MEGAHUNT

RM2x0016

# 参考手册

MH2435

基于 32 位 RISC 核心的安全处理器

## 引言

本文档面向应用程序开发人员，完整介绍了如何使用 MH2435 系列微控制器的存储器和外设。包含各个功能模块的内部结构、所有可能的功能描述、各种工作模式的使用和寄存器配置等详细信息。

该系列微控制器包含不同存储容量、封装和外设配置的型号，文档中统一简称为"MH2435"（特殊说明除外）。

如需产品的基本配置（如内置 Flash 和 RAM 的容量、外设模块的种类和数量等）、管脚的数量和分配、电气特性、封装信息和定购代码等，请参阅数据手册。

## 相关文档

- 可从兆讯恒达获取：
  - MH2435 数据手册
  - MH2435 应用手册

# 目录

1. 文档约定 ..... 46
1.1. 寄存器相关缩写词列表 ..... 46
1.2. 术语、定义及缩略语 ..... 46
2. 存储器和系统架构 ..... 48
2.1. 系统架构 ..... 48
2.2. 存储器映射 ..... 48
2.2.1. 嵌入式 SRAM ..... 50
2.2.2. 位段 ..... 50
3. 电源控制器(PWR) ..... 51
3.1. 电源 ..... 51
3.2. 电源监控器 ..... 51
3.2.1. 上电复位(POR)和掉电复位(PDR) ..... 51
3.2.2. 可编程电压检测器(PVD) ..... 51
3.3. 低功耗模式 ..... 52
3.3.1. 降低系统时钟 ..... 52
3.3.2. 外设时钟门控 ..... 52
3.3.3. 睡眠模式 ..... 53
3.3.4. 停止模式 ..... 53
3.3.5. 待机模式 ..... 54
3.3.6. 掉电模式 ..... 55
3.4. PWR 寄存器 ..... 55
3.4.1. PWR 控制寄存器(PWR_CR) ..... 55
3.4.2. PWR 状态寄存器(PWR_CSR) ..... 57
4. 嵌套向量中断控制器(NVIC) ..... 60
4.1. NVIC 特性 ..... 60
4.2. 中断和异常向量 ..... 60
5. 外部中断和事件控制器(EXTI) ..... 63
5.1. 外部中断/事件控制器 (EXTI) ..... 63

5.1.1. EXTI 主要特性 ..... 63
5.1.2. EXTI 框图 ..... 63
5.1.3. 唤醒事件管理 ..... 64
5.1.4. 功能描述 ..... 64
5.1.5. 硬件中断选择 ..... 64
5.1.6. 硬件事件选择 ..... 64
5.1.7. 软件中断/事件的选择 ..... 64
5.1.8. 外部中断/事件线映射 ..... 64
5.2. EXTI 寄存器 ..... 65
5.2.1. 中断屏蔽寄存器 (EXTI_IMR) ..... 65
5.2.2. 事件屏蔽寄存器 (EXTI_EMR) ..... 66
5.2.3. 上升沿触发选择寄存器 (EXTI_RTSR) ..... 66
5.2.4. 下降沿触发选择寄存器 (EXTI_FTSR) ..... 67
5.2.5. 软件中断事件寄存器 (EXTI_SWIER) ..... 67
5.2.6. 挂起寄存器 (EXTI_PR) ..... 68
6. 复位和时钟控制(RCC) ..... 69
6.1. 复位 ..... 69
6.1.1. 系统复位 ..... 69
6.1.2. 电源复位 ..... 69
6.1.3. 备份域复位 ..... 69
6.2. 时钟 ..... 69
6.2.1. HSI 时钟 ..... 70
6.2.2. PLL 配置 ..... 71
6.2.3. 时钟安全系统(CSS) ..... 71
6.2.4. 时钟输出功能 ..... 71
6.3. RCC 寄存器 ..... 72
6.3.1. RCC 时钟控制寄存器 (CR) ..... 72
6.3.2. RCC PLL 配置寄存器 (PLLCFGR) ..... 74
6.3.3. RCC 时钟配置寄存器 (CFGR) ..... 76
6.3.4. RCC 时钟中断寄存器 (CIR) ..... 78
6.3.5. RCC AHB1 外设复位寄存器 (AHB1RSTR) ..... 80

6.3.6. RCC AHB2 外设复位寄存器 (AHB2RSTR) ..... 82
6.3.7. RCC AHB3 复位寄存器 (AHB3RSTR) ..... 82
6.3.8. RCC AHB4 复位寄存器 (AHB4RSTR) ..... 83
6.3.9. RCC APB1 复位寄存器 (APB1RSTR) ..... 83
6.3.10. RCC APB2 复位寄存器 (APB2RSTR) ..... 86
6.3.11. RCC AHB1 外设时钟使能寄存器 (AHB1ENR) ..... 88
6.3.12. RCC AHB2 外设时钟使能寄存器 (AHB2ENR) ..... 90
6.3.13. RCC AHB3 外设时钟使能寄存器 (AHB3ENR) ..... 91
6.3.14. RCC AHB4 外设时钟使能寄存器 (AHB4ENR) ..... 91
6.3.15. RCC APB1 外设时钟使能寄存器 (APB1ENR) ..... 92
6.3.16. RCC APB2 外设时钟使能寄存器 (APB2ENR) ..... 94
6.3.17. RCC AHB1 低功耗时钟门控寄存器 (AHB1LPENR) ..... 96
6.3.18. RCC AHB2 低功耗时钟门控寄存器 (AHB2LPENR) ..... 98
6.3.19. RCC AHB3 低功耗时钟门控寄存器 (AHB3LPENR) ..... 99
6.3.20. RCC AHB4 低功耗时钟门控寄存器 (AHB4LPENR) ..... 99
6.3.21. RCC APB1 低功耗时钟门控寄存器 (APB1LPENR) ..... 100
6.3.22. RCC APB2 低功耗时钟门控寄存器 (APB2LPENR) ..... 103
6.3.23. RCC 备份域控制寄存器 (BDCR) ..... 104
6.3.24. RCC 时钟控制和状态寄存器 (CSR) ..... 105
6.3.25. RCC 扩频时钟生成寄存器 (SSCGR) ..... 107
6.3.26. RCC PLL2 配置寄存器 (PLL2CFGR) ..... 108
6.3.27. RCC 专用时钟配置寄存器 (DCKCFGR) ..... 109
6.3.28. RCC PLL3 EXT 时钟配置寄存器 (PLL3EXTCFGR) ..... 110
6.3.29. RCC PLL3 参数加载寄存器 (PLL3LOADR) ..... 110
6.3.30. RCC 子系统时钟分频寄存器 (SPRER) ..... 110
7. 通用 I/O(GPIO) ..... 112
7.1. GPIO 简介 ..... 112
7.2. GPIO 主要特性 ..... 112
7.3. GPIO 功能描述 ..... 113
7.3.1. 通用 I/O(GPIO) ..... 113
7.3.2. I/O 引脚复用器和映射 ..... 114

7.3.3. I/O 端口控制寄存器 ..... 115
7.3.4. I/O 端口数据寄存器 ..... 115
7.3.5. I/O 数据位操作 ..... 115
7.3.6. GPIO 锁定机制 ..... 115
7.3.7. I/O 复用功能输入/输出 ..... 115
7.3.8. 外部中断线/唤醒线 ..... 116
7.3.9. 输入配置 ..... 116
7.3.10. 输出配置 ..... 116
7.3.11. 复用功能配置 ..... 117
7.3.12. 模拟配置 ..... 118
7.3.13. 将 HSE 振荡器引脚用作 GPIO PH0/PH1 端口引脚 ..... 119
7.4. GPIO 寄存器 ..... 119
7.4.1. GPIO 端口模式寄存器(GPIOx_MODER)(x = A...I) ..... 119
7.4.2. GPIO 端口输出类型寄存器 (GPIOx_OTYPER) (x = A..I) ..... 119
7.4.3. GPIO 端口输出速度寄存器 (GPIOx_OSPEEDR) (x = A..I) ..... 120
7.4.4. GPIO 端口上拉/下拉寄存器 (GPIOx_PUPDR) (x = A..I) ..... 120
7.4.5. GPIO 端口输入数据寄存器 (GPIOx_IDR) (x = A..I) ..... 121
7.4.6. GPIO 端口输出数据寄存器 (GPIOx_ODR) (x = A..I) ..... 121
7.4.7. GPIO 端口置位/复位寄存器 (GPIOx_BSRR) (x = A..I) ..... 121
7.4.8. GPIO 端口配置锁定寄存器 (GPIOx_LCKR) (x = A..I) ..... 122
7.4.9. GPIO 复用功能低位寄存器 (GPIOx_AFRL) (x = A..I) ..... 123
7.4.10. GPIO 复用功能高位寄存器 (GPIOx_AFRH) (x = A..I) ..... 123
7.4.11. GPIO 驱动能力配置寄存器 (GPIOx_DSR) (x = A..I) ..... 124
8. 系统配置控制器(SYSCFG) ..... 125
8.1. SYSCFG 寄存器 ..... 125
8.1.1. SYSCFG 外设模式配置寄存器(SYSCFG_PMC) ..... 125
8.1.2. SYSCFG 外部中断配置寄存器 1(SYSCFG_EXTICR1) ..... 125
8.1.3. SYSCFG 外部中断配置寄存器 2(SYSCFG_EXTICR2) ..... 127
8.1.4. SYSCFG 外部中断配置寄存器 3(SYSCFG_EXTICR3) ..... 128
8.1.5. SYSCFG 外部中断配置寄存器 4(SYSCFG_EXTICR4) ..... 129
8.1.6. SYSCFG 外设模式配置寄存器 1(SYSCFG_PMC1) ..... 130

9. 直接存储器访问控制器(DMA)..... 132
9.1. DMA 简介..... 132
9.2. DMA 主要特性..... 132
9.3. DMA 功能说明..... 133
9.3.1. 一般说明..... 133
9.3.2. DMA 事务..... 134
9.3.3. 通道选择..... 134
9.3.4. 仲裁器..... 135
9.3.5. DMA 数据流..... 135
9.3.6. 源、目标和传输模式..... 136
9.3.7. 指针递增..... 139
9.3.8. 循环模式..... 139
9.3.9. 双缓冲区模式..... 140
9.3.10. 可编程数据宽度、封装/解封、字节序..... 140
9.3.11. 单次传输和突发传输..... 142
9.3.12. FIFO..... 142
9.3.13. DMA 传输完成..... 145
9.3.14. DMA 传输暂停..... 145
9.3.15. 流控制器..... 146
9.3.16. 可能的 DMA 配置汇总..... 146
9.3.17. 流配置过程..... 146
9.3.18. 错误管理..... 147
9.3.19. 链模式..... 148
9.3.20. DMA 中断..... 148
9.4. DMA 寄存器..... 149
9.4.1. DMA 低中断状态寄存器(DMA_LISR)..... 149
9.4.2. DMA 高中断状态寄存器(DMA_HISR)..... 150
9.4.3. DMA 低中断标志清除寄存器(DMA_LIFCR)..... 151
9.4.4. DMA 低中断标志清除寄存器(DMA_HIFCR)..... 151
9.4.5. DMA 数据流 x 配置寄存器(DMA_SxCR)(x = 0..7)..... 152
9.4.6. DMA 数据流 x 数据项数寄存器 (DMA_SxNDTR) (x = 0..7)..... 155

9.4.7. DMA 数据流 x 外设地址寄存器 (DMA_SxPAR) (x = 0..7) ..... 156
9.4.8. DMA 数据流 x 存储器 0 地址寄存器 (DMA_SxM0AR) (x = 0..7) ..... 156
9.4.9. DMA 数据流 x 存储器 1 地址寄存器 (DMA_SxM1AR) (x = 0..7) ..... 157
9.4.10. DMA 数据流 x FIFO 控制寄存器 (DMA_SxFCR) (x = 0..7) ..... 157
9.4.11. DMA 数据流链块地址寄存器(DMA_SxLR) (x = 0..7) ..... 158
10. 循环冗余校验计算单元 (CRC) ..... 159
10.1. CRC 简介 ..... 159
10.2. CRC 主要特性 ..... 159
10.3. CRC 功能描述 ..... 159
10.4. CRC 寄存器 ..... 159
10.4.1. CRC 数据寄存器(CRC_DR) ..... 159
10.4.2. CRC 独立数据寄存器(CRC_IDR) ..... 160
10.4.3. CRC 控制寄存器(CRC_CR) ..... 160
10.4.4. CRC 控制状态寄存器(CRC_CSR) ..... 161
10.4.5. CRC 初始值寄存器(CRC_INI) ..... 161
10.4.6. CRC 结果异或寄存器(CRC_XOR) ..... 162
11. Quad-SPI 控制器(QSPI) ..... 163
11.1. Quad-SPI 简介 ..... 163
11.2. Quad-SPI 主要特性 ..... 163
11.3. Quad-SPI 功能说明 ..... 163
11.3.1. Quad-SPI 引脚与时钟 ..... 163
11.3.2. Quad-SPI 时钟相位与极性 ..... 164
11.3.3. Quad-SPI 传输模式 ..... 164
11.3.4. Quad-SPI 数据大小端转换 ..... 165
11.3.5. Quad-SPI 接收数据帧长度 ..... 165
11.3.6. Quad-SPI 发送字节长度 ..... 165
11.3.7. Quad-SPI DMA 控制器接口 ..... 166
11.3.8. Quad-SPI 状态标志 ..... 166
11.3.9. Quad-SPI 中断 ..... 167
11.3.10. Quad-SPI 配置 ..... 167
11.4. Quad-SPI 寄存器 ..... 169

11.4.1. Quad-SPI 控制寄存器 0 (CTRL0) ..... 169
11.4.2. Quad-SPI 控制寄存器 1 (CTRL1) ..... 170
11.4.3. Quad-SPI 串行传输使能寄存器 (SSIENR) ..... 171
11.4.4. Quad-SPI 从选择寄存器 (SER) ..... 171
11.4.5. Quad-SPI 速率配置寄存器 (BAUDR) ..... 172
11.4.6. Quad-SPI 发送 FIFO 阈值寄存器 (TXFTLR) ..... 172
11.4.7. Quad-SPI 接收 FIFO 阈值寄存器 (RXFTLR) ..... 172
11.4.8. Quad-SPI 发送 FIFO 数据数量寄存器 (TXFLR) ..... 173
11.4.9. Quad-SPI 接收 FIFO 数据数量寄存器 (RXFLR) ..... 173
11.4.10. Quad-SPI 状态寄存器 (SR) ..... 174
11.4.11. Quad-SPI 中断屏蔽寄存器 (IMR) ..... 175
11.4.12. Quad-SPI 中断状态寄存器 (ISR) ..... 175
11.4.13. Quad-SPI 源中断状态寄存器 (RISR) ..... 176
11.4.14. Quad-SPI 发送上溢中断清除寄存器 (TXOICR) ..... 177
11.4.15. Quad-SPI 接收上溢中断清除寄存器 (RXOICR) ..... 177
11.4.16. Quad-SPI 接收下溢中断清除寄存器 (RXUICR) ..... 177
11.4.17. Quad-SPI 多主机冲突中断清除寄存器 (MSTICR) ..... 178
11.4.18. Quad-SPI 中断清除寄存器 (ICR) ..... 178
11.4.19. Quad-SPI DMA 控制寄存器 (DMACR) ..... 179
11.4.20. Quad-SPI DMA 发送阈值寄存器 (DMATDLR) ..... 179
11.4.21. Quad-SPI DMA 接收阈值寄存器 (DMARDLR) ..... 179
11.4.22. Quad-SPI 数据寄存器 (DR) ..... 180
11.4.23. Quad-SPI 接收采样延迟寄存器 (RX_SAMPLY_DLY) ..... 180
11.4.24. Quad-SPI 控制寄存器 (SPI_CTRL2) ..... 181
11.4.25. Quad-SPI 发送数据字节长度寄存器 (BYTE_NUM) ..... 182
12. 模数转换器(ADC) ..... 183
12.1. ADC 简介 ..... 183
12.2. ADC 主要特性 ..... 183
12.3. ADC 功能说明 ..... 183
12.3.1. ADC 开关控制 ..... 185
12.3.2. ADC 时钟 ..... 185

12.3.3. ADC 通道选择 ..... 185
12.3.4. 单次转换模式 ..... 185
12.3.5. 连续转换模式 ..... 186
12.3.6. 时序图 ..... 186
12.3.7. 模拟看门狗 ..... 187
12.3.8. 扫描模式 ..... 187
12.3.9. 注入通道管理 ..... 188
12.3.10. 不连续采样模式 ..... 189
12.4. 数据对齐 ..... 189
12.5. 可独立设置各通道采样时间 ..... 190
12.6. 外部触发转换和触发极性 ..... 191
12.7. 快速转换模式 ..... 192
12.8. 数据管理 ..... 192
12.8.1. 使用 DMA ..... 192
12.8.2. 在不使用 DMA 的情况下管理转换序列 ..... 193
12.8.3. 在不使用 DMA 和溢出检测的情况下进行转换 ..... 193
12.9. 多重 ADC 模式 ..... 193
12.9.1. 注入同时模式 ..... 196
12.9.2. 规则同时模式 ..... 197
12.9.3. 交替模式 ..... 198
12.9.4. 交替触发模式 ..... 200
12.9.5. 混合型规则/注入同时模式 ..... 202
12.9.6. 规则同时 + 交替触发组合模式 ..... 202
12.10. 温度传感器和内部参考电压 ..... 203
12.11. 锂电池电压检测 ..... 204
12.12. ADC 中断 ..... 204
12.13. ADC 寄存器 ..... 204
12.13.1. ADC 状态寄存器(ADC_SR) ..... 204
12.13.2. ADC 控制寄存器 1 (ADC_CR1) ..... 205
12.13.3. ADC 控制寄存器 2 (ADC_CR2) ..... 207
12.13.4. ADC 采样时间寄存器 1 (ADC_SMPR1) ..... 209

12.13.5. ADC 采样时间寄存器 2 (ADC_SMPR2)..... 210
12.13.6. ADC 注入通道数据偏移寄存器 X (ADC_JOFRx)(x=1..4)..... 211
12.13.7. ADC 看门狗高阈值寄存器 (ADC_HTR)..... 211
12.13.8. ADC 看门狗低阈值寄存器 (ADC_LTR)..... 211
12.13.9. ADC 规则序列寄存器 1 (ADC_SQR1)..... 212
12.13.10. ADC 规则序列寄存器 2 (ADC_SQR2)..... 212
12.13.11. ADC 规则序列寄存器 3 (ADC_SQR3)..... 213
12.13.12. ADC 注入序列寄存器 (ADC_JSQR)..... 213
12.13.13. ADC 注入数据寄存器 x (ADC_JDRx) (x= 1..4)..... 214
12.13.14. ADC 规则数据寄存器 (ADC_DR)..... 215
12.13.15. ADC 通用状态寄存器 (ADC_CSR)..... 215
12.13.16. ADC 通用控制寄存器 (ADC_CCR)..... 216
12.13.17. 适用于双重和三重模式的 ADC 通用规则数据寄存器（ADC_CDR)..... 218
13. 数模转换器(DAC)..... 219
13.1. DAC 简介..... 219
13.2. DAC 主要特性..... 219
13.3. DAC 功能说明..... 220
13.3.1. DAC 通道使能..... 220
13.3.2. DAC 输出缓冲器使能..... 220
13.3.3. DAC 数据格式..... 220
13.3.4. DAC 转换..... 221
13.3.5. DAC 输出电压..... 222
13.3.6. DAC 触发选择..... 222
13.3.7. DMA 请求..... 223
13.3.8. 生成噪声..... 223
13.3.9. 生成三角波..... 224
13.4. DAC 双通道转换..... 225
13.4.1. 独立触发（不产生波形）..... 225
13.4.2. 独立触发（生成单个 LFSR）..... 225
13.4.3. 独立触发（生成不同 LFSR）..... 226
13.4.4. 独立触发（生成单个三角波）..... 226

13.4.5. 独立触发（生成不同三角波）.....226
13.4.6. 同步软件启动.....227
13.4.7. 同步触发（不产生波形）.....227
13.4.8. 同步触发（生成单个 LFSR）.....227
13.4.9. 同步触发（生成不同 LFSR）.....227
13.4.10. 同步触发（生成单个三角波）.....228
13.4.11. 同步触发（生成不同三角波）.....228
13.5. DAC 寄存器.....228
13.5.1. DAC 控制寄存器（DAC_CR）.....228
13.5.2. DAC 软件触发寄存器 (DAC_SWTRIGR).....231
13.5.3. DAC 1 通道 12 位右对齐数据保持寄存器 (DAC_DHR12R1).....232
13.5.4. DAC 1 通道 12 位左对齐数据保持寄存器 (DAC_DHR12L1).....232
13.5.5. DAC 1 通道 8 位右对齐数据保持寄存器 (DAC_DHR8R1).....232
13.5.6. DAC 2 通道 12 位右对齐数据保持寄存器 (DAC_DHR12R2).....233
13.5.7. DAC 2 通道 12 位左对齐数据保持寄存器 (DAC_DHR12L2).....233
13.5.8. DAC 2 通道 8 位右对齐数据保持寄存器 (DAC_DHR8R2).....233
13.5.9. 双 DAC 12 位右对齐数据保持寄存器 (DAC_DHR12RD).....234
13.5.10. 双 DAC 12 位左对齐数据保持寄存器 (DAC_DHR12LD).....234
13.5.11. 双 DAC 8 位右对齐数据保持寄存器 (DAC_DHR8RD).....235
13.5.12. DAC 1 通道数据输出寄存器 (DAC_DOR1).....235
13.5.13. DAC 2 通道数据输出寄存器 (DAC_DOR2).....235
13.5.14. DAC 状态寄存器 (DAC_SR).....236
14. 高级控制定时器(TIM1/TIM8).....237
14.1. TIM1/TIM8 简介.....237
14.2. TIM1/TIM8 主要特性.....237
14.3. TIM1/TIM8 功能说明.....238
14.3.1. 时基单元.....238
14.3.2. 计数器模式.....240
14.3.3. 重复计数器.....250
14.3.4. 时钟选择.....251
14.3.5. 捕获/比较通道.....254

14.3.6. 输入捕获模式 256
14.3.7. PWM 输入模式 257
14.3.8. 强制输出模式 258
14.3.9. 输出比较模式 258
14.3.10. PWM 模式 259
14.3.11. 互补输出和死区插入 262
14.3.12. 使用断路功能 263
14.3.13. 发生外部事件时清除 OCxREF 信号 265
14.3.14. 生成 6 步 PWM 266
14.3.15. 单脉冲模式 267
14.3.16. 编码器接口模式 269
14.3.17. 定时器输入异或功能 270
14.3.18. 连接霍尔传感器 271
14.3.19. TIMx 与外部触发同步 272
14.3.20. 定时器同步 275
14.3.21. 调试模式 275
14.4. TIM1 和 TIM8 寄存器 275
14.4.1. TIM1 和 TIM8 控制寄存器 1(TIMx_CR1) 275
14.4.2. TIM1 和 TIM8 控制寄存器 2(TIMx_CR2) 277
14.4.3. TIM1 和 TIM8 从模式控制寄存器(TIMx_SMCR) 278
14.4.4. TIM1 和 TIM8 DMA/中断使能寄存器(TIMx_DIER) 280
14.4.5. TIM1 和 TIM8 状态寄存器(TIMx_SR) 282
14.4.6. TIM1 和 TIM8 事件生成寄存器(TIMx_EGR) 283
14.4.7. TIM1 和 TIM8 捕获/比较模式寄存器 1(TIMx_CCMR1) 284
14.4.8. TIM1 和 TIM8 捕获/比较模式寄存器 2(TIMx_CCMR2) 287
14.4.9. TIM1 和 TIM8 捕获/比较使能寄存器(TIMx_CCER) 289
14.4.10. TIM1 和 TIM8 计数器(TIMx_CNT) 291
14.4.11. TIM1 和 TIM8 预分频器(TIMx_PSC) 292
14.4.12. TIM1 和 TIM8 自动重载寄存器(TIMx_ARR) 292
14.4.13. TIM1 和 TIM8 重复计数器寄存器(TIMx_RCR) 292
14.4.14. TIM1 和 TIM8 捕获/比较寄存器 1 (TIMx_CCR1) 293

14.4.15. TIM1 和 TIM8 捕获/比较比较器 2 (TIMx_CCR2)..... 293
14.4.16. TIM1 和 TIM8 捕获/比较比较器 3 (TIMx_CCR3)..... 294
14.4.17. TIM1 和 TIM8 捕获/比较比较器 4 (TIMx_CCR4)..... 294
14.4.18. TIM1 和 TIM8 断路和死区寄存器(TIMx_BDTR)..... 295
14.4.19. TIM1 和 TIM8 DMA 控制寄存器(TIMx_DCR)..... 296
14.4.20. TIM1 和 TIM8 全传输 DMA 地址(TIMx_DMAR)..... 297
15. 通用定时器(TIM2/TIM3/TIM4/TIM5)..... 299
15.1. TIM2 到 TIM5 简介..... 299
15.2. TIM2 到 TIM5 主要特性..... 299
15.3. TIM2 到 TIM5 功能说明..... 300
15.3.1. 时基单元..... 300
15.3.2. 计数器模式..... 302
15.3.3. 时钟选择..... 310
15.3.4. 捕获/比较通道..... 313
15.3.5. 输入捕获模式..... 315
15.3.6. PWM 输入模式..... 316
15.3.7. 强制输出模式..... 317
15.3.8. 输出比较模式..... 317
15.3.9. PWM 模式..... 318
15.3.10. 单脉冲模式..... 320
15.3.11. 发生外部事件时清除 OCxREF 信号..... 322
15.3.12. 编码器接口模式..... 322
15.3.13. 定时器输入异或功能..... 324
15.3.14. 定时器与外部触发同步..... 324
15.3.15. 定时器同步..... 327
15.3.16. 调试模式..... 331
15.4. TIM2 到 TIM5 寄存器..... 332
15.4.1. TIMx 控制寄存器 1(TIMx_CR1)..... 332
15.4.2. TIMx 控制寄存器 2(TIMx_CR2)..... 333
15.4.3. TIMx 从模式控制寄存器(TIMx_SMCR)..... 334
15.4.4. TIMx DMA/中断使能寄存器(TIMx_DIE)..... 336

15.4.5. TIMx 状态寄存器(TIMx_SR) 337
15.4.6. TIMx 事件生成寄存器(TIMx_EGR) 338
15.4.7. TIMx 捕获/比较模式寄存器 1(TIMx_CCMR1) 339
15.4.8. TIMx 捕获/比较模式寄存器 2(TIMx_CCMR2) 342
15.4.9. TIMx 捕获/比较使能寄存器(TIMx_CCER) 343
15.4.10. TIMx 计数器(TIMx_CNT) 345
15.4.11. TIMx 预分频器(TIMx_PSC) 345
15.4.12. TIMx 自动重载寄存器(TIMx_ARR) 346
15.4.13. TIMx 捕获/比较寄存器 1 (TIMx_CCR1) 346
15.4.14. TIMx 捕获/比较比较器 2 (TIMx_CCR2) 346
15.4.15. TIMx 捕获/比较比较器 3 (TIMx_CCR3) 347
15.4.16. TIMx 捕获/比较比较器 4 (TIMx_CCR4) 348
15.4.17. TIMx DMA 控制寄存器(TIMx_DCR) 348
15.4.18. TIMx 全传输 DMA 地址(TIMx_DMAR) 349
15.4.19. TIM2 选项寄存器 1(TIM2_OR) 350
15.4.20. TIM5 选项寄存器 1(TIM5_OR) 350
16. 通用定时器(TIM9~TIM14) 351
16.1. TIM9 到 TIM14 简介 351
16.2. TIM9 到 TIM14 主要特性 351
16.2.1. TIM9/TIM12 主要特性 351
16.2.2. TIM10/TIM11 和 TIM13/TIM14 主要特性 352
16.3. TIM9 到 TIM14 功能说明 353
16.3.1. 时基单元 353
16.3.2. 计数器模式 355
16.3.3. 时钟选择 359
16.3.4. 捕获/比较通道 361
16.3.5. 输入捕获模式 362
16.3.6. PWM 输入模式(仅适用于 TIM9/12) 363
16.3.7. 强制输出模式 363
16.3.8. 输出比较模式 364
16.3.9. PWM 模式 365

16.3.10. 单脉冲模式 ..... 366
16.3.11. TIM9/12 外部触发同步 ..... 368
16.3.12. 定时器同步 ..... 370
16.3.13. 调试模式 ..... 370
16.4. TIM9 和 TIM12 寄存器 ..... 370
16.4.1. TIM9/12 控制寄存器 1 (TIMx_CR1) ..... 370
16.4.2. TIM9/12 控制寄存器 2 (TIMx_CR2) ..... 371
16.4.3. TIM9/12 从模式控制寄存器(TIMx_SMCR) ..... 372
16.4.4. TIM9/12 中断使能寄存器(TIMx_DIER) ..... 373
16.4.5. TIM9/12 状态寄存器(TIMx_SR) ..... 374
16.4.6. TIM9/12 事件生成寄存器(TIMx_EGR) ..... 375
16.4.7. TIM9/12 捕获/比较模式寄存器(TIMx_CCMR1) ..... 375
16.4.8. TIM9/12 捕获/比较使能寄存器(TIMx_CCER) ..... 378
16.4.9. TIM9/12 计数器(TIMx_CNT) ..... 379
16.4.10. TIM9/12 预分频器 (TIMx_PSC) ..... 380
16.4.11. TIM9/12 自动重载寄存器 (TIMx_ARR) ..... 380
16.4.12. TIM9/12 捕获/比较寄存器 1 (TIMx_CCR1) ..... 380
16.4.13. TIM9/12 捕获/比较比较器 2 (TIMx_CCR2) ..... 381
16.5. TIM10/11/13/14 寄存器 ..... 381
16.5.1. TIM10/11/13/14 控制寄存器 1 (TIMx_CR1) ..... 381
16.5.2. TIM10/11/13/14 状态寄存器 (TIMx_SR) ..... 382
16.5.3. TIM10/11/13/14 事件生成寄存器(TIMx_EGR) ..... 383
16.5.4. TIM10/11/13/14 捕获/比较模式寄存器 1(TIMx_CCMR1) ..... 383
16.5.5. TIM10/11/13/14 捕获/比较使能寄存器 (TIMx_CCER) ..... 386
16.5.6. TIM10/11/13/14 计数器(TIMx_CNT) ..... 386
16.5.7. TIM10/11/13/14 预分频器(TIMx_PSC) ..... 387
16.5.8. TIM10/11/13/14 自动重载寄存器(TIMx_ARR) ..... 387
16.5.9. TIM10/11/13/14 捕获/比较寄存器 1(TIMx_CCR1) ..... 387
16.5.10. TIM10/11/13/14 选项寄存器 1(TIM11_OR) ..... 388
17. 基本定时器(TIM6/TIM7) ..... 389
17.1. TIM6 和 TIM7 简介 ..... 389

17.2. TIM6 和 TIM7 主要特性 ..... 389
17.3. TIM6 和 TIM7 功能说明 ..... 390
17.3.1. 时基单元 ..... 390
17.3.2. 计数模式 ..... 392
17.3.3. 时钟源 ..... 394
17.3.4. 调试模式 ..... 395
17.4. TIM6 和 TIM7 寄存器 ..... 395
17.4.1. TIM6 和 TIM7 控制寄存器 1 (TIMx_CR1) ..... 395
17.4.2. TIM6 和 TIM7 控制寄存器 2 (TIMx_CR2) ..... 396
17.4.3. TIM6 和 TIM7 DMA/中断使能寄存器(TIMx_DIER) ..... 397
17.4.4. TIM6 和 TIM7 状态寄存器(TIMx_SR) ..... 397
17.4.5. TIM6 和 TIM7 事件生成寄存器 (TIMx_EGR) ..... 397
17.4.6. TIM6 和 TIM7 计数器 (TIMx_CNT) ..... 398
17.4.7. TIM6 和 TIM7 预分频器 (TIMx_PSC) ..... 398
17.4.8. TIM6 和 TIM7 自动重载寄存器 (TIMx_ARR) ..... 398
18. 独立看门狗(IWDG) ..... 399
18.1. IWDG 简介 ..... 399
18.2. IWDG 主要特性 ..... 399
18.3. IWDG 功能说明 ..... 399
18.3.1. 硬件看门狗 ..... 399
18.3.2. 寄存器访问保护 ..... 399
18.3.3. 调试模式 ..... 399
18.4. IWDG 寄存器 ..... 400
18.4.1. 关键字寄存器 (IWDG_KR) ..... 400
18.4.2. 预分频器寄存器 (IWDG_PR) ..... 401
18.4.3. 重载寄存器 (IWDG_RLR) ..... 401
18.4.4. 状态寄存器 (IWDG_SR) ..... 402
19. 窗口看门狗(WWDG) ..... 403
19.1. WWDG 简介 ..... 403
19.2. WWDG 主要特性 ..... 403
19.3. WWDG 功能说明 ..... 403

19.4. 如何设置看门狗超时 ..... 404
19.5. 调试模式 ..... 405
19.6. WWDG 寄存器 ..... 405
19.6.1. 控制寄存器 (WWDG_CR) ..... 405
19.6.2. 配置寄存器 (WWDG_CFR) ..... 405
19.6.3. 状态寄存器 (WWDG_SR) ..... 406
20. 内部集成电路接口(I2C) ..... 407
20.1. I2C 简介 ..... 407
20.2. I2C 主要特性 ..... 407
20.3. I2C 功能说明 ..... 408
20.3.1. 模式选择 ..... 408
20.3.2. I2C 从模式 ..... 410
20.3.3. I2C 主模式 ..... 412
20.3.4. 错误条件 ..... 416
20.3.5. 可编程噪声滤波器 ..... 417
20.3.6. SDA/SCL 线控制 ..... 417
20.3.7. SMBus ..... 417
20.3.8. DMA 请求 ..... 419
20.3.9. 数据包错误校验 ..... 421
20.4. I2C 中断 ..... 421
20.5. I2C 调试模式 ..... 422
20.6. I2C 寄存器 ..... 422
20.6.1. I2C 控制寄存器 1 (CR1) ..... 422
20.6.2. I2C 控制寄存器 2 (CR2) ..... 424
20.6.3. I2C 自有地址寄存器 1 (OAR1) ..... 425
20.6.4. I2C 自有地址寄存器 2 (OAR2) ..... 426
20.6.5. I2C 数据寄存器 (DR) ..... 426
20.6.6. I2C 状态寄存器 1 (SR1) ..... 427
20.6.7. I2C 状态寄存器 2 (SR2) ..... 430
20.6.8. I2C 时钟控制寄存器 (CCR) ..... 431
20.6.9. I2C TRISE 寄存器 (TRISE) ..... 432

20.6.10. I2C FLTR 寄存器 (FLTR) ..... 432
21. 通用同步异步收发器(USART) ..... 434
21.1. USART 简介 ..... 434
21.2. USART 主要特性 ..... 434
21.3. USART 功能说明 ..... 435
21.3.1. USART 字符说明 ..... 436
21.3.2. 发送器 ..... 437
21.3.3. 接收器 ..... 439
21.3.4. 小数波特率生成 ..... 444
21.3.5. USART 接收器对时钟偏差的容差 ..... 448
21.3.6. 多处理器通信 ..... 449
21.3.7. 奇偶校验控制 ..... 450
21.3.8. LIN(局域互连网络)模式 ..... 451
21.3.9. USART 同步模式 ..... 453
21.3.10. 单线半双工通信 ..... 455
21.3.11. 智能卡 ..... 456
21.3.12. IrDA SIR ENDEC 模块 ..... 457
21.3.13. 使用 DMA 进行连续通信 ..... 459
21.3.14. 硬件流控制 ..... 461
21.4. USART 中断 ..... 462
21.5. USART 模式配置 ..... 463
21.6. USART 寄存器 ..... 463
21.6.1. 状态寄存器 (USART_SR) ..... 463
21.6.2. 数据寄存器 (USART_DR) ..... 465
21.6.3. 波特率寄存器 (USART_BRR) ..... 466
21.6.4. 控制寄存器 1(USART_CR1) ..... 466
21.6.5. 控制寄存器 2 (USART_CR2) ..... 468
21.6.6. 控制寄存器 3 (USART_CR3) ..... 470
21.6.7. 保护时间和预分频器寄存器 (USART_GTPR) ..... 471
21.6.8. 接收 FIFO 控制寄存器 (USART_FCR) ..... 472
22. 串行外设接口 (SPI) ..... 473

22.1. SPI 简介 ..... 473
22.2. SPI 和 I2S 主要特性 ..... 473
22.2.1. SPI 特性 ..... 473
22.2.2. I2S 特性 ..... 474
22.3. SPI 功能说明 ..... 474
22.3.1. 一般说明 ..... 474
22.3.2. 把 SPI 配置成从器件 ..... 476
22.3.3. 把 SPI 配置成主器件 ..... 478
22.3.4. 配置 SPI 进行半双工通信 ..... 480
22.3.5. 数据发送和接收过程 ..... 480
22.3.6. CRC 计算 ..... 486
22.3.7. 状态标志 ..... 487
22.3.8. 关闭 SPI ..... 488
22.3.9. 使用 DMA 进行 SPI 通信 ..... 489
22.3.10. 错误标志 ..... 491
22.3.11. SPI 中断 ..... 492
22.4. I2S 功能说明 ..... 493
22.4.1. 一般说明 ..... 493
22.4.2. I2S 全双工 ..... 494
22.4.3. 支持的音频协议 ..... 494
22.4.4. 时钟发生器 ..... 501
22.4.5. I2S 主模式 ..... 502
22.4.6. I2S 从模式 ..... 503
22.4.7. 状态标志 ..... 504
22.4.8. 错误标志 ..... 505
22.4.9. I2S 中断 ..... 506
22.4.10. DMA 特性 ..... 506
22.5. SPI 和 I2S 寄存器 ..... 506
22.5.1. SPI 控制寄存器 1 (SPI_CR1)（不用于 I2S 模式） ..... 506
22.5.2. SPI 控制寄存器 2 (SPI_CR2) ..... 508
22.5.3. SPI 状态寄存器 (SPI_SR) ..... 509

22.5.4. SPI 数据寄存器 (SPI_DR) ..... 510
22.5.5. 多项式寄存器 (SPI_CRCPR)（不用于 I2S 模式） ..... 511
22.5.6. SPI RX CRC 寄存器 (SPI_RXCRCR)（不用于 I2S 模式） ..... 511
22.5.7. SPI TX CRC 寄存器 (SPI_TXCRCR)（不用于 I2S 模式） ..... 511
22.5.8. SPI_I2S 配置寄存器 (SPI_I2SCFGR) ..... 512
22.5.9. SPI_I2S 预分频器寄存器 (SPI_I2SPR) ..... 513
22.5.10. SPI_I2S 寄存器 3 (SPI_CR3) ..... 514
22.5.11. SPI_I2S 状态寄存器 2 (SPI_SR2) ..... 515
22.5.12. SPI_I2S FIFO 清除寄存器 (SPI_FIFO_CLR) ..... 515
23. SDIO 接口 (SDIO) ..... 517
23.1. SDIO 主要特性 ..... 517
23.2. SDIO 总线拓扑 ..... 517
23.3. SDIO 功能说明 ..... 519
23.3.1. SDIO 适配器 ..... 520
23.3.2. SDIO APB2 接口 ..... 528
23.4. 卡功能说明 ..... 529
23.4.1. 卡识别模式 ..... 529
23.4.2. 智能卡复位 ..... 529
23.4.3. 工作电压范围验证 ..... 529
23.4.4. 卡识别过程 ..... 530
23.4.5. 块写入 ..... 530
23.4.6. 块读取 ..... 531
23.4.7. 流访问、流写入和流读取（仅限多媒体卡） ..... 531
23.4.8. 擦除：组擦除和扇区擦除 ..... 532
23.4.9. 宽总线选择或取消选择 ..... 533
23.4.10. 保护管理 ..... 533
23.4.11. 卡状态寄存器 ..... 535
23.4.12. SD 状态寄存器 ..... 537
23.4.13. SD I/O 模式 ..... 541
23.4.14. 命令和响应 ..... 541
23.5. 响应格式 ..... 544

23.5.1. R1(正常响应命令)..... 544
23.5.2. R1b 544
23.5.3. R2(CID 和 CSD 寄存器)..... 545
23.5.4. R3(OCR 寄存器)..... 545
23.5.5. R4(快速 I/O)..... 545
23.5.6. R4b 545
23.5.7. R5(中断请求)..... 546
23.5.8. R6 546
23.6. SDIO I/O 卡专用操作..... 547
23.6.1. 通过触发 SDIO_D2 执行的 SDIO I/O 读取等待操作..... 547
23.6.2. 通过停止 SDIO_CK 执行的 SDIO 读取等待操作..... 547
23.6.3. SDIO 挂起/恢复操作..... 547
23.6.4. SDIO 中断..... 548
23.7. CE-ATA 专用操作..... 548
23.7.1. 命令完成信号禁止..... 548
23.7.2. 命令完成信号使能..... 548
23.7.3. CE-ATA 中断..... 548
23.7.4. 中止 CMD61..... 548
23.8. 硬件流控制..... 548
23.9. SDIO 寄存器..... 549
23.9.1. SDIO 电源控制寄存器(SDIO_POWER)..... 549
23.9.2. SDIO 时钟控制寄存器(SDIO_CLKCR)..... 549
23.9.3. SDIO 参数寄存器(SDIO_ARG)..... 550
23.9.4. SDIO 命令寄存器(SDIO_CMD)..... 550
23.9.5. SDIO 命令响应寄存器(SDIO_RESPCMD)..... 551
23.9.6. SDIO 响应 1..4 寄存器(SDIO_RESPx)..... 552
23.9.7. SDIO 数据定时寄存器(SDIO_DTIMER)..... 552
23.9.8. SDIO 数据长度寄存器(SDIO_DLEN)..... 553
23.9.9. SDIO 数据控制寄存器(SDIO_DCTRL)..... 553
23.9.10. SDIO 数据计数寄存器(SDIO_DCOUNT)..... 554
23.9.11. SDIO 状态寄存器(SDIO_STA)..... 555

23.9.12. SDIO 中断清零寄存器(SDIO_ICR) ..... 556
23.9.13. SDIO 屏蔽寄存器(SDIO_MASK) ..... 557
23.9.14. SDIO FIFO 计数寄存器(SDIO_FIFOCNT) ..... 560
23.9.15. SDIO 数据 FIFO 寄存器(SDIO_FIFO) ..... 560
24. 控制器区域网络(bxCAN) ..... 561
24.1. bxCAN 简介 ..... 561
24.2. bxCAN 主要特性 ..... 561
24.3. bxCAN 一般说明 ..... 561
24.3.1. CAN 2.0B 主动内核 ..... 562
24.3.2. 控制、状态和配置寄存器 ..... 562
24.3.3. 发送邮箱 ..... 562
24.3.4. 验收筛选器 ..... 562
24.4. bxCAN 工作模式 ..... 563
24.4.1. 初始化模式 ..... 563
24.4.2. 正常模式 ..... 564
24.4.3. 睡眠模式（低功耗） ..... 564
24.5. 测试模式 ..... 565
24.5.1. 静默模式 ..... 565
24.5.2. 环回模式 ..... 566
24.5.3. 环回与静默模式组合 ..... 566
24.6. 调试模式 ..... 566
24.7. bxCAN 功能说明 ..... 567
24.7.1. 发送处理 ..... 567
24.7.2. 时间触发通信模式 ..... 568
24.7.3. 接收处理 ..... 568
24.7.4. 标识符筛选 ..... 570
24.7.5. 消息存储 ..... 573
24.7.6. 错误管理 ..... 574
24.7.7. 位时序 ..... 575
24.8. bxCAN 中断 ..... 576
24.9. CAN 寄存器 ..... 578

24.9.1. 寄存器访问保护 578
24.9.2. CAN 控制和状态寄存器 578
24.9.3. CAN 邮箱寄存器 587
24.9.4. CAN 筛选器寄存器 592
25. 以太网(ETH) 596
25.1. 以太网简介 596
25.2. 以太网主要特性 596
25.2.1. MAC 内核特性 596
25.2.2. DMA 特性 597
25.2.3. PTP 特性 598
25.3. 以太网引脚 598
25.4. 以太网功能说明：SMI、MII 和 RMII 599
25.4.1. 站管理接口：SMI 599
25.4.2. 介质独立接口：MII 602
25.4.3. 精简介质独立接口：RMII 603
25.4.4. MII/RMII 选择 604
25.5. 以太网功能说明：MAC 802.3 605
25.5.1. MAC 802.3 帧格式 605
25.5.2. MAC 帧发送 608
25.5.3. MAC 帧接收 614
25.5.4. MAC 中断 618
25.5.5. MAC 过滤 618
25.5.6. MAC 回环模式 620
25.5.7. MAC 管理计数器：MMC 620
25.5.8. 电源管理：PMT 621
25.5.9. 精密时间协议(IEEE1588 PTP) 623
25.6. 以太网功能说明：DMA 控制器操作 628
25.6.1. 使用 DMA 进行传送的初始化 629
25.6.2. 主机总线突发访问 629
25.6.3. 主机数据缓冲区对齐 629
25.6.4. 缓冲区大小计算 630

25.6.5. DMA 仲裁器 ..... 630
25.6.6. 对 DMA 的错误响应 ..... 630
25.6.7. Tx DMA 配置 ..... 630
25.6.8. Rx DMA 配置 ..... 640
25.6.9. DMA 中断 ..... 650
25.7. 以太网中断 ..... 651
25.8. 以太网寄存器 ..... 652
25.8.1. MAC 寄存器 ..... 652
25.8.2. MMC 寄存器 ..... 669
25.8.3. IEEE 1588 时间戳寄存器 ..... 674
25.8.4. DMA 寄存器 ..... 681
26. 通用串行总线 (USB) ..... 695
26.1. USB 简介 ..... 695
26.1.1. 缩写描述 ..... 695
26.1.2. 模块简介 ..... 695
26.1.3. 参考文档 ..... 695
26.2. USB 主要特性 ..... 695
26.2.1. 通用特性 ..... 696
26.2.2. 主机模式特性 ..... 696
26.2.3. 设备模式特性 ..... 696
26.3. USB 功能说明 ..... 697
26.3.1. USB 结构框图 ..... 697
26.3.2. USB PHY ..... 697
26.4. OTG 双角色设备 ..... 697
26.4.1. ID 线检测 ..... 697
26.5. 设备模式下的 USB 功能说明 ..... 698
26.5.1. USB 设备状态 ..... 698
26.5.2. USB 设备端点 ..... 699
26.6. 主机模式下的 USB 功能说明 ..... 700
26.6.1. 支持 SRP 功能的 USB 主机 ..... 701
26.6.2. USB 主机状态 ..... 701

26.6.3. 主机通道 ..... 702
26.6.4. 主机调度器 ..... 703
26.7. USB 功耗模式 ..... 704
26.8. FIFO RAM 分配 ..... 705
26.8.1. 设备模式 ..... 705
26.8.2. 主机模式 ..... 705
26.9. USB 中断 ..... 706
26.10. USB 寄存器 ..... 706
26.10.1. USB 控制和状态寄存器 (GOTGCTL) ..... 707
26.10.2. USB 中断寄存器 (GOTGINT) ..... 709
26.10.3. USB 配置寄存器 (GAHBCFG) ..... 710
26.10.4. USB 配置寄存器 (GUSBCFG) ..... 711
26.10.5. USB 复位寄存器 (GRSTCTL) ..... 712
26.10.6. USB 模块中断寄存器 (GINTSTS) ..... 714
26.10.7. USB 中断屏蔽寄存器 (GINTMSK) ..... 717
26.10.8. USB 接收状态读取寄存器 (GRXSTSR) ..... 720
26.10.9. USB 接收状态出栈寄存器 (GRXSTSP) ..... 721
26.10.10. USB 接收 FIFO 大小寄存器 (GRXFSIZ) ..... 721
26.10.11. USB 非周期发送 FIFO 配置寄存器 (GNPTXFSIZ) ..... 722
26.10.12. USB 非周期性发送 FIFO/队列状态寄存器 (GNPTXSTS) ..... 722
26.10.13. USB 通用模块配置寄存器 (GCFG) ..... 722
26.10.14. USB 主机周期性发送 FIFO 大小寄存器 (HPTXFSIZ) ..... 723
26.10.15. USB 设备 IN 端点发送 FIFO 大小寄存器 (DIEPTXFn) ..... 724
26.10.16. USB 主机配置寄存器 (HCFG) ..... 724
26.10.17. USB 主机帧间隔寄存器 (HFIR) ..... 725
26.10.18. USB 主机帧编号/帧剩余时间寄存器 (HFNUM) ..... 725
26.10.19. USB Host 周期性发送 FIFO/队列状态寄存器 (HPTXSTS) ..... 726
26.10.20. USB 主机全体通道中断寄存器 (HAINT) ..... 727
26.10.21. USB 主机全体通道中断屏蔽寄存器 (HAINTMSK) ..... 727
26.10.22. USB 主机端口控制和状态寄存器 (HPRT) ..... 728
26.10.23. USB 主机通道特性寄存器 n (HCCHARn) ..... 730

26.10.24. USB 主机通道分离控制寄存器 n (HCSPLTn) ..... 731
26.10.25. USB 主机通道中断寄存器 n (HCINTn) ..... 731
26.10.26. USB 主机通道中断屏蔽寄存器 n (HCINTMSKn) ..... 732
26.10.27. USB 主机通道传输大小寄存器 n (HCTSIZn) ..... 733
26.10.28. USB 主机通道 DMA 地址寄存器 n (HCDMAn) ..... 734
26.10.29. USB 设备配置寄存器 (DCFG) ..... 734
26.10.30. USB 设备控制寄存器 (DCTL) ..... 735
26.10.31. USB 设备状态寄存器 (DSTS) ..... 736
26.10.32. USB 设备 IN 端点通用中断屏蔽寄存器 (DIEPMSK) ..... 737
26.10.33. USB 设备 OUT 端点通用中断屏蔽寄存器 (DOEPMSK) ..... 738
26.10.34. USB 设备全体端点中断寄存器 (DAINT) ..... 739
26.10.35. USB 全体端点中断屏蔽寄存器 (DAINTMSK) ..... 739
26.10.36. USB 设备 VBUS 放电时间寄存器 (DVBUSDIS) ..... 740
26.10.37. USB 设备 VBUS 脉冲时间寄存器 (DVBUSPULSE) ..... 740
26.10.38. USB 设备 IN 端点 FIFO 空中断屏蔽寄存器 (DIEPEMPMSK) ..... 741
26.10.39. USB 设备单个端点中断寄存器 (DEACHINT) ..... 741
26.10.40. USB 设备单个端点中断屏蔽寄存器 (DEACHINTMSK) ..... 741
26.10.41. USB 设备 IN 端点 1 中断屏蔽寄存器 (DIEPEACHMSK1) ..... 741
26.10.42. USB 设备 OUT 端点 1 中断屏蔽寄存器 (DOEPEACHMSK1) ..... 742
26.10.43. USB 设备端点控制寄存器 n (DIEPCTLn) ..... 742
26.10.44. USB 设备端点中断寄存器 n (DIEPINTn) ..... 744
26.10.45. USB 设备 IN 端点 0 传输大小寄存器 (DIEPTSIZ0) ..... 745
26.10.46. USB 设备端点传输大小寄存器 n (DIEPTSIZn) ..... 745
26.10.47. USB 设备 IN 端点 DMA 地址寄存器 n (DIEPDMAn) ..... 746
26.10.48. USB 设备 IN 端点发送 FIFO 状态寄存器 (DTXFSTSn) ..... 747
26.10.49. USB 设备控制 OUT 端点 0 控制寄存器 (DOEPCTL0) ..... 747
26.10.50. USB 设备端点控制寄存器 (DOEPCTLn) ..... 748
26.10.51. USB 设备端点中断寄存器 n (DOEPINTn) ..... 750
26.10.52. USB 设备 OUT 端点 0 传输大小寄存器 (DOEPTSIZ0) ..... 751
26.10.53. USB 设备端点传输大小寄存器 n (DOEPTSIZn) ..... 752
26.10.54. USB 设备 OUT 端点 DMA 地址寄存器 n (DOEPDMAn) ..... 753

26.10.55. USB 电源和时钟门控控制寄存器 (PCGCCTL)..... 753
27. OTP 控制器(OTP)..... 754
27.1. OTP 简介..... 754
27.2. OTP 功能描述..... 754
27.2.1. OTP 只读锁定..... 754
27.2.2. OTP 编程操作保护..... 754
27.3. OTP_CTRL 寄存器..... 754
27.3.1. OTP 控制状态寄存器（OTP_CS）..... 754
27.3.2. OTP 启动保护寄存器（OTP_PROT）..... 755
27.3.3. OTP 编程擦除地址寄存器（OTP_ADDR）..... 755
27.3.4. OTP 编程数据寄存器（OTP_PDATA）..... 756
27.3.5. OTP 时序寄存器（OTP_TIM）..... 756
27.3.6. OTP 主存区域只读区域寄存器 3（OTP_RO3）..... 756
27.3.7. OTP 主存区只读锁定寄存器 3（OTP_ROL3）..... 757
28. 真随机数发生器(TRNG)..... 758
28.1. TRNG 简介..... 758
28.2. TRNG 主要特性..... 758
28.3. TRNG 功能描述..... 758
28.4. TRNG 寄存器..... 758
28.4.1. 控制状态寄存器（RNG_CSR）..... 758
28.4.2. 数据寄存器（RNG_DATA）..... 759
28.4.3. 模拟控制寄存器（RNG_AMA）..... 759
28.4.4. RNG FIFO Index（RNG_INDEX）..... 759
28.4.5. 伪随机序列寄存器（RNG_PN0）..... 760
28.4.6. 伪随机序列寄存器（RNG_PN1）..... 760
28.4.7. 伪随机序列寄存器（RNG_PN2）..... 761
28.4.8. 伪随机序列寄存器（RNG_PN3）..... 761
29. 实时时钟(RTC)..... 762
29.1. RTC 简介..... 762
29.2. RTC 特性..... 762
29.3. RTC 寄存器..... 762

29.3.1. RTC 控制状态寄存器（RTC_CS） ..... 762
29.3.2. RTC 计数初始值寄存器（RTC_REF） ..... 763
29.3.3. RTC 闹钟设置寄存器（RTC_ARM） ..... 763
29.3.4. RTC 当前计数值寄存器（RTC_TIM） ..... 763
29.3.5. RTC 中断清除寄存器（RTC_INTCLR） ..... 764
30. XIP Flash 控制器(XFC) ..... 765
30.1. XFC 简介 ..... 765
30.2. XFC 主要特性 ..... 765
30.3. XFC 寄存器 ..... 765
30.3.1. XFC 命令寄存器(XFC_CMDR) ..... 765
30.3.2. XFC 地址寄存器 (XFC_ADDRR) ..... 766
30.3.3. XFC 读写数据量寄存器(XFC_BNR) ..... 766
30.3.4. XFC 写数据 FIFO 寄存器(XFC_WFIFOR) ..... 767
30.3.5. XFC 读数据 FIFO 寄存器(XFC_RFIFOR) ..... 767
30.3.6. XFC 配置寄存器(XFC_CFGR) ..... 767
30.3.7. XFC 写 REG 寄存器(XFC_WREGR) ..... 768
30.3.8. XFC 读 REG 寄存器(XFC_RREGR) ..... 769
30.3.9. XFC 状态寄存器(XFC_SR) ..... 769
30.3.10. XFC 状态清除寄存器(XFC_SCR) ..... 770
30.3.11. XFC 休眠配置寄存器(XFC_LPCFGR) ..... 770
30.3.12. XFC DMA 配置寄存器(XFC_DMACR) ..... 771
30.3.13. XFC FIFO 配置寄存器(XFC_FCR) ..... 772
31. SDRAM 控制器(SDRAM) ..... 773
31.1. SDRAM 简介 ..... 773
31.2. SDRAM 主要特性 ..... 773
31.3. SDRAM 的功能描述 ..... 773
31.3.1. SDRAM 上电初始化 ..... 773
31.3.2. SDRAM 的模式寄存器配置 ..... 775
31.3.3. 自动刷新 ..... 775
31.3.4. 低功耗模式 ..... 775
31.3.5. SDRAM 的地址映射 ..... 775

31.4. SDRAM 寄存器 ..... 776
31.4.1. SDRAM 配置寄存器(SDRAM_SCONR) ..... 776
31.4.2. SDRAM 时序寄存器 0(SDRAM_STMG0R) ..... 776
31.4.3. SDRAM 时序寄存器 1(SDRAM_STMG1R) ..... 778
31.4.4. SDRAM 控制寄存器(SDRAM_SCTLR) ..... 778
31.4.5. SDRAM 刷新间隔寄存器(SDRAM_SREFR) ..... 779
31.4.6. SDRAM 片选寄存器 x(SDRAM_SCSLRx)(x=0,1) ..... 779
32. 图像处理加速器(DMA2D) ..... 781
32.1. DMA2D 简介 ..... 781
32.2. DMA2D 主要特性 ..... 781
32.3. DMA2D 功能说明 ..... 782
32.3.1. 概述 ..... 782
32.3.2. DMA2D 控制 ..... 782
32.3.3. DMA2D 前景层 FIFO 和背景层 FIFO ..... 782
32.3.4. DMA2D 前景层和背景层像素格式转换器 (PFC) ..... 783
32.3.5. DMA2D 前景层 FIFO 和背景层 CLUT 接口 ..... 784
32.3.6. DMA2D 混合器 ..... 785
32.3.7. DMA2D 输出 PFC ..... 785
32.3.8. DMA2D 输出 FIFO ..... 786
32.3.9. DMA2D 输出 FIFO 字节重新排序 ..... 786
32.3.10. DMA2D AHB 主设备端口定时器 ..... 786
32.3.11. DMA2D 事务 ..... 786
32.3.12. DMA2D 配置 ..... 787
32.3.13. DMA2D 传输控制（启动、挂起、中止和完成） ..... 789
32.3.14. 水印 ..... 790
32.3.15. 错误管理 ..... 790
32.3.16. AHB 死区 ..... 790
32.4. DMA2D 中断 ..... 790
32.5. DMA2D 寄存器 ..... 791
32.5.1. DMA2D 控制寄存器 (DMA2D_CR) ..... 791
32.5.2. DMA2D 中断状态寄存器 (DMA2D_ISR) ..... 792

32.5.3. DMA2D 中断标志清零寄存器(DMA2D_IFCR) ..... 793
32.5.4. DMA2D 前景层存储器地址寄存器(DMA2D_FGMAR) ..... 793
32.5.5. DMA2D 前景层偏移寄存器(DMA2D_FGOR) ..... 794
32.5.6. DMA2D 背景层存储器地址寄存器(DMA2D_BGMAR) ..... 794
32.5.7. DMA2D 背景层偏移寄存器(DMA2D_BGOR) ..... 795
32.5.8. DMA2D 前景层 PFC 控制寄存器(DMA2D_FGPFCCR) ..... 795
32.5.9. DMA2D 前景层颜色寄存器(DMA2D_FGCOLR) ..... 797
32.5.10. DMA2D 背景层 PFC 控制寄存器(DMA2D_BGPFCCR) ..... 798
32.5.11. DMA2D 背景层颜色寄存器(DMA2D_BGCOLR) ..... 799
32.5.12. DMA2D 前景层 CLUT 存储器地址寄存器(DMA2D_FGCMAR) ..... 800
32.5.13. DMA2D 背景层 CLUT 存储器地址寄存器(DMA2D_BGCMAR) ..... 800
32.5.14. DMA2D 输出 PFC 控制寄存器(DMA2D_OPFCCR) ..... 800
32.5.15. DMA2D 输出颜色寄存器(DMA2D_OCOLR) ..... 801
32.5.16. DMA2D 输出存储器地址寄存器(DMA2D_OMAR) ..... 802
32.5.17. DMA2D 输出偏移寄存器(DMA2D_OOR) ..... 802
32.5.18. DMA2D 行数寄存器(DMA2D_NLR) ..... 803
32.5.19. DMA2D 行水印寄存器(DMA2D_LWR) ..... 803
32.5.20. DMA2D AHB 主设备定时器配置寄存器(DMA2D_AMTCR) ..... 804
33. JPEG 解码控制器(JPEGD) ..... 805
33.1. JPEGD 简介 ..... 805
33.2. JPEGD 的功能描述 ..... 805
33.3. JPEGD 寄存器 ..... 805
33.3.1. JPEGD 配置寄存器 0(JPEGD_CONFR0) ..... 805
33.3.2. JPEGD 配置寄存器 1(JPEGD_CONFR1) ..... 805
33.3.3. JPEGD 配置寄存器 2(JPEGD_CONFR2) ..... 806
33.3.4. JPEGD 配置寄存器 3(JPEGD_CONFR3) ..... 806
33.3.5. JPEGD 配置寄存器 4-7(JPEGD_CONFR4-7) ..... 807
33.3.6. JPEGD 输入 FIFO 阈值寄存器 (JPEGD_INTH) ..... 807
33.3.7. JPEGD 输出 FIFO 阈值寄存器 (JPEGD_OUTTH) ..... 808
33.3.8. JPEGD 控制寄存器 (JPEGD_CR) ..... 808
33.3.9. JPEGD 控制寄存器 (JPEGD_SR) ..... 809

33.3.10. JPEGD 数据状态清除寄存器 (JPEGD_CFR)..... 810
33.3.11. JPEGD 数据输入寄存器 (JPEGD_DIR)..... 810
33.3.12. JPEGD 数据输出寄存器 (JPEGD_DOR)..... 810
34. LCD-TFT 控制器(LTDC)..... 812
34.1. LTDC 简介..... 812
34.2. LTDC 主要特性..... 812
34.3. LTDC 功能说明..... 813
34.3.1. LTDC 框图..... 813
34.3.2. LTDC 复位和时钟..... 813
34.3.3. LTDC 引脚和信号接口..... 813
34.4. LTDC 可编程参数..... 814
34.4.1. LTDC 全局配置参数..... 814
34.4.2. 层可编程参数..... 816
34.5. LTDC 中断..... 819
34.6. LTDC 编程步骤..... 820
34.7. LTDC 寄存器..... 821
34.7.1. LTDC 同步大小配置寄存器 (LTDC_SSCR)..... 821
34.7.2. LTDC 后沿配置寄存器 (LTDC_BPCR)..... 821
34.7.3. LTDC 有效宽度配置寄存器 (LTDC_AWCR)..... 822
34.7.4. LTDC 总宽度配置寄存器 (LTDC_TWCR)..... 822
34.7.5. LTDC 全局控制寄存器 (LTDC_GCR)..... 823
34.7.6. LTDC 影子重载配置寄存器 (LTDC_SRCR)..... 824
34.7.7. LTDC 背景色配置寄存器 (LTDC_BCCR)..... 825
34.7.8. LTDC 中断使能寄存器 (LTDC_IER)..... 825
34.7.9. LTDC 中断状态寄存器 (LTDC_ISR)..... 826
34.7.10. LTDC 中断清零寄存器 (LTDC_ICR)..... 826
34.7.11. LTDC 行中断位置配置寄存器 (LTDC_LIPCR)..... 827
34.7.12. LTDC 当前位置状态寄存器 (LTDC_CPSR)..... 827
34.7.13. LTDC 当前显示状态寄存器 (LTDC_CDSR)..... 828
34.7.14. LTDC 第 x 层控制寄存器 (LTDC_LxCR)（其中, x = 1..2)..... 828
34.7.15. LTDC 第 x 层窗口水平位置配置寄存器 (LTDC_LxWHPCR)（其中 x=1..2)..... 829

34.7.16. LTDC 第 x 层窗口垂直位置配置寄存器 (LTDC_LxWVPCR)（其中 x=1..2)..... 830
34.7.17. LTDC 第 x 层色键配置寄存器 (LTDC_LxCKCR)（其中 x=1..2) ..... 831
34.7.18. LTDC 中断使能寄存器 (LTDC_IER)..... 831
34.7.19. LTDC 第 x 层常数 Alpha 配置寄存器 (LTDC_LxCACR)（其中 x=1..2)..... 832
34.7.20. LTDC 第 x 层默认颜色配置寄存器 (LTDC_LxDCCR)（其中 x=1..2)..... 832
34.7.21. LTDC 第 x 层混合系数配置寄存器 (LTDC_LxBFCR)（其中 x=1..2) ..... 833
34.7.22. LTDC 第 x 层颜色帧缓冲区地址寄存器 (LTDC_LxCFBAR)（其中 x=1..2) ..... 834
34.7.23. LTDC 第 x 层颜色帧缓冲区长度寄存器 (LTDC_LxCFBLR)（其中 x=1..2)..... 834
34.7.24. LTDC 第 x 层颜色帧缓冲区长度寄存器 (LTDC_LxCFBLR)（其中 x=1..2)..... 835
34.7.25. LTDC 第 x 层 CLUT 写寄存器 (LTDC_LxCLUTWR) （其中 x=1..2)..... 835
34.7.26. LTDC AHB BURST 配置寄存器 (LTDC_CFG0)..... 836
35. 修订历史..... 837

# 表格

表 1-1 寄存器相关缩写词 ..... 46
表 1-2 术语、定义及缩略语 ..... 46
表 2-1 存储映射表 ..... 48
表 2-2 SRAM 模块构成 ..... 50
表 3-1 低功耗模式汇总 ..... 52
表 3-2 停止工作模式 ..... 53
表 4-1 MH2435 向量表 ..... 60
表 7-1 端口位配置表 ..... 112
表 7-2 灵活的 SWJ-DP 引脚分配 ..... 114
表 9-1 DMA1 请求映射 ..... 134
表 9-2 DMA2 请求映射 ..... 135
表 9-3 源和目的地址 ..... 136
表 9-4 双缓冲区模式下的源和目标地址寄存器(DBM=1) ..... 140
表 9-5 封装/解封和字节序行为(位 PINC = MINC = 1) ..... 141
表 9-6 PSIZE 与 MSIZE 确定时对 NDT 的限制条件 ..... 142
表 9-7 Stream0-Stream5 FIFO 阈值配置要求 ..... 143
表 9-8 Stream6-Stream7 FIFO 阈值配置要求 ..... 144
表 9-9 可能的 DMA 配置 ..... 146
表 9-10 DMA 中断请求 ..... 148
表 11-1 不同模式下引脚配置 ..... 163
表 11-2 时钟相位与极性 ..... 164
表 11-3 Quad-SPI 中断 ..... 167
表 12-1 ADC 引脚 ..... 184
表 12-2 模拟看门狗通道选择 ..... 187
表 12-3 配置触发极性 ..... 191
表 12-4 规则通道的外部触发 ..... 191
表 12-5 注入通道的外部触发 ..... 191
表 12-6 ADC 中断 ..... 204
表 13-1 外部触发器 ..... 222

表 14-1 计数方向与编码器信号的关系 ..... 269
表 14-2 TIMx 内部触发连接 ..... 280
表 14-3 具有断路功能的互补通道 OCx 和 OCxN 的输出控制位 ..... 290
表 15-1 计数方向与编码器信号的关系 ..... 323
表 15-2 TIMx 内部触发连接 ..... 336
表 15-3 标准 OCx 通道输出控制位 ..... 345
表 16-1 TIMx 内部触发连接 ..... 373
表 16-2 标准 OCx 通道的输出控制位 ..... 379
表 16-3 标准 OCx 通道的输出控制位 ..... 386
表 18-1 32 kHz (LSI) 频率条件下 IWDG 超时周期的最小值/最大值(1) ..... 400
表 19-1 30 MHz ($f_{PCLK1}$) 时的超时值 ..... 405
表 20-1 符合 Thd:dat(max) 的 DNF[3:0] 最大值 ..... 417
表 20-2 SMBUS 与 I2C ..... 418
表 20-3 I2C 中断请求 ..... 421
表 21-1 通过采样数据进行噪声检测 ..... 443
表 21-2 采用 16 倍过采样时，在 $f_{PCLK}$ = 42/84MHz 下编程波特率时的误差计算 ..... 445
表 21-3 采用 8 倍过采样时，在 $f_{PCLK}$ = 42/84MHz 下编程波特率时的误差计算 ..... 446
表 21-4 采用 16 倍过采样时，在 $f_{PCLK}$ = 45/90MHz 下编程波特率时的误差计算 ..... 446
表 21-5 采用 8 倍过采样时，在 $f_{PCLK}$ = 45/90MHz 下编程波特率时的误差计算 ..... 447
表 21-6 DIV_Fraction 为 0 时的 USART 接收器容差 ..... 448
表 21-7 DIV_Fraction 不为 0 时的 USART 接收器容差 ..... 448
表 21-8 帧格式 ..... 450
表 21-9 USART 中断请求 ..... 462
表 21-10 USART 模式配置(1) ..... 463
表 22-1 SPI 中断请求 ..... 492
表 22-2 $I^2S$ 中断请求 ..... 506
表 23-1 SDIO I/O 定义 ..... 520
表 23-2 命令格式 ..... 524
表 23-3 短响应格式 ..... 524
表 23-4 长响应格式 ..... 524
表 23-5 命令路径状态标志 ..... 524

表 23-6 数据令牌格式.....527
表 23-7 传输 FIFO 状态标志.....527
表 23-8 接收 FIFO 状态标志.....528
表 23-9 卡状态.....536
表 23-10 SD 状态.....538
表 23-11 速度等级代码字段.....539
表 23-12 移动性能字段.....539
表 23-13 AU_SIZE 字段.....539
表 23-14 最大 AU 大小.....540
表 23-15 擦除大小字段.....540
表 23-16 擦除超时字段.....540
表 23-17 擦除偏移字段.....540
表 23-18 面向块的写入命令.....542
表 23-19 面向块的写保护命令.....543
表 23-20 擦除命令.....543
表 23-21 I/O 模式命令.....543
表 23-22 锁定卡.....544
表 23-23 应用程序特定的命令.....544
表 23-24 R1 响应.....544
表 23-25 R2 响应.....545
表 23-26 R3 响应.....545
表 23-27 R4 响应.....545
表 23-28 R4b 响应.....545
表 23-29 R5 响应.....546
表 23-30 R6 响应.....546
表 24-1 发送邮箱映射.....573
表 24-2 接收邮箱映射.....574
表 25-1 复用功能映射.....598
表 25-2 管理帧格式.....600
表 25-3 时钟选择.....601
表 25-4 TX 接口信号编码.....602

表 25-5 RX 接口信号编码 ..... 603
表 25-6 帧状态 ..... 615
表 25-7 目标地址过滤 ..... 619
表 25-8 源地址过滤 ..... 620
表 25-9 接收描述符 0 – 位 7、5 和 0 的编码（仅正常描述符格式，EDFE=0） ..... 645
表 25-10 时间戳快照对寄存器位的相依性 ..... 676
表 26-1 缩写描述 ..... 695
表 32-1 输入时支持的颜色模式 ..... 783
表 32-2 存储器中的数据顺序 ..... 783
表 32-3 Alpha 模式配置 ..... 784
表 32-4 支持的 CLUT 颜色模式 ..... 785
表 32-5 存储器中的 CLUT 数据顺序 ..... 785
表 32-6 输出时支持的颜色模式 ..... 785
表 32-7 存储器中的数据顺序 ..... 786
表 32-8 DMA2D 中断请求 ..... 790
表 34-1 LTDC 引脚和信号接口 ..... 813
表 34-2 像素数据映射与颜色格式的关系 ..... 817
表 34-3 LTDC 中断请求 ..... 820
表 35-1 修订历史 ..... 837

# 图片

图 3-1 上电复位/掉电复位波形 ..... 51
图 5-1 外部中断/事件控制器框图 ..... 63
图 5-2 外部中断/事件 GPIO 映射 ..... 65
图 6-1 时钟树 ..... 70
图 7-1 输入浮空/上拉/下拉配置 ..... 116
图 7-2 输出配置 ..... 117
图 7-3 复用功能配置 ..... 118
图 7-4 高阻态模拟配置 ..... 118
图 9-1 DMA 框图 ..... 133

图 9-2 通道选择 ..... 134
图 9-3 外设到存储器模式 ..... 137
图 9-4 存储器到外设模式 ..... 138
图 9-5 存储器到存储器模式 ..... 139
图 9-6 FIFO 结构 ..... 143
图 11-1 Quad-SPI 数据大小端转换 ..... 165
图 12-1 单个 ADC 框图 ..... 184
图 12-2 时序图 ..... 186
图 12-3 模拟看门狗的保护区域 ..... 187
图 12-4 注入转换延迟 ..... 188
图 12-5 12 位数据的右对齐 ..... 190
图 12-6 12 位数据的左对齐 ..... 190
图 12-7 6 位数据的左对齐 ..... 190
图 12-8 多重 ADC 框图(1) ..... 194
图 12-9 4 通道的注入同时模式：双重 ADC 模式 ..... 196
图 12-10 4 通道的注入同时模式：三重 ADC 模式 ..... 197
图 12-11 16 通道的规则同时模式：双重 ADC 模式 ..... 197
图 12-12 16 通道的规则同时模式：三重 ADC 模式 ..... 198
图 12-13 连续转换模式下 1 通道的交替模式：双重 ADC 模式 ..... 199
图 12-14 连续转换模式下 1 通道的交替模式：三重 ADC 模式 ..... 200
图 12-15 交替触发：各个 ADC 的注入组 ..... 201
图 12-16 交替触发：不连续采样模式下的 4 个注入通道（各个 ADC） ..... 201
图 12-17 交替触发：各个 ADC 的注入组 ..... 202
图 12-18 交替 + 规则同时 ..... 202
图 12-19 在注入转换期间出现的触发事件 ..... 203
图 13-1 DAC 通道框图 ..... 220
图 13-2 DAC 单通道模式下的数据寄存器 ..... 221
图 13-3 DAC 双通道模式下的数据寄存器 ..... 221
图 13-4 关闭触发 (TEN = 0) 时的转换时序图 TEN = 0 ..... 222
图 13-5 DAC LFSR 寄存器计算算法 ..... 223
图 13-6 LFSR 产生波形的 DAC 转换（使能软件触发） ..... 224

图 13-7 生成 DAC 三角波 ..... 224
图 13-8 生成三角波波形的 DAC 转换（使能软件触发） ..... 225
图 14-1 高级控制定时器框图 ..... 238
图 14-2 预分频器分频由 1 变为 2 时的计数器时序图 ..... 239
图 14-3 预分频器分频由 1 变为 4 时的计数器时序图 ..... 240
图 14-4 计数器时序图，内部时钟 1 分频 ..... 241
图 14-5 计数器时序图，内部时钟 2 分频 ..... 241
图 14-6 计数器时序图，内部时钟 4 分频 ..... 242
图 14-7 计数器时序图，内部时钟 N 分频 ..... 242
图 14-8 计数器时序图，ARPE=0 时更新事件(TIMx_ARR 未预装载) ..... 243
图 14-9 计数器时序图，ARPE=1 时更新事件(TIMx_ARR 预装载) ..... 243
图 14-10 计数器时序图，内部时钟 1 分频 ..... 244
图 14-11 计数器时序图，内部时钟 2 分频 ..... 245
图 14-12 计数器时序图，内部时钟 4 分频 ..... 245
图 14-13 计数器时序图，内部时钟 N 分频 ..... 246
图 14-14 计数器时序图，未使用重复计数器时更新事件 ..... 246
图 14-15 计数器时序图，内部时钟 1 分频，TIMx_ARR = 0x6 ..... 247
图 14-16 计数器时序图，内部时钟 2 分频 ..... 248
图 14-17 计数器时序图，内部时钟 4 分频，TIMx_ARR = 0x36 ..... 248
图 14-18 计数器时序图，内部时钟 N 分频 ..... 249
图 14-19 计数器时序图，ARPE=1 时的更新事件(计数器下溢) ..... 249
图 14-20 计数器时序图，ARPE=1 时的更新事件(计数器上溢) ..... 250
图 14-21 不同模式和 TIMx_RCR 寄存器设置下的更新频率示例 ..... 251
图 14-22 正常模式下的控制电路，内部时钟 1 分频 ..... 252
图 14-23 TI2 外部时钟连接示例 ..... 252
图 14-24 TI2 外部时钟连接示例 ..... 253
图 14-25 外部触发输入模块 ..... 253
图 14-26 外部时钟模式 2 下的控制电路 ..... 254
图 14-27 捕获/比较通道(例如：通道 1 输入阶段) ..... 255
图 14-28 捕获/比较通道 1 主电路 ..... 255
图 14-29 捕获/比较通道的输出阶段(通道 1 到 3) ..... 256

图 14-30 捕获/比较通道的输出阶段(通道 4) ..... 256
图 14-31 PWM 输入模式时序 ..... 258
图 14-32 输出比较模式，翻转 OC1 ..... 259
图 14-33 边沿对齐模式 PWM 波形(ARR=8) ..... 260
图 14-34 中心对齐模式 PWM 波形(ARR=8) ..... 261
图 14-35 带死区插入的互补输出 ..... 262
图 14-36 延迟时间大于负脉冲宽度的死区波形 ..... 262
图 14-37 延迟时间大于正脉冲宽度的死区波形 ..... 263
图 14-38 输出的断路响应行为 ..... 265
图 14-39 清除 TIMx 的 OCxREF ..... 266
图 14-40 COM 事件生成 6 步 PWM 示例(OSSR=1) ..... 267
图 14-41 单脉冲模式示例 ..... 268
图 14-42 编码器接口模式下的计数器工作示例 ..... 270
图 14-43 TI1FP1 极性反相时的编码器接口模式示例 ..... 270
图 14-44 霍尔传感器接口示例 ..... 272
图 14-45 复位模式下的控制电路 ..... 273
图 14-46 门控模式下的控制电路 ..... 274
图 14-47 触发模式下的控制电路 ..... 274
图 14-48 外部时钟模式 2+触发模式下的控制电路 ..... 275
图 15-1 通用定时器框图 ..... 300
图 15-2 预分频器分频由 1 变为 2 时的计数器时序图 ..... 301
图 15-3 预分频器分频由 1 变为 4 时的计数器时序图 ..... 301
图 15-4 计数器时序图，内部时钟 1 分频 ..... 302
图 15-5 计数器时序图，内部时钟 2 分频 ..... 303
图 15-6 计数器时序图，内部时钟 4 分频 ..... 303
图 15-7 计数器时序图，内部时钟 N 分频 ..... 303
图 15-8 计数器时序图，ARPE=0 时更新事件(TIMx_ARR 未预装载) ..... 304
图 15-9 计数器时序图，ARPE=1 时更新事件(TIMx_ARR 已预装载) ..... 304
图 15-10 计数器时序图，内部时钟 1 分频 ..... 305
图 15-11 计数器时序图，内部时钟 2 分频 ..... 305
图 15-12 计数器时序图，内部时钟 4 分频 ..... 306

图 15-13 计数器时序图，内部时钟 N 分频.....306
图 15-14 计数器时序图，更新事件.....307
图 15-15 计数器时序图，内部时钟 1 分频，TIMx_ARR=0x6.....308
图 15-16 计数器时序图，内部时钟 2 分频.....308
图 15-17 计数器时序图，内部时钟 4 分频，TIMx_ARR=0x36.....309
图 15-18 计数器时序图，内部时钟 N 分频.....309
图 15-19 计数器时序图，ARPE=1 时的更新事件(计数器下溢).....310
图 15-20 计数器时序图，ARPE=1 时的更新事件(计数器上溢).....310
图 15-21 正常模式下的控制电路，内部时钟 1 分频.....311
图 15-22 TI2 外部时钟连接示例.....311
图 15-23 外部时钟模式 1 下的控制电路.....312
图 15-24 外部触发输入模块.....313
图 15-25 外部时钟模式 2 下的控制电路.....313
图 15-26 捕获/比较通道(例如：通道 1 输入阶段).....314
图 15-27 捕获/比较通道 1 主电路.....314
图 15-28 捕获/比较通道的输出阶段(通道 1).....315
图 15-29 PWM 输入模式时序.....316
图 15-30 输出比较模式，翻转 OC1.....318
图 15-31 边沿对齐模式的 PWM 波形(ARR=8).....319
图 15-32 中心对齐模式 PWM 波形(ARR=8).....320
图 15-33 单脉冲模式示例.....321
图 15-34 清除 TIMx 的 OCxREF.....322
图 15-35 编码器接口模式下的计数器工作示例.....323
图 15-36 TI1FP1 极性反相时的编码器接口模式示例.....324
图 15-37 复位模式下的控制电路.....325
图 15-38 门控模式下的控制电路.....325
图 15-39 触发模式下的控制电路.....326
图 15-40 外部时钟模式 2 + 触发模式下的控制电路.....327
图 15-41 主/从定时器示例.....327
图 15-42 使用定时器 1 的 OC1REF 对定时器 2 实施门控控制.....328
图 15-43 使用定时器 1 的使能信号对定时器 2 实施门控控制.....329

图 15-44 使用定时器 1 的更新事件触发定时器 2.....330
图 15-45 使用定时器 1 的使能信号触发定时器 2.....330
图 15-46 使用定时器 1 的 TI1 输入触发定时器 1 和定时器 2.....331
图 16-1 通用定时器框图(TIM9 和 TIM12).....352
图 16-2 通用定时器框图 (TIM10/11/13/14).....353
图 16-3 预分频器分频由 1 变为 2 时的计数器时序图.....354
图 16-4 预分频器分频由 1 变为 4 时的计数器时序图.....355
图 16-5 计数器时序图，内部时钟 1 分频.....356
图 16-6 计数器时序图，内部时钟 2 分频.....356
图 16-7 计数器时序图，内部时钟 4 分频.....357
图 16-8 计数器时序图，内部时钟 N 分频.....357
图 16-9 计数器时序图， ARPE=0 时更新事件(TIMx_ARR 未预装载).....358
图 16-10 计数器时序图， ARPE=1 时更新事件(TIMx_ARR 已预装载).....358
图 16-11 正常模式下的控制电路，内部时钟 1 分频.....359
图 16-12 TI2 外部时钟连接示例.....360
图 16-13 外部时钟模式 1 下的控制电路.....360
图 16-14 捕获/比较通道(例如：通道 1 输入阶段).....361
图 16-15 捕获/比较通道 1 主电路.....361
图 16-16 捕获/比较通道的输出阶段(通道 1).....362
图 16-17 PWM 输入模式时序.....363
图 16-18 输出比较模式，翻转 OC1.....365
图 16-19 比边沿对齐模式的 PWM 波形 (ARR=8).....366
图 16-20 单脉冲模式示例.....367
图 16-21 复位模式下的控制电路.....368
图 16-22 门控模式下的控制电路.....369
图 16-23 触发模式下的控制电路.....370
图 17-1 基本定时器框图.....389
图 17-2 预分频器分频由 1 变为 2 时的计数器时序图.....391
图 17-3 预分频器分频由 1 变为 4 时的计数器时序图.....391
图 17-4 计数器时序图，内部时钟 1 分频.....392
图 17-5 计数器时序图，内部时钟 2 分频.....393

图 17-6 计数器时序图，内部时钟 4 分频 ..... 393
图 17-7 计数器时序图，内部时钟 N 分频 ..... 393
图 17-8 计数器时序图，ARPE = 0 时更新事件(TIMx_ARR 未预装载) ..... 394
图 17-9 计数器时序图，ARPE = 1 时更新事件(TIMx_ARR 已预装载) ..... 394
图 17-10 正常模式下的控制电路，内部时钟 1 分频 ..... 395
图 18-1 独立看门狗框图 ..... 400
图 19-1 看门狗框图 ..... 403
图 19-2 窗口看门狗时序图 ..... 404
图 20-1 I2C 总线协议 ..... 409
图 20-2 I2C 框图 ..... 409
图 20-3 从发送器的传输时序图 ..... 411
图 20-4 从接收器的传输时序图 ..... 411
图 20-5 主发送器的传输时序图 ..... 414
图 20-6 主接收器的传输时序图 ..... 415
图 20-7 I2C 中断映射图 ..... 422
图 21-1 字长编程 ..... 437
图 21-2 可配置的停止位 ..... 438
图 21-3 发送时的 TC/TXE 行为 ..... 439
图 21-4 16 倍或 8 倍过采样时的起始位检测 ..... 440
图 21-5 16 倍过采样时的数据采样 ..... 442
图 21-6 8 倍过采样时的数据采样 ..... 443
图 21-7 使用空闲线路检测时的静音模式 ..... 449
图 21-8 使用地址标记检测时的静音模式 ..... 450
图 21-9 LIN 模式下的断路检测(11 位断路长度 - LBDL 位置 1) ..... 452
图 21-10 LIN 模式下的断路检测与帧错误检测 ..... 453
图 21-11 USART 同步发送示例 ..... 454
图 21-12 USART 数据时钟时序图(M=0) ..... 454
图 21-13 USART 数据时钟时序图(M=1) ..... 455
图 21-14 RX 数据建立/保持时间 ..... 455
图 21-15 ISO 7816-3 异步协议 ..... 456
图 21-16 使用 1.5 个停止位检测奇偶校验错误 ..... 457

图 21-17 IrDA SIR ENDEC - 框图.....458
图 21-18 IrDA 数据调制 (3/16) - 正常模式.....459
图 21-19 使用 DMA 进行发送.....460
图 21-20 使用 DMA 进行接收.....461
图 21-21 两个 USART 间的硬件流控制.....461
图 21-22 RTS 流控制.....462
图 21-23 CTS 流控制.....462
图 22-1 单个主器件/单个从器件应用.....475
图 22-2 数据时钟时序图.....476
图 22-3 TI 模式 – 从模式，单次传输.....478
图 22-4 TI 模式 – 从模式，连续传输.....478
图 22-5 TI 模式 – 主模式，单次传输.....479
图 22-6 TI 模式 – 主模式，连续传输.....480
图 22-7 主/全双工模式下的 TXE/RXNE/BSY 行为（在连续传输的情况下）.....482
图 22-8 从/全双工模式下的 TXE/RXNE/BSY 行为（在连续传输的情况下）.....483
图 22-9 主设备只发送模式下的 TXE/ BSY 行为（在连续传输的情况下）.....484
图 22-10 从设备只发送模式下的 TXE/ BSY 行为（在连续传输的情况下）.....484
图 22-11 只接收模式下的 RXNE 行为（在连续传输的情况下）.....485
图 22-12 发送时的 TXE/BSY 行为（在间断传输的情况下）.....485
图 22-13 使用 DMA 进行发送.....490
图 22-14 使用 DMA 进行接收.....490
图 22-15 TI 模式帧格式错误检测.....492
图 22-16 I²S 框图.....493
图 22-17 I²S 全双工框图.....494
图 22-18 I²S Philips 协议波形(16/32 位全精度，CPOL = 0).....495
图 22-19 I²S Philips 标准波形(24 位帧，CPOL = 0).....495
图 22-20 发送 0x8EAA33.....496
图 22-21 接收 0x8EAA33.....496
图 22-22 I²S Philips 标准(16 位扩展为 32 位数据包帧，CPOL = 0).....496
图 22-23 示例.....497
图 22-24 MSB 对齐的 16 位或 32 位全精度长度，CPOL = 0.....497

图 22-25 MSB 对齐的 24 位帧长度，CPOL = 0 ..... 497
图 22-26 扩展为 32 位数据包帧的 MSB 对齐的 16 位，CPOL = 0 ..... 498
图 22-27 LSB 对齐的 16 位或 32 位全精度，CPOL = 0 ..... 498
图 22-28 LSB 对齐的 24 位帧长度，CPOL = 0 ..... 498
图 22-29 发送 0x3478AE 所需的操作 ..... 499
图 22-30 接收 0x3478AE 所需的操作 ..... 499
图 22-31 扩展为 32 位数据包帧的 LSB 对齐的 16 位，CPOL = 0 ..... 499
图 22-32 扩展为 32 位数据包帧的 LSB 对齐的 16 位示例 ..... 500
图 22-33 PCM 标准波形(16 位) ..... 500
图 22-34 PCM 标准波形(16 位扩展到 32 位数据包帧) ..... 500
图 22-35 音频采样频率定义 ..... 501
图 22-36 I²S 时钟发生器架构 ..... 501
图 23-1 SDIO “无响应”和“无数据”操作 ..... 518
图 23-2 SDIO(多个)块读取操作 ..... 518
图 23-3 SDIO(多个)块写入操作 ..... 518
图 23-4 SDIO 连续读取操作 ..... 519
图 23-5 SDIO 连续写入操作 ..... 519
图 23-6 SDIO 适配器 ..... 520
图 23-7 控制单元 ..... 521
图 23-8 SDIO 适配器命令路径 ..... 522
图 23-9 命令路径状态机(CPSM) ..... 523
图 23-10 SDIO 命令传输 ..... 523
图 23-11 数据路径 ..... 525
图 23-12 数据路径状态机(DPSM) ..... 526
图 24-1 CAN 网络拓扑结构 ..... 562
图 24-2 双 CAN 框图 ..... 563
图 24-3 bxCAN 工作模式 ..... 565
图 24-4 静默模式下的 bxCAN ..... 565
图 24-5 环回模式下的 bxCAN ..... 566
图 24-6 组合模式下的 bxCAN ..... 566
图 24-7 发送邮箱状态 ..... 568

图 24-8 接收 FIFO 状态.....569
图 24-9 筛选器组尺度配置 - 寄存器构成.....571
图 24-10 筛选器编号示例.....572
图 24-11 筛选机制-示例.....573
图 24-12 CAN 错误状态图.....574
图 24-13 位时序.....575
图 24-14 CAN 帧.....576
图 24-15 事件标志与中断产生.....577
图 24-16 每个邮箱由 4 个寄存器组成.....587
图 25-1 ETH 框图.....599
图 25-2 SMI 接口信号.....600
图 25-3 MDIO 时序和帧结构 - 写周期.....601
图 25-4 MDIO 时序和帧结构 – 读周期.....601
图 25-5 介质独立接口信号.....602
图 25-6 精简介质独立接口信号.....603
图 25-7 RMII 时钟源.....604
图 25-8 时钟方案.....604
图 25-9 地址字段格式.....606
图 25-10 MAC 帧格式.....607
图 25-11 带标记的 MAC 帧格式.....608
图 25-12 发送位序.....612
图 25-13 无冲突发送.....613
图 25-14 有冲突发送.....613
图 25-15 MII 和 RMII 模式下的帧发送.....614
图 25-16 接收位序.....617
图 25-17 无错误接收.....617
图 25-18 有错误接收.....617
图 25-19 出现假载波指示的接收.....618
图 25-20 MAC 内核中断屏蔽方案.....618
图 25-21 唤醒帧过滤寄存器.....621
图 25-22 网络时间同步.....624

图 25-23 使用精密校准方法更新系统时间.....625
图 25-24 PTP 触发输出与 TIM2 ITR1 的连接.....627
图 25-25 PPS 输出.....628
图 25-26 描述符环形结构与链接结构.....628
图 25-27 默认模式下的 TxDMA 操作.....631
图 25-28 OSF 模式下的 TxDMA 操作.....633
图 25-29 常规发送描述符.....634
图 25-30 增强的发送描述符.....639
图 25-31 DMA 接收操作.....641
图 25-32 常规 Rx DMA 描述符结构.....643
图 25-33 使能带有 IEEE1588 时间戳的增强的接收描述符字段格式.....648
图 25-34 中断方案.....651
图 25-35 以太网 MAC 远程唤醒帧过滤寄存器(ETH_MACRWUFFR).....661
图 26-1 USB-BlockDiagram.....697
图 26-2 USB Interrupt hierarchy.....706
图 31-1 软件初始化流程.....774
图 34-1 LTDC 框图.....813
图 34-2 LTDC 同步时序.....814
图 34-3 层窗口可编程参数.....817
图 34-4 两层与背景混合.....819
图 34-5 中断事件.....820

# 1. 文档约定

在文档写作中黑色字体表明已经确定的内容，尚未完全确定的内容以蓝色字体表示，有疑问或需要醒目标示的内容则以红色字体表示。

另外文中会用到许多数值的缩写遵循约定俗成的标准我们规定如下：

在描述数据容量如 RAM 容量时 1K 代表 1024，1M 代表 1,048,576；如 1K ARL 表项表示 1024 个 ARL 表项。

在描述其他数据如频率数据速率等时 1K 代表 1000，1M 代表 1,000,000，1G 代表 1,000,000,000；如 133MHz 表示 133,000,000Hz，100Mbps 表示 100,000,000bps。

在对寄存器的描述中使用了下列缩写：

## 1.1. 寄存器相关缩写词列表

寄存器缩略描述表：

表 1-1 寄存器相关缩写词

| 缩写词 | 说明 |
|---|---|
| 读/写（rw） | 软件可以读写这些位。 |
| 只读（r） | 软件只能读取这些位。 |
| 只写（w） | 软件只能写入该位。读取该位时将返回复位值。 |
| 读取/清零（rc_w1） | 软件可以读取该位，也可以通过写入 1 将该位清零。写入“0”对该位的值无影响。 |
| 读取/清零（rc_w0） | 软件可以读取该位，也可以通过写入 0 将该位清零。写入“1”对该位的值无影响。 |
| 读取/读取清零（rc_r） | 软件可以读取该位。读取该位时，将自动清零。写入“0”对该位的值无影响。 |
| 读取/置位（rs） | 软件可以读取该位，也可将其置 1。写入“0”对该位的值无影响。 |
| 保留（res） | 保留位，必须保持复位值。 |

## 1.2. 术语、定义及缩略语

表 1-2 术语、定义及缩略语

| Abbreviations | Full Spelling |
|---|---|
| AES | Advanced Encryption Standard |
| AHB | Advanced High Performance Bus |
| ANSI | American National Standards Institute |
| APB | Advanced Peripheral Bus |
| CBC | Cipher Block Chaining |
| CPHA | Clock Phase |
| CPOL | Clock Polarity |
| CPU | Central Processing Unit |
| CRC | Cyclic Redundancy Check |
| DEMA | Differential Electromagnetic Analysis |
| DES | Data Encryption Standard |
| DPA | Differential Power Analysis |
| DFA | Differential Fault Analysis |
| EC | Elliptic Curve |

| Abbreviations | Full Spelling |
|---|---|
| ECB | Electronic Codebook |
| ECC | Error Code Correction |
| EMA | Electromagnetic Analysis |
| ETU | Elementary Time Unit |
| FA | Fault Analysis |
| FD | Frequency Detector |
| FIFO | First In First Out |
| FIPS | Federal Information Processing Standards |
| GD | Glitch Detector |
| GPIO | General Purpose Inputs/Outputs |
| ISO/IEC 7816 | International Standard ISO/IEC7816 module |
| LD | Light Detector |
| MAC | Message Authentication Code |
| N/A | Not Applicable |
| NMI | Non Maskable Interrupt |
| NIST | National Institute of Standards and Technology |
| NVIC | Nested Vector Interrupt Controller |
| NVM | Non-Volatile Memory |
| OTP | One Time Programmable |
| PKE | Public Key Algorithm Acceleration Engine |
| PKCS | Public Key Cryptography Standards |
| RAM | Random Access Memory |
| RNG | Random Number Generator |
| RO | Read Only |
| RW | Read and Write |
| RSA | Rivest, Shamir, Adleman Algorithm |
| SCP | Symmetric Algorithm Co-Processor |
| SHA | Security Hash Algorithm |
| SPA | Simple Power Analysis |
| SPI | Serial Peripheral Interface |
| TD | Temperature Detector |
| TRNG | True Random Number Generator |
| VD | Voltage Detector |
| WDT | Watchdog Timer |
| WO | Write Only |

# 2. 存储器和系统架构

## 2.1. 系统架构

MH2435 采用 32 位多层总线结构，该结构可使系统中的多个主机和从机之间的并行通信成为可能。多层总线结构包括一个 AHB 互联矩阵、两个 AHB 总线和两个 APB 总线。

## 2.2. 存储器映射

程序存储器、数据存储器、寄存器和 I/O 端口排列在同一个顺序的 4 GB 地址空间内。

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

下表显示了芯片的存储器映射，包括代码、SRAM 和外设区域。

表 2-1 存储映射表

| 区域 | 外设 | 总线 | 地址范围 |
|---|---|---|---|
| | LTDC | AHB | 0x4002 CC00 - 0x4002 CFFF |
| | DCACHE | | 0x4002 C400 - 0x4002 C7FF |
| | ICACHE | | 0x4002 C000 - 0x4002 C3FF |
| | DMA2D | | 0x4002 B000 - 0x4002 BBFF |
| | DMA2 | | 0x4002 6400 - 0x4002 67FF |
| | DMA1 | | 0x4002 6000 - 0x4002 63FF |
| | XFC | | 0x4002 3C00 - 0x4002 3FFF |
| | SDRAM_REG | AHB3 | 0xA000 2000 - 0xA000 2FFF |
| | CRYPT_ASYMC | AHB2 | 0x5008 4000 - 0x5008 4FFF |
| | OTP | | 0x5007 6000 - 0x5007 67FF |
| | Quad-SPI | | 0x5007 5400 - 0x5007 57FF |
| | CRYPT_SYMC | | 0x5007 3000 - 0x5007 3FFF |
| | TRNG | | 0x5006 0800 - 0x5006 0BFF |
| | USB1 | | 0x5000 0000 - 0x5003 FFFF |
| | USB2 | AHB1 | 0x4004 0000 - 0x4007 FFFF |
| | ETH | | 0x4002 9000 - 0x4002 93FF |
| | | | 0x4002 8C00 - 0x4002 8FFF |
| | | | 0x4002 8800 - 0x4002 8BFF |
| | | | 0x4002 8400 - 0x4002 87FF |
| | | | 0x4002 8000 - 0x4002 83FF |
| | RCC | | 0x4002 3800 - 0x4002 3BFF |
| | CRC | | 0x4002 3000 - 0x4002 33FF |
| | JPEG Decoder | | 0x4002 2800 - 0x4002 2BFF |
| | GPIOI | | 0x4002 2000 - 0x4002 23FF |
| | GPIOH | | 0x4002 1C00 - 0x4002 1FFF |
| | GPIOG | | 0x4002 1800 - 0x4002 1BFF |
| | GPIOF | | 0x4002 1400 - 0x4002 17FF |
| | GPIOE | | 0x4002 1000 - 0x4002 13FF |
| | GPIOD | | 0x4002 0C00 - 0x4002 0FFF |

| 区域 | 外设 | 总线 | 地址范围 |
|---|---|---|---|
| | GPIOC | | 0x4002 0800 - 0x4002 0BFF |
| | GPIOB | | 0x4002 0400 - 0x4002 07FF |
| | GPIOA | | 0x4002 0000 - 0x4002 03FF |
| | PHERCFG | APB2 | 0x4001 6400 - 0x4001 67FF |
| | SPI5 | | 0x4001 5000 - 0x4001 53FF |
| | TIM11 | | 0x4001 4800 - 0x4001 4BFF |
| | TIM10 | | 0x4001 4400 - 0x4001 47FF |
| | TIM9 | | 0x4001 4000 - 0x4001 43FF |
| | EXTI | | 0x4001 3C00 - 0x4001 3FFF |
| | SYSCFG | | 0x4001 3800 - 0x4001 3BFF |
| | SPI4 | | 0x4001 3400 - 0x4001 37FF |
| | SPI1 | | 0x4001 3000 - 0x4001 33FF |
| | SDIO | | 0x4001 2C00 - 0x4001 2FFF |
| | ADC | | 0x4001 2000 - 0x4001 23FF |
| | USART6 | | 0x4001 1400 - 0x4001 17FF |
| | USART1 | | 0x4001 1000 - 0x4001 13FF |
| | TIM8 | | 0x4001 0400 - 0x4001 07FF |
| | TIM1 | | 0x4001 0000 - 0x4001 03FF |
| | UART7 | APB1 | 0x4000 7800 - 0x4000 7BFF |
| | DAC | | 0x4000 7400 - 0x4000 77FF |
| | PWR | | 0x4000 7000 - 0x4000 73FF |
| | CAN2 | | 0x4000 6800 - 0x4000 6BFF |
| | CAN1 | | 0x4000 6400 - 0x4000 67FF |
| | I2C3 | | 0x4000 5C00 - 0x4000 5FFF |
| | I2C2 | | 0x4000 5800 - 0x4000 5BFF |
| | I2C1 | | 0x4000 5400 - 0x4000 57FF |
| | UART4 | | 0x4000 4C00 - 0x4000 4FFF |
| | USART3 | | 0x4000 4800 - 0x4000 4BFF |
| | USART2 | | 0x4000 4400 - 0x4000 47FF |
| | I2S3ext | | 0x4000 4000 - 0x4000 43FF |
| | SPI3/I2S3 | | 0x4000 3C00 - 0x4000 3FFF |
| | SPI2/I2S2 | | 0x4000 3800 - 0x4000 3BFF |
| | I2S2ext | | 0x4000 3400 - 0x4000 37FF |
| | IWDG | | 0x4000 3000 - 0x4000 33FF |
| | WWDG | | 0x4000 2C00 - 0x4000 2FFF |
| | TIM14 | | 0x4000 2000 - 0x4000 23FF |
| | TIM13 | | 0x4000 1C00 - 0x4000 1FFF |
| | TIM12 | | 0x4000 1800 - 0x4000 1BFF |
| | TIM7 | | 0x4000 1400 - 0x4000 17FF |
| | TIM6 | | 0x4000 1000 - 0x4000 13FF |
| | TIM5 | | 0x4000 0C00 - 0x4000 0FFF |
| | TIM4 | | 0x4000 0800 - 0x4000 0BFF |
| | TIM3 | | 0x4000 0400 - 0x4000 07FF |
| | TIM2 | | 0x4000 0000 - 0x4000 03FF |
| SRAM | SRAM1 | AHB | 0x2000 0000 - 0x2002 7FFF |
| | SRAM2 | | 0x2002 8000 - 0x2002 FFFF |
| | SRAM3 | | 0x2003 0000 - 0x200F FFFF |

| 区域 | 外设 | 总线 | 地址范围 |
|---|---|---|---|
| Code | XFC | AHB | 0x0800 0000 - 0x0FFF FFFF |

### 2.2.1. 嵌入式 SRAM

系统 SRAM 可按字节、半字（16 位）或全字（32 位）访问。读写操作以 CPU 速度执行，且等待周期为 0。系统 SRAM 分为三个块，可以被同时访问，地址范围及容量如下表所示。

表 2-2 SRAM 模块构成

| 模块 | 容量 | 地址范围 | MEM_MODE = 0001 | MEM_MODE = 0010 |
|---|---|---|---|---|
| SRAM1 | 160KB | 0x2000 0000 - 0x2002 7FFF | | |
| SRAM2 | 32KB | 0x2002 8000 - 0x2002 FFFF | | |
| SRAM3 | 832KB | 0x2003 0000 - 0x200F FFFF | | |

### 2.2.2. 位段

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

外设寄存器和 SRAM 均映射到一个位段区域，这样可实现单个位段的读写操作。

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

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

其中：

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

例子：

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

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

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

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

# 3. 电源控制器(PWR)

## 3.1. 电源

芯片的工作电压 (VDD33) 要求介于 2.7 V 到 3.6 V 之间。

当主电源 VDD33 断电时，可通过 VBAT33 为实时时钟 (RTC)供电。

## 3.2. 电源监控器

### 3.2.1. 上电复位(POR)和掉电复位(PDR)

芯片内部集成有 POR/PDR 电路，可以从 2.7 V 开始正常工作。

当 VDD33/AVD33 低于指定阈值 VPOR/PDR 时，器件无需外部复位电路便会保持复位状态。

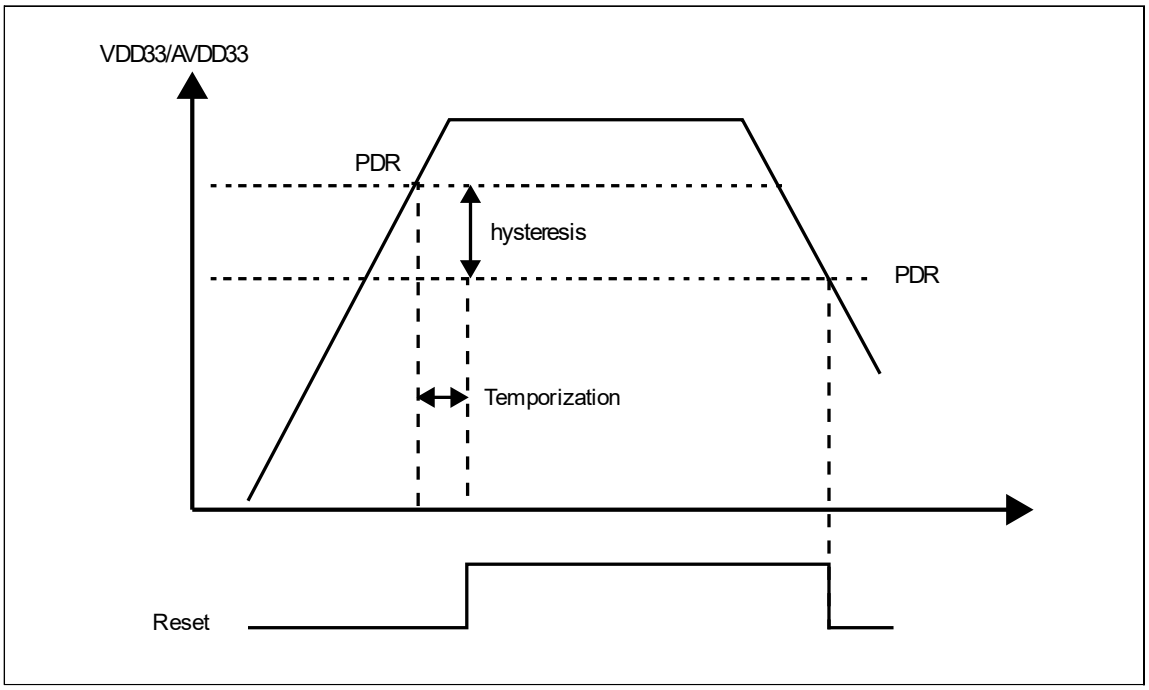



图 3-1 上电复位/掉电复位波形

### 3.2.2. 可编程电压检测器(PVD)

可以使用 PVD 监视 VDD33 电源，将其与 PWR 控制寄存器 (PWR_CR) 中 PLS[2:0] 位所选的阈值进行比较。

通过设置 PVDE 位来使能 PVD。

PWR 状态寄存器 (PWR_CSR) 中提供了 PVDO 标志，用于指示 VDD33 是大于还是小于 PVD 阈值。该事件内部连接到 EXTI 线 16，如果通过 EXTI 寄存器使能，则可以产生中断。当 VDD33 降至 PVD 阈值以下以及/或者当 VDD 升至 PVD 阈值以上时，可以产生 PVD 输出中断，具体取决于 EXTI 线 16 上升沿/下降沿的配置。该功能的用处之一就是可以在中断服务程序中执行紧急关闭系统的任务。

## 3.3. 低功耗模式

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

器件有三个低功耗模式：

- 睡眠模式（内核停止，外设保持运行）
- 停止模式（所有时钟都停止）
- 待机模式（1.1 V 域断电）
- 掉电模式

此外，可通过下列方法之一降低运行模式的功耗：

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

表 3-1 低功耗模式汇总

| 模式名称 | 进入 | 唤醒 | 对 1.1 V 域时钟的影响 | 对 VDD33 域时钟的影响 |
|---|---|---|---|---|
| 睡眠 | WFI | 任意中断 | CPU CLK 关闭<br>对其它时钟或模拟时钟源无影响 | 无 |
| | WFE | 唤醒事件 | | |
| 停止 | PDDS 和 LPDS 位 + SLEEPDEEP 位 + WFI 或 WFE | 1). 任意 EXTI 线（在 EXTI 寄存器中配置，内部线和外部线）<br>2). RTC、IWDG、RSTN、USB1/2 Suspend Exit、ENET 唤醒 | 所有 1.1 V 域时钟都关闭 | HSI 和 HSE 振荡器关闭 |
| 待机 | PDDS 位 + SLEEPDEEP 位 + WFI 或 WFE | 1). PA0<br>2). RTC<br>3). IWDG、RSTN 唤醒<br>4). PA7、PD11 高电平；PA14、PE13 低电平唤醒 | 所有 1.1 V 域时钟都关闭 | HSI 和 HSE 振荡器关闭 |
| 掉电 | PD_MODE_EN 位+ PDDS 位 + SLEEPDEEP 位 + WFI 或 WFE | 1). RSTN 唤醒<br>2). PA7、PD11 高电平；PA14、PE13 低电平唤醒 | 所有 1.1 V 域时钟都关闭 | HSI 和 HSE 振荡器关闭 |

### 3.3.1. 降低系统时钟

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

### 3.3.2. 外设时钟门控

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

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

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

### 3.3.3. 睡眠模式

#### 进入睡眠模式

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

- 立即休眠：如果 SLEEPONEXIT 位清零，MCU 将在执行 WFI 或 WFE 指令时立即进入睡眠模式。
- 退出时休眠：如果 SLEEPONEXIT 位置 1，MCU 将在退出优先级最低的 ISR 时立即进入睡眠模式。

#### 退出睡眠模式

如果使用 WFI 指令进入睡眠模式, 则嵌套向量中断控制器 (NVIC) 确认的任意外设中断都会将器件从睡眠模式唤醒。

如果使用 WFE 指令进入睡眠模式, MCU 将在有事件发生时立即退出睡眠模式。唤醒事件可通过以下方式产生：

- 在外设的控制寄存器使能一个中断，但不在 NVIC 中使能，同时使能内核系统控制寄存器中的 SEVONPEND 位。当 MCU 从 WFE 恢复时，需要清除相应外设的中断挂起位和外设 NVIC 中断通道挂起位（在 NVIC 中断清除挂起寄存器中）。
- 配置一个外部或内部 EXTI 线为事件模式。当 CPU 从 WFE 恢复时，因为对应事件线的挂起位没有被置位，不必清除相应外设的中断挂起位或 NVIC 中断通道挂起位。

由于没有在进入/退出中断时浪费时间，此模式下的唤醒时间最短。

### 3.3.4. 停止模式

停止模式基于内核深度睡眠模式与外设时钟门控。VDD11 既可以配置为正常模式，也可以配置为低功耗模式。在停止模式下， 1.1V 域中的所有时钟都会停止， PLL、HSI 和 HSE RC 振荡器也被禁止，内部 SRAM 可配置是否保持。

将 XFC->XFC_LPCFGR 寄存器中的 FDPD 位置 1 后，QSPI Flash 还会在器件进入停止模式时进入掉电模式。QSPI Flash 处于掉电模式时，将器件从停止模式唤醒将需要额外的启动延时。

表 3-2 停止工作模式

| 停止模式 | LPDS 位 | FDS 位 | 唤醒延迟 |
|---|---|---|---|
| STOP MR | 0 | 0 | HSI RC 启动时间 |
| STOP MR-FDPD | 0 | 1 | HSI RC 启动时间 +<br>Flash 从掉电模式唤醒的时间 |
| STOP LP | 1 | 0 | HSI RC 启动时间 +<br>VDD11 从 LP 模式唤醒的时间 |
| STOP LP-FDPD | 1 | 1 | HSI RC 启动时间 +<br>Flash 从掉电模式唤醒的时间 +<br>VDD11 从 LP 模式唤醒的时间 |

#### 进入停止模式

要进一步降低停止模式的功耗, 可将内部 VDD11 设置为低功耗模式。通过 PWR 电源控制寄存器 (PWR_CR) 的 LPDS 位进行配置。

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

如果正在访问 APB 域，停止模式的进入则延迟到 APB 访问结束后执行。

在停止模式下，可以通过对各控制位进行编程来选择以下功能：

- 独立的看门狗 (IWDG)：IWDG 通过写入其密钥寄存器来启动，而且一旦启动便无法停止，除非复位。
- 外部 32.768 kHz 振荡器 (LSE OSC)：通过 RCC 备份域控制寄存器 (RCC_BDCR) 中的 LSEON 位进行配置。

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

**退出停止模式**

通过发出中断或唤醒事件退出停止模式时，将选择 HSI RC 振荡器作为系统时钟。

当 VDD11 在低功耗模式下工作时，将器件从停止模式唤醒将需要额外的延时。在停止模式下一直开启 VDD11 正常模式虽然可以缩短启动时间，但功耗却增大。

### 3.3.5. 待机模式

待机模式下可达到更低功耗。待机模式基于内核深度睡眠模式，其中 VDD11 被关闭。因此 1.1V 域断电，PLL、 HSI 振荡器和 HSE 振荡器也将关闭。除备份域和待机电路中的寄存器外，SRAM 和寄存器内容都将丢失。

**进入待机模式**

在待机模式下，可以通过对各控制位进行编程来选择以下功能：

- 独立的看门狗 (IWDG)：IWDG 通过写入其密钥寄存器来启动。而且一旦启动便无法停止，除非复位。
- 外部 32.768 kHz 振荡器 (LSE OSC)：通过 RCC 备份域控制寄存器 (RCC_BDCR) 中的 LSEON 位进行配置。

**退出待机模式**

检测到外部复位(RSTN 引脚)、IWDG 复位、WKUP 引脚上升沿、RTC 闹钟、PA7/PD11 高电平、PA14/PE13 低电平时，微控制器退出待机模式。从待机模式唤醒后，除 PWR 状态寄存器 (PWR_CSR) 外，所有寄存器都将复位。

从待机模式唤醒后，程序将按照复位后的方式重新执行。PWR 状态寄存器 (PWR_CSR) 中的 SBF 状态标志指示 MCU 已处于待机模式。

**待机模式下的 IO 状态**

在待机模式下，除以下各部分以外，所有 I/O 引脚都处于高阻态：

- 复位引脚（仍可用）
- WKUP 引脚 (PA0)（如果使能）
- PA7/PD11 下拉输入
- PA14/PE13 上拉输入

### 3.3.6. 掉电模式

掉电模式下可达到最低功耗。掉电模式基于内核深度睡眠模式，其中全部 VDD11 掉电。因此 PLL、HSI 振荡器和 HSE 振荡器也将关闭。除备份域和掉电模式唤醒电路外，SRAM 和寄存器内容都将丢失。

**进入掉电模式**

在掉电模式下，可以通过对各控制位进行编程来选择以下功能：

- 外部 32.768 kHz 振荡器 (LSE OSC)：通过 RCC 备份域控制寄存器 (RCC_BDCR) 中的 LSEON 位进行配置。

**退出掉电模式**

检测到外部复位(RSTN 引脚)、PA7/PD11 高电平、PA14/PE13 低电平时，微控制器退出掉电模式。从掉电模式唤醒后，所有寄存器都将复位。

从掉电模式唤醒后，程序将按照复位后的方式重新执行。PWR 状态寄存器 (PWR_CSR) 中的 PDF 状态标志指示 MCU 已处于掉电模式。

**掉电模式下的 IO 状态**

在待机模式下，除以下各部分以外，所有 I/O 引脚都处于高阻态：

- 复位引脚（仍可用）
- PA7/PD11 下拉输入
- PA14/PE13 上拉输入

## 3.4. PWR 寄存器

### 3.4.1. PWR 控制寄存器(PWR_CR)

地址偏移： 0x00

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 |
|---|---|---|---|---|---|---|
| PD_OTP | PD_CRYPT | Reserved | PD_DMA2D | PD_SRAM1 | | PD_SRAM2 |
| rw | rw | | rw | rw | rw | rw |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 |
|---|---|---|---|---|---|---|
| PD_SRAM3 | | Reserved | | | | PD_CCM |
| rw | rw | | | | | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 |
|---|---|---|---|---|---|---|
| PD_ROM | Reserved | PD_MEM | Reserved | | LPDS | Reserved |
| rw | | rw | | | rw | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|
| PLS | | | PVDE | CSBF | CWUF | PDDS |
| rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31 | PD_OTP | OTP 模块电源控制<br>0：OTP 模块上电<br>1：OTP 模块掉电 |
| 位 30 | PD_CRYPT | CRYPT 模块电源控制<br>0：CRYPT 模块上电<br>1：CRYPT 模块掉电 |
| 位 29 | 保留 | 必须保持复位值 |
| 位 28 | PD_DMA2D | DMA2D 模块电源控制<br>0：DMA2D 模块上电<br>1：DMA2D 模块掉电 |
| 位 27:26 | PD_SRAM1 | STOP 模式下 SRAM1 电源控制<br>00, 11：SRAM1 上电<br>01：SRAM1 省电保持<br>10：SRAM1 掉电 |
| 位 25:24 | PD_SRAM2 | STOP 模式下 SRAM2 电源控制<br>00, 10, 11：SRAM2 上电<br>01：SRAM2 省电保持 |
| 位 23:22 | PD_SRAM3 | STOP 模式下 SRAM3 电源控制<br>00, 11：SRAM3 上电<br>01：SRAM3 省电保持<br>10：SRAM3 掉电 |
| 位 21:18 | 保留 | 必须保持复位值 |
| 位 17:16 | PD_CCM | STOP 模式下 CCM 电源控制<br>00, 10, 11：CCM 上电<br>01：CCM 省电保持 |
| 位 15 | PD_ROM | ROM 模块电源控制<br>0：ROM 模块上电<br>1：ROM 模块掉电 |
| 位 14 | 保留 | 必须保持复位值 |
| 位 13 | PD_MEM | LTDC、JPEG Decoder、DMA1、DMA2 模块 MEM 电源控制<br>0：LTDC/JPEG Decoder/DMA1/DMA2 模块 MEM 上电<br>1：LTDC/JPEG Decoder/DMA1/DMA2 模块 MEM 省电保持<br>*注意：先关闭相关模块时钟后再对该寄存器进行配置* |
| 位 12:11 | 保留 | 必须保持复位值 |
| 位 10 | LPDS | STOP 模式 LPLDO 切换控制<br>0：进入 Stop 时，不切换 LPLDO<br>1：进入 Stop 时，切换成 LPLDO |
| 位 9 | 保留 | 必须保持复位值 |
| 位 8 | DBP | 保护 RCC_BDCR 寄存器和 PWR_CSR 寄存器的 BRE 位，防止误写访问。对这些寄存器的写访问，必须将此位设置为启用。<br>0：访问 RCC_BDCR 寄存器和 PWR_CSR 寄存器的 BRE 位已禁用<br>1：访问 RCC_BDCR 寄存器和 PWR_CSR 寄存器的 BRE 位已启用 |
| 位 7:5 | PLS | PVD 级别选择<br>这些位由软件写入，用于选择电压检测器检测的电压阈值<br>000：2.01 V<br>001：2.1 V<br>010：2.19 V<br>011：2.28 V |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 100：2.37 V<br>101：2.46 V<br>110：2.55 V<br>111：2.64 V |
| 位 4 | PVDE | 使能电源电压检测器<br>此位由软件置 1 和清零。<br>0：禁止 PVD<br>1：使能 PVD |
| 位 3 | CSBF | 将待机标志清零<br>此位始终读为 0。<br>0：无操作<br>1：写 1 将 SBF 待机标志清零。 |
| 位 2 | CWUF | 将唤醒标志清零<br>此位始终读为 0。<br>0：无操作<br>1：写 1 操作 2 个系统时钟周期后将 WUF 唤醒标志清零 |
| 位 1:0 | PDDS | 由软件设置和清除。它与 LPDS 位一起工作。<br>00：当 CPU 进入 DeepSleep 时，进入 Stop 模式<br>01：当 CPU 进入 DeepSleep 时，进入 Standby 模式<br>10, 11：当 CPU 进入 DeepSleep 时，进入 PowerDown 模式 |

### 3.4.2. PWR 状态寄存器(PWR_CSR)

地址偏移： 0x04

复位值： 0x5806 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CRYPT_POF | Reserved | DMA2D_POF | SRAM_POF | Reserved | | | |
| ro | | ro | ro | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | STANDBY_IWDG_PAUSE_EN | STOP_IWDG_PAUSE_EN | PD_MODE_EN |
| | | | | | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| PDMODE_FLAG_CLR | Reserved | | | | | | PA0_WKEN |
| rw | | | | | | | rw |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PE13_WKEN | PA14_WKEN | PD11_WKEN | PA7_WKEN | PDF | PVDO | SBF | WUF |
| rw | rw | rw | rw | ro | ro | ro | ro |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31 | CRYPT_POF | CRYPT 模块电源状态<br>0：CRYPT 模块未上电<br>1：CRYPT 模块已上电 |
| 位 30 | 保留 | 必须保持复位值 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 29 | DMA2D_POF | DMA2D 模块电源状态<br>0：DMA2D 模块未上电<br>1：DMA2D 模块已上电 |
| 位 28 | SRAM_POF | SRAM 模块电源状态<br>0：SRAM 模块未上电<br>1：SRAM 模块已上电 |
| 位 27:19 | 保留 | 必须保持复位值 |
| 位 18 | STANDBY_IWDG_PAUSE_EN | STANDBY 模式下停止独立看门狗<br>0：STANDBY 模式下独立看门狗计数器仍继续工作<br>1：STANDBY 模式下独立看门狗计数器停止 |
| 位 17 | STOP_IWDG_PAUSE_EN | STOP 模式下停止独立看门狗<br>0：STOP 模式下独立看门狗计数器仍继续工作<br>1：STOP 模式下独立看门狗计数器停止 |
| 位 16 | PD_MODE_EN | PowerDown 模式使能<br>0：不使能 PowerDown 模式，当 PDDS=2、3，执行 WFI 或 WFE 后只进入 CPU Sleep 模式<br>1：使能 PowerDown 模式，当 PDDS=2、3，执行 WFI 或 WFE 后可进入 PowerDown 模式 |
| 位 15 | PDMODE_FLAG_CLR | 清除 PD_MODE_FLAG 信号，高脉冲有效，需软件置 1 清 0 |
| 位 14:9 | 保留 | 必须保持复位值 |
| 位 8 | PA0_WKEN | PA0 上升沿唤醒使能（Standby 有效）<br>0：关闭唤醒，PA0 作为普通 GPIO<br>1：使能唤醒，硬件强制 PA0 作为输入下拉。 |
| 位 7 | PE13_WKEN | PE13 低电平唤醒使能（Standby、PowerDown 有效）<br>0：关闭唤醒<br>1：使能唤醒 |
| 位 6 | PA14_WKEN | PA14 低电平唤醒使能（Standby、PowerDown 有效）<br>0：关闭唤醒<br>1：使能唤醒 |
| 位 5 | PD11_WKEN | PD11 高电平唤醒使能（Standby、PowerDown 有效）<br>0：关闭唤醒<br>1：使能唤醒 |
| 位 4 | PA7_WKEN | PA7 高电平唤醒使能（Standby、PowerDown 有效）<br>0：关闭唤醒<br>1：使能唤醒 |
| 位 3 | PDF | PowerDown 标志<br>0：未进入过 PowerDown 模式<br>1：进入过 PowerDown 模式 |
| 位 2 | PVDO | PVD 输出<br>此位通过硬件置 1 和清零。仅当通过 PVDE 位使能 PVD 时此位才有效。<br>0：VDD 高于 PLS[2:0] 位选择的 PVD 阈值。<br>1：VDD 低于 PLS[2:0] 位选择的 PVD 阈值。<br>*注意：PVD 在进入待机模式时停止。因此，进入待机模式或执行复位后，此位等于 0，直到 PVDE 位置 1。* |
| 位 1 | SBF | 待机标志<br>此位由硬件置 1，清零则只能通过 POR/PDR（上电复 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 位/掉电复位）或将 PWR_CR 寄存器中的 CSBF 位置 1 来实现。<br>0：未进入待机模式<br>1：在此次复位前进入待机模式 |
| 位 0 | WUF | 此位由硬件设置，并通过系统重置或在 PWR_CR 寄存器中设置 CWUF 位来清除。<br>0：未发生唤醒事件<br>1：从 PA0 接收到一个唤醒事件。<br>*注意：如果使能 PA0 引脚（将 PA0_WKEN 位置 1）时 PA0 引脚已为高电平，系统将检测到另一唤醒事件。* |

# 4. 嵌套向量中断控制器(NVIC)

## 4.1. NVIC 特性

嵌套向量中断控制器 NVIC 包含以下特性：

- 具有多达 110 个可屏蔽中断通道（不包括内核的 16 根中断线）；
- 16 个可编程优先级（使用了 4 位中断优先级）；
- 低延迟异常和中断处理；
- 电源管理控制；
- 系统控制寄存器的实现。

嵌套向量中断控制器 (NVIC) 和处理器内核接口紧密配合，可以实现低延迟的中断处理和晚到中断的高效处理。

包括内核异常在内的所有中断均通过 NVIC 进行管理。

## 4.2. 中断和异常向量

表 4-1 MH2435 向量表

| 位置 | 优先级 | 优先级类型 | 名称 | 说明 | 地址 |
|---|---|---|---|---|---|
| | - | - | - | 保留 | 0x0000 0000 |
| | -3 | 固定 | Reset | 复位 | 0x0000 0004 |
| | -2 | 固定 | NMI | 不可屏蔽中断。<br>RCC 时钟安全系统(CSS)连接到 NMI 向量。 | 0x0000 0008 |
| | -1 | 固定 | HardFault | 所有类型的错误 | 0x0000 000C |
| | 0 | 可设置 | MemManage | 存储器管理 | 0x0000 0010 |
| | 1 | 可设置 | BusFault | 预取指失败，存储器访问失败 | 0x0000 0014 |
| | 2 | 可设置 | UsageFault | 未定义的指令或非法状态 | 0x0000 0018 |
| | - | - | - | 保留 | 0x0000 001C<br>–<br>0x0000 002B |
| | 3 | 可设置 | SVCall | 通过 SWI 指令调用的系统服务 | 0x0000 002C |
| | 4 | 可设置 | Debug Monitor | 调试监控器 | 0x0000 0030 |
| | - | - | - | 保留 | 0x0000 0034 |
| | 5 | 可设置 | PendSV | 可挂起的系统服务 | 0x0000 0038 |
| | 6 | 可设置 | SysTick | 系统嘀嗒定时器 | 0x0000 003C |
| 0 | 7 | 可设置 | WWDG | 窗口看门狗中断 | 0x0000 0040 |
| 1 | 8 | 可设置 | PVD | 连接到 EXTI 线的可编程电压检测(PVD)中断 | 0x0000 0044 |
| 2 | 9 | 可设置 | - | 保留 | 0x0000 0048 |
| 3 | 10 | 可设置 | - | 保留 | 0x0000 004C |
| 4 | 11 | 可设置 | XFC | XIP QFlash 全局中断 | 0x0000 0050 |
| 5 | 12 | 可设置 | RCC | RCC 全局中断 | 0x0000 0054 |
| 6 | 13 | 可设置 | EXTI0 | EXTI 线 0 中断 | 0x0000 0058 |
| 7 | 14 | 可设置 | EXTI1 | EXTI 线 1 中断 | 0x0000 005C |
| 8 | 15 | 可设置 | EXTI2 | EXTI 线 2 中断 | 0x0000 0060 |

| 位置 | 优先级 | 优先级类型 | 名称 | 说明 | 地址 |
|---|---|---|---|---|---|
| 9 | 16 | 可设置 | EXTI3 | EXTI 线 3 中断 | 0x0000 0064 |
| 10 | 17 | 可设置 | EXTI4 | EXTI 线 4 中断 | 0x0000 0068 |
| 11 | 18 | 可设置 | DMA1_Stream0 | DMA1 流 0 全局中断 | 0x0000 006C |
| 12 | 19 | 可设置 | DMA1_Stream1 | DMA1 流 1 全局中断 | 0x0000 0070 |
| 13 | 20 | 可设置 | DMA1_Stream2 | DMA1 流 2 全局中断 | 0x0000 0074 |
| 14 | 21 | 可设置 | DMA1_Stream3 | DMA1 流 3 全局中断 | 0x0000 0078 |
| 15 | 22 | 可设置 | DMA1_Stream4 | DMA1 流 4 全局中断 | 0x0000 007C |
| 16 | 23 | 可设置 | DMA1_Stream5 | DMA1 流 5 全局中断 | 0x0000 0080 |
| 17 | 24 | 可设置 | DMA1_Stream6 | DMA1 流 6 全局中断 | 0x0000 0084 |
| 18 | 25 | 可设置 | ADC | ADC1、 ADC2 和 ADC3 全局中断 | 0x0000 0088 |
| 19 | 26 | 可设置 | CAN1_TX | CAN1 TX 中断 | 0x0000 008C |
| 20 | 27 | 可设置 | CAN1_RX0 | CAN1 RX0 中断 | 0x0000 0090 |
| 21 | 28 | 可设置 | CAN1_RX1 | CAN1 RX1 中断 | 0x0000 0094 |
| 22 | 29 | 可设置 | CAN1_SCE | CAN1 SCE 中断 | 0x0000 0098 |
| 23 | 30 | 可设置 | EXTI9_5 | EXTI 线 [9:5] 中断 | 0x0000 009C |
| 24 | 31 | 可设置 | TIM1_BRK_TIM9 | TIM1 刹车中断和 TIM9 全局中断 | 0x0000 00A0 |
| 25 | 32 | 可设置 | TIM1_UP_TIM10 | TIM1 更新中断和 TIM10 全局中断 | 0x0000 00A4 |
| 26 | 33 | 可设置 | TIM1_TRG_COM_TIM11 | TIM1 触发和换相中断与 TIM11 全局中断 | 0x0000 00A8 |
| 27 | 34 | 可设置 | TIM1_CC | TIM1 捕获比较中断 | 0x0000 00AC |
| 28 | 35 | 可设置 | TIM2 | TIM2 全局中断 | 0x0000 00B0 |
| 29 | 36 | 可设置 | TIM3 | TIM3 全局中断 | 0x0000 00B4 |
| 30 | 37 | 可设置 | TIM4 | TIM4 全局中断 | 0x0000 00B8 |
| 31 | 38 | 可设置 | I2C1_EV | I2C1 事件中断 | 0x0000 00BC |
| 32 | 39 | 可设置 | I2C1_ER | I2C1 错误中断 | 0x0000 00C0 |
| 33 | 40 | 可设置 | I2C2_EV | I2C2 事件中断 | 0x0000 00C4 |
| 34 | 41 | 可设置 | I2C2_ER | I2C2 错误中断 | 0x0000 00C8 |
| 35 | 42 | 可设置 | SPI1 | SPI1 全局中断 | 0x0000 00CC |
| 36 | 43 | 可设置 | SPI2 | SPI2 全局中断 | 0x0000 00D0 |
| 37 | 44 | 可设置 | USART1 | USART1 全局中断 | 0x0000 00D4 |
| 38 | 45 | 可设置 | USART2 | USART2 全局中断 | 0x0000 00D8 |
| 39 | 46 | 可设置 | USART3 | USART3 全局中断 | 0x0000 00DC |
| 40 | 47 | 可设置 | EXTI15_10 | EXTI 线 [15:10] 中断 | 0x0000 00E0 |
| 41 | 48 | 可设置 | RTC | RTC 全局中断 | 0x0000 00E4 |
| 42 | 49 | 可设置 | USB1 WKUP | 连接到 EXTI 线的 USB1 唤醒中断 | 0x0000 00E8 |
| 43 | 50 | 可设置 | TIM8_BRK_TIM12 | TIM8 刹车中断和 TIM12 全局中断 | 0x0000 00EC |
| 44 | 51 | 可设置 | TIM8_UP_TIM13 | TIM8 更新中断和 TIM13 全局中断 | 0x0000 00F0 |
| 45 | 52 | 可设置 | TIM8_TRG_COM_TIM14 | TIM8 触发和换相中断与 TIM14 全局中断 | 0x0000 00F4 |
| 46 | 53 | 可设置 | TIM8_CC | TIM8 捕捉比较中断 | 0x0000 00F8 |
| 47 | 54 | 可设置 | DMA1_Stream7 | DMA1 流 7 全局中断 | 0x0000 00FC |
| 48 | 55 | 可设置 | - | 保留 | 0x0000 0100 |
| 49 | 56 | 可设置 | SDIO | SDIO 全局中断 | 0x0000 0104 |
| 50 | 57 | 可设置 | TIM5 | TIM5 全局中断 | 0x0000 0108 |
| 51 | 58 | 可设置 | SPI3 | SPI3 全局中断 | 0x0000 010C |
| 52 | 59 | 可设置 | UART4 | UART4 全局中断 | 0x0000 0110 |
| 53 | 60 | 可设置 | - | 保留 | 0x0000 0114 |

| 位置 | 优先级 | 优先级类型 | 名称 | 说明 | 地址 |
|---|---|---|---|---|---|
| 54 | 61 | 可设置 | TIM6_DAC | TIM6 全局中断，DAC1 和 DAC2 下溢错误中断 | 0x0000 0118 |
| 55 | 62 | 可设置 | TIM7 | TIM7 全局中断 | 0x0000 011C |
| 56 | 63 | 可设置 | DMA2_Stream0 | DMA2 流 0 全局中断 | 0x0000 0120 |
| 57 | 64 | 可设置 | DMA2_Stream1 | DMA2 流 1 全局中断 | 0x0000 0124 |
| 58 | 65 | 可设置 | DMA2_Stream2 | DMA2 流 2 全局中断 | 0x0000 0128 |
| 59 | 66 | 可设置 | DMA2_Stream3 | DMA2 流 3 全局中断 | 0x0000 012C |
| 60 | 67 | 可设置 | DMA2_Stream4 | DMA2 流 4 全局中断 | 0x0000 0130 |
| 61 | 68 | 可设置 | ETH | 以太网全局中断 | 0x0000 0134 |
| 62 | 69 | 可设置 | ETH_WKUP | 连接到 EXTI 线的以太网唤醒中断 | 0x0000 0138 |
| 63 | 70 | 可设置 | CAN2_TX | CAN2 TX 中断 | 0x0000 013C |
| 64 | 71 | 可设置 | CAN2_RX0 | CAN2 RX0 中断 | 0x0000 0140 |
| 65 | 72 | 可设置 | CAN2_RX1 | CAN2 RX1 中断 | 0x0000 0144 |
| 66 | 73 | 可设置 | CAN2_SCE | CAN2 SCE 中断 | 0x0000 0148 |
| 67 | 74 | 可设置 | USB1 | USB1 全局中断 | 0x0000 014C |
| 68 | 75 | 可设置 | DMA2_Stream5 | DMA2 流 5 全局中断 | 0x0000 0150 |
| 69 | 76 | 可设置 | DMA2_Stream6 | DMA2 流 6 全局中断 | 0x0000 0154 |
| 70 | 77 | 可设置 | DMA2_Stream7 | DMA2 流 7 全局中断 | 0x0000 0158 |
| 71 | 78 | 可设置 | USART6 | USART6 全局中断 | 0x0000 015C |
| 72 | 79 | 可设置 | I2C3_EV | I2C3 事件中断 | 0x0000 0160 |
| 73 | 80 | 可设置 | I2C3_ER | I2C3 错误中断 | 0x0000 0164 |
| 74 | 81 | 可设置 | USB2_EP1_OUT | USB2 端点 1 输出全局中断 | 0x0000 0168 |
| 75 | 82 | 可设置 | USB2_EP1_IN | USB2 端点 1 输入全局中断 | 0x0000 016C |
| 76 | 83 | 可设置 | USB2_WKUP | 连接到 EXTI 线的 USB2 唤醒中断 | 0x0000 0170 |
| 77 | 84 | 可设置 | USB2 | USB2 全局中断 | 0x0000 0174 |
| 78 | 85 | 可设置 | - | 保留 | 0x0000 0178 |
| 79 | 86 | 可设置 | CRYP_SYMC | CRYP 对称加密全局中断 | 0x0000 017C |
| 80 | 87 | 可设置 | TRNG | 真随机数发生器全局中断 | 0x0000 0180 |
| 81 | 88 | 可设置 | FPU | FPU 全局中断 | 0x0000 0184 |
| 82 | 89 | 可设置 | UART7 | UART7 全局中断 | 0x0000 0188 |
| 83 | 90 | 可设置 | - | 保留 | 0x0000 018C |
| 84 | 91 | 可设置 | SPI4 | SPI4 全局中断 | 0x0000 0190 |
| 85 | 92 | 可设置 | SPI5 | SPI5 全局中断 | 0x0000 0194 |
| 86 | 93 | - | - | 保留 | 0x0000 0198 |
| 87 | 94 | - | - | 保留 | 0x0000 019C |
| 88 | 95 | 可设置 | LCD-TFT | LCD-TFT 全局中断 | 0x0000 01A0 |
| 89 | 96 | - | - | 保留 | 0x0000 01A4 |
| 90 | 97 | 可设置 | DMA2D | DMA2D 全局中断 | 0x0000 01A8 |
| 91 | 98 | - | - | 保留 | 0x0000 01AC |
| 92 | 99 | - | - | 保留 | 0x0000 01B0 |
| 93 | 100 | - | - | 保留 | 0x0000 01B4 |
| 94 | 101 | - | - | 保留 | 0x0000 01B8 |
| 95 | 102 | - | - | 保留 | 0x0000 01BC |
| 96 | 103 | 可设置 | Quad-SPI | Quad-SPI 全局中断 | 0x0000 01C0 |
| 97 | 104 | - | - | 保留 | 0x0000 01C4 |
| 98 | 105 | 可设置 | JPEG Decoder | JPEG Decoder 全局中断 | 0x0000 01C8 |
| 99 | 106 | 可设置 | CRYP_ASYMC | CRYP 非对称加密全局中断 | 0x0000 01CC |

# 5. 外部中断和事件控制器(EXTI)

## 5.1. 外部中断/事件控制器 (EXTI)

外部中断/事件控制器包含多达 21 个用于产生事件/中断请求的边沿检测器。每根输入线都可单独进行配置，以选择类型（中断或事件）和相应的触发事件（上升沿触发、下降沿触发或边沿触发）。每根输入线还可单独屏蔽。挂起寄存器用于保持中断请求的状态线。

### 5.1.1. EXTI 主要特性

EXTI 控制器的主要特性如下：

- 每个中断/事件线上都具有独立的触发和屏蔽；
- 每个中断线都具有专用的状态位；
- 支持多达 21 个软件事件/中断请求；
- 检测脉冲宽度低于 APB2 时钟宽度的外部信号。

### 5.1.2. EXTI 框图

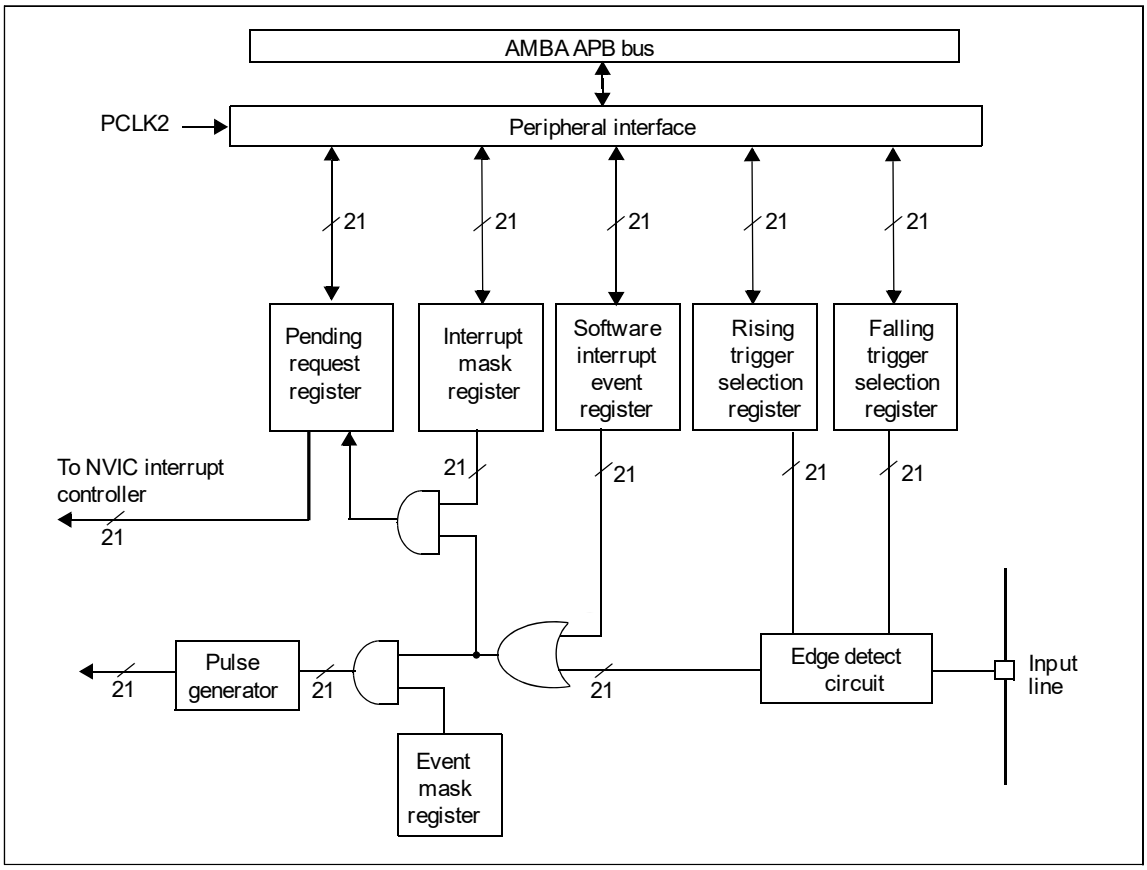



图 5-1 外部中断/事件控制器框图

### 5.1.3. 唤醒事件管理

芯片能够处理外部或内部事件来唤醒内核(WFE)。唤醒事件可通过以下方式产生：

- 在外设的控制寄存器使能一个中断，但不在 NVIC 中使能，同时使能系统控制寄存器中的 SEVONPEND 位。当 MCU 从 WFE 恢复时，需要清除相应外设的中断挂起位和外设 NVIC 中断通道挂起位(在 NVIC 中断清除挂起寄存器中)。
- 配置一个外部或内部 EXTI 线为事件模式，当 CPU 从 WFE 恢复后，因为对应事件线的挂起位没有被置位，不必清除相应外设的中断挂起位或 NVIC 中断通道挂起位。

### 5.1.4. 功能描述

要产生中断，必须先配置好并使能中断线。根据需要的边沿检测设置 2 个触发寄存器，同时在中断屏蔽寄存器的相应位写‘1’使能中断请求。当外部中断线上出现选定信号沿时，便会产生中断请求，对应的挂起位也会置‘1’。在挂起寄存器的对应位写‘1’，将清除该中断请求。

要产生事件，必须先配置好并使能事件线。根据需要的边沿检测通过设置 2 个触发寄存器，同时在事件屏蔽寄存器的相应位写‘1’允许事件请求。当事件线上出现选定信号沿时，便会产生事件脉冲，对应的挂起位不会置‘1’。

通过在软件中对软件中断/事件寄存器写‘1’，也可以产生中断/事件请求。

### 5.1.5. 硬件中断选择

要将一根线配置为中断源，请执行以下步骤：

- 配置相应的屏蔽位(EXTI_IMR)；
- 配置中断线的触发选择位(EXTI_RTSR 和 EXTI_FTSR)；
- 配置对应到外部中断控制器(EXTI)的 NVIC 中断通道的使能和屏蔽位，使得 21 个中断线中的请求可以被正确地响应。

### 5.1.6. 硬件事件选择

要将一根线配置为事件源，请执行以下步骤：

- 配置相应的屏蔽位(EXTI_IMR)；
- 配置事件线的触发选择位(EXTI_RTSR 和 EXTI_FTSR)。

### 5.1.7. 软件中断/事件的选择

可将这根线配置为软件中断/事件线。下面是产生软件中断的过程：

- 配置相应的屏蔽位(EXTI_IMR, EXTI_EMR)；
- 在软件中断寄存器设置相应的请求位(EXTI_SWIER)。

### 5.1.8. 外部中断/事件线映射

多个 GPIO 通过以下方式连接到 16 个外部中断/事件线：

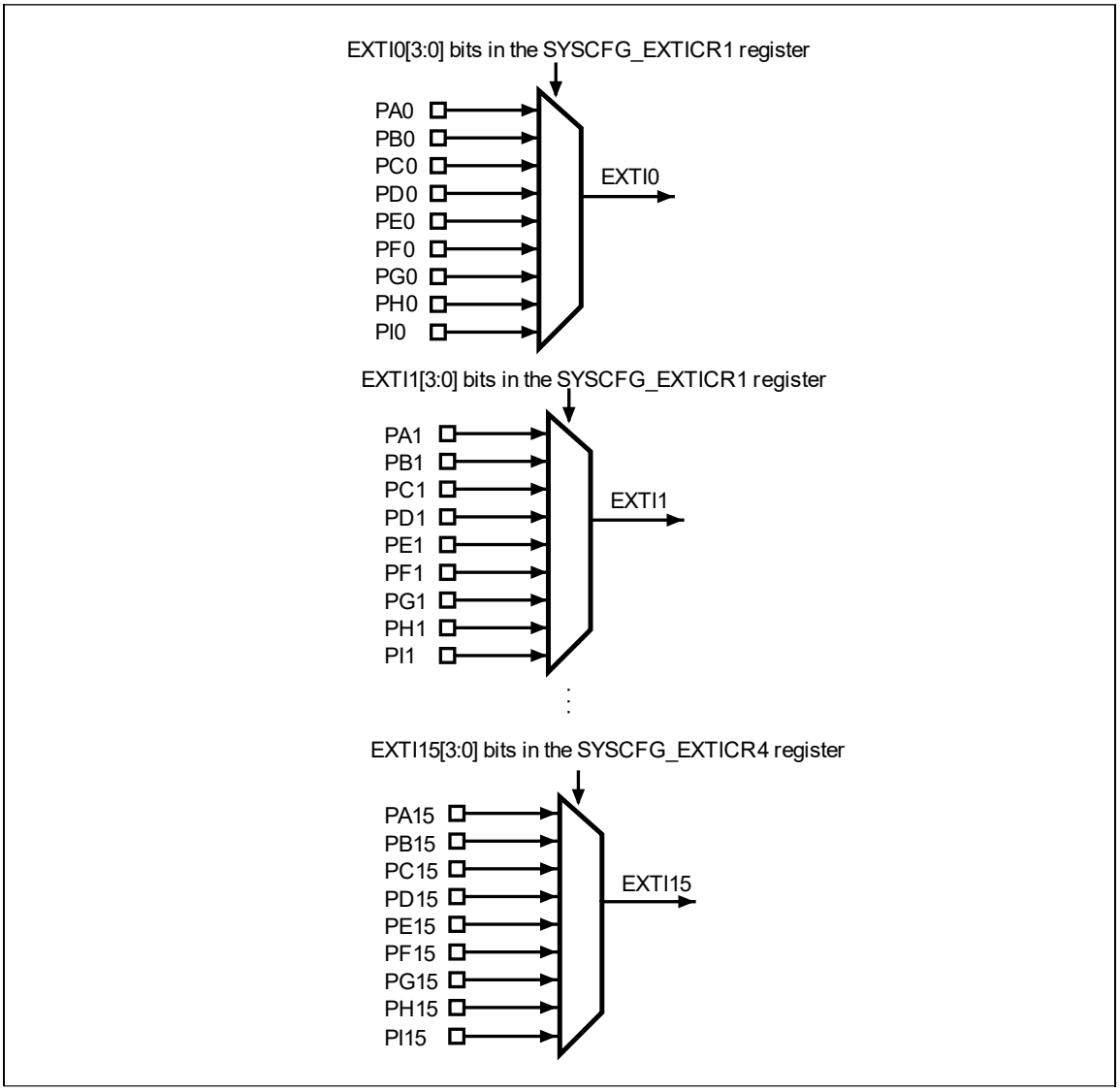



图 5-2 外部中断/事件 GPIO 映射

另外五根 EXTI 线连接方式如下:

- EXTI 线 16 连接到 PVD 输出;
- EXTI 线 17 连接到 RTC 中断事件;
- EXTI 线 18 连接到 USB1 唤醒事件;
- EXTI 线 19 连接到以太网唤醒事件;
- EXTI 线 20 连接到 USB2 唤醒事件。

## 5.2. EXTI 寄存器

本节对 EXTI 寄存器进行了详细介绍。

可通过字节（ 8 位)、半字（ 16 位）或字（ 32 位）对 EXTI 寄存器进行访问。

### 5.2.1. 中断屏蔽寄存器 (EXTI_IMR)

偏移地址：0x00

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | MR20 | MR19 | MR18 | MR17 | MR16 |
| | | | | | | | | | | | rw | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MR15 | MR14 | MR13 | MR12 | MR11 | MR10 | MR9 | MR8 | MR7 | MR6 | MR5 | MR4 | MR3 | MR2 | MR1 | MR0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:21 | 保留 | 必须保持复位值 |
| 位 20:0 | MRx | x 线上的中断屏蔽 (Interrupt mask on line x)<br>0：屏蔽来自 x 线的中断请求<br>1：开放来自 x 线的中断请求 |

### 5.2.2. 事件屏蔽寄存器 (EXTI_EMR)

偏移地址：0x04

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | MR20 | MR19 | MR18 | MR17 | MR16 |
| | | | | | | | | | | | rw | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MR15 | MR14 | MR13 | MR12 | MR11 | MR10 | MR9 | MR8 | MR7 | MR6 | MR5 | MR4 | MR3 | MR2 | MR1 | MR0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:21 | 保留 | 必须保持复位值 |
| 位 20:0 | MRx | x 线上的事件屏蔽 (Event mask on line x)<br>0：屏蔽来自 x 线的事件请求<br>1：开放来自 x 线的事件请求 |

### 5.2.3. 上升沿触发选择寄存器 (EXTI_RTSR)

偏移地址：0x08

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | TR20 | TR19 | TR18 | TR17 | TR16 |
| | | | | | | | | | | | rw | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TR15 | TR14 | TR13 | TR12 | TR11 | TR10 | TR9 | TR8 | TR7 | TR6 | TR5 | TR4 | TR3 | TR2 | TR1 | TR0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:21 | 保留 | 必须保持复位值 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 20:0 | TRx | x 线的上升沿触发事件配置位 (Rising trigger event configuration bit of linex)<br>0：禁止输入线上升沿触发（事件和中断）<br>1：允许输入线上升沿触发（事件和中断） |

*注意：外部唤醒线为边沿触发的，在这些线上不能出现毛刺信号。如果在向 EXTI_RTSR 寄存器写入值的同时外部中断线上产生上升沿，挂起位将被置位。*

*在同一中断线上，可以同时设置上升沿和下降沿触发。即任一边沿都可触发中断。*

### 5.2.4. 下降沿触发选择寄存器 (EXTI_FTSR)

偏移地址：0x0C

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Reserved | | | | | | | | | | | TR20 | TR19 | TR18 | TR17 | TR16 |
| | | | | | | | | | | | rw | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TR15 | TR14 | TR13 | TR12 | TR11 | TR10 | TR9 | TR8 | TR7 | TR6 | TR5 | TR4 | TR3 | TR2 | TR1 | TR0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:21 | 保留 | 必须保持复位值 |
| 位 20:0 | TRx | x 线的下降沿触发事件配置位 (Falling trigger event configuration bit of linex)<br>0：禁止输入线下降沿触发（事件和中断）<br>1：允许输入线下降沿触发（事件和中断） |

*注意：外部唤醒线为边沿触发的，在这些线上不能出现毛刺信号。如果在向 EXTI_FTSR 寄存器写入值的同时外部中断线上产生下降沿，挂起位将被置位。*

*在同一中断线上，可以同时设置上升沿和下降沿触发。即任一边沿都可触发中断。*

### 5.2.5. 软件中断事件寄存器 (EXTI_SWIER)

偏移地址： 0x10

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | SWIER20 | SWIER19 | SWIER18 | SWIER17 | SWIER16 |
| | | | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| SWIER15 | SWIER14 | SWIER13 | SWIER12 | SWIER11 | SWIER10 | SWIER9 | SWIER8 |
| rw | rw | rw | rw | rw | rw | rw | rw |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SWIER7 | SWIER6 | SWIER5 | SWIER4 | SWIER3 | SWIER2 | SWIER1 | SWIER0 |
| rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:21 | 保留 | 必须保持复位值 |
| 位 20:0 | SWIERx | 当该位为“0”时，写“1”将设置 EXTI_PR 中相应的挂起位。如果在 EXTI_IMR 和 EXTI_EMR 中允许产生该中断，则产生中断请求。<br>通过清除 EXTI_PR 的对应位（写入“1”），可以清除该位为“0”。 |

### 5.2.6. 挂起寄存器 (EXTI_PR)

偏移地址： 0x14

复位值：未定义

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | PR20 | PR19 | PR18 | PR17 | PR16 |
| | | | | | | | | | | | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PR15 | PR14 | PR13 | PR12 | PR11 | PR10 | PR9 | PR8 | PR7 | PR6 | PR5 | PR4 | PR3 | PR2 | PR1 | PR0 |
| rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:21 | 保留 | 必须保持复位值 |
| 位 20:0 | PRx | 挂起位 (Pending bit)<br>0：没有发生触发请求<br>1：发生了选择的触发请求<br>当在外部中断线上发生了选择的边沿事件，该位被置“1”。在此位中写入“1”可以清除它，也可以通过改变边沿检测的极性清除。 |

# 6. 复位和时钟控制(RCC)

## 6.1. 复位

共有三种类型的复位，分别为系统复位、电源复位和备份域复位。

### 6.1.1. 系统复位

除了时钟控制寄存器 CSR 中的复位标志和备份域中的寄存器外, 系统复位会将其它全部寄存器都复位为复位值。

只要发生以下事件之一，就会产生系统复位：

- NRST 引脚低电平（外部复位）；
- 窗口看门狗计数结束（WWDG 复位）；
- 独立看门狗计数结束（IWDG 复位）；
- 软件复位（SW 复位）。

### 6.1.2. 电源复位

只要发生以下事件之一，就会产生电源复位：

- 上电/掉电复位（POR/PDR 复位）或；
- 欠压 (BOR) 复位。

### 6.1.3. 备份域复位

备份域复位会将所有 RTC 寄存器复位为各自的复位值。

只要发生以下事件之一，就会产生备份域复位：

- 在电源 VDD33 和 VBAT33 都已掉电后，VBAT33 又再上电。

## 6.2. 时钟

可以使用三种不同的时钟源来驱动系统时钟 (SYSCLK)：

- HSI 振荡器时钟
- HSE 振荡器时钟
- PLL1 时钟

器件具有以下两个次级时钟源：

- 32 kHz 低速内部 RC (LSI RC)，该 RC 用于驱动独立看门狗，也可选择提供给 RTC。
- 32.768 kHz 低速外部晶振（LSE 晶振），用于驱动 RTC 时钟 (RTCCLK)对于每个时钟源来说，在未使用时都可单独打开或者关闭，以降低功耗。

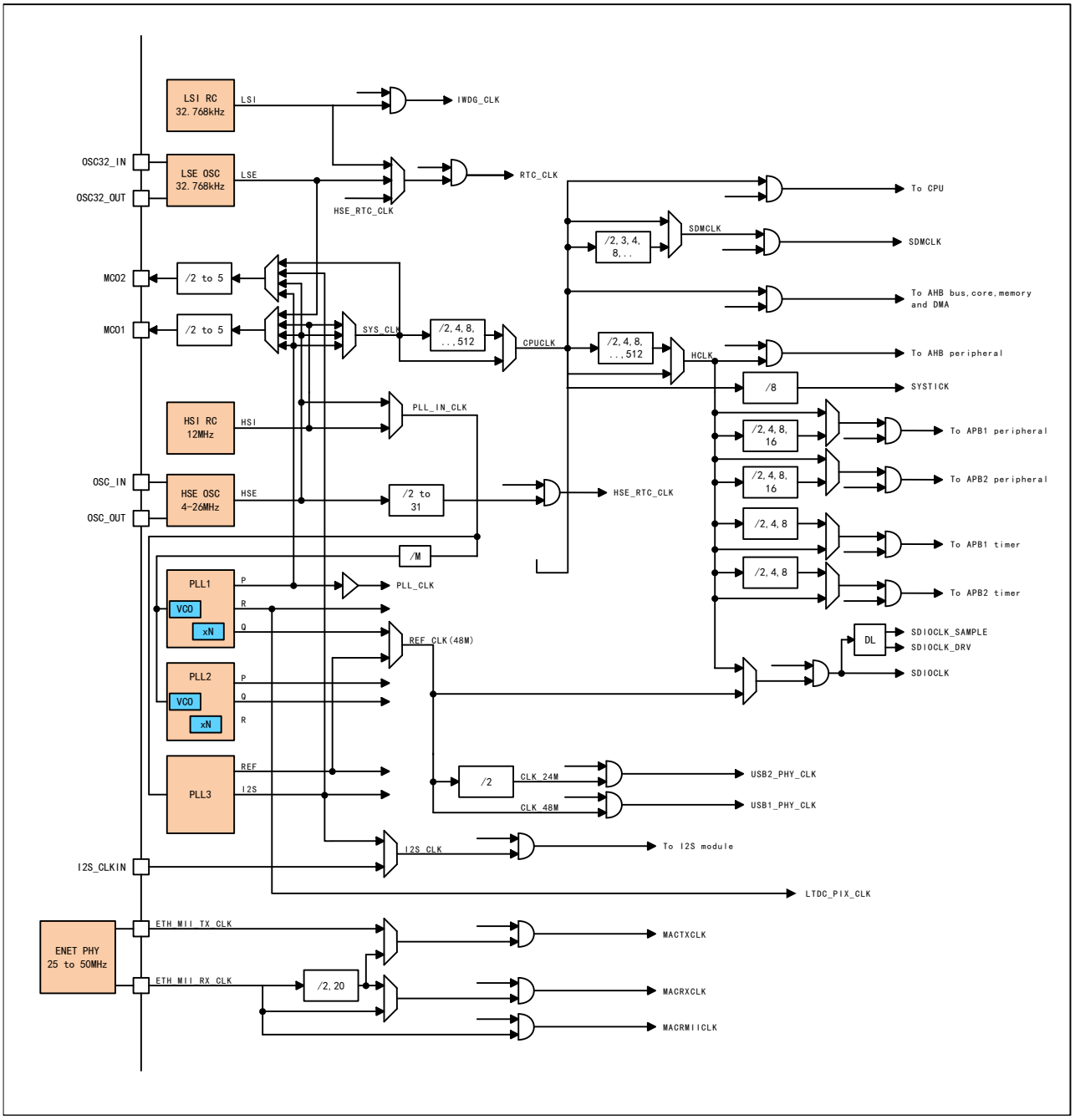



图 6-1 时钟树

时钟控制器为应用带来了高度的灵活性，用户在运行内核和外设时可选择使用外部晶振或者使用振荡器，既可采用最高的频率，也可为以太网、USB1、USB2、I2S 和 SDIO 等需要特定时钟的外设保证合适的频率。

可通过多个预分频器配置 AHB 频率、APB1 和 APB2。AHB 域的最大频率为 132MHz。APB1 和 APB2 的最大允许频率为 66MHz。

### 6.2.1. HSI 时钟

HSI 时钟信号由内部 12MHz RC 振荡器生成，可直接用作系统时钟，或者用作 PLL 输入。

HSI RC 振荡器的优点是成本较低（无需使用外部组件）。此外，其启动速度也要比 HSE 晶振块，但即使校

准后，其精度也不及外部晶振或陶瓷谐振器。

### 6.2.2. PLL 配置

芯片具有三个 PLL：

- PLL1 由 HSE 或 HSI 振荡器提供时钟信号，并具有两个不同的输出时钟：
  - 第一个输出用于生成高速系统时钟（最高达 264 MHz）。
  - 第二个输出用于生成 USB1, USB2 PHY 的参考时钟 (48 MHz) 和 SDIO 时钟 (48 MHz)等。
- PLL2 (PLLI2S) HSE 或 HSI 振荡器提供时钟信号。
- PLL3 用于生成精确时钟，并具有多个不同的输出时钟。
  - 27.12M 时钟。
  - 48MHz 参考时钟(USB、SDIO、SCI)。
  - I2S 时钟。
  - CLK_EXT 时钟，可配置 24M/36M/48M 时钟。

由于在 PLL 使能后 PLL 配置参数便不可更改，所以建议先对 PLL 进行配置，然后再使能（选择 HSI 或 HSE 振荡器作为 PLL 时钟源，并配置分频系数 M、N、P 和 Q）。

PLL2 使用与 PLL1 相同的输入时钟（PLLM[5:0] 和 PLLSRC 位为两个 PLL 所共用）。但是， PLL2 具有专门的使能/禁止和分频系数（N 和 R）配置位。在 PLL2 使能后，配置参数便不能更改。

当进入停机和待机模式后，三个 PLL 将由硬件禁止；如将 HSE 或 PLL（由 HSE 提供时钟信号）用作系统时钟，则在 HSE 发生故障时，三个 PLL 也将由硬件禁止。

### 6.2.3. 时钟安全系统(CSS)

时钟安全系统可通过软件激活。激活后，时钟监测器将在 HSE 振荡器启动延迟后使能，并在此振荡器停止时被关闭。

如果 HSE 时钟发生故障，此振荡器将自动禁止，一个时钟故障事件将发送到高级控制定时器 TIM1 和 TIM8 的断路输入，并且同时还将生成一个中断来向软件通知此故障（时钟安全系统中断，CSSI），以使 MCU 能够执行救援操作。CSSI 与 NMI（不可屏蔽中断）异常向量相链接。

*注意：*

*当 CSS 使能后，如果 HSE 时钟偶发故障，则 CSS 将生成一个中断，进而促使 NMI 自动生成。NMI 将 无限期执行，除非将 CSS 中断挂起位清零。因此，应用程序必须在 NMI ISR 中将 CSS 中断清零，具体方式为在时钟中断寄存器 (RCC_CIR) 中将 CSSC 位置 1。如果直接或间接使用 HSE 振荡器作为系统时钟（间接是指该振荡器直接用作 PLL 的输入时钟，并且该PLL 时钟为系统时钟）并且检出故障，则系统时钟将切换到 HSI 振荡器并且 HSE 振荡器将被禁止。如果 HSE 振荡器时钟是充当系统时钟的 PLL 的时钟源，则在发生故障时，PLL 也会被禁止。在此情况下，如果 PLL2 已使能，则在 HSE 发生故障时也会将其禁止。*

### 6.2.4. 时钟输出功能

共有两个微控制器时钟输出 (MCO) 引脚：

- MCO1

  用户可通过可配置的预分配器（从 1 到 5）向 MCO1 引脚 (PA8) 输出四个不同的时钟源：
  - HSI 时钟
  - LSE 时钟
  - HSE 时钟

- PLL1 时钟

所需的时钟源通过 RCC 时钟配置寄存器 (RCC_CFGR) 中的 MCO1PRE 和 MCO1 位选择。

- MCO2

用户可通过可配置的预分配器（从 1 到 5）向 MCO2 引脚 (PC9) 输出四个不同的时钟源：

- HSE 时钟
- PLL3 时钟
- 系统时钟 (SYSCLK)
- PLL1 时钟

所需的时钟源通过 RCC 时钟配置寄存器 (RCC_CFGR) 中的 MCO2PRE 和 MCO2 位选择。

对于不同的 MCO 引脚，必须将相应的 GPIO 端口在复用功能模式下进行设置。MCO 输出时钟不得超过 150 MHz（最大 I/O 速度）。

## 6.3. RCC 寄存器

有关寄存器说明中使用的缩写，请参见寄存器相关缩写词列表。

### 6.3.1. RCC 时钟控制寄存器 (CR)

**偏移地址：** 0x00

**复位值：** 0x00008083

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | PLL3RDY | PLL3ON | PLL2RDY | PLL2ON | PLL1RDY | PLL1ON | res. | | | | CSSON | HSEBYP | HSERDY | HSEON |
| | | ro | rw | ro | rw | ro | rw | | | | | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| HSICAL | | | | | | | | HSITRIM | | | | | res. | HSIRDY | HSION |
| ro | | | | | | | | rw | | | | | | ro | rw |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:30 | 保留 | 必须保持复位值 |
| 29 | PLL3RDY | PLL3 时钟就绪标志<br>由硬件置 1，用以指示 PLL3 已锁定。<br>0：PLL3 未锁定<br>1：PLL3 已锁定 |
| 28 | PLL3ON | PLL3 使能<br>由软件置 1 和清零，用于使能 PLL3<br>当进入停机或待机模式时由硬件清零。<br>0：PLL3 关闭<br>1：PLL3 开启 |
| 27 | PLL2RDY | PLL2 时钟就绪标志<br>由硬件置 1，用以指示 PLL2 已锁定。<br>0：PLL2 未锁定<br>1：PLL2 已锁定 |

| 位域 | 名称 | 描述 |
|---|---|---|
| 26 | PLL2ON | PLL2 使能<br>由软件置 1 和清零，用于使能 PLL2<br>当进入停机或待机模式时由硬件清零。<br>0：PLL2 关闭<br>1：PLL2 开启 |
| 25 | PLL1RDY | PLL1 时钟就绪标志<br>由硬件置 1，用以指示 PLL1 已锁定。<br>0：PLL1 未锁定<br>1：PLL1 已锁定 |
| 24 | PLL1ON | PLL1 使能<br>由软件置 1 和清零，用于使能 PLL1<br>当进入停机或待机模式时由硬件清零。<br>0：PLL1 关闭<br>1：PLL1 开启 |
| 23:20 | 保留 | 必须保持复位值 |
| 19 | CSSON | 时钟安全系统使能<br>由软件置 1 和清零，用于使能时钟安全系统。当 CSSON 置位时，时钟监测器将在 HSE 振荡器就绪时由硬件使能，并在检出振荡器故障时由硬件禁止。<br>0：时钟安全系统关闭（时钟监测器关闭）<br>1：时钟安全系统打开（如果 HSE 振荡器稳定，则时钟监测器打开；如果不稳定，则关闭） |
| 18 | HSEBYP | HSE 时钟旁路<br>由软件置 1 和清零，用于用外部时钟旁路振荡器。外部时钟必须通过 HSEON 位使能才能为<br>器件使用。<br>HSEBYP 只有在 HSE 振荡器已禁止的情况下才可写入。<br>0：不旁路 HSE 振荡器<br>1：外部时钟旁路 HSE 振荡器 |
| 17 | HSERDY | HSE 时钟就绪标志<br>由硬件置 1，用以指示 HSE 振荡器已稳定。在将 HSEON 位清零后，HSERDY 将在 6 个 HSE 振荡器时钟周期后转为低电平。<br>0：HSE 振荡器未就绪<br>1：HSE 振荡器已就绪 |
| 16 | HSEON | HSE 时钟使能<br>由软件置 1 和清零。<br>由硬件清零，用于在进入停机或待机模式时停止 HSE 振荡器。如果 HSE 振荡器直接或间接用于作为系统时钟，则此位不可复位。<br>0：HSE 振荡器关闭<br>1：HSE 振荡器打开 |
| 15:8 | HSICAL | 内部高速时钟校准<br>这些位在启动时自动初始化。 |
| 7:3 | HSITRIM | 内部高速时钟微调<br>通过这些位，可在 HSICAL[7:0] 位基础上实现可由用户编程的微调值。可通过编程使其适应电压和温度的差异，使内部 HSI RC 的频率更为准确。 |
| 2 | 保留 | 必须保持复位值 |
| 1 | HSIRDY | 内部高速时钟就绪标志<br>由硬件置 1，用以指示 HSI 振荡器已稳定。在将 HSION 位清零后，HSIRDY |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 将在 6 个 HSI 时钟周期后转为低电平。<br>0：HSI 振荡器未就绪<br>1：HSI 振荡器已就绪 |
| 0 | HSION | 内部高速时钟使能<br>由软件置 1 和清零。<br>由硬件置 1，用于在脱离停机或待机模式时或者在直接或间接用作系统时钟的 HSE 振荡器发生故障时强制 HSI 振荡器打开。如果 HSI 直接或间接用于作为系统时钟，则此位不可清零。<br>0：HSI 振荡器关闭<br>1 ：HSI 振荡器打开 |

### 6.3.2. RCC PLL 配置寄存器 (PLLCFGR)

**偏移地址：** 0x04

**复位值：** 0x24043010

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PLL1_R | | | | PLL1_Q | | | | res. | PLLSRC | res. | | | PLL1P_EN | PLL1_P | |
| rw | | | | rw | | | | | rw | | | | rw | rw | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | PLL1_N | | | | | | | | | PLLM | | | | | |
| | rw | | | | | | | | | rw | | | | | |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:28 | PLL1_R | PLL1_R：适用于 PLL1 分频系数<br>由软件置 1 和清零，用于控制 PLL1 时钟频率。这些位仅在 PLL1 已禁止时写入。<br>PLL1_R 时钟频率 = VCO 频率 / PLL1_R 并且 2 ≤ PLL1_R ≤ 15<br>0000：PLLR = 0，错误配置<br>0001：PLLR = 1，错误配置<br>0010：PLLR = 2<br>…<br>1111：PLLR = 15 |
| 27:24 | PLL1_Q | PLL1_Q：适用于 PLL1 分频系数<br>由软件置 1 和清零，用于控制 PLL1 时钟频率。这些位仅在 PLL1 已禁止时写入。<br>PLL1_Q 时钟频率 = VCO 频率 / PLL1_Q 并且 2 ≤ PLL1_Q ≤ 15<br>0000：PLL1_Q = 0，错误配置<br>0001：PLL1_Q = 1，错误配置<br>0010：PLL1_Q = 2<br>0011：PLL1_Q = 3<br>0100：PLL1_Q = 4<br>…<br>1111：PLL1_Q = 15 |
| 23 | 保留 | 必须保持复位值 |

| 位域 | 名称 | 描述 |
|---|---|---|
| 22 | PLLSRC | PLL 输入时钟源<br>由软件置 1 和清零，用于选择 PLL 时钟源。此位只有在 PLL 禁止时才可写入。<br>0：选择 HSI 时钟作为 PLL 时钟输入<br>1：选择 HSE 振荡器时钟作为 PLL 时钟输入 |
| 21:19 | 保留 | 必须保持复位值 |
| 18 | PLL1P_EN | PLL1P Bypass 配置<br>0：旁路 PLL1_P 分频系数，系统时钟频率 = VCO 频率<br>1：使能 PLL1_P 分频系数 |
| 17:16 | PLL1_P | 适用于 PLL1 分频系数<br>由软件置 1 和清零，用于控制 PLL1 输出时钟的频率。这些位只能在 PLL1 已禁止时写入。<br>PLL1_P 输出时钟频率 = VCO 频率 / PLL1_P 并且 PLL1_P = 2、4、6 或 8<br>00：PLL1_P = 2<br>01：PLL1_P = 4<br>10：PLL1_P = 6<br>11：PLL1_P = 8 |
| 15 | 保留 | 必须保持复位值 |
| 14:6 | PLL1_N | 适用于 VCO 的 PLL1 倍频系数<br>由软件置 1 和清零，用于控制 VCO 的倍频系数。这些位只能在 PLL1 已禁止时写入。写入这些位时只允许使用半字和字访问。<br>VCO 输出频率 = VCO 输入频率 × PLL1_N 并且 192 ≤ PLL1_N ≤ 432<br>000000000：PLL1_N = 0，错误配置<br>000000001：PLL1_N = 1，错误配置<br>...<br>011000000：PLL1_N = 192<br>...<br>110110000：PLL1_N = 432<br>110110001：PLL1_N = 433，错误配置<br>...<br>111111111：PLL1_N = 511，错误配置 |
| 5:0 | PLLM | PLL1 和 PLL2 输入时钟的分频系数<br>由软件置 1 和清零，用于在 VCO 之前对 PLL1 和 PLL2 输入时钟进行分频。这些位只有在 PLL1 和 PLL2 已禁止时才可写入。<br>注意： 软件必须正确设置这些位，确保 VCO 输入频率介于 1 和 2 MHz 之间。<br>VCO 输入频率 = PLL 输入时钟频率 / PLLM 并且 2 ≤ PLLM ≤ 63<br>000000：PLLM = 0，错误配置<br>000001：PLLM = 1，错误配置<br>000010：PLLM = 2<br>000011：PLLM = 3<br>000100：PLLM = 4<br>...<br>111110：PLLM = 62<br>111111：PLLM = 63 |

### 6.3.3. RCC 时钟配置寄存器 (CFGR)

**偏移地址：** 0x08

**复位值：** 0x00000000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MCO2 | | MCO2PRE | | | MCO1PRE | | | I2SSRC | MCO1 | | RTCPRE | | | | |
| rw | | rw | | | rw | | | rw | rw | | rw | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PPRE2 | | | PPRE1 | | | res. | | HPRE | | | | SWS | | SW | |
| rw | | | rw | | | | | rw | | | | ro | | rw | |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:30 | MCO2 | 微控制器时钟输出 2<br>由软件置 1 和清零。<br>00：选择系统时钟 (SYSCLK) 输出到 MCO2 引脚<br>01：选择 PLL3 时钟输出到 MCO2 引脚<br>10：选择 HSE 振荡器时钟输出到 MCO2 引脚<br>11：选择 PLL1 时钟输出到 MCO2 引脚 |
| 29:27 | MCO2PRE | MCO2 预分频器<br>由软件置 1 和清零，用于配置 MCO2 的预分频器。<br>0xx：无分频<br>100：2 分频<br>101：3 分 频<br>110：4 分频<br>111：5 分频 |
| 26:24 | MCO1PRE | MCO1 预分频器<br>由软件置 1 和清零，用于配置 MCO1 的预分频器。<br>0xx：无分频<br>100：2 分频<br>101：3 分 频<br>110：4 分频<br>111：5 分频 |
| 23 | I2SSRC | I2S 时钟选择<br>由软件置 1 和清零。通过此位可在 PLL3 时钟和外部时钟之间选择 I2S 时钟源。<br>0：PLL3 时钟用作 I2S 时钟源<br>1：在 I2S_CKIN 引脚上映射的外部时钟用作 I2S 时钟源 |
| 22:21 | MCO1 | 微控制器时钟输出 1<br>由软件置 1 和清零。<br>00：选择 HSI 时钟输出到 MCO1 引脚<br>01：选择 LSE 振荡器输出到 MCO1 引脚<br>10：选择 HSE 振荡器时钟输出到 MCO1 引脚<br>11：选择 PLL1 时钟输出到 MCO1 引脚 |
| 20:16 | RTCPRE | 适用于 RTC 时钟的 HSE 分频系数<br>由软件置 1 和清零，用于对 HSE 时钟输入时钟进行分频，进而为 RTC 生 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 成 1 MHz 的时钟。<br>00000：无时钟<br>00001：无时钟<br>00010：HSE/2<br>00011：HSE/3<br>00100：HSE/4<br>…<br>11110：HSE/30<br>11111 ： HSE/31 |
| 15:13 | PPRE2 | APB2 预分频器<br>由软件置位和清零，用于控制 APB2 时钟分频系数。<br>注意：APB2 时钟不可超过 90 MHz。<br>0xx：AHB 时钟不分频<br>100：AHB 时钟 2 分频<br>101：AHB 时钟 4 分频<br>110：AHB 时钟 8 分频<br>111：AHB 时钟 16 分频 |
| 12:10 | PPRE1 | APB1 预分频器<br>由软件置位和清零，用于控制 APB1 时钟分频系数。<br>注意：APB1 时钟不可超过 90 MHz。<br>0xx ： AHB 时钟不分频<br>100：AHB 时钟 2 分频<br>101：AHB 时钟 4 分频<br>110：AHB 时钟 8 分频<br>111：AHB 时钟 16 分频 |
| 9:8 | 保留 | 必须保持复位值 |
| 7:4 | HPRE | AHB 预分频器<br>由软件置位和清零，用于控制 AHB 时钟分频系数。<br>0xxx：系统时钟不分频<br>1000：系统时钟 2 分 频<br>1001：系统时钟 4 分频<br>1010：系统时钟 8 分频<br>1011：系统时钟 16 分频<br>1100：系统时钟 64 分频<br>1101：系统时钟 128 分 频<br>1110：系统时钟 256 分频<br>1111：系统时钟 512 分频 |
| 3:2 | SWS | 系统时钟切换状态<br>由硬件置 1 和清零，用于指示用作系统时钟的时钟源。<br>00：HSI 振荡器用作系统时钟<br>01：HSE 振荡器用作系统时钟<br>10：PLL1 用作系统时钟<br>11：不适用 |
| 1:0 | SW | 系统时钟切换<br>由软件置 1 和清零，用于选择系统时钟源。<br>由硬件置 1，用于在退出停机或待机模式时或者在直接或间接用作系统时钟的 HSE 振荡器发生故障时强制 HSI 的选择。<br>00：选择 HSI 振荡器作为系统时钟 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 01：选择 HSE 振荡器作为系统时钟<br>10：选择 PLL1 作为系统时钟<br>11：不允许 |

### 6.3.4. RCC 时钟中断寄存器 (CIR)

**偏移地址：** 0x0C

**复位值：** 0x00000000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | | | | | CSSC | res. | PLL2RDYC | PLL1RDYC | HSERDYC | HSIRDYC | LSERDYC | LSIRDYC |
| | | | | | | | | wo | | wo | wo | wo | wo | wo | wo |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | PLL2RDYIE | PLL1RDYIE | HSERDYIE | HSIRDYIE | LSERDYIE | LSIRDYIE | CSSF | res. | PLL2RDYF | PLL1RDYF | HSERDYF | HSIRDYF | LSERDYF | LSIRDYF |
| | | rw | rw | rw | rw | rw | rw | ro | | ro | ro | ro | ro | ro | ro |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:24 | 保留 | 必须保持复位值 |
| 23 | CSSC | 时钟安全系统中断清零<br>此位由软件置 1，用于将 CSSF 标志清零。<br>0：无操作<br>1：将 CSSF 标志清零 |
| 22 | 保留 | 必须保持复位值 |
| 21 | PLL2RDYC | PLL2 就绪中断清零<br>此位由软件置 1，用于将 PLL2RDYF 标志清零。<br>0：无操作<br>1：清零 PLLI2SRDYF |
| 20 | PLL1RDYC | PLL1 就绪中断清零<br>此位由软件置 1，用于将 PLL1RDYF 标志清零。<br>0：无操作<br>1：清零 PLLRDYF |
| 19 | HSERDYC | HSE 就绪中断清零<br>此位由软件置 1，用于将 HSERDYF 标志清零。<br>0：无操作<br>1：清零 HSERDYF |
| 18 | HSIRDYC | HSI 就绪中断清零<br>此位由软件置 1，用于将 HSIRDYF 标志清零。<br>0：无操作<br>1：清零 HSIRDYF |
| 17 | LSERDYC | LSE 就绪中断清零<br>此位由软件置 1，用于将 LSERDYF 标志清零。<br>0：无操作<br>1：清零 LSERDYF |
| 16 | LSIRDYC | LSI 就绪中断清零 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 此位由软件置 1，用于将 LSIRDYF 标志清零。<br>0：无操作<br>1：清零 LSIRDYF |
| 15:14 | 保留 | 必须保持复位值 |
| 13 | PLL2RDYIE | PLL2 就绪中断使能<br>由软件置 1 和清零，用于使能/禁止由 PLL2 锁定引起的中断。<br>0：禁止 PLL2 锁定中断<br>1：使能 PLL2 锁定中断 |
| 12 | PLL1RDYIE | PLL1 就绪中断使能<br>由软件置 1 和清零，用于使能/禁止由 PLL1 锁定引起的中断。<br>0：禁止 PLL1 锁定中断<br>1：使能 PLL1 锁定中断 |
| 11 | HSERDYIE | HSE 就绪中断使能<br>由软件置 1 和清零，用于使能/禁止由 HSE 振荡器稳定所引起的中断。<br>0：禁止 HSE 就绪中断<br>1：使能 HSE 就绪中断 |
| 10 | HSIRDYIE | HSI 就绪中断使能<br>由软件置 1 和清零，用于使能/禁止由 HSI 振荡器稳定所引起的中断。<br>0：禁止 HSI 就绪中断<br>1：使能 HSI 就绪中断 |
| 9 | LSERDYIE | LSE 就绪中断使能<br>由软件置 1 和清零，用于使能/禁止由 LSE 振荡器稳定所引起的中断。<br>0：禁止 LSE 就绪中断<br>1：使能 LSE 就绪中断 |
| 8 | LSIRDYIE | LSI 就绪中断使能<br>由软件置 1 和清零，用于使能/禁止由 LSI 振荡器稳定所引起的中断。<br>0：禁止 LSI 就绪中断<br>1：使能 LSI 就绪中断 |
| 7 | CSSF | 时钟安全系统中断标志<br>当在 HSE 振荡器中检出故障时由硬件置 1。<br>CSSC 位置 1 时由软件清零。<br>0：当前未因 HSE 时钟故障而引起时钟安全中断<br>1：因 HSE 时钟故障而引起时钟安全中断 |
| 6 | 保留 | 必须保持复位值 |
| 5 | PLL2RDYF | PLL2 就绪中断标志<br>当 PLL2 锁定并且 PLL2RDYIE 置 1 时由硬件置 1。<br>PLL2RDYC 位置 1 时由软件清零。<br>0：当前未因 PLL2 锁定而引起时钟就绪中断<br>1：因 PLL2 锁定而引起时钟就绪中断 |
| 4 | PLL1RDYF | PLL1 就绪中断标志<br>当 PLL 锁定并且 PLL1RDYIE 置 1 时由硬件置 1。<br>PLLRDYC 位置 1 时由软件清零。<br>0：当前未因 PLL1 锁定而引起时钟就绪中断<br>1：因 PLL1 锁定而引起时钟就绪中断 |
| 3 | HSERDYF | HSE 就绪中断标志<br>当外部高速时钟变为稳定且 HSERDYIE 置 1 时由硬件置 1。<br>HSERDYC 位置 1 时由软件清零。<br>0：当前未因 HSE 振荡器引起时钟就绪中断 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 1：因 HSE 振荡器引起时钟就绪中断 |
| 2 | HSIRDYF | HSI 就绪中断标志<br>当内部高速时钟变为稳定且 HSIRDYIE 置 1 时由硬件置 1。<br>HSIRDYC 位置 1 时由软件清零。<br>0：当前未因 HSI 振荡器引起时钟就绪中断<br>1：因 HSI 振荡器引起时钟就绪中断 |
| 1 | LSERDYF | LSE 就绪中断标志<br>当外部低速时钟变为稳定且 LSERDYIE 置 1 时由硬件置 1。<br>LSERDYC 位置 1 时由软件清零。<br>0：当前未因 LSE 振荡器引起时钟就绪中断<br>1：因 LSE 振荡器引起时钟就绪中断 |
| 0 | LSIRDYF | LSI 就绪中断标志<br>当内部低速时钟变为稳定且 LSIRDYIE 置 1 时由硬件置 1。<br>LSIRDYC 位置 1 时由软件清零。<br>0：当前未因 LSI 振荡器引起时钟就绪中断<br>1：因 LSI 振荡器引起时钟就绪中断 |

### 6.3.5. RCC AHB1 外设复位寄存器 (AHB1RSTR)

**偏移地址：** 0x10

**复位值：** 0x00000000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GLBRST | res. | USB2RST | res. | | | ETMACRST | res. | | | | | | | | |
| rw | | rw | | | | rw | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | CRCRST | res. | JPEGDRST | res. | GPIOIRST | GPIOHRST | GPIOGRST | GPIOFRST | GPIOERST | GPIODRST | GPIOCRST | GPIOBRST | GPIOARST |
| | | | rw | | rw | | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31 | GLBRST | 全局软复位<br>0：不复位系统<br>1：复位系统，同外部 NSRT 复位效果 |
| 30 | 保留 | 必须保持复位值 |
| 29 | USB2RST | USB2 模块复位<br>由软件置 1 和清零。<br>0：不复位 USB2 模块<br>1：复位 USB2 模块 |
| 28:26 | 保留 | 必须保持复位值 |
| 25 | ETMACRST | 以太网 MAC 复位<br>由软件置 1 和清零。<br>0：不复位以太网 MAC<br>1：复位以太网 MAC |
| 24:13 | 保留 | 必须保持复位值 |

| 位域 | 名称 | 描述 |
|---|---|---|
| 12 | CRCRST | CRC 复位<br>由软件置 1 和清零。<br>0：不复位 CRC<br>1：复位 CRC |
| 11 | 保留 | 必须保持复位值 |
| 10 | JPEGDRST | JPEG Decoder 复位<br>由软件置 1 和清零。<br>0：不复位 JPEG Decoder<br>1：复位 JPEG Decoder |
| 9 | 保留 | 必须保持复位值 |
| 8 | GPIOIRST | IO 端口 I 复位<br>由软件置 1 和清零。<br>0：不复位 IO 端口 I<br>1：复位 IO 端口 I |
| 7 | GPIOHRST | IO 端口 H 复位<br>由软件置 1 和清零。<br>0：不复位 IO 端口 H<br>1：复位 IO 端口 H |
| 6 | GPIOGRST | IO 端口 G 复位<br>由软件置 1 和清零。<br>0：不复位 IO 端口 G<br>1：复位 IO 端口 G |
| 5 | GPIOFRST | IO 端口 F 复位<br>由软件置 1 和清零。<br>0：不复位 IO 端口 F<br>1：复位 IO 端口 F |
| 4 | GPIOERST | IO 端口 E 复位<br>由软件置 1 和清零。<br>0：不复位 IO 端口 E<br>1：复位 IO 端口 E |
| 3 | GPIODRST | IO 端口 D 复位<br>由软件置 1 和清零。<br>0：不复位 IO 端口 D<br>1：复位 IO 端口 D |
| 2 | GPIOCRST | IO 端口 C 复位<br>由软件置 1 和清零。<br>0：不复位 IO 端口 C<br>1：复位 IO 端口 C |
| 1 | GPIOBRST | IO 端口 B 复位<br>由软件置 1 和清零。<br>0：不复位 IO 端口 B<br>1：复位 IO 端口 B |
| 0 | GPIOARST | IO 端口 A 复位<br>由软件置 1 和清零。<br>0：不复位 IO 端口 A<br>1：复位 IO 端口 A |

### 6.3.6. RCC AHB2 外设复位寄存器 (AHB2RSTR)

**偏移地址：** 0x14

**复位值：** 0x00000000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | | USB1RST | TRNGSRST | res. | SYMCRST | Quad-SPIRST | res. | | ASYMCRST | res. | | |
| | | | | | rw | rw | | rw | rw | | | rw | | | |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:11 | 保留 | 必须保持复位值 |
| 10 | USB1RST | USB1 模块复位<br>由软件置 1 和清零。<br>0：不复位 USB1 模块<br>1：复位 USB1 模块 |
| 9 | TRNGSRST | 真随机数发生器模块复位<br>由软件置 1 和清零。<br>0：不复位随机数发生器模块<br>1：复位随机数发生器模块 |
| 8 | 保留 | 必须保持复位值 |
| 7 | SYMCRST | 对称算法模块复位<br>由软件置 1 和清零。<br>0：不复位对称算法模块<br>1：复位对称算法模块 |
| 6 | Quad-SPIRST | Quad-SPI 模块复位<br>由软件置 1 和清零。<br>0：不复位 Quad-SPI 模块<br>1：复位 Quad-SPI 模块 |
| 5:4 | 保留 | 必须保持复位值 |
| 3 | ASYMCRST | 非对称算法模块复位<br>由软件置 1 和清零。<br>0：不复位非对称算法模块<br>1：复位非对称算法模块 |
| 2:0 | 保留 | 必须保持复位值 |

### 6.3.7. RCC AHB3 复位寄存器 (AHB3RSTR)

**偏移地址：** 0x18

**复位值：** 0x00000000

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:1 | 保留 | 必须保持复位值 |
| 0 | SDRAMCRST | SDRAM 模块复位<br>由软件置 1 和清零。<br>0：不复位 SDRAM 模块<br>1：复位 SDRAM 模块 |

### 6.3.8. RCC AHB4 复位寄存器 (AHB4RSTR)

**偏移地址：** 0x1C

**复位值：** 0x00000000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | | | | | | | IPIRST | LTDCRST | res. | DMA2DRST | DMA2RST | DMA1RST |
| | | | | | | | | | | rw | rw | | rw | rw | rw |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:6 | 保留 | 必须保持复位值 |
| 5 | IPIRST | IPI 复位<br>由软件置 1 和清零。<br>0：不复位 IPI<br>1：复位 IPI |
| 4 | LTDCRST | LTDC 复位<br>由软件置 1 和清零。<br>0：不复位 LTDC<br>1：复位 LTDC |
| 3 | 保留 | 必须保持复位值 |
| 2 | DMA2DRST | DMA2D 复位<br>由软件置 1 和清零。<br>0：不复位 DMA2D<br>1：复位 DMA2D |
| 1 | DMA2RST | DMA2 复位<br>由软件置 1 和清零。<br>0：不复位 DMA2<br>1：复位 DMA2 |
| 0 | DMA1RST | DMA1 复位<br>由软件置 1 和清零。<br>0：不复位 DMA1<br>1：复位 DMA1 |

### 6.3.9. RCC APB1 复位寄存器 (APB1RSTR)

**偏移地址：** 0x20

**复位值：** 0x00000000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UART8RST | UART7RST | DACRST | PWRRST | res. | CAN2RST | CAN1RST | res. | I2C3RST | I2C2RST | I2C1RST | UART5RST | UART4RST | UART3RST | UART2RST | res. |
| rw | rw | rw | rw | | rw | rw | | rw | rw | rw | rw | rw | rw | rw | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SPI3RST | SPI2RST | res. | | WWDGRST | res. | | TIM14RST | TIM13RST | TIM12RST | TIM7RST | TIM6RST | TIM5RST | TIM4RST | TIM3RST | TIM2RST |
| rw | rw | | | rw | | | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31 | UART8RST | UART8 复位<br>由软件置 1 和清零。<br>0：不复位 UART8<br>1：复位 UART8 |
| 30 | UART7RST | UART7 复位<br>由软件置 1 和清零。<br>0：不复位 UART7<br>1：复位 UART7 |
| 29 | DACRST | DAC 复位<br>由软件置 1 和清零。<br>0：不复位 DAC 接口<br>1：复位 DAC 接口 |
| 28 | PWRRST | 电源接口复位<br>由软件置 1 和清零。<br>0：不复位电源接口<br>1：复位电源接口 |
| 27 | 保留 | 必须保持复位值 |
| 26 | CAN2RST | CAN2 复位<br>由软件置 1 和清零。<br>0：不复位 CAN2<br>1：复位 CAN2 |
| 25 | CAN1RST | CAN1 复位<br>由软件置 1 和清零。<br>0：不复位 CAN1<br>1：复位 CAN1 |
| 24 | 保留 | 必须保持复位值 |
| 23 | I2C3RST | I2C3 复位<br>由软件置 1 和清零。<br>0：不复位 I2C3<br>1：复位 I2C3 |
| 22 | I2C2RST | I2C2 复位<br>由软件置 1 和清零。<br>0：不复位 I2C2<br>1：复位 I2C2 |
| 21 | I2C1RST | I2C1 复位<br>由软件置 1 和清零。 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 0：不复位 I2C1<br>1：复位 I2C1 |
| 20 | UART5RST | UART5 复位<br>由软件置 1 和清零。<br>0：不复位 UART5<br>1：复位 UART5 |
| 19 | UART4RST | USART4 复位<br>由软件置 1 和清零。<br>0：不复位 UART4<br>1：复位 UART4 |
| 18 | UART3RST | USART3 复位<br>由软件置 1 和清零。<br>0：不复位 USART3<br>1：复位 USART3 |
| 17 | UART2RST | USART2 复位<br>由软件置 1 和清零。<br>0：不复位 USART2<br>1：复位 USART2 |
| 16 | 保留 | 必须保持复位值 |
| 15 | SPI3RST | SPI3 复位<br>由软件置 1 和清零。<br>0：不复位 SPI3<br>1：复位 SPI3 |
| 14 | SPI2RST | SPI2 复位<br>由软件置 1 和清零。<br>0：不复位 SPI2<br>1：复位 SPI2 |
| 13:12 | 保留 | 必须保持复位值 |
| 11 | WWDGRST | 窗口看门狗复位<br>由软件置 1 和清零。<br>0：不复位窗口看门狗<br>1：复位窗口看门狗 |
| 10:9 | 保留 | 必须保持复位值 |
| 8 | TIM14RST | TIM14 复位<br>由软件置 1 和清零。<br>0：不复位 TIM14<br>1：复位 TIM14 |
| 7 | TIM13RST | TIM13 复位<br>由软件置 1 和清零。<br>0：不复位 TIM13<br>1：复位 TIM13 |
| 6 | TIM12RST | TIM12 复位<br>由软件置 1 和清零。<br>0：不复位 TIM12<br>1：复位 TIM12 |
| 5 | TIM7RST | TIM7 复位<br>由软件置 1 和清零。<br>0：不复位 TIM7 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 1：复位 TIM7 |
| 4 | TIM6RST | TIM6 复位<br>由软件置 1 和清零。<br>0：不复位 TIM6<br>1：复位 TIM6 |
| 3 | TIM5RST | TIM5 复位<br>由软件置 1 和清零。<br>0：不复位 TIM5<br>1：复位 TIM5 |
| 2 | TIM4RST | TIM4 复位<br>由软件置 1 和清零。<br>0：不复位 TIM4<br>1：复位 TIM4 |
| 1 | TIM3RST | TIM3 复位<br>由软件置 1 和清零。<br>0：不复位 TIM3<br>1：复位 TIM3 |
| 0 | TIM2RST | TIM2 复位<br>由软件置 1 和清零。<br>0：不复位 TIM2<br>1：复位 TIM2 |

### 6.3.10. RCC APB2 复位寄存器 (APB2RSTR)

**偏移地址：** 0x24

**复位值：** 0x00000000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | | | PHERCFGRST | res. | | | | SPI5RST | res. | TIM11RST | TIM10RST | TIM9RST |
| | | | | | | rw | | | | | rw | | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | SYSCFGRST | SPI4RST | SPI1RST | SDIORST | res. | | ADCRST | res. | | USART6RST | USART1RST | res. | | TIM8RST | TIM1RST |
| | rw | rw | rw | rw | | | rw | | | rw | rw | | | rw | rw |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:26 | 保留 | 必须保持复位值 |
| 25 | PHERCFGRST | PHERCFG 复位<br>由软件置 1 和清零。<br>0：不复位 PHERCFG<br>1：复位 PHERCFG |
| 24:21 | 保留 | 必须保持复位值 |
| 20 | SPI5RST | SPI5 复位<br>由软件置 1 和清零。<br>0：不复位 SPI5 |

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 1：复位 SPI5 |
| 19 | 保留 | 必须保持复位值 |
| 18 | TIM11RST | TIM11 复位<br>由软件置 1 和清零。<br>0：不复位 TIM11<br>1：复位 TIM11 |
| 17 | TIM10RST | TIM10 复位<br>由软件置 1 和清零。<br>0：不复位 TIM10<br>1：复位 TIM10 |
| 16 | TIM9RST | TIM9 复位<br>由软件置 1 和清零。<br>0：不复位 TIM9<br>1：复位 TIM9 |
| 15 | 保留 | 必须保持复位值 |
| 14 | SYSCFGRST | SYSCFG 复位<br>由软件置 1 和清零。<br>0：不复位 SYSCFG<br>1：复位 SYSCFG |
| 13 | SPI4RST | SPI4 复位<br>由软件置 1 和清零。<br>0：不复位 SPI4<br>1：复位 SPI4 |
| 12 | SPI1RST | SPI1 复位<br>由软件置 1 和清零。<br>0：不复位 SPI1<br>1：复位 SPI1 |
| 11 | SDIORST | SDIO 复位<br>由软件置 1 和清零。<br>0：不复位 SDIO 模块<br>1：复位 SDIO 模块 |
| 10:9 | 保留 | 必须保持复位值 |
| 8 | ADCRST | ADC 接口复位（所有 ADC 共用）<br>由软件置 1 和清零。<br>0：不复位 ADC 接口<br>1：复位 ADC 接口 |
| 7:6 | 保留 | 必须保持复位值 |
| 5 | USART6RST | USART6 复位<br>由软件置 1 和清零。<br>0：不复位 USART6<br>1：复位 USART6 |
| 4 | USART1RST | USART1 复位<br>由软件置 1 和清零。<br>0：不复位 USART1<br>1：复位 USART1 |
| 3:2 | 保留 | 必须保持复位值 |
| 1 | TIM8RST | TIM8 复位<br>由软件置 1 和清零。 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 0：不复位 TIM8<br>1：复位 TIM8 |
| 0 | TIM1RST | TIM1 复位<br>由软件置 1 和清零。<br>0：不复位 TIM1<br>1：复位 TIM1 |

### 6.3.11. RCC AHB1 外设时钟使能寄存器 (AHB1ENR)

**偏移地址：** 0x30

**复位值：** 0x00100000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | USB2EN | ETHMACPTPEN | ETHMACRXEN | ETHMACTXEN | ETHMACEN | res. | | | | CCMEN | res. | | | |
| | | rw | rw | rw | rw | rw | | | | | rw | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | CRCEN | res. | JPEGEN | res. | GPIOIEN | GPIOHEN | GPIOGEN | GPIOFEN | GPIOEEN | GPIODEN | GPIOCEN | GPIOBEN | GPIOAEN |
| | | | rw | | rw | | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:30 | 保留 | 必须保持复位值 |
| 29 | USB2EN | USB2 时钟使能<br>由软件置 1 和清零。<br>0：禁止 USB2 时钟<br>1：使能 USB2 时钟 |
| 28 | ETHMACPTPEN | 以太网 PTP 时钟使能<br>由软件置 1 和清零。<br>0：禁止以太网 PTP 时钟<br>1：使能以太网 PTP 时钟 |
| 27 | ETHMACRXEN | 以太网接收时钟使能<br>由软件置 1 和清零。<br>0：禁止以太网接收时钟<br>1：使能以太网接收时钟 |
| 26 | ETHMACTXEN | 以太网发送时钟使能<br>由软件置 1 和清零。<br>0：禁止以太网发送时钟<br>1：使能以太网发送时钟 |
| 25 | ETHMACEN | 以太网 MAC 时钟使能<br>由软件置 1 和清零。<br>0：禁止以太网 MAC 时钟<br>1：使能以太网 MAC 时钟 |
| 24:21 | 保留 | 必须保持复位值 |
| 20 | CCMEN | CCM 数据 RAM 时钟使能<br>由软件置 1 和清零。 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 0：禁止 CCM 数据 RAM 时钟<br>1：使能 CCM 数据 RAM 时钟 |
| 19:13 | 保留 | 必须保持复位值 |
| 12 | CRCEN | CRC 时钟使能<br>由软件置 1 和清零。<br>0：禁止 CRC 时钟<br>1：使能 CRC 时钟 |
| 11 | 保留 | 必须保持复位值 |
| 10 | JPEGEN | JPEG Decoder 时钟使能<br>由软件置 1 和清零。<br>0：禁止 JPEG Decoder 时钟<br>1：使能 JPEG Decoder 时钟 |
| 9 | 保留 | 必须保持复位值 |
| 8 | GPIOIEN | IO 端口 I 时钟使能<br>由软件置 1 和清零。<br>0：禁止 IO 端口 I 时钟<br>1：使能 IO 端口 I 时钟 |
| 7 | GPIOHEN | IO 端口 H 时钟使能<br>由软件置 1 和清零。<br>0：禁止 IO 端口 H 时钟<br>1：使能 IO 端口 H 时钟 |
| 6 | GPIOGEN | IO 端口 G 时钟使能<br>由软件置 1 和清零。<br>0：禁止 IO 端口 G 时钟<br>1：使能 IO 端口 G 时钟 |
| 5 | GPIOFEN | IO 端口 F 时钟使能<br>由软件置 1 和清零。<br>0：禁止 IO 端口 F 时钟<br>1：使能 IO 端口 F 时钟 |
| 4 | GPIOEEN | IO 端口 E 时钟使能<br>由软件置 1 和清零。<br>0：禁止 IO 端口 E 时钟<br>1：使能 IO 端口 E 时钟 |
| 3 | GPIODEN | IO 端口 D 时钟使能<br>由软件置 1 和清零。<br>0：禁止 IO 端口 D 时钟<br>1：使能 IO 端口 D 时钟 |
| 2 | GPIOCEN | IO 端口 C 时钟使能<br>由软件置 1 和清零。<br>0：禁止 IO 端口 C 时钟<br>1：使能 IO 端口 C 时钟 |
| 1 | GPIOBEN | IO 端口 B 时钟使能<br>由软件置 1 和清零。<br>0：禁止 IO 端口 B 时钟<br>1：使能 IO 端口 B 时钟 |
| 0 | GPIOAEN | IO 端口 A 时钟使能<br>由软件置 1 和清零。<br>0：禁止 IO 端口 A 时钟 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 1：使能 IO 端口 A 时钟 |

### 6.3.12. RCC AHB2 外设时钟使能寄存器 (AHB2ENR)

**偏移地址：** 0x34

**复位值：** 0x00000020

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | | USB1EN | TRNGEN | res. | SYMCEN | Quad-SPIEN | OTPEN | res. | ASYMCEN | BPUEN | res. | |
| | | | | | rw | rw | | rw | rw | rw | | rw | rw | | |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:11 | 保留 | 必须保持复位值 |
| 10 | USB1EN | USB1 时钟使能<br>由软件置 1 和清零。<br>0：USB1 时钟禁用<br>1：使能 USB1 时钟 |
| 9 | TRNGEN | 真随机数发生器时钟使能<br>由软件置 1 和清零。<br>0：禁止随机数发生器时钟<br>1：使能随机数发生器时钟 |
| 8 | 保留 | 必须保持复位值 |
| 7 | SYMCEN | 对称算法模块使能<br>由软件置 1 和清零。<br>0：禁止 对称算法模块<br>1：使能 对称算法模块 |
| 6 | Quad-SPIEN | Quad-SPI 模块使能<br>由软件置 1 和清零。<br>0：禁止 Quad-SPI 模块<br>1：使能 Quad-SPI 模块 |
| 5 | OTPEN | OTP 模块使能<br>由软件置 1 和清零。<br>0：禁止 OTP 模块<br>1：使能 OTP 模块 |
| 4 | 保留 | 必须保持复位值 |
| 3 | ASYMCEN | 非对称算法模块使能<br>由软件置 1 和清零。<br>0：禁止 非对称算法模块<br>1：使能 非对称算法模块 |
| 2 | BPUEN | BPU 模块使能<br>由软件置 1 和清零。<br>0：禁止 BPU 模块<br>1：使能 BPU 模块 |
| 1:0 | 保留 | 必须保持复位值 |

### 6.3.13. RCC AHB3 外设时钟使能寄存器 (AHB3ENR)

**偏移地址：** 0x38

**复位值：** 0x00000000

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:1 | 保留 | 必须保持复位值 |
| 0 | SDRAMCEN | SDRAM 模块使能<br>由软件置 1 和清零。<br>0：禁止 SDRAM 模块<br>1：使能 SDRAM 模块 |

### 6.3.14. RCC AHB4 外设时钟使能寄存器 (AHB4ENR)

**偏移地址：** 0x3C

**复位值：** 0x00000000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | | | | | | | IPIEN | LTDCEN | res. | DMA2DEN | DMA2EN | DMA1EN |
| | | | | | | | | | | rw | rw | | rw | rw | rw |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:6 | 保留 | 必须保持复位值 |
| 5 | IPIEN | IPI 时钟使能<br>由软件置 1 和清零。<br>0：禁止 IPI 时钟<br>1：使能 IPI 时钟 |
| 4 | LTDCEN | LTDC 时钟使能<br>由软件置 1 和清零。<br>0：禁止 LTDC 时钟<br>1：使能 LTDC 时钟 |
| 3 | 保留 | 必须保持复位值 |
| 2 | DMA2DEN | DMA2D 时钟使能<br>由软件置 1 和清零。<br>0：禁止 DMA2D 时钟<br>1：使能 DMA2D 时钟 |
| 1 | DMA2EN | DMA2 时钟使能<br>由软件置 1 和清零。<br>0：禁止 DMA2 时钟<br>1：使能 DMA2 时钟 |
| 0 | DMA1EN | DMA1 时钟使能 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 由软件置 1 和清零。<br>0：禁止 DMA1 时钟<br>1：使能 DMA1 时钟 |

### 6.3.15. RCC APB1 外设时钟使能寄存器 (APB1ENR)

**偏移地址：** 0x40

**复位值：** 0x00000000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UART8EN | UART7EN | DACEN | PWREN | res. | CAN2EN | CAN1EN | res. | I2C3EN | I2C2EN | I2C1EN | UART5EN | UART4EN | UART3EN | UART2EN | res. |
| rw | rw | rw | rw | | rw | rw | | rw | rw | rw | rw | rw | rw | rw | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SPI3EN | SPI2EN | res. | | WWDGEN | res. | | TIM14EN | TIM13EN | TIM12EN | TIM7EN | TIM6EN | TIM5EN | TIM4EN | TIM3EN | TIM2EN |
| rw | rw | | | rw | | | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31 | UART8EN | UART8 时钟使能<br>由软件置 1 和清零。<br>0：禁止 UART8 时钟<br>1：使能 UART8 时钟 |
| 30 | UART7EN | UART7 时钟使能<br>由软件置 1 和清零。<br>0：禁止 UART7 时钟<br>1：使能 UART7 时钟 |
| 29 | DACEN | DAC 接口时钟使能<br>由软件置 1 和清零。<br>0：禁止 DAC 接口时钟<br>1：使能 DAC 接口时钟 |
| 28 | PWREN | 电源接口时钟使能<br>由软件置 1 和清零。<br>0：禁止电源接口时钟<br>1：使能电源接口时钟 |
| 27 | 保留 | 必须保持复位值 |
| 26 | CAN2EN | CAN 2 时钟使能<br>由软件置 1 和清零。<br>0：禁止 CAN 2 时钟<br>1：使能 CAN 2 时钟 |
| 25 | CAN1EN | CAN 1 时钟使能<br>由软件置 1 和清零。<br>0：禁止 CAN 1 时钟<br>1：使能 CAN 1 时钟 |
| 24 | 保留 | 必须保持复位值 |
| 23 | I2C3EN | I2C3 时钟使能 |

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 由软件置 1 和清零。<br>0：禁止 I2C3 时钟<br>1：使能 I2C3 时钟 |
| 22 | I2C2EN | I2C2 时钟使能<br>由软件置 1 和清零。<br>0：禁止 I2C2 时钟<br>1：使能 I2C2 时钟 |
| 21 | I2C1EN | I2C1 时钟使能<br>由软件置 1 和清零。<br>0：禁止 I2C1 时钟<br>1：使能 I2C1 时钟 |
| 20 | UART5EN | UART5 时钟使能<br>由软件置 1 和清零。<br>0：禁止 UART5 时钟<br>1：使能 UART5 时钟 |
| 19 | UART4EN | UART4 时钟使能<br>由软件置 1 和清零。<br>0：禁止 UART4 时钟<br>1：使能 UART4 时钟 |
| 18 | UART3EN | USART3 时钟使能<br>由软件置 1 和清零。<br>0：禁止 USART3 时钟<br>1：使能 USART3 时钟 |
| 17 | UART2EN | USART2 时钟使能<br>由软件置 1 和清零。<br>0：禁止 USART2 时钟<br>1：使能 USART2 时钟 |
| 16 | 保留 | 必须保持复位值 |
| 15 | SPI3EN | SPI3 时钟使能<br>由软件置 1 和清零。<br>0：禁止 SPI3 时钟<br>1：使能 SPI3 时钟 |
| 14 | SPI2EN | SPI2 时钟使能<br>由软件置 1 和清零。<br>0：禁止 SPI2 时钟<br>1：使能 SPI2 时钟 |
| 13:12 | 保留 | 必须保持复位值 |
| 11 | WWDGEN | 窗口看门狗时钟使能<br>由软件置 1 和清零。<br>0：禁止窗口看门狗时钟<br>1：使能窗口看门狗时钟 |
| 10:9 | 保留 | 必须保持复位值 |
| 8 | TIM14EN | TIM14 时钟使能<br>由软件置 1 和清零。<br>0：禁止 TIM14 时钟<br>1：使能 TIM14 时钟 |
| 7 | TIM13EN | TIM13 时钟使能<br>由软件置 1 和清零。 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 0：禁止 TIM13 时钟<br>1：使能 TIM13 时钟 |
| 6 | TIM12EN | TIM12 时钟使能<br>由软件置 1 和清零。<br>0：禁止 TIM12 时钟<br>1：使能 TIM12 时钟 |
| 5 | TIM7EN | TIM7 时钟使能<br>由软件置 1 和清零。<br>0：禁止 TIM7 时钟<br>1：使能 TIM7 时钟 |
| 4 | TIM6EN | TIM6 时钟使能<br>由软件置 1 和清零。<br>0：禁止 TIM6 时钟<br>1：使能 TIM6 时钟 |
| 3 | TIM5EN | TIM5 时钟使能<br>由软件置 1 和清零。<br>0：禁止 TIM5 时钟<br>1：使能 TIM5 时钟 |
| 2 | TIM4EN | TIM4 时钟使能<br>由软件置 1 和清零。<br>0：禁止 TIM4 时钟<br>1：使能 TIM4 时钟 |
| 1 | TIM3EN | TIM3 时钟使能<br>由软件置 1 和清零。<br>0：禁止 TIM3 时钟<br>1：使能 TIM3 时钟 |
| 0 | TIM2EN | TIM2 时钟使能<br>由软件置 1 和清零。<br>0：禁止 TIM2 时钟<br>1：使能 TIM2 时钟 |

### 6.3.16. RCC APB2 外设时钟使能寄存器 (APB2ENR)

**偏移地址：** 0x44

**复位值：** 0x00000000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | | | PHERCFGEN | res. | | | | SPI5EN | res. | TIM11EN | TIM10EN | TIM9EN |
| | | | | | | rw | | | | | rw | | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | SYSCFGEN | SPI4EN | SPI1EN | SDIOEN | ADC3EN | ADC2EN | ADC1EN | res. | | USART6EN | USART1EN | res. | | TIM8EN | TIM1EN |
| | rw | rw | rw | rw | rw | rw | rw | | | rw | rw | | | rw | rw |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:26 | 保留 | 必须保持复位值 |
| 25 | PHERCFGEN | PHERCFG 时钟使能<br>由软件置 1 和清零。<br>0：禁止 PHERCFG 时钟<br>1：使能 PHERCFG 时钟 |
| 24:21 | 保留 | 必须保持复位值 |
| 20 | SPI5EN | SPI5 时钟使能<br>由软件置 1 和清零。<br>0：禁止 SPI5 时钟<br>1：使能 SPI5 时钟 |
| 19 | 保留 | 必须保持复位值 |
| 18 | TIM11EN | TIM11 时钟使能<br>由软件置 1 和清零。<br>0：禁止 TIM11 时钟<br>1：使能 TIM11 时钟 |
| 17 | TIM10EN | TIM10 时钟使能<br>由软件置 1 和清零。<br>0：禁止 TIM10 时钟<br>1：使能 TIM10 时钟 |
| 16 | TIM9EN | TIM9 时钟使能<br>由软件置 1 和清零。<br>0：禁止 TIM9 时钟<br>1：使能 TIM9 时钟 |
| 15 | 保留 | 必须保持复位值 |
| 14 | SYSCFGEN | 系统配置控制器时钟使能<br>由软件置 1 和清零。<br>0：禁止系统配置控制器时钟<br>1：使能系统配置控制器时钟 |
| 13 | SPI4EN | SPI4 时钟使能<br>由软件置 1 和清零。<br>0：禁止 SPI4 时钟<br>1：使能 SPI4 时钟 |
| 12 | SPI1EN | SPI1 时钟使能<br>由软件置 1 和清零。<br>0：禁止 SPI1 时钟<br>1：使能 SPI1 时钟 |
| 11 | SDIOEN | SDIO 时钟使能<br>由软件置 1 和清零。<br>0：禁止 SDIO 模块时钟<br>1：使能 SDIO 模块时钟 |
| 10 | ADC3EN | ADC3 时钟使能<br>由软件置 1 和清零。<br>0：禁止 ADC3 时钟<br>1：使能 ADC3 时钟 |
| 9 | ADC2EN | ADC2 时钟使能<br>由软件置 1 和清零。<br>0：禁止 ADC2 时钟<br>1：使能 ADC2 时钟 |

| 位域 | 名称 | 描述 |
|---|---|---|
| 8 | ADC1EN | ADC1 时钟使能<br>由软件置 1 和清零。<br>0：禁止 ADC1 时钟<br>1：使能 ADC1 时钟 |
| 7:6 | 保留 | 必须保持复位值 |
| 5 | USART6EN | USART6 时钟使能<br>由软件置 1 和清零。<br>0：禁止 USART6 时钟<br>1：使能 USART6 时钟 |
| 4 | USART1EN | USART1 时钟使能<br>由软件置 1 和清零。<br>0：禁止 USART1 时钟<br>1：使能 USART1 时钟 |
| 3:2 | 保留 | 必须保持复位值 |
| 1 | TIM8EN | TIM8 时钟使能<br>由软件置 1 和清零。<br>0：禁止 TIM8 时钟<br>1：使能 TIM8 时钟 |
| 0 | TIM1EN | TIM1 时钟使能<br>由软件置 1 和清零。<br>0：禁止 TIM1 时钟<br>1：使能 TIM1 时钟 |

### 6.3.17. RCC AHB1 低功耗时钟门控寄存器 (AHB1LPENR)

**偏移地址：** 0x50

**复位值：** 0x00000000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | USB2LPEN | ETHMACPTPLPEN | ETHMACRXLPEN | ETHMACTXLPEN | ETHMACLPEN | res. | | | | | | SRAM3LPEN | SRAM2LPEN | SRAM1LPEN |
| | | rw | rw | rw | rw | rw | | | | | | | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| XFC_LPEN | res. | | CRCLPEN | res. | JPEGLPEN | res. | GPIOILPEN | GPIOHLPEN | GPIOGLPEN | GPIOFLPEN | GPIOELPEN | GPIODLPEN | GPIOCLPEN | GPIOBLPEN | GPIOALPEN |
| rw | | | rw | | rw | | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:30 | 保留 | 必须保持复位值 |
| 29 | USB2LPEN | 睡眠模式期间的 USB2 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 USB2 时钟<br>1：睡眠模式期间使能 USB2 时钟 |
| 28 | ETHMACPTPLPEN | 睡眠模式期间的以太网 PTP 时钟使能<br>由软件置 1 和清零。 |

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 0：睡眠模式期间禁止以太网 PTP 时钟<br>1：睡眠模式期间使能 Ethernet PTP 时钟 |
| 27 | ETHMACRXLPEN | 睡眠模式期间的以太网接收时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止以太网接收时钟<br>1：睡眠模式期间使能以太网接收时钟 |
| 26 | ETHMACTXLPEN | 睡眠模式期间的以太网发送时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止以太网发送时钟<br>1：睡眠模式期间使能以太网发送时钟 |
| 25 | ETHMACLPEN | 睡眠模式期间的以太网 MAC 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止以太网 MAC 时钟<br>1：睡眠模式期间使能以太网 MAC 时钟 |
| 24:19 | 保留 | 必须保持复位值 |
| 18 | SRAM3LPEN | 睡眠模式期间的 SRAM3 接口时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 SRAM3 接口时钟<br>1：睡眠模式期间使能 SRAM3 接口时钟 |
| 17 | SRAM2LPEN | 睡眠模式期间的 SRAM2 接口时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 SRAM2 接口时钟<br>1 ：睡眠模式期间使能 SRAM2 接口时钟 |
| 16 | SRAM1LPEN | 睡眠模式期间的 SRAM1 接口时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 SRAM1 接口时钟<br>1：睡眠模式期间使能 SRAM1 接口时钟 |
| 15 | XFC_LPEN | 睡眠模式期间的 XIP Flash 接口时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 XIP Flash 接口时钟<br>1：睡眠模式期间使能 XIP Flash 接口时钟 |
| 14:13 | 保留 | 必须保持复位值 |
| 12 | CRCLPEN | 睡眠模式期间的 CRC 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 CRC 时钟<br>1：睡眠模式期间使能 CRC 时钟 |
| 11 | 保留 | 必须保持复位值 |
| 10 | JPEGLPEN | 睡眠模式期间的 JPEG DECODER 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 JPEG DECODER 时钟<br>1：睡眠模式期间使能 JPEG DECODER 时钟 |
| 9 | 保留 | 必须保持复位值 |
| 8 | GPIOILPEN | 睡眠模式期间的 IO 端口 I 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 IO 端口 I 时钟<br>1：睡眠模式期间使能 IO 端口 I 时钟 |
| 7 | GPIOHLPEN | 睡眠模式期间的 IO 端口 H 时钟使能<br>由软件置 1 和清零。 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 0：睡眠模式期间禁止 IO 端口 H 时钟<br>1：睡眠模式期间使能 IO 端口 H 时钟 |
| 6 | GPIOGLPEN | 睡眠模式期间的 IO 端口 G 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 IO 端口 G 时钟<br>1：睡眠模式期间使能 IO 端口 G 时钟 |
| 5 | GPIOFLPEN | 睡眠模式期间的 IO 端口 F 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 IO 端口 F 时钟<br>1：睡眠模式期间使能 IO 端口 F 时钟 |
| 4 | GPIOELPEN | 睡眠模式期间的 IO 端口 E 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 IO 端口 E 时钟<br>1：睡眠模式期间使能 IO 端口 E 时钟 |
| 3 | GPIODLPEN | 睡眠模式期间的 IO 端口 D 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 IO 端口 D 时钟<br>1：睡眠模式期间使能 IO 端口 D 时钟 |
| 2 | GPIOCLPEN | 睡眠模式期间的 IO 端口 C 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 IO 端口 C 时钟<br>1：睡眠模式期间使能 IO 端口 C 时钟 |
| 1 | GPIOBLPEN | 睡眠模式期间的 IO 端口 B 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 IO 端口 B 时钟<br>1：睡眠模式期间使能 IO 端口 B 时钟 |
| 0 | GPIOALPEN | 睡眠模式期间的 IO 端口 A 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 IO 端口 A 时钟<br>1：睡眠模式期间使能 IO 端口 A 时钟 |

### 6.3.18. RCC AHB2 低功耗时钟门控寄存器 (AHB2LPENR)

**偏移地址：** 0x54

**复位值：** 0x00000000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | | USB1LPEN | TRNGLPEN | res. | SYMCLPEN | Quad-SPILPEN | OTPLPEN | res. | ASYMCLPEN | res. | | |
| | | | | | rw | rw | | rw | rw | rw | | rw | | | |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:11 | 保留 | 必须保持复位值 |
| 10 | USB1LPEN | 睡眠模式期间的 USB1 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 USB1 时钟<br>1：睡眠模式期间使能 USB1 时钟 |
| 9 | TRNGLPEN | 睡眠模式期间的真随机数发生器时钟使能 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 由软件置 1 和清零。<br>0：睡眠模式期间禁止真随机数发生器时钟<br>1：睡眠模式期间使能真随机数发生器时钟 |
| 8 | 保留 | 必须保持复位值 |
| 7 | SYMCLPEN | 睡眠模式期间的对称算法时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止对称算法时钟<br>1：睡眠模 式期间使能对称算法时钟 |
| 6 | Quad-SPILPEN | 睡眠模式期间的 Quad-SPI 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 Quad-SPI 时钟<br>1：睡眠模式期间使能 Quad-SPI 时钟 |
| 5 | OTPLPEN | 睡眠模式期间的 OTP 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 OTP 时钟<br>1：睡眠模式期间使能 OTP 时钟 |
| 4 | 保留 | 必须保持复位值 |
| 3 | ASYMCLPEN | 睡眠模式期间的非对称算法时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止非对称算法时钟<br>1：睡眠模式期间使能非对称算法时钟 |
| 2:0 | 保留 | 必须保持复位值 |

### 6.3.19. RCC AHB3 低功耗时钟门控寄存器 (AHB3LPENR)

**偏移地址：** 0x58

**复位值：** 0x00000000

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:1 | 保留 | 必须保持复位值 |
| 0 | SDRAMCLPEN | 睡眠模式期间的 SDRAM 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 SDRAM 时钟<br>1：睡眠模式期间使能 SDRAM 时钟 |

### 6.3.20. RCC AHB4 低功耗时钟门控寄存器 (AHB4LPENR)

**偏移地址：** 0x5C

**复位值：** 0x00000000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | | | | | | | IPILPEN | LTDCLPEN | res. | DMA2DLPEN | DMA2LPEN | DMA1LPEN |
| | | | | | | | | | | rw | rw | | rw | rw | rw |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:6 | 保留 | 必须保持复位值 |
| 5 | IPILPEN | 睡眠模式期间的 IPI 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 IPI 时钟<br>1：睡眠模式期间使能 IPI 时钟 |
| 4 | LTDCLPEN | 睡眠模式期间的 LTDC 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 LTDC 时钟<br>1：睡眠模式期间使能 LTDC 时钟 |
| 3 | 保留 | 必须保持复位值 |
| 2 | DMA2DLPEN | 睡眠模式期间的 DMA2D 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 DMA2D 时钟<br>1：睡眠模式期间使能 DMA2D 时钟 |
| 1 | DMA2LPEN | 睡眠模式期间的 DMA2 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 DMA2 时钟<br>1：睡眠模式期间使能 DMA2 时钟 |
| 0 | DMA1LPEN | 睡眠模式期间的 DMA1 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 DMA1 时钟<br>1：睡眠模式期间使能 DMA1 时钟 |

### 6.3.21. RCC APB1 低功耗时钟门控寄存器 (APB1LPENR)

**偏移地址：** 0x60

**复位值：** 0xF6FEC9FF

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UART8LPEN | UART7LPEN | DACLPEN | PWRLPEN | res. | CAN2LPEN | CAN1LPEN | res. | I2C3LPEN | I2C2LPEN | I2C1LPEN | UART5LPEN | UART4LPEN | UART3LPEN | UART2LPEN | res. |
| rw | rw | rw | rw | | rw | rw | | rw | rw | rw | rw | rw | rw | rw | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SPI3LPEN | SPI2LPEN | res. | | WWDGLPEN | res. | | TIM14LPEN | TIM13LPEN | TIM12LPEN | TIM7LPEN | TIM6LPEN | TIM5LPEN | TIM4LPEN | TIM3LPEN | TIM2LPEN |
| rw | rw | | | rw | | | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31 | UART8LPEN | 睡眠模式期间的 UART8 时钟使能<br>由软件置 1 和清零。 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 0：睡眠模式期间禁止 UART8 时钟<br>1：睡眠模式期间使能 UART8 时钟 |
| 30 | UART7LPEN | 睡眠模式期间的 UART7 时钟使能<br>0：睡眠模式期间禁止 UART7 时钟<br>1：睡眠模式期间使能 UART7 时钟 |
| 29 | DACLPEN | 睡眠模式期间的 DAC 接口时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 DAC 接口时钟<br>1：睡眠模式期间使能 DAC 接口时钟 |
| 28 | PWRLPEN | 睡眠模式期间的电源接口时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止电源接口时钟<br>1：睡眠模式期间使能电源接口时钟 |
| 27 | 保留 | 必须保持复位值 |
| 26 | CAN2LPEN | 睡眠模式期间的 CAN 2 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 CAN 2 时钟<br>1：睡眠模式期间使能 CAN 2 时钟 |
| 25 | CAN1LPEN | 睡眠模式期间的 CAN 1 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 CAN 1 时钟<br>1：睡眠模式期间使能 CAN 1 时钟 |
| 24 | 保留 | 必须保持复位值 |
| 23 | I2C3LPEN | 睡眠模式期间的 I2C3 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 I2C3 时钟<br>1：睡眠模式期间使能 I2C3 时钟 |
| 22 | I2C2LPEN | 睡眠模式期间的 I2C2 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 I2C2 时钟<br>1：睡眠模式期间使能 I2C2 时钟 |
| 21 | I2C1LPEN | 睡眠模式期间的 I2C1 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 I2C1 时钟<br>1：睡眠模式期间使能 I2C1 时钟 |
| 20 | UART5LPEN | 睡眠模式期间的 UART5 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 UART5 时钟<br>1：睡眠模式期间使能 UART5 时钟 |
| 19 | UART4LPEN | 睡眠模式期间的 UART4 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 UART4 时钟<br>1：睡眠模式期间使能 UART4 时钟 |
| 18 | UART3LPEN | 睡眠模式期间的 UART3 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 UART3 时钟<br>1：睡眠模式期间使能 UART3 时钟 |
| 17 | UART2LPEN | 睡眠模式期间的 UART2 时钟使能 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 由软件置 1 和清零。<br>0：睡眠模式期间禁止 UART2 时钟<br>1：睡眠模式期间使能 UART2 时钟 |
| 16 | 保留 | 必须保持复位值 |
| 15 | SPI3LPEN | 睡眠模式期间的 SPI3 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 SPI3 时钟<br>1：睡眠模式期间使能 SPI3 时钟 |
| 14 | SPI2LPEN | 睡眠模式期间的 SPI2 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 SPI2 时钟<br>1：睡眠模式期间使能 SPI2 时钟 |
| 13:12 | 保留 | 必须保持复位值 |
| 11 | WWDGLPEN | 睡眠模式期间的窗口看门狗时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止窗口看门狗时钟<br>1：睡眠模式期间使能窗口看门狗时钟 |
| 10:9 | 保留 | 必须保持复位值 |
| 8 | TIM14LPEN | 睡眠模式期间的 TIM14 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 TIM14 时钟<br>1：睡眠模式期间使能 TIM14 时钟 |
| 7 | TIM13LPEN | 睡眠模式期间的 TIM13 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 TIM13 时钟<br>1：睡眠模式期间使能 TIM13 时钟 |
| 6 | TIM12LPEN | 睡眠模式期间的 TIM12 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 TIM12 时钟<br>1：睡眠模式期间使能 TIM12 时钟 |
| 5 | TIM7LPEN | 睡眠模式期间的 TIM7 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 TIM7 时钟<br>1：睡眠模式期间使能 TIM7 时钟 |
| 4 | TIM6LPEN | 睡眠模式期间的 TIM6 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 TIM6 时钟<br>1：睡眠模式期间使能 TIM6 时钟 |
| 3 | TIM5LPEN | 睡眠模式期间的 TIM5 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 TIM5 时钟<br>1：睡眠模式期间使能 TIM5 时钟 |
| 2 | TIM4LPEN | 睡眠模式期间的 TIM4 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 TIM4 时钟<br>1：睡眠模式期间使能 TIM4 时钟 |
| 1 | TIM3LPEN | 睡眠模式期间的 TIM3 时钟使能<br>由软件置 1 和清零。 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 0：睡眠模式期间禁止 TIM3 时钟<br>1：睡眠模式期间使能 TIM3 时钟 |
| 0 | TIM2LPEN | 睡眠模式期间的 TIM2 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 TIM2 时钟<br>1：睡眠模式期间使能 TIM2 时钟 |

### 6.3.22. RCC APB2 低功耗时钟门控寄存器 (APB2LPENR)

**偏移地址：** 0x64

**复位值：** 0x08377F33

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | | | | | | | | SPI5LPEN | res. | TIM11LPEN | TIM10LPEN | TIM9LPEN |
| | | | | | | | | | | | rw | | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | SYSCFGLPEN | SPI4LPEN | SPI1LPEN | SDIOLPEN | ADC3LPEN | ADC2LPEN | ADC1LPEN | res. | | USART6LPEN | USART1LPEN | res. | | TIM8LPEN | TIM1LPEN |
| | rw | rw | rw | rw | rw | rw | rw | | | rw | rw | | | rw | rw |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:21 | 保留 | 必须保持复位值 |
| 20 | SPI5LPEN | 睡眠模式期间的 SPI5 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 SPI5 时钟<br>1：睡眠模式期间使能 SPI5 时钟 |
| 19 | 保留 | 必须保持复位值 |
| 18 | TIM11LPEN | 睡眠模式期间的 TIM11 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 TIM11 时钟<br>1：睡眠模式期间使能 TIM11 时钟 |
| 17 | TIM10LPEN | 睡眠模式期间的 TIM10 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 TIM10 时钟<br>1：睡眠模式期间使能 TIM10 时钟 |
| 16 | TIM9LPEN | 睡眠模式期间的 TIM9 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 TIM9 时钟<br>1：睡眠模式期间使能 TIM9 时钟 |
| 15 | 保留 | 必须保持复位值 |
| 14 | SYSCFGLPEN | 睡眠模式期间的系统配置控制器时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止系统配置控制器时钟<br>1：睡眠模式期间使能系统配置控制器时钟 |
| 13 | SPI4LPEN | 睡眠模式期间的 SPI4 时钟使能 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 由软件置 1 和清零。<br>0：睡眠模式期间禁止 SPI4 时钟<br>1：睡眠模式期间使能 SPI4 时钟 |
| 12 | SPI1LPEN | 睡眠模式期间的 SPI1 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 SPI1 时钟<br>1：睡眠模式期间使能 SPI1 时钟 |
| 11 | SDIOLPEN | 睡眠模式期间的 SDIO 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 SDIO 模块时钟<br>1：睡眠模式期间使能 SDIO 模块时钟 |
| 10 | ADC3LPEN | 睡眠模式期间的 ADC 3 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 ADC 3 时钟<br>1：睡眠模式期间使能 ADC 3 时钟 |
| 9 | ADC2LPEN | 睡眠模式期间的 ADC2 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 ADC2 时钟<br>1：睡眠模式期间使能 ADC2 时钟 |
| 8 | ADC1LPEN | 睡眠模式期间的 ADC1 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 ADC1 时钟<br>1：睡眠模式期间使能 ADC1 时钟 |
| 7:6 | 保留 | 必须保持复位值 |
| 5 | USART6LPEN | 睡眠模式期间的 USART6 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 USART6 时钟<br>1：睡眠模式期间使能 USART6 时钟 |
| 4 | USART1LPEN | 睡眠模式期间的 USART1 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 USART1 时钟<br>1：睡眠模式期间使能 USART1 时钟 |
| 3:2 | 保留 | 必须保持复位值 |
| 1 | TIM8LPEN | 睡眠模式期间的 TIM8 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 TIM8 时钟<br>1：睡眠模式期间使能 TIM8 时钟 |
| 0 | TIM1LPEN | 睡眠模式期间的 TIM1 时钟使能<br>由软件置 1 和清零。<br>0：睡眠模式期间禁止 TIM1 时钟<br>1：睡眠模式期间使能 TIM1 时钟 |

### 6.3.23. RCC 备份域控制寄存器 (BDCR)

**偏移地址：** 0x70

**复位值：** 0x00008201

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | | | RTCSEL | | res. | | | | | LSEBYP | LSERDY | LSEON |
| | | | | | | rw | | | | | | | rw | ro | rw |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:17 | 保留 | 必须保持复位值 |
| 16 | BDRST | BDCR 寄存器软复位<br>0：不复位<br>1：复位 BDCR 寄存器 |
| 15:10 | 保留 | 必须保持复位值 |
| 9:8 | RTCSEL | RTC 时钟源选择<br>由软件置 1，用于选择 RTC 的时钟源。选择 RTC 时钟源后，除非复位该寄存器，否则不可再更改。可使用 BDRST 位对该寄存器进行复位。<br>00：无时钟<br>01：LSE 振荡器时钟用作 RTC 时钟<br>10：LSI 振荡器时钟用作 RTC 时钟<br>11：HSE 振荡器时钟分频用作 RTC 时钟 |
| 7:3 | 保留 | 必须保持复位值 |
| 2 | LSEBYP | 外部低速振荡器旁路<br>由软件置 1 和清零，用于旁路调试模式下的振荡器。只有在禁止 LSE 时钟后才能写入该位。<br>0：不旁路 LSE 振荡器<br>1：旁路 LSE 振荡器 |
| 1 | LSERDY | 外部低速振荡器就绪<br>由硬件置 1 和清零，用于指示外部 32 kHz 振荡器已稳定。<br>0：LSE 时钟未就绪<br>1：LSE 时钟就绪 |
| 0 | LSEON | 外部低速振荡器使能<br>由软件置 1 和清零。<br>0：LSE 时钟关闭<br>1：LSE 时钟开启 |

### 6.3.24. RCC 时钟控制和状态寄存器 (CSR)

**偏移地址：** 0x74

**复位值：** 0x0C000002

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LPWRRSTF | WWDGRSTF | IWDGRSTF | SFTRSTF | PORRSTF | PINRSTF | MATTRSTF | RMVF | BATTRSTF | GLBRSTF | res. | | | | | |
| ro | ro | ro | ro | ro | ro | ro | wo | ro | ro | | | | | | |

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 31 | LPWRRSTF | 低功耗复位标志<br>发生低功耗管理复位时，由硬件置 1。<br>通过写入 RMVF 位清零。<br>0：未发生低功耗管理复位<br>1：发生低功耗管理复位 |
| 30 | WWDGRSTF | 窗口看门狗复位标志<br>发生窗口看门狗复位时，由硬件置 1。<br>通过写入 RMVF 位清零。<br>0：未发生窗口看门狗复位<br>1：发生窗口看门狗复位 |
| 29 | IWDGRSTF | 独立看门狗复位标志<br>发生来自 VDD 域的独立看门狗复位时，由硬件置 1。<br>通过写入 RMVF 位清零。<br>0：未发生看门狗复位<br>1：发生看门狗复位 |
| 28 | SFTRSTF | 系统软件复位标志<br>发生软件复位时，由硬件置 1。<br>通过写入 RMVF 位清零。<br>0：未发生软件复位<br>1：发生软件复位 |
| 27 | PORRSTF | POR/PDR 复位标志<br>发生 POR/PDR 复位时，由硬件置 1。<br>通过写入 RMVF 位清零。<br>0：未发生 POR/PDR 复位<br>1：发生 POR/PDR 复位 |
| 26 | PINRSTF | 引脚复位标志<br>发生来自 NRST 引脚的复位时，由硬件置 1。<br>通过写入 RMVF 位清零。<br>0：未发生来自 NRST 引脚的复位<br>1：发生来自 NRST 引脚的复位 |
| 25 | MATTRSTF | 主 LDO 域复位标志<br>发生主 LDO 域复位时，由硬件置 1。<br>通过写入 RMVF 位清零。<br>0：未发生主 LDO 域复位<br>1：发生主 LDO 域复位 |
| 24 | RMVF | 清除复位标志<br>由软件置 1，用于将复位标志清零。<br>0：无操作<br>1：清零复位标志 |
| 23 | BATTRSTF | 备份域复位标志<br>发生备份域复位时，由硬件置 1。<br>通过写入 RMVF 位清零。<br>0：未发生备份域复位 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 1：发生备份域复位 |
| 22 | GLBRSTF | 全局复位标志(比 SFTRSTF 复位范围大)<br>发生全局软复位时，由硬件置 1。<br>通过写入 RMVF 位清零。<br>0：未发生全局软复位<br>1：发生软全局软复位 |
| 21:2 | 保留 | 必须保持复位值 |
| 1 | LSIRDY | 内部低速振荡器就绪<br>由硬件置 1 和清零，用于指示内部 32.768 kHz 振荡器已稳定。<br>0：LSI RC 振荡器未就绪<br>1：LSI RC 振荡器就绪 |
| 0 | 保留 | 必须保持复位值 |

### 6.3.25. RCC 扩频时钟生成寄存器 (SSCGR)

**偏移地址：** 0x80

**复位值：** 0x00000000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SSCGEN | SPREADSEL | res. | | INCSTEP | | | | | | | | | | | |
| rw | rw | | | rw | | | | | | | | | | | |

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 31 | SSCGEN | 扩频调制使能<br>由软件置 1 和清零。<br>0：扩频调制禁止。(CR[24]=PLL1ON 位清零后写入)<br>1：扩频调制使能。(CR[24]=PLL1ON 位置 1 前写入) |
| 30 | SPREADSEL | 扩频选择<br>由软件置 1 和清零。<br>CR[24]=PLL1ON 位置 1 前写入。<br>0：中心扩频<br>1：向下扩频 |
| 29:28 | 保留 | 必须保持复位值 |
| 27:13 | INCSTEP | 增量步长<br>由软件置 1 和清零。CR[24]=PLL1ON 位置 1 前写入。<br>设置调制曲线振幅输入。 |
| 12:0 | MODPER | 调制周期<br>由软件置 1 和清零。CR[24]=PLL1ON 位置 1 前写入。<br>设置调制曲线周期输入。 |

### 6.3.26. RCC PLL2 配置寄存器 (PLL2CFGR)

**偏移地址：** 0x84

**复位值：** 0x24043000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PLL2_R | | | | PLL2_Q | | | | res. | | | | | PLL2_PEN | PLL2_P | |
| rw | | | | rw | | | | | | | | | rw | rw | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | PLL2_N | | | | | | | | | res. | | | | | |
| | rw | | | | | | | | | | | | | | |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:28 | PLL2_R | PLL2_R：适用于 PLL2 分频系数<br>由软件置 1 和清零，用于控制 PLL2 时钟频率。这些位仅在 PLL2 已禁止时写入。<br>PLL2_R 时钟频率 = VCO 频率 / PLL2_R 并且 2 ≤ PLL2_R ≤ 15<br>0000：PLL2_R = 0，错误配置<br>0001：PLL2_R = 1，错误配置<br>0010：PLL2_R = 2<br>…<br>1111：PLL2_R = 15 |
| 27:24 | PLL2_Q | PLL2_Q：适用于 PLL2 分频系数<br>由软件置 1 和清零，用于控制 PLL2 时钟频率。这些位仅在 PLL2 已禁止时写入。<br>PLL2_Q 时钟频率 = VCO 频率 / PLL2_Q 并且 2 ≤ PLL2_Q ≤ 15<br>0000：PLL2_Q = 0，错误配置<br>0001：PLL2_Q = 1，错误配置<br>0010：PLL2_Q = 2<br>0011：PLL2_Q = 3<br>0100：PLL2_Q = 4<br>…<br>1111：PLL2_Q = 15 |
| 23:19 | 保留 | 必须保持复位值 |
| 18 | PLL2_PEN | PLL2P Bypass 配置<br>0：旁路 PLL2_P 分频系数，系统时钟频率 = VCO 频率<br>1：使能 PLL2_P 分频系数 |
| 17:16 | PLL2_P | 适用于 PLL2 分频系数<br>由软件置 1 和清零，用于控制 PLL2 输出时钟的频率。这些位只能在 PLL2 已禁止时写入。<br>PLL2_P 输出时钟频率 = VCO 频率 / PLL2_P 并且 PLL2_P = 2、4、6 或 8<br>00：PLL2_P = 2<br>01：PLL2_P = 4<br>10：PLL2_P = 6 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 11：PLL2_P = 8 |
| 15 | 保留 | 必须保持复位值 |
| 14:6 | PLL2_N | 适用于 VCO 的 PLL2 倍频系数<br>由软件置 1 和清零，用于控制 VCO 的倍频系数。这些位只能在 PLL2 已禁止时写入。写入这些位时只允许使用半字和字访问。<br>VCO 输出频率 = VCO 输入频率 × PLL2_N 并且 192 ≤ PLL2_N ≤ 432<br>000000000 ： PLL2_N = 0，错误配置<br>000000001：PLL2_N = 1，错误配置<br>…<br>011000000：PLL2_N = 192<br>…<br>110110000：PLL2_N = 432<br>110110001：PLL2_N = 433，错误配置<br>…<br>111111111：PLL2_N = 511，错误配置 |
| 5:0 | 保留 | 必须保持复位值 |

### 6.3.27. RCC 专用时钟配置寄存器 (DCKCFGR)

**偏移地址：** 0x8C

**复位值：** 0x20000004

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

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:19 | 保留 | 必须保持复位值 |
| 28 | SDIO_SEL | SDIO 时钟源选择<br>0：使用 REFCLK_SEL 时钟作为 SDIO 时钟源<br>1：使用 APB2 时钟作为 SDIO 时钟源 |
| 27 | REFCLK_SEL | USB 参考时钟源选择<br>0：使用 PLL3 作为参考时钟源<br>1：使用 PLL1_Q 作为参考时钟源 |
| 16:2 | 保留 | 必须保持复位值 |
| 1:0 | LTDC_SEL | LTDC PIXCLK 时钟选择<br>00：PLL2_R<br>01：PLL2_Q<br>10：PLL2_P<br>11：PLL1_R |

### 6.3.28. RCC PLL3 EXT 时钟配置寄存器 (PLL3EXTCFGR)

**偏移地址：** 0xD4

**复位值：** 0x00083147

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

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:22 | 保留 | 必须保持复位值 |
| 21:0 | FREQ_EXT | CLK_EXT 输出频率配置字，可配置 24/36/48MHz 时钟数据 |

### 6.3.29. RCC PLL3 参数加载寄存器 (PLL3LOADR)

**偏移地址：** 0xE4

**复位值：** 0x00000000

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:2 | 保留 | 必须保持复位值 |
| 1 | FREQ_EXT_LOAD | 完成 RCC_PLL3EXTCFGR 寄存器配置后，软件需将 FREQ_EXT_LOAD 置 1 加载并清 0 |
| 0 | 保留 | 必须保持复位值 |

### 6.3.30. RCC 子系统时钟分频寄存器 (SPRER)

**偏移地址：** 0xE8

**复位值：** 0x00000000

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 必须保持复位值 |
| 7:4 | SDRAM_PRE | SDRAM 子系统时钟分频配置，由 CPUCLK 分频得到<br>0xxx: CPUCLK not divided |

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 1000: CPUCLK divided by 2<br>1001: CPUCLK divided by 3<br>1010: CPUCLK divided by 4<br>1011: CPUCLK divided by 8<br>1100: CPUCLK divided by 16<br>1101: CPUCLK divided by 128<br>1110: CPUCLK divided by 256<br>1111: CPUCLK divided by 512 |
| 3:0 | CPU_PRE | CPU 子系统时钟分频配置，由 PLLCLK 分频得到<br>0xxx: PLLCLK not divided<br>1000: PLLCLK divided by 2<br>1001: PLLCLK divided by 4<br>1010: PLLCLK divided by 8<br>1011: PLLCLK divided by 16<br>1100: PLLCLK divided by 64<br>1101: PLLCLK divided by 128<br>1110: PLLCLK divided by 256<br>1111: PLLCLK divided by 512 |

# 7. 通用 I/O(GPIO)

## 7.1. GPIO 简介

每个通用 I/O 端口包括:

4 个 32 位配置寄存器 (GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR 和 GPIOx_PUPDR)。

2 个 32 位数据寄存器(GPIOx_IDR 和 GPIOx_ODR)和 1 个 32 位置位/复位寄存器 (GPIOx_BSRR)。

此外，所有 GPIO 都包括 1 个 32 位锁定寄存器 (GPIOx_LCKR) 和 2 个 32 位复用功能选择寄存器（GPIOx_AFRH 和 GPIOx_AFRL）以及 1 个驱动能力配置寄存器(GPIOx_DSR)。

## 7.2. GPIO 主要特性

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

表 7-1 端口位配置表

| MODE(i)[1:0] | OTYPER(i) | OSPEED(i)[1:0] | DSR(i)[1:0] | PUPD(i)[1:0] | | I/O 配置 | |
|---|---|---|---|---|---|---|---|
| 01 | 0 | SPEED[1:0] | DSR[1:0] | 0 | 0 | GP 输出 | PP |
| | 0 | | | 0 | 1 | GP 输出 | PP + PU |
| | 0 | | | 1 | 0 | GP 输出 | PP + PD |
| | 0 | | | 1 | 1 | 保留 | |
| | 1 | | | 0 | 0 | GP 输出 | OD |
| | 1 | | | 0 | 1 | GP 输出 | OD + PU |
| | 1 | | | 1 | 0 | GP 输出 | OD + PD |
| | 1 | | | 1 | 1 | 保留(GP 输出 OD) | |
| 10 | 0 | SPEED[1:0] | DSR[1:0] | 0 | 0 | AF | PP |
| | 0 | | | 0 | 1 | AF | PP + PU |
| | 0 | | | 1 | 0 | AF | PP + PD |
| | 0 | | | 1 | 1 | 保留 | |
| | 1 | | | 0 | 0 | AF | OD |
| | 1 | | | 0 | 1 | AF | OD + PU |

| MODE(i)[1:0] | OTYPER(i) | OSPEED(i)[1:0] | | DSR(i)[1:0] | | PUPD(i)[1:0] | | I/O 配置 | |
|---|---|---|---|---|---|---|---|---|---|
| | 1 | | | | | 1 | 0 | AF | OD + PD |
| | 1 | | | | | 1 | 1 | 保留 | |
| 00 | x | x | x | x | x | 0 | 0 | 输入 | 浮空 |
| | x | x | x | x | x | 0 | 1 | 输入 | PU |
| | x | x | x | x | x | 1 | 0 | 输入 | PD |
| | x | x | x | x | x | 1 | 1 | 保留(输入浮空) | |
| 11 | x | x | x | x | x | 0 | 0 | 输入/输出 | 模拟 |
| | x | x | x | x | x | 0 | 1 | 保留 | |
| | x | x | x | x | x | 1 | 0 | | |
| | x | x | x | x | x | 1 | 1 | | |

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

## 7.3. GPIO 功能描述

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

- 输入浮空；
- 输入上拉；
- 输入下拉；
- 模拟功能；
- 具有上拉或下拉功能的开漏输出；
- 具有上拉或下拉功能的推挽输出；
- 具有上拉或下拉功能的复用功能推挽；
- 具有上拉或下拉功能的复用功能开漏。

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

### 7.3.1. 通用 I/O(GPIO)

在复位期间及复位刚刚完成后，复用功能尚未激活，大多数 I/O 端口被配置为输入浮空模式。复位后，调试引脚处于复用功能上拉/下拉状态：

- PA15：JTDI 处于上拉状态；
- PA14：JTCK/SWCLK 处于下拉状态；
- PA13：JTMS/SWDAT 处于上拉状态；
- PB4：NJTRST 处于上拉状态；
- PB3：JTDO 处于浮空状态。

当引脚配置为输出后，写入到输出数据寄存器 (GPIOx_ODR) 的值将在 I/O 引脚上输出。可以在推挽模式下或开漏模式下使用输出驱动器（仅驱动低电平，高电平为高阻态）。

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

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

### 7.3.2. I/O 引脚复用器和映射

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

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

- 完成复位后，所有 I/O 都会连接到系统的复用功能 0 (AF0)。
- 外设的复用功能映射到 AF1 至 AF13。
- 内核 EVENTOUT 映射到 AF15。

下面的图对此结构进行了说明。

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

要将 I/O 配制成所需功能，请按照以下步骤操作：

- **系统功能**

  将 I/O 连接到 AF0，然后根据所用功能进行配置：

  - JTAG/SWD：在各器件复位后，会将这些引脚指定为专用引脚，可供片上调试模块立即使用（不受 GPIO 控制器控制）。
  - MCO1 和 MCO2：这些引脚必须配置为复用功能模式。

*注意： 可禁止部分或全部 JTAG/SWD 引脚，以释放相关联的引脚供 GPIO 使用。*

表 7-2 灵活的 SWJ-DP 引脚分配

| 可用的调试端口 | 用到的 SWJ I/O 引脚 | | | | |
|---|---|---|---|---|---|
| | PA13/ JTMS/ SWDIO | PA14/ JTCK/ SWCLK | PA15/ JTDI | PB3/ JTDO | PB4/ NJTRST |
| 全部 SWJ (JTAG-DP + SW-DP) - 复位状态 | X | X | X | X | X |
| 全部 SWJ (JTAG-DP + SW-DP)，但不包括 NJTRST | X | X | X | X | |
| 禁止 JTAG-DP 和使能 SW-DP | X | X | | | |
| 禁止 JTAG-DP 和禁止 SW-DP | 已释放 | | | | |

- **GPIO**

  在 GPIOx_MODER 寄存器中将所需 I/O 配置为输出或输入。

- **外设复用功能**

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

  对于其它外设：

  - 在 GPIOx_MODER 寄存器中将所需 I/O 配置为复用功能
  - 通过 GPIOx_OTYPER、GPIOx_PUPDR 和 GPIOx_OSPEEDER 寄存器，分别选择类型、上拉/下拉以及输出速度
  - 在 GPIOx_AFRL 或 GPIOx_AFRH 寄存器中，将 I/O 连接到所需 AFx

- **EVENTOUT**

  配置用于输出内核 EVENTOUT 信号的 I/O 引脚（通过将其连接到 AF15）。

有关系统和外设的复用功能 I/O 引脚映射的详细信息，请参见数据手册中的“复用功能映射”表。

### 7.3.3. I/O 端口控制寄存器

每个 GPIO 有 4 个 32 位存储器映射的控制寄存器(GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR)，可配置多达 16 个 I/O。GPIOx_MODER 寄存器用于选择 I/O 方向（输入、输出、AF、模拟）。GPIOx_OTYPER 和 GPIOx_OSPEEDR 寄存器分别用于选择输出类型（推挽或开漏）和速度 (无论采用哪种 I/O 方向，都会直接将 I/O 速度引脚连接到相应的 GPIOx_OSPEEDR 寄存器位)。无论采用哪种 I/O 方向，GPIOx_PUPDR 寄存器都用于选择上拉/下拉。

### 7.3.4. I/O 端口数据寄存器

每个 GPIO 都具有 2 个 16 位数据寄存器：输入和输出数据寄存器（GPIOx_IDR 和 GPIOx_ODR）。GPIOx_ODR 用于存储待输出数据，可对其进行读/写访问。通过 I/O 输入的数据存储到输入数据寄存器 (GPIOx_IDR) 中，它是一个只读寄存器。

有关寄存器说明的详细信息，请参见 GPIO 端口输入数据寄存器 (GPIOx_IDR)(x = A..I) 和 GPIO 端口输出数据寄存器 (GPIOx_ODR) (x = A..I)。

### 7.3.5. I/O 数据位操作

置位复位寄存器 (GPIOx_BSRR) 是一个 32 位寄存器，它允许应用程序在输出数据寄存器(GPIOx_ODR)中对各个单独的数据位执行置位和复位操作。置位复位寄存器的大小是 GPIOx_ODR 的二倍。

GPIOx_ODR 中的每个数据位对应于 GPIOx_BSRR 中的两个控制位：BSRR(i) 和 BSRR(i+SIZE)。当写入 1 时，BSRR(i) 位会置位对应的 ODR(i) 位。当写入 1 时，BSRR(i+SIZE) 位会清零 ODR(i) 对应的位。

在 GPIOx_BSRR 中向任何位写入 0 都不会对 GPIOx_ODR 中的对应位产生任何影响。如果在 GPIOx_BSRR 中同时尝试对某个位执行置位和清零操作，则置位操作优先。

使用 GPIOx_BSRR 寄存器更改 GPIOx_ODR 中各个位的值是一个“单次”操作，不会锁定 GPIOx_ODR 位。随时都可以直接访问 GPIOx_ODR 位。GPIOx_BSRR 寄存器提供了一种执行原子按位处理的方法。

在对 GPIOx_ODR 进行位操作时，软件无需禁止中断：在一次原子 AHB1 写访问中，可以修改一个或多个位。

### 7.3.6. GPIO 锁定机制

通过将特定的写序列应用到 GPIOx_LCKR 寄存器，可以冻结 GPIO 控制寄存器。冻结的寄存器包括 GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR、GPIOx_AFRL 和 GPIOx_AFRH。

要对 GPIOx_LCKR 寄存器执行写操作，必须应用特定的写/读序列。当正确的 LOCK 序列应用到此寄存器的第 16 位后，会使用 LCKR[15:0] 的值来锁定 I/O 的配置（在写序列期间，LCKR[15:0] 的值必须相同）。将 LOCK 序列应用到某个端口位后，在执行下一次复位之前，将无法对该端口位的值进行修改。每个 GPIOx_LCKR 位都会冻结控制寄存器（GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR、GPIOx_AFRL 和 GPIOx_AFRH）中的对应位。

LOCK 序列只能通过对 GPIOx_LCKR 寄存器进行字（32 位长）访问的方式来执行，因为 GPIOx_LCKR 的第 16 位必须与 [15:0] 位同时置位。

### 7.3.7. I/O 复用功能输入/输出

有两个寄存器可用来从每个 I/O 可用的 16 个复用功能输入/输出中进行选择。借助这些寄存器，可根据应用程序的要求将某个复用功能连接到其它某个引脚。

这意味着可使用 GPIOx_AFRL 和 GPIOx_AFRH 复用功能寄存器在每个 GPIO 上复用多个可用的外设功能。这样一来，应用程序可为每个 I/O 选择任何一个可用功能。由于 AF 选择信号由复用功能输入和复用功能输出共用，所以只需为每个 I/O 的复用功能输入/输出选择一个通道即可。

要了解在每个 GPIO 引脚上复用了哪些功能，请参见数据手册。

*注意：对于每个 I/O 而言，应用程序一次只能为其选择一个可用的外设功能。*

### 7.3.8. 外部中断线/唤醒线

所有端口都具有外部中断功能。要使用外部中断线，必须将端口配置为输入模式。

### 7.3.9. 输入配置

对 I/O 端口进行编程作为输入时：

- 输出缓冲器被关闭；
- 施密特触发器输入被打开；
- 根据 GPIOx_PUPDR 寄存器中的值决定是否打开上拉和下拉电阻；
- 输入数据寄存器每隔 1 个 AHB1 时钟周期对 I/O 引脚上的数据进行一次采样；
- 对输入数据寄存器的读访问可获取 I/O 状态。

下图说明了 I/O 端口位的输入配置。

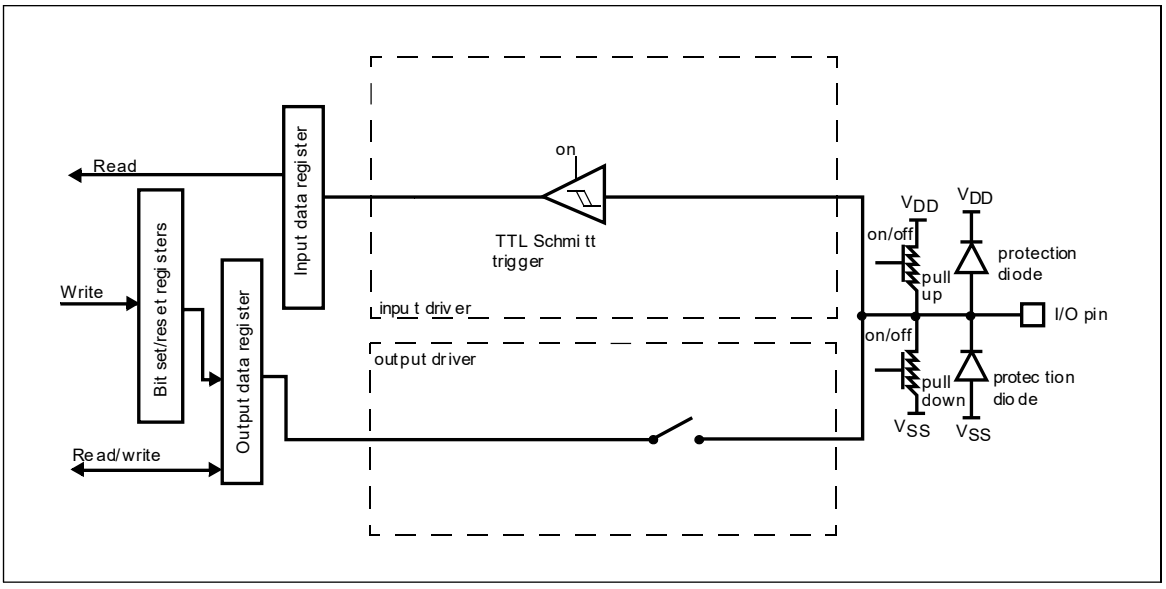



图 7-1 输入浮空/上拉/下拉配置

### 7.3.10. 输出配置

对 I/O 端口进行编程作为输出时：

- 输出缓冲器被打开：
- 开漏模式：输出寄存器中的“0”可激活 N-MOS，而输出寄存器中的“1”会使端口保持高组态 (Hi-Z)（P-MOS 始终不激活）。
- 推挽模式：输出寄存器中的“0”可激活 N-MOS，而输出寄存器中的“1”可激活 P-MOS。
- 施密特触发器输入被打开。

- 根据 GPIOx_PUPDR 寄存器中的值决定是否打开弱上拉电阻和下拉电阻。
- 输入数据寄存器每隔 1 个 AHB1 时钟周期对 I/O 引脚上的数据进行一次采样。
- 对输入数据寄存器的读访问可获取 I/O 状态。
- 对输出数据寄存器的读访问可获取最后的写入值。

下图说明了 I/O 端口位的输出配置。

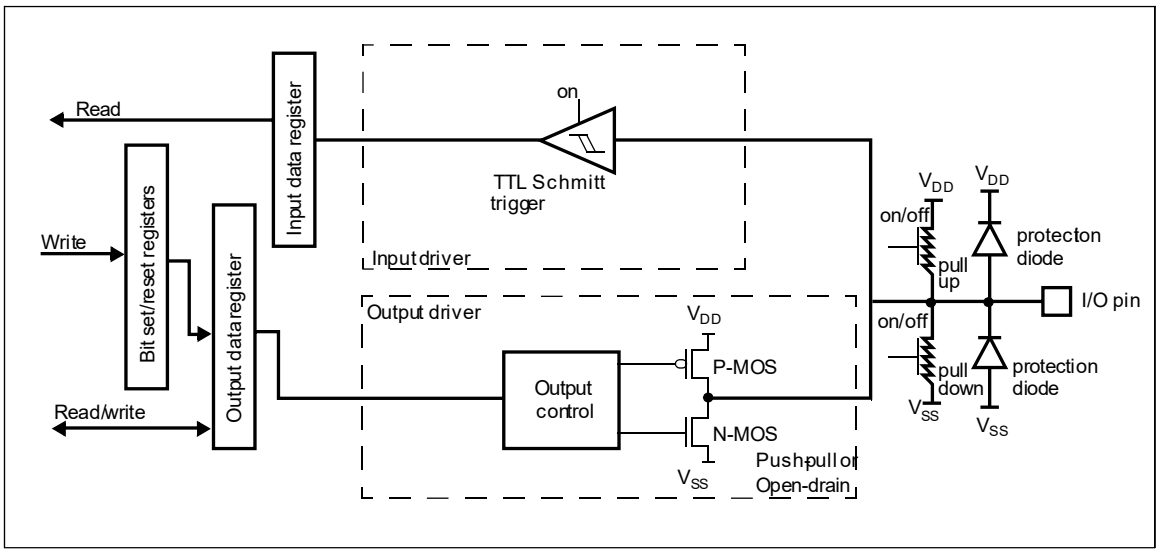



图 7-2 输出配置

### 7.3.11. 复用功能配置

对 I/O 端口进行编程作为复用功能时:

- 可将输出缓冲器配置为开漏或推挽。
- 输出缓冲器由来自外设的信号驱动（发送器使能和数据）。
- 施密特触发器输入被打开。
- 根据 GPIOx_PUPDR 寄存器中的值决定是否打开弱上拉电阻和下拉电阻。
- 输入数据寄存器每隔 1 个 AHB1 时钟周期对 I/O 引脚上的数据进行一次采样。
- 对输入数据寄存器的读访问可获取 I/O 状态。

下图说明了 I/O 端口位的复用功能配置。

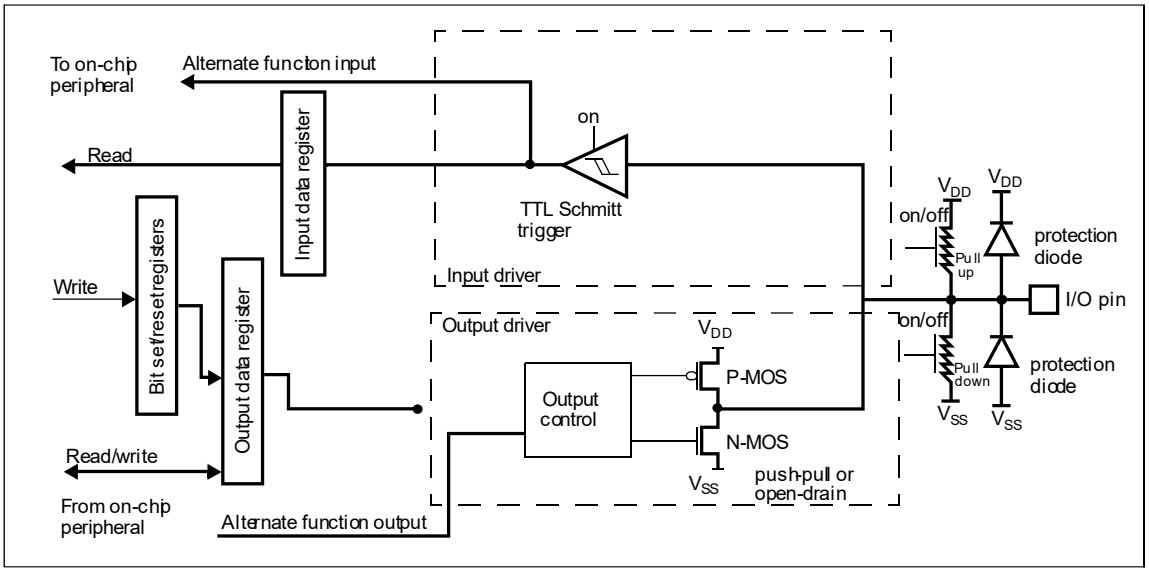



图 7-3 复用功能配置

## 7.3.12. 模拟配置

对 I/O 端口进行编程作为模拟配置时：

- 输出缓冲器被禁止；
- 施密特触发器输入停用，I/O 引脚的每个模拟输入的功耗变为零。施密特触发器的输出被强制处理为恒定值 (0)；
- 弱上拉和下拉电阻被关闭；
- 对输入数据寄存器的读访问值为“0”。

*注意： 在模拟配置中，I/O 引脚不能为 5 V 容忍。*

下图说明了 I/O 端口位的高阻态模拟输入配置。

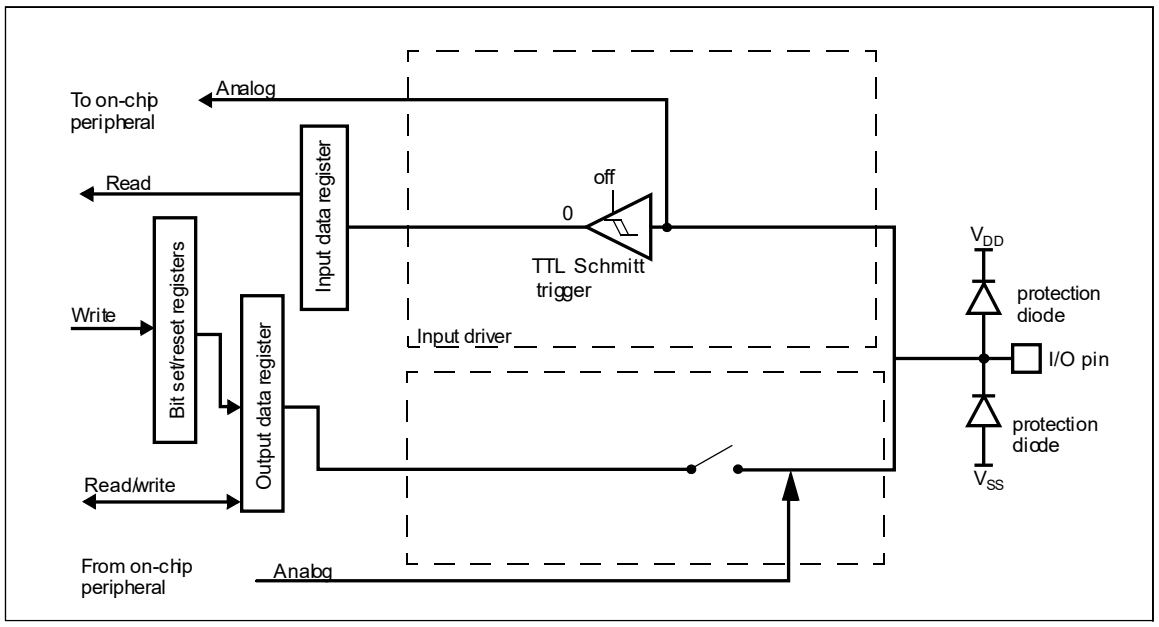



图 7-4 高阻态模拟配置

### 7.3.13. 将 HSE 振荡器引脚用作 GPIO PH0/PH1 端口引脚

当 HSE 振荡器处于关闭状态时，可分别将 HSE 振荡器引脚 OSC_IN 和 OSC_OUT 用作通用 PH0 I/O 和 PH1 I/O。(完成复位后，HSE 振荡器处于关闭状态)。当 HSE 振荡器处于开启状态时，PH0/PH1 I/O 只能被配置为 OSC_IN/OSC_OUT HSE 振荡器引脚。可通过在 RCC_CR 寄存器中将 HSEON 位置 1 来完成此操作。

## 7.4. GPIO 寄存器

本节对 GPIO 寄存器进行了详细介绍。

可通过字节（8 位)、半字（16 位）或字（32 位）对 GPIO 寄存器进行访问。

### 7.4.1. GPIO 端口模式寄存器(GPIOx_MODER)(x = A...I)

偏移地址： 0x00

复位值:

- 0xA800 0000（端口 A）
- 0x0000 0280（端口 B）
- 0x0000 0000（其它端口）

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MODER15[1:0] | | MODER14[1:0] | | MODER13[1:0] | | MODER12[1:0] | | MODER11[1:0] | | MODER10[1:0] | | MODER9[1:0] | | MODER8[1:0] | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MODER7[1:0] | | MODER6[1:0] | | MODER5[1:0] | | MODER4[1:0] | | MODER3[1:0] | | MODER2[1:0] | | MODER1[1:0] | | MODER0[1:0] | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 2y:2y+1 | MODERy[1:0] | 端口 x 配置位(y = 0..15)。<br>这些位通过软件写入，用于配置 I/O 方向模式。<br>00：输入（复位状态）<br>01：通用输出模式<br>10：复用功能模式<br>11：模拟模式 |

### 7.4.2. GPIO 端口输出类型寄存器 (GPIOx_OTYPER) (x = A..I)

偏移地址： 0x04

复位值： 0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:16 | 保留 | 必须保持复位值 |
| 位 15:0 | OTy[1:0] | 端口 x 配置位(y = 0..15)。<br>这些位通过软件写入，用于配置 I/O 端口的输出类型。<br>0：输出推挽（复位状态）<br>1：输出开漏 |

### 7.4.3. GPIO 端口输出速度寄存器 (GPIOx_OSPEEDR) (x = A..I)

偏移地址： 0x08

复位值:

- 0x0C00 0000（端口 A）
- 0x0000 00C0（端口 B）
- 0x0000 0000（其它端口）

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OSPEEDR15 [1:0] | | OSPEEDR14 [1:0] | | OSPEEDR13 [1:0] | | OSPEEDR12 [1:0] | | OSPEEDR11 [1:0] | | OSPEEDR10 [1:0] | | OSPEEDR9 [1:0] | | OSPEEDR8 [1:0] | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OSPEEDR7 [1:0] | | OSPEEDR6 [1:0] | | OSPEEDR5 [1:0] | | OSPEEDR4 [1:0] | | OSPEEDR3 [1:0] | | OSPEEDR2 [1:0] | | OSPEEDR1 [1:0] | | OSPEEDR0 [1:0] | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 2y+1:2y | OSPEEDRy[1:0] | 端口 x 配置位(y = 0..15)。<br>这些位通过软件写入，用于配置 I/O 输出速度。<br>00：低速<br>01：中速<br>10：快速<br>11：高速 |

### 7.4.4. GPIO 端口上拉/下拉寄存器 (GPIOx_PUPDR) (x = A..I)

偏移地址： 0x0C

复位值:

- 0x6400 0000（端口 A）
- 0x0000 0100（端口 B）
- 0x0000 0000（其它端口）

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PUPDR15[1:0] | | PUPDR14[1:0] | | PUPDR13[1:0] | | PUPDR12[1:0] | | PUPDR11[1:0] | | PUPDR10[1:0] | | PUPDR9[1:0] | | PUPDR8[1:0] | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PUPDR7[1:0] | | PUPDR6[1:0] | | PUPDR5[1:0] | | PUPDR4[1:0] | | PUPDR3[1:0] | | PUPDR2[1:0] | | PUPDR1[1:0] | | PUPDR0[1:0] | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 2y+1:2y | PUPDRy[1:0] | 端口 x 配置位 (y = 0..15)。<br>这些位通过软件写入，用于配置 I/O 上拉或下拉。<br>00：无上拉或下拉<br>01：上拉<br>10：下拉<br>11：保留 |

### 7.4.5. GPIO 端口输入数据寄存器 (GPIOx_IDR) (x = A..I)

偏移地址： 0x10

复位值： 0x0000 XXXX（其中 X 表示未定义）

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IDR15 | IDR14 | IDR13 | IDR12 | IDR11 | IDR10 | IDR9 | IDR8 | IDR7 | IDR6 | IDR5 | IDR4 | IDR3 | IDR2 | IDR1 | IDR0 |
| r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:16 | 保留 | 必须保持复位值 |
| 位 15:0 | IDRy | 端口输入数据(y = 0..15)。<br>这些位为只读。它们包含相应 I/O 端口的输入值。 |

### 7.4.6. GPIO 端口输出数据寄存器 (GPIOx_ODR) (x = A..I)

偏移地址： 0x14

复位值： 0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ODR15 | ODR14 | ODR13 | ODR12 | ODR11 | ODR10 | ODR9 | ODR8 | ODR7 | ODR6 | ODR5 | ODR4 | ODR3 | ODR2 | ODR1 | ODR0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:16 | 保留 | 必须保持复位值 |
| 位 15:0 | ODRy | 端口输出数据(y = 0..15)。<br>这些位可通过软件读取和写入。<br>*注：对于原子置位/复位，通过写入 GPIOx_BSRR 或 GPIOx_BRR 寄存器，可分别置位和/或复位 ODR 位(x = A..I)。* |

### 7.4.7. GPIO 端口置位/复位寄存器 (GPIOx_BSRR) (x = A..I)

偏移地址： 0x18

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BR15 | BR14 | BR13 | BR12 | BR11 | BR10 | BR9 | BR8 | BR7 | BR6 | BR5 | BR4 | BR3 | BR2 | BR1 | BR0 |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BS15 | BS14 | BS13 | BS12 | BS11 | BS10 | BS9 | BS8 | BS7 | BS6 | BS5 | BS4 | BS3 | BS2 | BS1 | BS0 |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:16 | Bry | 端口 x 复位位 y (y = 0..15)。<br>这些位为只写。读取这些位可返回值 0x0000。<br>0：不会对相应的 ODRx 位执行任何操作<br>1：复位相应的 ODRx 位 |
| 位 15:0 | Bsy | 端口 x 置位位 y (y= 0..15)。<br>这些位为只写。读取这些位可返回值 0x0000。<br>0：不会对相应的 ODRx 位执行任何操作<br>1：置位相应的 ODRx 位 |

### 7.4.8. GPIO 端口配置锁定寄存器 (GPIOx_LCKR) (x = A..I)

当正确的写序列应用到第 16 位 (LCKK) 时，此寄存器将用于锁定端口位的配置。位[15:0] 的值用于锁定 GPIO 的配置。在写序列期间，不能更改 LCKR[15:0] 的值。将 LOCK 序列应用到某个端口位后，在执行下一次 MCU 复位或外设复位之前，将无法对该端口位的值进行修改。

*注：可使用特定的写序列对 GPIOx_LCKR 寄存器执行写操作。在此锁定序列期间只允许使用字访问（32 位长）。*

*每个锁定位冻结一个特定的配置寄存器（控制寄存器和复用功能寄存器）。*

偏移地址： 0x1C

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | LCKK |
| | | | | | | | | | | | | | | | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LCK15 | LCK14 | LCK13 | LCK12 | LCK11 | LCK10 | LCK9 | LCK8 | LCK7 | LCK6 | LCK5 | LCK4 | LCK3 | LCK2 | LCK1 | LCK0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:17 | 保留 | 必须保持复位值 |
| 位 16 | LCKK | 锁定键<br>可随时读取此位。可使用锁定键写序列对其进行修改。<br>0：端口配置锁定键未激活<br>1：端口配置锁定键已激活。在下一次 MCU 复位或外设复位之前，GPIOx_LCKR 寄存器始终处于锁定状态。<br>锁定键写序列：<br>WR LCKR[16] = '1' + LCKR[15:0]<br>WR LCKR[16] = '0' + LCKR[15:0]<br>WR LCKR[16] = '1' + LCKR[15:0]<br>RD LCKR |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | RD LCKR[16] = ‘1’（此读操作为可选操作，但它可确认锁定已激活）<br>注：在锁定键写序列期间，不能更改 LCK[15:0] 的值。<br>锁定序列中的任何错误都将中止锁定操作。<br>在任一端口位上的第一个锁定序列之后，对 LCKK 位的任何读访问都将返回‘1’，直 到下一次 MCU 复位或外设复位为止。 |
| 位 15:0 | LCKy | 端口 x 锁定位 y (y= 0..15)<br>这些位都是读/写位，但只能在 LCKK 位等于“0”时执行写操作。<br>0：端口配置未锁定<br>1：端口配置已锁定 |

### 7.4.9. GPIO 复用功能低位寄存器 (GPIOx_AFRL) (x = A..I)

偏移地址： 0x20

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AFRL7[3:0] | | | | AFRL6[3:0] | | | | AFRL5[3:0] | | | | AFRL4[3:0] | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| AFRL3[3:0] | | | | AFRL2[3:0] | | | | AFRL1[3:0] | | | | AFRL0[3:0] | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | AFRLy[3:0] | 端口 x 引脚 y 的复用功能选择 (y = 0..7)<br>这些位通过软件写入，用于配置复用功能 I/O。<br>AFRLy 选择：<br>0000: AF0　　1000: AF8<br>0001: AF1　　1001: AF9<br>0010: AF2　　1010: AF10<br>0011: AF3　　1011: AF11<br>0100: AF4　　1100: AF12<br>0101: AF5　　1101: AF13<br>0110: AF6　　1110: AF14<br>0111: AF7　　1111: AF15 |

### 7.4.10. GPIO 复用功能高位寄存器 (GPIOx_AFRH) (x = A..I)

偏移地址： 0x24

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AFRH15[3:0] | | | | AFRH14[3:0] | | | | AFRH13[3:0] | | | | AFRH12[3:0] | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| AFRH11[3:0] | | | | AFRH10[3:0] | | | | AFRH9[3:0] | | | | AFRH8[3:0] | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | AFRHy[3:0] | 端口 x 引脚 y 的复用功能选择 (y = 8..15)<br>这些位通过软件写入，用于配置复用功能 I/O。<br>AFRHy 选择:<br>0000: AF0　　1000: AF8<br>0001: AF1　　1001: AF9<br>0010: AF2　　1010: AF10<br>0011: AF3　　1011: AF11<br>0100: AF4　　1100: AF12<br>0101: AF5　　1101: AF13<br>0110: AF6　　1110: AF14<br>0111: AF7　　1111: AF15 |

### 7.4.11. GPIO 驱动能力配置寄存器 (GPIOx_DSR) (x = A..I)

偏移地址： 0xF0

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DS15 | | DS14 | | DS13 | | DS12 | | DS11 | | DS10 | | DS9 | | DS8 | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DS7 | | DS6 | | DS5 | | DS4 | | DS3 | | DS2 | | DS1 | | DS0 | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | DSy[1:0] | 端口 x 引脚 y 的驱动能力选择 (y = 0..15)<br>这些位通过软件写入，用于配置 I/O 的驱动能力。<br>00：8mA　　10：16mA<br>01：12mA　　11：20mA |

# 8. 系统配置控制器(SYSCFG)

系统配置控制器主要用于管理对可执行代码的存储区域的地址重映射、选择以太网 PHY 接口以及管理 GPIO 的外部中断线连接。

## 8.1. SYSCFG 寄存器

### 8.1.1. SYSCFG 外设模式配置寄存器(SYSCFG_PMC)

地址偏移： 0x04

复位值： 0x00000003

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| MII_RMII_SEL | Reserved | | | | | | |
| rw | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | SDIO_DMA_MODE |
| | | | | | | | rw |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:24 | 保留 | 必须保持复位值 |
| 位 23 | MII_RMII_SEL | 以太网 PHY 接口选择<br>由软件置 1 和清零。这些位控制以太网 MAC 的 PHY 接口。<br>0：选择 MII 接口<br>1：选择 RMII Why 接口<br>*注意：必须在 MAC 处于复位状态且在使能 MAC 时钟之前完成此配置。* |
| 位 23:9 | 保留 | 必须保持复位值 |
| 位 8 | SDIO_DMA_MODE | SDIO DMA 搬运模式<br>0：按 4-Word 搬运<br>1：按 1-Word 搬运 |
| 位 7:0 | 保留 | 必须保持复位值 |

### 8.1.2. SYSCFG 外部中断配置寄存器 1(SYSCFG_EXTICR1)

地址偏移： 0x08

复位值： 0x00000000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:16 | 保留 | 必须保持复位值 |
| 位 15:12 | EXTI3[3:0] | EXTI3 配置<br>这些位通过软件写入，以选择 EXTI3 外部中断的源输入。<br>0000：PA[3] 引脚<br>0001：PB[3] 引脚<br>0010：PC[3] 引脚<br>0011：PD[3] 引脚<br>0100：PE[3] 引脚<br>0101：PF[3] 引脚<br>0110：PG[3] 引脚<br>0111：PH[3] 引脚<br>1000：PI[3] 引脚 |
| 位 11:8 | EXTI2[3:0] | EXTI2 配置<br>这些位通过软件写入，以选择 EXTI2 外部中断的源输入。<br>0000：PA[2] 引脚<br>0001：PB[2] 引脚<br>0010：PC[2] 引脚<br>0011：PD[2] 引脚<br>0100：PE[2] 引脚<br>0101：PF[2] 引脚<br>0110：PG[2] 引脚<br>0111：PH[2] 引脚<br>1000：PI[2] 引脚 |
| 位 7:4 | EXTI1[3:0] | EXTI1 配置<br>这些位通过软件写入，以选择 EXTI1 外部中断的源输入。<br>0000：PA[1] 引脚<br>0001：PB[1] 引脚<br>0010：PC[1] 引脚<br>0011：PD[1] 引脚<br>0100：PE[1] 引脚<br>0101：PF[1] 引脚<br>0110：PG[1] 引脚<br>0111：PH[1] 引脚<br>1000：PI[1] 引脚 |
| 位 3:0 | EXTI0[3:0] | EXTI0 配置<br>这些位通过软件写入，以选择 EXTI0 外部中断的源输入。<br>0000：PA[0] 引脚<br>0001：PB[0] 引脚<br>0010：PC[0] 引脚<br>0011：PD[0] 引脚 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 0100： PE[0] 引脚<br>0101： PF[0] 引脚<br>0110： PG[0] 引脚<br>0111： PH[0] 引脚<br>1000： PI[0] 引脚 |

### 8.1.3. SYSCFG 外部中断配置寄存器 2(SYSCFG_EXTICR2)

地址偏移： 0x0C

复位值： 0x00000000

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

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:16 | 保留 | 必须保持复位值 |
| 位 15:12 | EXTI7[3:0] | EXTI7 配置<br>这些位通过软件写入，以选择 EXTI7 外部中断的源输入。<br>0000： PA[7] 引脚<br>0001： PB[7] 引脚<br>0010： PC[7] 引脚<br>0011： PD[7] 引脚<br>0100： PE[7] 引脚<br>0101： PF[7] 引脚<br>0110： PG[7] 引脚<br>0111： PH[7] 引脚<br>1000： PI[7] 引脚 |
| 位 11:8 | EXTI6[3:0] | EXTI6 配置<br>这些位通过软件写入，以选择 EXTI6 外部中断的源输入。<br>0000： PA[6] 引脚<br>0001： PB[6] 引脚<br>0010： PC[6] 引脚<br>0011： PD[6] 引脚<br>0100： PE[6] 引脚<br>0101： PF[6] 引脚<br>0110： PG[6] 引脚<br>0111： PH[6] 引脚<br>1000： PI[6] 引脚 |
| 位 7:4 | EXTI5[3:0] | EXTI5 配置<br>这些位通过软件写入，以选择 EXTI5 外部中断的源输入。<br>0000： PA[5] 引脚<br>0001： PB[5] 引脚<br>0010： PC[5] 引脚 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 0011： PD[5] 引脚<br>0100： PE[5] 引脚<br>0101： PF[5] 引脚<br>0110： PG[5] 引脚<br>0111： PH[5] 引脚<br>1000： PI[5] 引脚 |
| 位 3:0 | EXTI4[3:0] | EXTI4 配置<br>这些位通过软件写入，以选择 EXTI4 外部中断的源输入。<br>0000： PA[4] 引脚<br>0001： PB[4] 引脚<br>0010： PC[4] 引脚<br>0011： PD[4] 引脚<br>0100： PE[4] 引脚<br>0101： PF[4] 引脚<br>0110： PG[4] 引脚<br>0111： PH[4] 引脚<br>1000： PI[4] 引脚 |

### 8.1.4. SYSCFG 外部中断配置寄存器 3(SYSCFG_EXTICR3)

地址偏移： 0x10

复位值： 0x00000000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EXTI11[3:0] | | | | EXTI10[3:0] | | | | EXTI9[3:0] | | | | EXTI8[3:0] | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:16 | 保留 | 必须保持复位值 |
| 位 15:12 | EXTI11[3:0] | EXTI11 配置<br>这些位通过软件写入，以选择 EXTI11 外部中断的源输入。<br>0000： PA[11] 引脚<br>0001： PB[11] 引脚<br>0010： PC[11] 引脚<br>0011： PD[11] 引脚<br>0100： PE[11] 引脚<br>0101： PF[11] 引脚<br>0110： PG[11] 引脚<br>0111： PH[11] 引脚<br>1000： PI[11] 引脚 |
| 位 11:8 | EXTI10[3:0] | EXTI10 配置<br>这些位通过软件写入，以选择 EXTI10 外部中断的源输入。<br>0000： PA[10] 引脚<br>0001： PB[10] 引脚 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 0010： PC[10] 引脚<br>0011： PD[10] 引脚<br>0100： PE[10] 引脚<br>0101： PF[10] 引脚<br>0110： PG[10] 引脚<br>0111： PH[10] 引脚<br>1000： PI[10] 引脚 |
| 位 7:4 | EXTI9[3:0] | EXTI9 配置<br>这些位通过软件写入，以选择 EXTI9 外部中断的源输入。<br>0000： PA[9] 引脚<br>0001： PB[9] 引脚<br>0010： PC[9] 引脚<br>0011： PD[9] 引脚<br>0100： PE[9] 引脚<br>0101： PF[9] 引脚<br>0110： PG[9] 引脚<br>0111： PH[9] 引脚<br>1000： PI[9] 引脚 |
| 位 3:0 | EXTI8[3:0] | EXTI8 配置<br>这些位通过软件写入，以选择 EXTI8 外部中断的源输入。<br>0000： PA[8] 引脚<br>0001： PB[8] 引脚<br>0010： PC[8] 引脚<br>0011： PD[8] 引脚<br>0100： PE[8] 引脚<br>0101： PF[8] 引脚<br>0110： PG[8] 引脚<br>0111： PH[8] 引脚<br>1000： PI[8] 引脚 |

### 8.1.5. SYSCFG 外部中断配置寄存器 4(SYSCFG_EXTICR4)

地址偏移： 0x14

复位值： 0x00000000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EXTI15[3:0] | | | | EXTI14[3:0] | | | | EXTI13[3:0] | | | | EXTI12[3:0] | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:16 | 保留 | 必须保持复位值 |
| 位 15:12 | EXTI15[3:0] | EXTI15 配置<br>这些位通过软件写入，以选择 EXTI15 外部中断的源输入。<br>0000： PA[15] 引脚 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 0001： PB[15] 引脚<br>0010： PC[15] 引脚<br>0011： PD[15] 引脚<br>0100： PE[15] 引脚<br>0101： PF[15] 引脚<br>0110： PG[15] 引脚<br>0111： PH[15] 引脚<br>1000： PI[15] 引脚 |
| 位 11:8 | EXTI14[3:0] | EXTI14 配置<br>这些位通过软件写入，以选择 EXTI14 外部中断的源输入。<br>0000： PA[14] 引脚<br>0001： PB[14] 引脚<br>0010： PC[14] 引脚<br>0011： PD[14] 引脚<br>0100： PE[14] 引脚<br>0101： PF[14] 引脚<br>0110： PG[14] 引脚<br>0111： PH[14] 引脚<br>1000： PI[14] 引脚 |
| 位 7:4 | EXTI13[3:0] | EXTI13 配置<br>这些位通过软件写入，以选择 EXTI13 外部中断的源输入。<br>0000： PA[13] 引脚<br>0001： PB[13] 引脚<br>0010： PC[13] 引脚<br>0011： PD[13] 引脚<br>0100： PE[13] 引脚<br>0101： PF[13] 引脚<br>0110： PG[13] 引脚<br>0111： PH[13] 引脚<br>1000： PI[13] 引脚 |
| 位 3:0 | EXTI12[3:0] | EXTI12 配置<br>这些位通过软件写入，以选择 EXTI12 外部中断的源输入。<br>0000： PA[12] 引脚<br>0001： PB[12] 引脚<br>0010： PC[12] 引脚<br>0011： PD[12] 引脚<br>0100： PE[12] 引脚<br>0101： PF[12] 引脚<br>0110： PG[12] 引脚<br>0111： PH[12] 引脚<br>1000： PI[12] 引脚 |

### 8.1.6. SYSCFG 外设模式配置寄存器 1(SYSCFG_PMC1)

地址偏移： 0x30

复位值： 0x00000001

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:24 | 保留 | 必须保持复位值 |
| 位 23:20 | SDIO_CLKD_SEL | SDIO_CLK 驱动时钟相位 延迟时间 = 配置值 * 2(ns) |
| 位 19:16 | SDIO_CLKS_SEL | SDIO_CLK 采样时钟相位 延迟时间 = 配置值 * 2(ns) |
| 位 15:0 | 保留 | 必须保持复位值 |

# 9. 直接存储器访问控制器(DMA)

## 9.1. DMA 简介

直接存储器访问(DMA)用于在外设与存储器之间以及存储器与存储器之间提供高速数据传输。可以在无需任何 CPU 操作的情况下通过 DMA 快速移动数据。这样节省的 CPU 资源可以供其他操作使用。

DMA 控制器基于复杂的总线矩阵架构，将功能强大的双 AHB 主总线架构与独立的 FIFO 结构合在一起，优化了系统带宽。

两个 DMA 控制器总共有 16 个数据流(每个控制器 8 个)，每一个 DMA 控制器都用于管理一个或多个外设存储器访问请求。每个数据流总共可以有多达 8 个通道(或称请求)。每个通道都有一个仲裁器，用于处理 DMA 请求间的优先级。

## 9.2. DMA 主要特性

DMA 主要特性：

- 双 AHB 主总线架构，一个用于存储器访问，另一个用于外设访问。
- 仅支持 32 位访问的 AHB 从编程接口。
- 每个 DMA 控制器有 8 个数据流，每个数据流有多达 8 个通道(或称请求)。
- Stream6 和 Stream7 内嵌 128 字节 FIFO，其余 6 个 Stream 内嵌 16 字节 FIFO，可用于 FIFO 模式或直接模式：
  - FIFO 模式：可通过软件阈值级别选取为 FIFO 的 1/4、1/2、3/4 或满。
  - 直接模式。

    每个 DMA 请求会立即启动对存储器的传输。当在直接模式(禁止 FIFO)下将 DMA 请求配置为存储器到外设模式传输数据时，DMA 仅会将一个数据从存储器预加载到内部 FIFO，从而确保一旦外设触发 DMA 请求时则立即传输数据。
- 通过硬件可以将两个数据流配置为：
  - 支持外设到存储器、存储器到外设和存储器到存储器的常规通道。
  - 在存储器支持双缓冲区通道。
- 8 个数据流的每一个都连接到专用硬件 DMA 通道(请求)。
- DMA 数据流请求之间的优先级可用软件编程(4 个级别：非常高、高、中、低)，在软件优先级相同的情况下可以通过硬件决定优先级(例如，请求 0 的优先级高于请求 1)。
- 每个数据流也支持存储器到存储器的软件触发(仅限 DMA2 控制器)。
- 可供每个数据流选择的通道请求多达 8 个。此选择可由软件配置，允许几个外设发起 DMA 请求
- 要传输的数据项的数目可以由 DMA 控制器或外设管理：
  - DMA 要传输的数据项的数目可用软件编程，从 1 至 4294967295。
  - 外设流控制器：要传输的数据项的数目未知并由目标外设控制，这些外设通过硬件发出传输结束的信号。
- 独立的源和目标传输宽度（字节、半字、字）：源和目标的数据宽度不相等时，DMA 自动封装/解封必要的传输数据来优化带宽。这个特性仅在 FIFO 模式下可用。
- 对源和目标的增量或非增量寻址。

- 支持 4 个、8 个、16 个、32 个、64 个和 128 个节拍的增量突发传输。突发增量的大小可由软件配置，通常等于外设 FIFO 大小的一半。
- 每个数据流都支持循环缓冲区管理。
- 6 个事件标志（DMA 半传输、DMA 传输完成、DMA 传输错误、DMA FIFO 错误、直接模式错误、DMA 链模式子块传输完成），进行逻辑或运算，从而产生每个数据流的单个中断请求。
- 每个数据流都支持链模式传输。

## 9.3. DMA 功能说明

### 9.3.1. 一般说明

下图显示了 DMA 的框图。

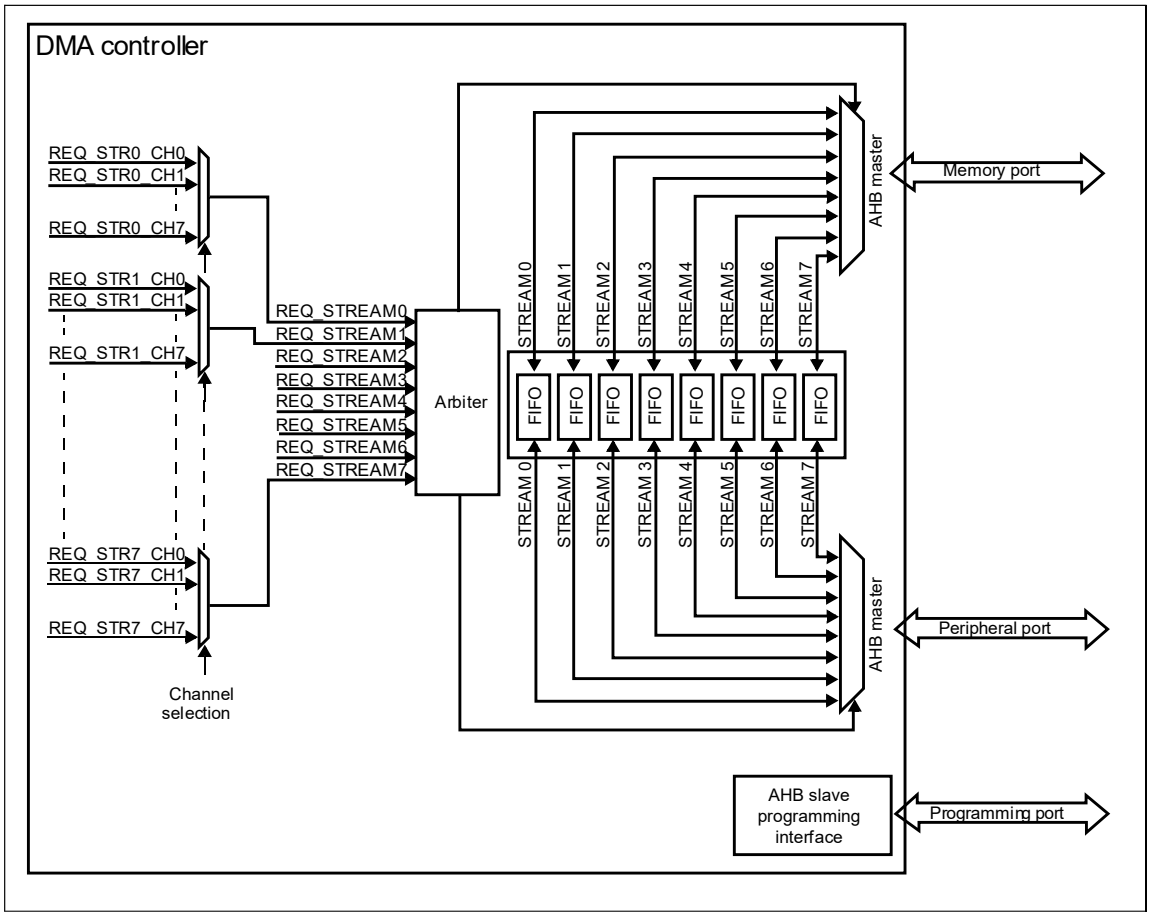



图 9-1 DMA 框图

DMA 控制器执行直接存储器传输：因为采用 AHB 主总线，它可以控制 AHB 总线矩阵来启动 AHB 事务。

它可以执行下列事务：

- 外设到存储器的传输；
- 存储器到外设的传输；
- 存储器到存储器的传输。

DMA 控制器提供两个 AHB 主端口：AHB 存储器端口（用于连接存储器）和 AHB 外设端口（用于连接

外设)。但是要执行存储器到存储器的传输，AHB 外设端口必须也能访问存储器。

AHB 从端口用于对 DMA 控制器进行编程（它仅支持 32 位访问)。

### 9.3.2. DMA 事务

DMA 事务由给定数目的数据传输序列组成。要传输的数据项的数目及其宽度（8 位、16 位 或 32 位）可用软件编程。

每个 DMA 传输包含三项操作:

- 通过 DMA_SxPAR 或 DMA_SxM0AR 寄存器寻址，从外设数据寄存器或存储器单元中加载数据。
- 通过 DMA_SxPAR 或 DMA_SxM0AR 寄存器寻址，将加载的数据存储到外设数据寄存器或存储器单元。
- DMA_SxNDTR 计数器在数据存储结束后递减，该计数器中包含仍需执行的事务数。

在产生事件后，外设会向 DMA 控制器发送请求信号。DMA 控制器根据通道优先级处理该请求。只要 DMA 控制器访问外设，DMA 控制器就会向外设发送确认信号。外设获得 DMA 控制器的确认信号后，便会立即释放其请求。一旦外设使请求失效，DMA 控制器就会释放确认信号。如果有更多请求，外设可以启动下一个事务。

### 9.3.3. 通道选择

每个数据流都与一个 DMA 请求相关联，此 DMA 请求可以从 8 个可能的通道请求中选出。 此选择由 DMA_SxCR 寄存器中的 CHSEL[2:0] 位控制。

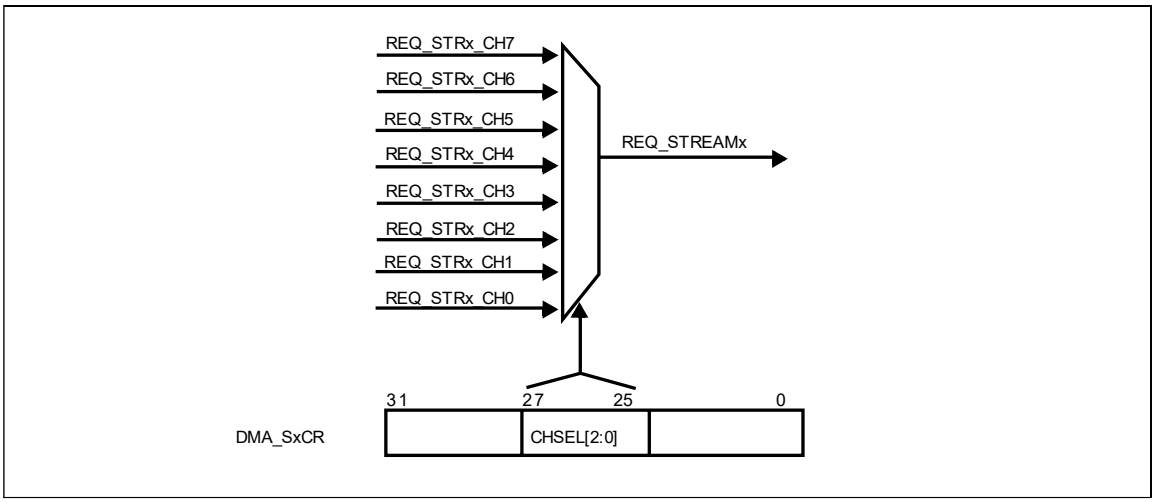



图 9-2 通道选择

下表给出了 DMA 请求映射的示例。

表 9-1 DMA1 请求映射

| 外设请求 | 数据流 0 | 数据流 1 | 数据流 2 | 数据流 3 | 数据流 4 | 数据流 5 | 数据流 6 | 数据流 7 |
|---|---|---|---|---|---|---|---|---|
| 通道 0 | SPI3_RX | - | SPI3_RX | SPI2_RX | SPI2_TX | SPI3_TX | - | SPI3_TX |
| 通道 1 | I2C1_RX | - | TIM7_UP | - | TIM7_UP | I2C1_RX | I2C1_TX | I2C1_TX |
| 通道 2 | TIM4_CH1 | - | I2S3_EXT_RX | TIM4_CH2 | I2S2_EXT_TX | I2S3_EXT_TX | TIM4_UP | TIM4_CH3 |
| 通道 3 | I2S3_EXT_RX | TIM2_UP TIM2_CH3 | I2C3_RX | I2S2_EXT_RX | I2C3_TX | TIM2_CH1 | TIM2_CH2 TIM2_CH4 | TIM2_CH4 TIM2_UP |
| 通道 4 | | USART3_RX | UART4_RX | USART3_TX | UART4_TX | USART2_RX | USART2_TX | |

| 外设请求 | 数据流 0 | 数据流 1 | 数据流 2 | 数据流 3 | 数据流 4 | 数据流 5 | 数据流 6 | 数据流 7 |
|---|---|---|---|---|---|---|---|---|
| 通道 5 | | UART7_TX | TIM3_CH4<br>TIM3_UP | UART7_RX | TIM3_TRIG<br>TIM3_CH1 | TIM3_CH2 | | TIM3_CH3 |
| 通道 6 | TIM5_CH3<br>TIM5_UP | TIM5_CH4<br>TIM5_TRIG | TIM5_CH1 | TIM5_TRIG<br>TIM5_CH4 | TIM5_CH2 | - | TIM5_UP | - |
| 通道 7 | - | TIM6_UP | I2C2_RX | I2C2_RX | USART3_TX | DAC1 | DAC2 | I2C2_TX |

表 9-2 DMA2 请求映射

| 外设请求 | 数据流 0 | 数据流 1 | 数据流 2 | 数据流 3 | 数据流 4 | 数据流 5 | 数据流 6 | 数据流 7 |
|---|---|---|---|---|---|---|---|---|
| 通道 0 | ADC1 | Quad-SPI<br>_RX | TIM8_CH1<br>TIM8_CH2<br>TIM8_CH3 | Quad-SPI<br>_TX | ADC1 | JPEG_IN | TIM1_CH1<br>TIM1_CH2<br>TIM1_CH3 | - |
| 通道 1 | - | | ADC2 | ADC2 | - | | | |
| 通道 2 | ADC3 | ADC3 | | SPI5_RX | SPI5_TX | - | XFC_TX | XFC_TX |
| 通道 3 | SPI1_RX | XFC_RX | SPI1_RX | SPI1_TX | XFC_TX | SPI1_TX | XFC_RX | XFC_RX |
| 通道 4 | SPI4_RX | SPI4_TX | USART1_RX | SDIO | - | USART1_RX | SDIO | USART1_TX |
| 通道 5 | - | USART6_RX | USART6_RX | SPI4_RX | SPI4_TX | - | USART6_TX | USART6_TX |
| 通道 6 | TIM1_TRIG | TIM1_CH1 | TIM1_CH2 | TIM1_CH1 | TIM1_CH4<br>TIM1_TRIG<br>TIM1_COM | TIM1_UP | TIM1_CH3 | - |
| 通道 7 | JPEG_OUT | TIM8_UP | TIM8_CH1 | TIM8_CH2 | TIM8_CH3 | SPI5_RX | SPI5_TX | TIM8_CH4<br>TIM8_TRIG<br>TIM8_COM |

### 9.3.4. 仲裁器

仲裁器为两个 AHB 主端口（存储器和外设端口）提供基于请求优先级的 8 个 DMA 数据流请求管理，并启动外设/存储器访问序列。

优先级管理分为两个阶段：

- 软件：每个数据流优先级都可以在 DMA_SxCR 寄存器中配置。分为四个级别：
  - 非常高优先级；
  - 高优先级；
  - 中优先级；
  - 低优先级。
- 硬件：如果两个请求具有相同的软件优先级，则编号低的数据流优先于编号高的数据流。例如，数据流 2 的优先级高于数据流 4。

### 9.3.5. DMA 数据流

8 个 DMA 控制器数据流都能够提供源和目标之间的单向传输链路。

每个数据流配置后都可以执行：

- 常规类型事务：存储器到外设、外设到存储器或存储器到存储器的传输。
- 双缓冲区类型事务：使用存储器的两个存储器指针的双缓冲区传输（当 DMA 正在进行自/至缓冲区的读/写操作时，应用程序可以进行至/自其它缓冲区的写/读操作）。

要传输的数据量（多达 4294967295）可以编程，并与连接到外设 AHB 端口的外设（请求 DMA 传输）

的源宽度相关。每个事务完成后，包含要传输的数据项总量的寄存器都会递减。

### 9.3.6. 源、目标和传输模式

源传输和目标传输在整个 4 GB 区域（地址在 0x0000 0000 和 0xFFFF FFFF 之间）都可以寻址外设和存储器。

传输方向使用 DMA_SxCR 寄存器中的 DIR[1:0] 位进行配置，有三种可能的传输方向：存储器到外设、外设到存储器或存储器到存储器。

表 9-3 源和目的地址

| DMA_SxCR 寄存器的位 DIR[1:0] | 方向 | 源地址 | 目标地址 |
|---|---|---|---|
| 00 | 外设到存储器 | DMA_SxPAR | DMA_SxM0AR |
| 01 | 存储器到外设 | DMA_SxM0AR | DMA_SxPAR |
| 10 | 存储器到存储器 | DMA_SxPAR | DMA_SxM0AR |
| 11 | 保留 | - | - |

当数据宽度(在 DMA_SxCR 寄存器的 PSIZE 或 MSIZE 位中编程)分别是半字或字时，写入 DMA_SxPAR 或 DMA_SxM0AR/M1AR 寄存器的外设或存储器地址必须分别在字或半字地址的边界对齐。

#### 外设到存储器模式

- 使能这种模式（将 DMA_SxCR 寄存器中的位 EN 置 1）时，每次产生外设请求，数据流都会启动数据源到 FIFO 的传输。
- 达到 FIFO 的阈值级别时，FIFO 的内容移出并存储到目标中。
- 如果 DMA_SxNDTR 寄存器达到零、外设请求传输终止（在使用外设流控制器的情况下）或 DMA_SxCR 寄存器中的 EN 位由软件清零，传输即会停止。
- 在直接模式下（当 DMA_SxFCR 寄存器中的 DMDIS 值为“0”时），不使用 FIFO 的阈值级别控制：每完成一次从外设到 FIFO 的数据传输后，相应的数据立即就会移出并存储到目标中。
- 只有赢得了数据流的仲裁后，相应数据流才有权访问 AHB 源或目标端口。系统使用在 DMA_SxCR 寄存器 PL[1:0] 位中为每个数据流定义的优先级执行仲裁。

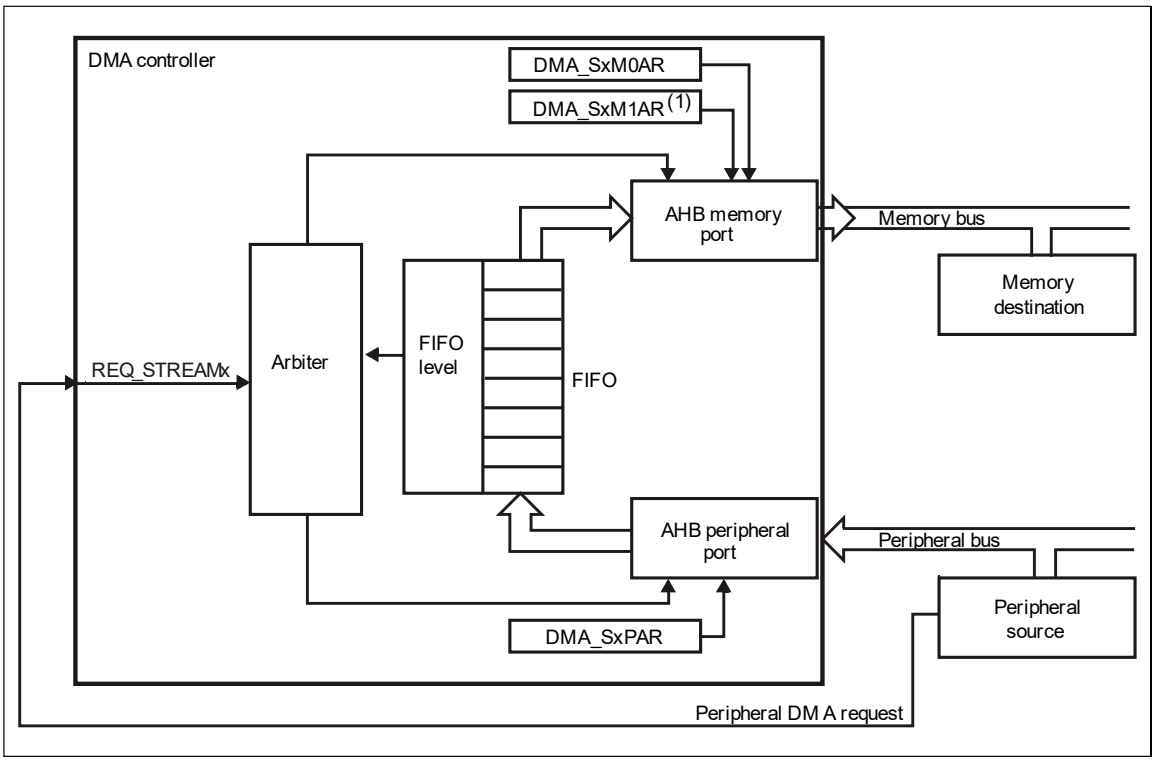



图 9-3 外设到存储器模式

- 用于双缓冲区模式。

### 存储器到外设模式

- 使能这种模式（将 DMA_SxCR 寄存器中的 EN 位置 1）时，数据流会立即启动传输，从源完全填充 FIFO。
- 每次发生外设请求，FIFO 的内容都会移出并存储到目标中。当 FIFO 的级别小于或等于预定义的阈值级别时，将使用存储器中的数据完全重载 FIFO。
- 如果 DMA_SxNDTR 寄存器达到零、外设请求传输终止（在使用外设流控制器的情况下）或 DMA_SxCR 寄存器中的 EN 位由软件清零，传输即会停止。
- 在直接模式下（当 DMA_SxFCR 寄存器中的 DMDIS 值为“0”时），不使用 FIFO 的阈值级别。一旦使能了数据流，DMA 便会预装载第一个数据，将其传输到内部 FIFO。一旦外设请求数据传输，DMA 便会将预装载的值传输到配置的目标。然后，它会使用要传输的下一个数据再次重载内部空 FIFO。预装载的数据大小为 DMA_SxCR 寄存器中 PSIZE 位字段的值。
- 只有赢得了数据流的仲裁后，相应数据流才有权访问 AHB 源或目标端口。系统使用在 DMA_SxCR 寄存器 PL[1:0] 位中为每个数据流定义的优先级执行仲裁。

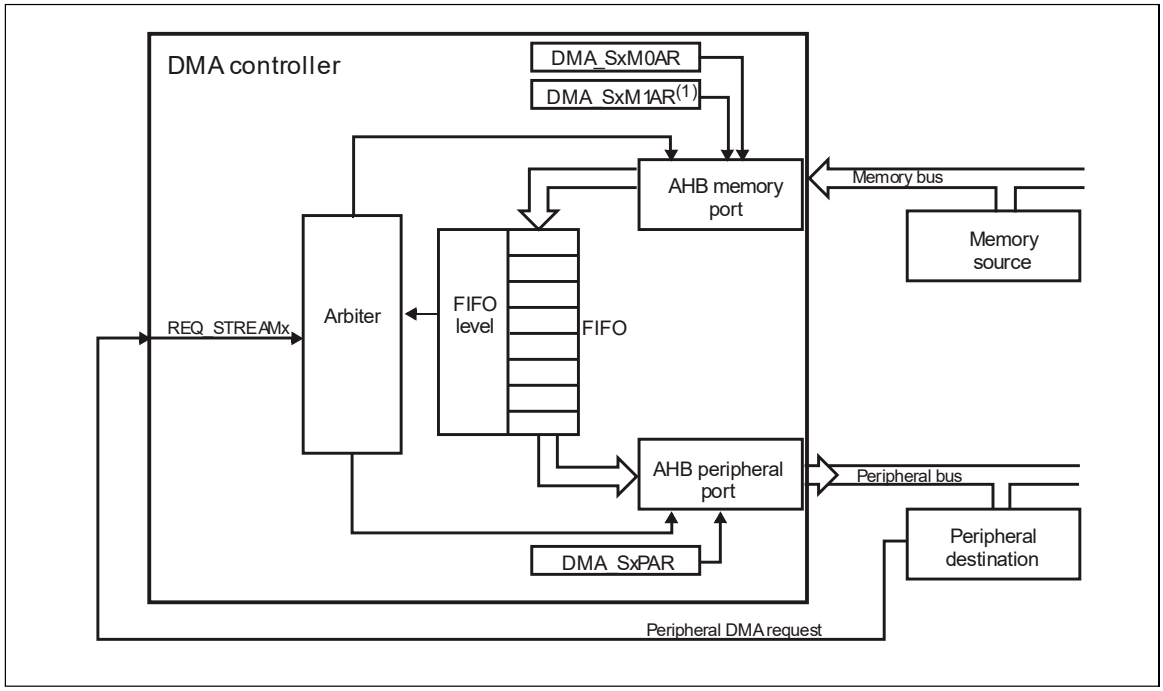



图 9-4 存储器到外设模式

- 用于双缓冲区模式

### 存储器到存储器模式

- DMA 通道在没有外设请求触发的情况下同样可以工作。此为图 9-5 中介绍的存储器到存储器模式。
- 通过将 DMA_SxCR 寄存器中的使能位 (EN) 置 1 来使能数据流时，数据流会立即开始填充 FIFO，直至达到阈值级别。达到阈值级别后，FIFO 的内容便会移出，并存储到目标中。
- 如果 DMA_SxNDTR 寄存器达到零或 DMA_SxCR 寄存器中的 EN 位由软件清零，传输即会停止。
- 只有赢得了数据流的仲裁后，相应数据流才有权访问 AHB 源或目标端口。系统使用 DMA_SxCR 寄存器 PL[1:0] 位中为每个数据流定义的优先级执行仲裁。

*注意：使用存储器到存储器模式时，不允许循环模式和直接模式。*

*只有 DMA2 控制器能够执行存储器到存储器的传输。*

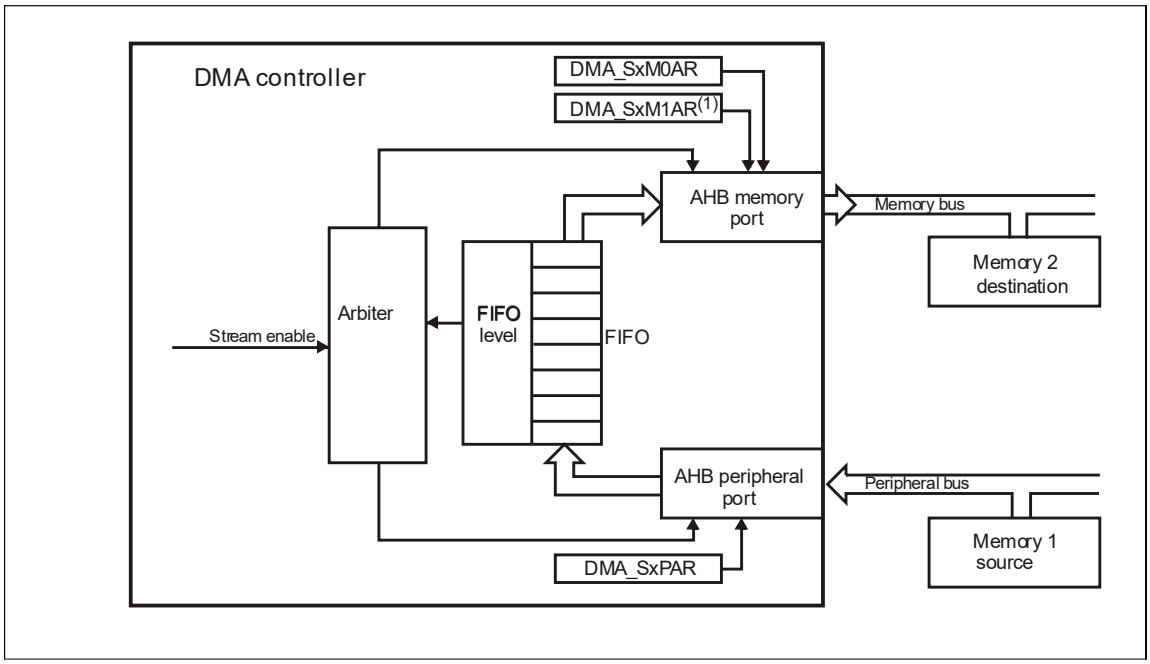



图 9-5 存储器到存储器模式

- 用于双缓冲区模式。

### 9.3.7. 指针递增

根据 DMA_SxCR 寄存器中 PINC 和 MINC 位的状态，外设和存储器指针在每次传输后可以自动向后递增或保持常量。

通过单个寄存器访问外设源或目标数据时，禁止递增模式十分有用。

如果使能了递增模式，则根据在 DMA_SxCR 寄存器 PSIZE 或 MSIZE 位中编程的数据宽度，下一次传输的地址将是前一次传输的地址递增 1（对于字节）、2（对于半字）或 4（对于字）。

为了优化封装操作，可以不管 AHB 外设端口上传输的数据的大小，将外设地址的增量偏移大小固定下来。DMA_SxCR 寄存器中的 PINCOS 位用于将增量偏移大小与外设 AHB 端口或 32 位地址（此时地址递增 4）上的数据大小对齐。PINCOS 位仅对 AHB 外设端口有影响。

如果将 PINCOS 位置 1，则不论 PSIZE 值是多少，下一次传输的地址总是前一次传输的地址递增 4（自动与 32 位地址对齐）。但是，AHB 存储器端口不受此操作影响。

如果 AHB 外设端口或 AHB 存储器端口分别请求突发事务，为了满足 AMBA 协议（在固定地址模式下不允许突发事务），则需要将 PINC 或 MINC 位置 1。

### 9.3.8. 循环模式

循环模式可用于处理循环缓冲区和连续数据流（例如 ADC 扫描模式）。可以使用 DMA_SxCR 寄存器中的 CIRC 位使能此特性。

当激活循环模式时，要传输的数据项的数目在数据流配置阶段自动用设置的初始值进行加载，并继续响应 DMA 请求。

*注意：在循环模式下，如果为存储器配置了突发模式，必须遵循下列规则：*

*DMA_SxNDTR = ((Mburst 节拍 ) × (Msize)/(Psize)) 的倍数，其中：*

- *(Mburst 节拍 ) = 4、8、16、32、64、128（取决于 DMA_SxCR 寄存器中的 MBURST 位）*

- *((Msize)/(Psize)) = 1、2、4、1/2 或 1/4（Msize 和 Psize 表示DMA_SxCR 寄存器中的 MSIZE 和 PSIZE 位。它们与字节相关）*
- *DMA_SxNDTR = AHB 外设端口上要传输的数据项的数目*

*例如：Mburst 节拍 = 8 (INCR8)，MSIZE =“00”（字节）和 PSIZE =“01”（半字），在此例中：DMA_SxNDTR 必须是 (8 × 1/2 = 4) 的倍数。*

*如果不遵循此公式，则 DMA 行为和数据完整性得不到保证。*

*NDTR 还必须是外设突发大小与外设数据大小乘积的倍数，否则会导致错误的DMA 行为。*

### 9.3.9. 双缓冲区模式

此模式可用于所有 DMA1 和 DMA2 数据流。

通过将 DMA_SxCR 寄存器中的 DBM 位置 1，即可使能双缓冲区模式。

除了有两个存储器指针之外，双缓冲区数据流的工作方式与常规（单缓冲区）数据流的一样。使能双缓冲区模式时，将自动使能循环模式（DMA_SxCR 中的 CIRC 位的状态是“无关”），并在每次事务结束时交换存储器指针。

在此模式下，每次事务结束时，DMA 控制器都从一个存储器目标交换为另一个存储器目标。这样，软件在处理一个存储器区域的同时，DMA 传输还可以填充/使用第二个存储器区域。如下表：双缓冲区模式下的源和目标地址寄存器 (DBM=1) 所述，双缓冲区数据流可以双向工作（存储器既可以是源也可以是目标）。

*注意：在双缓冲区模式下使能数据流时，可遵循下列条件，实时更新 AHB 存储器的基址（DMA_SxM0AR 或 DMA_SxM1AR）：*

- *当 DMA_SxCR 寄存器中的 CT 位为“0”时，可以写入 DMA_SxM1AR 寄存器。当 CT =“1”时，试图写入此寄存器会将错误标志位 (TEIF) 置 1，并自动禁止数据流。*
- *当 DMA_SxCR 寄存器中的 CT 位为“1”时，可以写入 DMA_SxM0AR 寄存器。当 CT =“0”时，试图写入此寄存器会将错误标志位 (TEIF) 置 1，并自动禁止数据流。*

*为避免出现任何错误状态，建议在 TCIF 标志位置位时立即更改基址。因为此时根据上述两个条件之一，目标存储器依据 DMA_SxCR 寄存器中 CT 值的情况，一定已从存储器 0 更改为存储器 1（或从存储器 1 更改为存储器 0）。*

*对于所有其它模式（双缓冲区模式除外），一旦使能数据流，存储器地址寄存器即被写保护。*

表 9-4 双缓冲区模式下的源和目标地址寄存器(DBM=1)

| DMA_SxCR 寄存器的位 DIR[1:0] | 方向 | 源地址 | 目标地址 |
|---|---|---|---|
| 00 | 外设到存储器 | DMA_SxPAR | DMA_SxM0AR/DMA_SxM1AR |
| 01 | 存储器到外设 | DMA_SxM0AR/DMA_SxM1AR | DMA_SxPAR |
| 10 | 不允许(1) | | |
| 11 | 保留 | - | - |

- *使能双缓冲区模式时，自动使能循环模式。由于存储器到存储器模式与循环模式不兼容，所以当使能双缓冲区模式时，不允许配置存储器到存储器模式。*

### 9.3.10. 可编程数据宽度、封装/解封、字节序

要传输的数据项数目必须在使能数据流之前编程到 DMA_SxNDTR（要传输数据项数目位，NDT）中，当流控制器是外设且 DMA_SxCR 中的 PFCTRL 位置为 1 时除外。

当使用内部 FIFO 时，源和目标数据的数据宽度可以通过 DMA_SxCR 寄存器的 PSIZE 和 MSIZE 位（可以是 8、16 或 32 位）编程。

当 PSIZE 和 MSIZE 不相等时：

- 在 DMA_SxNDTR 寄存器中配置的要传输的数据项数目的数据宽度等于外设总线的宽度（由 DMA_SxCR 寄存器中的 PSIZE 位配置）。例如，在外设到存储器、存储器到外设或存储器到存储器传

输的情况下，如果将 PSIZE[1:0] 位配置为半字，则要传输的字节数等于 2 × NDT。

- DMA 控制器仅按小字节序寻址源和目标。相关内容在下表：封装/解封和字节序行为（位 PINC = MINC = 1）中介绍。

在封装/解封数据的过程中，如果在数据完全封装/解封前中断操作，则有数据损坏的危险。因此，为了确保数据一致性，可将数据流配置成生成突发传输：在这种情况下，属于一个突发的每组传输不可分割（请参见：单次传输和突发传输）。

在直接模式下（DMA_SxFCR 寄存器中的 DMDIS = 0），不能进行数据封装/解封。这种情况下，不允许源与目标的传输数据宽度不同，二者必须相等，并由 DMA_SxCR 中的 PSIZE 位定义，MSIZE 位的状态是“无关”。

表 9-5 封装/解封和字节序行为(位 PINC = MINC = 1)

| AHB 存储器端口宽度 | AHB 外设端口宽度 | 要传输的数据项的数目 (NDT) | 存储器传输数目 | 存储器端口地址/字节通道 | 外设传输数目 | 外设端口地址/字节通道 | |
|---|---|---|---|---|---|---|---|
| | | | | | | PINCOS = 1 | PINCOS = 0 |
| 8 | 8 | 4 | 1234 | 0x0/B0[7:0]<br>0x1/B1[7:0]<br>0x2/B2[7:0]<br>0x3/B3[7:0] | 1<br>2<br>3<br>4 | 0x0/B0[7:0]<br>0x4/B1[7:0]<br>0x8/B2[7:0]<br>0xC/B3[7:0] | 0x0/B0[7:0]<br>0x1/B1[7:0]<br>0x2/B2[7:0]<br>0x3/B3[7:0] |
| 8 | 16 | 2 | 1234 | 0x0/B0[7:0]<br>0x1/B1[7:0]<br>0x2/B2[7:0]<br>0x3/B3[7:0] | 1<br>2 | 0x0/B1|B0[15:0]<br>0x4/B3|B2[15:0] | 0x0/B1|B0[15:0]<br>0x2/B3|B2[15:0] |
| 8 | 32 | 1 | 1234 | 0x0/B0[7:0]<br>0x1/B1[7:0]<br>0x2/B2[7:0]<br>0x3/B3[7:0] | 1 | 0x0/B3|B2|B1|B0[31:0] | 0x0/B3|B2|B1|B0[31:0] |
| 16 | 8 | 4 | 1 2 | 0x0/B1|B0[15:0]<br>0x2/B3|B2[15:0] | 1<br>2<br>3<br>4 | 0x0/B0[7:0]<br>0x4/B1[7:0]<br>0x8/B2[7:0]<br>0xC/B3[7:0] | 0x0/B0[7:0]<br>0x1/B1[7:0]<br>0x2/B2[7:0]<br>0x3/B3[7:0] |
| 16 | 16 | 2 | 1 2 | 0x0/B1|B0[15:0]<br>0x2/B1|B0[15:0] | 1<br>2 | 0x0/B1|B0[15:0]<br>0x4/B3|B2[15:0] | 0x0/B1|B0[15:0]<br>0x2/B3|B2[15:0] |
| 16 | 32 | 1 | 1<br>2 | 0x0/B1|B0[15:0]<br>0x2/B3|B2[15:0] | 1 | 0x0/B3|B2|B1|B0[31:0] | 0x0/B3|B2|B1|B0[31:0] |
| 32 | 8 | 4 | 1 | 0x0/B3|B2|B1|B0[31:0] | 1<br>2<br>3<br>4 | 0x0/B0[7:0]<br>0x4/B1[7:0]<br>0x8/B2[7:0]<br>0xC/B3[7:0] | 0x0/B0[7:0]<br>0x1/B1[7:0]<br>0x2/B2[7:0]<br>0x3/B3[7:0] |
| 32 | 16 | 2 | 1 | 0x0/B3|B2|B1|B0[31:0] | 1<br>2 | 0x0/B1|B0[15:0]<br>0x4/B3|B2[15:0] | 0x0/B1|B0[15:0]<br>0x2/B3|B2[15:0] |
| 32 | 32 | 1 | 1 | 0x0/B3|B2|B1|B0[31:0] | 1 | 0x0/B3|B2|B1|B0[31:0] | 0x0/B3|B2|B1|B0[31:0] |

*注意：外设端口可以是源或目标（在存储器到存储器传输的情况下，也可能是存储器源）。必须配置 PSIZE、MSIZE 和 NDT[31:0]，以确保最后一次传输的完整性。当外设端口的数据宽度（ PSIZE 位）小于存储器端口的数据宽度（ MSIZE 位）时，可能会发生数据传输不完整的情况。此限制条件汇总在下表中。*

表 9-6 PSIZE 与 MSIZE 确定时对 NDT 的限制条件

| DMA_SxCR 的 PSIZE[1:0] | DMA_SxCR 的 MSIZE[1:0] | DMA_SxNDTR 的 NDT[31:0] |
|---|---|---|
| 00（8 位） | 01（16 位） | 必须是 2 的倍数 |
| 00（8 位） | 10（32 位） | 必须是 4 的倍数 |
| 01（16 位） | 10（32 位） | 必须是 2 的倍数 |

### 9.3.11. 单次传输和突发传输

DMA 控制器可以产生单次传输或 4 个、8 个、16 个、32 个、64 个和 128 个节拍的增量突发传输。

突发大小通过软件针对两个 AHB 端口独立配置，配置时使用 DMA_SxCR 寄存器中的 MBURST[2:0] 和 PBURST[2:0] 位。

突发大小指示突发中的节拍数，而不是传输的字节数。

为确保数据一致性，形成突发的每一组传输都不可分割：在突发传输序列期间，AHB 传输会锁定，并且 AHB 总线矩阵的仲裁器不解除对 DMA 主总线的授权。

根据单次或突发配置的情况，每个 DMA 请求在 AHB 外设端口上相应地启动不同数量的传输。

- 当 AHB 外设端口被配置为单次传输时，根据 DMA_SxCR 寄存器 PSIZE[1:0] 位的值，每个 DMA 请求产生一次字节、半字或字的数据传输。
- 当 AHB 外设端口被配置为突发传输时，根据 DMA_SxCR 寄存器 PBURST[2:0] 和 PSIZE[1:0] 位的值，每个 DMA 请求相应地生成 4 个、8 个、16 个、32 个、64 个和 128 个节拍的字节、半字或字的传输。

对于需要配置 MBURST 和 MSIZE 位的 AHB 存储器端口，必须考虑与上述相同的内容。

在直接模式下，数据流只能生成单次传输，而 MBURST[2:0] 和 PBURST[2:0] 位由硬件强制配置。

必须选择地址指针(DMA_SxPAR 或 DMA_SxM0AR 寄存器)，以确保一个突发块内的所有传输在等于传输大小的地址边界对齐。

选择突发配置必须要遵守 AHB 协议，即突发传输不得越过 1 KB 地址边界，因为可以分配给单个从设备的最小地址空间是 1 KB。这意味着突发块传输不应越过 1 KB 地址边界，否则就 会产生一个 AHB 错误，并且 DMA 寄存器不会报告这个错误。

*注意：仅在使能指针递增模式时允许突发模式：*

- *当 PINC 位为"0"时，也应将 PBURST 位清为"00"*
- *当 MINC 位为"0"时，也应将 MBURST 位清为"00"*

### 9.3.12. FIFO

#### FIFO 结构

FIFO 用于在源数据传输到目标之前临时存储这些数据。

其中数据流 0~5 有一个独立的 4 字 FIFO，数据流 6~7 有一个独立的 32 字 FIFO，阈值级别均可由软件配置为 1/4、1/2、3/4 或满。

为了使能 FIFO 阈值级别，必须通过将 DMA_SxFCR 寄存器中的 DMDIS 位置 1 来禁止直接模式。

FIFO 的结构随源与目标数据宽度而不同，相关内容在下图：FIFO 结构中介绍。

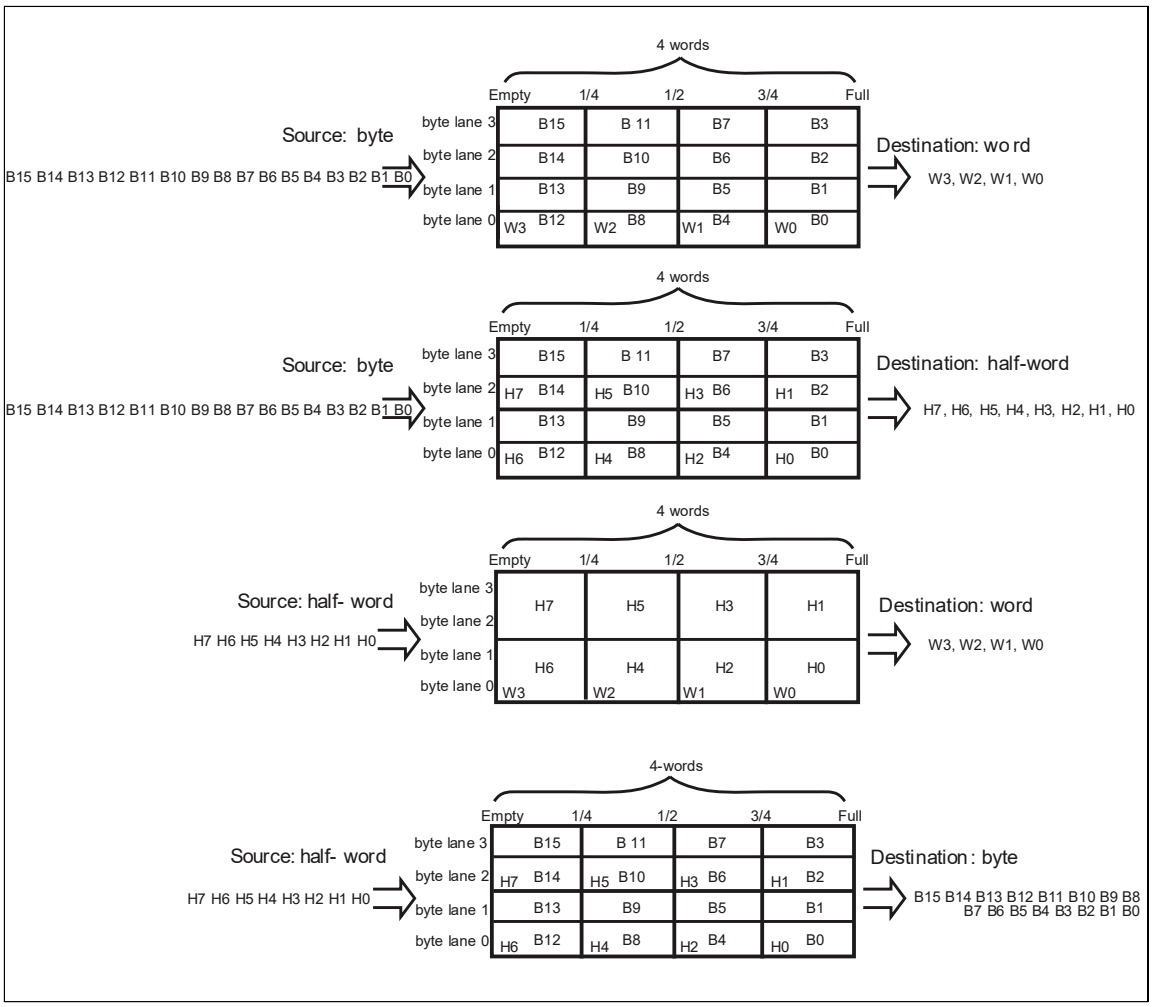



图 9-6 FIFO 结构

## FIFO 阈值与突发配置

存在 16 字节和 128 字节两种 FIFO 规格，stream0-stream5 为 16 字节，stream-stream7 是 128 字节。

对于 16 字节 FIFO 而言，burst 长度最大配置为 incr16，此时 size 需配置为 byte。

对于 128 字节 FIFO 而言，burst 长度最大配置为 incr128，此时 size 需配置为 byte。

由于 DMA 传输机制和 FIFO 阈值相关，因此 FIFO 阈值的设定需要和 burst 和 size 的配置匹配，才能保证正确传输。一次传输的数据量为 burst*size，因此需要保证 FIFO 阈值对应的数据量大于等于 burst*size，且（FIFO 总容量-FIFO 阈值）大于等于 burst*size（FIFO 未满的情况下）。具体配置要求参加如下表格：

表 9-7 Stream0-Stream5 FIFO 阈值配置要求

| pisze/msize | FIFO 阈值 | pburst/mbusrt =single | pburst/mbusrt =inc4 | pburst/mbusrt =inc8 | pburst/mbusrt =inc16 |
|---|---|---|---|---|---|
| 8bit | 1/4 | 允许 | 允许 | 禁止 | 禁止 |
| | 1/2 | 允许 | 允许 | 允许 | 禁止 |
| | 3/4 | 允许 | 允许 | 禁止 | 禁止 |
| | 满 | 允许 | 允许 | 允许 | 允许 |
| 16bit | 1/4 | 允许 | 禁止 | 禁止 | 禁止 |
| | 1/2 | 允许 | 允许 | 禁止 | 禁止 |

| pisze/msize | FIFO 阈值 | pburst/mbusrt =single | pburst/mbusrt =inc4 | pburst/mbusrt =inc8 | pburst/mbusrt =inc16 |
|---|---|---|---|---|---|
| | 3/4 | 允许 | 禁止 | 禁止 | 禁止 |
| | 满 | 允许 | 允许 | 允许 | 禁止 |
| 32bit | 1/4 | 允许 | 禁止 | 禁止 | 禁止 |
| | 1/2 | 允许 | 禁止 | 禁止 | 禁止 |
| | 3/4 | 允许 | 禁止 | 禁止 | 禁止 |
| | 满 | 允许 | 允许 | 禁止 | 禁止 |

表 9-8 Stream6-Stream7 FIFO 阈值配置要求

| pisze/ msize | FIFO 阈值 | pburst/ mbusrt= single | pburst/ mbusrt =inc4 | pburst/ mbusrt =inc8 | pburst/ mbusrt =inc16 | pburst/ mbusrt =inc32 | pburst/ mbusrt =inc64 | pburst/ mbusrt =inc128 |
|---|---|---|---|---|---|---|---|---|
| 8bit | 1/4 | 允许 | 允许 | 允许 | 允许 | 允许 | 禁止 | 禁止 |
| | 1/2 | 允许 | 允许 | 允许 | 允许 | 允许 | 允许 | 禁止 |
| | 3/4 | 允许 | 允许 | 允许 | 允许 | 允许 | 禁止 | 禁止 |
| | 满 | 允许 | 允许 | 允许 | 允许 | 允许 | 允许 | 允许 |
| 16bit | 1/4 | 允许 | 允许 | 允许 | 允许 | 禁止 | 禁止 | 禁止 |
| | 1/2 | 允许 | 允许 | 允许 | 允许 | 允许 | 禁止 | 禁止 |
| | 3/4 | 允许 | 允许 | 允许 | 允许 | 禁止 | 禁止 | 禁止 |
| | 满 | 允许 | 允许 | 允许 | 允许 | 允许 | 允许 | 禁止 |
| 32bit | 1/4 | 允许 | 允许 | 允许 | 禁止 | 禁止 | 禁止 | 禁止 |
| | 1/2 | 允许 | 允许 | 允许 | 允许 | 禁止 | 禁止 | 禁止 |
| | 3/4 | 允许 | 允许 | 允许 | 禁止 | 禁止 | 禁止 | 禁止 |
| | 满 | 允许 | 允许 | 允许 | 允许 | 允许 | 禁止 | 禁止 |

### FIFO 刷新

当复位 DMA_SxCR 寄存器中的 EN 位来禁止数据流，以及配置数据流来管理外设到存储器或存储器到存储器的传输时，可以刷新 FIFO。因为如果禁止数据流时仍有某些数据存留在 FIFO 中，DMA 控制器会将剩余的数据继续传输到目标（即使已经有效禁止了数据流）。刷新完成时，会将 DMA_LISR 或 DMA_HISR 寄存器中的传输完成状态位 (TCIFx) 置 1。

在这种情况下，剩余数据计数器 DMA_SxNDTR 保持的值指示在目标存储器现有多少可用数据项。

请注意在 FIFO 刷新操作期间，如果 FIFO 中要传输到存储器的剩余数据项的数目（以字节为单位）小于存储器数据宽度（例如在 MSIZE 配置为字时 FIFO 中为 2 个字节），则会使用在 DMA_SxCR 寄存器中的 MSIZE 位设置的数据宽度发送数据。这意味着将使用非预期值写入存储器。软件可以读取 DMA_SxNDTR 寄存器来确定包含良好数据的存储器区域（起始地址和最后地址）。

如果 FIFO 中剩余数据项的数目小于突发大小，并且数据流通过将 DMA_SxCR 寄存器 MBURST 位置 1 进行配置来管理在 AHB 存储器端口上的突发传输，则使用单次传输来完成 FIFO 的刷新。

### 直接模式

默认情况下，FIFO 以直接模式操作（将 DMA_SxFCR 中的 DMDIS 位置 1），不使用 FIFO 阈值级别。如果在每次 DMA 请求之后，系统需要至/自存储器的立即和单独传输，这种模式非常有用。

当在直接模式（禁止 FIFO）下将 DMA 配置为以存储器到外设模式传输数据时，DMA 会将一个数据从存储器预加载到内部 FIFO，从而确保一旦外设触发 DMA 请求时则立即传输数据。

为了避免 FIFO 饱和，建议使用高优先级配置相应的数据流。

该模式仅限以下方式的传输：

- 源和目标传输宽度相等，并均由 DMA_SxCR 中的 PSIZE[1:0] 位定义（MSIZE[1:0] 位的状态是“无关”）。
- 不可能进行突发传输（DMA_SxCR 中的 PBURST[2:0] 和 MBURST[2:0] 位的状态是“无关”）。

当实现存储器到存储器传输时不得使用直接模式。

### 9.3.13. DMA 传输完成

以下各种事件均可以结束传输过程，并将 DMA_LISR 或 DMA_HISR 状态寄存器中的 TCIFx 位置 1：

- 在 DMA 流控制器模式下：
  - 在存储器到外设模式下，DMA_SxNDTR 计数器已达到零。
  - 传输结束前禁止了数据流（通过将 DMA_SxCR 寄存器中的 EN 位清零），并在传输是外设到存储器或存储器到存储器的模式时，所有的剩余数据均已从 FIFO 刷新到存储器。
- 在外设流控制器模式下：
  - 已从外设生成最后的外部突发请求或单独请求，并当 DMA 在外设到存储器模式下工作时，余数据已从 FIFO 传输到存储器。
  - 数据流由软件禁止，并当 DMA 在外设到存储器模式下工作时，剩余数据已从 FIFO 传输到存储器。

*注意：仅在外设到存储器模式下，传输的完成取决于 FIFO 中要传输到存储器的剩余数据。这种情况不适用于存储器到外设模式。*

*如果是在非循环模式下配置数据流，传输结束后（即要传输的数据数目达到零），除非软件重新对数据流编程并重新使能数据流（通过将 DMA_SxCR 寄存器中的 EN 位置 1），否则 DMA 即会停止传输（通过硬件将 DMA_SxCR 寄存器中的 EN 位清零）并且不再响应任何 DMA 请求。*

### 9.3.14. DMA 传输暂停

可以随时暂停 DMA 传输以供稍后重新开始；也可以在 DMA 传输结束前明确禁止暂停功能。分为两种情况：

- 数据流禁止传输，以后不从停止点重新开始暂停。这种情况下，只需将 DMA_SxCR 寄存器中的 EN 位清零来禁止数据流，除此之外不需要任何其他操作。禁止数据流可能要花费一些时间（需要首先完成正在进行的传输）。需要将传输完成中断标志（DMA_LISR 或 DMA_HISR 寄存器中的 TCIF）置 1 来指示传输结束。现在 DMA_SxCR 中的 EN 位的值是“0”，借此确认数据流已经终止传输。DMA_SxNDTR 寄存器包含数据流停止时剩余数据项的数目，这样软件便可以确定数据流中断前已传输了多少数据项。
- 数据流在 DMA_SxNDTR 寄存器中要传输的剩余数据项数目达到 0 之前暂停传输。目的 是以后通过重新使能数据流重新开始传输。为了在传输停止点重新开始传输，软件必须在通过写入 DMA_SxCR 寄存器中的 EN 位（然后检查确认该位为‘0’）禁止数据流之后，首先读取 DMA_SxNDTR 寄存器来了解已经收集的数据项的数目。然后：
  - 必须更新外设和/或存储器地址以调整地址指针。
  - 必须使用要传输的剩余数据项的数目（禁止数据流时读取的值）更新 SxNDTR 寄存器。
  - 然后可以重新使能数据流，从停止点重新开始传输。

*注意：传输完成中断标志（DMA_LISR 或 DMA_HISR 中的 TCIF）置 1 将指示因数据流中断而结束传输。*

### 9.3.15. 流控制器

控制要传输的数据数目的实体称为流控制器。此流控制器使用 DMA_SxCR 寄存器中的 PFCTRL 位针对每个数据流独立配置。

流控制器可以是:

- DMA 控制器：在这种情况下，要传输的数据项的数目在使能 DMA 数据流之前由软件编程到 DMA_SxNDTR 寄存器。
- 外设源或目标：当要传输的数据项的数目未知时属于这种情况。当所传输的是最后的数据时，外设通过硬件向 DMA 控制器发出指示。仅限能够发出传输结束信号的外设支持此功能，也就是:
  - SDIO

当外设流控制器用于给定数据流时，写入 DMA_SxNDTR 的值对 DMA 传输没有作用。实际上，不论写入什么值，一旦使能数据流，硬件即会将该值强制置为 0xFFFFFFFF 来执行下列方案:

- 预期的数据流中断：DMA_SxCR 寄存器中的 EN 位由软件重置为 0，以在外设发送最后的数据硬件信号（单独或突发）之前停止数据流。这样，在外设到存储器 DMA 传输的情况下，数据流即会关闭并触发 FIFO 刷新。状态寄存器中相应数据流的 TCIFx 标志置 1 以指示 DMA 完成传输。要了解 DMA 传输期间传输的数据项的数目，请读取 DMA_SxNDTR 寄存器并应用下列公式:
  - 传输的数据数目 = 0xFFFFFFFF - DMA_SxNDTR
- 因接收到最后的数据硬件信号而引起的正常数据流中断：当外设请求最后的传输（单独或突发）并当此传输完成时，自动中断数据流。相应流的 TCIFx 标志在状态寄存器中置 1 以指示 DMA 传输完成。要了解传输的数据项的数目，请读取 DMA_SxNDTR 寄存器并应用与上面相同的公式。
- DMA_SxNDTR 寄存器达到 0: 状态寄存器中相应数据流的 TCIFx 标志置 1 以指示强制的 DMA 传输完成。即使尚未置位最后的数据硬件信号（单独或突发），也会自动关闭数据流。已传输的数据不会丢失。这意味着即使在外设流控制模式下，DMA 在单独的事务中最多处理 4294967295 个数据项。

*注意：当在存储器到存储器模式下配置时，DMA 始终是流控制器，而 PFCTRL 位由硬件强制置为 0。在外设流控制器模式下禁止循环模式。*

### 9.3.16. 可能的 DMA 配置汇总

下表汇总了各种可能的 DMA 配置。

表 9-9 可能的 DMA 配置

| DMA 传输模式 | 源 | 目标 | 流控制器 | 循环模式 | 传输类型 | 直接模式 | 双缓冲区模式 |
|---|---|---|---|---|---|---|---|
| 外设到存储器 | AHB 外设端口 | AHB 存储器端口 | DMA | 允许 | 单独 | 允许 | 允许 |
| | | | | | 突发 | 禁止 | |
| | | | 外设 | 禁止 | 单独 | 允许 | 禁止 |
| | | | | | 突发 | 禁止 | |
| 存储器到外设 | AHB 存储器端口 | AHB 外设端口 | DMA | 允许 | 单独 | 允许 | 允许 |
| | | | | | 突发 | 禁止 | |
| | | | 外设 | 禁止 | 单独 | 允许 | 禁止 |
| | | | | | 突发 | 禁止 | |
| 存储器到存储器 | AHB 外设端口 | AHB 存储器端口 | 仅 DMA | 禁止 | 单独 | 禁止 | 禁止 |
| | | | | | 突发 | | |

### 9.3.17. 流配置过程

配置 DMA 数据流 x（其中 x 是数据流编号）时应遵守下面的顺序:

- 如果使能了数据流，通过重置 DMA_SxCR 寄存器中的 EN 位将其禁止，然后读取此位以确认没有正在进行的数据流操作。将此位写为 0 不会立即生效，因为实际上只有所有当前传输都已完成时才会将其写为 0。当所读取 EN 位的值为 0 时，才表示可以配置数据流。因此在开始任何数据流配置之前，需要等待 EN 位置 0。应将先前的数据块 DMA 传输中在状态寄存器（DMA_LISR 和 DMA_HISR）中置 1 的所有数据流专用的位置 0，然后才可重新使能数据。
- 在 DMA_SxPAR 寄存器中设置外设端口寄存器地址。外设事件发生后，数据会从此地址移动到外设端口或从外设端口移动到此地址。
- 在 DMA_SxMA0R 寄存器（在双缓冲区模式的情况下还有 DMA_SxMA1R 寄存器）中设置存储器地址。外设事件发生后，将从此存储器读取数据或将数据写入此存储器。
- 在 DMA_SxNDTR 寄存器中配置要传输的数据项的总数。每出现一次外设事件或每出现一个节拍的突发传输，该值都会递减。
- 使用 DMA_SxCR 寄存器中的 CHSEL[2:0] 选择 DMA 通道（请求）。
- 如果外设用作流控制器而且支持此功能，请将 DMA_SxCR 寄存器中的 PFCTRL 位置 1。
- 使用 DMA_SxCR 寄存器中的 PL[1:0] 位配置数据流优先级。
- 配置 FIFO 的使用情况（使能或禁止，发送和接收阈值）。
- 配置数据传输方向、外设和存储器增量/固定模式、单独或突发事务、外设和存储器数据宽度、循环模式、双缓冲区模式和传输完成一半和/或全部完成，和/或 DMA_SxCR 寄存器中错误的中断。
- 通过将 DMA_SxCR 寄存器中的 EN 位置 1 激活数据流。

一旦使能了流，即可响应连接到数据流的外设发出的任何 DMA 请求。

一旦在 AHB 目标端口上传输了一半数据，传输一半标志 (HTIF) 便会置 1，如果传输一半中断使能位 (HTIE) 置 1, 还会生成中断。传输结束时, 传输完成标志 (TCIF) 便会置 1, 如果传输完成中断使能位 (TCIE) 置 1，还会生成中断。

### 9.3.18. 错误管理

DMA 控制器可以检测到以下错误：

- 传输错误：当发生下列情况时，传输错误中断标志 (TEIFx) 将置 1：
  - DMA 读或写访问期间发生总线错误。
  - 软件请求在双缓冲区模式下写访问存储器地址寄存器，但是已使能数据流，并且当前目标存储器是受写入存储器地址寄存器操作影响的存储器。
- FIFO 错误：如果发生下列情况，FIFO 错误中断标志 (FEIFx) 将置 1：
  - 检测到 FIFO 下溢情况。
  - 检测到 FIFO 上溢情况（在存储器到存储器模式下由 DMA 内部管理请求和传输，所以在此模式下不检测溢出情况）。
  - 当 FIFO 阈值级别与存储器突发大小不兼容时使能流。
- 直接模式错误：只有当在直接模式下工作并且已将 DMA_SxCR 寄存器中的 MINC 位清零时，才能在外设到存储器模式下将直接模式错误中断标志 (DMEIFx) 置 1。当在先前数据未完全传输到存储器（因为存储器总线未得到授权）的情况下发生 DMA 请求时，该标志将置 1。在这种情况下，该标志指示有两个数据项相继传输到相同的目标地址，如果目标不能管理这种情况，会发生问题。

在直接模式下，如果出现下列条件，FIFO 错误标志也会置 1：

- 在外设到存储器模式下，如果未对存储器总线授权支持多个外设请求，FIFO 可能饱和（上溢）。
- 在存储器到外设模式下，如果在外设请求发生前存储器总线未得到授权，则可能发生下溢的情况。

如果由于突发大小与 FIFO 阈值级别之间的不兼容而引起 TEIFx 或 FEIFx 标志置 1，则硬件自动将相应

数据流配置寄存器 (DMA_SxCR) 中的 EN 位清零，从而禁止错误的数据流。

如果由于上溢或下溢情况引起 DMEIFx 或 FEIFx 标志置 1，则不会自动禁止错误的数据流， 而是由软件决定是否通过重置 DMA_SxCR 寄存器中的 EN 位禁止数据流。这是因为当发生这种错误时没有数据丢失。

当 DMA_LISR 或 DMA_HISR 寄存器中数据流的错误中断标志（TEIF、FEIF、DMEIF）置 1 时，如果 DMA_SxCR 或 DMA_SxFCR 寄存器中相应的中断使能位（TEIE、FEIE、DMIE）也置 1，则会生成一个中断。

*注意：当 FIFO 上溢或下溢的情况发生时，因为直到上溢或下溢的情况被清除，数据流才会确认外设请求，所以数据不会丢失。如果此确认过程花费过多时间，则外设本身会检测到其内部缓冲器上溢或下溢的情况，数据可能丢失。*

### 9.3.19. 链模式

链模式可实现多块的连续搬运，节约多次 DMA 的配置时间，每次链传输可更新外设地址寄存器、存储地址寄存器、传输长度寄存器、下次链传输信息部分起始地址寄存器。

链地址寄存器 DMA_S0LINK-DMA_S7LINK 中的值需要保证按字对齐。

**操作流程**

以 stream0 通道为例：

- 配置 DMA_S0CR,DMAS0FCR,配置通道基本信息，使能 link_en，地址寄存器和数量寄存器不需配置；
- 配置 DMA_S0LINK 寄存器，填入首块的信息起始地址 link_addr，link_addr 需保证按字对齐；
- 启动 stream0 使能；
- DMA 自动从(link_addr+0x0)加载外设地址到寄存器 DMA_S0PAR；
- (link_addr+0x4)加载存储地址到寄存器 DMA_S0M0AR；
- (link_addr+0x8)加载数据长度到寄存器 DMA_S0NDTR；
- (link_addr+0xc)加载下一次链传输的信息起始地址到寄存器 DMA_S0LINK；
- DMA 开启第一个链传输，传输完成后产生链完成中断；
- 若加载的下一链的信息起始地址不为 0（即当前 DMA_S0LINK 的值），则继续下一链的传输；若为 0，则结束全部链传输，产生 DMA 传输完成中断。

### 9.3.20. DMA 中断

对于每个 DMA 数据流，可在发生以下事件时产生中断：

- 达到半传输；
- 传输完成；
- 传输错误；
- FIFO 错误（上溢、下溢或 FIFO 级别错误）；
- 直接模式错误；
- 链模式子块传输完成。

表 9-10 DMA 中断请求

| 中断事件 | 中断标志 | 使能控制位 |
|---|---|---|
| 半传输 | HTIF | HTIE |
| 传输完成 | TCIF | TCIE |
| 传输错误 | TEIF | TEIE |

| 中断事件 | 中断标志 | 使能控制位 |
|---|---|---|
| FIFO 上溢/下溢 | FEIF | FEIE |
| 直接模式错误 | DMEIF | DMEIE |
| 链模式子块传输完成 | LTCIF | LTCIE |

*注意： 在将使能控制位置1'前，应将相应的事件标志清零，否则会立即产生中断。*

## 9.4. DMA 寄存器

### 9.4.1. DMA 低中断状态寄存器(DMA_LISR)

偏移地址： 0x0000

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | TCIF3 | HTIF3 | TEIF3 | DMEIF3 | LTCIF3 | FEIF3 | TCIF2 | HTIF2 | TEIF2 | DMEIF2 | LTCIF2 | FEIF2 |
| | | | | r | r | r | r | r | r | r | r | r | r | r | r |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | TCIF1 | HTIF1 | TEIF1 | DMEIF1 | LTCIF1 | FEIF1 | TCIF0 | HTIF0 | TEIF0 | DMEIF0 | LTCIF0 | FEIF0 |
| | | | | r | r | r | r | r | r | r | r | r | r | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:28 | 保留 | 必须保持复位值 |
| 位 15:12 | 保留 | 必须保持复位值 |
| 位 27、21、11、5 | TCIFx | 数据流 x 传输完成中断标志(x = 3..0)<br>此位将由硬件置 1, 由软件清零, 软件只需将 1 写入 DMA_LIFCR 寄存器的相应位。<br>0：数据流 x 上无传输完成事件<br>1：数据流 x 上发生传输完成事件 |
| 位 26、20、10、4 | HTIFx | 数据流 x 半传输中断标志(x=3..0)<br>此位将由硬件置 1, 由软件清零, 软件只需将 1 写入 DMA_LIFCR 寄存器的相应位。<br>0：数据流 x 上无半传输事件<br>1：数据流 x 上发生半传输事件 |
| 位 25、19、9、3 | TEIFx | 数据流 x 传输错误中断标志(x=3..0)<br>此位将由硬件置 1, 由软件清零, 软件只需将 1 写入 DMA_LIFCR 寄存器的相应位。<br>0：数据流 x 上无传输错误<br>1：数据流 x 上发生传输错误 |
| 位 24、18、8、2 | DMEIFx | 数据流 x 直接模式错误中断标志 (x=3..0)<br>此位将由硬件置 1, 由软件清零, 软件只需将 1 写入 DMA_LIFCR 寄存器的相应位。<br>0：数据流 x 上无直接模式错误<br>1：数据流 x 上发生直接模式错误 |
| 位 23、17、7、1 | LTCIFx | 数据流 x 链模式子块传输完成中断标志 (x = 3..0)<br>此位将由硬件置 1, 由软件清零, 软件只需将 1 写入 DMA_LIFCR 寄存器的相应位。<br>0：数据流 x 未出现链模式子块传输完成事件 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 1：数据流 x 链模式中的一个子块传输完成事件 |
| 位 22、16、6、0 | FEIFx | 数据流 x FIFO 错误中断标志(x=3..0)<br>此位将由硬件置 1, 由软件清零, 软件只需将 1 写入 DMA_LIFCR 寄存器的相应位。<br>0：数据流 x 上无 FIFO 错误事件<br>1：数据流 x 上发生 FIFO 错误事件 |

### 9.4.2. DMA 高中断状态寄存器(DMA_HISR)

偏移地址： 0x0004

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | TCIF3 | HTIF3 | TEIF3 | DMEIF3 | LTCIF3 | FEIF3 | TCIF2 | HTIF2 | TEIF2 | DMEIF2 | LTCIF2 | FEIF2 |
| | | | | r | r | r | r | r | r | r | r | r | r | r | r |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | TCIF1 | HTIF1 | TEIF1 | DMEIF1 | LTCIF1 | FEIF1 | TCIF0 | HTIF0 | TEIF0 | DMEIF0 | LTCIF0 | FEIF0 |
| | | | | r | r | r | r | r | r | r | r | r | r | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:28 | 保留 | 必须保持复位值 |
| 位 15:12 | 保留 | 必须保持复位值 |
| 位 27、21、11、5 | TCIFx | 数据流 x 传输完成中断标志 (x = 7..4)<br>此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_HIFCR 寄存器的相应位。<br>0：数据流 x 上无传输完成事件<br>1：数据流 x 上发生传输完成事件 |
| 位 26、20、10、4 | HTIFx | 数据流 x 半传输中断标志 (x = 7..4)<br>此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_HIFCR 寄存器的相应位。<br>0：数据流 x 上无半传输事件<br>1：数据流 x 上发生半传输事件 |
| 位 25、19、9、3 | TEIFx | 数据流 x 传输错误中断标志 (x = 7..4)<br>此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_HIFCR 寄存器的相应位。<br>0：数据流 x 上无传输错误<br>1：数据流 x 上发生传输错误 |
| 位 24、18、8、2 | DMEIFx | 数据流 x 直接模式错误中断标志 (x = 7..4)<br>此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_HIFCR 寄存器的相应位。<br>0：数据流 x 上无直接模式错误<br>1：数据流 x 上发生直接模式错误 |
| 位 23、17、7、1 | LTCIFx | 数据流 x 链模式子块传输完成中断标志 (x = 7..4)<br>此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_HIFCR 寄存器的相应位。<br>0：数据流 x 未出现链模式子块传输完成事件<br>1：数据流 x 链模式中的一个子块传输完成事件 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 22、16、6、0 | FEIFx | 数据流 x FIFO 错误中断标志 (x = 7..4)<br>此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_HIFCR 寄存器的相应位。<br>0：数据流 x 上无 FIFO 错误事件<br>1：数据流 x 上发生 FIFO 错误事件 |

### 9.4.3. DMA 低中断标志清除寄存器(DMA_LIFCR)

偏移地址： 0x0008

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Reserved | | | | CTCIF3 | CHTIF3 | CTEIF3 | CDMEIF3 | CLTCIF3 | CFEIF3 | CTCIF2 | CHTIF2 | CTEIF2 | CDMEIF2 | CLTCIF2 | CFEIF2 |
| | | | | r | r | r | r | r | r | r | r | r | r | r | r |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Reserved | | | | CTCIF1 | CHTIF1 | CTEIF1 | CDMEIF1 | CLTCIF1 | CFEIF1 | CTCIF0 | CHTIF0 | CTEIF0 | CDMEIF0 | CLTCIF0 | CFEIF0 |
| | | | | r | r | r | r | r | r | r | r | r | r | r | r |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:28 | 保留 | 必须保持复位值 |
| 位 15:12 | 保留 | 必须保持复位值 |
| 位 27、21、11、5 | CTCIFx | 数据流 x 传输完成中断标志清零 (x = 3..0)<br>将 1 写入此位时，DMA_LISR 寄存器中相应的 CTCIFx 标志将清零 |
| 位 26、20、10、4 | CHTIFx | 数据流 x 半传输中断标志清零 (x = 3..0)<br>将 1 写入此位时，DMA_LISR 寄存器中相应的 CHTIFx 标志将清零 |
| 位 25、19、9、3 | CTEIFx | 数据流 x 传输错误中断标志清零 (x = 3..0)<br>将 1 写入此位时，DMA_LISR 寄存器中相应的 CTEIFx 标志将清零 |
| 位 24、18、8、2 | CDMEIFx | 数据流 x 直接模式错误中断标志清零(x = 3..0)<br>将 1 写入此位时，DMA_LISR 寄存器中相应的 DMEIFx 标志将清零 |
| 位 23、17、7、1 | CLTCIFx | 数据流 x 链模式子块中断标志清零 (x = 3..0)<br>将 1 写入此位时，DMA_LISR 寄存器中相应的 CLTCIFx 标志将清零 |
| 位 22、16、6、0 | CFEIFx | 数据流 x FIFO 错误中断标志清零 (x = 3..0)<br>将 1 写入此位时，DMA_LISR 寄存器中相应的 CFEIFx 标志将清零 |

### 9.4.4. DMA 低中断标志清除寄存器(DMA_HIFCR)

偏移地址： 0x000C

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Reserved | | | | CTCIF3 | CHTIF3 | CTEIF3 | CDMEIF3 | CLTCIF3 | CFEIF3 | CTCIF2 | CHTIF2 | CTEIF2 | CDMEIF2 | CLTCIF2 | CFEIF2 |
| | | | | r | r | r | r | r | r | r | r | r | r | r | r |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Reserved | | | | CTCIF1 | CHTIF1 | CTEIF1 | CDMEIF1 | CLTCIF1 | CFEIF1 | CTCIF0 | CHTIF0 | CTEIF0 | CDMEIF0 | CLTCIF0 | CFEIF0 |
| | | | | r | r | r | r | r | r | r | r | r | r | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:28 | 保留 | 必须保持复位值 |
| 位 15:12 | 保留 | 必须保持复位值 |
| 位 27、21、11、5 | CTCIFx | 数据流 x 传输完成中断标志清零 (x = 7..4)<br>将 1 写入此位时, DMA_HISR 寄存器中相应的 CTCIFx 标志将清零 |
| 位 26、20、10、4 | CHTIFx | 数据流 x 半传输中断标志清零 (x = 7..4)<br>将 1 写入此位时, DMA_HISR 寄存器中相应的 CHTIFx 标志将清零 |
| 位 25、19、9、3 | CTEIFx | 数据流 x 传输错误中断标志清零(x = 7..4)<br>将 1 写入此位时, DMA_HISR 寄存器中相应的 CTEIFx 标志将清零 |
| 位 24、18、8、2 | CDMEIFx | 数据流 x 直接模式错误中断标志清零 (x = 7..4)<br>将 1 写入此位时, DMA_HISR 寄存器中相应的 CDMEIFx 标志将清零 |
| 位 23、17、7、1 | CLTCIFx | 数据流 x 链模式子块中断标志清零 (x = 7..4)<br>将 1 写入此位时, DMA_HISR 寄存器中相应的 CLTCIFx 标志将清零 |
| 位 22、16、6、0 | CFEIFx | 数据流 x FIFO 错误中断标志清零(x = 7..4)<br>将 1 写入此位时, DMA_HISR 寄存器中相应的 CFEIFx 标志将清零 |

### 9.4.5. DMA 数据流 x 配置寄存器(DMA_SxCR)(x = 0..7)

偏移地址： 0x10 + 0x18 * 数据流编号

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LINKEN | MBURST | | | CHSEL | | | LCIE | PBURST | | | Reserved | CT | DBM | PL | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PINCOS | MSIZE | | PSIZE | | MINC | PINC | CIRC | DIR | | PFCTRL | TCIE | HTIE | TEIE | DMEIE | EN |
| | | | | r | r | r | r | r | r | r | r | r | r | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31 | LINKEN | 链模式使能<br>此位将由软件置 1 和清零。<br>1：链模式使能<br>0：链模式关闭<br>此位受到保护，只有 EN 为“0”时才可以写入 |
| 位 30:28 | MBURST | 存储器突发传输配置<br>这些位将由软件置 1 和清零。<br>000：single 传输<br>001：incr4 传输<br>010：incr8 传输<br>011：incr16 传输<br>100：incr32 传输(仅 stream6,stream7 可用）<br>101：incr64 传输(仅 stream6,stream7 可用）<br>110：incr128 传输(仅 stream6,stream7 可用）<br>111：保留<br>这些位受到保护，只有 EN 为“0”时才可以写入 在直接模式中，当位 EN =“1”时，这些位由硬件强制置为 0x0。 |
| 位 27:25 | CHSEL | 通道选择 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 这些位将由软件置 1 和清零。<br>000：选择通道 0<br>001：选择通道 1<br>010：选择通道 2<br>011：选择通道 3<br>100：选择通道 4<br>101：选择通道 5<br>110：选择通道 6<br>111：选择通道 7<br>这些位受到保护，只有 EN 为“0”时才可以写入 |
| 位 24 | LTCIE | 传输完成中断使能<br>此位由软件置 1 和清零。<br>0：禁止 LTC 中断<br>1：使能 LTC 中断 |
| 位 23:21 | PBURST | 外设突发传输配置<br>这些位将由软件置 1 和清零。<br>000 : single 传输<br>001 : incr4 传输<br>010：incr8 传输<br>011：incr16 传输<br>100：incr32 传输(仅 stream6,stream7 可用）<br>101：incr64 传输(仅 stream6,stream7 可用）<br>110：incr128 传输(仅 stream6,stream7 可用）<br>111：保留<br>这些位受到保护，只有 EN 为“0”时才可以写入在直接模式中，当位 EN =“1”时，这些位由硬件强制置为 0x0。 |
| 位 20 | 保留 | 必须保持复位值 |
| 位 19 | CT | 当前目标（仅在双缓冲区模式下）<br>此位由硬件置 1 和清零，也可由软件写入。<br>0：当前目标存储器为存储器 0（使用 DMA_SxM0AR 指针寻址）<br>1：当前目标存储器为存储器 1（使用 DMA_SxM1AR 指针寻址）<br>只有 EN 为“0”时，此位才可以写入，以指示第一次传输的目标存储区。在使能数据流 后，此位相当于一个状态标志，用于指示作为当前目标的存储区。 |
| 位 18 | DBM | 双缓冲区模式<br>此位由软件置 1 和清零。<br>0：传输结束时不切换缓冲区<br>1：DMA 传输结束时切换目标存储区<br>此位受到保护，只有 EN 为“0”时才可以写入。 |
| 位 17：16 | PL | 优先级<br>这些位将由软件置 1 和清零。<br>00：低<br>01：中<br>10：高<br>11：非常高<br>这些位受到保护，只有 EN 为“0”时才可以写入。 |
| 位 15 | PINCOS | 外设增量偏移量<br>此位由软件置 1 和清零<br>0：用于计算外设地址的偏移量与 PSIZE 相关 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 1：用于计算外设地址的偏移量固定为 4（32 位对齐）<br>如果位 PINC =“0”，则此位没有意义。此位受到保护，只有 EN 为“0”时才可以写入。 如果选择直接模式或者 PBURST 不等于“00”，则当使能数据流（位 EN =“1”）时，此位由硬件强制置为低电平。 |
| 位 14:13 | MSIZE | 存储器数据大小<br>这些位将由软件置 1 和清零。<br>00：字节（8 位）<br>01：半字（16 位）<br>10：字（32 位）<br>11：保留<br>这些位受到保护，只有 EN 为“0”时才可以写入。 在直接模式下，当位 EN =“1”时，MSIZE 位由硬件强制置为与 PSIZE 相同的值。 |
| 位 12:11 | PSIZE | 外设数据大小<br>这些位将由软件置 1 和清零。<br>00：字节（8 位）<br>01：半字（16 位）<br>10：字（32 位）<br>11：保留<br>这些位受到保护，只有 EN 为“0”时才可以写入 |
| 位 10 | MINC | 存储器递增模式<br>此位由软件置 1 和清零。<br>0：存储器地址指针固定<br>1：每次数据传输后，存储器地址指针递增（增量为 MSIZE 值）<br>此位受到保护，只有 EN 为“0”时才可以写入。 |
| 位 9 | PINC | 外设递增模式<br>此位由软件置 1 和清零。<br>0：外设地址指针固定<br>1：每次数据传输后，外设地址指针递增（增量为 PSIZE 值）<br>此位受到保护，只有 EN 为“0”时才可以写入。 |
| 位 8 | CIRC | 循环模式<br>此位由软件置 1 和清零，并可由硬件清零。<br>0：禁止循环模式<br>1：使能循环模式<br>如果外设为流控制器（位 PFCTRL=1）且使能数据流（位 EN=1），此位由硬件自动强制清零。如果 DBM 位置 1，当使能数据流（位 EN =“1”）时，此位由硬件自动强制置 1。 |
| 位 7:6 | DIR | 数据传输方向<br>这些位将由软件置 1 和清零。<br>00：外设到存储器<br>01：存储器到外设<br>10：存储器到存储器<br>11：保留<br>这些位受到保护，只有 EN 为“0”时才可以写入。 |
| 位 5 | PFCTRL | 外设流控制器<br>此位由软件置 1 和清零。<br>0：DMA 是流控制器<br>1：外设是流控制器<br>此位受到保护，只有 EN 为“0”时才可以写入。 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 选择存储器到存储器模式（位 DIR[1:0]=10）后，此位由硬件自动强制清零。 |
| 位 4 | TCIE | 传输完成中断使能<br>此位由软件置 1 和清零。<br>0：禁止 TC 中断<br>1：使能 TC 中断 |
| 位 3 | HTIE | 半传输中断使能<br>此位由软件置 1 和清零。<br>0：禁止 HT 中断<br>1：使能 HT 中断 |
| 位 2 | TEIE | 传输错误中断使能<br>此位由软件置 1 和清零。<br>0：禁止 TE 中断<br>1：使能 TE 中断 |
| 位 1 | DMEIE | 直接模式错误中断使能<br>此位由软件置 1 和清零。<br>0：禁止 DME 中断<br>1：使能 DME 中断 |
| 位 0 | EN | 数据流使能/读作低电平时数据流就绪标志<br>此位由软件置 1 和清零。<br>0：禁止数据流<br>1：使能数据流<br>以下情况下，此位可由硬件清零：<br>DMA 传输结束时（准备好配置数据流）<br>AHB 主总线出现传输错误时<br>存储器 AHB 端口上的 FIFO 阈值与突发大小不兼容时<br>此位读作 0 时，软件可以对配置和 FIFO 位寄存器编程。EN 位读作 1 时，禁止向这些寄存 器执行写操作。<br>*注意： 将 EN 位置"1"以启动新传输之前，DMA_LISR 或 DMA_HISR 寄存器中与数据流相对应的事件标志必须清零。* |

### 9.4.6. DMA 数据流 x 数据项数寄存器 (DMA_SxNDTR) (x = 0..7)

偏移地址： 0x14 + 0x18 * 数据流编号

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | NDT | 要传输的数据项数目<br>要传输的数据项数目（0 到 4294967295）。<br>只有在禁止数据流时，才能向此寄存器执行写操作。<br>使能数据流后，此寄存器为只读，用于指示要传输的剩余数据项数。每次 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | DMA 传输后，此寄存器将递减。<br>传输完成后，此寄存器保持为零（数据流处于正常模式时），或者在以下情况下自动以先前编程的值重载：<br>以循环模式配置数据流时。<br>通过将 EN 位置“1”来重新使能数据流时<br>如果该寄存器的值为零，则即使使能数据流，也无法完成任何事务。 |

### 9.4.7. DMA 数据流 x 外设地址寄存器 (DMA_SxPAR) (x = 0..7)

偏移地址： 0x18 + 0x18 * 数据流编号

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:0 | PAR | 外设地址<br>读/写数据的外设数据寄存器的基址。<br>这些位受到写保护，只有 DMA_SxCR 寄存器中的 EN 为“0”时才可以写入。 |

### 9.4.8. DMA 数据流 x 存储器 0 地址寄存器 (DMA_SxM0AR) (x = 0..7)

偏移地址： 0x1C + 0x18 * 数据流编号

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| M0AR[31:16] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| M0AR[15:0] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:0 | M0AR | 存储器 0 地址（用于双缓冲区模式）<br>读/写数据的存储区 0 的基址。<br>此寄存器仅用于双缓冲区模式。<br>这些位受到写保护，只有在以下情况下才可以写入：<br>禁止数据流（DMA_SxCR 寄存器中的位 EN=“0”）<br>使能数据流（DMA_SxCR 寄存器中的 EN=“1”）并且 DMA_SxCR 寄存器中的位 CT =“1”（在双缓冲模式下）。 |

### 9.4.9. DMA 数据流 x 存储器 1 地址寄存器 (DMA_SxM1AR) (x = 0..7)

偏移地址： 0x20 + 0x18 * 数据流编号

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| M1AR[31:16] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| M1AR[15:0] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | M1AR | 存储器 1 地址（用于双缓冲区模式）<br>读/写数据的存储区 1 的基址。<br>此寄存器仅用于双缓冲区模式。<br>这些位受到写保护，只有在以下情况下才可以写入：<br>禁止数据流（DMA_SxCR 寄存器中的位 EN=“0”）<br>使能数据流（DMA_SxCR 寄存器中的 EN=“1”）并且 DMA_SxCR 寄存器中的位 CT =“0”。 |

### 9.4.10. DMA 数据流 x FIFO 控制寄存器 (DMA_SxFCR) (x = 0..7)

偏移地址： 0x24 + 0x24 * 数据流编号

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:8 | 保留 | 必须保持复位值 |
| 位 7 | FEIE | FIFO 错误中断使能<br>此位由软件置 1 和清零。<br>0：禁止 FE 中断<br>1：使能 FE 中断 |
| 位 6 | 保留 | 必须保持复位值 |
| 位 5:3 | FS[2:0] | FIFO 状态<br>这些位为只读。<br>000：0 <fifo_level< 1/4<br>001：1/4 ≤ fifo_level< 1/2<br>010：1/2 ≤ fifo_level< 3/4<br>011：3/4 ≤ fifo_level< 满<br>100：FIFO 为空 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 101：FIFO 已满<br>其它：无意义在直接模式（DMDIS 位为零）下，这些位无意义。 |
| 位 2 | DMDIS | 直接模式禁止<br>此位由软件置 1 和清零。<br>它可由硬件置 1。<br>0：使能直接模式<br>1：禁止直接模式 此位受到保护，只有 EN 为“0”时才可以写入。<br>如果选择存储器到存储器模式（DMA_SxCR 中的 DIR 位为“10”），并且 DMA_SxCR 寄存器中的 EN 位为“1”，则此位由硬件置 1，因为在存储器到存储器配置不能使用直接模式。 |
| 位 1:0 | FTH[1:0] | FIFO 阈值选择<br>这些位将由软件置 1 和清零。<br>00：FIFO 容量的 1/4<br>01：FIFO 容量的 1/2<br>10：FIFO 容量的 3/4<br>11：FIFO 完整容量在直接模式（DMDIS 值为零）下，不使用这些位。 这些位受到保护，只有 EN 为“1”时才可以写入。 |

### 9.4.11. DMA 数据流链块地址寄存器(DMA_SxLR) (x = 0..7)

偏移地址： 0xD0 + 0x04 * 数据流编号

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:0 | LAR | 链块地址<br>读/写数据的链块数据寄存器的基址。<br>这些位受到写保护，只有 DMA_SxCR 寄存器中的 EN 为“0”时才可以写入。 |

# 10. 循环冗余校验计算单元 (CRC)

## 10.1. CRC 简介

循环冗余校验(CRC)计算单元是根据选定的生产多项式得到任一 16/32 位的 CRC 计算结果。在其他的应用中，CRC 技术主要应用于核实数据传输或者数据存储的正确性和完整性。

## 10.2. CRC 主要特性

- 支持 CRC-16/32 两种 CRC 计算方式；
- 支持 CRC-16 多项式：0x8005 和 0x1021；
- 支持 CRC-32 多项式：0x04C11DB7；
- 输入数据可配置由硬件进行大小端翻转；
- 输出数据可配置由硬件进行大小端翻转；
- 支持指定 CRC 计算初始值；
- 支持选择 CRC 计算的 BYTE 数量；
- 一个 32 位数据寄存器用于输入/输出；
- CRC 计算时间： 4 个 AHB 时钟周期(HCLK)；
- 通用 8 位寄存器(可用于存放临时数据)。

## 10.3. CRC 功能描述

CRC 计算单元含有 1 个 32 位数据寄存器：

- 对该寄存器进行写操作时，作为输入寄存器，可以输入要进行 CRC 计算的新数据。
- 对该寄存器进行读操作时，返回上一次 CRC 计算的结果。

每一次写入数据寄存器，其计算结果是前一次 CRC 计算结果和新计算结果的组合(对整个 32 位字进行 CRC 计算，而不是逐字节地计算)。

在 CRC 计算期间会暂停 CPU 的写操作，因此可以对寄存器 CRC_DR 进行背靠背写入或者连续地写-读操作。

可以通过设置寄存器 CRC_CR 的 RESET 位来重置寄存器 CRC_DR 为 0xFFFF FFFF。该操作不影响寄存器 CRC_IDR 内的数据。

## 10.4. CRC 寄存器

### 10.4.1. CRC 数据寄存器(CRC_DR)

地址偏移： 0x00

复位值： 0x00000000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | DR | 数据寄存器位<br>写入 CRC 计算器的新数据时，作为输入寄存器<br>读取时返回 CRC 计算的结果 |

### 10.4.2. CRC 独立数据寄存器(CRC_IDR)

地址偏移： 0x04

复位值： 0x00000000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:8 | 保留 | 必须保持复位值 |
| 位 7:0 | IDR | 通用 8 位数据寄存器位<br>可用于临时存放 1 字节的数据。<br>寄存器 CRC_CR 的 RESET 位产生的 CRC 复位对本寄存器没有影响<br>此寄存器不参与 CRC 计算，可以存放任何数据 |

### 10.4.3. CRC 控制寄存器(CRC_CR)

地址偏移： 0x08

复位值： 0x00000000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:1 | 保留 | 必须保持复位值 |
| 位 0 | RESET | 复位 CRC 计算单元，设置数据寄存器为 0xFFFF FFFF。<br>只能对该位写‘1’，它由硬件自动清‘0’。 |

### 10.4.4. CRC 控制状态寄存器(CRC_CSR)

地址偏移： 0x0C

复位值： 0x000000CA

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | BYTE_NUM | | XOROUT | REVOUT | BYTE_REV | REVIN | TYPE | POLY |
| | | | | | | | | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:8 | 保留 | 必须保持复位值 |
| 位 7:6 | BYTE_NUM | 选择 CRC 计算的 byte 数量<br>00：只对 byte0 进行 CRC 运算<br>01：byte0,byte1<br>10：byte0,byte1,byte2<br>11：byte0,byte1,byte2,byte3<br>*注：配置为 CRC16 时只能配置 0,1；*<br>*配置为 CRC32 时可以全部配置。* |
| 位 5 | XOR_OUT | CRC 计算结果和 0xffff_ffff 进行异或；<br>1：CRC32 和 0xffff_ffff 进行异或,CRC16 和 0x0000_ffff 进行异或<br>0：计算结果直接输出 |
| 位 4 | REV_OUT | CRC 计算结果高低位反转<br>1：反转<br>0：不反转 |
| 位 3 | BYTE_REV | CRC 输入 byte 大小端翻转<br>CRC32<br>1：计算顺序 byte3->byte2->byte1->byte0<br>0：计算顺序 byte0->byte1->byte2->byte3<br>*注：该寄存器会受 BYTE_NUM 选择而改变* |
| 位 2 | REV_IN | CRC 8-bit 输入大小端反转进行计算，如 bit7 作为 bit0 参与运算，bit6 作为 bit1，以此类推<br>1：反转<br>0：不反转 |
| 位 1 | TYPE | CRC 类型选择<br>1：CRC32<br>0：CRC16 |
| 位 0 | POLY | CRC16 的多项式选择，当选择 CRC32 时，该位无效<br>1：0x1021<br>0：0x8005 |

### 10.4.5. CRC 初始值寄存器(CRC_INI)

地址偏移： 0x10

复位值： 0xFFFFFFFF

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | CRC_INI | CRC 计算初始值<br>写入该寄存器后需要配置 RESET 寄存器后载入该值<br>CRC16 时低 16 位有效 |

### 10.4.6. CRC 结果异或寄存器(CRC_XOR)

地址偏移： 0x14

复位值： 0xFFFFFFFF

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | CRC_XOR | CRC 结果异或值<br>写入该寄存器后需要配置 RESET 寄存器后载入该值<br>CRC16 时低 16 位有效 |

# 11. Quad-SPI 控制器(QSPI)

## 11.1. Quad-SPI 简介

Quad-SPI 接口可使用单线(Motorola)、双线、四线模式与外部设备进行通信，通信模式可由软件配置选择。单线模式在设备重置后默认选择。模块接口为 SPI 主设备，向外部从设备提供通信时钟（SCK）。

## 11.2. Quad-SPI 主要特性

- 支持 SPI 主设备模式;
- 可编程的时钟极性和相位;
- 支持单线、双线、四线通讯;
- 支持传输数据大小端转换;
- 两个具有 DMA 功能的 32 位宽度嵌入式 Rx 和 Tx FIFO;
- 高达 32 的 RX 和 Tx FIFO 深度;
- 可选择 4 至 32 位数据大小;
- 支持多主机冲突检测;
- 可编程的接收采样延时时间。

## 11.3. Quad-SPI 功能说明

### 11.3.1. Quad-SPI 引脚与时钟

#### 引脚描述

- D0/MOSI：数据线 0，或主输出/从输入。
- D1/MISO：数据线 1，或主输入/从输出。
- D2：数据线 2。
- D3：数据线 3。
- SCK：串行时钟输出。
- CS：从器件选择。

各模式引脚配置使用见下表。

表 11-1 不同模式下引脚配置

| 引脚 | 单线 SPI 模式 | 双线 SPI 模式 | 四线 SPI 模式 |
|---|---|---|---|
| D0/MOSI | MOSI | D0 | D0 |
| D1/MISO | MISO | D1 | D1 |
| D2 | | | D2 |
| D3 | | | D3 |
| SCK | SCK | SCK | SCK |
| CSN | CSN | CSN | CSN |

#### 模块时钟

Quad-SPI 模块挂载在 AHB 时钟域上，最大支持 AHB 时钟的 2 分频。可通过配置 BAUDR 寄存器中的 SCKDV 位调整模块 SCK 输出频率。

SCKDV 位域的 LSB 总是会保持 0 值，不受写入影响，以保证分频值为偶数。若分频值为 0，则 SCK 的时钟输出关闭。

SCK 频率计算方式如下：

**QuadSPI_CLK = AHB_CLK**

**SCK = QuadSPI_CLK / SCKDV**

### 11.3.2. Quad-SPI 时钟相位与极性

可通过 CTRLR0 寄存器中的 SCPOL 与 SCPH 位，调整时钟的极性和相位。

当 SCPOL(时钟极性)为 0 时，空闲时 SCK 保持低电平。

当 SCPOL(时钟极性)为 1 时，空闲时 SCK 保持高电平。

当 SCPH (时钟相位)为 0 时，在 SCK 引脚上的第一个边沿采样。

当 SCPH (时钟相位)为 1 时，在 SCK 引脚上的第二个边沿采样。

时钟相位组合模式见下表。

表 11-2 时钟相位与极性

| 模式 | SCPOL | SCPH | 传输特性 |
|---|---|---|---|
| Mode_0 | 0 | 0 | CSN 的下降沿表示数据传输的起始；<br>主机和从机在时钟的的第一个边沿处（上升沿）采样数据；<br>数据要求在第一个时钟边沿（上升沿）之前就有效； |
| Mode_1 | 0 | 1 | CSN 信号有效且 SCK 的第一个边沿（上升沿）表示数据传输的起始；<br>主机和从机在时钟的的第二个边沿处（下降沿）采样数据；<br>数据在第一个时钟边沿（上升沿）发送； |
| Mode_2 | 1 | 0 | CSN 的下降沿表示数据传输的起始；<br>主机和从机在时钟的的第一个边沿处（下降沿）采样数据；<br>数据要求在第一个时钟边沿（下降沿）之前就有效； |
| Mode_3 | 1 | 1 | CSN 信号有效且 SCK 的第一个边沿（下降沿）表示数据传输的起始；<br>主机和从机在时钟的的第二个边沿处（上升沿）采样数据；<br>数据在第一个时钟边沿（下降沿）发送； |

### 11.3.3. Quad-SPI 传输模式

#### 发送和接收模式

当 TMOD=0x0 时，接收和发送逻辑均会生效。会按照选择的帧格式正常产生数据传输。发送数据从发送 FIFO 中取出并通过 txd 线发送到目标设备，并通过 rxd 线接收回复数据。从目标设备接收到的数据会先经过内部移位寄存器后放置在接收 FIFO 中。

*注意：发送和接收模式（TMOD=0x0）只应用在 QUADSPI 配置为单线模式时。*

**只发/只写模式**

当 TMOD=0x1 时，接收数据无效并且不会存储到接收 FIFO 中。数据发送按选择的帧格式正常传输。发送数据从发送 FIFO 中取出并通过 txd 线发送到目标设备。在数据帧的末尾，内部接收移位寄存器不会加载新的接收数据到接收 FIFO 中。若内部接收移位寄存器中存在数据，会造成下一次的发送数据被覆写。当进入只写模式后，应该屏蔽接收相关的中断。

**只收/只读模式**

当 TMOD=0x2 时，发送数据变为无效。在传输过程中 txd 线保持固定值，接收数据通路正常。当进入只读模式后，应该屏蔽发送相关的中断。

### 11.3.4. Quad-SPI 数据大小端转换

通过 CTRLR0 寄存器的 SECONV 位，可设置 DR 寄存器数据的大小端。转换示例见下图。

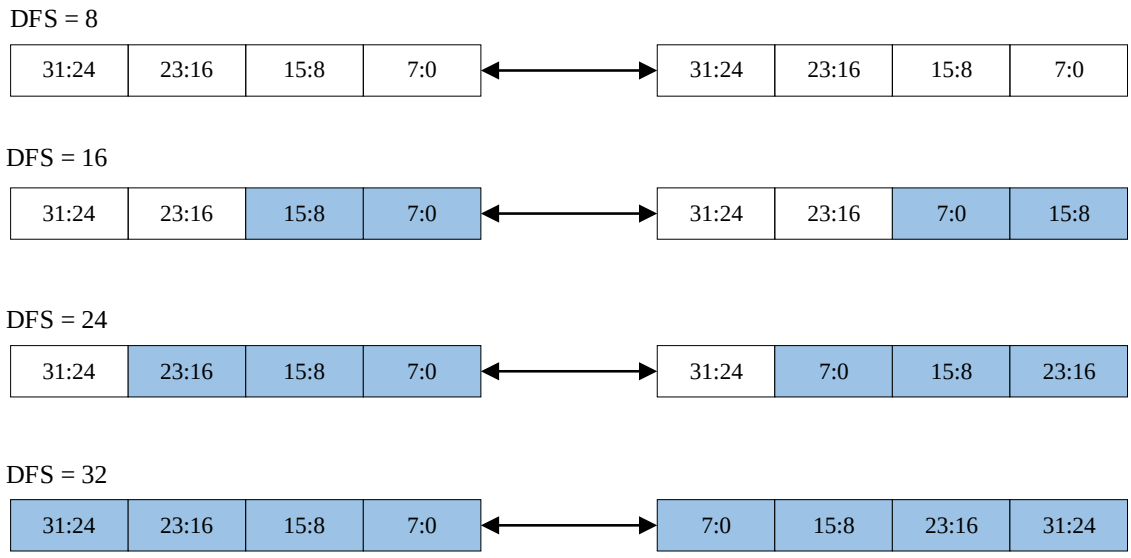



图 11-1 Quad-SPI 数据大小端转换

### 11.3.5. Quad-SPI 接收数据帧长度

通过 CTRLR1 寄存器的 NDF 位，可设置接收数据帧长度。

QUADSPI 会连续接收串行数据直到接收的数据帧数量等于 NDF+1。

*注意：接收数据帧数量(NDF)按接收数据帧计数，实际接收字节长度 = (NDF+1)*DFS/8*

### 11.3.6. Quad-SPI 发送字节长度

通过 BYTE_NUM 寄存器，设置发送数据字节长度。

当 byte_num 配置为 0 时，片选线(CSN)会在间隔 1 个 SCK 周期时间没有新数据发送后拉高。

当 byte_num 配置非 0 值时，若发送数据字节数小于 byte_num 值，片选线(CSN)会一直为拉低状态。发送数据大于等于 byte_num 值，片选线(CSN) 会在间隔 1 个 SCK 周期时间没有新数据发送后拉高。配置 byte_num 可防止由于数据传输过程不连贯（数据间间隔超过 1 个 SCK 周期的时间）造成的 CSN 线异常拉

高。

*注意：*

- *与接收数据帧数量 NDF 不同，byte_num 按发送字节个数计数。*
- *在双线/四线模式下，byte_num 不计算指令和地址的长度，只计算发送数据的字节数。*
- *在标准 SPI(单线)模式下，则需要用户考虑指令 inst 和地址 addr 及数据 data 的字节数，一起算进 byte_num 中，这是单线模式与双/四线模式的区别。*

### 11.3.7. Quad-SPI DMA 控制器接口

通过 CTRLR1 寄存器的 TDMAE 位与 RDMAE 位，可设置 DMA 发送与 DMA 接收的使能状态。

配置 DMATDLR 寄存器，设置 DMA 发送阈值。当发送 FIFO 中有效数据个数小于或等于 DMATDL，且已使能 QUADSPI 的 DMA 发送(TDMAE=0x1)的情况下，会生成 DMA 发送请求。

配置 DMARDLR 寄存器，设置 DMA 接收阈值。当接收 FIFO 中有效数据个数大于或等于 DMARDL+1，且已使能 QUADSPI 的 DMA 接收(RDMAE=0x1)的情况下，会生成 DMA 接收请求。

### 11.3.8. Quad-SPI 状态标志

**数据冲突错误（DCOL）**

当 QUADSPI 被线路上存在的其他主机打断传输过程时，会发生数据冲突错误，这会自动将 DCOL 位置 1。

读取 SR 寄存器将 DCOL 位清零。

**接收 FIFO 满（RFF）**

此标志置 1 时，表示接收 FIFO 数据数量大于等于设定的接收 FIFO 阈值（RFF+1）。

当接收 FIFO 距离接收 FIFO 阈值存在一个或更多空位置时，此位清除。

**接收 FIFO 非空（RFNE）**

此标志置 1 时，表示接收 FIFO 存在一个或更多数据时。

当接收 FIFO 没有数据时，此位清除。

**发送 FIFO 空（TFE）**

此标志置 1 时，表示发送 FIFO 不存在数据。

当发送 FIFO 存在一个或更多数据时，此位清除。

**发送 FIFO 未满（TFNF）**

此标志置 1 时，表示当前发送 FIFO 中数据数量小于等于设定的发送 FIFO 阈值（TFT）。

当发送 FIFO 大于设定的发送 FIFO 阈值时，此位清除。

**BUSY**

BUSY 标志由硬件置 1 和清零。用于指示 QUADSPI 的通信状态。

当 BUSY 置 1 时，标志 QUADSPI 正处于传输过程中。

当 BUSY 清零时，表示 QUADSPI 处于关闭或空闲状态。

### 11.3.9. Quad-SPI 中断

表 11-3 Quad-SPI 中断

| 中断事件 | 事件标志 | 使能控制位 |
|---|---|---|
| 多主机冲突中断 | MSTIS | MSTIM |
| 接收 FIFO 满中断 | RXFIS | RXFIM |
| 接收 FIFO 上溢中断 | RXOIS | RXOIM |
| 接收 FIFO 下溢中断 | RXUIS | RXUIM |
| 发送 FIFO 上溢中断 | TXOIS | TXOIM |
| 发送 FIFO 空中断 | TXEIS | TXEIM |

### 11.3.10. Quad-SPI 配置

**单线(Motorola)模式**

通过 CTRLR0 寄存器中的 SPI_FRF 位为 0x0，选择 QUADSPI 以单线模式运行。

编程步骤：

- 设置清除 SSI_EN 位，保证 QUADSPI 模块为关闭状态。
- 配置 SER 位，使能模块从机通路。
- 配置 SCPOL 和 SCPH，设置定义数据传输和串行时钟之前的关系。
- 配置 BAUDR 寄存器，选择串行时钟运行频率。
- 配置 SSTE 位，保证 CSN 信号在数据传输过程中状态符合预期。
- 配置 TMOD 位，选择传输模式是只发、只收或是收发同时进行。
- 按需求配置 DFS_32 与 SECONV 位，选择传输数据大小和数据的大小端。
- 配置 SPI_FRF 选择单线模式。
- 配置 TXFTLR 和 RXFTLR 寄存器，设置模块发送与接收 FIFO 的阈值。
- 配置 NDF 位，设置接收数据帧数量。
- 配置 BYTE_NUM 寄存器，设置发送数据字节数。
- 按需求配置 IMR 寄存器，启用对应中断。
- 按需求配置 DMACR 和 DMATDLR/ DMARDLR 寄存器，设置 DMA 请求阈值，以及是否启用 DMA。
- 设置 SSI_EN 位，启用 QUADSPI 模块。

**双线模式**

通过 CTRLR0 寄存器中的 SPI_FRF 位为 0x1,选择 QUADSPI 以双线 SPI 模式运行。

双线/四线 SPI 模式下，QUADSPI 支持增强 SPI 模式，可通过配置 SPI_CTRLR0 寄存器，对地址及指令的传输模式和长度等参数进行配置，详细见 SPI_CTRLR0 寄存器描述。

编程步骤：

- 设置清除 SSI_EN 位，保证 QUADSPI 模块为关闭状态。
- 配置 SER 位，使能模块从机通路。
- 配置 SCPOL 和 SCPH，设置定义数据传输和串行时钟之前的关系。

- 配置 BAUDR 寄存器，选择串行时钟运行频率。
- 配置 SSTE 位，保证 CSN 信号在数据传输过程中状态符合预期。
- 配置 TMOD 位，选择传输模式是读取或是写入。
- 按需求配置 DFS_32 与 SECONV 位，选择传输数据大小和数据的大小端。
- 配置 SPI_FRF 选择双线 SPI 模式。
- 配置 SPI_CTRLR0 寄存器，选择地址及指令长度和传输方式，以及等待周期。
- 配置 RX_SAMPLY_DLY 寄存器，设置接收延迟采样参数。
- 配置 TXFTLR 和 RXFTLR 寄存器，设置模块发送与接收 FIFO 的阈值。
- 配置 NDF 位，设置接收数据帧数量。
- 配置 BYTE_NUM 寄存器，设置发送数据字节数。
- 按需求配置 IMR 寄存器，启用对应中断。
- 按需求配置 DMACR 和 DMATDLR/DMARDLR 寄存器，设置 DMA 请求阈值，以及是否启用 DMA。
- 设置 SSI_EN 位，启用 QUADSPI 模块。

**四线模式**

通过 CTRLR0 寄存器中的 SPI_FRF 位为 0x2,选择 QUADSPI 以四线 SPI 模式运行。

双线/四线 SPI 模式下，QUADSPI 支持增强 SPI 模式，可通过配置 SPI_CTRLR0 寄存器，对地址及指令的传输模式和长度等参数进行配置，详细见 SPI_CTRLR0 寄存器描述。

编程步骤:

- 设置清除 SSI_EN 位，保证 QUADSPI 模块为关闭状态。
- 配置 SER 位，使能模块从机通路。
- 配置 SCPOL 和 SCPH，设置定义数据传输和串行时钟之前的关系。
- 配置 BAUDR 寄存器，选择串行时钟运行频率。
- 配置 SSTE 位，保证 CSN 信号在数据传输过程中状态符合预期。
- 配置 TMOD 位，选择传输模式是读取或是写入。
- 按需求配置 DFS_32 与 SECONV 位，选择传输数据大小和数据的大小端。
- 配置 SPI_FRF 选择四线 SPI 模式。
- 配置 SPI_CTRLR0 寄存器，选择地址及指令长度和传输方式，以及等待周期。
- 配置 RX_SAMPLY_DLY 寄存器，设置接收延迟采样参数。
- 配置 TXFTLR 和 RXFTLR 寄存器，设置模块发送与接收 FIFO 的阈值。
- 配置 NDF 位，设置接收数据帧数量。
- 配置 BYTE_NUM 寄存器，设置发送数据字节数。
- 按需求配置 IMR 寄存器，启用对应中断。
- 按需求配置 DMACR 和 DMATDLR/DMARDLR 寄存器，设置 DMA 请求阈值，以及是否启用 DMA。
- 设置 SSI_EN 位，启用 QUADSPI 模块。

## 11.4. Quad-SPI 寄存器

### 11.4.1. Quad-SPI 控制寄存器 0 (CTRL0)

偏移地址： 0x00

复位值： 0x0107 0000

当 QUADSPI 为开启状态(SSIENR = 0x1)时，此寄存器不可写入

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | SECONV | SSTE | RSD | SPI_FRF | | DFS_32 | | | | |
| | | | | | | rw | rw | | rw | rw | rw | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | TMOD | | SCPOL | SCPH | Reserved | | | | | |
| | | | | | | rw | rw | rw | rw | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:26 | 保留 | 必须保持复位值。 |
| 25 | SECONV | 设立大小端<br>0：关闭大小端转换.<br>1：启动大小端转换 |
| 24 | SSTE | 从选择翻转使能<br>0：传输过程中从选择线(CS)保持为低<br>1：传输过程中从选择线(CS)在数据间翻转，当 CS 为高时 SCK 会保持默认值 |
| 23 | 保留 | 必须保持复位值。 |
| 22:21 | SPI_FRF | SPI 模式选择<br>0：标准单线 SPI 模式<br>1：双线 SPI 模式<br>2：四线 SPI 模式<br>3：保留 |
| 20:16 | DFS_32 | 数据帧大小(Data Frame Size)<br>■ 0x3 (FRAME_04BITS): 按 4 位串行数据传输<br>■ 0x4 (FRAME_05BITS): 按 5 位串行数据传输<br>■ 0x5 (FRAME_06BITS): 按 6 位串行数据传输<br>■ 0x6 (FRAME_07BITS): 按 7 位串行数据传输<br>■ 0x7 (FRAME_08BITS): 按 8 位串行数据传输<br>■ 0x8 (FRAME_09BITS): 按 9 位串行数据传输<br>■ 0x9 (FRAME_10BITS): 按 10 位串行数据传输<br>■ 0xa (FRAME_11BITS): 按 11 位串行数据传输<br>■ 0xb (FRAME_12BITS): 按 12 位串行数据传输<br>■ 0xc (FRAME_13BITS): 按 13 位串行数据传输<br>■ 0xd (FRAME_14BITS): 按 14 位串行数据传输<br>■ 0xe (FRAME_15BITS): 按 15 位串行数据传输<br>■ 0xf (FRAME_16BITS): 按 16 位串行数据传输<br>■ 0x10 (FRAME_17BITS): 按 17 位串行数据传输<br>■ 0x11 (FRAME_18BITS): 按 18 位串行数据传输 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | ■ 0x12 (FRAME_19BITS): 按 19 位串行数据传输<br>■ 0x13 (FRAME_20BITS): 按 20 位串行数据传输<br>■ 0x14 (FRAME_21BITS): 按 21 位串行数据传输<br>■ 0x15 (FRAME_22BITS): 按 22 位串行数据传输<br>■ 0x16 (FRAME_23BITS): 按 23 位串行数据传输<br>■ 0x17 (FRAME_24BITS): 按 24 位串行数据传输<br>■ 0x18 (FRAME_25BITS): 按 25 位串行数据传输<br>■ 0x19 (FRAME_26BITS): 按 26 位串行数据传输<br>■ 0x1a (FRAME_27BITS): 按 27 位串行数据传输<br>■ 0x1b (FRAME_28BITS): 按 28 位串行数据传输<br>■ 0x1c (FRAME_29BITS): 按 29 位串行数据传输<br>■ 0x1d (FRAME_30BITS): 按 30 位串行数据传输<br>■ 0x1e (FRAME_31BITS): 按 31 位串行数据传输<br>■ 0x1f (FRAME_32BITS): 按 32 位串行数据传输<br>*注意:*<br>*当 SPI_FRF=0x01（双线模式）时，DFS 应为 2 的倍数。*<br>*当 SPI_FRF=0x10（四线模式）时，DFS 应为 4 的倍数。* |
| 15:10 | 保留 | 必须保持复位值。 |
| 9:8 | TMOD | 传输模式<br>0：接收&发送<br>1：只发模式<br>2：只收模式<br>3：保留<br><br>当 SPI_FRF 配置为双线、四线模式时只有如下两个有效配置<br>1：写模式<br>2：读模式 |
| 7 | SCPOL | 串行时钟极性<br>0：空闲时 SCK 保持低电平<br>1：空闲时 SCK 保持高电平 |
| 6 | SCPH | 串行时钟相位<br>0：从第一个时钟边沿开始采样数据<br>1：从第二个时钟边沿开始采样数据 |
| 5:0 | 保留 | 必须保持复位值。 |

### 11.4.2. Quad-SPI 控制寄存器 1 (CTRL1)

偏移地址： 0x04

复位值： 0x0000 0000

当 QUADSPI 为开启状态(SSIENR = 0x1)时，此寄存器不可写入

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15:0 | NDF | 数据帧数量<br>当 TMOD=0x10 或 TMOD=0x11 时，此位域设置 QUADSPI 连续接收的数据数量。QUADSPI 会连续接收串行数据直到接收的数据数量等于 NDF+1。<br>*注意：接收数据帧数量(NDF)按接收次数计数，实际接收字节长度=(NDF+1)*DFS/8* |

### 11.4.3. Quad-SPI 串行传输使能寄存器 (SSIENR)

偏移地址： 0x08

复位值： 0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:1 | 保留 | 必须保持复位值。 |
| 0 | EN | QUADSPI 模块使能<br>0：关闭 QUADSPI 模块<br>1：使能 QUADSPI 模块 |

### 11.4.4. Quad-SPI 从选择寄存器 (SER)

偏移地址： 0x10

复位值： 0x0000 0000

当 QUADSPI 模块关闭(SSI_EN=0x0)时，此寄存器可写入，可清除。

当 QUADSPI 模块开启(SSI_EN=0x1)时，此寄存器可写入，不可清除。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:4 | 保留 | 必须保持复位值。 |
| 3:0 | SER | 从选择使能标志<br>0：无从机使能<br>1：使能从机 0<br>其他：保留<br>*注意：此位域应始终配置为 0x1。* |

### 11.4.5. Quad-SPI 速率配置寄存器 (BAUDR)

偏移地址： 0x14

复位值： 0x0000 0000

当 QUADSPI 为开启状态(SSIENR = 0x1)时，此寄存器不可写入

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15:0 | SCKDV | QUADSPI 时钟分频<br>此位域 LSB 总是会保持 0 值，不受写入影响，以保证分频值为偶数。若分频值为 0，则 SCK 的时钟输出关闭。<br>SCK 计算方式如下：<br>SCK = QUADSPI_CLK/SCKDV |

### 11.4.6. Quad-SPI 发送 FIFO 阈值寄存器 (TXFTLR)

偏移地址： 0x18

复位值： 0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:5 | 保留 | 必须保持复位值。 |
| 4:0 | TFT | 发送 FIFO 阈值<br>设置发送 FIFO 控制器触发中断的阈值。<br>若当前发送 FIFO 中数据数量小于等于 TFT 时，发送 FIFO 空中断会被触发 |

### 11.4.7. Quad-SPI 接收 FIFO 阈值寄存器 (RXFTLR)

偏移地址： 0x1C

复位值： 0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:5 | 保留 | 必须保持复位值。 |
| 4:0 | RFT | 接收 FIFO 阈值<br>设置接收 FIFO 控制器触发中断的阈值。<br>若当前接收 FIFO 中数据数量大于等于 RFF+1 时，接收 FIFO 满中断会被触发 |

### 11.4.8. Quad-SPI 发送 FIFO 数据数量寄存器 (TXFLR)

偏移地址： 0x20

复位值： 0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:6 | 保留 | 必须保持复位值。 |
| 5:0 | TXFLR | 发送 FIFO 数据数量<br>当前在发送 FIFO 中的有效数据个数 |

### 11.4.9. Quad-SPI 接收 FIFO 数据数量寄存器 (RXFLR)

偏移地址： 0x24

复位值： 0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:6 | 保留 | 必须保持复位值。 |
| 5:0 | RXFLR | 接收 FIFO 数据数量<br>当前接收 FIFO 中有效数据个数 |

### 11.4.10. Quad-SPI 状态寄存器 (SR)

偏移地址： 0x28

复位值： 0x0000 0006

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | DCOL | RSD | RFF | RFNE | TFE | TFNF | BUSY |
| | | | | | | | | | r | | r | r | r | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:7 | 保留 | 必须保持复位值。 |
| 6 | DCOL | 数据冲突错误<br>当 QUADSPI 被线路上存在的其他主机打断传输过程，此位置位。读取以清除此位。<br>0：无错误<br>1：数据冲突错误 |
| 5 | 保留 | 必须保持复位值。 |
| 4 | RFF | 接收 FIFO 满<br>当接收 FIFO 满时此位置位。<br>当接收 FIFO 存在一个或更多空位置时，此位清除。<br>0：接收 FIFO 未满<br>1：接收 FIFO 已满 |
| 3 | RFNE | 接收 FIFO 非空<br>当接收 FIFO 为空时，此位清除。<br>当接收 FIFO 存在一个或更多数据时，此位置位。<br>0：接收 FIFO 为空<br>1：接收 FIFO 非空 |
| 2 | TFE | 发送 FIFO 空<br>当发送 FIFO 为空时，此位置位。<br>当发送 FIFO 存在一个或更多有效数据时，此位清除。<br>此位域变化不会请求中断。<br>0：发送 FIFO 非空<br>1：发送 FIFO 空 |
| 1 | TFNF | 发送 FIFO 未满<br>当发送 FIFO 存在一个或更多空位置时，此位置位。当发送 FIFO 满时，此位清除。<br>0：发送 FIFO 满<br>1：发送 FIFO 未满 |
| 0 | BUSY | QUADSPI 忙标志<br>当 QUADSPI 处于传输过程中时，此位置位。<br>当 QUADSPI 处于空闲或关闭状态时，此位清除。<br>0：QUADSPI 处于空闲或关闭状态<br>1：QUADSPI 处于数据传输状态 |

### 11.4.11. Quad-SPI 中断屏蔽寄存器 (IMR)

偏移地址： 0x2C

复位值： 0x0000 003F

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | | | | MSTIM | RXFIM | RXOIM | RXUIM | TXOIM | TXEIM |
| | | | | | | | | | | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:6 | 保留 | 必须保持复位值。 |
| 5 | MSTIM | 多主机冲突中断屏蔽位<br>0：屏蔽多主机冲突中断<br>1：不屏蔽多主机冲突中断 |
| 4 | RXFIM | 接收 FIFO 满中断屏蔽位<br>0：屏蔽接收 FIFO 满中断<br>1：不屏蔽接收 FIFO 满中断 |
| 3 | RXOIM | 接收 FIFO 上溢中断屏蔽位<br>0：屏蔽接收 FIFO 上溢中断<br>1：不屏蔽接收 FIFO 上溢中断 |
| 2 | RXUIM | 接收 FIFO 下溢中断屏蔽位<br>0：屏蔽接收 FIFO 下溢中断<br>1：不屏蔽接收 FIFO 下溢中断 |
| 1 | TXOIM | 发送 FIFO 上溢中断屏蔽位<br>0：屏蔽发送 FIFO 上溢中断<br>1：不屏蔽发送 FIFO 上溢中断 |
| 0 | TXEIM | 发送 FIFO 空中断屏蔽位<br>0：屏蔽发送 FIFO 空中断<br>1：不屏蔽发送 FIFO 空中断 |

### 11.4.12. Quad-SPI 中断状态寄存器 (ISR)

偏移地址： 0x30

复位值： 0x0000 0000

此寄存器只反馈未被中断屏蔽的中断状态。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | | | | MSTIS | RXFIS | RXOIS | RXUIS | TXOIS | TXEIS |
| | | | | | | | | | | r | r | r | r | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:6 | 保留 | 必须保持复位值。 |
| 5 | MSTIS | 多主机冲突中断状态 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 0：多主机冲突中断未触发<br>1：多主机冲突中断已触发 |
| 4 | RXFIS | 接收 FIFO 满中断状态<br>0：接收 FIFO 满中断未触发<br>1：接收 FIFO 满中断已触发 |
| 3 | RXOIS | 接收 FIFO 上溢中断状态<br>0：接收 FIFO 上溢中断未触发<br>1：接收 FIFO 上溢中断已触发 |
| 2 | RXUIS | 接收 FIFO 下溢中断状态<br>0：接收 FIFO 下溢中断未触发<br>1：接收 FIFO 下溢中断已触发 |
| 1 | TXOIS | 发送 FIFO 上溢中断状态<br>0：发送 FIFO 上溢中断未触发<br>1：发送 FIFO 上溢中断已触发 |
| 0 | TXEIS | 发送 FIFO 空中断状态<br>0：发送 FIFO 空中断未触发<br>1：发送 FIFO 空中断已触发 |

### 11.4.13. Quad-SPI 源中断状态寄存器 (RISR)

偏移地址： 0x34

复位值： 0x0000 0000

此寄存器反馈中断状态不受中断屏蔽影响，无论对应中断是否被屏蔽。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | | | | MSTIR | RXFIR | RXOIR | RXUIR | TXOIR | TXEIR |
| | | | | | | | | | | r | r | r | r | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:6 | 保留 | 必须保持复位值。 |
| 5 | MSTIR | 多主机冲突源中断状态<br>0：多主机冲突源中断未触发<br>1：多主机冲突源中断已触发 |
| 4 | RXFIR | 接收 FIFO 满源中断状态<br>0：接收 FIFO 满源中断未触发<br>1：接收 FIFO 满源中断已触发 |
| 3 | RXOIR | 接收 FIFO 上溢源中断状态<br>0：接收 FIFO 上溢源中断未触发<br>1：接收 FIFO 上溢源中断已触发 |
| 2 | RXUIR | 接收 FIFO 下溢源中断状态<br>0：接收 FIFO 下溢源中断未触发<br>1：接收 FIFO 下溢源中断已触发 |
| 1 | TXOIR | 发送 FIFO 上溢源中断状态<br>0：发送 FIFO 上溢源中断未触发<br>1：发送 FIFO 上溢源中断已触发 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 0 | TXEIR | 发送 FIFO 空源中断状态<br>0：发送 FIFO 空源中断未触发<br>1：发送 FIFO 空源中断已触发 |

### 11.4.14. Quad-SPI 发送上溢中断清除寄存器 (TXOICR)

偏移地址： 0x38

复位值： 0x0000 0000

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:1 | 保留 | 必须保持复位值。 |
| 0 | TXOICR | 清发送 FIFO 上溢中断<br>读取此寄存器清除发送 FIFO 上溢中断。<br>此寄存器写入无效。 |

### 11.4.15. Quad-SPI 接收上溢中断清除寄存器 (RXOICR)

偏移地址： 0x3C

复位值： 0x0000 0000

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:1 | 保留 | 必须保持复位值。 |
| 0 | RXOICR | 清接收 FIFO 上溢中断<br>读取此寄存器清除接收 FIFO 上溢中断。<br>此寄存器写入无效。 |

### 11.4.16. Quad-SPI 接收下溢中断清除寄存器 (RXUICR)

偏移地址： 0x40

复位值： 0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:1 | 保留 | 必须保持复位值。 |
| 0 | RXUICR | 清接收 FIFO 下溢中断<br>读取此寄存器清除接收 FIFO 下溢中断。<br>此寄存器写入无效。 |

### 11.4.17. Quad-SPI 多主机冲突中断清除寄存器 (MSTICR)

偏移地址： 0x44

复位值： 0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:1 | 保留 | 必须保持复位值。 |
| 0 | MSTICR | 清多主机冲突中断<br>读取此寄存器清除多主机冲突中断。<br>此寄存器写入无效。 |

### 11.4.18. Quad-SPI 中断清除寄存器 (ICR)

偏移地址： 0x48

复位值： 0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:1 | 保留 | 必须保持复位值。 |
| 0 | ICR | 清中断<br>读取此寄存器清除发送 FIFO 上溢中断、接收 FIFO 上溢中断、接收 FIFO 下溢中断、多主机冲突中断。<br>此寄存器写入无效。 |

### 11.4.19. Quad-SPI DMA 控制寄存器 (DMACR)

偏移地址： 0x4C

复位值： 0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:2 | 保留 | 必须保持复位值。 |
| 1 | TDMAE | 使能 QUADSPI 的 DMA 发送<br>0：关闭 DMA 发送<br>1：使能 DMA 发送 |
| 0 | RDMAE | 使能 QUADSPI 的 DMA 接收<br>0：关闭 DMA 接收<br>1：使能 DMA 接收 |

### 11.4.20. Quad-SPI DMA 发送阈值寄存器 (DMATDLR)

偏移地址： 0x50

复位值： 0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:5 | 保留 | 必须保持复位值。 |
| 4:0 | DMATDL | DMA 发送阈值<br>当发送 FIFO 中有效数据个数小于或等于 DMATDL，且已使能 QUADSPI 的 DMA 发送(TDMAE=0x1)的情况下，会生成 DMA 发送请求。 |

### 11.4.21. Quad-SPI DMA 接收阈值寄存器 (DMARDLR)

偏移地址： 0x54

复位值： 0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:5 | 保留 | 必须保持复位值。 |
| 4:0 | DMARDL | DMA 接收阈值<br>当接收 FIFO 中有效数据个数大于或等于 DMARDL+1，且已使能 QUADSPI 的 DMA 接收(RDMAE=0x1)的情况下，会生成 DMA 接收请求。 |

### 11.4.22. Quad-SPI 数据寄存器 (DR)

偏移地址： 0x60

复位值： 0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | DR | 数据寄存器<br>在写入此寄存器时，需保证数据右对齐。读取此寄存器时，读取的数据会自动右对齐。<br>写入，向发送 FIFO 缓冲区填充数据。<br>读取，从接收 FIFO 缓冲区读取数据<br>*注意：*<br>*写入操作只能在QUADSPI 使能(EN=0x1)后进行，在QUADSPI 关闭(EN=0x0)后FIFO 缓冲区会重置。* |

### 11.4.23. Quad-SPI 接收采样延迟寄存器 (RX_SAMPLY_DLY)

偏移地址： 0xF0

复位值： 0x0000 0000

当 QUADSPI 为开启状态(SSIENR = 0x1)时，此寄存器不可写入

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:6 | 保留 | 必须保持复位值。 |
| 5:0 | RX_SAMPLE_DLY | 接收采样延迟<br>此寄存器用于配置接收端口采样延迟 |

### 11.4.24. Quad-SPI 控制寄存器 (SPI_CTRL2)

偏移地址： 0xF4

复位值： 0x0000 0200

当 QUADSPI 为开启状态(SSIENR = 0x1)时，此寄存器不可写入。

此寄存器相关配置只在 QUADSPI 双线(SPI_FRF=0x01)、四线(SPI_FRF=0x10)模式下生效。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WAIT_CYCLES | | | | | RSD | INST_L | | Reserved | | ADDR_L | | | | TRANS_TYPE | |
| rw | rw | rw | rw | rw | | rw | rw | | | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15:11 | WAIT_CYCLES | 等待周期<br>双线四线模式下控制帧传输和数据接收之间的等待周期数量。 |
| 10 | 保留 | 必须保持复位值。 |
| 9:8 | INST_L | 指令长度<br>双线/四线模式下指令的位长度.<br>0：0 位(无指令)<br>1：4 位指令长度<br>2：8 位指令长度<br>3：16 位指令长度 |
| 7:6 | 保留 | 必须保持复位值。 |
| 5:2 | ADDR_L | 地址长度<br>此位域定义传输的地址长度。配置完成后才可启动 SPI 传输。<br>0x0：0 位地址宽度<br>0x1：4 位地址宽度<br>0x2：8 位地址宽度<br>0x3：12 位地址宽度<br>0x4：16 位地址宽度<br>0x5：20 位地址宽度<br>0x6：24 位地址宽度<br>0x7：28 位地址宽度<br>0x8：32 位地址宽度<br>0x9：36 位地址宽度<br>0xa：40 位地址宽度<br>0xb：44 位地址宽度<br>0xc：48 位地址宽度<br>0xd：52 位地址宽度 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 0xe：56 位地址宽度<br>0xf：60 位地址宽度 |
| 1:0 | TRANS_TYPE | 地址指令传输格式<br>0：指令和地址均按标准 SPI 模式发送<br>1：指令按标准 SPI 模式发送，地址跟随 SPI_FRF 指定格式发送<br>2：指令和地址均跟随 SPI_FRF 指定格式发送<br>3：保留 |

### 11.4.25. Quad-SPI 发送数据字节长度寄存器 (BYTE_NUM)

偏移地址： 0x12C

复位值： 0x0000 0200

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:13 | 保留 | 必须保持复位值。 |
| 12:0 | BYTE_NUM | 发送字节数<br>当 byte_num 配置为 0 时，片选线(CS)会在间隔 1 个 SCLK 周期时间没有新数据发送后拉高。<br>当 byte_num 配置非 0 值时，若发送数据字节数小于 byte_num 值，片选线(CS)会一直为拉低状态。发送数据大于等于 byte_num 值，片选线(CS) 会在间隔 1 个 SCLK 周期时间没有新数据发送后拉高。<br>注意：<br>与接收数据帧数量 NDF 不同，byte_num 按发送字节个数计数。而 NDF 按接收次数计数，实际接收字节数=(NDF+1)*DFS/8。<br>在双线/四线模式下，byte_num 不计算指令和地址的长度，只计算发送数据的字节数。<br>在标准 SPI(单线)模式下，则需要用户考虑指令 inst 和地址 addr 及数据 data 的字节数，一起算进 byte_num 中，这是单线模式与双/四线模式的区别。 |

# 12. 模数转换器(ADC)

## 12.1. ADC 简介

12 位 ADC 是逐次趋近型模数转换器。它具有多达 19 个复用通道，可测量来自 16 个外部源、Temp、Vref 和 VBAT _LI 通道的信号。这些通道的 A/D 转换可在单次、连续、扫描或不连续采样模式下进行。ADC 的结果存储在一个左对齐或右对齐的 16 位数据寄存器中。

ADC 具有模拟看门狗特性，允许应用检测输入电压是否超过了用户自定义的阈值上限或下限。

## 12.2. ADC 主要特性

- 可配置 12 位、10 位、8 位或 6 位分辨率；
- 在转换结束、注入转换结束以及发生模拟看门狗或溢出事件时产生中断；
- 单次和连续转换模式；
- 用于自动将通道 0 转换为通道“n”的扫描模式；
- 数据对齐以保持内置数据一致性；
- 可独立设置各通道采样时间；
- 外部触发器选项，可为规则转换和注入转换配置极性；
- 不连续采样模式；
- 双重/三重模式（具有 2 个或更多 ADC 的器件提供）；
- 双重/三重 ADC 模式下可配置的 DMA 数据存储；
- 双重/三重交替模式下可配置的转换间延迟；
- ADC 转换类型（参见数据手册）；
- ADC 电源要求：全速运行时为 2.4 V 到 3.6 V，慢速运行时为 1.8 V；
- ADC 输入范围：$V_{REF-} \leq VIN \leq V_{REF+}$；
- 规则通道转换期间可产生 DMA 请求。

## 12.3. ADC 功能说明

下图和表分别给出了单个 ADC 的框图及引脚说明。

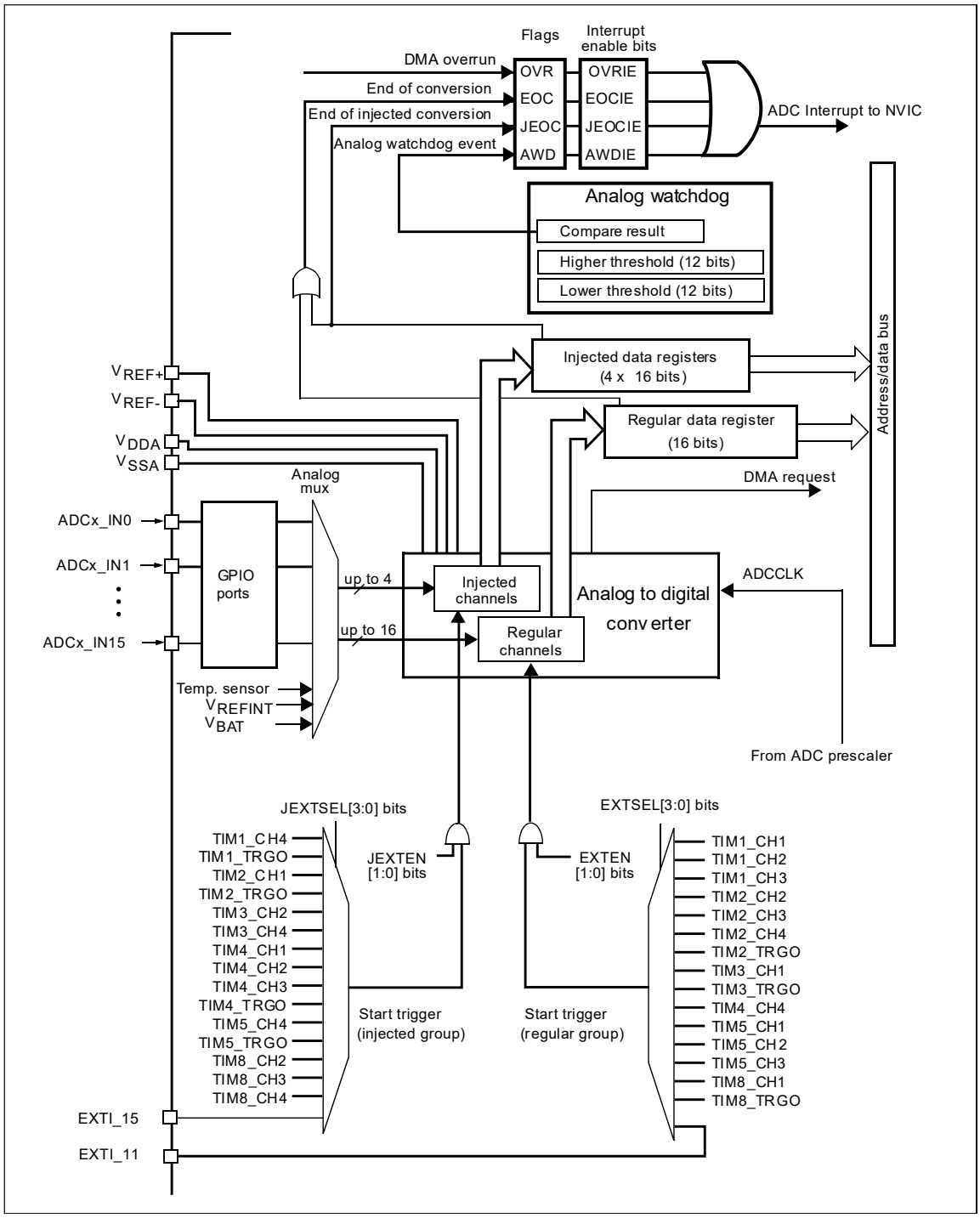



图 12-1 单个 ADC 框图

表 12-1 ADC 引脚

| 名称 | 信号类型 | 备注 |
|---|---|---|
| $V_{REF+}$ | 正模拟参考电压输入 | ADC 正参考电压，1.8V ≤ $V_{REF+}$ ≤ $V_{DDA}$ |
| $V_{DDA}$ | 模拟电源输入 | 模拟电源电压等于 $V_{DD}$，<br>全速运行时，2.4V ≤ $V_{DDA}$ ≤ $V_{DD}$ (3.6 V)<br>低速运行时，1.8V ≤ $V_{DDA}$ ≤ $V_{DD}$ (3.6 V) |
| $V_{REF-}$ | 负模拟参考电压输入 | ADC 低/负参考电压，$V_{REF-}$ = $V_{SSA}$ |

| 名称 | 信号类型 | 备注 |
| --- | --- | --- |
| $V_{SSA}$ | 模拟电源接地输入 | 模拟电源接地电压等于 VSS |
| ADCx_IN[15:0] | 模拟输入信号 | 16 个模拟输入通道 |

### 12.3.1. ADC 开关控制

可通过将 ADC_CR2 寄存器中的 ADON 位置 1 来为 ADC 供电。首次将 ADON 位置 1 时，会将 ADC 从掉电模式中唤醒。

SWSTART 或 JSWSTART 位置 1 时，启动 AD 转换。

可通过将 ADON 位清零来停止转换并使 ADC 进入掉电模式。在此模式下，ADC 几乎不耗电。

### 12.3.2. ADC 时钟

ADC 具有两个时钟方案：

用于模拟电路的时钟：ADCCLK，所有 ADC 共用

此时钟来自于经可编程预分频器分频的 APB2 时钟，该预分频器允许 ADC 在 $f_{PCLK2}$/2、/4、/6 或 /8 下工作。

用于数字接口的时钟（用于寄存器读/写访问）

此时钟等效于 APB2 时钟。可以通过 RCC APB2 外设时钟使能寄存器 (RCC_APB2ENR)分别为每个 ADC 使能/禁止数字接口时钟。

### 12.3.3. ADC 通道选择

有 16 条复用通道。可以将转换分为两组：规则转换和注入转换。每个组包含一个转换序列，该序列可按任意顺序在任意通道上完成。例如，可按以下顺序对序列进行转换：ADC_IN3、ADC_IN8、ADC_IN2、ADC_IN2、ADC_IN0、ADC_IN2、ADC_IN2、ADC_IN15。

- 一个规则转换组最多由 16 个转换构成。必须在 ADC_SQRx 寄存器中选择转换序列的规则通道及其顺序。规则转换组中的转换总数必须写入 ADC_SQR1 寄存器中的 L[3:0] 位。
- 一个注入转换组最多由 4 个转换构成。必须在 ADC_JSQR 寄存器中选择转换序列的注入通道及其顺序。注入转换组中的转换总数必须写入 ADC_JSQR 寄存器中的 L[1:0] 位。

如果在转换期间修改 ADC_SQRx 或 ADC_JSQR 寄存器，将复位当前转换并向 ADC 发送一个新的启动脉冲，以转换新选择的组。

**温度传感器、$V_{REFINT}$ 和 $V_{BAT_LI}$ 内部通道**

温度传感器内部连接到通道 ADC1_IN16，内部参考电压 $V_{REFINT}$ 连接到 ADC1_IN17，$V_{BAT_LI}$ 通道连接到通道 ADC2_IN16。

### 12.3.4. 单次转换模式

在单次转换模式下，ADC 执行一次转换。CONT 位为 0 时，可通过以下方式启动此模式：

- 将 ADC_CR2 寄存器中的 SWSTART 位置 1（仅适用于规则通道）；
- 将 JSWSTART 位置 1（适用于注入通道）；
- 外部触发（适用于规则通道或注入通道）。

完成所选通道的转换之后：

- 如果转换了规则通道：
  - 转换数据存储在 16 位 ADC_DR 寄存器中；
  - EOC（转换结束）标志置 1；
  - EOCIE 位置 1 时将产生中断；
- 如果转换了注入通道：
  - 转换数据存储在 16 位 ADC_JDR1 寄存器中；
  - JEOC（注入转换结束）标志置 1；
  - JEOCIE 位置 1 时将产生中断；
  - 然后，ADC 停止。

### 12.3.5. 连续转换模式

在连续转换模式下，ADC 结束一个转换后立即启动一个新的转换。CONT 位为 1 时，可通过外部触发或将 ADC_CR2 寄存器中的 SWSTRT 位置 1 来启动此模式（仅适用于规则通道）。每次转换之后：

- 如果转换了规则通道组：
  - 上次转换的数据存储在 16 位 ADC_DR 寄存器中；
  - EOC（转换结束）标志置 1；
  - EOCIE 位置 1 时将产生中断。

*注意：无法连续转换注入通道。连续模式下唯一的例外情况是，注入通道配置为在规则通道之后自动转换（使用 JAUTO 位），请参见自动注入一节。*

### 12.3.6. 时序图

如图所示，ADC 在开始精确转换之前需要一段稳定时间 $t_{STAB}$。ADC 开始转换并经过 15 个时钟周期后，EOC 标志置 1，转换结果存放在 16 位 ADC 数据寄存器中。

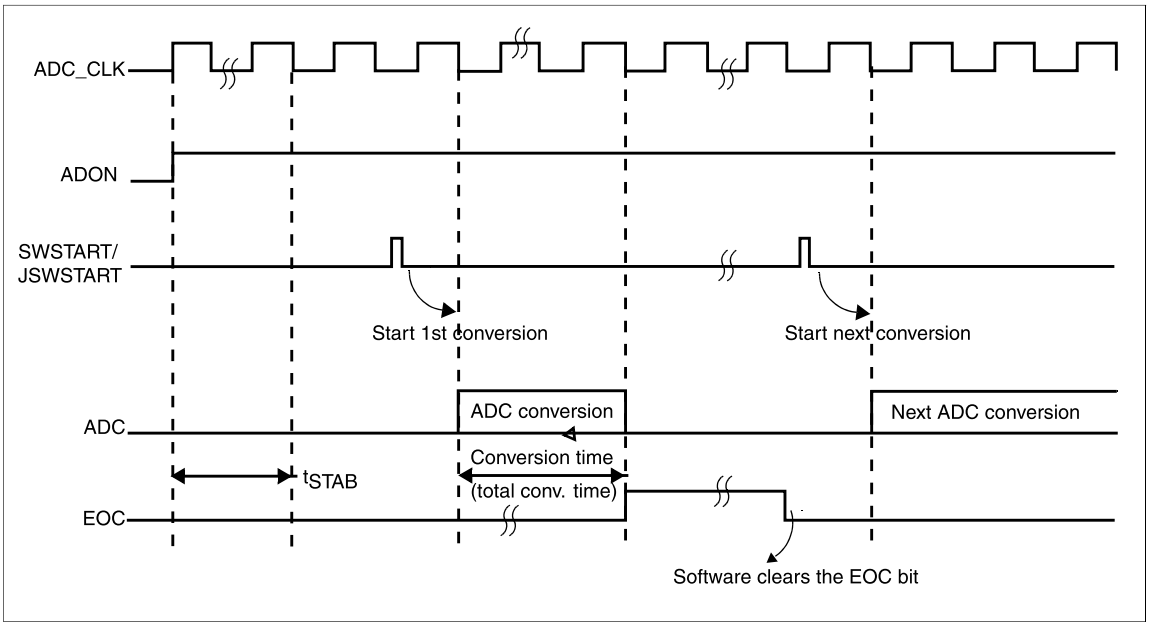



图 12-2 时序图

### 12.3.7. 模拟看门狗

如果 ADC 转换的模拟电压低于阈值下限或高于阈值上限，则 AWD 模拟看门狗状态位会置 1。这些阈值在 ADC_HTR 和 ADC_LTR 16 位寄存器的 12 个最低有效位中进行编程。可以使用 ADC_CR1 寄存器中的 AWDIE 位使能中断。

阈值与 ADC_CR2 寄存器中的 ALIGN 位的所选对齐方式无关。在对齐之前，会将模拟电压与阈值上限和下限进行比较。

下表介绍了应如何配置 ADC_CR1 寄存器才能在一个或多个通道上使能模拟看门狗。

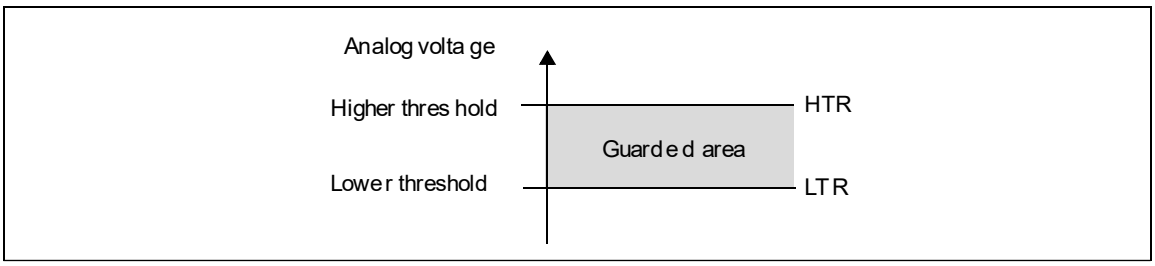



图 12-3 模拟看门狗的保护区域

表 12-2 模拟看门狗通道选择

| 模拟看门狗保护的通道 | ADC_CR1 寄存器控制位（ x = 无关） | | |
|---|---|---|---|
| | AWDSGL 位 | AWDEN 位 | JAWDEN 位 |
| 无 | x | 0 | 0 |
| 所有注入通道 | 0 | 0 | 1 |
| 所有规则通道 | 0 | 1 | 0 |
| 所有规则通道和注入通道 | 0 | 1 | 1 |
| 单个(1)注入通道 | 1 | 0 | 1 |
| 单个(1)规则通道 | 1 | 1 | 0 |
| 单个(1)规则通道或注入通道 | 1 | 1 | 1 |

*1. 通过 AWDCH[4:0] 位选择*

### 12.3.8. 扫描模式

此模式用于扫描一组模拟通道。

通过将 ADC_CR1 寄存器中的 SCAN 位置 1 来选择扫描模式。将此位置 1 后，ADC 会扫描在 ADC_SQRx 寄存器（对于规则通道）或 ADC_JSQR 寄存器（对于注入通道）中选择的所有通道。为组中的每个通道都执行一次转换。每次转换结束后，会自动转换该组中的下一个通道。如果将 CONT 位置 1，规则通道转换不会在组中最后一个所选通道处停止，而是再次从第一个所选通道继续转换。

如果将 DMA 位置 1，则在每次规则通道转换之后，均使用直接存储器访问 (DMA) 控制器将转换自规则通道组的数据（存储在 ADC_DR 寄存器中）传输到 SRAM。

在以下情况下， ADC_SR 寄存器中的 EOC 位置 1:

- 如果 EOCS 位清零，在每个规则组序列转换结束时。
- 如果 EOCS 位置 1，在每个规则通道转换结束时。

从注入通道转换的数据始终存储在 ADC_JDRx 寄存器中。

### 12.3.9. 注入通道管理

#### 触发注入

要使用触发注入，必须将 ADC_CR1 寄存器中的 JAUTO 位清零。

- 通过外部触发或将 ADC_CR2 寄存器中的 SWSTART 位置 1 来启动规则通道组转换。
- 如果在规则通道组转换期间出现外部注入触发或者 JSWSTART 位置 1，则当前的转换会复位，并且注入通道序列会切换为单次扫描模式。
- 然后，规则通道组的规则转换会从上次中断的规则转换处恢复。
- 如果在注入转换期间出现规则事件，注入转换不会中断，但在注入序列结束时会执行规则序列。下图显示了相应的时序图。

*注意： 使用触发注入时，必须确保触发事件之间的间隔长于注入序列。例如，如果序列长度为 30 个 ADC 时钟周期（即采样时间为 3 个时钟周期的两次转换），则触发事件的最小间隔不能小于 31 个 ADC 时钟周期。*

#### 自动注入

如果将 JAUTO 位置 1，则注入组中的通道会在规则组通道之后自动转换。这可用于转换最多由 20 个转换构成的序列，这些转换在 ADC_SQRx 和 ADC_JSQR 寄存器中编程。

在此模式下，必须禁止注入通道上的外部触发。

如果 CONT 位和 JAUTO 位均已置 1，则在转换规则通道之后会继续转换注入通道。

*注意： 不能同时使用自动注入和不连续采样模式。*

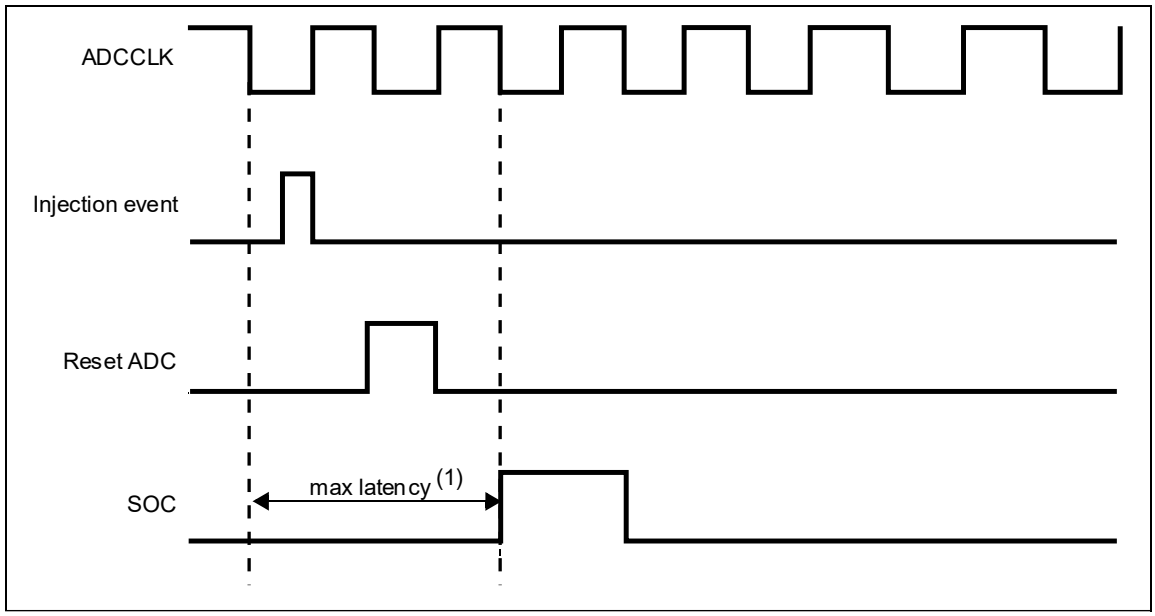



图 12-4 注入转换延迟

### 12.3.10. 不连续采样模式

#### 规则组

可将 ADC_CR1 寄存器中的 DISCEN 位置 1 来使能此模式。该模式可用于转换含有 n (n ≤ 8)个转换的短序列，该短序列是在 ADC_SQRx 寄存器中选择的转换序列的一部分。可通过写入 ADC_CR1 寄存器中的 DISCNUM[2:0] 位来指定 n 的值。

出现外部触发时，将启动在 ADC_SQRx 寄存器中选择的接下来 n 个转换，直到序列中的所有转换均完成为止。通过 ADC_SQR1 寄存器中的 L[3:0] 位定义总序列长度。

示例：

- n = 3，要转换的通道 = 0、1、2、3、6、7、9、10
- 第 1 次触发：转换序列 0、1、2
- 第 2 次触发：转换序列 3、6、7
- 第 3 次触发：转换序列 9、10 并生成 EOC 事件
- 第 4 次触发：转换序列 0、1、2

*注意： 在不连续采样模式下转换规则组时，不会出现翻转。转换完所有子组后，下一个触发信号将启动第一个子组的转换。在上述示例中，第 4 次触发重新转换了第 1 个子组中的通道 0、 1 和 2。*

#### 注入组

可将 ADC_CR1 寄存器中的 JDISCEN 位置 1 来使能此模式。在出现外部触发事件之后，可使用该模式逐通道转换在 ADC_JSQR 寄存器中选择的序列。

出现外部触发时，将启动在 ADC_JSQR 寄存器中选择的下一个通道转换，直到序列中的所有转换均完成为止。通过 ADC_JSQR 寄存器中的 JL[1:0] 位定义总序列长度。

示例：

- n = 1，要转换的通道 = 1、2、3
- 第 1 次触发：转换通道 1
- 第 2 次触发：转换通道 2
- 第 3 次触发：转换通道 3 并生成 EOC 和 JEOC 事件
- 第 4 次触发：通道 1

*注意：转换完所有注入通道后，下一个触发信号将启动第一个注入通道的转换。在上述示例中，第 4 次触发重新转换了第 1 个注入通道。*

*不能同时使用自动注入和不连续采样模式。*

*不得同时为规则组和注入组设置不连续采样模式。只能针对一个组使能不连续采样模式。*

## 12.4. 数据对齐

ADC_CR2 寄存器中的 ALIGN 位用于选择转换后存储的数据的对齐方式。可选择左对齐和右对齐两种方式。

注入通道组的转换数据将减去 ADC_JOFRx 寄存器中写入的用户自定义偏移量，因此结果可以是一个负值。SEXT 位表示扩展的符号值。

对于规则组中的通道，不会减去任何偏移量，因此只有十二个位有效。

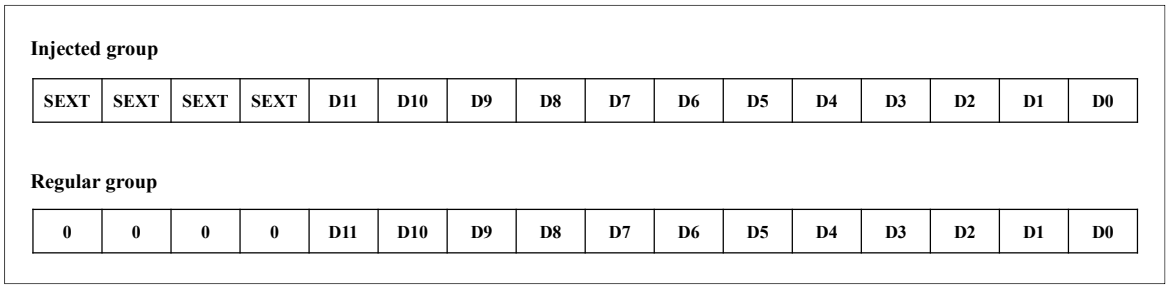



图 12-5 12 位数据的右对齐

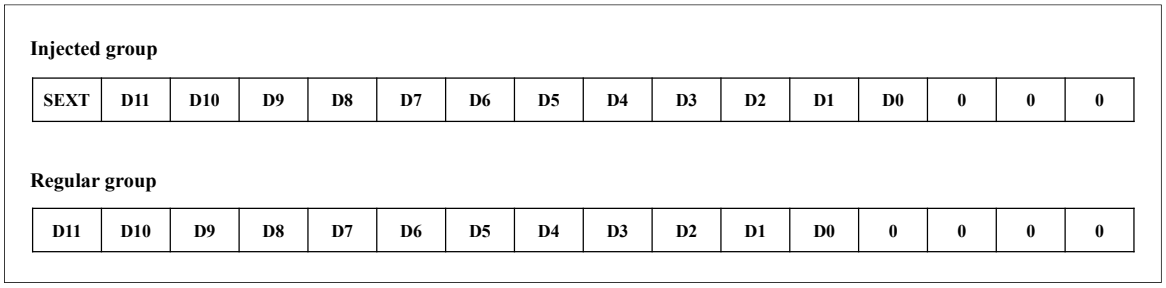



图 12-6 12 位数据的左对齐

特例：采用左对齐时，数据基于半字进行对齐，除了分辨率设置为 6 位时。分辨率设置为 6 位时，数据基于字节进行对齐，如下图所示。

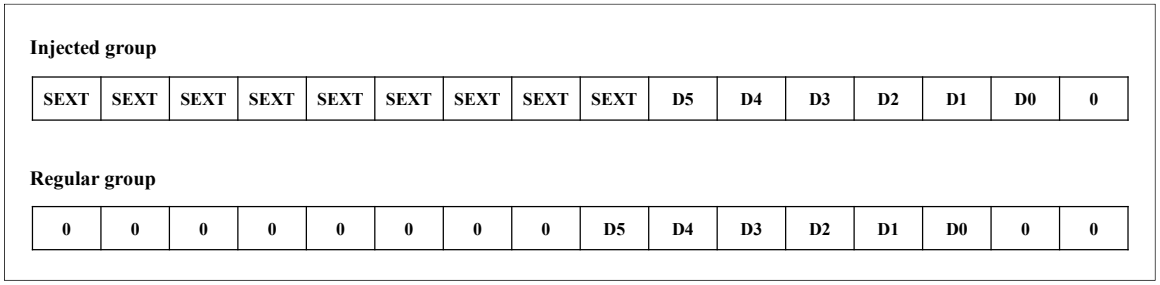



图 12-7 6 位数据的左对齐

## 12.5. 可独立设置各通道采样时间

ADC 会在数个 ADCCLK 周期内对输入电压进行采样，可使用 ADC_SMPR1 和 ADC_SMPR2 寄存器中的 SMP[2:0] 位修改周期数。每个通道均可以使用不同的采样时间进行采样。

总转换时间的计算公式如下：

$$T_{conv} = \text{采样时间} + 12 \text{ 个周期}$$

示例：

ADCCLK = 30 MHz 且采样时间 = 3 个周期时：

$$T_{conv} = 3 + 12 = 15 \text{ 个周期} = 0.5\ \mu s\text{（APB2 为 60 MHz 时）}$$

## 12.6. 外部触发转换和触发极性

可以通过外部事件（例如，定时器捕获、EXTI 中断线）触发转换。如果 EXTEN[1:0] 控制位（对于行规转换）或 JEXTEN[1:0] 位（对于注入转换）不等于“0b00”，则外部事件能够以所选极性触发转换。下表提供了 EXTEN[1:0] 和 JEXTEN[1:0] 值与触发极性之间的对应关系。

表 12-3 配置触发极性

| 源 | EXTEN[1:0]/JEXTEN[1:0] |
|---|---|
| 禁止触发检测 | 0 |
| 在上升沿时检测 | 1 |
| 在下降沿时检测 | 10 |
| 在上升沿和下降沿均检测 | 11 |

*注意： 可以实时更改外部触发的极性。*

EXTSEL[3:0] 和 JEXTSEL[3:0] 控制位用于从 16 个可能事件中选择可触发规则组转换和注入组转换的事件。

下表给出了可用于规则转换的外部触发。

表 12-4 规则通道的外部触发

| 源 | 类型 | EXTSEL[3:0] |
|---|---|---|
| TIM1_CH1 事件 | 片上定时器的内部信号 | 0000 |
| TIM1_CH2 事件 | | 0001 |
| TIM1_CH3 事件 | | 0010 |
| TIM2_CH2 事件 | | 0011 |
| TIM2_CH3 事件 | | 0100 |
| TIM2_CH4 事件 | | 0101 |
| TIM2_TRGO 事件 | | 0110 |
| TIM3_CH1 事件 | | 0111 |
| TIM3_TRGO 事件 | | 1000 |
| TIM4_CH4 事件 | | 1001 |
| TIM5_CH1 事件 | | 1010 |
| TIM5_CH2 事件 | | 1011 |
| TIM5_CH3 事件 | | 1100 |
| TIM8_CH1 事件 | | 1101 |
| TIM8_TRGO 事件 | | 1110 |
| EXTI 线 11 | 外部引脚 | 1111 |

下表给出了可用于注入转换的外部触发。

表 12-5 注入通道的外部触发

| 源 | 连接类型 | JEXTSEL[3:0] |
|---|---|---|
| TIM1_CH4 事件 | 片上定时器的内部信号 | 0000 |
| TIM1_TRGO 事件 | | 0001 |
| TIM2_CH1 事件 | | 0010 |
| TIM2_TRGO 事件 | | 0011 |
| TIM3_CH2 事件 | | 0100 |
| TIM3_CH4 事件 | | 0101 |
| TIM4_CH1 事件 | | 0110 |
| TIM4_CH2 事件 | | 0111 |

| 源 | 连接类型 | JEXTSEL[3:0] |
|---|---|---|
| TIM4_CH3 事件 | | 1000 |
| TIM4_TRGO 事件 | | 1001 |
| TIM5_CH4 事件 | | 1010 |
| TIM5_TRGO 事件 | | 1011 |
| TIM8_CH2 事件 | | 1100 |
| TIM8_CH3 事件 | | 1101 |
| TIM8_CH4 事件 | | 1110 |
| EXTI 线 15 | 外部引脚 | 1111 |

可通过将 ADC_CR2 寄存器中的 SWSTART（对于规则转换）或 JSWSTART（对于注入转换）位置 1 来产生软件源触发事件。

可通过注入触发中断规则组转换。

*注意： 可以实时更改触发选择。不过，当更改触发选择时，会在 1 个 APB 时钟周期的时间范围内禁止触发检测。这是为了避免在转换期间出现意外检测。*

## 12.7. 快速转换模式

可通过降低 ADC 分辨率来执行快速转换。RES 位用于选择数据寄存器中可用的位数。每种分辨率的最小转换时间如下：

- 12 位： 3 + 12 = 15 ADCCLK 周期
- 10 位： 3 + 10 = 13 ADCCLK 周期
- 8 位： 3 + 8 = 11 ADCCLK 周期
- 6 位： 3 + 6 = 9 ADCCLK 周期

## 12.8. 数据管理

### 12.8.1. 使用 DMA

由于规则通道组只有一个数据寄存器，因此，对于多个规则通道的转换，使用 DMA 非常有帮助。这样可以避免丢失在下一次写入之前还未被读出的 ADC_DR 寄存器中的数据。

在使能 DMA 模式的情况下（ADC_CR2 寄存器中的 DMA 位置 1），每完成规则通道组中的一个通道转换后，都会生成一个 DMA 请求。这样便可将转换的数据从 ADC_DR 寄存器传输到用软件选择的目标位置。

尽管如此，如果数据丢失（溢出），则会将 ADC_SR 寄存器中的 OVR 位置 1 并生成一个中断（如果 OVRIE 使能位已置 1）。随后会禁止 DMA 传输并且不再接受 DMA 请求。在这种情况下，如果生成 DMA 请求，则会中止正在进行的规则转换并忽略之后的规则触发。随后需要将所使用的 DMA 流中的 OVR 标志和 DMAEN 位清零，并重新初始化 DMA 和 ADC，以将需要的转换通道数据传输到正确的存储器单元。只有这样，才能恢复转换并再次使能数据传输。注入通道转换不会受到溢出错误的影响。

在 DMA 模式下，当 OVR = 1 时，传送完最后一个有效数据后会阻止 DMA 请求，这意味着传输到 RAM 的所有数据均被视为有效。

在最后一次 DMA 传输（DMA 控制器的 DMA_SxRTR 寄存器中配置的传输次数）结束时：

- 如果将 ADC_CR2 寄存器中的 DDS 位清零，则不会向 DMA 控制器发出新的 DMA 请求（这可避免产生溢出错误）。不过，硬件不会将 DMA 位清零。必须将该位写入 0 然后写入 1 才能启动新的传输。
- 如果将 DDS 位置 1，则可继续生成请求。从而允许在双缓冲区循环模式下配置 DMA。要在使用

DMA 时将 ADC 从 OVR 状态中恢复，请按以下步骤操作：

- 重新初始化 DMA（调整目标地址和 NDTR 计数器）；
- 将 ADC_SR 寄存器中的 ADC OVR 位清零；
- 触发 ADC 以开始转换。

### 12.8.2. 在不使用 DMA 的情况下管理转换序列

如果转换过程足够慢，则可使用软件来处理转换序列。在这种情况下，必须将 ADC_CR2 寄存器中的 EOCS 位置 1，才能使 EOC 状态位在每次转换结束时置 1，而不仅是在序列结束时置 1。当 EOCS = 1 时，会自动使能溢出检测。因此，每当转换结束时，EOC 都会置 1，并且可以读取 ADC_DR 寄存器。溢出管理与使用 DMA 时的管理相同。

要在 EOCS 位置 1 时将 ADC 从 OVR 状态中恢复，请按以下步骤操作：

- 将 ADC_SR 寄存器中的 ADC OVR 位清零；
- 触发 ADC 以开始转换。

### 12.8.3. 在不使用 DMA 和溢出检测的情况下进行转换

ADC 在转换一个或多个通道时不是每次都读取数据的情况下，这可能会很有用（例如，存在模拟看门狗时）。为此，必须禁止 DMA (DMA = 0) 并且仅在序列结束 (EOCS = 0) 时才将 EOC 位置 1。在此配置中，溢出检测已禁止。

## 12.9. 多重 ADC 模式

可使用双重（具有两个 ADC）和三重（具有三个 ADC）ADC 模式（参见下图）。

在多重 ADC 模式下，通过 ADC1 主器件到 ADC2 和 ADC3 从器件的交替触发或同时触发来启动转换，具体取决于 ADC_CCR 寄存器中的 MULTI[4:0] 位所选的模式。

*注意： 在多重 ADC 模式下，配置外部事件触发转换时，应用必须设置为仅主器件触发而禁止从器件触发，以防止出现意外触发而启动不需要的从转换。*

可实现以下四种模式：

- 注入同时模式；
- 规则同时模式；
- 交替模式；
- 交替触发模式。

也可按以下方式组合使用上述模式：

- 注入同时模式 + 规则同时模式。
- 规则同时模式 + 交替触发模式。

*注意： 在多重 ADC 模式下，可在多模式数据寄存器 (ADC_CDR) 中读取转换的数据。可在多模式状态寄存器 (ADC_CSR) 中读取状态位。*

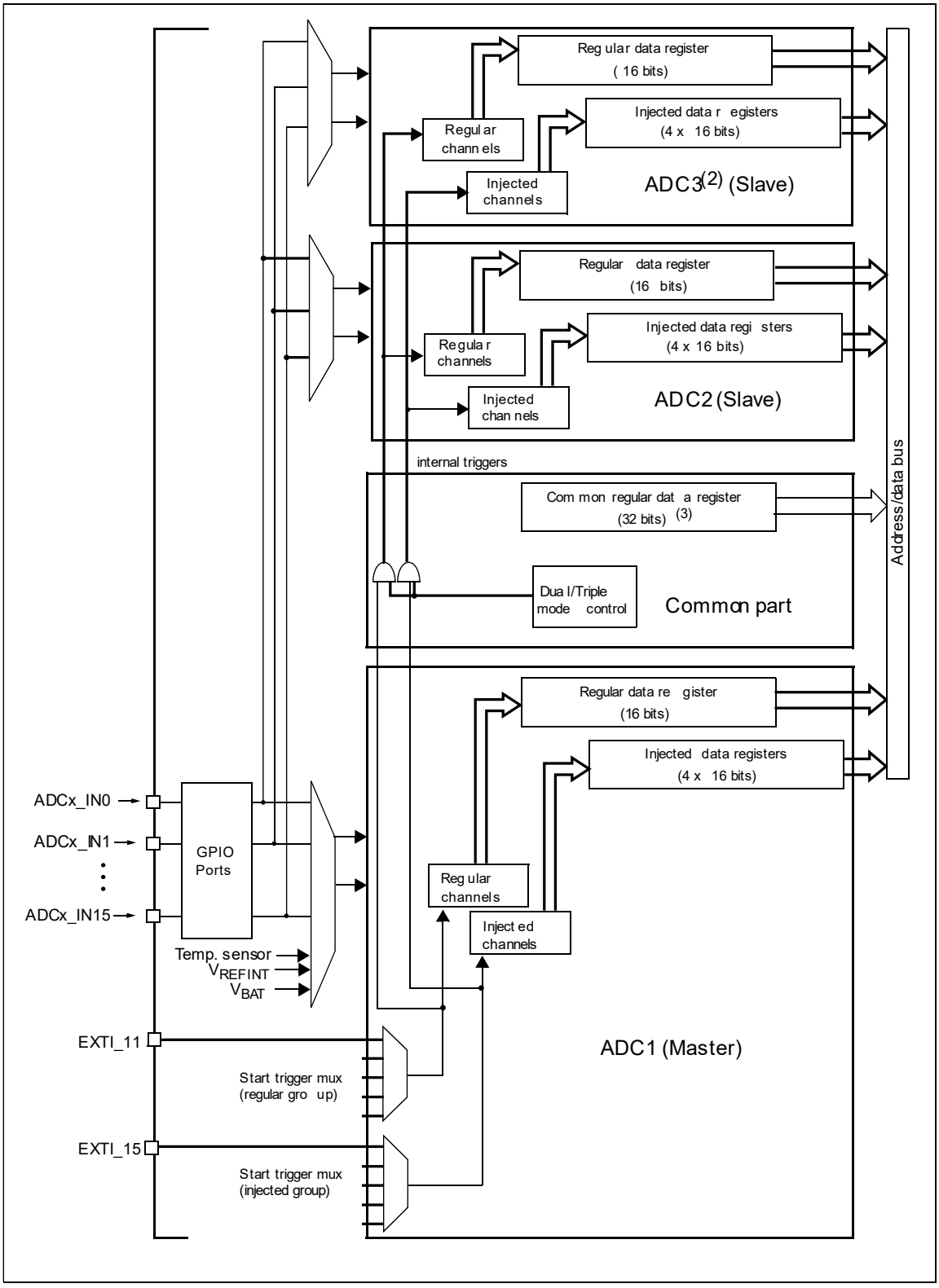



图 12-8 多重 ADC 框图(1)

1. 尽管 ADC2 和 ADC3 上存在外部触发，但它们并未显示在此图中。
2. 在双重 ADC 模式下，不存在 ADC3 从器件部分。
3. 在三重 ADC 模式下， ADC 通用数据寄存器 (ADC_CDR) 包含 ADC1、ADC2 和 ADC3 的规则转

换数据。按照所选的存储顺序使用全部 32 个寄存器位。

在双重 ADC 模式下， ADC 通用数据寄存器 (ADC_CDR) 包含 ADC1 和 ADC2 的规则转换数据。使用全部 32 个寄存器位。

- **多重 ADC 模式下的 DMA 请求：**

  在多重 ADC 模式下，可将 DMA 配置为使用三种不同的模式来传输转换的数据。在所有情况下，要使用的 DMA 流均连接到 ADC：

  - **DMA 模式 1**：每发出一个 DMA 请求（一个数据项可用），就会传输一个表示 ADC 转换的数据项的半字。

    在双重 ADC 模式下，发出第一个请求时传输 ADC1 的数据，发出第二个请求时传输 ADC2 的数据，依次类推。

    在三重 ADC 模式下，发出第一个请求时传输 ADC1 的数据，发出第二个请求时传输 ADC2 的数据，发出第三个请求时传输 ADC3 的数据；重复此序列。因此 DMA 首先传输 ADC1 的数据，随后传输 ADC2 的数据，再传输 ADC3 的数据，依次类推。

    DMA 模式 1 用于三重规则同时模式。

    示例：

    三重规则同时模式：生成 3 个连续的 DMA 请求（每个请求对应一个转换数据项）

    第 1 个请求： ADC_CDR[31:0] = ADC1_DR[15:0]

    第 2 个请求： ADC_CDR[31:0] = ADC2_DR[15:0]

    第 3 个请求： ADC_CDR[31:0] = ADC3_DR[15:0]

    第 4 个请求： ADC_CDR[31:0] = ADC1_DR[15:0]

  - **DMA 模式 2**：每发送一个 DMA 请求（两个数据项可用），就会以字的形式传输表示两个 ADC 转换数据项的两个半字。

    在双重 ADC 模式下，发出第一个请求时会传输 ADC2 和 ADC1 的数据（ADC2 数据占用高位半字， ADC1 数据占用低位半字），依此类推。

    在三重 ADC 模式下，将生成三个 DMA 请求：发出第一个请求时，会传输 ADC2 和 ADC1 的数据（ADC2 数据占用高位半字， ADC1 数据占用低位半字）。发出第二个请求时，会传输 ADC1 和 ADC3 的数据（ADC1 数据占用高位半字， ADC3 数据占用低位半字）。发出第三个请求时，会传输 ADC3 和 ADC2 的数据（ADC3 数据占用高位半字， ADC2 数据占用低位半字），依此类推。

    DMA 模式 2 用于交替模式和规则同时模式（仅适用于双重 ADC 模式）。

    示例：

    a) 双重交替模式：每当有 2 个数据项可用时，就会生成一个 DMA 请求：

       第 1 个请求： ADC_CDR[31:0] = ADC2_DR[15:0] | ADC1_DR[15:0]

       第 2 个请求： ADC_CDR[31:0] = ADC2_DR[15:0] | ADC1_DR[15:0]

    b) 三重交替模式：每当有 2 个数据项可用时，就会生成一个 DMA 请求

       第 1 个请求： ADC_CDR[31:0] = ADC2_DR[15:0] | ADC1_DR[15:0]

       第 2 个请求： ADC_CDR[31:0] = ADC1_DR[15:0] | ADC3_DR[15:0]

       第 3 个请求： ADC_CDR[31:0] = ADC3_DR[15:0] | ADC2_DR[15:0]

       第 4 个请求： ADC_CDR[31:0] = ADC2_DR[15:0] | ADC1_DR[15:0]

  - **DMA 模式 3**：此模式与 DMA 模式 2 相似。唯一的区别是：在这种模式下，每发送一个 DMA 请求（两个数据项可用），就会以半字的形式传输表示两个 ADC 转换数据项的两个字节。此模式下的数据传输顺序与 DMA 模式 2 相似。

DMA 模式 3 用于分辨率为 6 位和 8 位时的交替模式。

示例：

a) 双重交替模式：每当有 2 个数据项可用时，就会生成一个 DMA 请求

第 1 个请求： ADC_CDR[15:0] = ADC2_DR[7:0] | ADC1_DR[7:0]

第 2 个请求： ADC_CDR[15:0] = ADC2_DR[7:0] | ADC1_DR[7:0]

b) 三重交替模式：每当有 2 个数据项可用时，就会生成一个 DMA 请求

第 1 个请求： ADC_CDR[15:0] = ADC2_DR[7:0] | ADC1_DR7:0]

第 2 个请求： ADC_CDR[15:0] = ADC1_DR[7:0] | ADC3_DR[15:0]

第 3 个请求： ADC_CDR[15:0] = ADC3_DR[7:0] | ADC2_DR[7:0]

第 4 个请求： ADC_CDR[15:0] = ADC2_DR[7:0] | ADC1_DR7:0]

**溢出检测：** 如果在其中一个相关的 ADC（双重和三重模式下的 ADC1 和 ADC2，仅有三重模式时的 ADC3）上检测到溢出，则不再发出 DMA 请求，以确保传输到 RAM 的所有数据都有效。对于与某个 ADC 对应的 EOC 位，有时可能会因为此 ADC 的数据寄存器包含有效数据而保持置 1。

### 12.9.1. 注入同时模式

此模式可转换注入通道组。外部触发源来自 ADC1 的注入组多路复用器（通过 ADC1_CR2 寄存器中的 JEXTSEL[3:0] 位进行选择）。同时触发可用于 ADC2 和 ADC3。

*注意：不要在两个/三个 ADC 上转换同一通道（转换同一通道时，不允许两个/三个 ADC 采样时间重叠）。*

*在同时模式下，必须使用同一长度来转换序列，或必须确保触发之间的间隔长于 2 个序列（双重 ADC 模式）/3 个序列（三重 ADC 模式）中的较长时间。否则，当序列较长的 ADC 完成上一次转换时，序列较短的 ADC 可能重新开始转换。*

*规则转换可在一个或所有 ADC 上执行。这种情况下，它们彼此之间都是独立的，而且会在出现注入事件时中断。它们会在注入转换组结束时恢复转换。*

#### 双重 ADC 模式

在 ADC1 或 ADC2 转换事件结束时：

- 转换的数据会存储在各个 ADC 接口的 ADC_JDRx 寄存器中。
- 当 ADC1/ADC2 的注入通道全部完成转换后，会生成一个 JEOC 中断（如果已在两个 ADC 接口中的一个接口上使能）。

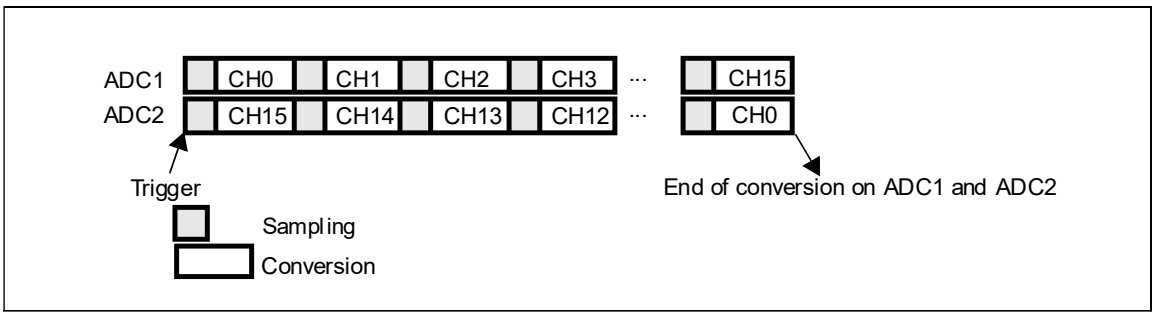



图 12-9 4 通道的注入同时模式：双重 ADC 模式

### 三重 ADC 模式

在 ADC1、ADC2 或 ADC3 转换事件结束时：

- 转换的数据会存储在各个 ADC 接口的 ADC_JDRx 寄存器中。
- 当 ADC1/ADC2/ADC3 的注入通道全部完成转换后，会生成一个 JEOC 中断（如果已在三个 ADC 接口中的一个接口上使能）。

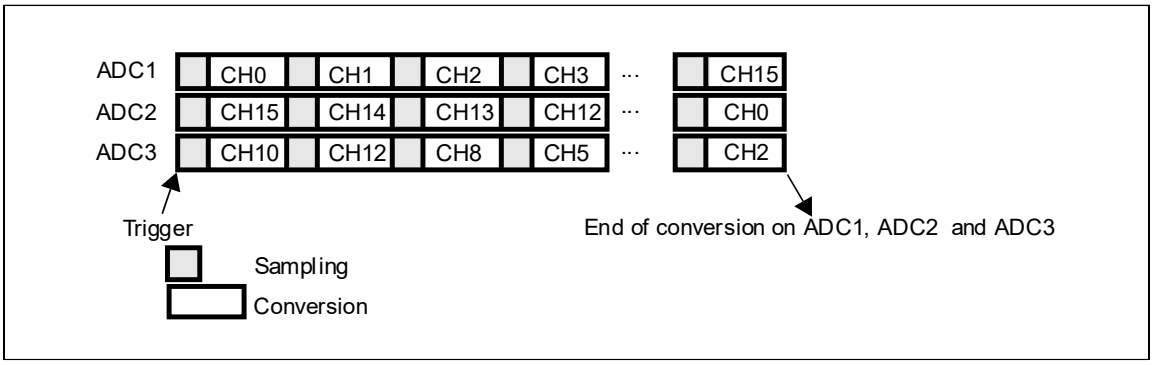



图 12-10 4 通道的注入同时模式：三重 ADC 模式

## 12.9.2. 规则同时模式

此模式可用于规则通道组。外部触发源来自 ADC1 的规则组多路复用器（通过 ADC1_CR2 寄存器中的 EXTSEL[3:0] 位进行选择）。同时触发可用于 ADC2 和 ADC3。

*注意：不要在两个/三个 ADC 上转换同一通道（转换同一通道时，不允许两个/三个 ADC 采样时间重叠）。*

*在规则同时模式下，必须使用同一长度来转换序列，或必须确保触发之间的间隔长于 2 个序列（双重 ADC 模式）/3 个序列（三重 ADC 模式）中的较长转换时间。否则，当序列较长的 ADC 完成上一次转换时，序列较短的 ADC 可能重新开始转换。*

*必须禁止注入转换*

### 双重 ADC 模式

在 ADC1 或 ADC2 转换事件结束时：

- 会生成一个 32 位 DMA 传输请求（如果 ADC_CCR 寄存器中的 DMA[1:0] 位等于 0b10）。此请求会将存储在 ADC_CDR 32 位寄存器高位半字中的 ADC2 转换数据传输到 SRAM，然后将存储在 ADC_CCR 低位半字中的 ADC1 转换数据传输到 SRAM。
- 当 ADC1/ADC2 的规则通道全部完成转换后，会生成一个 EOC 中断（如果已在两个 ADC 接口中的一个接口上使能）。

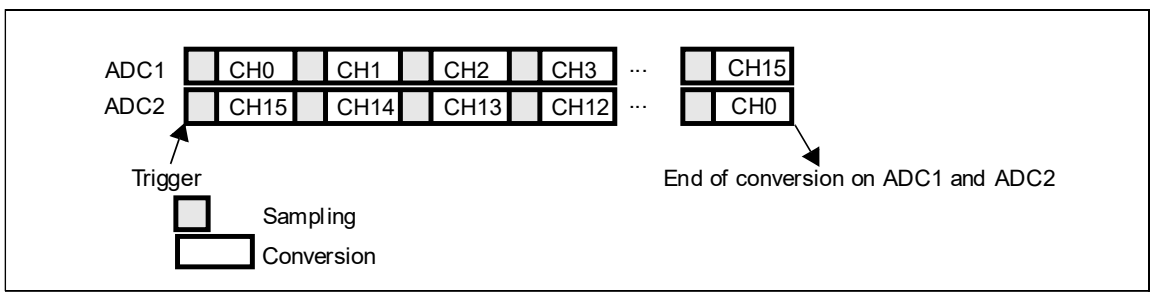



图 12-11 16 通道的规则同时模式：双重 ADC 模式

### 三重 ADC 模式

在 ADC1、 ADC2 或 ADC3 转换事件结束时:

- 会生成三个 32 位 DMA 传输请求（如果 ADC_CCR 寄存器中的 DMA[1:0] 位等于 0b01）。之后会发生三次从 ADC_CDR 32 位寄存器到 SRAM 的传输: 首先传输 ADC1 转换数据, 然后是 ADC2 转换数据, 最后是 ADC3 转换数据。每次出现新的三阶段转换时都会重复这一过程。
- 当 ADC1/ADC2/ADC3 的规则通道全部完成转换后, 会生成一个 EOC 中断（如果已在三个 ADC 接口中的一个接口上使能）。

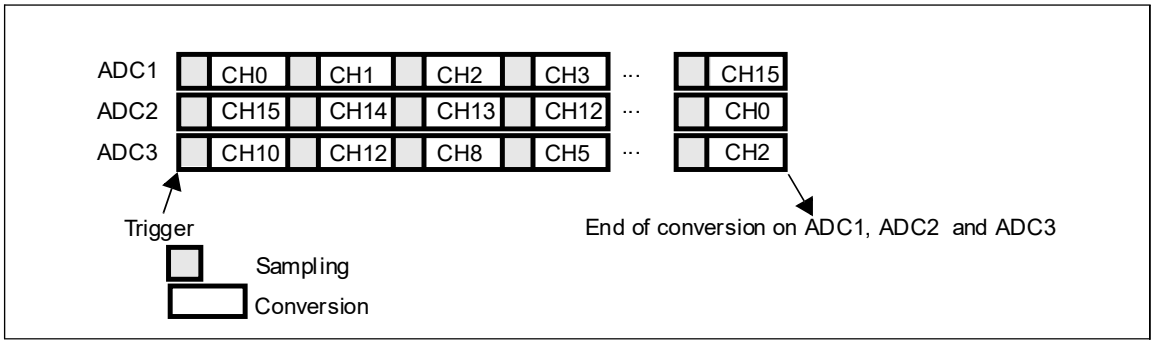



图 12-12 16 通道的规则同时模式：三重 ADC 模式

## 12.9.3. 交替模式

此模式只能用于规则组（通常为一个通道）。外部触发源来自 ADC1 的规则通道多路复用器。

### 双重 ADC 模式

出现外部触发之后:

- ADC1 立即启动;
- 经过几个 ADC 时钟周期延迟后 ADC2 启动。

交替模式下 2 个转换之间的最小延迟通过 ADC_CCR 寄存器中的 DELAY 位进行配置。但是，如果某个 ADC 的互补 ADC 仍在对其输入进行采样，则该 ADC 无法启动转换（在给定时间内，只有一个 ADC 能够对输入信号采样）。在这种情况下，延迟时间为采样时间 + 2 个 ADC 时钟周期。例如，如果两个 ADC 的 DELAY = 5 个时钟周期，且采样时间为 15 个时钟周期，则 ADC1 和 ADC2 之间的转换延迟为 17 个时钟周期。

如果 ADC1 和 ADC2 上的 CONT 位均置 1，则这两个 ADC 所选规则通道会连续进行转换。

*注意： 如果转换序列中断（例如 DMA 传输结束时），则必须首先通过在独立模式下进行配置来将多重 ADC 定序器复位（位 DUAL[4:0] = 00000），然后才可以对交替模式进行编程。*

*ADC2 生成一个 EOC 中断之后（如果已通过 EOCIE 位使能），会生成一个 32 位 DMA 传输请求（如果 ADC_CCR 寄存器中的 DMA[1:0] 位等于 0b10）。此请求首先会将存储在 ADC_CDR 32 位寄存器高位半字中的 ADC2 转换数据传输到 SRAM，然后将存储在寄存器低位半字中的 ADC1 转换数据传输到 SRAM。*

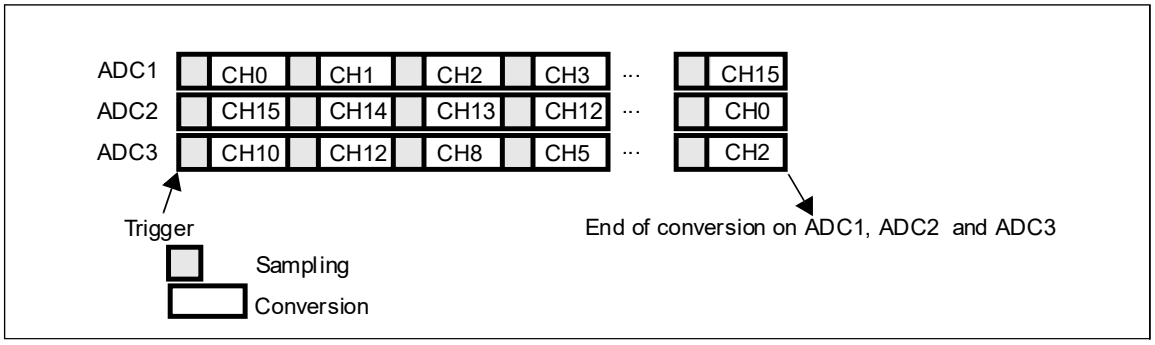



图 12-13 连续转换模式下 1 通道的交替模式：双重 ADC 模式

### 三重 ADC 模式

出现外部触发之后：

- ADC1 立即启动；
- 经过几个 ADC 时钟周期延迟后 ADC2 启动；
- 在 ADC2 转换经过几个 ADC 时钟周期的延迟后 ADC3 启动‘’

交替模式下 2 个转换之间的最小延迟通过 ADC_CCR 寄存器中的 DELAY 位进行配置。但是，如果某个 ADC 的互补 ADC 仍在对其输入进行采样，则该 ADC 无法启动转换（在给定时间内，只有一个 ADC 能够对输入信号采样）。在这种情况下，延迟时间为采样时间 +2 个 ADC 时钟周期。例如，如果这三个 ADC 的 DELAY = 5 个时钟周期，且采样时间为 15 个时钟周期，则 ADC1、ADC2 和 ADC3 之间的转换延迟为 17 个时钟周期。

如果 ADC1、ADC2 和 ADC3 上的 CONT 位均置 1，则这些 ADC 所选规则通道会连续进行转换。

*注意： 如果转换序列中断（例如 DMA 传输结束时），则必须首先通过在独立模式下进行配置来将多重 ADC 定序器复位（位 DUAL[4:0] = 00000），然后才可以对交替模式进行编程。*

*在此模式下，每当出现 2 个可用数据项时，就会生成一个 DMA 传输请求（如果 ADC_CCR 寄存器中的 DMA[1:0] 位等于 0b10）。此请求首先会将存储在 ADC_CDR 32 位寄存器低位半字中的第一批转换数据传输到 SRAM，然后将存储在 ADC_CDR 高位半字中的第二批转换数据传输到 SRAM。具体顺序如下：*

*第 1 个请求： ADC_CDR[31:0] = ADC2_DR[15:0] | ADC1_DR[15:0]*

*第 2 个请求： ADC_CDR[31:0] = ADC1_DR[15:0] | ADC3_DR[15:0]*

*第 3 个请求： ADC_CDR[31:0] = ADC3_DR[15:0] | ADC2_DR[15:0]*

*第 4 个请求： ADC_CDR[31:0] = ADC2_DR[15:0] | ADC1_DR[15:0], ...*

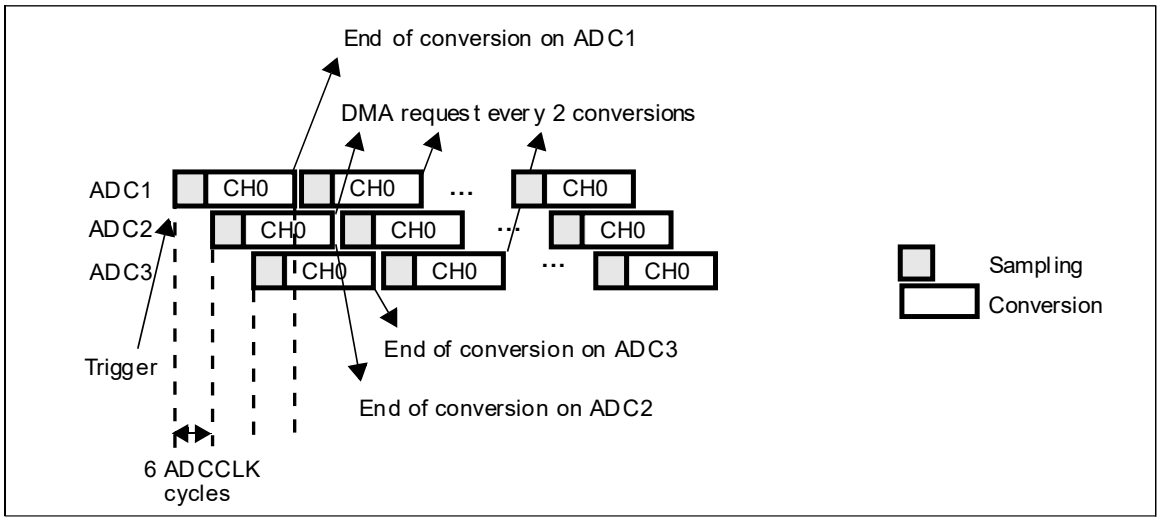



图 12-14 连续转换模式下 1 通道的交替模式：三重 ADC 模式

### 12.9.4. 交替触发模式

此模式只能用于注入组。外部触发源来自 ADC1 的注入组多路复用器。

*注意：规则转换可在一个或所有 ADC 上使能。在这种情况下，规则转换彼此之间是独立的。当 ADC 必须执行注入转换时，会中断规则转换。它会在注入转换完成后恢复。*

*如果转换序列中断（例如 DMA 传输结束时），则必须首先通过在独立模式下进行配置来将多重 ADC 定序器复位（位 DUAL[4:0] = 00000），然后才可以对交替模式进行编程。*

*两次触发事件之间的时间间隔必须大于或等于 1 个 ADC 时钟周期。同一个 ADC 上可启动转换的两个触发事件之间的最小时间间隔与单个 ADC 模式下相同。*

#### 双重 ADC 模式

- 发生第一次触发时，将转换 ADC1 中注入组的所有通道；
- 发生第二次触发时，将转换 ADC2 中注入组的所有通道；
- 以此类推。

当组中的所有注入 ADC1 通道都转换完成后，会生成一个 JEOC 中断（如果已使能）。

当组中的所有注入 ADC2 通道都转换完成后，会生成一个 JEOC 中断（如果已使能）。

如果在组中的所有注入通道都完成转换后出现另一个外部触发，则可通过转换组中的注入 ADC1 通道来重新启动交替触发过程。

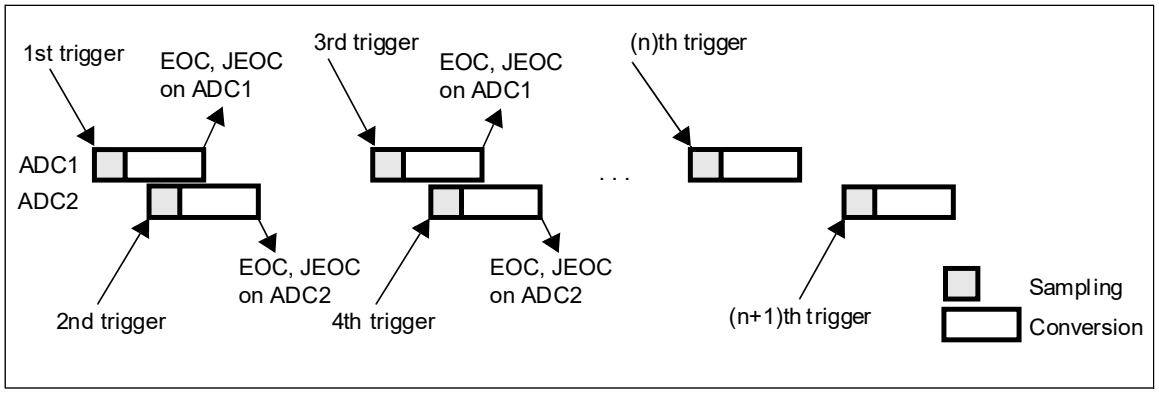



图 12-15 交替触发：各个 ADC 的注入组

如果使能 ADC1 和 ADC2 的注入不连续采样模式:

- 发生第一次触发时，将转换第一个注入 ADC1 通道;
- 发生第二次触发时，将转换第一个注入 ADC2 通道;
- 以此类推。

当组中的所有注入 ADC1 通道都转换完成后，会生成一个 JEOC 中断（如果已使能）。

当组中的所有注入 ADC2 通道都转换完成后，会生成一个 JEOC 中断（如果已使能）。

如果注入组中的所有通道都完成转换后出现另一个外部触发，则会重新启动交替触发过程。

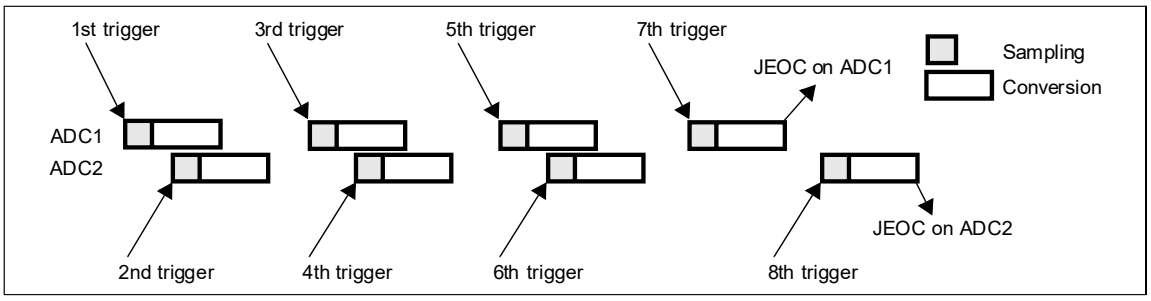



图 12-16 交替触发：不连续采样模式下的 4 个注入通道（各个 ADC）

### 三重 ADC 模式

- 发生第一次触发时，将转换 ADC1 中注入组的所有通道。
- 发生第二次触发时，将转换 ADC2 中注入组的所有通道。
- 发生第三次触发时，将转换 ADC3 中注入组的所有通道。
- 以此类推。

当组中的所有注入 ADC1 通道都转换完成后，会生成一个 JEOC 中断（如果已使能）。

当组中的所有注入 ADC2 通道都转换完成后，会生成一个 JEOC 中断（如果已使能）。

当组中的所有注入 ADC3 通道都转换完成后，会生成一个 JEOC 中断（如果已使能）。

如果在组中的所有注入通道都完成转换后出现另一个外部触发，则可通过转换组中的注入 ADC1 通道来重新启动交替触发过程。

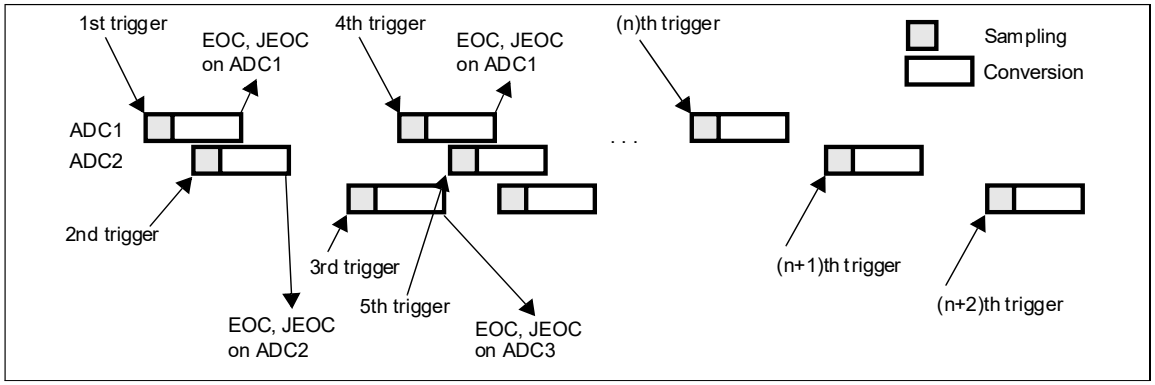



图 12-17 交替触发：各个 ADC 的注入组

### 12.9.5. 混合型规则/注入同时模式

可以中断规则组的同时转换，然后开始注入组的同时转换。

*注意：在混合型规则/注入同时模式下，必须使用同一长度来转换序列，或必须确保触发之间的间隔长于 2 个序列（双重 ADC 模式）/3 个序列（三重 ADC 模式）中的较长转换时间。否则，当序列较长的 ADC 完成上一次转换时，序列较短的 ADC 可能重新开始转换。*

### 12.9.6. 规则同时 + 交替触发组合模式

可以中断规则组的同时转换，然后开始注入组的交替触发转换。下图说明了交替触发模式中断同时规则转换的行为。

注入事件后立即开始注入交替转换。当规则转换处于运行状态时，为确保在注入转换后实现同步，所有的（主/从）ADC 规则转换均将停止，并会在注入转换结束时得以恢复运行。

*注意：在规则同时 + 交替触发组合模式下，必须使用同一长度来转换序列，或必须确保触发之间的间隔长于 2 个序列（双重 ADC 模式）/3 个序列（三重 ADC 模式）的长转换时间。否则，当序列较长的 ADC 完成上一次转换时，序列较短的 ADC 可能重新开始转换。*

*如果转换序列中断（例如 DMA 传输结束时），则必须首先通过在独立模式下进行配置来将多重 ADC 定序器复位（位 DUAL[4:0] = 00000），然后才可以对交替模式进行编程。*

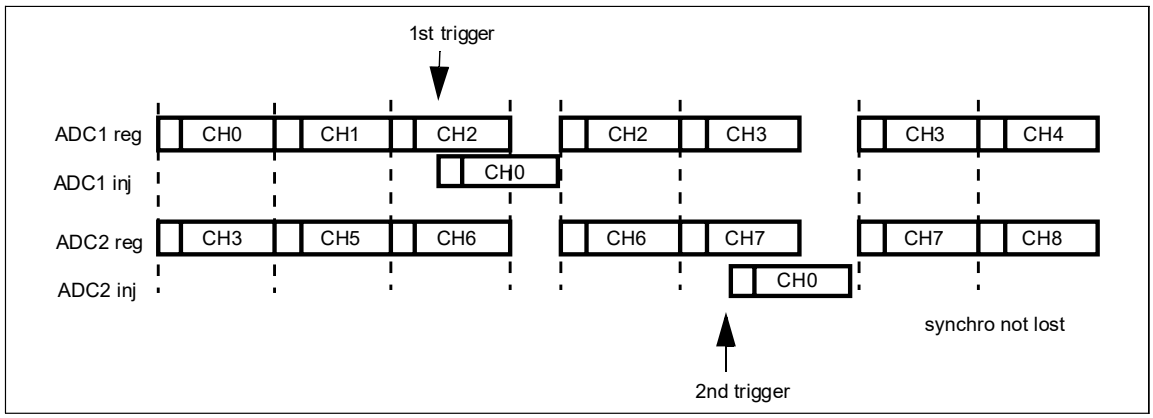



图 12-18 交替 + 规则同时

在已导致规则转换中断的注入转换期间出现的触发将被忽略。下图说明了这种情况下的行为（第 2 次触发被忽略）。

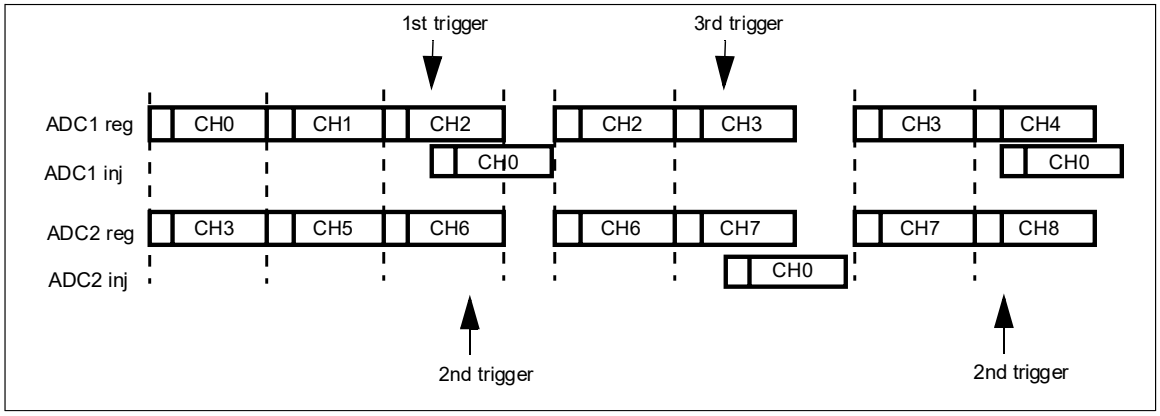



图 12-19 在注入转换期间出现的触发事件

## 12.10. 温度传感器和内部参考电压

温度传感器可用于测量器件的环境温度 (TA)。内部连接到 ADC1_IN16 通道，而 ADC1 用于将传感器输出电压转换为数字值，不使用时可将传感器置于掉电模式。

内部参考电压(VREFINT)内部连接到 ADC1_IN17 通道。

*注意：必须将 TSVREFE 位置 1 才能同时对两个通道进行转换。ADC1_IN16 和 ADC1_IN17 (VREFINT)。*

**温度传感器主要特性**

- 支持的温度范围：-40 °C 到 125 °C
- 精度：±1.5 °C

**读取温度**

要使用传感器，请执行以下操作：

- 选择 ADC1_IN16 输入通道。
- 选择一个采样时间，该采样时间要大于数据手册中所指定的最低采样时间。
- 在 ADC_CCR 寄存器中将 TSVREFE 位置 1，以便将温度传感器从掉电模式中唤醒。
- 通过将 SWSTART 位置 1（或通过外部触发）开始 ADC 转换。
- 读取 ADC 数据寄存器中生成的 VSENSE 数据。
- 使用以下公式计算温度：

$$\textbf{温度(单位为℃)} = \{(\mathbf{V_{SENSE}} - \mathbf{V_{25}}) / \textbf{Avg_Slope}\} + \mathbf{25}$$

其中：

- $V_{25}$ = 25 °C 时的 $V_{SENSE}$ 值。
- Avg_Slope = 温度与 $V_{SENSE}$ 曲线的平均斜率（以 mV/°C 或 µV/°C 表示）。

有关 V25 和 Avg_Slope 实际值的相关信息，请参见数据手册。

*注意：传感器从掉电模式中唤醒需要一个启动时间，启动时间过后其才能正确输出 VSENSE。ADC 在上电后同样需要一个启动时间，因此，为尽可能减少延迟间，应同时将 ADON 和 TSVREFE 位置 1。*

*温度传感器的输出电压随温度线性变化。由于工艺不同，该线性函数的偏移量取决于各个芯片（芯片之间的温度变化可达 45 °C）。*

*内部温度传感器更适用于对温度变量而非绝对温度进行测量的应用情况。如果需要读取精确温度，则应使用外部温度传感器。*

## 12.11. 锂电池电压检测

锂电池电压(VBAT_LI )固定连接到 ADC2_IN16 输入通道，为保证锂电池电压检测范围(0~4.2V)，ADC_IN16 内部增加了分压。使用以下公式计算锂电池的电压值：

$$V_{BAT_LI} = V_{ADC2_IN16} \times 2.25$$

## 12.12. ADC 中断

当模拟看门狗状态位和溢出状态位分别置 1 时，规则组和注入组在转换结束时可能会产生中断。可以使用单独的中断使能位以实现灵活性。

ADC_SR 寄存器中存在另外两个标志，但这两个标志不存在中断相关性：

- JSTRT（开始转换注入组的通道）
- STRT（开始转换规则组的通道）

表 12-6 ADC 中断

| 中断事件 | 事件标志 | 使能控制位 |
|---|---|---|
| 结束规则组的转换 | EOC | EOCIE |
| 结束注入组的转换 | JEOC | JEOCIE |
| 模拟看门狗状态位置 1 | AWD | AWDIE |
| 溢出 (Overrun) | OVR | OVRIE |

## 12.13. ADC 寄存器

有关寄存器说明中使用的缩写，请参见第 1.1 节。

必须在字级别（32 位）对外设寄存器执行写入操作。而读访问可支持字节（8 位）、半字（16 位）或字（32 位）。

### 12.13.1. ADC 状态寄存器(ADC_SR)

偏移地址： 0x00

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | | | | OVR | STRT | JSTRT | JEOC | EOC | AWD |
| | | | | | | | | | | rc_w0 | rc_w0 | rc_w0 | rc_w0 | rc_w0 | rc_w0 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:6 | 保留 | 必须保持复位值。 |
| 5 | OVR | 溢出<br>数据丢失时，硬件将该位置 1（在单一模式或双重/三重模式下）。但需要通过软件清零。溢出检测仅在 DMA = 1 或 EOCS = 1 时使能。<br>0：未发生溢出 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 1：发生溢出 |
| 4 | STRT | 规则通道开始标志<br>规则通道转换开始时，硬件将该位置 1。但需要通过软件清零。<br>0：未开始规则通道转换<br>1：已开始规则通道转换 |
| 3 | JSTRT | 注入通道开始标志<br>注入组转换开始时，硬件将该位置 1。但需要通过软件清零。<br>0：未开始注入组转换<br>1：已开始注入组转换 |
| 2 | JEOC | 注入通道转换结束<br>组内所有注入通道转换结束时，硬件将该位置 1。但需要通过软件清零。<br>0：转换未完成<br>1：转换已完成 |
| 1 | EOC | 规则通道转换结束<br>规则组通道转换结束后，硬件将该位置 1。通过软件或通过读取 ADC_DR 寄存器将该位清零。<br>0：转换未完成 (EOCS=0) 或转换序列未完成 (EOCS=1)<br>1：转换已完成 (EOCS=0) 或转换序列已完成 (EOCS=1) |
| 0 | AWD | 模拟看门狗标志<br>当转换电压超过在 ADC_LTR 和 ADC_HTR 寄存器中编程的值时，硬件将该位置 1。但需要通过软件清零。<br>0：未发生模拟看门狗事件<br>1：发生模拟看门狗事件 |

### 12.13.2. ADC 控制寄存器 1 (ADC_CR1)

偏移地址： 0x04

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | OVREI | RES | | AWDEN | JAWDEN | Reserved | | | | | |
| | | | | | rw | rw | rw | rw | rw | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DISCNUM[2:0] | | | JDISCEN | DISC EN | JAUTO | AWDSGL | SCAN | JEOCIE | AWDIE | AWDCH[4:0] | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:27 | 保留 | 必须保持复位值。 |
| 26 | OVRIE | 溢出中断使能<br>通过软件将该位置 1 和清零可使能/禁止溢出中断。<br>0：禁止溢出中断<br>1：使能溢出中断。OVR 位置 1 时产生中断。 |
| 25:24 | RES[1:0] | 分辨率<br>通过软件写入这些位可选择转换的分辨率。<br>00：12 位（15 ADCCLK 周期）<br>01：10 位（13 ADCCLK 周期） |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 10：8 位（11 ADCCLK 周期）<br>11：6 位（9 ADCCLK 周期） |
| 23 | AWDEN | 规则通道上的模拟看门狗使能<br>此位由软件置 1 和清零。<br>0：在规则通道上禁止模拟看门狗<br>1：在规则通道上使能模拟看门狗 |
| 22 | JAWDEN | 注入通道上的模拟看门狗使能<br>此位由软件置 1 和清零。<br>0：在注入通道上禁止模拟看门狗<br>1：在注入通道上使能模拟看门狗 |
| 21:16 | 保留 | 必须保持复位值。 |
| 15:13 | DISCNUM[2:0] | 不连续采样模式通道计数<br>软件将写入这些位，用于定义在接收到外部触发后于不连续采样模式下转换的规则通道数。<br>000：1 个通道<br>001：2 个通道<br>...<br>111：8 个通道 |
| 12 | JDISCEN | 注入通道的不连续采样模式<br>通过软件将该位置 1 和清零可使能/禁止注入通道的不连续采样模式。<br>0：禁止注入通道的不连续采样模式<br>1：使能注入通道的不连续采样模式 |
| 11 | DISCEN | 规则通道的不连续采样模式<br>通过软件将该位置 1 和清零可使能/禁止规则通道的不连续采样模式。<br>0：禁止规则通道的不连续采样模式<br>1：使能规则通道的不连续采样模式 |
| 10 | JAUTO | 注入组自动转换<br>通过软件将该位置 1 和清零可在规则组转换后分别使能/禁止注入组自动转换。<br>0：禁止注入组自动转换<br>1：使能注入组自动转换 |
| 9 | AWDSGL | 在扫描模式下使能单一通道上的看门狗<br>通过软件将该位置 1 和清零可分别使能/禁止通过 AWDCH[4:0] 位确定的通道上的模拟看门狗。<br>0：在所有通道上使能模拟看门狗<br>1：在单一通道上使能模拟看门狗 |
| 8 | SCAN | 扫描模式<br>通过软件将该位置 1 和清零可使能/禁止扫描模式。在扫描模式下，转换通过 ADC_SQRx 或 ADC_JSQRx 寄存器选择的输入。<br>0：禁止扫描模式<br>1：使能扫描模式<br>注意： EOCIE 位置 1 时将生成 EOC 中断：<br>–如果 EOCS 位清零，在每个规则组序列转换结束时<br>–如果 EOCS 位置 1，在每个规则通道转换结束时<br>注意：JEOCIE 位置 1 时，JEOC 中断仅在最后一个通道转换结束时生成。 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 7 | JEOCIE | 注入通道的中断使能<br>通过软件将该位置 1 和清零可使能/禁止注入通道的转换结束中断。<br>0：禁止 JEOC 中断<br>1：使能 JEOC 中断。JEOC 位置 1 时产生中断。 |
| 6 | AWDIE | 模拟看门狗中断使能<br>通过软件将该位置 1 和清零可使能/禁止模拟看门狗中断。<br>0：禁止模拟看门狗中断<br>1：使能模拟看门狗中断 |
| 5 | EOCIE | EOC 中断使能<br>通过软件将该位置 1 和清零可使能/禁止转换结束中断。<br>0：禁止 EOC 中断<br>1：使能 EOC 中断 EOC 位置 1 时产生中断。 |
| 4:0 | AWDCH[4:0] | 模拟看门狗通道选择位<br>这些位将由软件置 1 和清零。它们用于选择由模拟看门狗监控的输入通道。<br>注意：<br>00000：ADC 模拟输入通道 0<br>00001：ADC 模拟输入通道 1<br>...<br>01111：ADC 模拟输入通道 15<br>10000：ADC 模拟输入通道 16<br>10001：ADC 模拟输入通道 17<br>10010：ADC 模拟输入通道 18<br>保留其它值 |

### 12.13.3. ADC 控制寄存器 2 (ADC_CR2)

偏移地址： 0x08

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | SWSTART | EXTEN | | EXTSEL[3:0] | | | | reserved | JSWSTART | JEXTEN | | JEXTSEL[3:0] | | | |
| | rw | rw | rw | rw | rw | rw | rw | | rw | rw | rw | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | ALIGN | EOCS | DDS | DMA | Reserved | | | | | | CONT | ADON |
| | | | | rw | rw | rw | rw | | | | | | | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | 保留 | 必须保持复位值。 |
| 30 | SWSTART | 开始转换规则通道<br>通过软件将该位置 1 可开始转换，而硬件会在转换开始后将该位清零。<br>0：复位状态<br>1：开始转换规则通道<br>*注意： 该位只能在 ADON = 1 时置 1，否则不会启动转换。* |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 29:28 | EXTEN | 规则通道的外部触发使能<br>通过软件将这些位置 1 和清零可选择外部触发极性和使能规则组的触发。<br>00：禁止触发检测<br>01：上升沿上的触发检测<br>10：下降沿上的触发检测<br>11：上升沿和下降沿上的触发检测 |
| 27:24 | EXTSEL[3:0] | 为规则组选择外部事件<br>这些位可选择用于触发规则组转换的外部事件。<br>0000：定时器 1 CC1 事件<br>0001：定时器 1 CC2 事件<br>0010：定时器 1 CC3 事件<br>0011：定时器 2 CC2 事件<br>0100：定时器 2 CC3 事件<br>0101：定时器 2 CC4 事件<br>0110：定时器 2 TRGO 事件<br>0111：定时器 3 CC1 事件<br>1000：定时器 3 TRGO 事件<br>1001：定时器 4 CC4 事件<br>1010：定时器 5 CC1 事件<br>1011：定时器 5 CC2 事件<br>1100：定时器 5 CC3 事件<br>1101：定时器 8 CC1 事件<br>1110：定时器 8 TRGO 事件<br>1111：EXTI 线 11 |
| 23 | 保留 | 必须保持复位值。 |
| 22 | JSWSTART | 开始转换注入通道<br>转换开始后，软件将该位置 1，而硬件将该位清零。<br>0：复位状态<br>1：开始转换注入通道<br>*注意： 该位只能在 ADON = 1 时置 1，否则不会启动转换。* |
| 21:20 | JEXTEN | 注入通道的外部触发使能<br>通过软件将这些位置 1 和清零可选择外部触发极性和使能注入组的触发。<br>00：禁止触发检测<br>01：上升沿上的触发检测<br>10：下降沿上的触发检测<br>11：上升沿和下降沿上的触发检测 |
| 19:16 | JEXTSEL[3:0] | 为注入组选择外部事件<br>这些位可选择用于触发注入组转换的外部事件。<br>0000：定时器 1 CC4 事件<br>0001：定时器 1 TRGO 事件<br>0010：定时器 2 CC1 事件<br>0011：定时器 2 TRGO 事件<br>0100：定时器 3 CC2 事件<br>0101：定时器 3 CC4 事件<br>0110：定时器 4 CC1 事件<br>0111：定时器 4 CC2 事件 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 1000：定时器 4 CC3 事件<br>1001：定时器 4 TRGO 事件<br>1010：定时器 5 CC4 事件<br>1011：定时器 5 TRGO 事件<br>1100：定时器 8 CC2 事件<br>1101：定时器 8 CC3 事件<br>1110：定时器 8 CC4 事件<br>1111：EXTI 线 15 |
| 15:12 | 保留 | 必须保持复位值。 |
| 11 | ALIGN | 数据对齐<br>此位由软件置 1 和清零。<br>0：右对齐<br>1：左对齐 |
| 10 | EOCS | 结束转换选择<br>此位由软件置 1 和清零。<br>0：在每个规则转换序列结束时将 EOC 位置 1。溢出检测仅在 DMA=1 时使能。<br>1：在每个规则转换结束时将 EOC 位置 1。使能溢出检测。 |
| 9 | DDS | DMA 禁止选择（对于单一 ADC 模式）<br>此位由软件置 1 和清零。<br>0：最后一次传输后不发出新的 DMA 请求（在 DMA 控制器中进行配置）<br>1：只要发生数据转换且 DMA = 1，便会发出 DMA 请求 |
| 8 | DMA | 直接存储器访问模式（对于单一 ADC 模式）<br>此位由软件置 1 和清零。<br>0：禁止 DMA 模式<br>1：使能 DMA 模式 |
| 7:2 | 保留 | 必须保持复位值。 |
| 1 | CONT | 连续转换<br>此位由软件置 1 和清零。该位置 1 时，转换将持续进行，直到该位清零。<br>0：单次转换模式<br>1：连续转换模式 |
| 0 | ADON | A/D 转换器开启/关闭<br>此位由软件置 1 和清零。<br>注意：<br>0：禁止 ADC 转换并转至掉电模式<br>1：使能 ADC |

### 12.13.4. ADC 采样时间寄存器 1 (ADC_SMPR1)

偏移地址： 0x0C

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | SMP18[2:0] | | | SMP17[2:0] | | | SMP16[2:0] | | | SMP15[2:1] | |
| | | | | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SMP15_0 | SMP14[2:0] | | | SMP13[2:0] | | | SMP12[2:0] | | | SMP11[2:0] | | | SMP10[2:0] | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:27 | 保留 | 必须保持复位值。 |
| 26:0 | SMPx[2:0] | 通道 X 采样时间选择<br>通过软件写入这些位可分别为各个通道选择采样时间。在采样周期期间，通道选择位必须保持不变。<br>注意：<br>000：3 个周期<br>001：15 个周期<br>010：28 个周期<br>011：56 个周期<br>100：84 个周期<br>101：112 个周期<br>110：144 个周期<br>111：480 个周期 |

### 12.13.5. ADC 采样时间寄存器 2 (ADC_SMPR2)

偏移地址： 0x10

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | SMP9[2:0] | | | SMP8[2:0] | | | SMP7[2:0] | | | SMP6[2:0] | | | SMP5[2:1] | |
| | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SMP5_0 | SMP4[2:0] | | | SMP3[2:0] | | | SMP2[2:0] | | | SMP1[2:0] | | | SMP0[2:0] | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:30 | 保留 | 必须保持复位值。 |
| 29:0 | SMPx[2:0] | 通道 X 采样时间选择<br>通过软件写入这些位可分别为各个通道选择采样时间。在采样周期期间，通道选择位必须保持不变。<br>注意：<br>000：3 个周期<br>001：15 个周期<br>010：28 个周期<br>011：56 个周期<br>100：84 个周期<br>101：112 个周期<br>110：144 个周期 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 111：480 个周期 |

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

偏移地址： 0x14-0x20

复位值： 0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:12 | 保留 | 必须保持复位值。 |
| 11:0 | JOFFSETx[11:0] | 注入通道 X 的数据偏移<br>通过软件写入这些位可定义在转换注入通道时从原始转换数据中减去的偏移量。可从 ADC_JDRx 寄存器中读取转换结果。 |

### 12.13.7. ADC 看门狗高阈值寄存器 (ADC_HTR)

偏移地址： 0x24

复位值： 0x0000 0FFF

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:12 | 保留 | 必须保持复位值。 |
| 11:0 | HT[11:0] | 模拟看门狗高阈值<br>通过软件写入这些位可为模拟看门狗定义高阈值。 |

### 12.13.8. ADC 看门狗低阈值寄存器 (ADC_LTR)

偏移地址： 0x28

复位值： 0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:12 | 保留 | 必须保持复位值。 |
| 11:0 | LT[11:0] | 模拟看门狗低阈值<br>通过软件写入这些位可为模拟看门狗定义低阈值。 |

### 12.13.9. ADC 规则序列寄存器 1 (ADC_SQR1)

偏移地址： 0x2C

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | L[3:0] | | | | SQ16[4:1] | | | |
| | | | | | | | | rw | rw | rw | rw | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SQ16_0 | SQ15[4:0] | | | | | SQ14[4:0] | | | | | SQ13[4:0] | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:24 | 保留 | 必须保持复位值。 |
| 23:20 | L[3:0] | 规则通道序列长度<br>通过软件写入这些位可定义规则通道转换序列中的转换总数。<br>0000：1 次转换<br>0001：2 次转换<br>…<br>1111：16 次转换 |
| 19:15 | SQ16[4:0] | 规则序列中的第十六次转换<br>通过软件写入这些位，并将通道编号 (0..18) 分配为转换序列中的第十六次转换。 |
| 14:10 | SQ15[4:0] | 规则序列中的第十五次转换 |
| 9:5 | SQ14[4:0] | 规则序列中的第十四次转换 |
| 4:0 | SQ13[4:0] | 规则序列中的第十三次转换 |

### 12.13.10. ADC 规则序列寄存器 2 (ADC_SQR2)

偏移地址： 0x30

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | SQ12[4:0] | | | | | SQ11[4:0] | | | | | SQ10[4:1] | | | |
| | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SQ10_0 | SQ9[4:0] | | | | | SQ8[4:0] | | | | | SQ7[4:0] | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:30 | 保留 | 必须保持复位值。 |
| 29:26 | SQ12[4:0] | 规则序列中的第十二次转换通过软件写入这些位，并将通道编号 (0..18) 分配为序列中的第十二次转换。 |
| 24:20 | SQ11[4:0] | 规则序列中的第十一次转换 |
| 19:15 | SQ10[4:0] | 规则序列中的第十次转换 |
| 14:10 | SQ9[4:0] | 规则序列中的第九次转换 |
| 9:5 | SQ8[4:0] | 规则序列中的第八次转换 |
| 4:0 | SQ7[4:0] | 规则序列中的第七次转换 |

### 12.13.11. ADC 规则序列寄存器 3 (ADC_SQR3)

偏移地址： 0x34

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | SQ6[4:0] | | | | | SQ5[4:0] | | | | | SQ4[4:1] | | | |
| | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:30 | 保留 | 必须保持复位值。 |
| 29:25 | SQ6[4:0] | 规则序列中的第六次转换<br>通过软件写入这些位，并将通道编号 (0..18) 分配为序列中的第六次转换。 |
| 24:20 | SQ5[4:0] | 规则序列中的第五次转换 |
| 19:15 | SQ4[4:0] | 规则序列中的第四次转换 |
| 14:10 | SQ3[4:0] | 规则序列中的第三次转换 |
| 9:5 | SQ2[4:0] | 规则序列中的第二次转换 |
| 4:0 | SQ1[4:0] | 规则序列中的第一次转换 |

### 12.13.12. ADC 注入序列寄存器 (ADC_JSQR)

偏移地址： 0x38

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | JL[1:0] | | JSQ4[4:1] | | | |
| | | | | | | | | | | rw | rw | rw | rw | rw | rw |

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:22 | 保留 | 必须保持复位值。 |
| 21:20 | JL[1:0] | 注入序列长度<br>通过软件写入这些位可定义注入通道转换序列中的转换总数。<br>00：1 次转换<br>01：2 次转换<br>10：3 次转换<br>11：4 次转换 |
| 19:15 | JSQ4[4:0] | 注入序列中的第四次转换<br>（当 JL[1:0] = 3 时，请参见下方的注释）<br>通过软件写入这些位，并将通道编号 (0..18) 分配为序列中的第四次转换。 |
| 14:10 | JSQ3[4:0] | 注入序列中的第三次转换<br>（当 JL[1:0] = 3 时，请参见下方的注释） |
| 9:5 | JSQ2[4:0] | 注入序列中的第二次转换<br>（当 JL[1:0] = 3 时，请参见下方的注释） |
| 4:0 | JSQ1[4:0] | 注入序列中的第一次转换<br>（当 JL[1:0] = 3 时，请参见下方的注释） |

*注意：*

*当 JL[1:0] = 3（定序器中有 4 次注入转换）时，ADC 将按以下顺序转换通道：JSQ1[4:0]、JSQ2[4:0]、JSQ3[4:0] 和 JSQ4[4:0]。*

*当 JL = 2（定序器中有 3 次注入转换）时，ADC 将按以下顺序转换通道：JSQ2[4:0]、JSQ3[4:0] 和 JSQ4[4:0]。*

*当 JL = 1（定序器中有 2 次注入转换）时，ADC 转换通道的顺序为：先是 JSQ3[4:0]，而后是 JSQ4[4:0]。*

*当 JL = 0（定序器中有 1 次注入转换）时，ADC 将仅转换 JSQ4[4:0] 通道。*

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

偏移地址： 0x3C - 0x48

复位值： 0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15:0 | JDATA[15:0] | 注入数据 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 这些位为只读。它们包括来自注入通道 X 的转换结果。数据有左对齐和右对齐两种方式。 |

### 12.13.14. ADC 规则数据寄存器 (ADC_DR)

偏移地址： 0x4C

复位值： 0x0000 0000

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:16 | 保留 | 必须保持复位值。 |
| 15:0 | DATA[15:0] | 规则数据<br>这些位为只读。它们包括来自规则通道的转换结果。数据有左对齐和右对齐两种方式。 |

### 12.13.15. ADC 通用状态寄存器 (ADC_CSR)

偏移地址： 0x00（该偏移地址与 ADC1 基地址 + 0x300 相关）

复位值： 0x0000 0000

该寄存器可提供不同 ADC 的状态位图像。但是，它为只读形式且不允许将不同的状态位清零。必须在对应的 ADC_SR 寄存器中将其写为 0，才能将各个状态位清零。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Reserved | | | | | | | | | | OVR3 | STRT3 | JSTRT3 | JEOC 3 | EOC3 | AWD3 |
| | | | | | | | | | | ADC3 | | | | | |
| | | | | | | | | | | r | r | r | r | r | r |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | OVR2 | STRT2 | JSTRT 2 | JEOC2 | EOC2 | AWD2 | Reserved | | OVR1 | STRT1 | JSTRT1 | JEOC 1 | EOC1 | AWD1 |
| | | ADC2 | | | | | | | | ADC1 | | | | | |
| | | r | r | r | r | r | r | | | r | r | r | r | r | r |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:22 | 保留 | 必须保持复位值。 |
| 21 | OVR3 | ADC3 的溢出标志<br>该位是 ADC3_SR 寄存器中 OVR 位的副本。 |
| 20 | STRT3 | ADC3 的规则通道开始标志<br>该位是 ADC3_SR 寄存器中 STRT 位的副本。 |
| 19 | JSTRT3 | ADC3 的注入通道开始标志<br>该位是 ADC3_SR 寄存器中 JSTRT 位的副本。 |
| 18 | JEOC3 | ADC3 的注入通道转换结束 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 该位是 ADC3_SR 寄存器中 JEOC 位的副本。 |
| 17 | EOC3 | ADC3 的转换结束<br>该位是 ADC3_SR 寄存器中 EOC 位的副本。 |
| 16 | AWD3 | ADC3 的模拟看门狗标志<br>该位是 ADC3_SR 寄存器中 AWD 位的副本。 |
| 15:14 | 保留 | 必须保持复位值。 |
| 13 | OVR2 | ADC2 的溢出标志<br>该位是 ADC2_SR 寄存器中 OVR 位的副本。 |
| 12 | STRT2 | ADC2 的规则通道开始标志<br>该位是 ADC2_SR 寄存器中 STRT 位的副本。 |
| 11 | JSTRT2 | ADC2 的注入通道开始标志<br>该位是 ADC2_SR 寄存器中 JSTRT 位的副本。 |
| 10 | JEOC2 | ADC2 的注入通道转换结束<br>该位是 ADC2_SR 寄存器中 JEOC 位的副本。 |
| 9 | EOC2 | ADC2 转换结束<br>该位是 ADC2_SR 寄存器中 EOC 位的副本。 |
| 8 | AWD2 | ADC2 的模拟看门狗标志<br>该位是 ADC2_SR 寄存器中 AWD 位的副本。 |
| 7:6 | 保留 | 必须保持复位值。 |
| 5 | OVR1 | ADC1 的溢出标志<br>该位是 ADC1_SR 寄存器中 OVR 位的副本。 |
| 4 | STRT1 | ADC1 的规则通道开始标志<br>该位是 ADC1_SR 寄存器中 STRT 位的副本。 |
| 3 | JSTRT1 | ADC1 的注入通道开始标志<br>该位是 ADC1_SR 寄存器中 JSTRT 位的副本。 |
| 2 | JEOC1 | ADC1 的注入通道转换结束<br>该位是 ADC1_SR 寄存器中 JEOC 位的副本。 |
| 1 | EOC1 | ADC1 的转换结束<br>该位是 ADC1_SR 寄存器中 EOC 位的副本。 |
| 0 | AWD1 | ADC1 的模拟看门狗标志<br>该位是 ADC1_SR 寄存器中 AWD 位的副本。 |

### 12.13.16. ADC 通用控制寄存器 (ADC_CCR)

偏移地址： 0x04（该偏移地址与 ADC1 基地址 + 0x300 相关）

复位值： 0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DMA[1:0] | | DDS | Res | DELAY[3:0] | | | | Reserved | | | MULTI[4:0] | | | | |
| rw | rw | rw | | rw | rw | rw | rw | | | | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:24 | 保留 | 必须保持复位值。 |
| 23 | TSVREFE | 温度传感器和 VREFINT 使能 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 通过软件将该位置 1 和清零可使能/禁止温度传感器和 VREFINT 通道。<br>0：禁止温度传感器和 VREFINT 通道<br>1：使能温度传感器和 VREFINT 通道 |
| 22:18 | 保留 | 必须保持复位值。 |
| 17:16 | ADCPRE | ADC 预分频器<br>由软件置 1 和清零，以选择 ADC 的时钟频率。该时钟为所有 ADC 所共用。<br>注意：<br>00：PCLK2 2 分频<br>01：PCLK2 4 分频<br>10：PCLK2 6 分频<br>11：PCLK2 8 分频 |
| 15:14 | DMA | 直接存储器访问模式（对于多个 ADC 模式）<br>此位由软件置 1 和清零。<br>00：禁止 DMA 模式<br>01：使能 DMA 模式 1（依次 2/3 半字 - 1、2、3 依次进行）<br>10：使能 DMA 模式 2（成对 2/3 半字 - 2 和 1、1 和 3、3 和 2 依次进行）<br>11：使能 DMA 模式 3（成对 2/3 字节 - 2 和 1、1 和 3、3 和 2 依次进行） |
| 13 | DDS | DMA 禁止选择（对于多个 ADC 模式）<br>此位由软件置 1 和清零。<br>0: 最后一次传输后不发出新的 DMA 请求（在 DMA 控制器中进行配置）DMA 位不通过硬件清零，但必须在生成新的 DMA 请求前，通过软件清零并设置为需要的模式。<br>1：只要数据发生转换且 DMA = 01、10 或 11，便会发出 DMA 请求。 |
| 12 | 保留 | 必须保持复位值。 |
| 11:8 | DELAY | 2 个采样阶段之间的延迟<br>由软件置 1 和清零。这些位在双重或三重交错模式下使用。<br>0000：5 * $T_{ADCCLK}$<br>0001：6 * $T_{ADCCLK}$<br>0010：7 * $T_{ADCCLK}$<br>...<br>1111：20 * $T_{ADCCLK}$ |
| 7:5 | 保留 | 必须保持复位值。 |
| 4:0 | MULTI[4:0] | 多重 ADC 模式选择<br>通过软件写入这些位可选择操作模式。<br>所有 ADC 均独立：<br>00000：独立模式<br>00001 到 01001：双重模式，ADC1 和 ADC2 一起工作，ADC3 独立<br>00001：规则同时 + 注入同时组合模式<br>00010：规则同时 + 交替触发组合模式<br>00011：Reserved<br>00101：仅注入同时模式<br>00110：仅规则同时模式 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 00111：仅交错模式<br>01001：仅交替触发模式<br>10001 到 11001：三重模式：ADC1、ADC2 和 ADC3 一起工作<br>10001：规则同时 + 注入同时组合模式<br>10010：规则同时 + 交替触发组合模式<br>10011：Reserved<br>10101：仅注入同时模式<br>10110：仅规则同时模式<br>10111：仅交错模式<br>11001：仅交替触发模式<br>其它所有组合均需保留且不允许编程<br>注意：在多重模式下，更改通道配置会生成中止，进而导致同步丢失。建议在更改配置前禁用多重 ADC 模式。 |

### 12.13.17. 适用于双重和三重模式的 ADC 通用规则数据寄存器（ADC_CDR)

偏移地址： 0x08（该偏移地址与 ADC1 基地址 + 0x300 相关）

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| DATA2[15:0] | | | | | | | | | | | | | | | |
| r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA1[15:0] | | | | | | | | | | | | | | | |
| r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:16 | DATA2[15:0] | 规则转换对中的第二个数据项<br>在双重模式下，这些位包含 ADC2 的规则数据。请参见双重 ADC 模式。<br>在三重模式下，这些位还可以包含 ADC2、ADC1 和 ADC3 的规则数据。请参见三重 ADC 模式。 |
| 15:0 | DATA1[15:0] | 规则转换对中的第一个数据项<br>在双重模式下，这些位包含 ADC1 的规则数据。请参见双重 ADC 模式。<br>在三重模式下，这些位还可以包含 ADC1、ADC3 和 ADC2 的规则数据。请参见三重 ADC 模式。 |

# 13. 数模转换器(DAC)

## 13.1. DAC 简介

DAC 模块是 12 位电压输出数模转换器。DAC 可以按 8 位或 12 位模式进行配置，并且可与 DMA 控制器配合使用。在 12 位模式下，数据可以采用左对齐或右对齐。DAC 有两个输出通道，每个通道各有一个转换器。在 DAC 双通道模式下，每个通道可以单独进行转换；当两个通道组合在一起同步执行更新操作时，也可以同时进行转换。

## 13.2. DAC 主要特性

- 两个 DAC 转换器：各对应一个输出通道；
- 12 位模式下数据采用左对齐或右对齐；
- 同步更新功能；
- 生成噪声波；
- 生成三角波；
- DAC 双通道单独或同时转换；
- 每个通道都具有 DMA 功能；
- DMA 下溢错误检测；
- 通过外部触发信号进行转换。

下图所示为 DAC 通道框图。

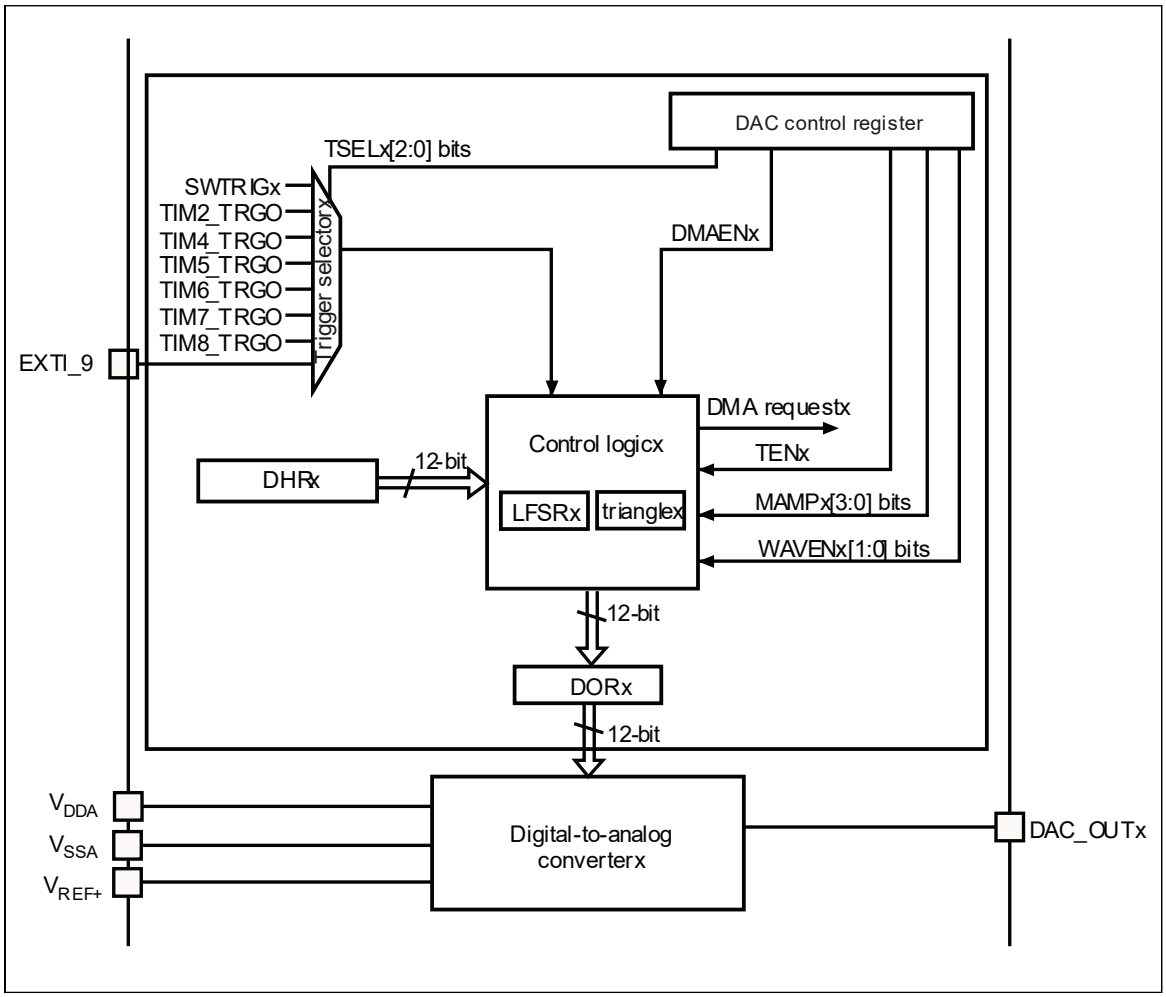



图 13-1 DAC 通道框图

使能 DAC 通道 x 后，相应 GPIO 引脚（PA4 或 PA5）将自动连接到模拟转换器输出(DAC_OUTx)。为了避免寄生电流消耗，应首先将 PA4 或 PA5 引脚配置为模拟模式 (AIN)。

## 13.3. DAC 功能说明

### 13.3.1. DAC 通道使能

将 DAC_CR 寄存器中的相应 ENx 位置 1, 即可接通对应 DAC 通道。经过一段启动时间 $t_{WAKEUP}$ 后, DAC 通道被真正使能。

*注意： ENx 位只会使能模拟 DAC Channelx 宏单元。即使 ENx 位复位，DAC Channelx 数字接口仍处于使能状态。*

### 13.3.2. DAC 输出缓冲器使能

DAC 集成了两个输出缓冲器，可用来降低输出阻抗并在不增加外部运算放大器的情况下直接驱动外部负载。通过 DAC_CR 寄存器中的相应 BOFFx 位，可使能或禁止各 DAC 通道输出缓冲器。

### 13.3.3. DAC 数据格式

根据所选配置模式，数据必须按如下方式写入指定寄存器：

- 对于 DAC 单通道 x，有三种可能的方式：
  - 8 位右对齐：软件必须将数据加载到 DAC_DHR8Rx [7:0] 位(存储到 DHRx[11:4]位)。
  - 12 位左对齐：软件必须将数据加载到 DAC_DHR12Lx [15:4] 位(存储 DHRx[11:0]位)。
  - 12 位右对齐：软件必须将数据加载到 DAC_DHR12Rx [11:0] 位(存储 DHRx[11:0]位)。

根据加载的 DAC_DHRyyyx 寄存器，用户写入的数据将移位并存储到相应的 DHRx（数据保持寄存器 x，即内部非存储器映射寄存器）。之后 DHRx 寄存器将被自动加载，或者通过软件或外部事件触发加载到 DORx 寄存器。

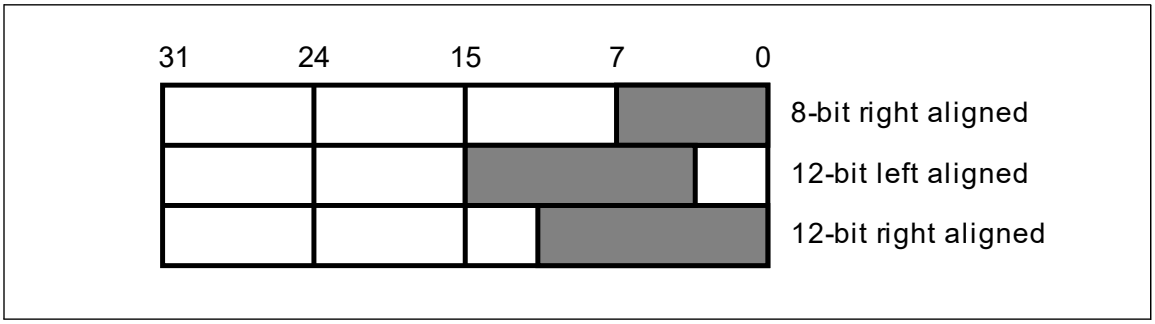



图 13-2 DAC 单通道模式下的数据寄存器

- 对于 DAC 双通道，有三种可能的方式：
  - 8 位右对齐：将 DAC 1 通道的数据加载到 DAC_DHR8RD [7:0] 位(存储到 DHR1[11:4] 位)，将 DAC 2 通道的数据加载到 DAC_DHR8RD [15:8] 位(存储到 DHR2[11:4] 位)。
  - 12 位左对齐：将 DAC 1 通道的数据加载到 DAC_DHR12RD [15:4] 位(存储到 DHR1[11:0] 位)，将 DAC 2 通道的数据加载到 DAC_DHR12RD [31:20] 位(存储到 DHR2[11:0] 位)。
  - 12 位右对齐：将 DAC 1 通道的数据加载到 DAC_DHR12RD [11:0] 位(存储到 DHR1[11:0] 位)，将 DAC 2 通道的数据加载到 DAC_DHR12RD [27:16] 位(存储到 DHR2[11:0] 位)。

根据加载的 DAC_DHRyyyD 寄存器，用户写入的数据将移位并存储到 DHR1 和 DHR2（数据保持寄存器，即内部非存储器映射寄存器）。之后 DHR1 和 DHR2 寄存器将被自动加载，或者通过软件或外部事件触发分别被加载到 DOR1 和 DOR2 寄存器。

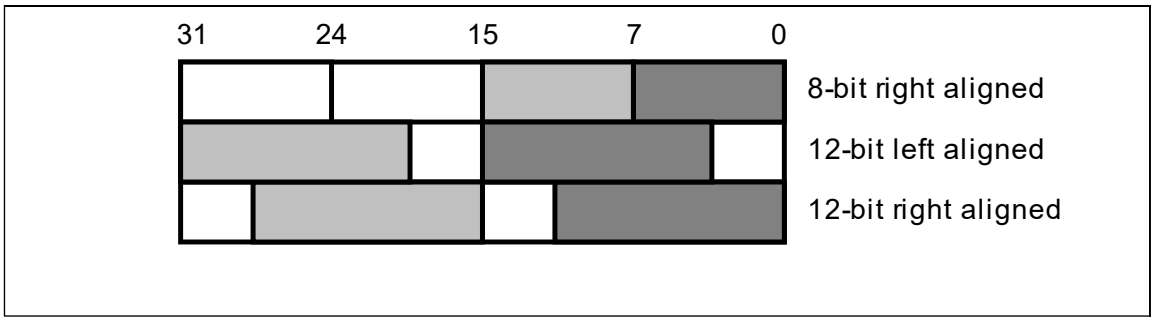



图 13-3 DAC 双通道模式下的数据寄存器

### 13.3.4. DAC 转换

DAC_DORx 无法直接写入，任何数据都必须通过加载 DAC_DHRx 寄存器（写入 DAC_DHR8Rx、DAC_DHR12Lx、DAC_DHR12Rx、DAC_DHR8RD、DAC_DHR12LD 或 DAC_DHR12LD）才能传输到 DAC 通道 x。

如果未选择硬件触发（DAC_CR 寄存器中的 TENx 位复位），那么经过一个 APB1 时钟周期后，

DAC_DHRx 寄存器中存储的数据将自动转移到 DAC_DORx 寄存器。但是，如果选择硬件触发（置位 DAC_CR 寄存器中的 TENx 位）且触发条件到来，将在三个 APB1 时钟周期后进行转移。

当 DAC_DORx 加载了 DAC_DHRx 内容时，模拟输出电压将在一段时间 $t_{SETTLING}$ 后可用，具体时间取决于电源电压和模拟输出负载。

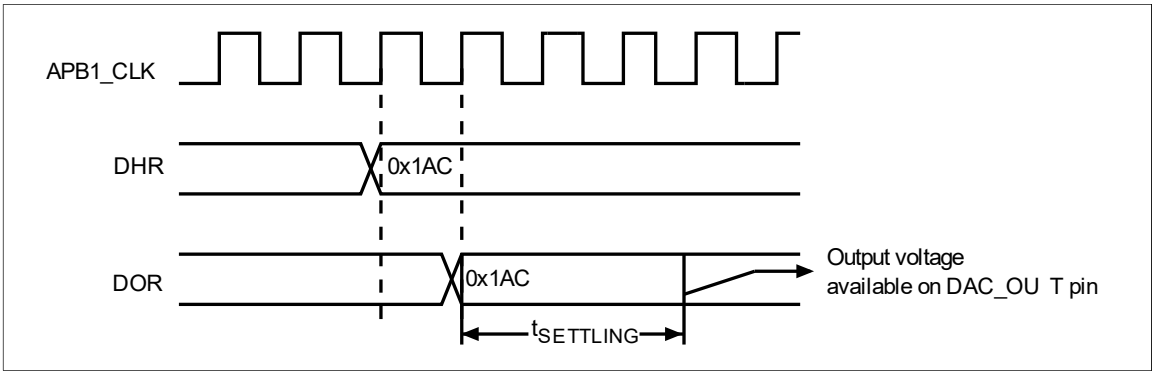



图 13-4 关闭触发（TEN = 0）时的转换时序图 TEN = 0

### 13.3.5. DAC 输出电压

经过线性转换后，数字输入会转换为 0 到 $V_{REF+}$ 之间的输出电压。

各 DAC 通道引脚的模拟输出电压通过以下公式确定：

$$\mathbf{DAC_{output}} = \mathbf{V_{REF}} \times \frac{\boldsymbol{DOR}}{\mathbf{4095}}$$

### 13.3.6. DAC 触发选择

如果 TENx 控制位置 1，可通过外部事件（定时计数器、外部中断线）触发转换。TSELx[2:0] 控制位将决定通过 8 个可能事件中的哪一个来触发转换，如下表所示。

表 13-1 外部触发器

| 源 | 类型 | TSEL[2:0] |
|---|---|---|
| Timer 6 TRGO event | 片上定时器的内部信号 | 0 |
| Timer 8 TRGO event | 1 | |
| Timer 7 TRGO event | 10 | |
| Timer 5 TRGO event | 11 | |
| Timer 2 TRGO event | 100 | |
| Timer 4 TRGO event | 101 | |
| EXTI line9 | 外部引脚 | 110 |
| SWTRIG | 软件控制位 | 111 |

每当 DAC 接口在所选定时器 TRGO 输出或所选外部中断线 9 上检测到上升沿时，DAC_DHRx 寄存器中存储的最后一个数据即会转移到 DAC_DORx 寄存器中。发生触发后再经过三个 APB1 周期，DAC_DORx 寄存器将会得到更新。

如果选择软件触发，一旦 SWTRIG 位置 1，转换即会开始。DAC_DHRx 寄存器内容加载到 DAC_DORx 寄存器中后，SWTRIG 即由硬件复位。

*注意：ENx 位置 1 时，无法更改 TSELx[2:0] 位。*

*如果选择软件触发，DAC_DHRx 寄存器的内容只需一个 APB1 时钟周期即可转移到 DAC_DORx 寄存器。*

### 13.3.7. DMA 请求

每个 DAC 通道都具有 DMA 功能。两个 DMA 通道用于处理 DAC 通道的 DMA 请求。

当 DMAENx 位置 1 时，如果发生外部触发（而不是软件触发），则将产生 DAC DMA 请求。 DAC_DHRx 寄存器的值随后转移到 DAC_DORx 寄存器。

在双通道模式下，如果两个 DMAENx 位均置 1，则将产生两个 DMA 请求。如果只需要一个 DMA 请求，应仅将相应 DMAENx 位置 1。这样，应用程序可以在双通道模式下通过一个 DMA 请求和一个特定 DMA 通道来管理两个 DAC 通道。

**DMA 下溢**

DAC DMA 请求没有缓冲队列。这样，如果第二个外部触发到达时尚未收到第一个外部触发的确认，将不会发出新的请求，并且 DAC_SR 寄存器中的 DAM 通道下溢标志 DMAUDRx 将置 1，以报告这一错误状况。 DMA 数据传输随即禁止，并且不再处理其他 DMA 请求。DAC 通道仍将继续转换旧有数据。

软件应通过写入“1”来将 DMAUDRx 标志清零，将所用 DMA 数据流的 DMAEN 位清零，并重新初始化 DMA 和 DAC 通道，以便正确地重新开始 DMA 传输。软件应修改 DAC 触发转换频率或减轻 DMA 工作负载，以避免再次发生 DMA 下溢。最后，可通过使能 DMA 数据传输和转换触发来继续完成 DAC 转换。

对于各 DAC 通道，如果使能 DAC_CR 寄存器中相应的 DMAUDRIEx 位，还将产生中断。

### 13.3.8. 生成噪声

为了生成可变振幅的伪噪声，可使用 LFSR（线性反馈移位寄存器）。将 WAVEx[1:0] 置为“01”即可选择生成噪声。LFSR 中的预加载值为 0xAAA。在每次发生触发事件后，经过三个 APB1 时钟周期，该寄存器会依照特定的计算算法完成更新。

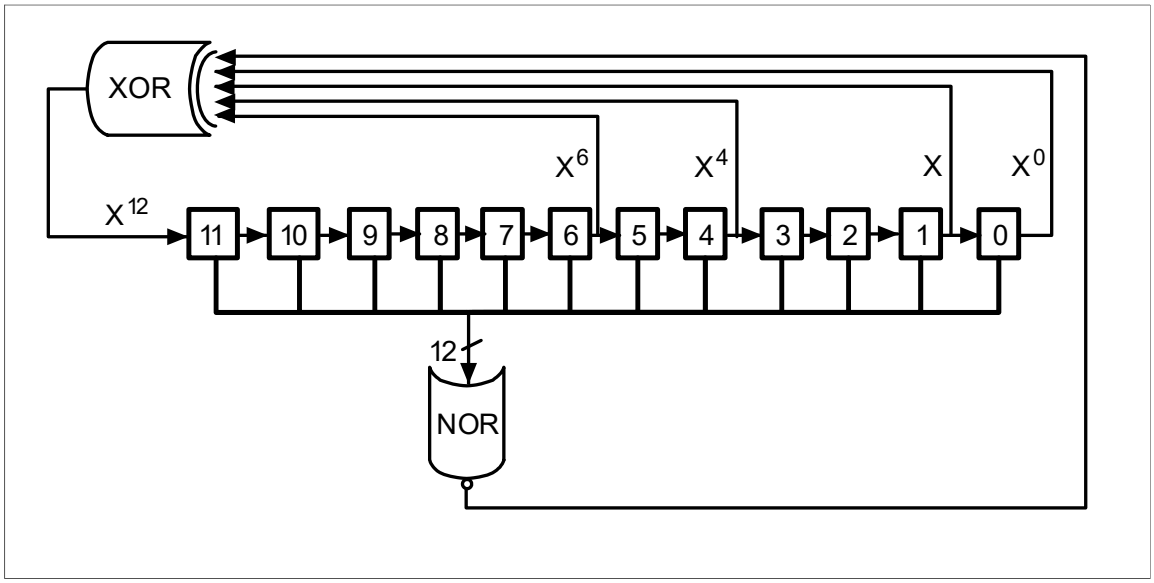



图 13-5 DAC LFSR 寄存器计算算法

LFSR 值可以通过 DAC_CR 寄存器中的 MAMPx[3:0] 位来部分或完全屏蔽，在不发生溢出的情况下，该值将与 DAC_DHRx 的内容相加，然后存储到 DAC_DORx 寄存器中。

如果 LFSR 为 0x0000，将向其注入“1”（防锁定机制）。

可以通过复位 WAVEx[1:0] 位来将 LFSR 波形产生功能关闭。

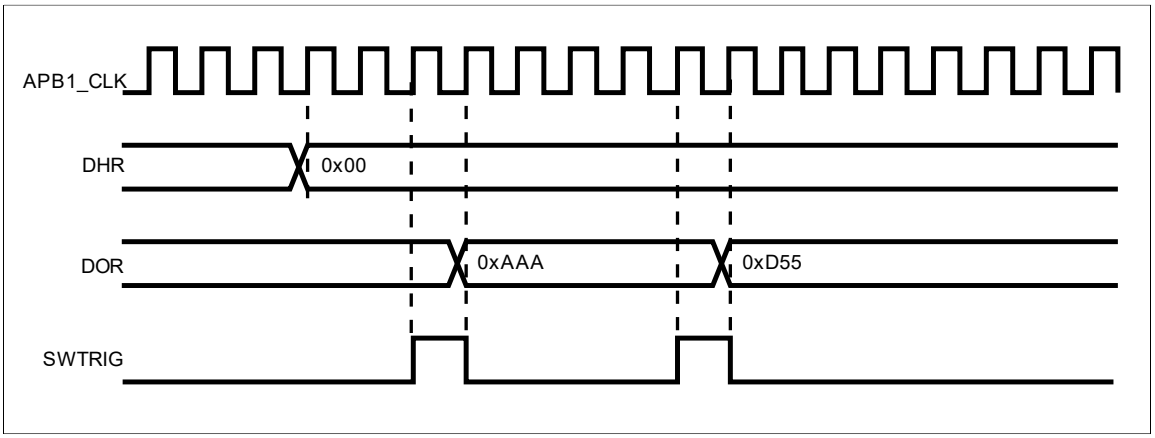



图 13-6 LFSR 产生波形的 DAC 转换（使能软件触发）

*注意： 要生成噪声，必须通过将 DAC_CR 寄存器中的 TENx 位置 1 来使能 DAC 触发。*

### 13.3.9. 生成三角波

可以在直流电流或慢变信号上叠加一个小幅三角波。将 WAVEx[1:0] 置为“10”即可选择 DAC 生成三角波。振幅通过 DAC_CR 寄存器中的 MAMPx[3:0] 位进行配置。每次发生触发事件后，经过三个 APB1 时钟周期，内部三角波计数器将会递增。在不发生溢出的情况下，该计数器的值将与 DAC_DHRx 寄存器内容相加，所得总和将存储到 DAC_DORx 寄存器中。只要小于 MAMPx[3:0] 位定义的最大振幅，三角波计数器就会一直递增。一旦达到配置的振幅，计数器将递减至零，然后再递增，以此类推。

可以通过复位 WAVEx[1:0] 位来将三角波产生功能关闭。

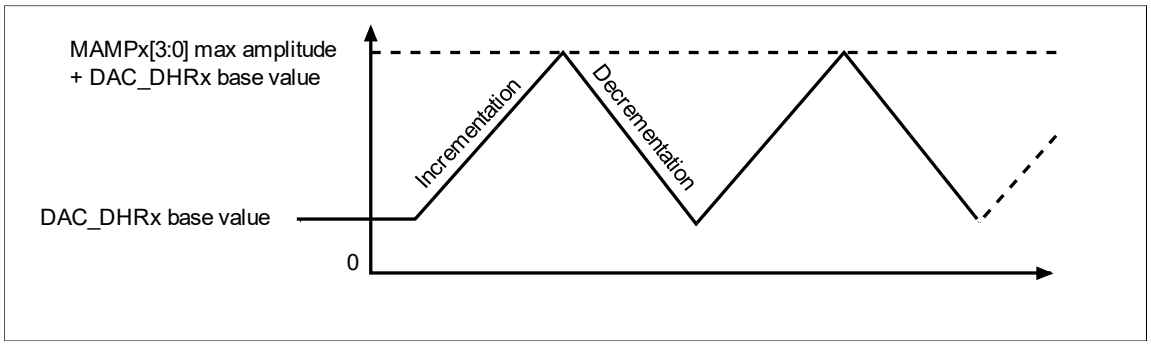



图 13-7 生成 DAC 三角波

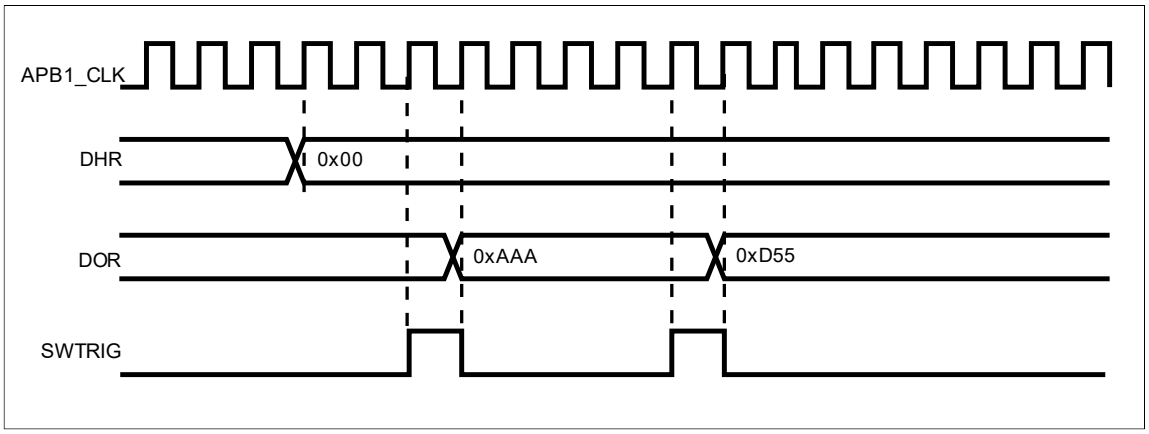



图 13-8 生成三角波波形的 DAC 转换（使能软件触发）

*注意：　要生成三角波，必须通过将 DAC_CR 寄存器中的 TENx 位置 1 来使能 DAC 触发。*

*MAMPx[3:0] 位必须在使能 DAC 之前进行配置，否则将无法更改*

## 13.4. DAC 双通道转换

为了在同时需要两个 DAC 通道的应用中有效利用总线带宽，DAC 模块实现了三个双寄存器：DHR8RD、DHR12RD 和 DHR12LD。这样，只需一个寄存器访问即可同时驱动两个 DAC 通道。

通过两个 DAC 通道和这三个双寄存器可以实现 11 种转换模式。但如果需要，所有这些转换模式也都可以通过单独的 DHRx 寄存器来实现。

下面几段内容将介绍所有这些模式。

### 13.4.1. 独立触发（不产生波形）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

- 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1。
- 将 TSEL1[2:0] 和 TSEL2[2:0] 设置为不同的值，以配置不同的触发源。
- 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD）。

DAC 1 通道触发信号到达时，DHR1 寄存器的内容转移到 DAC_DOR1（三个 APB1 时钟周期之后）。

DAC 2 通道触发信号到达时，DHR2 寄存器的内容转移到 DAC_DOR2（三个 APB1 时钟周期之后）。

### 13.4.2. 独立触发（生成单个 LFSR）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

- 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1。
- 将 TSEL1[2:0] 和 TSEL2[2:0] 设置为不同的值，以配置不同的触发源。
- 将两个 DAC 通道的 WAVEx[1:0] 设置为“01”，并在 MAMPx[3:0] 位中配置相同的 LFSR 掩码值。
- 将 DAC 双通道数据加载到所需 DHR 寄存器（DHR12RD、DHR12LD 或 DHR8RD）。

DAC 通道 1 触发信号到达时，LFSR1 计数器内容（使用相同的掩码）与 DHR1 寄存器内容相加，所得总和转移到 DAC_DOR1 中（三个 APB1 时钟周期之后）。LFSR1 计数器随即更新。

DAC 通道 2 触发信号到达时，LFSR2 计数器内容（使用相同的掩码）与 DHR2 寄存器内容相加，所得总

和转移到 DAC_DOR2 中（三个 APB1 时钟周期之后）。LFSR2 计数器随即更新。

### 13.4.3. 独立触发（生成不同 LFSR）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

- 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1。
- 将 TSEL1[2:0] 和 TSEL2[2:0] 设置为不同的值，以配置不同的触发源。
- 将两个 DAC 通道的 WAVEx[1:0] 设置为“01”，并在 MAMP1[3:0] 和 MAMP2[3:0] 位中设置不同的 LFSR 掩码值。
- 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD）。

DAC 通道 1 触发信号到达时，LFSR1 计数器内容（使用 MAMP1[3:0] 配置的掩码）与 DHR1 寄存器内容相加，所得总和转移到 DAC_DOR1 中（三个 APB1 时钟周期之后）。LFSR1 计数器随即更新。

DAC 通道 2 触发信号到达时，LFSR2 计数器内容（使用 MAMP2[3:0] 配置的掩码）与 DHR2 寄存器内容相加，所得总和转移到 DAC_DOR2 中（三个 APB1 时钟周期之后）。LFSR2 计数器随即更新。

### 13.4.4. 独立触发（生成单个三角波）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

- 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1。
- 将 TSEL1[2:0] 和 TSEL2[2:0] 设置为不同的值，以配置不同的触发源。
- 将两个 DAC 通道的 WAVEx[1:0] 设置为“1x”，并在 MAMPx[3:0] 位中配置相同的最大振幅值。
- 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD）

DAC 通道 1 触发信号到达时，DAC 通道 1 三角波计数器内容（使用相同的三角波振幅）与 DHR1 寄存器内容相加，所得总和转移到 DAC_DOR1 中（三个 APB1 时钟周期之后）。DAC 通道 1 三角波计数器随即更新。

DAC 通道 2 触发信号到达时，DAC 通道 2 三角波计数器内容（使用相同的三角波振幅）与 DHR2 寄存器内容相加，所得总和转移到 DAC_DOR2 中（三个 APB1 时钟周期之后）。DAC 通道 2 三角波计数器随即更新。

### 13.4.5. 独立触发（生成不同三角波）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

- 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1。
- 将 TSEL1[2:0] 和 TSEL2[2:0] 设置为不同的值，以配置不同的触发源。
- 将两个 DAC 通道的 WAVEx[1:0] 设置为“1x”，并在 MAMP1[3:0] 和 MAMP2[3:0] 位中设置不同的最大振幅值。
- 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD）。

DAC 通道 1 触发信号到达时，DAC 通道 1 三角波计数器内容（使用 MAMP1[3:0] 配置的三角波振幅）与 DHR1 寄存器内容相加，所得总和转移到 DAC_DOR1 中（三个 APB1 时钟周期之后）。 DAC 通道 1 三角波计数器随即更新。

DAC 通道 2 触发信号到达时，DAC 通道 2 三角波计数器内容（使用 MAMP2[3:0] 配置的三角波振幅）与 DHR2 寄存器内容相加，所得总和转移到 DAC_DOR2 中（三个 APB1 时钟周期之后）。DAC 通道 2

三角波计数器随即更新。

### 13.4.6. 同步软件启动

要将 DAC 配置为此转换模式，需要遵循以下顺序:

- 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD)。
- 在此配置中，DHR1 和 DHR2 寄存器内容会在一个 APB1 时钟周期后分别转移到 DAC_DOR1 和 DAC_DOR2 中。

### 13.4.7. 同步触发（不产生波形）

要将 DAC 配置为此转换模式，需要遵循以下顺序:

- 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1。
- 将 TSEL1[2:0] 和 TSEL2[2:0] 设置为相同的值，以便为两个 DAC 通道配置相同的触发源。
- 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD)。

当触发信号到达时， DHR1 和 DHR2 寄存器内容将分别转移到 DAC_DOR1 和 DAC_DOR2 中（三个 APB1 时钟周期之后)。

### 13.4.8. 同步触发（生成单个 LFSR）

要将 DAC 配置为此转换模式，需要遵循以下顺序:

- 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1。
- 将 TSEL1[2:0] 和 TSEL2[2:0] 设置为相同的值，以便为两个 DAC 通道配置相同的触发源。
- 将两个 DAC 通道的 WAVEx[1:0] 设置为“01”，并在 MAMPx[3:0] 位中配置相同的 LFSR 掩码值。
- 将 DAC 双通道数据加载到所需 DHR 寄存器（DHR12RD、DHR12LD 或 DHR8RD)。

触发信号到达时，LFSR1 计数器内容（使用相同的掩码）与 DHR1 寄存器内容相加，所得总和转移到 DAC_DOR1 中（三个 APB1 时钟周期之后)。 LFSR1 计数器随即更新。同时，LFSR2 计数器内容（使用相同的掩码）与 DHR2 寄存器内容相加，所得总和转移到 DAC_DOR2 中（三个 APB1 时钟周期之后)。LFSR2 计数器随即更新。

### 13.4.9. 同步触发（生成不同 LFSR）

要将 DAC 配置为此转换模式，需要遵循以下顺序:

- 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1。
- 将 TSEL1[2:0] 和 TSEL2[2:0] 设置为相同的值，以便为两个 DAC 通道配置相同的触发源。
- 将两个 DAC 通道的 WAVEx[1:0] 设置为“01”，并在 MAMP1[3:0] 和 MAMP2[3:0] 位中设置不同的 LFSR 掩码值。
- 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD)。

触发信号到达时，LFSR1 计数器内容（使用 MAMP1[3:0] 配置的掩码）与 DHR1 寄存器内容相加，所得总和转移到 DAC_DOR1 中（三个 APB1 时钟周期之后)。LFSR1 计数器随即更新。

同时，LFSR2 计数器内容（使用 MAMP2[3:0] 配置的掩码）与 DHR2 寄存器内容相加，所得总和转移到

DAC_DOR2 中（三个 APB1 时钟周期之后）。 LFSR2 计数器随即更新。

### 13.4.10. 同步触发（生成单个三角波）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

- 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1。
- 将 TSEL1[2:0] 和 TSEL2[2:0] 设置为相同的值，以便为两个 DAC 通道配置相同的触发源。
- 将两个 DAC 通道的 WAVEx[1:0] 设置为“1x”，并在 MAMPx[3:0] 位中配置相同的最大振幅值。
- 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD）。

触发信号到达时，DAC 通道 1 三角波计数器内容（使用相同的三角波振幅）与 DHR1 寄存器内容相加，所得总和转移到 DAC_DOR1 中（三个 APB1 时钟周期之后）。DAC 通道 1 三角波计数器随即更新。

同时，DAC 通道 2 三角波计数器内容（使用相同的三角波振幅）与 DHR2 寄存器内容相加，所得总和转移到 DAC_DOR2 中（三个 APB1 时钟周期之后）。DAC 通道 2 三角波计数器随即更新

### 13.4.11. 同步触发（生成不同三角波）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

- 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1。
- 将 TSEL1[2:0] 和 TSEL2[2:0] 设置为相同的值，以便为两个 DAC 通道配置相同的触发源。
- 将两个 DAC 通道的 WAVEx[1:0] 设置为“1x”，并在 MAMP1[3:0] 和 MAMP2[3:0] 位中设置不同的最大振幅值。
- 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD）。

触发信号到达时，DAC 通道 1 三角波计数器内容（使用 MAMP1[3:0] 配置的三角波振幅）与 DHR1 寄存器内容相加，所得总和转移到 DAC_DOR1 中（三个 APB1 时钟周期之后）。DAC 通道 1 三角波计数器随即更新。

同时，DAC 通道 2 三角波计数器内容（使用 MAMP2[3:0] 配置的三角波振幅）与 DHR2 寄存器内容相加，所得总和转移到 DAC_DOR2 中（三个 APB1 时钟周期之后）。DAC 通道 2 三角波计数器随即更新。

## 13.5. DAC 寄存器

有关寄存器说明中使用的缩写，请参见第 1.1 节。

外设寄存器必须按字（32 位）进行写访问。

### 13.5.1. DAC 控制寄存器（DAC_CR）

偏移地址： 0x00

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | DMAUDRIE2 | DMAEN2 | MAMP2[3:0] | | | | WAVE2[1:0] | | TSEL2[2:0] | | | TEN2 | BOFF2 | EN2 |
| | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | DMAUDRIE1 | DMAEN1 | MAMP1[3:0] | | | | WAVE1[1:0] | | TSEL1[2:0] | | | TEN1 | BOFF1 | EN1 |
| | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:30 | 保留 | 必须保持复位值。 |
| 29 | DMAUDRIE2 | DAC 2 通道 DMA 下溢中断使能<br>此位由软件置 1 和清零。<br>0：禁止 DAC 2 通道 DMA 下溢中断<br>1：使能 DAC 2 通道 DMA 下溢中断 |
| 28 | DMAEN2 | DAC 2 通道 DMA 使能<br>此位由软件置 1 和清零。<br>0：禁止 DAC 2 通道 DMA 模式<br>1：使能 DAC 2 通道 DMA 模式 |
| 27:24 | MAMP2[3:0] | DAC 2 通道掩码/振幅选择器<br>这些位由软件写入，用于在生成噪声波模式下选择掩码，或者在生成三角波模式下选择振幅。<br>0000：不屏蔽 LFSR 的位 0/三角波振幅等于 1<br>0001：不屏蔽 LFSR 的位 [1:0]/三角波振幅等于 3<br>0010：不屏蔽 LFSR 的位 [2:0]/三角波振幅等于 7<br>0011：不屏蔽 LFSR 的位 [3:0]/三角波振幅等于 15<br>0100：不屏蔽 LFSR 的位 [4:0]/三角波振幅等于 31<br>0101：不屏蔽 LFSR 的位 [5:0]/三角波振幅等于 63<br>0110：不屏蔽 LFSR 的位 [6:0]/三角波振幅等于 127<br>0111：不屏蔽 LFSR 的位 [7:0]/三角波振幅等于 255<br>1000：不屏蔽 LFSR 的位 [8:0]/三角波振幅等于 511<br>1001：不屏蔽 LFSR 的位 [9:0]/三角波振幅等于 1023<br>1010：不屏蔽 LFSR 的位 [10:0]/三角波振幅等于 2047<br>1011：不屏蔽 LFSR 的位 [11:0]/三角波振幅等于 4095 |
| 23:22 | WAVE2[1:0] | DAC 2 通道噪声/三角波生成使能<br>这些位由软件置 1 或清零。<br>00：禁止生成波<br>01：使能生成噪声波<br>1x：使能生成三角波<br>*注意：只在位 TEN2 = 1（使能 DAC 2 通道触发）时使用* |
| 21:19 | TSEL2[2:0] | DAC 2 通道触发器选择<br>这些位用于选择 DAC 2 通道的外部触发事件<br>000：定时器 6 TRGO 事件<br>001：定时器 8 TRGO 事件<br>010：定时器 7 TRGO 事件<br>011：定时器 5 TRGO 事件<br>100：定时器 2 TRGO 事件<br>101：定时器 4 TRGO 事件 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 110：外部中断线 9<br>111：软件触发<br>*注意： 只在位 TEN2 = 1（使能 DAC 2 通道触发）时使用。* |
| 18 | TEN2 | DAC 2 通道触发使能<br>此位由软件置 1 和清零，以使能/禁止 DAC 2 通道触发<br>0: 禁止 DAC 2 通道触发，写入 DAC_DHRx 寄存器的数据在一个 APB1 时钟周期之后转移到 DAC_DOR2 寄存器<br>1: 使能 DAC 2 通道触发，DAC_DHRx 寄存器的数据在三个 APB1 时钟周期之后转移到 DAC_DOR2 寄存器<br>*注意： 如果选择软件触发，DAC_DHRx 寄存器的内容只需一个 APB1 时钟周期即可转移到 DAC_DOR2 寄存器。* |
| 17 | BOFF2 | DAC 2 通道输出缓冲器禁止<br>此位由软件置 1 和清零，以使能/禁止 DAC 2 通道输出缓冲器。<br>0：使能 DAC 2 通道输出缓冲器<br>1：禁止 DAC 2 通道输出缓冲器 |
| 16 | EN2 | DAC 2 通道使能<br>此位由软件置 1 和清零，以使能/禁止 DAC 2 通道。<br>0：禁止 DAC 2 通道<br>1：使能 DAC 2 通道 |
| 15:14 | 保留 | 必须保持复位值。 |
| 13 | DMAUDRIE1 | DAC 1 通道 DMA 下溢中断使能<br>此位由软件置 1 和清零。<br>0：禁止 DAC 1 通道 DMA 下溢中断<br>1：使能 DAC 1 通道 DMA 下溢中断 |
| 12 | DMAEN1 | DAC 1 通道 DMA 使能<br>此位由软件置 1 和清零。<br>0：禁止 DAC 1 通道 DMA 模式<br>1：使能 DAC 1 通道 DMA 模式 |
| 11:8 | MAMP1[3:0] | DAC 1 通道掩码/振幅选择器<br>这些位由软件写入，用于在生成噪声波模式下选择掩码，或者在生成三角波模式下选择振幅。<br>0000：不屏蔽 LFSR 的位 0/三角波振幅等于 1<br>0001：不屏蔽 LFSR 的位 [1:0]/三角波振幅等于 3<br>0010：不屏蔽 LFSR 的位 [2:0]/三角波振幅等于 7<br>0011：不屏蔽 LFSR 的位 [3:0]/三角波振幅等于 15<br>0100：不屏蔽 LFSR 的位 [4:0]/三角波振幅等于 31<br>0101：不屏蔽 LFSR 的位 [5:0]/三角波振幅等于 63<br>0110：不屏蔽 LFSR 的位 [6:0]/三角波振幅等于 127<br>0111：不屏蔽 LFSR 的位 [7:0]/三角波振幅等于 255<br>1000：不屏蔽 LFSR 的位 [8:0]/三角波振幅等于 511<br>1001：不屏蔽 LFSR 的位 [9:0]/三角波振幅等于 1023<br>1010：不屏蔽 LFSR 的位 [10:0]/三角波振幅等于 2047<br>1011：不屏蔽 LFSR 的位 [11:0]/三角波振幅等于 4095 |
| 7:6 | WAVE1[1:0] | DAC 1 通道噪声/三角波生成使能<br>这些位将由软件置 1 和清零。<br>00：禁止生成波<br>01：使能生成噪声波<br>1x：使能生成三角波 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | *注意： 只在位 TEN1 = 1（使能 DAC 1 通道触发）时使用。* |
| 5:3 | TSEL1[2:0] | DAC 1 通道触发器选择<br>这些位用于选择 DAC 1 通道的外部触发事件。<br>000：定时器 6 TRGO 事件<br>001：定时器 8 TRGO 事件<br>010：定时器 7 TRGO 事件<br>011：定时器 5 TRGO 事件<br>100：定时器 2 TRGO 事件<br>101：定时器 4 TRGO 事件<br>110：外部中断线 9<br>111：软件触发<br>*注意： 只在位 TEN1 = 1（使能 DAC 1 通道触发）时使用。* |
| 2 | TEN1 | DAC 1 通道触发使能<br>此位由软件置 1 和清零，以使能/禁止 DAC 1 通道触发。<br>0: 禁止 DAC 1 通道触发，写入 DAC_DHRx 寄存器的数据在一个 APB1 时钟周期之后转移到 DAC_DOR1 寄存器<br>1: 使能 DAC 1 通道触发，DAC_DHRx 寄存器的数据在三个 APB1 时钟周期之后转移到 DAC_DOR1 寄存器<br>*注意：如果选择软件触发，DAC_DHRx 寄存器的内容只需一个 APB1 时钟周期即可转移到 DAC_DOR1 寄存器。* |
| 1 | BOFF1 | DAC 1 通道输出缓冲器禁止<br>此位由软件置 1 和清零，以使能/禁止 DAC 1 通道输出缓冲器。<br>0：使能 DAC 1 通道输出缓冲器<br>1：禁止 DAC 1 通道输出缓冲器 |
| 0 | EN1 | DAC 1 通道使能<br>此位由软件置 1 和清零，以使能/禁止 DAC 1 通道。<br>0：禁止 DAC 1 通道<br>1：使能 DAC 1 通道 |

### 13.5.2. DAC 软件触发寄存器 (DAC_SWTRIGR)

偏移地址： 0x04

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | | | | | | | | SWTRIG2 | SWTRIG1 |
| | | | | | | | | | | | | | | w | w |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:2 | 保留 | 必须保持复位值。 |
| 1 | SWTRIG2 | DAC 2 通道软件触发<br>此位由软件置 1 和清零，以使能/禁止软件触发。<br>0：禁止软件触发<br>1：使能软件触发<br>*注意： 一旦 DAC_DHR2 寄存器值加载到 DAC_DOR2 寄存器中，该位即会由硬件清零（一个 APB1 时钟周期之后）。* |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 0 | SWTRIG1 | DAC 1 通道软件触发<br>此位由软件置 1 和清零，以使能/禁止软件触发。<br>0：禁止软件触发<br>1：使能软件触发<br>*注意： 一旦 DAC_DHR1 寄存器值加载到 DAC_DOR1 寄存器中，该位即会由硬件清零（一个 APB1 时钟周期之后）。* |

### 13.5.3. DAC 1 通道 12 位右对齐数据保持寄存器 (DAC_DHR12R1)

偏移地址： 0x08

复位值： 0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | DACC1DHR[11:0] | | | | | | | | | | | |
| | | | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:12 | 保留 | 必须保持复位值。 |
| 11:0 | DACC1DHR[11:0] | DAC 1 通道 12 位右对齐数据<br>这些位由软件写入，用于为 DAC 1 通道指定 12 位数据。 |

### 13.5.4. DAC 1 通道 12 位左对齐数据保持寄存器 (DAC_DHR12L1)

偏移地址： 0x0C

复位值： 0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DACC1DHR[11:0] | | | | | | | | | | | | Reserved | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15:4 | DACC1DHR[11:0] | DAC 1 通道 12 位左对齐数据<br>这些位由软件写入，用于为 DAC 1 通道指定 12 位数据。 |
| 3:0 | 保留 | 必须保持复位值。 |

### 13.5.5. DAC 1 通道 8 位右对齐数据保持寄存器 (DAC_DHR8R1)

偏移地址： 0x10

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | | DACC1DHR[7:0] | | | | | | | |
| | | | | | | | | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 必须保持复位值。 |
| 7:0 | DACC1DHR[7:0] | DAC 1 通道 8 位右对齐数据<br>这些位由软件写入，用于为 DAC 1 通道指定 8 位数据。 |

### 13.5.6. DAC 2 通道 12 位右对齐数据保持寄存器 (DAC_DHR12R2)

偏移地址： 0x14

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | DACC2DHR[11:0] | | | | | | | | | | | |
| | | | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:12 | 保留 | 必须保持复位值。 |
| 11:0 | DACC2DHR[11:0] | DAC 2 通道 12 位右对齐数据<br>这些位由软件写入，用于为 DAC 2 通道指定 12 位数据。 |

### 13.5.7. DAC 2 通道 12 位左对齐数据保持寄存器 (DAC_DHR12L2)

偏移地址： 0x18

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DACC2DHR[11:0] | | | | | | | | | | | | Reserved | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15:4 | DACC2DHR[11:0] | DAC 2 通道 12 位左对齐数据<br>这些位由软件写入，用于为 DAC 2 通道指定 12 位数据。 |
| 3:0 | 保留 | 必须保持复位值。 |

### 13.5.8. DAC 2 通道 8 位右对齐数据保持寄存器 (DAC_DHR8R2)

偏移地址： 0x1C

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | | DACC2DHR[7:0] | | | | | | | |
| | | | | | | | | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 必须保持复位值。 |
| 7:0 | DACC2DHR[7:0] | DAC 2 通道 8 位右对齐数据<br>这些位由软件写入，用于为 DAC 2 通道指定 8 位数据。 |

### 13.5.9. 双 DAC 12 位右对齐数据保持寄存器 (DAC_DHR12RD)

偏移地址： 0x20

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | DACC2DHR[11:0] | | | | | | | | | | | |
| | | | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | DACC1DHR[11:0] | | | | | | | | | | | |
| | | | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:28 | 保留 | 必须保持复位值。 |
| 27:16 | DACC2DHR[11:0] | DAC 2 通道 12 位右对齐数据<br>这些位由软件写入，用于为 DAC 2 通道指定 12 位数据。 |
| 15:12 | 保留 | 必须保持复位值。 |
| 11:0 | DACC1DHR[11:0] | DAC 1 通道 12 位右对齐数据<br>这些位由软件写入，用于为 DAC 1 通道指定 12 位数据。 |

### 13.5.10. 双 DAC 12 位左对齐数据保持寄存器 (DAC_DHR12LD)

偏移地址： 0x24

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DACC2DHR[11:0] | | | | | | | | | | | | Reserved | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DACC1DHR[11:0] | | | | | | | | | | | | Reserved | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:20 | DACC2DHR[11:0] | DAC 2 通道 12 位左对齐数据 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 这些位由软件写入，用于为 DAC 2 通道指定 12 位数据。 |
| 19:16 | 保留 | 必须保持复位值。 |
| 15:4 | DACC1DHR[11:0] | DAC 1 通道 12 位左对齐数据<br>这些位由软件写入，用于为 DAC 1 通道指定 12 位数据。 |
| 3:0 | 保留 | 必须保持复位值。 |

### 13.5.11. 双 DAC 8 位右对齐数据保持寄存器 (DAC_DHR8RD)

偏移地址： 0x28

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DACC2DHR[7:0] | | | | | | | | DACC1DHR[7:0] | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15:8 | DACC2DHR[7:0] | DAC 2 通道 8 位右对齐数据<br>这些位由软件写入，用于为 DAC 2 通道指定 8 位数据。 |
| 7:0 | DACC1DHR[7:0] | DAC 1 通道 8 位右对齐数据<br>这些位由软件写入，用于为 DAC 1 通道指定 8 位数据。 |

### 13.5.12. DAC 1 通道数据输出寄存器 (DAC_DOR1)

偏移地址： 0x2C

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | DACC1DOR[11:0] | | | | | | | | | | | |
| | | | | r | r | r | r | r | r | r | r | r | r | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:12 | 保留 | 必须保持复位值。 |
| 11:0 | DACC1DOR[11:0] | DAC 1 通道数据输出<br>这些位为只读，其中包含 DAC 1 通道的数据输出。 |

### 13.5.13. DAC 2 通道数据输出寄存器 (DAC_DOR2)

偏移地址： 0x30

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | DACC2DOR[11:0] | | | | | | | | | | | |
| | | | | r | r | r | r | r | r | r | r | r | r | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:12 | 保留 | 必须保持复位值。 |
| 11:0 | DACC2DOR[11:0] | DAC 2 通道数据输出<br>这些位为只读，其中包含 DAC 2 通道的数据输出。 |

### 13.5.14. DAC 状态寄存器 (DAC_SR)

偏移地址： 0x34

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DACC2DHR[11:0] | | DMAUDR2 | Reserved | | | | | | | | | | | | |
| | | rc_w1 | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DACC1DHR[11:0] | | DMAUDR1 | Reserved | | | | | | | | | | | | |
| | | rc_w1 | | | | | | | | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:30 | 保留 | 必须保持复位值。 |
| 29 | DMAUDR2 | DAC 2 通道 DMA 下溢标志<br>此位由硬件置 1，由软件清零（写入 1）。<br>0：DAC 2 通道未发生 DMA 下溢错误状况<br>1：DAC 2 通道发生 DMA 下溢错误状况（当前所选触发源以高于 DMA 服务能力的频率驱动 DAC 2 通道转换） |
| 28:14 | 保留 | 必须保持复位值。 |
| 13 | DMAUDR1 | DAC 1 通道 DMA 下溢标志<br>此位由硬件置 1，由软件清零（写入 1）。<br>0：DAC 1 通道未发生 DMA 下溢错误状况<br>1：DAC 1 通道发生 DMA 下溢错误状况（当前所选触发源以高于 DMA 服务能力的频率驱动 DAC 1 通道转换） |
| 12:0 | 保留 | 必须保持复位值。 |

# 14. 高级控制定时器(TIM1/TIM8)

## 14.1. TIM1/TIM8 简介

高级控制定时器 (TIM1//TIM8) 包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。

此类定时器可用于各种用途，包括测量输入信号的脉冲宽度（输入捕获），或者生成输出波形（输出比较、PWM 和带死区插入的互补 PWM）。

使用定时器预分频器和 RCC 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。

高级控制定时器（TIM1/TIM8）和通用 (TIMx) 定时器彼此完全独立，不共享任何资源，它们可以同步操作。

## 14.2. TIM1/TIM8 主要特性

- 16 位递增、递减、递增/递减自动重载计数器。
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 到 65536 之间。
- 多达 6 个独立通道，可用于：
  - 输入捕获（但通道 5 和通道 6 除外）；
  - 输出比较；
  - PWM 生成（边沿和中心对齐模式）；
  - 单脉冲模式输出。
- 带可编程死区的互补输出。
- 使用外部信号控制定时器且可实现多个定时器互连的同步电路。
- 重复计数器，用于仅在给定数目的计数器周期后更新定时器寄存器。
- 2 个断路输入，用于将定时器的输出信号置于用户可选的安全配置中。
- 发生如下事件时生成中断/DMA 请求：
  - 更新：计数器上溢/下溢、计数器初始化（通过软件或内部/外部触发）；
  - 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数）；
  - 输入捕获；
  - 输出比较。
- 支持定位用增量（正交）编码器和霍尔传感器电路。
- 触发输入用作外部时钟或逐周期电流管理。

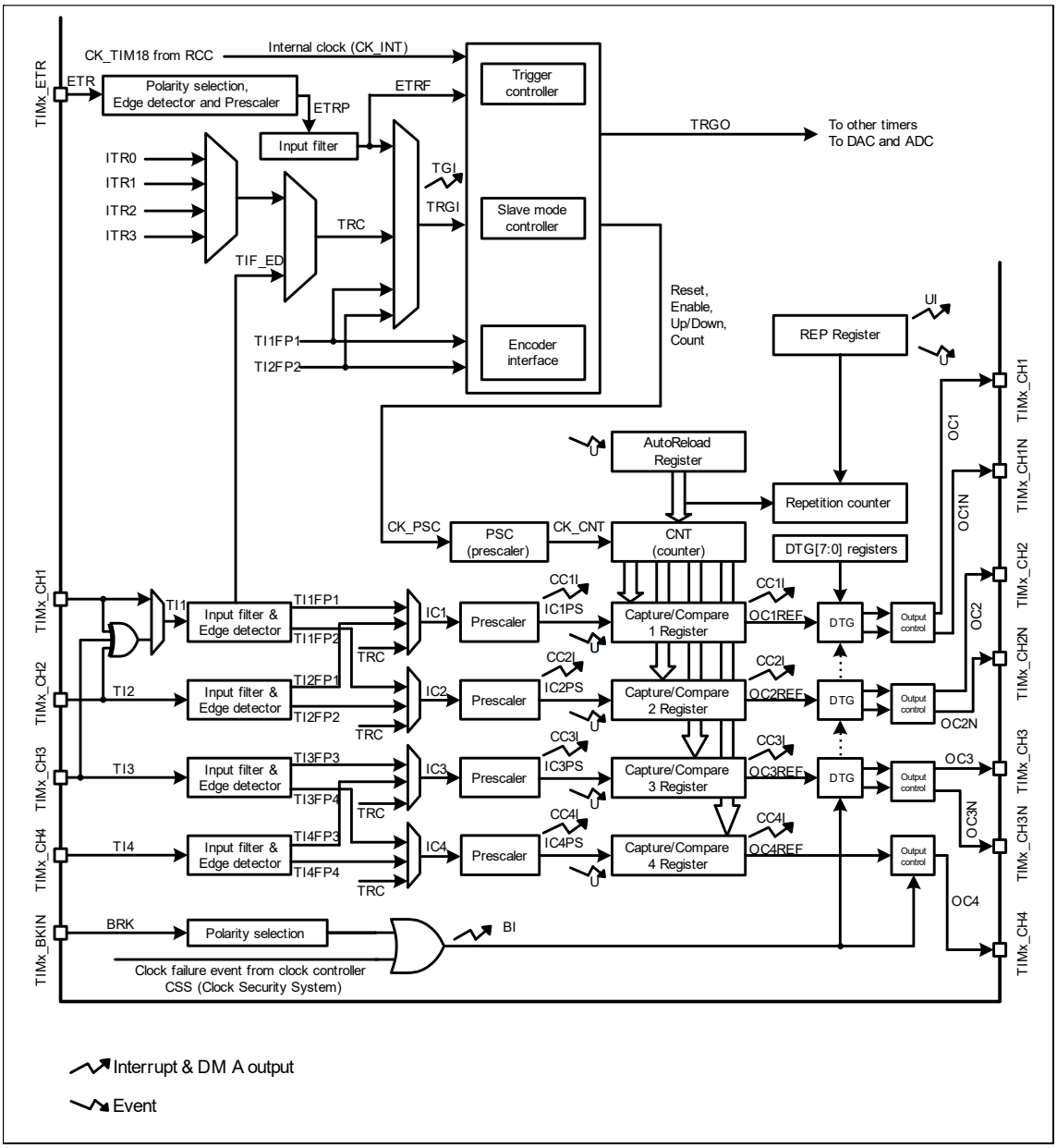



图 14-1 高级控制定时器框图

# 14.3. TIM1/TIM8 功能说明

## 14.3.1. 时基单元

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

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

时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)

- 自动重载寄存器 (TIMx_ARR)
- 重复计数器寄存器 (TIMx_RCR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以直接传送到影子寄存器，也可以在每次发生更新事件 (UEV) 时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值（或者在递减计数时达到下溢值）并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。

*注意，计数器将在 TIMx_CR1 寄存器的 CEN 位置 1 时刻的一个时钟周期后开始计数。*

**预分频器说明**

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

下图以一些示例说明在预分频比实时变化时计数器的行为。

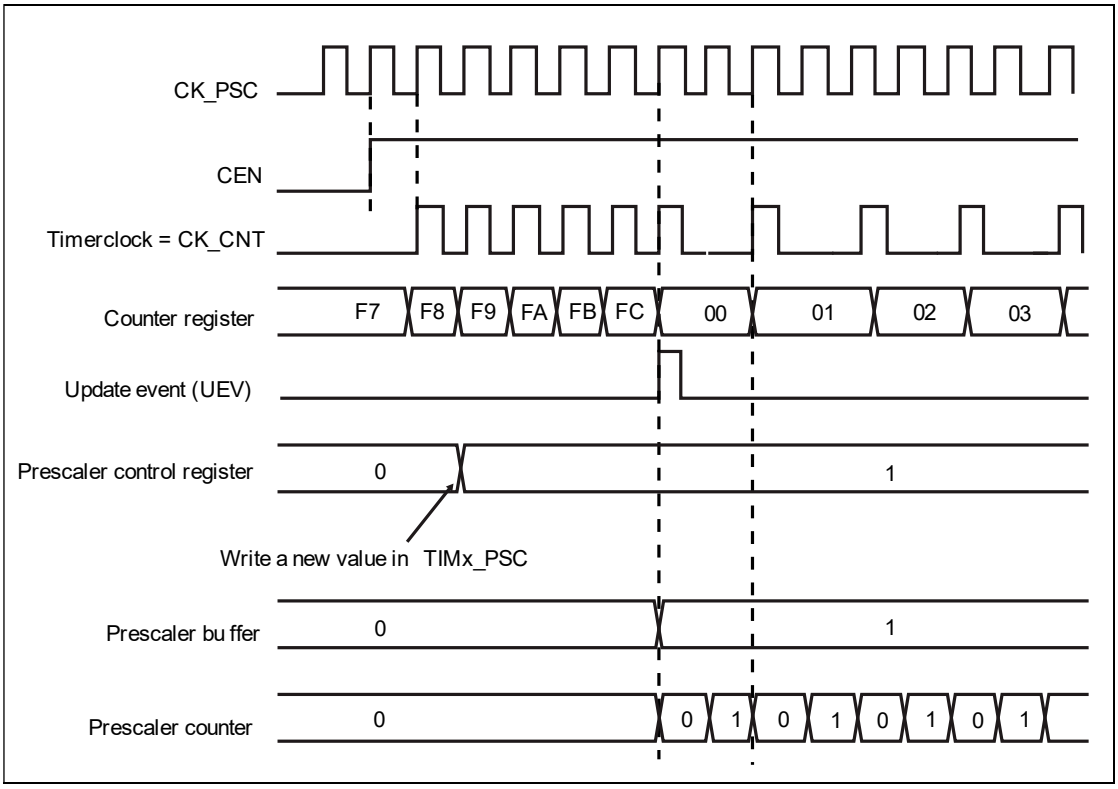



图 14-2 预分频器分频由 1 变为 2 时的计数器时序图

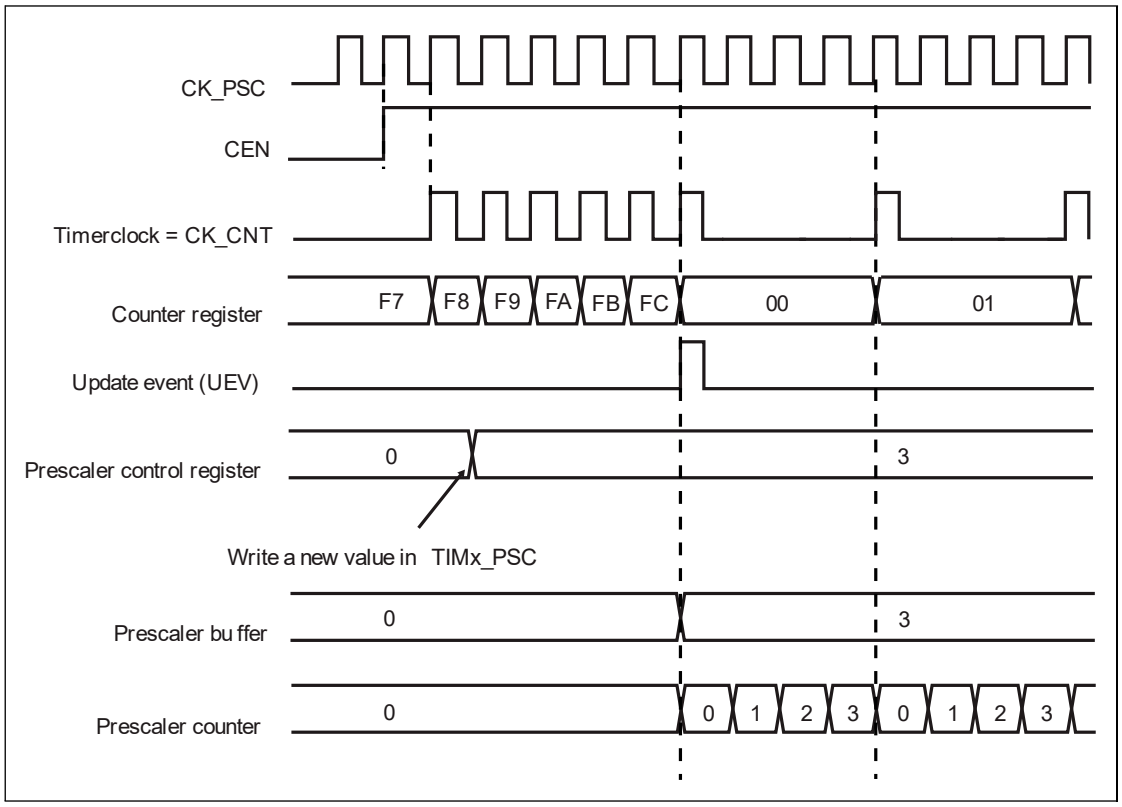



图 14-3 预分频器分频由 1 变为 4 时的计数器时序图

## 14.3.2. 计数器模式

**递增计数模式**

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

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

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

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

- 重复计数器中将重新装载 TIMx_RCR 寄存器的内容；
- 自动重载影子寄存器将以预装载值 (TIMx_ARR) 进行更新；
- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）；
- 以下各图以一些示例说明当 TIMx_ARR=0x36 时不同时钟频率下计数器的行为。

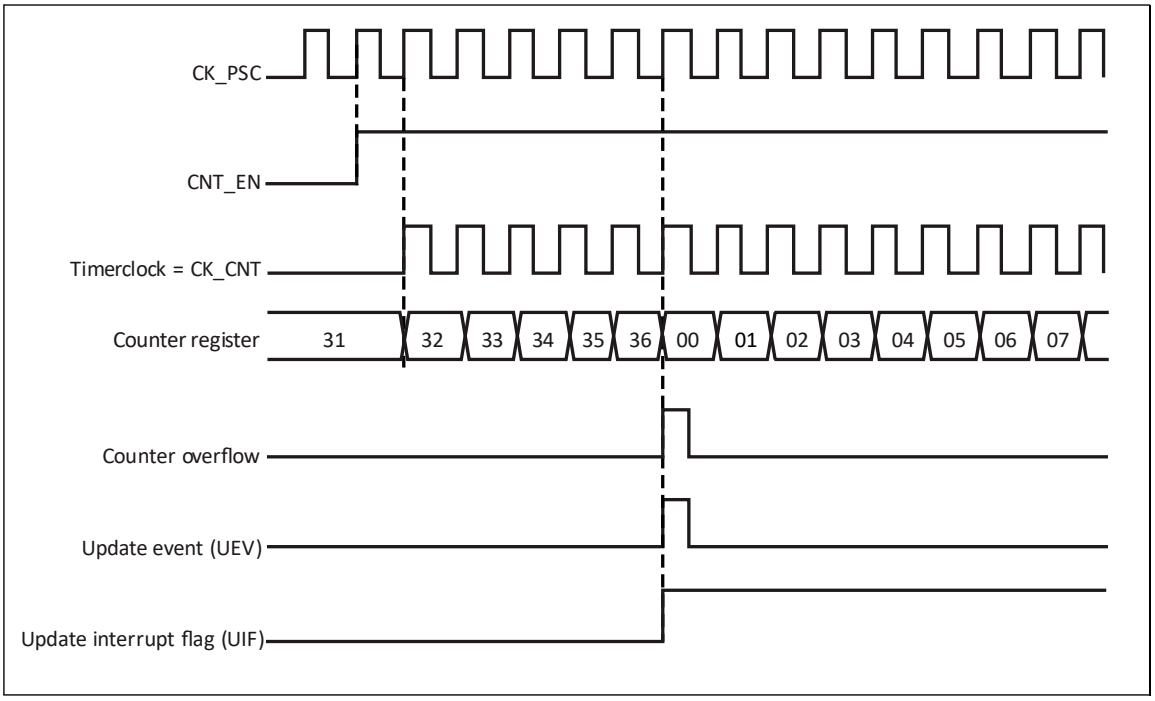



图 14-4 计数器时序图，内部时钟 1 分频

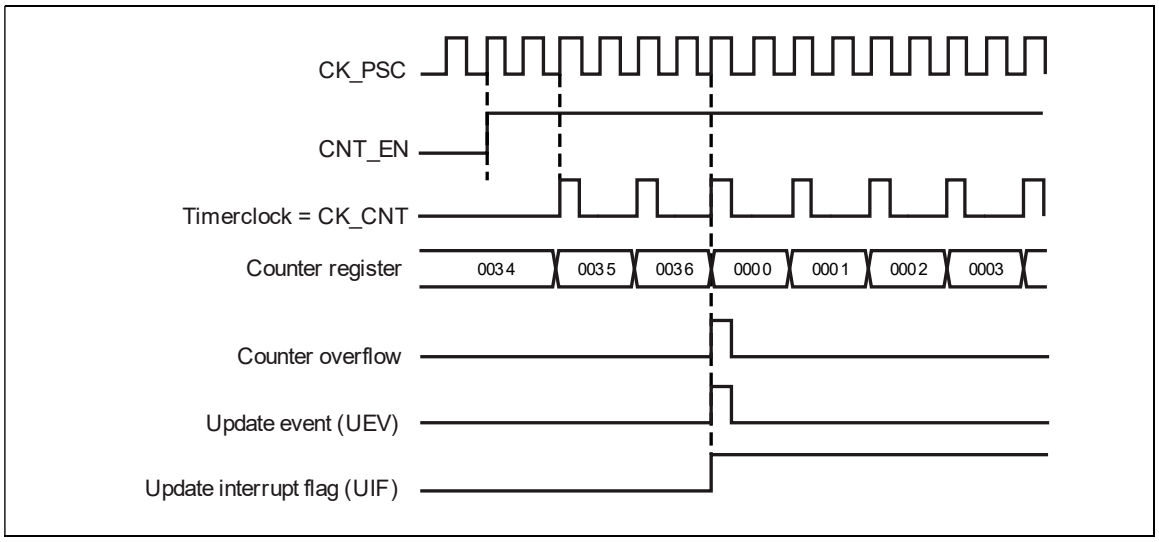



图 14-5 计数器时序图，内部时钟 2 分频

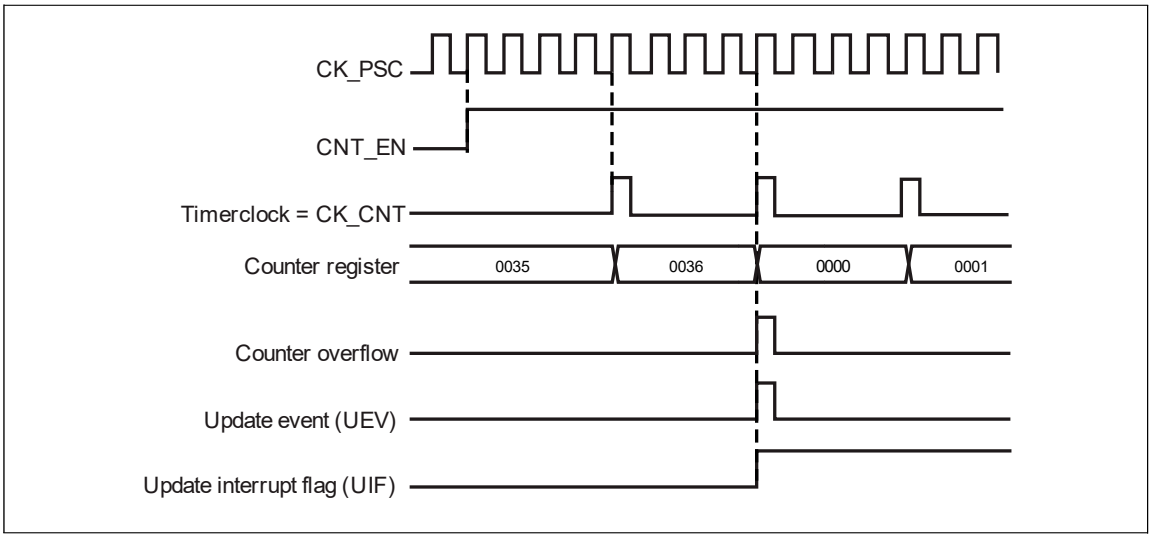



图 14-6 计数器时序图，内部时钟 4 分频

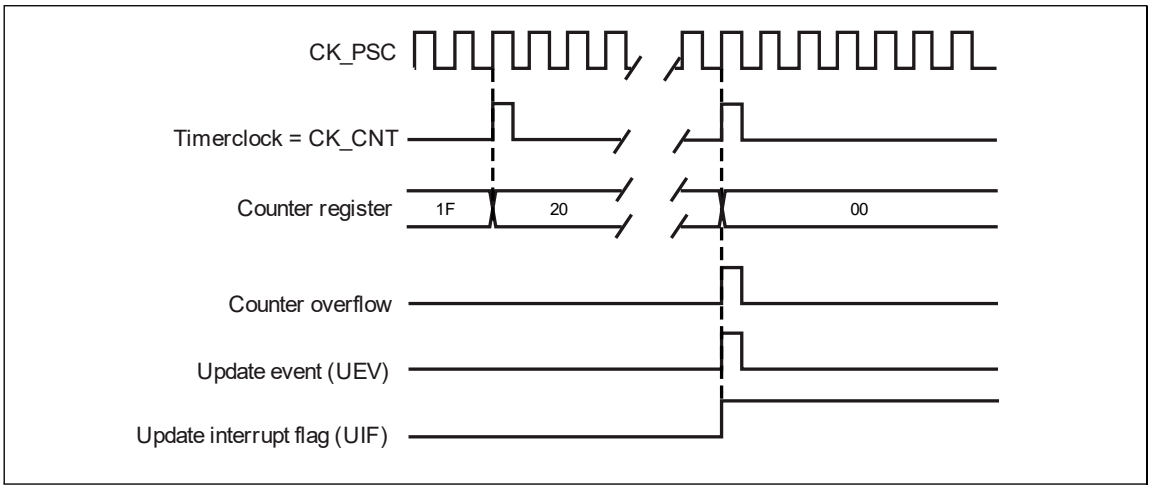



图 14-7 计数器时序图，内部时钟 N 分频

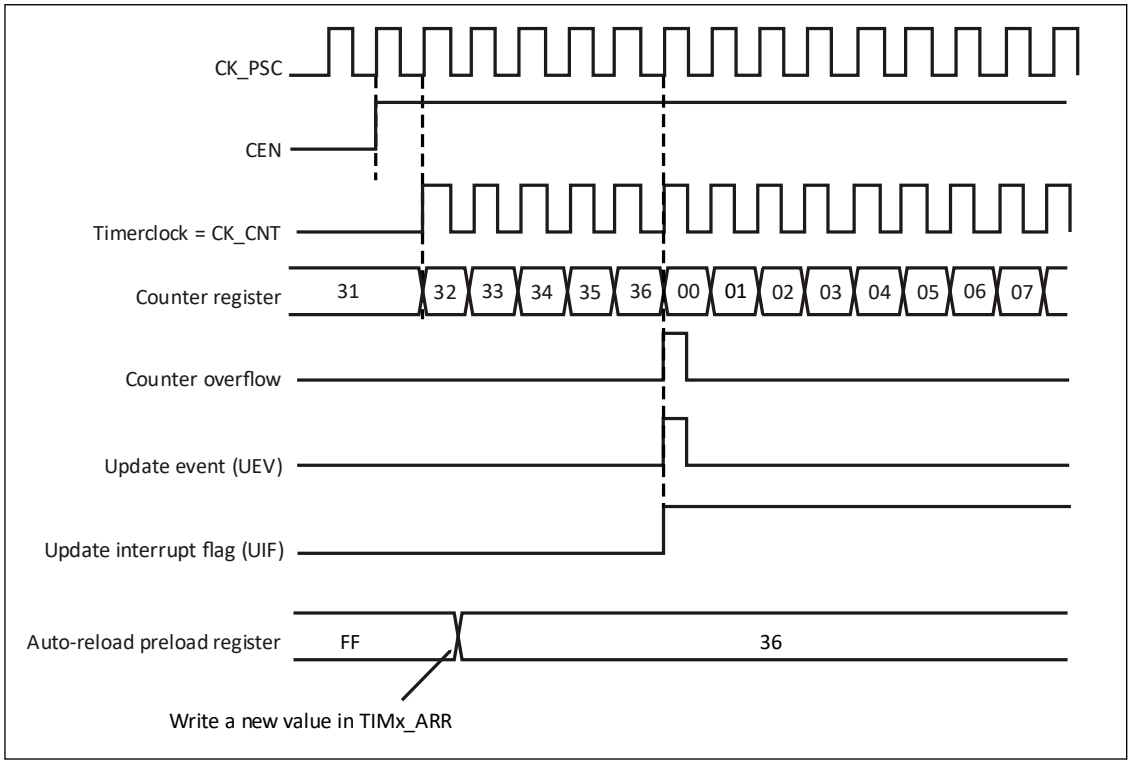



图 14-8 计数器时序图，ARPE=0 时更新事件(TIMx_ARR 未预装载)

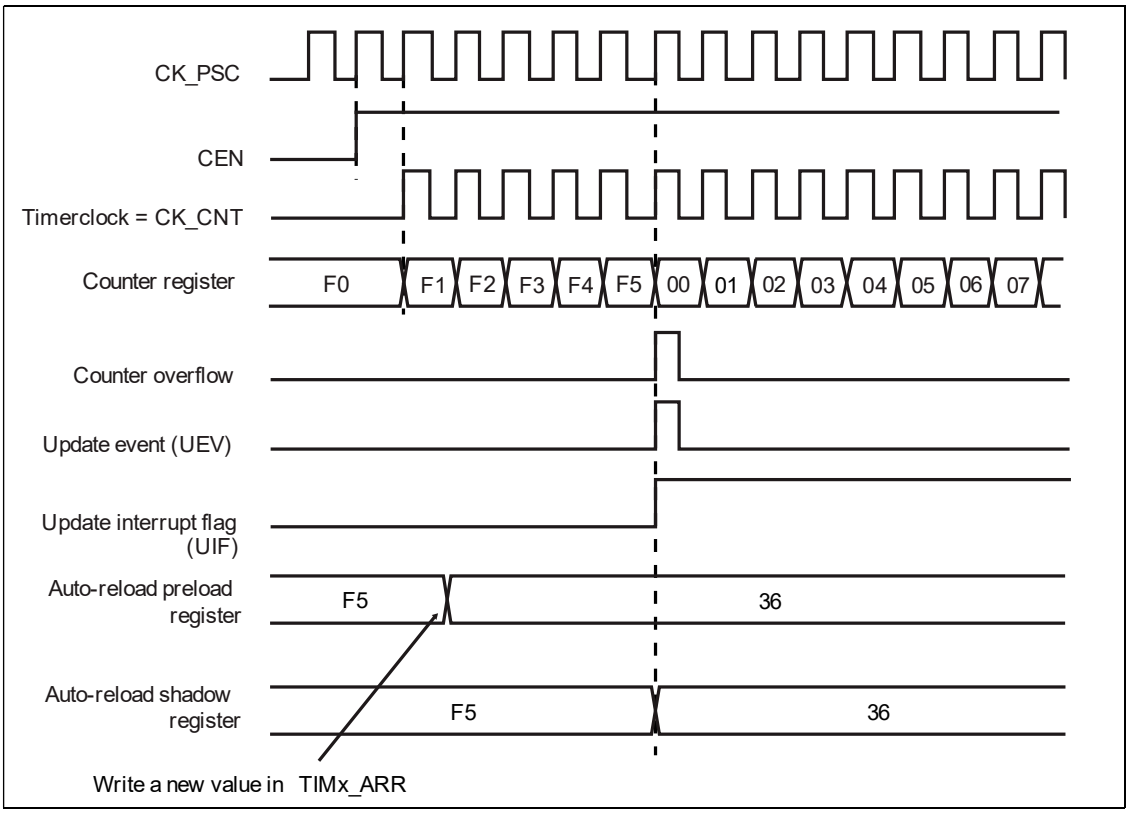



图 14-9 计数器时序图，ARPE=1 时更新事件(TIMx_ARR 预装载)

**递减计数模式**

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

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

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

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器会重新从当前自动重载值开始计数，而预分频器计数器则重新从 0 开始计数（但预分频比保持不变）。

此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

- 重复计数器中将重新装载 TIMx_RCR 寄存器的内容；
- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）；
- 自动重载活动寄存器将以预装载值（TIMx_ARR 寄存器的内容）进行更新。注意，自动重载寄存器会在计数器重载之前被更新，因此下一个周期就是预期的值。

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

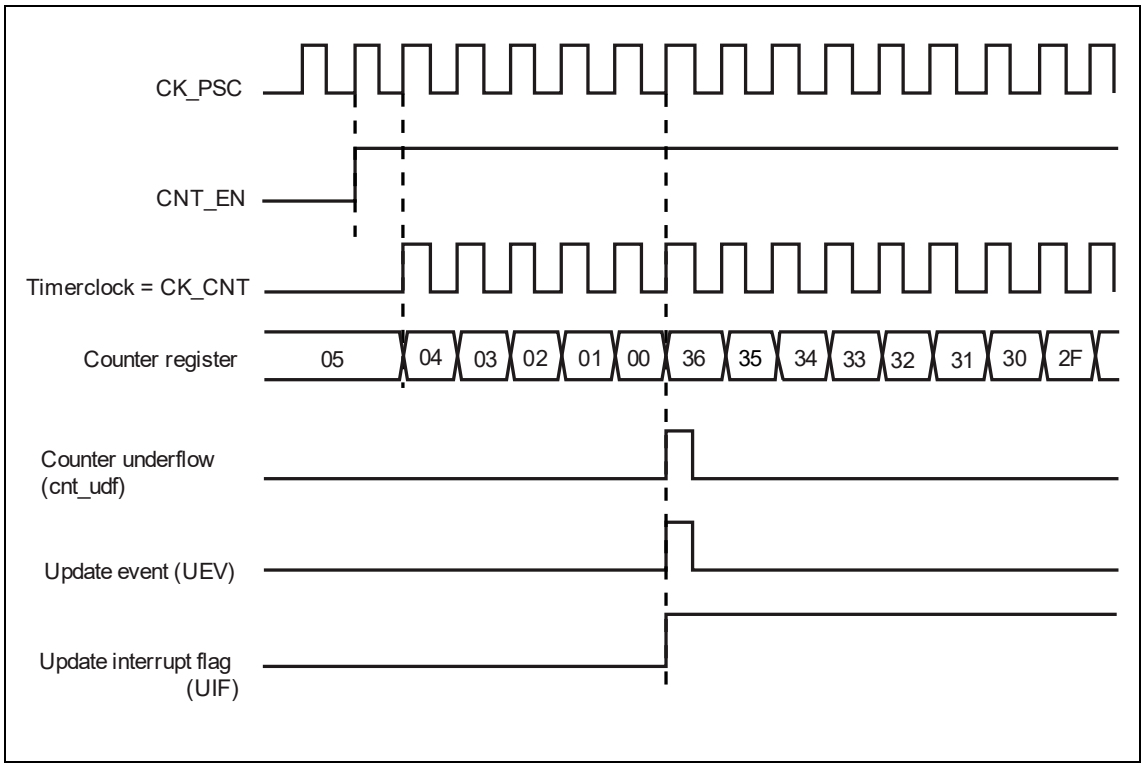



图 14-10 计数器时序图，内部时钟 1 分频

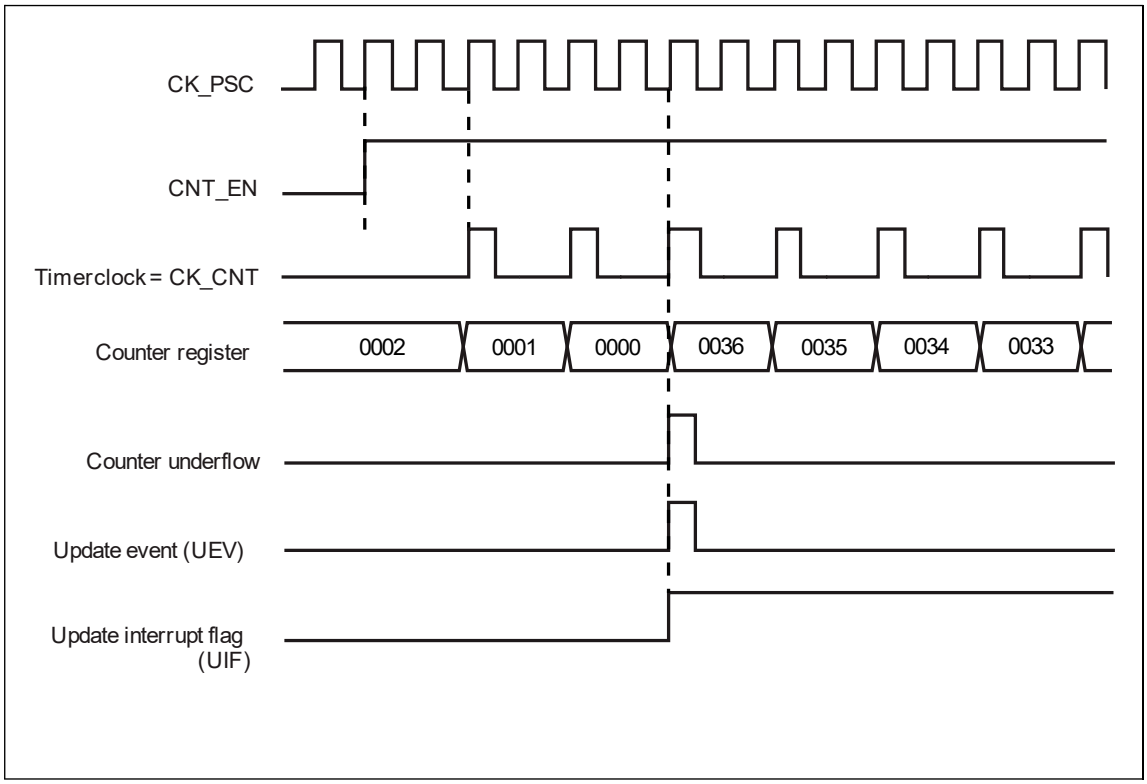



图 14-11 计数器时序图，内部时钟 2 分频

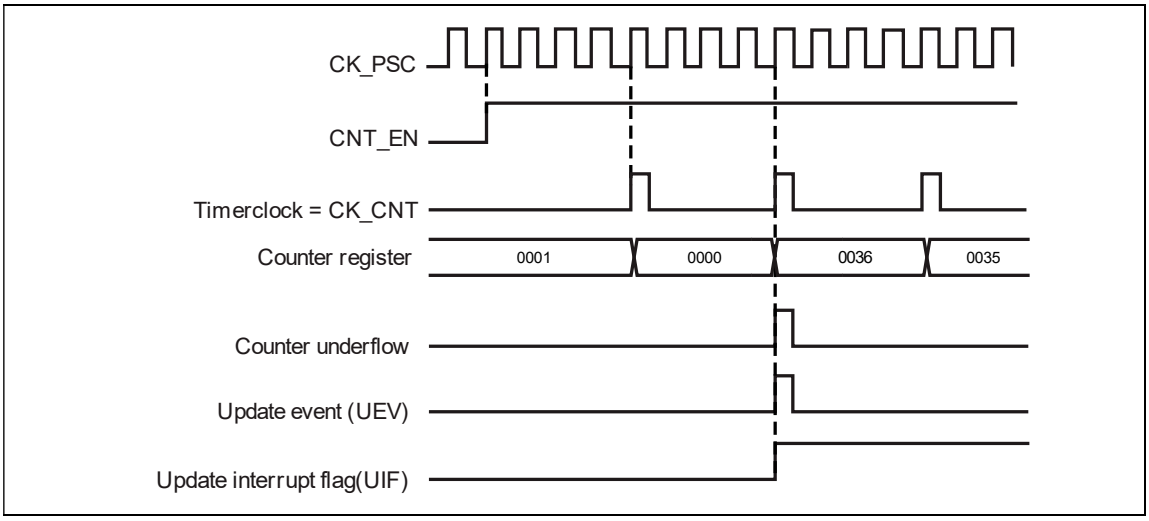



图 14-12 计数器时序图，内部时钟 4 分频

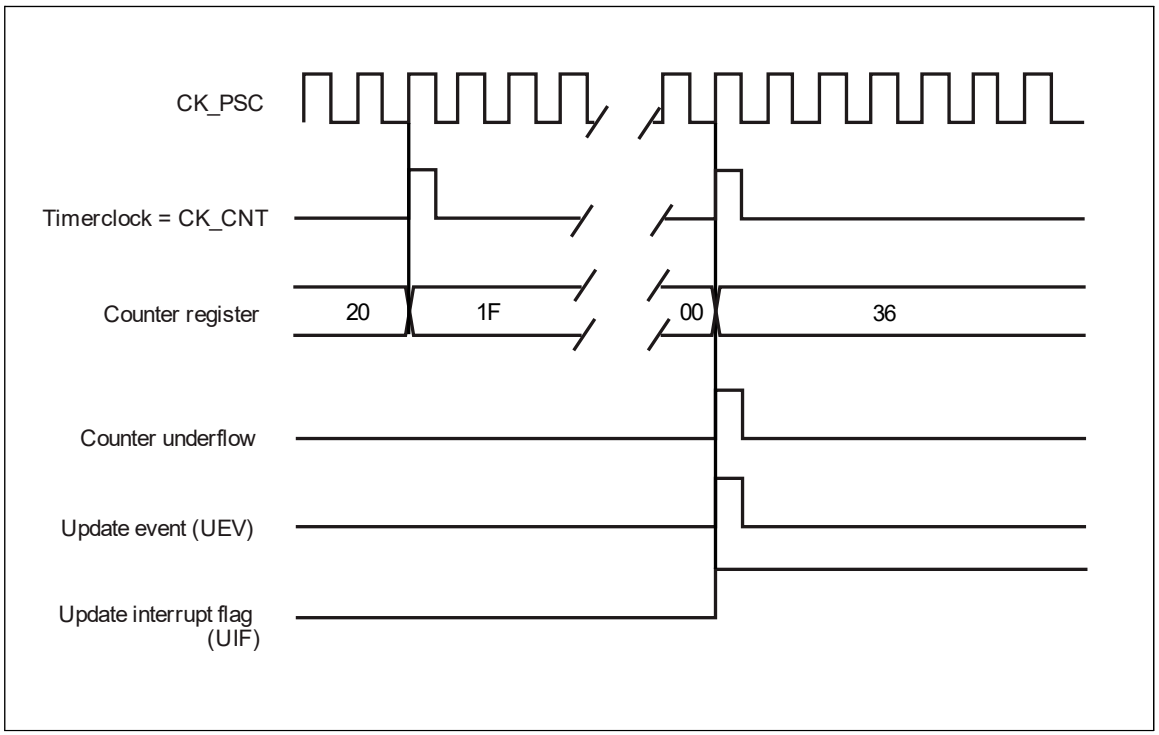



图 14-13 计数器时序图，内部时钟 N 分频

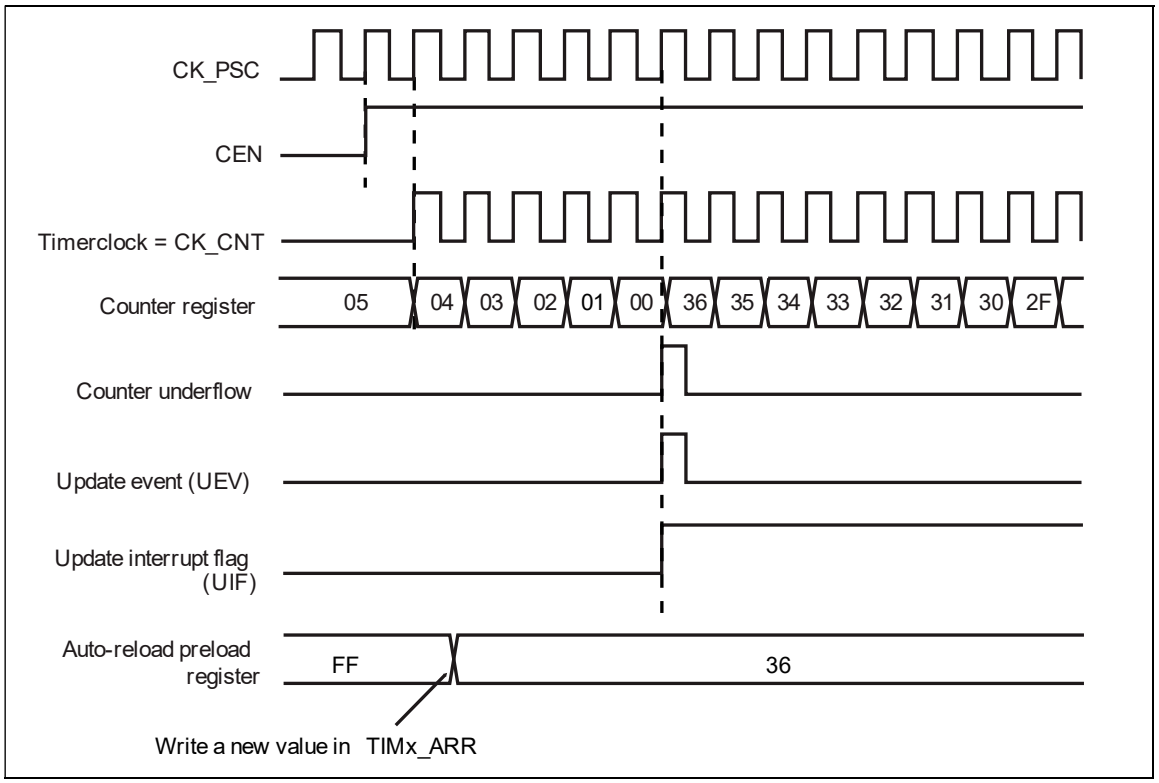



图 14-14 计数器时序图，未使用重复计数器时更新事件

### 中心对齐模式（递增/递减计数）

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

当 TIMx_CR1 寄存器中的 CMS 位不为“00”时，中心对齐模式有效。将通道配置为输出模式时，其输出比较中断标志将在以下模式下置 1，即：计数器递减计数（中心对齐模式 1，CMS =“01”）、计数器递增计数（中心对齐模式 2， CMS =“10”） 以及计数器递增/递减计数（中心对齐模式 3， CMS =“11”）。

在此模式下，TIMx_CR1 寄存器的 DIR 方向位不可写入值，而是由硬件更新并指示当前计数器方向。

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

UEV 更新事件可通过软件禁止，只需将 TIMx_CR1 寄存器中的 UDIS 位置 1。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器仍会根据当前自动重载值进行递增和递减计数。

此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会产生 UEV 更新事件，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

重复计数器中将重新装载 TIMx_RCR 寄存器的内容

预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）

自动重载活动寄存器将以预装载值（TIMx_ARR 寄存器的内容）进行更新。注意，如果更新操作是由计数器上溢触发的，则 ARR 寄存器在计数器重载之前更新，因此，下一个计数周期就是我们所希望的新的周期长度（计数器被重载新的值）。

以下各图以一些示例说明不同时钟频率下计数器的行为。

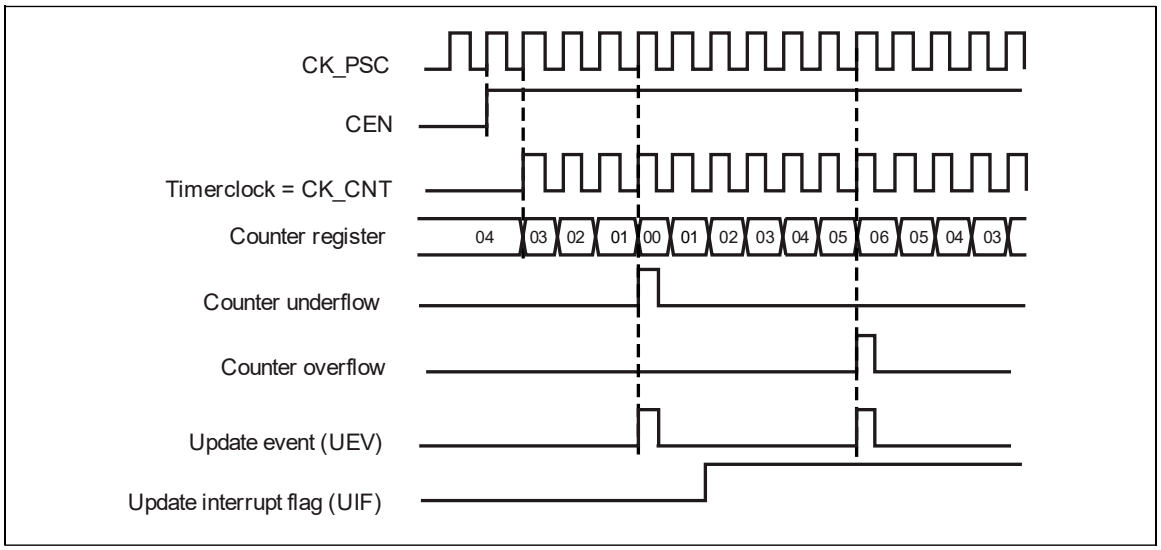



图 14-15 计数器时序图，内部时钟 1 分频，TIMx_ARR = 0x6

*注：此处使用中心对齐模式 1*

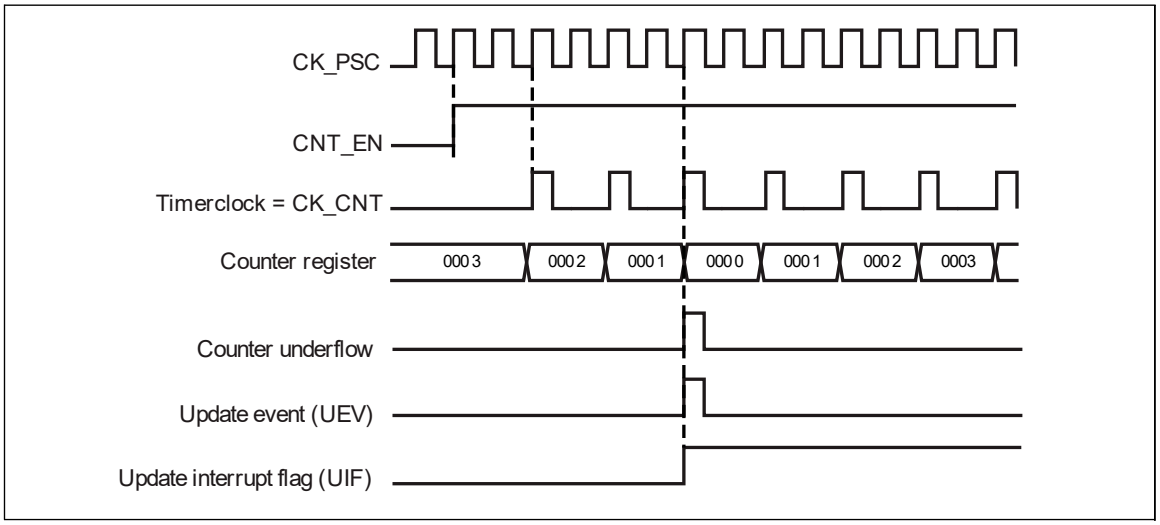



图 14-16 计数器时序图，内部时钟 2 分频

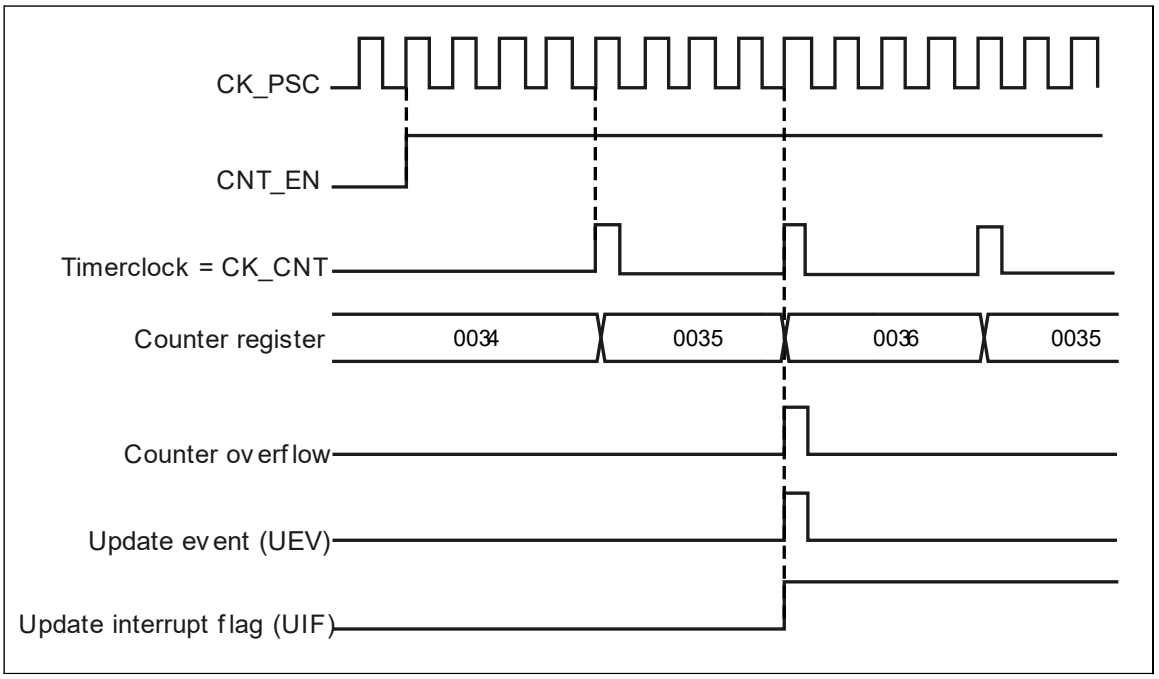



图 14-17 计数器时序图，内部时钟 4 分频，TIMx_ARR = 0x36

*注：中心对齐模式 2 或模式 3 与上溢 UIF 结合使用*

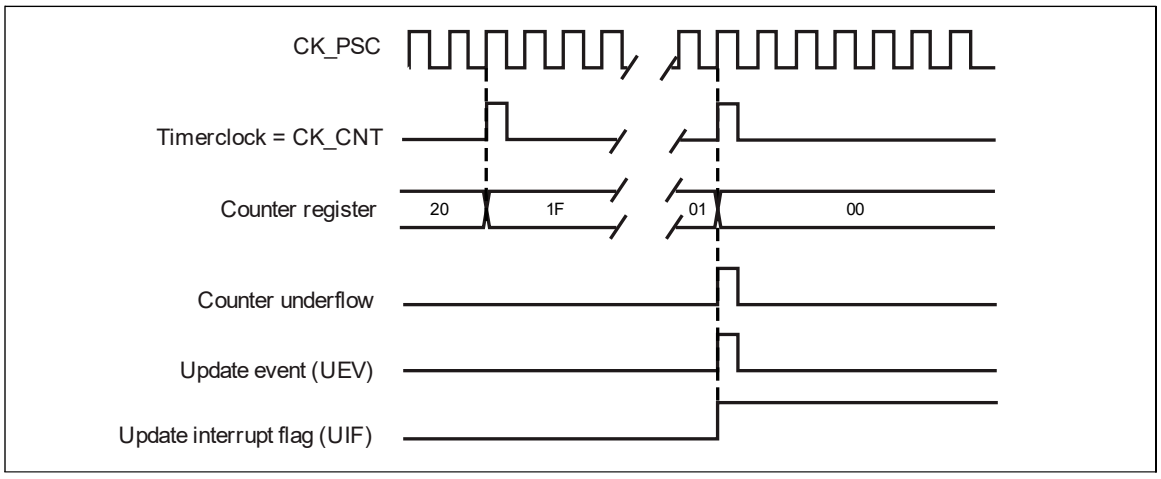



图 14-18 计数器时序图，内部时钟 N 分频

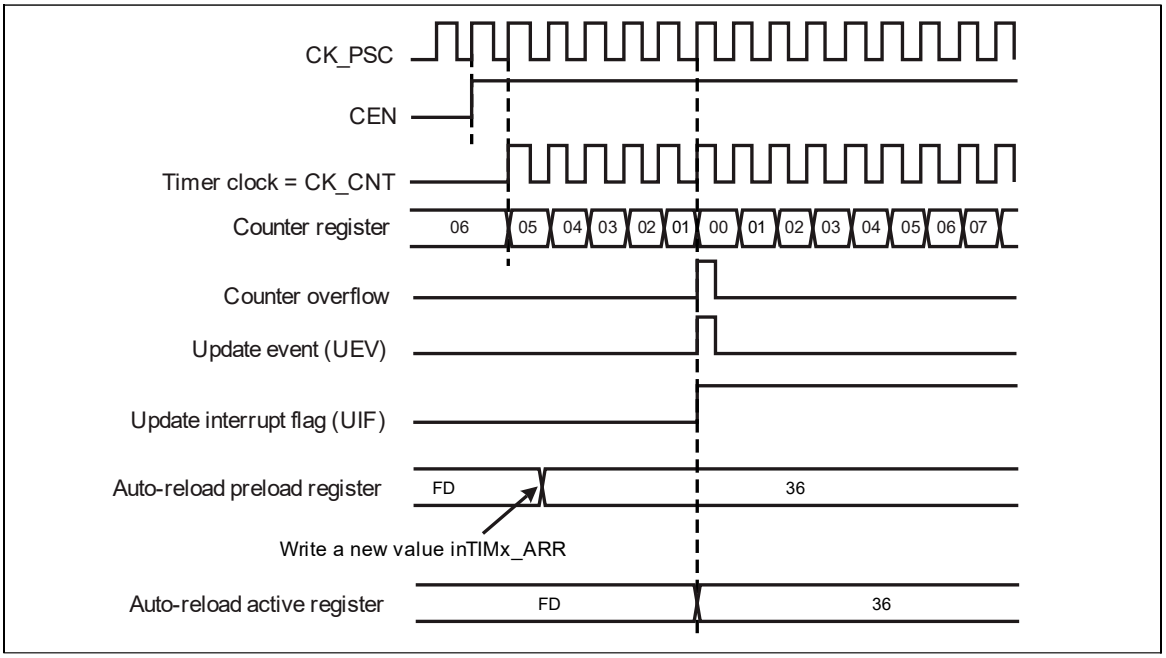



图 14-19 计数器时序图，ARPE=1 时的更新事件(计数器下溢)

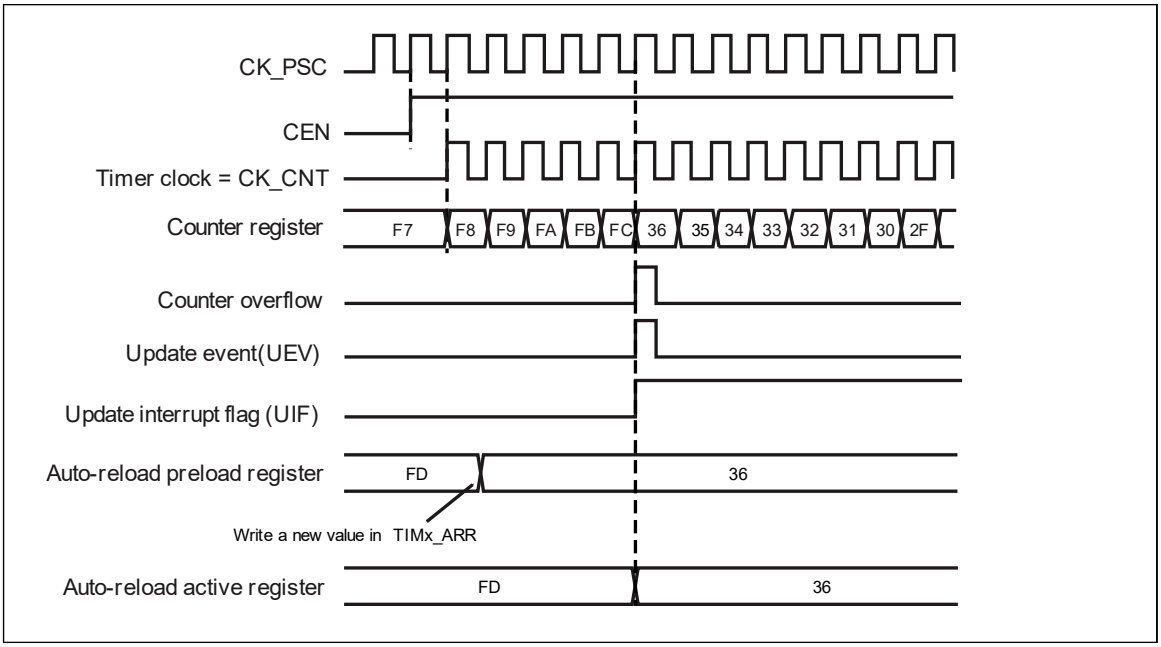



图 14-20 计数器时序图，ARPE=1 时的更新事件(计数器上溢)

### 14.3.3. 重复计数器

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

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

重复计数器在下列情况下递减：

- 递增计数模式下的每个计数器上溢；
- 递减计数模式下的每个计数器下溢；
- 中心对齐模式下每个计数器上溢和计数器下溢。尽管这使得最大重复次数不超过 128 个 PWM 周期，但在每个 PWM 周期内可更新占空比两次。当在中心对齐模式下，每个 PWM 周期仅刷新一次比较寄存器时，由于模式的对称性，最大分辨率为 $2xT_{ck}$。

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

在中心对齐模式下，如果 RCR 值为奇数，更新事件将在上溢或下溢时发生，这取决于何时写入 RCR 寄存器以及何时启动计数器：如果在启动计数器前写入 RCR，则 UEV 在上溢时发生。如果在启动计数器后写入 RCR，则 UEV 在下溢时发生。

例如，如果 RCR = 3 ，UEV 将在每个周期的第四个上溢或下溢事件时产生（取决于何时写入 RCR）。

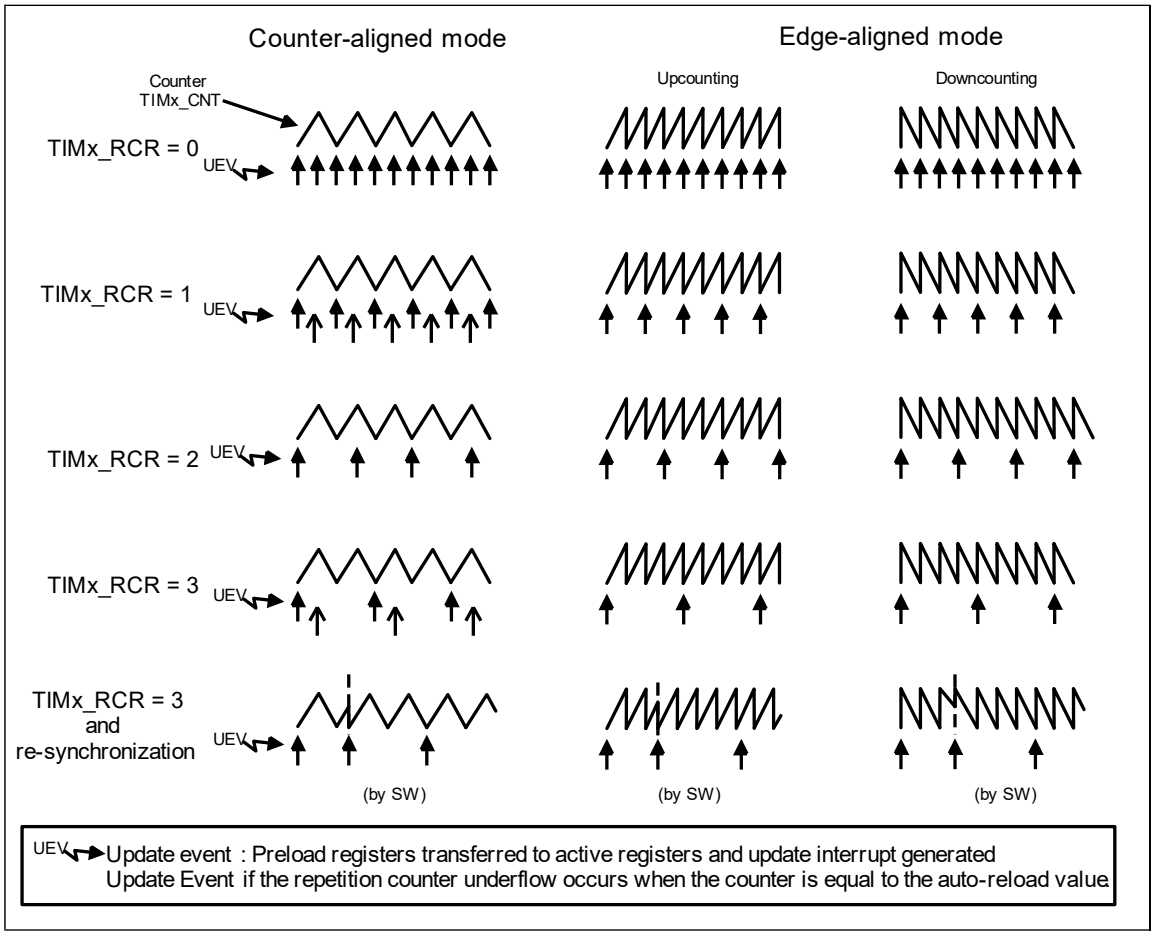



图 14-21 不同模式和 TIMx_RCR 寄存器设置下的更新频率示例

### 14.3.4. 时钟选择

计数器时钟可由下列时钟源提供:

- 内部时钟 (CK_INT)
- 外部时钟模式 1：外部输入引脚
- 外部时钟模式 2：外部触发输入 ETR
- 内部触发输入(ITRx)：使用一个定时器作为另一个定时器的预分频器，例如可将定时器 1 配置为定时器 2 的预分频器。

#### 内部时钟源 (CK_INT)

如果禁止从模式控制器 (SMS=000)，则 CEN 位、DIR 位（TIMx_CR1 寄存器中）和 UG 位（TIMx_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（UG 除外，仍保持自动清零）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 CK_INT 提供。

下图显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）

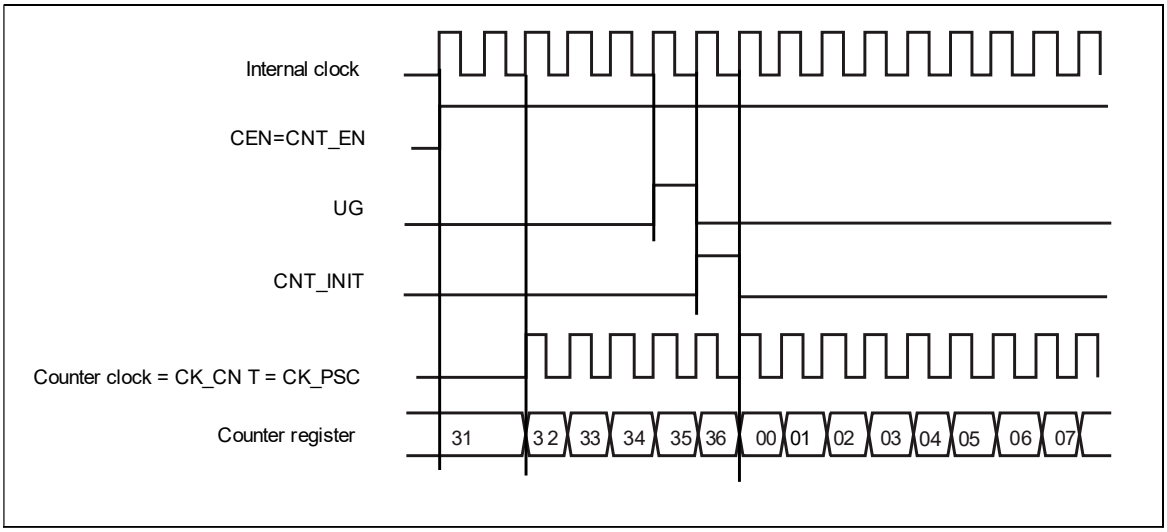



图 14-22 正常模式下的控制电路，内部时钟 1 分频

### 外部时钟源模式 1

当 TIMx_SMCR 寄存器中的 SMS=111 时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

例如，要使递增计数器在 TI2 输入出现上升沿时计数，请执行以下步骤：

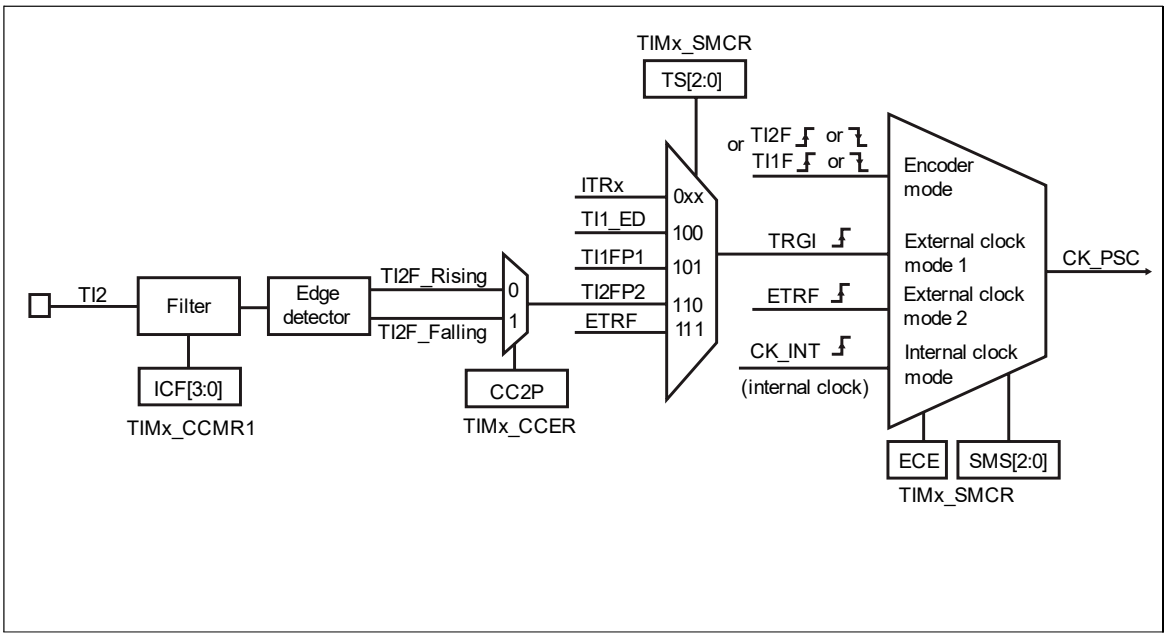



图 14-23 TI2 外部时钟连接示例

1. 通过在 TIMx_CCMR1 寄存器中写入 CC2S =“01”来配置通道 2，使其能够检测 TI2 输入的上升沿。
2. 通过在 TIMx_CCMR1 寄存器中写入 IC2F[3:0] 位来配置输入滤波带宽（如果不需要任何滤波器，请保持 IC2F=0000）。
3. 通过在 TIMx_CCER 寄存器中写入 CC2P=0 和 CC2NP=0 来选择上升沿极性。
4. 通过在 TIMx_SMCR 寄存器中写入 SMS=111，使定时器在外部时钟模式 1 下工作。
5. 通过在 TIMx_SMCR 寄存器中写入 TS=110 来选择 TI2 作为触发输入源。

6. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

*注： 由于捕获预分频器不用于触发操作，因此无需对其进行配置。*

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。

TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入的重新同步电路引起的。

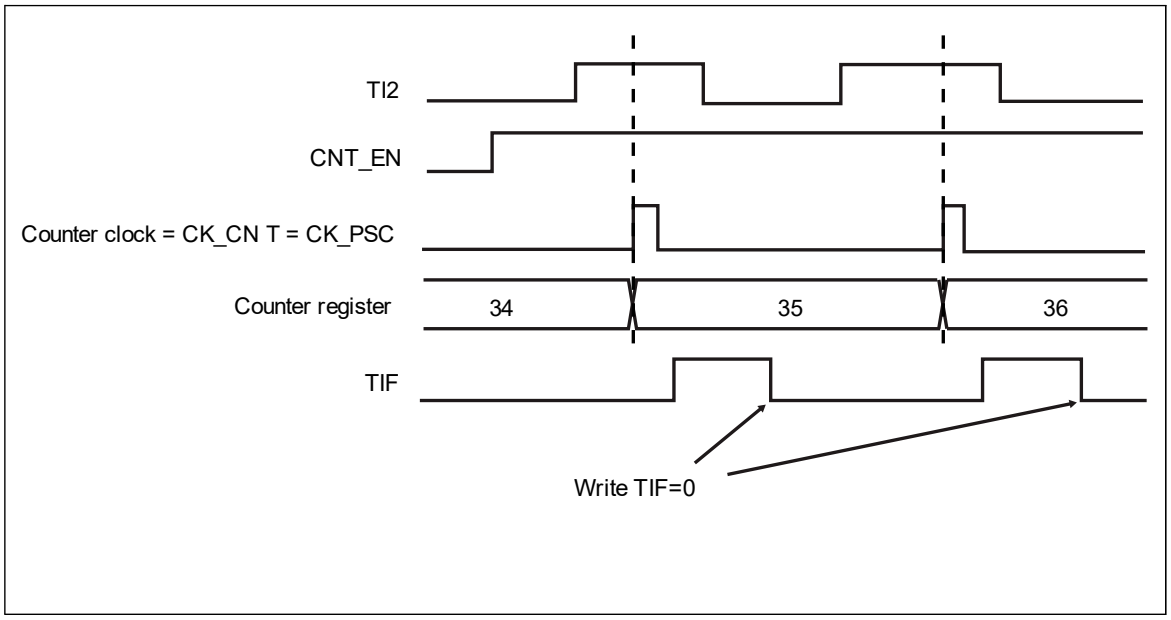



图 14-24 TI2 外部时钟连接示例

### 外部时钟源模式 2

通过在 TIMx_SMCR 寄存器中写入 ECE=1 可选择此模式，计数器可在外部触发输入 ETR 出现上升沿或下降沿时计数。下图简要介绍了外部触发输入模块。

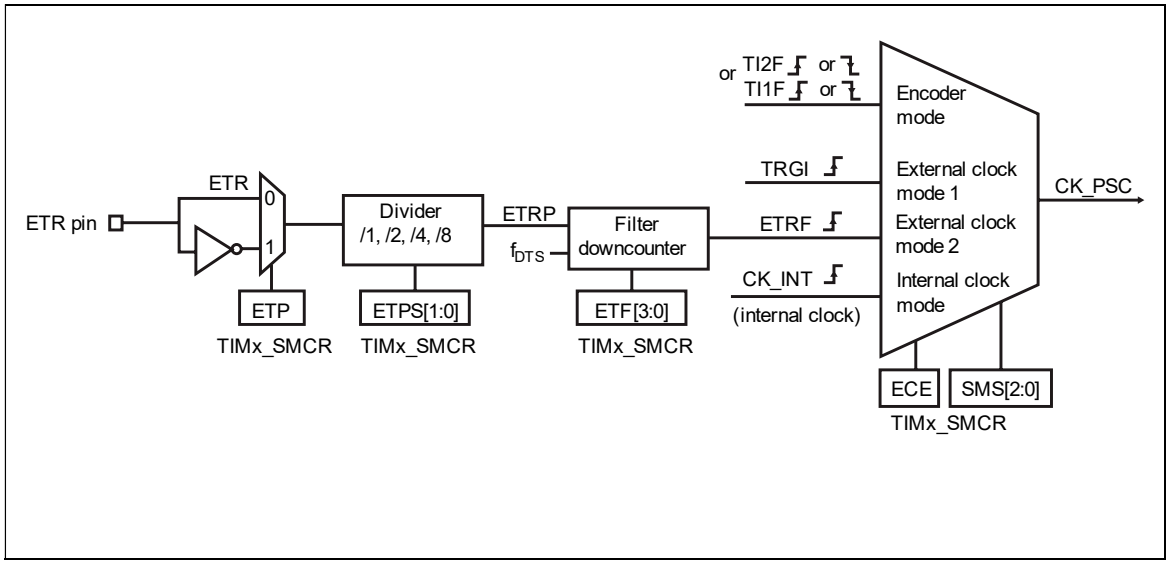



图 14-25 外部触发输入模块

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

1. 由于此例中不需滤波器，因此在 TIMx_SMCR 寄存器中写入 ETF[3:0]=0000。
2. 通过在 TIMx_SMCR 寄存器中写入 ETPS[1:0]=01 来设置预分频器。
3. 通过在 TIMx_SMCR 寄存器中写入 ETP=0 来选择 ETR 引脚的上升沿检测。
4. 通过在 TIMx_SMCR 寄存器中写入 ECE=1 来使能外部时钟模式 2。
5. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

ETR 每出现 2 个上升沿，计数器计数一次，ETR 的上升沿与实际计数器时钟之间的延迟是由于 ETRP 信号的重新同步电路引起的。

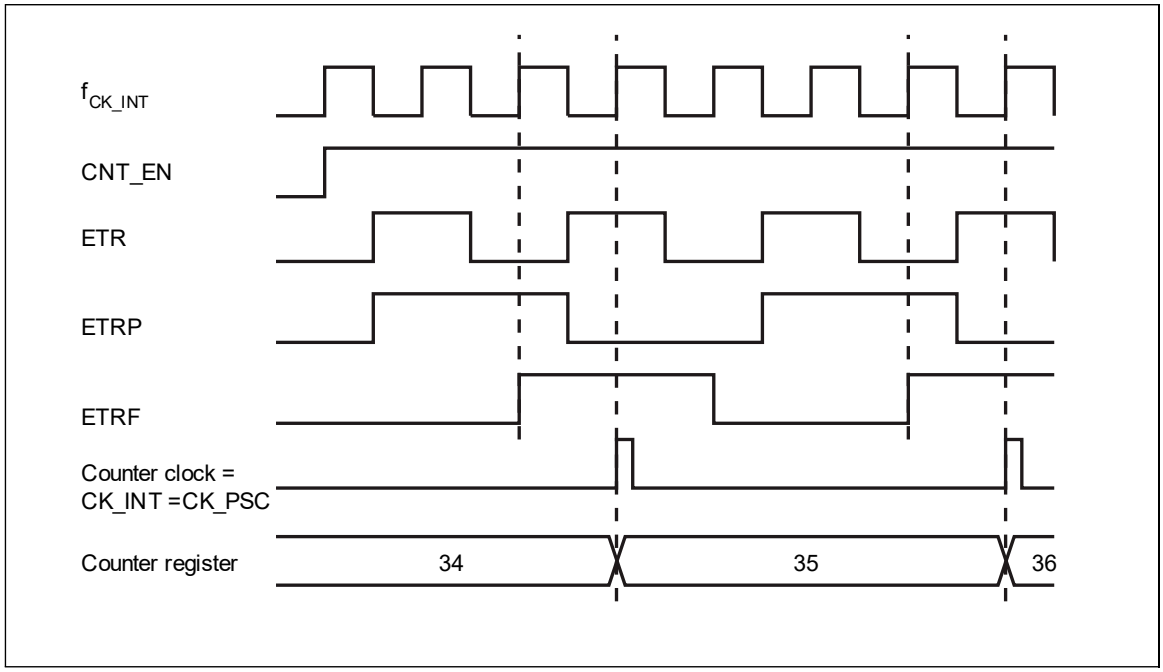



图 14-26 外部时钟模式 2 下的控制电路

### 14.3.5. 捕获/比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入阶段（数字滤波、多路复用和预分频器）和一个输出阶段（比较器和输出控制）构建而成。

图 20-27 到图 20-30 概括介绍了一个捕获/比较通道。

输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 TIxF。然后，带有极性选择功能的边沿检测器生成一个信号 (TIxFPx)，该信号可用作从模式控制器的触发输入，也可用作捕获命令。该信号先进行预分频 (ICxPS)，而后再进入捕获寄存器。

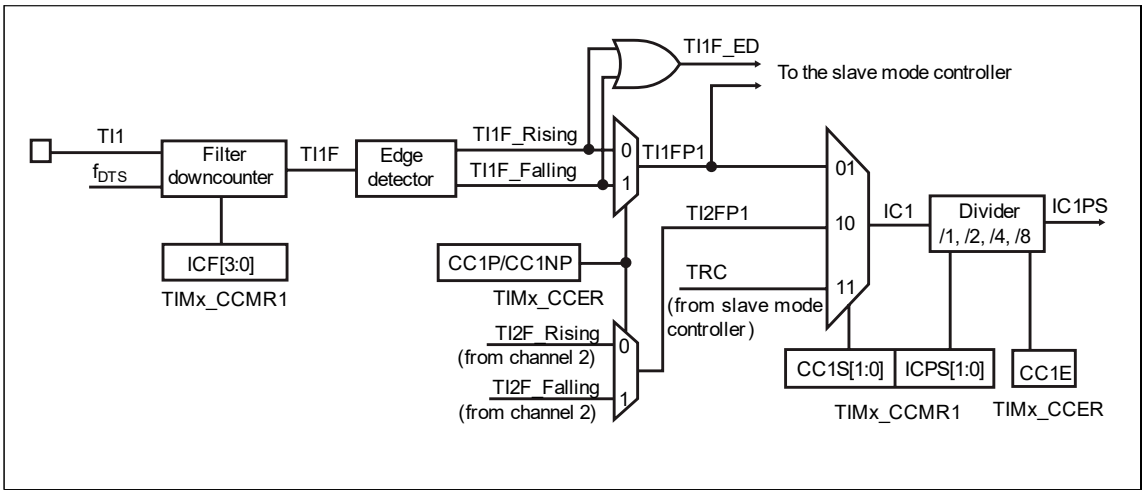



图 14-27 捕获/比较通道(例如：通道 1 输入阶段)

输出阶段生成一个中间波形作为基准： OCxRef（高电平有效）。链的末端决定最终输出信号的极性。

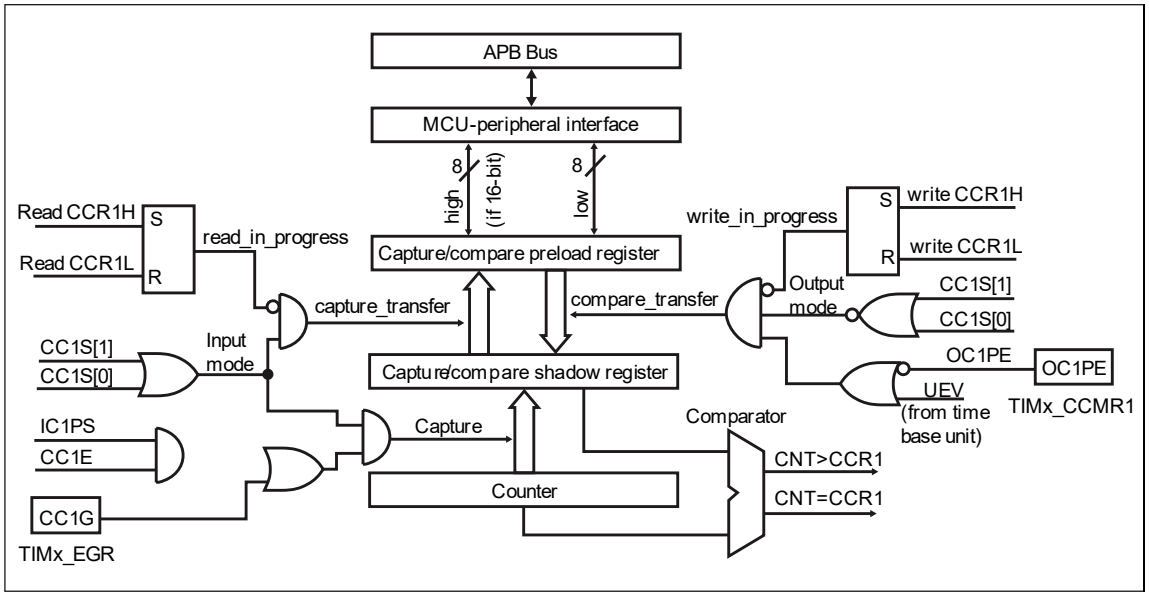



图 14-28 捕获/比较通道 1 主电路

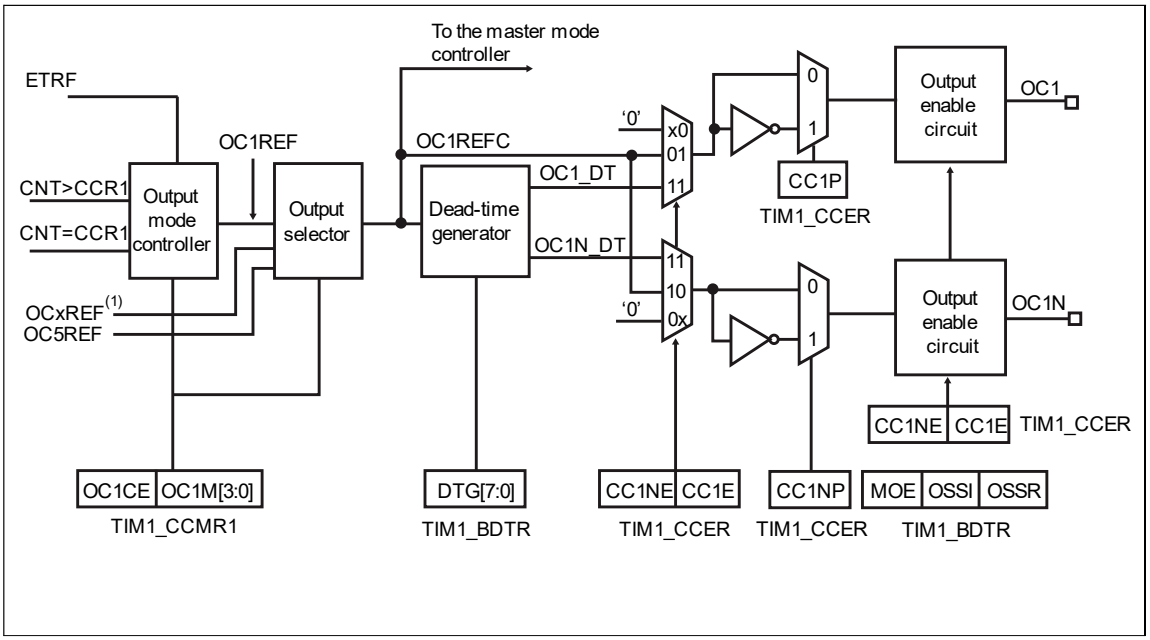



图 14-29 捕获/比较通道的输出阶段(通道 1 到 3)

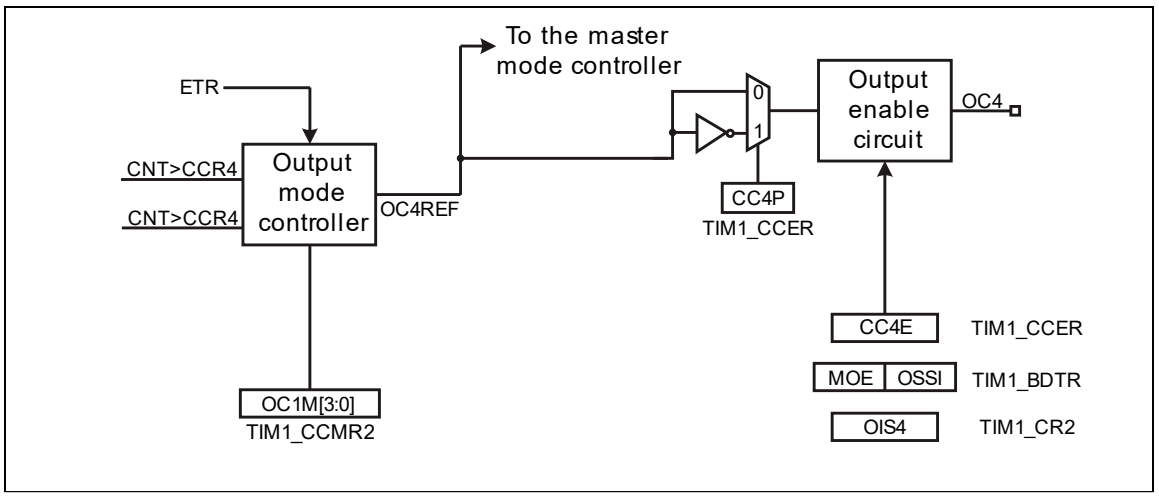



图 14-30 捕获/比较通道的输出阶段(通道 4)

捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。始终可通过读写操作访问预装载寄存器。

在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容复制到预装载寄存器中。

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

### 14.3.6. 输入捕获模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获/比较寄存器 (TIMx_CCRx) 来锁存计数器的值。发生捕获事件时，会将相应的 CCXIF 标志（TIMx_SR 寄存器）置 1，并可发送中断或 DMA 请求（如果已使能）。如果发生捕获事件时 CCxIF 标志已处于高位，则会将重复捕获标志 CCxOF（TIMx_SR 寄存器）置 1。可通过软件将 CCxIF 清零，或读取存储在 TIMx_CCRx 寄存器中的已捕获数据。向 CCxOF 写入“0”后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIMx_CCR1 中。具体操作步骤如下：

- 选择有效输入：TIMx_CCR1 必须连接到 TI1 输入，因此向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 TIMx_CCR1 寄存器将处于只读状态。
- 根据连接到定时器的信号，对所需的输入滤波带宽进行编程（如果输入为 Tix 输入，则对 TIMx_CCMRx 寄存器中的 ICxF 位进行编程）。假设信号边沿变化时，输入信号最多在 5 个内部时钟周期内发生抖动。因此，我们必须将滤波带宽设置为大于 5 个内部时钟周期。在检测到 8 个具有新电平的连续采样（以 $f_{DTS}$ 频率采样）后，可以确认 TI1 上的跳变沿。然后向 TIMx_CCMR1 寄存器中的 IC1F 位写入 0011。
- 通过在 TIMx_CCER 寄存器中将 CC1P 位和 CC1NP 位写入 0，选择 TI1 上的有效转换边沿（本例中为上升沿）。
- 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 TIMx_CCMR1 寄存器中的 IC1PS 位写入“00”） 。
- 通过将 TIMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
- 如果需要，可通过将 TIMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求，并且/或者通过将该寄存器中的 CC1DE 位置 1 来使能 DMA 请求。

发生输入捕获时：

- 发生有效跳变沿时，TIMx_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
- 根据 CC1IE 位生成中断。
- 根据 CC1DE 位生成 DMA 请求。

要处理重复捕获，建议在读出捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出标志之后与读取数据之前可能出现的重复捕获信息。

*注意：通过软件将 TIMx_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断和/或 DMA 请求。*

### 14.3.7. PWM 输入模式

此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这两个 ICx 信号在边沿处有效，但极性相反。
- 选择两个 TIxFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。

例如，可通过以下步骤对应用于 TI1 的 PWM 的周期（位于 TIMx_CCR1 寄存器中）和占空比（位于 TIMx_CCR2 寄存器中）进行测量(取决于 CK_INT 频率和预分频器的值)：

- 选择 TIMx_CCR1 的有效输入：向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01(选择 TI1)。
- 选择 TI1FP1 的有效极性(用于在 TIMx_CCR1 中捕获和计数器清零)：向 CC1P 位和 CC1NP 位写入“0”(上升沿有效)。
- 选择 TIMx_CCR2 的有效输入：向 TIMx_CCMR1 寄存器中的 CC2S 写入 10(选择 TI1)。
- 选择 TI1FP2 的有效极性(用于在 TIMx_CCR2 中捕获)：向 CC2P 位和 CC2NP 位写入 CC2P/CC2NP=“ 10”(下降沿有效)。
- 选择有效触发输入：向 TIMx_SMCR 寄存器中的 TS 位写入 101（选择 TI1FP1）。
- 将从模式控制器配置为复位模式：向 TIMx_SMCR 寄存器中的 SMS 位写入 0100。
- 使能捕获：向 TIMx_CCER 寄存器中的 CC1E 位和 CC2E 位写入“1”。

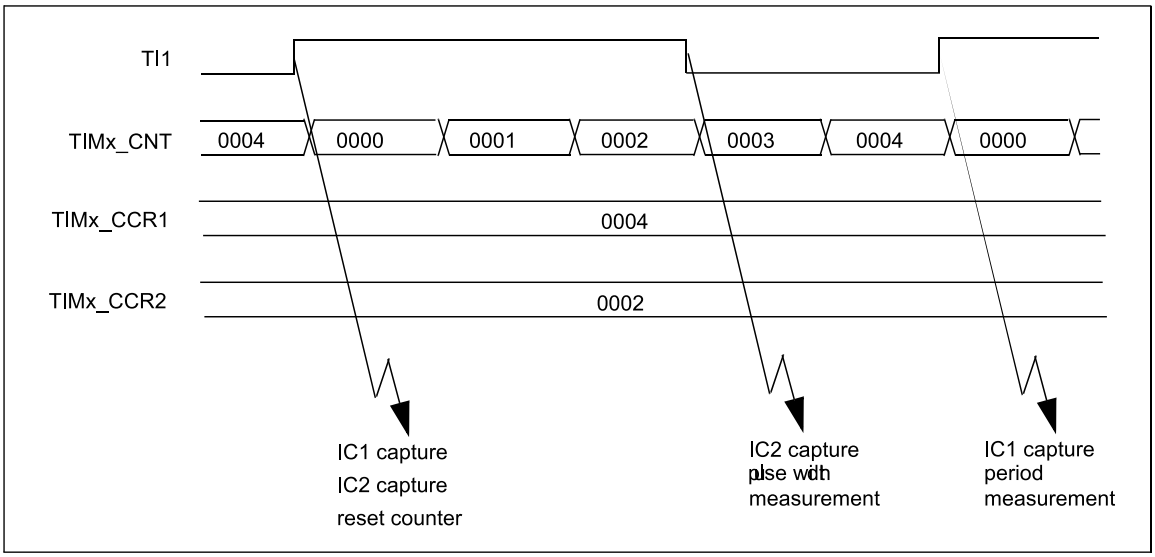



图 14-31 PWM 输入模式时序

### 14.3.8. 强制输出模式

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

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

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

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入 0100，可将 OCxREF 信号强制设置为低电平。

无论如何，TIMx_CCRx 影子寄存器与计数器之间的比较仍会执行，而且允许将标志置 1。因此可发送相应的中断和 DMA 请求。下面的输出比较模式一节对此进行了介绍。

### 14.3.9. 输出比较模式

此功能用于控制输出波形，或指示已经过某一时间段。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

- 将为相应的输出引脚分配一个可编程值, 该值由输出比较模式(TIMx_CCMRx 寄存器中 的 OCxM 位) 和输出极性（TIMx_CCER 寄存器中的 CCxP 位）定义。匹配时，输出引脚既可保持其电平 (OCXM=000)，也可设置为有效电平 (OCXM=001)、无效电平 (OCXM=010) 或进行翻转 (OCxM=011)。
- 将中断状态寄存器中的标志置 1（TIMx_SR 寄存器中的 CCxIF 位）。
- 如果相应中断使能位（TIMx_DIER 寄存器中的 CCXIE 位） 置 1，将生成中断。
- 如果相应使能位(TIMx_DIER 寄存器的 CCxDE 位, TIMx_CR2 寄存器的 CCDS 位, 用来选择 DMA 请求）置 1，将发送 DMA 请求。

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

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

步骤

1. 选择计数器时钟（内部、外部、预分频器）。
2. 在 TIMx_ARR 和 TIMx_CCRx 寄存器中写入所需数据。
3. 如果要生成中断请求，则需将 CCxIE 位置 1。
4. 选择输出模式。例如：
   a) 当 CNT 与 CCRx 匹配时，写入 OCxM = 0011 以翻转 OCx 输出引脚
   b) 写入 OCxPE = 0 以禁止预装载寄存器
   c) 写入 CCxP = 0 以选择高电平有效极性
   d) 写入 CCxE = 1 使能输出
5. 通过将 TIMx_CR1 寄存器中的 CEN 位置 1 来使能计数器。

可通过软件随时更新 TIMx_CCRx 寄存器以控制输出波形，前提是未使能预加载寄存器（OCxPE="0"，否则 TIMx_CCRx 影子寄存器仅在下一个更新事件 UEV 时进行更新）。下图给出了一个示例。

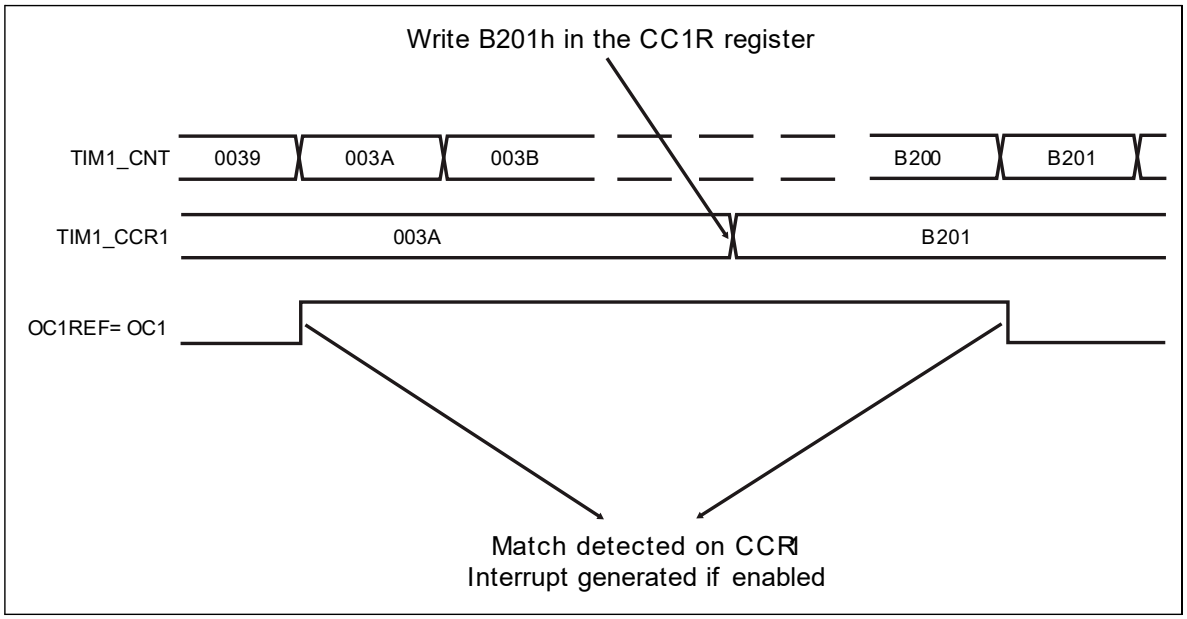



图 14-32 输出比较模式，翻转 OC1

### 14.3.10. PWM 模式

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

通过向 TIMx_CCMRx 寄存器的 OCxM 位写入"110"（PWM 模式 1）或"111"（PWM 模式 2），可以独立选择各通道（每个 OCx 输出对应一个 PWM）的 PWM 模式。必须通过将 TIMx_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，最后通过将 TIMx_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器（在递增计数或中心对齐模式下）。

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

OCx 极性可使用 TIMx_CCER 寄存器的 CCxP 位来编程 。可编程为高电平有效或低电平有效。通过 CCxE、CCxNE、MOE、OSSI 和 OSSR 位（TIMx_CCER 和 TIMx_BDTR 寄存器）的组合使能 OCx 输出。

有关详细信息，请参见 TIMx_CCER 寄存器说明。

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

根据 TIMx_CR1 寄存器中的 CMS 位状态，定时器能够产生边沿对齐模式或中心对齐模式的 PWM 信号。

**PWM 边沿对齐模式**

- 递增计数配置

当 TIMx_CR1 寄存器中的 DIR 位为低时执行递增计数。

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

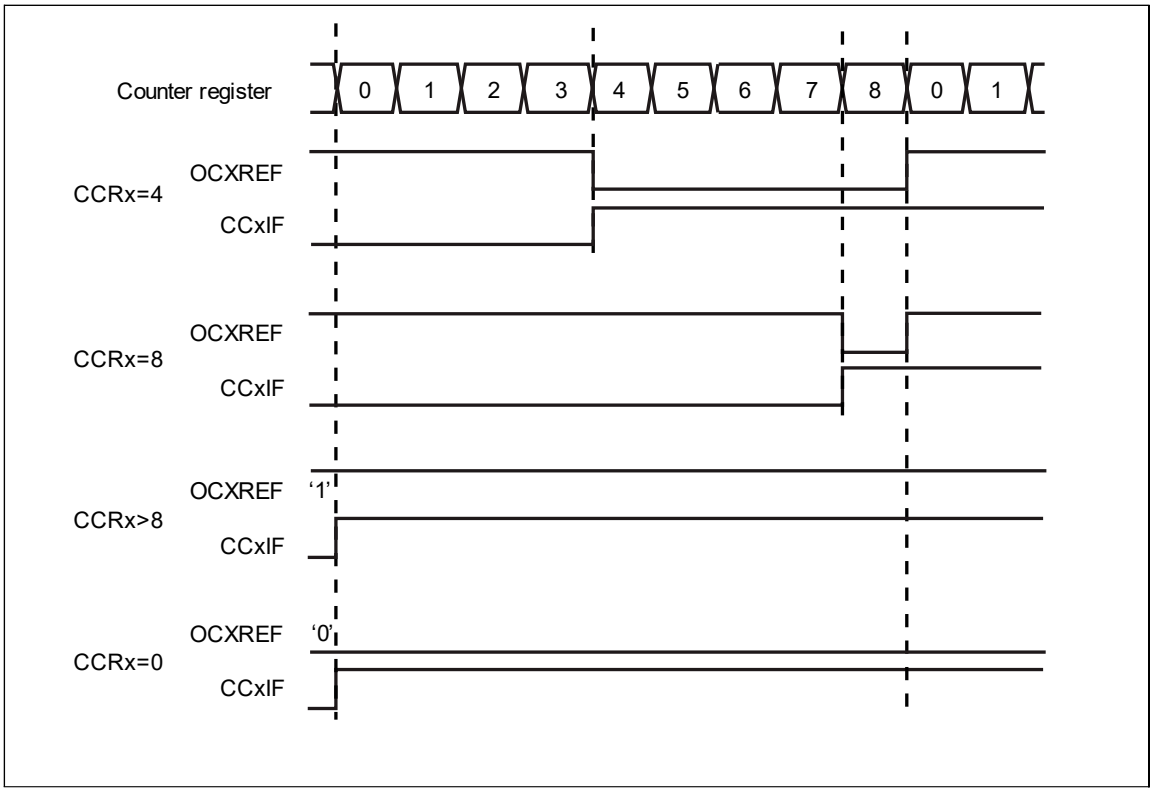



图 14-33 边沿对齐模式 PWM 波形(ARR=8)

- 递减计数配置

当 TIMx_CR1 寄存器中的 DIR 位为高时执行递减计数。

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

**PWM 中心对齐模式**

当 TIMx_CR1 寄存器中的 CMS 位不为“00”（其余所有配置对 OCxRef/OCx 信号具有相同的作用），中心对齐模式生效。根据 CMS 位的配置，可以在计数器递增计数、递减计数或同时递增和递减计数时将比较标志置 1。TIMx_CR1 寄存器中的方向位 (DIR) 由硬件更新，不得通过软件更改。

下图显示了中心对齐模式的 PWM 波形，在此例中：

- TIMx_ARR=8
- PWM 模式为 PWM 模式 1

- 在根据 TIMx_CR1 寄存器中 CMS=01 而选择的中心对齐模式 1 下，当计数器递减计数时，比较标志置 1。

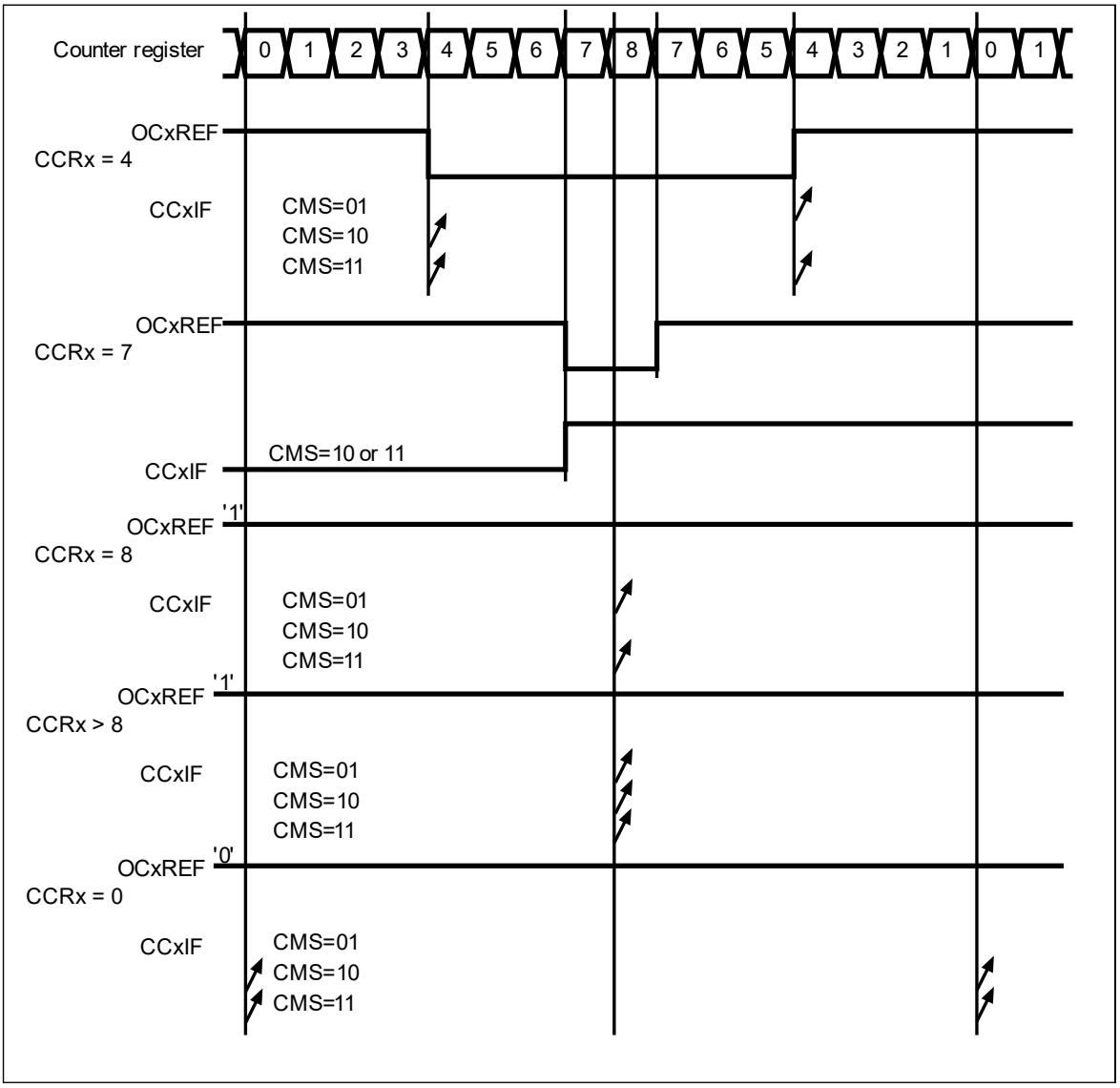



图 14-34 中心对齐模式 PWM 波形(ARR=8)

中心对齐模式使用建议

- 启动中心对齐模式时将使用当前的递增/递减计数配置。这意味着计数器将根据写入 TIMx_CR1 寄存器中 DIR 位的值进行递增或递减计数。此外，不得同时通过软件修改 DIR 和 CMS 位。
- 不建议在运行中心对齐模式时对计数器执行写操作，否则将发生意想不到的结果。尤其是：
- 如果写入计数器中的值大于自动重载值 (TIMx_CNT>TIMx_ARR)，计数方向不会更新。例如，如果计数器之前递增计数，则继续递增计数。
- 如果向计数器写入 0 或 TIMx_ARR 的值，计数方向会更新，但不生成更新事件 UEV。
- 使用中心对齐模式最为保险的方法是：在启动计数器前通过软件生成更新（将 TIMx_EGR 寄存器中的 UG 位置 1），并且不要在计数器运行过程中对其执行写操作。

### 14.3.11. 互补输出和死区插入

高级控制定时器 (TIM1 和 TIM8) 可以输出两路互补信号，并管理输出的关断与接通瞬间。

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

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

互补信号 OCx 和 OCxN 通过以下多个控制位的组合进行激活：TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位以及 TIMx_BDTR 和 TIMx_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位。应当注意，切换至空闲状态（MOE 下降到 0）的时刻，死区仍然有效。

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

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

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

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

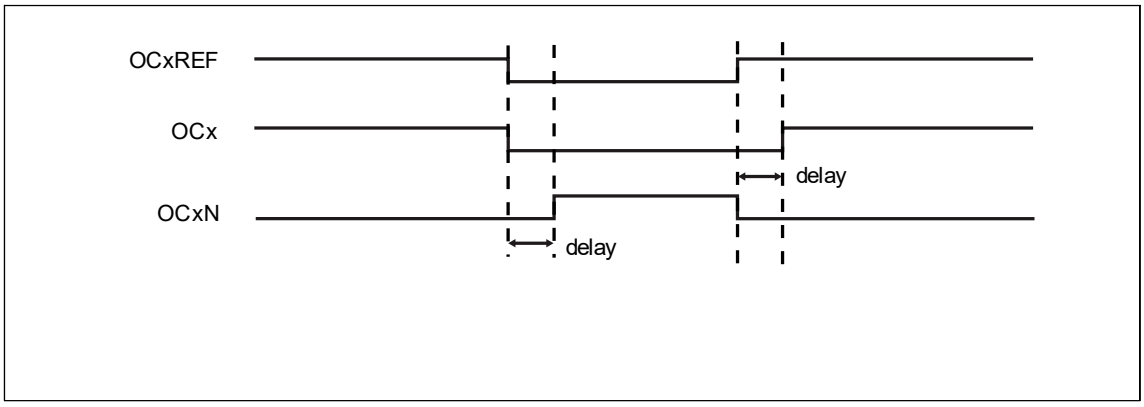



图 14-35 带死区插入的互补输出

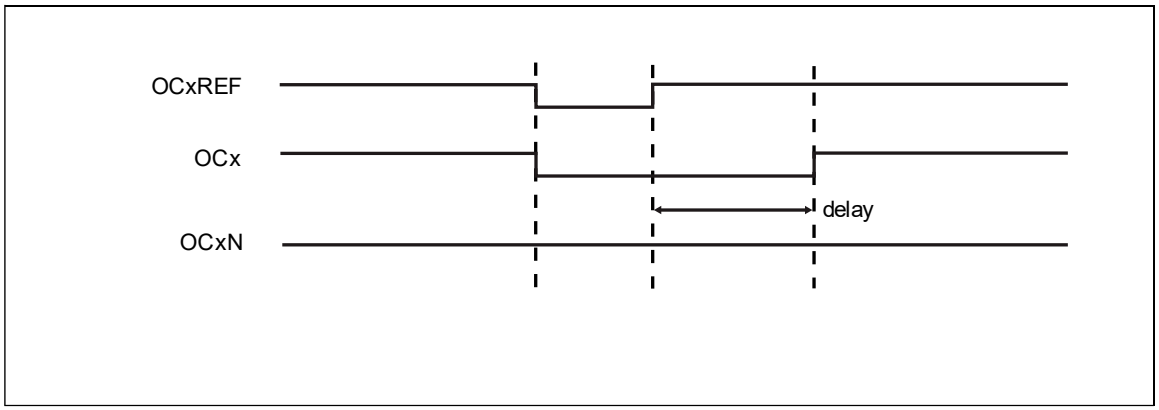



图 14-36 延迟时间大于负脉冲宽度的死区波形

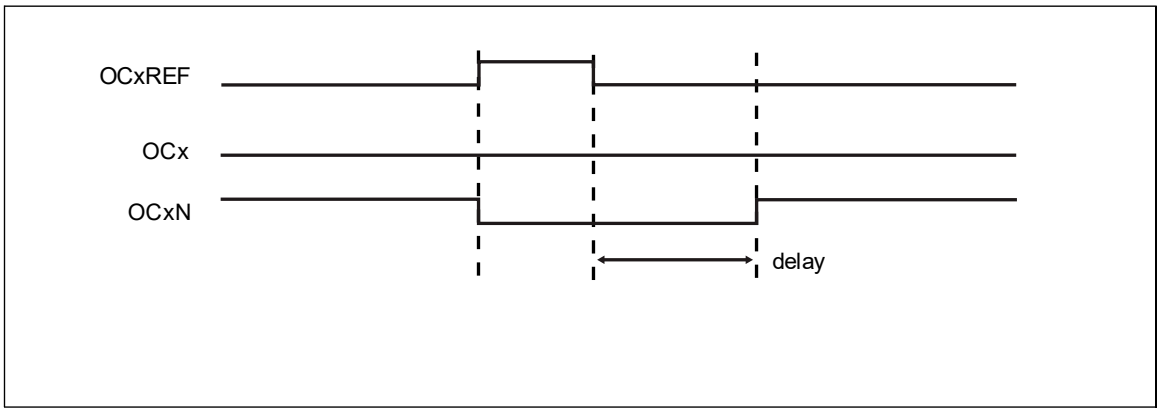



图 14-37 延迟时间大于正脉冲宽度的死区波形

死区延迟对于所有通道均相同，可通过 TIMx_BDTR 寄存器中的 DTG 位进行编程。

**将 OCxREF 重定向到 OCx 或 OCxN**

在输出模式（强制输出模式、输出比较模式或 PWM 模式）下，通过配置 TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位，可将 OCxREF 重定向到 OCx 输出或 OCxN 输出。

通过此功能，可以在一个输出上发送特定波形（如 PWM 或静态有效电平），而同时使互补输出保持其无效电平。或者使两个输出同时保持无效电平，或者两个输出同时处于有效电平，两者互补并且带死区。

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

### 14.3.12. 使用断路功能

使用断路功能时，根据其它控制位（TIMx_BDTR 寄存器中的 MOE、OSSI 和 OSSR 位以及 TIMx_CR2 寄存器中的 OISx 和 OISxN 位）修改输出使能信号和无效电平。任何情况下，OCx 和 OCxN 输出都不能同时置为有效电平。

断路源可以是断路输入引脚，也可以是时钟故障事件，后者由复位时钟控制器中的时钟安全系统 (CSS) 生成。

退出复位状态后，断路功能处于禁止状态，MOE 位处于低电平。将 TIMx_BDTR 寄存器中的 BKE 位置 1，可使能断路功能。断路输入的极性可通过该寄存器中的 BKP 位来选择。BKE 和 BKP 位可同时修改。对 BKE 和 BKP 位执行写操作时，写操作会在 1 个 APB 时钟周期的延迟后生效。因此，执行写操作后，需要等待 1 个 APB 时钟周期，才能准确回读该位。

由于 MOE 下降沿可能是异步信号，因此在实际信号（作用于输出）与同步控制位（位于 TIMx_BDTR 寄存器中）之间插入了再同步电路，从而在异步信号与同步信号之间产生延迟。具体而言，如果在 MOE 处于低电平时向其写入 1，则必须首先插入延迟（空指令），才能准确进行读取。这是因为写入的是异步信号，而读取的却是同步信号。

发生断路（断路输入上出现所选电平）时：

- MOE 位异步清零，使输出处于无效状态、空闲状态或复位状态（通过 OSSI 位进行选择）。即使 MCU 振荡器关闭，该功能仍然有效。
- MOE=0 时，将以 TIMx_CR2 寄存器 OISx 位中编程的电平驱动每个输出通道。如果 OSSI=0，则定时器将释放使能输出，否则使能输出始终保持高电平。
- 使用互补输出时：
- 输出首先置于复位状态或无效状态（取决于极性）。这是异步操作，因此即使没有为定时器提供时钟，

该操作仍有效。

- 如果定时器时钟仍存在，则将重新激活死区发生器，进而在死区后以 OISx 和 OISxN 位中编程的电平驱动输出。即使在这种情况下，也不能同时将 OCx 和 OCxN 驱动至其有效电平。请注意，MOE 进行再同步，因此死区的持续时间会比通常情况长一些（约 2 个 ck_tim 时钟周期）。
- 如果 OSSI=0，则定时器会释放使能输出，否则只要 CCxE 位或 CCxNE 位处于高电平，使能输出就会保持或变为高电平。
- 将断路状态标志（TIMx_SR 寄存器中的 BIF 位）置 1。如果 TIMx_DIER 寄存器中的 BIE 位置 1，可产生中断。如果 TIMx_DIER 寄存器中的 BDE 位置 1，可发送 DMA 请求。
- 如果 TIMx_BDTR 寄存器中的 AOE 位置 1，则 MOE 位会在发生下一更新事件 (UEV)时自动再次置 1。这一特性有许多用处，比如，可用于实现调节器的功能。否则 MOE 将始终保持低电平，直到再次向该位写入“1”。这种情况下，这一特性可用于确保安全。可以将断路输入连接到功率驱动器的警报、温度传感器或任何安全元件。

*注意： 断路输入为电平有效。因此，当断路输入有效电平时，不能将 MOE 位置 1（自动或通过软件）。同时，不能将状态标志 BIF 清零。*

断路可由 BRK 输入生成，该输入具有可编程极性，其使能位 BKE 位于 TIMx_BDTR 寄存器中。

断路有以下两种生成方案：

- 使用 BRK 输入生成，该输入具有可编程极性，其使能位 BKE 位于 TIMx_BDTR 寄存器中。
- 由软件通过 TIMx_EGR 寄存器中的 BG 位生成。

除断路输入和输出管理外，断路电路内部还实施了写保护，用以保护应用的安全。通过该功能，用户可冻结多个参数配置（死区持续时间、OCx/OCxN 极性和禁止时的状态、OCxM 配置、断路使能和极性）。可以通过 TIMx_BDTR 寄存器中的 LOCK 位，从 3 种保护级别中进行选择。MCU 复位后只能对 LOCK 位执行一次写操作。

下图所示为输出对断路响应行为的示例：

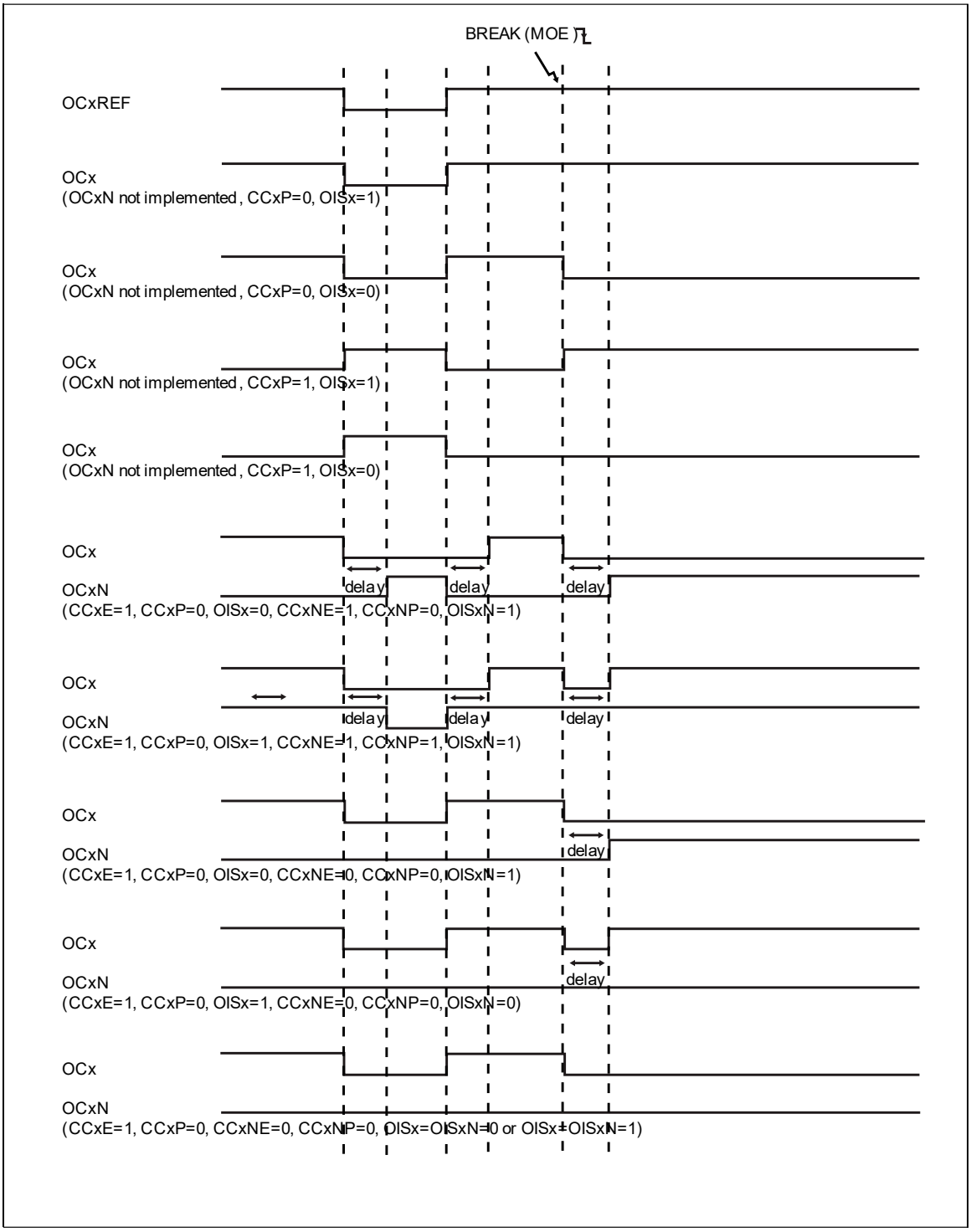



图 14-38 输出的断路响应行为

### 14.3.13. 发生外部事件时清除 OCxREF 信号

对于给定通道，在 ETRF 输入施加高电平（相应 TIMx_CCMRx 寄存器中的 OCxCE 使能位置“1”），可将 OCxREF 信号驱动为低电平。OCxREF 信号将保持低电平，直到发生下一更新事件 UEV。

此功能仅能用于输出比较模式和 PWM 模式，而不适用于强制输出模式。

例如， ETR 信号可以连接到比较器的输出，用于控制电流。此时，ETR 必须如下配置：

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

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

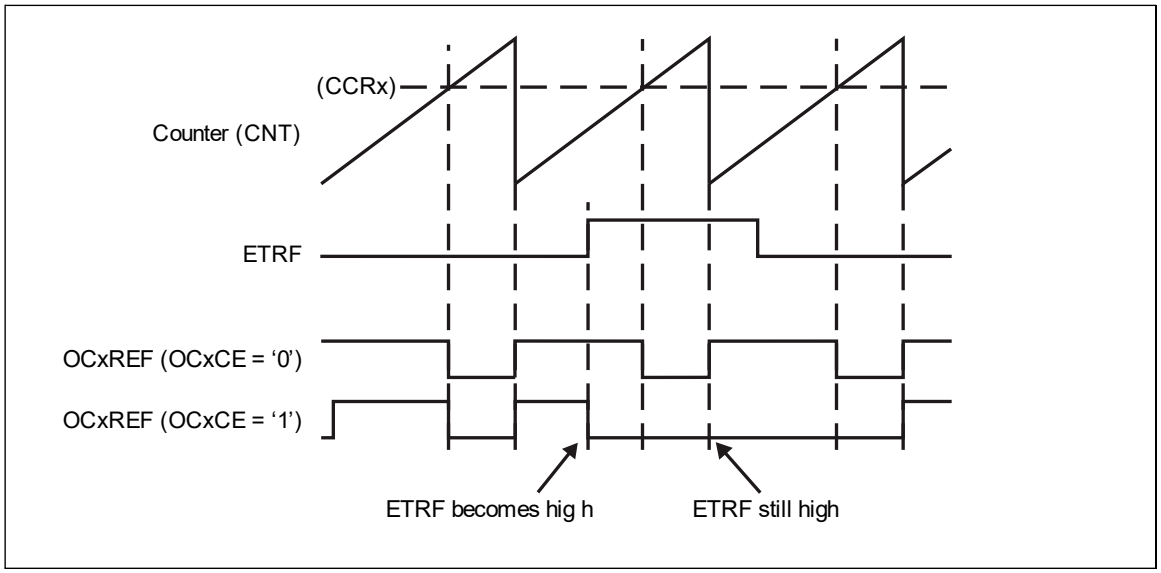



图 14-39 清除 TIMx 的 OCxREF

*注意：如果 PWM 的占空比为 100% (CCRx>ARR)，则下次计数器溢出时会再次使能 OCxREF。*

### 14.3.14. 生成 6 步 PWM

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

发生 COM 事件时，某个标志位（TIMx_SR 寄存器中的 COMIF 位）将会置 1。这时，如果 TIMx_DIER 寄存器中的 COMIE 位置 1，将产生中断；如果 TIMx_DIER 寄存器中 COMDE 位置 1，则将产生 DMA 请求。

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

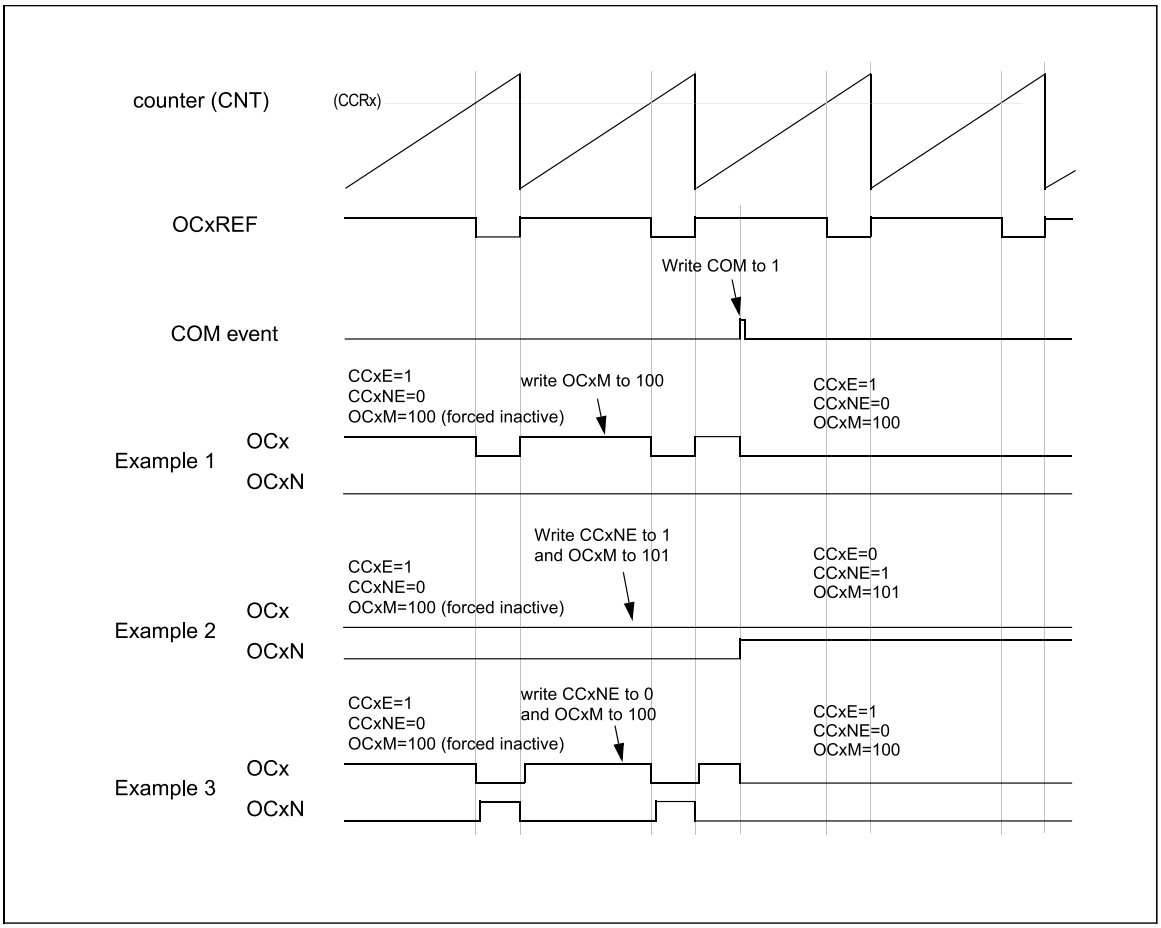



图 14-40 COM 事件生成 6 步 PWM 示例(OSSR=1)

### 14.3.15. 单脉冲模式

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

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。将 TIMx_CR1 寄存器中的 OPM 位置 1，即可选择单脉冲模式。这样发生下一更新事件 UEV 时，计数器将自动停止。

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

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

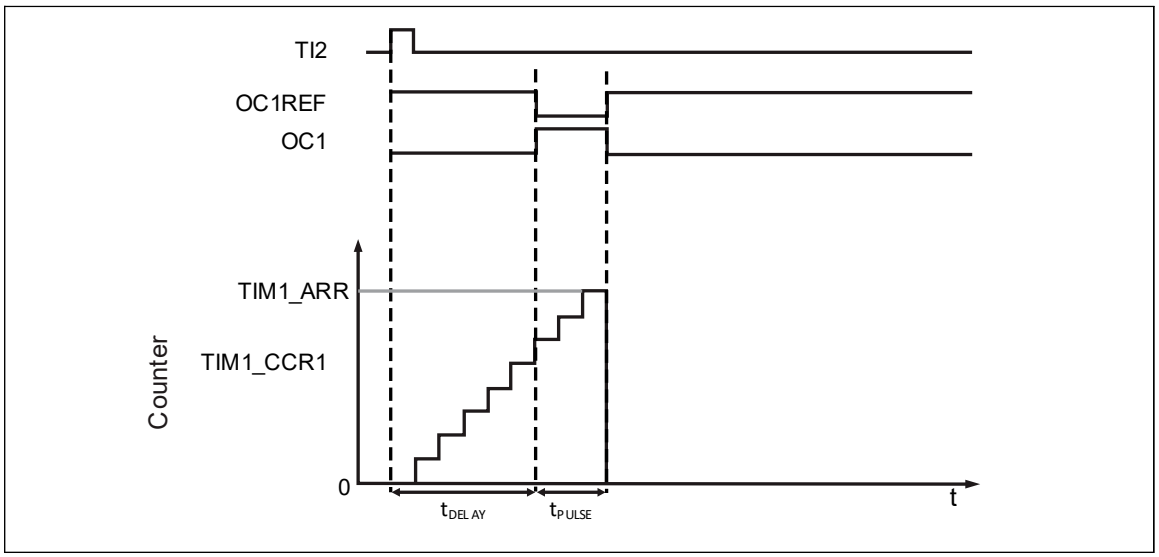



图 14-41 单脉冲模式示例

例如，用户希望达到这样的效果：在 TI2 输入引脚检测到正沿时，经过 $t_{DELAY}$ 的延迟，在 OC1 上产生一个长度为 $t_{PULSE}$ 的正脉冲。

使用 TI2FP2 作为触发 1：

- 在 TIMx_CCMR1 寄存器中写入 CC2S =“01”，以将 TI2FP2 映射到 TI2。
- 在 TIMx_CCER 寄存器中写入 CC2P =“0”和 CC2NP=“0”，使 TI2FP2 能够检测上升沿。
- 在 TIMx_SMCR 寄存器中写入 TS =“110”，以将 TI2FP2 配置为从模式控制器的触发(TRGI)。
- 在 TIMx_SMCR 寄存器中写入 SMS =“110”（触发模式），以使用 TI2FP2 启动计数器。

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

- $t_{DELAY}$ 由写入 TIMx_CCR1 寄存器的值定义。
- $t_{PULSE}$ 由自动重载值与比较值 (TIMx_ARR - TIMx_CCR1) 之差来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从“0”变为“1”，在计数器达到自动重载值时由“1”变为“0”。为此，应在 TIMx_CCMR1 寄存器中写入 OC1M=111，以使能 PWM 模式 2。如果需要，可选择在 TIMx_CCMR1 寄存器的 OC1PE 和 TIMx_CR1 寄存器的 ARPE 中写入“1”，以使能预装载寄存器。这种情况下，必须在 TIMx_CCR1 寄存器中写入比较值并在 TIMx_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI2 上的外部触发事件。本例中，CC1P 的值为“0”。

在本例中，TIMx_CR1 寄存器中的 DIR 和 CMS 位应为低。

由于仅需要 1 个脉冲（单脉冲模式），因此应向 TIMx_CR1 寄存器的 OPM 位写入“1”，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。 TIMx_CR1 寄存器中的 OPM 位置“0”时，即选择重复模式。

**特例： OCx 快速使能：**

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

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

### 14.3.16. 编码器接口模式

选择编码器接口模式时，如果计数器仅在 TI2 边沿处计数，在 TIMx_SMCR 寄存器中写入 SMS="001"；如果计数器仅在 TI1 边沿处计数，写入 SMS="010"；如果计数器在 TI1 和 TI2 边沿处均计数，则写入 SMS="011"。

通过编程 TIMx_CCER 寄存器的 CC1P 和 CC2P 位，选择 TI1 和 TI2 极性。如果需要，还可对输入滤波器进行编程。 CC1NP 和 CC2NP 必须保持低电平。

TI1 和 TI2 两个输入用于连接增量编码器。请参见下表。如果使能计数器（在 TIMx_CR1 寄存器的 CEN 位中写入"1"），则计数器的时钟由 TI1FP1 或 TI2FP2 上的每次有效信号转换提供。 TI1FP1 和 TI2FP2 是进行输入滤波器和极性选择后 TI1 和 TI2 的信号，如果不进行滤波和反相，则 TI1FP1=TI1，TI2FP2=TI2。将根据两个输入的信号转换序列，产生计数脉冲和方向信号。根据该信号转换序列，计数器相应递增或递减计数，同时硬件对 TIMx_CR1 寄存器的 DIR 位进行相应修改。任何输入（TI1 或 TI2）发生信号转换时，都会计算 DIR 位，无论计数器是仅在 TI1 或 TI2 边沿处计数，还是同时在 TI1 和 TI2 处计数。

编码器接口模式就相当于带有方向选择的外部时钟。这意味着，计数器仅在 0 到 TIMx_ARR 寄存器中的自动重载值之间进行连续计数（根据具体方向，从 0 递增计数到 ARR，或从 ARR 递减计数到 0）。因此，在启动前必须先配置 TIMx_ARR。同样，捕获、比较、预分频器、重复计数器及触发输出功能继续正常工作。编码器模式和外部时钟模式 2 不兼容，因此不能同时选择。

在此模式下，计数器会根据增量编码器的速度和方向自动进行修改，因此，其内容始终表示编码器的位置。计数方向对应于所连传感器的旋转方向。下表汇总了可能的组合（假设 TI1 和 TI2 不同时切换）。

表 14-1 计数方向与编码器信号的关系

| Active edge | Level on opposite signal (TI1FP1 for TI2, TI2FP2 for TI1) | TI1FP1 signal | | TI2FP2 signal | |
|---|---|---|---|---|---|
| | | Rising | Falling | Rising | Falling |
| Counting on TI1 only | High | Down | Up | No Count | No Count |
| | Low | Up | Down | No Count | No Count |
| Counting on TI2 only | High | No Count | No Count | Up | Down |
| | Low | No Count | No Count | Down | Up |
| Counting on TI1 and TI2 | High | Down | Up | Up | Down |
| | Low | Up | Down | Down | Up |

外部增量编码器可直接与 MCU 相连，无需外部接口逻辑。不过，通常使用比较器将编码器的差分输出转换为数字信号。这样大幅提高了抗噪声性能。用于指示机械零位的第三个编码器输出可与外部中断输入相连，用以触发计数器复位。

下图以计数器工作为例，说明了计数信号的生成和方向控制。同时也说明了选择双边沿时如何对输入抖动进行补偿。将传感器靠近其中一个切换点放置时可能出现这种情况。本例中假设配置如下：

- CC1S="01"（TIMx_CCMR1 寄存器， TI1FP1 映射到 TI1 上）。
- CC2S="01"（TIMx_CCMR2 寄存器， TI1FP2 映射到 TI2 上）。
- CC1P="0"，CC1NP="0"，且 IC1F ="0000"（TIMx_CCER 寄存器，TI1FP1 未反相， TI1FP1=TI1）。
- CC2P="0"， CC2NP="0"，且 IC2F ="0000"（TIMx_CCER 寄存器，TI1FP2 未反相，TI1FP2= TI2）。
- SMS="011"（TIMx_SMCR 寄存器，两个输入在上升沿和下降沿均有效）。
- CEN="1"（TIMx_CR1 寄存器，使能计数器）。

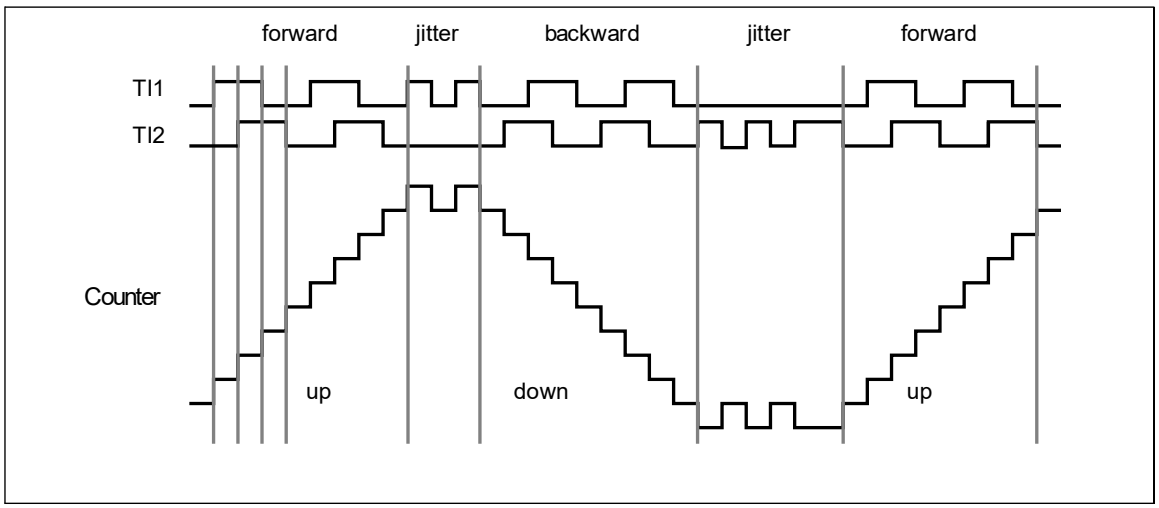



图 14-42 编码器接口模式下的计数器工作示例

下图举例说明 TI1FP1 极性反相时计数器的行为（除 CC1P=“ 1”外，其它配置与上例相同）。

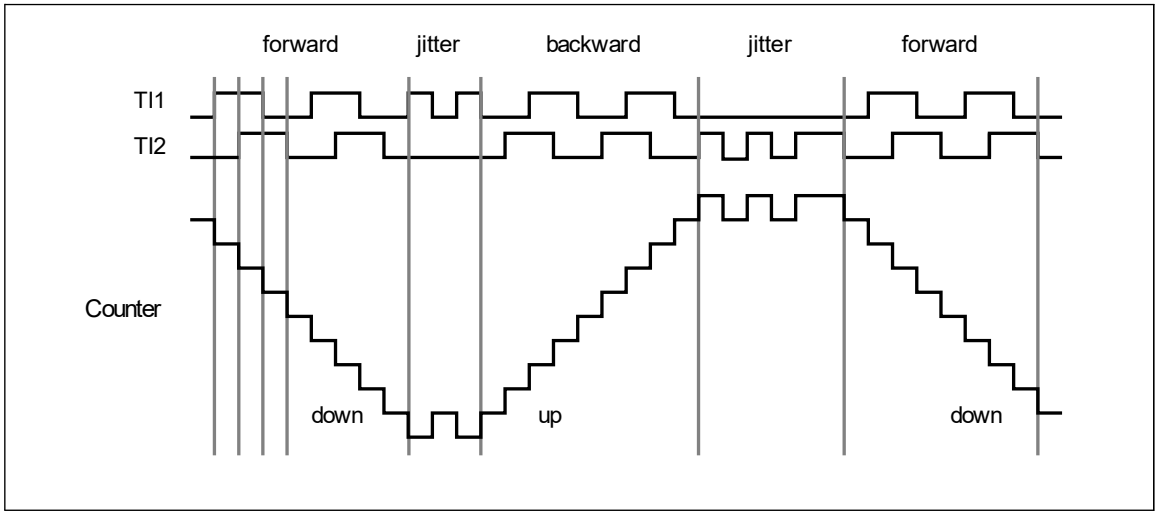



图 14-43 TI1FP1 极性反相时的编码器接口模式示例

定时器配置为编码器接口模式时，会提供传感器当前位置的相关信息。使用另一个配置为捕获模式的定时器测量两个编码器事件之间的周期，可获得动态信息（速度、加速度和减速度）。指示机械零位的编码器输出即可用于此目的。根据两个事件之间的时间间隔，还可定期读取计数器。如果可能，可以将计数器值锁存到第三个输入捕获寄存器来实现此目的（捕获信号必须为周期性信号，可以由另一个定时器产生）；还可以通过由实时时钟生成的 DMA 请求读取计数器值。

### 14.3.17. 定时器输入异或功能

通过 TIMx_CR2 寄存器中的 TI1S 位，可将通道 1 的输入滤波器连接到异或门的输出，从而将 TIMx_CH1 到 TIMx_CH3 这三个输入引脚组合在一起。

异或输出可与触发或输入捕获等所有定时器输入功能配合使用。下面章节以连接霍尔传感器为例介绍了此功能。

### 14.3.18. 连接霍尔传感器

可通过用于生成电机驱动 PWM 信号的高级控制定时器（TIM1 或 TIM8）以及称为“接口定时器”的另一个定时器 TIMx（TIM2、TIM3、TIM4 或 TIM5），实现与霍尔传感器的连接。3 个定时器输入引脚（TIMx_CH1、 TIMx_CH2 和 TIMx_CH3）通过异或门连接到 TI1 输入通道（通过将 TIMx_CR2 寄存器中的 TI1S 位置 1 来选择），并由“接口定时器”进行捕获。

从模式控制器配置为复位模式；从输入为 TI1F_ED。这样，每当 3 个输入中有一个输入发生切换时，计数器会从 0 开始重新计数。这样将产生由霍尔输入的任何变化而触发的时基。

在“接口定时器”上，捕获/比较通道 1 配置为捕获模式，捕获信号为 TRC。捕获值对应于输入上两次变化的间隔时间，可提供与电机转速相关的信息。

“接口定时器”可用于在输出模式下产生脉冲，以通过触发 COM 事件更改高级控制定时器(TIM1 或 TIM8)各个通道的配置。TIM1 定时器用于生成电机驱动 PWM 信号。为此，必须对接口定时器通道进行编程，以便在编程的延迟过后产生正脉冲（在输出比较或 PWM 模式中）。该脉冲通过 TRGO 输出发送到高级控制定时器（TIM1 或 TIM8）。

示例：霍尔输入与一个 TIMx 定时器相连接，每当霍尔输入发生更改，需要在所编程的延迟过后更改高级控制定时器 TIM1 的 PWM 配置。

- 向 TIMx_CR2 寄存器的 TI1S 位写入“1”，使 3 个定时器输入经过异或运算后进入 TI1 输入通道。
- 时基编程：向 TIMx_ARR 写入其最大值（计数器必须通过 TI1 的变化清零）。设置预分频器，以得到最大计数器周期，该周期长于传感器上两次变化的间隔时间。
- 将通道 1 编程为捕获模式（选择 TRC）：向 TIMx_CCMR1 寄存器的 CC1S 位写入“11”。如果需要，还可以编程数字滤波器。
- 将通道 2 编程为 PWM 2 模式，并具有所需延迟：向 TIMx_CCMR1 寄存器的 OC2M 位写入“111”，CC2S 位写入“00”。
- 选择 OC2REF 作为 TRGO 上的触发输出：向 TIMx_CR2 寄存器的 MMS 位写入“101”。

在高级控制定时器 TIM1 中，必须选择正确的 ITR 输入作为触发输入，定时器编程为可产生 PWM 信号，捕获/比较控制信号进行预装载（TIMx_CR2 寄存器的 CCPC=1），并且 COM 事件由触发输入控制（TIMx_CR2 寄存器中 CCUS=1）。发生 COM 事件后，在 PWM 控制位（CCxE、 OCxM）中写入下一步的配置，此操作可在由 OC2REF 上升沿产生的中断子程序中完成。下图为本示例的示意图。

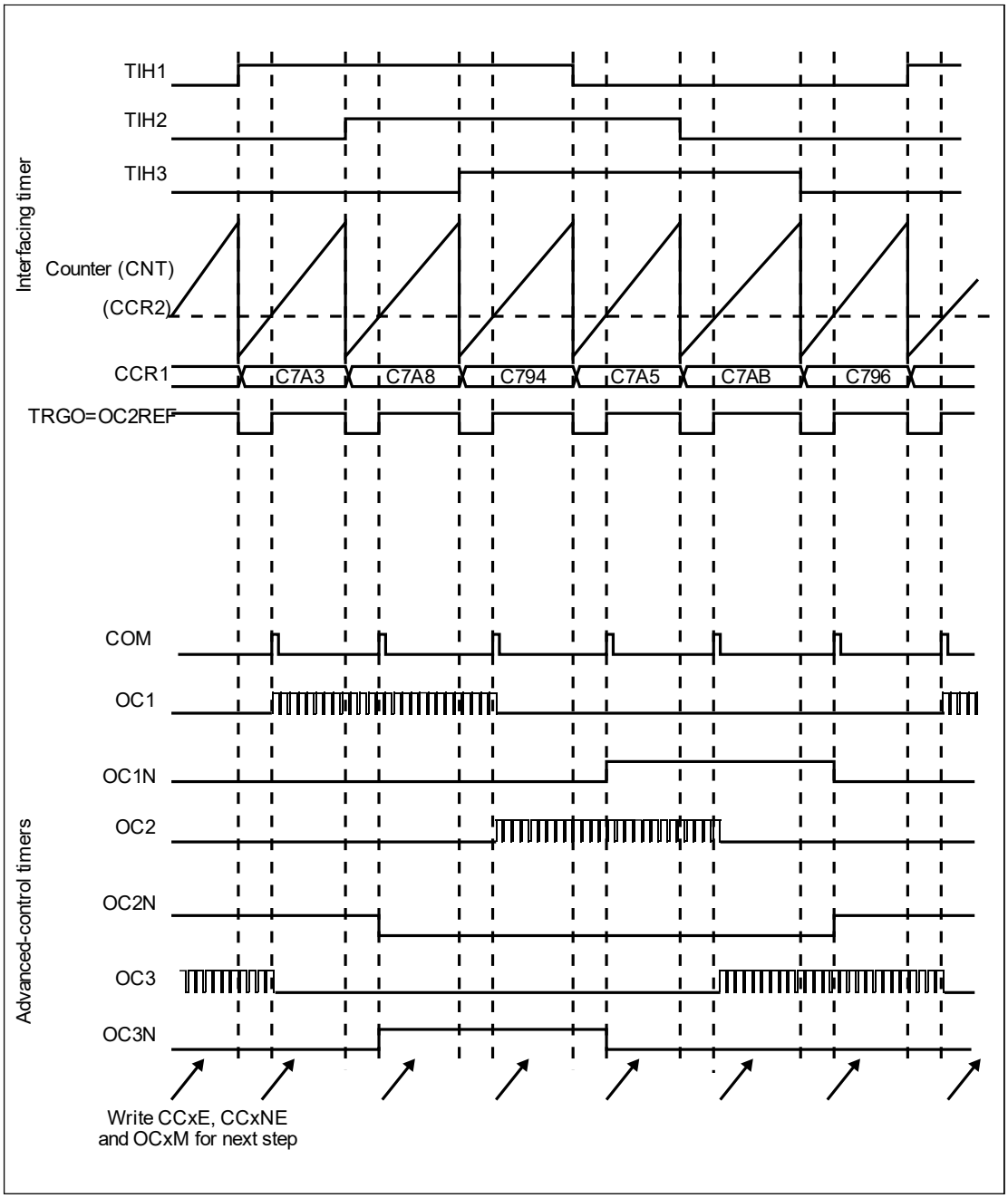



图 14-44 霍尔传感器接口示例

## 14.3.19. TIMx 与外部触发同步

TIMx 定时器可与外部触发以下列模式实现同步：复位模式、门控模式和触发模式。

**从模式：复位模式**

当触发输入信号产生变化时，计数器及其预分频器可重新初始化。此外，如果 TIMx_CR1 寄存器中的 URS 位处于低电平，则会生成更新事件 UEV。然后，所有预装载寄存器（TIMx_ARR 和 TIMx_CCRx）都将更新。

在以下示例中，TI1 输入上出现上升沿时，递增计数器清零：

- 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波时间（本例中不需要任何滤波，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获

源，即 TIMx_CCMR1 寄存器中的 CC1S = 01。在 TIMx_CCER 寄存器中写入 CC1P=0 和 CC1NP='0'，验证极性（仅检测上升沿）。

- 在 TIMx_SMCR 寄存器中写入 SMS=100，将定时器配置为复位模式。在 TIMx_SMCR 寄存器中写入 TS=101，选择 TI1 作为输入源。
- 在 TIMx_CR1 寄存器中写入 CEN=1，启动计数器。

计数器使用内部时钟计数，然后正常运转，直到出现 TI1 上升沿。当 TI1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时，触发标志（TIMx_SR 寄存器中的 TIF 位）置 1，使能中断或 DMA 后，还可发送中断或 DMA 请求（取决于 TIMx_DIER 寄存器中的 TIE 和 TDE 位）。

下图显示了自动重载寄存器 TIMx_ARR=0x36 时的相关行为。 TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

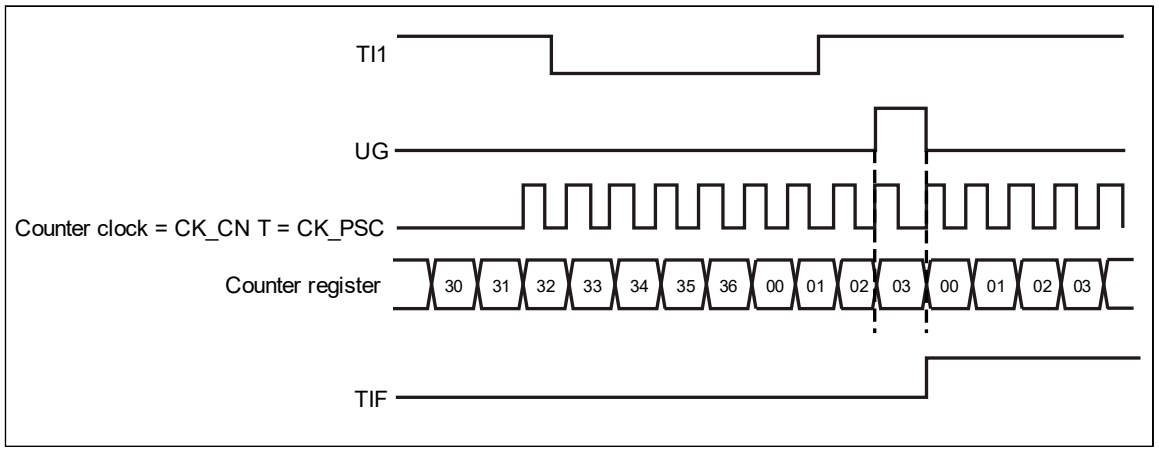



图 14-45 复位模式下的控制电路

**从模式：门控模式**

输入信号的电平可用来使能计数器。

在以下示例中，递增计数器仅在 TI1 输入为低电平时计数：

- 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波时间（本例中不需要任何滤波，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S=01。在 TIMx_CCER 寄存器中写入 CC1P=1 和 CC1NP="0"，以确定极性（仅检测低电平）。
- 在 TIMx_SMCR 寄存器中写入 SMS=101，将定时器配置为门控模式。在 TIMx_SMCR 寄存器中写入 TS=101，选择 TI1 作为输入源。
- 在 TIMx_CR1 寄存器中写入 CEN=1，使能计数器（在门控模式下，如果 CEN=0，则无论触发输入电平如何，计数器都不启动）。

只要 TI1 为低电平，计数器就开始根据内部时钟计数，直到 TI1 变为高电平时停止计数。计数器启动或停止时， TIMx_SR 寄存器中的 TIF 标志都会置 1。

TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

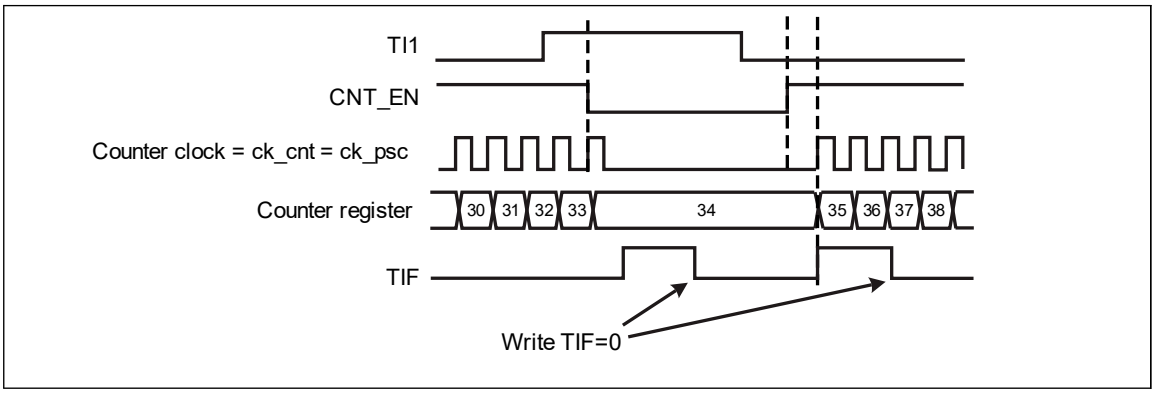



图 14-46 门控模式下的控制电路

**从模式：触发模式**

所选输入上发生某一事件时可以启动计数器。

在以下示例中，TI2 输入上出现上升沿时，递增计数器启动：

- 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波时间（本例中不需要任何滤波，因此保持 IC2F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC2S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC2S=01。在 TIMx_CCER 寄存器中写入 CC2P=1 和 CC2NP=0，以确定极性（仅检测低电平）。
- 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIMx_SMCR 寄存器中写入 TS=110，选择 TI2 作为输入源。

当 TI2 出现上升沿时，计数器开始根据内部时钟计数，并且 TIF 标志置 1。

TI2 的上升沿与实际计数器启动之间的延迟是由于 TI2 输入的重新同步电路引起的。

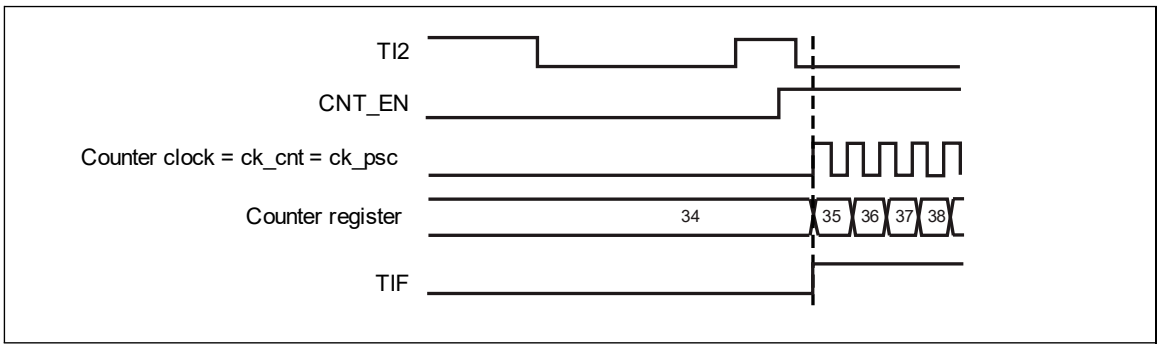



图 14-47 触发模式下的控制电路

**从模式：外部时钟模式 2 + 触发模式**

外部时钟模式 2 可与另一种从模式（外部时钟模式 1 和编码器模式除外）结合使用。这种情况下， ETR 信号用作外部时钟输入，在复位模式、门控模式或触发模式下工作时，可选择另一个输入作为触发输入。不建议通过 TIMx_SMCR 寄存器中的 TS 位来选择 ETR 作为 TRGI。

在以下示例中，只要 TI1 出现上升沿，递增计数器即会在 ETR 信号的每个上升沿处递增：

1. 通过对 TIMx_SMCR 寄存器进行如下编程，配置外部触发输入电路：

- ETF = 0000：无滤波器
- ETPS = 00：禁止预分频器
- ETP = 0：检测 ETR 的上升沿，并写入 ECE=1，以使能外部时钟模式 2。

2. 如下配置通道 1，以检测 TI 的上升沿：

- IC1F=0000：无滤波器。
- 由于捕获预分频器不用于触发操作，因此无需对其进行配置。
- TIMx_CCMR1 寄存器中 CC1S=01，只选择输入捕获源。
- TIMx_CCER 寄存器中 CC1P=0 且 CC1NP=“0”，以确定极性（仅检测上升沿）。

3. 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIMx_SMCR 寄存器中写入 TS=101，选择 TI1 作为输入源。

TI1 出现上升沿时将使能计数器并且 TIF 标志置 1。然后计数器在 ETR 出现上升沿时计数。ETR 信号的上升沿与实际计数器复位之间的延迟是由于 ETRP 输入的重新同步电路引起的。

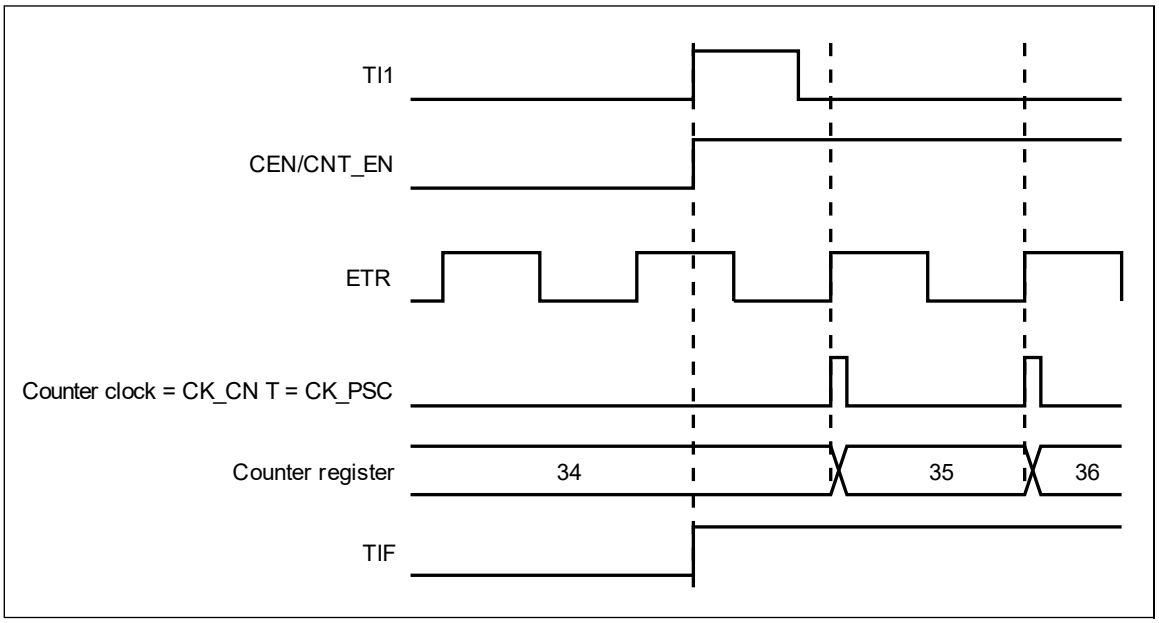



图 14-48 外部时钟模式 2+触发模式下的控制电路

### 14.3.20. 定时器同步

TIM 定时器从内部链接在一起，以实现定时器同步或级联。有关详细信息，请参见章节：定时器同步。

### 14.3.21. 调试模式

当微控制器进入调试模式(内核停止)时，TIMx 计数器会根据 DBG 模块中的 DBG_TIMx_STOP 配置位选择继续正常工作或者停止工作。

## 14.4. TIM1 和 TIM8 寄存器

有关寄存器说明中使用的缩写，请参见第 1.1 节。

外设寄存器必须按半字（16 位）或字（32 位）进行写访问。而读访问则可按字节（8 位）、半字（ 16 位）或字（ 32 位）进行。

### 14.4.1. TIM1 和 TIM8 控制寄存器 1(TIMx_CR1)

偏移地址：0x00

复位值：0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 15:10 | 保留 | 必须保持复位值 |
| 位 9:8 | CKD[1:0] | 时钟分频<br>指示定时器时钟 (CK_INT) 频率与死区发生器以及数字滤波器(ETR、 Tix)所使用的死区及采样时钟 ($t_{DTS}$) 之间的分频比,<br>00：$t_{DTS} = t_{CK_INT}$<br>01：$t_{DTS} = 2 * t_{CK_INT}$<br>10：$t_{DTS} = 4 * t_{CK_INT}$<br>11：保留，不要设置成此值 |
| 位 7 | ARPE | 自动重载预装载使能<br>0：TIMx_ARR 寄存器不进行缓冲<br>1：TIMx_ARR 寄存器进行缓冲 |
| 位 6:5 | CMS[1:0] | 中心对齐模式选择<br>00：边沿对齐模式。计数器根据方向位 (DIR) 递增计数或递减计数。<br>01：中心对齐模式 1。计数器交替进行递增计数和递减计数。仅当计数器递减计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CxS=00）的输出比较中断标志才置 1。<br>10：中心对齐模式 2。计数器交替进行递增计数和递减计数。仅当计数器递增计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CxS=00）的输出比较中断标志才置 1。<br>11：中心对齐模式 3。计数器交替进行递增计数和递减计数。当计数器递增计数或递减计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CxS=00）的输出比较中断标志都会置 1。<br>*注意：只要计数器处于使能状态 (CEN=1)，就不得从边沿对齐模式切换为中心对齐模式。* |
| 位 4 | DIR | 方向<br>0：计数器递增计数<br>1：计数器递减计数<br>*注意： 当定时器配置为中心对齐模式或编码器模式时，该位为只读状态* |
| 位 3 | OPM | 单脉冲模式<br>0：计数器在发生更新事件时不会停止计数<br>1：计数器在发生下一更新事件时停止计数（将 CEN 位清零） |
| 位 2 | URS | 更新请求源<br>此位由软件置 1 和清零，用以选择 UEV 事件源。<br>0：失能时，所有以下事件都会生成更新中断或 DMA 请求。此类事件包括:<br>• 计数器上溢/下溢<br>• 将 UG 位置 1<br>• 通过从模式控制器生成的更新事件<br>1：使能时，只有计数器上溢/下溢会生成更新中断或 DMA 请求。 |
| 位 1 | UDIS | 更新禁止<br>此位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。<br>0：使能 UEV。更新 (UEV) 事件可通过以下事件之一生成:<br>• 计数器上溢/下溢 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | • 将 UG 位置 1<br>• 通过从模式控制器生成的更新事件<br>然后更新影子寄存器的值。<br>1：禁止 UEV。不会生成更新事件，各影子寄存器的值(ARR、PSC 和 CCRx)保持不变。<br>但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。 |
| 位 0 | CEN | 计数器使能<br>0：禁止计数器<br>1：使能计数器<br>*注意：只有事先通过软件将 CEN 位置 1，才可以使用外部时钟、门控模式和编码器模式。而触发模式可通过硬件自动将 CEN 位置 1。* |

### 14.4.2. TIM1 和 TIM8 控制寄存器 2(TIMx_CR2)

偏移地址：0x04

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESET | OIS4 | OIS3N | OIS3 | OIS2N | OIS2 | OIS1N | OIS1 | TI1S | MMS[2:0] | | | CCDS | CCUS | RESET | CCPC |
| | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15 | 保留 | 必须保持复位值。 |
| 14 | OIS4 | 输出空闲状态 4（OC4 输出）<br>参见 OIS1 位 |
| 13 | OIS3N | 输出空闲状态 3（OC3N 输出）<br>参见 OIS1N 位 |
| 12 | OIS3 | 输出空闲状态 3（OC3 输出）<br>参见 OIS1 位 |
| 11 | OIS2N | 输出空闲状态 2（OC2N 输出）<br>参见 OIS1N 位 |
| 10 | OIS2 | 输出空闲状态 2（OC2 输出）<br>参见 OIS1 位 |
| 9 | OIS1N | 输出空闲状态 1（OC1N 输出）<br>0：当 MOE=0 时，经过死区时间后 OC1N=0<br>1：当 MOE=0 时，经过死区时间后 OC1N=1<br>*注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，此位即无法修改。* |
| 8 | OIS1 | 输出空闲状态 1（OC1 输出）<br>0：当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）OC1=0<br>1：当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）OC1=1<br>*注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，此位即无法修改。* |
| 7 | TI1S | TI1 选择<br>0：TIMx_CH1 引脚连接到 TI1 输入 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 1：TIMx_CH1、CH2 和 CH3 引脚连接到 TI1 输入（异或组合） |
| 6:4 | MMS[1:0] | 主模式选择<br>这些位可选择主模式下将要发送到从定时器以实现同步的信息(TRGO)。这些位的组合如下：<br>000：复位 – TIMx_EGR 寄存器中的 UG 位用作触发输出(TRGO)。如果复位由触发输入生成（从模式控制器配置为复位模式），则 TRGO 上的信号相比实际复位会有延迟。<br>001：使能 – 计数器使能信号 CNT_EN 用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号可由 CEN 控制位产生。<br>当配置为门控模式时，也可由触发输入产生。当计数器使能信号由触发输入控制时，TRGO 上会存在延迟，选择主/从模式时除外（请参见 TIMx_SMCR 寄存器中 MSM 位的说明）。<br>010：更新 – 选择更新事件作为触发输出 (TRGO)。例如，主定时器可用作从定时器的预分频器。<br>011：比较脉冲 – 一旦发生输入捕获或比较匹配事件，当 CC1IF 被置 1 时（即使已为高电平），触发输出都会发送一个正脉冲。(TRGO)。<br>100：比较 – OC1REF 信号用作触发输出 (TRGO)<br>101：比较 – OC2REF 信号用作触发输出 (TRGO)<br>110：比较 – OC3REF 信号用作触发输出 (TRGO)<br>111：比较 – OC4REF 信号用作触发输出 (TRGO) |
| 3 | CCDS | 捕获/比较 DMA 选择<br>0：发生 CCx 事件时发送 CCx DMA 请求<br>1：发生更新事件时发送 CCx DMA 请求 |
| 2 | CCUS | 捕获/比较控制更新选择<br>0：如果捕获/比较控制位 (CCPC=1) 进行预装载，仅通过将 COMG 位置 1 来对这些位进行更新<br>1：如果捕获/比较控制位 (CCPC=1) 进行预装载，可通过将 COMG 位置 1 或 TRGI 的上升沿对这些位进行更新。<br>*注意：此位仅对具有互补输出的通道有效。* |
| 1 | 保留 | 必须保持复位值。 |
| 0 | CCPC | 捕获/比较预装载控制<br>0：CCxE、CCxNE 和 OCxM 位未进行预装载<br>1：CCxE、CCxNE 和 OCxM 位进行了预装载，写入这些位后，仅当发生换向事件 (COM)（COMG 位置 1 或在 TRGI 上检测到上升沿，取决于 CCUS 位）时才会对这些位进行更新。<br>*注意：此位仅对具有互补输出的通道有效。* |

### 14.4.3. TIM1 和 TIM8 从模式控制寄存器(TIMx_SMCR)

偏移地址：0x08

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ETP | ECS | ETPS[1:0] | | ETF[3:0] | | | | MSM | TS[2:0] | | | Res. | SMS[2:0] | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | Res. | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15 | ETP | 外部触发极性<br>此位可选择将 ETR 还是 ETR 用于触发操作<br>0：ETR 未反相，高电平或上升沿有效。<br>1：ETR 反相，低电平或下降沿有效。 |
| 14 | ECE | 外部时钟使能<br>此位可使能外部时钟模式 2。<br>0：禁止外部时钟模式 2<br>1：使能外部时钟模式 2。计数器时钟由 ETRF 信号的任意有效边沿提供。<br>注意：<br>1：将 ECE 位置 1 与选择外部时钟模式 1 并将 TRGI 连接到 ETRF（SMS=111 且 TS=111）具有相同效果。<br>2：外部时钟模式 2 可以和以下从模式同时使用：复位模式、门控模式和触发模式。不过此类情况下 TRGI 不得连接 ETRF(TS 位不得为 111)。<br>3：如果同时使能外部时钟模式 1 和外部时钟模式 2，则外部时钟输入为 ETRF。 |
| 13:12 | ETPS[1:0] | 外部触发预分频器<br>外部触发信号 ETRP 频率不得超过 TIMxCLK 频率的 1/4。可通过使能预分频器来降低 ETRP 频率。这种方法在输入快速外部时钟时非常有用。<br>00：预分频器关闭<br>01：2 分频 ETRP 频率<br>10：4 分频 ETRP 频率<br>11：8 分频 ETRP 频率 |
| 11:8 | ETF[3:0] | 外部触发滤波器<br>此位域可定义 ETRP 信号的采样频率和适用于 ETRP 的数字滤波时间。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿：<br>0000：无滤波器，按 fDTS 频率进行采样<br>0001：$f_{SAMPLING} = f_{CK_INT}$，N=2<br>0010：$f_{SAMPLING} = f_{CK_INT}$，N=4<br>0011：$f_{SAMPLING} = f_{CK_INT}$，N=8<br>0100：$f_{SAMPLING} = f_{DTS}/2$，N=6<br>0101：$f_{SAMPLING} = f_{DTS}/2$，N=8<br>0110：$f_{SAMPLING} = f_{DTS}/4$，N=6<br>0111：$f_{SAMPLING} = f_{DTS}/4$，N=8<br>1000：$f_{SAMPLING} = f_{DTS}/8$，N=6<br>1001：$f_{SAMPLING} = f_{DTS}/8$，N=8<br>1010：$f_{SAMPLING} = f_{DTS}/16$，N=5<br>1011：$f_{SAMPLING} = f_{DTS}/16$，N=6<br>1100：$f_{SAMPLING} = f_{DTS}/16$，N=8<br>1101：$f_{SAMPLING} = f_{DTS}/32$，N=5<br>1110：$f_{SAMPLING} = f_{DTS}/32$，N=6<br>1111：$f_{SAMPLING} = f_{DTS}/32$，N=8 |
| 7 | MSM | 主/从模式<br>0：不执行任何操作<br>1：当前定时器的触发输入事件 (TRGI) 的动作被推迟，以使当 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 前定时器与其从定时器实现完美同步（通过 TRGO）。此设置适用于由单个外部事件对多个定时器进行同步的情况。 |
| 6:4 | TS[2:0] | 触发选择<br>此位域可选择将要用于同步计数器的触发输入。<br>000：内部触发 0 (ITR0)<br>001：内部触发 1 (ITR1)<br>010：内部触发 2 (ITR2)<br>011：内部触发 3 (ITR3)<br>100：TI1 边沿检测器 (TI1F_ED)<br>101：滤波后的定时器输入 1 (TI1FP1)<br>110：滤波后的定时器输入 2 (TI2FP2)<br>111：外部触发输入 (ETRF)<br>*注意：这些位只能在未使用的情况下（例如，SMS=000 时）进行更改，以避免转换时出现错误的边沿检测。* |
| 3 | 保留 | 必须保持复位值。 |
| 2:0 | SMS | 从模式选择<br>选择外部信号时，触发信号 (TRGI) 的有效边沿与外部输入上所选的极性相关（请参见输入控制寄存器和控制寄存器说明）。<br>000: 禁止从模式 – 如果 CEN ="1"，预分频器时钟直接由内部时钟提供。<br>001：编码器模式 1 – 计数器根据 TI1FP1 电平在 TI2FP2 边沿递增/递减计数。<br>010：编码器模式 2 – 计数器根据 TI2FP2 电平在 TI1FP1 边沿递增/递减计数。<br>011：编码器模式 3 – 计数器在 TI1FP1 和 TI2FP2 的边沿计数，计数的方向取决于另外一个信号的电平。<br>100: 复位模式 – 在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器并生成一个寄存器更新事件。<br>101：门控模式 – 触发输入 (TRGI) 为高电平时使能计数器时钟。只要触发输入变为低电平，计数器立即停止计数（但不复位）。计数器的启动和停止都是受控的。<br>110：触发模式 – 触发信号 TRGI 出现上升沿时启动计数器（但不复位）。只控制计数器的启动。<br>111：外部时钟模式 1 – 由所选触发信号 (TRGI) 的上升沿提供计数器时钟。<br>*注意：如果将 TI1F_ED 选作触发输入（TS="100"），则不得使用门控模式。实际上，TI1F 每次转换时，TI1F_ED 都输出 1 个脉冲，而门控模式检查的则是触发信号的电平。* |

表 14-2 TIMx 内部触发连接

| 从 TIM | ITR0(TS = 000) | ITR1(TS = 001) | ITR2(TS = 010) | ITR3(TS = 011) |
|---|---|---|---|---|
| TIM1 | TIM5 | TIM2 | TIM3 | TIM4 |
| TIM8 | TIM1 | TIM2 | TIM4 | TIM5 |

### 14.4.4. TIM1 和 TIM8 DMA/中断使能寄存器(TIMx_DIER)

偏移地址：0x0C

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESET | TDE | COMDE | CC4DE | CC3DE | CC2DE | CC1DE | UDE | BIE | TIE | COMIE | CC4IE | CC3IE | CC2IE | CC1E | UIE |
| | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15 | 保留 | 必须保持复位值。 |
| 14 | TDE | 触发 DMA 请求使能<br>0：禁止 DMA 请求<br>1：使能 DMA 请求 |
| 13 | COMDE | COM DMA 请求使能<br>0：禁止 COM DMA 请求<br>1：使能 COM DMA 请求 |
| 12 | CC4DE | 捕获/比较 4 DMA 请求使能<br>0：禁止 CC4 DMA 请求<br>1：使能 CC4 DMA 请求 |
| 11 | CC3DE | 捕获/比较 3 DMA 请求使能<br>0：禁止 CC3 DMA 请求<br>1：使能 CC3 DMA 请求 |
| 10 | CC2DE | 捕获/比较 2 DMA 请求使能<br>0：禁止 CC2 DMA 请求<br>1：使能 CC2 DMA 请求 |
| 9 | CC1DE | 捕获/比较 1 DMA 请求使能<br>0：禁止 CC1 DMA 请求<br>1：使能 CC1 DMA 请求 |
| 8 | UDE | 更新 DMA 请求使能<br>0：禁止更新 DMA 请求<br>1：使能更新 DMA 请求 |
| 7 | BIE | 断路中断使能<br>0：禁止断路中断<br>1：使能断路中断 |
| 6 | TIE | 触发信号 (TGRI) 中断使能<br>0：禁止触发信号 (TGRI) 中断<br>1：使能触发信号 (TGRI) 中断 |
| 5 | COMIE | COM 中断使能<br>0：禁止 COM 中断<br>1：使能 COM 中断 |
| 4 | CC4IE | 捕获/比较 4 中断使能<br>0：禁止 CC4 中断<br>1：使能 CC4 中断 |
| 3 | CC3IE | 捕获/比较 3 中断使能<br>0：禁止 CC3 中断<br>1：使能 CC3 中断 |
| 2 | CC2IE | 捕获/比较 2 中断使能<br>0：禁止 CC2 中断<br>1：使能 CC2 中断 |
| 1 | CC1IE | 捕获/比较 1 中断使能<br>0：禁止 CC1 中断<br>1：使能 CC1 中断 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 0 | UIE | 更新中断使能<br>0：禁止更新中断<br>1：使能更新中断 |

### 14.4.5. TIM1 和 TIM8 状态寄存器(TIMx_SR)

偏移地址：0x10

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | CC4OF | CC3OF | CC2OF | CC1OF | Res. | BIF | TIF | COMIF | CC4IF | CC3IF | CC2IF | CC1IF | UIF |
| | | | rc_w0 | rc_w0 | rc_w0 | rc_w0 | Res. | rc_w0 | rc_w0 | rc_w0 | rc_w0 | rc_w0 | rc_w0 | rc_w0 | rc_w0 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:13 | 保留 | 必须保持复位值。 |
| 12 | CC4OF | 捕获/比较 4 重复捕获标志<br>请参见 CC1OF 说明 |
| 11 | CC3OF | 捕获/比较 3 重复捕获标志<br>请参见 CC1OF 说明 |
| 10 | CC2OF | 捕获/比较 2 重复捕获标志<br>请参见 CC1OF 说明 |
| 9 | CC1OF | 捕获/比较 1 重复捕获标志<br>仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将该位清零。<br>0：未检测到重复捕获。<br>1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。 |
| 8 | 保留 | 必须保持复位值。 |
| 7 | BIF | 断路中断标志<br>只要断路输入变为有效状态，此标志便由硬件置 1。断路输入无效后可通过软件对其清零。<br>0：未发生断路事件。<br>1：在断路输入上检测到有效电平。 |
| 6 | TIF | 触发中断标志<br>在除门控模式以外的所有模式下，当使能从模式控制器后在 TRGI 输入上检测到有效边沿时，该标志将由硬件置 1。选择门控模式时，该标志将在计数器启动或停止时置 1。但需要通过软件清零。<br>0：未发生触发事件。<br>1：触发中断挂起。 |
| 5 | COMIF | COM 中断标志<br>此标志在发生 COM 事件时（捕获/比较控制位 CCxE、CCxNE 和 OCxM 已更新时）由硬件置 1。但需要通过软件清零。<br>0：未发生 COM 事件。<br>1：COM 中断挂起。 |
| 4 | CC4IF | 捕获/比较 4 中断标志<br>请参见 CC1IF 说明 |
| 3 | CC3IF | 捕获/比较 3 中断标志<br>请参见 CC1IF 说明 |
| 2 | CC2IF | 捕获/比较 2 中断标志 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 请参见 CC1IF 说明 |
| 1 | CC1IF | 捕获/比较 1 中断标志<br>**如果通道 CC1 配置为输出：**<br>当计数器与比较值匹配时，此标志由硬件置 1，中心对齐模式下除外（请参见 TIMx_CR1 寄存器中的 CMS 位说明）。但需要通过软件清零。<br>0：不匹配。<br>1：TIMx_CNT 计数器的值与 TIMx_CCR1 寄存器的值匹配。当 TIMx_CCR1 的值大于 TIMx_ARR 的值时，CC1IF 位将在计数器发生上溢（递增计数模式和增减计数模式下）或下溢（递减计数模式下）时变为高电平。<br>**如果通道 CC1 配置为输入：**<br>此位将在发生捕获事件时由硬件置 1。通过软件或读取 TIMx_CCR1 寄存器将该位清零。<br>0：未发生输入捕获事件<br>1：TIMx_CCR1 寄存器中已捕获到计数器值（IC1 上已检测到与所选极性匹配的边沿） |
| 0 | UIF | 更新中断标志<br>该位在发生更新事件时通过硬件置 1。但需要通过软件清零。<br>0：未发生更新。<br>1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：<br>TIMx_CR1 寄存器中的 UDIS=0，并且重复计数器值上溢或下溢时（重复计数器 = 0 时更新）。<br>TIMx_CR1 寄存器中的 URS = 0 且 UDIS = 0，并且由软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。<br>TIMx_CR1 寄存器中的 URS=0 且 UDIS=0，并且 CNT 由触发事件重新初始化时 |

### 14.4.6. TIM1 和 TIM8 事件生成寄存器(TIMx_EGR)

偏移地址：0x14

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | BG | TG | COMG | CC4G | CC3G | CC2G | CC1G | UG |
| | | | | | | | | w | w | w | w | w | w | w | w |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:8 | 保留 | 必须保持复位值。 |
| 7 | BG | 断路生成<br>此位由软件置 1 以生成事件，并由硬件自动清零。<br>0：不执行任何操作<br>1：生成断路事件。MOE 位清零且 BIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。 |
| 6 | TG | 生成触发信号<br>此位由软件置 1 以生成事件，并由硬件自动清零。<br>0：不执行任何操作<br>1：TIMx_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 或 DMA 传输事件。 |
| 5 | COMG | 捕获/比较控制更新生成<br>该位可通过软件置 1，并由硬件自动清零<br>0：不执行任何操作<br>1：CCPC 位置 1 时，可更新 CCxE、CCxNE 和 OCxM 位<br>*注意： 此位仅对具有互补输出的通道有效。* |
| 4 | CC4G | 捕获/比较 4 生成<br>请参见 CC1G 说明 |
| 3 | CC3G | 捕获/比较 3 生成<br>请参见 CC1G 说明 |
| 2 | CC2G | 捕获/比较 2 生成<br>请参见 CC1G 说明 |
| 1 | CC1G | 捕获/比较 1 生成<br>此位由软件置 1 以生成事件，并由硬件自动清零。<br>0：不执行任何操作<br>1：通道 1 上生成捕获/比较事件：<br>**如果通道 CC1 配置为输出：**<br>使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。<br>**如果通道 CC1 配置为输入：**<br>TIMx_CCR1 寄存器中将捕获到计数器当前值。使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。 |
| 0 | UG | 更新生成<br>该位可通过软件置 1，并由硬件自动清零。<br>0：不执行任何操作<br>1：重新初始化计数器并生成一个寄存器更新事件。<br>请注意，预分频器计数器也将清零（但预分频比不受影响）。如果选择中心对齐模式或 DIR=0（递增计数），计数器将清零；如果 DIR=1（递减计数），计数器将使用自动重载值 (TIMx_ARR)。 |

### 14.4.7. TIM1 和 TIM8 捕获/比较模式寄存器 1(TIMx_CCMR1)

偏移地址：0x18

复位值：0x0000

这些通道可用于输入（捕获模式）或输出（比较模式）模式。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入模式和输出模式下的功能均不同。对于任一给定位，OCxx 用于说明通道配置为输出时该位对应的功能，ICxx 则用于说明通道配置为输入时该位对应的功能。因此，必须注意同一个位在输入阶段和输出阶段具有不同的含义。

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OC2CE | OC2M[2:0] | | | OC2PE | OC2FE | CC2S[1:0] | | OC1CE | OC1M[2:0] | | | OC1PE | OC1FE | CC1S[1:0] | |
| IC2F[3:0] | | | | IC2PSC[1:0] | | | | IC1F[3:0] | | | | IC1PSC[1:0] | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

**输出比较模式：**

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15 | OC2CE | 输出比较 2 清零使能 |
| 14:12 | OC2M[2:0] | 输出比较 2 模式 |
| 11 | OC2PE | 输出比较 2 预装载使能 |
| 10 | OC2FE | 输出比较 2 快速使能 |
| 9:8 | CC2S[1:0] | 捕获/比较 2 选择<br>此位域定义通道方向（输入/输出）以及所使用的输入。<br>00：CC2 通道配置为输出<br>01：CC2 通道配置为输入，IC2 映射到 TI2 上<br>10：CC2 通道配置为输入，IC2 映射到 TI1 上<br>11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效<br>*注意： 仅当通道关闭时（TIMx_CCER 中的 CC2E =“0”），才可向 CC2S 位写入数据。* |
| 7 | OC1CE | 输出比较 1 清零使能<br>0：OC1Ref 不受 ETRF 输入影响<br>1：ETRF 输入上检测到高电平时，OC1Ref 立即清零。 |
| 6:4 | OC1M | 输出比较 1 模式<br>这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 和 OC1N 的有效电平则取决于 CC1P 位和 CC1NP 位。<br>000：冻结 – 输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。（该模式用于生成时基）。<br>001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。<br>010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。<br>011：翻转 – TIMx_CNT=TIMx_CCR1 时，OC1REF 发生翻转。<br>100：强制变为无效电平 – OC1REF 强制变为低电平。<br>101：强制变为有效电平 – OC1REF 强制变为高电平。<br>110：PWM 模式 1 – 在递增计数模式下，只要 TIMx_CNT < TIMx_CCR1，通道 1 便为有效状态，否则为无效状态。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，通道 1 便为无效状态（OC1REF =“0”），否则为有效状态（OC1REF=“1”）。<br>111：PWM 模式 2 – 在递增计数模式下，只要 TIMx_CNT < TIMx_CCR1，通道 1 便为无效状态，否则为有效状态。在递减计数模式下，只要 TIMx_CNT > TIMx_CCR1，通道 1 便为有效状态，否则为无效状态。<br>注意：<br>1：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S=“00”（通道配置为输出），这些位即无法修改。<br>2：在 PWM 模式 1 或 PWM 模式 2 下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，OCREF 电平才会发生更改。<br>3：此位域将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成 COM 事件时，OC1M 有效位才会从预装载位获取新值。 |
| 3 | OC1PE | 输出比较 1 预装载使能 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 0: 禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。<br>1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到活动寄存器中。<br>注意：<br>1：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S="00"（通道配置为输出），这些位即无法修改。<br>2：只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式（TIMx_CR1 寄存器中的 OPM 位置 1）。其它情况下则无法保证该行为。 |
| 2 | OC1FE | 输出比较 1 快速使能<br>此位用于加快触发输入事件对 CC 输出的影响。<br>0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期。<br>1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OCFE 才会起作用。 |
| 1:0 | CC1S | 捕获/比较 1 选择<br>此位域定义通道方向（输入/输出）以及所使用的输入。<br>00：CC1 通道配置为输出<br>01：CC1 通道配置为输入，IC1 映射到 TI1 上<br>10：CC1 通道配置为输入，IC1 映射到 TI2 上<br>11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效<br>注意： 仅当通道关闭时（TIMx_CCER 中的 CC1E ="0"），才可向 CC1S 位写入数据。 |

**输入捕获模式：**

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:12 | IC2F | 输入捕获 2 滤波器 |
| 11:10 | IC2PSC[1:0] | 输入捕获 2 预分频器 |
| 9:8 | CC2S | 捕获/比较 2 选择<br>此位域定义通道方向（输入/输出）以及所使用的输入。<br>00：CC2 通道配置为输出<br>01：CC2 通道配置为输入，IC2 映射到 TI2 上<br>10：CC2 通道配置为输入，IC2 映射到 TI1 上<br>11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效<br>*注意： 仅当通道关闭时（TIMx_CCER 中的 CC2E ="0"），才可向 CC2S 位写入数据。* |
| 7:4 | IC1F[3:0] | 输入捕获 1 滤波器<br>此位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿：<br>0000：无滤波器，按 fDTS 频率进行采样<br>0001：$f_{SAMPLING} = f_{CK_INT}$，N=2<br>0010：$f_{SAMPLING} = f_{CK_INT}$，N=4 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 0011：$f_{SAMPLING} = f_{CK_INT}$，N=8<br>0100：$f_{SAMPLING} = f_{DTS}/2$，N=6<br>0101：$f_{SAMPLING} = f_{DTS}/2$，N=8<br>0110：$f_{SAMPLING} = f_{DTS}/4$，N=6<br>0111：$f_{SAMPLING} = f_{DTS}/4$，N=8<br>1000：$f_{SAMPLING} = f_{DTS}/8$，N=6<br>1001：$f_{SAMPLING} = f_{DTS}/8$，N=8<br>1010：$f_{SAMPLING} = f_{DTS}/16$，N=5<br>1011：$f_{SAMPLING} = f_{DTS}/16$，N=6<br>1100：$f_{SAMPLING} = f_{DTS}/16$，N=8<br>1101：$f_{SAMPLING} = f_{DTS}/32$，N=5<br>1110：$f_{SAMPLING} = f_{DTS}/32$，N=6<br>1111：$f_{SAMPLING} = f_{DTS}/32$，N=8 |
| 3:2 | IC1PSC | 输入捕获 1 预分频器<br>此位域定义 CC1 输入 (IC1) 的预分频比。<br>只要 CC1E=“0”（TIMx_CCER 寄存器），预分频器便立即复位。<br>00：无预分频器，捕获输入上每检测到一个边沿便执行捕获<br>01：每发生 2 个事件便执行一次捕获<br>10：每发生 4 个事件便执行一次捕获<br>11：每发生 8 个事件便执行一次捕获 |
| 1:0 | CC1S | 捕获/比较 1 选择<br>此位域定义通道方向（输入/输出）以及所使用的输入。<br>00：CC1 通道配置为输出<br>01：CC1 通道配置为输入，IC1 映射到 TI1 上<br>10：CC1 通道配置为输入，IC1 映射到 TI2 上<br>11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效<br>*注意： 仅当通道关闭时（TIMx_CCER 中的 CC1E =“0”），才可向 CC1S 位写入数据。* |

### 14.4.8. TIM1 和 TIM8 捕获/比较模式寄存器 2(TIMx_CCMR2)

偏移地址：0x18

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OC4CE | OC4M[2:0] | | | OC4PE | OC4FE | CC4S[1:0] | | OC3CE | OC1M[2:0] | | | OC3PE | OC3FE | CC3S[1:0] | |
| IC4F[3:0] | | | | IC4PSC[1:0] | | | | IC3F[3:0] | | | | IC3PSC[1:0] | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

**输出比较模式：**

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15 | OC4CE | 输出比较 4 清零使能 |
| 14:12 | OC4M | 输出比较 4 模式 |
| 11 | OC4PE | 输出比较 4 预装载使能 |
| 10 | OC4FE | 输出比较 4 快速使能 |
| 9:8 | CC4S | 捕获/比较 4 选择<br>此位域定义通道方向（输入/输出）以及所使用的输入。 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 00：CC4 通道配置为输出<br>01：CC4 通道配置为输入，IC4 映射到 TI4 上<br>10：CC4 通道配置为输入，IC4 映射到 TI3 上<br>11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效<br>*注意：仅当通道关闭时（TIMx_CCER 中的 CC4E =“0”），才可向 CC4S 位写入数据。* |
| 7 | OC3CE | 输出比较 3 清零使能 |
| 6:4 | OC3M | 输出比较 3 模式 |
| 3 | OC3PE | 输出比较 3 预装载使能 |
| 2 | OC3FE | 输出比较 3 快速使能 |
| 1:0 | CC3S | 捕获/比较 3 选择<br>此位域定义通道方向（输入/输出）以及所使用的输入。<br>00：CC3 通道配置为输出<br>01：CC3 通道配置为输入，IC3 映射到 TI3 上<br>10：CC3 通道配置为输入，IC3 映射到 TI4 上<br>11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效<br>*注意：仅当通道关闭时（TIMx_CCER 中的 CC3E =“0”），才可向 CC3S 位写入数据。输入捕获模式* |

**输入捕获模式：**

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 15:12 | IC4F | 输入捕获 4 滤波器 |
| 11:10 | IC4PSC | 输入捕获 4 预分频器 |
| 9:8 | CC4S | 捕获/比较 4 选择<br>此位域定义通道方向（输入/输出）以及所使用的输入。<br>00：CC4 通道配置为输出<br>01：CC4 通道配置为输入，IC4 映射到 TI4 上<br>10：CC4 通道配置为输入，IC4 映射到 TI3 上<br>11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效<br>*注意：仅当通道关闭时（TIMx_CCER 中的 CC4E =“0”），才可向 CC4S 位写入数据。* |
| 7:4 | IC3F | 输入捕获 3 滤波器 |
| 3:2 | IC3PSC | 输入捕获 3 预分频器 |
| 1:0 | CC3S | 捕获/比较 3 选择<br>此位域定义通道方向（输入/输出）以及所使用的输入。<br>00：CC3 通道配置为输出<br>01：CC3 通道配置为输入，IC3 映射到 TI3 上<br>10：CC3 通道配置为输入，IC3 映射到 TI4 上<br>11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | *注意： 仅当通道关闭时（TIMx_CCER 中的 CC3E =“0”），才可向 CC3S 位写入数据。* |

### 14.4.9. TIM1 和 TIM8 捕获/比较使能寄存器(TIMx_CCER)

偏移地址：0x20

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Reserved | | CC4P | CC4E | CC3NP | CC3NE | CC3P | CC3E | CC2NP | CC2NE | CC2P | CC2E | CC1NP | CC1NE | CC1P | CC1E |
| | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 15:14 | 保留 | 必须保持复位值。 |
| 13 | CC4P | 捕获/比较 4 输出极性<br>请参见 CC1P 说明 |
| 12 | CC4E | 捕获/比较 4 输出使能<br>请参见 CC1E 说明 |
| 11 | CC3NP | 捕获/比较 3 互补输出极性<br>请参见 CC1NP 说明 |
| 10 | CC3NE | 捕获/比较 3 互补输出使能<br>请参见 CC1NE 说明 |
| 9 | CC3P | 捕获/比较 3 输出极性<br>请参见 CC1P 说明 |
| 8 | CC3E | 捕获/比较 3 输出使能<br>请参见 CC1E 说明 |
| 7 | CC2NP | 捕获/比较 2 互补输出极性<br>请参见 CC1NP 说明 |
| 6 | CC2NE | 捕获/比较 2 互补输出使能<br>请参见 CC1NE 说明 |
| 5 | CC2P | 捕获/比较 2 输出极性<br>请参见 CC1P 说明 |
| 4 | CC2E | 捕获/比较 2 输出使能<br>请参见 CC1E 说明 |
| 3 | CC1NP | 捕获/比较 1 互补输出极性<br>CC1 通道配置为输出：<br>0：OC1N 高电平有效。<br>1：OC1N 低电平有效。<br>CC1 通道配置为输入：<br>此位与 CC1P 配合使用，用以定义 TI1FP1 和 TI2FP1 的极性。<br>请参见 CC1P 说明。<br>*注意： 此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1NP 有效位才会从预装载位获取新值。*<br>*注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3 且 CC1S=“00”（通道配置为输出），此位立即变为不可写状态。* |
| 2 | CC1NE | 捕获/比较 1 互补输出使能<br>0：关闭 —OC1N 未激活。OC1N 电平是 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的函数。 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 1：开启 – 在相应输出引脚上输出 OC1N 信号，具体取决于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位。<br>*注意：此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时， CC1NE 有效位才会从预装载位获取新值。* |
| 1 | CC1P | 捕获/比较 1 输出极性<br>**CC1 通道配置为输出：**<br>0：OC1 高电平有效<br>1：OC1 低电平有效<br>**CC1 通道配置为输入：**<br>CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的有效极性。<br>00：非反相/上升沿触发<br>电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式或编码器模式下执行触发操作）。<br>01：反相/下降沿触发<br>电路对 TIxFP1 下升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式或编码器模式下执行触发操作）。<br>10：保留，不使用此配置。<br>11：未反相/边沿触发。<br>电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。编码器模式下不得使用此配置。<br>*注意：此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1P 有效位才会从预装载位获取新值。*<br>*注意： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3，此位立即变为不可写状态。* |
| 0 | CC1E | 捕获/比较 1 输出使能<br>**CC1 通道配置为输出：**<br>0: 关闭 – OC1 未激活。OC1 电平是 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的函数。<br>1：开启 – OC1 信号是相应输出引脚上的输出，具体取决于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位。<br>**CC1 通道配置为输入：**<br>此位决定了是否可以实际将计数器值捕获到输入捕获/比较寄存器 1 (TIMx_CCR1) 中。<br>0：禁止捕获。<br>1：使能捕获。<br>*注意：此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时， CC1E 有效位才会从预装载位获取新值* |

表 14-3 具有断路功能的互补通道 OCx 和 OCxN 的输出控制位

| 控制位 | | | | | 输出状态 | |
| --- | --- | --- | --- | --- | --- | --- |
| MOE 位 | OSSI 位 | OSSR 位 | CCxE 位 | CCxNE 位 | OCx 输出状态 | OCxN 输出状态 |
| 1 | X | 0 | 0 | 0 | 禁止输出（不由定时器驱动）<br>OCx=0,OCx_EN=0 | 禁止输出（不由定时器驱动）<br>OCxN=0,OCxN_EN=0 |

| 控制位 | | | | | 输出状态 | |
|---|---|---|---|---|---|---|
| | | 0 | 0 | 1 | 禁止输出（不由定时器驱动）<br>OCx=0,OCx_EN=0 | OCxREF+ 极性<br>OCxN=OCxREF 异或<br>CCxNP,OCxN_EN=1 |
| | | 0 | 1 | 0 | OCxREF+极性<br>OCx_=OCxREF 异或 CCxP、<br>OCx_EN=1 | 禁止输出（不由定时器驱动）<br>OCxN=0,OCxN_EN=0 |
| | | 0 | 1 | 1 | CCxREF+极性+死区<br>OCx_EN=1 | OCxREF 互补项（而非 OCREF）+极性+死区<br>OCxN_EN=1 |
| | | 1 | 0 | 0 | 禁止输出（不由定时器驱动）<br>OCx=CCxP,OCx_EN=0 | 禁止输出（不由定时器驱动）<br>OCxN=CCxNP,OCxN_EN=0 |
| | | 1 | 0 | 1 | 关闭状态（输出使能位无效状态）<br>OCx=CCxP,OCx_EN=1 | OCxREF+极性<br>OCxN_=OCxREF 异或<br>CCxNP、OCxN_EN=1 |
| | | 1 | 1 | 0 | OCxREF+极性<br>OCx_=OCxREF 异或 CCxP、<br>OCx_EN=1 | 关闭状态(输出使能位无效状态）<br>OCxN=CCxNP,OCxN_EN=1 |
| | | 1 | 1 | 1 | OCxREF+极性+死区<br>OCx_EN=1 | OCREF 互补项（而非 OCREF）+极性+死区<br>OCxN_EN=1 |
| 0 | 0 | X | 0 | 0 | 禁止输出（不由定时器驱动）<br>OCx=CCxP,OCx_EN=0 | 禁止输出（不由定时器驱动）<br>OCxN=CCxNP,OCxN_EN=0 |
| | 0 | | 0 | 1 | 禁止输出（不由定时器驱动）<br>异步：OCx=CCxP,Ocx_EN=0,OCxN=CCxNP,OCxN_EN=0<br>如果存在时钟，在死区后 OCx=OISx 且 OCxN=OISxN,从而假定 OISx 和 OISxN 在有效状态下与 OCX 和 OCxN 不对应 | |
| | 0 | | 1 | 0 | | |
| | 0 | | 1 | 1 | | |
| | 1 | | 0 | 0 | 禁止输出（不由定时器驱动）<br>OCx=CCxP,OCx_EN=0 | 禁止输出（不由定时器驱动）<br>OCxN=CCxNP,OCxN_EN=0 |
| | 1 | | 0 | 1 | 关闭状态（输出使能位无效状态）<br>异步：OCx=CCxP,OCx_EN=1,OCxN=CCxNP,OCxN_EN=1<br>如果存在时钟，在死区后 Ocx=OISx 且 OCxN=OISxN,从而假定 OISx 和 OISxN 在有效状态下与 OCX 和 OCxN 不对应 | |
| | 1 | | 1 | 0 | | |
| | 1 | | 1 | 1 | | |

## 14.4.10. TIM1 和 TIM8 计数器(TIMx_CNT)

偏移地址：0x24

复位值：0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:0 | CNT[15:0] | 计数器值 |

### 14.4.11. TIM1 和 TIM8 预分频器(TIMx_PSC)

偏移地址：0x28

复位值：0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:0 | PSC[15:0] | 预分频器值<br>计数器时钟频率 (CK_CNT) 等于 $f_{CK_PSC}$ / (PSC[15:0] + 1)。<br>PSC 包含每次发生更新事件（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时，或在配置为“复位模式”时通过触发控制器清零时）时要装载到活动预分频器寄存器的值。 |

### 14.4.12. TIM1 和 TIM8 自动重载寄存器(TIMx_ARR)

偏移地址：0x2C

复位值：0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:0 | ARR[15:0] | 自动重载值<br>ARR 为要装载到实际自动重载寄存器的值。<br>当自动重载值为空时，计数器不工作。 |

### 14.4.13. TIM1 和 TIM8 重复计数器寄存器(TIMx_RCR)

偏移地址：0x30

复位值：0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:8 | 保留 | 必须保持复位值。 |
| 7:0 | REP[7:0] | 重复计数器值<br>使能预装载寄存器时，用户可通过这些位设置比较寄存器的更新频率（即从预装载寄存器向活动寄存器周期性传输数据）；使能更新中断时，也可设置更新中断的生成速率。<br>与 REP_CNT 相关的减计数器每次计数到 0 时，都将生成一个更新事件并且计数器从 REP 值重新开始计数。由于只有生成重复更新事件 U_RC 时，REP_CNT 才会重载 REP |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 值, 因此在生成下一重复更新事件之前, 无论向 TIMx_RCR 寄存器写入何值都无影响。<br>这意味着 PWM 模式下 (REP+1) 相当于:<br>边沿对齐模式下的 PWM 周期数<br>中心对齐模式下的 PWM 半周期数 |

### 14.4.14. TIM1 和 TIM8 捕获/比较寄存器 1 (TIMx_CCR1)

偏移地址：0x34

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CCR1[15:0] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 15:0 | CCR1[15:0] | 捕获/比较 1 值<br>如果通道 CC1 配置为输出:<br>CCR1 是捕获/比较寄存器 1 的预装载值。如果没有通过 TIMx_CCMR 寄存器中的 OC1PE 位来使能预装载功能, 写入的数值会被直接传输至当前寄存器中。否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器 1）。<br>实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC1 输出上发出信号的值。<br>如果通道 CC1 配置为输入:<br>CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。 |

### 14.4.15. TIM1 和 TIM8 捕获/比较比较器 2 (TIMx_CCR2)

偏移地址：0x38

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CCR2[15:0] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 15:0 | CCR2[15:0] | 捕获/比较 2 值<br>如果通道 CC2 配置为输出:<br>CCR2 为要装载到实际捕获/比较 2 寄存器的值（预装载值）。<br>如果没有通过 TIMx_CCMR2 寄存器中的 OC2PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 2 寄存器中。<br>实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 较并在 OC2 输出上发出信号的值。<br>如果通道 CC2 配置为输入：<br>CCR2 为上一个输入捕获 2 事件 (IC2) 发生时的计数器值。 |

### 14.4.16. TIM1 和 TIM8 捕获/比较比较器 3 (TIMx_CCR3)

偏移地址：0x3C

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CCR3[15:0] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 15:0 | CCR3[15:0] | 捕获/比较值<br>如果通道 CC3 配置为输出：<br>CCR3 为要装载到实际捕获/比较 3 寄存器的值（预装载值）。<br>如果没有通过 TIMx_CCMR3 寄存器中的 OC3PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 3 寄存器中。<br>实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC3 输出上发出信号的值。<br>如果通道 CC3 配置为输入：<br>CCR3 为上一个输入捕获 3 事件 (IC3) 发生时的计数器值。 |

### 14.4.17. TIM1 和 TIM8 捕获/比较比较器 4 (TIMx_CCR4)

偏移地址：0x40

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CCR4[15:0] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 15:0 | CCR4[15:0] | 捕获/比较值<br>如果通道 CC4 配置为输出：<br>CCR4 为要装载到实际捕获/比较 4 寄存器的值（预装载值）。<br>如果没有通过 TIMx_CCMR4 寄存器中的 OC4PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 4 寄存器中。<br>实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC4 输出上发出信号的值。<br>如果通道 CC4 配置为输入：<br>CCR4 为上一个输入捕获 4 事件 (IC4) 发生时的计数器值。 |

### 14.4.18. TIM1 和 TIM8 断路和死区寄存器(TIMx_BDTR)

偏移地址：0x44

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MOE | AOE | BKP | BKE | OSSR | OSSI | LOCK[1:0] | | DTG[7:0] | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15 | MOE | 主输出使能<br>只要断路输入变为有效状态，此位便由硬件异步清零。此位由软件置 1，也可根据 AOE 位状态自动置 1。此位仅对配置为输出的通道有效。<br>0：OC 和 OCN 输出禁止或被强制为空闲状态。<br>1：如果 OC 和 OCN 输出相应使能位（TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位）均置 1，则使能 OC 和 OCN 输出。 |
| 14 | AOE | 自动输出使能<br>0：MOE 只能由软件置 1<br>1：MOE 可由软件置 1，也可在发生下一更新事件时自动置 1(如果断路输入无效)<br>*注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。* |
| 13 | BKP | 断路极性<br>0：断路输入 BRK 为低电平有效<br>1：断路输入 BRK 为高电平有效<br>*注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。*<br>*注意：对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。* |
| 12 | BKE | 断路使能<br>0：禁止断路输入（BRK 和 CSS 时钟故障事件）<br>1：使能断路输入（BRK 和 CSS 时钟故障事件）<br>*注意：编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。*<br>*注意：对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。* |
| 11 | OSSR | 运行模式下的关闭状态选择<br>此位在 MOE=1 时作用于配置为输出模式且具有互补输出的通道。如果定时器中没有互补输出，则不存在 OSSR。<br>0：处于无效状态时，禁止 OC/OCN 输出（OC/OCN 使能输出信号=0）。<br>1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便使能 OC/OCN 输出并将其设为无效电平。然后设置 OC/OCN 使能输出信号=1<br>*注意：编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 后，此位即无法修改。* |
| 10 | OSSI | 空闲模式下的关闭状态选择<br>此位在 MOE=0 时作用于配置为输出的通道。<br>0：处于无效状态时，禁止 OC/OCN 输出（OC/OCN 使能输出信号=0）。<br>1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便将 OC/OCN 输出首先强制为其空闲电平。然后设置 OC/OCN 使能输出信号=1 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | *注意：编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 后，此位即无法修改。* |
| 9:8 | LOCK[1:0] | 锁定配置<br>这些位用于针对软件错误提供写保护。<br>00：关闭锁定 – 不对任何位提供写保护。<br>01：锁定级别 1，此时无法对 TIMx_BDTR 寄存器中的 DTG 位、TIMx_CR2 寄存器中的 OISx 和 OISxN 位以及 TIMx_BDTR 寄存器中的 BKE/BKP/AOE 位执行写操作。<br>10：锁定级别 2，此时无法对锁定级别 1 中适用的各位、CC 极性位（TIMx_CCER 寄存器中的 CCxP/CCxNP 位，只要通过 CCxS 位将相关通道配置为输出）以及 OSSR 和 OSSI 位执行写操作。<br>11：锁定级别 3，此时无法对锁定级别 2 中适用的各位、CC 控制位（TIMx_CCMRx 寄存器中的 OCxM 和 OCxPE 位，只要通过 CCxS 位将相关通道配置为输出）执行写操作。<br>*注意：复位后只能对 LOCK 位执行一次写操作。对 TIMx_BDTR 寄存器执行写操作后其中的内容将冻结，直到下一次复位。* |
| 7:0 | DTG[7:0] | 配置死区发生器<br>此位域定义插入到互补输出之间的死区持续时间。DT 与该持续时间相对应。<br>DTG[7:5] = 0xx => DT=DTG[7:0] x $t_{dtg}$，其中 $t_{dtg}$ = $t_{DTS}$。<br>DTG[7:5] = 10x => DT=(64+DTG[5:0]) x $t_{dtg}$，其中 $t_{dtg}$ = 2 x $t_{DTS}$。<br>DTG[7:5] = 110 => DT=(32+DTG[4:0]) x $t_{dtg}$，其中 $t_{dtg}$ = 8 x $t_{DTS}$。<br>DTG[7:5] = 111 => DT=(32+DTG[4:0]) x $t_{dtg}$，其中 $t_{dtg}$ = 16 x $t_{DTS}$。<br>示例：如果 $T_{DTS}$=125ns (8MHz)，则可能的死区值为：<br>0 到 15875 ns（步长为 125 ns），<br>16 us 到 31750 ns（步长为 250 ns），<br>32 us 到 63us（步长为 1 us），<br>64 us 到 126 us（步长为 2 us）<br>*注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，此位域即无法修改。* |

### 14.4.19. TIM1 和 TIM8 DMA 控制寄存器(TIMx_DCR)

偏移地址：0x48

复位值：0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:13 | 保留 | 必须保持复位值。 |
| 12:8 | DBL[4:0] | DMA 连续传送长度<br>该 5 位定义了 DMA 在连续模式下的传送长度（当对 TIMx_DMAR 寄存器进行读或写时，定时器则进行一次连续传送）。<br>00000：1 次传输 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 00001：2 次传输<br>00010：3 次传输<br>…<br>10001：18 次传输 |
| 7:5 | 保留 | 必须保持复位值。 |
| 4:0 | DBA[4:0] | DMA 基址<br>该 5 位向量定义 DMA 传输的基址（通过 TIMx_DMAR 地址进行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。<br>示例：<br>00000：TIMx_CR1，<br>00001：TIMx_CR2，<br>00010：TIMx_SMCR，<br>…<br>示例：考虑以下传输：DBL = 7 次传输，DBA = TIMx_CR1。这种情况下将向/从自 TIMx_CR1 地址开始的 7 个寄存器传输数据。 |

### 14.4.20. TIM1 和 TIM8 全传输 DMA 地址(TIMx_DMAR)

偏移地址：0x4C

复位值：0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:0 | DMAB[15:0] | DMA 连续传送寄存器<br>对 DMAR 寄存器执行读或写操作将访问位于如下地址的寄存器 (TIMx_CR1 地址) + (DBA + DMA 索引) x 4 其中 TIMx_CR1 地址为控制寄存器 1 的地址，DBA 为 TIMx_DCR 寄存器中配置的 DMA 基址，DMA 索引由 DMA 传输自动控制，其范围介于 0 到 DBL（TIMx_DCR 寄存器中配置的 DBL）之间。 |

**DMA 连续传送功能使用方法示例**

本例中，定时器 DMA 连续传送功能用于将 CCRx 寄存器（x = 2、3、4）的内容更新为通过 DMA 传输到 CCRx 寄存器中的多个半字。

具体操作步骤如下：

- 将相应的 DMA 通道配置如下：
  - DMA 通道外设地址为 DMAR 寄存器地址。
  - DMA 通道存储器地址为包含要通过 DMA 传输到 CCRx 寄存器的数据的 RAM 缓冲区地址。
  - 要传输的数据量 = 3（参见下文注释）。
  - 禁止循环模式。
- 通过将 DBA 和 DBL 位域配置如下来配置 DCR 寄存器：

- DBL = 3 次传输，DBA = 0xE。

- 使能 TIMx 更新 DMA 请求（DIER 寄存器中的 UDE 位置 1）。
- 使能 TIMx 。
- 使能 DMA 通道。

*注意：本例适用于每个 CCRx 寄存器只更新一次的情况。如果每个 CCRx 寄存器要更新两次，则要传输的数据量应为 6。下面以包含 data1、data2、data3、data4、data5 和 data6 的 RAM 缓冲区为例。数据将按照如下方式传输到 CCRx 寄存器：在第一个更新 DMA 请求期间，data1 传输到 CCR2，data2 传输到 CCR3，data3 传输到 CCR4；在第二个更新 DMA 请求期间，data4 传输到 CCR2，data5 传输到 CCR3，data6 传输到 CCR4。*

# 15. 通用定时器(TIM2/TIM3/TIM4/TIM5)

## 15.1. TIM2 到 TIM5 简介

通用定时器包含一个 16 位或 32 位自动重载计数器，该计数器由可编程预分频器驱动。
它们可用于多种用途，包括测量输入信号的脉冲宽度（输入捕获）或生成输出波形（输出比较和 PWM）。
使用定时器预分频器和 RCC 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。
这些定时器彼此完全独立，不共享任何资源，可以同步操作。

## 15.2. TIM2 到 TIM5 主要特性

通用 TIMx 定时器具有以下特性：

- 16 位(TIM3 和 TIM4)或 32 位(TIM2 和 TIM5)递增、递减和递增/递减自动重载计数器。
- 16 位可编程预分频器，用于对计数器时钟频率进行分频(即运行时修改)，分频系数介于 1 到 65536 之间。
- 多达 4 个独立通道，可用于：
  - 输入捕获。
  - 输出比较。
  - PWM 生成(边沿和中心对齐模式)。
  - 单脉冲模式输出。
- 使用外部信号控制定时器且可实现多个定时器互连的同步电路。
- 发生如下事件时生成中断/DMA 请求：
  - 更新：计数器上溢/下溢、计数器初始化(通过软件或内部/外部触发)。
  - 触发事件(计数器启动、停止、初始化或通过内部/外部触发计数)。
  - 输入捕获。
  - 输出比较。
- 支持定位用增量(正交)编码器和霍尔传感器电路
- 外部时钟触发输入或逐周期电流管理

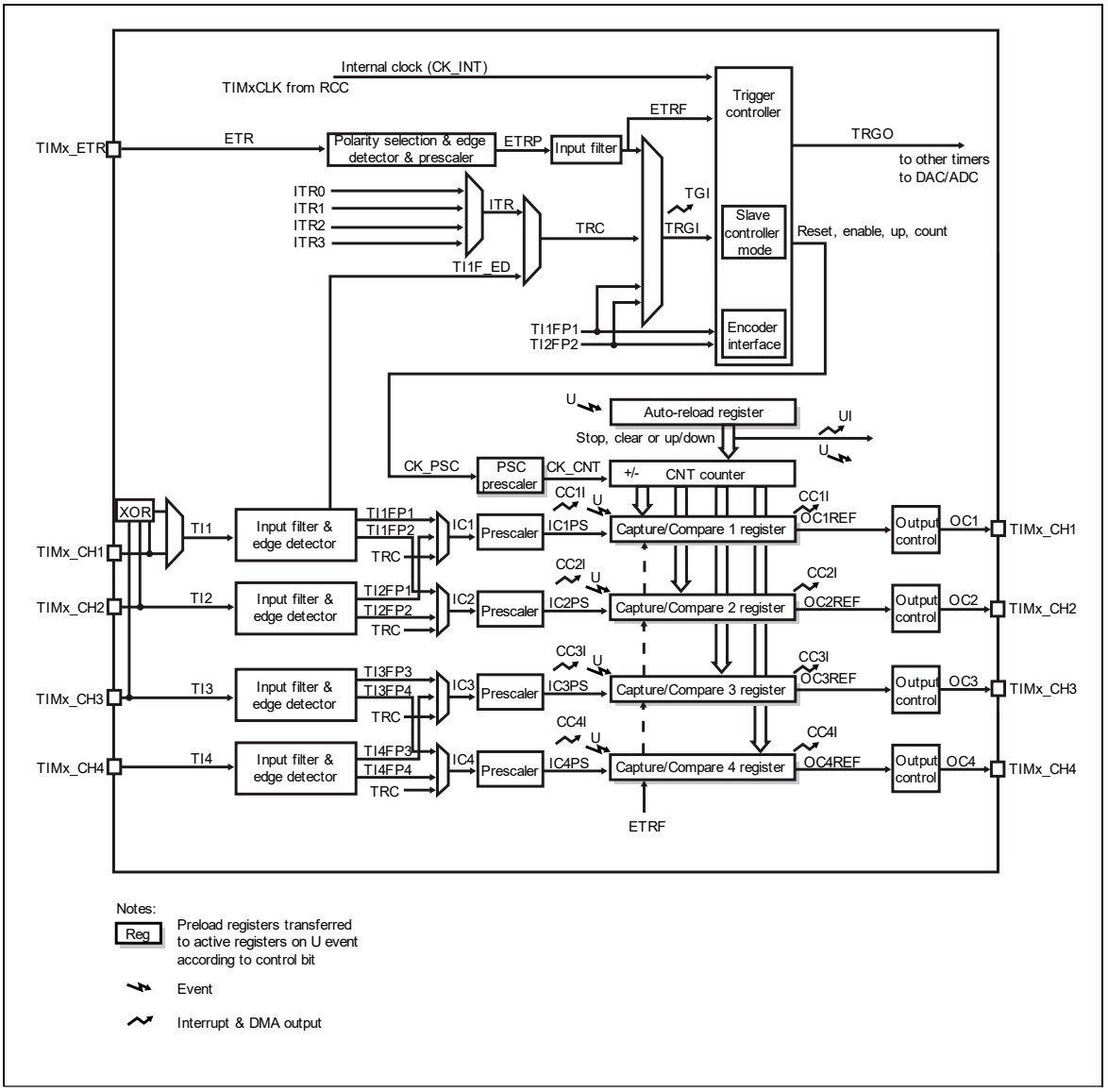



图 15-1 通用定时器框图

## 15.3. TIM2 到 TIM5 功能说明

### 15.3.1. 时基单元

可编程定时器的主要模块由一个 16 位/32 位计数器及其相关的自动重装寄存器组成。此计数器可采用递增方式计数。计数器的时钟可通过预分频器进行分频。

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

时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以直接传送到影子寄存器，也可以在每次发生更新事件 (UEV)时传送到影子寄存器，这取决

于 TIMx_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值（或者在递减计数时达到下溢值）并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位 (CEN)置 1 时，才会启动计数器。

请注意，真正的计数器使能信号 CNT_EN 在 CEN 置 1 的一个时钟周期后被置 1。

**预分频器说明**

预分频器可对计数器时钟频率进行分频，分频系数介于 1 到 65536 之间。该预分频器基于 16 位/32 位寄存器（TIMx_PSC 寄存器）所控制的 16 位计数器。由于该控制寄存器具有缓冲功能，因此预分频器可实现实时更改。而新的预分频比将在下一更新事件发生时被采用。下图给出了在预分频比发生实时变化时一些计数器行为的示例：

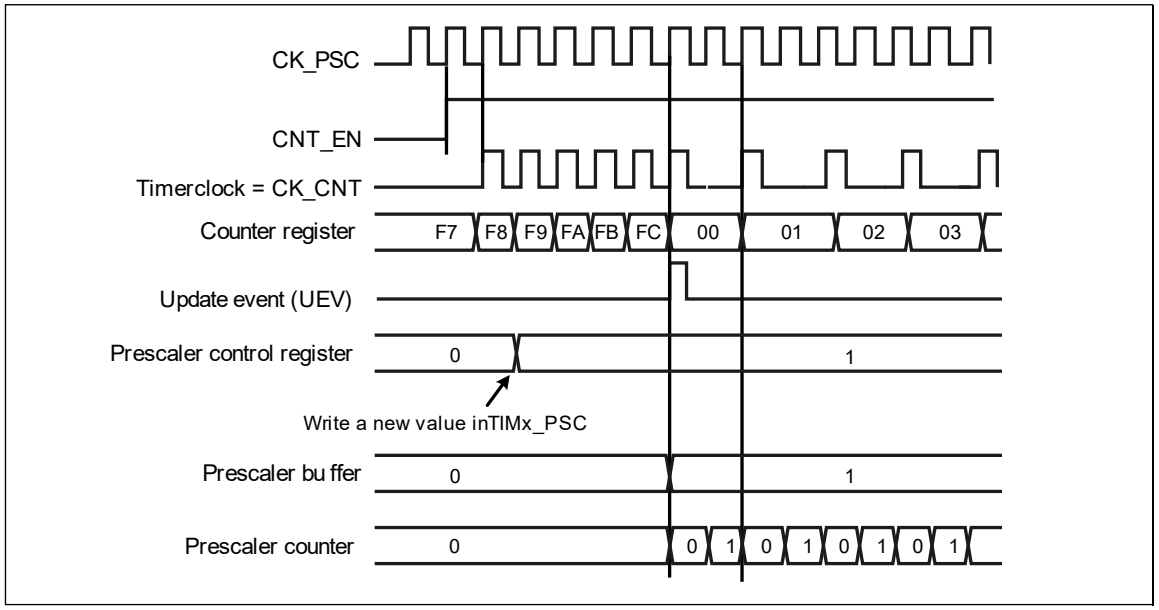



图 15-2 预分频器分频由 1 变为 2 时的计数器时序图

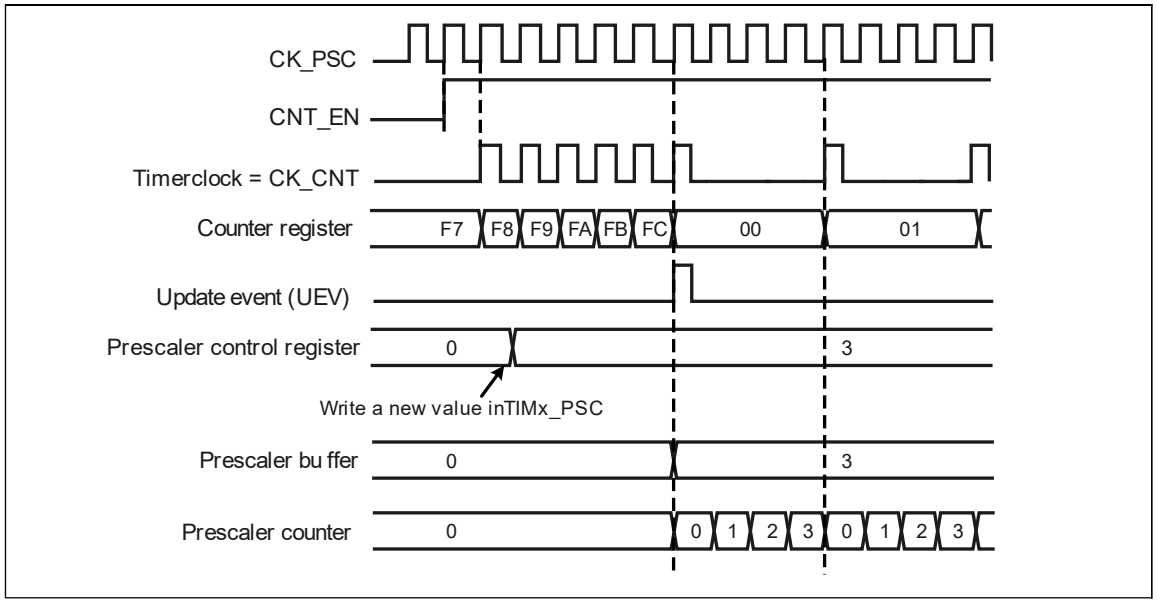



图 15-3 预分频器分频由 1 变为 4 时的计数器时序图

### 15.3.2. 计数器模式

**递增计数模式**

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

每次发生计数器上溢时会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）;
- 自动重载影子寄存器将以预装载值进行更新。

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

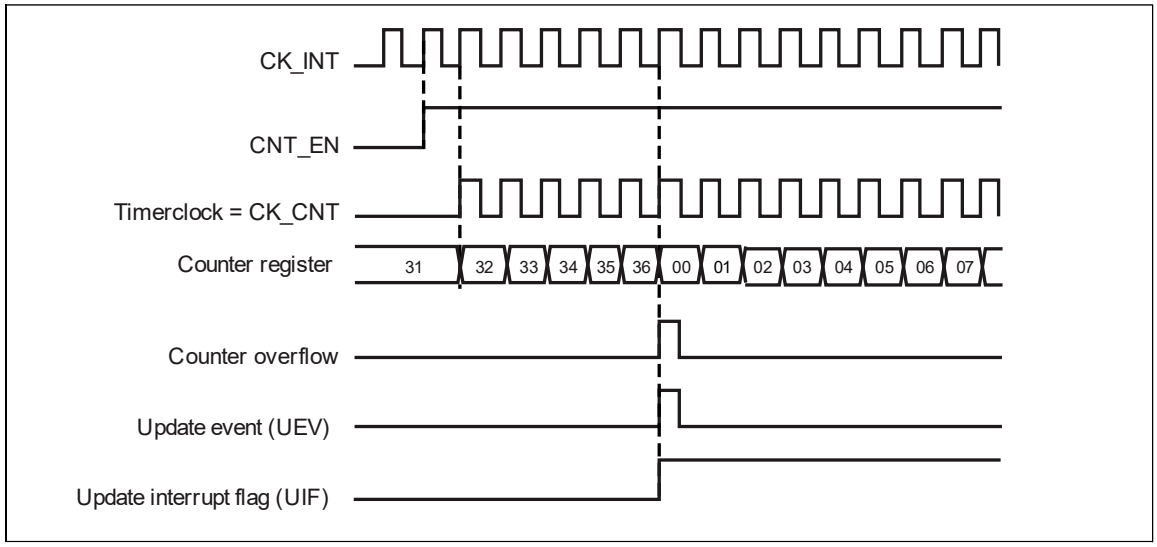



图 15-4 计数器时序图，内部时钟 1 分频

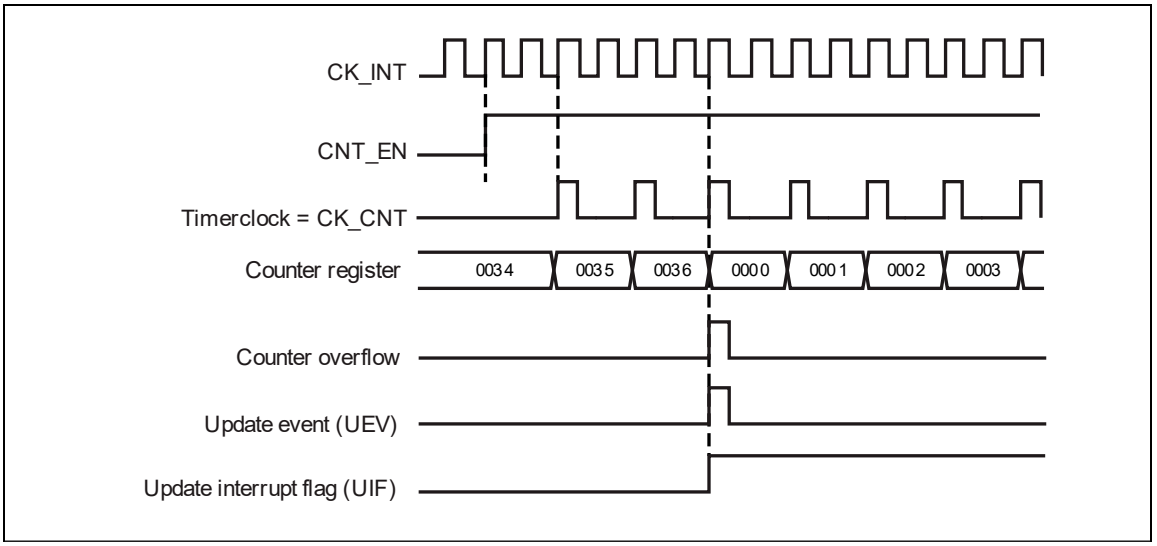



图 15-5 计数器时序图，内部时钟 2 分频

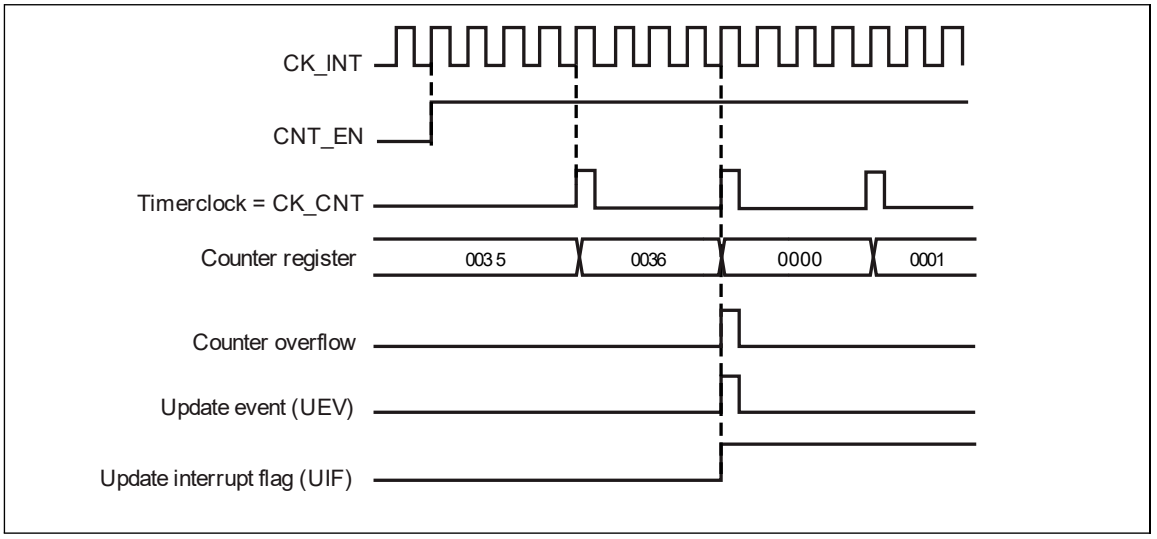



图 15-6 计数器时序图，内部时钟 4 分频

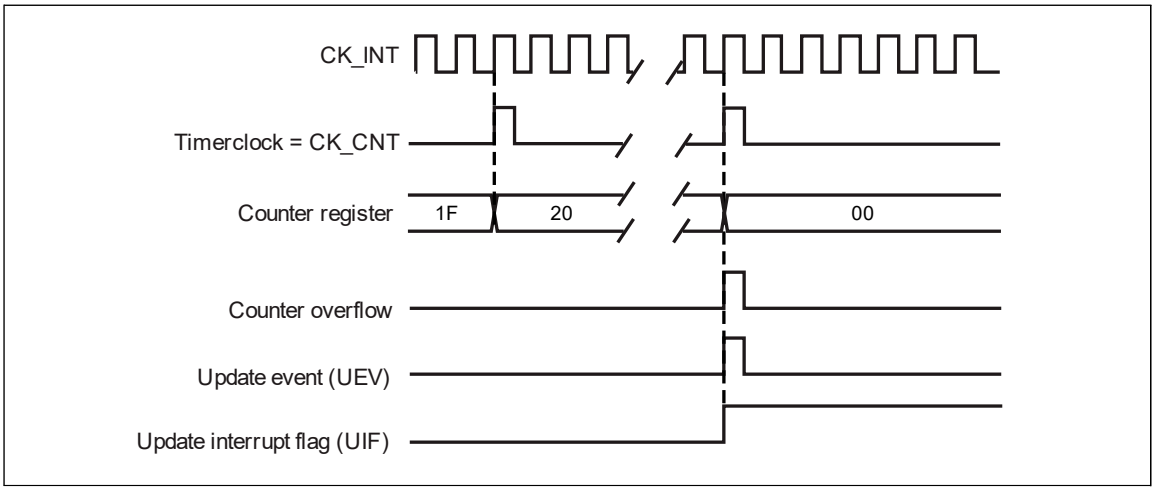



图 15-7 计数器时序图，内部时钟 N 分频

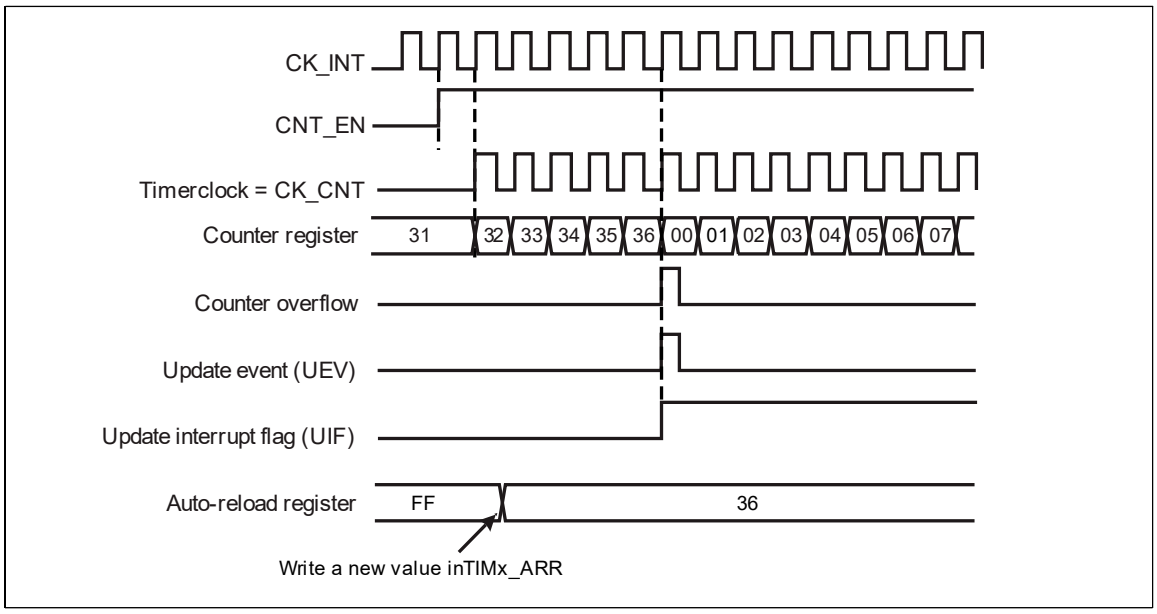



图 15-8 计数器时序图，ARPE=0 时更新事件(TIMx_ARR 未预装载)

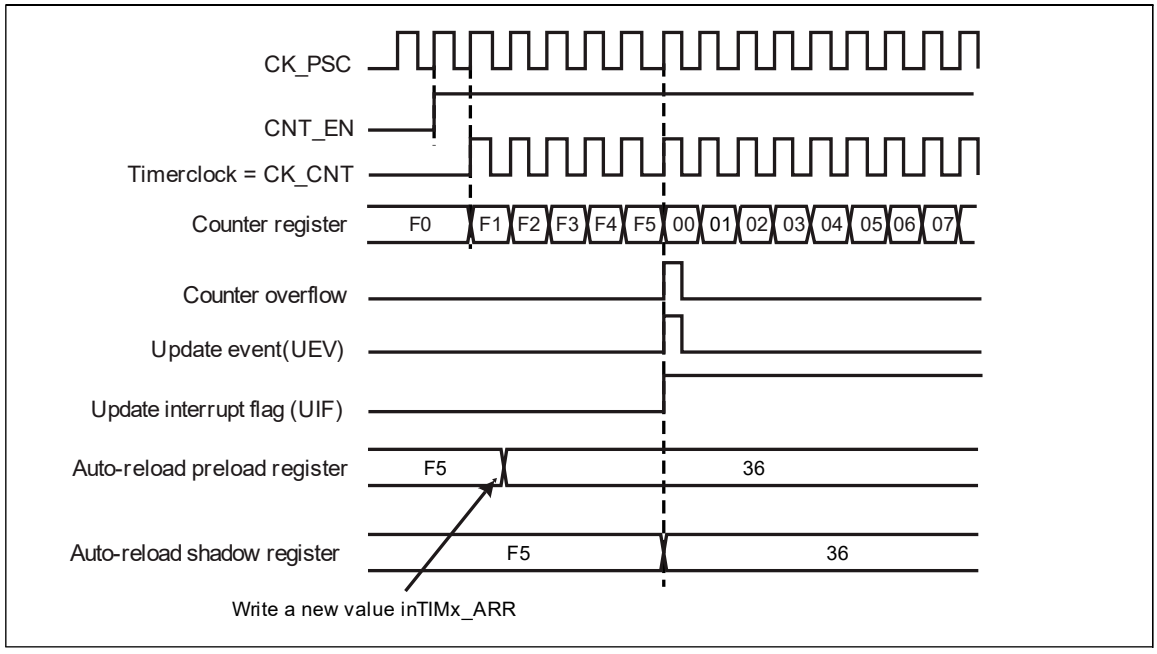



图 15-9 计数器时序图，ARPE=1 时更新事件(TIMx_ARR 已预装载)

**递减计数模式**

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

每次发生计数器下溢时会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器会重新从当前自动重载值开始计数，而预分频器计数器则重新从 0 开始计数（但预分频比保持不变）。

此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件

UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。
- 自动重载活动寄存器将以预装载值（TIMx_ARR 寄存器的内容）进行更新。注意，自动重载寄存器会在计数器重载之前得到更新，因此，下一个计数周期就是我们所希望的新的周期长度。

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

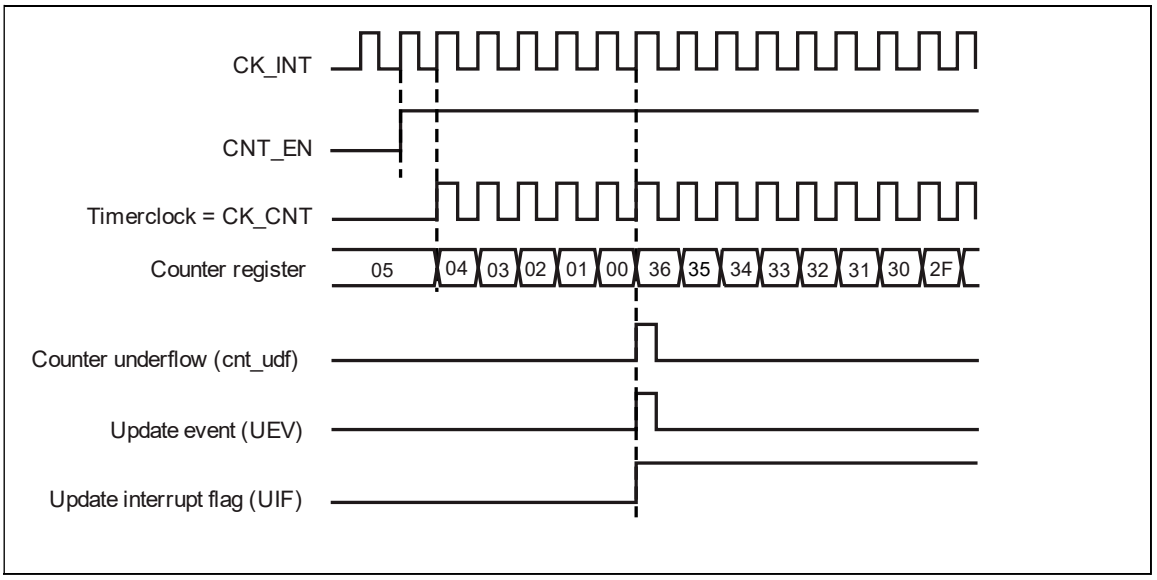



图 15-10 计数器时序图，内部时钟 1 分频

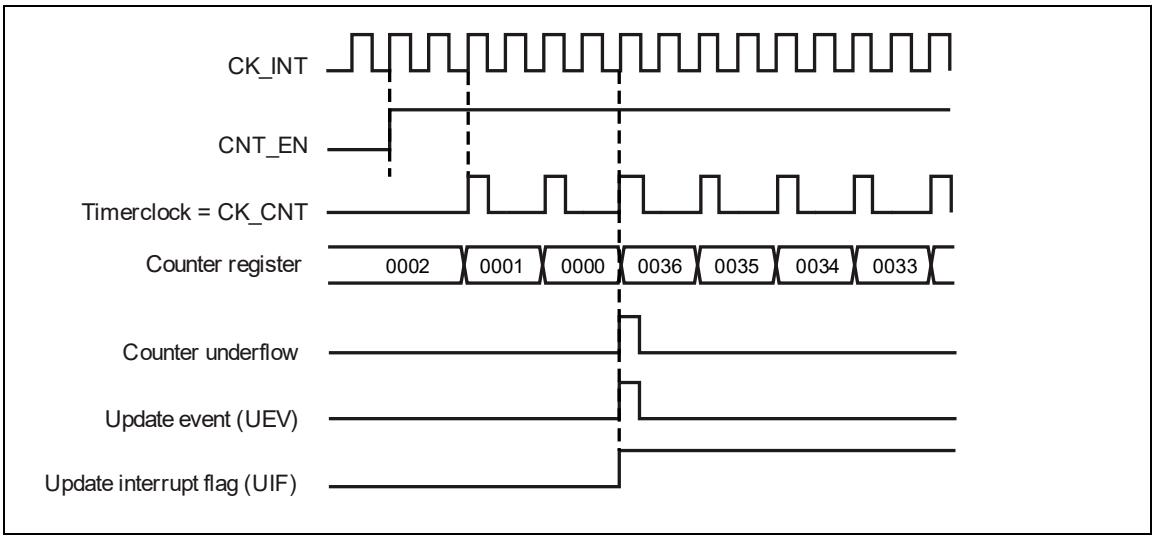



图 15-11 计数器时序图，内部时钟 2 分频

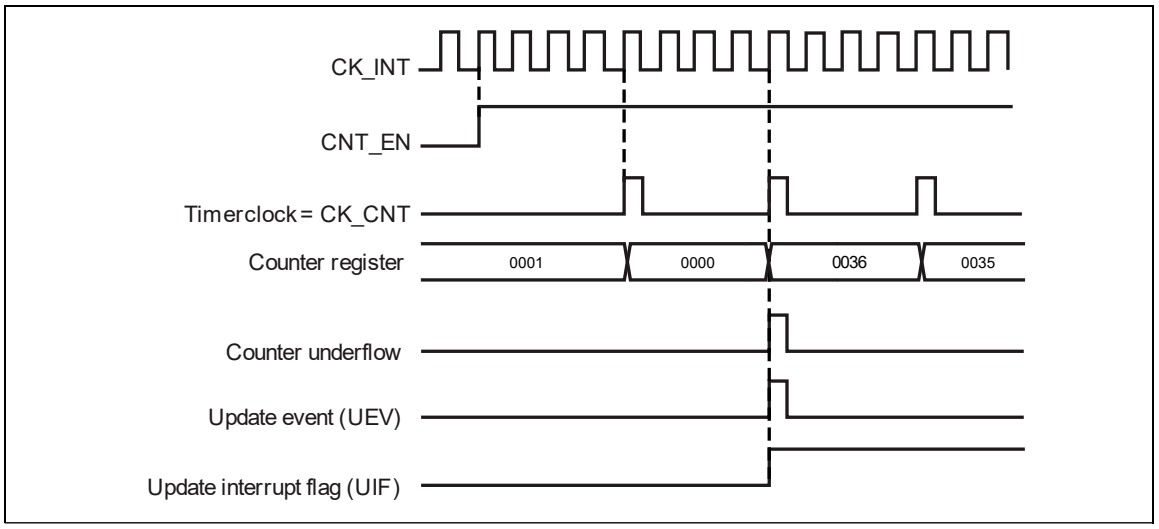



图 15-12 计数器时序图，内部时钟 4 分频

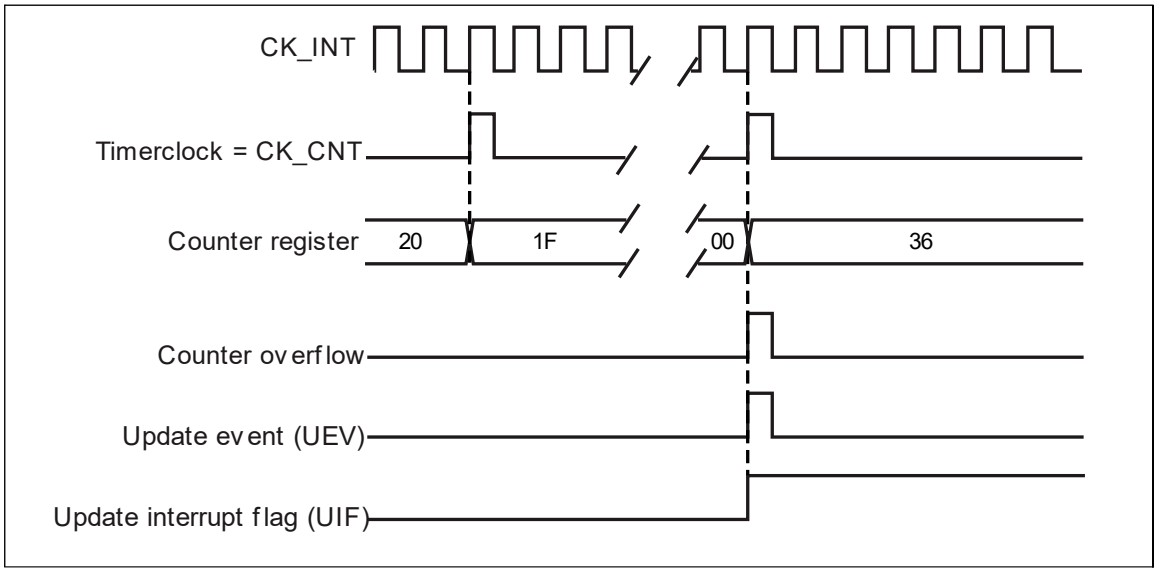



图 15-13 计数器时序图，内部时钟 N 分频

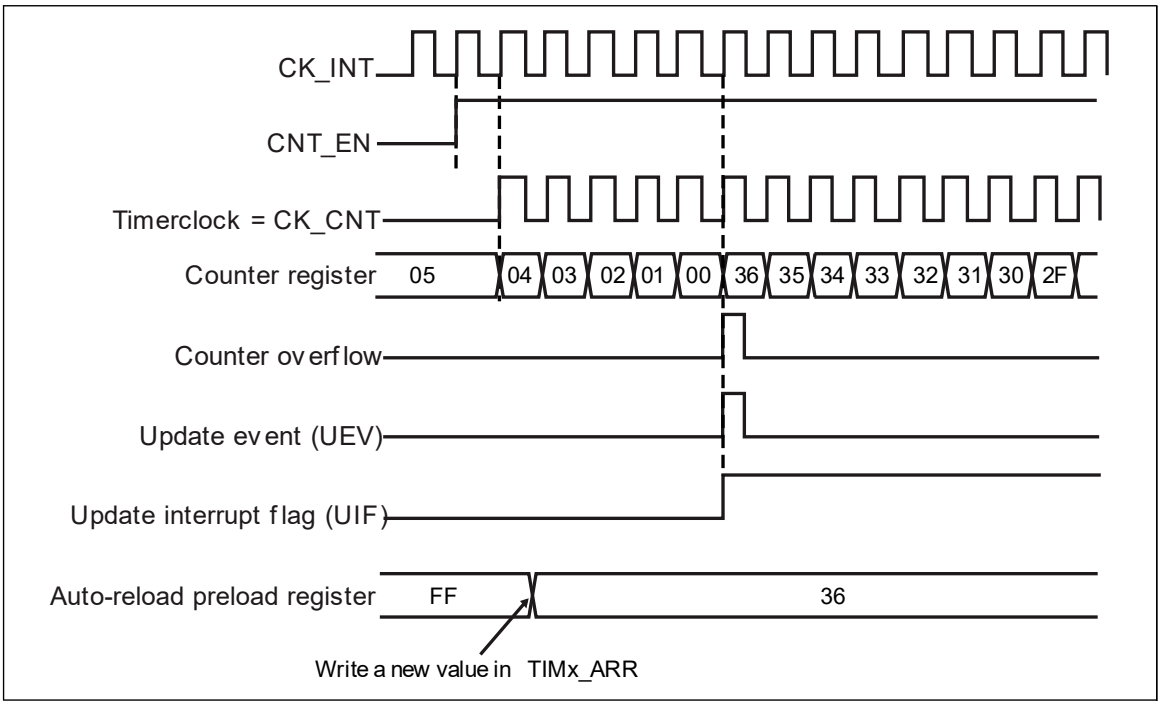



图 15-14 计数器时序图，更新事件

**中心对齐模式（递增/递减计数）**

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

当 TIMx_CR1 寄存器中的 CMS 位不为“00”时，中心对齐模式有效。将通道配置为输出模式时，其输出比较中断标志将在以下模式下置 1，即：计数器递减计数（中心对齐模式 1，CMS =“01”）、计数器递增计数（中心对齐模式 2， CMS =“10”）以及计数器递增/递减计数（中心对齐模式 3， CMS =“11”）。

此模式下无法写入方向位（ TIMx_CR1 寄存器中的 DIR 位）。而是由硬件更新并指示当前计数器方向。

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

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器仍会根据当前自动重载值进行递增和递减计数。

此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

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

以下各图以一些示例说明不同时钟频率下计数器的行为。

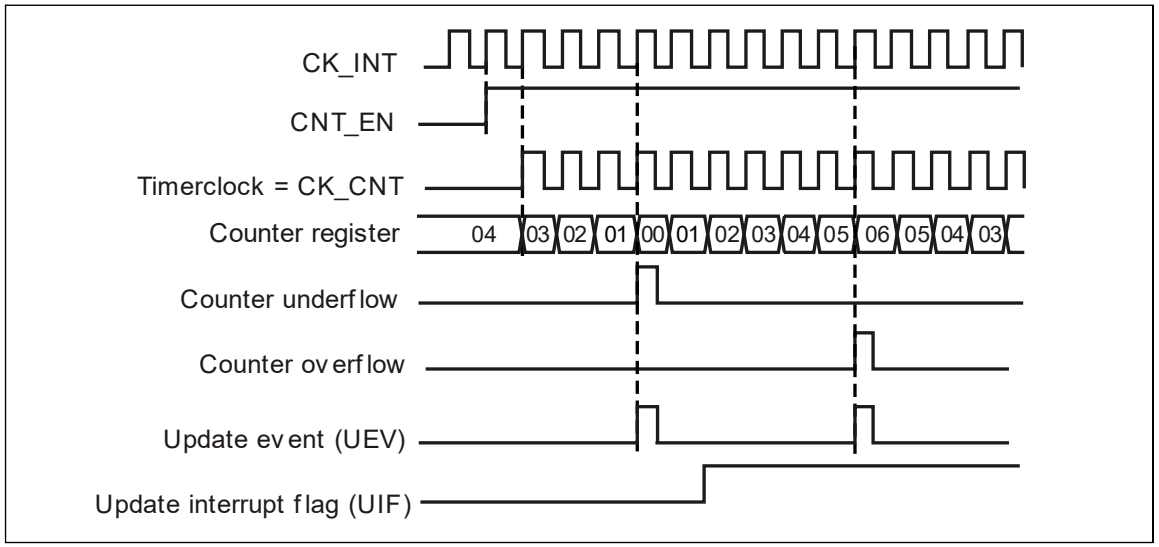



图 15-15 计数器时序图，内部时钟 1 分频，TIMx_ARR=0x6

*注：此处使用中心对齐模式 1*

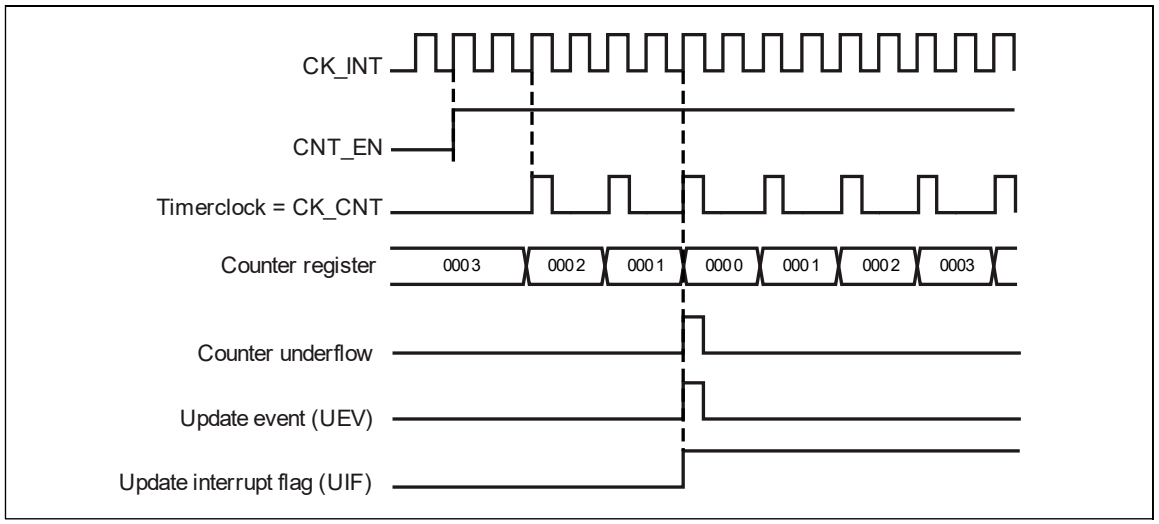



图 15-16 计数器时序图，内部时钟 2 分频

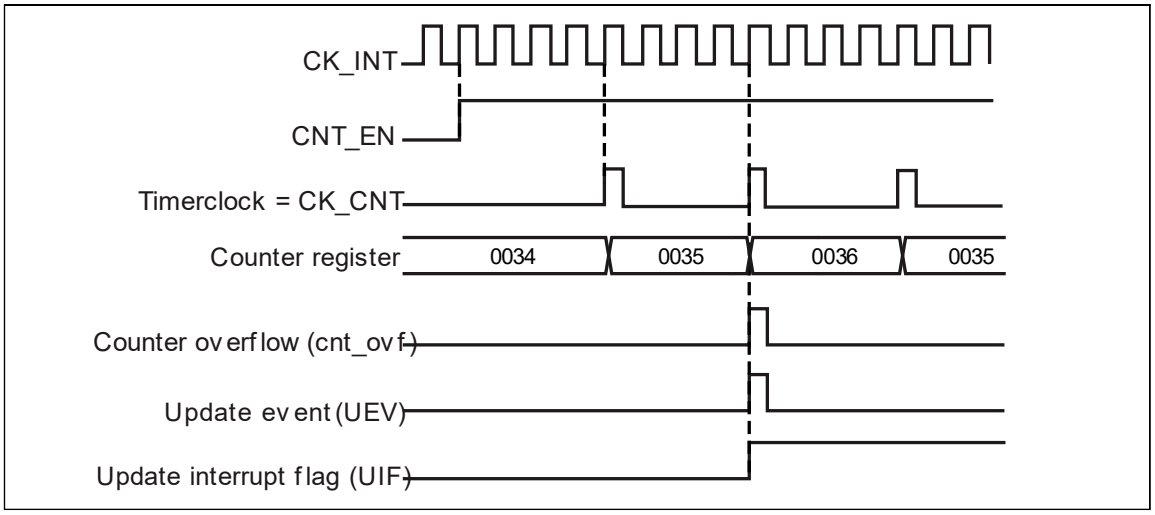



图 15-17 计数器时序图，内部时钟 4 分频，TIMx_ARR=0x36

*注：中心对齐模式 2 或模式 3 与上溢 UIF 结合使用*

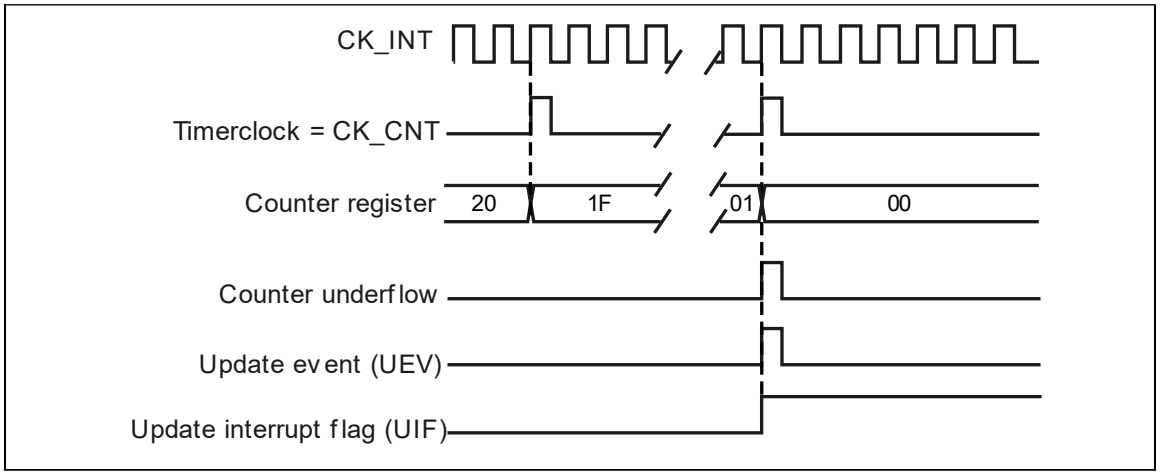



图 15-18 计数器时序图，内部时钟 N 分频

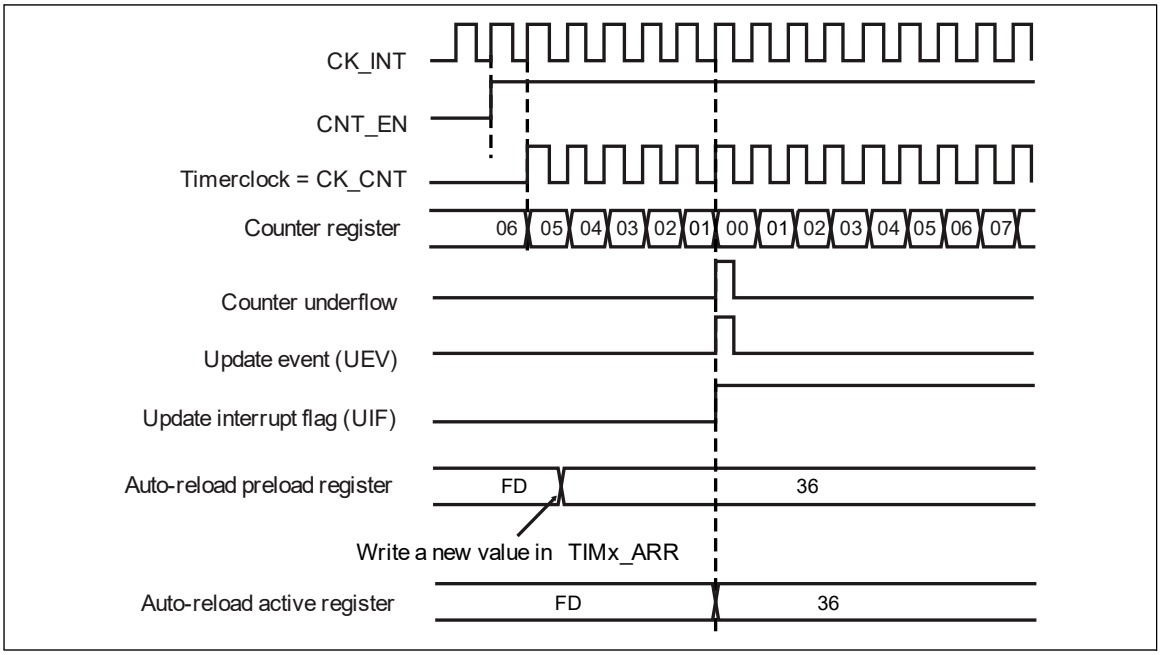



图 15-19 计数器时序图，ARPE=1 时的更新事件(计数器下溢)

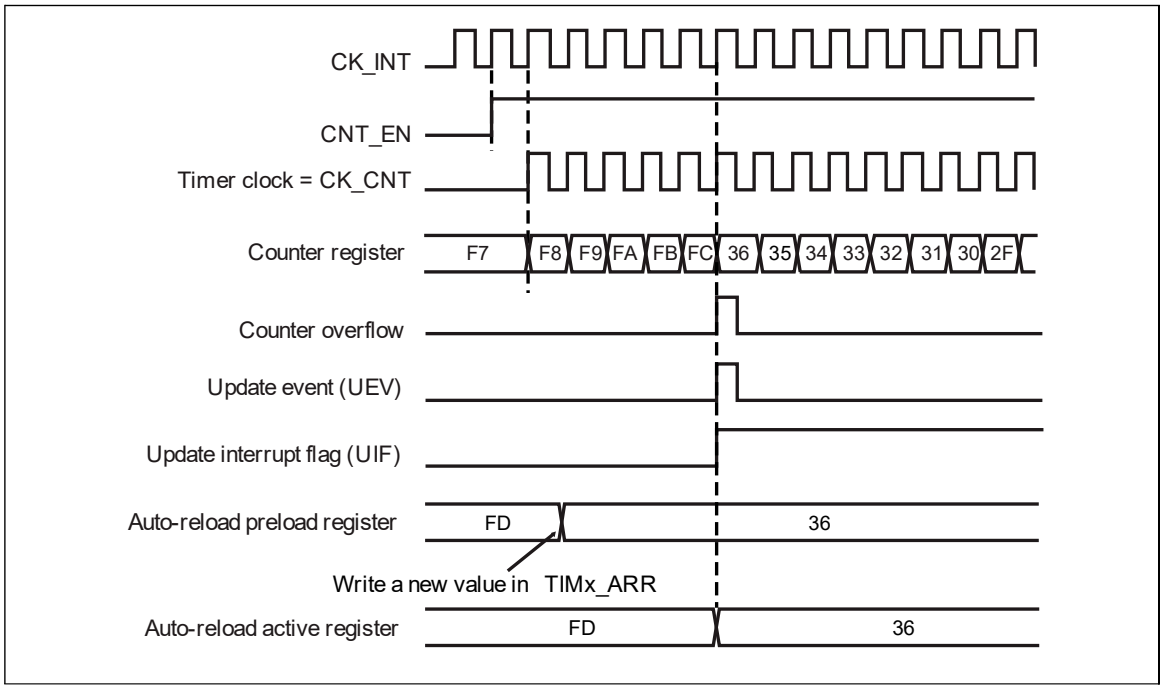



图 15-20 计数器时序图，ARPE=1 时的更新事件(计数器上溢)

### 15.3.3. 时钟选择

计数器时钟可由下列时钟源提供:

- 内部时钟 (CK_INT)。
- 外部时钟模式 1：外部输入引脚 (TIx)。

- 外部时钟模式 2：外部触发输入 (ETR)，仅适用于 TIM2、 TIM3 和 TIM4。
- 内部触发输入 (ITRx)：使用一个定时器作为另一个定时器的预分频器，例如可以将定时器配置为定时器 2 的预分频器。

### 内部时钟源 (CK_INT)

如果禁止从模式控制器（TIMx_SMCR 寄存器中 SMS=000），则 CEN 位、DIR 位（TIMx_CR1 寄存器中）和 UG 位（TIMx_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（UG 除外，仍自动清零）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 CK_INT 提供。

下图显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

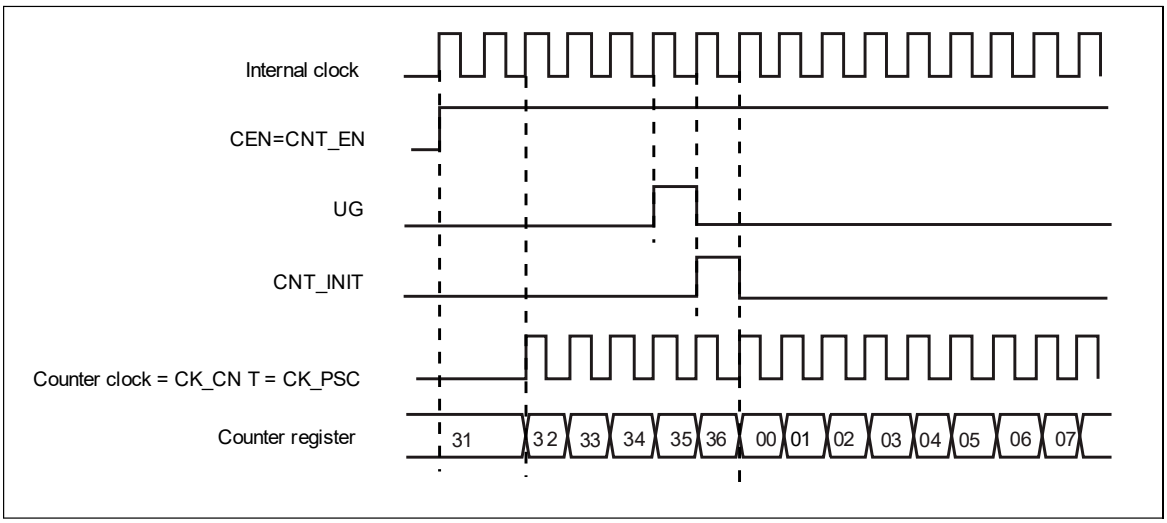



图 15-21 正常模式下的控制电路，内部时钟 1 分频

### 外部时钟源模式 1

当 TIMx_SMCR 寄存器中的 SMS=111 时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

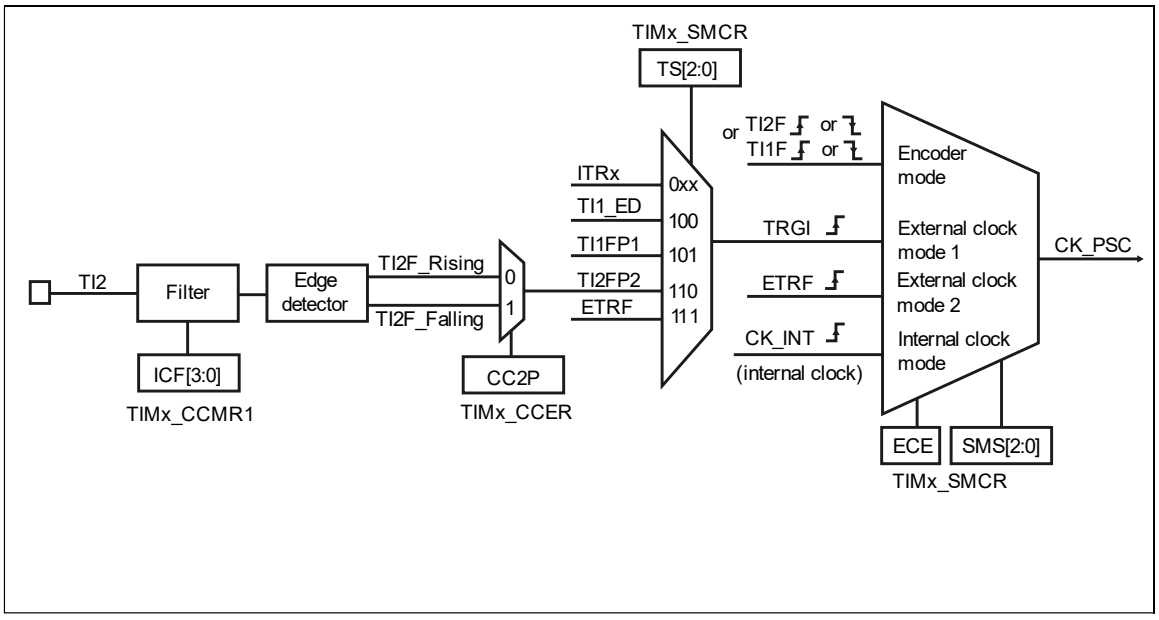



图 15-22 TI2 外部时钟连接示例

例如，要使递增计数器在 TI2 输入出现上升沿时计数，请执行以下步骤：

- 通过在 TIMx_CCMR1 寄存器中写入 CC2S="01"来配置通道 2，使其能够检测 TI2 输入的上升沿。
- 通过在 TIMx_CCMR1 寄存器中写入 IC2F[3:0] 位来配置输入滤波时间（如果不需要任何滤波，请保持 IC2F=0000）。

*注意： 由于捕获预分频器不用于触发操作，因此无需对其进行配置。*

- 通过在 TIMx_CCER 寄存器中写入 CC2P=0 和 CC2NP=0 来选择上升沿极性。
- 通过在 TIMx_SMCR 寄存器中写入 SMS=111，使定时器在外部时钟模式 1 下工作。
- 通过在 TIMx_SMCR 寄存器中写入 TS=110 来选择 TI2 作为输入源。
- 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。

TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入的重新同步电路引起的。

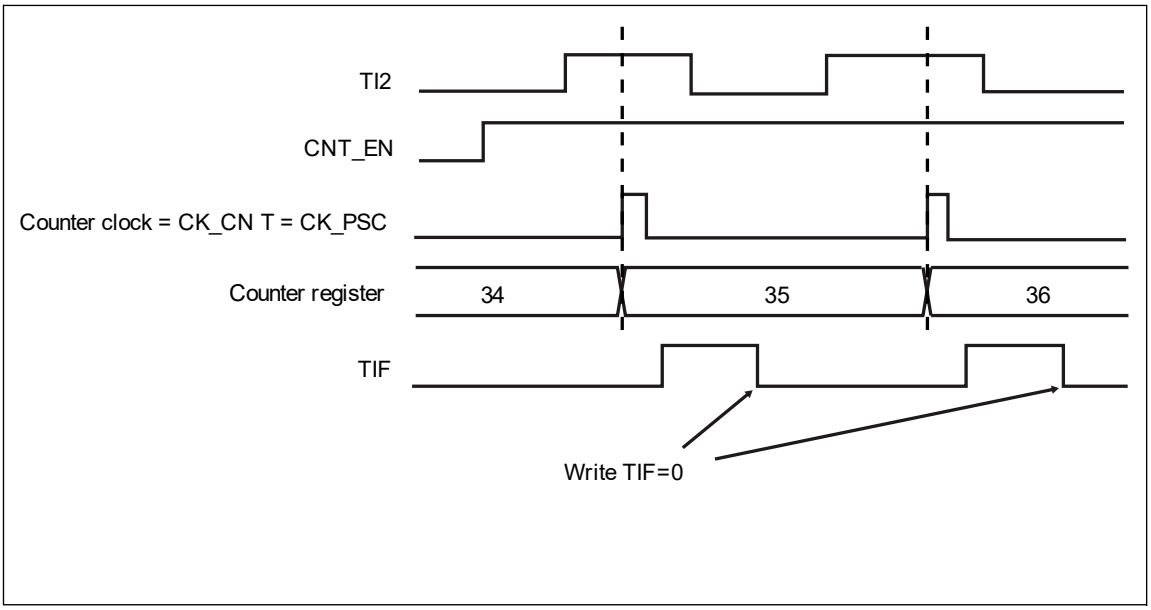



图 15-23 外部时钟模式 1 下的控制电路

**外部时钟源模式 2**

通过在 TIMx_SMCR 寄存器中写入 ECE=1 可选择此模式。

计数器可在外部触发输入 ETR 出现上升沿或下降沿时计数。

下图简要介绍了外部触发输入模块。

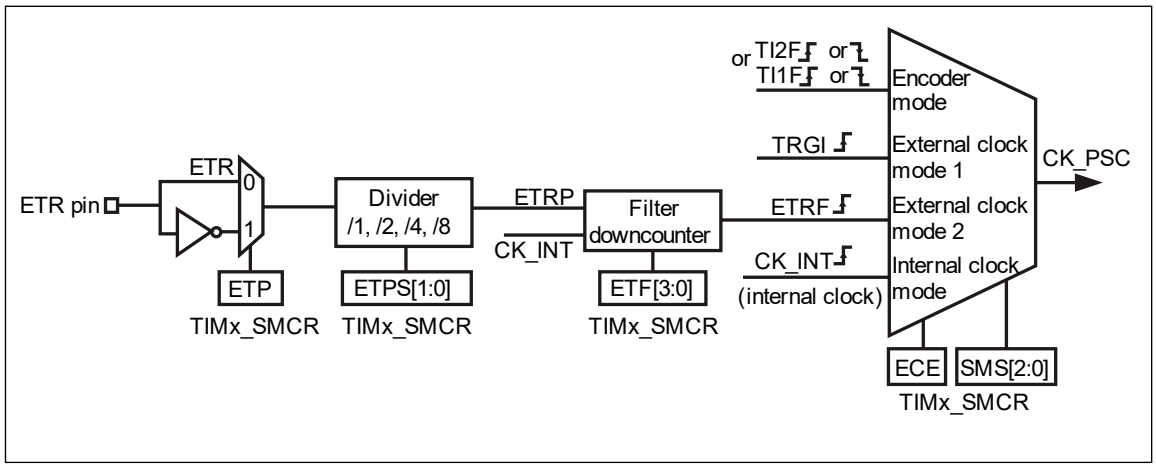



图 15-24 外部触发输入模块

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

- 由于此例中不需滤波器，因此在 TIMx_SMCR 寄存器中写入 ETF[3:0]=0000。
- 通过在 TIMx_SMCR 寄存器中写入 ETPS[1:0]=01 来设置预分频器。
- 通过在 TIMx_SMCR 寄存器中写入 ETP=0 来选择 ETR 引脚的上升沿检测。
- 通过在 TIMx_SMCR 寄存器中写入 ECE=1 来使能外部时钟模式 2。
- 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

ETR 每出现 2 个上升沿，计数器计数一次。

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

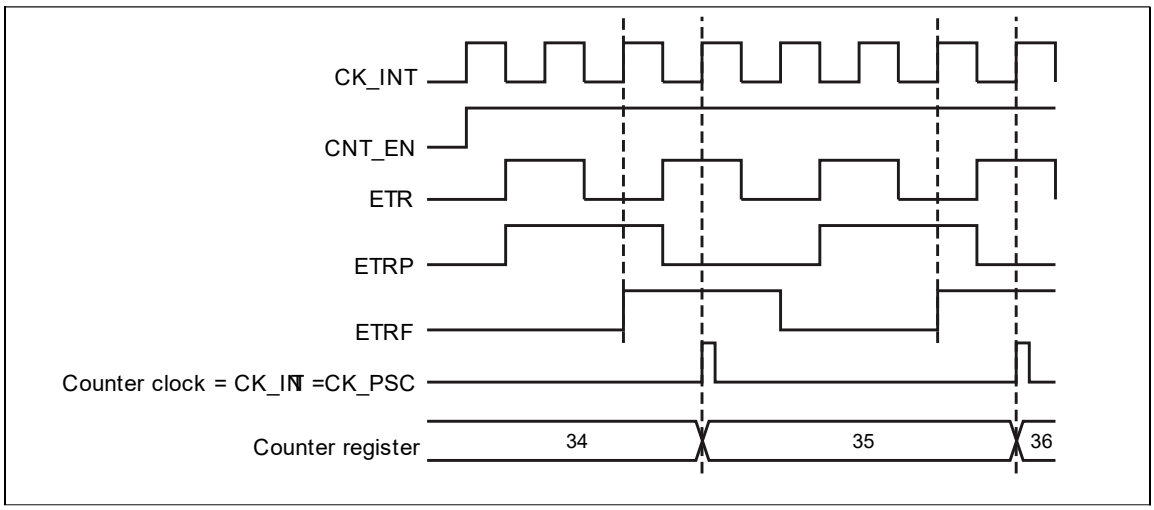



图 15-25 外部时钟模式 2 下的控制电路

### 15.3.4. 捕获/比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入阶段（数字滤波、多路复用和预分频器）和一个输出阶段（比较器和输出控制）构建而成。

下图概括介绍了一个捕获/比较通道。

输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 TIxF。然后，带有极性选择功能的边沿检测器生成一个信号 (TIxFPx)，该信号可用作从模式控制器的触发输入，也可用作捕获命令。该信号先进行预分频 (ICxPS)，而后再进入捕获寄存器。

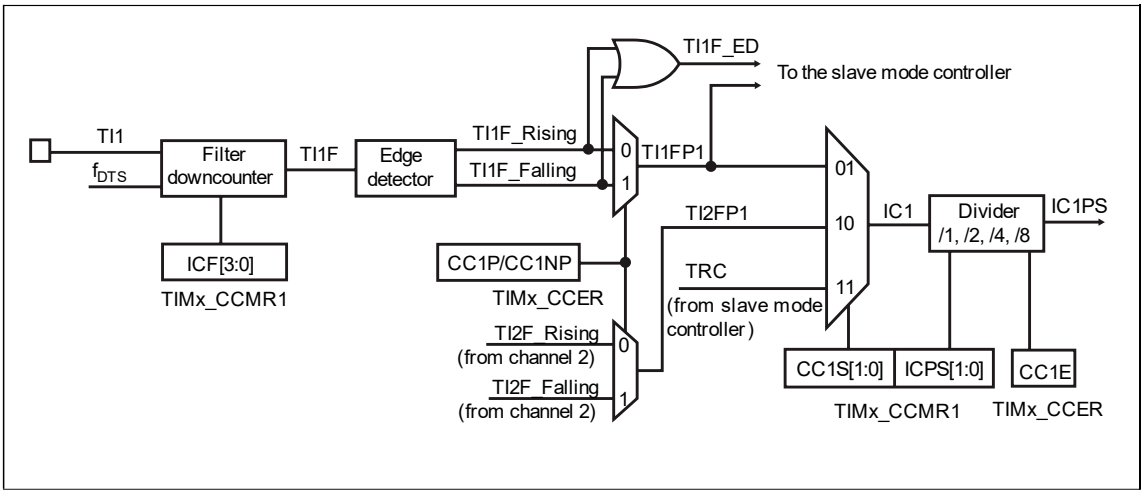



图 15-26 捕获/比较通道(例如：通道 1 输入阶段)

输出阶段生成一个中间波形作为基准：OCxRef（ 高电平有效)。链的末端决定最终输出信号的极性。

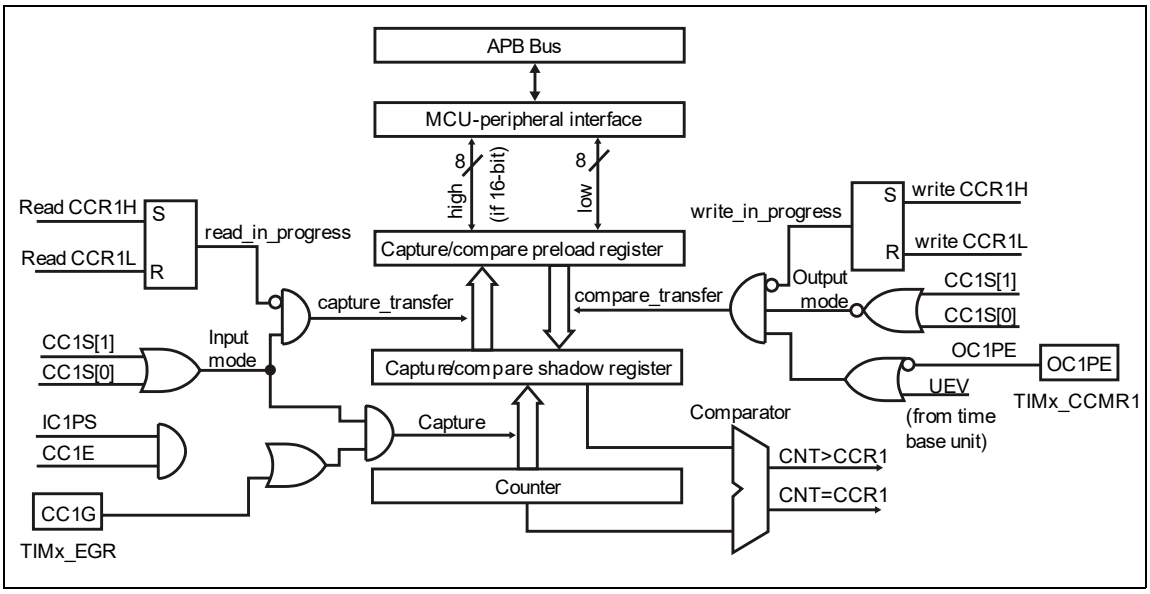



图 15-27 捕获/比较通道 1 主电路

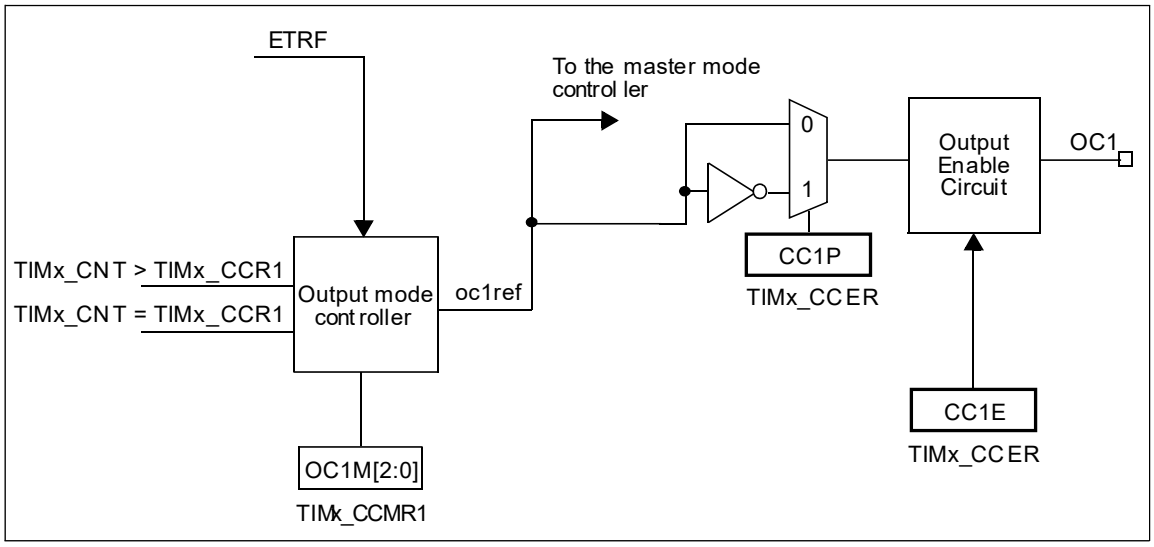



图 15-28 捕获/比较通道的输出阶段(通道 1)

捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。始终可通过读写操作访问预装载寄存器。

在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容复制到预装载寄存器中。

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

### 15.3.5. 输入捕获模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获/比较寄存器(TIMx_CCRx)来锁存计数器的值。发生捕获事件时，会将相应的 CCXIF 标志（TIMx_SR 寄存器）置 1，并可发送中断或 DMA 请求（如果已使能）。如果发生捕获事件时 CCxIF 标志已处于高位，则会将重复捕获标志 CCxOF（ TIMx_SR 寄存器）置 1。可通过软件向 CCxIF 写入 0 来给 CCxIF 清零，或读取存储在 TIMx_CCRx 寄存器中的已捕获数据。向 CCxOF 写入 0 后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIMx_CCR1 中。具体操作步骤如下：

- 选择有效输入：TIMx_CCR1 必须连接到 TI1 输入，因此向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 TIMx_CCR1 寄存器将处于只读状态。
- 根据连接到定时器的信号，对所需的输入滤波时间进行编程（如果输入为 TIx 输入之一，则对 TIMx_CCMRx 寄存器中的 ICxF 位进行编程）。假设信号变化时，输入信号最多在 5 个内部时钟周期内发生抖动。因此，我们必须将滤波时间设置为大于 5 个内部时钟周期。在检测到 8 个具有新电平的连续采样（以 $f_{DTS}$ 频率采样）后，可以确认 TI1 上的跳变沿。然后向 TIMx_CCMR1 寄存器中的 IC1F 位写入 0011。
- 通过向 TIMx_CCER 寄存器中的 CC1P 位和 CC1NP 位写入 0，选择 TI1 通道的有效转换边沿（本例中为上升沿）。
- 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 TIMx_CCMR1 寄存器中的 IC1PS 位写入 00）。
- 通过将 TIMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
- 如果需要，可通过将 TIMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求，并且/或者通过将该寄存器中的 CC1DE 位置 1 来使能 DMA 请求。

发生输入捕获时：

- 发生有效跳变沿时， TIMx_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样

CC1OF 捕获溢出标志会被置 1。

- 根据 CC1IE 位生成中断。
- 根据 CC1DE 位生成 DMA 请求。

要处理重复捕获，建议在读出捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出标志之后与读取数据之前可能出现的重复捕获信息。

*注意：通过软件将 TIMx_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断和/或 DMA 请求。*

### 15.3.6. PWM 输入模式

此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这两个 ICx 信号在边沿处有效，但极性相反。
- 选择两个 TIxFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。

例如，可通过以下步骤对应用于 TI1 的 PWM 的周期（位于 TIMx_CCR1 寄存器中）和占空比（位于 TIMx_CCR2 寄存器中）进行测量（取决于 CK_INT 频率和预分频器的值）：

- 选择 TIMx_CCR1 的有效输入：向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01（选择 TI1）。
- 选择 TI1FP1 的有效极性（用于 TIMx_CCR1 中的捕获和计数器清零）：向 CC1P 位和 CC1NP 位写入“0”（上升沿有效）。
- 选择 TIMx_CCR2 的有效输入：向 TIMx_CMR1 寄存器中的 CC2S 写入 10（选择 TI1）。
- 选择 TI1FP2 的有效极性（用于 TIMx_CCR2 中的捕获）：向 CC2P 位和 CC2NP 位写入“1”（下降沿有效）。
- 选择有效触发输入：向 TIMx_SMCR 寄存器中的 TS 位写入 101（选择 TI1FP1）。
- 将从模式控制器配置为复位模式：向 TIMx_SMCR 寄存器中的 SMS 位写入 100。
- 使能捕获：向 TIMx_CCER 寄存器中的 CC1E 位和 CC2E 位写入“1”。

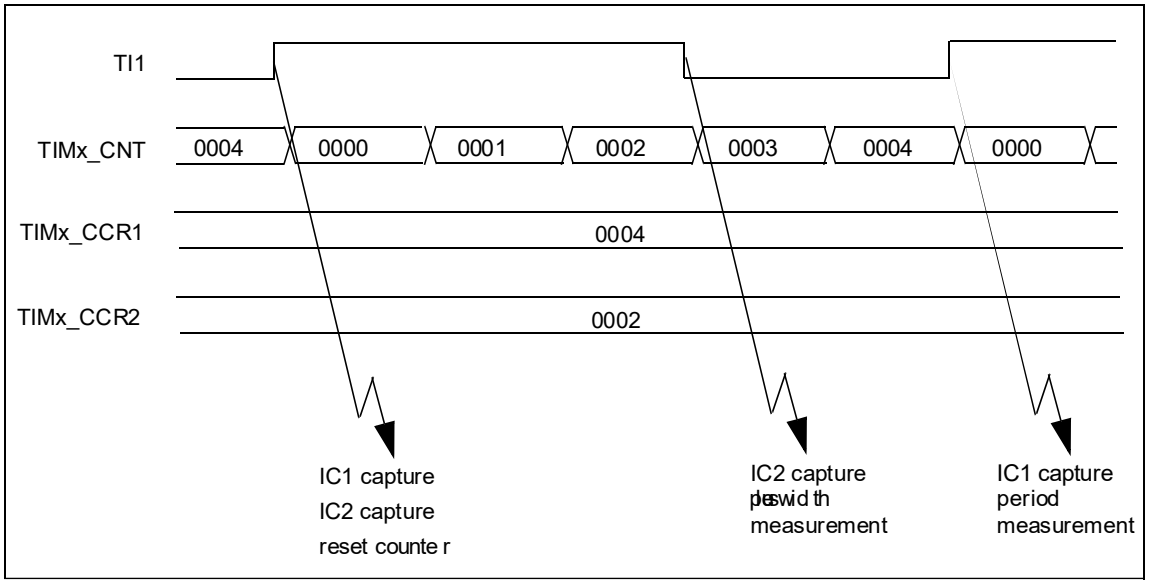



图 15-29 PWM 输入模式时序

### 15.3.7. 强制输出模式

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

要将输出比较信号 (OCXREF/OCx) 强制设置为有效电平，只需向相应 TIMx_CCMRx 寄存器中的 OCxM 位写入 101。ocxref 进而强制设置为高电平（OCxREF 始终为高电平有效），同时 OCx 获取 CCxP 极性位的相反值。

例如： CCxP=0（OCx 高电平有效）=> OCx 强制设置为高电平。

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入 100，可将 ocxref 信号强制设置为低电平

无论如何，TIMx_CCRx 影子寄存器与计数器之间的比较仍会执行，而且允许将标志置 1。因此可发送相应的中断和 DMA 请求。输出比较模式一节对此进行了介绍。

### 15.3.8. 输出比较模式

此功能用于控制输出波形，或指示已经过某一时间段。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

- 将为相应的输出引脚分配一个可编程值，该值由输出比较模式（TIMx_CCMRx 寄存器中的 OCxM 位）和输出极性（TIMx_CCER 寄存器中的 CCxP 位）定义。匹配时，输出引脚既可保持其电平 (OCXM=000)，也可设置为有效电平 (OCXM=001)、无效电平(OCXM=010) 或进行翻转 (OCxM=011)。
- 将中断状态寄存器中的标志置 1（TIMx_SR 寄存器中的 CCxIF 位）。
- 如果相应中断使能位（TIMx_DIER 寄存器中的 CCXIE 位）置 1，将生成中断。
- 如果相应 DMA 使能位（TIMx_DIER 寄存器的 CCxDE 位，TIMx_CR2 寄存器的 CCDS 位，用来选择 DMA 请求）置 1，将发送 DMA 请求。

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

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

步骤：

- 选择计数器时钟（内部、外部、预分频器）。
- 在 TIMx_ARR 和 TIMx_CCRx 寄存器中写入所需数据。
- 如果要生成中断和/或 DMA 请求，将 CCxIE 位和/或 CCxDE 位置 1。
- 选择输出模式。例如，当 CNT 与 CCRx 匹配、未使用预装载 CCRx 并且 OCx 使能且为高电平有效时，必须写入 OCxM=011、OCxPE=0、CCxP=0 和 CCxE=1 来翻转 OCx 输出引脚。
- 通过将 TIMx_CR1 寄存器中的 CEN 位置 1 来使能计数器。

可随时通过软件更新 TIMx_CCRx 寄存器以控制输出波形，前提是未使能预装载寄存器（OCxPE=0，否则仅当发生下一个更新事件 UEV 时，才会更新 TIMx_CCRx 影子寄存器）。

下图列出了相关示例。

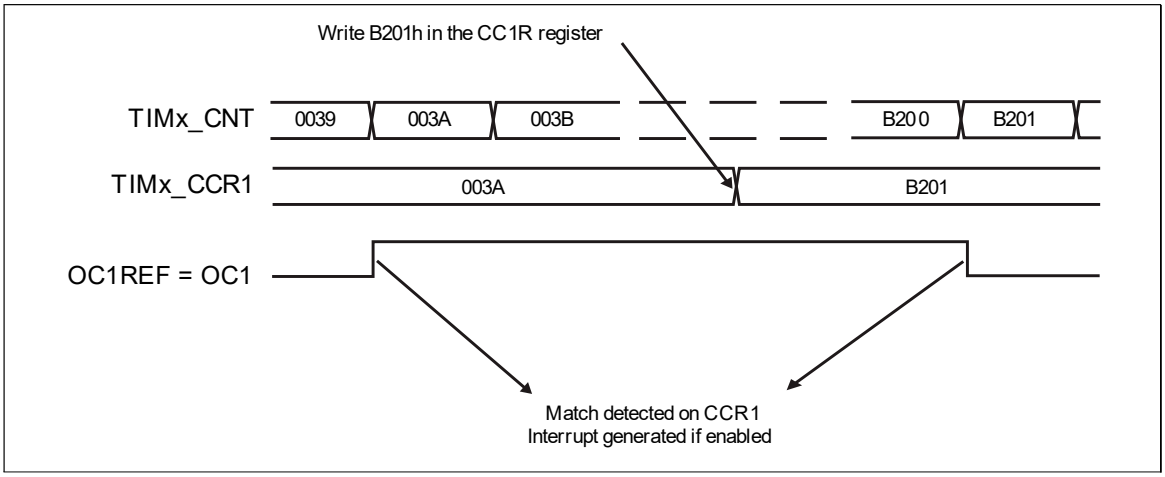



图 15-30 输出比较模式，翻转 OC1

### 15.3.9. PWM 模式

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

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入 110（PWM 模式 1）或 111（PWM 模式 2），可以独立选择各通道（每个 OCx 输出对应一个 PWM）的 PWM 模式。必须通过将 TIMx_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器, 最后通过将 TIMx_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器。

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

OCx 极性可使用 TIMx_CCER 寄存器的 CCxP 位来编程。既可以设为高电平有效，也可以设为低电平有效。OCx 输出通过将 TIMx_CCER 寄存器中的 CCxE 位置 1 来使能。有关详细信息，请参见 TIMx_CCERx 寄存器说明。

在 PWM 模式(1 或 2)下, TIMx_CNT 始终与 TIMx_CCRx 进行比较, 以确定是 TIMx_CCRx≤TIMx_CNT 还是 TIMx_CNT≤TIMx_CCRx（取决于计数器计数方向)。不过，为了与 ETRF 相符（在下一个 PWM 周期之前，ETR 信号上的一个外部事件能够清除 OCxREF），OCREF 信号仅在以下情况下变为有效状态:

- 比较结果发生改变，或
- 输出比较模式（TIMx_CCMRx 寄存器中的 OCxM 位）从“冻结”配置（不进行比较，OCxM=“000”）切换为任一 PWM 模式（OCxM=“110”或“111”）。

定时器运行期间，可以通过软件强制 PWM 输出。

根据 TIMx_CR1 寄存器中的 CMS 位状态，定时器能够产生边沿对齐模式或中心对齐模式的 PWM 信号。

#### PWM 边沿对齐模式

- 递增计数配置

  当 TIMx_CR1 寄存器中的 DIR 位为低时执行递增计数。

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

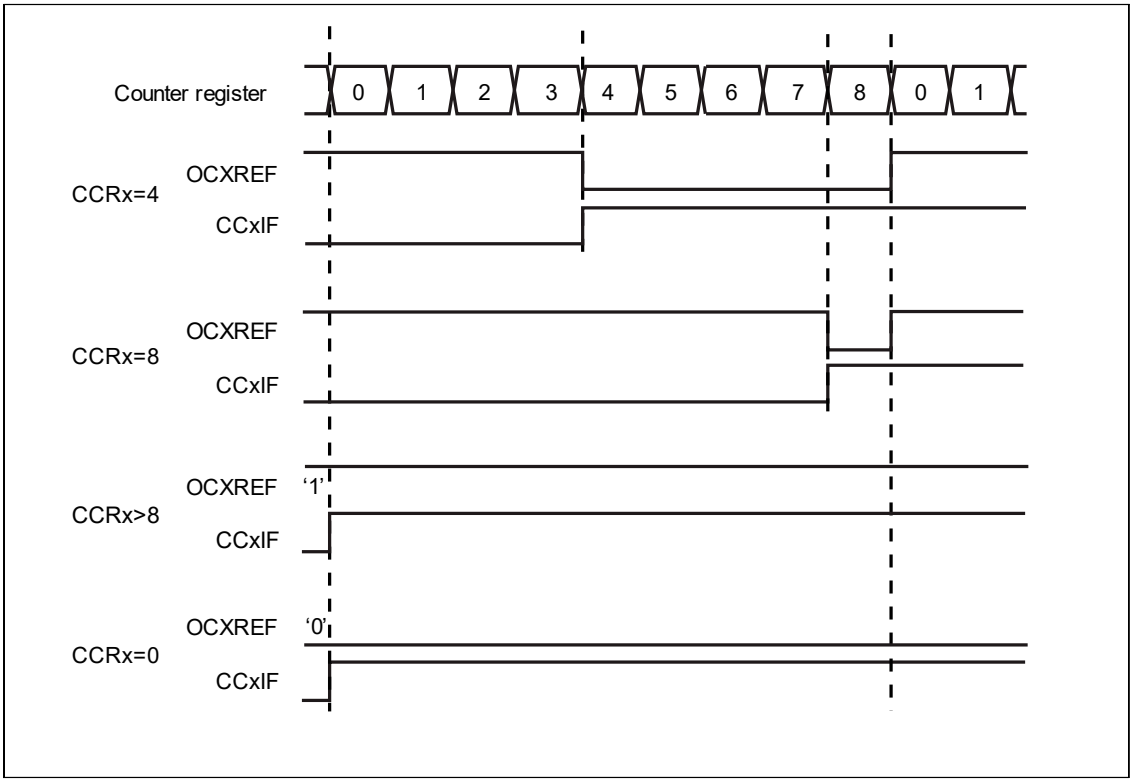



图 15-31 边沿对齐模式的 PWM 波形(ARR=8)

- 递增计数配置

  当 TIMx_CR1 寄存器中的 DIR 位为高时执行递减计数。

  在 PWM 模式 1 下，只要 TIMx_CNT > TIMx_CCRx，参考信号 ocxref 便为低电平，否则为高电平。如果 TIMx_CCRx 中的比较值大于 TIMx_ARR 中的自动重载值，则 ocxref 保持为“1”。此模式下不可能产生 0% 的 PWM 波形。

**PWM 中心对齐模式**

当 TIMx_CR1 寄存器中的 CMS 位不为“00”时（其余所有配置对 ocxref/OCx 信号具有相同的作用），中心对齐模式生效。根据 CMS 位的配置，可以在计数器递增计数、递减计数或同时递增和递减计数时将比较标志置 1。TIMx_CR1 寄存器中的方向位 (DIR) 由硬件更新，不得通过软件更改。下图显示了中心对齐模式的 PWM 波形，在此例中：

- TIMx_ARR=8，
- PWM 模式为 PWM 模式 1，
- 在根据 TIMx_CR1 寄存器中 CMS=01 而选择的中心对齐模式 1 下，当计数器递减计数时，比较标志置 1。

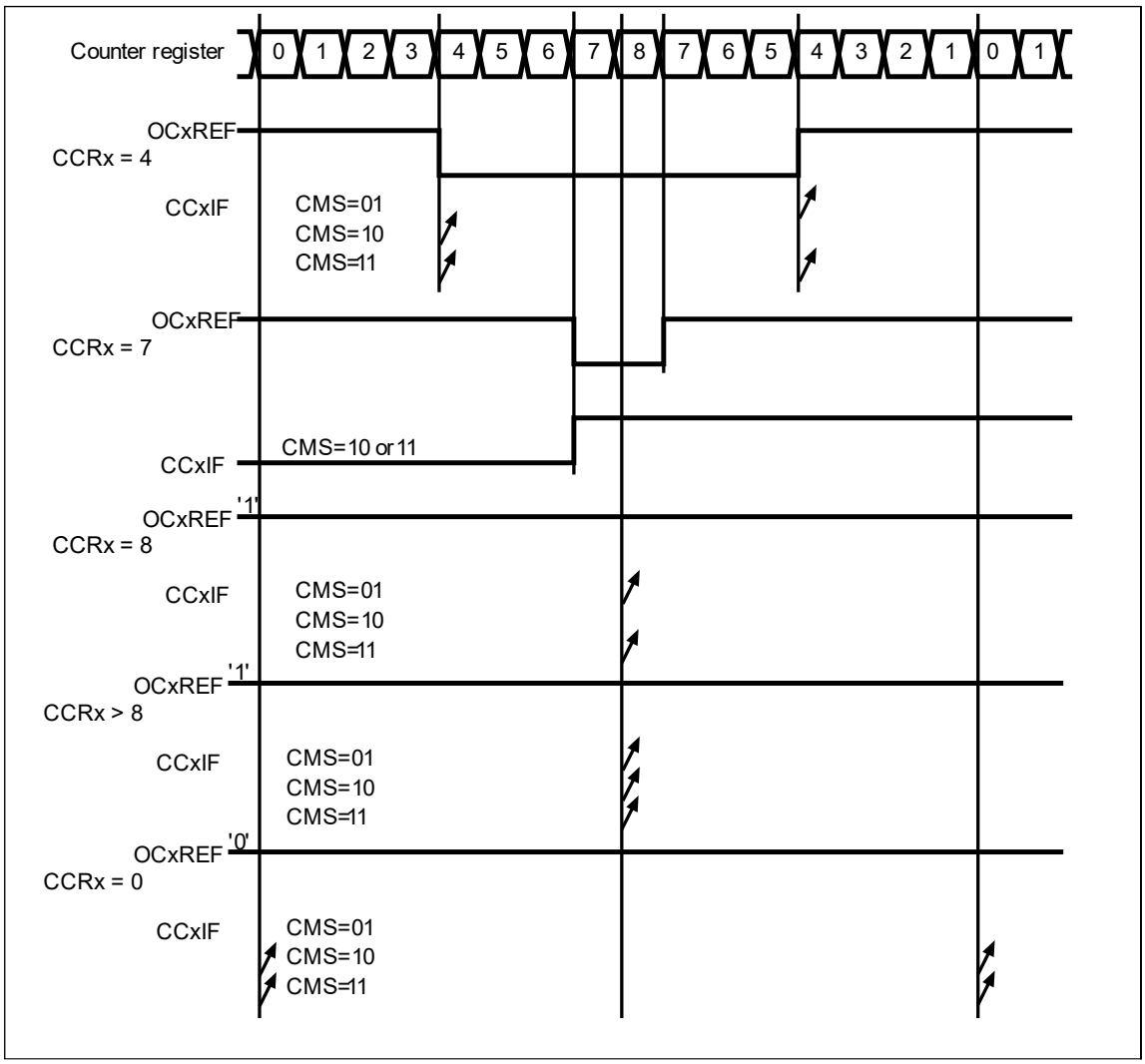



图 15-32 中心对齐模式 PWM 波形(ARR=8)

中心对齐模式使用建议：

- 启动中心对齐模式时将使用当前的递增/递减计数配置。这意味着计数器将根据写入 TIMx_CR1 寄存器中 DIR 位的值进行递增或递减计数。此外，不得同时通过软件修改 DIR 和 CMS 位。
- 不建议在运行中心对齐模式时对计数器执行写操作，否则将发生意想不到的结果。尤其是：
- 如果写入计数器中的值大于自动重载值 (TIMx_CNT>TIMx_ARR)，计数方向不会更新。例如，如果计数器之前递增计数，则继续递增计数。
- 如果向计数器写入 0 或 TIMx_ARR 的值，计数方向会更新，但不生成更新事件 UEV。
- 使用中心对齐模式最为保险的方法是：在启动计数器前通过软件生成更新（将 TIMx_EGR 寄存器中的 UG 位置 1），并且不要在计数器运行过程中对其执行写操作。

### 15.3.10. 单脉冲模式

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

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。将 TIMx_CR1 寄存器中的 OPM 位置 1，即可选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

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

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

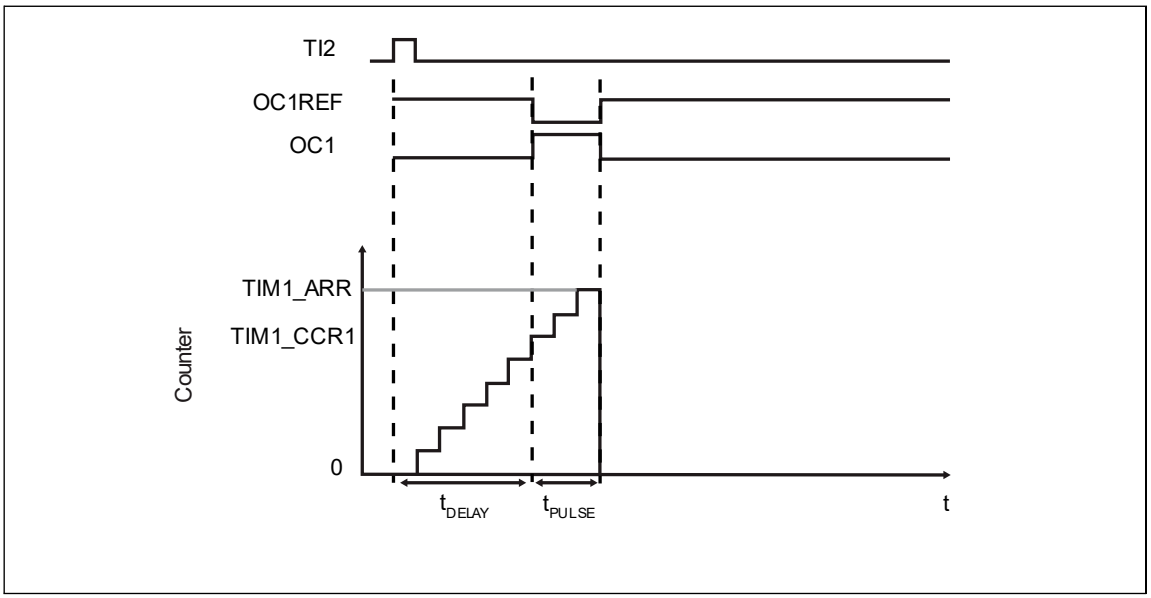



图 15-33 单脉冲模式示例

例如，用户希望达到这样的效果：在 TI2 输入引脚检测到正沿时，经过 $t_{DELAY}$ 的延迟，在 OC1 上产生一个长度为 $t_{PULSE}$ 的正脉冲。

使用 TI2FP2 作为触发 1：

- 在 TIMx_CCMR1 寄存器中写入 CC2S=01，将 TI2FP2 映射到 TI2。
- 在 TIMx_CCER 寄存器中写入 CC2P=0 和 CC2NP=“0”，使 TI2FP2 能够检测上升沿。
- 在 TIMx_SMCR 寄存器中写入 TS=110，将 TI2FP2 配置为从模式控制器的触发 (TRGI)。
- 在 TIMx_SMCR 寄存器中写入 SMS=“110”（触发模式），使用 TI2FP2 启动计数器。

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

- $t_{DELAY}$ 由写入 TIMx_CCR1 寄存器的值定义。
- $t_{PULSE}$ 由自动重载值与比较值 (TIMx_ARR - TIMx_CCR1) 之差来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从“0”变为“1”，在计数器达到自动重载值时由“1”变为“0”。为此，应在 TIMx_CCMR1 寄存器中写入 OC1M=111，以使能 PWM 模式 2。如果需要，可选择在 TIMx_CCMR1 寄存器的 OC1PE 和 TIMx_CR1 寄存器的 ARPE 中写入“1”，以使能预装载寄存器。这种情况下，必须在 TIMx_CCR1 寄存器中写入比较值并在 TIMx_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI2 上的外部触发事件。本例中，CC1P 的值为“0”。

在本例中，TIMx_CR1 寄存器中的 DIR 和 CMS 位应为低。

由于仅需要 1 个脉冲（单脉冲模式），因此应向 TIMx_CR1 寄存器的 OPM 位写入“1”，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。TIMx_CR1 寄存器中的 OPM 位置“0”时，即选择重复模式。

**特例： OCx 快速使能：**

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

小值)。

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

### 15.3.11. 发生外部事件时清除 OCxREF 信号

对于给定通道，在 ETRF 输入施加高电平（相应 TIMx_CCMRx 寄存器中的 OCxCE 使能位置“1”），可使 OCxREF 信号变为低电平。OCxREF 信号将保持低电平，直到发生下一更新事件 (UEV)。

此功能仅能用于输出比较模式和 PWM 模式，而不适用于强制输出模式。

例如， ETR 信号可以连接到比较器的输出，用于控制电流。此时，ETR 必须如下配置：

必须关闭外部触发预分频器：TIMx_SMCR 寄存器中的 ETPS[1:0] 位置 “00”。

必须禁止外部时钟模式 2：TIMx_SMCR 寄存器中的 ECE 位置 “0”。

外部触发极性 (ETP) 和外部触发滤波器 (ETF) 可根据应用需要进行配置。

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

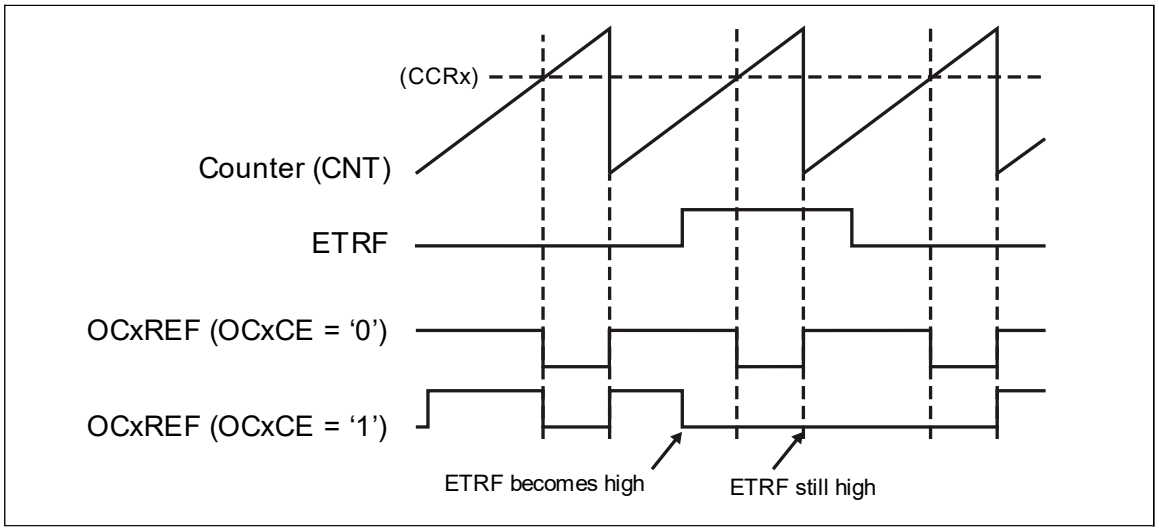



图 15-34 清除 TIMx 的 OCxREF

### 15.3.12. 编码器接口模式

选择编码器接口模式时，如果计数器仅在 TI2 边沿处计数，在 TIMx_SMCR 寄存器中写入 SMS=001；如果计数器仅在 TI1 边沿处计数，写入 SMS=010；如果计数器在 TI1 和 TI2 边沿处均计数，则写入 SMS=011。

通过编程 TIMx_CCER 寄存器的 CC1P 和 CC2P 位, 选择 TI1 和 TI2 极性。如果需要, 还可对输入滤波器进行编程。

TI1 和 TI2 两个输入用于连接增量编码器。请参见下表。如果使能计数器（在 TIMx_CR1 寄存器的 CEN 位中写入“1”), 则计数器的时钟由 TI1FP1 或 TI2FP2 上的每次有效信号转换提供。TI1FP1 和 TI2FP2 是进行输入滤波器和极性选择后 TI1 和 TI2 的信号, 如果不进行滤波和反相, 则 TI1FP1=TI1，TI2FP2=TI2。将根据两个输入的信号转换序列，产生计数脉冲和方向信号。根据该信号转换序列，计数器相应递增或递减计数，同时硬件对 TIMx_CR1 寄存器的 DIR 位进行相应修改。任何输入（TI1 或 TI2）发生信号转换时，都会计算 DIR 位，无论计数器是仅在 TI1 或 TI2 边沿处计数，还是同时在 TI1 和 TI2 处计数。

编码器接口模式就相当于带有方向选择的外部时钟。这意味着，计数器仅在 0 到 TIMx_ARR 寄存器中的自动重载值之间进行连续计数（根据具体方向，从 0 递增计数到 ARR，或从 ARR 递减计数到 0）。因此，在启动前必须先配置 TIMx_ARR。同样，捕获、比较、预分频器、触发输出功能继续正常工作。

在此模式下，计数器会根据增量编码器的速度和方向自动进行修改，因此，其内容始终表示编码器的位置。计数方向对应于所连传感器的旋转方向。下表汇总了可能的组合（假设 TI1 和 TI2 不同时切换）。

表 15-1 计数方向与编码器信号的关系

| Active edge | Level on opposite signal (TI1FP1 for TI2, TI2FP2 for TI1) | TI1FP1 signal | | TI2FP2 signal | |
|---|---|---|---|---|---|
| | | Rising | Falling | Rising | Falling |
| Counting on TI1 only | High | Down | Up | No Count | No Count |
| | Low | Up | Down | No Count | No Count |
| Counting on TI2 only | High | No Count | No Count | Up | Down |
| | Low | No Count | No Count | Down | Up |
| Counting on TI1 and TI2 | High | Down | Up | Up | Down |
| | Low | Up | Down | Down | Up |

外部增量编码器可直接与 MCU 相连，无需外部接口逻辑。不过，通常使用比较器将编码器的差分输出转换为数字信号。这样大幅提高了抗噪声性能。用于指示机械零位的第三个编码器输出可与外部中断输入相连，用以触发计数器复位。

下图以计数器工作为例，说明了计数信号的生成和方向控制。同时也说明了选择双边沿时如何对输入抖动进行补偿。将传感器靠近其中一个切换点放置时可能出现这种情况。本例中假设配置如下：

- CC1S =“01”（TIMx_CCMR1 寄存器， TI1FP1 映射到 TI1 上）。
- CC2S =“01”（TIMx_CCMR2 寄存器， TI1FP2 映射到 TI2 上）。
- CC1P =“0”, CC1NP=“0”, 且 IC1F =“0000”（TIMx_CCER 寄存器，TI1FP1 未反相， TI1FP1=TI1）。
- CC2P =“0”， CC2NP=“0”，且 IC2F =“0000”（TIMx_CCER 寄存器，TI1FP2 未反相，TI1FP2= TI2）。
- SMS =“011”（TIMx_SMCR 寄存器，两个输入在上升沿和下降沿均有效）。
- CEN =“1”（TIMx_CR1 寄存器，使能计数器）。

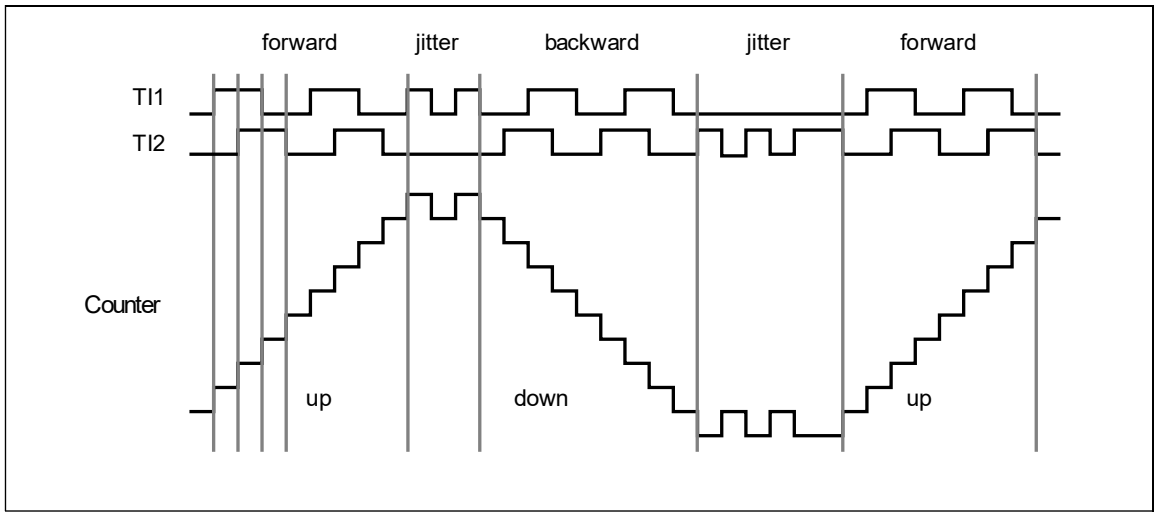



图 15-35 编码器接口模式下的计数器工作示例

下图举例说明 TI1FP1 极性反相时计数器的行为（除 CC1P=“ 1”外，其它配置与上例相同）。

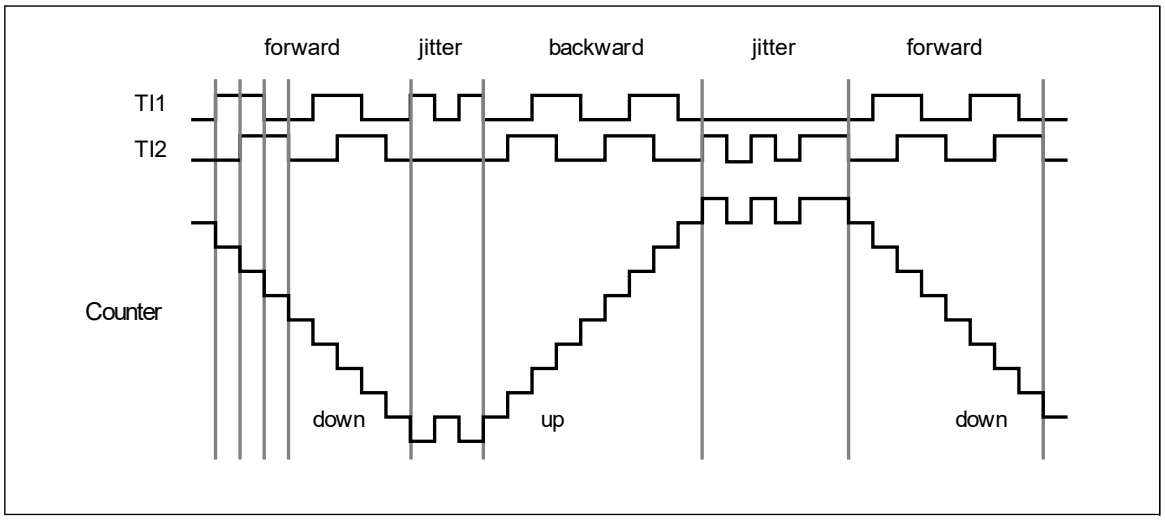



图 15-36 TI1FP1 极性反相时的编码器接口模式示例

定时器配置为编码器接口模式时，会提供传感器当前位置的相关信息。使用另一个配置为捕获模式的定时器测量两个编码器事件之间的周期，可获得动态信息（速度、加速度和减速度）。指示机械零位的编码器输出即可用于此目的。根据两个事件之间的时间间隔，还可定期读取计数器。如果可能，可以将计数器值锁存到第三个输入捕获寄存器来实现此目的（捕获信号必须为周期性信号，可以由另一个定时器产生）；还可以通过由实时时钟生成的 DMA 请求读取计数器值。

### 15.3.13. 定时器输入异或功能

借助 TIMx_CR2 寄存器中的 TI1S 位，可将通道 1 的输入滤波器连接到异或门的输出，从而将 TIMx_CH1 到 TIMx_CH3 这三个输入引脚组合在一起。

异或输出可与触发或输入捕获等所有定时器输入功能配合使用。

### 15.3.14. 定时器与外部触发同步

TIMx 定时器以下列模式与外部触发实现同步：复位模式、门控模式和触发模式。

**从模式：复位模式**

当触发输入信号发生变化时，计数器及其预分频器可重新初始化。此外，如果 TIMx_CR1 寄存器中的 URS 位处于低电平，则会生成更新事件 UEV。然后，所有预装载寄存器（TIMx_ARR 和 TIMx_CCRx）都将更新。

在以下示例中， TI1 输入上出现上升沿时，递增计数器清零：

- 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波时间（本例中不需要任何滤波，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。 CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S = 01。在 TIMx_CCER 寄存器中写入 CC1P=0 和 CC1NP=0，验证极性（仅检测上升沿）。
- 在 TIMx_SMCR 寄存器中写入 SMS=100，将定时器配置为复位模式。在 TIMx_SMCR 寄存器中写入 TS=101，选择 TI1 作为输入源。
- 在 TIMx_CR1 寄存器中写入 CEN=1，启动计数器。

计数器使用内部时钟计数，然后正常运转，直到出现 TI1 上升沿。当 TI1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时，触发标志（TIMx_SR 寄存器中的 TIF 位）置 1，使能中断或 DMA 后，还

可发送中断或 DMA 请求（取决于 TIMx_DIER 寄存器中的 TIE 和 TDE 位）。

下图显示了自动重载寄存器 TIMx_ARR=0x36 时的相关行为。TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

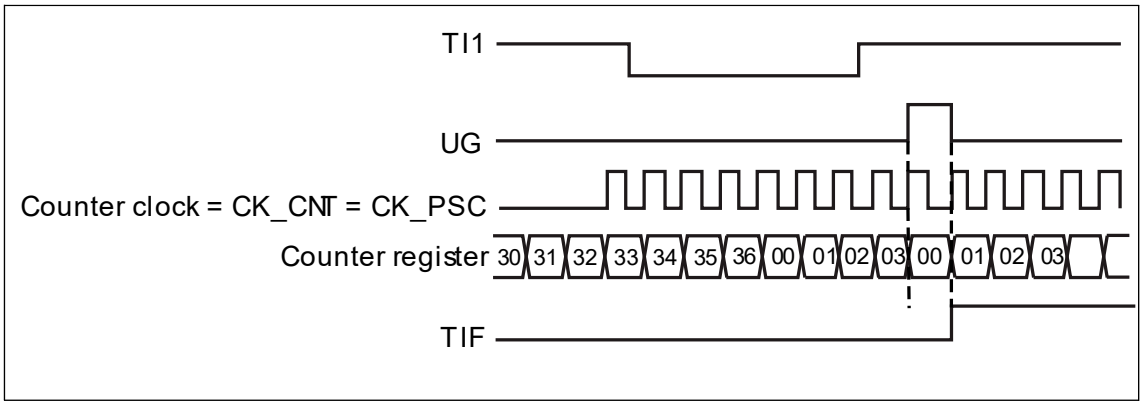



图 15-37 复位模式下的控制电路

**从模式：门控模式**

输入信号的电平可用来使能计数器。

在以下示例中，递增计数器仅在 TI1 输入为低电平时计数：

- 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波时间（本例中不需要任何滤波，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S=01。在 TIMx_CCER 寄存器中写入 CC1P=1，以确定极性（仅检测低电平）。
- 在 TIMx_SMCR 寄存器中写入 SMS=101，将定时器配置为门控模式。在 TIMx_SMCR 寄存器中写入 TS=101，选择 TI1 作为输入源。
- 在 TIMx_CR1 寄存器中写入 CEN=1，使能计数器（在门控模式下，如果 CEN=0，则无论触发输入电平如何，计数器都不启动）。

只要 TI1 为低电平，计数器就开始根据内部时钟计数，直到 TI1 变为高电平时停止计数。计数器启动或停止时， TIMx_SR 寄存器中的 TIF 标志都会置 1。

TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

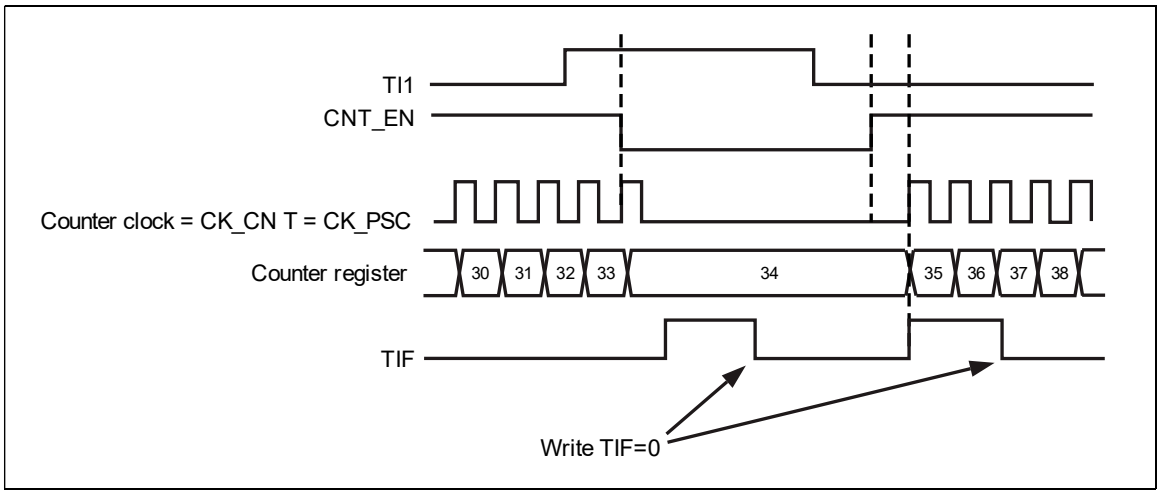



图 15-38 门控模式下的控制电路

1. 由于门控模式作用于电平而非边沿，因此在门控模式下，“CCxP=CCxNP=1”（同时检测上升沿和下降

沿）不发挥任何作用。

**从模式：触发模式**

所选输入上发生某一事件时可以启动计数器。

在以下示例中， TI2 输入上出现上升沿时，递增计数器启动：

- 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波时间（本例中不需要任何滤波，因此保持 IC2F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC2S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC2S=01。在 TIMx_CCER 寄存器中写入 CC2P=1，以确定极性（仅检测低电平）。
- 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIMx_SMCR 寄存器中写入 TS=110，选择 TI2 作为输入源。

当 TI2 出现上升沿时，计数器开始根据内部时钟计数，并且 TIF 标志置 1。

TI2 的上升沿与实际计数器启动之间的延迟是由于 TI2 输入的重新同步电路引起的。

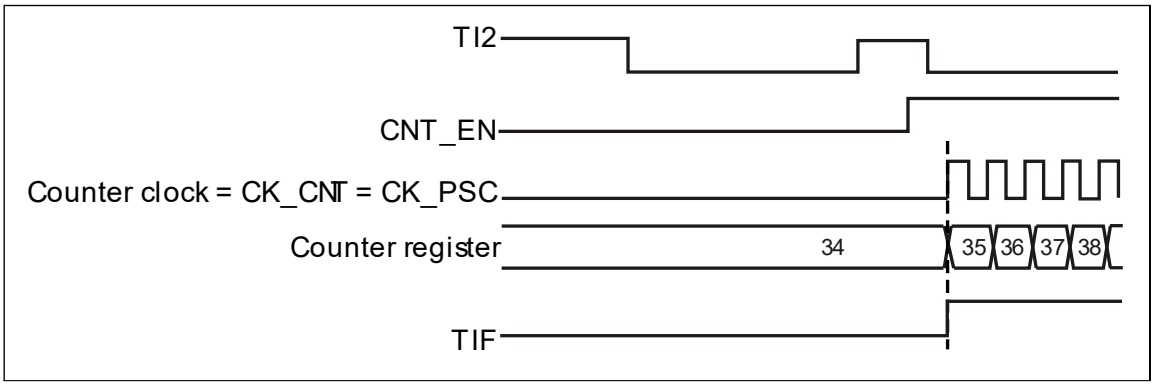



图 15-39 触发模式下的控制电路

**从模式：外部时钟模式 2 + 触发模式**

外部时钟模式 2 可与另一种从模式（外部时钟模式 1 和编码器模式除外）结合使用。这种情况下， ETR 信号用作外部时钟输入，在复位模式、门控模式或触发模式下工作时，可选择另一个输入作为触发输入。不建议通过 TIMx_SMCR 寄存器中的 TS 位来选择 ETR 作为 TRGI。

在以下示例中，只要 TI1 出现上升沿，递增计数器即会在 ETR 信号的每个上升沿处递增：

- 通过对 TIMx_SMCR 寄存器进行如下编程，配置外部触发输入电路：
  - ETF = 0000：无滤波器
  - ETPS = 00：禁止预分频器
  - ETP = 0：检测 ETR 的上升沿，并写入 ECE=1，以使能外部时钟模式 2。
- 如下配置通道 1，以检测 TI 的上升沿：
  - IC1F = 0000：无滤波器
  - 由于捕获预分频器不用于触发操作，因此无需对其进行配置
  - TIMx_CCMR1 寄存器中的 CC1S = 01，只选择输入捕获源
  - TIMx_CCER 寄存器中的 CC1P = 0，以确定极性（仅检测上升沿）。
- 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIMx_SMCR 寄存器中写入 TS=101，选择 TI1 作为输入源。

TI1 出现上升沿时将使能计数器并且 TIF 标志置 1。然后计数器在 ETR 出现上升沿时计数。

ETR 信号的上升沿与实际计数器复位之间的延迟是由于 ETRP 输入的重新同步电路引起的。

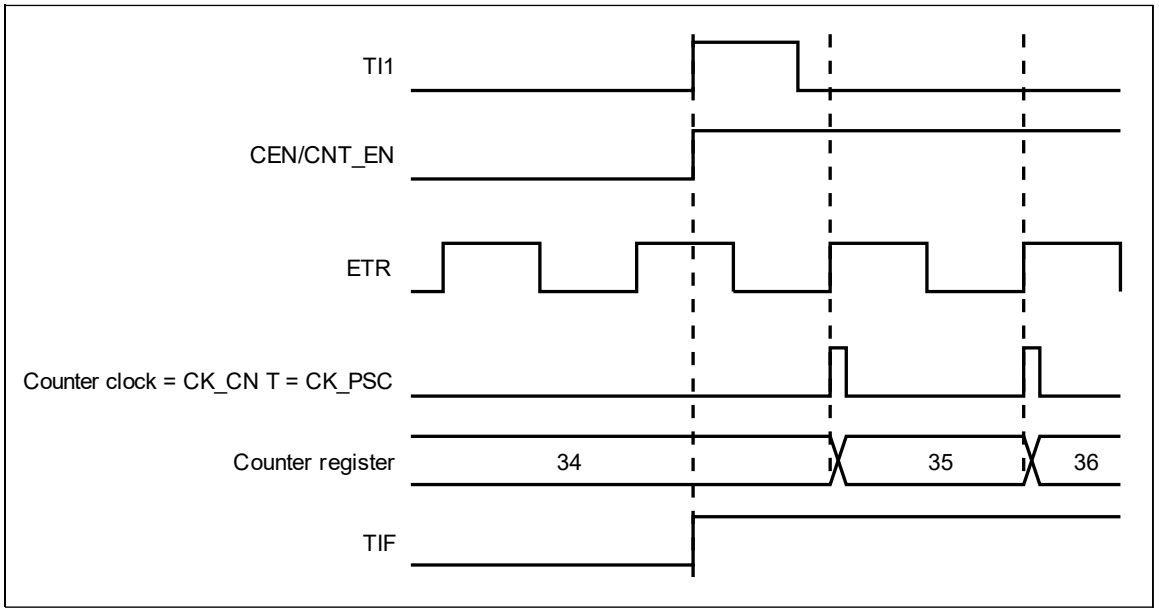



图 15-40 外部时钟模式 2 + 触发模式下的控制电路

## 15.3.15. 定时器同步

TIMx 定时器从内部连接在一起，以实现定时器同步或级联。当某个定时器配置为主模式时，可对另一个配置为从模式的定时器的计数器执行复位、启动、停止操作或为其提供时钟。

下图 为主/从定时器示例简要介绍了触发选择和主模式选择框图。

将一个定时器用作另一个定时器的预分频器。

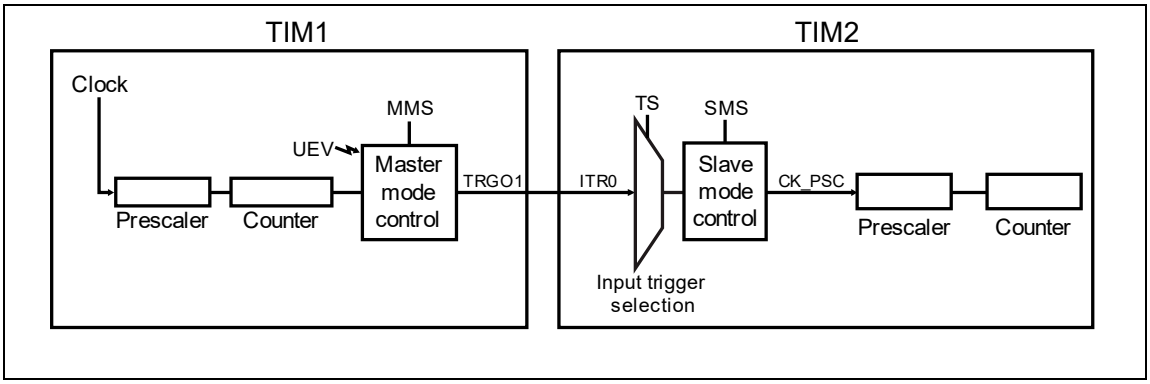



图 15-41 主/从定时器示例

例如，可以将定时器 1 配置为定时器 2 的预分频器。请参见上图。为此：

- 将定时器 1 配置为主模式，以便每次发生更新事件 UEV 时都输出一个周期性触发信号。如果在 TIM1_CR2 寄存器中写入 MMS=010，则每次生成更新事件时，TRGO1 都会输出一个上升沿。
- 要将定时器 1 的 TRGO1 输出连接到定时器 2，必须将定时器 2 配置为从模式，使用 ITR0 作为内部触发。通过 TIM2_SMCR 寄存器中的 TS 位（写入 TS=000）可对此进行选择。
- 然后将从模式控制器设为外部时钟模式 1（在 TIM2_SMCR 寄存器中写入 SMS=111）。这样一来，定

时器 2 的时钟将由定时器 1 周期性触发信号的上升沿（与定时器 1 的计数器上溢对应）提供。

- 最后必须通过将这两个定时器的相应 CEN 位（TIMx_CR1 寄存器）置 1 同时使能二者。

*注意： 如果选择定时器 1 的 OCx 信号作为触发输出 (MMS=1xx)，该信号的上升沿将用于驱动定时器 2 的计数器。*

**使用一个定时器使能另一个定时器**

本例中通过定时器 1 的输出比较 1 来使能定时器 2。相关连接图请参见上图。仅当定时器 1 的 OC1REF 为高电平时，定时器 2 才根据分频后的内部时钟进行计数。两个计数器的时钟频率都基于 CK_INT 通过预分频器执行 3 分频 ($f_{CK_CNT} = f_{CK_INT}/3$)。

- 将定时器 1 配置为主模式，发送其输出比较 1 参考信号 (OC1REF) 作为触发输出（TIM1_CR2 寄存器中的 MMS=100）。
- 配置定时器 1 的 OC1REF 波形（TIM1_CCMR1 寄存器）。
- 配置定时器 2 以接收来自定时器 1 的输入触发（TIM2_SMCR 寄存器中的 TS=000）。
- 将定时器 2 配置为门控模式（TIM2_SMCR 寄存器中的 SMS=101）。
- 通过向 CEN 位（TIM2_CR1 寄存器）写入“1”使能定时器 2。
- 通过向 CEN 位（TIM1_CR1 寄存器）写入“1”启动定时器 1。

*注意： 计数器 2 的时钟与计数器 1 不同步，此模式仅影响定时器 2 的计数器使能信号*

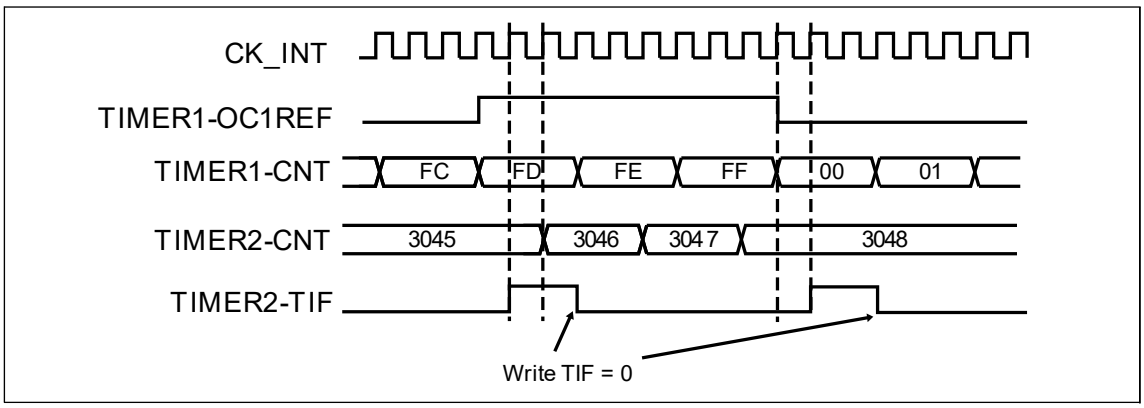



图 15-42 使用定时器 1 的 OC1REF 对定时器 2 实施门控控制

在上图的示例中，定时器 2 的计数器和预分频器在启动前未进行初始化。因此从各自的当前值开始计数。启动定时器 1 之前，通过复位这两个定时器可以从指定值开始计数。这样便可以在定时器计数器中写入所需的任意值。两个定时器都可通过软件使用 TIMx_EGR 寄存器中的 UG 位轻松复位。

在下一示例中，定时器 1 与定时器 2 同步。定时器 1 为主模式，从 0 开始计数。定时器 2 为从模式，从 0xE7 开始计数。两个定时器的预分频比相同。在 TIM1_CR1 寄存器中通过向 CEN 位写入“0”来禁止定时器 1 时，定时器 2 将停止：

- 将定时器 1 配置为主模式，发送其输出比较 1 参考信号 (OC1REF) 作为触发输出 (TIM1_CR2 寄存器中的 MMS=100）。
- 配置定时器 1 的 OC1REF 波形（TIM1_CCMR1 寄存器）。
- 配置定时器 2 以接收来自定时器 1 的输入触发（TIM2_SMCR 寄存器中的 TS=000）。
- 将定时器 2 配置为门控模式（TIM2_SMCR 寄存器中的 SMS=101）。
- 通过向 UG 位（TIM1_EGR 寄存器）写入“1”复位定时器 1。
- 通过向 UG 位（TIM2_EGR 寄存器）写入“1”复位定时器 2。
- 通过在定时器 2 的计数器 (TIM2_CNTL) 中写入“0xE7”使定时器 2 初始化为 0xE7。
- 通过向 CEN 位（TIM2_CR1 寄存器）写入“1”使能定时器 2。

- 通过向 CEN 位（TIM1_CR1 寄存器）写入“1”启动定时器 1。
- 通过向 CEN 位（TIM1_CR1 寄存器）写入“0”停止定时器 1。

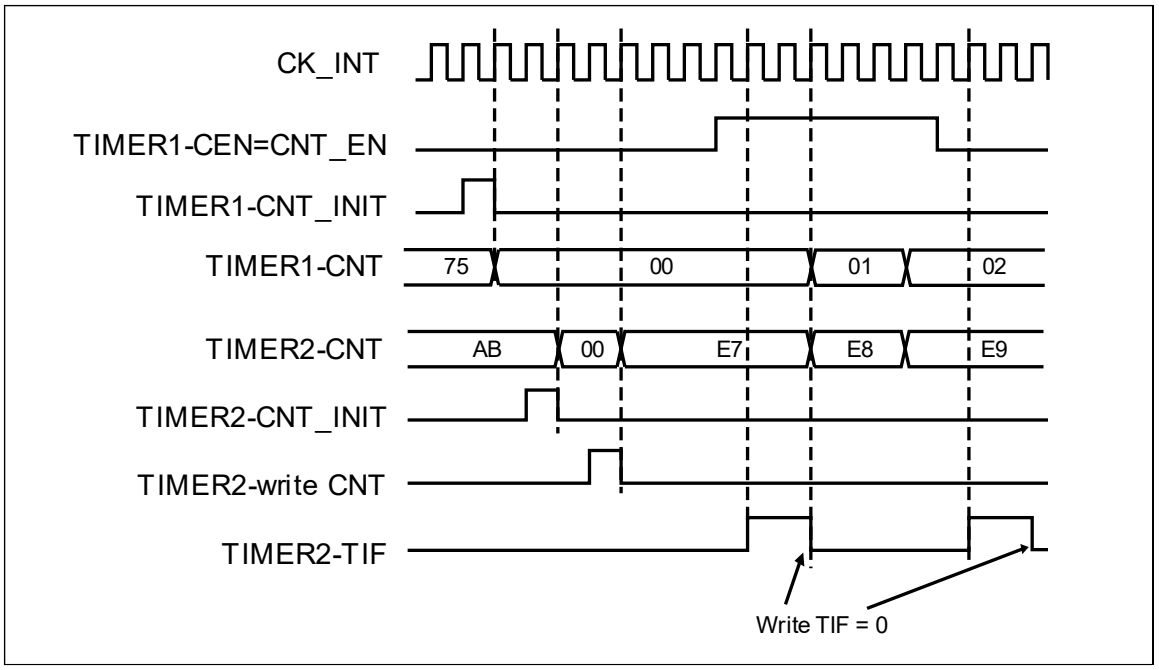



图 15-43 使用定时器 1 的使能信号对定时器 2 实施门控控制

**使用一个定时器启动另一个定时器**

本例中使用定时器 1 的更新事件使能定时器 2。相关连接图，请参见图 21-41。只要定时器 1 生成更新事件，定时器 2 便根据分频后的内部时钟从当前值（可以不为 0）开始计数。定时器 2 收到触发信号时，其 CEN 位自动置 1，并且计数器开始计数，直到向 TIM2_CR1 寄存器的 CEN 位写入“0”后停止计数。两个计数器的时钟频率都基于 CK_INT 通过预分频器执行 3 分频（$f_{CK_CNT} = f_{CK_INT}/3$）。

- 将定时器 1 配置为主模式，发送其更新事件（UEV）作为触发输出（TIM1_CR2 寄存器中的 MMS=010）。
- 配置定时器 1 的周期（TIM1_ARR 寄存器）。
- 配置定时器 2 以接收来自定时器 1 的输入触发（TIM2_SMCR 寄存器中的 TS=000）。
- 将定时器 2 配置为触发模式（TIM2_SMCR 寄存器中的 SMS=110）。
- 通过向 CEN 位（TIM1_CR1 寄存器）写入“1”启动定时器 1。

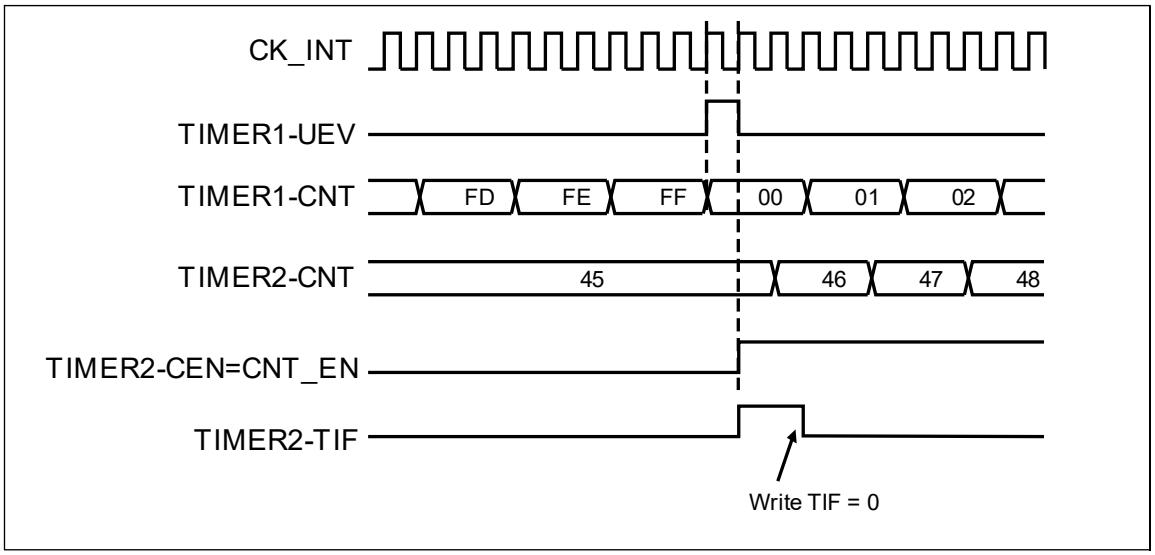



图 15-44 使用定时器 1 的更新事件触发定时器 2

如上述示例所示，用户可以在开始计数之前初始化两个计数器。下图显示了与上图具有相同配置，只不过处于触发模式（ TIM2_SMCR 寄存器中的 SMS=110） 而非门控模式的计数行为。

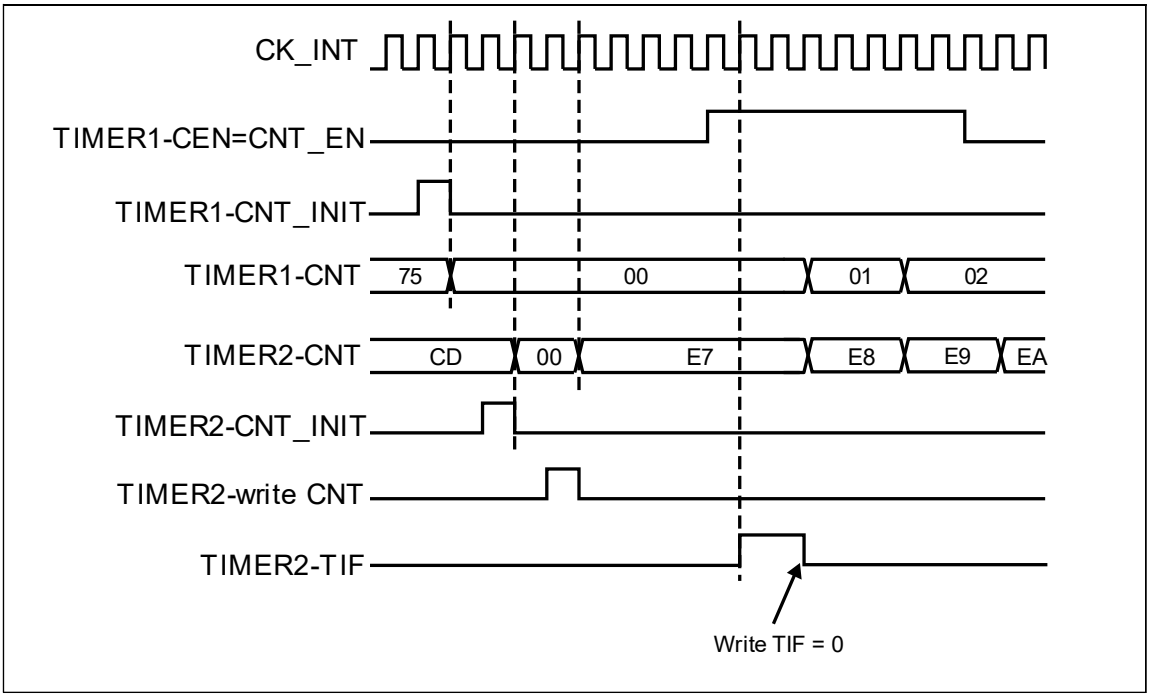



图 15-45 使用定时器 1 的使能信号触发定时器 2

**将一个定时器用作另一个定时器的预分频器**

例如，可以将定时器 1 配置为定时器 2 的预分频器。相关连接图，请参见图 21-41。为此：

- 将定时器 1 配置为主模式，发送其更新事件 (UEV) 作为触发输出（TIM1_CR2 寄存器中的 MMS=010）。这样便会在计数器每次发生上溢时输出一个周期性信号。
- 配置定时器 1 的周期（TIM1_ARR 寄存器）。
- 配置定时器 2 以接收来自定时器 1 的输入触发（TIM2_SMCR 寄存器中的 TS=000）。
- 将定时器 2 配置为外部时钟模式（TIM2_SMCR 寄存器中的 SMS=111）。

- 通过向 CEN 位（TIM2_CR1 寄存器）写入“1”启动定时器 2。
- 通过向 CEN 位（TIM1_CR1 寄存器）写入“1”启动定时器 1。

**使用一个外部触发同步的启动 2 个定时器**

本例中，定时器 1 的 TI1 输入出现上升沿时使能定时器 1，使能定时器 1 的同时使能定时器 2。相关连接图，请参见图 21-41。要确保两个计数器对齐，定时器 1 必须配置为主/从模式（对应的 TI1 为从，对应定时器 2 为主）：

- 将定时器 1 配置为主模式，发送其使能信号作为触发输出（TIM1_CR2 寄存器中的 MMS=001）。
- 将定时器 1 配置为从模式以接收来自 TI1 的输入触发（TIM1_SMCR 寄存器中的 TS=100）。
- 将定时器 1 配置为触发模式（TIM1_SMCR 寄存器中的 SMS=110）。
- 通过写入 MSM=1（TIMx_SMCR 寄存器）将定时器 1 配置为主/从模式。
- 配置定时器 2 以接收来自定时器 1 的输入触发（TIM2_SMCR 寄存器中的 TS=000）。
- 将定时器 2 配置为触发模式（TIM2_SMCR 寄存器中的 SMS=110）。

当 TI1（定时器 1）出现上升沿时，两个计数器开始根据内部时钟同步计数，并且两个 TIF 标志都置 1。

*注意： 本例中，两个定时器都在启动之前进行了初始化（通过将各自的 UG 位置 1）。两个计数器都从 0 开始计数，但可以通过对任意一个计数器寄存器 (TIMx_CNT) 进行写操作，在二者之间轻松插入一个偏移量。可注意到主/从模式在定时器 1 的 CNT_EN 与 CK_PSC 之间产生了延迟。*

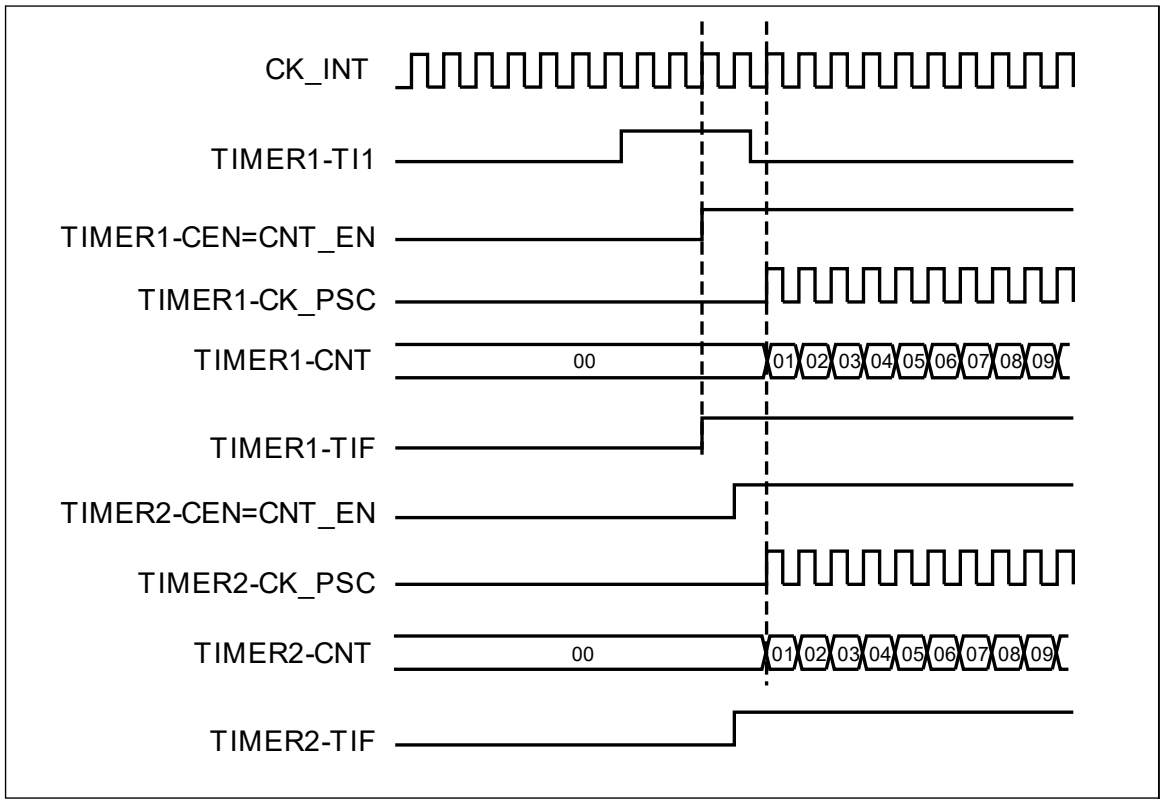



图 15-46 使用定时器 1 的 TI1 输入触发定时器 1 和定时器 2

## 15.3.16. 调试模式

当微控制器进入调试模式(内核停止)时， TIMx 计数器会根据 DBGMCU 模块中的 DBG_TIMx_STOP 配置位选择继续正常工作或者停止工作。

## 15.4. TIM2 到 TIM5 寄存器

有关寄存器说明中使用的缩写，请参见第 1.1 节。

外设寄存器必须按半字（16 位）或字（32 位）进行写访问。而读访问则可按字节（8 位）、半字（ 16 位）或字（ 32 位）进行。

### 15.4.1. TIMx 控制寄存器 1(TIMx_CR1)

偏移地址：0x00

复位值：0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:10 | 保留 | 必须保持复位值。 |
| 9:8 | CKD[1:0] | 时钟分频<br>此位域指示定时器时钟 (CK_INT) 频率与死区发生器以及数字滤波器（ETR、TIx）所使用的死区及采样时钟 ($t_{DTS}$) 之间的分频比，<br>00：$t_{DTS} = t_{CK_INT}$<br>01：$t_{DTS} = 2 * t_{CK_INT}$<br>10：$t_{DTS} = 4 * t_{CK_INT}$<br>11：保留，不要设置成此值 |
| 7 | ARPE | 自动重载预装载使能<br>0：TIMx_ARR 寄存器不进行缓冲<br>1：TIMx_ARR 寄存器进行缓冲 |
| 6:5 | CMS[1:0] | 中心对齐模式选择<br>00：边沿对齐模式。计数器根据方向位 (DIR) 递增计数或递减计数。<br>01：中心对齐模式 1。计数器交替进行递增计数和递减计数。仅当计数器递减计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CxS=00）的输出比较中断标志才置 1。<br>10：中心对齐模式 2。计数器交替进行递增计数和递减计数。仅当计数器递增计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CxS=00）的输出比较中断标志才置 1。<br>11：中心对齐模式 3。计数器交替进行递增计数和递减计数。当计数器递增计数或递减计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CxS=00）的输出比较中断标志都会置 1。<br>*注意： 只要计数器处于使能状态 (CEN=1)，就不得从边沿对齐模式切换为中心对齐模式。* |
| 4 | DIR | 方向<br>0：计数器递增计数<br>1：计数器递减计数<br>*注意： 当定时器配置为中心对齐模式或编码器模式时，该位为只读状态。* |
| 3 | OPM | 单脉冲模式<br>0：计数器在发生更新事件时不会停止计数<br>1：计数器在发生下一更新事件时停止计数（将 CEN 位清零） |
| 2 | URS | 更新请求源 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 此位由软件置 1 和清零，用以选择 UEV 事件源。<br>0：失能时，所有以下事件都会生成更新中断或 DMA 请求。此类事件包括：<br>计数器上溢/下溢<br>将 UG 位置 1<br>通过从模式控制器生成的更新事件<br>1：使能时，只有计数器上溢/下溢会生成更新中断或 DMA 请求。 |
| 1 | UDIS | 更新禁止<br>此位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。<br>0：使能 UEV。更新 (UEV) 事件可通过以下事件之一生成：<br>计数器上溢/下溢<br>将 UG 位置 1<br>通过从模式控制器生成的更新事件<br>然后更新影子寄存器的值。<br>1：禁止 UEV。不会生成更新事件，各影子寄存器的（ARR、PSC 和 CCRx）保持不变。<br>但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。 |
| 0 | CEN | 计数器使能<br>0：禁止计数器<br>1：使能计数器<br>*注意： 只有事先通过软件将 CEN 位置 1，才可以使用外部时钟、门控模式和编码器模式。而触发模式可通过硬件自动将 CEN 位置 1。* |

### 15.4.2. TIMx 控制寄存器 2(TIMx_CR2)

偏移地址：0x04

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | TI1S | MMS[2:0] | | | CCDS | Reserved | | |
| | | | | | | | | rw | rw | rw | rw | rw | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:8 | 保留 | 必须保持复位值。 |
| 7 | TI1S | TI1 选择<br>0：TIMx_CH1 引脚连接到 TI1 输入<br>1：TIMx_CH1、TIMx_CH2 和 TIMx_CH3 引脚连接到 TI1 输入（异或组合） |
| 6:4 | MMS | 主模式选择<br>这些位可选择主模式下将要发送到从定时器以实现同步的信息 (TRGO)。这些位的组合如下：<br>000：复位 – TIMx_EGR 寄存器中的 UG 位用作触发输出 (TRGO)。如果复位由触发输入生成（从模式控制器配置为复位模式），则 TRGO 上的信号相比实际复位会有延迟。<br>001：使能 – 计数器使能信号 (CNT_EN) 用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号可由 CEN 控制位产生。当配置为门控模式时，也可由触发 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 输入产生。当计数器使能信号由触发输入控制时，TRGO 上会存在延迟，选择主/从模式时除外（请参见 TIMx_SMCR 寄存器中 MSM 位的说明）。<br>010：更新--选择更新事件作为触发输出 (TRGO)。例如，主定时器可用作从定时器的预分频器。<br>011：比较脉冲 — 一旦发生输入捕获或比较匹配事件，当 CC1IF 被置 1 时（即使已为高电平），触发输出都会发送一个正脉冲 (TRGO)。(TRGO)<br>100：比较 — OC1REF 信号用作触发输出 (TRGO)<br>101：比较 — OC2REF 信号用作触发输出 (TRGO)<br>110：比较 — OC3REF 信号用作触发输出 (TRGO)<br>111：比较 — OC4REF 信号用作触发输出 (TRGO) |
| 3 | CCDS | 捕获/比较 DMA 选择<br>0：发生 CCx 事件时发送 CCx DMA 请求<br>1：发生更新事件时发送 CCx DMA 请求 |
| 2:0 | 保留 | 保必须保持复位值。 |

### 15.4.3. TIMx 从模式控制寄存器(TIMx_SMCR)

偏移地址：0x08

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ETP | ECS | ETPS[1:0] | | ETF[3:0] | | | | MSM | TS[2:0] | | | Res. | SMS[2:0] | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15 | ETP | 外部触发极性<br>此位可选择将 ETR 还是 ETR 用于触发操作<br>0：ETR 未反相，高电平或上升沿有效。<br>1：ETR 反相，低电平或下降沿有效。 |
| 14 | ECE | 外部时钟使能<br>此位可使能外部时钟模式 2。<br>0：禁止外部时钟模式 2<br>1：使能外部时钟模式 2。计数器时钟由 ETRF 信号的任意有效边沿提供。<br>注意：<br>1：将 ECE 位置 1 与选择外部时钟模式 1 并将 TRGI 连接到 ETRF（SMS=111 且 TS=111）具有相同效果。<br>2：外部时钟模式 2 可以和以下从模式同时使用：复位模式、门控模式和触发模式。不过此类情况下 TRGI 不得连接 ETRF（TS 位不得为 111）。<br>3：如果同时使能外部时钟模式 1 和外部时钟模式 2，则外部时钟输入为 ETRF。 |
| 13:12 | ETPS[1:0] | 外部触发预分频器<br>外部触发信号 ETRP 频率不得超过 TIMxCLK 频率的 1/4。可通过使能预分频器来降低 ETRP 频率。这种方法在输入快速外部时钟时非常有用。<br>00：预分频器关闭<br>01：2 分频 ETRP 频率 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 10：4 分频 ETRP 频率<br>11：8 分频 ETRP 频率 |
| 11:8 | ETF[3:0] | 外部触发滤波器<br>此位域可定义 ETRP 信号的采样频率和适用于 ETRP 的数字滤波时间。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿：<br>0000：无滤波器，按 fDTS 频率进行采样<br>0001：$f_{SAMPLING} = f_{CK_INT}$，N=2<br>0010：$f_{SAMPLING} = f_{CK_INT}$，N=4<br>0011：$f_{SAMPLING} = f_{CK_INT}$，N=8<br>0100：$f_{SAMPLING} = f_{DTS}/2$，N=6<br>0101：$f_{SAMPLING} = f_{DTS}/2$，N=8<br>0110：$f_{SAMPLING} = f_{DTS}/4$，N=6<br>0111：$f_{SAMPLING} = f_{DTS}/4$，N=8<br>1000：$f_{SAMPLING} = f_{DTS}/8$，N=6<br>1001：$f_{SAMPLING} = f_{DTS}/8$，N=8<br>1010：$f_{SAMPLING} = f_{DTS}/16$，N=5<br>1011：$f_{SAMPLING} = f_{DTS}/16$，N=6<br>1100：$f_{SAMPLING} = f_{DTS}/16$，N=8<br>1101：$f_{SAMPLING} = f_{DTS}/32$，N=5<br>1110：$f_{SAMPLING} = f_{DTS}/32$，N=6<br>1111：$f_{SAMPLING} = f_{DTS}/32$，N=8 |
| 7 | MSM | 主/从模式<br>0：不执行任何操作<br>1：当前定时器的触发输入事件 (TRGI) 的动作被推迟，以使当前定时器与其从定时器实现完美同步（通过 TRGO）。此设置适用于由单个外部事件对多个定时器进行同步的情况。 |
| 6:4 | TS[2:0] | 触发选择<br>此位域可选择将要用于同步计数器的触发输入。<br>000：内部触发 0 (ITR0)<br>001：内部触发 1 (ITR1)<br>010：内部触发 2 (ITR2)<br>011：内部触发 3 (ITR3)<br>100：TI1 边沿检测器 (TI1F_ED)<br>101：滤波后的定时器输入 1 (TI1FP1)<br>110：滤波后的定时器输入 2 (TI2FP2)<br>111：外部触发输入 (ETRF)<br>*注意：这些位只能在未使用的情况下（例如，SMS=000 时）进行更改，以避免转换时出现错的边沿检测。* |
| 3 | 保留 | 必须保持复位值。 |
| 2:0 | SMS | 从模式选择<br>选择外部信号时，触发信号 (TRGI) 的有效边沿与外部输入上所选的极性相关（请参见输入控制寄存器和控制寄存器说明）。<br>000：禁止从模式--如果 CEN =“1”，预分频器时钟直接由内部时钟提供。<br>001：编码器模式 1--计数器根据 TI1FP1 电平在 TI2FP2 边沿递增/递减计数。<br>010：编码器模式 2--计数器根据 TI2FP2 电平在 TI1FP1 边沿 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 递增/递减计数。<br>011: 编码器模式 3--计数器在 TI1FP1 和 TI2FP2 的边沿计数，计数的方向取决于另外一个信号的电平。<br>100: 复位模式--在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器并生成一个寄存器更新事件。<br>101: 门控模式--触发输入 (TRGI) 为高电平时使能计数器时钟。只要触发输入变为低电平，计数器立即停止计数（但不复位）。计数器的启动和停止都是受控的。<br>110: 触发模式--触发信号 TRGI 出现上升沿时启动计数器（但不复位）。只控制计数器的启动。<br>111: 外部时钟模式 1--由所选触发信号 (TRGI) 的上升沿提供计数器时钟。<br>*注意： 如果将 TI1F_ED 选作触发输入（TS="100"），则不得使用门控模式。实际上，TI1F 每次转换时，TI1F_ED 都输出 1 个脉冲，而门控模式检查的则是触发信号的电平。* |

表 15-2 TIMx 内部触发连接

| 从 TIM | ITR0(TS = 000) | ITR1(TS = 001) | ITR2(TS = 010) | ITR3(TS = 011) |
|---|---|---|---|---|
| TIM2 | TIM1 | TIM8 | TIM3 | TIM4 |
| TIM3 | TIM1 | TIM2 | TIM5 | TIM4 |
| TIM4 | TIM1 | TIM2 | TIM3 | TIM8 |
| TIM5 | TIM2 | TIM3 | TIM4 | TIM8 |

### 15.4.4. TIMx DMA/中断使能寄存器(TIMx_DIE)

偏移地址：0x0C

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | TDE | Res. | CC4DE | CC3DE | CC2DE | CC1DE | UDE | Res. | TIE | Res. | CC4IE | CC3IE | CC2IE | CC1E | UIE |
| | rw | | rw | rw | rw | rw | rw | | rw | | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15 | 保留 | 必须保持复位值。 |
| 14 | TDE | 触发 DMA 请求使能<br>0：禁止触发 DMA 请求。<br>1：使能触发 DMA 请求。 |
| 13 | 保留 | 始终读为 0 |
| 12 | CC4DE | 捕获/比较 4 DMA 请求使能<br>0：禁止 CC4 DMA 请求。<br>1：使能 CC4 DMA 请求。 |
| 11 | CC3DE | 捕获/比较 3 DMA 请求使能<br>0：禁止 CC3 DMA 请求。<br>1：使能 CC3 DMA 请求。 |
| 10 | CC2DE | 捕获/比较 2 DMA 请求使能<br>0：禁止 CC2 DMA 请求。<br>1：使能 CC2 DMA 请求。 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 9 | CC1DE | 捕获/比较 1 DMA 请求使能<br>0：禁止 CC1 DMA 请求。<br>1：使能 CC1 DMA 请求。 |
| 8 | UDE | 更新 DMA 请求使能<br>0：禁止更新 DMA 请求。<br>1：使能更新 DMA 请求。 |
| 7 | 保留 | 必须保持复位值。 |
| 6 | TIE | 触发信号(TRGI)中断使能<br>0：禁止触发信号 (TRGI) 中断。<br>1：使能触发信号 (TRGI) 中断。 |
| 5 | 保留 | 必须保持复位值。 |
| 4 | CC4IE | 捕获/比较 4 中断使能<br>0：禁止 CC4 中断。<br>1：使能 CC4 中断。 |
| 3 | CC3IE | 捕获/比较 3 中断使能<br>0：禁止 CC3 中断<br>1：使能 CC3 中断 |
| 2 | CC2IE | 捕获/比较 2 中断使能<br>0：禁止 CC2 中断<br>1：使能 CC2 中断 |
| 1 | CC1IE | 捕获/比较 1 中断使能<br>0：禁止 CC1 中断<br>1：使能 CC1 中断 |
| 0 | UIE | 更新中断使能<br>0：禁止更新中断<br>1：使能更新中断 |

### 15.4.5. TIMx 状态寄存器(TIMx_SR)

偏移地址：0x10

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | CC4OF | CC3OF | CC2OF | CC1OF | Reserved | | TIF | Res. | CC4IF | CC3IF | CC2IF | CC1IF | UIF |
| | | | rc_w0 | rc_w0 | rc_w0 | rc_w0 | | | rc_w0 | | rc_w0 | rc_w0 | rc_w0 | rc_w0 | rc_w0 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:13 | 保留 | 必须保持复位值。 |
| 12 | CC4OF | 捕获/比较 4 重复捕获标志<br>请参见 CC1OF 说明 |
| 11 | CC3OF | 捕获/比较 3 重复捕获标志<br>请参见 CC1OF 说明 |
| 10 | CC2OF | 捕获/比较 2 重复捕获标志<br>请参见 CC1OF 说明 |
| 9 | CC1OF | 捕获/比较 1 重复捕获标志<br>仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将该位清零。<br>0：未检测到重复捕获 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 1:TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。 |
| 8:7 | 保留 | 必须保持复位值。 |
| 6 | TIF | 触发中断标志<br>在除门控模式以外的所有模式下，当使能从模式控制器后在 TRGI 输入上检测到有效边沿时，该标志将由硬件置 1。选择门控模式时，该标志将在计数器启动或停止时置 1。但需要通过软件清零。<br>0：未发生触发信号 (TRGI) 事件<br>1：触发信号 (TRGI) 中断挂起 |
| 5 | 保留 | 必须保持复位值。 |
| 4 | CC4IF | 捕获/比较 4 中断标志<br>请参见 CC1IF 说明 |
| 3 | CC3IF | 捕获/比较 3 中断标志<br>请参见 CC1IF 说明 |
| 2 | CC2IF | 捕获/比较 2 中断标志<br>请参见 CC1IF 说明 |
| 1 | CC1IF | 捕获/比较 1 中断标志<br>**如果通道 CC1 配置为输出：**<br>当计数器与比较值匹配时，此标志由硬件置 1，中心对齐模式下除外（请参见 TIMx_CR1 寄存器中的 CMS 位说明）。但需要通过软件清零。<br>0：不匹配<br>1：TIMx_CNT 计数器的值与 TIMx_CCR1 寄存器的值匹配。当 TIMx_CCR1 的值大于 TIMx_ARR 的值时，CC1IF 位将在计数器发生上溢（递增计数模式和增减计数模式下）或下溢（递减计数模式下）时变为高电平。<br>**如果通道 CC1 配置为输入：**<br>此位将在发生捕获事件时由硬件置 1。通过软件或读取 TIMx_CCR1 寄存器将该位清零。<br>0：未发生输入捕获事件<br>1：TIMx_CCR1 寄存器中已捕获到计数器值（IC1 上已检测到与所选极性匹配的边沿） |
| 0 | UIF | 更新中断标志<br>该位在发生更新事件时通过硬件置 1。但需要通过软件清零。<br>0：未发生更新。<br>1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：<br>上溢或下溢（对于 TIM2 到 TIM5）以及当 TIMx_CR1 寄存器中 UDIS = 0 时。<br>TIMx_CR1 寄存器中的 URS = 0 且 UDIS = 0，并且由软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。<br>TIMx_CR1 寄存器中的 URS=0 且 UDIS=0，并且 CNT 由触发事件重新初始化时（参见同步控制寄存器说明）。 |

### 15.4.6. TIMx 事件生成寄存器(TIMx_EGR)

偏移地址：0x14

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | TG | Res. | CC4G | CC3G | CC2G | CC1G | UG |
| | | | | | | | | | w | | w | w | w | w | w |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:7 | 保留 | 必须保持复位值。 |
| 6 | TG | 产生触发信号<br>此位由软件置 1 以生成事件，并由硬件自动清零。<br>0：不执行任何操作<br>1：TIMx_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。 |
| 5 | 保留 | 必须保持复位值。 |
| 4 | CC4G | 捕获/比较 4 生成<br>请参见 CC1G 说明 |
| 3 | CC3G | 捕获/比较 3 生成<br>请参见 CC1G 说明 |
| 2 | CC2G | 捕获/比较 2 生成<br>请参见 CC1G 说明 |
| 1 | CC1G | 捕获/比较 1 生成<br>此位由软件置 1 以生成事件，并由硬件自动清零。<br>0：不执行任何操作<br>1：通道 1 上生成捕获/比较事件：<br>**如果通道 CC1 配置为输出：**<br>使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。<br>**如果通道 CC1 配置为输入：**<br>TIMx_CCR1 寄存器中将捕获到计数器当前值。使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。 |
| 0 | UG | 更新生成<br>该位可通过软件置 1，并由硬件自动清零。<br>0：不执行任何操作<br>1：重新初始化计数器并生成寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。如果选择中心对齐模式或 DIR=0（递增计数），计数器将清零；如果 DIR=1（递减计数），计数器将使用自动重载值 (TIMx_ARR)。 |

### 15.4.7. TIMx 捕获/比较模式寄存器 1(TIMx_CCMR1)

偏移地址：0x18

复位值：0x0000

这些通道可用于输入（捕获模式）或输出（比较模式）模式。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入模式和输出模式下的功能均不同。对于任一给定位，OCxx 用于说明通道配置为输出时该位对应的功能，ICxx 则用于说明通道配置为输入时该位对应的功能。因此，必须注意同一个位在输入阶段和输出阶段具有不同的含义。

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OC2CE | OC2M[2:0] | | | OC2PE | OC2FE | CC2S[1:0] | | OC1CE | OC1M[2:0] | | | OC1PE | OC1FE | CC1S[1:0] | |
| IC2F[3:0] | | | | IC2PSC[1:0] | | | | IC1F[3:0] | | | | IC1PSC[1:0] | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

**输出比较模式：**

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15 | OC2CE | 输出比较 2 清零使能 |
| 14:12 | OC2M[2:0] | 输出比较 2 模式 |
| 11 | OC2PE | 输出比较 2 预装载使能 |
| 10 | OC2FE | 输出比较 2 快速使能 |
| 9:8 | CC2S[1:0] | 捕获/比较 2 选择<br>此位域定义通道方向（输入/输出）以及所使用的输入。<br>00：CC2 通道配置为输出<br>01：CC2 通道配置为输入，IC2 映射到 TI2 上<br>10：CC2 通道配置为输入，IC2 映射到 TI1 上<br>11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效<br>*注意： 仅当通道关闭时（TIMx_CCER 中的 CC2E =“0”），才可向 CC2S 位写入数据。* |
| 7 | OC1CE | 输出比较 1 清零使能<br>0：OC1Ref 不受 ETRF 输入影响<br>1：ETRF 输入上检测到高电平时，OC1Ref 立即清零。 |
| 6:4 | OC1M | 输出比较 1 模式<br>这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 和 OC1N 的有效电平则取决于 CC1P 位和 CC1NP 位。<br>000：冻结 – 输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。（该模式用于生成时基）。<br>001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。<br>010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。<br>011：翻转 – TIMx_CNT=TIMx_CCR1 时，OC1REF 发生翻转。<br>100：强制变为无效电平 – OC1REF 强制变为低电平。<br>101：强制变为有效电平 – OC1REF 强制变为高电平。<br>110：PWM 模式 1 – 在递增计数模式下，只要 TIMx_CNT < TIMx_CCR1，通道 1 便为有效状态，否则为无效状态。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，通道 1 便为无效状态（OC1REF=“0”），否则为有效状态（OC1REF=“1”）。<br>111：PWM 模式 2--在递增计数模式下，只要 TIMx_CNT < TIMx_CCR1，通道 1 便为无效状态，否则为有效状态。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，通道 1 便为有效状态，否则为无效状态。<br>注意：<br>1：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S=“00”（通道配置为输出），这些位即无法修改。<br>2：在 PWM 模式 1 或 PWM 模式 2 下，仅当比较结果发生改变或输出 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 比较模式由“冻结”模式切换到“PWM”模式时，OCREF 电平才会发生更改。<br>3：此位域将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成 COM 事件时，OC1M 有效位才会从预装载位获取新值。 |
| 3 | OC1PE | 输出比较 1 预装载使能<br>0：禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立 即使用新值。<br>1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到活动寄存器中。<br>*注意：*<br>*1：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S=“00”（通道配置为输出），这些位即无法修改。*<br>*2：只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式（TIMx_CR1 寄存器中的 OPM 位置 1）。其它情况下则无法保证该行为。* |
| 2 | OC1FE | 输出比较 1 快速使能<br>此位用于加快触发输入事件对 CC 输出的影响。<br>0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期。<br>1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OCFE 才会起作用。 |
| 1:0 | CC1S | 捕获/比较 1 选择<br>此位域定义通道方向（输入/输出）以及所使用的输入。<br>00：CC1 通道配置为输出<br>01：CC1 通道配置为输入，IC1 映射到 TI1 上<br>10：CC1 通道配置为输入，IC1 映射到 TI2 上<br>11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效<br>*注意： 仅当通道关闭时（TIMx_CCER 中的 CC1E =“0”），才可向 CC1S 位写入数据。输入捕获模式* |

**输入捕获模式：**

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:12 | IC2F | 输入捕获 2 滤波器 |
| 11:10 | IC2PSC[1:0] | 输入捕获 2 预分频器 |
| 9:8 | CC2S | 捕获/比较 2 选择<br>此位域定义通道方向（输入/输出）以及所使用的输入。<br>00：CC2 通道配置为输出<br>01：CC2 通道配置为输入，IC2 映射到 TI2 上<br>10：CC2 通道配置为输入，IC2 映射到 TI1 上<br>11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效<br>*注意： 仅当通道关闭时（TIMx_CCER 中的 CC2E =“0”），才可向 CC2S 位写入数据。* |
| 7:4 | IC1F[3:0] | 输入捕获 1 滤波器<br>此位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿：<br>0000：无滤波器，按 $f_{DTS}$ 频率进行采样 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 0001：$f_{SAMPLING} = f_{CK_INT}$，N=2<br>0010：$f_{SAMPLING} = f_{CK_INT}$，N=4<br>0011：$f_{SAMPLING} = f_{CK_INT}$，N=8<br>0100：$f_{SAMPLING} = f_{DTS}$ /2，N=6<br>0101：$f_{SAMPLING} = f_{DTS}$ /2，N=8<br>0110：$f_{SAMPLING} = f_{DTS}$ /4，N=6<br>0111：$f_{SAMPLING} = f_{DTS}$ /4，N=8<br>1000：$f_{SAMPLING} = f_{DTS}$ /8，N=6<br>1001：$f_{SAMPLING} = f_{DTS}$ /8，N=8<br>1010：$f_{SAMPLING} = f_{DTS}$ /16，N=5<br>1011：$f_{SAMPLING} = f_{DTS}$ /16，N=6<br>1100：$f_{SAMPLING} = f_{DTS}$ /16，N=8<br>1101：$f_{SAMPLING} = f_{DTS}$ /32，N=5<br>1110：$f_{SAMPLING} = f_{DTS}$ /32，N=6<br>1111：$f_{SAMPLING} = f_{DTS}$ /32，N=8 |
| 3:2 | IC1PSC | 输入捕获 1 预分频器<br>此位域定义 CC1 输入 (IC1) 的预分频比。<br>只要 CC1E="0"（TIMx_CCER 寄存器），预分频器便立即复位。<br>00：无预分频器，捕获输入上每检测到一个边沿便执行捕获<br>01：每发生 2 个事件便执行一次捕获<br>10：每发生 4 个事件便执行一次捕获<br>11：每发生 8 个事件便执行一次捕获 |
| 1:0 | CC1S | 捕获/比较 1 选择<br>此位域定义通道方向（输入/输出）以及所使用的输入。<br>00：CC1 通道配置为输出<br>01：CC1 通道配置为输入，IC1 映射到 TI1 上<br>10：CC1 通道配置为输入，IC1 映射到 TI2 上<br>11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效<br>*注意：仅当通道关闭时（TIMx_CCER 中的 CC1E ="0"），才可向 CC1S 位写入数据。* |

### 15.4.8. TIMx 捕获/比较模式寄存器 2(TIMx_CCMR2)

偏移地址：0x1C

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OC4CE | OC4M[2:0] | | | OC4PE | OC4FE | CC4S[1:0] | | OC3CE | OC1M[2:0] | | | OC3PE | OC3FE | CC3S[1:0] | |
| IC4F[3:0] | | | | IC4PSC[1:0] | | | | IC3F[3:0] | | | | IC3PSC[1:0] | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

**输出比较模式：**

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15 | OC4CE | 输出比较 4 清零使能 |
| 14:12 | OC4M | 输出比较 4 模式 |
| 11 | OC4PE | 输出比较 4 预装载使能 |
| 10 | OC4FE | 输出比较 4 快速使能 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 9:8 | CC4S | 捕获/比较 4 选择<br>此位域定义通道方向（输入/输出）以及所使用的输入。<br>00：CC4 通道配置为输出<br>01：CC4 通道配置为输入，IC4 映射到 TI4 上<br>10：CC4 通道配置为输入，IC4 映射到 TI3 上<br>11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效<br>*注意： 仅当通道关闭时（TIMx_CCER 中的 CC4E =“0”），才可向 CC4S 位写入数据。* |
| 7 | OC3CE | 输出比较 3 清零使能 |
| 6:4 | OC3M | 输出比较 3 模式 |
| 3 | OC3PE | 输出比较 3 预装载使能 |
| 2 | OC3FE | 输出比较 3 快速使能 |
| 1:0 | CC3S | 捕获/比较 3 选择<br>此位域定义通道方向（输入/输出）以及所使用的输入。<br>00：CC3 通道配置为输出<br>01：CC3 通道配置为输入，IC3 映射到 TI3 上<br>10：CC3 通道配置为输入，IC3 映射到 TI4 上<br>11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效<br>*注意： 仅当通道关闭时（TIMx_CCER 中的 CC3E =“0”），才可向 CC3S 位写入数据。输入捕获模式* |

**输入捕获模式：**

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:12 | IC4F | 输入捕获 4 滤波器 |
| 11:10 | IC4PSC | 输入捕获 4 预分频器 |
| 9:8 | CC4S | 捕获/比较 4 选择<br>此位域定义通道方向（输入/输出）以及所使用的输入。<br>00：CC4 通道配置为输出<br>01：CC4 通道配置为输入，IC4 映射到 TI4 上<br>10：CC4 通道配置为输入，IC4 映射到 TI3 上<br>11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效<br>*注意： 仅当通道关闭时（TIMx_CCER 中的 CC4E =“0”），才可向 CC4S 位写入数据。* |
| 7:4 | IC3F | 输入捕获 3 滤波器 |
| 3:2 | IC3PSC | 输入捕获 3 预分频器 |
| 1:0 | CC3S | 捕获/比较 3 选择<br>此位域定义通道方向（输入/输出）以及所使用的输入。<br>00：CC3 通道配置为输出<br>01：CC3 通道配置为输入，IC3 映射到 TI3 上<br>10：CC3 通道配置为输入，IC3 映射到 TI4 上<br>11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效<br>*注意： 仅当通道关闭时（TIMx_CCER 中的 CC3E =“0”），才可向 CC3S 位写入数据。* |

### 15.4.9. TIMx 捕获/比较使能寄存器(TIMx_CCER)

偏移地址：0x20

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CC4NP | Res. | CC4P | CC4E | CC3NP | Res. | CC3P | CC3E | CC2NP | Res. | CC2P | CC2E | CC1NP | Res. | CC1P | CC1E |
| rw | | rw | rw | rw | | rw | rw | rw | | rw | rw | rw | | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15 | CC4NP | 捕获/比较 4 输出极性<br>请参见 CC1NP 说明 |
| 14 | 保留 | 必须保持复位值。 |
| 13 | CC4P | 捕获/比较 4 输出极性<br>请参见 CC1P 说明 |
| 12 | CC4E | 捕获/比较 4 输出使能<br>请参见 CC1E 说明 |
| 11 | CC3NP | 捕获/比较 3 输出极性<br>请参见 CC1NP 说明 |
| 10 | 保留 | 必须保持复位值。 |
| 9 | CC3P | 捕获/比较 3 输出极性<br>请参见 CC1P 说明 |
| 8 | CC3E | 捕获/比较 3 输出使能<br>请参见 CC1E 说明 |
| 7 | CC2NP | 捕获/比较 2 输出极性<br>请参见 CC1NP 说明 |
| 6 | 保留 | 必须保持复位值。 |
| 5 | CC2P | 捕获/比较 2 输出极性<br>请参见 CC1P 说明 |
| 4 | CC2E | 捕获/比较 2 输出使能<br>请参见 CC1E 说明 |
| 3 | CC1NP | 捕获/比较 1 输出极性<br>**CC1 通道配置为输出：**<br>在这种情况下， CC1NP 必须保持清零。<br>**CC1 通道配置为输入：**<br>此位与 CC1P 配合使用，用以定义 TI1FP1/TI2FP1 的极性。请参见 CC1P 说明。 |
| 2 | 保留 | 必须保持复位值。 |
| 1 | CC1P | 捕获/比较 1 输出极性<br>**CC1 通道配置为输出：**<br>0：OC1 高电平有效<br>1：OC1 低电平有效<br>**CC1 通道配置为输入：**<br>CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的极性。<br>00：非反相/上升沿触发<br>电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式或编码器模式下执行触发操作）。<br>01：反相/下降沿触发电路对 TIxFP1 下降沿敏感 （在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式或编码器模式下执行触发操作）。 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 10：保留，不使用此配置。<br>11：非反相/上升沿和下降沿均触发电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。编码器模式下不得使用此配置。 |
| 0 | CC1E | 捕获/比较 1 输出使能<br>CC1 通道配置为输出：<br>0：关闭 – OC1 未激活<br>1：开启 – 在相应输出引脚上输出 OC1 信号<br>CC1 通道配置为输入：<br>此位决定了是否可以实际将计数器值捕获到输入捕获/比较寄存器 1 (TIMx_CCR1) 中。<br>0：禁止捕获<br>1：使能捕获 |

表 15-3 标准 OCx 通道输出控制位

| CCxE 位 | OCx 输出状态 |
|---|---|
| 0 | 禁止输出（Ocx=“0”，OCx_EN=“0”） |
| 1 | Ocx=OCxREF+极性，Ocx_EN=“1” |

### 15.4.10. TIMx 计数器(TIMx_CNT)

偏移地址：0x24

复位值：0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:0 | CNT[15:0] | 计数器值 |

### 15.4.11. TIMx 预分频器(TIMx_PSC)

偏移地址：0x28

复位值：0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:0 | PSC[15:0] | 预分频器值<br>计数器时钟频率 (CK_CNT) 等于 $f_{CK_PSC}$ / (PSC[15:0] + 1)。<br>PSC 包含每次发生更新事件（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时，或在配置为“复位模式”时通过触发控制器清零时）时要装载到活动预分频器寄存器的值。 |

### 15.4.12. TIMx 自动重载寄存器(TIMx_ARR)

偏移地址：0x2C

复位值：0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:0 | ARR[15:0] | 自动重载值<br>ARR 为要装载到实际自动重载寄存器的值。 |

### 15.4.13. TIMx 捕获/比较寄存器 1 (TIMx_CCR1)

偏移地址：0x34

复位值：0x0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CCR1[31:16](depending on times) | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CCR1[15:0] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | CCR2[31:16] | 捕获/比较 2 的高 16 位（对于 TIM2 和 TIM5）。 |
| 15:0 | CCR2[15:0] | 捕获/比较 2 的低 16 位<br>**如果通道 CC2 配置为输出：**<br>CCR2 为要装载到实际捕获/比较 2 寄存器的值（预装载值）。<br>如果没有通过 TIMx_CCMR 寄存器中的 OC2PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 2 寄存器中。<br>实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC2 输出上发出信号的值。<br>**如果通道 CC2 配置为输入：**<br>CCR2 为上一个输入捕获 2 事件 (IC2) 发生时的计数器值。 |

### 15.4.14. TIMx 捕获/比较比较器 2 (TIMx_CCR2)

偏移地址：0x38

复位值：0x0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CCR2[31:16] (depending on times) | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CCR2[15:0] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | CCR3[31:16] | 捕获/比较 3 的高 16 位（对于 TIM2 和 TIM5）。 |
| 15:0 | CCR3[15:0] | 捕获/比较 3 的低 16 位<br>如果通道 CC3 配置为输出：<br>CCR3 为要装载到实际捕获/比较 3 寄存器的值（预装载值）。<br>如果没有通过 TIMx_CCMR 寄存器中的 OC3PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 3 寄存器中。<br>实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC3 输出上发出信号的值。<br>如果通道 CC3 配置为输入：<br>CCR3 为上一个输入捕获 3 事件 (IC3) 发生时的计数器值。 |

### 15.4.15. TIMx 捕获/比较比较器 3 (TIMx_CCR3)

偏移地址：0x3C

复位值：0x0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CCR3[31:16] (depending on times) | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CCR3[15:0] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | CCR3[31:16] | 捕获/比较 3 的高 16 位（对于 TIM2 和 TIM5）。 |
| 15:0 | CCR3[15:0] | 捕获/比较 3 的低 16 位<br>如果通道 CC3 配置为输出：<br>CCR3 为要装载到实际捕获/比较 3 寄存器的值（预装载值）。<br>如果没有通过 TIMx_CCMR 寄存器中的 OC3PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 3 寄存器中。<br>实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC3 输出上发出信号的值。<br>如果通道 CC3 配置为输入：<br>CCR3 为上一个输入捕获 3 事件 (IC3) 发生时的计数器值。 |

### 15.4.16. TIMx 捕获/比较比较器 4 (TIMx_CCR4)

偏移地址：0x40

复位值：0x0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CCR4[31:16] (depending on times) | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CCR4[15:0] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | CCR4[31:16] | 捕获/比较 4 的高 16 位（对于 TIM2 和 TIM5）。 |
| 15:0 | CCR4[15:0] | 捕获/比较 4 的低 16 位<br>1.如果 CC4 通道配置为输出（CC4S 位）：<br>CCR4 为要装载到实际捕获/比较 4 寄存器的值（预装载值）。<br>如果没有通过 TIMx_CCMR 寄存器中的 OC4PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 4 寄存器中。<br>实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC4 输出上发出信号的值。<br>2. 如果 CC4 通道配置为输入（TIMx_CCMR4 寄存器中的 CC4S 位）： CCR4 为上一个输入捕获 4 事件 (IC4) 发生时的计数器值。 |

### 15.4.17. TIMx DMA 控制寄存器(TIMx_DCR)

偏移地址：0x48

复位值：0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:13 | 保留 | 必须保持复位值。 |
| 12:8 | DBL[4:0] | DMA 连续传送长度<br>该 5 位定义了 DMA 在连续模式下的传送长度（当对 TIMx_DMAR 寄存器进行读或写时，定时器则进行一次连续传送）。<br>TIMx_DMAR 地址）<br>00000：1 次传输<br>00001：2 次传输<br>00010：3 次传输<br>...<br>10001：18 次传输 |
| 7:5 | 保留 | 必须保持复位值。 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 4:0 | DBA[4:0] | DMA 基址<br>该 5 位向量定义 DMA 传输的基址（通过 TIMx_DMAR 地址进行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。<br>示例：<br>00000：TIMx_CR1，<br>00001：TIMx_CR2，<br>00010：TIMx_SMCR，<br>...<br>示例：考虑以下传输：DBL = 7 次传输，DBA = TIMx_CR1。这种情况下将向/从自 TIMx_CR1 地址开始的 7 个寄存器传输数据。 |

### 15.4.18. TIMx 全传输 DMA 地址(TIMx_DMAR)

偏移地址：0x4C

复位值：0x0000

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 15:0 | DMAB[15:0] | DMA 连续传送寄存器<br>对 DMAR 寄存器执行读或写操作将访问位于如下地址的寄存器 (TIMx_CR1 地址) + (DBA + DMA 索引) x 4 其中 TIMx_CR1 地址为控制寄存器 1 的地址，DBA 为 TIMx_DCR 寄存器中配置的 DMA 基址，DMA 索引由 DMA 传输自动控制，其范围介于 0 到 DBL（TIMx_DCR 寄存器中配置的 DBL）之间。 |

**DMA 连续传送功能使用方法示例**

本例中，定时器 DMA 连续传送功能用于将 CCRx 寄存器（x = 2、3、4）的内容更新为通过 DMA 传输到 CCRx 寄存器中的多个半字。

具体操作步骤如下：

- 将相应的 DMA 通道配置如下：
  - DMA 通道外设地址为 DMAR 寄存器地址
  - DMA 通道存储器地址为包含要通过 DMA 传输到 CCRx 寄存器的数据的 RAM 缓冲区地址。
  - 要传输的数据量 = 3（参见下文注释）。
  - 禁止循环模式。
- 通过将 DBA 和 DBL 位域配置如下来配置 DCR 寄存器：
  - DBL = 3 次传输，DBA = 0xE。
- 使能 TIMx 更新 DMA 请求（DIER 寄存器中的 UDE 位置 1）。
- 使能 TIMx
- 使能 DMA 通道

*注意：*

*本例适用于每个 CCRx 寄存器只更新一次的情况。如果每个 CCRx 寄存器要更新两次，则要传输的数据量应为 6。下面以包含 data1、data2、data3、data4、data5 和 data6 的 RAM 缓冲区为例。数据将按照如下方式传输到 CCRx 寄存器：在第一个更新 DMA 请求期间，data1 传输到 CCR2，data2 传输到 CCR3，data3 传输到 CCR4；在第二个更新 DMA 请求期间，data4 传输到 CCR2，data5 传输到 CCR3，data6 传输到 CCR4。*

### 15.4.19. TIM2 选项寄存器 1(TIM2_OR)

偏移地址：0x50

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Reserved | | | ITR1_RMP | | Reserved | | | | | | | | | |
| | | | | rw | rw | | | | | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:12 | 保留 | 必须保持复位值。 |
| 11:10 | ITR1_RMP | 内部触发 1 重映射<br>由软件置 1 和清零。<br>00：TIM8_TRGOUT<br>01：PTP 触发输出连接到 TIM2_ITR1<br>10：保留<br>11：保留 |
| 9:0 | 保留 | 必须保持复位值。 |

### 15.4.20. TIM5 选项寄存器 1(TIM5_OR)

偏移地址：0x50

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | TI4_RMP | | Reserved | | | | | |
| | | | | | | | | rw | rw | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:8 | 保留 | 必须保持复位值。 |
| 7:6 | TI4_RMP | 定时器输入 4 重映射<br>由软件置 1 和清零。<br>00: TIM5 的通道 4 连接到 GPIO: 请参见数据手册中的复用功能映射表。<br>01：为进行校准，LSI 内部时钟连接到 TIM5_CH4 输入。<br>10：为进行校准，LSE 内部时钟连接到 TIM5_CH4 输入。 |
| 5:0 | 保留 | 必须保持复位值。 |

# 16. 通用定时器(TIM9~TIM14)

## 16.1. TIM9 到 TIM14 简介

TIM9 到 TIM14 通用定时器包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。
它们可用于多种用途，包括测量输入信号的脉冲宽度（输入捕获），或者生成输出波形（输出比较、 PWM）。
使用定时器预分频器和 RCC 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。
TIM9 到 TIM14 定时器彼此完全独立，不共享任何资源。

## 16.2. TIM9 到 TIM14 主要特性

### 16.2.1. TIM9/TIM12 主要特性

TIM9 到 TIM14 通用定时器具有以下特性：

- 16 位自动重载递增计数器（属于中等容量器件）；
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（即运行时修改），分频系数介于 1 和 65536 之间；
- 多达 2 个独立通道，可用于：
  - 输入捕获；
  - 输出比较；
  - PWM 生成（边沿对齐模式）；
  - 单脉冲模式输出；
- 使用外部信号控制定时器且可实现多个定时器互连的同步电路；
- 发生如下事件时生成中断：
  - 更新：计数器上溢、计数器初始化（通过软件或内部触发）；
  - 触发事件（计数器启动、停止、初始化或者由内部触发计数）；
  - 输入捕获；
  - 输出比较。

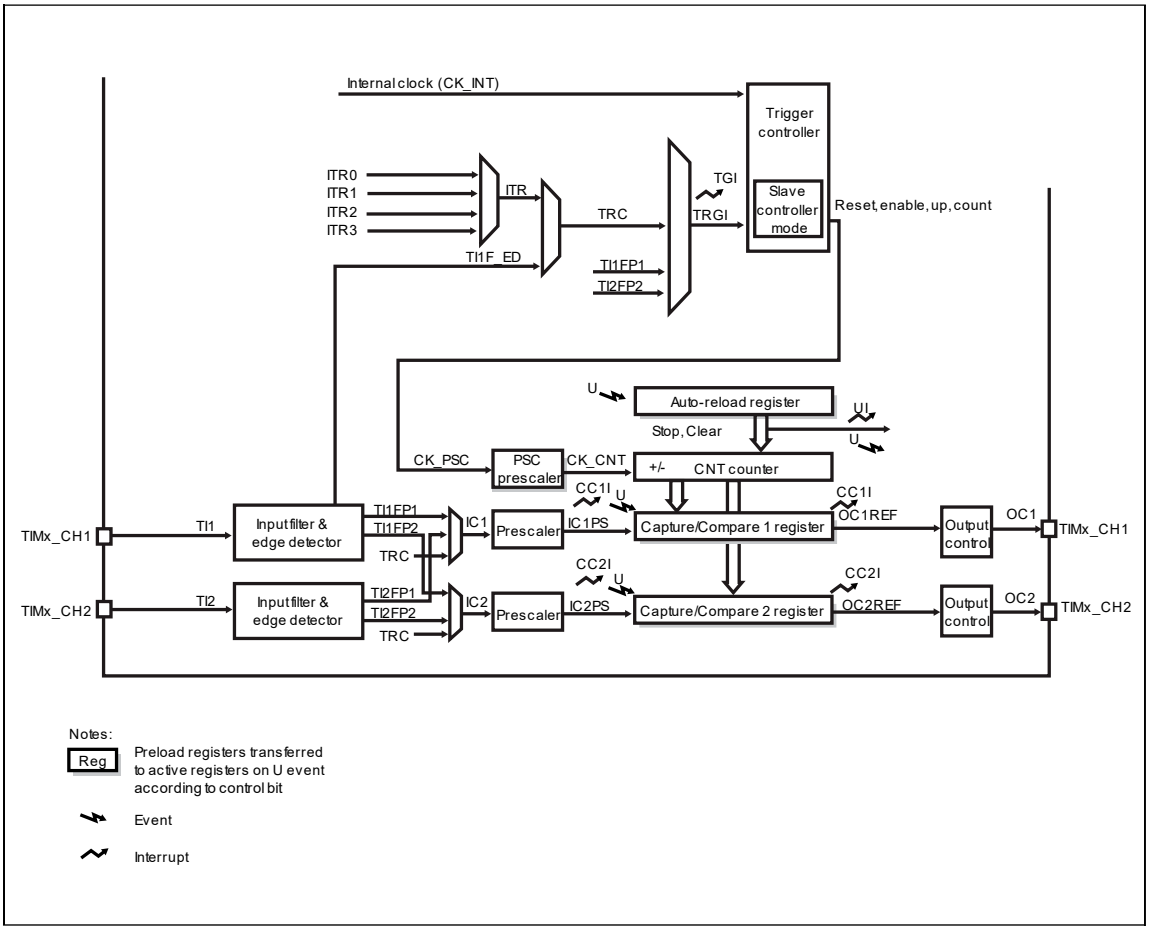



图 16-1 通用定时器框图(TIM9 和 TIM12)

### 16.2.2. TIM10/TIM11 和 TIM13/TIM14 主要特性

通用定时器 TIM10/TIM11 和 TIM13/TIM14 具有以下特性:

- 16 位自动重载递增计数器。
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（即运行时修改），分频系数介于 1 和 65536 之间。
- 独立通道，可用于:
  - 输入捕获。
  - 输出比较。
  - PWM 生成（边沿对齐模式）。
  - 单脉冲模式输出。
- 发生如下事件时生成中断:
  - 更新：计数器上溢、计数器初始化（通过软件）。
  - 输入捕获。
  - 输出比较。

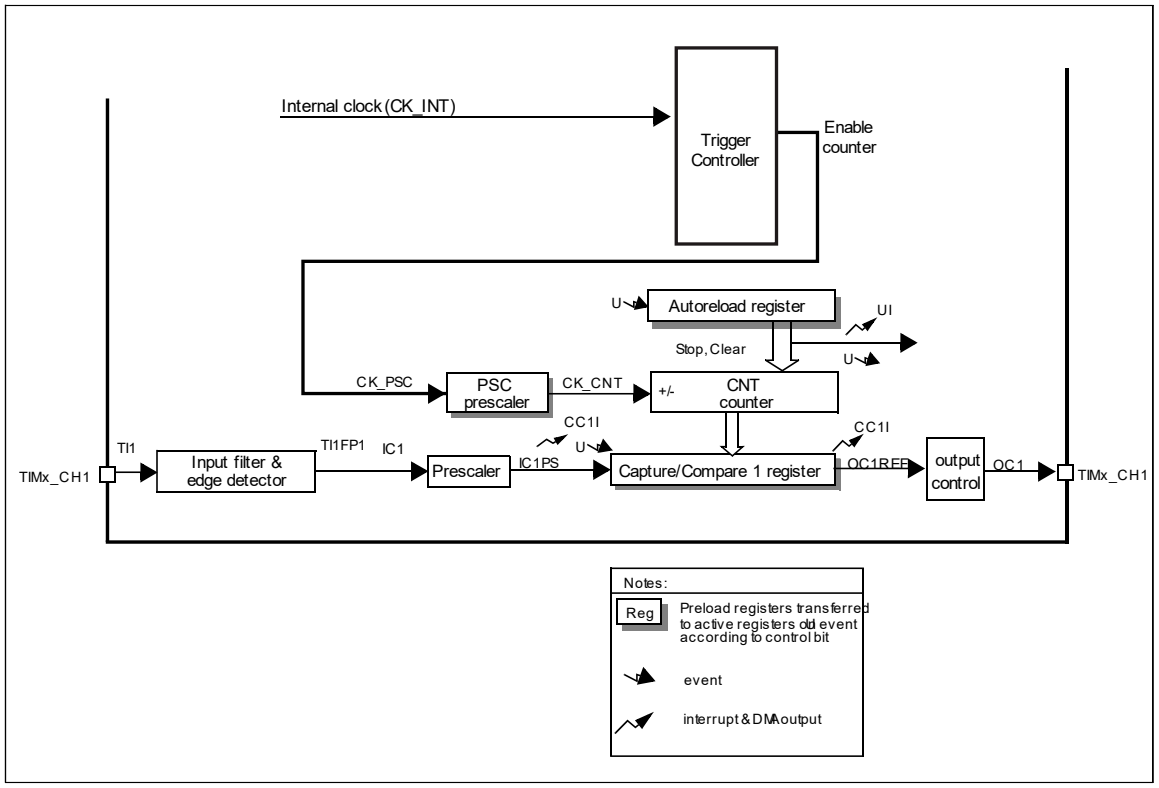



图 16-2 通用定时器框图 (TIM10/11/13/14)

# 16.3. TIM9 到 TIM14 功能说明

## 16.3.1. 时基单元

定时器的主要模块由一个 16 位计数器及其相关的自动重载寄存器组成。计数器采用递增方式计数。

计数器的时钟可通过预分频器进行分频。

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

时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以直接传送到影子寄存器，也可以在每次发生更新事件 (UEV)时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位 (CEN)置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。

*注意，计数器将在 TIMx_CR1 寄存器的 CEN 位置 1 时刻的一个时钟周期后开始计数。*

**预分频器说明**

预分频器可对计数器时钟频率进行分频，分频系数介于 1 和 65536 之间。该预分频器基于 TIMx_PSC 寄

存器中的 16 位寄存器所控制的 16 位计数器。由于该控制寄存器具有缓冲功能，因此预分频器可实现实时更改。而新的预分频比将在下一更新事件发生时被采用。

下图以一些示例说明在预分频比实时变化时计数器的行为：

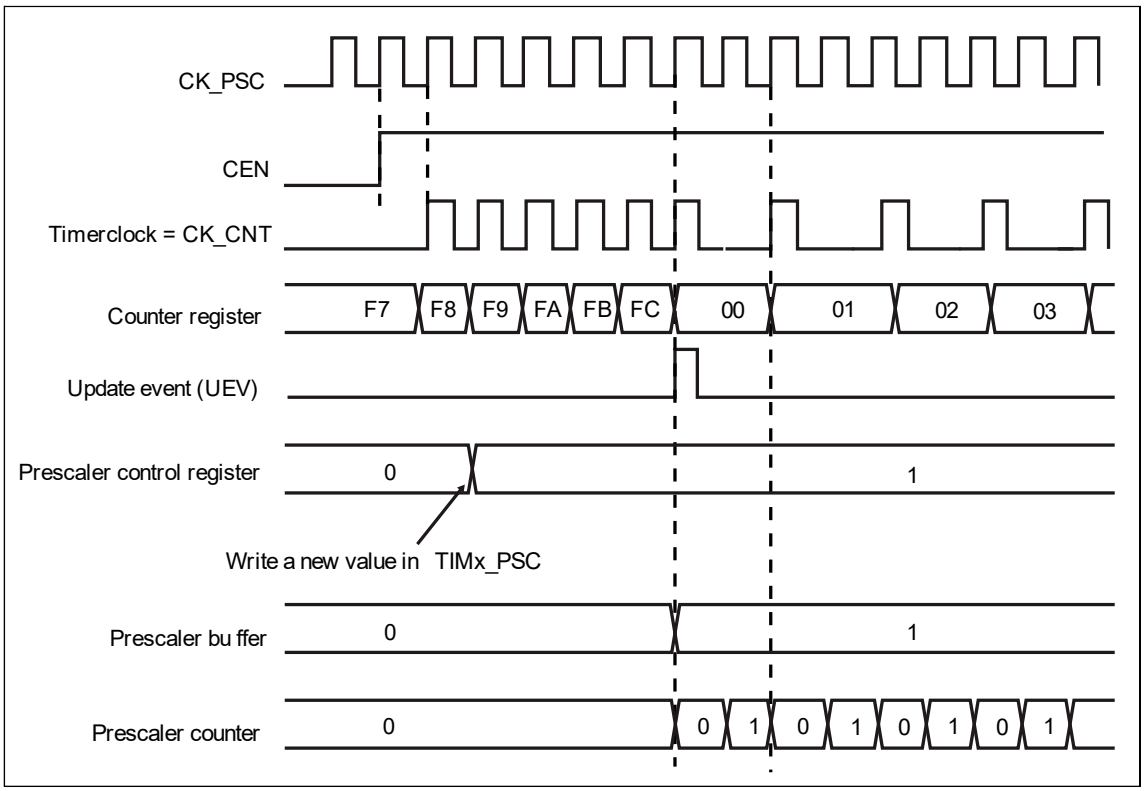



图 16-3 预分频器分频由 1 变为 2 时的计数器时序图

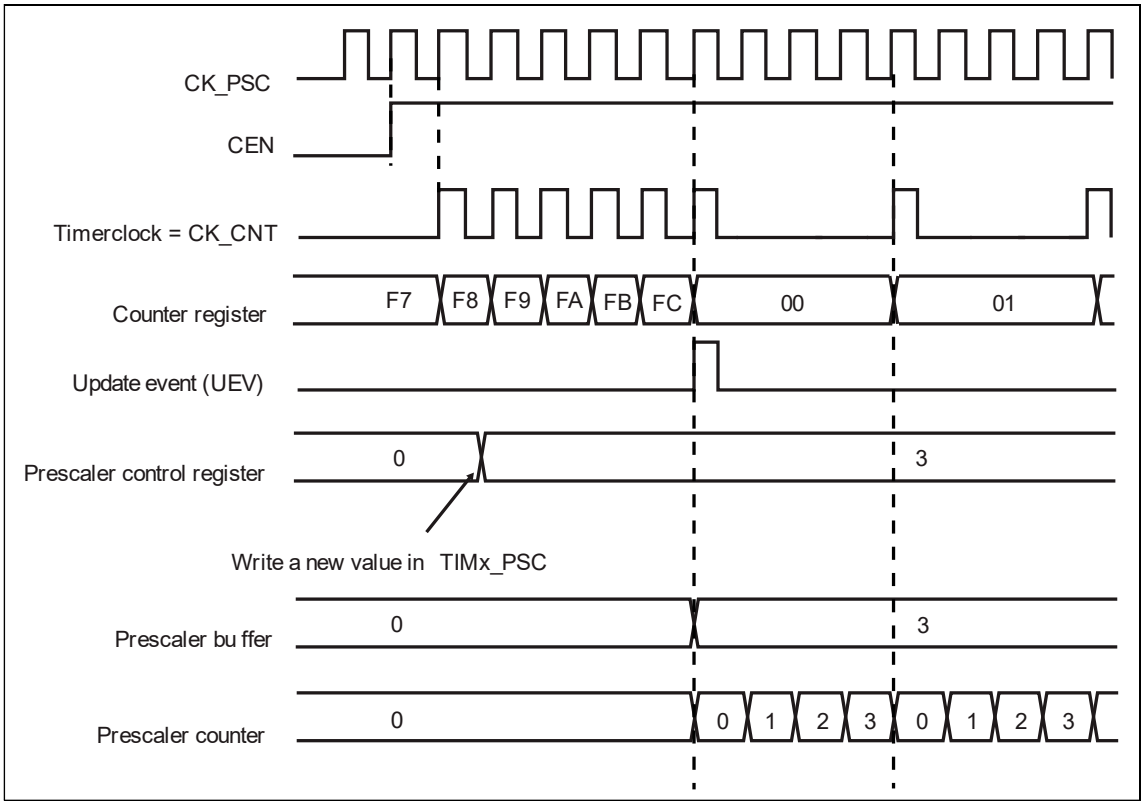



图 16-4 预分频器分频由 1 变为 4 时的计数器时序图

## 16.3.2. 计数器模式

**递增计数模式**

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

将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用 TIM9 和 TIM12 上的从模式控制器）也会生成更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

- 自动重载影子寄存器将以预装载值 (TIMx_ARR) 进行更新。
- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。

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

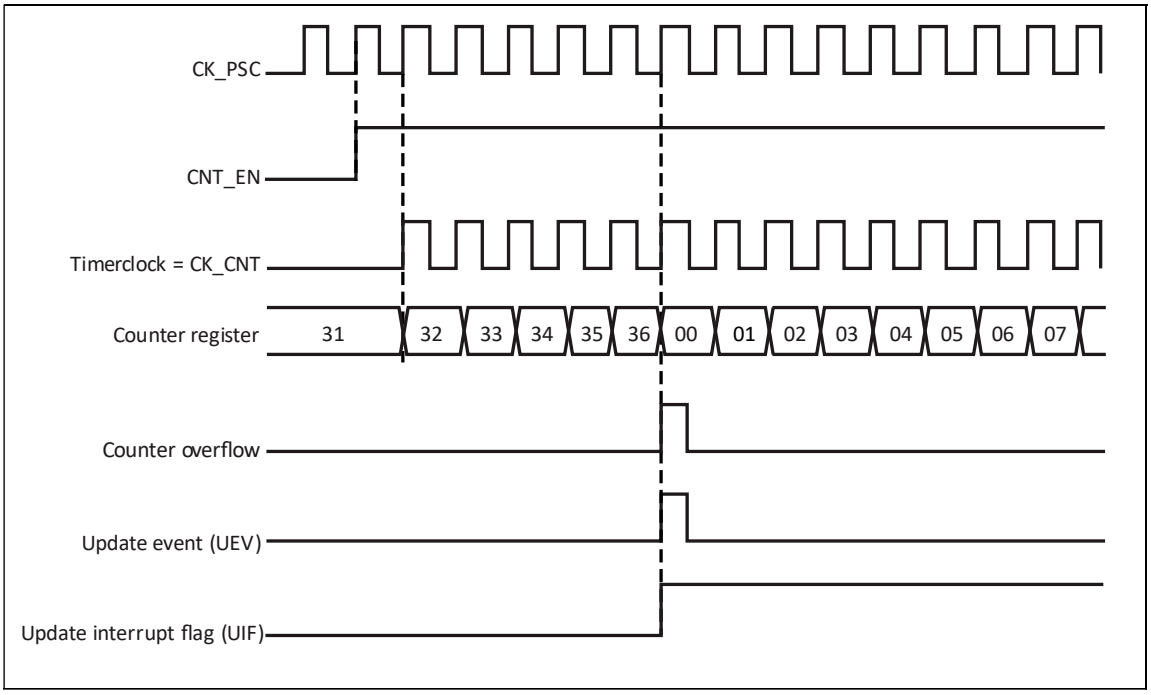



图 16-5 计数器时序图，内部时钟 1 分频

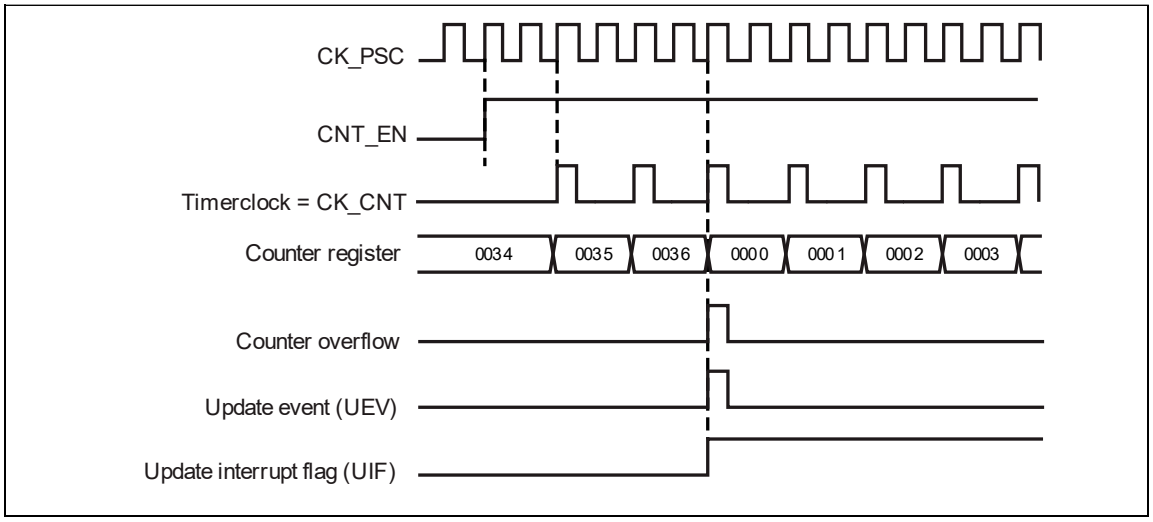



图 16-6 计数器时序图，内部时钟 2 分频

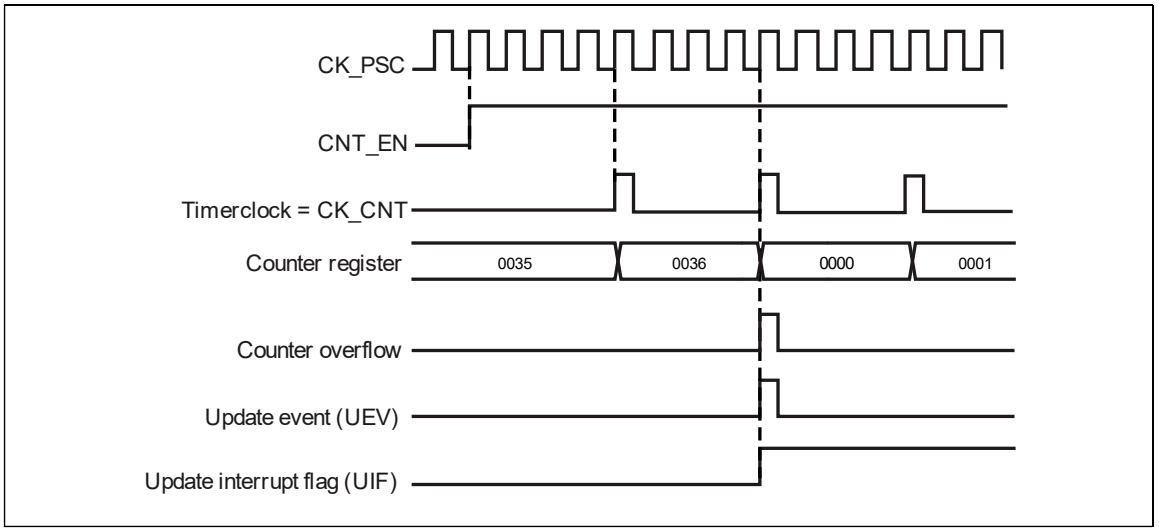



图 16-7 计数器时序图，内部时钟 4 分频

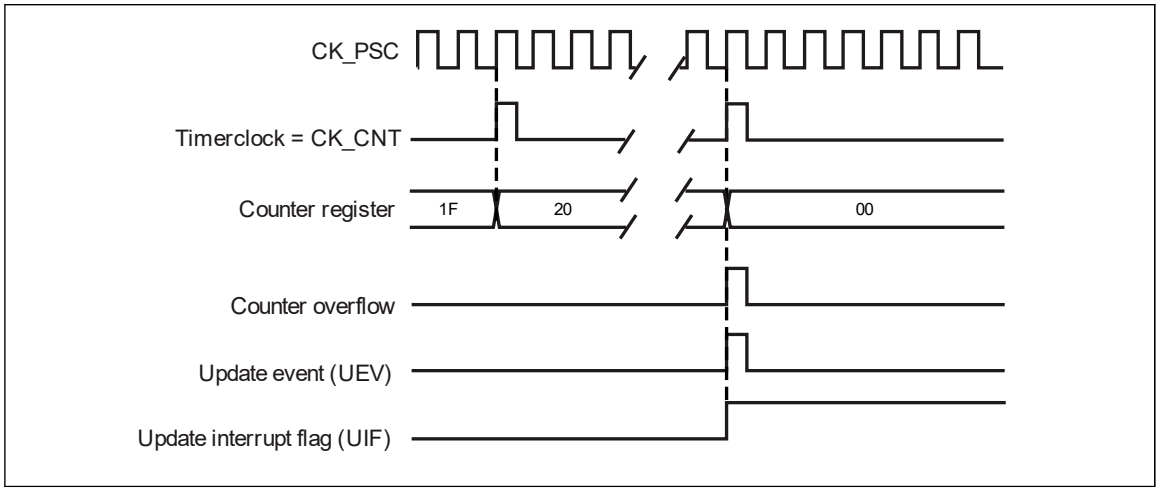



图 16-8 计数器时序图，内部时钟 N 分频

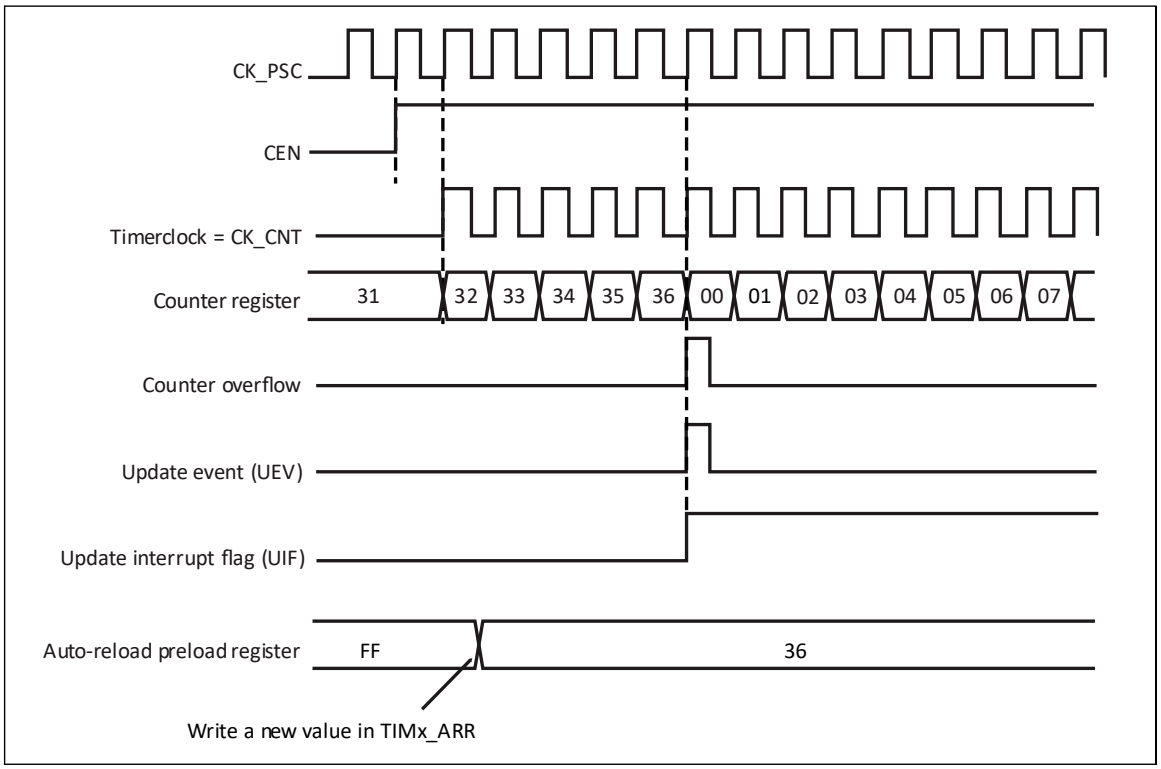



图 16-9 计数器时序图， ARPE=0 时更新事件(TIMx_ARR 未预装载)

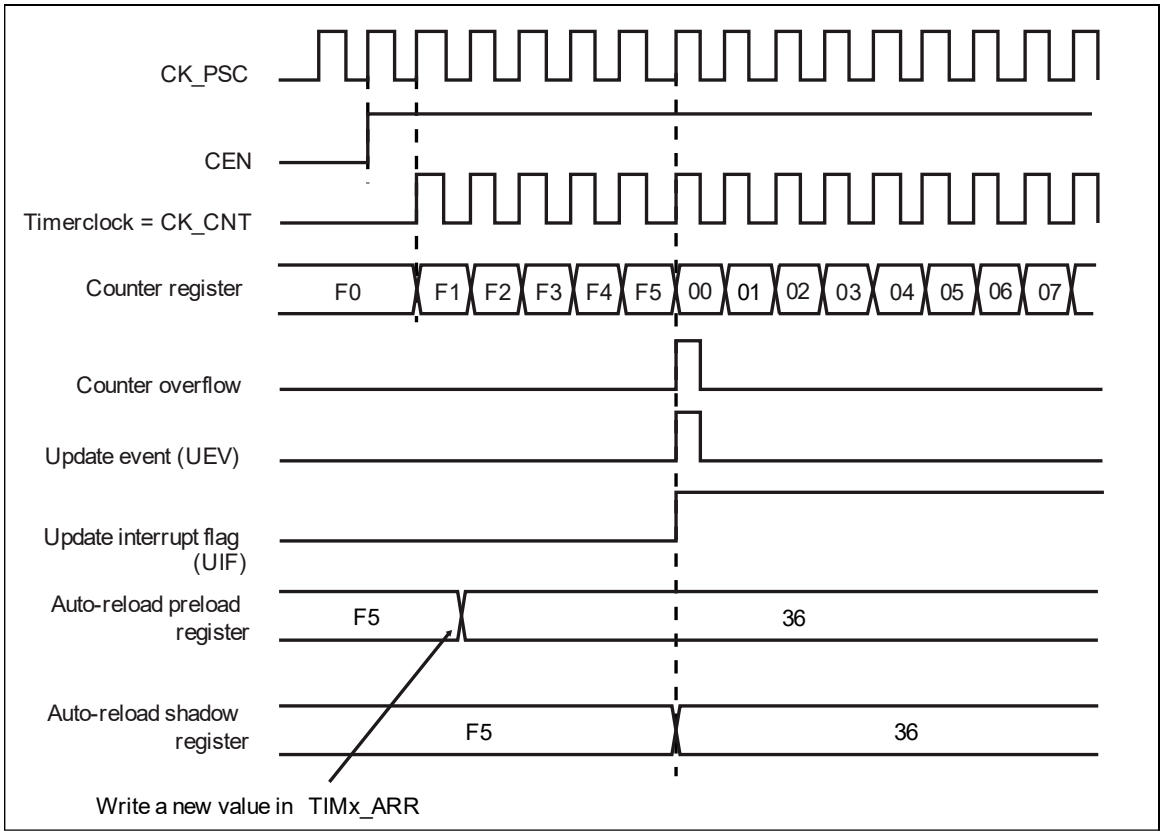



图 16-10 计数器时序图， ARPE=1 时更新事件(TIMx_ARR 已预装载)

### 16.3.3. 时钟选择

计数器时钟可由下列时钟源提供:

- 内部时钟 (CK_INT)。
- 外部时钟模式 1（针对 TIM9 和 TIM12）：外部输入引脚 (TIx)。
- 内部触发输入 (ITRx)（针对 TIM9 和 TIM12）：连接来自其它计数器的触发输出。

#### 内部时钟源 (CK_INT)

内部时钟源是 TIM10/TIM11 和 TIM13/TIM14 的默认时钟源。

对于 TIM9 和 TIM12，在禁止从模式控制器后选择内部时钟源（SMS=“000”）。然后，将 TIMx_CR1 寄存器中的 CEN 位和 TIMx_EGR 寄存器中的 UG 位用作控制位，并且只能通过软件对其进行更改（保持清零状态的 UG 除外）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 CK_INT 提供。

下图显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

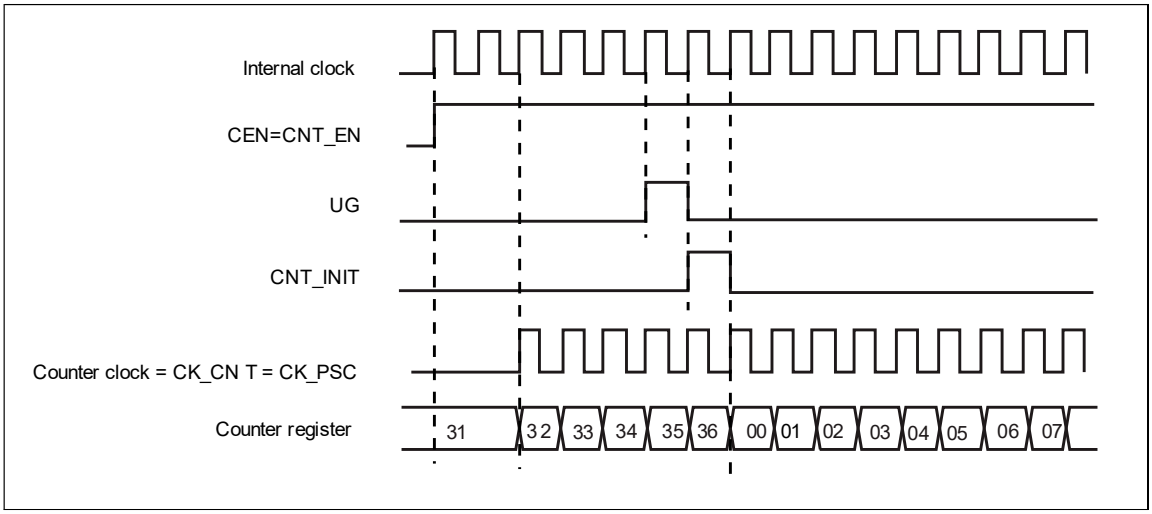



图 16-11 正常模式下的控制电路，内部时钟 1 分频

#### 外部时钟源模式 1（TIM9 和 TIM12）

当 TIMx_SMCR 寄存器中的 SMS=“111”时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

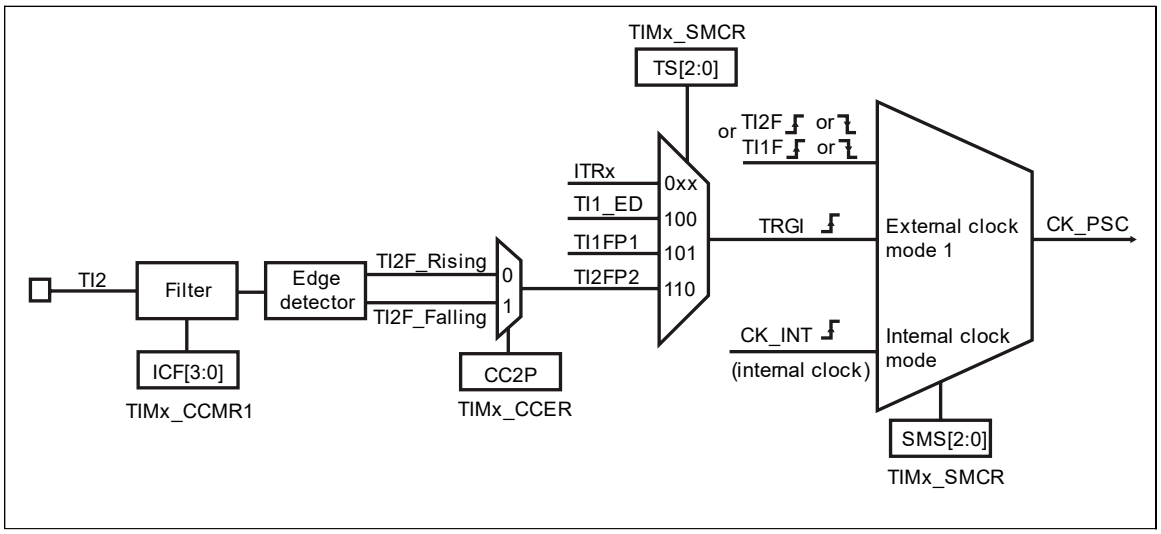



图 16-12 TI2 外部时钟连接示例

例如，要使递增计数器在 TI2 输入出现上升沿时计数，请执行以下步骤：

- 通过在 TIMx_CCMR1 寄存器中写入 CC2S =“01”来配置通道 2，使其能够检测 TI2 输入的上升沿。
- 通过在 TIMx_CCMR1 寄存器中写入 IC2F[3:0] 位来配置输入滤波时间（如果不需要任何滤波器，请保持 IC2F=“0000”）。
- 通过在 TIMx_CCER 寄存器中写入 CC2P=“0”和 CC2NP=“0”来选择上升沿极性。
- 通过在 TIMx_SMCR 寄存器中写入 SMS=“111”，使定时器在外部时钟模式 1 下工作。
- 通过在 TIMx_SMCR 寄存器中写入 TS=“110”来选择 TI2 作为触发输入源。
- 通过在 TIMx_CR1 寄存器中写入 CEN=“1”来使能计数器。

*注意： 由于捕获预分频器不用于触发操作，因此无需对其进行配置。*

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。

TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入的重新同步电路引起的。

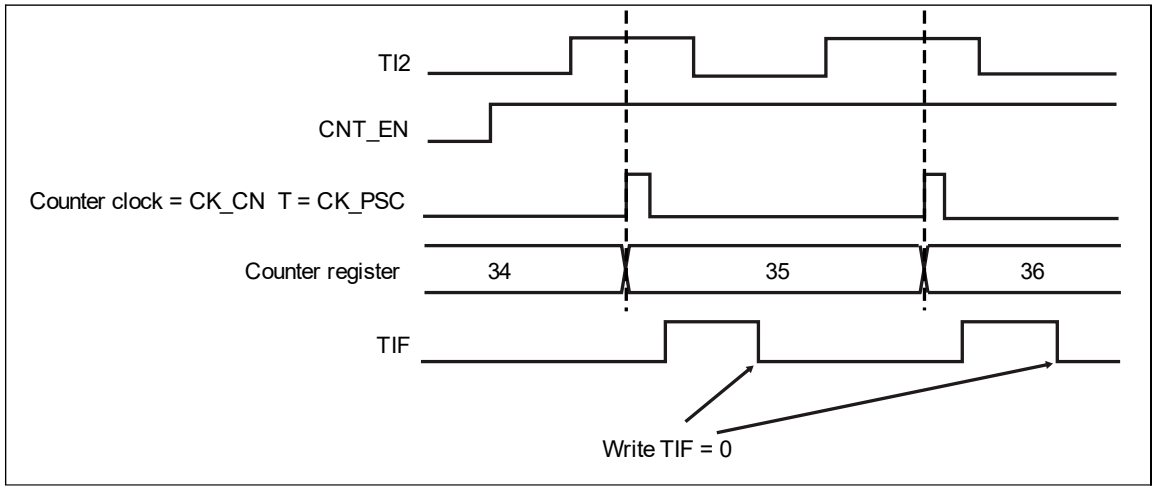



图 16-13 外部时钟模式 1 下的控制电路

### 16.3.4. 捕获/比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入阶段（数字滤波、多路复用和预分频器）和一个输出阶段（比较器和输出控制）构建而成。

下图对捕获/比较通道做了简要说明。

输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 TIxF。然后，带有极性选择功能的边沿检测器生成一个信号 (TIxFPx)， 该信号可用作从模式控制器的触发输入，也可用作捕获命令。该信号先进行预分频 (ICxPS)，而后再进入捕获寄存器。

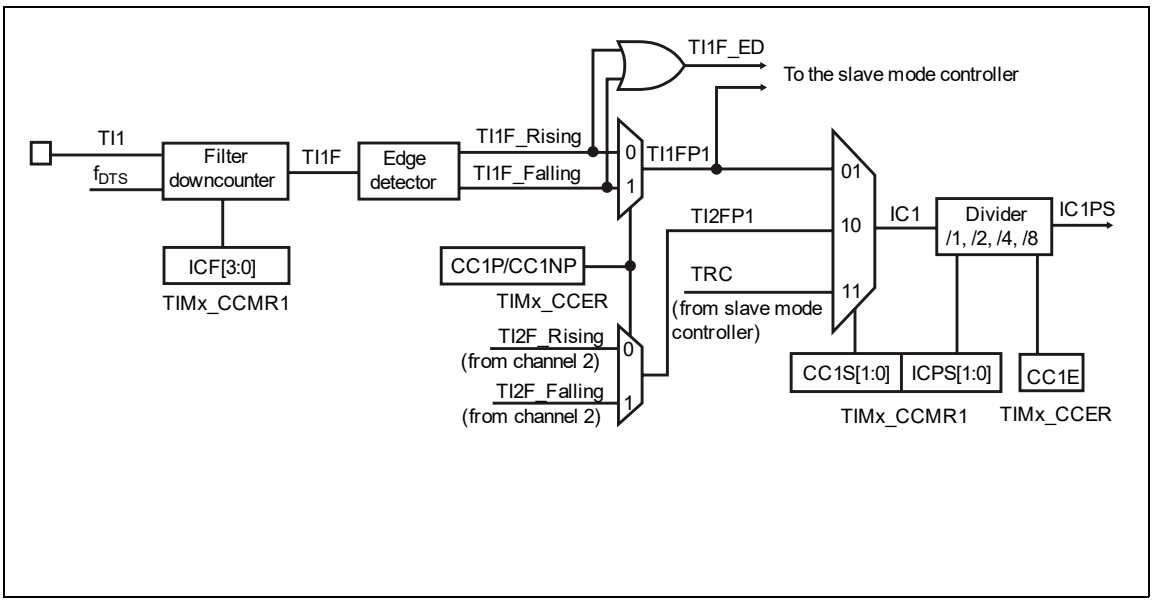



图 16-14 捕获/比较通道(例如：通道 1 输入阶段)

输出阶段生成一个中间波形作为基准： OCxRef（高电平有效）。 链的末端决定最终输出信号的极性。

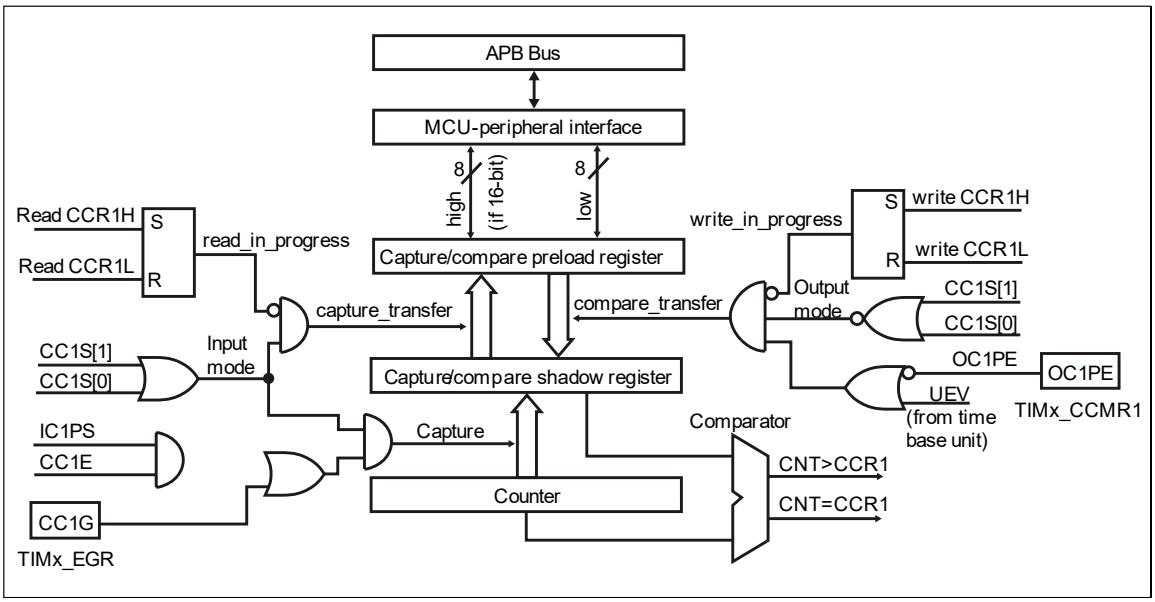



图 16-15 捕获/比较通道 1 主电路

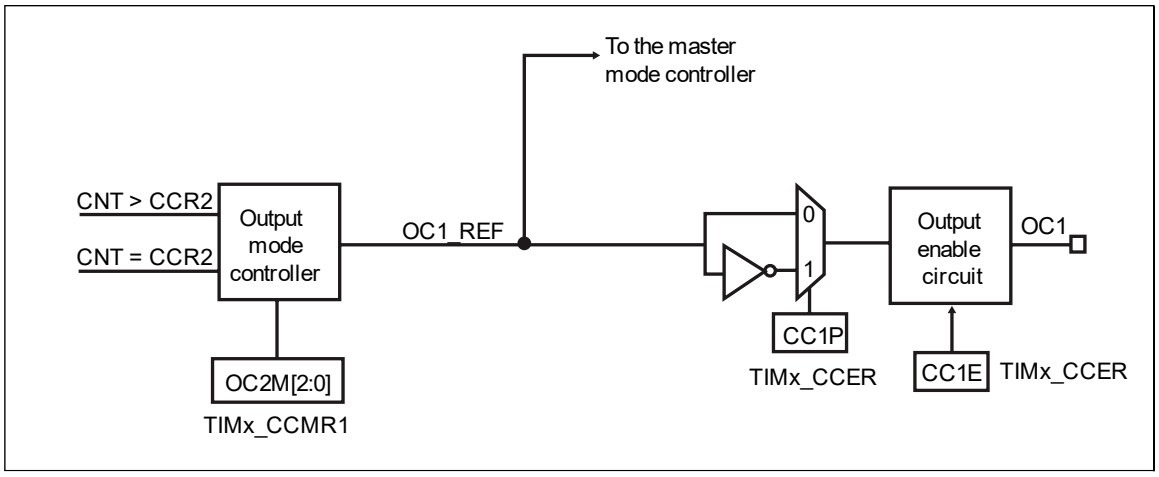



图 16-16 捕获/比较通道的输出阶段(通道 1)

捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。始终可通过读写操作访问预装载寄存器。

在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容复制到预装载寄存器中。

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

### 16.3.5. 输入捕获模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获/比较寄存器(TIMx_CCRx)来锁存计数器的值。发生捕获事件时，会将相应的 CCXIF 标志（TIMx_SR 寄存器）置 1，并可发送中断或 DMA 请求（如果已使能）。如果发生捕获事件时 CCxIF 标志已处于高位，则会将重复捕获标志 CCxOF（TIMx_SR 寄存器）置 1。可通过软件向 CCxIF 写入 0 来给 CCxIF 清零，或读取存储在 TIMx_CCRx 寄存器中的已捕获数据。向 CCxOF 写入“0”后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIMx_CCR1 中。具体操作步骤如下：

- 选择有效输入： TIMx_CCR1 必须连接到 TI1 输入，因此向 TIMx_CCMR1 寄存器中的 CC1S 位写入“01”。只要 CC1S 不等于“00”，就会将通道配置为输入模式，并且 TIMx_CCR1 寄存器将处于只读状态。
- 根据连接到定时器的信号，对所需的输入滤波时间进行编程（如果输入为 TIx 输入之一，则对 TIMx_CCMRx 寄存器中的 ICxF 位进行编程）。信号变化时，输入信号最多在 5 个内部时钟周期内发生抖动。因此，我们必须将滤波时间设置为大于 5 个内部时钟周期。在检测到 8 个具有新电平的连续采样（以 fDTS 频率采样）后，可以确认 TI1 上的跳变沿。然后向 TIMx_CCMR1 寄存器中的 IC1F 位写入“0011”。
- 通过在 TIMx_CCER 寄存器中将 CC1P 位和 CC1NP 位编程为“00”，选择 TI1 上的有效转换边沿（本例中为上升沿）。
- 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 TIMx_CCMR1 寄存器中的 IC1PS 位写入“00”）。
- 通过将 TIMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
- 如果需要，可通过将 TIMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求。

发生输入捕获时：

- 发生有效跳变沿时，TIMx_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。

- 根据 CC1IE 位生成中断。

要处理重复捕获，建议在读出捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出标志之后与读取数据之前可能出现的重复捕获信息。

*注意： 通过软件将 TIMx_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断请求。*

### 16.3.6. PWM 输入模式(仅适用于 TIM9/12)

此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处:

- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这两个 ICx 信号在边沿处有效，但极性相反。
- 选择两个 TIxFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。

例如，可通过以下步骤对应用于 TI1 的 PWM 的周期（位于 TIMx_CCR1 寄存器中）和占空比（位于 TIMx_CCR2 寄存器中）进行测量（取决于 CK_INT 频率和预分频器的值）:

- 选择 TIMx_CCR1 的有效输入：向 TIMx_CCMR1 寄存器中的 CC1S 位写入“01”（选择 TI1）。
- 选择 TI1FP1 的有效极性（同时用于 TIMx_CCR1 中的捕获和计数器清零）：将 CC1P 位和 CC1NP 位编程为“00”（上升沿有效）。
- 选择 TIMx_CCR2 的有效输入：向 TIMx_CCMR1 寄存器中的 CC2S 位写入“10”（选择 TI1）。
- 选择 TI1FP2 的有效极性（用于 TIMx_CCR2 中的捕获）：将 CC2P 位和 CC2NP 位编程为“11”（下降沿有效）。
- 选择有效触发输入：向 TIMx_SMCR 寄存器中的 TS 位写入“101”（选择 TI1FP1）。
- 将从模式控制器配置为复位模式：向 TIMx_SMCR 寄存器中的 SMS 位写入“100”。
- 使能捕获：向 TIMx_CCER 寄存器中的 CC1E 位和 CC2E 位写入“1”。

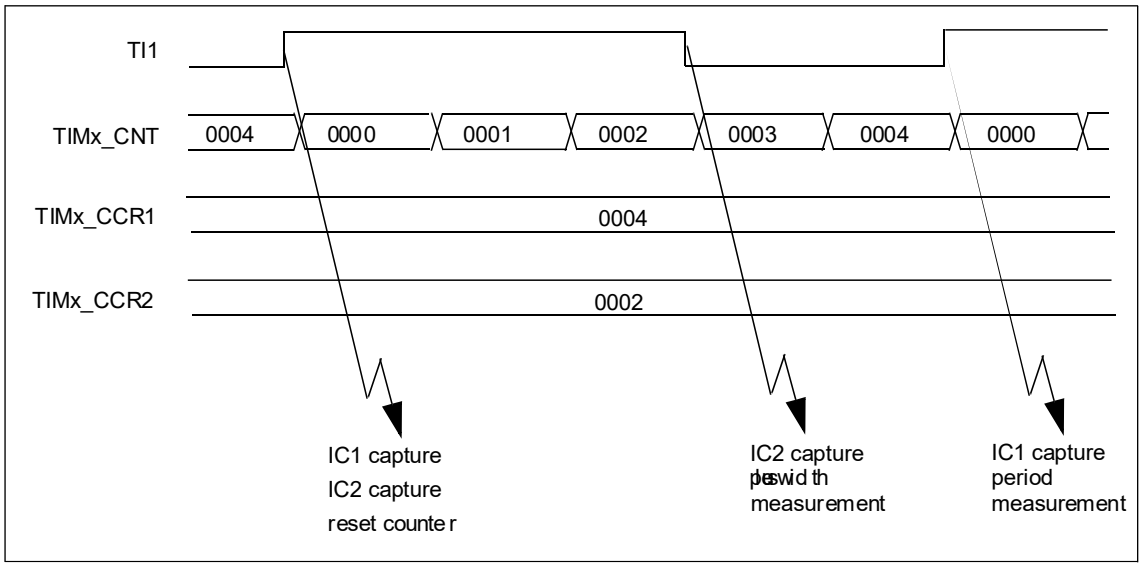



图 16-17 PWM 输入模式时序

1. *PWM 输入模式只能与 TIMx_CH1/TIMx_CH2 信号配合使用，因为只有 TI1FP1 和 TI2FP2 与从模式控制器相连。*

### 16.3.7. 强制输出模式

在输出模式（TIMx_CCMRx 寄存器中的 CCxS 位 =“00”）下，可直接由软件将每个输出比较信号（OCxREF

和 OCx）强制设置为有效电平或无效电平，而无需考虑输出比较寄存器和计数器之间的任何比较结果。

要将输出比较信号 (OCXREF/OCx) 强制设置为有效电平，只需向相应 TIMx_CCMRx 寄存器中的 OCxM 位写入“101”。OCXREF 进而强制设置为高电平（OCxREF 始终为高电平有效），同时 OCx 获取 CCxP 极性位的相反值。

例如： CCxP=“0”（OCx 高电平有效） => 将 OCx 强制设置为高电平。

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入“100”，可将 OCxREF 信号强制设置为低电平。

无论如何，TIMx_CCRx 影子寄存器与计数器之间的比较仍会执行，而且允许将标志置 1。因此可相应发送中断请求。下面的输出比较模式一节对此进行了介绍。

### 16.3.8. 输出比较模式

此功能用于控制输出波形，或指示已经过某一时间段。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

- 将为相应的输出引脚分配一个可编程值，该值由输出比较模式（TIMx_CCMRx 寄存器中的 OCxM 位）和输出极性（TIMx_CCER 寄存器中的 CCxP 位）定义。匹配时，输出引脚既可保持其电平（OCXM=“000”），也可设置为有效电平（OCxM=“001”）、无效电平（OCxM=“010”）或进行翻转（OCxM=“011”）。
- 将中断状态寄存器中的标志置 1（TIMx_SR 寄存器中的 CCxIF 位）。
- 如果相应中断使能位（TIMx_DIER 寄存器中的 CCXIE 位）置 1，将生成中断。

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

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

步骤：

- 选择计数器时钟（内部、外部、预分频器）。
- 在 TIMx_ARR 和 TIMx_CCRx 寄存器中写入所需数据。
- 如果要生成中断请求，则需将 CCxIE 位置 1。
- 选择输出模式。例如：
  - 当 CNT 与 CCRx 匹配时，写入 OCxM =“011”以翻转 OCx 输出引脚；
  - 写入 OCxPE =“0”以禁止预装载寄存器；
  - 写入 CCxP =“0”以选择高电平有效极性；
  - 写入 CCxE =“1”以使能输出。
- 通过将 TIMx_CR1 寄存器中的 CEN 位置 1 来使能计数器。

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

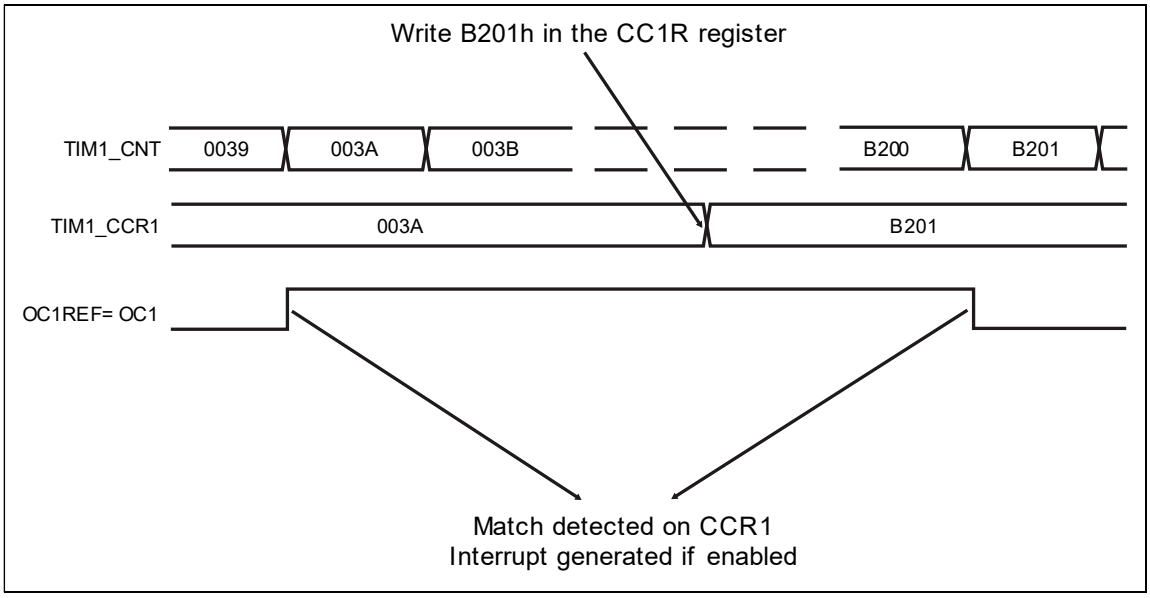



图 16-18 输出比较模式，翻转 OC1

### 16.3.9. PWM 模式

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

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入 110（PWM 模式 1）或 111（PWM 模式 2），可以独立选择各通道 （每个 OCx 输出对应一个 PWM）的 PWM 模式。必须通过将 TIMx_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，最后通过将 TIMx_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器（在递增计数或中心对齐模式下）。

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

OCx 极性可使用 TIMx_CCER 寄存器的 CCxP 位来编程。既可以设为高电平有效，也可以设为低电平有效。OCx 输出通过将 TIMx_CCER 寄存器中的 CCxE 位置 1 来使能。有关详细信息，请参见 TIMx_CCERx 寄存器说明。

在 PWM 模式(1 或 2)下, TIMx_CNT 总是与 TIMx_CCRx 进行比较, 以确定是 TIMx_CNT ≤ TIMx_CCRx。

因为计数器采用递增方式计数，所以定时器能够在边沿对齐模式下生成 PWM。

**PWM 边沿对齐模式**

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

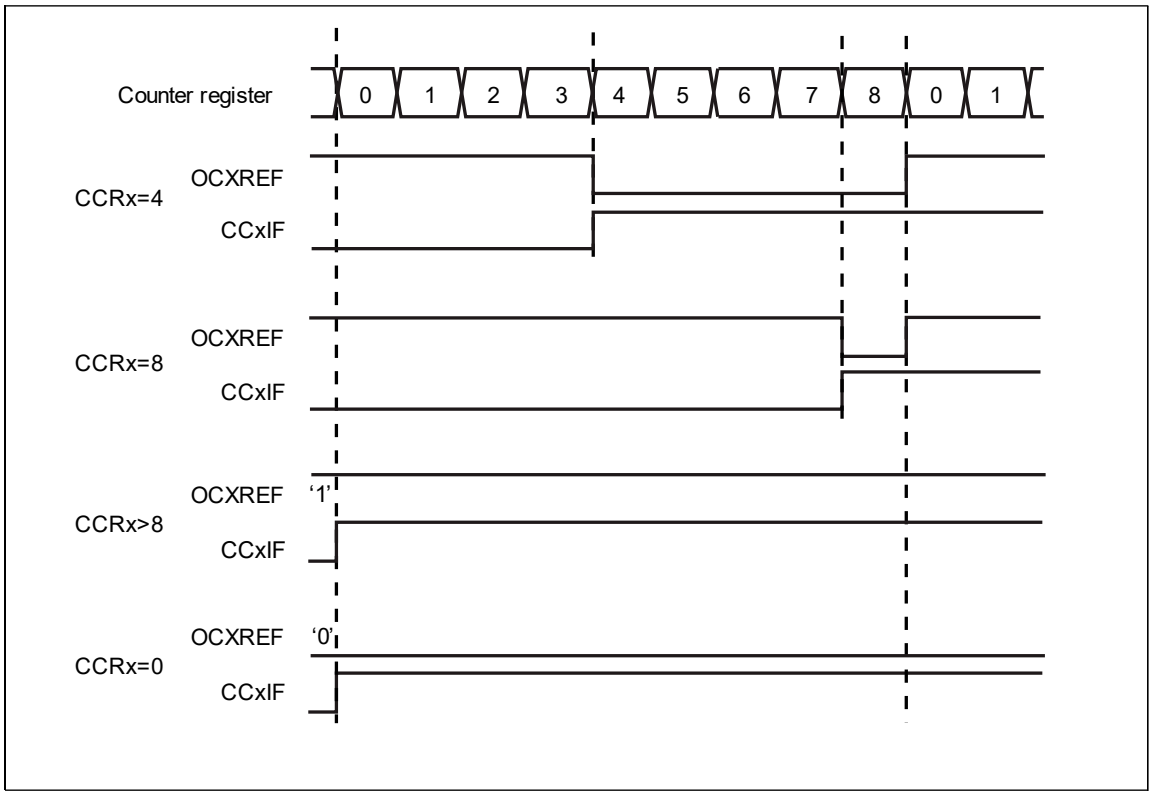



图 16-19 比边沿对齐模式的 PWM 波形 (ARR=8)

### 16.3.10. 单脉冲模式

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

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。将 TIMx_CR1 寄存器中的 OPM 位置 1，即可选择单脉冲模式。这样发生下一更新事件 UEV 时，计数器将自动停止。

只有当比较值与计数器初始值不同时，才能正确产生一个脉冲。启动前（定时器等待触发时），必须进行如下配置：CNT < CCRx ≤ ARR（特别注意，0 < CCRx）

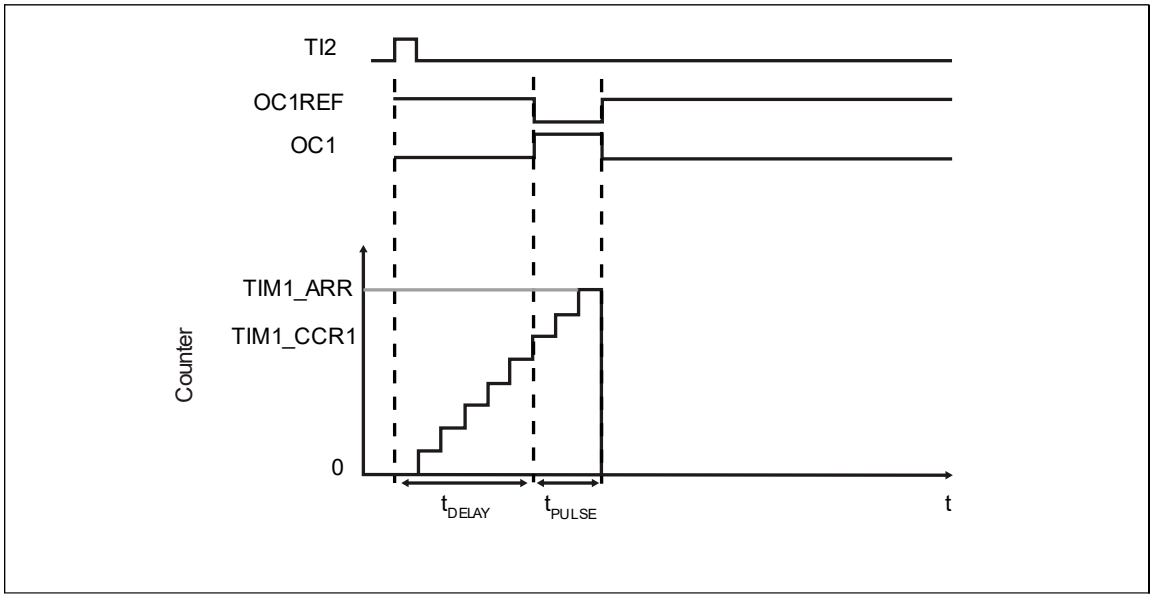



图 16-20 单脉冲模式示例

例如，用户希望达到这样的效果：在 TI2 输入引脚检测到正沿时，经过 $t_{DELAY}$ 的延迟，在 OC1 上产生一个长度为 $t_{PULSE}$ 的正脉冲。

使用 TI2FP2 作为触发 1：

- 在 TIMx_CCMR1 寄存器中写入 CC2S =“01”，以将 TI2FP2 映射到 TI2。
- 在 TIMx_CCER 寄存器中写入 CC2P =“0”和 CC2NP=“0”，使 TI2FP2 能够检测上升沿
- 在 TIMx_SMCR 寄存器中写入 TS =“110”，以将 TI2FP2 配置为从模式控制器的触发(TRGI)。
- 在 TIMx_SMCR 寄存器中写入 SMS =“110”（触发模式），以使用 TI2FP2 启动计数器。

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

- $t_{DELAY}$ 由写入 TIMx_CCR1 寄存器的值定义。
- $t_{PULSE}$ 由自动重载值与比较值 (TIMx_ARR - TIMx_CCR1) 之差来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从“0”变为“1”，在计数器达到自动重载值时由“1”变为“0”。为此，应在 TIMx_CCMR1 寄存器中写入 OC1M=“111”，以使能 PWM 模式 2。如果需要，可选择在 TIMx_CCMR1 寄存器的 OC1PE 和 TIMx_CR1 寄存器的 ARPE 中写入“1”，以使能预装载寄存器。这种情况下，必须在 TIMx_CCR1 寄存器中写入比较值并在 TIMx_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI2 上的外部触发事件。本例中，CC1P 的值为“0”。

由于仅需要 1 个脉冲（单脉冲模式），因此应向 TIMx_CR1 寄存器的 OPM 位写入“1”，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。TIMx_CR1 寄存器中的 OPM 位置“0”时，即选择重复模式。

**特例： OCx 快速使能**

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

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

### 16.3.11. TIM9/12 外部触发同步

TIM9/12 定时器可以下列模式与外部触发实现同步：复位模式、门控模式和触发模式。

**从模式：复位模式**

当触发输入信号产生变化时，计数器及其预分频器可重新初始化。此外，如果 TIMx_CR1 寄存器中的 URS 位处于低电平，则会生成更新事件 UEV。然后，所有预装载寄存器（TIMx_ARR 和 TIMx_CCRx）都将更新。

在以下示例中，TI1 输入上出现上升沿时，递增计数器清零：

- 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波时间（本例中不需要任何滤波器，因此保持 IC1F=“0000”）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S =“01”。将 TIMx_CCER 寄存器中的 CC1P 和 CC1NP 编程为“00”，以验证极性（仅检测上升沿）
- 在 TIMx_SMCR 寄存器中写入 SMS=“100”，将定时器配置为复位模式。在 TIMx_SMCR 寄存器中写入 TS=“101”，选择 TI1 作为输入源。
- 通过在 TIMx_CR1 寄存器中写入 CEN=“1”来启动动计数器。
- 计数器使用内部时钟计数，然后正常运转，直到出现 TI1 上升沿。当 TI1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时，触发标志（TIMx_SR 寄存器中的 TIF 位）置 1，使能中断，还可发送中断请求（取决于 TIMx_DIER 寄存器中的 TIE 位）。

下图显示了自动重载寄存器 TIMx_ARR=0x36 时的相关行为。TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

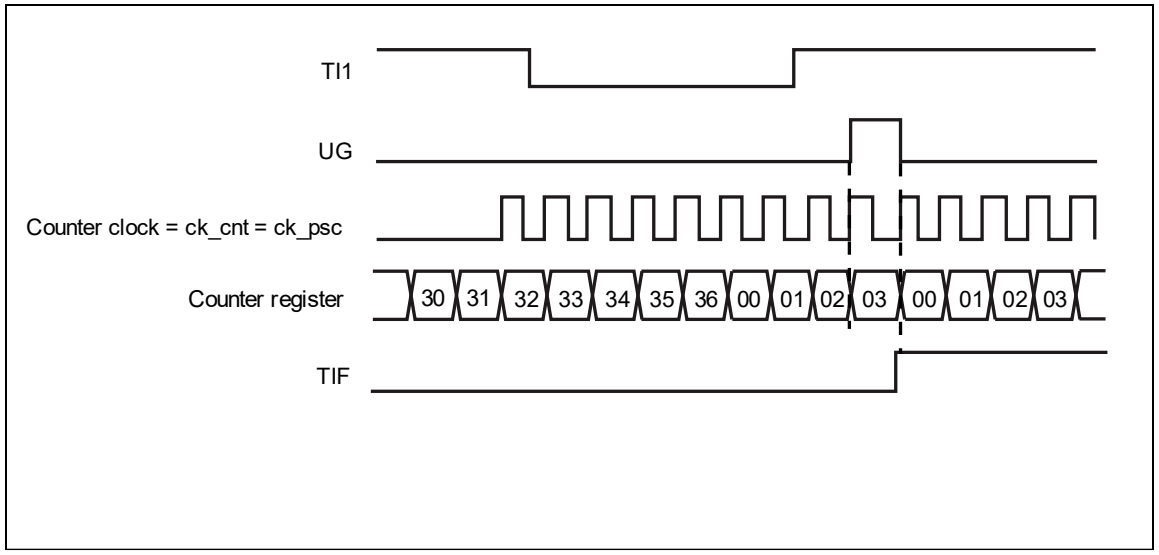



图 16-21 复位模式下的控制电路

**从模式：门控模式**

输入信号的电平可用来使能计数器。

在以下示例中，递增计数器仅在 TI1 输入为低电平时计数：

- 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波时间（本例中不需要任何滤波器，因此保持 IC1F=“0000”）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S =“01”。将 TIMx_CCER 寄存器中的 CC1P 和 CC1NP 分

别编程为“1”和“0”，以确定极性（仅检测低电平）。

- 在 TIMx_SMCR 寄存器中写入 SMS=“101”，将定时器配置为门控模式。在 TIMx_SMCR 寄存器中写入 TS=“101”，选择 TI1 作为输入源。
- 在 TIMx_CR1 寄存器中写入 CEN=“ 1”，使能计数器（在门控模式下，如果 CEN=“0”，则不论触发输入电平如何，计数器都不启动）。

只要 TI1 为低电平，计数器就开始根据内部时钟计数，直到 TI1 变为高电平时停止计数。计数器启动或停止时，TIMx_SR 寄存器中的 TIF 标志都会置 1。

TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

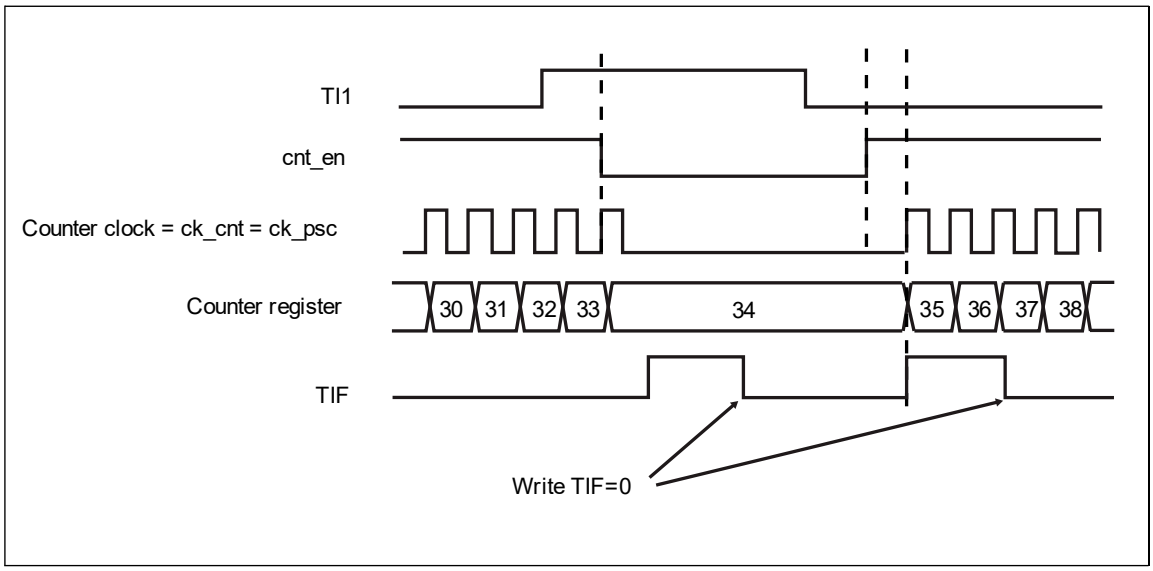



图 16-22 门控模式下的控制电路

**从模式：触发模式**

所选输入上发生某一事件时可以启动计数器。

在以下示例中，TI2 输入上出现上升沿时，递增计数器启动：

- 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波时间（本例中不需要任何滤波器，因此保持 IC2F=“0000”）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC2S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC2S=“01”。将 TIMx_CCER 寄存器中的 CC2P 和 CC2NP 分别编程为“1”和“0”，以确定极性（仅检测低电平）。
- 在 TIMx_SMCR 寄存器中写入 SMS=“110”，将定时器配置为触发模式。在 TIMx_SMCR 寄存器中写入 TS=“110”，选择 TI2 作为输入源。

当 TI2 出现上升沿时，计数器开始根据内部时钟计数，并且 TIF 标志置 1。

TI2 的上升沿与实际计数器启动之间的延迟是由于 TI2 输入的重新同步电路引起的。

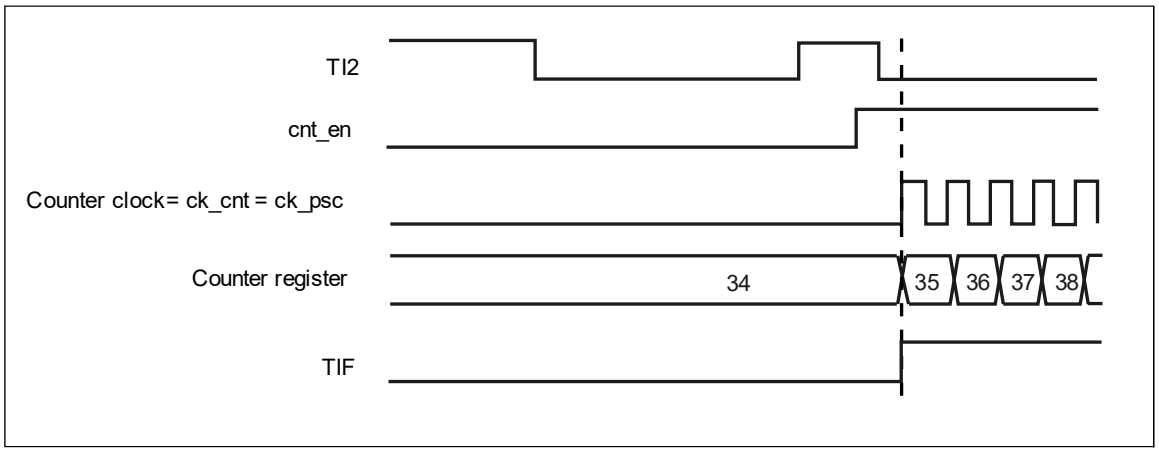



图 16-23 触发模式下的控制电路

### 16.3.12. 定时器同步

TIM 定时器从内部链接在一起，以实现定时器同步或级联。

### 16.3.13. 调试模式

当微控制器进入调试模式(内核停止)时，TIMx 计数器会根据 DBG 模块中的 DBG_TIMx_STOP 配置位选择继续正常工作或者停止工作。

## 16.4. TIM9 和 TIM12 寄存器

有关寄存器说明中使用的缩写，请参见第 1.1 节。

外设寄存器必须按半字（16 位）或字（32 位）进行写访问。而读访问则可按字节（8 位）、半字（ 16 位）或字（ 32 位）进行。

### 16.4.1. TIM9/12 控制寄存器 1 (TIMx_CR1)

偏移地址：0x00

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | CKD[1:0] | | ARPE | Reserved | | | OPM | URS | UDIS | CEN |
| | | | | | | rw | rw | rw | | | | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:10 | 保留 | 必须保持复位值。 |
| 9:8 | CKD | 时钟分频<br>此位域指示定时器时钟 (CK_INT) 频率与数字滤波器所使用的采样时钟 (TIx) 之间的分频比，<br>00：$t_{DTS} = t_{CK_INT}$<br>01：$t_{DTS} = 2 \times t_{CK_INT}$<br>10：$t_{DTS} = 4 \times t_{CK_INT}$<br>11：保留 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 7 | ARPE | 自动重载预装载使能<br>0：TIMx_ARR 寄存器不进行缓冲。<br>1：TIMx_ARR 寄存器进行缓冲。 |
| 6:4 | 保留 | 必须保持复位值。 |
| 3 | OPM | 单脉冲模式<br>0：计数器在发生更新事件时不会停止计数<br>1：计数器在发生下一更新事件时停止计数（将 CEN 位清零） |
| 2 | URS | 更新请求源<br>此位由软件置 1 和清零，用以选择 UEV 事件源。<br>0：失能后，所有以下事件都会生成更新中断：<br>计数器上溢<br>将 UG 位置 1<br>1：使能后，只有计数器上溢会生成更新中断。 |
| 1 | UDIS | 更新禁止<br>此位由软件置 1 和清零，用以使能/禁止更新事件 (UEV) 生成。<br>0：使能 UEV。UEV 可通过以下事件之一生成：<br>计数器上溢<br>将 UG 位置 1<br>然后更新影子寄存器的值。<br>1：禁止 UEV。不会生成 UEV，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。如果<br>UG 位置 1，则会重新初始化计数器和预分频器。 |
| 0 | CEN | 计数器使能<br>0：禁止计数器<br>1：使能计数器<br>在单脉冲模式下，当发生更新事件时会自动将 CEN 位清零。 |

### 16.4.2. TIM9/12 控制寄存器 2 (TIMx_CR2)

偏移地址：0x04

复位值：0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:7 | 保留 | 必须保持复位值。 |
| 6:4 | MMS | 主模式选择<br>这些位用于选择主模式下将要发送到从定时器以实现同步的信息 (TRGO)。这些位的组合如下：<br>000：复位 – TIMx_EGR 寄存器中的 UG 位用作触发输出 (TRGO)。如果复位由触发输入生成（从模式控制器配置为复位模式），则 TRGO 上的信号相比实际复位会有延迟。<br>001：使能 – 计数器使能信号 (CNT_EN) 用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号由 CEN 控制位与门控模式下的触发 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 输入的逻辑或运算组合而成。<br>当计数器使能信号由触发输入控制时，TRGO 上会存在延迟，选择主/从模式时除外（请参见 TIMx_SMCR 寄存器中对 MSM 位的说明）。<br>010：更新 – 选择更新事件作为触发输出 (TRGO)。例如，主定时器可用作从定时器的预分频器。<br>011：比较脉冲 – 一旦发生输入捕获或比较匹配事件，当 CC1IF 被置 1 时（即使已为高电平），触发输出都会发送一个正脉冲。(TRGO)。<br>100：比较 – OC1REF 信号用作触发输出 (TRGO)。<br>101：比较 – OC2REF 信号用作触发输出 (TRGO)。<br>110：保留。<br>111：保留。 |
| 3:0 | 保留 | 必须保持复位值。 |

### 16.4.3. TIM9/12 从模式控制寄存器(TIMx_SMCR)

偏移地址：0x08

复位值：0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:8 | 保留 | 必须保持复位值。 |
| 7 | MSM | 主/从模式<br>0：不执行任何操作。<br>1：延迟事件对触发输入 (TRGI) 的影响，以使当前定时器与其从定时器实现完美同步（通过 TRGO）。此设置适用于要在发生单个外部事件时对多个定时器进行同步的情况。 |
| 6:4 | TS | 触发选择<br>此位域可选择将要用于同步计数器的触发输入。<br>000：内部触发 0 (ITR0)<br>001：内部触发 1 (ITR1)<br>010：内部触发 2 (ITR2)<br>011：内部触发 3 (ITR3)<br>100：TI1 边沿检测器 (TI1F_ED)<br>101：滤波后的定时器输入 1 (TI1FP1)<br>110：滤波后的定时器输入 2 (TI2FP2)<br>111：保留<br>*注意：这些位只能在未使用的情况下（例如，SMS="000"时）进行更改，以避免转换时出现错误的边沿检测。* |
| 3 | 保留 | 必须保持复位值。 |
| 2:0 | SMS | 从模式选择<br>选择外部信号时，触发信号 (TRGI) 的有效边沿与外部输入上所选的极性相关（请参见输入控制寄存器和控制寄存器说明）。<br>000：禁止从模式 – 如果 CEN ="1"，预分频器时钟直接由内部时钟提供 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 001：保留<br>010：保留<br>011：保留<br>100：复位模式 – 在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器并生成一个寄存器更新事件<br>101：门控模式 – 触发输入 (TRGI) 为高电平时使能计数器时钟。只要触发输入变为低电平，计数器立即停止计数（但不复位）。计数器的启动和停止都是受控的。<br>110：触发模式 – 触发信号 TRGI 出现上升沿时启动计数器（但不复位）。只控制计数器的启动<br>111：外部时钟模式 1 – 在所选触发 (TRGI) 的上升沿驱动计数器<br>*注意： 如果将 TI1F_ED 选作触发输入（TS="100"），则不得使用门控模式。实际上，TI1F 每次转换时，TI1F_ED 都输出 1 个脉冲，而门控模式检查的则是触发信号的电平。* |

表 16-1 TIMx 内部触发连接

| 从 TIM | ITR0<br>(TS = 000) | ITR1<br>(TS = 001) | ITR2<br>(TS = 010) | ITR3<br>(TS = 011) |
|---|---|---|---|---|
| TIM2 | TIM1 | TIM8 | TIM3 | TIM4 |
| TIM3 | TIM1 | TIM2 | TIM5 | TIM4 |
| TIM4 | TIM2 | TIM2 | TIM3 | TIM8 |
| TIM5 | TIM2 | TIM3 | TIM4 | TIM8 |
| TIM9 | TIM2 | TIM3 | TIM10 | TIM11 |
| TIM12 | TIM4 | TIM5 | TIM13 | TIM14 |

### 16.4.4. TIM9/12 中断使能寄存器(TIMx_DIER)

偏移地址：0x0C

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | TIE | Res | | | CC2IE | CC1IE | UIE |
| | | | | | | | | | rw | | | | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:7 | 保留 | 必须保持复位值。 |
| 6 | TIE | 触发信号 (TGRI) 中断使能<br>0：禁止触发信号 (TGRI) 中断。<br>1：使能触发信号 (TGRI) 中断。 |
| 5:3 | 保留 | 必须保持复位值。 |
| 2 | CC2IE | 捕获/比较 2 中断使能<br>0：禁止 CC2 中断。<br>1：使能 CC2 中断。 |
| 1 | CC1IE | 捕获/比较 1 中断使能<br>0：禁止 CC1 中断。<br>1：使能 CC1 中断。 |
| 0 | UIE | 更新中断使能 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 0：禁止更新中断。<br>1：使能更新中断。 |

### 16.4.5. TIM9/12 状态寄存器(TIMx_SR)

偏移地址：0x10

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | CC2OF | CC1OF | Reserved | | TIF | Reserved | | | CC2IF | CC1IF | UIF |
| | | | | | rc_w0 | rc_w0 | | | rc_w0 | | | | rc_w0 | rc_w0 | rc_w0 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:11 | 保留 | 必须保持复位值。 |
| 10 | CC2OF | 捕获/比较 2 重复捕获标志<br>请参见 CC1OF 说明 |
| 9 | CC1OF | 捕获/比较 1 重复捕获标志<br>仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将该位清零。<br>0：未检测到重复捕获。<br>1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。 |
| 8:7 | 保留 | 必须保持复位值。 |
| 6 | TIF | 触发中断标志<br>在除门控模式以外的所有模式下，当使能从模式控制器后在 TRGI 输入上检测到有效边沿时，该标志将由硬件置 1。选择门控模式时，该标志将在计数器启动或停止时置 1。但需要通过软件清零。<br>0：未发生触发事件。<br>1：触发中断挂起。 |
| 5:3 | 保留 | 必须保持复位值。 |
| 2 | CC2IF | 捕获/比较 2 中断标志<br>请参见 CC1IF 说明 |
| 1 | CC1IF | 捕获/比较 1 中断标志<br>**如果通道 CC1 配置为输出：**<br>当计数器与比较值匹配时，此标志由硬件置 1。但需要通过软件清零。<br>0：不匹配。<br>1：TIMx_CNT 计数器的值与 TIMx_CCR1 寄存器的值匹配。当 TIMx_CCR1 的值大于 TIMx_ARR 的值时, CC1IF 位将在计数器发生上溢时变为高电平。<br>**如果通道 CC1 配置为输入：**<br>此位将在发生捕获事件时由硬件置 1。通过软件或读取 TIMx_CCR1 寄存器将该位清零。<br>0：未发生输入捕获事件。<br>1：TIMx_CCR1 寄存器中已捕获到计数器值（IC1 上已检测到与所选极性匹配的边沿）。 |
| 0 | UIF | 更新中断标志<br>该位在发生更新事件时通过硬件置 1。但需要通过软件清零。<br>0：未发生更新。<br>1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1： |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 上溢且当 TIMx_CR1 寄存器中 UDIS =“0”时。<br>当由于 TIMx_CR1 寄存器中 URS =“0”且 UDIS =“0”而通过软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。<br>当由于 TIMx_CR1 寄存器中 URS =“0”且 UDIS =“0”而通过触发事件（请参见同步控制寄存器说明）重新初始化 CNT 时。 |

### 16.4.6. TIM9/12 事件生成寄存器(TIMx_EGR)

偏移地址：0x14

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Reserved | | | | | | | | | TG | Reserved | | | CC2G | CC1G | UG |
| | | | | | | | | | w | | | | w | w | w |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 15:7 | 保留 | 必须保持复位值。 |
| 6 | TG | 生成触发信号<br>此位由软件置 1 以生成事件，并由硬件自动清零。<br>0：不执行任何操作<br>1：TIMx_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断 |
| 5:3 | 保留 | 必须保持复位值。 |
| 2 | CC2G | 捕获/比较 2 生成<br>请参见 CC1G 说明 |
| 1 | CC1G | 捕获/比较 1 生成<br>此位由软件置 1 以生成事件，并由硬件自动清零。<br>0：不执行任何操作<br>1：通道 1 上生成捕获/比较事件：<br>**如果通道 CC1 配置为输出：**<br>使能后，CC1IF 标志置 1 并发送相应中断。<br>**如果通道 CC1 配置为输入：**<br>TIMx_CCR1 寄存器中将捕获到计数器的当前值。使能后，CC1IF 标志置 1 并发送相应中断。<br>如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。 |
| 0 | UG | 更新生成<br>该位可通过软件置 1，并由硬件自动清零。<br>0：不执行任何操作<br>1：重新初始化计数器并生成寄存器更新事件。预分频器计数器也将清零，但预分频比不受影响。计数器清零 |

### 16.4.7. TIM9/12 捕获/比较模式寄存器(TIMx_CCMR1)

偏移地址：0x18

复位值：0x0000

这些通道可用于输入（捕获模式）或输出（比较模式）模式。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入模式和输出模式下的功能均不同。对于任一给定位，OCxx 用于说明通道配置为输出时该位对应的功能，ICxx 则用于说明通道配置为输入时该位对应的功能。因此，必须注意同一个位

在输入阶段和输出阶段具有不同的含义。

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | OC2M[2:0] | | | OC2PE | OC2FE | CC2S[1:0] | | Res. | OC1M[2:0] | | | OC1PE | OC1FE | CC1S[1:0] | |
| IC2F[3:0] | | | | IC2PSC[1:0] | | | | IC1F[3:0] | | | | IC1PSC[1:0] | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

**输出比较模式：**

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15 | 保留 | 必须保持复位值。 |
| 14:12 | OC2M[2:0] | 输出比较 2 模式 |
| 11 | OC2PE | 输出比较 2 预装载使能 |
| 10 | OC2FE | 输出比较 2 快速使能 |
| 9:8 | CC2S[1:0] | 捕获/比较 2 选择<br>此位域定义通道方向（输入/输出）以及所使用的输入。<br>00：CC2 通道配置为输出<br>01：CC2 通道配置为输入，IC2 映射到 TI2 上<br>10：CC2 通道配置为输入，IC2 映射到 TI1 上<br>11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效<br>*注意： 仅当通道关闭时（TIMx_CCER 中的 CC2E = 0），才可向 CC2S 位写入数据。* |
| 7 | 保留 | 必须保持复位值。 |
| 6:4 | OC1M | 输出比较 1 模式<br>这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 和 OC1N 的有效电平则分别取决于 CC1P 位和 CC1NP 位。<br>000：冻结 – 输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成 任何影响。（该模式用于生成时基）。<br>001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。<br>010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。<br>011：翻转 – TIMx_CNT=TIMx_CCR1 时，OC1REF 发生翻转。<br>100：强制变为无效电平 – OC1REF 强制变为低电平。<br>101：强制变为有效电平 – OC1REF 强制变为高电平。<br>110：PWM 模式 1 – 在递增计数模式下，只要 TIMx_CNT < TIMx_CCR1，通道 1 便为有效 状态，否则为无效状态。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，通道 1 便为无 效状态（OC1REF=“0”），否则为有效状态（OC1REF=“1”）。<br>111：PWM 模式 2 – 在递增计数模式下，只要 TIMx_CNT < TIMx_CCR1，通道 1 便为无效状态，否则为有效状态。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，通道 1 便为有效状态，否则为无效状态。<br>*注意： 在 PWM 模式 1 或 PWM 模式 2 下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，OCREF 电平才会发生更改。* |
| 3 | OC1PE | 输出比较 1 预装载使能<br>0：禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | TIMx_CCR1 写入数据，写入后将立即使用新值。<br>1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到活动寄存器中。<br>*注意：只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式（TIMx_CR1 寄存器中的 OPM 位置 1）。其它情况下则无法保证该行为。* |
| 2 | OC1FE | 输出比较 1 快速使能<br>此位用于加快触发输入事件对 CC 输出的影响。<br>0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期。<br>1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。<br>仅当通道配置为 PWM1 或 PWM2 模式时，OC1FE 才会起作用。 |
| 1:0 | CC1S | 捕获/比较 1 选择<br>此位域定义通道方向（输入/输出）以及所使用的输入。<br>00：CC1 通道配置为输出<br>01：CC1 通道配置为输入，IC1 映射到 TI1 上<br>10：CC1 通道配置为输入，IC1 映射到 TI2 上<br>11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存 器）选择内部触发输入时有效<br>*注意： 仅当通道关闭时（TIMx_CCER 中的 CC1E = 0），才可向 CC1S 位写入数据。* |

**输入捕获模式：**

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:12 | IC2F | 输入捕获 2 滤波器 |
| 11:10 | IC2PSC[1:0] | 输入捕获 2 预分频器 |
| 9:8 | CC2S | 捕获/比较 2 选择<br>此位域定义通道方向（输入/输出）以及所使用的输入。<br>00：CC2 通道配置为输出<br>01：CC2 通道配置为输入，IC2 映射到 TI2 上<br>10：CC2 通道配置为输入，IC2 映射到 TI1 上<br>11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存 器）选择内部触发输入时有效<br>注意： 仅当通道关闭时（TIMx_CCER 中的 CC2E = 0），才可向 CC2S 位写入数据。 |
| 7:4 | IC1F | 输入捕获 1 滤波器<br>此位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿：<br>0000：无滤波器，以 $f_{DTS}$ 频率进行采样<br>0001：$f_{SAMPLING}=f_{CK_INT}$，N=2<br>0010：$f_{SAMPLING}=f_{CK_INT}$，N=4<br>0011：$f_{SAMPLING}=f_{CK_INT}$，N=8<br>0100：$f_{SAMPLING}=f_{DTS}/2$，N=6 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 0101：$f_{SAMPLING}$ =$f_{DTS}$/2，N=8<br>0110：$f_{SAMPLING}$ =$f_{DTS}$/4，N=6<br>0111：$f_{SAMPLING}$ =$f_{DTS}$/4，N=8<br>1000：$f_{SAMPLING}$=$f_{DTS}$/8，N=6<br>1001：$f_{SAMPLING}$ =$f_{DTS}$/8，N=8<br>1010：$f_{SAMPLING}$ =$f_{DTS}$/16，N=5<br>1011：$f_{SAMPLING}$ =$f_{DTS}$/16，N=6<br>1100：$f_{SAMPLING}$ =$f_{DTS}$/16，N=8<br>1101：$f_{SAMPLING}$ =$f_{DTS}$/32，N=5<br>1110：$f_{SAMPLING}$ =$f_{DTS}$/32，N=6<br>1111：$f_{SAMPLING}$ =$f_{DTS}$/32，N=8 |
| 3:2 | IC1PSC | 输入捕获 1 预分频器<br>此位域定义 CC1 输入 (IC1) 的预分频比。<br>只要 CC1E=“0”（TIMx_CCER 寄存器），预分频器便立即复位。<br>00：无预分频器，捕获输入上每检测到一个边沿便执行捕获<br>01：每发生 2 个事件便执行一次捕获<br>10：每发生 4 个事件便执行一次捕获<br>11：每发生 8 个事件便执行一次捕获 |
| 1:0 | CC1S | 捕获/比较 1 选择<br>此位域定义通道方向（输入/输出）以及所使用的输入。<br>00：CC1 通道配置为输出<br>01：CC1 通道配置为输入，IC1 映射到 TI1 上<br>10：CC1 通道配置为输入，IC1 映射到 TI2 上<br>11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效<br>*注意： 仅当通道关闭时（TIMx_CCER 中的 CC1E = 0），才可向 CC1S 位写入数据。* |

### 16.4.8. TIM9/12 捕获/比较使能寄存器(TIMx_CCER)

偏移地址：0x20

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | CC2NP | TG | CC2P | CC2E | CC1NP | Res. | CC1P | CC1E |
| | | | | | | | | rw | | rw | rw | rw | | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:8 | 保留 | 必须保持复位值。 |
| 7 | CC2NP | 捕获/比较 2 输出极性<br>请参见 CC1NP 说明 |
| 6 | 保留 | 必须保持复位值。 |
| 5 | CC2P | 捕获/比较 2 输出极性<br>请参见 CC1P 说明 |
| 4 | CC2E | 捕获/比较 2 输出使能<br>请参见 CC1E 说明 |
| 3 | CC1NP | 捕获/比较 1 互补输出极性<br>CC1 通道配置为输出：CC1NP 必须保持清零<br>CC1 通道配置为输入：CC1NP 与 CC1P 配合使用可定义 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | TI1FP1/TI2FP1 的极性（请参见 CC1P 说明）。 |
| 2 | 保留 | 必须保持复位值。 |
| 1 | CC1P | 捕获/比较 1 输出极性<br>**CC1 通道配置为输出：**<br>0：OC1 高电平有效。<br>1：OC1 低电平有效。<br>**CC1 通道配置为输入：**<br>CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的极性。<br>00: 未反相/上升沿触发电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式或编码器模式下执行触发操作）。<br>01：反相/下降沿触发电路对 TIxFP1 下降沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式或编码器模式下执行触发操作）。<br>10：保留，不使用此配置。<br>*注意：11：未反相/上升沿和下降沿均触发电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行 捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。编码器模式下不得使用此配置。* |
| 0 | CC1E | 捕获/比较 1 输出使能<br>**CC1 通道配置为输出：**<br>0：关闭 – OC1 未激活。<br>1：开启 – 在相应输出引脚上输出 OC1 信号。<br>**CC1 通道配置为输入**:<br>该位决定了输入捕获/比较寄存器 1 (TIMx_CCR1) 是否能够实际捕获到计数器的值。<br>0：禁止捕获。<br>1：使能捕获。 |

表 16-2 标准 OCx 通道的输出控制位

| CCxE 位 | OCx 输出状态 |
|---|---|
| 0 | 禁止输出（Ocx=“0”，OCx_EN=“0”） |
| 1 | Ocx=OCxREF+极性，Ocx_EN=“1” |

### 16.4.9. TIM9/12 计数器(TIMx_CNT)

偏移地址：0x24

复位值：0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:0 | CNT[15:0] | 计数器值 |

### 16.4.10. TIM9/12 预分频器 (TIMx_PSC)

偏移地址：0x28

复位值：0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:0 | PSC[15:0] | 预分频器值<br>计数器时钟频率 CK_CNT 等于 $f_{CK_PSC}$ / (PSC[15:0] + 1)。<br>PSC 包含在每次发生更新事件时要装载到实际预分频器寄存器的值。 |

### 16.4.11. TIM9/12 自动重载寄存器 (TIMx_ARR)

偏移地址：0x2C

复位值：0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:0 | ARR[15:0] | 自动重载值<br>ARR 为要装载到实际自动重载寄存器的值。<br>当自动重载值为空时，计数器不工作。 |

### 16.4.12. TIM9/12 捕获/比较寄存器 1 (TIMx_CCR1)

偏移地址：0x34

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CCR1[15:0] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:0 | CCR1[15:0] | 捕获/比较 1 值<br>**如果通道 CC1 配置为输出：**<br>CCR1 是捕获/比较寄存器 1 的预装载值。<br>如果没有通过 TIMx_CCMR 寄存器中的 OC1PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器 1）。<br>实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC1 输出上发出信号的值。 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | **如果通道 CC1 配置为输入：**<br>CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。 |

### 16.4.13. TIM9/12 捕获/比较比较器 2 (TIMx_CCR2)

偏移地址：0x38

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CCR2[15:0] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:0 | CCR2[15:0] | 捕获/比较 2 值<br>**如果通道 CC2 配置为输出：**<br>CCR2 为要装载到实际捕获/比较 2 寄存器的值（预装载值）。<br>如果没有通过 TIMx_CCMR2 寄存器中的 OC2PE 位来使能预装载功能，写入的数值会被直接传 输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 2 寄存器中。<br>实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC2 输出上发出信号的值。<br>**如果通道 CC2 配置为输入：**<br>CCR2 为上一个输入捕获 2 事件 (IC2) 发生时的计数器值。 |

## 16.5. TIM10/11/13/14 寄存器

外设寄存器的写访问仅支持半字（16 位）或字（32 位）。而读访问可支持字节（8 位）、半字（ 16 位）或字（ 32 位）。

### 16.5.1. TIM10/11/13/14 控制寄存器 1 (TIMx_CR1)

偏移地址：0x00

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | CKD[1:0] | | ARPE | Reserved | | | | URS | UDIS | CEN |
| | | | | | | rw | rw | rw | | | | | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:10 | 保留 | 必须保持复位值。 |
| 9:8 | CKD | 时钟分频<br>此位域指示定时器时钟 (CK_INT) 频率与数字滤波器所使用的采样时钟（ETR、TIx）之间的分频比，<br>00：$t_{DTS} = t_{CK_INT}$<br>01：$t_{DTS} = 2 \times t_{CK_INT}$<br>10：$t_{DTS} = 4 \times t_{CK_INT}$<br>11：保留 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 7 | ARPE | 自动重载预装载使能<br>0：TIMx_ARR 寄存器不进行缓冲<br>1：TIMx_ARR 寄存器进行缓冲 |
| 6:3 | 保留 | 必须保持复位值。 |
| 2 | URS | 更新请求源<br>此位由软件置 1 和清零，用以选择更新中断 (UEV) 源。<br>0：失能后，所有以下事件都会生成 UEV：<br>计数器上溢<br>将 UG 位置 1<br>1：使能后，只有计数器上溢会生成 UEV。 |
| 1 | UDIS | 更新禁止<br>此位由软件置 1 和清零，用以使能/禁止更新中断 (UEV) 事件生成。<br>0：使能 UEV。UEV 可通过以下事件之一生成：<br>计数器上溢<br>将 UG 位置 1<br>然后更新影子寄存器的值。<br>1：禁止 UEV。不会生成 UEV，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。如果 UG 位置 1，则会重新初始化计数器和预分频器。 |
| 0 | CEN | 计数器使能<br>0：禁止计数器<br>1：使能计数器 |

### 16.5.2. TIM10/11/13/14 状态寄存器 (TIMx_SR)

偏移地址：0x10

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | CC1OF | ARPE | | | | | | | CC1IF | UIF |
| | | | | | | rw_w0 | | | | | | | | rw_w0 | rw_w0 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:10 | 保留 | 必须保持复位值。 |
| 9 | CC1OF | 捕获/比较 1 重复捕获标志<br>仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将该位清零。<br>0：未检测到重复捕获。<br>1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。 |
| 8:2 | 保留 | 必须保持复位值。 |
| 1 | CC1IF | 捕获/比较 1 中断标志<br>**如果通道 CC1 配置为输出：**<br>当计数器与比较值匹配时，此标志由硬件置 1。但需要通过软件清零。<br>0：不匹配。<br>1：TIMx_CNT 计数器的值与 TIMx_CCR1 寄存器的值匹配。当 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | TIMx_CCR1 的值大于 TIMx_ARR 的值时，CC1IF 位将在计数器发生上溢时变为高电平。<br>**如果通道 CC1 配置为输入：**<br>此位将在发生捕获事件时由硬件置 1。通过软件或读取 TIMx_CCR1 寄存器将该位清零。<br>0：未发生输入捕获事件。<br>1：TIMx_CCR1 寄存器中已捕获到计数器值（IC1 上已检测到与所选极性匹配的边沿）。 |
| 0 | UIF | 更新中断标志<br>该位在发生更新事件时通过硬件置 1。但需要通过软件清零。<br>0：未发生更新。<br>1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：<br>上溢且当 TIMx_CR1 寄存器中 UDIS =“0”时。<br>当由于 TIMx_CR1 寄存器中 URS =“0”且 UDIS =“0”而通过软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。 |

### 16.5.3. TIM10/11/13/14 事件生成寄存器(TIMx_EGR)

偏移地址：0x14

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | CC1G | UG |
| | | | | | | | | | | | | | | w | w |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:2 | 保留 | 必须保持复位值。 |
| 1 | CC1G | 捕获/比较 1 生成<br>此位由软件置 1 以生成事件，并由硬件自动清零。<br>0：不执行任何操作。<br>1：通道 1 上生成捕获/比较事件：<br>**如果通道 CC1 配置为输出：**<br>使能后，CC1IF 标志置 1 并发送相应的中断。<br>**如果通道 CC1 配置为输入：**<br>TIMx_CCR1 寄存器中将捕获到计数器当前值。使能后，CC1IF 标志置 1 并发送相应中断。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。 |
| 0 | UG | 更新生成<br>该位可通过软件置 1，并由硬件自动清零。<br>0：不执行任何操作。<br>1：重新初始化计数器并生成寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。计数器清零。 |

### 16.5.4. TIM10/11/13/14 捕获/比较模式寄存器 1(TIMx_CCMR1)

偏移地址：0x18

复位值：0x0000

这些通道可用于输入（捕获模式）或输出（比较模式）模式。通道方向通过配置相应的 CCxS 位进行定义。

此寄存器的所有其他位在输入模式和输出模式下的功能均不同。对于任一给定位，OCxx 用于说明通道配置为输出时该位对应的功能，ICxx 则用于说明通道配置为输入时该位对应的功能。因此，必须注意同一个位在输入阶段和输出阶段具有不同的含义。

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | OC1M[2:0] | | | OC1PE | OC1FE | CC1S[1:0] | |
| Reserved | | | | | | | | IC1F[3:0] | | | | IC1PSC[1:0] | | | |
| | | | | | | | | rw | rw | rw | rw | rw | rw | rw | rw |

**输出比较模式：**

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:7 | 保留 | 必须保持复位值。 |
| 6:4 | OC1M | 输出比较 1 模式<br>这些位定义提供 OC1 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 的有效电平则取决于 CC1P 位。<br>000：冻结。输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。<br>001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。<br>010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。<br>011：翻转 – TIMx_CNT = TIMx_CCR1 时，OC1REF 发生翻转。<br>100：强制变为无效电平 – OC1REF 强制变为低电平。<br>101：强制变为有效电平 – OC1REF 强制变为高电平。<br>110：PWM 模式 1 –只要 TIMx_CNT < TIMx_CCR1，通道 1 便为有效状态，否则为无效状态。<br>111：PWM 模式 2 – 只要 TIMx_CNT < TIMx_CCR1，通道 1 便为无效状态，否则为有效 状态。<br>*注意：在 PWM 模式 1 或 PWM 模式 2 下，仅当比较结果发生改变或输出比较模式由"冻结"模式切换到"PWM"模式时，OCREF 电平才会发生更改。* |
| 3 | OC1PE | 输出比较 1 预装载使能<br>0：禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立 即使用新值。<br>1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到活动寄存器中。<br>*注意：只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式（TIMx_CR1 寄存器中的 OPM 位置 1）。其它情况下则无法保证该行为。* |
| 2 | OC1FE | 输出比较 1 快速使能<br>此位用于加快触发输入事件对 CC 输出的影响。<br>0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期。<br>1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 为 PWM1 或 PWM2 模式时，OC1FE 才会起作用。 |
| 1:0 | CC1S | 捕获/比较 1 选择<br>此位域定义通道方向（输入/输出）以及所使用的输入。<br>00：CC1 通道配置为输出。<br>01：CC1 通道配置为输入，IC1 映射到 TI1 上。<br>10：保留<br>11：保留<br>*注意： 仅当通道关闭时（TIMx_CCER 中的 CC1E = 0），才可向 CC1S 位写入数据。* |

**输入捕获模式：**

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:8 | 保留 | 必须保持复位值。 |
| 7:4 | IC1F | 输入捕获 1 滤波器<br>此位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器<br>组成，每 N 个事件才视为一个有效边沿：<br>0000：无滤波器，以 $f_{DTS}$ 频率进行采样<br>0001：$f_{SAMPLING} = f_{CK_INT}$，N=2<br>0010：$f_{SAMPLING} = f_{CK_INT}$，N=4<br>0011：$f_{SAMPLING} = f_{CK_INT}$，N=8<br>0100：$f_{SAMPLING} = f_{DTS}/2$，N=6<br>0101：$f_{SAMPLING} = f_{DTS}/2$，N=8<br>0110：$f_{SAMPLING} = f_{DTS}/4$，N=6<br>0111：$f_{SAMPLING} = f_{DTS}/4$，N=8<br>1000：$f_{SAMPLING} = f_{DTS}/8$，N=6<br>1001：$f_{SAMPLING} = f_{DTS}/8$，N=8<br>1010：$f_{SAMPLING} = f_{DTS}/16$，N=5<br>1011：$f_{SAMPLING} = f_{DTS}/16$，N=6<br>1100：$f_{SAMPLING} = f_{DTS}/16$，N=8<br>1101：$f_{SAMPLING} = f_{DTS}/32$，N=5<br>1110：$f_{SAMPLING} = f_{DTS}/32$，N=6<br>1111：$f_{SAMPLING} = f_{DTS}/32$，N=8 |
| 3:2 | IC1PSC | 输入捕获 1 预分频器<br>此位域定义 CC1 输入 (IC1) 的预分频比。<br>只要 CC1E="0"（TIMx_CCER 寄存器），预分频器便立即复位。<br>00：无预分频器，捕获输入上每检测到一个边沿便执行捕获<br>01：每发生 2 个事件便执行一次捕获<br>10：每发生 4 个事件便执行一次捕获<br>11：每发生 8 个事件便执行一次捕获 |
| 1:0 | CC1S | 捕获/比较 1 选择<br>此位域定义通道方向（输入/输出）以及所使用的输入。<br>00：CC1 通道配置为输出<br>01：CC1 通道配置为输入，IC1 映射到 TI1 上<br>10：保留<br>11：保留<br>*注意： 仅当通道关闭时（TIMx_CCER 中的 CC1E = 0），才可向 CC1S 位写入数据。* |

### 16.5.5. TIM10/11/13/14 捕获/比较使能寄存器 (TIMx_CCER)

偏移地址：0x20

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | CC1NP | Res. | CC1P | CC1E |
| | | | | | | | | | | | | rw | | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:4 | 保留 | 必须保持复位值。 |
| 3 | CC1NP | 捕获/比较 1 互补输出极性<br>CC1 通道配置为输出：CC1NP 必须保持清零。<br>CC1 通道配置为输入：CC1NP 位与 CC1P 配合使用可定义 TI1FP1 的极性（请参见 CC1P 说明）。 |
| 2 | 保留 | 必须保持复位值。 |
| 1 | CC1P | 捕获/比较 1 输出极性<br>**CC1 通道配置为输出：**<br>0：OC1 高电平有效<br>1：OC1 低电平有效<br>**CC1 通道配置为输入：**<br>CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的极性。<br>00：未反相/上升沿触发电路对 TI1FP1 上升沿敏感（捕获模式），TI1FP1 未反相。<br>01：反相/下降沿触发电路对 TI1FP1 下降沿敏感（捕获模式），TI1FP1 反相。<br>10：保留，不使用此配置<br>11：未反相/上升沿和下降沿均触发电路对 TI1FP1 上升沿和下降沿均敏感（捕获模式），TI1FP1 未反相。 |
| 0 | CC1E | 捕获/比较 1 输出使能<br>**CC1 通道配置为输出：**<br>0：关闭——OC1 未激活<br>1：开启——在相应输出引脚上输出 OC1 信号<br>**CC1 通道配置为输入**：<br>该位决定了输入捕获/比较寄存器 1 (TIMx_CCR1) 是否能够实际捕获到计数器的值。<br>0：禁止捕获<br>1：使能捕获 |

表 16-3 标准 OCx 通道的输出控制位

| CCxE 位 | OCx 输出状态 |
|---|---|
| 0 | 禁止输出（Ocx=“0”，OCx_EN=“0”） |
| 1 | Ocx=OCxREF+极性，Ocx_EN=“1” |

### 16.5.6. TIM10/11/13/14 计数器(TIMx_CNT)

偏移地址：0x24

复位值：0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:0 | CNT[15:0] | 计数器值 |

### 16.5.7. TIM10/11/13/14 预分频器(TIMx_PSC)

偏移地址：0x28

复位值：0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:0 | PSC[15:0] | 预分频器值<br>计数器时钟频率 CK_CNT 等于 $f_{CK_PSC}$ / (PSC[15:0] + 1)。<br>PSC 包含在每次发生更新事件时要装载到实际预分频器寄存器的值。 |

### 16.5.8. TIM10/11/13/14 自动重载寄存器(TIMx_ARR)

偏移地址：0x2C

复位值：0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:0 | ARR[15:0] | 自动重载值<br>ARR 为要装载到实际自动重载寄存器的值。<br>当自动重载值为空时，计数器不工作。 |

### 16.5.9. TIM10/11/13/14 捕获/比较寄存器 1(TIMx_CCR1)

偏移地址：0x34

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CCR1[15:0] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:0 | CCR1[15:0] | 捕获/比较 1 值 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | **如果通道 CC1 配置为输出：**<br>CCR1 为要装载到实际捕获/比较 1 寄存器的值（预装载值）。如果没有通过 TIMx_CCMR 寄存器中的 OC1PE 位来使能预装载功能，写入的数值会被直接传 输至当前寄存器中。否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器 1）实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC1 输出上发出信号的值。<br>**如果通道 CC1 配置为输入：**<br>CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。 |

### 16.5.10. TIM10/11/13/14 选项寄存器 1(TIM11_OR)

偏移地址：0x50

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Reserved | | | | | | | | | | | | | | TI1_RMP[1:0] | |
| | | | | | | | | | | | | | | rw | |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 15:2 | 保留 | 必须保持复位值。 |
| 1:0 | TI1_RMP[1:0] | TIM11 输入 1 重映射功能<br>由软件置 1 和清零。<br>00,01,11: TIM11 通道 1 连接到 GPIO（请参见数据手册中的复用功能映射表）。<br>10： HSE_RTC 时钟（由可编程预分频器分频的 HSE）连接到 TIM11_CH1 输入，用于测量用途 |

# 17. 基本定时器(TIM6/TIM7)

## 17.1. TIM6 和 TIM7 简介

基本定时器 TIM6 和 TIM7 包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。

此类定时器不仅可用作通用定时器以生成时基，还可以专门用于驱动数模转换器 (DAC)。实际上，此类定时器内部连接到 DAC 并能够通过其触发输出驱动 DAC。

这些定时器彼此完全独立，不共享任何资源。

## 17.2. TIM6 和 TIM7 主要特性

基本定时器（ TIM6 和 TIM7）的特性包括:

- 16 位自动重载递增计数器;
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（即运行时修改），分频系数介于 1 和 65536 之间;
- 用于触发 DAC 的同步电路;
- 发生如下更新事件时会生成中断/DMA 请求：计数器上溢。

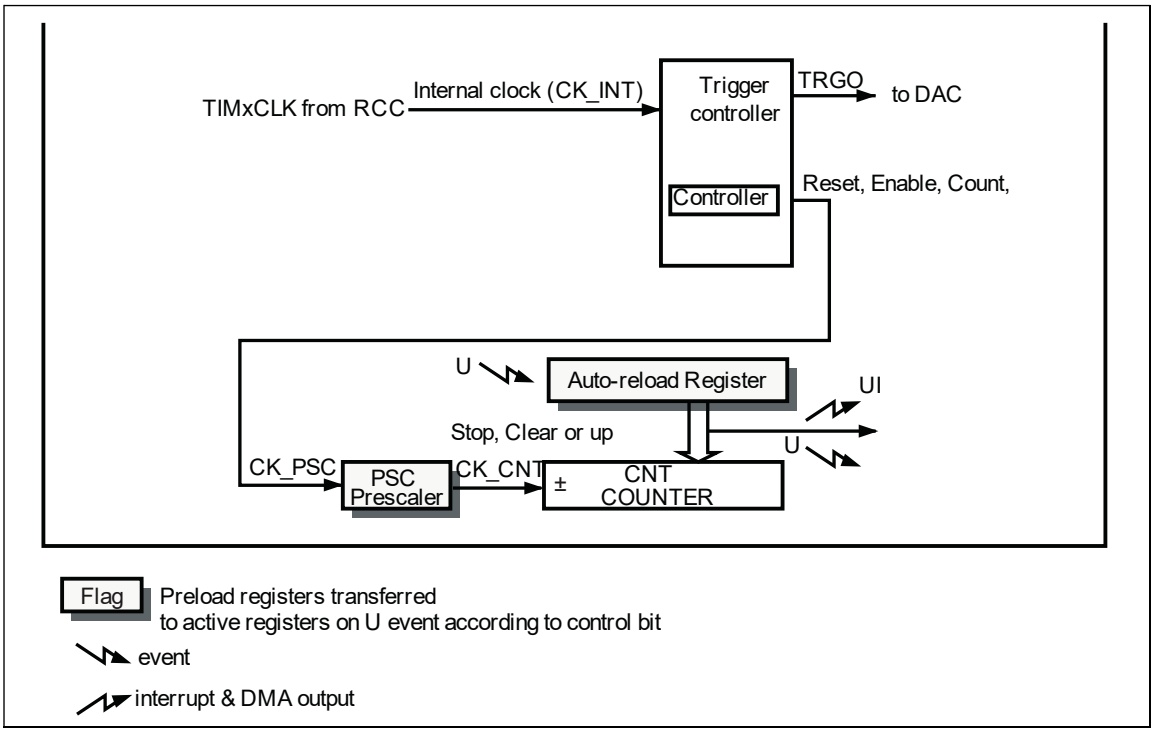



图 17-1 基本定时器框图

## 17.3. TIM6 和 TIM7 功能说明

### 17.3.1. 时基单元

可编程定时器的主要模块由一个 16 位递增计数器及其相关的自动重载寄存器组成。计数器的时钟可通过预分频器进行分频。

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

时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)

自动重载寄存器是预装载的。每次尝试对自动重载寄存器执行读写操作时，都会访问预装载寄存器。预装载寄存器的内容既可以直接传送到影子寄存器，也可以在每次发生更新事件 UEV 时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位 (CEN)置 1 时，才会启动计数器。

请注意，实际的计数器使能信号 CNT_EN 在 CEN 置 1 的一个时钟周期后被置 1。

**预分频器说明**

预分频器可对计数器时钟频率进行分频，分频系数介于 1 和 65536 之间。该预分频器基于 TIMx_PSC 寄存器中的 16 位寄存器所控制的 16 位计数器。由于 TIMx_PSC 控制寄存器有缓冲，因此可对预分频器进行实时更改。而新的预分频比将在下一更新事件发生时被采用。下图给出了在预分频比发生实时变化时一些计数器行为的示例。

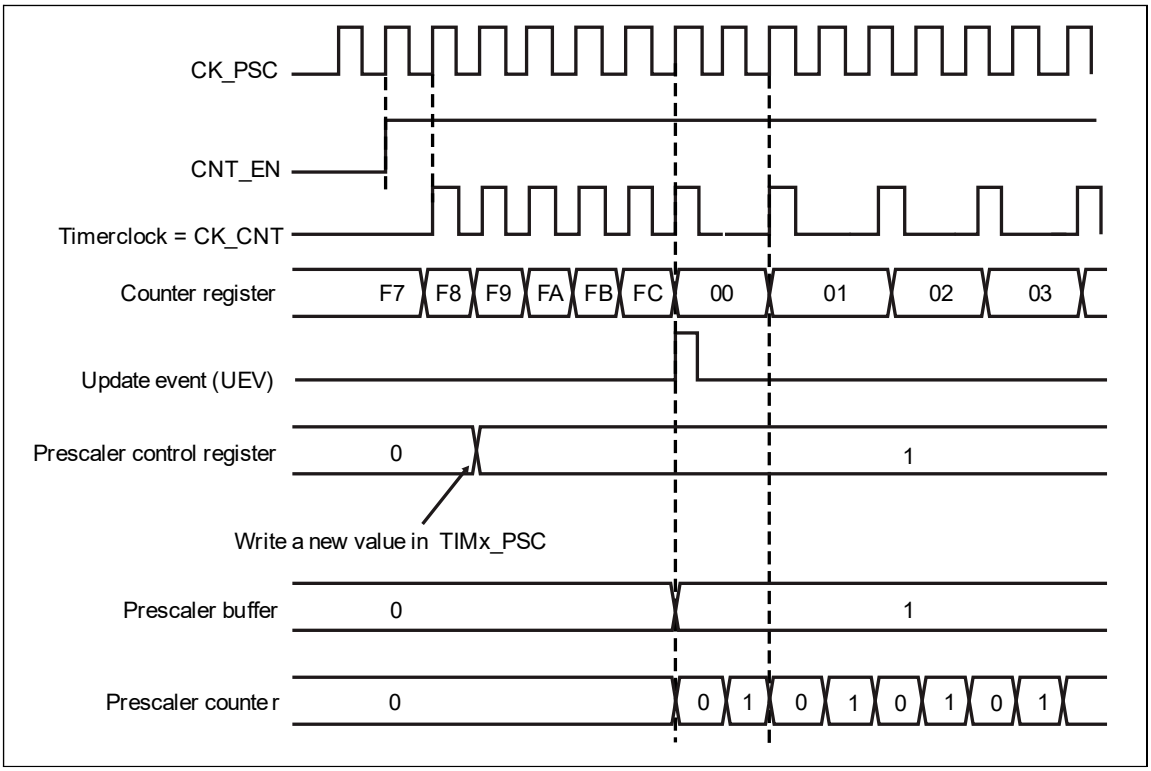



图 17-2 预分频器分频由 1 变为 2 时的计数器时序图

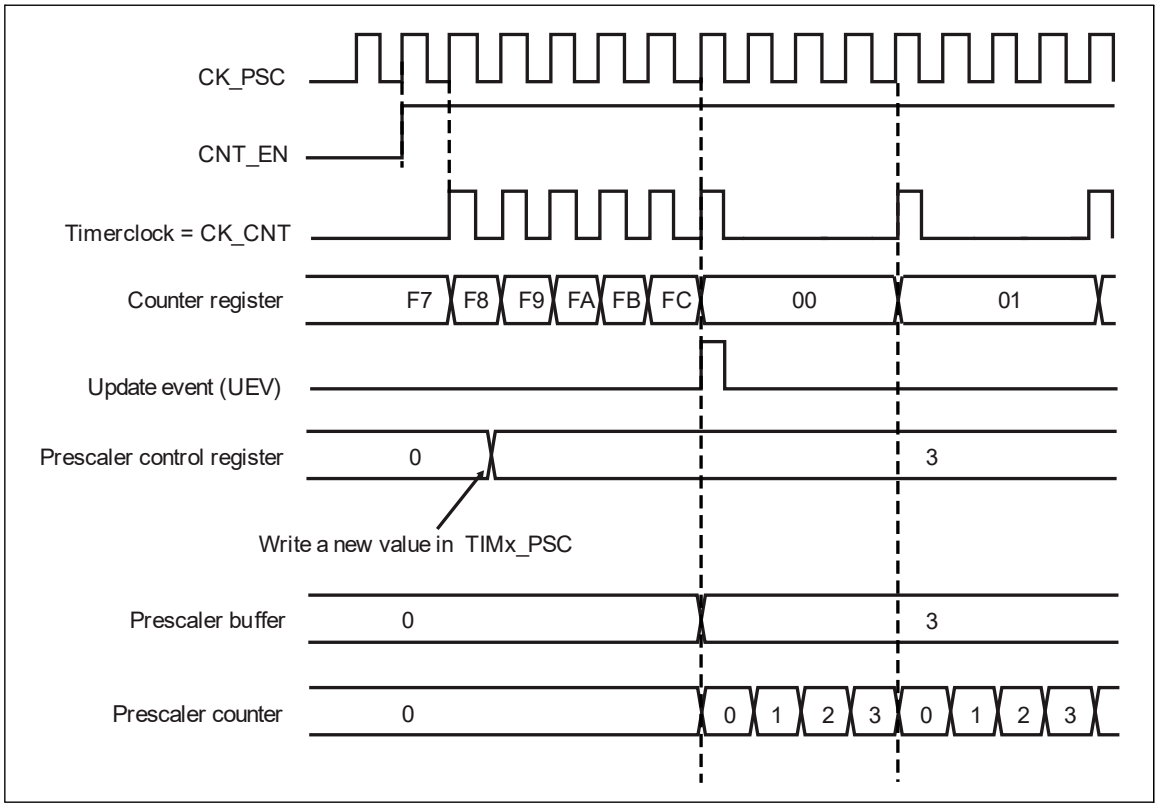



图 17-3 预分频器分频由 1 变为 4 时的计数器时序图

### 17.3.2. 计数模式

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

每次发生计数器上溢时会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。这样，直到 UDIS 位中写入 0 前便不会生成任何更新事件，但计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。

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

- 使用预装载值（TIMx_PSC 寄存器的内容）重新装载预分频器的缓冲区；
- 使用预装载值 (TIMx_ARR) 更新自动重载影子寄存器。

以下各图显示了当 TIMx_ARR=0x36 时不同时钟频率下计数器行为的示例。

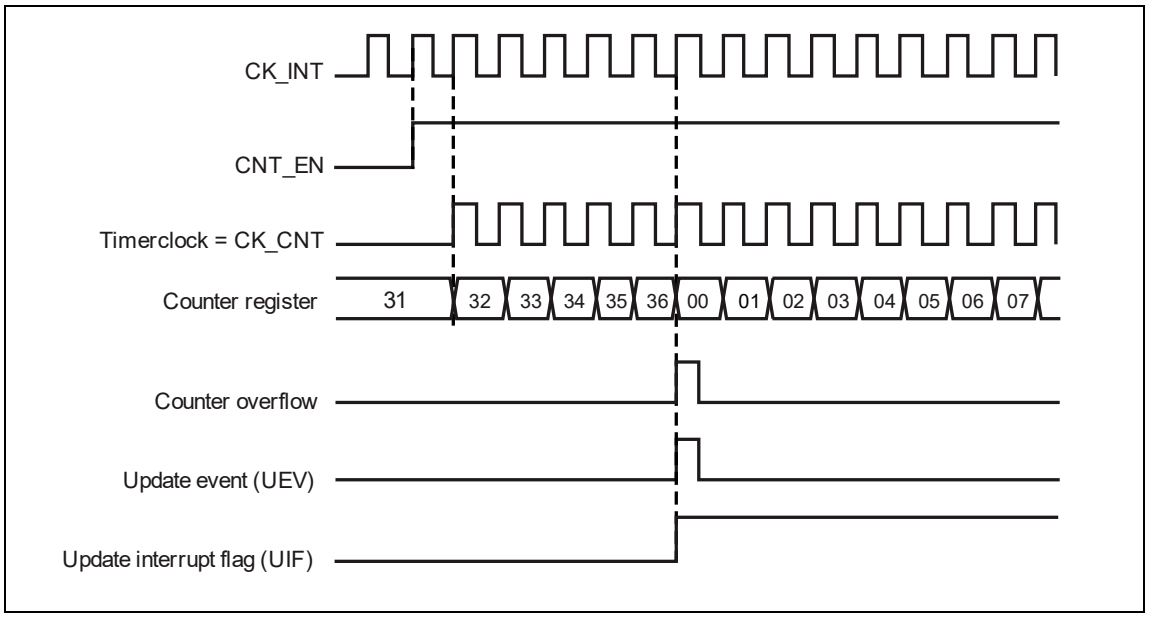



图 17-4 计数器时序图，内部时钟 1 分频

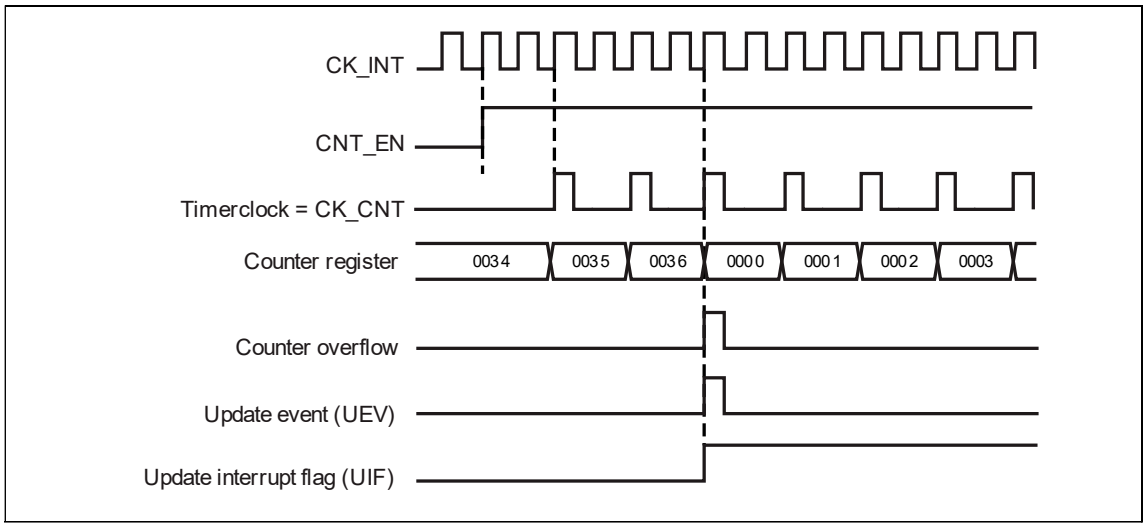



图 17-5 计数器时序图，内部时钟 2 分频

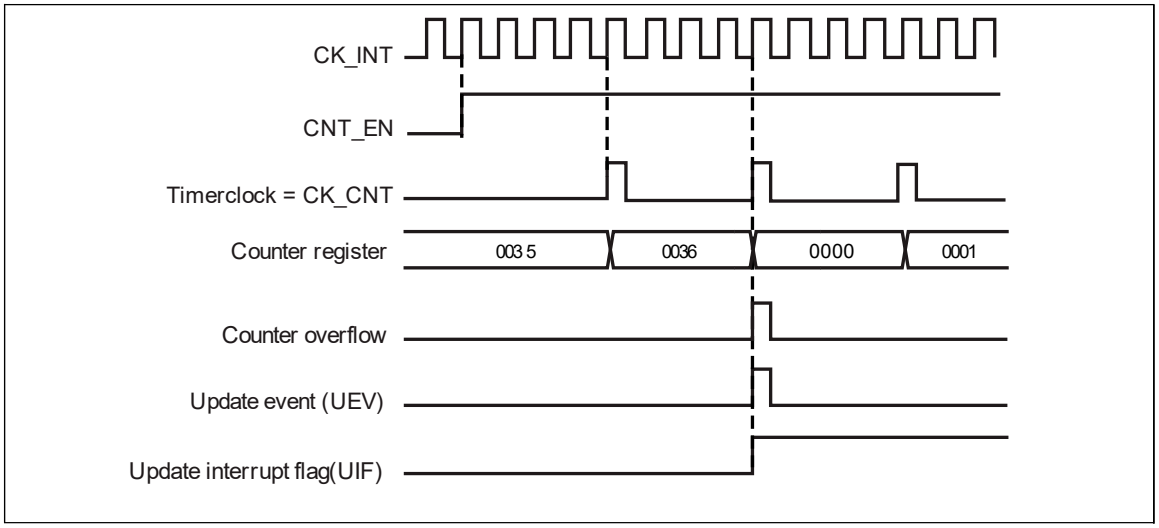



图 17-6 计数器时序图，内部时钟 4 分频

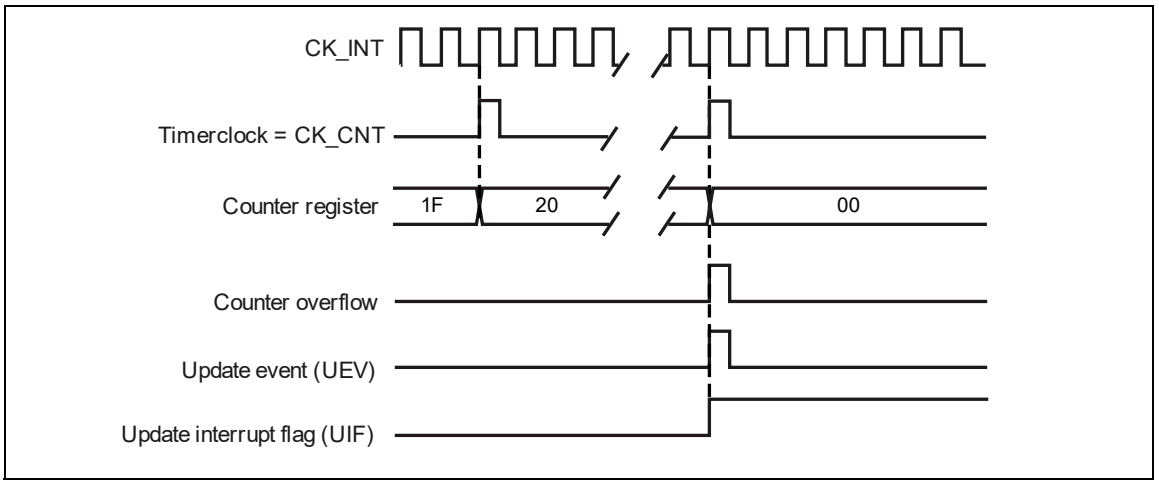



图 17-7 计数器时序图，内部时钟 N 分频

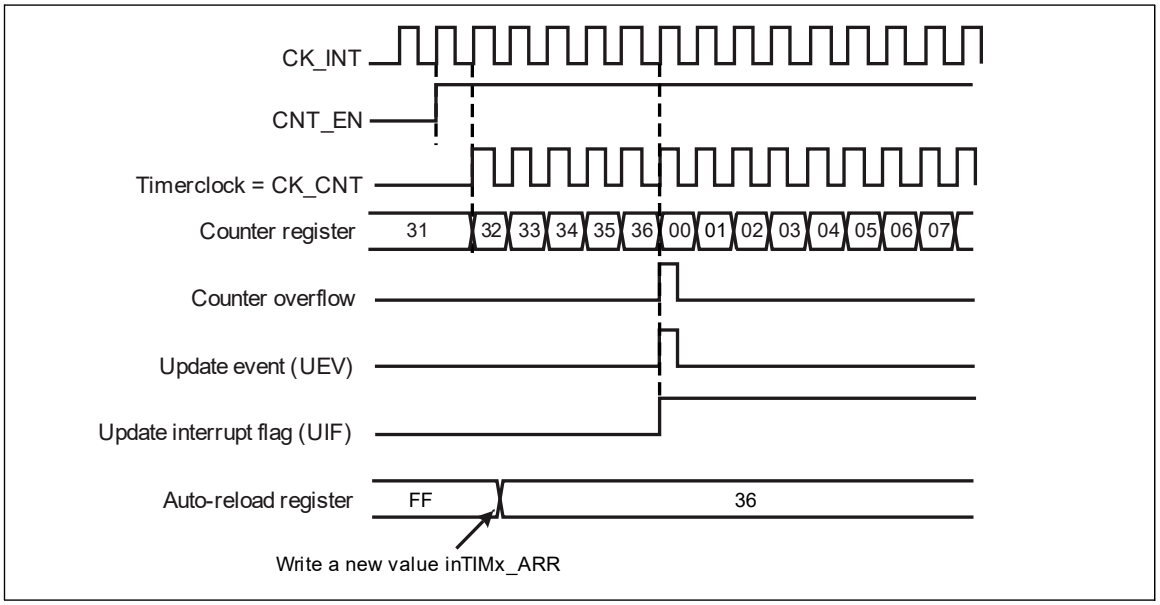



图 17-8 计数器时序图，ARPE = 0 时更新事件(TIMx_ARR 未预装载)

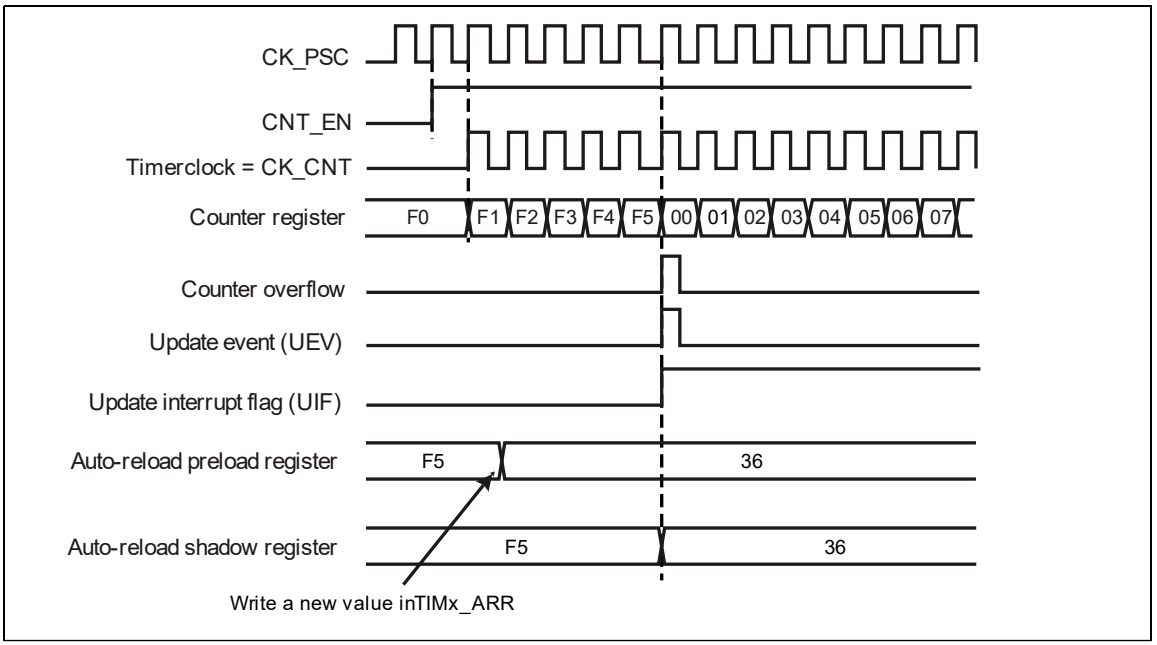



图 17-9 计数器时序图，ARPE = 1 时更新事件(TIMx_ARR 已预装载)

### 17.3.3. 时钟源

计数器时钟由内部时钟 (CK_INT) 源提供。

CEN（TIMx_CR1 寄存器中）和 UG 位（TIMx_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（保持自动清零的 UG 除外）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 CK_INT 提供。下图显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

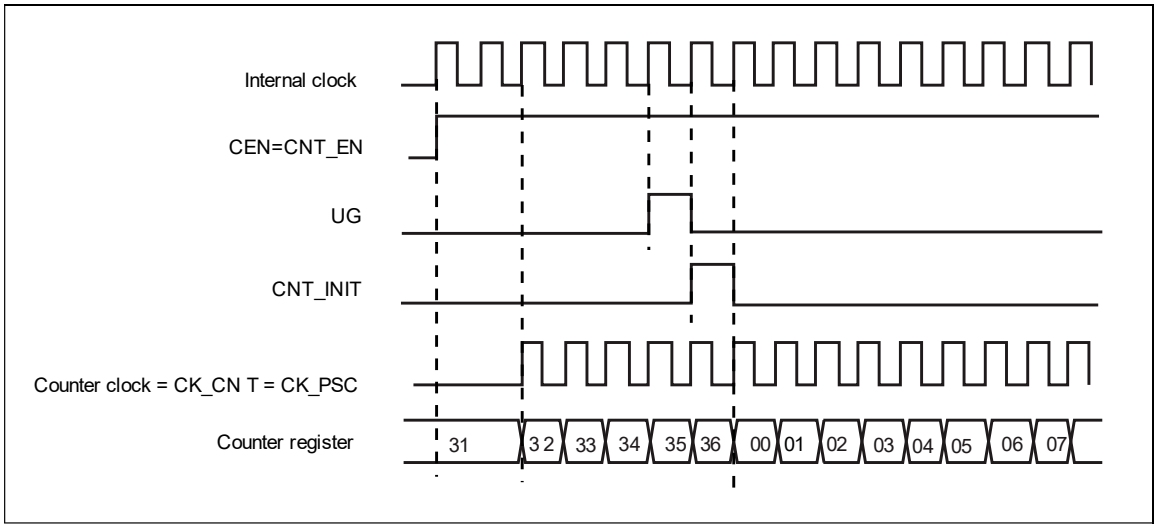



图 17-10 正常模式下的控制电路，内部时钟 1 分频

### 17.3.4. 调试模式

当微控制器进入调试模式(内核停止)时，TIMx 计数器会根据 DBG 模块中的 DBG_TIMx_STOP 配置位选择继续正常工作或者停止工作。

## 17.4. TIM6 和 TIM7 寄存器

有关寄存器说明中使用的缩写，请参见第 1.1 节。

外设寄存器必须按半字（16 位）或字（32 位）进行写访问。而读访问则可按字节（8 位）、半字（ 16 位）或字（ 32 位）进行。

### 17.4.1. TIM6 和 TIM7 控制寄存器 1 (TIMx_CR1)

偏移地址：0x00

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | ARPE | Reserved | | | OPM | URS | UDIS | CEN |
| | | | | | | | | rw | | | | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:8 | 保留 | 必须保持复位值。 |
| 7 | ARPE | 自动重载预装载使能<br>0：TIMx_ARR 寄存器不进行缓冲。<br>1：TIMx_ARR 寄存器进行缓冲。 |
| 6:4 | 保留 | 必须保持复位值。 |
| 3 | OPM | 单脉冲模式<br>0：计数器在发生更新事件时不会停止计数<br>1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）。 |
| 2 | URS | 更新请求源<br>此位由软件置 1 和清零，用以选择 UEV 事件源。 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 0：失能时，所有以下事件都会生成更新中断或 DMA 请求。此类事件包括：<br>计数器上溢/下溢<br>将 UG 位置 1<br>通过从模式控制器生成的更新事件<br>1：使能时，只有计数器上溢/下溢会生成更新中断或 DMA 请求 |
| 1 | UDIS | 更新禁止<br>此位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。<br>0：使能 UEV。更新 (UEV) 事件可通过以下事件之一生成：<br>计数器上溢/下溢<br>将 UG 位置 1<br>通过从模式控制器生成的更新事件然后更新影子寄存器的值。<br>1：禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR 和 PSC）保持不变。但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。 |
| 0 | CEN | 计数器使能<br>0：禁止计数器<br>1：使能计数器<br>*注意：只有事先通过软件将 CEN 位置 1，才可以使用门控模式。而触发模式可通过硬件自动将 CEN 位置 1。*<br>*在单脉冲模式下，当发生更新事件时会自动将 CEN 位清零。* |

### 17.4.2. TIM6 和 TIM7 控制寄存器 2 (TIMx_CR2)

偏移地址：0x04

复位值：0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:7 | 保留 | 必须保持复位值。 |
| 6:4 | MMS | 主模式选择<br>这些位用于选择主模式下将要发送到从定时器以实现同步的信息 (TRGO)。这些位的组合如下：<br>000：复位 – IMx_EGR 寄存器中的 UG 位用作触发输出 (TRGO)。如果复位由触发输入生成（从模式控制器配置为复位模式），则 TRGO 上的信号相比实际复位会有延迟。<br>001：使能 – 数器使能信号 (CNT_EN) 用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号由 CEN 控制位与门控模式下的触发输入的逻辑或运算组合而成。<br>当计数器使能信号由触发输入控制时，TRGO 上会存在延迟，选择主/从模式时除外（请参见 TIMx_SMCR 寄存器中对 MSM 位的说明）<br>010：更新 – 选择更新事件作为触发输出 (TRGO)。例如，主定时器可用作从定时器的预分频器。 |
| 3:0 | 保留 | 必须保持复位值。 |

### 17.4.3. TIM6 和 TIM7 DMA/中断使能寄存器(TIMx_DIER)

偏移地址：0x0C

复位值：0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:9 | 保留 | 必须保持复位值。 |
| 8 | UDE | 更新 DMA 请求使能<br>0：禁止更新 DMA 请求。<br>1：使能更新 DMA 请求。 |
| 7:1 | 保留 | 必须保持复位值。 |
| 0 | UIE | 更新中断使能<br>0：禁止更新中断。<br>1：使能更新中断。 |

### 17.4.4. TIM6 和 TIM7 状态寄存器(TIMx_SR)

偏移地址：0x10

复位值：0x0000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | UIF |
| | | | | | | | | | | | | | | | rc_w0 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:1 | 保留 | 必须保持复位值。 |
| 0 | UIF | 更新中断标志<br>该位在发生更新事件时通过硬件置 1。但需要通过软件清零。<br>0：未发生更新。<br>1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：<br>上溢或下溢并且当 TIMx_CR1 寄存器中 UDIS = 0 时。<br>当由于 TIMx_CR1 寄存器中 URS = 0 且 UDIS = 0 而通过软件使用 TIMx_EGR 寄存器中 的 UG 位重新初始化 CNT 时。 |

### 17.4.5. TIM6 和 TIM7 事件生成寄存器 (TIMx_EGR)

偏移地址：0x14

复位值：0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:1 | 保留 | 必须保持复位值。 |
| 0 | UG | 更新生成<br>该位可通过软件置 1，并由硬件自动清零。<br>0：不执行任何操作。<br>1：重新初始化定时器计数器并生成寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。 |

### 17.4.6. TIM6 和 TIM7 计数器 (TIMx_CNT)

偏移地址：0x24

复位值：0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:0 | CNT[15:0] | 计数器值 |

### 17.4.7. TIM6 和 TIM7 预分频器 (TIMx_PSC)

偏移地址：0x28

复位值：0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:0 | PSC[15:0] | 预分频器值<br>计数器时钟频率 CK_CNT 等于 $f_{CK_PSC}$ / (PSC[15:0] + 1)。<br>PSC 包含在每次发生更新事件时要装载到实际预分频器寄存器的值。 |

### 17.4.8. TIM6 和 TIM7 自动重载寄存器 (TIMx_ARR)

偏移地址：0x2C

复位值：0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:0 | ARR[15:0] | 自动重载值<br>ARR 为要装载到实际自动重载寄存器的值。<br>当自动重载值为空时，计数器不工作。 |

# 18. 独立看门狗(IWDG)

## 18.1. IWDG 简介

此器件具有两个嵌入式看门狗外设，具有安全性高、定时准确及使用灵活的优点。两个看门狗外设（独立和窗口）均可用于检测并解决由软件错误导致的故障；当计数器达到给定的超 时值时，触发一个中断（仅适用于窗口型看门狗）或产生系统复位。

独立看门狗 (IWDG) 由其专用低速时钟 (LSI) 驱动，因此即便在主时钟发生故障时仍然保持工作状态。窗口看门狗 (WWDG) 时钟由 APB1 时钟经预分频后提供，通过可配置的时间窗口来检测应用程序非正常的过迟或过早的操作。

IWDG 最适合应用于那些需要看门狗作为一个在主程序之外，能够完全独立工作，并且对时间精度要求较低的场合。WWDG 最适合那些要求看门狗在精确计时窗口起作用的应用程序。

## 18.2. IWDG 主要特性

- 自由运行递减计数器。
- 时钟由独立 RC 振荡器提供（可在待机和停止模式下运行）。
- 当递减计数器值达到 0x000 时产生复位（如果看门狗已激活）。

## 18.3. IWDG 功能说明

下图给出了独立看门狗模块的功能框图。

当通过对关键字寄存器 (IWDG_KR) 写入值 0xCCCC 启动独立看门狗时，计数器开始从复位值 0xFFF 递减计数。当计数器计数到终值 (0x000) 时会产生一个复位信号（IWDG 复位）。

任何时候将关键字 0xAAAA 写到 IWWDG_KR 寄存器中，IWDG_RLR 的值就会被重载到计数器，从而避免产生看门狗复位。

### 18.3.1. 硬件看门狗

如果通过器件选项位使能“硬件看门狗”功能，上电时将自动使能看门狗；如果在计数器计数结束前，若软件没有向关键字寄存器写入相应的值，则系统会产生复位。

### 18.3.2. 寄存器访问保护

IWDG_PR 和 IWDG_RLR 寄存器具有写访问保护。若要修改寄存器，必须首先对 IWDG_KR 寄存器写入代码 0x5555。而写入其他值则会破坏该序列，从而使寄存器访问保护再次生效。这意味着重装载操作（即写入 0xAAAA）也会启动写保护功能。

状态寄存器指示预分频值和递减计数器是否正在被更新。

### 18.3.3. 调试模式

当微控制器进入调试模式(内核停止)时，IWDG 计数器会根据 DBG 模块中的 DBG_IWDG_STOP 配置位选择继续正常工作或者停止工作。

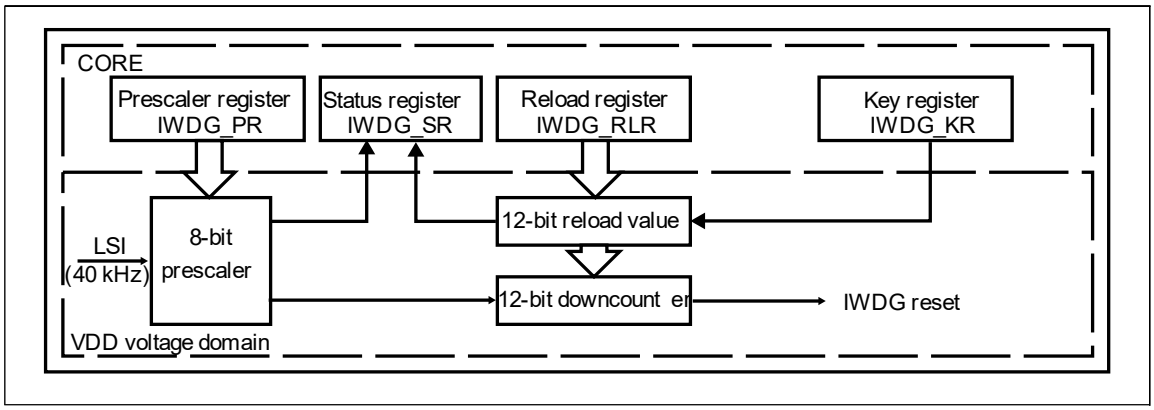



图 18-1 独立看门狗框图

*注意：看门狗功能由 VDD 电压域供电，在停止模式和待机模式下是否工作可以通过寄存器 PWR_CSR 寄存器配置。*

表 18-1 32 kHz (LSI) 频率条件下 IWDG 超时周期的最小值/最大值(1)

| 预分频器 | PR[2:0] 位 | 最短超时 (ms)<br>RL[11:0]= 0x000 | 最长超时 (ms)<br>RL[11:0]= 0xFFF |
|---|---|---|---|
| /4 | 0 | 0.125 | 512 |
| /8 | 1 | 0.25 | 1024 |
| /16 | 2 | 0.5 | 2048 |
| /32 | 3 | 1 | 4096 |
| /64 | 4 | 2 | 8192 |
| /128 | 5 | 4 | 16384 |
| /256 | 6 | 8 | 32768 |

1. *这些时间均针对 32 kHz 时钟给出。实际上 MCU 内部的 RC 频率会在 30kHz 到 60kHz 之间变化。此外，即使 RC 振荡器的频率是精确的，确切的时序仍然依赖于 APB 接口时钟与 RC 振荡器时钟之间的相位差，因此总会有一个完整的 RC 周期是不确定的。*

## 18.4. IWDG 寄存器

有关寄存器说明中使用的缩写，请参见第 1.1 节。

外设寄存器可支持半字（16 位）或字（32 位）访问。

### 18.4.1. 关键字寄存器 (IWDG_KR)

偏移地址：0x00

复位值：0x0000 0000（通过待机模式复位）

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15:0 | KEY[15:0] | 键值（只写位，读为 0000h）<br>必须每隔一段时间便通过软件对这些位写入键值 AAAAh，否则当计数器计数 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 到 0 时，看门狗会产生复位。<br>写入键值 5555h 可使能对 IWDG_PR 和 IWDG_RLR 寄存器的访问<br>写入键值 CCCCh 可启动看门狗 |

### 18.4.2. 预分频器寄存器 (IWDG_PR)

偏移地址：0x04

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:3 | 保留 | 必须保持复位值。 |
| 2:0 | PR[2:0] | 预分频器<br>这些位受写访问保护，请参见第 18.3.2 节。通过软件设置这些位来选择计数器时钟的预分频因子。若要更改预分频器的分频系数，IWDG_SR 的 PVU 位必须为 0。<br>000：4 分频<br>001：8 分频<br>010：16 分频<br>011：32 分频<br>100：64 分频<br>101：128 分频<br>110：256 分频<br>111：256 分频<br>*注意：读取该寄存器会当前的预分频器值。如果正在对该寄存器执行写操作，则读取的值可能不是最新的/ 有效的。因此，只有在 IWDG_SR 寄存器中的 PVU 位为 0 时，从寄存器读取的值才有效。* |

### 18.4.3. 重载寄存器 (IWDG_RLR)

偏移地址： 0x08

复位值： 0x0000 0FFF（待机模式时复位）

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:12 | 保留 | 必须保持复位值。 |
| 11:0 | RL[11:0] | 看门狗计数器重载值<br>这些位受写访问保护，这个值由软件设置，每次对 IWDR_KR 寄存器写 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 入值 AAAAh 时，这个值就会重装载到看门狗计数器中。之后，看门狗计数器便从该装载的值开始递减计数。超时周期由该值和时钟预分频器共同决定。<br>若要更改重载值，IWDG_SR 中的 RVU 位必须为 0。<br>*注意： 读取该寄存器会返回当前的重载值。如果正在对该寄存器执行写操作，则读取的值可能不是最新的/有效的。因此，只有在 IWDG_SR 寄存器中的 RVU 位为 0 时，从寄存器读取的值才有效。* |

### 18.4.4. 状态寄存器 (IWDG_SR)

偏移地址： 0x0C

复位值： 0x0000 0000（待机模式时不复位）

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:2 | 保留 | 必须保持复位值。 |
| 1 | RVU | 看门狗计数器重载值更新<br>可通过硬件将该位置 1 以指示重载值正在更新。当在完成重载值更新操作后，会通过硬件将该位复位。<br>重载值只有在 RVU 位为 0 时才可更新。 |
| 0 | PVU | 看门狗预分频器值更新<br>可通过硬件将该位置 1 以指示预分频器值正在更新。当完成预分频器值更新操作后，会通过硬件将该位复位。<br>预分频器值只有在 PVU 位为 0 时才可更新。 |

*注意：如果应用使用多个重载值或预分频器值，则必须等到 RVU 位被清零后才能更改重载值，而且必须等到 PVU 位被清零后才能更改预分频器值。但是，在更新预分频器和/或重载值之后，则无需等到 RVU 或 PVU 复位后再继续执行代码（即便进入低功耗模式，也会继续执行写操作至完成）*

# 19. 窗口看门狗(WWDG)

## 19.1. WWDG 简介

窗口看门狗通常被用来监测，由外部干扰或不可预见的逻辑条件造成的应用程序背离正常的运行序列而产生的软件故障。除非递减计数器的值在 T6 位变成 0 前被刷新，看门狗电路在达到预置的时间周期时，会产生一个 MCU 复位。如果在递减计数器达到窗口寄存器值之前刷新控制寄存器中的 7 位递减计数器值，也会产生 MCU 复位。这意味着必须在限定的时间窗口内刷新计数器。

## 19.2. WWDG 主要特性

- 可编程的自由运行递减计数器；
- 复位条件；
- 当递减计数器值小于 0x40 时复位（如果看门狗已激活）；
- 在窗口之外重载递减计数器时复位（如果看门狗已激活）；
- 提前唤醒中断 (EWI)：当递减计数器等于 0x40 时触发（如果已使能且看门狗已激活）。

## 19.3. WWDG 功能说明

如果激活看门狗（WWDG_CR 寄存器中的 WDGA 位置 1），则当 7 位递减计数器（T[6:0] 位）从 0x40 滚动到 0x3F（T6 已清零）时会引发复位。当计数器值大于窗口寄存器中所存储的值时，如果软件重载计数器，则会产生复位。

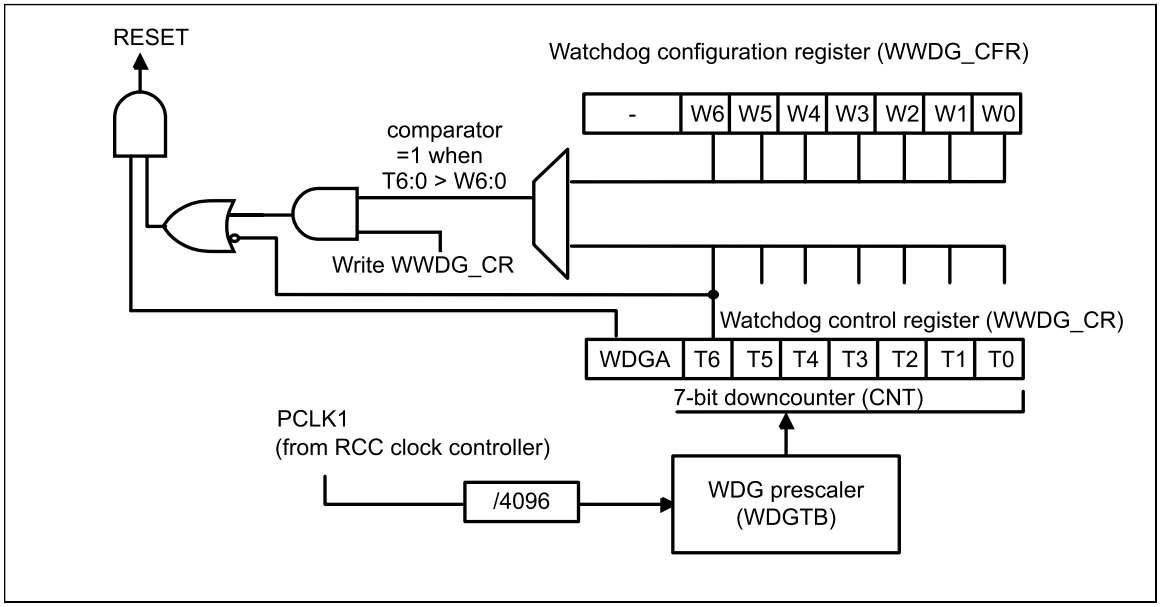



图 19-1 看门狗框图

应用程序在正常运行过程中必须定期地写入 WWDG_CR 寄存器以防止 MCU 发生复位。只有当计数器值低于窗口寄存器值时，才能执行此操作。存储在 WWDG_CR 寄存器中的值必须介于 0xFF 和 0xC0 之间：

**使能看门狗**

在系统复位后，看门狗总是处于关闭状态。可通过设置 WWDG_CR 寄存器中的 WDGA 位来使能看门狗，之后除非执行复位操作，否则不能再次关闭。

**控制递减计数器**

递减计数器处于自由运行状态：即使禁止看门狗，递减计数器仍继续递减计数。当使能看门狗时，必须将 T6 位置 1，以防止立即复位。

T[5:0] 位包含了看门狗产生复位之前的计时数目；复位前的延时时间在一个最小值和一个最大值之间变化，这是因为写入 WWDG_CR 寄存器时，预分频值是未知的。配置寄存器 (WWDG_CFR) 包含窗口的上限：为防止发生复位，当递减计数器的值低于窗口寄存器值且大于 0x3F 时必须重载。 下图介绍了窗口看门狗的工作过程。

*注意： 可使用 T6 位产生软件复位（将 WDGA 位置 1 并将 T6 位清零）。*

**看门狗中断高级特性**

如果在产生实际复位之前必须执行特定的安全操作或数据记录，则可使用提前唤醒中断(EWI)。通过设置 WWDG_CFR 寄存器中的 EWI 位使能 EWI 中断。当递减计数器的值为 0x40 时，将生成 EWI 中断。在复位器件之前，可以使用相应的中断服务程序 (ISR) 来触发特定操作（例如通信或数据记录）。

在某些应用中，可以使用 EWI 中断来管理软件系统检查和/或系统恢复/功能退化，而不会生成 WWDG 复位。在这种情况下，相应的中断服务程序 (ISR) 可用来重载 WWDG 计数器以避免 WWDG 复位，然后再触发所需操作。

通过将 0 写入 WWDG_SR 寄存器中的 EWIF 位来清除 EWI 中断。

*注意： 当由于在更高优先级任务中有系统锁定而无法使用 EWI 中断时，最终会产生 WWDG 复位。*

## 19.4. 如何设置看门狗超时

可以使用下图中的公式来计算 WWDG 超时。

*警告：写入 WWDG_CR 寄存器时，始终将 1 写入 T6 位，以避免生成立即复位。*

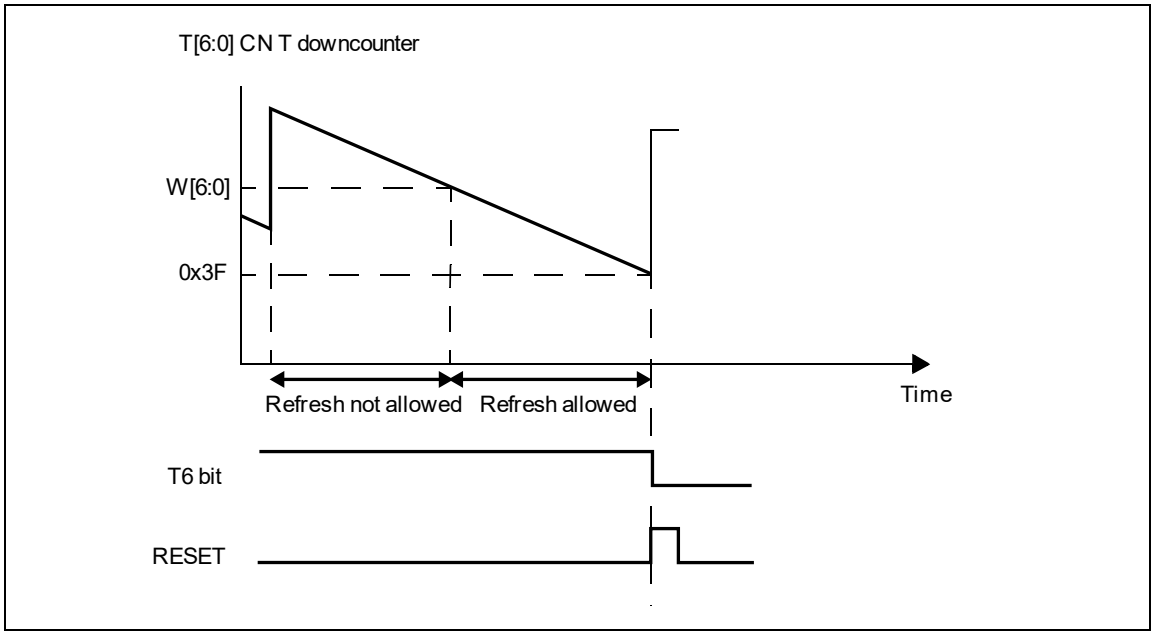



图 19-2 窗口看门狗时序图

超时值的计算公式如下：

$$t_{WWDG} = t_{PCLK1} \times 4096 \times 2^{WDGTB[1:0]} \times (T[5:0] + 1) \quad (ms)$$

其中：

$t_{WWDG}$ ： WWDG 超时

$t_{PCLK1}$ ： APB1 时钟周期，以 ms 为测量单位

有关 TWWDG 的最小值和最大值，请参见下表。

表 19-1 30 MHz ($f_{PCLK1}$) 时的超时值

| 预分频器 | WDGTB | 最小超时 (µs)<br>T[5:0] = 0x00 | 最大超时 (ms)<br>T[5:0] = 0x3F |
|---|---|---|---|
| 1 | 0 | 136.53 | 8.74 |
| 2 | 1 | 273.07 | 17.48 |
| 4 | 2 | 546.13 | 34.95 |
| 8 | 3 | 1092.27 | 69.91 |

## 19.5. 调试模式

当微控制器进入调试模式(内核停止）时，WWDG 计数器会根据 DBG 模块中的 DBG_WWDG_STOP 配置位选择继续正常工作或者停止工作。

## 19.6. WWDG 寄存器

有关寄存器说明中使用的缩写，请参见第 1.1 节。

外设寄存器可支持半字（16 位）或字（32 位）访问。

### 19.6.1. 控制寄存器 (WWDG_CR)

偏移地址：0x00

复位值：0x0000 007F

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | WDGA | T[6:0] | | | | | | |
| | | | | | | | | rs | rw | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 必须保持复位值。 |
| 7 | WDGA | 激活位<br>此位由软件置 1，只有复位后才由硬件清零。当 WDGA = 1 时，看门狗可产生复位。<br>0：禁止看门狗<br>1：使能看门狗 |
| 6:0 | T[6:0] | 7 位计数器 (MSB 到 LSB)<br>这些位用来存储看门狗计数器的值。它每隔 (4096 x $2^{WDGTB[1:0]}$) PCLK1 个周期递减一次。当它从 0x40 滚动到 0x3F (T6 清零) 时会产生复位。 |

### 19.6.2. 配置寄存器 (WWDG_CFR)

偏移地址： 0x04

复位值： 0x0000 007F

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | EWI | WDGTB[1:0] | | W[6:0] | | | | | | |
| | | | | | | rs | rw | | rw | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:10 | 保留 | 必须保持复位值。 |
| 9 | EWI | 提前唤醒中断<br>置 1 后，只要计数器值达到 0x40 就会产生中断。此中断只有在复位后才由硬件清零。 |
| 8:7 | WDGTB[1:0] | 定时器时基<br>可按如下方式修改预分频器的时基：<br>00：CK 计数器时钟 (PCLK1 div 4096) 分频器 1<br>01：CK 计数器时钟 (PCLK1 div 4096) 分频器 2<br>10：CK 计数器时钟 (PCLK1 div 4096) 分频器 4<br>11：CK 计数器时钟 (PCLK1 div 4096) 分频器 8 |
| 6:0 | W[6:0] | 7 位窗口值<br>这些位包含用于与递减计数器进行比较的窗口值。 |

### 19.6.3. 状态寄存器 (WWDG_SR)

偏移地址： 0x08

复位值： 0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | EWIF |
| | | | | | | | | | | | | | | | rc_w0 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:1 | 保留 | 必须保持复位值。 |
| 0 | EWIF | 提前唤醒中断标志<br>当计数器值达到 0x40 时此位由硬件置 1。它必须由软件通过写入 0 来清零。写入 1 不起作用。如果不使能中断，此位也会被置 1。 |

# 20. 内部集成电路接口(I2C)

## 20.1. I2C 简介

I2C（内部集成电路）总线接口用作微控制器和 I2C 串行总线之间的接口。它提供多主模式功能，可以控制所有 I2C 总线特定的序列、协议、仲裁和时序。它支持标准和快速模式。它还与 SMBus 2.0 兼容。

它可以用于多种用途，包括 CRC 生成和验证、SMBus（系统管理总线）以及 PMBus（电源管理总线）。

根据器件的不同，可利用 DMA 功能来减轻 CPU 的工作量。

## 20.2. I2C 主要特性

- 并行总线/I2C 协议转换器；
- 多主模式功能：同一接口既可用作主模式也可用作从模式；
- I2C 主模式特性：
  - 时钟生成；
  - 起始位和停止位生成；
- I2C 从模式特性：
  - 可编程 I2C 地址检测；
  - 双寻址模式，可对 2 个从地址应答；
  - 停止位检测；
- 7 位/10 位寻址以及广播呼叫的生成和检测；
- 支持不同的通信速度：
  - 标准速度（高达 100 kHz）；
  - 快速速度（高达 400 kHz）；
- 状态标志：
  - 发送/接收模式标志；
  - 字节传输结束标志；
  - I2C 忙碌标志；
- 错误标志：
  - 主模式下的仲裁丢失情况；
  - 地址/数据传输完成后的应答失败；
  - 检测误放的起始位和停止位；
  - 禁止时钟延长后出现的上溢/下溢；
- 2 个中断向量：
  - 一个中断由成功的地址/数据字节传输事件触发；
  - 一个中断由错误状态触发；
- 可选的时钟延长；
- 带 DMA 功能的 1 字节缓冲；

- 可配置的 PEC（数据包错误校验）生成或验证：
  - 在 Tx 模式下，可将 PEC 值作为最后一个字节进行传送；
  - 针对最后接收字节的 PEC 错误校验；
- SMBus 2.0 兼容性：
  - 25 ms 时钟低电平超时延迟；
  - 10 ms 主器件累计时钟低电平延长时间；
  - 25 ms 从器件累计时钟低电平延长时间；
  - 具有 ACK 控制的硬件 PEC 生成/验证；
  - 支持地址解析协议 (ARP)；
- PMBus 兼容性。

## 20.3. I2C 功能说明

除了接收和发送数据之外，此接口还可以从串行格式转换为并行格式，反之亦然。中断由软件使能或禁止。该接口通过数据引脚 (SDA) 和时钟引脚 (SCL) 连接到 I2C 总线。它可以连接到标准（高达 100 kHz）或快速（高达 400 kHz）I2C 总线。

### 20.3.1. 模式选择

该接口在工作时可选用以下四种模式之一：

- 从发送器
- 从接收器
- 主发送器
- 主接收器

默认情况下，它以从模式工作。接口在生成起始位后会自动由从模式切换为主模式，并在出现仲裁丢失或生成停止位时从主模式切换为从模式，从而实现多主模式功能。

#### 通信流程

在主模式下，I2C 接口会启动数据传输并生成时钟信号。串行数据传输始终是在出现起始位时开始，在出现停止位时结束。起始位和停止位均在主模式下由软件生成。

在从模式下，该接口能够识别其自身地址（7 或 10 位） 以及广播呼叫地址。广播呼叫地址检测可由软件使能或禁止。

数据和地址均以 8 位字节传输，MSB 在前。起始位后紧随地址字节（7 位地址占据一个字节； 10 位地址占据两个字节）。地址始终在主模式下传送。

在字节传输 8 个时钟周期后是第 9 个时钟脉冲，在此期间接收器必须向发送器发送一个应答位。详见下图：

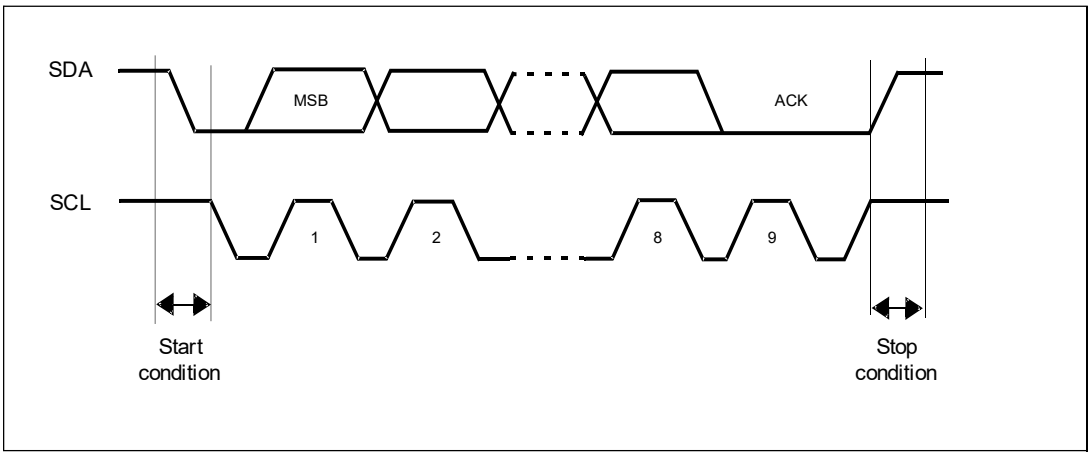



图 20-1 I2C 总线协议

应答位可由软件使能或禁止。I2C 接口地址（7 位/10 位双寻址模式和/ 或广播呼叫地址）可通过软件进行选择。

I2C 接口框图如下:

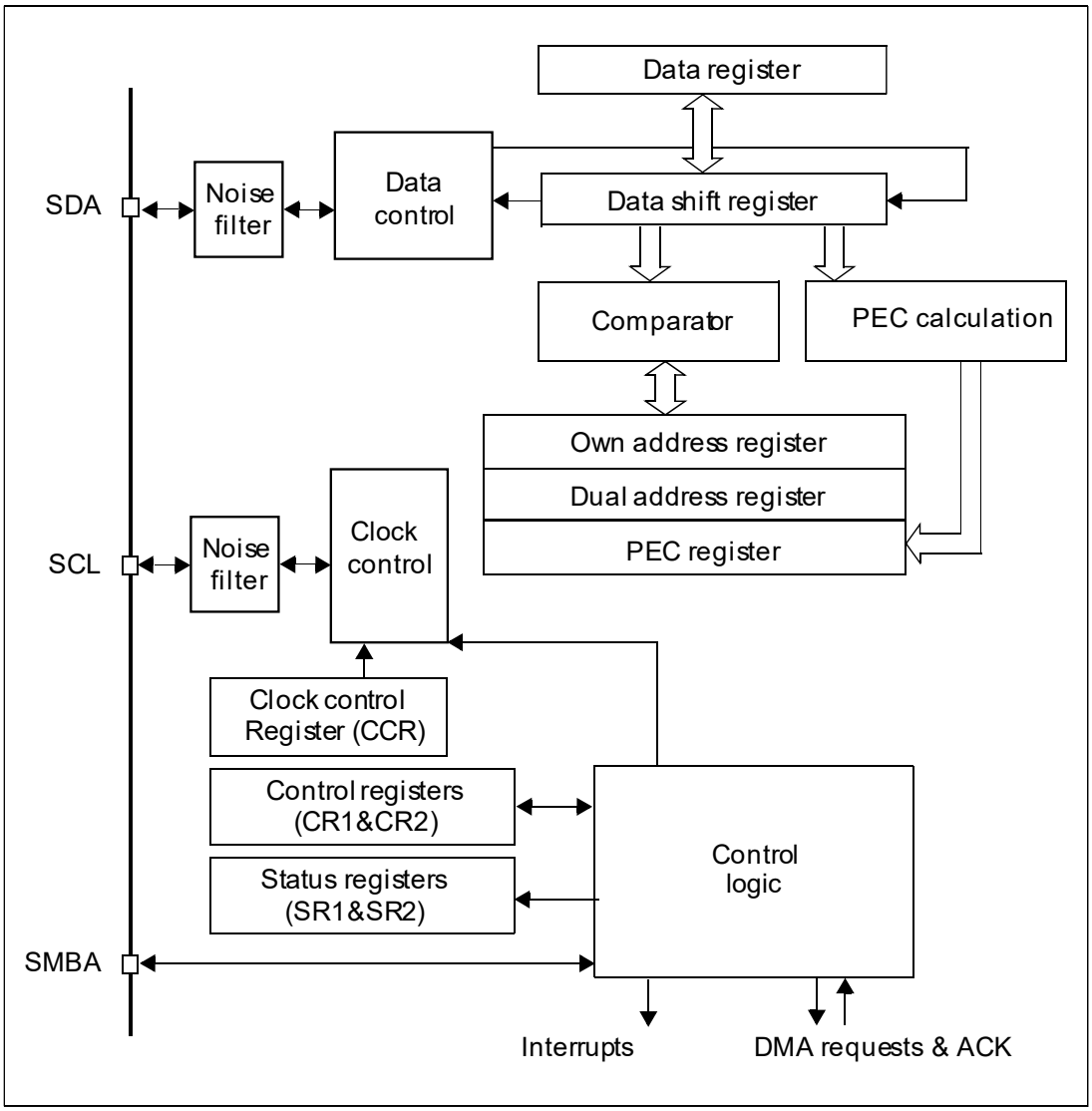



图 20-2 I2C 框图

### 20.3.2. I2C 从模式

默认情况下，I2C 接口在从模式下工作。要将工作模式由默认的从模式切换为主模式，需要生成一个起始位。

为了生成正确的时序，必须在 I2C_CR2 寄存器中对外设输入时钟进行编程。外设输入时钟频率的下限为：

- 标准模式下 2 MHz
- 快速模式下 4 MHz

检测到起始位后，便会立即接收到来自 SDA 线的地址并将其送到移位寄存器。之后，会将其与接口地址 (OAR1) 和 OAR2（如果 ENDUAL=1）或者广播呼叫地址（如果 ENGC = 1）进行比较。

**注意：** 在 10 位寻址模式下，比较对象还包括头序列(11110xx0)，其中，xx 表示该地址的两个最高有效位。

**头或地址不匹配：** 接口会忽略它并等待下一个起始位。

**头匹配**(仅针对 10 位模式)：如果 ACK 位置 1，则接口会生成一个应答脉冲并等待 8 位从地址。

**地址匹配**：接口会依次：

- 发出应答脉冲（如果 ACK 位置 1）。
- ADDR 位会由硬件置 1 并在 ITEVFEN 位置 1 时生成一个中断。
- 如果 ENDUAL=1，则软件必须读取 DUALF 位状态来核对哪些从地址进行了应答。

在 10 位模式下，完成地址序列接收后，从模式始终处于接收模式。在接收到重复起始位以及一个匹配地址位和最低有效位均置 1 的头序列 (11110xx1) 后，它会进入发送模式。

TRA 位指示从设备是处于接收模式还是处于发送模式。

#### 从发送器

在接收到地址并将 ADDR 清零后，从设备会通过内部移位寄存器将 DR 寄存器中的字节发送到 SDA 线。

从设备会延长 SCL 低电平时间，直到 ADDR 位清零且 DR 寄存器中填满待发送数据为止。

接收到应答脉冲时：

- TxE 位会由硬件置 1 并在 ITEVFEN 和 ITBUFEN 位均置 1 时生成一个中断。

如果在下一次数据传输结束之前 TxE 位已置 1 但某些数据尚未写入 I2C_DR 寄存器，则 BTF 位会置 1，而接口会一直延长 SCL 低电平，直到通过软件对 I2C_SR1 读操作，以及对 I2C_DR 写操作后，把 BTF 清零为止。

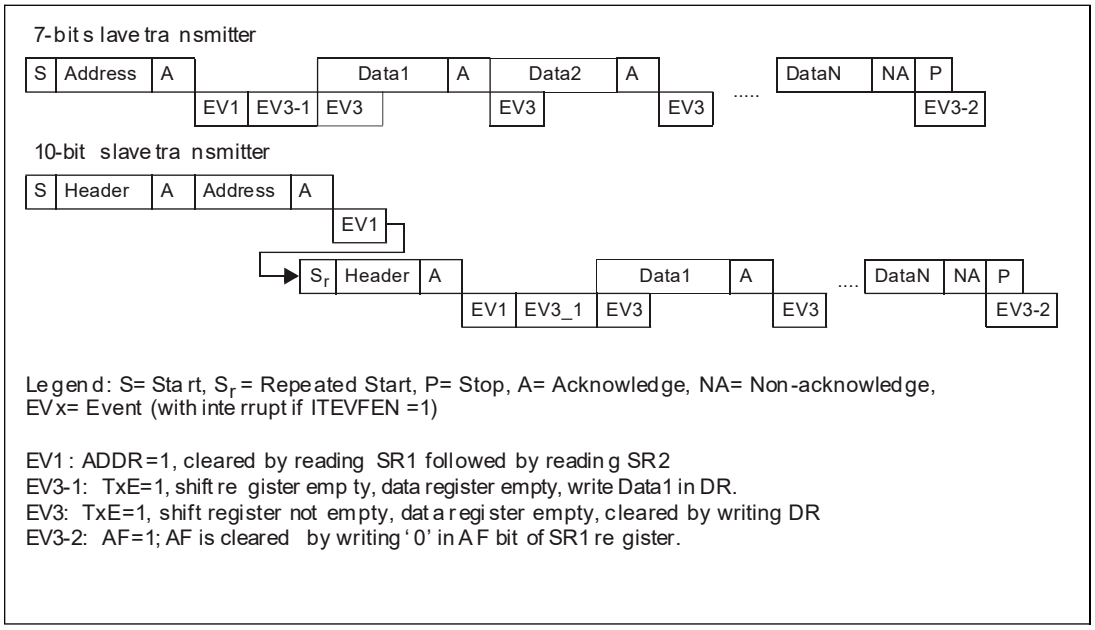



图 20-3 从发送器的传输时序图

1. EV1 和 EV3_1 事件可延长 SCL 低电平时间，直到相应的软件序列结束为止。
2. 如果软件序列在下一个字节传输结束之前未能完成，EV3 事件会延长 SCL 低电平时间。

**从接收器**

在接收到地址并将 ADDR 位清零后，从设备会通过内部移位寄存器接收 SDA 线中的字节并将其保存到 DR 寄存器。在每个字节接收完成后，接口都会依次：

- 发出应答脉冲（如果 ACK 位置 1）。
- RxNE 位会由硬件置 1 并在 ITEVFEN 和 ITBUFEN 位均置 1 时生成一个中断。

如果在下一次数据接收结束之前 RxNE 位已置 1 但 DR 寄存器中的数据尚未读取，则 BTF 位会置 1，而接口会一直延长 SCL 低电平，直到软件通过读取 I2C_DR 寄存器来把 BTF 清零。

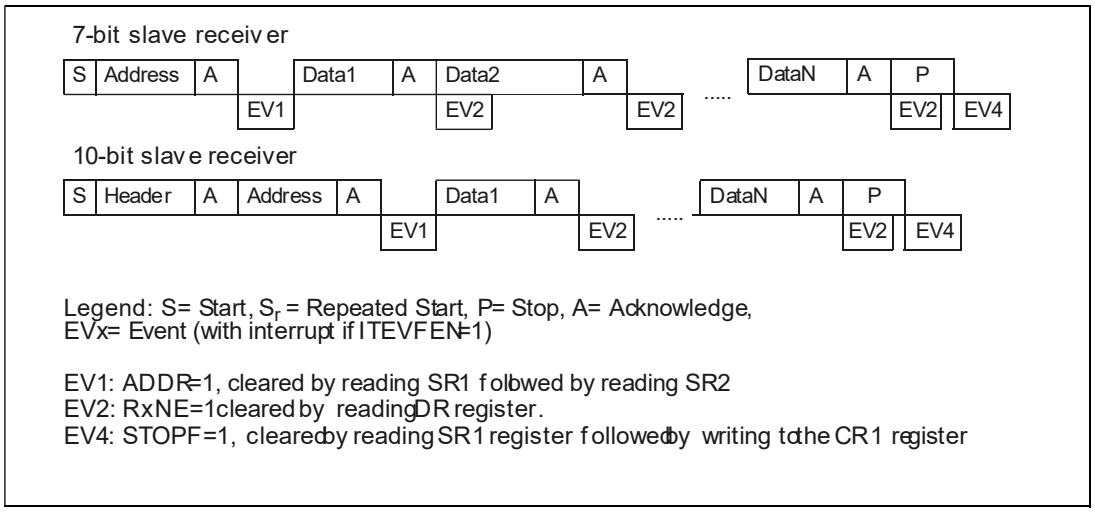



图 20-4 从接收器的传输时序图

1. EV1 事件可延长 SCL 低电平时间，直到相应的软件序列结束为止。

2. 如果软件序列在下一个字节接收结束之前未能完成， EV2 事件会延长 SCL 低电平时间。
3. 检查了 SR1 寄存器内容之后，用户应针对各个被置位的标志执行完整的清零序列。

对于 ADDR 和 STOPF 标志，需要在 I2C 中断程序中执行以下序列：

对 SR1 执行读操作

if (ADDR == 1) {对 SR1 执行读操作；对 SR2 执行读操作}

if (STOPF == 1) {对 SR1 执行读操作；对 CR1 执行写操作}

这样做的目的是为了确保 ADDR 和 STOPF 这两个标志都清零（如果二者均为被置位）。

**关闭从设备通信**

传输完最后一个数据字节之后，主设备会生成一个停止位。接口会检测此条件并：

- 将 STOPF 位置 1 并在 ITEVFEN 位置 1 时生成一个中断。

通过先读取 SR1 寄存器然后写入 CR1 寄存器的方式将 STOPF 位清零。

### 20.3.3. I2C 主模式

在主模式下，I2C 接口会启动数据传输并生成时钟信号。串行数据传输始终是在出现起始位时开始，在出现停止位时结束。只要通过 START 位在总线上生成了起始位，即会选中主模式。

在主模式中要求执行以下序列。

- 在 I2C_CR2 寄存器中对外设输入时钟进行编程，以生成正确的时序；
- 配置时钟控制寄存器；
- 配置上升时间寄存器；
- 对 I2C_CR1 寄存器进行编程，以便使能外设；
- 将 I2C_CR1 寄存器的 START 位置 1，以生成起始位。

外设输入时钟频率的下限为：

- 标准模式下 2 MHz；
- 快速模式下 4 MHz。

**起始位**

START 位置 1 后，接口会在 BUSY 位清零后生成一个起始位并切换到主模式（M/SL 位置 1）。

**注意：** 在主设备下，START 位置 1 后，接口会在当前字节传输结束后生成一个重复起始位。

起始位发出之后：

- SB 位会由硬件置 1 并在 ITEVFEN 位置 1 时生成一个中断。
- 接下来主设备会等待软件对 SR1 执行读操作，然后把从设备地址写入 DR 寄存器。

**从地址传输**

接下来从地址会通过内部移位寄存器发送到 SDA 线。

- 在 10 位寻址模式中，发送头序列会产生以下事件：
  - ADD10 位会由硬件置 1 并在 ITEVFEN 位置 1 时生成一个中断。
    接下来主设备会等待软件读取 SR1 ，然后把第二个地址字节写入 DR 寄存器（请参见传输序列）。

- ADDR 位会由硬件置 1 并在 ITEVFEN 位置 1 时生成一个中断。
  - 接下来主设备会等待对 SR1 寄存器执行读操作，然后对 SR2 寄存器执行读操作（请参见传输序列）。
- 在 7 位寻址模式下，会发送一个地址字节。
  地址字节被发出后，
  - ADDR 位会由硬件置 1 并在 ITEVFEN 位置 1 时生成一个中断。
    接下来主设备会等待对 SR1 寄存器执行读操作，然后对 SR2 寄存器执行读操作（请参见传输序列）。

主设备会根据发送的从地址字节 LSB 来决定是进入发送模式还是接收模式。

- 在 7 位寻址模式下，
  - 要进入发送模式，主设备会发送从地址并将 LSB 复位。
  - 要进入接收模式，主设备会发送从地址并将 LSB 置 1。
- 在 10 位寻址模式下，
  - 要进入发送模式，主设备会先发送头序列 (11110xx0)，然后发送从地址（其中 xx 表 示该地址的两个最高有效位）。
  - 要进入接收模式，主设备会先发送头序列 (11110xx0)，然后发送从地址。接下来会发送一个重复起始位，然后再发送头序列 (11110xx1)（其中 xx 表示地址的两个最高有效位）。

TRA 位指示主设备是处于接收模式还是处于发送模式。

**主发送器**

在发送出地址并将 ADDR 清零后，主设备会通过内部移位寄存器将 DR 寄存器中的字节发送到 SDA 线。

主设备会一直等待，直到首个数据字节被写入 I2C_DR 为止。

接收到应答脉冲后，TxE 位会由硬件置 1 并在 ITEVFEN 和 ITBUFEN 位均置 1 时生成一个中断。

如果在上一次数据传输结束之前 TxE 位已置 1 但数据字节尚未写入 DR 寄存器，则 BTF 位会置 1，而接口会一直延长 SCL 低电平，等待 I2C_DR 寄存器被写入，以将 BTF 清零。

**结束通信**

当最后一个字节写入 DR 寄存器后，软件会将 STOP 位置 1 以生成一个停止位。接口会自动返回从模式（M/SL 位清零）。

**注意：** 当 TxE 或 BTF 中的任何一个置 1 时，应在 EV8_2 事件期间对停止位进行编程。

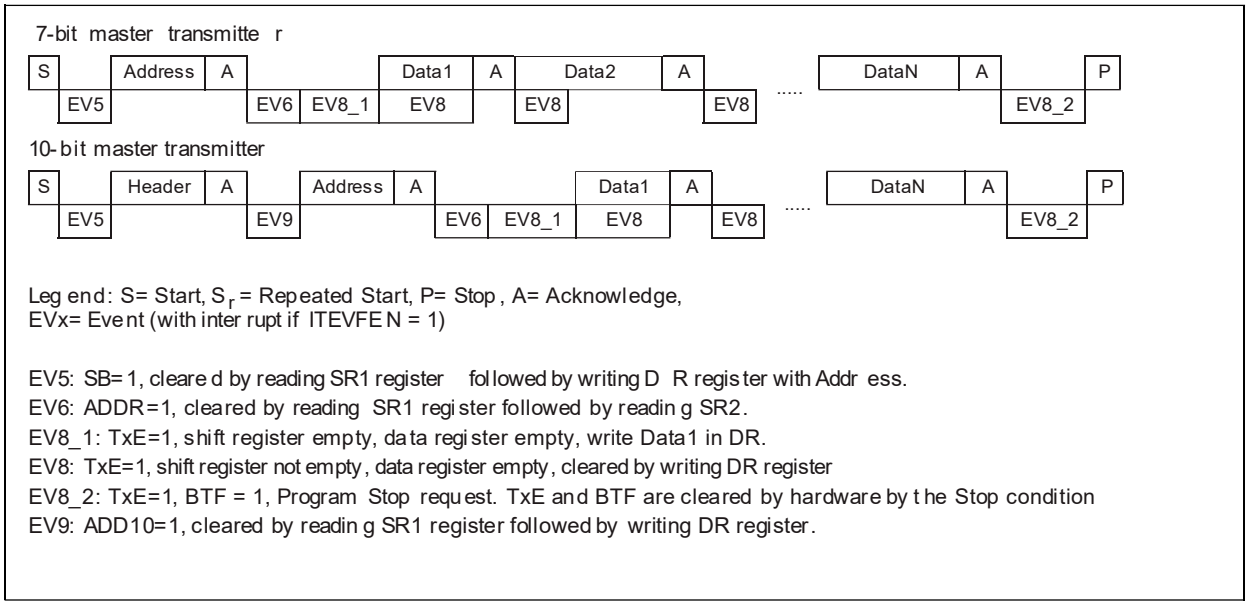



图 20-5 主发送器的传输时序图

1. EV5、EV6、EV9、EV8_1 和 EV8_2 事件可延长 SCL 低电平时间，直到相应的软件序列结束为止。
2. 如果软件序列在下一个字节传输结束之前未能完成，EV8 事件会延长 SCL 低电平时间。

### 主接收器

完成地址传输并将 ADDR 位清零后，I2C 接口会进入主接收模式。在此模式下，接口会通过内部移位寄存器接收 SDA 线中的字节并将其保存到 DR 寄存器。在每个字节传输结束后，接口都会依次:

1. 发出应答脉冲（如果 ACK 位置 1）。
2. RxNE 位置 1 并在 ITEVFEN 和 ITBUFEN 位均置 1 时生成一个中断。

如果在上一次数据接收结束之前 RxNE 位已置 1 但 DR 寄存器中的数据尚未读取，则 BTF 位会由硬件置 1，而接口会一直延长 SCL 低电平，等待 I2C_DR 寄存器被写入，以将 BTF 清零。

### 结束通信

主设备会针对自从设备接收的最后一个字节发送 NACK。在接收到此 NACK 之后，从设备会释放对 SCL 和 SDA 线的控制。随后，主设备可发送一个停止位/重复起始位。

1. 为了在最后一个接收数据字节后生成非应答脉冲，必须在读取倒数第二个数据字节后（倒数第二个 RxNE 事件之后）立即将 ACK 位清零。
2. 要生成停止位/重复起始位，软件必须在读取倒数第二个数据字节后（倒数第二个 RxNE 事件之后）将 STOP/START 位置 1。
3. 在只接收单个字节的情况下，会在 EV6 期间（在 ADDR 标志清零之前）禁止应答并在 EV6 之后生成停止位。

生成停止位后，接口会自动返回从模式（M/SL 位清零）。

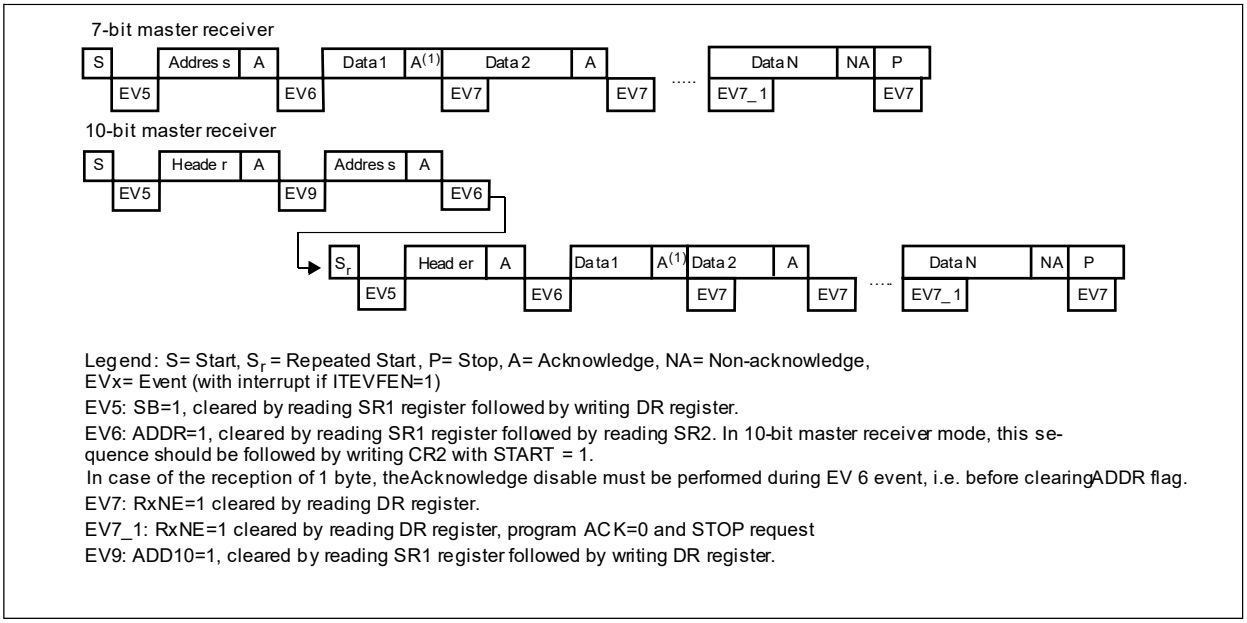



图 20-6 主接收器的传输时序图

1. 如果接收单个字节，则为 NA。
2. EV5、EV6 和 EV9 事件可延长 SCL 低电平时间，直到相应的软件序列结束为止。
3. 如果软件序列在下一个字节接收结束之前未能完成，EV7 事件会延长 SCL 低电平时间。
4. EV7_1 软件序列必须在当前字节传输的 ACK 脉冲之前结束。

如果 EV7_1 软件序列未能在当前字节传输的 ACK 脉冲之前结束，建议采用下列步骤。

必须遵循以下步骤以确保：

- 在结束上一个数据接收之前及时将 ACK 位复位
- 在上一次数据接收后将 STOP 位置位且不接收补充数据。

**针对 2 字节的接收：**

- 等待 ADDR = 1（SCL 保持低电平，直到 ADDR 标志清零）；
- 将 ACK 复位，POS 置位；
- 将 ADDR 标志清零；
- 等待 BTF = 1（数据 1 在 DR 中，数据 2 在移位寄存器中，SCL 保持低电平，直到读取了数据 1 为止）；
- 将 STOP 置位；
- 读取数据 1 和数据 2。

**针对 N > 2 的字节接收，从 N-2 数据接收开始**

- 等待 BTF = 1（数据 N-2 在 DR 中，数据 N-1 在移位寄存器中，SCL 保持低电平，直到读取了数据 N-2 为止）；
- 将 ACK 复位；
- 读取数据 N-2；
- 等待 BTF = 1（数据 N-1 在 DR 中，数据 N 在移位寄存器中，SCL 保持低电平，直到读取了数据

N-1 为止）；

- 将 STOP 置位；
- 读取数据 N-1 和数据 N。

### 20.3.4. 错误条件

以下错误条件可能导致通信失败。

#### 总线错误 (BERR)

当 I2C 接口在传输地址或数据期间检测到外部停止位或起始位时，会出现此错误。在这种情况下：

- BERR 位置 1 并在 ITERREN 位置 1 时生成一个中断；
- 在从模式下：硬件会丢弃数据并释放数据线：
  - 在误放起始位的情况下，从模式会认为它是重复起始位并会等待一个地址或停止位；
  - 在误放停止位的情况下，从模式会按停止位操作且硬件会释放数据线；
- 在主模式下：不会释放数据线且不会影响当前传输状态。由软件决定是否中止当前传输。

#### 应答失败 (AF)

当接口检测到未应答脉冲会出现此错误。在这种情况下：

- AF 位置 1 并在 ITERREN 位置 1 时生成一个中断；
- 接收到 NACK 的发送器必须复位通信：
  - 在从模式下：硬件释放数据线；
  - 在主模式下：必须由软件生成一个停止位或重复起始位。

#### 仲裁丢失 (ARLO)

当 I2C 接口检测到仲裁丢失时会出现此错误。在这种情况下，

- ARLO 位会由硬件置 1（并在 ITERREN 位置 1 时生成一个中断）。
- I2C 接口会自动返回从模式（M/SL 位清零）。I2C 失去仲裁时，将无法在本次传输中对自身从地址进行应答，但是能够在赢得仲裁的主设备发出重复起始位后对自身设备进行应答。
- 硬件释放数据线。

#### 上溢/下溢错误 (OVR)

当时钟延长已禁止且 I2C 接口正在接收数据时，从模式中可能出现上溢错误。接口已经收到一个字节 (RxNE=1)，但是收到下一个字节之前 DR 中的数据未被读走。在这种情况下，

- 会丢失接收的最后一个字节。
- 在出现上溢错误时，软件应将 RxNE 位清零，而发送器应重新发送最后一个字节。

当时钟延长已禁止且 I2C 接口正在发送数据时，从模式中可能出现下溢错误。下一个字节的时钟信号到来之前，从设备还未把下一个要发送的数据写进 DR（TxE=1）。在这种情况下，

- 会再次发送 DR 寄存器中的字节。
- 用户应确保在出现下溢错误期间从接收器端接收到的数据被丢弃，并确保在 I2C 总线标准指定的时钟低电平时间内写入下一个字节。

对于要传送的首个字节，必须在 ADDR 清零之后且出现首个 SCL 上升沿之前将其写入 DR 寄存器。否则，接收器必须丢弃首个数据。

### 20.3.5. 可编程噪声滤波器

在快速模式下， I2C 标准要求将 SDA 和 SCL 线上尖峰脉宽在 50 ns 以内的噪声都抑止掉。

SDA 和 SCL I/O 中采用了模拟噪声滤波器。此滤波器默认为使能，通过将 I2C_FLTR 寄存器中的 ANOFF 位置 1 可禁止它。

将 DNF[3:0] 位配置为一个非零值可使能数字噪声滤波器。这可以抑制 SDA 和 SCL 输入上脉宽小于 DNF[3:0] * $T_{PCLK}$ 的噪声。

使能数字噪声滤波器后，SDA 保持时间可增加 (DNF[3:0] +1) * $T_{PCLK}$。

为了符合 I2C 总线规范版本 2.1 (Thd:dat) 中对最大保持时间的要求，必须使用下面表格所示的限制条件对 DNF 位进行编程并假定模拟滤波器已禁止。

表 20-1 符合 Thd:dat(max) 的 DNF[3:0] 最大值

| PCLK frequency | Maximum DNF value | |
|---|---|---|
| | Sm mode | Fm mode |
| 2 <= FPCLK1 <= 5 | 2 | 0 |
| 5 < FPCLK1 <= 10 | 12 | 0 |
| 10 < FPCLK1 <= 20 | 15 | 1 |
| 20 < FPCLK1 <= 30 | 15 | 7 |
| 30 < FPCLK1 <= 40 | 15 | 13 |
| 40 < FPCLK1 <= 50 | 15 | 15 |

*注意： DNF[3:0] 必须在 I2C 已禁止(PE = 0) 的情况下配置。如果模拟滤波器也已使能，则需将数字滤波器添加到模拟滤波器中。*

### 20.3.6. SDA/SCL 线控制

- 如果时钟延长已使能：
  - 发送模式：如果 TxE=1 且 BTF=1：接口会在发送数据之前保持时钟线为低电平，以等待微控制器将字节写入数据寄存器（缓冲寄存器和移位寄存器均为空）。
  - 接收模式：如果 RxNE=1 且 BTF=1：接口会在接收数据之后保持时钟线为低电平，以等待微控制器将字节读入数据寄存器（缓冲寄存器和移位寄存器均已满）。
- 如果从模式中的时钟延长已禁止：
  - RxNE=1，且在下一个数据接收完成之前还未读走 DR 中的数据就出现上溢错误。会丢失接收的最后一个字节。
  - TxE=1，且在下一个数据的始终到来之前还未把发送数据写到 DR 中，就出现下溢错误。会再次发送同一字节。不会对写冲突进行管理。

### 20.3.7. SMBus

#### 前言

系统管理总线 (SMBus) 是一个双线制接口，各器件可通过它在彼此之间或者与系统的其余部 分进行通信。它以 I2C 的工作原理为基础。SMBus 可针对系统和电源管理相关的任务提供控制总线。系统可使用 SMBus 与设备进行消息传递，而无需切换各个控制线。

系统管理总线规范涉及三类器件。从器件，用于接收或响应命令。主器件，用于发出命令、生成时钟和中止传输。主机，专用的主器件，可提供连接系统 CPU 的主接口。主机必须具有主 - 从设备功能，并且必须

支持 SMBus 主机通知协议。系统中只允许存在一个主机。

**SMBus 与 I2C 的相似之处**

- 双线制总线协议（1 个时钟总线，1 个数据总线）+ 可选 SMBus 报警线。
- 主从通信，主器件提供时钟。
- 多主器件功能。
- SMBus 数据格式与 I2C 7 位地址格式相似。

**SMBus 与 I2C 之间的差异**

下表介绍了 SMBus 与 I2C 之间的差异。

表 20-2 SMBUS 与 I2C

| SMBus | I2C |
|---|---|
| 最大速度 100 kHz | 最大速度 400 kHz |
| 最小时钟速度 10 kHz | 无最小时钟速度 |
| 35 ms 时钟低电平超时 | 无超时 |
| 逻辑电平固定 | 逻辑电平取决于 $V_{DD}$ |
| 地址类型不同（保留、动态等） | 7 位、10 位和广播从模式地址类型 \| |
| 总线协议不同（快速命令、过程调用等） | 无总线协议 |

**SMBus 应用用途**

通过系统管理总线，器件可以提供制造商信息、告诉系统它的型号/部件号、保存暂停事件的状态、报告不同的错误类型、接受控制参数并返回其状态。SMBus 可针对系统和电源管理相 关的任务提供控制总线。

**器件标识**

系统管理总线中作为从器件的任何器件均具有一个唯一地址，被称为从地址。有关保留的从地址列表的信息，请参见 SMBus 规范版本 2.0。

**总线协议**

SMBus 规范支持多达 9 类总线协议。有关这些协议和 SMBus 地址类型的详细信息，请参见 SMBus 规范版本 2.0。这些协议应通过用户软件实施。

**地址解析协议 (ARP)**

通过为各个从器件动态分配一个新的唯一地址可解决 SMBus 从地址冲突的问题。地址解析协议 (ARP) 具有以下属性：

- 地址分配采用标准的 SMBus 物理层仲裁机制。
- 器件通电时，分配的地址保持不变；器件断电时，也允许保留地址。
- 完成地址分配后不会带来额外的 SMBus 数据包开销。(即，对已分配的从地址进行后续访问与访问固定地址的器件所产生的开销相同)。
- 任何 SMBus 主器件都可以遍历总线。

#### 唯一的器件标识符 (UDID)

为了提供一种机制来针对地址分配隔离各个器件，各器件必须具有唯一的器件标识符 (UDID)。 有关 128 位 UDID 和 ARP 的详细信息，请参见 SMBus 规范版本 2.0

#### SMBus 报警模式

SMBus 报警是带有中断线的可选信号，主要用于希望扩展它们的控制能力而牺牲一个引脚的器件。SMBA 是与 SCL 和 SDA 信号类似的线与信号。SMBA 与 SMBus 广播地址配合使用。使用 SMBus 调用的消息长度为 2 字节。

将 I2C_CR1 寄存器中的 ALERT 位置 1 后一个只具有从功能的器件可通过 SMBA 向主机发出信号，指示它想要通信。主机会处理该中断并通过报警响应地址（简称 ARA，其值为 0001 100X）同步访问所有 SMBA 器件。只有那些将 SMBA 拉到低电平的器件会确认报警响应地址。通过 I2C_SR1 寄存器中的 SMBALERT 状态标志可确定这一状态。主机会执行修改后的接收字节操作。由从发送器件提供的 7 位器件地址被放置在字节的 7 个最高有效位。第 8 位可以是 0 或 1。

如果有不止一个器件将 SMBA 拉为低电平，则在从地址传输期间，具有最高优先级（最低位地址）的器件会通过标准仲裁获得通信权限。在确认从地址之后，器件必须释放 SMBA。如 果消息传输结束后主机检测到 SMBA 仍为低电平，会再次读取 ARA。

未实现 SMBA 信号的主机会定期访问 ARA。

有关 SMBus 报警模式的详细信息，请参见 SMBus 规范版本 2.0。

#### 超时错误

SMBus 和 I2C 之间存在一些定时规范方面的差异。

SMBus 定义了一个时钟低电平超时， TIMEOUT 为 35ms。另外，SMBus 还指定 TLOW：SEXT 作为从器件的累积时钟低电平延长时间。SMBus 指定 TLOW：MEXT 作为主器件的累积时钟低电平延长时间。有关这些超时的详细信息，请参见 SMBus 规范版本 2.0。

I2C_SR1 寄存器中的 Timeout 或 Tlow 错误状态标志表明了此特性的状态。

#### 如何在 SMBus 模式下使用该接口

要从 I2C 模式切换到 SMBus 模式，应执行以下步骤。

- 将 I2C_CR1 寄存器中的 SMBus 位置 1。
- 根据应用的要求配置 I2C_CR1 寄存器中的 SMBTYPE 和 ENARP 位。

如果要将某个器件配置为主器件，请按照主模式中介绍的起始位生成步骤进行操作。否则按照从模式中的顺序操作。

该应用需要通过软件控制各种 SMBus 协议。

- ENARP=1 且 SMBTYPE=0 时使用 SMB 器件默认地址。
- ENARP=1 且 SMBTYPE=1 时使用 SMB 主机头字段。
- SMBALERT=1 时使用 SMB 报警响应地址。

### 20.3.8. DMA 请求

DMA 请求（在使能后）仅用于数据传输。当发送数据寄存器变空以及接收数据寄存器变满时会生成 DMA 请求。进行 I2C 数据传输之前，必须先初始化并使能 DMA 。I2C_CR2 寄存器中的 DMAEN 位必须在

ADDR 事件之前置 1。在主模式或从模式下，如果已使能时钟延长，DMAEN 位也可以在 ADDR 事件期间于 ADDR 标志清零之前置 1。结束当前字节传输之前，必须发出 DMA 请求。当传输的数据量达到相应 DMA 通道编程设定的值时，DMA 控制器会发送一个结束传输 EOT 信号给 I2C 接口，并生成一个传输完成中断（如果已使能）：

- 主发送器：在 EOT 中断后的中断程序中，禁止 DMA 请求，然后在等到 BTF 事件后设置 停止位。
- 主接收器：
  - 当要接收的字节数等于或大于二时，DMA 控制器会在收到倒数第二个数据字节（第 N - 1 个数据时）发送一个硬件信号 EOT_1。如果 I2C_CR2 寄存器中的 LAST 位置 1，I2C 会在 EOT_1 后的下一个字节之后自动发送一个 NACK。用户可在 DMA 传输完成中断（如果已使能）程序中生成停止位。
  - 当必须接收单个字节时，必须在 EV6 事件期间于 ADDR 标志清零之前对 NACK 进行编程，即当 ADDR=1 时编程设定 ACK=0。接下来，用户可在 ADDR 标志清零之后或者在执行 DMA 传输完成中断程序时编程设定停止位。

**使用 DMA 进行发送**

将 I2C_CR2 寄存器中的 DMAEN 位置 1 可以使能 DMA 模式进行发送。当 TXE 位置 1 时，数据将由 DMA 从预置的存储区装载进 I2C_DR 寄存器。要映射一个 DMA 通道以便进行 I2C 发送，请按以下步骤操作：其中的 x 表示通道编号。

1. 设置 DMA_CPARx 寄存器中的 I2C_DR 寄存器地址。每次发生 TxE 事件后，数据都会从存储器移动到此地址。
2. 设置 DMA_CMARx 寄存器中的存储器地址。每次发生 TxE 事件后，数据都会从此存储器加载到 I2C_DR。
3. 在 MA_CNDTRx 寄存器中配置要传输的总字节数，在每次 TxE 事件后，此值都会递减
4. 使用 DMA_CCRx 寄存器中的 PL[0:1] 位来配置通道优先级。
5. 在完成半数传输或全部传输（取决于应用的需求）之后，将 DMA_CCRx 寄存器中的 DIR 位置 1 并配置中断。
6. 将 DMA_CCRx 寄存器中的 EN 位置 1 以激活通道。

当传输的数据量达到 DMA 控制器中编程设定的值时，DMA 控制器会发送一个结束传输 EOT/EOT_1 信号给 I2C 接口，而 DMA 会在 DMA 通道中断向量上生成一个中断（如果已使能）。

*注意： 如果使用 DMA 进行传送，请勿使能 I2C_CR2 寄存器中的 ITBUFEN 位。*

**使用 DMA 进行接收**

将 I2C_CR2 寄存器中的 DMAEN 位置 1 可以使能 DMA 模式进行接收。接收数据字节时，数据会从 I2C_DR 寄存器加载到使用 DMA 外设配置的存储区中（参见 DMA 规范）。要映射一个 DMA 通道以便进行 I2C 接收，请按以下步骤操作：其中的 x 表示通道编号。

1. 设置 DMA_CPARx 寄存器中的 I2C_DR 寄存器地址。每次发生 RxNE 事件后，数据都会从此地址移动到存储器。
2. 设置 DMA_CMARx 寄存器中的存储器地址。每次发生 RXNE 事件后，数据都会从 I2C_DR 寄存器加载到此存储区。
3. 在 MA_CNDTRx 寄存器中配置要传输的总字节数，在每次 RxNE 事件后，此值都会递减。
4. 使用 DMA_CCRx 寄存器中的 PL[0:1] 位来配置通道优先级。
5. 在完成半数传输或全部传输（取决于应用的需求）之后，将 DMA_CCRx 寄存器中的 DIR 位重新置 1

并配置中断。

6. 将 DMA_CCRx 寄存器中的 EN 位置 1 以激活通道。

当传输的数据量达到 DMA 控制器中编程设定的值时，DMA 控制器会发送一个结束传输 EOT/EOT_1 信号给 I2C 接口，而 DMA 会在 DMA 通道中断向量上生成一个中断（如果已使能）。

*注意： 如果使用 DMA 进行接收，请勿使能 I2C_CR2 寄存器中的 ITBUFEN 位。*

### 20.3.9. 数据包错误校验

PEC 计算器的应用目的是提高通信的可靠性。PEC 对各个位使用 CRC-8 多项式 $C(x) = x^8 + x^2 + x + 1$ 公式进行串行计算。

- 将 I2C_CR1 寄存器中的 ENPEC 位置 1 即可使能 PEC 计算。PEC 是针对所有消息字节（包括地址和 R/W 位）的 CRC-8 计算。
  - 在发送过程中：在与最后一个字节对应的 TxE 事件发生后，将 I2C_CR1 寄存器中的 PEC 传输位置 1 。PEC 会在最后一个传输的字节之后进行传送。
  - 在接收过程中：在与最后一个字节对应的 RxNE 事件发生之后，将 I2C_CR1 寄存器中的 PEC 位置 1，以便接收器在接收到的下一个字节不等于内部计算的 PEC 时发送一个 NACK。在主接收器中，无论校验结果如何，PEC 后都将发送 NACK 。在从模式下，必须在 CRC 接收的 ACK 之前设置 PEC。在主模式下，必须在复位 ACK 时设置 PEC 位。
- I2C_SR1 寄存器中还提供 PECERR 错误标志/中断。
- 如果 DMA 和 PEC 计算均已使能：
  - 在发送过程中：当 I2C 接口接收来自 DMA 控制器的 EOT 信号时，会在最后一个字节之后自动发送 PEC。
  - 在接收过程中：当 I2C 接口接收来自 DMA 控制器的 EOT_1 信号时，会自动将下一个字节视为 PEC 并对其进行校验。在 PEC 接收之后会生成一个 DMA 请求。
- 为了允许进行中间 PEC 传输，可使用 I2C_CR2 寄存器中的一个控制位（LAST 位）来确定它是否真的是最后一个 DMA 传输。如果确实是主接收器的最后一个 DMA 请求，则会在接收最后一个字节后自动发送一个 NACK。
- PEC 计算会因仲裁丢失而失效。

## 20.4. I2C 中断

下表列出了 I2C 中断请求列表。

表 20-3 I2C 中断请求

| 中断事件 | 事件标志 | 使能控制位 |
|---|---|---|
| 发送起始位（主模式） | SB | ITEVFEN |
| 地址已发送（主模式）或地址匹配（从模式） | ADDR | |
| 10 位地址的头段已发送（主模式） | ADD10 | |
| 已收到停止位（从模式） | STOPF | |
| 完成数据字节传输 | BTF | |
| 接收缓冲区非空 | RxNE | ITEVFEN 和 ITBUFEN |
| 发送缓冲区为空 | TxE | |
| 总线错误 | BERR | ITERREN |
| 仲裁丢失（主模式） | ARLO | |
| 应答失败 | AF | |
| 上溢/下溢 | OVR | |

| 中断事件 | 事件标志 | 使能控制位 |
| --- | --- | --- |
| PEC 错误 | PECERR | |
| 超时/Tlow 错误 | TIMEOUT | |
| SMBus 报警 | SMBALERT | |

*注意： SB、ADDR、ADD10、STOPF、BTF、RxNE 和 TxE 通过逻辑或映射到同一个中断通道上。BERR、ARLO、AF、OVR、PECERR、TIMEOUT 和 SMBALERT 通过逻辑或映射到同一个中断通道上。*

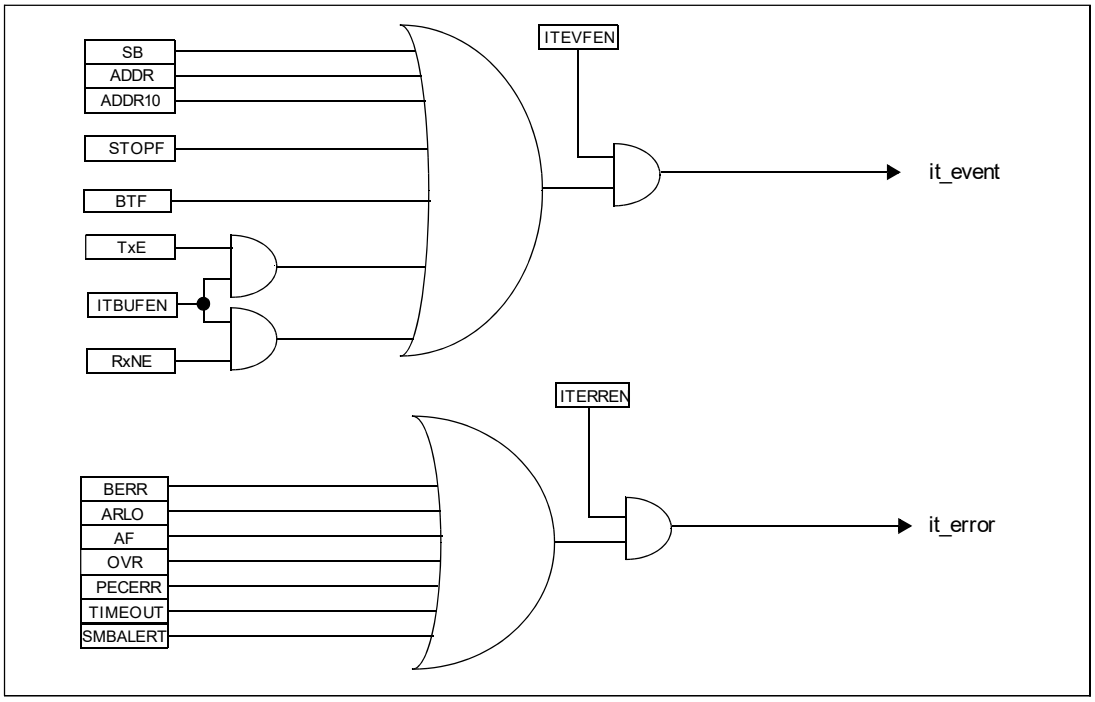



图 20-7 I2C 中断映射图

## 20.5. I2C 调试模式

当微控制器进入调试模式(内核停止)时，SMBUS 超时会根据 DBG 模块中的 DBG_I2Cx_SMBUS_TIMEOUT 配置位选择继续正常工作或者停止工作。

## 20.6. I2C 寄存器

外设寄存器可支持半字（16 位）或字（32 位）访问。

### 20.6.1. I2C 控制寄存器 1 (CR1)

**偏移地址：** 0x00

**复位值：** 0x00000000

*注意： STOP、START 或 PEC 位置 1 之后，在硬件将该位清零之 前，软件不能对 I2C_CR1 执行任何写操作。否则，可能会再次发出 STOP、START 或 PEC 置位请求。*

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SWRST | res. | ALERT | PEC | POS | ACK | STOP | START | NOSTRETCH | ENGC | ENPEC | ENARP | SMBTYPE | res. | SMBUS | PE |
| rw | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | | rw | rw |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15 | SWRST | 软件复位<br>当置 1 时，I2C 处于复位状态。在复位此位之前，确保 I2C 线已释放且总线空闲。<br>0：I2C 外设未处于复位状态<br>1：I2C 外设处于复位状态<br>*注意： 当出现错误或锁定状态后，可使用此位重新初始化外设。例如，如果 BUSY 位已置 1 但因母线干扰而不能复位，则可使用 SWRST 位退出此状态。* |
| 14 | 保留 | 必须保持复位值。 |
| 13 | ALERT | SMBus 报警<br>此位由软件置 1 和清零，并可在 PE=0 时由硬件清零。<br>0：释放 SMBA 引脚使其变成高电平。报警响应地址头后跟 NACK。<br>1：驱动 SMBA 引脚使其变成低电平。报警响应地址头后跟 ACK。 |
| 12 | PEC | 数据包错误校验<br>此位由软件置 1 和清零，并可在 PEC 传输完成时由硬件清零，或者在 PE=0 时或在检测到起始位或停止位时由硬件清零。<br>0：不传输 PEC<br>1：PEC 传输（在 Tx 或 Rx 模式下）<br>*注意： PEC 计算会因仲裁丢失而失效。* |
| 11 | POS | 应答/PEC 位置（针对接收数据）<br>此位由软件置 1 和清零，并可在 PE=0 时由硬件清零。<br>0: ACK 位控制移位寄存器 中当前正在接收的字节的 (N)ACK。PEC 位指示移位寄存器中的当前字节是一个 PEC。<br>1：ACK 位控制移位寄存器中要接收的下一个字节的 (N)ACK。PEC 位指示移位寄存器的下一个字节是一个 PEC。<br>*注意： POS 位只能用于主设备接收 2 个字节时。它必须在数据开始接收之前进行配置。* |
| 10 | ACK | 应答使能<br>此位由软件置 1 和清零，并可在 PE=0 时由硬件清零。<br>0：不返回应答<br>1：在接收一个字节（匹配地址或 数据）之后返回应答 |
| 9 | STOP | 生成停止位<br>该位由软件置 1 和清零，也可在检测到停止位时由硬件清零，在检测到超时错误时由硬件置 1。<br>在主模 式下：<br>0：不生成停止位。<br>1：在传输当前字节或发送当前起始位后生成停止位。<br>在从模式下：<br>0：不生成停止位。<br>1：完成当前字节传输后释放 SCL 和 SDA 线。 |
| 8 | START | 生成起始位<br>此位由软件置 1 和清零，并可在起始位发送完成后或 PE=0 时由硬件清零。 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 在主模式下:<br>0：不生成 起始位<br>1：生成重复起始位<br>在从模式下:<br>0：不生成起始位<br>1：在总线空闲时生成起始位 |
| 7 | NOSTRETCH | 禁止时钟延长（从模式）<br>在从模式下，当 ADDR 或 BTF 标志置 1 时，此位用于禁止时钟延长，直到软件将其复位为止。<br>0：使能时钟延长<br>1：禁止时钟延长 |
| 6 | ENGC | 广播呼叫使能<br>0：禁止广播呼叫。不对地址 00h 应答。<br>1：使能广播呼叫。对地址 00h 应答。 |
| 5 | ENPEC | PEC 使能<br>0：禁止 PEC 计算<br>1：使能 PEC 计算 |
| 4 | ENARP | ARP 使能<br>0：禁止 ARP<br>1：使能 ARP<br>SMBTYPE=0 时识别 SMBus 器件默认地址<br>SMBTYPE=1 时识别 SMBus 主机地址 |
| 3 | SMBTYPE | SMBus 类型<br>0：SMBus 器件<br>1：SMBus 主机 |
| 2 | 保留 | 必须保持复位值。 |
| 1 | SMBUS | SMBus 模式<br>0：I2C 模式<br>1：SMBus 模式 |
| 0 | PE | 外设使能<br>0：禁止外设<br>1：使能外设<br>*注意： 如果此位在通信进行过程中复位，在结束本次通信后会带 IDLE 状态，外设被禁止。由于通信结束时 PE=0，所有位均会复位。*<br>*在主模式下，此位不能在通信结束之前复位。* |

### 20.6.2. I2C 控制寄存器 2 (CR2)

**偏移地址：** 0x04

**复位值：** 0x00000000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | LAST | DMAEN | ITBUFEN | ITEVTEN | ITERREN | res. | FREQ[6:0] | | | | | | |
| | | | rw | rw | rw | rw | rw | | rw | | | | | | |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:13 | 保留 | 必须保持复位值 |

| 位域 | 名称 | 描述 |
|---|---|---|
| 12 | LAST | 最后一次 DMA 传输<br>0：下一个 DMA EOT 不是最后一次传输<br>1：下一个 DMA EOT 是最后一次传输<br>*注意： 此位用于主接收模式，可对最后接收的数据生成 NACK。* |
| 11 | DMAEN | DMA 请求使能<br>0：禁止 DMA 请求<br>1：当 TxE=1 或 RxNE =1 时使能 DMA 请求 |
| 10 | ITBUFEN | 缓冲中断使能<br>0：TxE = 1 或 RxNE = 1 时不生成任何中断。<br>1：TxE = 1 或 RxNE = 1 时生成事件中断（与 DMAEN 状态无关） |
| 9 | ITEVTEN | 事件中断使能<br>0：禁止事件中断<br>1：使能事件中断<br>满足以下条件时将生成此中断：<br>* SB = 1（主模式）<br>* ADDR = 1（主/从模式）<br>* ADD10= 1（主模式）<br>* STOPF = 1（从模式）<br>* BTF = 1，无 TxE 或 RxNE 事件<br>* ITBUFEN = 1 且 TxE 事件置 1<br>* ITBUFEN = 1 且 RxNE 事件置 1 |
| 8 | ITERREN | 错误中断使能<br>0：禁止错误中断<br>1：使能错误中断<br>满足以下任意条件时将生成此中断：<br>* BERR = 1<br>* ARLO = 1<br>* AF = 1<br>* OVR = 1<br>* PECERR = 1<br>* TIMEOUT = 1<br>* SMBALERT = 1 |
| 7 | 保留 | 必须保持复位值 |
| 6:0 | FREQ[6:0] | 外设时钟频率<br>外设时钟频率必须使用 APB 时钟频率进行配置（I2C 外设连接到 APB）。<br>允许的最小频率为 2 MHz，最大频率则受限于 APB 最大频率 (90 MHz)。<br>0b0000000：不允许<br>0b0000001：不允许<br>0b0000010：2 MHz<br>…<br>0b1011010：90 MHz<br>大于 0b1011010：不允许 |

### 20.6.3. I2C 自有地址寄存器 1 (OAR1)

**偏移地址：** 0x08

**复位值：** 0x00000000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDMODE | res. | res. | | | | ADD[9:8] | | ADD[7:1] | | | | | | | ADD0 |
| rw | | | | | | rw | | rw | | | | | | | rw |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15 | ADDMODE | 寻址模式（从模式）<br>0：7 位从地址（无法应答 10 位地址）<br>1：10 位从地址（无法应答 7 位地址） |
| 14 | 保留 | 应通过软件始终保持为 1。 |
| 13:10 | 保留 | 保留，必须保持复位值 |
| 9:8 | ADD[9:8] | 接口地址<br>7 位寻址模式：无意义<br>10 位寻址模式：地址的第 9:8 位 |
| 7:1 | ADD[7:1] | 接口地址<br>地址的第 7:1 位 |
| 0 | ADD0 | 接口地址<br>7 位寻址模式：无意义<br>10 位寻址模式：地址的第 0 位 |

### 20.6.4. I2C 自有地址寄存器 2 (OAR2)

**偏移地址：** 0x0C

**复位值：** 0x00000000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | | | | | ADD2[7:1] | | | | | | | ENDUAL |
| | | | | | | | | rw | | | | | | | rw |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 必须保持复位值 |
| 7:1 | ADD2[7:1] | 接口地址<br>双寻址模式下的地址第 7:1 位 |
| 0 | ENDUAL | 双寻址模式使能<br>0：7 位寻址模式下仅对 OAR1 地址响应<br>1：7 位寻址模式下能对 OAR1 和 OAR2 两个地址响应 |

### 20.6.5. I2C 数据寄存器 (DR)

**偏移地址：** 0x10

**复位值：** 0x00000000

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 7:0 | DR[7:0] | 8 位数据寄存器<br>接收的字节或者要发送到总线的字节。<br>–发送模式：在 DR 寄存器中写入第一个字节时自动开始发送字节。如果在启动传送 (TxE=1) 后立即将下一个要传送的数据置于 DR 中，则可以保持连续的传送流–接收模式：将接收到的字节复制到 DR 中 (RxNE=1)。如果在接收下一个数据字节 (RxNE=1)之前读取 DR，则可保持连续的传送流。<br>注意：<br>在从模式下，地址并不会复制到 DR 中。<br>硬件不对写冲突进行管理（TxE=0 时也可对 DR 执行写操作）。<br>如果发出 ACK 脉冲时出现 ARLO 事件，则不会将接收到的字节复制到 DR 寄存器，因而也无法读取字节。 |

### 20.6.6. I2C 状态寄存器 1 (SR1)

**偏移地址：** 0x14

**复位值：** 0x00000000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SMBALERT | TIMEOUT | res. | PECERR | OVR | AF | ARLO | BERR | TxE | RxNE | res. | STOPF | ADD10 | BTF | ADDR | SB |
| rc_w0 | rc_w0 | | rc_w0 | rc_w0 | rc_w0 | rc_w0 | rc_w0 | r | r | | r | r | r | r | r |

| 位域 | 名称 | 描述 |
|---|---|---|
| 15 | SMBALERT | SMBus 报警<br>在 SMBus 主机模式下：<br>0：无 SMBALERT<br>1：引脚上发生 SMBALERT 事件<br>在 SMBus 从模式 下：<br>0：无 SMBALERT 响应地址头<br>1：接收到指示 SMBALERT 低电平的 SMBALERT 响应地址头<br>* 由软件写入 0 来清零，或在 PE=0 时由硬件清零。 |
| 14 | TIMEOUT | 超时或 $T_{low}$ 错误<br>0：无超时错误<br>1：SCL 低电平时长持续 25 ms（超时）<br>或<br>主器件累计时钟低电平延长时间超过 10 ms (Tlow~:mext)<br>或<br>从器件累计时钟低电平延长时间超过 25 ms (Tlow~:sext)<br>* 在从模式下置 1 时：从器件复位通信 且硬件释放数据线<br>* 在主模式下置 1 时：由硬件发送停止位<br>* 由软件写入 0 来清零，或在 PE=0 时由硬件清零。<br>*注意： 此功能 仅在 SMBus 模式下可用。* |
| 13 | 保留 | 必须保持复位值 |
| 12 | PECERR | 接收期间 PEC 错误<br>0：无 PEC 错误：接收器在接收 PEC 后返回 ACK（如果 ACK=1）<br>1：PEC 错误：接收器在 接收 PEC 后返回 NACK（无论 ACK 什么值）<br>* 由软件写入 0 来清零，或在 PE=0 时由硬件清零。<br>*注意：接收到错误的 CRC 时，如 果在结束 CRC 接收之前 PEC 控制位没有置 1，则 PECERR 位在从模* |

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| | | *式下不会置 1。不过可以通过读取 PEC 值来判定接收到的 CRC 是否 正确。* |
| 11 | OVR | 上溢/下溢<br>0：未发生上溢/下溢<br>1：上溢或下溢<br>* 在从模式下由硬件置 1，前提是满足 NOSTRETCH=1 且：<br>* 接收过程中接收到一个新字节（包括 ACK 脉冲）但尚未读取 DR 寄存器。新接收的字节将丢失。<br>* 发送过程中将发送一个新字节但尚未向 DR 寄存器写入数据。同一字节发送两次。<br>* 由软件写入 0 来清零，或在 PE=0 时由硬件清零。<br>*注意： 如果 DR 写操作时间与出现 SCL 上升沿的时间非常接近，则发出的数据不确定，并且出现数据保持时间错误。* |
| 10 | AF | 应答失败<br>0：未发生应答失败<br>1：应答失败<br>* 无应答返回时由硬件置 1。<br>* 由软件写入 0 来清零，或在 PE=0 时由硬件清零。 |
| 9 | ARLO | 仲裁丢失（主模式）<br>0：未检测到仲裁丢失<br>1：检测到仲裁丢失<br>当接口在竞争总线是输给另一个主设备时，由 硬件将该位置 1<br>* 由软件写入 0 来清零，或在 PE=0 时由硬件清零。<br>发生 ARLO 事件后，接口会自动切换回从模式 (M/SL=0)。<br>*注意： 在 SMBUS 中，从模式下的数据仲裁仅发生在数据阶段或发送确认期间（不适用于地址确认）。* |
| 8 | BERR | 总线错误<br>0：无误放的起始或停止位<br>1：存在误放的起始或停止位<br>* SCL 为高电平时，若接口在字节传输期间检测到某个无效位置出现 SDA 上升沿或下降沿，则会由硬件将该位置 1。<br>* 由软件写入 0 来清零，或在 PE=0 时由硬件清零。 |
| 7 | TxE | 数据寄存器为空（发送器）<br>0：数据寄存器非空<br>1：数据寄存器为空<br>* 发送过程中 DR 为空时该位置 1。TxE 不会在地址阶段置 1。<br>* 由软件写入 DR 寄存器来清零，或在出现起始、停止位或者 PE=0 时由硬件清零。<br>如果接收到 NACK 或要发送的下一个字节为 PEC (PEC=1)，TxE 将不会置 1<br>*注意： 写入第一个要发送的数据或在 BTF 置 1 时写入数据都无法将 TxE 清零，因为这两种情况下数据寄存器仍为空。* |
| 6 | RxNE | 数据寄存器非空（接收器）<br>0：数据寄存器为空<br>1：数据寄存器非空<br>* 接收模式下数据寄存器非空时置 1。RxNE 不会在地址阶段置 1。<br>* 由软件读取或写入 DR 寄存器来清零，或在 PE=0 时由硬件清零。<br>发生 ARLO 事件时 RxNE 不会置 1。<br>*注意： BTF 置 1 时无法通过读取数据将 RxNE 清零，因为此时数据寄存器仍为满。* |
| 5 | 保留 | 必须保持复位值 |

| 位域 | 名称 | 描述 |
|---|---|---|
| 4 | STOPF | 停止位检测（从模式）<br>0：未检测到停止位<br>1：检测到停止位<br>* 从设备在应答脉冲后（如果 ACK=1）检测到停止位，由硬件置 1。<br>* 由软件分别对 SR1 寄存器和 CR1 寄存器执行读操作和写操作来清零，或在 PE=0 时由硬件清零。<br>*注意： 收到 NACK 后 STOPF 位不会置 1。*<br>*建议在 STOPF 置 1 后执行完整的清零序列（首先读取 SR1，然后写入 CR1）。请参见从接收器的传输序列图。* |
| 3 | ADD10 | 发送 10 位头（主模式）<br>0：未发生 ADD10 事件。<br>1：主器件已发送第一个地址字节（头）。<br>* 主器件在 10 位 地址模式下已发送第一个字节时由硬件置 1。<br>* 由软件在读取 SR1 寄存器后在 DR 寄存器中写入第二个地址字节来清零，或在 PE=0 时由硬件清零。<br>*注意： 收到 NACK 后 ADD10 位不会置 1* |
| 2 | BTF | 字节传输完成<br>0：数据字节传输未完成<br>1：数据字节传输成功完成<br>* 由硬件置 1，前提是满足 NOSTRETCH=0 且：<br>* 接收过程中接收到一个新字节（包括 ACK 脉冲）但尚未读取 DR 寄存器 (RxNE=1)。<br>* 发送过程中将发送一个新字节但尚未向 DR 寄存器写入数据 (TxE=1)。<br>* 由软件读或写 DR 寄存器来清零，或在发送过程中出现起始或停止位后由硬件清零，也可以在 PE=0 时由硬件清零。<br>*注意： 收到 NACK 后 BTF 位不会置 1*<br>*如果下一个要发送的字节为 PEC（I2C_SR2 寄存器中的 TRA=1，I2C_CR1 寄存器中的 PEC=1），则 BTF 位不会置 1* |
| 1 | ADDR | 地址已发送（主模式）/地址匹配（从模式）<br>由软件在读取 SR1 寄存器后读取 SR2 寄存器来清零，或在 PE=0 时由硬件清零。<br>地址匹配（从模式）<br>0：地址不匹配或未接收到地址。<br>1：接收到的地址匹配。<br>* 当接收到的从地址与 OAR 寄存器内容、广播呼叫地址或 SMBus 器件默认地址匹配时，或者识别到 SMBus 主机或 SMBus 报警时，该位由硬件置 1。（根据配置确定何时使能） 。<br>*注意：*<br>*在从模式下，建议在 ADDR 置 1 后执行完整的清零序列（首先读取 SR1，然后写入 SR2）。请参见从接收器的传输序列图。*<br><br>地址已发送（主模式）<br>0：地址发送未结束<br>1：地址发送结束<br>* 在 10 位寻址模式下，接收到第二个地址字节的 ACK 后该位置 1。<br>* 在 7 位寻址模式下，接收到地址字节的 ACK 后该位置 1。<br>**注意：**<br>收到 NACK 后 ADDR 位不会置 1 |

| 位域 | 名称 | 描述 |
|---|---|---|
| 0 | SB | 起始位 (Start bit)（主模式）<br>0：无起始位<br>1：起始位已经发送。<br>* 生成启动条件时置 1。<br>* 由软件在读取 SR1 寄存器后写入 DR 寄存器来清零，或在 PE=0 时由硬件清零 |

### 20.6.7. I2C 状态寄存器 2 (SR2)

**偏移地址：** 0x18

**复位值：** 0x00000000

**注意：** 读取 I2C_SR1 后再读取 I2C_SR2 可将 ADDR 标志清零，即使 ADDR 标志在读取 I2C_SR1 之后置 1 也如此。因此，必须仅在 I2C_SR1 中的 ADDR 位已置 1 或者 STOPF 位已清零后读取 I2C_SR2。

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PEC[7:0] | | | | | | | | DUALF | SMBHOST | SMBDEFAULT | GENCALL | res. | TRA | BUSY | MSL |
| r | | | | | | | | r | r | r | r | | r | r | r |

| 位域 | 名称 | 描述 |
|---|---|---|
| 15:8 | PEC[7:0] | 数据包错误校验寄存器<br>ENPEC=1 时，此寄存器包含内部 PEC。 |
| 7 | DUALF | 双标志（从模式）<br>0：接收到的地址与 OAR1 匹配<br>1：接收到的地址与 OAR2 匹配<br>* 出现停止位、重复起始位或 PE=0 时由硬件清零。 |
| 6 | SMBHOST | SMBus 主机头（从模式）<br>0：无 SMBus 主机地址<br>1：SMBTYPE=1 且 ENARP=1 时接收到 SMBus 主机地址。<br>* 出现停止位、重复起始位或 PE=0 时由硬件清零。 |
| 5 | SMBDEFAULT | SMBus 器件默认地址（从模式）<br>0：无 SMBus 器件默认地址<br>1：ENARP=1 时接收到 SMBus 器件默认地址<br>* 出现停止位、重复起始位或 PE=0 时由硬件清零。 |
| 4 | GENCALL | 广播呼叫地址（从模式）<br>0：无广播呼叫<br>1：ENGC=1 时接收到广播呼叫地址<br>* 出现停止位、重复起始位或 PE=0 时由硬件清零。 |
| 3 | 保留 | 必须保持复位值 |
| 2 | TRA | 发送器/接收器<br>0：接收器<br>1：发送器<br>此位在整个地址阶段的结尾处根据地址字节的 R/W 位状态进行置 1。<br>同样，检测到停止位 (STOPF=1)、重复起始位、总线仲裁丢失 (ARLO=1) 或当 PE=0 时该位也由硬件清零。 |
| 1 | BUSY | 总线忙碌<br>0：总线上无通信 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 1：总线正在进行通信<br>* 检测到 SDA 或 SCL 低电平时由硬件置 1<br>* 检测到停止位时由硬件清零。<br>该位指示总线上是否正在进行通信。即使禁止接口 (PE=0) 后此信息也会更新。 |
| 0 | MSL | 主/从模式<br>0：从模式<br>1：主模式<br>* 接口进入主模式时 (SB=1) 由硬件置 1。<br>* 检测到总线上的停止位、仲裁丢失 (ARLO=1) 或当 PE=0 时由硬件清零。 |

### 20.6.8. I2C 时钟控制寄存器 (CCR)

**偏移地址：** 0x1C

**复位值：** 0x00000000

**注意：** fPCLK1 必须至少为 2MHz，才能达到标准模式 I2C 频率。必须至少为 4MHz 才能达到快速模式 I2C 频率。必须为 10MHz 的倍数才能实现最大 400 kHz 的 I2C 快速模式时钟。CCR 寄存器必须仅在禁止 I2C (PE = 0) 的情况下配置。

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F/S | DUTY | res. | | CCR[11:0] | | | | | | | | | | | |
| rw | rw | | | rw | | | | | | | | | | | |

| 位域 | 名称 | 描述 |
|---|---|---|
| 15 | F/S | I2C 主模式选择<br>0：标准模式 I2C<br>1：快速模式 I2C |
| 14 | DUTY | 快速模式占空比<br>0：快速模式 $T_{low}/T_{high}$ = 2<br>1：快速模式 $T_{low}/T_{high}$ = 16/9 (参见 CCR) |
| 13:12 | 保留 | 必须保持复位值 |
| 11:0 | CCR[11:0] | 快速/标准模式下的时钟控制寄存器（主模式）<br>控制主模式下的 SCL 时钟。<br>标准模式或 SMBus 模式:<br>$T_{high}$ = CCR * TPCLK1<br>$T_{low}$ = CCR * TPCLK1<br><br>快速模式:<br>如果 DUTY = 0:<br>$T_{high}$ = CCR * TPCLK1<br>$T_{low}$ = 2 * CCR * TPCLK1<br>如果 DUTY = 1：(达到 400kHz)<br>$T_{high}$ = 9 * CCR * TPCLK1<br>$T_{low}$ = 16 * CCR * TPCLK1<br><br>例如：要在标准模式下生成 100 kHz 的 SCL 频率:<br>如果 FREQR = 08，TPCLK1 = 125 ns，则必须将 CCR 编程为 0x28 (0x28 <=> 40d x 125 ns = 5000 ns)。 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | **注意：**<br>1. 允许的最小值为 0x04，但快速占空比模式例外，其最小值为 0x01<br>2. 这些时间均未经过滤波。<br>3. CCR 寄存器必须仅在禁止 I2C (PE = 0) 的情况下配置。 |

### 20.6.9. I2C TRISE 寄存器 (TRISE)

**偏移地址：** 0x20

**复位值：** 0x00000002

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:7 | 保留 | 必须保持复位值 |
| 6:0 | TRISE[6:0] | 快速/标准模式下的最大上升时间（主模式）<br>这些位必须编程为 I2C 总线规范中给定的最大 SCL 上升时间加 1。<br>例如：标准模式下允许的最大 SCL 上升时间为 1000 ns。<br>如果 I2C_CR2 寄存器中 FREQ[6:0] 位的值等于 0x08 且 TPCLK1 = 125 ns，则 TRISE[6:0] 位必须编程为 09h。<br>(1000 ns / 125 ns = 8 + 1)<br>滤波器值也可以叠加到 TRISE[6:0]。<br>如果结果不为整数，则 TRISE[6:0] 必须编程为整数部分，以符合 $T_{high}$ 参数要求。<br>*注意：*<br>*TRISE[6:0] 必须仅在禁止 I2C (PE = 0) 的情况下配置 。* |

### 20.6.10. I2C FLTR 寄存器 (FLTR)

**偏移地址：** 0xFC

**复位值：** 0x1C040404

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | ANOFF[1:0] | | DNOFF[2:0] | | | FLTR_NUM2[7:0] | | | | | | | |
| | | | rw | | rw | | | rw | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FLTR_NUM1[7:0] | | | | | | | | FLTR_NUM0[7:0] | | | | | | | |
| rw | | | | | | | | rw | | | | | | | |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:29 | 保留 | 必须保持复位值 |
| 28:27 | ANOFF[1:0] | 模拟噪声滤波器关闭<br>BIT1:1 表示开启 SDA 模拟滤波，0 表示关闭 SDA 模拟滤波<br>BIT0:1 表示开启 SCL 模拟滤波，0 表示关闭 SCL 模拟滤波<br>**注意：** |

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 必须仅在禁止 I2C (PE = 0) 的情况下配置。 |
| 26:24 | DNOFF[2:0] | 模拟噪声滤波器关闭<br>BIT2:1 表示开启 ALERT 数字滤波，0 表示关闭 ALERT 数字滤波<br>BIT1:1 表示开启 SDA 数字滤波，0 表示关闭 SDA 数字滤波<br>BIT0:1 表示开启 SCL 数字滤波，0 表示关闭 SCL 数字滤波 |
| 23:16 | FLTR_NUM2[7:0] | ALERT 数字滤波采样比对次数，最小值为 2 |
| 15:8 | FLTR_NUM1[7:0] | SDA 数字滤波采样比对次数，最小值为 2 |
| 7:0 | FLTR_NUM0[7:0] | SCL 数字滤波采样比对次数，最小值为 2 |

# 21. 通用同步异步收发器(USART)

## 21.1. USART 简介

通用同步异步收发器 (USART) 能够灵活地与外部设备进行全双工数据交换，满足外部设备对工业标准 NRZ 异步串行数据格式的要求。USART 通过小数波特率发生器提供了多种波特率。

它支持同步单向通信和半双工单线通信；还支持 LIN（局域互连网络）、智能卡协议与 IrDA（红外线数据协会）SIR ENDEC 规范，以及调制解调器操作 (CTS/RTS)。而且，它还支持多处理器通信。

通过配置多个缓冲区使用 DMA 可实现高速数据通信。

## 21.2. USART 主要特性

- 全双工异步通信；
- NRZ 标准格式（标记/空格）；
- 可配置为 16 倍过采样或 8 倍过采样，因而为速度容差与时钟容差的灵活配置提供了可能；
- 小数波特率发生器系统：
  - 通用可编程收发波特率（有关最大 APB 频率时的波特率值，请参见数据手册）。
- 数据字长度可编程（8 位或 9 位）；
- 停止位可配置 - 支持 0.5、1、1.5 或 2 个停止位；
- 集成深度为 8 的接收 FIFO；
- LIN 主模式同步停止符号发送功能和 LIN 从模式停止符号检测功能：
  - 对 USART 进行 LIN 硬件配置时可生成 13 位停止符号和检测 10/11 位停止符号。
- 用于同步发送的发送器时钟输出；
- IrDA SIR 编码解码器：
  - 正常模式下，支持 3/16 位持续时间。
- 智能卡仿真功能：
  - 智能卡接口支持符合 ISO 7816-3 标准中定义的异步协议智能卡；
  - 智能卡工作模式下，支持 0.5 或 1.5 个停止位。
- 单线半双工通信；
- 使用 DMA（直接存储器访问）实现可配置的多缓冲区通信：
  - 使用 DMA 在预留的 SRAM 缓冲区中收/发字节。
- 发送器和接收器具有单独使能位；
- 传输检测标志：
  - 接收缓冲区已满；
  - 接收缓冲区将满(FIFO 模式下)；
  - 发送缓冲区为空；
  - 传输结束标志。
- 奇偶校验控制：

  - 发送奇偶校验位；
  - 检查接收的数据字节的奇偶性。
- 四个错误检测标志：
  - 溢出错误；
  - 噪声检测；
  - 帧错误；
  - 奇偶校验错误。
- 十一个具有标志位的中断源：
  - CTS 变化；
  - LIN 停止符号检测；
  - 发送数据寄存器为空；
  - 发送完成；
  - 接收数据寄存器已满；
  - 接收数据寄存器将满(FIFO 模式下)；
  - 接收到线路空闲；
  - 溢出错误；
  - 帧错误；
  - 噪声错误；
  - 奇偶校验错误。
- 多处理器通信 - 如果地址不匹配，则进入静默模式；
- 从静默模式唤醒（通过线路空闲检测或地址标记检测）；
- 两个接收器唤醒模式：地址位（MSB，第 9 位），线路空闲。

## 21.3. USART 功能说明

接口通过三个引脚从外部连接到其它设备。任何 USART 双向通信均需要至少两个引脚：接收数据输入引脚 (RX) 和发送数据引脚输出 (TX)：

**RX**： 接收数据输入引脚就是串行数据输入引脚。过采样技术可区分有效输入数据和噪声，从而用于恢复数据。

**TX**： 发送数据输出引脚。如果关闭发送器，该输出引脚模式由其 I/O 端口配置决定。如果使能了发送器但没有待发送的数据，则 TX 引脚处于高电平。在单线和智能卡模式下，该 I/O 用于发送和接收数据。

正常 USART 模式下，通过这些引脚以帧的形式发送和接收串行数据：

- 发送或接收前保持空闲线路；
- 起始位；
- 数据（字长 8 位或 9 位），最低有效位在前；
- 用于指示帧传输已完成的 0.5 个、1 个、1.5 个、2 个停止位；
- 该接口使用小数波特率发生器 - 带 12 位尾数和 4 位小数；
- 状态寄存器 (USART_SR)；
- 数据寄存器 (USART_DR)；
- 波特率寄存器 (USART_BRR) - 12 位尾数和 4 位小数；

- 智能卡模式下的保护时间寄存器 (USART_GTPR)。

有关各个位的定义，请参见 USART 寄存器。

在同步模式下连接时需要以下引脚：

- **SCLK**： 发送器时钟输出。该引脚用于输出发送器数据时钟，以便按照 SPI 主模式进行同步发送（起始位和结束位上无时钟脉冲，可通过软件向最后一个数据位发送时钟脉冲）。 RX 上可同步接收并行数据。这一点可用于控制带移位寄存器的外设。时钟相位和极性可通过软件编程。在智能卡模式下，SCLK 可向智能卡提供时钟。在硬件流控制模式下需要以下引脚：
- **nCTS**：“清除以发送”用于在当前传输结束时阻止数据发送（高电平时）。
- **nRTS**：“请求以发送”用于指示 USART 已准备好接收数据（低电平时）。

### 21.3.1. USART 字符说明

可通过对 USART_CR1 寄存器中的 M 位进行编程来选择 8 位或 9 位的字长。

TX 引脚在起始位工作期间处于低电平状态。在停止位工作期间处于高电平状态。

空闲字符可理解为整个帧周期内电平均为“1”（停止位的电平也是“1”），该字符后是下一个数据帧的起始位。

停止字符可理解为在一个帧周期内接收到的电平均为“0”。发送器在中断帧的末尾插入 1 或 2 个停止位（逻辑“1”位）以确认起始位。

发送和接收由通用波特率发生器驱动，发送器和接收器的使能位分别置 1 时将生成相应的发送时钟和接收时钟。

下面给出了各个块的详细信息。

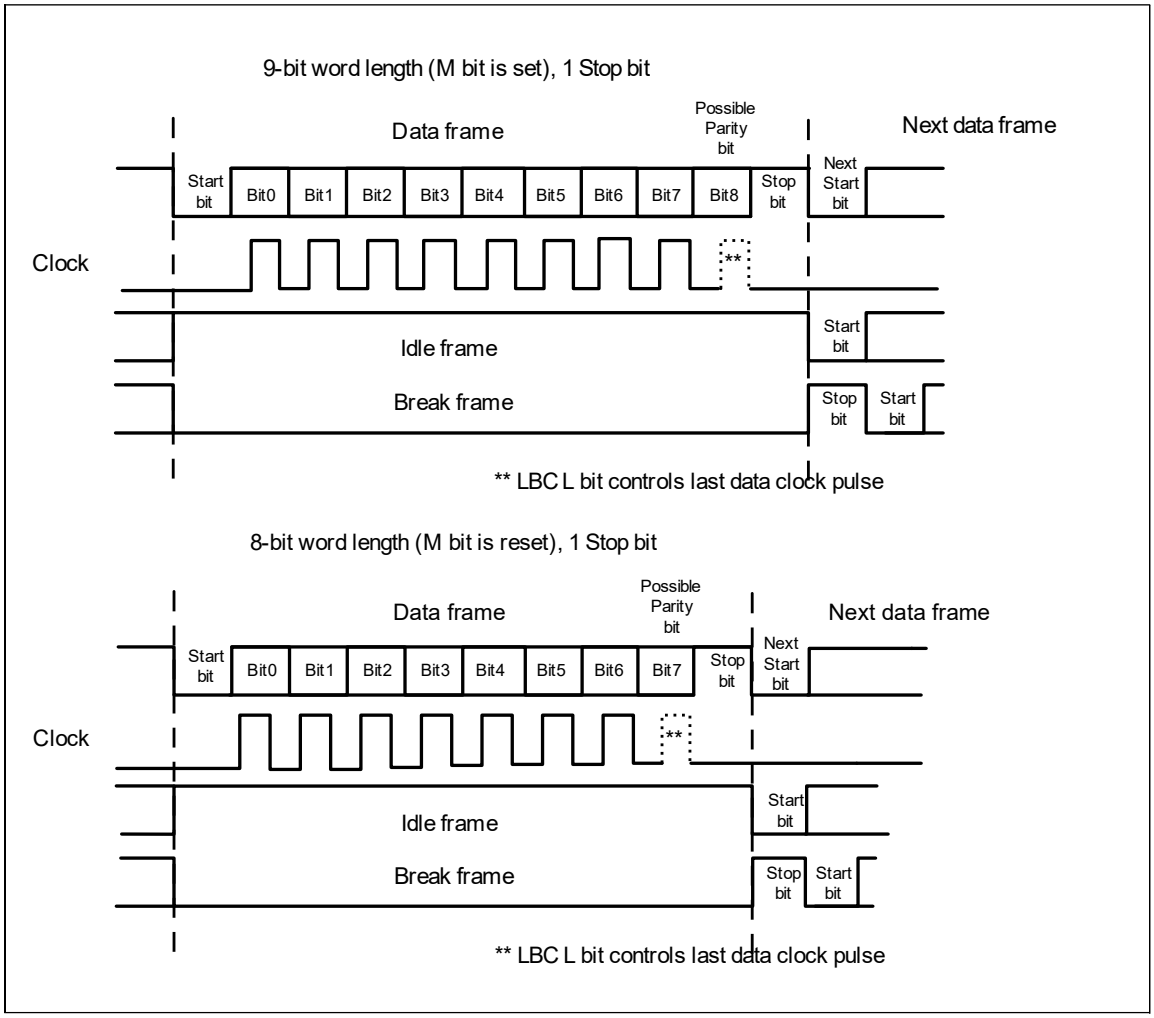



图 21-1 字长编程

## 21.3.2. 发送器

发送器可发送 8 位或 9 位的数据字，具体取决于 M 位的状态。发送使能位 (TE) 置 1 时，发送移位寄存器中的数据在 TX 引脚输出，相应的时钟脉冲在 SCLK 引脚输出。

**字符发送**

USART 发送期间，首先通过 TX 引脚移出数据的最低有效位。该模式下，USART_DR 寄存器的缓冲区 (TDR) 位于内部总线和发送移位寄存器之间。

每个字符前面都有一个起始位，其逻辑电平在一个位周期内为低电平。字符由可配置数量的停止位终止。

USART 支持以下停止位：0.5、1、1.5 和 2 个停止位。

*注意： 数据发送期间不应复位 TE 位。发送期间复位 TE 位会冻结波特率计数器，从而将损坏 TX 引脚上的数据。当前传输的数据将会丢失。*

*使能 TE 位后，将会发送空闲帧。*

**可配置的停止位**

可以在控制寄存器 2 的位 13 和 位 12 中编程将随各个字符发送的停止位的数量。

- 1 个停止位： 这是停止位数量的默认值。
- 2 个停止位： 正常 USART 模式、单线模式和调制解调器模式支持该值。

- 0.5 个停止位： 在智能卡模式下接收数据时使用。
- 个停止位： 在智能卡模式下发送和接收数据时使用。

空闲帧发送将包括停止位。

m = 0 时，中断发送是 10 个低电平位，然后是已配置数量的停止位；m = 1 时，中断发送是 11 个低电平位，然后是已配置数量的停止位。无法传送长中断（中断长度大于 10/11 个低电平位）。

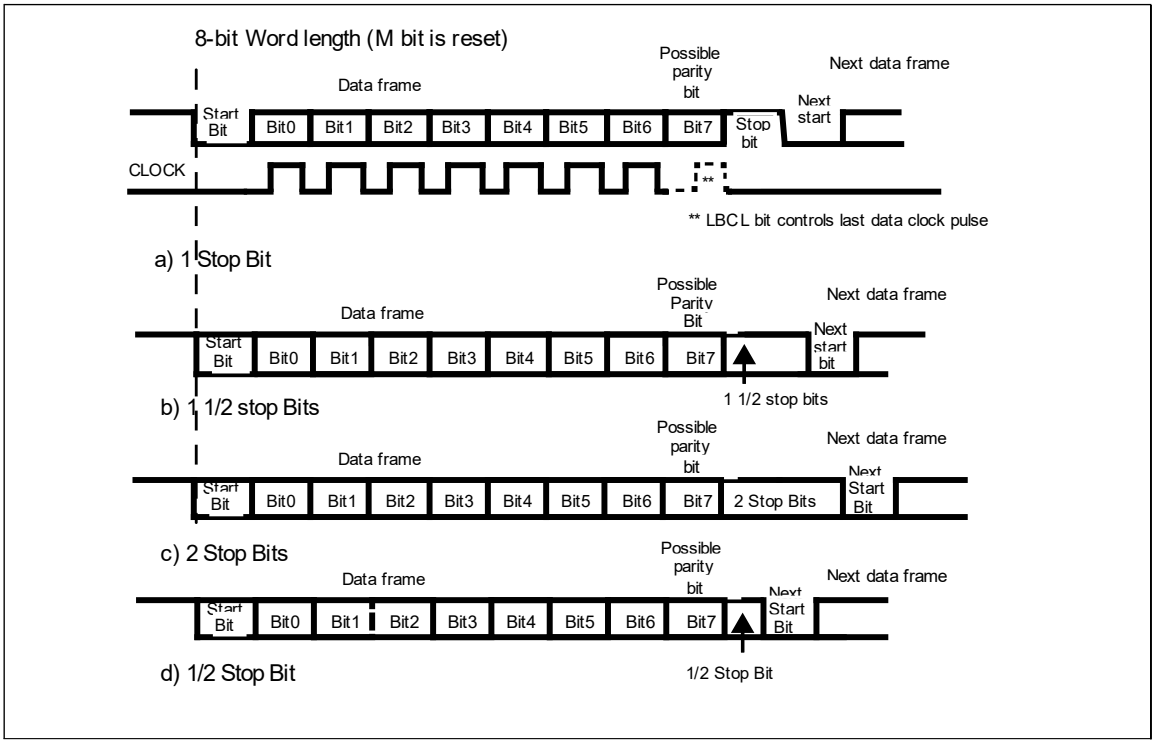



图 21-2 可配置的停止位

步骤：

- 通过向 USART_CR1 寄存器中的 UE 位写入 1 使能 USART。
- 对 USART_CR1 中的 M 位进行编程以定义字长。
- 对 USART_CR2 中的停止位数量进行编程。
- 如果将进行多缓冲区通信，请选择 USART_CR3 中的 DMA 使能 (DMAT)。按照多缓冲区通信中的解释说明配置 DMA 寄存器。
- 使用 USART_BRR 寄存器选择所需波特率。
- 将 USART_CR1 中的 TE 位置 1 以便在首次发送时发送一个空闲帧。
- 在 USART_DR 寄存器中写入要发送的数据（该操作将清零 TXE 位）。为每个要在单缓冲区模式下发送的数据重复这一步骤。
- 向 USART_DR 寄存器写入最后一个数据后，等待至 TC=1。这表明最后一个帧的传送已完成。禁止 USART 或进入暂停模式时需要此步骤，以避免损坏最后一次发送。

**单字节通信**

始终通过向数据寄存器写入数据来将 TXE 位清零。

TXE 位由硬件置 1，它表示：

- 数据已从 TDR 移到移位寄存器中且数据发送已开始。

- TDR 寄存器为空。
- USART_DR 寄存器中可写入下一个数据，而不会覆盖前一个数据。

TXEIE 位置 1 时该标志位会生成中断。

发送时，要传入 USART_DR 寄存器的写指令中存有 TDR 寄存器中的数据，该数据将在当前发送结束时复制到移位寄存器中。

未发送时，要传入 USART_DR 寄存器的写指令直接将数据置于移位寄存器中，数据发送开始时，TXE 位立即置 1。

如果帧已发送（停止位后）且 TXE 位置 1，TC 位将变为高电平。如果 USART_CR1 寄存器中的 TCIE 位置 1，将生成中断。

向 USART_DR 寄存器中写入最后一个数据后，必须等待至 TC=1，之后才可禁止 USART 或使微控制器进入低功率模式。

TC 位通过以下软件序列清零：

- 从 USART_SR 寄存器读取数据；
- 向 USART_DR 寄存器写入数据。

*注意： 还可通过向 TC 位写入"0"将其清零。建议仅在多缓冲区通信时使用此清零序列。*

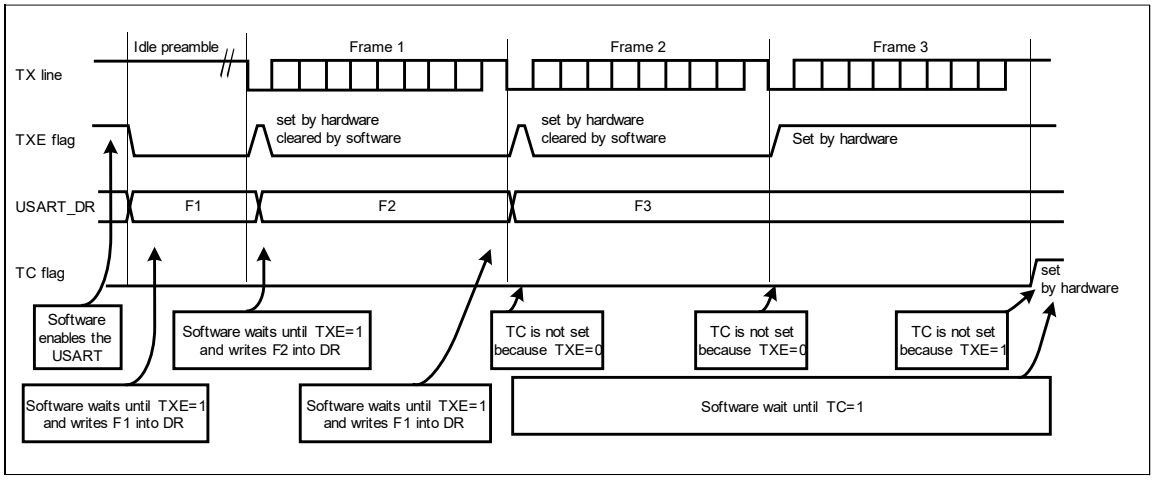



图 21-3 发送时的 TC/TXE 行为

**中断字符**

将 SBK 位置 1 将发送一个中断字符。中断帧的长度取决于 M 位。

如果 SBK 位置"1"，当前字符发送完成后，将在 TX 线路上发送一个中断字符。中断字符发送完成时（发送中断字符的停止位期间），该位由硬件复位。 USART 在上一个中断帧的末尾插入一个逻辑"1"位，以确保识别下个帧的起始位。

*注意： 如果软件在中断发送开始前对 SBK 位进行了复位，将不会发送中断字符。对于两个连续的中断，应在上一个中断的停止位发送完成后将 SBK 位置 1。*

**空闲字符**

将 TE 位置 1 会驱动 USART 在第一个数据帧之前发送一个空闲帧。

### 21.3.3. 接收器

USART 可接收 8 位或 9 位的数据字，具体取决于 USART_CR1 寄存器中的 M 位。

### 起始位检测

16 倍或 8 倍过采样时，起始位检测序列相同。

在 USART 中，识别出特定序列的采样时会检测起始位。该序列为： 1 1 1 0 X 0 X 0 X 0 0 0 0。

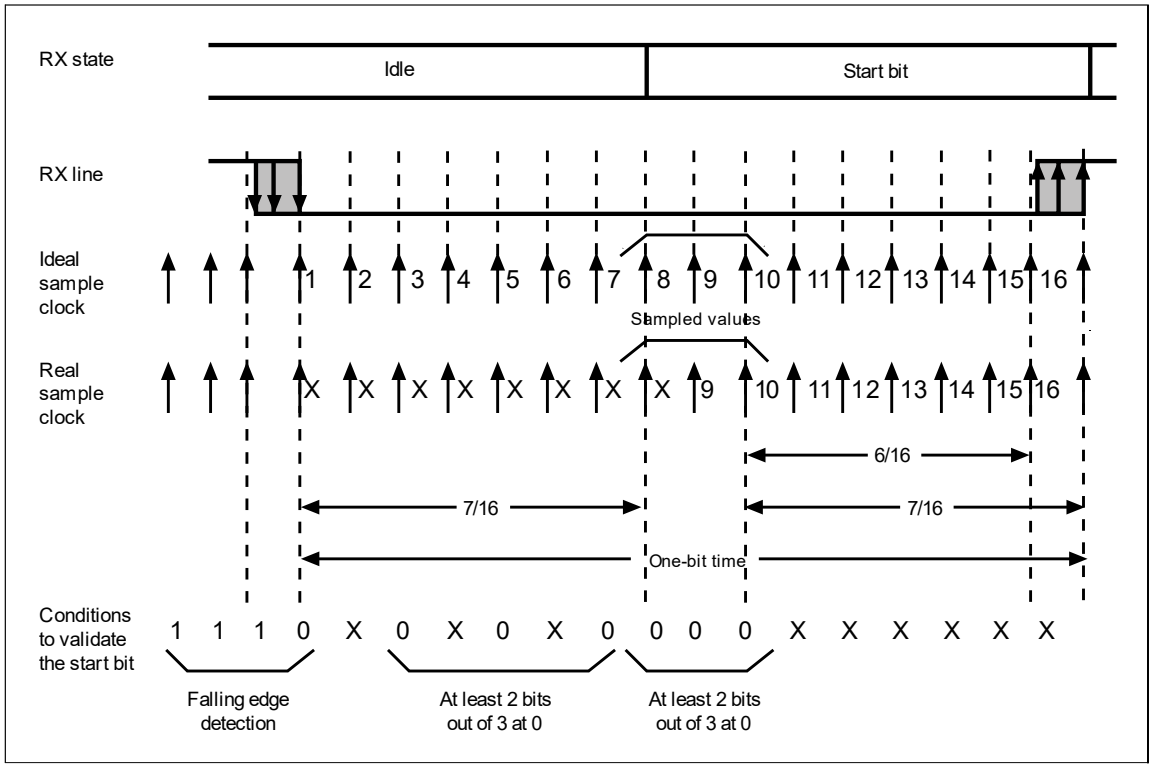



图 21-4 16 倍或 8 倍过采样时的起始位检测

*注意： 如果序列不完整，起始位检测将中止，接收器将返回空闲状态（无标志位置 1）等待下降沿。*

*如果 3 个采样位均为 0（针对第 3 位、第 5 位和第 7 位进行首次采样时检测到这 3 位均为 0；*

*针对第 8 位、第 9 位和第 10 位进行第二次采样时检测到这 3 位均为 0），可确认起始位（ RXNE 标志位置 1， RXNEIE=1 时生成中断）。*

*如果两次采样时（对第 3 位、第 5 位和第 7 位进行采样以及对第 8 位、第 9 位和第 10 位进行采样），3 个采样位中至少有 2 个为 0，则可验证起始位（RXNE 标志位置 1， RXNEIE=1 时生成中断）但 NE 噪声标志位置 1。如果不满足此条件，则启动检测中止，接收器返回空闲状态（无标志位置 1）。*

*如果其中一次采样时（对第 3 位、第 5 位和第 7 位进行采样或对第 8 位、第 9 位和第 10 位进行采样），3 个采样位中有 2 个为 0，则可验证起始位但 NE 噪声标志位置 1。*

### 字符接收

USART 接收期间，首先通过 RX 引脚移入数据的最低有效位。该模式下，USART_DR 寄存器的缓冲区 (RDR) 位于内部总线和接收移位寄存器之间。

步骤：

- 通过向 USART_CR1 寄存器中的 UE 位写入 1 使能 USART。
- 对 USART_CR1 中的 M 位进行编程以定义字长。
- 对 USART_CR2 中的停止位数量进行编程。
- 如果将进行多缓冲区通信，请选择 USART_CR3 中的 DMA 使能 (DMAR)。按照多缓冲区通信中的解释说明配置 DMA 寄存器。

- 步骤 3 使用波特率寄存器 USART_BRR 选择所需波特率。
- 将 RE 位 USART_CR1 置 1。这一操作将使能接收器开始搜索起始位。

接收到字符时

- RXNE 位置 1。这表明移位寄存器的内容已传送到 RDR。也就是说，已接收到并可读取数据（以及其相应的错误标志）。
- 如果 RXNEIE 位置 1，则会生成中断。
- 如果接收期间已检测到帧错误、噪声错误或上溢错误，错误标志位可置 1。
- 在多缓冲区模式下，每接收到一个字节后 RXNE 均置 1，然后通过 DMA 对数据寄存器执行读操作清零。
- 在单缓冲区模式下，通过软件对 USART_DR 寄存器执行读操作将 RXNE 位清零。 RXNE 标志也可以通过向该位写入零来清零。 RXNE 位必须在结束接收下一个字符前清零，以避免发生上溢错误。

*注意： 接收数据时，不应将 RE 位复位。如果接收期间禁止了 RE 位，则会中止接收当前字节。*

**中断字符**

接收到中断字符时，USART 将会按照帧错误对其进行处理。

**空闲字符**

检测到空闲帧时，处理步骤与接收到数据的情况相同；如果 IDLEIE 位为 1，则会产生中断。

**上溢错误**

如果在 RXNE 未复位时接收到字符，则会发生上溢错误。 RXNE 位清零前，数据无法从移位寄存器传送到 RDR 寄存器。

每接收到一个字节后，RXNE 标志位都将置 1。当 RXNE 标志位是 1 时，如果在接收到下一个数据或尚未处理上一个 DMA 请求时，则会发生上溢错误。发生上溢错误时：

- ORE 位置 1。
- RDR 中的内容不会丢失。对 USART_DR 执行读操作时可使用先前的数据。
- 移位寄存器将被覆盖。之后，上溢期间接收到的任何数据都将丢失。
- 如果 RXNEIE 位置 1 或 EIE 与 DMAR 位均为 1，则会生成中断。
- 通过先后对 USART_SR 寄存器和 USART_DR 寄存器执行读操作将 ORE 位清除。
- 在 FIFO 模式下 ORE 位清除会清空接收缓存区的所有数据。

注意： ORE 位置 1 时表示至少 1 个数据丢失。存在两种可能：

- 如果 RXNE=1，则最后一个有效数据存储于接收寄存器 RDR 中并且可进行读取；
- 如果 RXNE=0，则表示最后一个有效数据已被读取，因此 RDR 中没有要读取的数据。接收到新（丢失）数据的同时已读取 RDR 中的最后一个有效数据时，会发生该情况。读取序列期间（在 USART_SR 寄存器读访问与 USART_DR 读访问之间）接收到新数据时也会发生该情况。

**选择合适的过采样方法**

接收器采用不同的用户可配置过采样技术（除了同步模式下），可以从噪声中提取有效数据。

可通过编程 USART_CR1 寄存器中的 OVER8 位来选择采样方法，且采样时钟可以是波特率时钟的 16 倍或 8 倍。

根据应用：

- 选择 8 倍过采样 (OVER8=1) 以获得更高的速度（高达 $f_{PCLK}/8$）。这种情况下接收器对时钟偏差的最大容差将会降低。

- 选择 16 倍过采样 (OVER8=0) 以增加接收器对时钟偏差的容差。这种情况下，最大速度限制为最高 $f_{PCLK}/16$。

可通过编程 USART_CR3 寄存器中的 ONEBIT 位选择用于评估逻辑电平的方法。有两种选择：

- 在已接收位的中心进行三次采样，从而进行多数表决。这种情况下，如果用于多数表决的 3 次采样结果不相等，NF 位置 1。
- 在已接收位的中心进行单次采样。

  根据应用：

  - 在噪声环境下工作时，请选择三次采样的多数表决法 (ONEBIT=0)；在检测到噪声时请拒绝数据，因为这表示采样过程中产生了干扰。
  - 线路无噪声时请选择单次采样法 (ONEBIT=1) 以增加接收器对时钟偏差的容差。这种情况下 NF 位始终不会置 1。

帧中检测到噪声时：

- 在 RXNE 位的上升沿时 NF 位置 1。
- 无效数据从移位寄存器传送到 USART_DR 寄存器。
- 单字节通信时无中断产生。然而，在 RXNE 位产生中断时，该位出现上升沿。多缓冲区通信时，USART_CR3 寄存器中的 EIE 位置 1 时将发出中断。

通过先后对 USART_SR 寄存器和 USART_DR 寄存器执行读操作将 NF 位清零。

*注意： 智能卡、IrDA 和 LIN 模式下不可采用 8 倍过采样。在这些模式下， OVER8 位由硬件强制清零。*

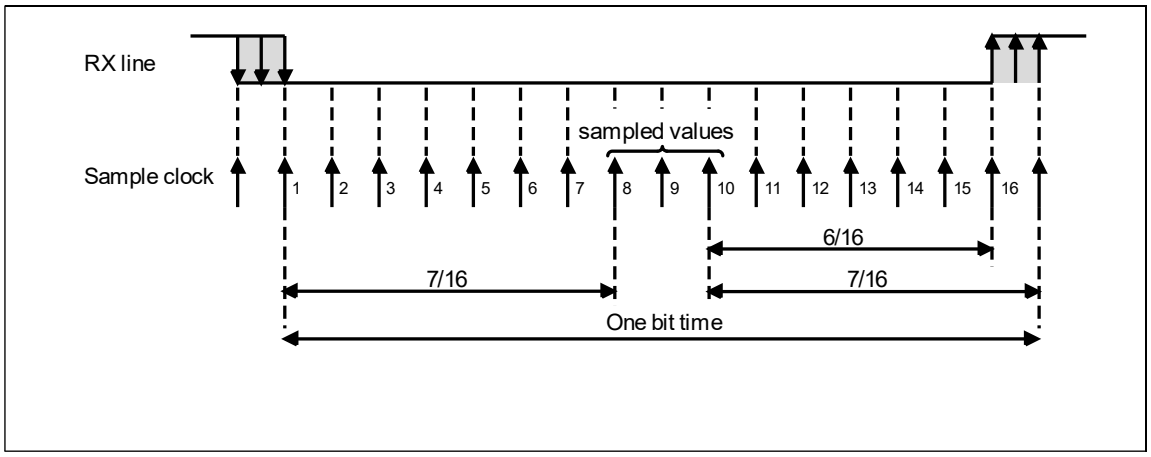



图 21-5 16 倍过采样时的数据采样

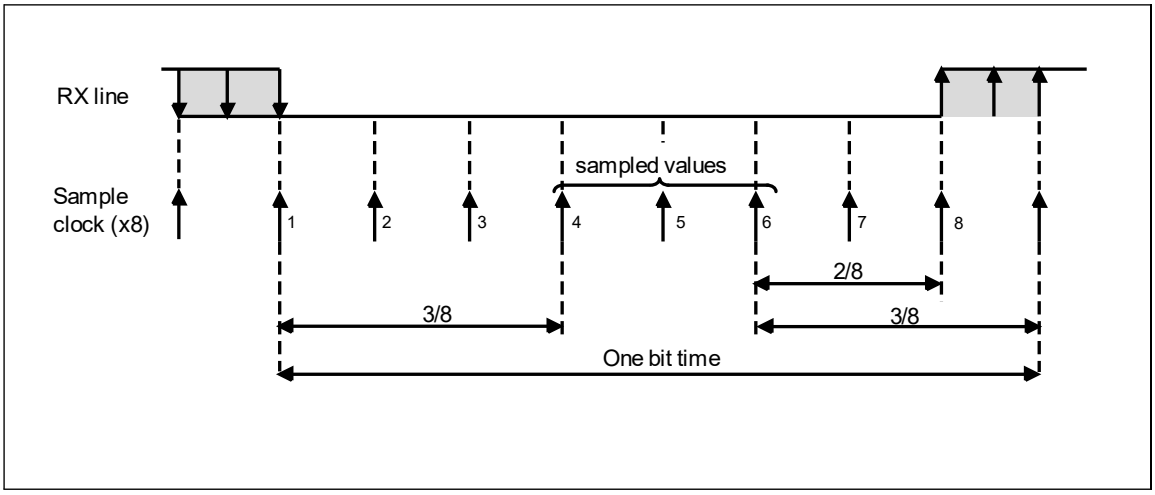



图 21-6 8 倍过采样时的数据采样

表 21-1 通过采样数据进行噪声检测

| 采样值 | NE 状态 | 接收的位值 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 1 | 0 |
| 10 | 1 | 0 |
| 11 | 1 | 1 |
| 100 | 1 | 0 |
| 101 | 1 | 1 |
| 110 | 1 | 1 |
| 111 | 0 | 1 |

**帧错误**

以下情况下将检测到帧错误：

接收数据时未在预期时间内识别出停止位，从而出现同步失效或过度的噪声。

检测到帧错误时：

- FE 位由硬件置 1。
- 无效数据从移位寄存器传送到 USART_DR 寄存器。
- 单字节通信时无中断产生。然而，在 RXNE 位产生中断时，该位出现上升沿。多缓冲区通信时，USART_CR3 寄存器中的 EIE 位置 1 时将发出中断。

通过先后对 USART_SR 寄存器和 USART_DR 寄存器执行读操作将 FE 位清零。

**接收期间可配置的停止位**

可通过控制寄存器 2 中的控制位配置要接收的停止位的数量 - 可以是 1 或 2 个（正常模式下），也可以是 0.5 或 1.5 个（智能卡模式下）。

- 0.5 个停止位（在智能卡模式下接收时）：不会对 0.5 个停止位进行采样。结果，选择 0.5 个停止位时，无法检测到帧错误和中断帧。
- 1 个停止位：将在第 8、第 9 和第 10 次采样时对 1 个停止位进行采样。
- 1.5 个停止位（在智能卡模式下）：在智能卡模式下发送时，设备必须检查数据是否正确发送。因此必须使能接收器块（USART_CR1 寄存器中的 RE =1）并检查停止位，以测试智能卡是否已检测到奇偶校验错误。发生奇偶校验错误时，智能卡会在采样时将数据信号强制为低电平，即 NACK 信号-，该信号被标记为帧错误。之后，FE 标志在 1.5 个停止位的末尾由 RXNE 置 1。在第 16、第 17 和第

18 次采样时对 1.5 个停止位进行采样（停止位采样开始后维持 1 个波特时钟周期）。1.5 个停止位可分为 2 个部分： 0.5 个波特时钟周期（未发生任何动作），然后是 1 个正常的停止位周期（一半时间处进行采样）。

- 2 个停止位：采样 2 个停止位时在第 8、第 9 和第 10 次采样时对第一个停止位进行采样。如果在第一个停止位期间检测到帧错误，则帧错误标志位将会置 1。发生帧错误时不检测第 2 个停止位。RXNE 标志将在第一个停止位末尾时置 1。

### 21.3.4. 小数波特率生成

对 USARTDIV 的尾数值和小数值进行编程时，接收器和发送器（Rx 和 Tx）的波特率均设置为相同值。

**公式 1：适用于标准 USART（包括 SPI 模式）的波特率：**

$$Tx/Rx\text{波特率} = \frac{f_{CK}}{8 \times (2 - OVER8) \times USARTDIV}$$

**公式 1：智能卡、 LIN 和 IrDA 模式下的波特率：**

$$Tx/Rx\text{波特率} = \frac{f_{CK}}{16 \times USARTDIV}$$

USARTDIV 是一个存放在 USART_BRR 寄存器中的无符号定点数。

- 当 OVER8=0 时，小数部分编码为 4 位并通过 USART_BRR 寄存器中的 DIV_fraction[3:0]位编程。
- 当 OVER8=1 时，小数部分编码为 3 位并通过 USART_BRR 寄存器中的 DIV_fraction[2:0]位编程，此时 DIV_fraction[3] 位必须保持清零状态。

*注意： 对 USART_BRR 执行写操作后，波特率计数器更新为波特率寄存器中的新值。因此，波特率寄存器的值不应在通信时发生更改。*

**OVER8=0 时如何从 USART_BRR 寄存器值中获取 USARTDIV**

**示例 1：**

如果 DIV_Mantissa = 0d27 且 DIV_Fraction = 0d12 (USART_BRR = 0x1BC)，则

尾数 (USARTDIV) = 0d27

小数 (USARTDIV) = 12/16 = 0d0.75

因此 USARTDIV = 0d27.75

**示例 2：**

要设定 USARTDIV = 0d25.62

这将导致：

DIV_Fraction = 16*0d0.62 = 0d9.92

最接近的实数为 0d10 = 0xA

DIV_Mantissa = 尾数 (0d25.620) = 0d25 = 0x19

则 USART_BRR = 0x19A，因此 USARTDIV = 0d25.625

**示例 3：**

要设定 USARTDIV = 0d50.99

这将导致：

DIV_Fraction = 16*0d0.99 = 0d15.84

最接近的实数为 0d16 = 0x10 => DIV_frac[3:0] 溢出 => 尾数必须添加进位

DIV_Mantissa = 尾数（ 0d50.990 + 进位） = 0d51 = 0x33

则 USART_BRR = 0x330，因此 USARTDIV = 0d51.000

**OVER8=1 时如何从 USART_BRR 寄存器中获取 USARTDIV**

**示例 1：**

如果 DIV_Mantissa = 0x27 且 DIV_Fraction[2:0]= 0d6 (USART_BRR = 0x1B6)，则

尾数 (USARTDIV) = 0d27

小数 (USARTDIV) = 6/8 = 0d0.75

因此 USARTDIV = 0d27.75

**示例 2：**

要设定 USARTDIV = 0d25.62

这将导致：

DIV_Fraction = 8*0d0.62 = 0d4.96

最接近的实数为 0d5 = 0x5

DIV_Mantissa = 尾数 (0d25.620) = 0d25 = 0x19

则 USART_BRR = 0x195 => USARTDIV = 0d25.625

**示例 3：**

要设定 USARTDIV = 0d50.99

这将导致：

DIV_Fraction = 8*0d0.99 = 0d7.92

最接近的实数为 0d8 = 0x8 => DIV_frac[2:0] 溢出 => 尾数必须添加进位

DIV_Mantissa = 尾数（ 0d50.990 + 进位） = 0d51 = 0x33

则 USART_BRR = 0x0330 => USARTDIV = 0d51.000

表 21-2 采用 16 倍过采样时，在 $f_{PCLK}$ = 42/84MHz 下编程波特率时的误差计算

| 16 倍过采样时 (OVER8=0) | | | | | | | |
|---|---|---|---|---|---|---|---|
| 波特率 | | $f_{PCLK}$ = 42 MHz | | | $f_{PCLK}$ = 84 MHz | | |
| 序号 | 所需值 | 实际值 | 波特率寄存器中编程的值 | 误差% = (计算值 - 所需值)/ 所需波特率 | 实际值 | 波特率寄存器中编程的值 | 误差 % |
| 1 | 1.2 KBps | 1.2 KBps | 2187.5 | 0 | 1.2 KBps | 4375 | 0 |
| 2 | 2.4 KBps | 2.4 KBps | 1093.75 | 0 | 2.4 KBps | 2187.5 | 0 |
| 3 | 9.6 KBps | 9.6 KBps | 273.4375 | 0 | 9.6 KBps | 546.875 | 0 |
| 4 | 19.2 KBps | 19.195 KBps | 136.75 | 0.02 | 19.2 KBps | 273.4375 | 0 |
| 5 | 38.4 KBps | 38.391 KBps | 68.375 | 0.02 | 38.391 KBps | 136.75 | 0.02 |
| 6 | 57.6 KBps | 57.613 KBps | 45.5625 | 0.02 | 57.613 KBps | 91.125 | 0.02 |
| 7 | 115.2 KBps | 115.068 KBps | 22.8125 | 0.11 | 115.226 KBps | 45.5625 | 0.02 |

| 16 倍过采样时 (OVER8=0) | | | | | | | |
|---|---|---|---|---|---|---|---|
| 8 | 230.4 KBps | 230.769 KBps | 11.375 | 0.16 | 230.137 KBps | 22.8125 | 0.11 |
| 9 | 460.8 KBps | 461.538 KBps | 5.6875 | 0.16 | 461.538 KBps | 11.375 | 0.16 |
| 10 | 921.6 KBps | 913.043 KBps | 2.875 | 0.93 | 923.076 KBps | 5.6875 | 0.93 |
| 11 | 1.792 MBps | 1.826 MBps | 1.4375 | 1.9 | 1.787 MBps | 2.9375 | 0.27 |
| 12 | 1.8432 MBps | 1.826 MBps | 1.4375 | 0.93 | 1.826 MBps | 2.875 | 0.93 |
| 13 | 3.584 MBps | N.A | N.A | N.A | 3.652 MBps | 1.4375 | 1.9 |
| 14 | 3.6864 MBps | N.A | N.A | N.A | 3.652 MBps | 1.4375 | 0.93 |
| 15 | 7.168 MBps | N.A | N.A | N.A | N.A | N.A | N.A |
| 16 | 7.3728 MBps | N.A | N.A | N.A | N.A | N.A | N.A |
| 17 | 9 MBps | N.A | N.A | N.A | N.A | N.A | N.A |
| 18 | 10.5 MBps | N.A | N.A | N.A | N.A | N.A | N.A |

表 21-3 采用 8 倍过采样时，在 $f_{PCLK}$ = 42/84MHz 下编程波特率时的误差计算

| 8 倍过采样 (OVER8=1) | | | | | | | |
|---|---|---|---|---|---|---|---|
| 波特率 | | $f_{PCLK}$ = 42 MHz | | | $f_{PCLK}$ = 84 MHz | | |
| 序号 | 所需值 | 实际值 | 波特率寄存器中编程的值 | 误差 % =（计算值 - 所需值）/所需波特率 | 实际值 | 波特率寄存器中编程的值 | 误差 % |
| 1 | 1.2 KBps | 1.2 KBps | 4375 | 0 | 1.2 KBps | 8750 | 0 |
| 2 | 2.4 KBps | 2.4 KBps | 2187.5 | 0 | 2.4 KBps | 4375 | 0 |
| 3 | 9.6 KBps | 9.6 KBps | 546.875 | 0 | 9.6 KBps | 1093.75 | 0 |
| 4 | 19.2 KBps | 19.195 KBps | 273.5 | 0.02 | 19.2 KBps | 546.875 | 0 |
| 5 | 38.4 KBps | 38.391 KBps | 136.75 | 0.02 | 38.391 KBps | 273.5 | 0.02 |
| 6 | 57.6 KBps | 57.613 KBps | 91.125 | 0.02 | 57.613 KBps | 182.25 | 0.02 |
| 7 | 115.2 KBps | 115.068 KBps | 45.625 | 0.11 | 115.226 KBps | 91.125 | 0.02 |
| 8 | 230.4 KBps | 230.769 KBps | 22.75 | 0.11 | 230.137 KBps | 45.625 | 0.11 |
| 9 | 460.8 KBps | 461.538 KBps | 11.375 | 0.16 | 461.538 KBps | 22.75 | 0.16 |
| 10 | 921.6 KBps | 913.043 KBps | 5.75 | 0.93 | 923.076 KBps | 11.375 | 0.93 |
| 11 | 1.792 MBps | 1.826 MBps | 2.875 | 1.9 | 1.787Mbps | 5.875 | 0.27 |
| 12 | 1.8432 MBps | 1.826 MBps | 2.875 | 0.93 | 1.826 MBps | 5.75 | 0.93 |
| 13 | 3.584 MBps | 3.5 MBps | 1.5 | 2.34 | 3.652 MBps | 2.875 | 1.9 |
| 14 | 3.6864 MBps | 3.82 MBps | 1.375 | 3.57 | 3.652 MBps | 2.875 | 0.93 |
| 15 | 7.168 MBps | N.A | N.A | N.A | 7 MBps | 1.5 | 2.34 |
| 16 | 7.3728 MBps | N.A | N.A | N.A | 7.636 MBps | 1.375 | 3.57 |
| 17 | 9 MBps | N.A | N.A | N.A | 9.333 MBps | 1.125 | 3.7 |
| 18 | 10.5 MBps | N.A | N.A | N.A | 10.5 MBps | 1 | 0 |

表 21-4 采用 16 倍过采样时，在 $f_{PCLK}$ = 45/90MHz 下编程波特率时的误差计算

| 16 倍过采样 (OVER8=0) | | | | | | | |
|---|---|---|---|---|---|---|---|
| 波特率 | | $f_{PCLK}$ = 45 MHz | | | $f_{PCLK}$ =90 MHz | | |
| 序号 | 所需值 | 实际值 | 波特率寄 | 误差 % =（计算 | 实际值 | 波特率寄 | 误差% |

| 16 倍过采样 (OVER8=0) | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | 存器中编程的值 | 值 - 所需值）/所需波特率 | | 存器中编程的值 | |
| 1 | 1.2 KBps | 1.2 KBps | 2343.75 | 0 | 1.2 KBps | 4687.5 | 0 |
| 2 | 2.4 KBps | 2.4 KBps | 1171.875 | 0 | 2.4 KBps | 2343.75 | 0 |
| 3 | 9.6 KBps | 9.601 KBps | 293.9375 | 0.01 | 9.6 KBps | 585.9375 | 0 |
| 4 | 19.2 KBps | 19.197 KBps | 146.05 | 0.02 | 19.202 KBps | 292.9375 | 0.01 |
| 5 | 38.4 KBps | 38.395 KBps | 73.25 | 0.01 | 38.395 KBps | 146.5 | 0.01 |
| 6 | 57.6 KBps | 57.618 KBps | 48.8125 | 0.03 | 57.618 KBps | 97.625 | 0.03 |
| 7 | 115.2 KBps | 115.089 KBps | 24.4375 | 0.09 | 115.236 KBps | 48.8125 | 0.03 |
| 8 | 230.4 KBps | 230.769 KBps | 12.1875 | 0.16 | 230.179 KBps | 24.4375 | 0.09 |
| 9 | 460.8 KBps | 459.183 KBps | 6.125 | 0.35 | 461.538 KBps | 12.1875 | 0.16 |
| 10 | 921.6 KBps | 918.367 KBps | 3.0625 | 0.35 | 918.367 KBps | 6.125 | 0.35 |
| 11 | 1.792 MBps | 1.8 MBps | 1.5625 | 0.45 | 1.8 MBps | 3.125 | 0.45 |
| 12 | 1.8432 MBps | 1.875 MBps | 1.5 | 1.73 | 1.837 MBps | 3.0625 | 0.35 |
| 13 | 3.584 MBps | N.A | N.A | N.A | 3.6 MBps | 1.5625 | 0.45 |
| 14 | 3.6864 MBps | N.A | N.A | N.A | 3.75 MBps | 1.5 | 1.73 |
| 15 | 7.168 MBps | N.A | N.A | N.A | N.A | N.A | N.A |
| 16 | 7.3728 MBps | N.A | N.A | N.A | N.A | N.A | N.A |
| 17 | 9 MBps | N.A | N.A | N.A | N.A | N.A | N.A |
| 18 | 10.5 MBps | N.A | N.A | N.A | N.A | N.A | N.A |

表 21-5 采用 8 倍过采样时，在 $f_{PCLK}$ = 45/90MHz 下编程波特率时的误差计算

| 8 倍过采样 (OVER8=1) | | | | | | | |
|---|---|---|---|---|---|---|---|
| 波特率 | | $f_{PCLK}$ = 45 MHz | | | $f_{PCLK}$ =90 MHz | | |
| 序号 | 所需值 | 实际值 | 波特率寄存器中编程的值 | 误差 % = (计算值 - 所需值）/所需波特率 | 实际值 | 波特率寄存器中编程的值 | 误差% |
| 1 | 1.2 KBps | 1.2 KBps | 4687.5 | 0 | 1.2 KBps | 9375 | 0 |
| 2 | 2.4 KBps | 2.4 KBps | 2343.75 | 0 | 2.4 KBps | 4687.5 | 0 |
| 3 | 9.6 KBps | 9.601 KBps | 585.875 | 0.01 | 9.6 KBps | 1171.875 | 0 |
| 4 | 19.2 KBps | 19.197 KBps | 293 | 0.02 | 19.202 KBps | 585.875 | 0.01 |
| 5 | 38.4 KBps | 38.395 KBps | 146.5 | 0.01 | 38.395 KBps | 293 | 0.01 |
| 6 | 57.6 KBps | 57.618 KBps | 97.625 | 0.03 | 57.618 KBps | 195.25 | 0.03 |
| 7 | 115.2 KBps | 115.089 KBps | 48.875 | 0.09 | 115.236 KBps | 97.625 | 0.03 |

| 8 倍过采样 (OVER8=1) | | | | | | | |
|---|---|---|---|---|---|---|---|
| 8 | 230.4 KBps | 230.769 KBps | 24.375 | 0.16 | 230.179 KBps | 48.875 | 0.09 |
| 9 | 460.8 KBps | 459.183 KBps | 12.25 | 0.35 | 461.538 KBps | 24.375 | 0.16 |
| 10 | 921.6 KBps | 918.367 KBps | 6.125 | 0.35 | 918.367 KBps | 12.25 | 0.35 |
| 11 | 1.792 MBps | 1.8 MBps | 3.125 | 0.45 | 1.8 MBps | 6.25 | 0.45 |
| 12 | 1.8432 MBps | 1.875 MBps | 3 | 1.73 | 1.837 MBps | 6.125 | 0.35 |
| 13 | 3.584 MBps | 3.462 MBps | 1.625 | 3.54 | 3.6 MBps | 3.125 | 0.45 |
| 14 | 3.6864 MBps | 3.75 MBps | 1.5 | 1.73 | 3.75 MBps | 3 | 1.73 |
| 15 | 7.168 MBps | N.A | N.A | N.A | 6.923 MBps | 1.625 | 3.54 |
| 16 | 7.3728 MBps | N.A | N.A | N.A | 7.5 MBps | 1.5 | 1.72 |
| 17 | 9 MBps | N.A | N.A | N.A | 9 MBps | 1.25 | 0 |
| 18 | 10.5 MBps | N.A | N.A | N.A | 10 MBps | 1.125 | 5 |

### 21.3.5. USART 接收器对时钟偏差的容差

仅当总时钟系统偏差小于 USART 接收器的容差时，USART 异步接收器器才能正常工作。影响总偏差的因素包括：

- DTRA：发送器误差引起的偏差（其中还包括发送器本地振荡器的偏差）；
- DQUANT：接收器的波特率量化引起的误差；
- DREC：接收器本地振荡器的偏差；
- DTCL：传输线路引起的偏差（通常是由于收发器所引起，它可能会在低电平到高电平转换时序与高电平到低电平转换时序之间引入不对称）。

DTRA + DQUANT + DREC + DTCL < USART 接收器的容差。

对于正常接收数据， USART 接收器的容差等于所容许的最大偏差，具体取决于以下选择：

- 由 USART_CR1 寄存器中的 M 位定义的 10 位或 11 位字符长度；
- 由 USART_CR1 寄存器中的 OVER8 位定义的 8 倍或 16 倍过采样；
- 是否使用小数波特率；
- 使用 1 位或 3 位对数据进行采样，取决于 USART_CR3 寄存器中 ONEBIT 位的值。

表 21-6 DIV_Fraction 为 0 时的 USART 接收器容差

| M 位 | OVER8 位 = 0 | | OVER8 位 = 1 | |
|---|---|---|---|---|
| | ONEBIT=0 | ONEBIT=1 | ONEBIT=0 | ONEBIT=1 |
| 0 | 3.75% | 4.38% | 2.50% | 3.75% |
| 1 | 3.41% | 3.97% | 2.27% | 3.41% |

表 21-7 DIV_Fraction 不为 0 时的 USART 接收器容差

| M 位 | OVER8 位 = 0 | | OVER8 位 = 1 | |
|---|---|---|---|---|
| | ONEBIT=0 | ONEBIT=1 | ONEBIT=0 | ONEBIT=1 |
| 0 | 3.33% | 3.88% | 2% | 3% |
| 1 | 3.03% | 3.53% | 1.82% | 2.73% |

### 21.3.6. 多处理器通信

可以与 USART 进行多处理器通信（多个 USART 连接在一个网络中）。例如，其中一个 USART 可以是主 USART，其 TX 输出与其它 USART 的 RX 输入相连接。其它 USART 为从 USART，其各自的 TX 输出在逻辑上通过与运算连在一起，并与主 USART 的 RX 输入相连接。

在多处理器配置中，理想情况下通常只有预期的消息接收方主动接收完整的消息内容，从而减少由所有未被寻址的接收器造成的冗余 USART 服务开销。

可通过静音功能将未被寻址的器件置于静音模式下。在静音模式下：

- 不得将接收状态位置 1。
- 禁止任何接收中断。
- USART_CR1 寄存器中的 RWU 位置 1。RWU 可由硬件自动控制，或在特定条件下由软件写入。

根据 USART_CR1 寄存器中 WAKE 位的设置，USART 可使用以下两种方法进入或退出静音模式：

- 如果 WAKE 位被复位，则进行空闲线路检测，
- 如果 WAKE 位置 1，则进行地址标记检测。

#### 空闲线路检测 (WAKE=0)

当向 RWU 位写入 1 时，USART 进入静音模式。

当检测到空闲帧时，它会被唤醒。此时 RWU 位会由硬件清零，但 USART_SR 寄存器中的 IDLE 位不会置 1。还可通过软件向 RWU 位写入 0。

使用空闲线路检测时静音模式行为的示例：

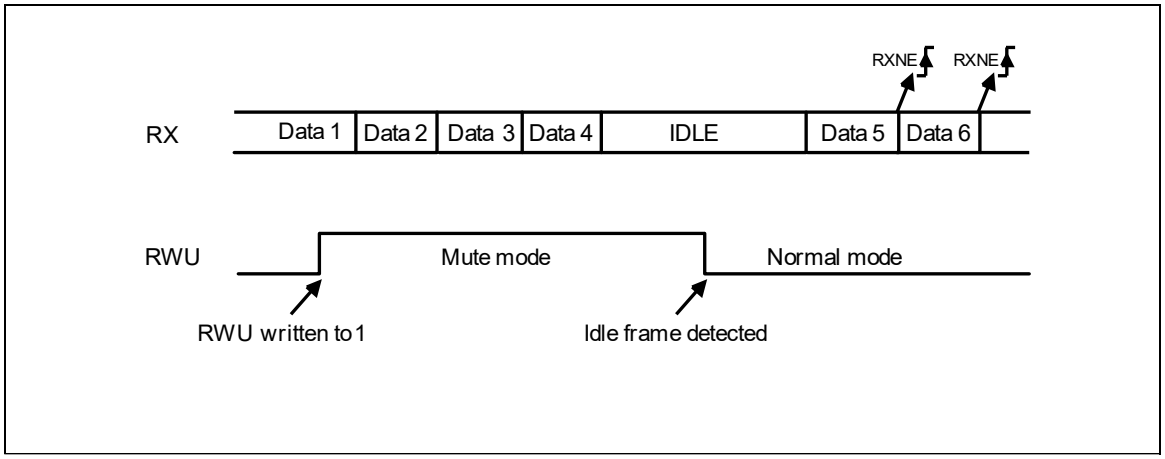



图 21-7 使用空闲线路检测时的静音模式

#### 地址标记检测 (WAKE=1)

在此模式下，如果字节的 MSB 为 1，则将这些字节识别为地址，否则将其识别为数据。在地址字节中，目标接收器的地址位于 4 个 LSB 上。接收器会将此 4 位字与其地址进行比较，该接收器的地址在 USART_CR2 寄存器的 ADD 位中进行设置。

当接收到与其编程地址不匹配的地址字符时，USART 会进入静音模式。此时，RWU 位将由硬件置 1。由于当时 USART 已经进入了静音模式，所以 RXNE 标志不会针对此地址字节置 1，也不会发出中断或 DMA 请求。

当接收到与编程地址匹配的地址字符时，它会退出静音模式。然后 RWU 位被清零，可以开始正常接收后续字节。由于 RWU 位已清零， RXNE 位会针对地址字符置 1。

当接收器的缓冲区不包含任何数据（USART_SR 寄存器中 RXNE=0）时，可向 RWU 位写入 0 或 1。否则会忽略写尝试。

使用地址标记检测时静音模式行为的示例:

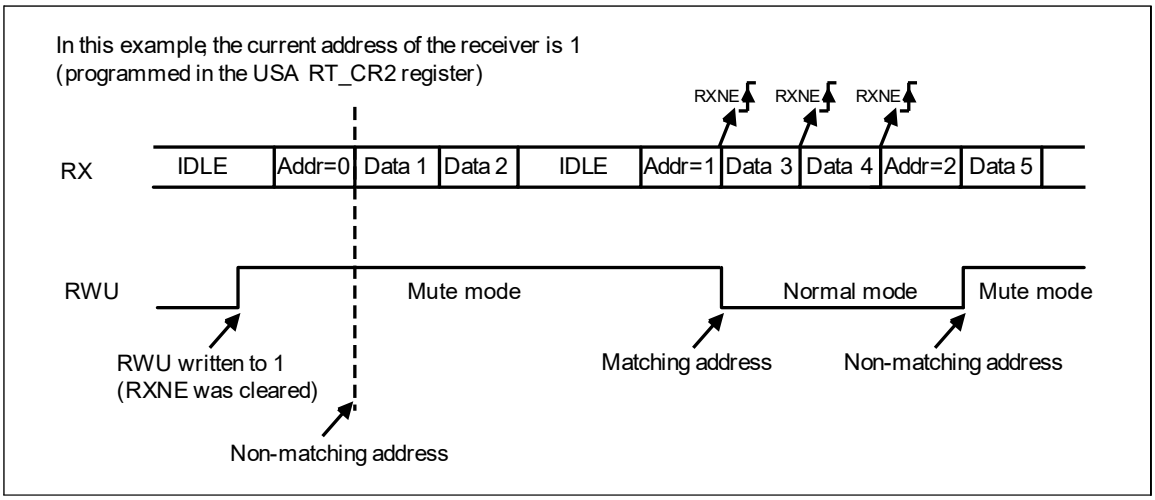



图 21-8 使用地址标记检测时的静音模式

## 21.3.7. 奇偶校验控制

将 USART_CR1 寄存器中的 PCE 位置 1，可以使能奇偶校验控制（发送时生成奇偶校验位，接收时进行奇偶校验检查）。根据 M 位定义的帧长度，表中列出了可能的 USART 帧格式。

表 21-8 帧格式

| M 位 | PCE 位 | USART 帧(1) |
|---|---|---|
| 0 | 0 | \| SB \| 8 位数据 \| STB \| |
| 0 | 1 | \| SB \| 7 位数据 \| PB \| STB \| |
| 1 | 0 | \| SB \| 9 位数据 \| STB \| |
| 1 | 1 | \| SB \| 8 位数据 PB \| STB \| |

1. 图注： SB：起始位， STB：停止位， P：奇偶校验位。

**偶校验**

对奇偶校验位进行计算，使帧和奇偶校验位中“1”的数量为偶数（帧由 7 个或 8 个 LSB 位组成，具体取决于 M 等于 0 还是 1）。

例如：数据=00110101； 4 个位置 1 =>如果选择偶校验（USART_CR1 寄存器中的 PS 位 = 0），则校验位是 0。

**奇校验**

对奇偶校验位进行计算，使帧和奇偶校验位中“1”的数量为奇数（帧由 7 个或 8 个 LSB 位组成，具体取决于 M 等于 0 还是 1）。

例如：数据=00110101； 4 个位置 1 =>如果选择奇校验（USART_CR1 寄存器中的 PS 位 = 1），则校验位是 1。

**接收时进行奇偶校验检查**

如果奇偶校验检查失败，则 USART_SR 寄存器中的 PE 标志置 1 ；如果 USART_CR1 寄存器中 PEIE 位置 1，则会生成中断。PE 标志由软件序列清零（从状态寄存器中读取，然后对 USART_DR 数据寄存器执行读或写访问）。

*注意：如果被地址标记唤醒：会使用数据的 MSB 位而非奇偶校验位来识别地址。此外，接收器不会对地址数据进行奇偶校验检查（奇偶校验出错时，*

*PE 不置 1）。*

**发送时的奇偶校验生成**

如果 USART_CR1 寄存器中的 PCE 位置 1，则在数据寄存器中所写入数据的 MSB 位会进行传送，但是会由奇偶校验位进行更改（如果选择偶校验 (PS=0)，则“1”的数量为偶数；如果选择奇校验 (PS=1)，则“1”的数量为奇数）。

*注意：用于管理发送过程的软件程序可以激活软件序列，进而将 PE 标志清零（从状态寄存器中读取，然后对数据寄存器执行读或写访问）。在半双工模式下工作时（具体取决于软件），这可能会导致 PE 标志意外清零。*

### 21.3.8. LIN(局域互连网络)模式

通过将 USART_CR2 寄存器中的 LINEN 位置 1 来选择 LIN 模式。在 LIN 模式下，必须将以下位清零：

- USART_CR2 寄存器中的 CLKEN 位，
- USART_CR3 寄存器中的 STOP[1:0]、SCEN、HDSEL 和 IREN 位。

**LIN 发送**

与正常的 USART 发送相比，在 LIN 主器件中发送时必须采用发送器中介绍的相同步骤，同时还具有以下区别：

- M 位清零以配置 8 位字长度。
- LINEN 位置 1 以进入 LIN 模式。此时，将 SBK 位置 1 会发送 13 个“0”位作为断路字符。然后会发送值为“1”的位以进行下一启动检测。

**LIN 接收**

断路检测电路在 USART 接口上实现。该检测完全独立于正常的 USART 接收器。在空闲状态或某个帧期间，只要发生断路即可检测出来。

接收器（USART_CR1 寄存器中 RE=1）使能后，电路便开始监测启动信号的 RX 输入。检测起始位的方法与搜索断路字符或数据的方法相同。检测到起始位后，电路会对接下来的位进行采样，方法与数据采样相同（第 8、第 9 和第 10 次采样）。如果 10 个（USART_CR2 寄存器中 LBDL = 0 时）或 11 个（USART_CR2 寄存器中 LBDL=1 时）连续位均检测为“0”，且其后跟随分隔符，则 USART_SR 寄存器中的 LBD 标志将会置 1。如果 LBDIE 位 =1，则会生成中断。在验证断路前，会对分隔符进行检查，因为它表示 RX 线路已恢复到高电平。

如果在第 10 或第 11 次采样前已对“1”采样，则断路检测电路会取消当前检测，并重新搜索起始位。

如果禁止 LIN 模式 (LINEN=0)，接收器会作为正常的 USART 继续工作，不会再进行断路检测。

如果使能 LIN 模式 (LINEN=1)，只要发生帧错误（例如，在“0”处检测到停止位，这种情况可能出现在任何断路帧中），接收器即会停止，直到断路检测电路接收到“1”（断路字不完整时）或接收到分隔符（检测到断路时）为止。

下图中显示了断路检测器状态机和断路标志的行为。

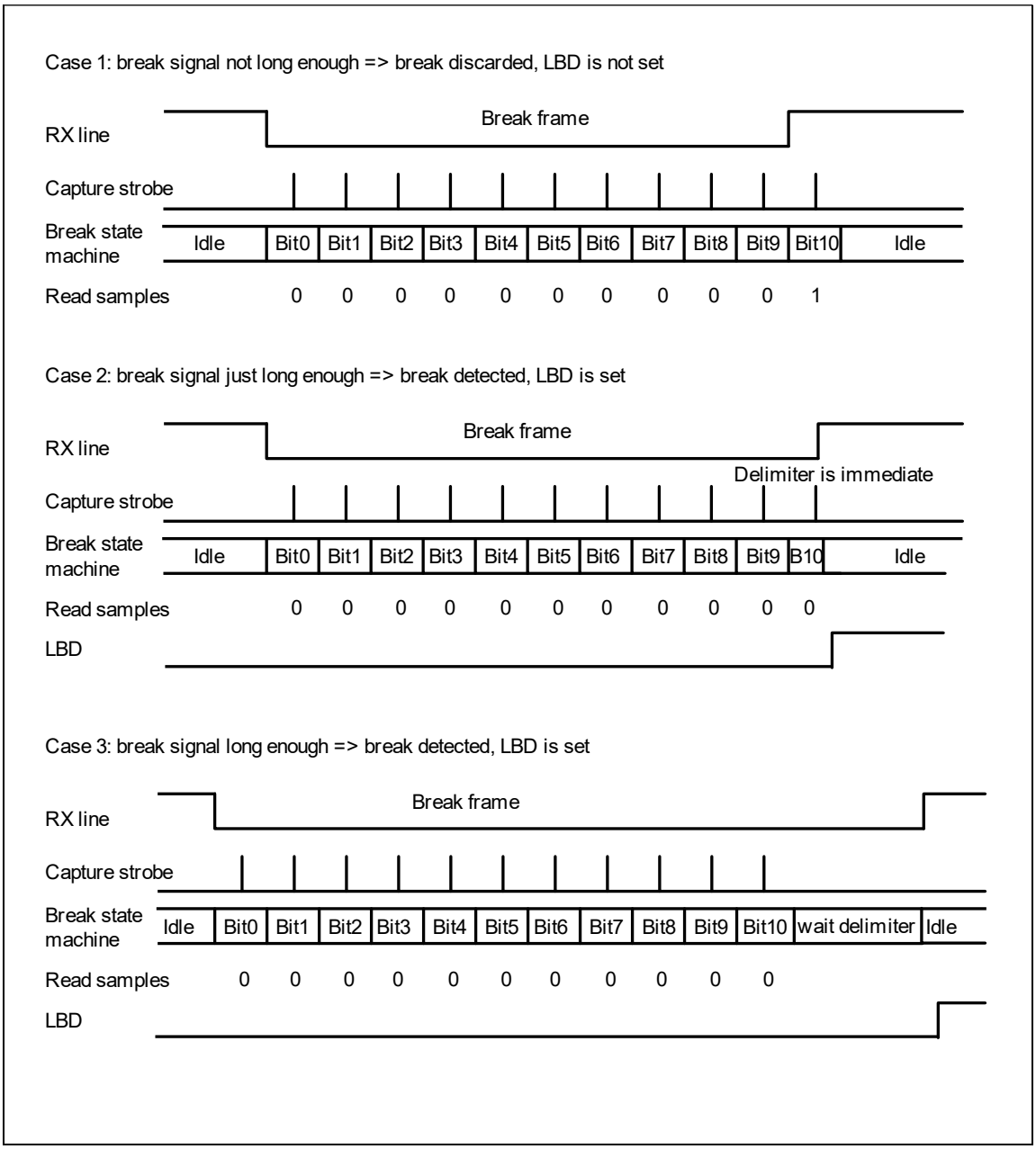



图 21-9 LIN 模式下的断路检测(11 位断路长度 - LBDL 位置 1)

下图列出了断路帧的示例。

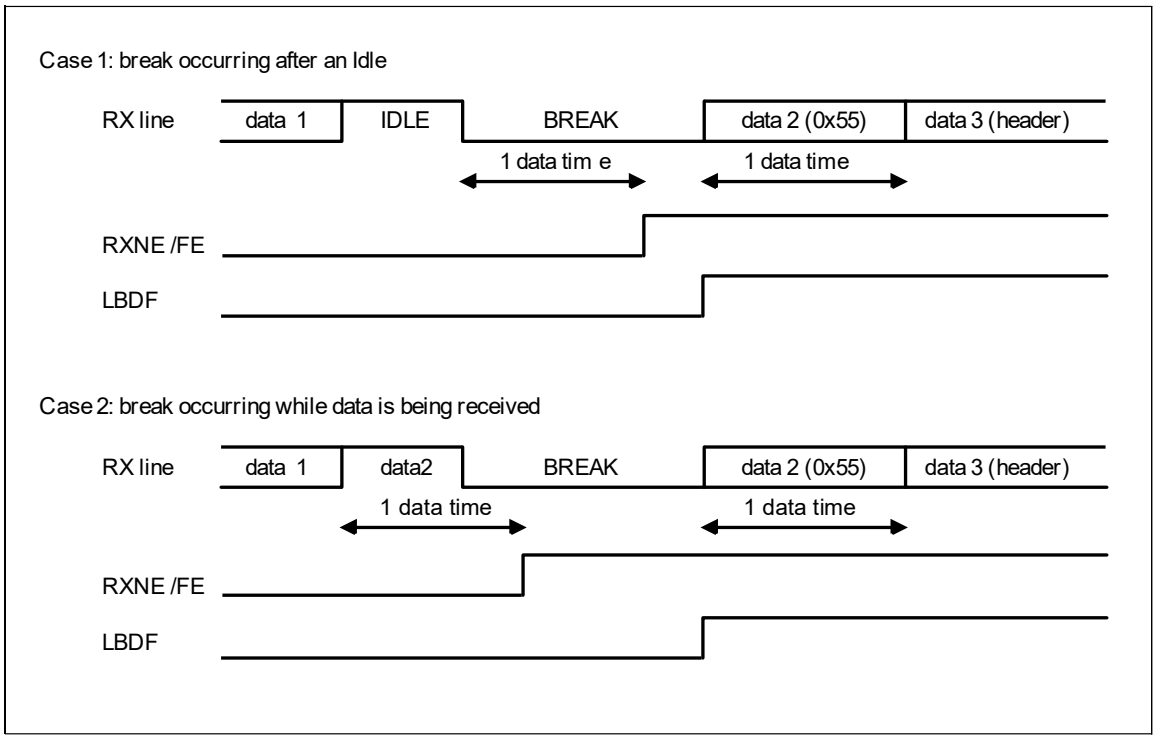



图 21-10 LIN 模式下的断路检测与帧错误检测

### 21.3.9. USART 同步模式

通过将 USART_CR2 寄存器中的 CLKEN 位写入 1 来选择同步模式。在同步模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN 位，
- USART_CR3 寄存器中的 SCEN、HDSEL 和 IREN 位。

通过 USART，用户可以在主模式下控制双向同步串行通信。SCLK 引脚是 USART 发送器时钟的输出。在起始位或停止位期间，不会向 SCLK 引脚发送时钟脉冲。在最后一个有效数据位（地址标记）期间，将会（也可能不会）生成时钟脉冲，这取决于 USART_CR2 寄存器中 LBCL 位的状态。通过 USART_CR2 寄存器中的 CPOL 位，用户可以选择时钟极性；通过 USART_CR2 寄存器中的 CPHA 位，用户可以选择外部时钟相位。

在空闲状态、报头模式和发送断路期间，外部 SCLK 时钟处于未激活状态。

USART 发送器在同步模式下的工作方式与异步模式下完全相同。但是由于 SCLK 与 TX 同步（根据 CPOL 和 CPHA），因此 TX 上的数据是同步的。

在此模式下，USART 接收器的工作方式与异步模式下不同。如果 RE=1，则数据在 SCLK 上采样（上升或下降沿，取决于 CPOL 和 CPHA），而不会进行任何过采样。此时必须确保建立时间和保持时间（取决于波特率： 1/16 位时间）符合要求。

*注意：SCLK 引脚可与 TX 引脚结合使用。因此，仅当使能发送器(TE=1) 且正在发送数据时(对数据寄存器 USART_DR 已被写入)，才会提供时钟。这意味着，没有发送数据的情况下无法接收同步数据。*

*当发送器和接收器(TE=RE=0) 都被禁止时，必须选择 LBCL、CPOL 和 CPHA 位，以确保时钟脉冲正常工作。当使能发送器或接收器时，不得对这些位进行更改。*

*建议按照相同指令将 TE 和 RE 位置 1，以尽量缩短接收器的建立时间和保持时间。*

*USART 只支持主模式：它不能接收或发送与输入时钟相关的数据(SCLK 始终为输出)。*

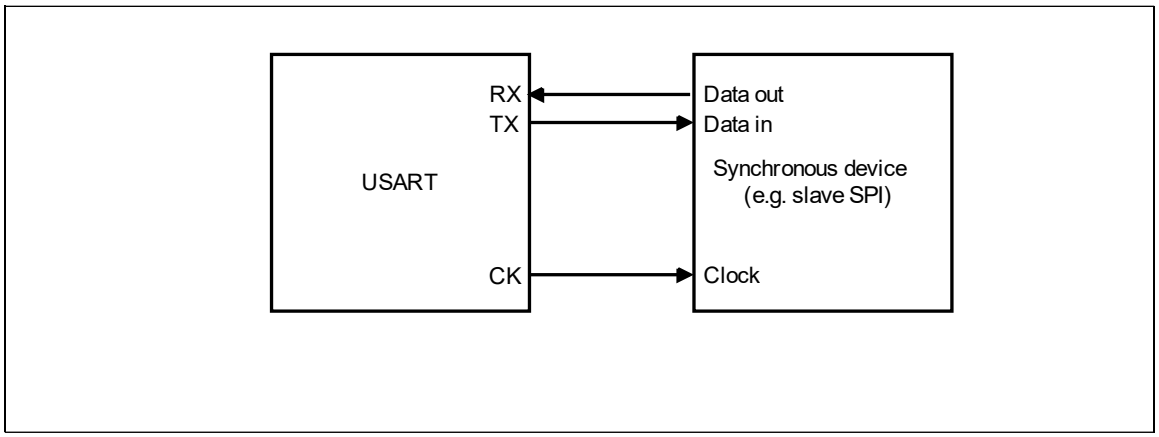



图 21-11 USART 同步发送示例

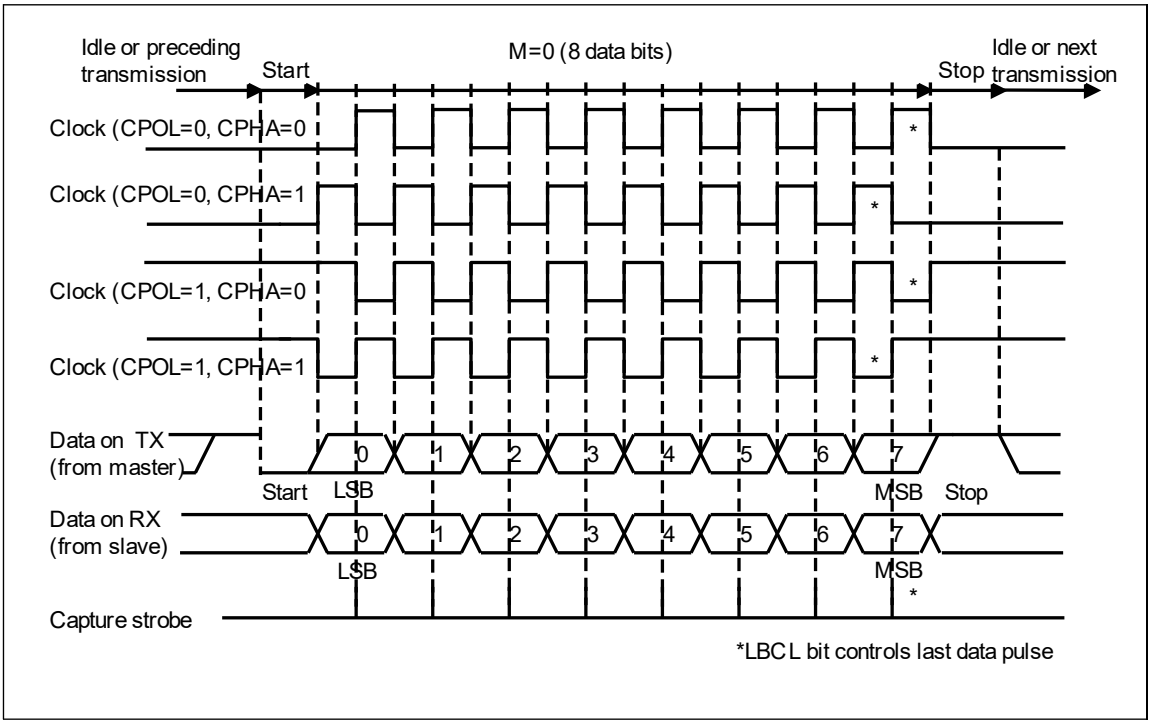



图 21-12 USART 数据时钟时序图(M=0)

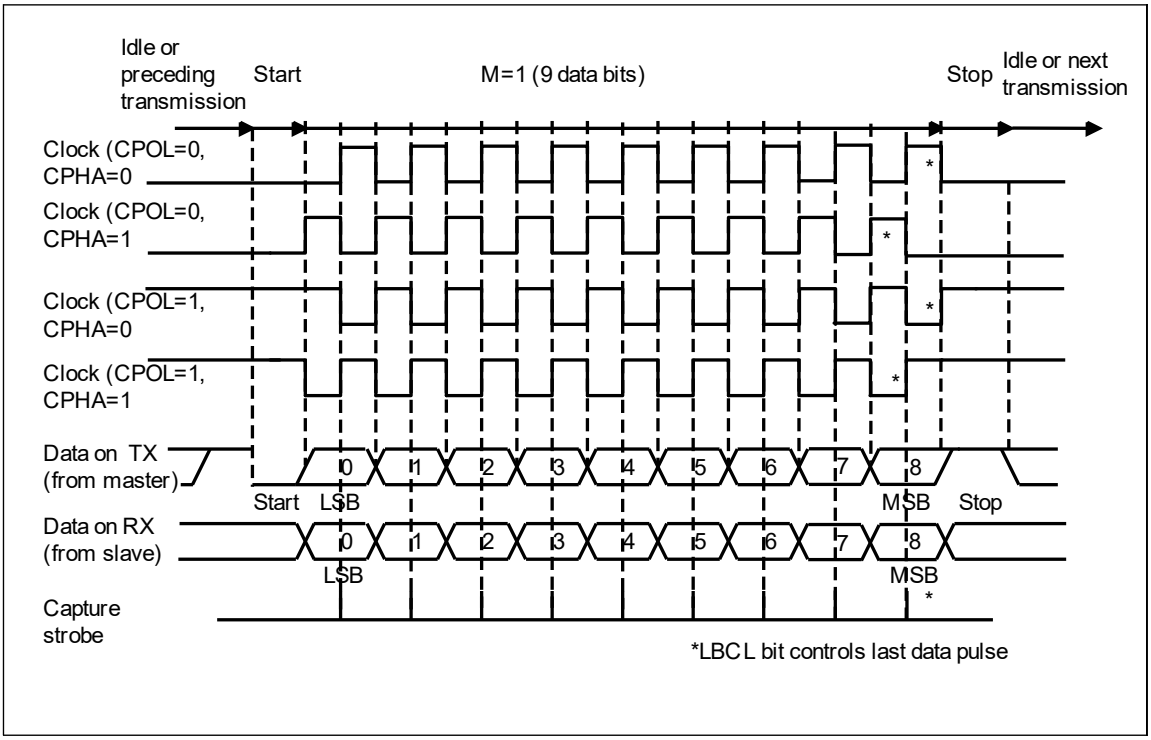



图 21-13 USART 数据时钟时序图(M=1)

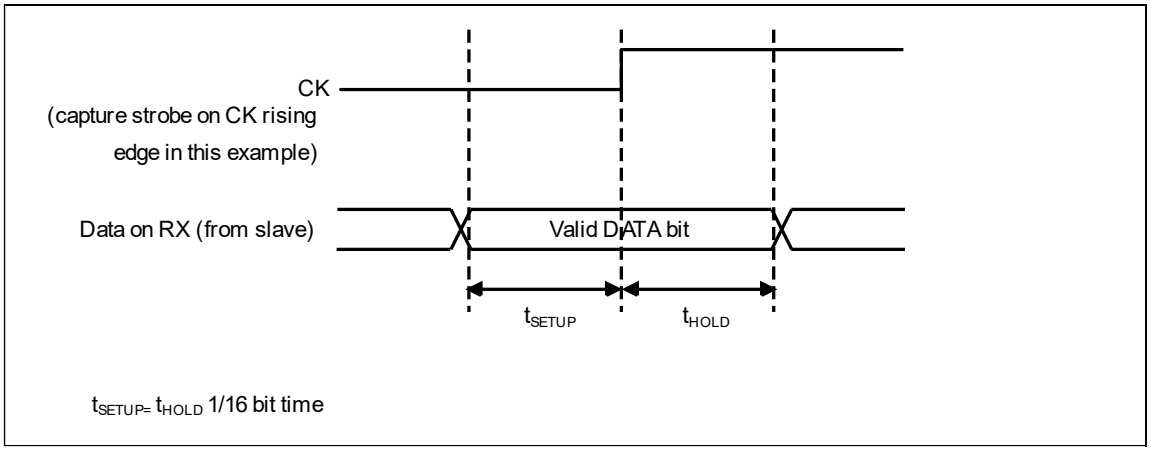



图 21-14 RX 数据建立/保持时间

*注意：智能卡模式下， SCLK 的功能有所不同。有关详细信息，请参见智能卡模式一章。*

### 21.3.10. 单线半双工通信

通过将 USART_CR3 寄存器中的 HDSEL 位置 1 来选择单线半双工模式。在此模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN 和 CLKEN 位，
- USART_CR3 寄存器中的 SCEN 和 IREN 位。

USART 可以配置为遵循单线半双工协议，其中 TX 和 RX 线路从内部相连接。使用控制位 “HALF DUPLEX SEL”（USART_CR3 寄存器中的 HDSEL 位），可以在半双工通信和全双工通信间进行选择。

一旦向 HDSEL 位写入 1：

- TX 和 RX 线路从内部相连接；
- 不能再使用 RX 引脚；
- 无数据传输时，TX 引脚始终处于释放状态。因此，它在空闲状态或接收过程中用作标准 I/O。这意味着，必须对 I/O 进行配置，以便在未受 USART 驱动时，使 TX 成为浮空输入（或高电平开漏输出）。

除此之外，通信与正常 USART 模式下的通信相似。此线路上的冲突必须由软件进行管理（例如，使用中央仲裁器）。尤其要注意，发送过程永远不会被硬件封锁，只要数据是在 TE 位置 1 的情况下写入，发送就会持续进行。

### 21.3.11. 智能卡

通过将 USART_CR3 寄存器中的 SCEN 位置 1 来选择智能卡模式。在智能卡模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN 位，
- USART_CR3 寄存器中的 HDSEL 和 IREN 位。

此外，可能需要将 CLKEN 位置 1，以便为智能卡提供时钟。

智能卡接口支持符合 ISO 7816-3 标准的异步协议智能卡。USART 应如下所示进行配置：

- 8 个位加奇偶校验：当 USART_CR1 寄存器中 M=1 且 PCE=1 时，
- 发送和接收时使用 1.5 个停止位：当 USART_CR2 寄存器中 STOP=11 时。

*注意：接收时也可以选择 0.5 个停止位，但为了避免在两种配置之间切换，建议发送和接收时均使用 1.5 个停止位。*

下图显示了有奇偶校验错误和无奇偶校验错误时数据线上情况的示例：

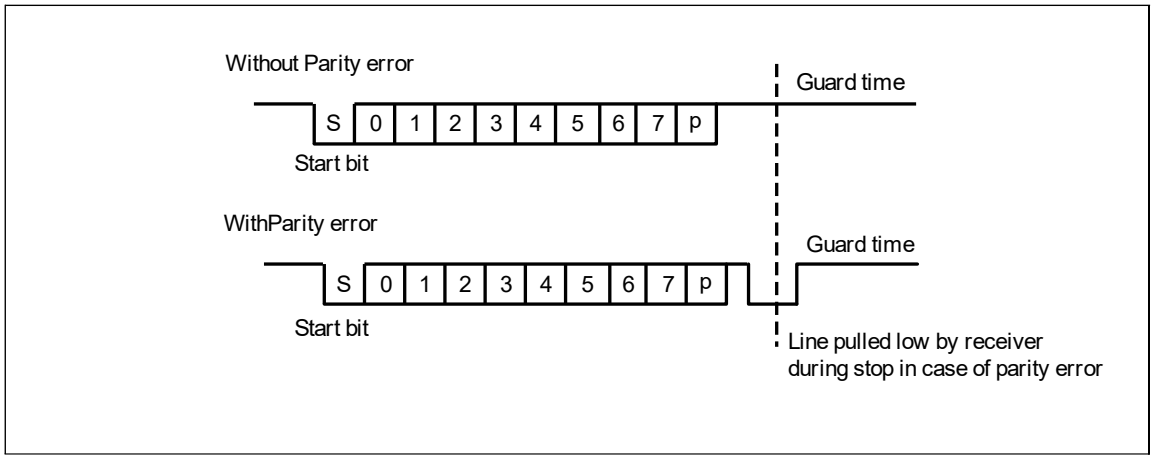



图 21-15 ISO 7816-3 异步协议

连接到智能卡时，USART 的 TX 输出会驱动一条双向线（它也由智能卡驱动）。必须将 TX 引脚配置为开漏引脚。

智能卡是一个单线半双工通信协议。

- 从发送移位寄存器发送数据会经过至少 1/2 个时钟周期的延迟。正常工作时，已满的发送移位寄存器会在下一个时钟边沿开始移位。在智能卡模式下，此发送过程还会进一步经过 1/2 波特时钟周期的延迟。
- 如果在接收一个使用 0.5 或 1.5 个停止位编程的帧期间检测到奇偶校验错误，则在完成接收帧后，发送线会被拉低一个时钟周期。这是为了向智能卡指出发送到 USART 的数据尚未正确接收。此 NACK 信号（将发送线拉低 1 个时钟周期）会导致发送器端（配置为 1.5 个停止位）出现帧错误。应用程序可根据协议重新发送数据。如果 NACK 控制位置 1，则接收器会向奇偶校验错误发送“NACK”信号；

否则不会发送 NACK 信号。

- 通过对保护时间寄存器进行编程，可以延迟 TC 标志的置位。正常工作时，当发送移位寄存器为空且没有新的发送请求出现时，会对 TC 标志进行置位。在智能卡模式下，空的发送移位寄存器会触发保护时间计数器，使其递增计数至保护时间寄存器中的值。在此期间，TC 标志被强制为低电平。当保护时间计数器达到设置值时，TC 置位为高电平。
- TC 标志的释放不受智能卡模式的影响。
- 如果在发送端检测到帧错误（由来自接收器的 NACK 信号引起），则发送端的接收块不会将 NACK 作为起始位进行检测。根据 ISO 协议，接收到的 NACK 信号的持续时间可以是 1 或 2 个时钟周期。
- 在接收端，如果检测到奇偶校验错误并发送了 NACK 信号，则接收端不会将 NACK 作为起始位进行检测。

*注意：中断字符在智能卡模式下无效。带有帧错误的 0x00 数据将被视为数据，而非中断。*

*当翻转 TE 位时，不会发送空闲帧。空闲帧（在其它配置中进行了定义）在 ISO 协议中未进行定义。*

下图详细介绍了 USART 如何对 NACK 信号采样。在本例中，USART 正在发送数据并配置了 1.5 个停止位。 USART 的接收部分已被使能，以检查数据的完整性和 NACK 信号。

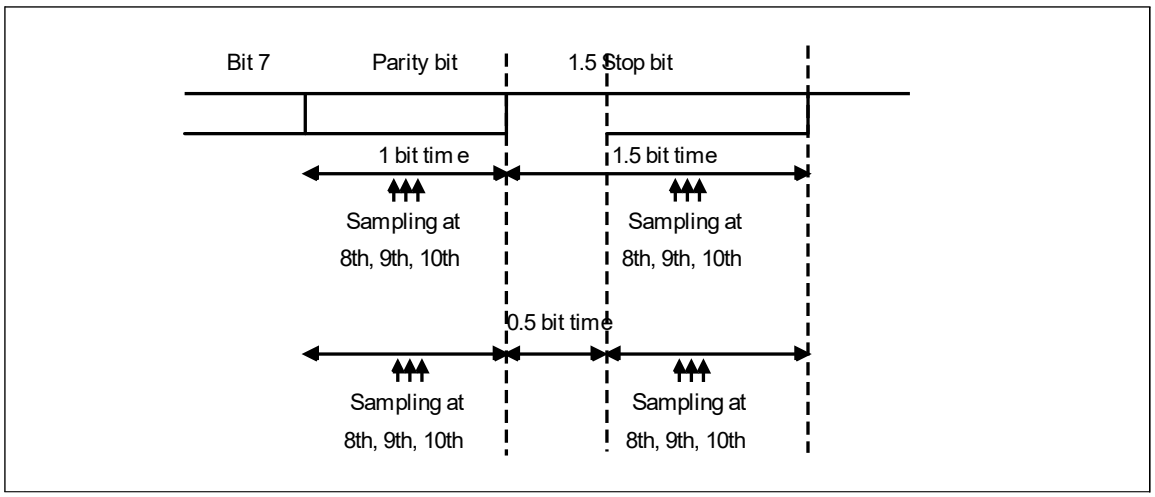



图 21-16 使用 1.5 个停止位检测奇偶校验错误

USART 可以通过 SCLK 输出为智能卡提供时钟。在智能卡模式下，SCLK 仅通过一个 5 位预分频器由内部外设输入时钟提供。分频比在预分频器寄存器 USART_GTPR 中进行配置。 SCLK 频率可在 $f_{CK}/2$ 到 $f_{CK}/62$ 之间进行编程，其中 $f_{CK}$ 为外设输入时钟。

### 21.3.12. IrDA SIR ENDEC 模块

通过将 USART_CR3 寄存器中的 IREN 位置 1 来选择 IrDA 模式。在 IrDA 模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN、STOP 和 CLKEN 位，
- USART_CR3 寄存器中的 SCEN 和 HDSEL 位。

IrDA SIR 物理层规定使用反相归零 (RZI) 调制方案，它以红外光脉冲表示逻辑 0。

SIR 发送编码器用于调制 USART 发出的非归零 (NRZ) 位流。输出脉冲流会发送到外部输出驱动器和红外线 LED 。USART 支持的 SIR ENDEC 比特率最高为 115.2Kbps。在正常模式下，所发送的脉冲宽度规定为一个位周期的 3/16。

SIR 接收解码器用于解调由红外探测器发出的归零位流，并将接收到的 NRZ 串行位流输出到 USART。在空闲状态下，解码器输入通常为高电平（标记状态）。发送编码器输出的极性与解码器输入相反。当解码器输入为低电平时，会检测到起始位。

- IrDA 是一个半双工通信协议。如果发送器忙（例如，USART 正在向 IrDA 编码器发送数据），则 IrDA 解码器会忽略 IrDA 接收线上的所有数据；如果接收器忙（例如，USART 正在接收来自 USART 的解码数据），则 IrDA 不会对 USART 发送到 IrDA 的 TX 上的数据进行编码。在接收数据时，应避免同时进行发送，因为这样做可能会破坏要发送的数据。
- “0”作为高电平脉冲发送，而“1”作为“0”发送。在正常模式下，脉冲宽度规定为所选位周期的 3/16。
- SIR 解码器用于将兼容 IrDA 的接收信号转换为 USART 的位流。
- SIR 接收逻辑将高电平状态视为逻辑“1”，将低电平脉冲视为逻辑“0”。
- 发送编码器输出的极性与解码器输入相反。SIR 输出在空闲时处于低电平状态。
- IrDA 规范要求脉冲容忍值要大于 1.41 us。可接受的脉冲宽度可通过寄存器设置。接收器端的干扰检测逻辑会滤除宽度小于 2 个 PSC 周期的脉冲（PSC 是在 IrDA 低功耗波特寄存器 USART_GTPR 中编程的预分频器值）。宽度小于 1 个 PSC 周期的脉冲都将被拒绝，但宽度大于 1 个而小于 2 个周期的脉冲可能被接受也可能被拒绝，而宽度大于 2 个周期的脉冲将被接受作为有效脉冲。当 PSC=0 时， IrDA 编码器/解码器不工作。
- 接收器能够与低功耗发送器进行通信。
- 在 IrDA 模式下，USART_CR2 寄存器中的 STOP 位必须配置为“1 个停止位”。

### IrDA 低功耗模式

**发送器：**

在低功耗模式下，脉冲宽度不再保持为位周期的 3/16 。此时的脉冲宽度为低功耗波特率的 3 倍，最小可为 1.42 MHz。通常此值是 1.8432 MHz (1.42 MHz < PSC< 2.12 MHz)。低功耗模式下的可编程分频器会对系统时钟进行分频，以达到此值。

**接收器：**

在低功耗模式下接收与在正常模式下接收类似。为进行干扰检测，USART 应丢弃持续时间短于 1/PSC 的脉冲。只有当持续时间大于 2 个 IrDA 低功耗波特时钟周期（USART_GTPR 寄存器中的 PSC 值）时，才是有效低电平。

*注意：宽度小于两个但大于一个 PSC 周期的脉冲可能被接受，也可能被拒绝。*

*接收器的建立时间应由软件进行管理。IrDA 物理层规范规定发送和接收之间至少要经过 10 ms 的延迟（IrDA 是一个半双工协议）。*

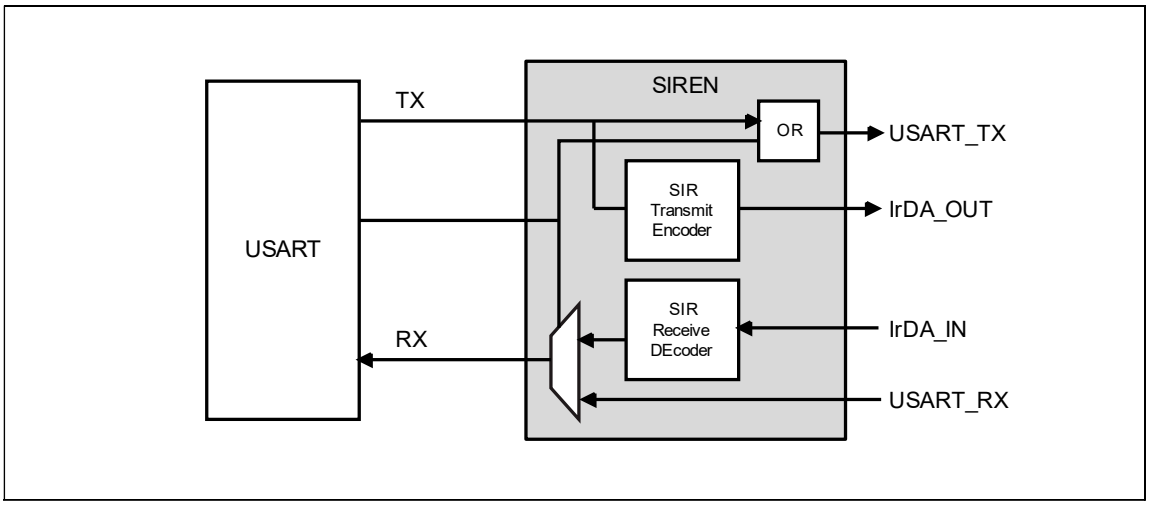



图 21-17 IrDA SIR ENDEC - 框图

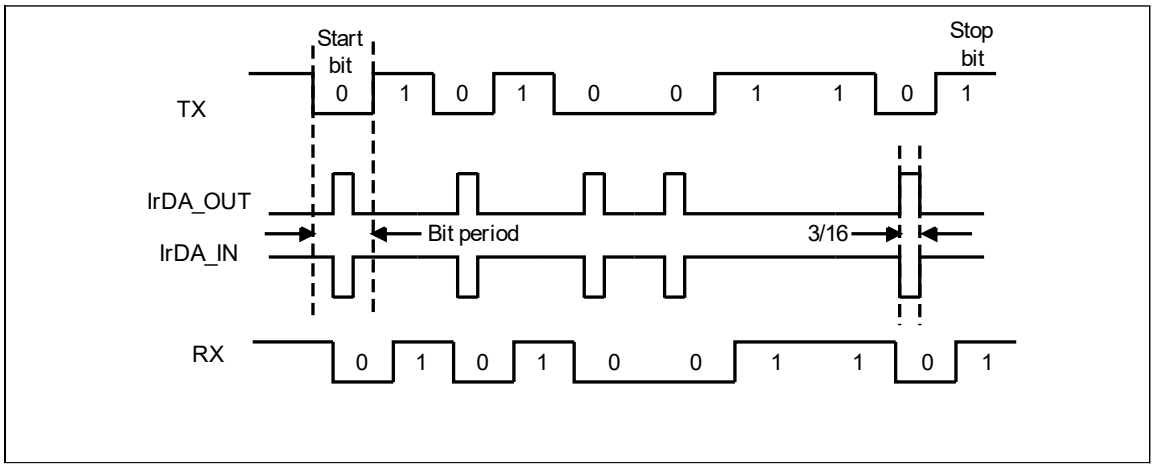



图 21-18 IrDA 数据调制 (3/16) - 正常模式

### 21.3.13. 使用 DMA 进行连续通信

USART 能够使用 DMA 进行连续通信。接收缓冲区和发送缓冲区的 DMA 请求是独立的。

**使用 DMA 进行发送**

将 USART_CR3 寄存器中的 DMAT 位置 1 可以使能 DMA 模式进行发送。当 TXE 位置 1 时，可将数据从 SRAM 区（通过 DMA 配置，参见 DMA 部分）加载到 USART_DR 寄存器。要映射一个 DMA 通道以进行 USART 发送，请按以下步骤操作（x 表示通道编号）：

- 在 DMA 控制寄存器中写入 USART_DR 寄存器地址，将其配置为传输的目标地址。每次发生 TXE 事件后，数据都会从存储器移动到此地址。
- 在 DMA 控制寄存器中写入存储器地址，将其配置为传输的源地址。每次发生 TXE 事件后，数据都会从这个存储区域加载到 USART_DR 寄存器中。
- 在 DMA 控制寄存器中配置要传输的总字节数。
- 在 DMA 寄存器中配置通道优先级
- 根据应用的需求，在完成一半或全部传输后产生 DMA 中断。
- 向 SR 寄存器中的 TC 位写入 0，将其清零。
- 在 DMA 寄存器中激活该通道。

当达到在 DMA 控制器中设置的数据传输量时，DMA 控制器会在 DMA 通道的中断向量上产生一个中断。

在发送模式下，DMA 对所有要发送的数据执行了写操作（DMA_ISR 寄存器中的 TCIF 标志置 1）后，可以对 TC 标志进行监视，以确保 USART 通信已完成。在禁止 USART 或进入停止模式前必须执行此步骤，以避免损坏最后一次发送。软件必须等待直到 TC=1 。TC 标志在所有数据发送期间都必须保持清零状态，然后在最后一帧发送结束后由硬件置 1。

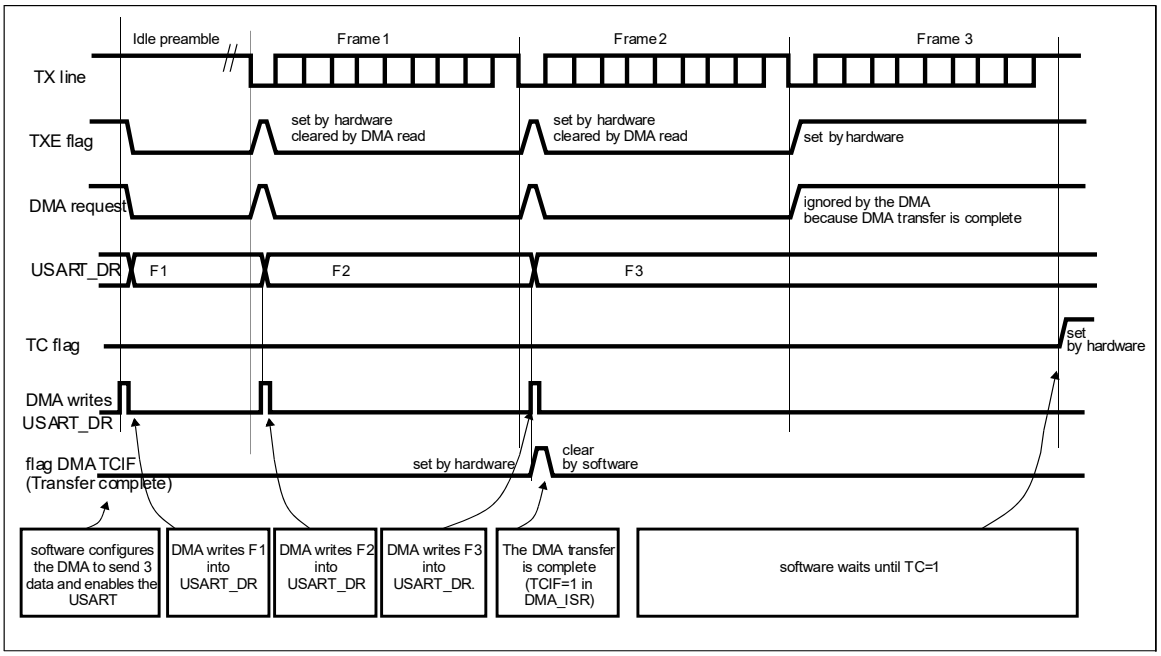



图 21-19 使用 DMA 进行发送

### 使用 DMA 进行接收

将 USART_CR3 寄存器中的 DMAR 位置 1 可以使能 DMA 模式进行接收。接收数据字节时，数据会从 USART_DR 寄存器加载到 SRAM 区域中。要映射一个 DMA 通道以进行 USART 接收，请按以下步骤操作：

- 在 DMA 控制寄存器中写入 USART_DR 寄存器地址，将其配置为传输的源地址。每次发生 RXNE 事件后，数据都会从此地址移动到存储器。
- 在 DMA 控制寄存器中写入存储器地址，将其配置为传输的目标地址。每次发生 RXNE 事件后，数据都会从 USART_DR 寄存器加载到此存储区。
- 在 DMA 控制寄存器中配置要传输的总字节数。
- 在 DMA 控制寄存器中配置通道优先级。
- 根据应用的需求，在完成一半或全部传输后产生中断。
- 在 DMA 控制寄存器中激活该通道。

当达到在 DMA 控制器中设置的数据传输量时，DMA 控制器会在 DMA 通道的中断向量上产生一个中断。在中断子程序中，USART_CR3 寄存器中的 DMAR 位应由软件清零。

*注意： 如果 DMA 用于接收，则不要使能 RXNEIE 位。*

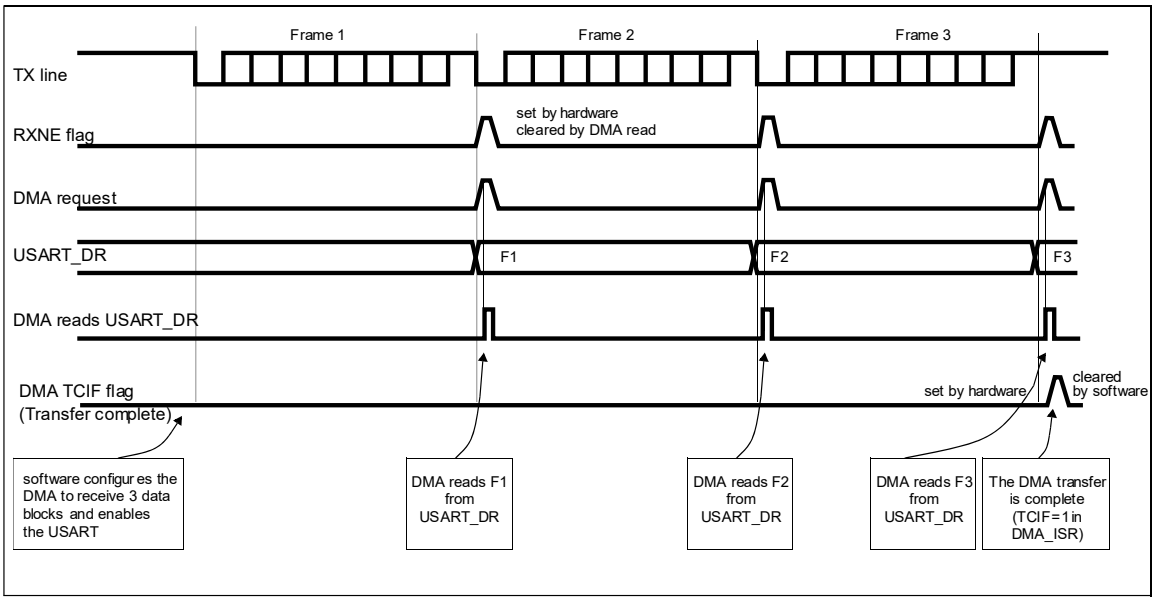



图 21-20 使用 DMA 进行接收

**多缓冲区通信中的错误标志和中断生成**

在多缓冲区通信中，如果事务中发生任何错误，都会在当前字节后放置错误标志。如果中断使能置 1，则会产生中断。在单字节接收过程中，与 RXNE 一同置位的帧错误、上溢错误和噪声标志具有单独的错误标志中断使能位（USART_CR3 寄存器中的 EIE 位）；如果该位置 1，则会因其中任何一个错误而在当前字节后产生中断。

## 21.3.14. 硬件流控制

使用 nCTS 输入和 nRTS 输出可以控制 2 个器件间的串行数据流。下图显示了在这种模式下如何连接 2 个器件：

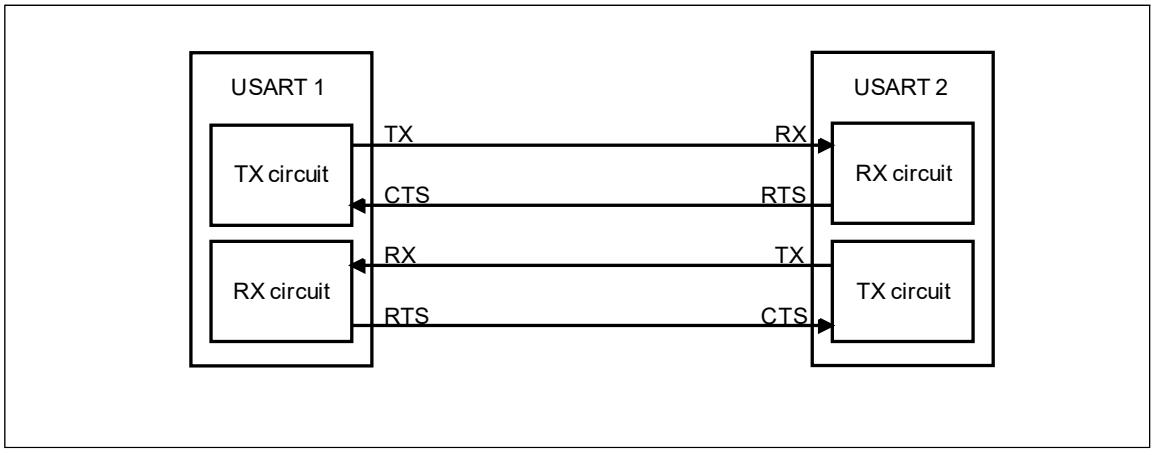



图 21-21 两个 USART 间的硬件流控制

分别向 USART_CR3 寄存器中的 RTSE 位和 CTSE 位写入 1，可以分别使能 RTS 和 CTS 流控制。

**RTS 流控制**

如果使能 RTS 流控制 (RTSE=1)，只要 USART 接收器准备好接收新数据，便会将 nRTS 变为有效（连接到低电平）。当接收寄存器已满时，会将 nRTS 变为无效，表明发送过程会在当前帧结束后停止。下图显示了在 RTS 流控制的情况下进行通信的示例。

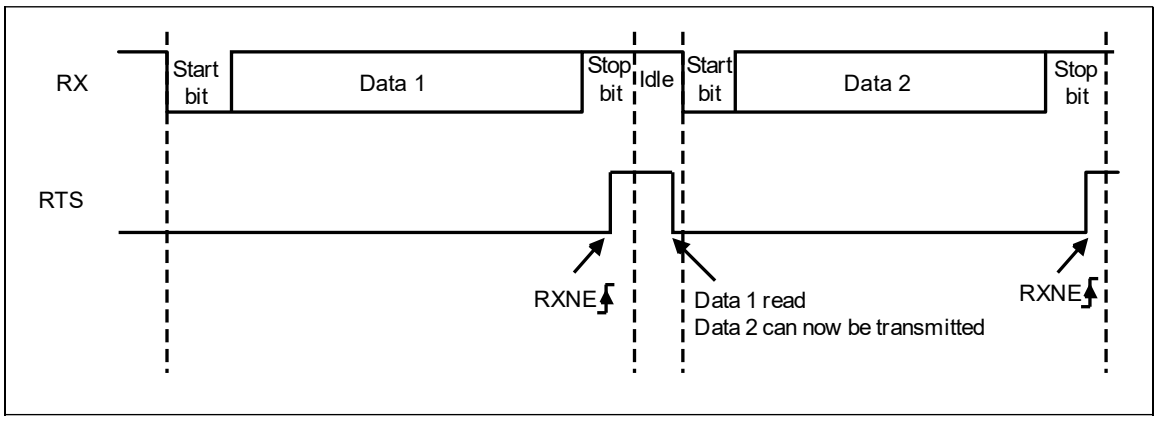



图 21-22 RTS 流控制

### CTS 流控制

如果使能 CTS 流控制 (CTSE=1)，则发送器会在发送下一帧前检查 nCTS。如果 nCTS 有效（连接到低电平），则会发送下一数据（假设数据已准备好发送，即 TXE=0）；否则不会进行发送。如果在发送过程中 nCTS 变为无效，则当前发送完成之后，发送器停止。

当 CTSE=1 时，只要 nCTS 发生变化，CTSIF 状态位便会由硬件自动置 1。这指示接收器是否已准备好进行通信。如果 USART_CR3 寄存器中的 CTSIE 位置 1，则会产生中断。

下图显示了在使能 CTS 流控制的情况下进行通信的示例。

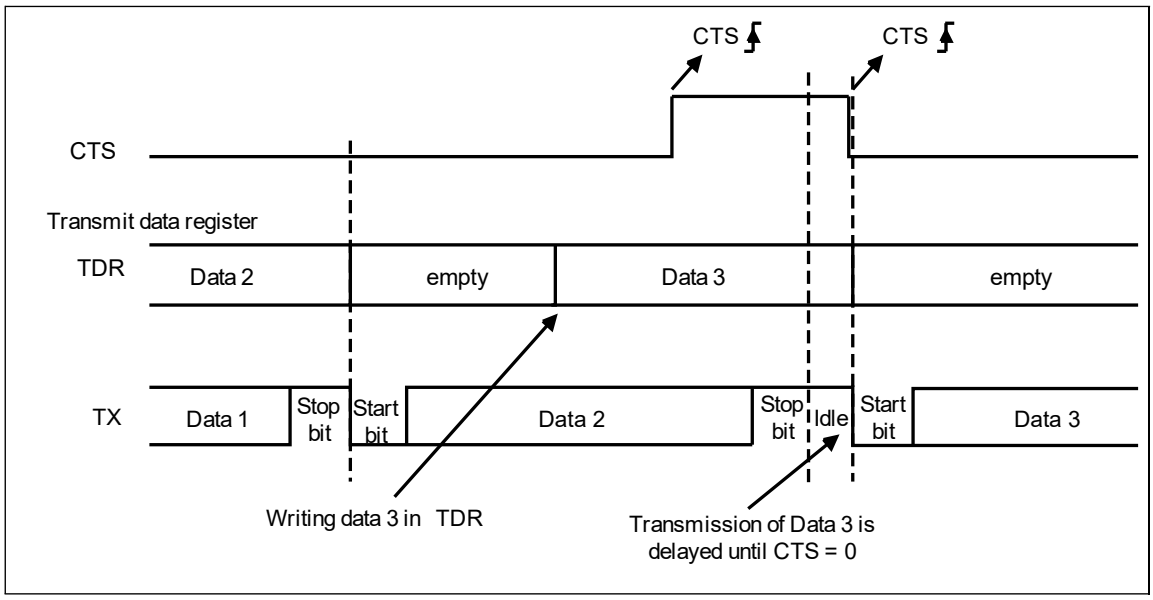



图 21-23 CTS 流控制

*注意： 停止帧的特殊行为： 使能 CTS 流后，发送器发送停止信号时将不检查 nCTS 输入状态。*

## 21.4. USART 中断

表 21-9 USART 中断请求

| 中断事件 | 事件标志 | 使能控制位 |
|---|---|---|
| 发送数据寄存器为空 | TXE | TXEIE |
| CTS 标志 | CTS | CTSIE |
| 发送完成 | TC | TCIE |

| 中断事件 | 事件标志 | 使能控制位 |
|---|---|---|
| 准备好读取接收到的数据 | RXNE | RXNEIE |
| 检测到上溢错误 | ORE | |
| 检测到空闲线路 | IDLE | IDLEIE |
| 奇偶校验错误 | PE | PEIE |
| 断路标志 | LBD | LBDIE |
| 多缓冲区通信中的噪声标志、上溢错误和帧错误 | NF 或 ORE 或 FE | EIE |
| 接收 FIFO 将满 | AFULL | AFIE |

USART 中断事件被连接到相同的中断向量（请参见下图）。

- 发送期间：发送完成、清除以发送或发送数据寄存器为空中断。
- 接收期间：空闲线路检测、上溢错误、接收数据寄存器不为空、奇偶校验错误、LIN 断路检测、噪声标志（仅限多缓冲区通信）和帧错误（仅限多缓冲区通信）。

如果相应的使能控制位置 1，则这些事件会生成中断。

## 21.5. USART 模式配置

表 21-10 USART 模式配置(1)

| USART 模式 | USART1 | USART2 | USART3 | UART4 | UART5 | USART6 | USART7 | USART8 |
|---|---|---|---|---|---|---|---|---|
| 异步模式 | X | X | X | X | X | X | X | X |
| 硬件流控制 | X | X | X | NA | NA | X | NA | NA |
| 多缓冲区通信(DMA) | X | X | X | X | X | X | X | X |
| 多处理器通信 | X | X | X | X | X | X | X | X |
| 同步 | X | X | X | NA | NA | X | NA | NA |
| 智能卡 | X | X | X | NA | NA | X | NA | NA |
| 半双工（单线模式） | X | X | X | X | X | X | X | X |
| IrDA | X | X | X | X | X | X | X | X |
| LIN | X | X | X | X | X | X | X | X |

1. X = 支持； NA = 不适用

## 21.6. USART 寄存器

外设寄存器可支持半字（16 位）或字（32 位）访问。

### 21.6.1. 状态寄存器 (USART_SR)

偏移地址： 0x00

复位值： 0x00C0 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | CTS | LBD | TXE | TC | RXNE | IDLE | ORE | NF | FE | PE |
| | | | | | | rc_w0 | rc_w0 | r | rc_w0 | rc_w0 | r | r | r | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:10 | 保留 | 必须保持复位值 |
| 9 | CTS | CTS 标志<br>如果 CTSE 位置 1，当 nCTS 输入变换时，此位由硬件置 1。通过软件将该位清零（通过向该位中写入 0）。如果 USART_CR3 寄存器中 CTSIE=1，则会生成中断。<br>0：nCTS 状态线上未发生变化<br>1：nCTS 状态线上发生变化<br>*注意： 该位不适用于 UART4、UART5、UART7、UART8。* |
| 8 | LBD | LIN 断路检测标志<br>检测到 LIN 断路时，该位由硬件置 1。通过软件将该位清零（通过向该位中写入 0）。如果 USART_CR2 寄存器中 LBDIE = 1，则会生成中断。<br>0：未检测到 LIN 断路<br>1：检测到 LIN 断路<br>*注意： 如果 LBDIE=1，则当 LBD=1 时生成中断* |
| 7 | TXE | 发送数据寄存器为空<br>当 TDR 寄存器的内容已传输到移位寄存器时，该位由硬件置 1。如果 USART_CR1 寄存器中 TXEIE 位 = 1，则会生成中断。通过对 USART_DR 寄存器执行写入操作将该位清零。<br>0：数据未传输到移位寄存器<br>1：数据传输到移位寄存器<br>*注意： 单缓冲区发送期间使用该位。* |
| 6 | TC | 发送完成<br>如果已完成对包含数据的帧的发送并且 TXE 置 1，则该位由硬件置 1。如果 USART_CR1 寄存器中 TCIE = 1，则会生成中断。该位由软件序列清零（读取 USART_SR 寄存器，然后写入 USART_DR 寄存器）。TC 位也可以通过向该位写入‘0’来清零。建议仅在多缓冲区通信时使用此清零序列。<br>0：传送未完成<br>1：传送已完成 |
| 5 | RXNE | 读取数据寄存器不为空<br>当 RDR 移位寄存器的内容已传输到 USART_DR 寄存器时，该位由硬件置 1。如果 USART_CR1 寄存器中 RXNEIE = 1，则会生成中断。通过对 USART_DR 寄存器执行读入操作将该位清零。RXNE 标志也可以通过向该位写入零来清零。建议仅在多缓冲区通信时使用此清零序列。<br>0：未接收到数据<br>1：已准备好读取接收到的数据 |
| 4 | IDLE | 检测到空闲线路<br>检测到空闲线路时，该位由硬件置 1。如果 USART_CR1 寄存器中 IDLEIE = 1，则会生成中断。该位由软件序列清零（读入 USART_SR 寄存器，然后读入 USART_DR 寄存器）。<br>0：未检测到空闲线路<br>1：检测到空闲线路<br>*注意： 直到 RXNE 位本身已置 1 时（即，当出现新的空闲线路时）IDLE 位才会被再次置 1。* |
| 3 | ORE | 上溢错误<br>在 RXNE = 1 的情况下，当移位寄存器中当前正在接收的字准备好传输到 RDR 寄存器时，该位由硬件置 1。如果 USART_CR1 寄存器中 RXNEIE = 1，则会生成中断。该位由软件序列清零（读入 USART_SR 寄存器，然后读入 USART_DR 寄存器）。 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 0：无上溢错误<br>1：检测到上溢错误<br>*注意：当该位置 1 时，RDR 寄存器的内容不会丢失，但移位寄存器会被覆盖。如果 EIE 位置 1，则在进行多缓冲区通信时会对 ORE 标志生成一个中断。* |
| 2 | NF | 检测到噪声标志<br>当在接收的帧上检测到噪声时，该位由硬件置 1。该位由软件序列清零（读入 USART_SR 寄存器，然后读入 USART_DR 寄存器）。<br>0：未检测到噪声<br>1：检测到噪声<br>*注意： 如果 EIE 位置 1，则在进行多缓冲区通信时，该位不会生成中断，因为该位出现的时间与本身生成中断的 RXNE 位因 NF 标志而生成的时间相同。*<br>*注意： 当线路无噪声时，可以通过将 ONEBIT 位编程为 1 提高 USART 对偏差的容差来禁止 NF 标志* |
| 1 | FE | 帧错误<br>当检测到去同步化、过度的噪声或中断字符时，该位由硬件置 1。该位由软件序列清零（读入 USART_SR 寄存器，然后读入 USART_DR 寄存器）。<br>0：未检测到帧错误<br>1：检测到帧错误或中断字符<br>*注意： 该位不会生成中断，因为该位出现的时间与本身生成中断的 RXNE 位出现的时间相同。如果当前正在传输的字同时导致帧错误和上溢错误，则会传输该字，且仅有 ORE 位被置 1。*<br>*如果 EIE 位置 1，则在进行多缓冲区通信时会对 FE 标志生成一个中断。* |
| 0 | PE | 奇偶校验错误<br>当在接收器模式下发生奇偶校验错误时，该位由硬件置 1。该位由软件序列清零（读取状态寄存器，然后对 USART_DR 数据寄存器执行读或写访问）。将 PE 位清零前软件必须等待 RXNE 标志被置 1。<br>如果 USART_CR1 寄存器中 PEIE = 1，则会生成中断。<br>0：无奇偶校验错误<br>1：奇偶校验错误 |

### 21.6.2. 数据寄存器 (USART_DR)

偏移地址： 0x04

复位值： 0xXXXX XXXX

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:9 | 保留 | 必须保持复位值 |
| 8:0 | DR[8:0] | 数据值包含接收到数据字符或已发送的数据字符，具体取决于所执行的操作是“读取”操作还是“写入”操作。<br>因为数据寄存器包含两个寄存器，一个用于发送 (TDR)，一个用于接收 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | (RDR)，因此它具有双重功能（读和写）。<br>在使能奇偶校验位的情况下（USART_CR1 寄存器中的 PCE 位被置 1）进行发送时，由于 MSB 的写入值（位 7 或位 8，具体取决于数据长度）会被奇偶校验位所取代，因此该值不起任何作用。<br>在使能奇偶校验位的情况下进行接收时，从 MSB 位中读取的值为接收到的奇偶校验位。 |

### 21.6.3. 波特率寄存器 (USART_BRR)

注意：如果 TE 或 RE 位分别被禁止，则波特计数器会停止计数。

偏移地址： 0x08

复位值： 0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:4 | DIV_Mantissa [11:0] | USARTDIV 的尾数<br>这 12 个位用于定义 USART 除数 (USARTDIV) 的尾数 |
| 3:0 | DIV_Fraction [3:0] | USARTDIV 的小数<br>这 4 个位用于定义 USART 除数 (USARTDIV) 的小数。当 OVER8 = 1 时，不考虑 DIV_Fraction3 位，且必须将该位保持清零。 |

### 21.6.4. 控制寄存器 1(USART_CR1)

偏移地址： 0x0C

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OVER8 | Reserved | UE | M | WAKE | PCE | PS | PEIE | TXEIE | TCIE | RXNEIE | IDLEIE | TE | RE | RWU | SBK |
| rw | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15 | OVER8 | 过采样模式<br>0：16 倍过采样<br>1：8 倍过采样<br>注意： 8 倍过采样在智能卡、IrDA 和 LIN 模式下不可用：当 SCEN=1、IREN=1 或 LINEN=1 时，OVER8 由硬件强制清零。 |
| 14 | 保留 | 必须保持复位值 |
| 13 | UE | USART 使能 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 该位清零后，USART 预分频器和输出将停止，并会结束当前字节传输以降低功耗。此位由软件置 1 和清零。<br>0：禁止 USART 预分频器和输出<br>1：使能 USART |
| 12 | M | 字长<br>该位决定了字长。该位由软件置 1 或清零。<br>0：1 起始位，8 数据位，n 停止位<br>1：1 起始位，9 数据位，n 停止位<br>注意： 在数据传输（发送和接收）期间不得更改 M 位 |
| 11 | WAKE | 唤醒方法<br>该位决定了 USART 唤醒方法，该位由软件置 1 或清零。<br>0：空闲线路<br>1：地址标记 |
| 10 | PCE | 奇偶校验控制使能<br>该位选择硬件奇偶校验控制（生成和检测）。使能奇偶校验控制时，计算出的奇偶校验位被插入到 MSB 位置（如果 M=1，则为第 9 位；如果 M=0，则为第 8 位），并对接收到的数据检查奇偶校验位。此位由软件置 1 和清零。一旦该位置 1，PCE 在当前字节的后面处于活动状态（在接收和发送时）。<br>0：禁止奇偶校验控制<br>1：使能奇偶校验控制 |
| 9 | PS | 奇偶校验选择<br>该位用于在使能奇偶校验生成/检测（PCE 位置 1）时选择奇校验或偶校验。该位由软件置 1 和清零。将在当前字节的后面选择奇偶校验。<br>0：偶校验<br>1：奇校验 |
| 8 | PEIE | PE 中断使能<br>此位由软件置 1 和清零。<br>0：禁止中断<br>1：当 USART_SR 寄存器中 PE=1 时，生成 USART 中断 |
| 7 | TXEIE | TXE 中断使能<br>此位由软件置 1 和清零。<br>0：禁止中断<br>1：当 USART_SR 寄存器中 TXE=1 时，生成 USART 中断。 |
| 6 | TCIE | 传送完成中断使能<br>此位由软件置 1 和清零。<br>0：禁止中断<br>1：当 USART_SR 寄存器中 TC=1 时，生成 USART 中断 |
| 5 | RXNEIE | RXNE 中断使能<br>此位由软件置 1 和清零。<br>0：禁止中断<br>1：当 USART_SR 寄存器中 ORE=1 或 RXNE=1 时，生成 USART 中断 |
| 4 | IDLEIE | IDLE 中断使能<br>此位由软件置 1 和清零。<br>0：禁止中断<br>1：当 USART_SR 寄存器中 IDLE=1 时，生成 USART 中断 |
| 3 | TE | 发送器使能<br>该位使能发送器。该位由软件置 1 和清零。 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 0：禁止发送器<br>1：使能发送器<br>注意：<br>1)：除了在智能卡模式下以外，传送期间 TE 位上的“0”脉冲（“0”后紧跟的是“1”）会在当前字的后面发送一个报头（空闲线路）。<br>2)：当 TE 置 1 时，在发送开始前存在 1 位的时间延迟。 |
| 2 | RE | 接收器使能<br>该位使能接收器。该位由软件置 1 和清零。<br>0：禁止接收器<br>1：使能接收器并开始搜索起始位 |
| 1 | RWU | 接收器唤醒<br>该位决定 USART 是否处于静音模式。该位由软件置 1 和清零，并可在识别出唤醒序列时由硬件清零。<br>0：接收器处于活动模式<br>1：接收器处于静音模式<br>注意：<br>1)：选择静音模式前（通过将 RWU 位置 1），USART 必须首先接收一个数据字节，否则当由空闲线路检测到唤醒时，它无法于静音模式下正常工作。<br>2)：在地址标记检测唤醒配置（WAKE 位 = 1）中，RXNE 位置 1 时，RWU 位不能由软件进行修改。 |
| 0 | SBK | 发送断路<br>该位用于发送断路字符。该位可由软件置 1 和清零。该位应由软件置 1，并在断路停止位期间由硬件重置。<br>0：不发送断路字符<br>1：将发送断路字符 |

### 21.6.5. 控制寄存器 2 (USART_CR2)

偏移地址： 0x10

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | LINEN | STOP[1:0] | | CLKEN | CPOL | CPHA | LBCL | Res. | LBDIE | LBDL | Res. | ADD[3:0] | | | |
| | rw | rw | rw | rw | rw | rw | rw | | rw | rw | | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:15 | 保留 | 必须保持复位值 |
| 14 | LINEN | LIN 模式使能<br>此位由软件置 1 和清零。<br>0：禁止 LIN 模式<br>1：使能 LIN 模式<br>LIN 模式可以使用 USART_CR1 寄存器中的 SBK 位发送 LIN 同步断路（13 个低位），并可检测 LIN 同步断路。 |
| 13:12 | STOP | 停止位<br>这些位用于编程停止位。 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 00：1 个停止位<br>01：0.5 个停止位<br>10：2 个停止位<br>11：1.5 个停止位<br>*注意：0.5 个停止位和 1.5 个停止位不适用于 UART4 、UART5、UART7、UART8。* |
| 11 | CLKEN | 时钟使能<br>该位允许用户使能 SCLK 引脚。<br>0：禁止 SCLK 引脚<br>1：使能 SCLK 引脚<br>该位不适用于 UART4 、UART5、UART7、UART8。 |
| 10 | CPOL | 时钟极性<br>该位允许用户在同步模式下选择 SCLK 引脚上时钟输出的极性。它与 CPHA 位结合使用可获得所需的时钟/数据关系<br>0：空闲时 SCLK 引脚为低电平。<br>1：空闲时 SCLK 引脚为高电平。<br>该位不适用于 UART4 、UART5、UART7、UART8。 |
| 9 | CPHA | 时钟相位<br>该位允许用户在同步模式下选择 SCLK 引脚上时钟输出的相位。它与 CPOL 位结合使用可获得所需的时钟/数据关系<br>0：在时钟第一个变化沿捕获数据<br>1：在时钟第二个变化沿捕获数据<br>*注意： 该位不适用于UART4 、UART5、UART7、UART8。* |
| 8 | LBCL | 最后一个位时钟脉冲<br>该位允许用户在同步模式下选择与发送的最后一个数据位 (MSB) 关联的时钟脉冲是否必须在 SCLK 引脚上输出。<br>0：最后一个数据位的时钟脉冲不在 SCLK 引脚上输出<br>1：最后一个数据位的时钟脉冲在 SCLK 引脚上输出<br>*注意：*<br>*1)：最后一位为发送的第 8 或第 9 个数据位，具体取决于 USART_CR1 寄存器中 M 位所选择的 8 位或 9 位格式。*<br>*2)：该位不适用于 UART4 、UART5、UART7、UART8。* |
| 7 | 保留 | 必须保持复位值 |
| 6 | LBDIE | LIN 断路检测中断使能<br>断路中断屏蔽（使用断路分隔符进行断路检测）<br>0：禁止中断<br>1：当 USART_SR 寄存器中 LBD = 1 时，生成中断 |
| 5 | LBDL | lin 断路检测长度<br>该位用于选择 11 位断路检测或 10 位断路检测。<br>0：10 位断路检测<br>1：11 位断路检测 |
| 4 | 保留 | 必须保持复位值 |
| 3:0 | ADD[3:0] | USART 节点的地址<br>该位域用于指定 USART 节点的地址。<br>将在多处理器通信时于静音模式下使用该位域，以通过地址标记检测进行唤醒。<br>*注意：*<br>*使能发送器时不应对这 3 个位（CPOL、CPHA、LBCL）进行写操作。* |

### 21.6.6. 控制寄存器 3 (USART_CR3)

偏移地址： 0x14

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | ONEBIT | CTSIE | CTSE | RTSE | DMAT | DMAR | SCEN | NACK | HDSEL | IRLP | IREN | EIE |
| | | | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:12 | 保留 | 必须保持复位值 |
| 11 | ONEBIT | 一个采样位方法使能<br>该位允许用户选择采样方法。选择一个采样位方法后，将禁止噪声检测标志 (NF)。<br>0：三个采样位方法<br>1：一个采样位方法 |
| 10 | CTSIE | CTS 中断使能<br>0：禁止中断<br>1：当 USART_SR 寄存器中 CTS = 1 时，生成中断<br>*注意： 该位不适用于 UART4、UART5、UART7、UART8* |
| 9 | CTSE | CTS 使能<br>0：禁止 CTS 硬件流控制<br>1：使能 CTS 模式，仅当 nCTS 输入有效 （连接到 0）时才发送数据。如果在发送数据时使 nCTS 输入无效，会在停止之前完成发送。如果使 nCTS 有效时数据已写入数据寄存器，则将延迟发送，直到 nCTS 有效。<br>*注意： 该位不适用于 UART4、UART5、UART7、UART8* |
| 8 | RTSE | RTS 使能<br>0：禁止 RTS 硬件流控制<br>1：使能 RTS 中断，仅当接收缓冲区中有空间时才会请求数据。发送完当前字符后应停止发送数据。 可以接收数据时使 nRTS 输出有效 （连接到 0）。<br>*注意： 该位不适用于 UART4、UART5、UART7、UART8* |
| 7 | DMAT | DMA 使能发送器<br>该位由软件置 1/ 复位。<br>1：针对发送使能 DMA 模式。<br>0：针对发送禁止 DMA 模式。 |
| 6 | DMAR | DMA 使能接收器<br>该位由软件置 1/ 复位。<br>1：针对接收使能 DMA 模式<br>0：针对接收禁止 DMA 模式 |
| 5 | SCEN | 智能卡模式使能<br>该位用于使能智能卡模式。<br>0：禁止智能卡模式<br>1：使能智能卡模式<br>*注意： 该位不适用于 UART4、UART5、UART7、UART8* |
| 4 | NACK | 智能卡 NACK 使能 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 0： 出现奇偶校验错误时禁止 NACK 发送<br>1：出现奇偶校验错误时使能 NACK 发送<br>*注意： 该位不适用于 UART4、UART5、UART7、UART8* |
| 3 | HDSEL | 半双工选择<br>选择单线半双工模式<br>0：未选择半双工模式<br>1：选择半双工模式 |
| 2 | IRLP | IrDA 低功耗<br>该位用于选择正常模式和低功耗 IrDA 模式<br>0：正常模式<br>1：低功耗模式 |
| 1 | IREN | IrDA 模式使能<br>此位由软件置 1 和清零。<br>0：禁止 IrDA<br>1：使能 IrDA |
| 0 | EIE | 错误中断使能<br>对于多缓冲区通信（USART_CR3 寄存器中 DMAR = 1），如果发生帧错误、上溢错误或出现噪声标志（USART_SR 寄存器中 FE = 1 或 ORE = 1 或 NF = 1），则需要使用错误中断使能位来使能中断生成。<br>0：禁止中断<br>1：当 USART_CR3 寄存器中的 DMAR = 1 并且 USART_SR 寄存器中的 FE = 1 或 ORE = 1 或 NF = 1 时，将生成中断。 |

### 21.6.7. 保护时间和预分频器寄存器 (USART_GTPR)

偏移地址： 0x18

复位值： 0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:8 | GT[7:0] | 保护时间值<br>该位域提供保护时间值 （以波特时钟数为单位）。<br>该位用于智能卡模式。经过此保护时间后，发送完成标志置 1。<br>*注意：该位不适用于 UART4、UART5、UART7、UART8* |
| 7:0 | PSC[7:0] | 预分频器值<br>在 IrDA 低功耗模式下：<br>PSC[7:0] = IrDA 低功耗波特率<br>用于编程预分频器，进行系统时钟分频以获得低功耗频率：<br>使用寄存器中给出的值（8 个有效位）对源时钟进行分频：<br>00000000：保留 - 不编程此值<br>源时钟 1 分频<br>00000010：源时钟 2 分频 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | ...<br>在正常 IrDA 模式下：PSC 必须设置为 00000001。<br>在智能卡模式下：<br>PSC[4:0]：预分频器值<br>用于编程预分频器，进行系统时钟分频以提供智能卡时钟。<br>将寄存器中给出的值(5 个有效位)乘以 2 得出源时钟频率的分频系数：<br>00000：保留 - 不编程此值<br>00001：源时钟 2 分频<br>00010：源时钟 4 分频<br>00011：源时钟 6 分频<br>*注意：*<br>*1)：如果使用智能卡模式，则位 [7:5] 不起作用。*<br>*2)：该位不适用于 UART4、UART5、UART7、UART8* |

### 21.6.8. 接收 FIFO 控制寄存器 (USART_FCR)

偏移地址： 0x1C

复位值： 0x0000 0002

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:10 | 保留 | 必须保持复位值 |
| 9:6 | FIFO_LEVEL [9:6] | 接收 FIFO 缓存数量_<br>当前接收 FIFO 中数据的个数 |
| 5 | AFULL | FIFO 将满标志<br>当 FIFO_LEVEL>=7 时，此位由硬件置 1。当 FIFO_LEVEL<7 时，此位由硬件自动清零。如果 AFIE=1，则会生成中断。<br>0：接收 FIFO 数据未满<br>1：接收 FIFO 数据将满 |
| 4 | AFIE | FIFO 将满中断使能<br>0：不使能 FIFO 中断<br>1：使能 FIFO 中断<br>*注意： 是否使能中断不影响AFULL 标志位状态。* |
| 3:2 | 保留 | 必须保持复位值 |
| 1 | FEN | 该位清零后， FIFO 功能将被静止，不会进行数据缓存及相应中断触发<br>0：使能<br>1：禁用 |
| 0 | 保留 | 必须保持复位值 |

# 22. 串行外设接口 (SPI)

## 22.1. SPI 简介

SPI 接口提供两个主要功能，支持 SPI 协议或 I²S 音频协议。默认情况下，选择的是 SPI 功能。可通过软件将接口从 SPI 切换到 I²S。

串行外设接口 (SPI) 可与外部器件进行半双工/全双工的同步串行通信。该接口可配置为主模式，在这种情况下，它可为外部从器件提供通信时钟 (SCK)。该接口还能够在多主模式配置下工作。

它可用于多种用途，包括基于双线的单工同步传输，其中一条可作为双向数据线，或使用 CRC 校验实现可靠通信。

I²S 也是同步串行通信接口。它可满足四种不同音频标准的要求，包括 I²S Philips 标准、MSB 和 LSB 对齐标准， 以及 PCM 标准。它可在全双工模式（使用 4 引脚）或半双工模式（使用 3 个引脚）下作为从器件或主器件工作。当 I2S 配置为通信主模式时，该接口可以向外部从器件提供主时钟。

## 22.2. SPI 和 I2S 主要特性

### 22.2.1. SPI 特性

- 基于三条线的全双工同步传输；
- 基于双线的单工同步传输，其中一条可作为双向数据线；
- 8 位或 16 位传输帧格式选择；
- 主模式或从模式操作；
- 多主模式功能；
- 8 个主模式波特率预分频器（最大值为 $f_{PCLK}/2$）；
- 从模式频率（最大值为 $f_{PCLK}/2$）；
- 对于主模式和从模式都可实现更快的通信；
- 对于主模式和从模式都可通过硬件或软件进行 NSS 管理：动态切换主/从操作；
- 可编程的时钟极性和相位；
- 可编程的数据顺序，最先移位 MSB 或 LSB；
- 可触发中断的专用发送和接收标志；
- SPI 总线忙状态标志；
- SPI TI 模式；
- 用于确保可靠通信的硬件 CRC 功能：
  - 在发送模式下可将 CRC 值作为最后一个字节发送；
  - 根据收到的最后一个字节自动进行 CRC 错误校验；
- 可触发中断的主模式故障、上溢和 CRC 错误标志；
- 具有 DMA 功能的 1 字节发送和接收缓冲器：发送和接收请求；
- 支持接收和发送 FIFO，深度为 16Byte；
- 接收发送 FIFO 可配置中断阈值。

### 22.2.2. I2S 特性

- 全双工通信;
- 半双工通信（仅作为发送器或接收器）;
- 主模式或从模式操作;
- 8 位可编程线性预分频器，可实现精确的音频采样频率（从 8 kHz 到 192 kHz）;
- 数据格式可以是 16 位、24 位或 32 位;
- 数据包帧由音频通道固定为 16 位（可容纳 16 位数据帧）或 32 位（可容纳 16 位、24 位、32 位数据帧）;
- 可编程的时钟极性（就绪时的电平状态）;
- 从发送模式下的下溢标志、接收模式下的上溢标志（主模式和从模式），以及接收和发送模式下的帧错误标志（仅从模式）;
- 发送和接收使用同一个 16 位数据寄存器;
- 支持的 I²S 协议:
  - I²S Phillps 标准;
  - MSB 对齐标准（左对齐）;
  - LSB 对齐标准（右对齐）;
  - PCM 标准（在 16 位通道帧或扩展为 32 位通道帧的 16 位数据帧上进行短帧和长帧同步）;
- 数据方向始终为 MSB 在前;
- 用于发送和接收的 DMA 功能（16 位宽）;
- 包含硬件实现的收发 FIFO,深度为 16Byte;
- 硬件收发 FIFO 支持可配中断阈值;
- 可输出主时钟以驱动外部音频元件。比率固定为 $256 \times F_S$（其中 $F_S$ 为音频采样频率）;
- 两个 I²S（I²S2 和 I²S3）均有专用的 PLL (PLLI2S)，可生成更为精确的时钟;
- I²S（I²S2 和 I²S3）时钟可由 I2S_CKIN 引脚上的外部时钟提供。

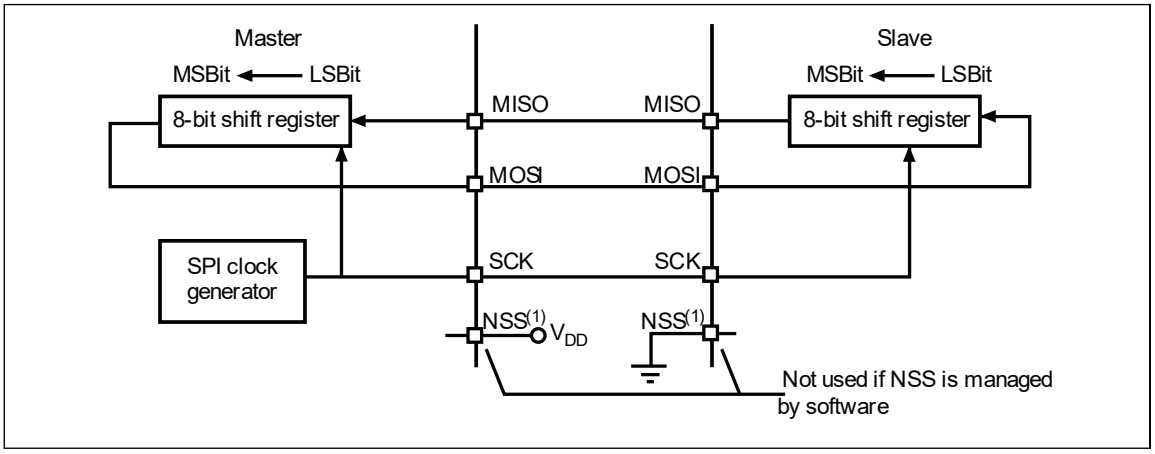
## 22.3. SPI 功能说明

### 22.3.1. 一般说明

通常， SPI 通过 4 个引脚与外部器件连接:

- MISO: 主输入/从输出数据。此引脚可用于在从模式下发送数据和在主模式下接收数据。
- MOSI: 主输出/从输入数据。此引脚可用于在主模式下发送数据和在从模式下接收数据。
- SCK: 用于 SPI 主器件的串行时钟输出以及 SPI 从器件的串行时钟输入。
- NSS: 从器件选择。这是用于选择从器件的可选引脚。此引脚用作“片选”，可让 SPI 主器件与从器件进行单独通信，从而并避免数据线上的竞争。从器件的 NSS 输入可由主器件上的标准 IO 端口驱动。NSS 引脚在使能（SSOE 位）时还可用作输出，并可在 SPI 处于主模式配置时驱动为低电平。通过这种方式，只要器件配置成 NSS 硬件管理模式，所有连接到该主器件 NSS 引脚的其它器件 NSS 引脚都将呈现低电平，并因此而作为从器件。当配置为主模式，且 NSS 配置为输入（MSTR=1 且 SSOE=0）时，如果 NSS 拉至低电平，SPI 将进入主模式故障状态: MSTR 位自动清零，并且器件配置为从模式。

下图给出了单个主器件和单个从器件之间的互连的基本示例。




1. *此处 NSS 引脚配置为输入。*

图 22-1 单个主器件/单个从器件应用

MOSI 引脚连接在一起，MISO 引脚连接在一起。通过这种方式，主器件和从器件之间以串行方式传输数据（最高有效位在前）。

通信始终由主器件发起。当主器件通过 MOSI 引脚向从器件发送数据时，从器件同时通过 MISO 引脚做出响应。这是一个数据输出和数据输入都由同一时钟进行同步的 1 全双工通信过程。

**从器件选择 (NSS) 引脚管理**

可以使用 SPI_CR1 寄存器中的 SSM 位设置硬件或软件管理从器件选择。

- 软件管理 NSS (SSM = 1)
- 从器件选择信息在内部由 SPI_CR1 寄存器中的 SSI 位的值驱动。外部 NSS 引脚空闲，可供其它应用使用。
- 硬件管理 NSS (SSM = 0)
- 根据 NSS 输出配置（SPI_CR1 寄存器中的 SSOE 位），硬件管理 NSS 有两种模式。
  - NSS 输出使能（SSM = 0 ，SSOE = 1）

    仅当器件在主模式下工作时才使用此配置。当主器件开始通信时，NSS 信号驱动为低电平，并保持到 SPI 被关闭为止。
  - NSS 输出禁止（SSM = 0 ，SSOE = 0）

    对于在主模式下工作的器件，此配置允许多主模式功能。对于设置为从模式的器件，NSS 引脚用作传统 NSS 输入：在 NSS 为低电平时片选该从器件，在 NSS 为 高电平时取消对它的片选。

**时钟相位和时钟极性**

通过 SPI_CR1 寄存器中的 CPOL 和 CPHA 位，可以用软件选择四种可能的时序关系。CPOL（时钟极性）位控制不传任何数据时的时钟电平状态。此位对主器件和从器件都有作用。如果复位 CPOL ，SCK 引脚在空闲状态处于低电平。如果将 CPOL 置 1，SCK 引脚在空闲状态处于高电平。

如果将 CPHA（时钟相位）位置 1，则 SCK 引脚上的第二个边沿（如果复位 CPOL 位，则为下降沿；如果将 CPOL 位置 1，则为上升沿）对 MSBit 采样。即，在第二个时钟边沿锁存数据。如果复位 CPHA 位，则 SCK 引脚上的第一个边沿（如果将 CPOL 位置 1，则为下降沿；如果复位 CPOL 位，则为上升沿）对 MSBit 采样。即在第一个时钟边沿锁存数据。

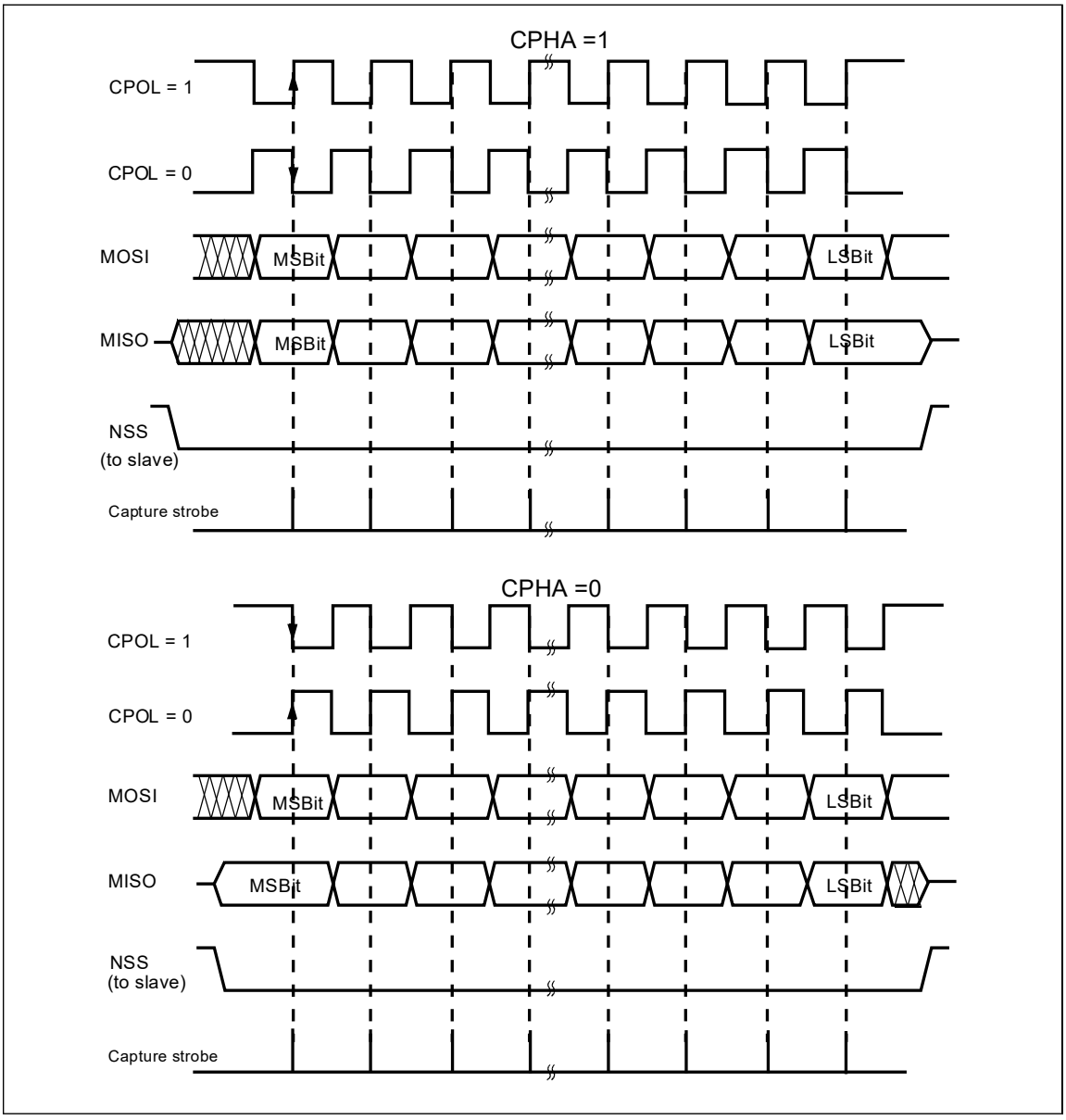
CPOL（时钟极性）和 CPHA（时钟相位）位的组合用于选择数据捕获时钟边沿。

*注意： 在切换 CPOL/CPHA 位之前，必须通过复位 SPE 位来关闭 SPI。*

必须以同一时序模式对主器件和从器件进行编程。

SCK 的空闲状态必须与 SPI_CR1 寄存器中选择的极性相对应(如果 CPOL=1, 则上拉 SCK；如果 CPOL=0, 则下拉 SCK)。

通过 SPI_CR1 寄存器中的 DFF 位选择数据帧格式（8 或 16 位），该格式决定了发送/接收过程中的数据长度。




1. *所示的时序为 SPI_CR1 寄存器中的 LSBFIRST 位复位时的时序。*

图 22-2 数据时钟时序图

**数据帧格式**

移出数据时 MSB 在前还是 LSB 在前取决于 SPI_CR1 寄存器中 LSBFIRST 位的值。

每个数据帧的长度均为 8 位或 16 位，具体取决于使用 SPI_CR1 寄存器中的 DFF 位。所选的数据帧格式适用于发送和/或接收。

### 22.3.2. 把 SPI 配置成从器件

在从模式配置中，从 SCK 引脚上接收主器件的串行时钟。SPI_CR1 寄存器的 BR[2:0] 位中设置的值不会

影响数据传输率。

*注意：建议在主器件发送时钟前使能 SPI 从器件。否则，数据传输可能会不正常。在主时钟的第一个边沿到来之前或者正在进行的通信结束之前，从器件的数据寄存器就需要准备就绪。在使能从器件和主器件之前，必须将通信时钟的极性设置为空闲时的时钟电平。*

按照下面的步骤把 SPI 模式配置成从模式：

- 设置 DFF 位，以定义 8 或 16 位数据帧格式。
- 选择 CPOL 和 CPHA 位，以定义数据传输和串行时钟之间的关系（四种关系中的一种）。要实现正确的数据传输，必须以相同方式在从器件和主器件中配置 CPOL 和 CPHA 位。如果通过 SPI_CR2 寄存器中的 FRF 位选择 TI 模式，则不需要此步骤。
- 帧格式（MSB 在前或 LSB 在前取决于 SPI_CR1 寄存器中 LSBFIRST 位的值）必须与主器件的帧格式相同。如果选择 TI 模式，则不需要此步骤。
- 在硬件模式下，NSS 引脚在整个字节发送序列期间都必须连接到低电平。在 NSS 软件模式下，将 SPI_CR1 寄存器中的 SSM 位置 1，将 SSI 位清零。如果选择 TI 模式，则不需要此步骤。
- 将 SPI_CR2 寄存器中的 FRF 位置 1，以选择 TI 模式协议进行串行通信。
- 将 MSTR 位清零，并将 SPE 位置 1（两个位均在 SPI_CR1 寄存器中）。

在此配置中，MOSI 引脚为数据输入，MISO 引脚为数据输出。

**发送序列**

数据字节在写周期内被并行加载到发送缓冲区中。

当从器件在其 MOSI 引脚上收到时钟信号和数据的最高有效位时，发送序列开始。其余位（8 位数据帧格式中的 7 个位，16 位数据帧格式中的 15 个位）将加载到移位寄存器中。 SPI_SR 寄存器中的 TXE 标志在数据从发送缓冲区传输到移位寄存器时置 1，并且在 SPI_CR2 寄存器中的 TXEIE 位置 1 时将生成中断。

**接收序列**

对于接收器，在数据传输完成时：

- 移位寄存器中的数据将传输到接收缓冲区，并且 RXNE 标志（SPI_SR 寄存器）置 1。
- 如果 SPI_CR2 寄存器中的 RXNEIE 位置 1，则生成中断。

在出现最后一个采样时钟边沿后，RXNE 位置 1，移位寄存器中接收的数据字节被拷贝到接收缓冲区中。当读取 SPI_DR 寄存器时，SPI 外设将返回此缓冲值。

通过读取 SPI_DR 寄存器将 RXNE 位清零。

**从模式下的 SPI TI 协议**

在从模式下，SPI 接口与 TI 协议兼容。可以使用 SPI_CR2 寄存器的 FRF 位来配置从 SPI 串行通信，以兼容此协议。

时钟极性和相位都被强制为遵循 TI 协议，和 SPI_CR1 中的设置无关。NSS 管理也特定于 TI 协议，这使用户无需通过设置 SPI_CR1 和 SPI_CR2 寄存器（例如 SSM、SSI、SSOE） 来对 NSS 管理进行设置。

在从模式下使用 SPI 波特率预分频器来控制 MISO 引脚状态切换到高阻态的时刻。可以使用任意波特率，因此可以非常灵活地确定此时刻。但是，波特率通常设置为外部主时钟波特率。MISO 信号变为高阻态的时间 ($t_{release}$) 取决于芯片内部电路同步以及通过 SPI_CR1 寄存器的 BR[2:0] 设置的波特率值。具体公式如下：

$$\frac{t_{baud_rate}}{2} + 4 \times t_{PCLK} < t_{release} < \frac{t_{baud_rate}}{2} + 6 \times t_{PCLK}$$

*注意：此特性不适用于 Motorola SPI 通信（FRF 位为 0）。*

要在从器件发送器模式下使用错误中断 (ERRIE = 1) 检测 TI 帧错误，必须通过将 SPI_CR1 寄存器中的 BIDIMODE 和 BIDIOE 置 1 来将 SPI 配置为双线单向模式。当 BIDIMODE 置为 0 时，OVR 将置 1，

因为始终不会读取数据寄存器从而始终生成溢出错误中断；而当 BIDIMODE 置 1 时，不会接收数据，也不会将 OVR 置 1。

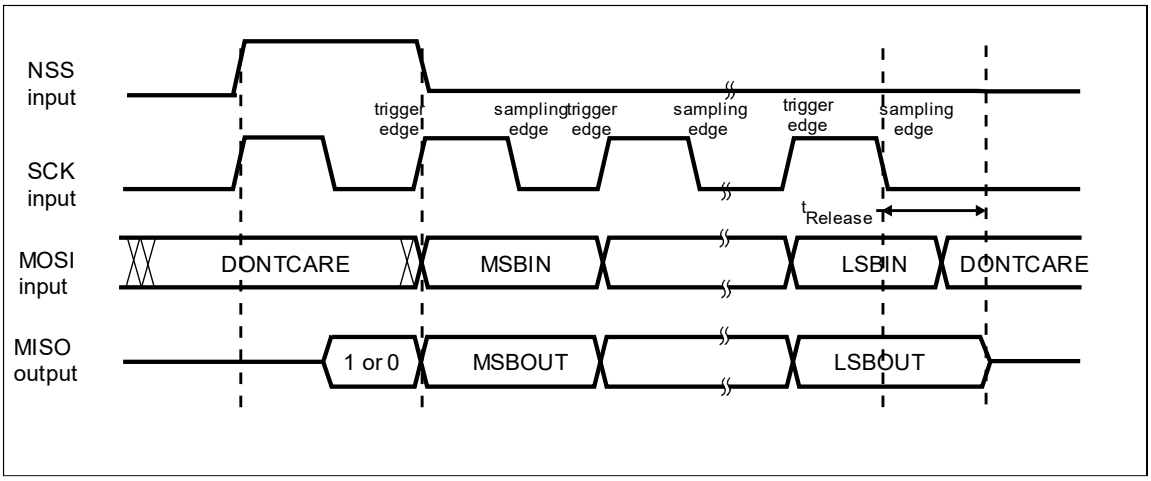



图 22-3 TI 模式 – 从模式，单次传输

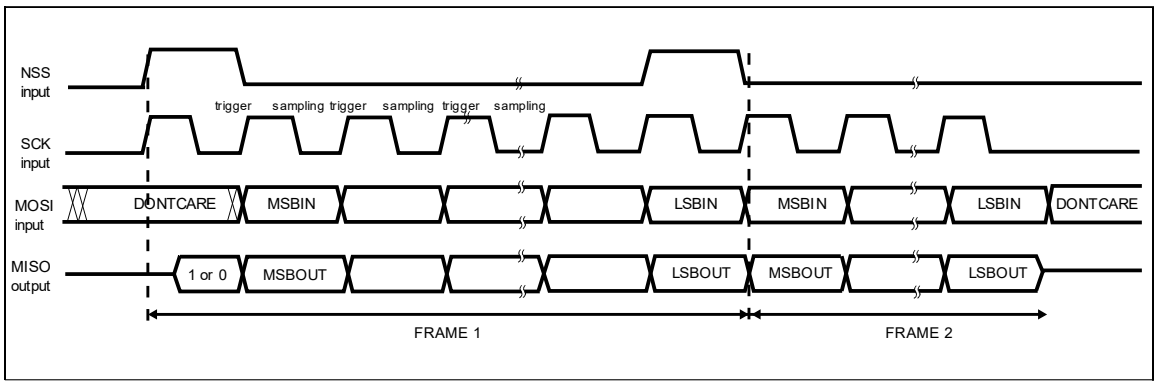



图 22-4 TI 模式 – 从模式，连续传输

### 22.3.3. 把 SPI 配置成主器件

在主模式配置下，在 SCK 引脚上输出串行时钟。

- 设置 BR[2:0] 位以定义串行时钟波特率（参见 SPI_CR1 寄存器）。
- 选择 CPOL 和 CPHA 位，以定义数据传输和串行时钟之间的关系（四种关系中的一种）。如果选择 TI 模式，则不需要此步骤。
- 设置 DFF 位，以定义 8 或 16 位
- 数据帧格式
- 配置 SPI_CR1 寄存器中的 LSBFIRST 位以定义帧格式。如果选择 TI 模式，则不需要此步骤。
- 如果 NSS 引脚配置成输入，在 NSS 硬件模式下，NSS 引脚在整个字节发送序列期间都连接到高电平信号；在 NSS 软件模式下，将 SPI_CR1 寄存器中的 SSM 和 SSI 位置 1。如果 NSS 引脚配置成输出，只应将 SSOE 位置 1。如果选择 TI 模式，则不需要此步骤。
- 将 SPI_CR2 中的 FRF 位置 1，以选择 TI 协议进行串行通信。
- MSTR 和 SPE 位必须置 1（仅当 NSS 引脚与高电平信号连接时，这两个位才保持置 1）。

在此配置中，MOSI 引脚为数据输出， MISO 引脚为数据输入。

**发送序列**

在发送缓冲区中写入字节时，发送序列开始。

在第一个位传输期间，数据字节（从内部总线）并行加载到移位寄存器中，然后以串行方式移 出到 MOSI 引脚，至于是 MSB 在前还是 LSB 在前则取决于 SPI_CR1 寄存器中的 LSBFIRST 位。TXE 标志在数据从发送缓冲区传输到移位寄存器时置 1，并且在 SPI_CR2 寄存器中的 TXEIE 位置 1 时将生成中断。

**接收序列**

对于接收器，在数据传输完成时：

- 移位寄存器中的数据将传输到接收缓冲区，并且 RXNE 标志置 1；
- 如果 SPI_CR2 寄存器中的 RXNEIE 位置 1，则生成中断。

在出现最后一个采样时钟边沿时，RXNE 位置 1，移位寄存器中接收的数据字节被拷贝到接收缓冲区中。当读取 SPI_DR 寄存器时，SPI 外设将返回此缓冲值。

通过读取 SPI_DR 寄存器将 RXNE 位清零。

如果在发送开始后将要发送的下一个数据置于发送缓冲区，则可保持连续的发送流。请注意，仅当 TXE 标志为 1 时，才可以对发送缓冲区执行写操作。

*注意： 如果与之通信的从器件需要在每个字节传输之间拉低片选信号，必须将该主器件的NSS 配置成GPIO，或使用另外GPIO，通过软件控制从器件的片选。*

**主模式下的 SPI TI 协议**

在主模式下，SPI 接口与 TI 协议兼容。可以使用 SPI_CR2 寄存器的 FRF 位来配置主 SPI 串行通信，以兼容此协议。

时钟极性和相位都被强制为遵循 TI 协议，和 SPI_CR1 中的设置无关。NSS 管理也特定于 TI 协议，这使用户无需通过设置 SPI_CR1 和 SPI_CR2 寄存器（例如 SSM、SSI、SSOE）来对 NSS 管理进行设置。

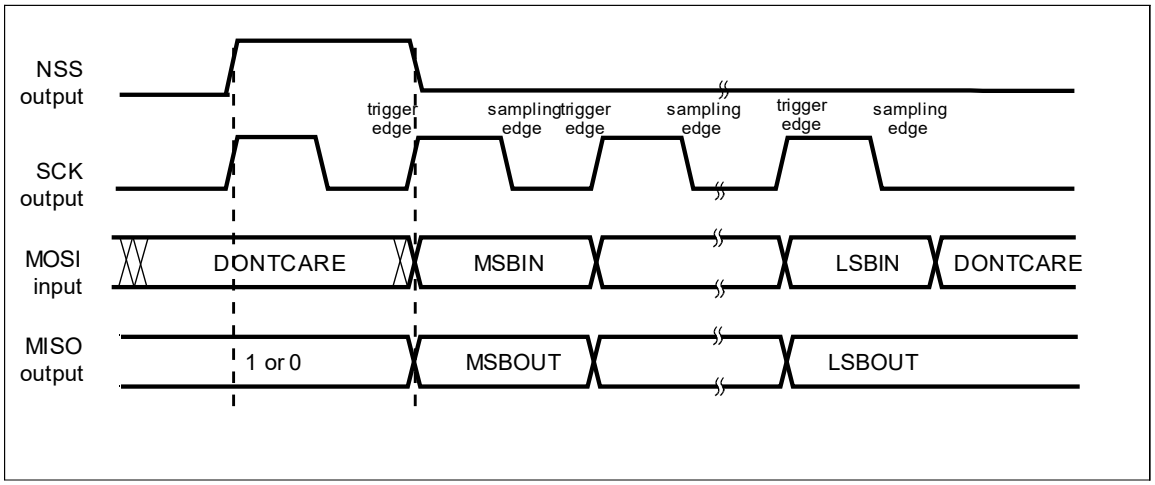



图 22-5 TI 模式 – 主模式，单次传输

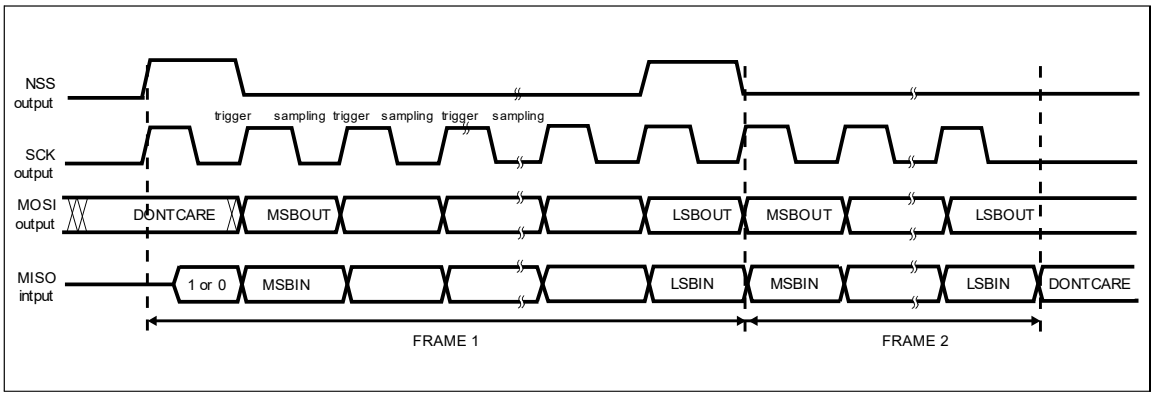



图 22-6 TI 模式 – 主模式，连续传输

### 22.3.4. 配置 SPI 进行半双工通信

SPI 能够在以下两种配置中以半双工模式工作。

- 1 个时钟和 1 条双向数据线;
- 1 个时钟和 1 条数据线（只接收或只发送）‘’

**1 个时钟和 1 条双向数据线 (BIDIMODE=1)**

可将 SPI_CR1 寄存器中的 BIDIMODE 位置 1 来使能此模式。在此模式下，SCK 用于时钟，MOSI（主模式下）或 MISO（从模式下）用于数据通信。通过 SPI_CR1 寄存器中的 BIDIOE 位来选择传输方向（输入/输出）。当该位置 1 时，数据线为输出，否则为输入。

**1 个时钟和 1 条单向数据线 (BIDIMODE=0)**

在此模式下，应用程序可使用 SPI 的只发送或只接收功能。

- 只发送模式类似于全双工模式（BIDIMODE=0、RXONLY=0）：在发送引脚（主模式下的 MOSI 或从模式下的 MISO）上发送数据，接收引脚（主模式下的 MISO 或从模式下的 MOSI）可用作通用 IO。在这种情况下，应用程序只需要忽略接收缓冲区（即使读取数据寄存器，它也不包含接收值）。
- 只接收模式下，应用程序可将 SPI_CR2 寄存器中的 RXONLY 位置 1 来关闭 SPI 输出功能。在这种情况下，发送 IO 引脚（主模式下的 MOSI 或从模式下的 MISO）可用于其它用途。

要在只接收模式下开始通信，配置并使能 SPI:

- 在主模式下，通信会立即开始，并在 SPE 位清零且当前接收结束时停止。在此模式下无需读取 BSY 标志。在进行 SPI 通信时，该标志始终置 1。
- 在从模式下，只要 NSS 被拉低（或在 NSS 软件模式下将 SSI 位清零）并且一直有来自主器件的 SCK 输入，SPI 就会继续接收。

### 22.3.5. 数据发送和接收过程

**接收和发送缓冲区**

在接收过程中，数据收到后，先存储到内部接收缓冲区中；而在发送过程中，先将数据存储到内部发送缓冲区中，然后发送数据。

对 SPI_DR 寄存器的读访问将返回接收缓冲值，而对 SPI_DR 寄存器的写访问会将写入的数据存储到发送缓冲区中。

**在主模式下启动通信序列**

- 在全双工模式下（BIDIMODE=0 且 RXONLY=0）:

- 将数据写入到 SPI_DR 寄存器（发送缓冲区）时，通信序列启动。
- 随后在第一个位的发送期间，将数据从发送缓冲区并行加载到 8 位移位寄存器中，然后以串行方式将其移出到 MOSI 引脚。
- 同时，将 MISO 引脚上接收的数据以串行方式移入 8 位移位寄存器，然后并行加载到 SPI_DR 寄存器（接收缓冲区）中。

- 在单向只接收模式下（BIDIMODE=0 且 RXONLY=1）：
  - 只要 SPE = 1，通信序列就立即开始。
  - 只有接收器激活，并且在 MISO 引脚上接收的数据以串行方式移入 8 位移位寄存器，然后并行加载到 SPI_DR 寄存器（接收缓冲区）中。
- 在双向模式下，进行发送时（BIDIMODE=1 且 BIDIOE=1）：
  - 将数据写入到 SPI_DR 寄存器（发送缓冲区）时，通信序列启动。
  - 随后在第一个位的发送期间，将数据从发送缓冲区并行加载到 8 位移位寄存器中，然后以串行方式将其移出到 MOSI 引脚。
  - 不接收任何数据。
- 在双向模式下，进行接收时（BIDIMODE=1 且 BIDIOE=0）：
  - 只要 SPE=1 且 BIDIOE=0，通信序列就立即开始。
  - 在 MOSI 引脚上接收的数据以串行方式移入 8 位移位寄存器，然后并行加载到 SPI_DR 寄存器（接收缓冲区）中。
  - 发送器没有激活，因此不会有数据以串行方式移出 MOSI 引脚。

**在从模式下启动通信序列**

- 在全双工模式下（BIDIMODE=0 且 RXONLY=0）：
  - 当从器件在其 MOSI 引脚上收到时钟信号和数据的第一个位时，通信序列开始。其余 7 个位将加载到移位寄存器中。
  - 同时，在第一个位的发送期间，将数据从发送缓冲区并行加载到 8 位移位寄存器中，然后以串行方式将其移出到 MISO 引脚。在 SPI 主器件启动传输前，软件必须已把要从器件发送的数据写入发送缓冲区。
- 在单向只接收模式下（BIDIMODE=0 且 RXONLY=1）：
  - 当从器件在其 MOSI 引脚上收到时钟信号和数据的第一个位时，通信序列开始。其余 7 个位将加载到移位寄存器中。
  - 发送器没有激活，因此不会有数据以串行方式移出 MISO 引脚。
- 在双向模式下，进行发送时（BIDIMODE=1 且 BIDIOE=1）：
  - 当从器件收到时钟信号，并将发送缓冲区中的第一个位在 MISO 引脚上发送时，通信序列开始。
  - 随后在第一个位的发送期间，将数据从发送缓冲区并行加载到 8 位移位寄存器中，然后以串行方式将其移出到 MISO 引脚。在 SPI 主器件启动传输前，软件必须已把要从器件发送的数据写入发送缓冲区。
  - 不接收任何数据。
- 在双向模式下，进行接收时（BIDIMODE=1 且 BIDIOE=0）
  - 当从器件在其 MISO 引脚上收到时钟信号和数据的第一个位时，通信序列开始。
  - 在 MISO 引脚上接收的数据以串行方式移入 8 位移位寄存器，然后并行加载到 SPI_DR 寄存器（接收缓冲区）中。
  - 发送器没有激活，因此不会有数据以串行方式移出 MISO 引脚。

### 处理数据发送与接收

将数据从发送缓冲区传输到移位寄存器时，TXE 标志（发送缓冲区为空）置 1。该标志表示内部发送缓冲区已准备好加载接下来的数据。如果 SPI_CR2 寄存器中的 TXEIE 位置 1，可产生中断。通过对 SPI_DR 寄存器执行写操作将 TXE 位清零。

*注意： 软件必须确保在尝试写入发送缓冲区之前 TXE 标志已置 1。否则，将覆盖之前写入发送缓冲区的数据。*

将数据从移位寄存器传输到接收缓冲区时，RXNE 标志（接收缓冲区非空）会在最后一个采样时钟边沿置 1。它表示已准备好从 SPI_DR 寄存器中读取数据。如果 SPI_CR2 寄存器中的 RXNEIE 位置 1，可产生中断。通过读取 SPI_DR 寄存器将 RXNE 位清零。

对于某些配置，可以在最后一次数据传输期间使用 BSY 标志来等待传输完成。

**主模式或从模式下的全双工发送和接收过程（BIDIMODE=0 且 RXONLY=0）**

软件必须遵循以下步骤来发送和接收数据：

- 通过将 SPE 位置 1 来使能 SPI。
- 将第一个要发送的数据项写入 SPI_DR 寄存器（此操作会将 TXE 标志清零）。
- 等待至 TXE=1，然后写入要发送的第二个数据项。然后等待至 RXNE=1，读取 SPI_DR 以获取收到的第一个数据项（此操作会将 RXNE 位清零）。对每个要发送/接收的数据项重复此操作，直到第 n-1 个接收的数据为止。
- 等待至 RXNE=1，然后读取最后接收的数据。
- 等待至 TXE=1，然后等待至 BSY=0，再关闭 SPI。
- 此外，还可以使用在 RXNE 或 TXE 标志所产生的中断对应的各个中断子程序来实现该过程。

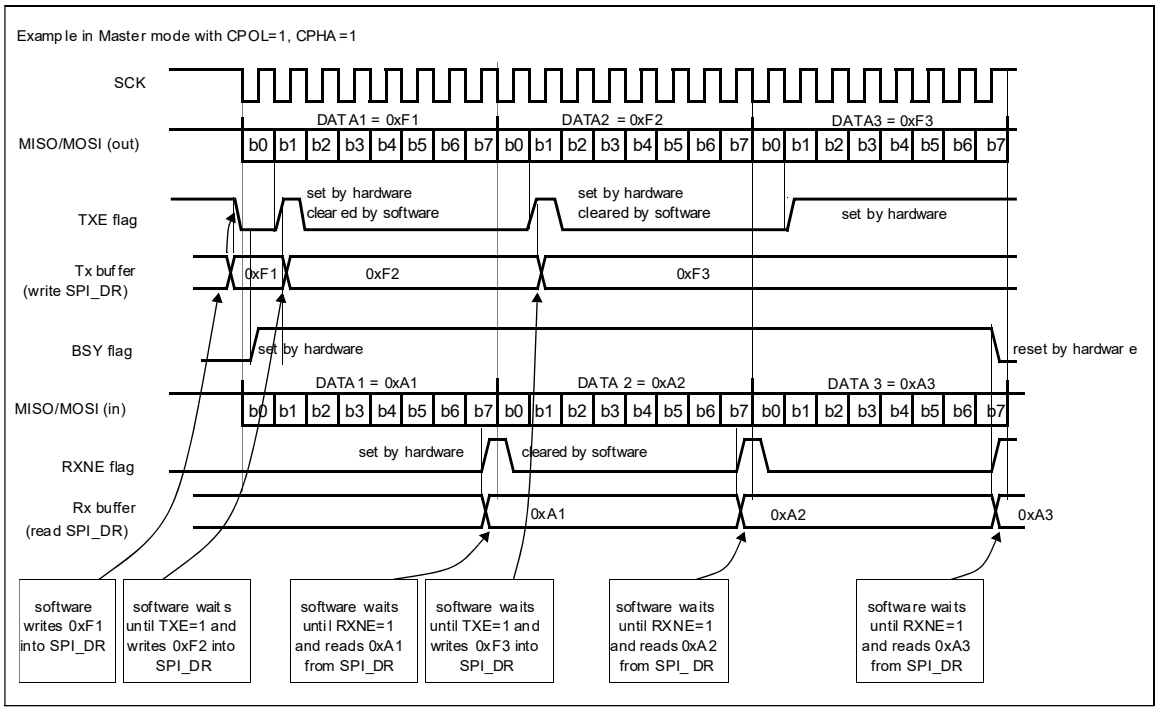



图 22-7 主/全双工模式下的 TXE/RXNE/BSY 行为（在连续传输的情况下）

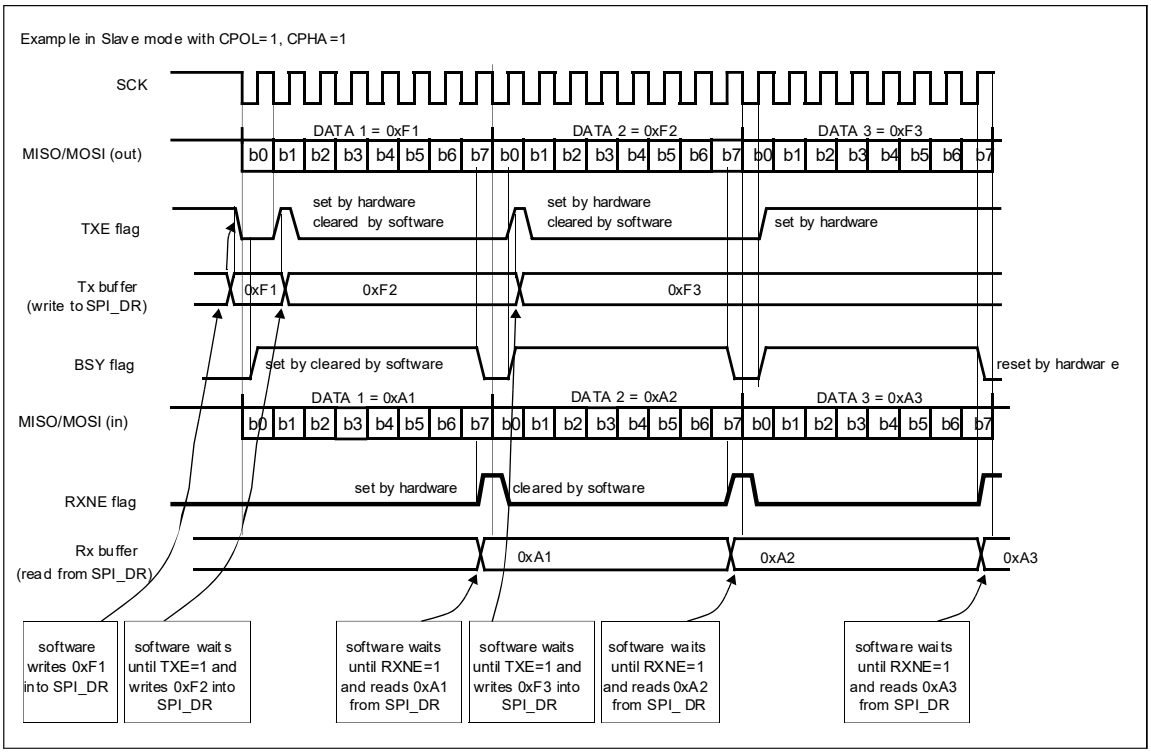



图 22-8 从/全双工模式下的 TXE/RXNE/BSY 行为（在连续传输的情况下）

**只发送模式下的数据发送过程（BIDIMODE=0 、RXONLY=0）**

在此模式下，可以按下文所述简化过程，并且可使用 BSY 位等待发送完成:

- 通过将 SPE 位置 1 来使能 SPI。
- 将第一个要发送的数据项写入 SPI_DR 寄存器（此操作会将 TXE 标志清零）。
- 等待至 TXE=1，然后写入下一个要发送的数据项。对每个要发送的数据项重复此步骤。
- 将最后一个数据项写入 SPI_DR 寄存器后，等待至 TXE=1，然后等待至 BSY=0，这表示最后的数据发送完成。

此外，还可以使用在 TXE 标志所产生的中断对应的中断子程序来实现该过程。

*注意：在不连续通信期间，在对 SPI_DR 执行写操作与 BSY 位置 1 之间有 2 个 APB 时钟周期的延迟。因此，在只发送模式下，写入最后的数据后，必须先等待 TXE 位置 1，然后等待 BSY 位清零。*

在只发送模式下，发送两个数据项后，SPI_SR 寄存器中的 OVR 标志将置 1，因为始终不会读取接收的数据。

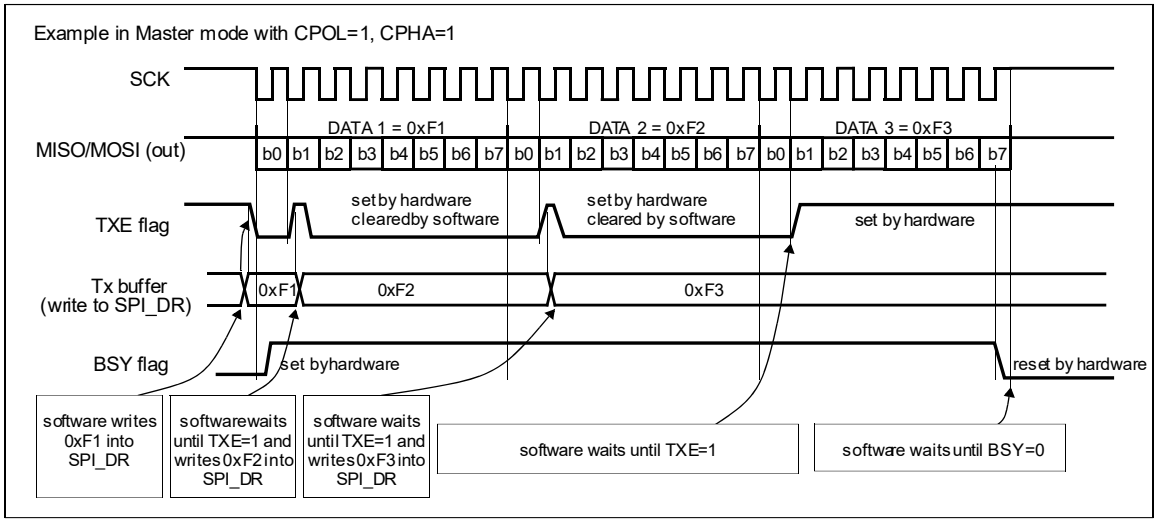



图 22-9 主设备只发送模式下的 TXE/ BSY 行为（在连续传输的情况下）

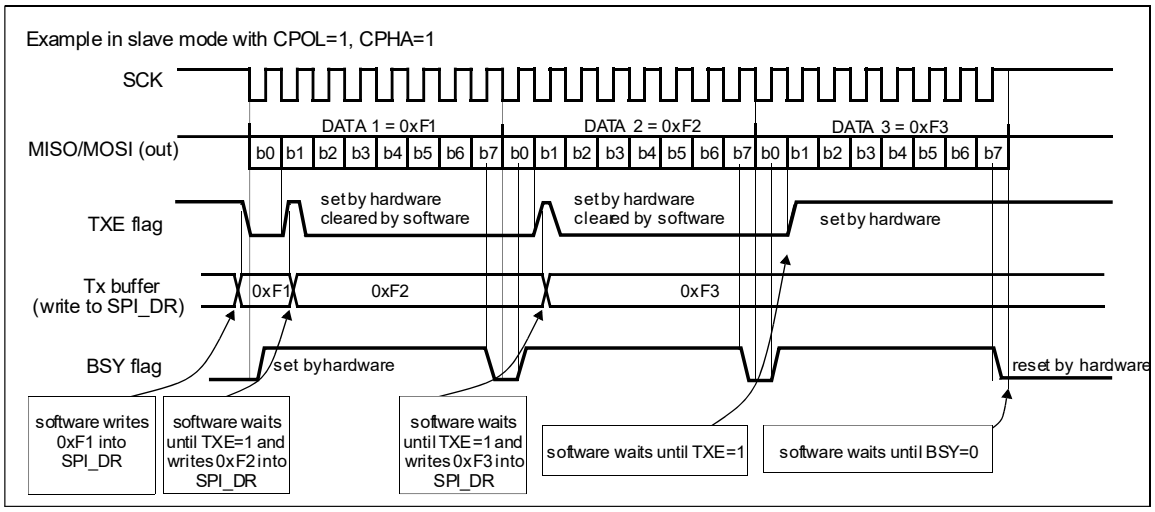



图 22-10 从设备只发送模式下的 TXE/ BSY 行为（在连续传输的情况下）

**单线双向模式下的发送过程（BIDIMODE=1 且 BIDIOE=1）**

在此模式下，过程与只发送模式下的过程相似，除了在使能 SPI 前必须将 SPI_CR2 寄存器中的 BIDIMODE 位和 BIDIOE 位均置 1。

**单向只接收过程（BIDIMODE=0 且 RXONLY=1）**

在此模式下，可以按如下所述简化过程：

- 将 SPI_CR2 寄存器中的 RXONLY 位置 1。
- 通过将 SPE 位置 1 使能 SPI：
  - 在主模式下，这会立即激活 SCK 时钟的产生，并以串行方式接收数据，直到关闭 SPI (SPE=0)。
  - 在从模式下，当 SPI 主器件将该从器件的 NSS 驱动为低电平并输出 SCK 时钟时，接收数据。
- 等待 RXNE=1，然后读取 SPI_DR 寄存器以获取接收的数据（此操作会将 RXNE 位清零）。对每个要接收的数据项重复此操作。

此外，还可以使用在 RXNE 标志所产生的中断对应的中断子程序来实现该过程。

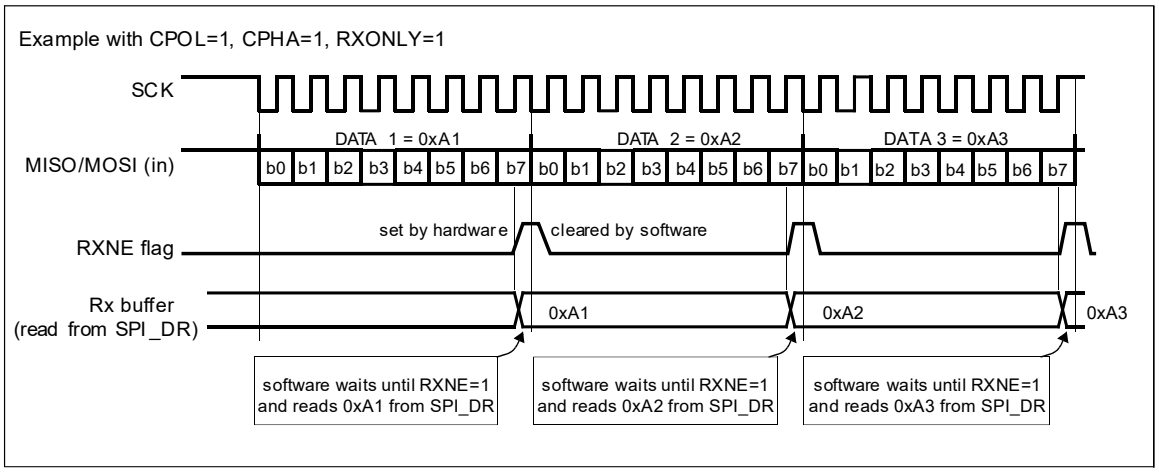



图 22-11 只接收模式下的 RXNE 行为（在连续传输的情况下）

### 单线双向模式下的接收过程（BIDIMODE=1 和 BIDIOE=0）

在此模式下，过程与只接收模式的过程相似，除了在使能 SPI 前必须将 SPI_CR2 寄存器中的 BIDIMODE 位置 1 并将 BIDIOE 清零。

### 连续传输和间断传输

在主模式下发送数据时，如果软件速度快到能在下一个数据传输完成之前响应该数据从数据寄存器传输到移位寄存器所触发的 TXE 中断，并能立即完成再下一个数据的写 SPI_DR 操作，则这种通信称为连续通信。在这种情况下，各数据项之间 SPI 时钟的生成不会间断，并且各数据传输之间不会清零 BSY 位。

相反，如果软件速度不够快，则可能导致通信中断。在这种情况下，各数据传输之间会清零 BSY 位。

在主设备仅接收模式 (RXONLY=1) 下，通信始终是连续的，且 BSY 标志始终读为 1。

在从模式下，通信的连续性由 SPI 主器件决定。任何情况下（即使通信是连续的），在各个数据传输之间 BSY 标志都会短暂变为低电平，持续时间为一个 SPI 时钟周期。

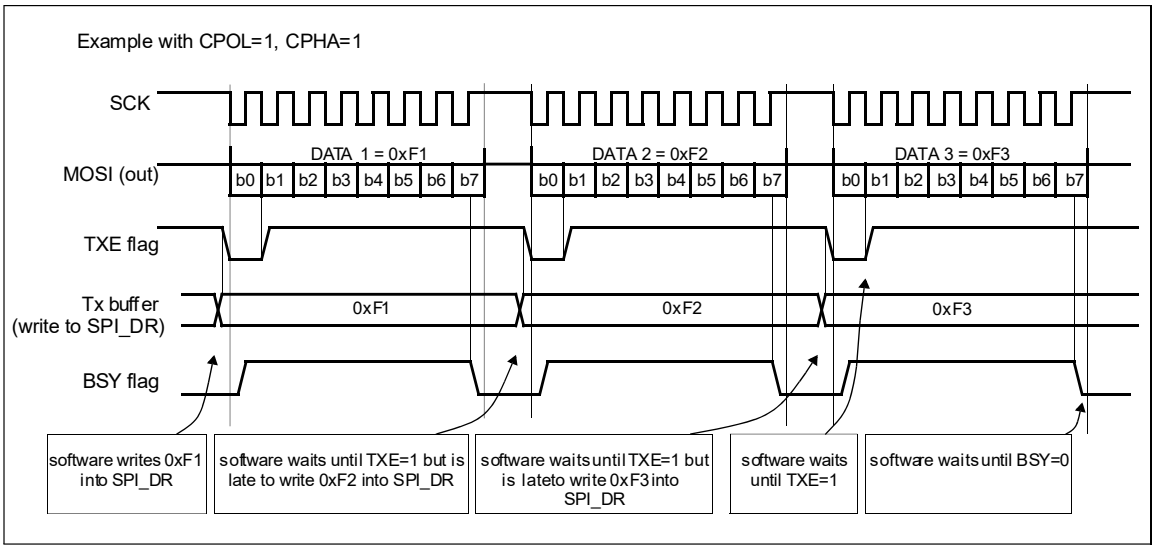



图 22-12 发送时的 TXE/BSY 行为（在间断传输的情况下）

### 22.3.6. CRC 计算

为确保通信的可靠性, SPI 模块实现了硬件 CRC 功能。针对发送的数据和接收的数据分别实现 CRC 计算。使用可编程的多项式对每个位来计算 CRC。在由 SPI_CR1 寄存器中的 CPHA 位和 CPOL 位定义的采样时钟边沿采样每个位来进行计算。

*注意： 该 SPI 模块提供两种 CRC 计算标准，具体取决于为发送和/或接收选择的数据帧格式：8 位数据采用 CR8 ，16 位数据采用 CRC16。*

通过将 SPI_CR1 寄存器中的 CRCEN 位置 1 来使能 CRC 的计算。此操作将复位 CRC 寄存器（SPI_RXCRCR 和 SPI_TXCRCR）。在全双工或只发送模式下，如果传输由软件（CPU 模式）管理，则在将最后传输的数据写入 SPI_DR 后，必须立即对 CRCNEXT 位执行写操作。最后一次数据传输结束时，将发送 SPI_TXCRCR 值。

在只接收模式下，如果传输由软件（CPU 模式）管理，则在接收到倒数第二个数据后，必须对 CRCNEXT 位执行写操作。在收到最后一个数据后会收到 CRC，然后执行 CRC 校验。

如果传输过程中出现数据损坏, 则在数据和 CRC 传输结束时, SPI_SR 寄存器中的 CRCERR 标志将置 1。

如果发送缓冲区中存在数据，则只有在发送数据字节后才会发送 CRC 值。在 CRC 发送期间， CRC 计算器处于关闭状态且寄存器值保持不变。

可通过以下步骤使用 CRC 进行 SPI 通信:

- 对 CPOL、CPHA、LSBFirst、BR、SSM、SSI 和 MSTR 值进行编程。
- 对 SPI_CRCPR 寄存器中的多项式进行编程。
- 通过将 SPI_CR1 寄存器中的 CRCEN 位置 1 来使能 CRC 计算。此操作还会将 SPI_RXCRCR 和 SPI_TXCRCR 寄存器清零。
- 通过将 SPI_CR1 寄存器中的 SPE 位置 1 使能 SPI。
- 启动并维持通信，直到只剩下一个字节或半字未发送或接收。
  - 在全双工或只发送模式下, 如果传输由软件管理, 则在向发送缓冲区写入最后一个字节或半字后，将 SPI_CR1 寄存器中的 CRCNEXT 位置 1，以表示在发送完最后一个字节后将发送 CRC。
  - 在只接收模式下，在接收倒数第二个数据后，立即将 CRCNEXT 位置 1，以便使 SPI 准备好在接收完最后一个数据后进入 CRC 阶段。在 CRC 传输期间，CRC 计算将冻结。
- 传输完最后一个字节或半字后，SPI 进入 CRC 传输和校验阶段。在全双工模式或只接收模式下，将接收的 CRC 与 SPI_RXCRCR 值进行比较。如果两个值不匹配, 则 SPI_SR 中的 CRCERR 标志将置 1，并且在 SPI_CR2 寄存器中的 ERRIE 位置 1 时会产生中断。

*注意： 当 SPI 处于从模式时，注意只能在时钟稳定（即，时钟处于空闲电平）时使能 CRC 计算。否则，可能导致 CRC 计算错误。因为，只要 CRCEN 位置 1，无论 SPE 位的值如何，只要有时钟输入，CRC 计算器就开始工作。*

在 SPI 通信时钟频率较高的情况下，发送 CRC 时务必小心。由于在 CRC 传输阶段 CPU 应尽可能空闲，因此禁止在 CRC 发送阶段调用函数，以便避免最后的数据和 CRC 接收出错。实际上，在发送/接收最后的数据之前必须对 CRCNEXT 位执行写操作。

SPI 通信时钟频率较高时，建议使用 DMA 模式来避免由于 CPU 访问影响 SPI 带宽而导致 SPI 速度性能下降。

如果将器件配置为从器件，并且使用 NSS 硬件模式，则需要在数据阶段和 CRC 阶段之间将 NSS 引脚保持为低电平。

当 SPI 配置为从模式并且 CRC 功能已使能时, 即使 NSS 引脚上为高电平, 也会进行 CRC 计算。例如，在多从模式环境下可能出现这种情况，此时通信主器件会交替寻址从器件。

在对从器件片选的切换期间内，应在主器件和从器件两端同时将 CRC 值清零，以重新同步主从双方的 CRC 计算。

要将 CRC 清零，请按以下步骤操作:

- 关闭 SPI (SPE = 0)
- 将 CRCEN 位清零
- 将 CRCEN 位置 1
- 使能 SPI (SPE = 1)

### 22.3.7. 状态标志

应用可通过三种状态标志监视 SPI 总线的状态。

**发送缓冲区为空 (TXE)**

此标志置 1 时，表示发送缓冲区为空，可以将待发送的下一个数据加载到缓冲区中。对 SPI_DR 寄存器执行写操作时，将清零 TXE 标志。

**接收缓冲区非空 (RXNE)**

此标志置 1 时，表示接收缓冲区中存在有效的已接收数据。读取 SPI_DR 时，将清零该标志。

**BUSY**

BSY 标志由硬件置 1 和清零（对此标志执行写操作没有任何作用）。BSY 标志用于指示 SPI 通信的状态。

BSY 置 1 时，表示 SPI 正忙于通信。在主模式下的双向通信接收模式（MSTR=1 且 BDM=1 且 BDOE=0）有一个例外情况，BSY 标志在接收过程中保持低电平。

如果软件要关闭 SPI 并进入停止模式（或关闭外设时钟），可使用 BSY 标志检测传输是否结束以避免破坏最后一个数据的传输。为此，必须严格遵循下述步骤。

BSY 标志还可用于避免在多主模式系统中发生写冲突。

传输开始时，BSY 标志将置 1，但在主模式下的双向通信接收模式（MSTR=1 且 BDM=1 且 BDOE=0）下例外。

在以下情况硬件将清零该标志：

- 传输完成时（主模式下的连续通信除外）；
- 关闭 SPI 时；
- 发生主模式故障时 (MODF=1)。

当通信不连续时，BSY 标志在各通信之间处于低电平。

当通信连续时：

- 在主模式下，BSY 标志在所有传输期间均保持高电平；
- 在从模式下，BSY 标志在各传输之间的一个 SPI 时钟周期内变为低电平。

*注意：　请勿使用 BSY 标志处理每次数据发送或接收，最好改用 TXE 标志和 RXNE 标志。*

**发送 FIFO 满 (TFF)**

此标志只有在使能发送 FIFO 模式(TFEN = 1)时才有效。此标志置 1 时，表示发送 FIFO 已满。此时再次填充数据到 DR 寄存器就会出现数据溢出丢失的情况。

清零 TFF 标志的方法：

- 等待 SPI 将数据传输到移位寄存器，此时 TFFLR 小于 16，硬件自动清除标志位；
- 配置 TFFLR = 1 时将清除发送 FIFO 中的所有数据，硬件自动清除标志位。

**接收 FIFO 满 (RFF)**

此标志只有在使能接收 FIFO 模式(RFEN = 1)时才有效。此标志置 1 时，表示接收 FIFO 已满。此时如果还有通讯数据传输，就会出现后续接收到的数据溢出丢失的情况，但 FIFO 中原有数据不会丢失。

清零 RFF 标志的方法：

- 读取 SPI_DR 寄存器，此时 RFFLR 小于 16，硬件自动清除标志位；
- 配置 RFFLR = 1 时将清除接收 FIFO 中的所有数据，硬件自动清除标志位。

**发送 FIFO 将空 (TFE)**

此标志只有在使能发送 FIFO 模式(TFEN = 1)时才有效。此标志置 1 时，表示发送 FIFO 中的数据量小于等于设置的将空阈值。

清零 TFE 标志的方法：

- 对 SPI_DR 寄存器执行写操作时，直到 FIFO 中的数量大于阈值，硬件自动清除标志位。

**接收 FIFO 将满 (RFP)**

此标志只有在使能接收 FIFO 模式(RFEN = 1)时才有效。此标志置 1 时，表示接收 FIFO 中的数据量大于等于设置的将空阈值。

清零 RFP 标志的方法：

- 对 SPI_DR 寄存器执行读操作时，直到 FIFO 中的数量小于阈值，硬件自动清除标志位。

### 22.3.8. 关闭 SPI

传输终止时，应用可通过关闭 SPI 外设来停止通信。这通过将 SPE 位清零来完成。

对于某些配置，在传输进行时关闭 SPI 并进入停止模式会导致当前传输受损，并且或者 BSY 标志可能不可靠。

为避免上述后果，建议在关闭 SPI 时按以下步骤操作：

**在主模式或全双工从模式（BIDIMODE=0、RXONLY=0）下**

- 等待 RXNE=1 以接收最后的数据；
- 等待 TXE=1；
- 然后等待 BSY=0；
- 关闭 SPI (SPE=0)，最后进入停止模式（或关闭外设时钟）。

**在主模式或单向只发送从模式（BIDIMODE=0、RXONLY=0）或双向通信发送模式（BIDIMODE=1 、BIDIOE=1）下**

在最后的数据写入 SPI_DR 寄存器后：

- 等待 TXE=1
- 然后等待 BSY=0
- 关闭 SPI (SPE=0)，最后进入停止模式（或关闭外设时钟）

**在单向只接收主模式（MSTR=1、BIDIMODE=0、RXONLY=1）或双向通信接收模式（MSTR=1、BIDIMODE=1、BIDIOE=0）下**

必须以特殊方式管理这种情况，以避免多余的 SPI 数据传输。以下序列仅适用于 SPI Motorola 配置（FRF 位置 0）：

- 等待倒数第二个数据（第 n-1 个）对应的 RxNE 标志置位；
- 然后等待一个 SPI 时钟周期（使用软件循环），才能关闭 SPI (SPE=0)；
- 再等待最后的 RXNE=1，然后进入停止模式（或关闭外设时钟）。

当 SPI 配置为 TI 模式（FRF 位置 1）时，必须按以下步骤操作以避免当关闭 SPI 时在 NSS 上产生不需要的脉冲：

- 等待倒数第二个数据（第 n-1 个）对应的 RxNE 标志置位。
- 在以下窗口帧中使用软件循环关闭 SPI (SPE = 0)：

- 至少一个 SPI 时钟周期后，
- LSB 数据开始传输前。

注意： 在双向通信接收主模式（MSTR=1、BDM=1、BDOE=0）下， BSY 标志在传输期间保持低电平。

**在只接收从模式（MSTR=0 、BIDIMODE=0 、RXONLY=1）或双向通信接收模式（MSTR=0 、BIDIMODE=1 、BIDOE=0）下**

- 可以随时关闭 SPI（写入 SPE=0）：当前传输完成后，SPI 才被真正关闭；
- 之后，如果要进入停止模式，则必须首先等待至 BSY = 0，然后才能进入停止模式（或关闭外设时钟）。

### 22.3.9. 使用 DMA 进行 SPI 通信

要以最大速度工作，需要给 SPI 不断提供要发送的数据，并及时读取接收缓冲区中的数据，以避免上溢。为加速传输，SPI 提供了 DMA 功能，以实现简单的请求/应答协议。

当使能 SPI_CR2 寄存器中的使能位时，将请求 DMA 访问。发送缓冲区和接收缓冲区会发出各自的 DMA 请求：

- 在发送过程中，每次 TXE 位置 1 都会发出 DMA 请求。DMA 随后对 SPI_DR 寄存器执行写操作（此操作会将 TXE 标志清零）。
- 在接收过程中，每次 RXNE 位置 1 都会发出 DMA 请求。DMA 随后对 SPI_DR 寄存器执行读操作（此操作会将 RXNE 标志清零）。

当 SPI 仅用于发送数据时，可以只使能 SPI Tx DMA 通道。在这种情况下，OVE 标志会置 1，因为未读取接收的数据。

当 SPI 仅用于接收数据时，可以只使能 SPI Rx DMA 通道。

在发送模式下，DMA 完成了所有要发送数据的传输（DMA_ISR 寄存器中的 TCIF 标志置 1）后，可以对 BSY 标志进行监视，以确保 SPI 通信已完成。在关闭 SPI 或进入停止模式前必须执行此步骤，以避免损坏最后一次数据的发送。软件必须首先等待 TXE=1，再等待 BSY=0。

*注意： 在不连续通信期间，在对 SPI_DR 执行写操作与 BSY 位置 1 之间有 2 个 APB 时钟周期的延迟。因此，必须在写入最后的数据后，先等待 TXE=1，再等待 BSY=0。*

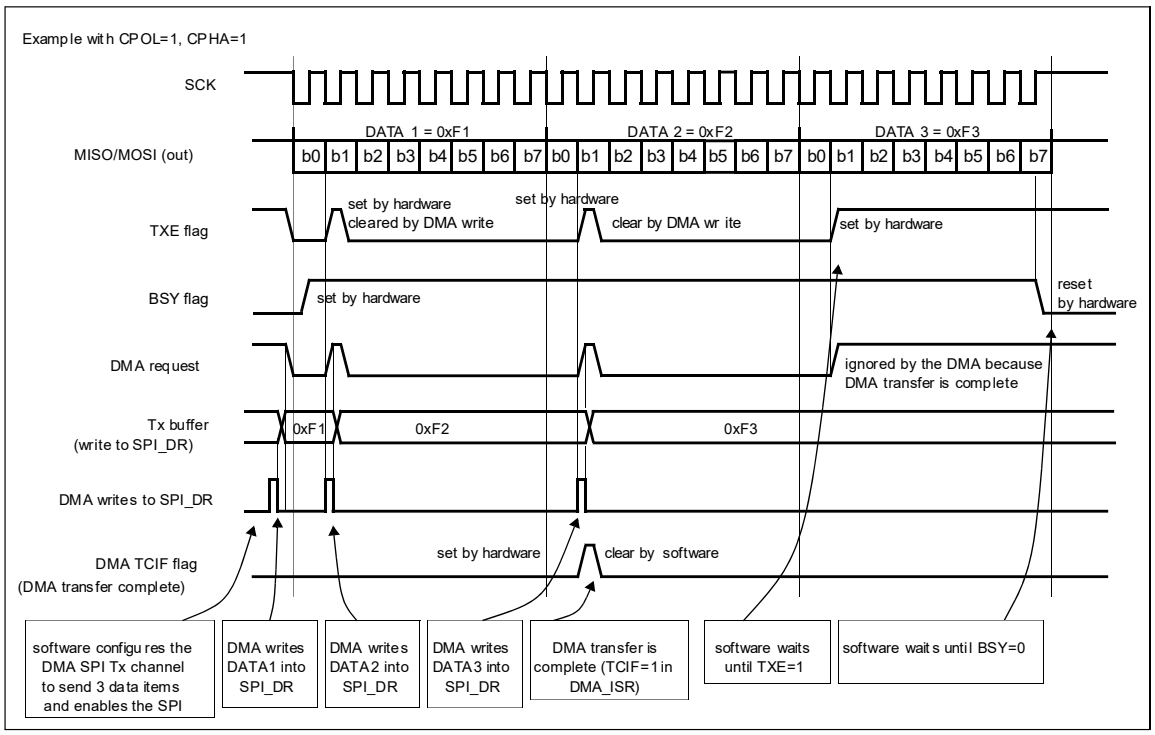



图 22-13 使用 DMA 进行发送

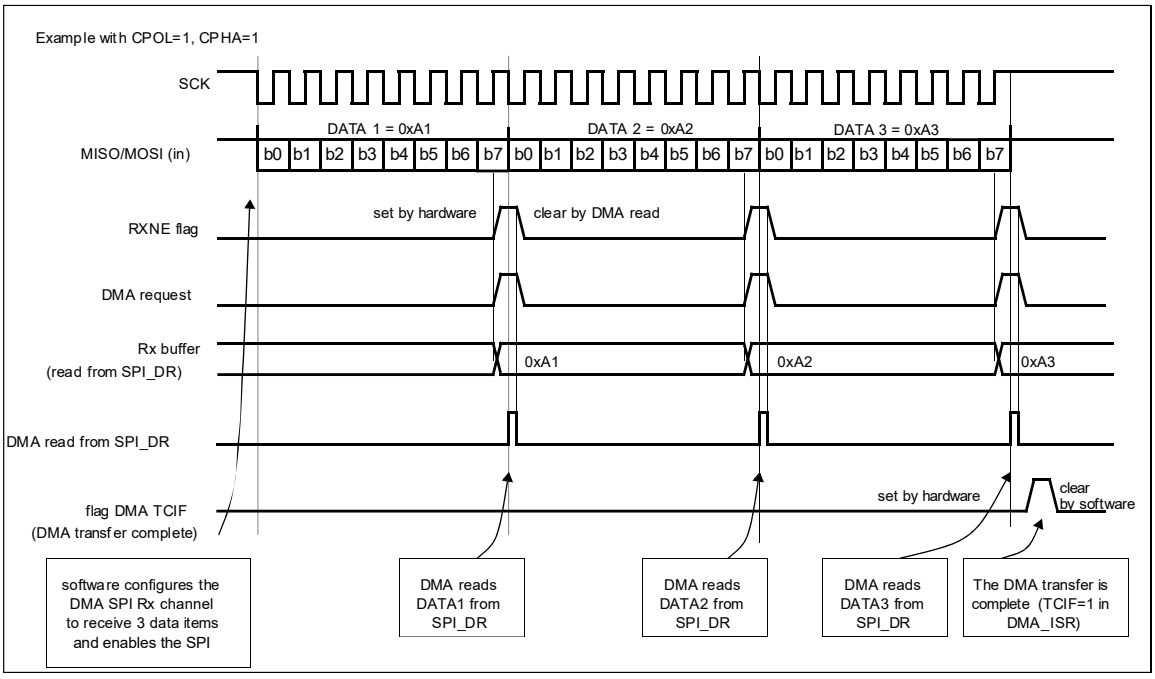



图 22-14 使用 DMA 进行接收

### DMA 功能与 CRC

当使能的 SPI 通信支持 CRC 通信和 DMA 模式时，在通信结束时会自动发送和接收 CRC，无需使用 CRCNEXT 位。接收 CRC 后，必须在 SPI_DR 寄存器中读取 CRC，以将 RXNE 标志清零。

如果传输过程中出现损坏，则在数据和 CRC 传输结束时，SPI_SR 寄存器中的 CRCERR 标志将置 1。

### 22.3.10. 错误标志

**主模式故障 (MODF)**

当主器件的 NSS 引脚拉低（NSS 硬件模式下）或 SSI 位为 0（NSS 软件模式下）时，会发生主模式故障，这会自动将 MODF 位置 1。主模式故障会在以下几方面影响 SPI 外设：

- 如果 ERRIE 位置 1，MODF 位将置 1，并生成 SPI 中断。
- SPE 位清零。这将关闭器件的所有输出，并关闭 SPI 接口。

MSTR 位清零，从而强制器件进入从模式。

使用以下软件序列将 MODF 位清零：

- 在 MODF 位置 1 时，对 SPI_SR 寄存器执行读或写访问。
- 然后，对 SPI_CR1 寄存器执行写操作。

为避免包含多个 MCU 的系统中发生多从模式冲突，必须在 MODF 位清零序列期间将 NSS 引脚拉高。在该清零序列后，可以将 SPE 和 MSTR 位恢复到原始状态。

作为安全措施，硬件不允许在 MODF 位置 1 时将 SPE 和 MSTR 位置 1。

在从器件中，不能将 MODF 位置 1。但是在多主模式配置中，器件可在 MODF 位置 1 时处于从模式。在这种情况下，MODF 位指示系统控制可能存在多主模式冲突。可使用中断程序从此状态完全恢复，方法是执行复位或返回到默认状态。

**溢出错误**

当主器件发送完数据字节，而从器件尚未将上一个收到的数据所产生的 RXNE 位清零时，将出现溢出情况。出现溢出错误时：

OVR 位置 1 并在 ERRIE 位置 1 时生成一个中断。

在这种情况下，接收器缓冲区内容不会被来自主器件的新数据更新。读取 SPI_DR 寄存器将返回此字节。主器件后续发送的所有其它字节均将丢失。

依次读取 SPI_DR 寄存器和 SPI_SR 寄存器可将 OVR 清除。

**CRC 错误**

当 SPI_CR1 寄存器中的 CRCEN 位置 1 时，此标志用于验证接收数据的有效性。如果移位寄存器中接收的值与 SPI_RXCRCR 的值不匹配，SPI_SR 寄存器中的 CRCERR 标志 将置 1。

**TI 模式帧格式错误**

如果 SPI 在从模式下工作，并配置为符合 TI 模式协议，则在持续通信期间出现 NSS 脉冲时，将检测到 TI 模式帧格式错误。出现此错误时，SPI_SR 寄存器中的 FRE 标志将置 1。发生错误时不会关闭 SPI，但会忽略 NSS 脉冲，并且 SPI 会等待至下一个 NSS 脉冲，然后再开始新的传输。由于错误检测可能导致丢失两个数据字节，因此数据可能会损坏。

读取 SPI_SR 寄存器时，将清零 FRE 标志。如果 ERRIE 位置 1，则检测到帧格式错误时将产生中断。在这种情况下， 由于无法保证数据的连续性，应关闭 SPI，并在重新使能从 SPI 后，由主器件重新发起通信。

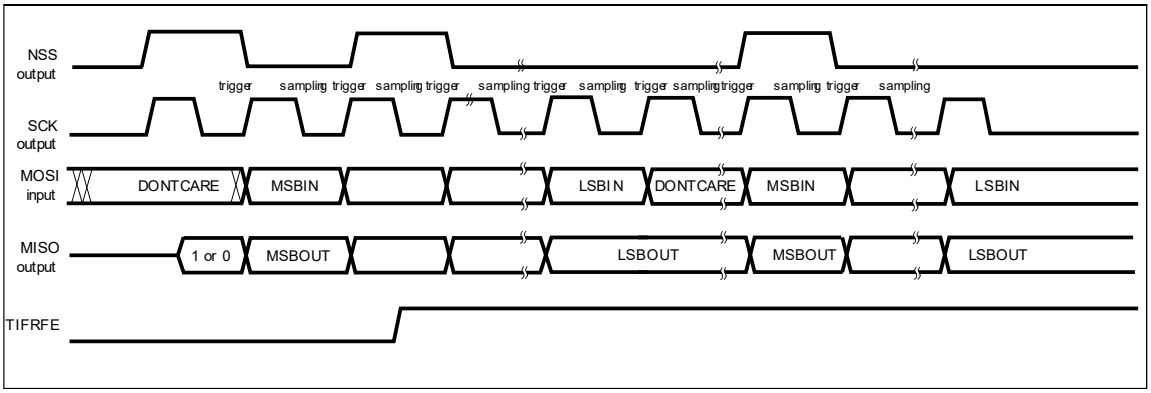



图 22-15 TI 模式帧格式错误检测

### 22.3.11. SPI 中断

表 22-1 SPI 中断请求

| 中断事件 | 事件标志 | 使能控制位 |
|---|---|---|
| 发送缓冲区为空 | TXE | TXEIE |
| 接收缓冲区非空 | RXNE | RXNEIE |
| 主模式故障 | MODF | ERRIE |
| 溢出错误 | OVR | |
| CRC 错误 | CRCERR | |
| TI 帧格式错误 | FRE | ERRIE |
| 发送 FIFO 满 | TFF | TFIE |
| 接受 FIFO 满 | RFF | RFIE |
| 发送 FIFO 将空 | TFE | TFEIE |
| 接收 FIFO 将满 | RFP | RFPIE |

# 22.4. I2S 功能说明

## 22.4.1. 一般说明

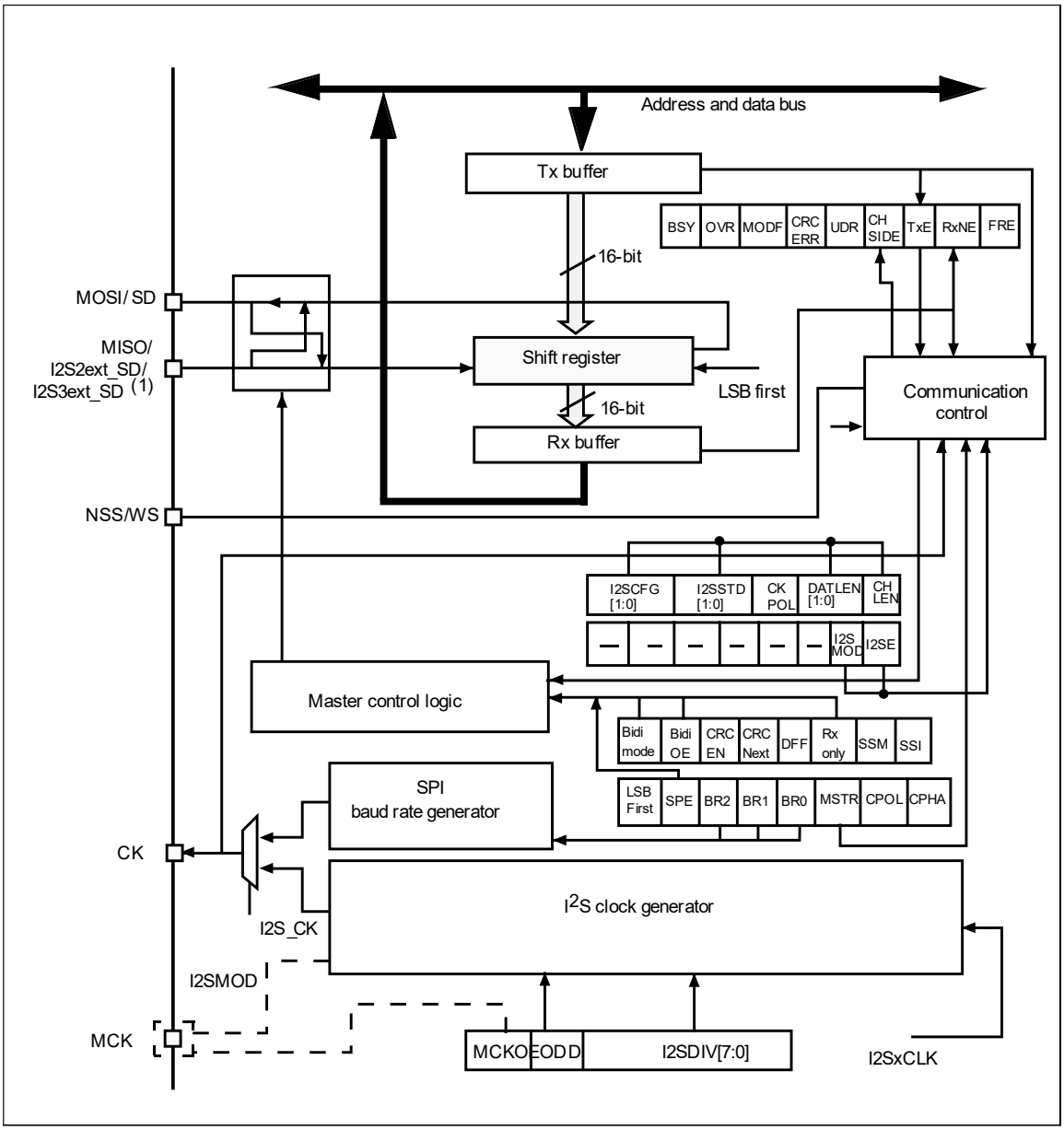



图 22-16 I²S 框图

1. I2S2ext_SD 和 I2S3ext_SD 为扩展 SD 引脚，用于控制 I2S 全双工模式。

当使能 I²S 功能（将 SPI_I2SCFGR 寄存器中的 I2SMOD 位置 1）时， SPI 可用作音频 I²S 接口。此接口使用几乎与 SPI 相同的引脚、标志和中断。

I²S 与 SPI 共用以下三个引脚：

- SD：串行数据（映射到 MOSI 引脚），用于发送或接收两个时分复用的数据通道上的数据（仅半双工模式）。
- WS：字选择（映射到 NSS 引脚），是主模式下的数据控制信号输出以及从模式下的数据控制信号输

入。

- CK：串行时钟（映射到 SCK 引脚），是主模式下的串行时钟输出以及从模式下的串行时钟输入。
- I2S2ext_SD 和 I2S3ext_SD：用于控制 I²S 全双工模式的附加引脚（映射到 MISO 引脚）。当某些外部音频设备需要使用主时钟输出时，可以使用附加引脚：
- MCK：当 I²S 配置为主模式（并且 SPI_I2SPR 寄存器中的 MCKOE 位置 1）时，使用主时钟（单独映射）输出此附加时钟，该时钟输出频率 256 × FS，其中 FS 为音频信号采样频率。

I²S 在主模式下使用自身的时钟发生器生成通信时钟。此时钟发生器也是主时钟输出的源。在 I²S 模式下可以使用两个额外的寄存器。一个是时钟发生器配置寄存器 SPI_I2SPR，另一个是通用 I²S 配置寄存器 SPI_I2SCFGR（音频标准、从/主模式、数据格式、数据包帧、时钟极性等）。

在 I²S 模式下不使用 SPI_CR1 寄存器和所有 CRC 寄存器。同样，也不使用 SPI_CR2 寄存器中的 SSOE 位以及 SPI_SR 中的 MODF 和 CRCERR 位。

I²S 使用和 SPI 相同的寄存器 (SPI_DR) 进行 16 位数据传输。

### 22.4.2. I2S 全双工

为支持 I2S 全双工模式, 除了 I2S2 和 I2S3, 还可以使用两个额外的 I²S, 它们称为扩展 I2S（I2S2_ext、I2S3_ext）。因此，第一个 I2S 全双工接口基于 I2S2 和 I2S2_ext，第二个基于 I2S3 和 I2S3_ext。

*注意：I2S2_ext 和 I2S3_ext 仅用于全双工模式。*

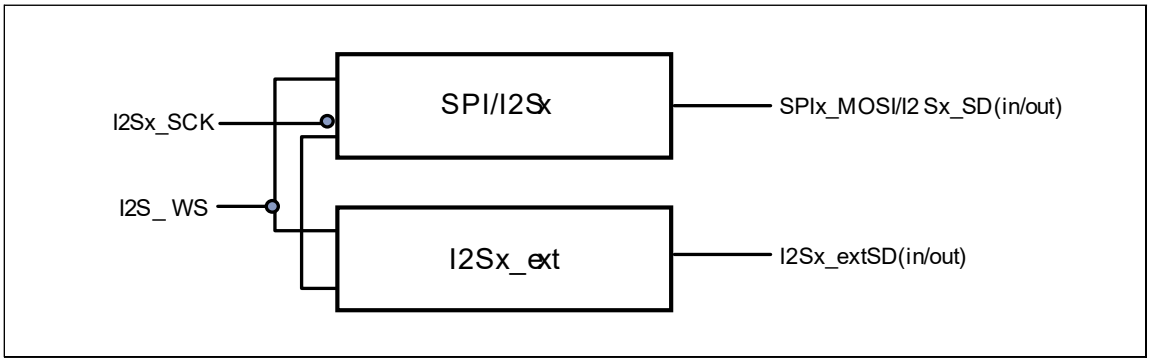



图 22-17 I²S 全双工框图

1. 其中 X 可以是 2 或 3。

I2Sx 可以在主模式下工作。因此：

- 只有 I2Sx 可在半双工模式下输出 SCK 和 WS。
- 只有 I2Sx 可在全双工模式下向 I2S2_ext 和 I2S3_ext 提供 SCK 和 WS。扩展 I2S (I2Sx_ext) 只能用于全双工模式。I2Sx_ext 始终在从模式下工作。
- I2Sx 和 I2Sx_ext 均可用于发送和接收。

### 22.4.3. 支持的音频协议

四线总线通常需要处理时分复用的两个通道（右通道和左通道）上的音频数据。但是，只有 一个 16 位寄存器进行发送和接收。所以，需由软件来把和通道对应的数据写入数据寄存 器，以及从数据寄存器中读取数据，并通过检查 SPI_SR 寄存器中的 CHSIDE 位来识别对应的通道。始终先发送“左通道数据”，而后再发送右通道数据（对于 PCM 协议来说，CHSIDE 没有意义）。

有四种数据和帧格式组合，可采用下列格式发送数据：

- 将 16 位数据封装在 16 位帧中；

- 将 16 位数据封装在 32 位帧中;
- 将 24 位数据封装在 32 位帧中;
- 将 32 位数据封装在 32 位帧中。

当使用 32 位数据包中的 16 位数据时，前 16 位 (MSB) 为有效位，16 位 LSB 被强制清零，无需任何软件操作或 DMA 请求（只需一个读/写操作）。

如果应用程序首选 DMA，则 24 位和 32 位数据帧需要对 SPI_DR 执行两次 CPU 读取或写入操作，或者需要两次 DMA 操作。24 位的数据帧，硬件会将 8 位非有效位扩展到带有 0 位的 32 位。

对于所有数据格式和通信标准而言，始终会先发送最高有效位（MSB 优先）。

I²S 接口支持四种音频标准，可使用 SPI_I2SCFGR 寄存器中的 I2SSTD[1:0] 和 PCMSYNC 位对其进行配置。

### I2S Philips 标准

使用 WS 信号来指示当前正在发送的数据所属的通道。该信号从当前通道数据的第一个位 (MSB) 之前的一个时钟开始有效。

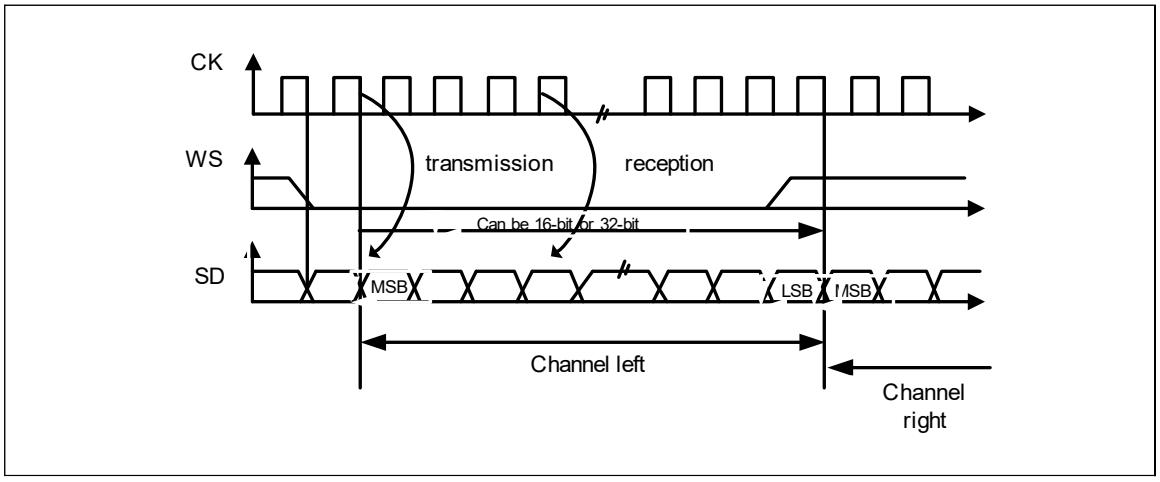



图 22-18 I²S Philips 协议波形(16/32 位全精度，CPOL = 0)

发送方在时钟信号 (CK) 的下降沿改变数据，接收方在上升沿读取数据。 WS 信号也在 CK 的 下降沿变化。

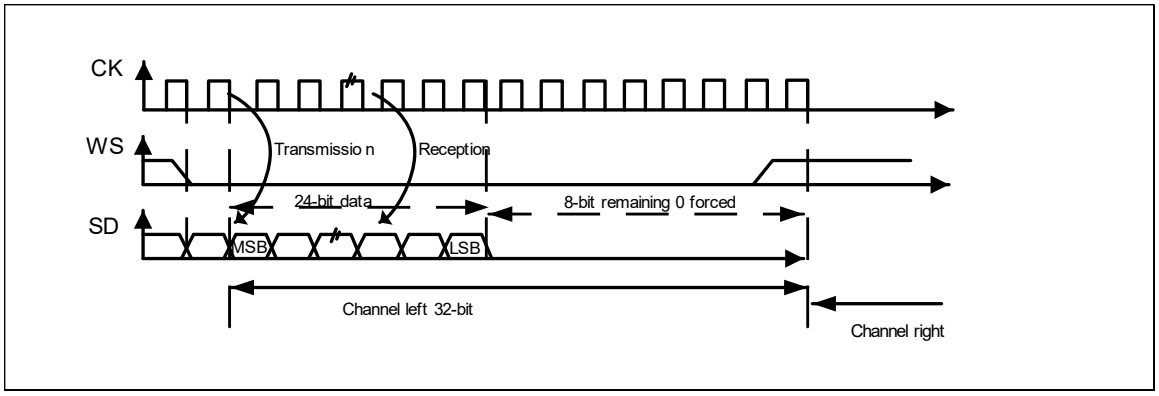



图 22-19 I²S Philips 标准波形(24 位帧，CPOL = 0)

该模式需要对 SPI_DR 执行两次写入或读取操作。

- 在发送模式下：
- 如果需要发送 0x8EAA33（24 位）：

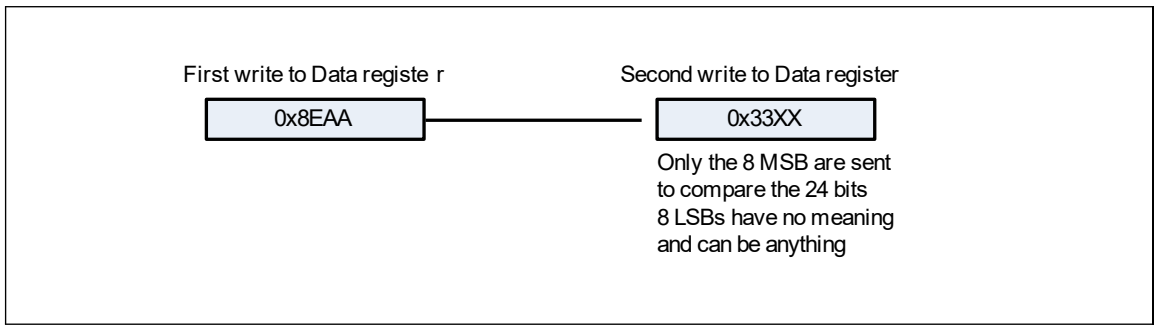



图 22-20 发送 0x8EAA33

- 在接收模式下：
- 如果接收数据 0x8EAA33：

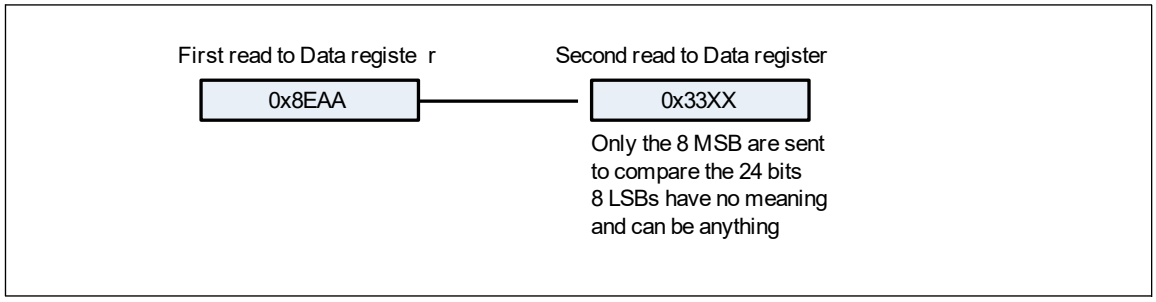



图 22-21 接收 0x8EAA33

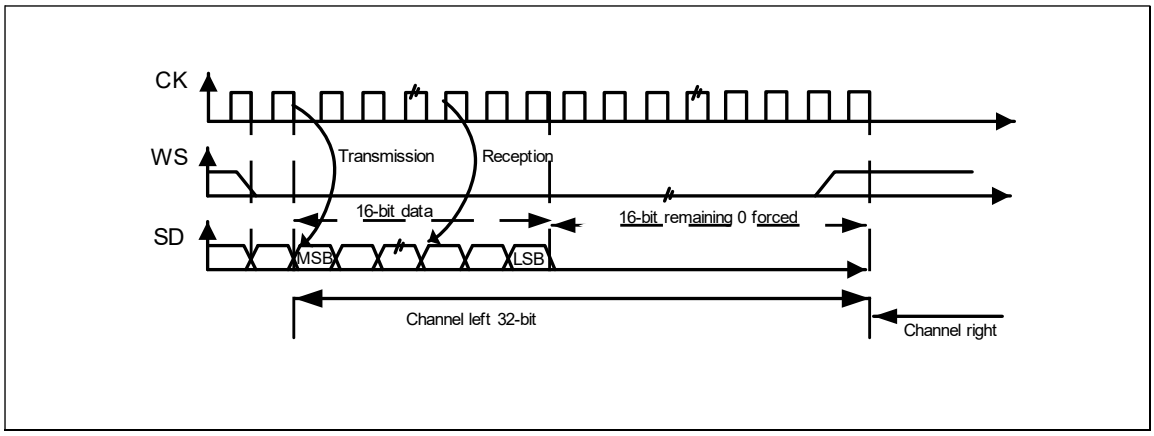



图 22-22 I²S Philips 标准(16 位扩展为 32 位数据包帧，CPOL = 0)

如果 I²S 配置成将 16 位数据扩展到 32 位帧，只需要访问一次寄存器 SPI_DR。硬件会将其余 16 个位强制设置为 0x0000，以便将数据扩展为 32 位格式。

如果要发送的数据或已接收的数据为 0x76A3（0x76A30000 扩展为 32 位），则需要执行示例中显示的操作。

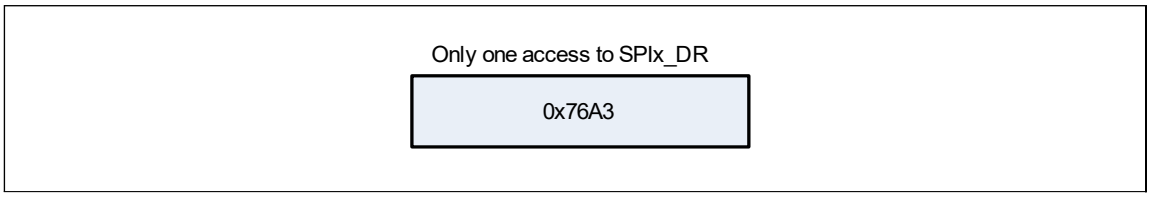



图 22-23 示例

发送时，每次将 16 位数据写入 SPI_DR，硬件就会置位 TXE 标志，并在中断使能的情况下触发中断，以把下一个要发送的数据加载到 SPI_DR。即使硬件填充的低 16 位 0x0000 还未发送，也会如此，因为低 16 位是由硬件发送。

接收时，接收到第一个半字（高 16 位），则硬件将 RXNE 标志置 1，并在中断使能的情况下触发中断。

这样，就延长了两个写入或读取操作之间的时间间隔，从而可防止出现下溢或上溢情况（具体取决于数据传输方向）。

### MSB 对齐标准

此标准同时生成 WS 信号和第一个数据位（即 MSBit）。

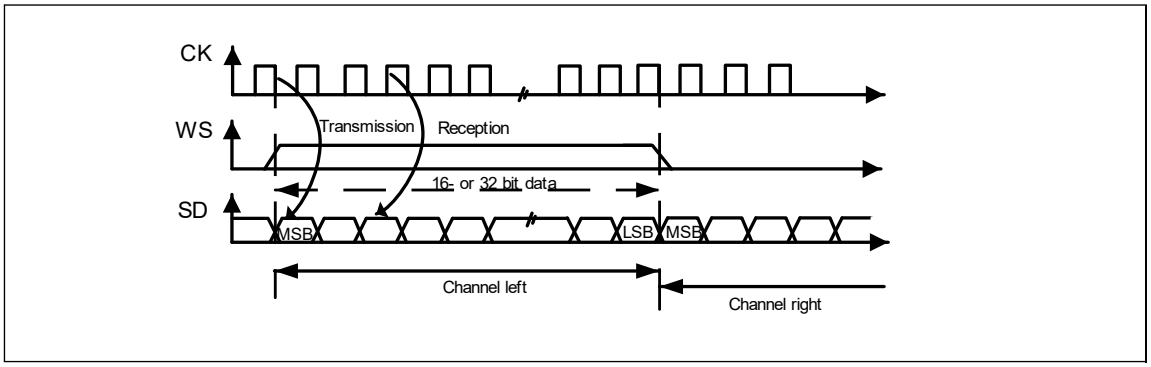



图 22-24 MSB 对齐的 16 位或 32 位全精度长度，CPOL = 0

发送方在时钟信号的下降沿改变数据；接收方在上升沿读取数据。

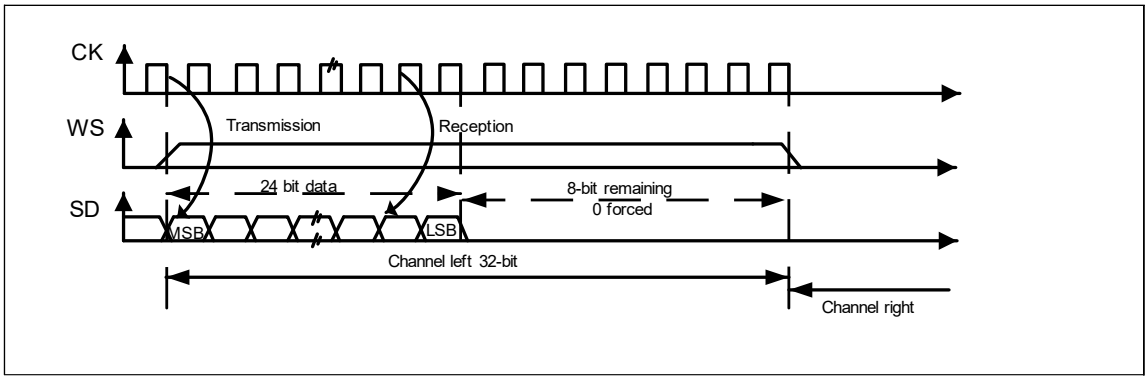



图 22-25 MSB 对齐的 24 位帧长度，CPOL = 0

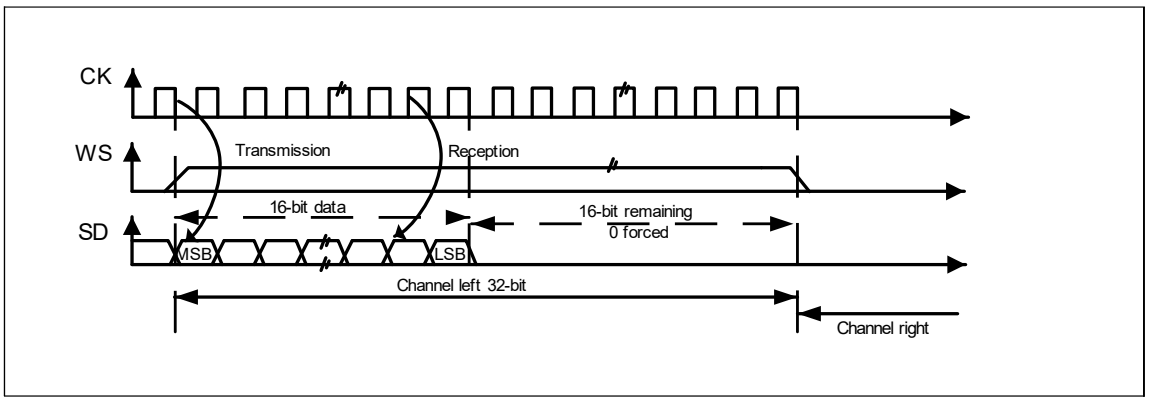



图 22-26 扩展为 32 位数据包帧的 MSB 对齐的 16 位，CPOL = 0

### LSB 对齐标准

该标准与 MSB 对齐标准类似（对于 16 位和 32 位全精度帧格式，没有任何不同）。

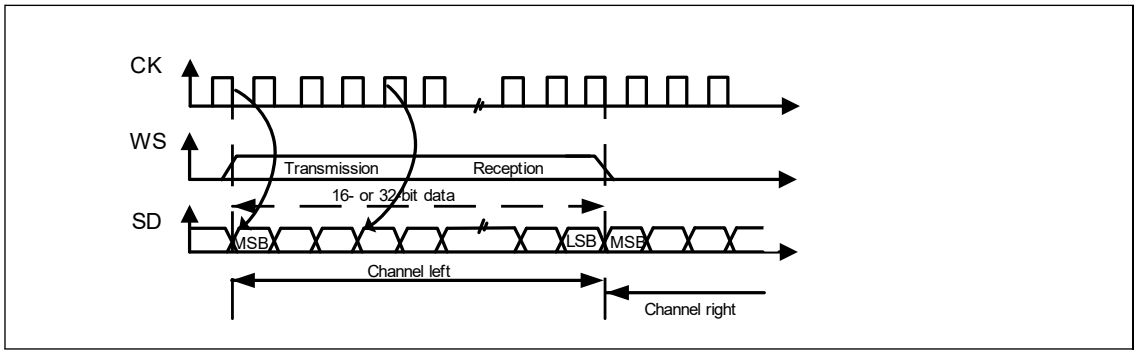



图 22-27 LSB 对齐的 16 位或 32 位全精度，CPOL = 0

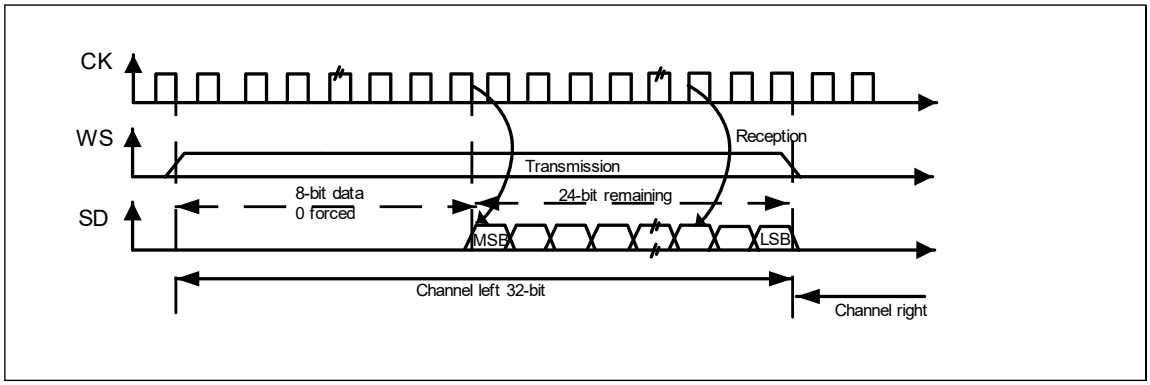



图 22-28 LSB 对齐的 24 位帧长度，CPOL = 0

- 在发送模式下：
- 如果需要发送数据 0x3478AE，则需要通过软件或 DMA 对 SPI_DR 寄存器执行两次写入操作。操作如下所示。

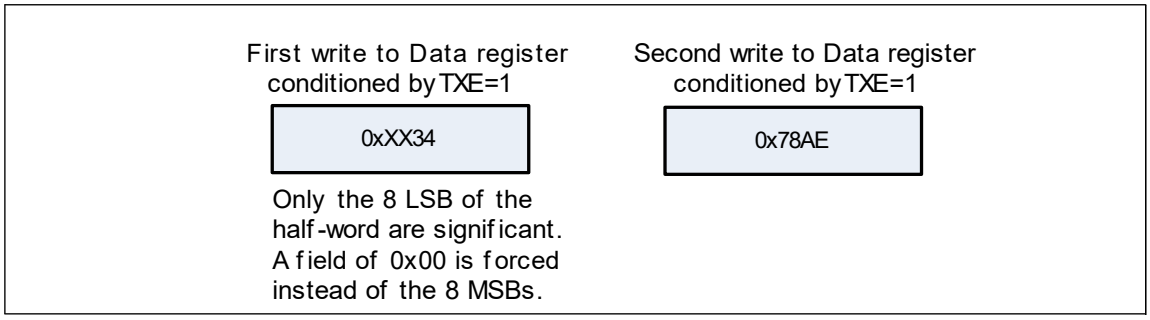



图 22-29 发送 0x3478AE 所需的操作

- 在接收模式下:
- 如果接收到数据 0x3478AE，则在每个 RXNE 事件时需要对 SPI_DR 执行两次连续的读取操作。

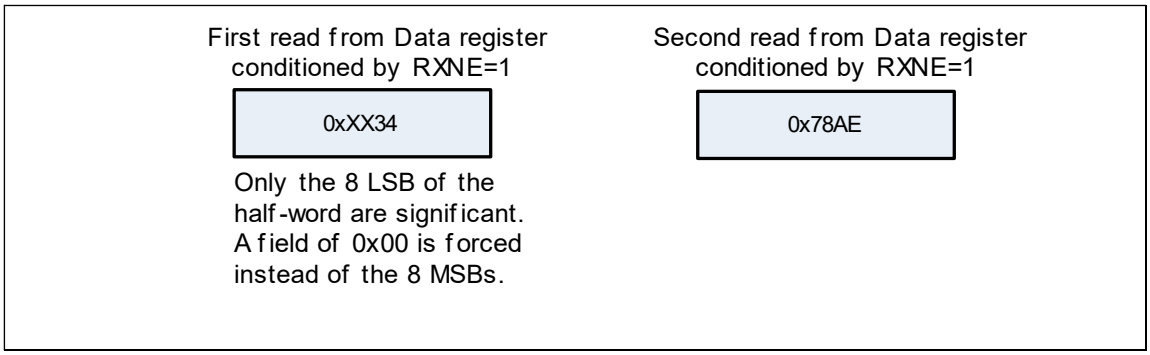



图 22-30 接收 0x3478AE 所需的操作

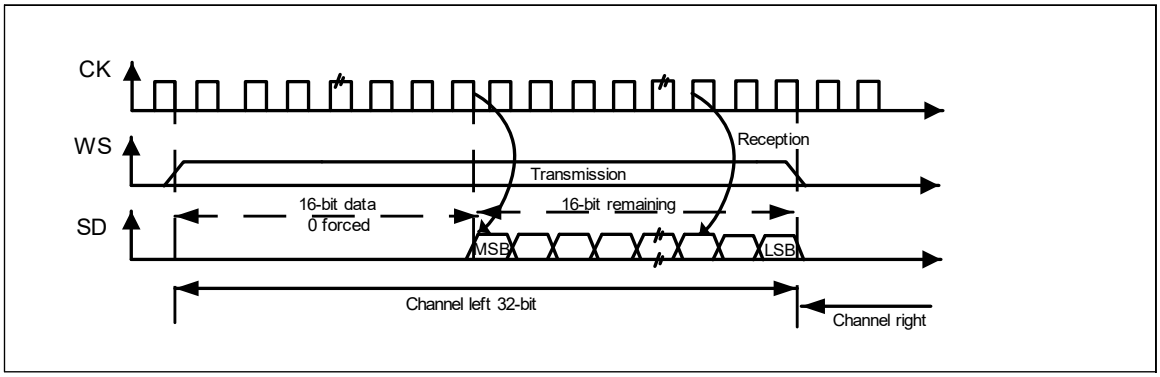



图 22-31 扩展为 32 位数据包帧的 LSB 对齐的 16 位，CPOL = 0

如果 I²S 配置成将 16 位数据扩展到 32 位帧，只需要访问一次寄存器 SPI_DR。硬件会将其余 16 个位强制设置为 0x0000，以便将数据扩展为 32 位格式。在这种情况下，其对应于半字 MSB。

如果要发送的数据或已接收的数据为 0x76A3（0x0000 76A3 扩展为 32 位），则需要执行下图中显示的操作。

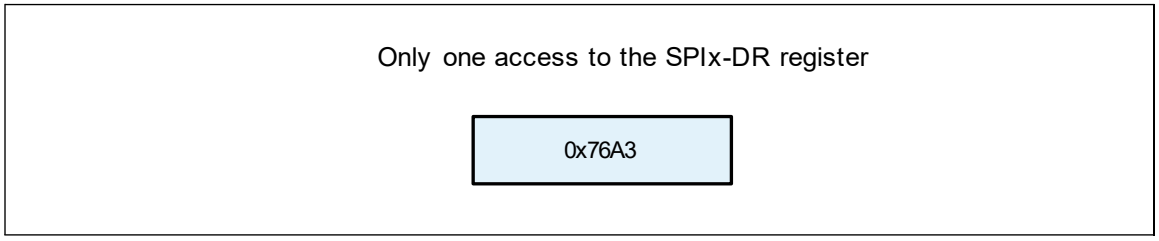



图 22-32 扩展为 32 位数据包帧的 LSB 对齐的 16 位示例

在发送模式下，TXE 置位时，应用程序需要写入要发送的数据（此例中，为 0x76A3）。首先发送 0x000 字段（扩展到 32 位）。有效数据 (0x76A3) 发送完毕后，TXE 会被再次置位。

在接收模式下，当接收到有效半字后（而非 0x0000 字段），即会置位 RXNE。

这样，就延长了两个写入或读取操作之间的时间间隔，以防止出现下溢或上溢情况。

### PCM 标准

对于 PCM 标准，无需使用通道信息。有两种 PCM 模式（短帧和长帧），并且可使用 SPI_I2SCFGR 中的 PCMSYNC 位来配置。

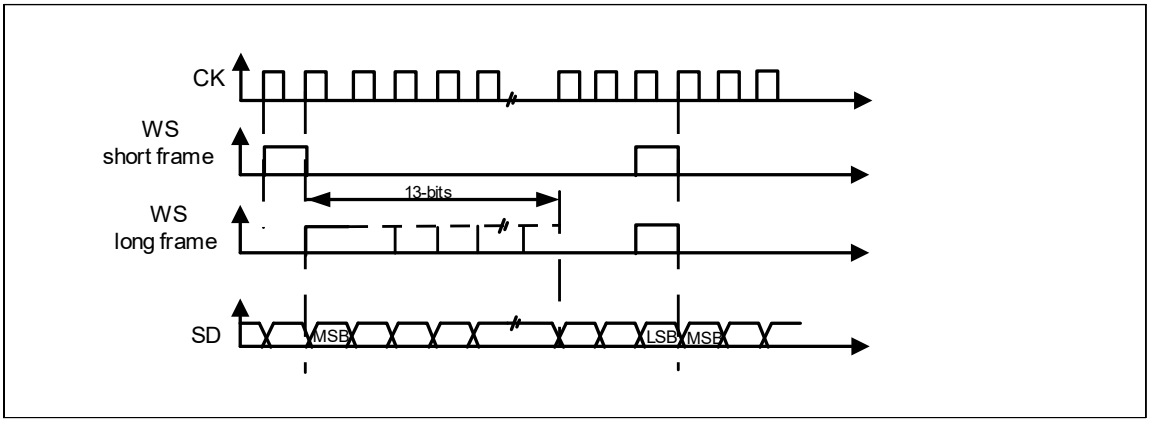



图 22-33 PCM 标准波形(16 位)

对于长帧同步，在主模式下会将 WS 信号持续 13 个周期。

对于短帧同步，WS 同步信号的持续时间仅为一个周期。

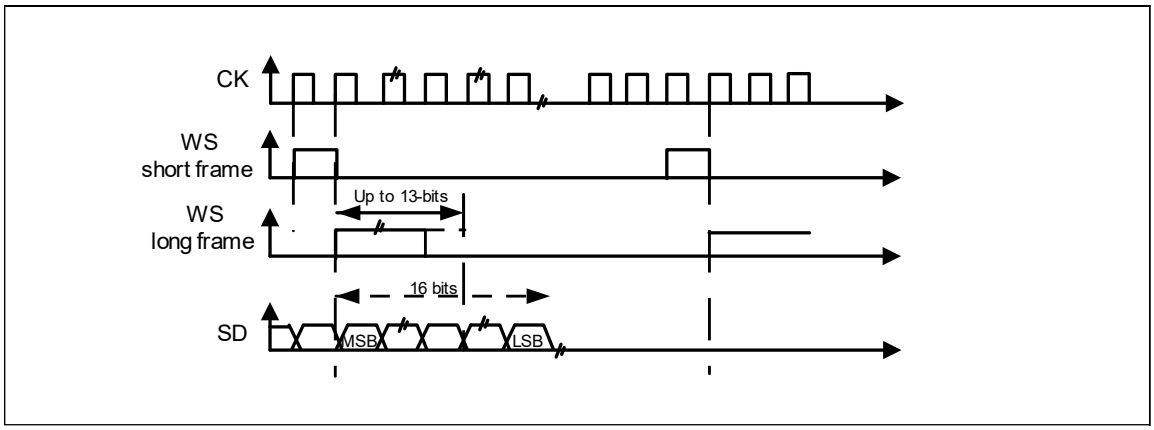



图 22-34 PCM 标准波形(16 位扩展到 32 位数据包帧)

*注意：对于两种模式(主/从模式)和两种同步(短/长同步)，即使在从模式下，也需要指定两组连续数据(以及两个同步信号)之间位的个数(SPI_I2SCFGR 寄存器中的 DATLEN 位和 CHLEN 位)。*

### 22.4.4. 时钟发生器

I²S 比特率用来确定 I²S 数据线上的数据流和 I²S 时钟信号频率。

I²S 比特率 = 每个通道的位数 × 通道数 × 音频采样频率

对于 16 位双通道音频，I²S 比特率的计算公式如下：

I²S 比特率 = 16 × 2 × $F_S$

如果数据包为 32 位宽，则 I²S 比特率 = 32 x 2 x $F_S$。

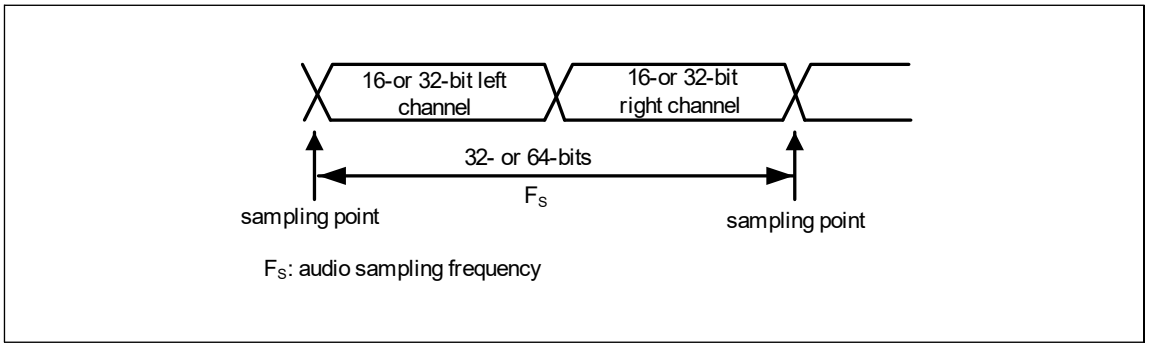



图 22-35 音频采样频率定义

配置主模式时，需要正确地对线性分频器进行设置，以便采用所需的音频频率进行通信。

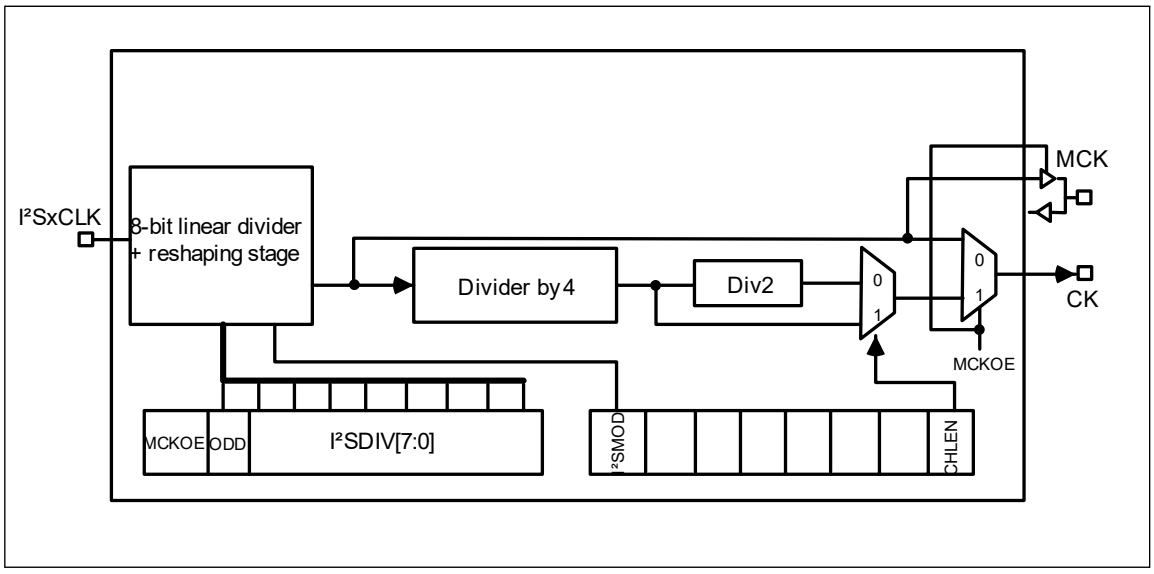



图 22-36 I²S 时钟发生器架构

1. 其中，X 可以是 2 或 3。

要实现高品质的音频性能，I2SxCLK 时钟源可为 PLLI2S 输出或外部时钟（映射到 I2S_CKIN 引脚）。

音频采样频率可能是 192 kHz 、96 kHz 或 48 kHz。为达到所需频率，需要根据以下公式对线性分频器进行编程：

输出主时钟（SPI_I2SPR 寄存器中的 MCKOE 置 1)时：

$F_S$ = I2SxCLK / [(16*2)*((2*I2SDIV)+ODD)*8)]（通道帧宽度为 16 位时）

$F_S$ = I2SxCLK / [(32*2)*((2*I2SDIV)+ODD)*4)]（通道帧宽度为 32 位时）

关闭主时钟输出（MCKOE 位清零）时：

$F_S$ = I2SxCLK / [(16*2)*((2*I2SDIV)+ODD))]（通道帧宽度为 16 位时）

$F_S$ = I2SxCLK / [(32*2)*((2*I2SDIV)+ODD))]（通道帧宽度为 32 位时）

### 22.4.5. I2S 主模式

$I^2S$ 可如下配置：

- 发送主器件或接收主器件（使用 I2Sx 的半双工模式）
- 同时收发的主器件（使用 I2Sx 和 I2Sx_ext 的全双工模式）。

$I^2S$ 工作在主模式， 串行时钟由引脚 CK 输出，字选信号由引脚 WS 产生。可以通过设置寄存器 SPI_I2SPR 的 MCKOE 位来选择输出或者不输出主时钟 (MCK)。

**步骤**

- 设置 SPI_I2SPR 寄存器的 I2SDIV[7:0] 位，以定义串行时钟波特率，从而达到适当的音频采样频率。SPI_I2SPR 寄存器的 ODD 位也需要设置。
- 设置 CKPOL 位，定义时钟在空闲时的电平状态。如果需要为外部 DAC/ADC 音频组件提供主时钟 MCK，则将 SPI_I2SPR 寄存器的 MCKOE 位置 1（I2SDIV 和 ODD 值应根据 MCK 输出的状态进行计算。有关详细信息，请参见时钟发生器）。
- 将 SPI_I2SCFGR 中的 I2SMOD 位置 1 以激活 $I^2S$ 功能，通过 I2SSTD[1:0] 和 PCMSYNC 位选择 $I^2S$ 标准，通过 DATLEN[1:0] 位选择数据长度并通过配置 CHLEN 位选择每个通道的位数。此外，通过 SPI_I2SCFGR 寄存器的 I2SCFG[1:0] 位选择 I2S 主模式和方向（发送器或接收器）。
- 如果需要，通过对 SPI_CR2 寄存器执行写操作来选择所有可能的中断源和 DMA 功能。
- SPI_I2SCFGR 寄存器的 I2SE 位必须置 1。

WS 和 CK 配置为输出模式。如果 SPI_I2SPR 的 MCKOE 位置 1，则 MCK 也是输出。

**发送序列**

将半字写入发送缓冲区后，发送序列随即开始。

通常，写入发送缓冲区的第一个数据对应于左通道数据。数据从发送缓冲区传输到移位寄存器时，TXE 置 1，并且必须将对应于右通道的数据写入发送缓冲区。CHSIDE 标志指示将发送的数据对应的通道。TXE 标志置 1 时，CHSIDE 标志有意义，因为该标志在 TXE 变为高电平时进行更新。

一个完整帧表示先进行左通道数据发送再进行右通道数据发送。不存在仅发送左通道的部分帧。

首位发送期间，数据按半字并行加载到 16 位移位寄存器中，然后以串行方式移位并输出到 MOSI/SD 引脚（MSB 在前）。每次数据从发送缓冲区传输到移位寄存器后，TXE 标志都将置 1，如果 SPI_CR2 寄存器的 TXEIE 位置 1，将产生中断。

有关各种 I2S 标准模式中的写操作的更多详细信息，请参见支持的音频协议。

为确保连续进行音频数据发送，必须在当前数据发送结束前将下一个要发送数据写入 SPI_DR。

要通过将 I2SE 清零来关闭 $I^2S$，必须等待 TXE 置 1 且 BSY 清零。

**接收序列**

此工作模式与发送模式相同，只有第 3 点存在不同（请参见上面所述的步骤），即通过 I2SCFG[1:0] 位设置主器件接收模式。

无论数据或通道长度如何，音频数据始终按 16 位数据包进行接收。这意味着，每当接收缓冲区满时，RXNE 标志即置 1，并且如果 SPI_CR2 寄存器的 RXNEIE 位置 1，还将产生中断。所接收的右通道或左通道的

音频值可通过一次或两次接收操作进入接收缓冲区，具体取决于数据和通道长度配置。

读取 SPI_DR 寄存器即会使 RXNE 位清零。

CHSIDE 在每次接收后进行更新。它由 I²S 单元所产生的 WS 信号触发。

如果在先前收到的数据尚未读取时又接收到新数据，将产生溢出错误并将 OVR 标志置 1。如果 SPI_CR2 寄存器的 ERRIE 位置 1，将产生中断以指示该错误。

要关闭 I²S，需要执行特定操作来确保 I2S 正确完成传输周期而不启动新的数据传输。该序列取决于数据和通道长度的配置，以及所选的音频协议模式。在以下情况下：

- 32 位通道长度上扩展的 16 位数据长度（DATLEN = 00 且 CHLEN = 1），使用 LSB 对齐模式 (I2SSTD = 10)
  - 等待倒数第二个 RXNE = 1 (n - 1)
  - 然后等待 17 个 I²S 时钟周期（使用软件循环）
  - 关闭 I²S (I2SE = 0)
- 32 位通道长度上扩展的 16 位数据长度（DATLEN = 00 且 CHLEN = 1），使用 MSB 对齐、I²S 或 PCM 模式（即 I2SSTD = 00 、I2SSTD = 01 或 I2SSTD = 11）
  - 等待最后一个 RXNE
  - 然后等待 1 个 I²S 时钟周期（使用软件循环）
  - 关闭 I²S (I2SE = 0)
- 对于 DATLEN 和 CHLEN 的所有其它组合，无论通过 I2SSTD 位选择何种音频模式，都将执行以下序列来关闭 I²S：
  - 等待倒数第二个 RXNE = 1 (n — 1)
  - 然后等待 1 个 I²S 时钟周期（使用软件循环）
  - 关闭 I²S (I2SE = 0)

*注意： 传输期间， BSY 标志保持低电平。*

### 22.4.6. I2S 从模式

I²S 可如下配置：

- 发送从器件或接收从器件（使用 I2Sx 的半双工模式）；
- 同时收发的从器件（使用 I2Sx 和 I2Sx_ext 的全双工模式）。

此工作模式所遵循的规则与 I²S 主模式基本相同。在从模式下，I²S 接口不产生时钟。时钟和 WS 信号从 I²S 接口所连接的外部主器件输入。这样用户便不需要配置时钟。

应遵循如下配置步骤：

- 将 SPI_I2SCFGR 寄存器的 I2SMOD 位置 1 以激活 I²S 功能，通过 I2SSTD[1:0] 位选择 I²S 标准，通过 DATLEN[1:0] 位选择数据长度并通过 CHLEN 位选择帧中每个通道的位数。此外，通过 SPI_I2SCFGR 寄存器的 I2SCFG[1:0] 位选择从器件的模式（发送或接收）。
- 如果需要，通过对 SPI_CR2 寄存器执行写操作来选择所有可能的中断源和 DMA 功能。
- SPI_I2SCFGR 寄存器的 I2SE 位必须置 1。

**发送序列**

当外部主器件发送时钟并且通过 NSS_WS 信号请求传输数据时，发送序列开始。必须首先使能从器件，然后外部主器件才能开始通信。主器件开始通信前，从器件还必须加载 I2S 数据寄存器。

对于 I²S、MSB 对齐和 LSB 对齐模式，要写入数据寄存器的第一个数据项对应于左通道的数据。通信开始时，数据从发送缓冲区传输到移位寄存器。TXE 标志随即置 1，以请求将右通道的数据写入 I²S 数据寄存

器。

CHSIDE 标志指示将发送的数据对应的通道。与主发送模式相比，在从模式下，CHSIDE 由来自外部主器件的 WS 信号触发。这意味着，从器件需要首先为发送第一个数据做好准备， 然后主器件才能产生时钟。WS 置位意味着首先发送左通道数据。

*注意： 必须要在主器件发出的第一个时钟出现在 CK 线上至少 2 个 PCLK 周期之前置位 I2SE。*

*首位发送期间，数据按半字从内部总线并行加载到 16 位移位寄存器中，然后以串行方式移位并输出到 MOSI/SD 引脚（MSB 在前）。每次数据从发送缓冲区传输到移位寄存器后，TXE 标志都将置 1，如果 SPI_CR2 寄存器的 TXEIE 位置 1，将产生中断。*

*请注意，仅当 TXE 标志为 1 时，才可以尝试向发送缓冲区写入数据。*

有关各种 I²S 标准模式中的写操作的更多详细信息，请参见支持的音频协议。

为确保连续进行音频数据发送，必须在当前数据发送结束前将下一个要发送数据写入 SPI_DR 寄存器。如果在数据尚未写入 SPI_DR 寄存器时下一个数据通信的首个时钟边沿到来，下溢标志将置 1 并可能产生中断。通过这种方式，软件可以获知所传输的数据不正确。 如果 SPI_CR2 寄存器的 ERRIE 位置 1，则当 SPI_SR 寄存器中的 UDR 标志变为 1 时，将产生中断。这种情况下，必须关闭 I²S 并从左通道开始重新启动数据传输。

要通过将 I2SE 位清零来关闭 I²S，必须等待 TXE 置 1 且 BSY 清零。

**接收序列**

此工作模式与发送模式相同，只有第 1 点存在不同（请参见上面所述的步骤），即通过 SPI_I2SCFGR 寄存器的 I2SCFG[1:0] 位设置从器件接收模式。

无论数据长度或通道长度如何，音频数据始终按 16 位数据包进行接收。这意味着，每当接收缓冲区填满时，SPI_SR 寄存器中的 RXNE 标志即置 1，并且如果 SPI_CR2 寄存器的 RXNEIE 位置 1，还将产生中断。所接收的右通道或左通道的音频值可能通过一次或两次接收操作进入接收缓冲区，具体取决于数据长度和通道长度配置。

每次接收要从 SPI_DR 读取的数据时， CHSIDE 标志都将更新。该标志由外部主器件所管理的外部 WS 线路触发。

读取 SPI_DR 寄存器即会使 RXNE 位清零。

如果在先前收到的数据尚未读取时又接收到新数据，将产生溢出错误并将 OVR 标志置 1。 如果 SPI_CR2 寄存器的 ERRIE 位置 1，将产生中断以指示该错误。

要在接收模式下关闭 I²S，必须在接收到最后一个 RXNE = 1 后立即将 I2SE 清零。

*注意： 外部主器件应能够通过音频通道以 16 位或 32 位数据包发送/接收数据。*

### 22.4.7. 状态标志

应用程序可通过三个状态标志来全面监视 I²S 总线的状态。

**忙标志 (BSY)**

BSY 标志由硬件置 1 和清零（写入此标志没有任何作用）。该标志表示 I²S 通信的状态。

硬件将 BSY 置 1 时，指示 I²S 正在忙于通信。在主接收模式 (I2SCFG = 11) 中，BSY 标志的情况例外，该标志在接收期间仍保持低电平。

如果软件需要关闭 I²S，可使用 BSY 标志检测传输是否结束。以避免破坏最后一个数据的传输。为此，必须严格遵循下述步骤。

在传输开始时（I²S 处于主接收器模式时除外）硬件将 BSY 标志置 1。

出现以下情况时， BSY 标志被硬件清零：

- 传输完成（主发送模式除外，在该模式下通信是连续的）；
- 关闭 I²S 时。

当通信连续时：

- 在主发送模式下，BSY 标志在所有传输期间均保持高电平；
- 在从模式下，BSY 标志在每次传输之间变为低电平并持续一个 I²S 时钟周期。

*注意：　请勿使用 BSY 标志处理每次数据发送或接收，最好改用 TXE 标志和 RXNE 标志。*

**发送缓冲区为空 (TXE)**

如果此标志置 1，表示发送缓冲区为空，可将要发送的下一个数据加载到其中。发送缓冲区已包含要发送的数据时， TXE 标志复位。关闭 I²S（I2SE 位复位）时，该标志也会复位。

**接收缓冲区非空 (RXNE)**

此标志置 1 时，表示接收缓冲区中存在有效的已接收数据。读取 SPI_DR 寄存器时，该标志复位。

**通道方向 (CHSIDE)**

在发送模式下，此标志将在 TXE 变为高电平时进行刷新。此标志指示 SD 上要传输的数据所属的通道。如果在从发送模式下发生下溢错误事件，此标志将不可靠，在恢复通信前需要关闭并重新开启 I2S。

在接收模式下，此标志将在数据接收到 SPI_DR 中时进行刷新。此标志指示已接收的数据所属的通道。请注意，如果发生错误（例如 OVR），此标志将失去意义，应通过关闭并重新使能 I²S（根据需要更改配置）来将其复位。

此标志在 PCM 标准中没有意义（短帧和长帧模式）。

当 SPI_SR 中的 OVR 或 UDR 标志置 1，并且 SPI_CR2 中的 ERRIE 位也置 1 时，将产生中断。中断源被清除后，可通过读取 SPI_SR 状态寄存器来将此中断清零。

### 22.4.8. 错误标志

I²S 单元共有三个错误标志。

**下溢标志 (UDR)**

在从发送模式下，如果在软件尚未将任何值加载到 SPI_DR 之前出现第一个数据发送时钟，此标志将置 1 。SPI_I2SCFGR 中的 I2SMOD 位置 1 时，可以使用此标志。如果 SPI_CR2 中的 ERRIE 位置 1，可产生中断。

UDR 位通过 SPI_SR 寄存器上的读操作进行清零。

**上溢标志 (OVR)**

如果在尚未从 SPI_DR 读取上一个数据时又接收到新数据，此标志将置 1。因此传入的数据将丢失。如果 SPI_CR2 中的 ERRIE 位置 1，可产生中断。

这种情况下，将不会用接收到的新数据更新接收缓冲区的内容。对 SPI_DR 寄存器执行的读操作将返回先前正确接收的数据。主器件后续发送的所有其它半字都将丢失。

要将 OVR 位清零，应首先对 SPI_DR 寄存器执行读操作，然后再对 SPI_SR 寄存器进行读访问。

**帧错误标志 (FRE)**

仅当 I²S 配置为从模式时，此标志才可由硬件置 1。如果外部主器件没有按照从器件期望的那样改变 WS 信号，则此标志将置 1。如果失去同步，要从此状态中恢复并将外部主器件与 I²S 从器件重新同步，应遵循如下步骤：

- 关闭 I²S。
- 在 WS 线上检测到正确的电平时将其重新使能（WS 在 I²S 模式下为高电平，在 MSB 对齐、LSB 对齐或 PCM 模式下为低电平）。

主器件与从器件之间的同步失效可能是由于 SCK 通信时钟或 WS 帧同步信号线上存在噪音干扰。如果 ERRIE 位置 1，可产生错误中断。读取状态寄存器时，同步失效标志 (FRE) 由软件清零。

### 22.4.9. I2S 中断

表 22-2 I²S 中断请求

| 中断事件 | 事件标志 | 使能控制位 |
|---|---|---|
| 发送缓冲区为空 | TXE | TXEIE |
| 接收缓冲区非空 | RXNE | RXNEIE |
| 上溢错误 | OVR | ERRIE |
| 下溢错误 | UDR | |
| 帧错误 | FRE | ERRIE |
| 发送 FIFO 满 | TFF | TFIE |
| 接受 FIFO 满 | RFF | RFIE |
| 发送 FIFO 将空 | TFE | TFEIE |
| 接收 FIFO 将满 | RFP | RFPIE |

### 22.4.10. DMA 特性

DMA 的工作方式与 SPI 模式完全相同。对于 I²S 没有任何区别。只是由于不需要数据传输保护机制，因此 I²S 模式没有 CRC 功能。

## 22.5. SPI 和 I2S 寄存器

外设寄存器可支持半字（16 位）或字（32 位）访问。

### 22.5.1. SPI 控制寄存器 1 (SPI_CR1)（不用于 I2S 模式）

偏移地址： 0x00

复位值： 0x0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BIDIMODE | BIDIOE | CRCEN | CRCNEXT | DFF | RXONLY | SSM | SSI | LSBFIRST | SPE | BR [2:0] | | | MSTR | CPOL | CPHA |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15 | BIDIMODE | 双向通信数据模式使能<br>0：选择双线单向通信数据模式<br>1：选择单线双向通信数据模式<br>*注意： 不适用于I2S 模式* |
| 14 | BIDIOE | 双向通信模式下的输出使能<br>此位结合 BIDIMODE 位，用于选择双向通信模式下的传输方向<br>0：禁止输出（只接收模式）<br>1：使能输出（只发送模式）<br>*注意： 在主模式下使用 MOSI 引脚；在从模式下使用 MISO 引脚。*<br>*不适用于I2S 模式* |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 13 | CRCEN | 硬件 CRC 计算使能<br>0：禁止 CRC 计算<br>1：使能 CRC 计算<br>*注意：为确保正确操作，只应在禁止 SPI(SPE ="0")时对此位执行写操作*<br>*不适用于 I2S 模式* |
| 12 | CRCNEXT | 下一次传输 CRC<br>0：数据阶段（无 CRC 阶段）<br>1：下一次传输为 CRC（CRC 阶段）<br>*注意： 当 SPI 配置为全双工或只发送模式时，只要最后一个数据写入 SPI_DR 寄存器，就必须对 CRCNEXT 执行写操作。*<br>当 SPI 配置为只接收模式时，必须在接收到倒数第二个数据之后将 CRCNEXT 置 1。<br>当传输由 DMA 管理时，此位应保持清零状态。<br>不适用于 I²S 模式 |
| 11 | DFF | 数据帧格式<br>0：为发送/接收选择 8 位数据帧格式<br>1：为发送/接收选择 16 位数据帧格式<br>*注意：为确保正确操作，只应在禁止 SPI(SPE ="0")时对此位执行写操作*<br>*适用于 I2S 模式* |
| 10 | RXONLY | 只接收<br>此位结合 BIDIMODE 位，用于选择双线单向模式下的传输方向。此位也适用于多从模式系统，在此类系统中，不会访问特定从器件，也不会损坏访问的从器件的输出。<br>0：全双工（发送和接收）<br>1：关闭输出（只接收模式）<br>*注意：不适用于 I2S 模式* |
| 9 | SSM | 软件从器件管理<br>当 SSM 位置 1 时，NSS 引脚输入替换为 SSI 位的值。<br>0：禁止软件从器件管理<br>1：使能软件从器件管理<br>*注意：不适用于 I2S 模式和 SPI TI 模式* |
| 8 | SSI | 内部从器件选择<br>仅当 SSM 位置 1 时，此位才有效。此位的值将作用到 NSS 引脚上，并忽略 NSS 引脚的 IO 值。<br>*注意：不适用于 I2S 模式和 SPI TI 模式* |
| 7 | LSBFIRST | 帧格式<br>0：先发送 MSB<br>1：先发送 LSB<br>*注意：正在通信时不应更改此位。*<br>*不适用于 I2S 模式和 SPI TI 模式* |
| 6 | SPE | SPI 使能<br>0：关闭外设<br>1：使能外设<br>*注意：*<br>*1). 不适用于 I2S 模式*<br>*2). 关闭 SPI 时，请按照关闭 SPI 中所述的步骤操作。* |
| 5:3 | BR[2:0] | 波特率控制<br>000: $f_{PCLK}/2$ 100: $f_{PCLK}/32$ |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 001: $f_{PCLK}/4$ 101: $f_{PCLK}/64$<br>010: $f_{PCLK}/8$ 110: $f_{PCLK}/128$<br>011: $f_{PCLK}/16$ 111: $f_{PCLK}/256$<br>*注意： 正在通信时不应更改这些位。*<br>*不适用于 I2S 模式* |
| 2 | MSTR | 主模式选择<br>0：从配置<br>1：主配置<br>*注意： 正在通信时不应更改此位。*<br>*不适用于 I2S 模式* |
| 1 | CPOL | 时钟极性<br>0：空闲状态时，SCK 保持低电平<br>1：空闲状态时，SCK 保持高电平<br>*注意： 正在通信时不应更改此位。*<br>*不适用于 I2S 模式和 SPI TI 模式* |
| 0 | CPHA | 时钟相位<br>0：从第一个时钟边沿开始采样数据<br>1：从第二个时钟边沿开始采样数据<br>*注意： 正在通信时不应更改此位。*<br>*不适用于 I2S 模式和 SPI TI 模式* |

### 22.5.2. SPI 控制寄存器 2 (SPI_CR2)

偏移地址： 0x04

复位值： 0x0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | | TXEIE | RXNEIE | ERRIE | FRF | Res. | SSOE | TXDMAEN | RXDMAEN |
| | | | | | | | | rw | rw | rw | rw | | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 必须保持复位值。 |
| 7 | TXEIE | 发送缓冲区空中断使能<br>0：屏蔽 TXE 中断<br>1：使能 TXE 中断。TXE 标志置 1 时产生中断请求。 |
| 6 | RXNEIE | 接收缓冲区非空中断使能<br>0：屏蔽 RXNE 中断<br>1：使能 RXNE 中断。RXNE 标志置 1 时产生中断请求。 |
| 5 | ERRIE | 错误中断使能<br>此位用于控制在错误状况发生时是否产生中断(SPI 模式中的 CRCERR、OVR、MODF，以及 UDR、OVR 和 FRE)。<br>0：屏蔽错误中断<br>1：使能错误中断 |
| 4 | FRF | 帧格式<br>0：SPI Motorola 模式 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 1：SPI TI 模式<br>*注意： 不适用于 I2S 模式* |
| 3 | 保留 | 必须保持复位值。 |
| 2 | SSOE | SS 输出使能<br>0：在主模式下禁止 SS 输出，可在多主模式配置下工作<br>1：在主模式下使能 SS 输出，不能在多主模式环境下工作<br>*注意： 不适用于 I2S 模式和 SPI TI 模式* |
| 1 | TXDMAEN | 发送缓冲区 DMA 使能<br>当此位置 1 时，每当 TXE 标志置 1 时，即产生 DMA 请求。<br>0：关闭发送缓冲区 DMA<br>1：使能发送缓冲区 DMA |
| 0 | RXDMAEN | 接收缓冲区 DMA 使能<br>当此位置 1 时，每当 RXNE 标志置 1 时，即产生 DMA 请求。<br>0：关闭接收缓冲区 DMA<br>1：使能接收缓冲区 DMA |

### 22.5.3. SPI 状态寄存器 (SPI_SR)

偏移地址： 0x08

复位值： 0x0002

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | FRE | BSY | OVR | MODF | CRCERR | UDR | CHSIDE | TXE | RXNE |
| | | | | | | | r | r | r | r | rc_w0 | r | r | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:9 | 保留 | 必须保持复位值。 |
| 8 | FRE | 帧格式错误<br>0：没有帧格式错误<br>1：出现帧格式错误<br>此标志由硬件置 1，当读取 SPIx_SR 寄存器时由软件清零。<br>*注意：当 SPI 在 TI 从模式或 I2S 从模式下工作时，使用此标志* |
| 7 | BSY | 忙标志<br>0：SPI（或 I2S）不繁忙<br>1：SPI（或 I2S）忙于通信或者发送缓冲区不为空<br>此标志由硬件置 1 和清零。<br>*注意：BSY 标志必须谨慎使用* |
| 6 | OVR | 上溢标志<br>0：未发生上溢<br>1：发生上溢<br>此标志由硬件置 1，可由软件序列复位。 |
| 5 | MODF | 模式故障<br>0：未发生模式故障<br>1：发生模式故障<br>此标志由硬件置 1，可由软件序列复位。 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | *注意：不适用于 I2S 模式* |
| 4 | CRCERR | CRC 错误标志<br>0：接收到的 CRC 值与 SPI_RXCRCR 值匹配<br>1：接收到的 CRC 值与 SPI_RXCRCR 值不匹配<br>此标志由硬件置 1，通过软件写入 0 来清零。<br>*注意：不适用于 I2S 模式* |
| 3 | UDR | 下溢标志<br>0：未发生下溢<br>1：发生下溢<br>此标志由硬件置 1，可由软件序列复位。<br>*注意：不适用于 SPI 模式* |
| 2 | CHSIDE | 通道信息<br>0：发送或接收左通道信息<br>1：发送或接收右通道信息<br>*注意：不适用于 SPI 模式。在 PCM 模式下没有意义* |
| 1 | TXE | 发送缓冲区为空<br>0：发送缓冲区非空<br>1：发送缓冲区为空 |
| 0 | RXNE | 接收缓冲区非空<br>0：接收缓冲区为空<br>1：接收缓冲区非空 |

### 22.5.4. SPI 数据寄存器 (SPI_DR)

偏移地址： 0x0C

复位值： 0x0000

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

| 31:16 | 保留 | 必须保持复位值。 |
|---|---|---|
| 15:0 | DR[15:0] | 数据寄存器<br>已接收或者要发送的数据。<br>数据寄存器分为 2 个缓冲区，一个用于写入（发送缓冲区），一个用于读取（接收缓冲区）。对数据寄存器执行写操作时，数据将写入发送缓冲区，从数据寄存器执行读取时，将返回接收缓冲区中的值。<br>**针对 SPI 模式的说明：**<br>发送或接收的数据为 8 位或 16 位，具体取决于数据帧格式选择位（SPI_CR1 寄存器中的 DFF）。必须在使能 SPI 前进行此项选择，以确保操作正确。<br>对于 8 位数据帧，缓冲区为 8 位，只有寄存器的 LSB (SPI_DR[7:0]) 用于发送/接收。在接收模式下，寄存器的 MSB (SPI_DR[15:8]) 强制为 0。<br>对于 16 位数据帧，缓冲区为 16 位，整个寄存器 SPI_DR[15:0] 均用于发送/接收。 |

### 22.5.5. 多项式寄存器 (SPI_CRCPR)（不用于 I2S 模式）

偏移地址： 0x10

复位值： 0x0007

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15:0 | CRCPOLY[15:0] | CRC 多项式寄存器<br>此寄存器包含用于 CRC 计算的多项式。<br>CRC 多项式 (0007h) 是此寄存器的复位值。可根据需要配置另一个多项式。<br>*注意： 不适用于 I2S 模式。* |

### 22.5.6. SPI RX CRC 寄存器 (SPI_RXCRCR)（不用于 I2S 模式）

偏移地址： 0x14

复位值： 0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15:0 | RXCRC[15:0] | 接收 CRC 寄存器<br>使能 CRC 计算后，RxCRC[15:0] 位将包含后续接收字节在计算后所得到的 CRC 值。当 SPI_CR1 寄存器中的 CRCEN 位写入 1 时，此寄存器复位。CRC 通过 SPI_CRCPR 寄存器中编程的多项式连续计算。<br>数据帧格式设置为 8 位数据（SPI_CR1 的 DFF 位清零）时，仅考虑 8 个 LSB 位。CRC 计算依据任意 CRC8 标准进行。<br>选择 16 位数据帧格式（SPI_CR1 寄存器的 DFF 位置 1）时，考虑此寄存器的全部 16 个位。CRC 计算依据任意 CRC16 标准进行。<br>*注意： 当 BSY 标志置 1 时，读取此寄存器可能返回一个不正确的值。*<br>*不适用于 I2S 模式。* |

### 22.5.7. SPI TX CRC 寄存器 (SPI_TXCRCR)（不用于 I2S 模式）

偏移地址： 0x18

复位值： 0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15:0 | TXCRC[15:0] | 发送 CRC 寄存器<br>使能 CRC 计算后, TxCRC[7:0] 位将包含后续发送字节在计算后所得到的 CRC 值。当 SPI_CR1 寄存器中的 CRCEN 位写入 1 时, 此寄存器复位。CRC 通过 SPI_CRCPR 寄存器中编程的多项式连续计算。<br>数据帧格式设置为 8 位数据 (SPI_CR1 的 DFF 位清零) 时, 仅考虑 8 个 LSB 位。CRC 计算依据任意 CRC8 标准进行。<br>选择 16 位数据帧格式 (SPI_CR1 寄存器的 DFF 位置 1) 时, 考虑此寄存器的全部 16 个位。CRC 计算依据任意 CRC16 标准进行。<br>*注意: 当 BSY 标志置 1 时, 读取此寄存器可能返回一个不正确的值。*<br>*不适用于 I2S 模式。* |

### 22.5.8. SPI_I2S 配置寄存器 (SPI_I2SCFGR)

偏移地址: 0x1C

复位值: 0x0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | I2SMOD | I2SE | I2SCFG | | PCMSYNC | Res | I2SSTD | | CKPOL | DATLEN | | CHLEN |
| | | | | rw | rw | rw | rw | rw | | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:12 | 保留 | 必须保持复位值。 |
| 11 | I2SMOD | $I^2S$ 模式选择<br>0: 选择 SPI 模式<br>1: 选择 $I^2S$ 模式<br>*注意: 应在 SPI 或 I2S 禁止时配置此位。* |
| 10 | I2SE | $I^2S$ 使能<br>0: 关闭 $I^2S$ 外设<br>1: 使能 $I^2S$ 外设<br>*注意: 不适用于 SPI 模式* |
| 9:8 | I2SCFG | $I^2S$ 配置模式<br>00: 从模式 - 发送<br>01: 从模式 - 接收<br>10: 主模式 - 发送<br>11: 主模式 - 接收<br>*注意: 应在 I2S 禁止时配置此位。*<br>*不适用于 SPI 模式* |
| 7 | PCMSYNC | PCM 帧同步 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 0：短帧同步<br>1：长帧同步<br>*注意： 只有在 I2SSTD = 11（使用 PCM 标准）时，此位才有意义*<br>*不适用于 SPI 模式* |
| 6 | 保留 | 必须保持复位值。 |
| 5:4 | I2SSTD | I²S 标准选择<br>00：I²S Philips 标准。<br>01：MSB 对齐标准（左对齐）<br>10：LSB 对齐标准（右对齐）<br>11：PCM 标准<br>*注意： 为确保正确运行，应在 I2S 关闭时配置这些位。*<br>*不适用于 SPI 模式。* |
| 3 | CKPOL | 空闲状态的时钟电平<br>0：空闲状态时钟为低电平<br>1：空闲状态时钟为高电平<br>*注意： 为确保正确运行，应在 I2S 关闭时配置此位。*<br>*不适用于 SPI 模式* |
| 2:1 | DATLEN | 传输的数据长度<br>00：16 位数据长度<br>01：24 位数据长度<br>10：32 位数据长度<br>11：不允许<br>*注意： 为确保正确运行，应在 I2S 关闭时配置这些位。*<br>*不适用于 SPI 模式。* |
| 0 | CHLEN | 通道长度（每个音频通道的位数）<br>0：16 位<br>1：32 位<br>只有在 DATLEN = 00 时，此位的写操作才有意义，否则无论填入何值，通道长度始终由硬件固定为 32 位。不适用于 SPI 模式。<br>*注意： 为确保正确运行，应在 I2S 关闭时配置此位。* |

### 22.5.9. SPI_I2S 预分频器寄存器 (SPI_I2SPR)

偏移地址： 0x20

复位值： 0000 0010 (0x0002)

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:10 | 保留 | 必须保持复位值。 |
| 9 | MCKOE | 主时钟输出使能<br>0：禁止主时钟输出<br>1：使能主时钟输出<br>*注意： 应在 I2S 禁止时配置此位。只有在 I2S 为主模式时，才会使用此位。* |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | *不适用于 SPI 模式。* |
| 8 | ODD | 预分频器的奇数因子<br>0：实际分频值为 = I2SDIV *2<br>1：实际分频值为 = (I2SDIV * 2)+1。<br>*注意： 应在 I2S 禁止时配置此位。只有在 I2S 为主模式时，才会使用此位。*<br>*不适用于 SPI 模式。* |
| 7:0 | I2SDIV | I2S 线性预分频器<br>I2SDIV [7:0] = 0 或 I2SDIV [7:0] = 1 为禁用值。<br>*注意： 应在 I2S 禁止时配置这些位。只有在 I2S 为主模式时，才会使用此位。*<br>*不适用于 SPI 模式。* |

### 22.5.10. SPI_I2S 寄存器 3 (SPI_CR3)

偏移地址： 0x40

复位值： 0x00000FF0

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | TFIE | RFIE | TFTH[3:0] | | | | RFTH[3:0] | | | | TFEIE | RFPIE | TFEN | RFEN |
| | | rw | rw | rw | | | | rw | | | | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:14 | 保留 | 必须保持复位值。 |
| 13 | TFIE | 发送 FIFO 满中断使能<br>0：禁止中断<br>1：使能中断 |
| 12 | RFIE | 接收 FIFO 满中断使能<br>0：禁止中断<br>1：使能中断 |
| 11:8 | TFTH[3:0] | 发送 FIFO 将空标志阈值<br>FIFO 数据小于等于 TFTH[3:0]+1 时，起将空标志<br>例如：<br>0 表示 FIFO 有小于等于 1 个数据时，起将空标志<br>F 表示 FIFO 有小于等于 16 个数据时，起将空标志 |
| 7:4 | RFTH[3:0] | 接收 FIFO 将满标志阈值<br>FIFO 数据大于等于 TFTH[3:0]+1 时，起将满标志<br>例如：<br>0 表示 FIFO 有大于等于 1 个数据时，起将满标志<br>F 表示 FIFO 有大于等于 16 个数据时，起将满标志 |
| 3 | TFEIE | 发送 FIFO 将空中断使能<br>0：禁止中断<br>1：使能中断 |
| 2 | RFPIE | 接收 FIFO 将满中断使能<br>0：禁止中断<br>1：使能中断 |
| 1 | TFEN | 发送 FIFO 控制使能 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 0：禁止发送 FIFO 功能<br>1：使能发送 FIFO 功能 |
| 0 | RFEN | 接收 FIFO 控制使能<br>0：禁止接收 FIFO 功能<br>1：使能接收 FIFO 功能 |

### 22.5.11. SPI_I2S 状态寄存器 2 (SPI_SR2)

偏移地址： 0x44

复位值： 0x0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | TFF | RFF | TFFLR[4:0] | | | | | RFFLR[4:0] | | | | | TFE |
| | | ro | ro | ro | | | | | ro | | | | | ro |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:14 | 保留 | 必须保持复位值。 |
| 13 | TFF | 发送 FIFO 满标志<br>0：发送 FIFO 未满<br>1：发送 FIFO 满 |
| 12 | RFF | 接收 FIFO 满标志<br>0：接收 FIFO 未满<br>1：接收 FIFO 满 |
| 11:7 | TFFLR[4:0] | 当前发送 FIFO 存储字节个数<br>FIFO 中的数据数量为 TFFLR[4:0] Bytes |
| 6:2 | RFFLR[4:0] | 当前接收 FIFO 存储字节个数<br>FIFO 中的数据数量为 RFFLR[4:0] Bytes |
| 1 | TFE | 发送 FIFO 将空标志<br>0：发送 FIFO 数据量大于将空阈值<br>1：发送 FIFO 数据量小于等于将空阈值 |
| 0 | RFP | 接收 FIFO 将满标志<br>0：接收 FIFO 数据量小于将满阈值<br>1：接收 FIFO 数据量大于等于将满阈值 |

*注意： 是否使能中断不影响 FIFO 标志位状态。*

### 22.5.12. SPI_I2S FIFO 清除寄存器 (SPI_FIFO_CLR)

偏移地址： 0x48

复位值： 0x0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:2 | 保留 | 必须保持复位值。 |
| 1 | TFCLR | 发送 FIFO 清除控制<br>配置 1 时将清除发送 FIFO 中的所有数据 |
| 0 | RFCLR | 接收 FIFO 清除控制<br>配置 1 时将清除接收 FIFO 中的所有数据 |

# 23. SDIO 接口 (SDIO)

## 23.1. SDIO 主要特性

SD/SDIO MMC 卡主机接口 (SDIO) 提供 APB2 外设总线与多媒体卡 (MMC)、SD 卡、SDIO 卡以及 CE-ATA 设备之间的接口。

多媒体卡协会网站 www.mmca.org 中提供了由 MMCA 技术委员会发布的多媒体卡系统规范。

SD 卡协会网站 www.sdcard.org 中提供了 SD 存储卡和 SD I/O 卡系统规范。

CE-ATA 工作组网站 www.ce-ata.org 中提供了 CE-ATA 系统规范。

SDIO 具有以下特性:

- 完全兼容多媒体卡系统规范版本 4.2。卡支持三种不同数据总线模式: 1 位（默认）、4 位和 8 位;
- 完全兼容先前版本的多媒体卡（向前兼容性）;
- 完全兼容 SD 存储卡规范版本 2.0;
- 完全兼容 SD I/O 卡规范版本 2.0: 卡支持两种不同数据总线模式: 1 位（默认）和 4 位;
- 完全支持 CE-ATA 功能（完全符合 CE-ATA 数字协议版本 1.1）;
- 对于 8 位模式，数据传输高达 48 MHz;
- 数据和命令输出使能信号，控制外部双向驱动程序。

*注意: SDIO 不具备兼容 SPI 的通信模式。*

*SD 存储卡协议是多媒体卡系统规范版本 2.11 定义的多媒体卡协议的超集。一些 SD 存储卡器件所需的命令，仅 SD-I/O 的卡或组合卡的 I/O 部分都不支持。部分上述命令（如擦除命令）无法在 SD I/O 器件中使用，因此在 SDIO 中不受支持。此外，一些命令在 SD 存储卡和 SD I/O 卡之间有所不同，因此在 SDIO 中不受支持。有关详细信息，请参见 SD I/O 卡规范版本 1.0。CE-ATA 支持通过 MMC 电气接口实现，这种接口使用一种利用现有 MMC 访问原语的协议。接口的电气和信令定义与 MMC 参考中定义的相同。*

*多媒体卡/SD 总线将卡连接到控制器。*

*当前版本的 SDIO 每次只支持一个 SD/SDIO/MMC4.2 卡，但支持多个 MMC4.1 或之前版本的卡。*

## 23.2. SDIO 总线拓扑

总线的通信基于命令和数据传输。

多媒体卡/SD/SD I/O 总线上的基本事务是命令/响应事务。这些类型的总线事务直接在命令或响应结构中传输其信息。此外，某些操作具有数据令牌。

与 SD/SDIO 存储卡的数据相互传输在数据块中执行。与 MMC 的数据相互传输在数据块或流中执行。与 CE-ATA 器件的数据相互传输在数据块中执行。

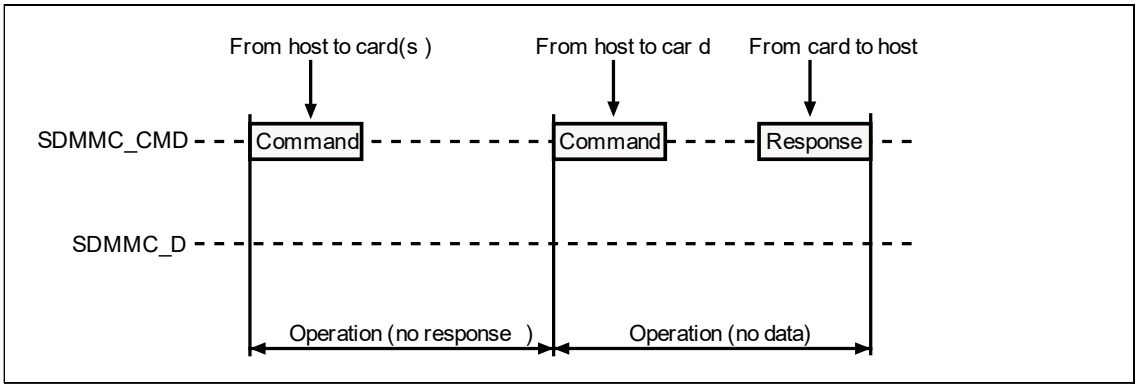



图 23-1 SDIO“无响应”和“无数据”操作

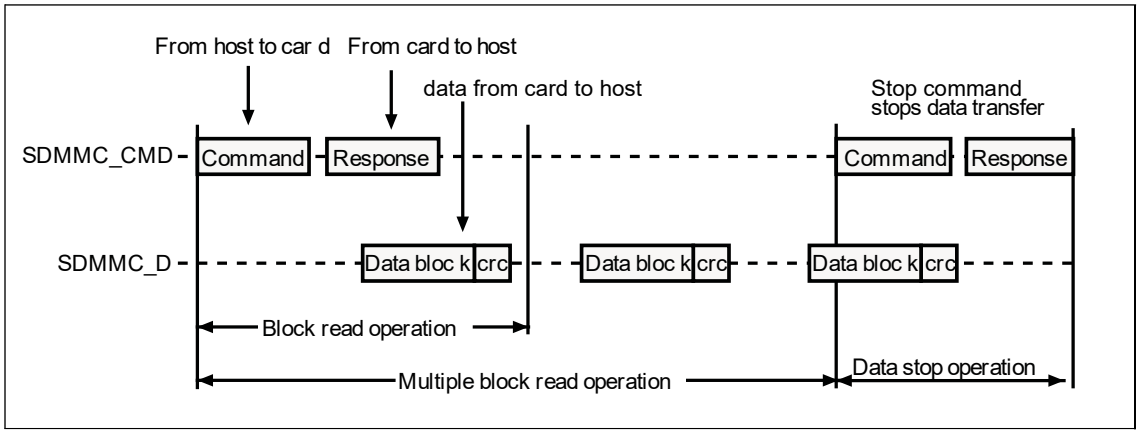
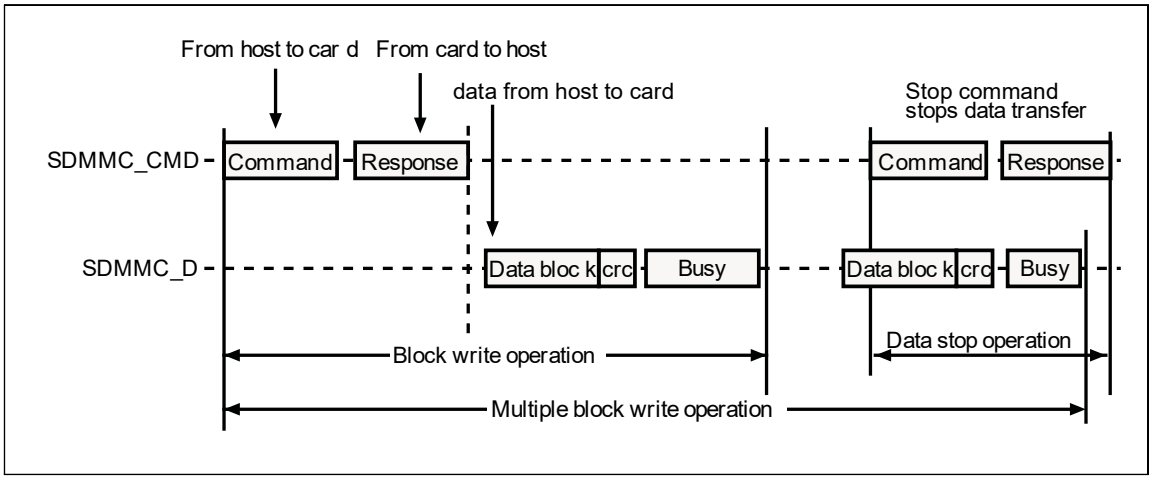



图 23-2 SDIO(多个)块读取操作




*注意： 只要发出"繁忙"信号（ SDIO_D0 被拉到低电平)， SDIO 便不会发送任何数据。*

图 23-3 SDIO(多个)块写入操作

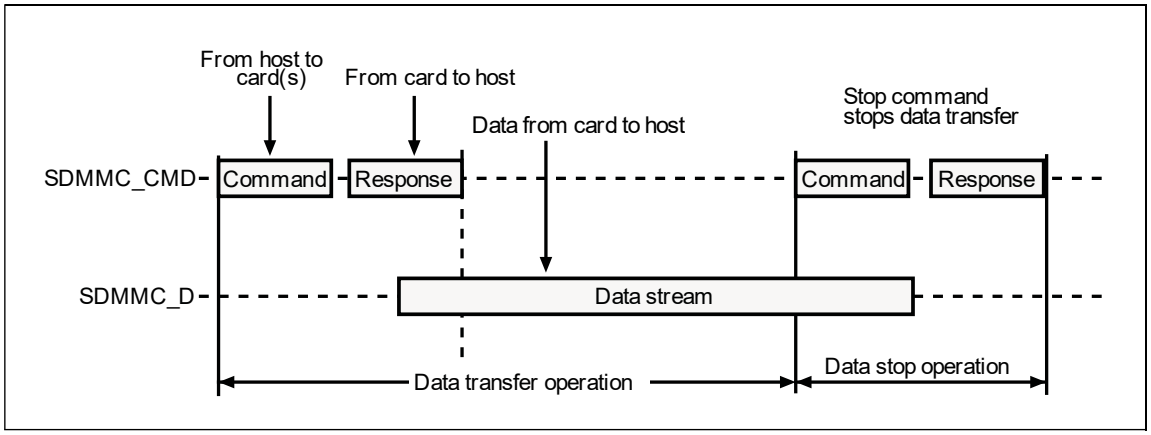



图 23-4 SDIO 连续读取操作

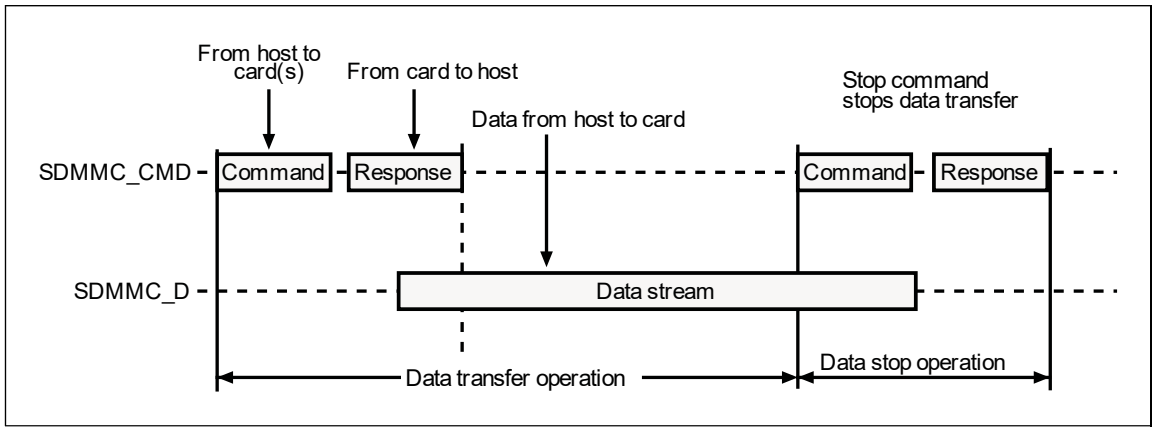



图 23-5 SDIO 连续写入操作

## 23.3. SDIO 功能说明

SDIO 由两部分组成：

- SDIO 适配器块提供特定于 MMC/SD/SD I/O 卡的所有功能，如时钟生成单元、命令和数据传输。
- APB2 接口访问 SDIO 适配器寄存器，并且生成中断和 DMA 请求信号。

默认情况下，SDIO_D0 用于数据传输。初始化后，主机可以更改数据总线宽度。如果多媒体卡连接到总线，则 SDIO_D0、SDIO_D[3:0] 或 SDIO_D[7:0] 可以用于数据传输。MMC V3.31 或更低版本仅支持 1 位数据，因此只能使用 SDIO_D0。如果 SD 或 SD I/O 卡连接到总线，则主机可以将数据传输配置为使用 SDIO_D0 或 SDIO_D[3:0]。所有数据线均以推挽模式运行。

SDIO_CMD 有两种操作模式：

- 开漏引脚，用于初始化（仅限于 MMC V3.31 或更低版本）。
- 推挽，用于命令传输（SD/SD I/O 卡 MMC4.2 还将推挽驱动程序用于初始化）。

SDIO_CK 是与卡相连的时钟：一个位在每个时钟周期内同时在命令和数据线上传输。对于 MMC V3.31，时钟频率可以在 0 MHz 到 20 MHz 之间变化，对于 MMC V4.0/4.2，可以在 0 到 48 MHz 之间变化，对于 SD/SD I/O 卡，可以在 0 到 25 MHz 之间变化。

SDIO 使用两个时钟信号：

- SDIO 适配器时钟 (SDIOCLK = 48 MHz)。
- APB2 总线时钟 (PCLK2)。

PCLK2 和 SDIO_CK 时钟频率必须满足以下条件：

$$\mathbf{Frequency(PCLK2)} \geq \mathbf{3/8} \times \mathbf{Frequency(SDIO_CK)}$$

下表中的信号在多媒体卡/SD/SD I/O 卡总线中使用。

表 23-1 SDIO I/O 定义

| 引脚 | 方向 | 说明 |
|---|---|---|
| SDIO_CK | 输出 | 多媒体卡/SD/SDIO 卡时钟。该引脚是从主机到卡之间的时钟。 |
| SDIO_CMD | 双向 | 多媒体卡/SD/SDIO 卡命令。该引脚是双向命令/响应信号。 |
| SDIO_D[7:0] | 双向 | 多媒体卡/SD/SDIO 卡数据。这些引脚是双向数据总线。 |

### 23.3.1. SDIO 适配器

下图显示了 SDIO 适配器的简化框图。

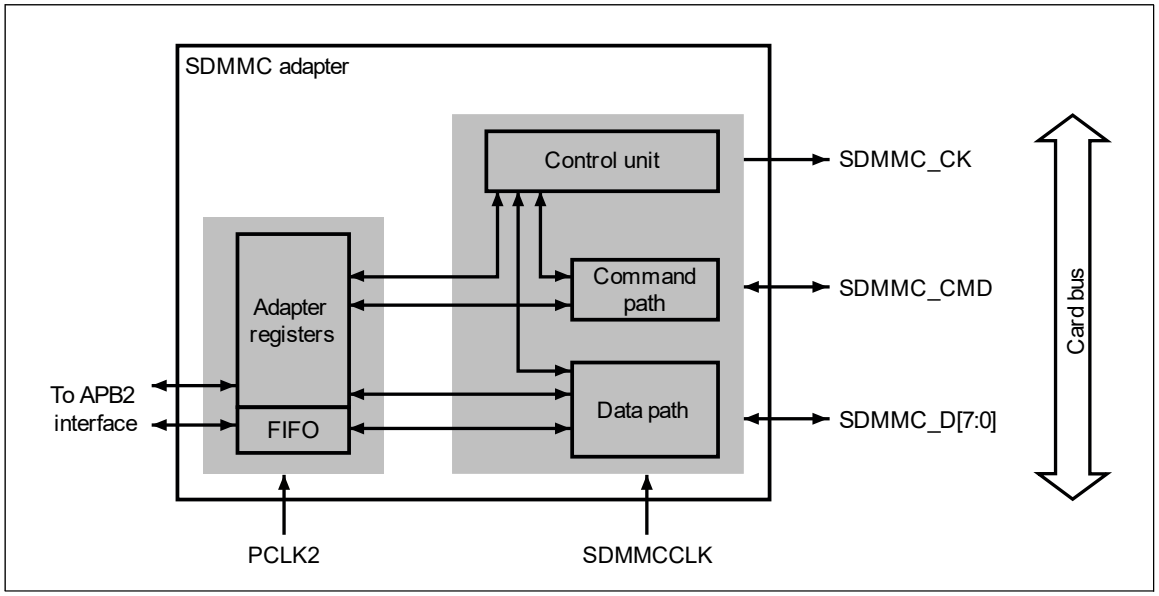



图 23-6 SDIO 适配器

SDIO 适配器是一个多媒体卡/安全数字存储卡总线主设备，提供与多媒体卡堆栈或安全数字存储卡的接口。该适配器由五个子单元组成：

- 适配器寄存器块；
- 控制单元；
- 命令路径；
- 数据路径；
- 数据 FIFO。

*注意：适配器寄存器和 FIFO 使用 APB2 总线时钟域 (PCLK2)。控制单元、命令路径和数据路径使用 SDIO 适配器时钟域 (SDIOCLK)。*

#### 适配器寄存器块

适配器寄存器模块包含所有系统寄存器。该模块还生成将多媒体卡中的静态标志清零的信号。当写入 1 到 SDIO 清零寄存器中对应位的位置时，将生成清零信号。

### 控制单元

控制单元包含电源管理功能和存储卡时钟的时钟分频器。

有三个电源阶段：

- 掉电；
- 上电；
- 通电。

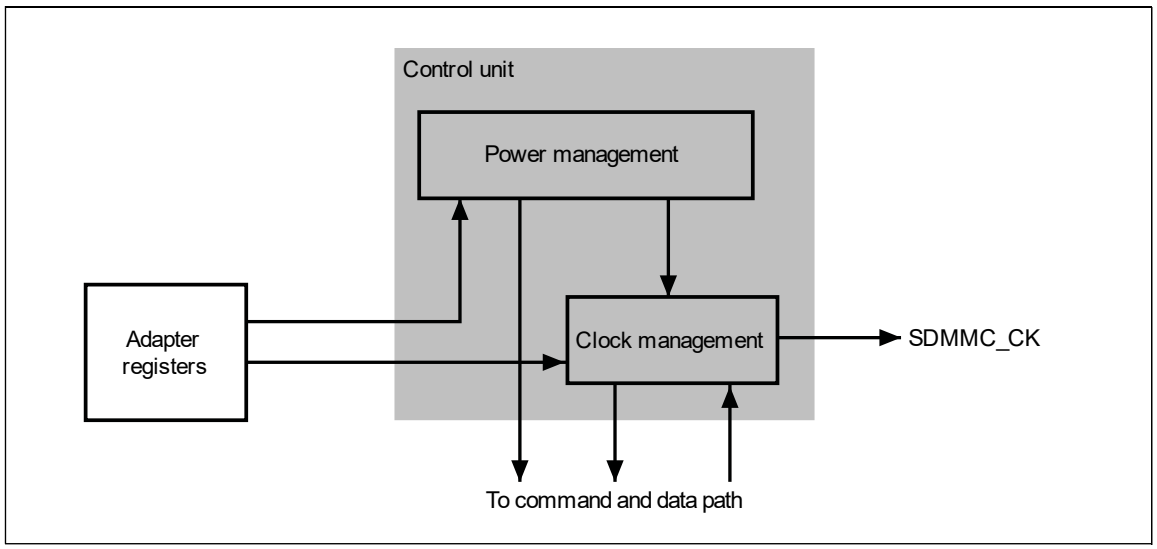



图 23-7 控制单元

控制单元如上图所示。控制单元由电源管理子单元和时钟管理子单元组成。

电源管理子单元会在断电阶段和上电阶段中禁止卡总线输出信号。

时钟管理子单元负责生成和控制 SDIO_CK 信号。SDIO_CK 输出可以使用时钟分频或时钟旁路模式。时钟输出在以下情况下无效：

- 复位后；
- 在断电或上电阶段中；
- 在使能了节电模式并且卡总线处于空闲状态的情况下（命令和数据路径子单元进入空闲阶段后的八个时钟周期之后）。

### 命令路径

命令路径单元向卡发送命令并从卡接收响应。

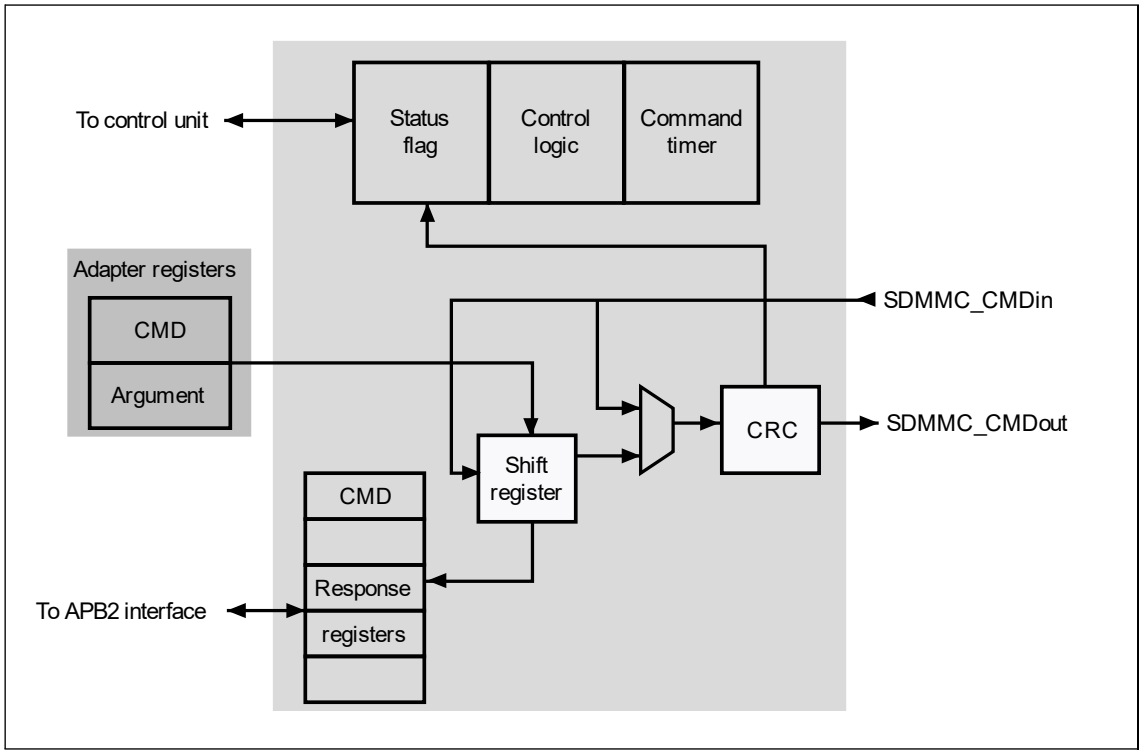



图 23-8 SDIO 适配器命令路径

- 命令路径状态机 (CPSM)
- 写入命令寄存器并且将使能位置 1 后，命令传输将开始。发送了命令后，命令路径 状态机 (CPSM) 将将状态标志置 1，并且在不需要响应时进入空闲状态。如果需要响应，则等待响应。收到响应时，将比较生成的 CRC 代码和内部生成代码，并且将相应的状态标志置 1。

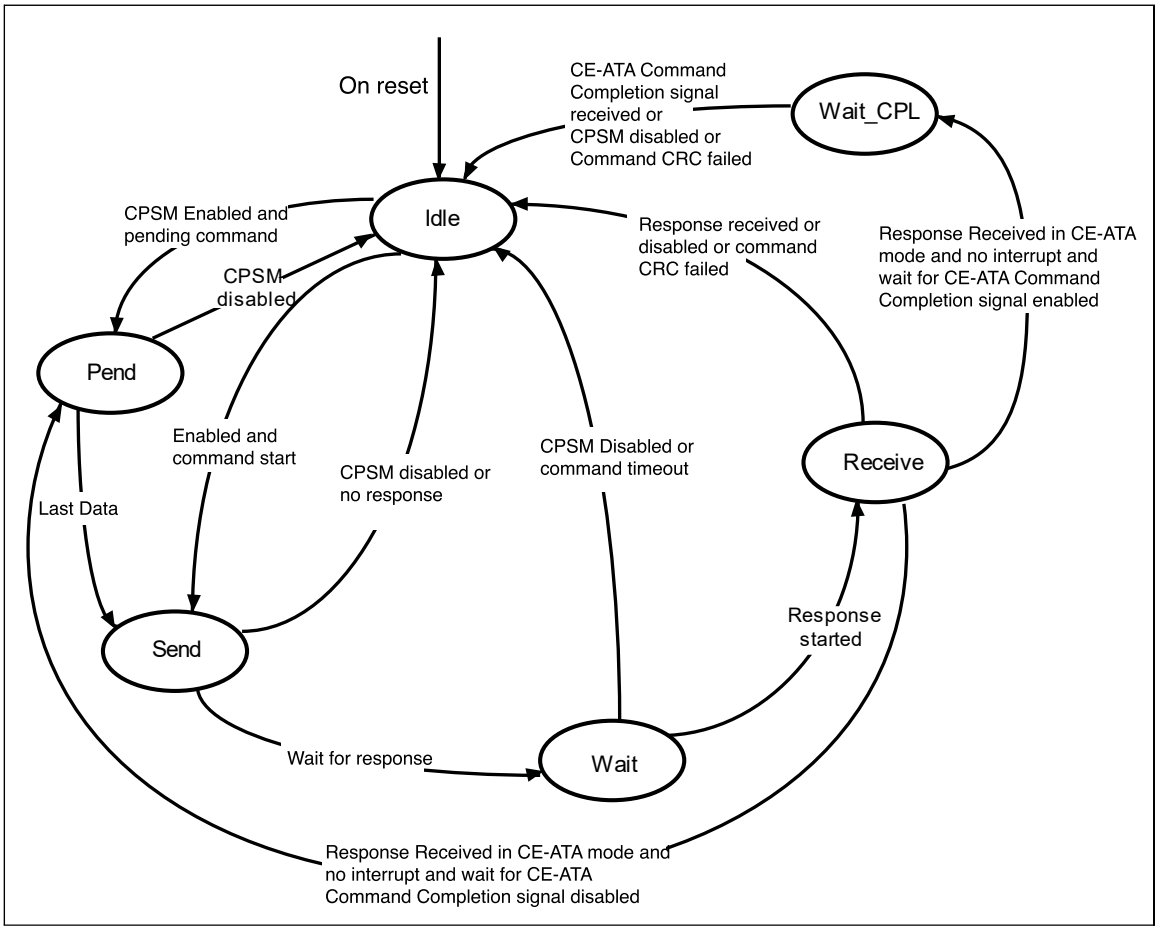



图 23-9 命令路径状态机(CPSM)

进入等待状态后，命令计时器便开始运行。如果 CPSM 尚未变为接收状态便已达到超时，则将超时标志置 1 并且进入空闲状态。

*注意：命令超时为固定值：64 个 SDIO_CK 时钟周期。*

*如果在命令寄存器中将中断位置 1，则禁止计时器运行并且 CPSM 等待来自其中一个卡的中断请求。如果在命令寄存器中将挂起位置 1，则 CPSM 进入挂起状态并等待来自数据路径子单元的 CmdPend 信号。*

*检测到 CmdPend 时，CPSM 将变为发送状态。这会使能数据计数器，用以触发停止命令传输。*

*注意：CPSM 保持空闲状态至少八个 SDIO_CK 周期以满足 NCC 和 NRC 时序限制。NCC 是两个主机命令间的最小延迟，NRC 是主机命令和卡响应之间的最小延迟。*

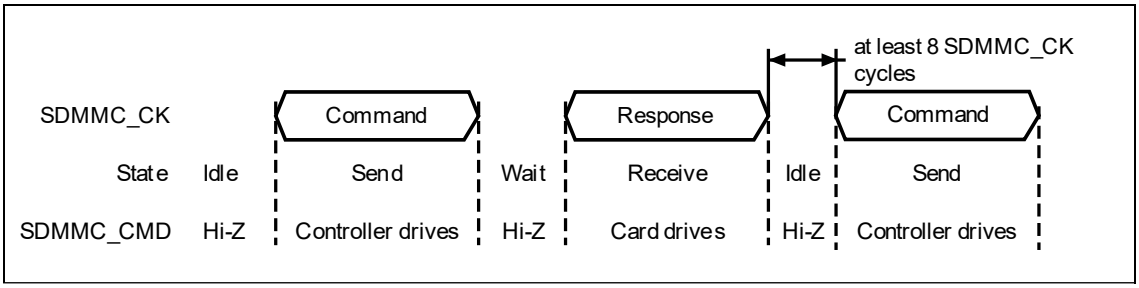



图 23-10 SDIO 命令传输

- **命令格式**

- 命令：命令是用于启动操作的令牌。命令从主机发送到单个卡（编址命令），或发送到所有已连接的卡（MMC V3.31 或更低版本可以使用广播命令）。命令在 CMD 线上以串行方式传输。所有命令都为固定长度 48 位。下表中显示了多媒体卡、SD 存储卡和 SDIO 卡的命令令牌的常规格式。CE-ATA 命令是 MMC 命令 V4.2 的扩展，因此格式相同。
- 命令路径以半双工模式运行，因此可以发送或者接收命令和响应。如果 CPSM 未处于发送状态，则 SDIO_CMD 输出处于 Hi-Z 状态。SDIO_CMD 上的数据与 SDIO_CK 的上升沿保持同步。

表 23-2 命令格式

| 位的位置 | 宽度 | 值 | 说明 |
|---|---|---|---|
| 47 | 1 | 0 | 起始位 |
| 46 | 1 | 1 | 传输位 |
| [45:40] | 6 | - | 命令索引 |
| [39:8] | 32 | - | 参数 |
| [7:1] | 7 | - | CRC7 |
| 0 | 1 | 1 | 结束位 |

- 响应：响应是一个令牌，它作为对先前接收命令的应答，从编址卡（或者，对于 MMC V3.31 或更低版本，从所有已连接的卡同步）发送到主机。响应在 CMD 线上以串行方式传输。

SDIO 支持两种响应类型。两种类型均使用 CRC 错误校验：

- 48 位短响应；
- 136 位长响应。

*注意：如果响应不包含 CRC（CMD1 响应），则设备驱动程序必须忽略 CRC 失败状态。*

表 23-3 短响应格式

| 位的位置 | 宽度 | 值 | 说明 |
|---|---|---|---|
| 47 | 1 | 0 | 起始位 |
| 46 | 1 | 1 | 传输位 |
| [45:40] | 6 | — | 命令索引 |
| [39:8] | 32 | — | 参数 |
| [7:1] | 7 | — | CRC7(或 1111111) |
| 0 | 1 | 1 | 结束位 |

表 23-4 长响应格式

| 位的位置 | 宽度 | 值 | 说明 |
|---|---|---|---|
| 135 | 1 | 0 | 起始位 |
| 134 | 1 | 0 | 传输位 |
| [133:128] | 6 | 111111 | 保留 |
| [127:1] | 127 | | CID 或 CSD(包括内部 CRC) |
| 0 | 1 | 1 | 结束位 |

命令寄存器包含命令索引(发送到卡的六位)和命令类型。它们确定命令是否需要影响以及响应长度的长度是 48 位还是 136 位。命令路径实施下表中的显示的状态标志：

表 23-5 命令路径状态标志

| 标志 | 说明 |
|---|---|
| CMDREND | 在响应 CRC 正常的情况下将此标志置 1 |
| CCRCFAIL | 在响应 CRC 失败的情况下将此标志置 1 |
| CMDSENT | 发送了不需要响应的命令式将此标志置 1 |

| 标志 | 说明 |
| --- | --- |
| CTIMEOUT | 响应超时 |
| CMDACT | 正在进行命令传输 |

CRC 生成计算 CRC 代码前面所有位的 CRC 校验和。这包括起始位、传输位、命令索引和命令参数。对于长响应格式，将为 CID 或 CSD 的前 120 位计算 CRC 校验和。请注意，CRC 计算中不使用起始位、传输位和六个保留位。

CRC 校验和是一个 7 位数值：

$$\mathbf{CRC}[\mathbf{6}:\mathbf{0}] = \mathbf{余数}[(\mathbf{M}(\mathbf{X}) * \mathbf{X}^{\mathbf{7}}) / \mathbf{G}(\mathbf{X})]$$

$$\mathbf{G}(\mathbf{X}) = \mathbf{X}^{\mathbf{7}} + \mathbf{X}^{\mathbf{3}} + \mathbf{1}$$

$$\mathbf{M}(\mathbf{X}) = \left(\mathbf{起始位}\right) * \mathbf{X}^{\mathbf{39}} + \cdots + \left(\mathbf{CRC前的最后一位}\right) * \mathbf{X}^{\mathbf{0}} \text{ 或者}$$

$$\mathbf{M}(\mathbf{X}) = \left(\mathbf{起始位}\right) * \mathbf{X}^{\mathbf{119}} + \cdots + \left(\mathbf{CRC前的最后一位}\right) * \mathbf{X}^{\mathbf{0}}$$

### 数据路径

数据路径子单元负责与卡相互传输数据。下图显示了数据路径的框图。

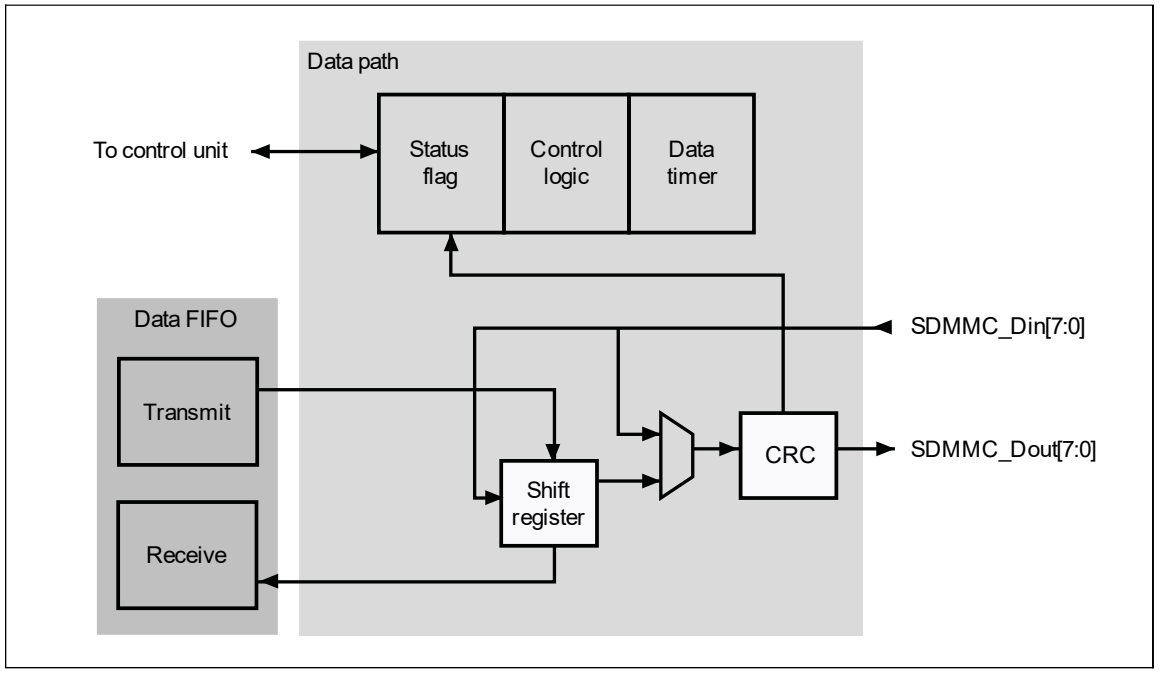



图 23-11 数据路径

可以使用时钟控制寄存器对卡数据总线宽度进行编程。如果使能了 4 位宽度的总线模式，则使用所有四个数据信号线 (SDIO_D[3:0]) 在每个时钟周期内传输 4 个数据位。如果使能了 8 位宽度的总线模式，则使用所有八个数据信号线 (SDIO_D[7:0]) 在每个时钟周期内传输 8 个数据位。如果未使能宽总线模式，则每个时钟周期仅传输一位，且使用 SDIO_D0。

根据传输方向（发送或接收），数据路径状态机 (DPSM) 将变为 Wait_S 或 Wait_R 状态 （如果已使能）：

- 发送：DPSM 变为 Wait_S 状态。如果传输 FIFO 中有数据，则 DPSM 变为发送状态并且数据路径子单元开始向卡发送数据。
- 接收：DPSM 变为 Wait_R 状态并等待起始位。在 DPSM 收到起始位时，将变为接收状态，并且数据路径子单元开始从卡中接收数据。

**数据路径状态机 (DPSM)**

DPSM 以 SDIO_CK 频率运行。卡总线信号上的数据与 SDIO_CK 的上升沿保持同步。

DPSM 有六种状态，如下图所示。

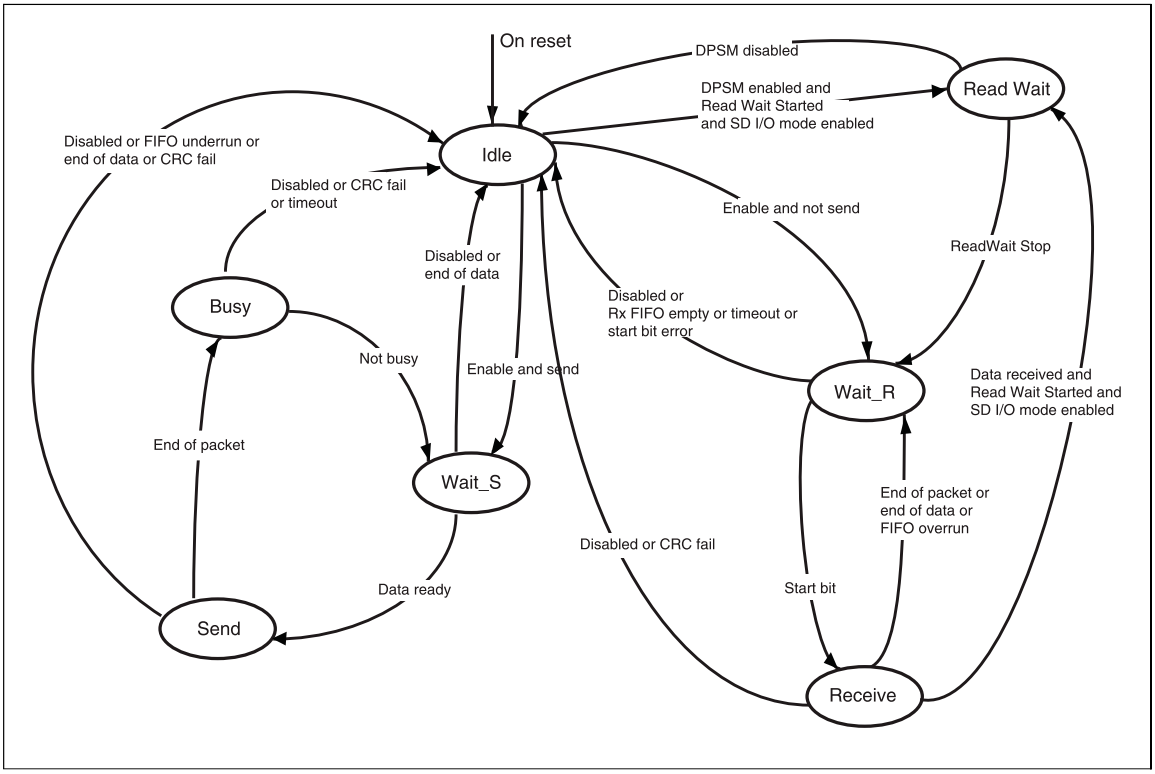



图 23-12 数据路径状态机(DPSM)

- 空闲：数据路径处于无效状态，SDIO_D[7:0] 输出处于 Hi-Z。当写入数据控制寄存器并将使能位置 1 后，DPSM 将使用新值来加载数据计数器，根据数据方向位，将变为 Wait_S 或 Wait_R 状态。
- Wait_R：如果数据计数器等于零，则 DPSM 会在接收 FIFO 为空时变为空闲状态。如果数据计数器不为零，则 DPSM 等待 SDIO_D 上的起始位。如果 DPSM 在超时之前收到了起始位，则 DPSM 变为接收状态，并加载数据块计数器。如果 DPSM 尚未检测到起始位便已超时或者发生起始位错误，则变为空闲状态并且将超时状态标志置 1。
- 接收：从卡收到的串行数据以字节为单位打包，并写入到数据 FIFO。根据数据控制寄存器中的传输模式，数据传输模式可以是块或流：
  - 在块模式中，当数据块计数器达到零时，DPSM 将等待直至其收到 CRC 代码。如果收到的代码与内部生成的 CRC 代码相匹配，则 DPSM 变为 Wait_R 状态。如果不匹配，则将 CRC 失败状态标志置 1 并且 DPSM 变为空闲状态。
  - 在流模式中，DPSM 在数据计数器不为零时接收数据。当计数器为零时，会将移位寄存器中其余数据写入到数据 FIFO，并且 DPSM 将变为 Wait_R 状态。

如果发生 FIFO 溢出错误，则 DPSM 将 FIFO 错误标志置 1 并变为空闲状态：

- Wait_S：如果数据计数器为零，则 DPSM 变为空闲状态。如果不为零，则 DPSM 将等待直至停止发出 FIFO 空标志，并变为发送状态。

- 发送：DPSM 开始向卡发送数据。根据数据控制寄存器中的传输模式，数据传输模式可以是块或流：
  - 在块模式中，当数据块计数器达到零时，DPSM 将发送一个内部生成的 CRC 代码和结束位，并变为繁忙状态。
  - 在流模式中，DPSM 将在使能位处于高电平并且数据计数器不为零时，向卡发送数据。然后它会变为空闲状态。

如果发生 FIFO 下溢错误，则 DPSM 将 FIFO 错误标志置 1 并变为空闲状态。

- 忙碌：DPSM 等待 CRC 状态标志：
  - 如果未收到正确的 CRC 状态，则变为空闲状态并且将 CRC 失败状态标志置 1。
  - 如果收到正确的 CRC 状态，则在 SDIO_D0 未处于低电平（卡未处于繁忙状态）的情况下变为 Wait_S 状态。
  - 如果 DPSM 处于繁忙状态时发生超时，则数据超时标志置 1 并变为空闲状态。

  当 DPSM 处于 Wait_R 或繁忙状态时将使能数据计时器，并生成数据超时错误：
  - 发送数据时，如果 DPSM 处于繁忙状态的时间超过了编程的超时周期，则发生超时错误。
  - 接收数据时，如果数据的结尾不为“真”，并且 DPSM 处于 Wait_R 状态的时间超过了编程的超时周期，则发生超时。
- 数据：可以将数据从卡发送到主机，或从主机到卡。数据通过数据线来传输。数据存储在 32 字的 FIFO 中，每个字为 32 位宽。

表 23-6 数据令牌格式

| 说明 | 起始位 | 数据 | CRC16 | 结束位 |
|---|---|---|---|---|
| 块数据 | 0 | | 是 | 1 |
| 流数据 | 0 | | 否 | 1 |

### 数据 FIFO

数据 FIFO（先进先出）子单元是一个数据缓冲器，带发送和接收单元。

FIFO 包含一个宽度为 32 位且深度为 32 字的数据缓冲器和发送/接收逻辑。由于数据 FIFO 在 APB2 时钟域 (PCLK2) 中运行，因此来自 SDIO 时钟域 (SDIOCLK) 的子单元中的所有信号都将重新同步。

根据 TXACT 和 RXACT 标志，可以将 FIFO 禁止、使能传输或使能接收。TXACT 和 RXACT 是由数据路径子单元驱动，并且两者互斥：

- 发出 TXACT 时，传输 FIFO 引用传输逻辑和数据缓冲器；
- 发出 RXACT 时，接收 FIFO 引用接收逻辑和数据缓冲器。

传输 FIFO：

如果使能了 SDIO 以用于传输，那么可以通过 APB2 接口将数据写入到传输 FIFO。可以通过 32 个连续地址来访问传输 FIFO。传输 FIFO 包含一个数据输出寄存器，其中存储了读取指针所指向的数据字。在数据路径子单元已加载了其移位寄存器之后，该子单元会将读取指针递增并将新数据推出。如果禁止了传输 FIFO，则停止发出所有状态标志。数据路径子单元在传输数据时置位 TXACT。

表 23-7 传输 FIFO 状态标志

| 标志 | 说明 |
|---|---|
| TXFIFOF | 当所有 32 个接收 FIFO 字都包含有效数据时，设置为高电平。 |
| TXFIFOE | 当接收 FIFO 不包含有效数据时，设置为高电平。 |
| TXFIFOHE | 当 8 个或更多个接收 FIFO 字包含有效数据时，设置为高电平。该标志可以用作 DMA |

| 标志 | 说明 |
| --- | --- |
| | 请求。 |
| TXDAVL | 当接收 FIFO 非空时，设置为高电平。该标志与 RXFIFOE 标志正好相反。 |
| TXUNDERR | 发生溢出错误时，设置为高电平。通过写入到 SDIO 清零寄存器可将该位清零。 |

接收 FIFO

当数据路径子单元接收到一个字的数据时，该子单元将在写入数据总线上驱动数据。在 写入操作完成后，写入指针递增。在读取端，读取指针的当前值指向的 FIFO 字内容被推入读取数据总线中。如果禁止了接收 FIFO，则会停止发出所有状态标志，并且将读取和写入指针复位。数据路径子单元在接收数据时置位 RXACT。表 135 列出了接收 FIFO 状态标志。可以通过 32 个连续地址来访问接收 FIFO。

表 23-8 接收 FIFO 状态标志

| 标志 | 说明 |
| --- | --- |
| RXFIFOF | 当所有 32 个接收 FIFO 字都包含有效数据时，设置为高电平 |
| RXFIFOE | 当接收 FIFO 不包含有效数据时，设置为高电平 |
| RXFIFOHE | 当 8 个或更多个接收 FIFO 字包含有效数据时，设置为高电平。该标志可以用作 DMA 请求 |
| RXDAVL | 当接收 FIFO 非空时，设置为高电平。该标志与 RXFIFOE 标志正好相反 |
| RXOVERR | 发生溢出错误时，设置为高电平。通过写入到 SDIO 清零寄存器可将该位清 |

### 23.3.2. SDIO APB2 接口

APB2 接口生成中断和 DMA 请求，并且访问 SDIO 适配器寄存器和数据 FIFO。该接口由数据路径、寄存器解码器和中断/DMA 逻辑构成。

**SDIO 中断**

中断逻辑生成一个中断请求信号，当至少其中一个所选状态标志为高电平时，便发出此信号。提供了一个屏蔽寄存器，用于选择将生成中断的条件。如果对应的屏蔽标志置 1，则状态标志将生成中断请求。

**SDIO/DMA 接口 – SDIO 和存储器之间的数据传输过程**

在显示的示例中，是使用 CMD24 (WRITE_BLOCK)，从 SDIO 主机控制器传输到 MMC（512 字节）。使用 DMA 控制器将存储器中的数据填充到 SDIO FIFO。

- 执行卡识别过程
- 增加 SDIO_CK 频率
- 发送 CMD7 来选择卡
- 配置 DMA2，如下所示:
  - 使能 DMA2 控制器并将所有挂起的中断清零。
  - 使用存储器位置的基址对 DMA2_Stream3 或 DMA2_Stream6 Channel4 源地址寄存器进行编程，并使用 SDIO_FIFO 寄存器地址对 DMA2_Stream3 或 DMA2_Stream6 Channel4 目标地址寄存器进行编程。
  - 对 DMA2_Stream3 或 DMA2_Stream6 Channel4 控制寄存器进行编程（存储器递增、非外设递增，外设和源宽度是字）。
  - 对 DMA2_Stream3 或 DMA2_Stream6 Channel4 进行编程以选择外设作为流控制器（在

DMA_S3CR 或 DMA_S6CR 配置寄存器中将 PFCTRL 位置 1)。

- 在 DMA2_Stream3 或 DMA2_Stream6 Channel4 中，将递增突发传输配置为 4 个节拍（至少来自于外设端)。
- 使能 DMA2_Stream3 或 DMA2_Stream6 Channel4。

- 发送 CMD24 (WRITE_BLOCK)，如下所示：
  - 对 SDIO 数据长度寄存器进行编程（在卡识别过程之前，SDIO 数据计时器寄存器应该已编程)。
  - 使用数据传输目标卡的地址位置对 SDIO 参数寄存器进行编程。
  - 对 SDIO 命令寄存器进行编程：CmdIndex 为 24 (WRITE_BLOCK)；WaitResp 为“1”（SDIO 卡主机等待响应)；CPSMEN 为“1”（使能 SDIO 卡主机，以发送命令)。其它字段是其各自的复位值。
  - 等待 SDIO_STA[6] = CMDREND 中断，然后对 SDIO 数据控制寄存器进行编程： DTEN 为“1”（使能 SDIO 卡主机以发送数据)；DTDIR 为“0”（从控制器到卡)；DTMODE 为“0”（块数据传输)；DMAEN 为“1”（使能 DMA)；DBLOCKSIZE 为 0x9（512 字节)。其它字段无关。
  - 等待 SDIO_STA[10] = DBCKEND。
- 轮询 DMA 通道使能状态寄存器，以确认没有任何通道仍处于使能状态。

## 23.4. 卡功能说明

### 23.4.1. 卡识别模式

在卡识别模式下, 主机复位所有卡, 验证运行电压范围, 识别卡并在总线上为每个卡设置相对卡地址 (RCA)。卡识别模式中的所有数据通信都仅使用命令行 (CMD)。

### 23.4.2. 智能卡复位

GO_IDLE_STATE 命令 (CMD0) 是软件复位命令，它会将多媒体卡和 SD 存储器置于空闲状态。IO_RW_DIRECT 命令 (CMD52) 可复位 SD I/O 卡。在上电或 CMD0 之后，所有卡输出总线驱动程序都处于高阻态状态，并且这些卡将使用默认的相对卡地址 (RCA=0x0001) 和默认的驱动程序阶段寄存器设置（最低速度，最高驱动电流容量）进行初始化。

### 23.4.3. 工作电压范围验证

所有卡都可以使用规范范围内的任何工作电压与 SDIO 卡主机进行通信。卡上的运行条件寄存器 (OCR) 定义了支持的最小和最大 VDD 值。

在有效负载存储器中存储卡标识号 (CID) 和卡特定数据 (CSD) 的卡仅能够在数据传输 VDD 条件下传播此信息。当 SDIO 卡主机模块和卡具有不兼容的 VDD 范围时，卡无法完成识别周期且无法发送 CSD 数据。因此，SEND_OP_COND (CMD1)、SD_APP_OP_COND（用于 SD 存储器的 ACMD41）和 IO_SEND_OP_COND（用于 SD I/O 的 CMD5）等特殊命令旨在提供一种机制，用于识别和拒绝与 SDIO 卡主机要求的 VDD 范围不匹配的卡。SDIO 卡主机会发送所需的 VDD 电压窗口作为这些命令的运算对象。无法在指定范围内执行数据传输的卡将断开与总线的连接，并且变为无效状态。

如果在使用这些命令时不将电压范围作为运算对象包括在内，则 SDIO 卡主机可能会查询每个卡并确定通用电压范围，然后再将超出范围的卡置于无效状态。当 SDIO 卡主机能够选择通用电压范围或者当用户要求就卡无法使用这一情况进行通知时，将会使用此查询。

### 23.4.4. 卡识别过程

多媒体卡和 SD 卡的卡识别过程有所不同。对于多媒体卡，识别过程是以时钟速率 $F_{od}$ 开始。SDIO_CMD 线输出驱动器是开漏引脚，在此识别过程中允许并行的卡操作。对准过程以如下方式完成：

- 激活总线。
- SDIO 卡主机广播 SEND_OP_COND (CMD1) 以接收运行条件。
- 响应是来自所有卡的运行条件寄存器的线与运算。
- 不兼容的卡将被置于无效状态。
- SDIO 卡主机向所有有效卡广播 ALL_SEND_CID (CMD2)。
- 有效卡同时以串行方式发送其 CID 号。如果卡的输出 CID 位与命令线上的位不匹配，则卡将停止传输，必须等待下一个识别周期。一个卡成功地将完整 CID 传输到 SDIO 卡主机并进入识别状态。
- SDIO 卡主机向该卡发出 SET_RELATIVE_ADDR (CMD3)。这一新地址称为相对卡地址 (RCA)；它比 CID 更短，可对卡进行寻址。分配的卡变为待机状态，它不会对进一步的识别周期进行响应，并且其输出将从开漏切换为推挽。
- SDIO 卡主机不断重复步骤 5 到 7，直到收到超时条件为止。

对于 SD 卡，识别过程是以时钟速率 $F_{od}$ 开始，并且 SDIO_CMD 线输出驱动是推挽驱动器，而非开漏引脚。对准过程以如下方式完成：

- 激活总线。
- SDIO 卡主机广播 SD_APP_OP_COND (ACMD41)。
- 卡以其运行条件寄存器的内容进行响应。
- 不兼容的卡将被置于无效状态。
- SDIO 卡主机向所有有效卡广播 ALL_SEND_CID (CMD2)。
- 这些卡将发回其唯一的卡识别号 (CID) 并进入识别状态。
- SDIO 卡主机向某个地址的有效卡发出 SET_RELATIVE_ADDR (CMD3)。这一新地址称为相对卡地址 (RCA)；它比 CID 更短，可对卡进行寻址。分配的卡变为待机状态。SDIO 卡主机可以重新发出此命令以更改 RCA。卡的 RCA 是最后分配的值。
- SDIO 卡主机对所有有效卡重复步骤 5 到 7。

对于 SDI/O 卡，对准过程以如下方式完成：

- 激活总线。
- SDIO 卡主机发送 IO_SEND_OP_COND (CMD5)。
- 卡以其运行条件寄存器的内容进行响应。
- 不兼容的卡将被设置无效状态。
- SDIO 卡主机向某个地址的有效卡发出 SET_RELATIVE_ADDR (CMD3)。这一新地址称为相对卡地址 (RCA)；它比 CID 更短，可对卡进行寻址。分配的卡变为待机状态。SDIO 卡主机可以重新发出此命令以更改 RCA。卡的 RCA 是最后分配的值。

### 23.4.5. 块写入

在块写入 (CMD24 - 27) 期间，一个或多个数据块从主机传输到卡，并且由主机在每个块的结尾追加一个 CRC。支持块写入的卡始终可以接受 WRITE_BL_LEN 所定义的数据块。如果 CRC 失败，卡将在 SDIO_D 线上指示失败，已传输的数据将废弃而不会写入，并且将忽略所有进一步传输的块（在多块写入模式中）。如果主机使用不完整的块（其累计长度没有进行块对齐）并且不允许块偏离（未设置 CSD 参数

WRITE_BLK_MISALIGN），则卡会在第一个偏离块之前检测块偏离错误。（ADDRESS_ERROR 错误位在状态寄存器中设置）。如果主机尝试在写保护区域进行写入，也将会中止写入操作。但是，在此情况下，卡会将 WP_VIOLATION 位置 1。

对 CID 和 CSD 寄存器的编程不需要先前的块长度设置。传输的数据也受到 CRC 保护。如果 CSD 或 CID 寄存器的某一部分存储在 ROM 中，则这一无法更改的部分必须与接收缓冲器的对应部分相匹配。如果不匹配，则卡将报告错误并且不会更改任何寄存器内容。某些卡可能需要很长且无法预测的时间来写入数据块。收到数据块并完成 CRC 校验后，卡开始进行写入，如果卡的写入缓冲器已满并且无法接受来自新 WRITE_BLOCK 命令的新数据，则会保持 SDIO_D 线的低电平。主机可能随时使用 SEND_STATUS 命令 (CMD13) 轮询卡状态，而卡将使用其状态进行响应。READY_FOR_DATA 状态位指示卡是否可以接受新数据以及写入进程是否仍在进行中。主机可以通过发出 CMD7（用于选择不同的卡）来取消选择卡，这会将卡置于断开连接状态并释放 SDIO_D 线，但不会中断写入操作。再次选择卡时，如果编程仍在进行中并且写入缓冲器不可用，则卡会通过将 SDIO_D 拉到低电平来重新激活忙碌指示。

### 23.4.6. 块读取

在块读取模式中，数据传输的基本单位是块，其最大大小在 CSD (READ_BL_LEN) 中定义。如果将 READ_BL_PARTIAL 置 1，则还可以传输更小的块，这些块的开始和结束地址全部包含在一个物理块中（如 READ_BL_LEN 所定义）。在每个块的结尾会追加一个 CRC，以确保数据传输完整性。CMD17 (READ_SINGLE_BLOCK) 会启动块读取，在完成传输后，卡将返回到传输状态。

CMD18 (READ_MULTIPLE_BLOCK) 会启动多个连续块的传输。

主机在多个块操作中可以随时中止读取，无论其类型如何。通过发送停止传输命令可中止传输。

在多个块读取操作期间（全部两种类型），如果卡检测到错误（例如，超出范围、地址未对齐或内部错误），卡将停止数据传输并保持处于数据状态。然后，主机必须通过发送停止传输命令来中止操作。读取错误将在停止传输命令的响应中报告。

如果主机在卡以预定义的块数量传输了多个块操作的最后一个块之后发送了停止传输命令，则会以非法命令来响应它，因为卡已经不再处于数据状态。如果主机使用不完整的块（其累计长度没有进行块对齐）并且不允许块偏离，则卡会在第一个偏离块的开头检测块偏离错误条件（在状态寄存器中 ADDRESS_ERROR 错误位置 1）。

### 23.4.7. 流访问、流写入和流读取（仅限多媒体卡）

在流模式中，数据以字节进行传输，并且不会在每个块的结尾追加 CRC。

#### 流写入(仅限多媒体卡)

WRITE_DAT_UNTIL_STOP (CMD20) 会启动从 SDIO 卡主机到卡的数据传输，从指定地址开始并继续，直到 SDIO 卡主机发出停止命令为止。如果允许不完整的块（CSD 参数 WRITE_BL_PARTIAL 置 1），则数据库可以在卡地址空间内的任何地址开始和停止，否则只能在时钟边界处开始和停止。由于未提前确定要传输的数据量，因此无法使用 CRC。如果在发送数据时达到了存储器范围的结尾并且 SD 卡主机未发出停止命令，则将丢弃已传输的任何多余的数据。

使用以下特定于卡的数据寄存器等式字段，可确定流写入操作的最大时钟频率：

$$\mathbf{Maximumspeed} = \mathbf{MIN}(\mathbf{TRANSPEED}, \frac{(\mathbf{8} \times \mathbf{2}^{\mathbf{writebllen}})(-\mathbf{NSAC})}{\mathbf{TACC} \times \mathbf{R2WFACTOR}})$$

- Maximumspeed = 最大写入频率
- TRANSPEED = 最大数据传输速率
- writebllen = 写入数据块最大长度

- NSAC = CLK 周期中数据读取访问时间 2
- TAAC = 数据读取访问时间 1
- R2WFACTOR = 写入速度系数

如果主机尝试使用更高的频率，卡可能无法处理数据并停止编程，在状态寄存器中将 OVERRUN 错误位置 1，同时忽略所有进一步数据传输，并在接收数据状态下等待停止命令。如果主机尝试在写保护区域进行写入，也将会中止写入操作。但是，在此情况下，卡会将 WP_VIOLATION 位置 1。

**流读取(仅限多媒体卡)**

READ_DAT_UNTIL_STOP (CMD11) 控制面向流的数据传输。

此命令指示卡从指定地址开始发送其数据，直到 SDIO 卡主机发送 STOP_TRANSMISSION (CMD12) 为止。由于命令是串行传输，因此停止命令在执行时会有延迟，数据传输会在停止命令的结束位之后停止。如果在发送数据时达到了存储器范围的结尾并且 SDIO 卡主机未发出停止命令，则发送的任何后续数据都被视为未定义。

使用以下特定于卡的数据寄存器等式字段，可确定流读取操作的最大时钟频率。

$$\mathbf{Maximumspeed} = \mathbf{MIN}(\mathbf{TRANSPEED}, \frac{(\mathbf{8} \times \mathbf{2^{readbllen}})(-\mathbf{NSAC})}{\mathbf{TACC} \times \mathbf{R2WFACTOR}})$$

- Maximumspeed = 最大读取频率
- TRANSPEED = 最大数据传输速率
- readbllen = 读取数据块最大长度
- writebllen = 写入数据块最大长度
- NSAC = CLK 周期中数据读取访问时间 2
- TAAC = 数据读取访问时间 1
- R2WFACTOR = 写入速度系数

如果主机尝试使用更高的频率，则卡无法持续进行数据传输。如果发生此情况，卡会在状态寄存器中将 UNDERRUN 错误位置 1，中止传输并在数据状态下等待停止命令。

### 23.4.8. 擦除：组擦除和扇区擦除

多媒体卡的可擦除单元是擦除组。擦除组以写入块来测量，这些写入块是卡的基本可写入单元。擦除组的大小是特定于卡的参数，在 CSD 中定义。

主机可以擦除某一连续范围内的擦除组。启动擦除是一个三步的过程。

首先，主机使用 ERASE_GROUP_START (CMD35) 命令定义范围的起始地址，然后使用 ERASE_GROUP_END (CMD36) 命令定义范围的最终地址，最后通过发出 ERASE (CMD38) 命令来启动擦除过程。擦除命令中的地址字段是以字节为单位的擦除组地址。卡将忽略小于擦除组大小的所有 LSB，从而有效地将地址向下取整为擦除组边界。

如果未按顺序接收擦除命令，则卡会在状态寄存器中将 ERASE_SEQ_ERROR 位置 1 并复位整个序列。

如果收到了一个乱序命令（两个都不是擦除命令，但 SEND_STATUS 除外），则卡会在状态寄存器中将 ERASE_RESET 状态位置 1、复位擦除序列并执行最后的命令。

如果擦除范围包括写保护块，则它们将保留不变，仅会擦除不受保护的块。状态寄存器中的 WP_ERASE_SKIP 状态位将会置 1。

卡通过将 SDIO_D 保持低电平来指示擦除正在进行中。实际擦除时间可能很长，主机可以发出 CMD7 来取消选择卡。

### 23.4.9. 宽总线选择或取消选择

使用 SET_BUS_WIDTH (ACMD6) 来选择或取消选择宽总线（4 位总线宽度）操作模式。加电或 GO_IDLE_STATE (CMD0) 后，默认总线宽度是 1 位。SET_BUS_WIDTH (ACMD6) 仅在传输状态下有效，这意味着只有在通过 SELECT/DESELECT_CARD (CMD7) 选择了卡之后才能更改总线宽度。

### 23.4.10. 保护管理

SDIO 卡主机模块中支持三种针对卡的写保护方法。

- 卡内部写保护；
- 机械写保护开关(仅由 SDIO 卡主机模块负责)；
- 密码保护的卡锁定操作。

**卡内部写保护**

可以防止对卡数据进行写入和擦除。通过在 CSD 中设置永久或临时写保护位，制造商或内容供应商可以将整个卡永久进行写保护。如果卡支持对扇区组进行写保护（通过在 CSD 中将 WP_GRP_ENABLE 位置 1），可以保护部分数据，并且写保护可以由应用程序进行更改。写保护是以 CSD 中指定的 WP_GRP_SIZE 扇区为单位。SET_WRITE_PROT 和 CLR_WRITE_PROT 命令控制编址组的保护。SEND_WRITE_PROT 命令类似于单个块读取命令。卡将发送一个数据块，其中包含 32 个写保护位（表示从指定地址开始的 32 个写保护组），后跟 16 个 CRC 位。写保护命令中的地址字段是以字节为单位的组地址。

卡将忽略小于组大小的所有 LSB。

**机械写保护开关**

一个位于卡侧面的机械拨片，用户可以利用它来设置或清除卡上的写保护。当拨片位于窗口打开的位置时，卡受到写保护，当窗口关闭时，可以更改卡内容。插座一侧的匹配开关向 SDIO 卡主机模块表明，卡受到写保护。SDIO 卡主机模块负责保护卡。卡的内部电路并不知道写保护开关的位置。

**密码保护**

密码保护功能使 SDIO 卡主机模块可以使用密码来锁定和解锁卡。密码存储在 128 位 PWD 寄存器中，其大小在 8 位 PWD_LEN 寄存器中设置。这些寄存器具有非易失性，因此断电不会擦除它们。锁定的卡将响应和执行某些命令。这意味着 SDIO 卡主机模块可以进行复位、初始化、选择以及查询状态，但是不允许访问卡的数据。设置了密码后（由 PWD_LEN 的非零值来指示），卡将在上电后自动锁定。与 CSD 和 CID 寄存器的写命令一样，仅在传输状态中才可以使用锁定/解锁命令。在此情况下，该命令不包括地址参数，必须首先选择卡然后才能使用。卡锁定/解锁命令具备常规单一块写入命令的结构和总线事务类型。传输的数据块包括命令的所有必需信息（密码设置模式、PWD 本身以及卡锁定/解锁）。命令数据块大小由 SDIO 卡主机模块在发送卡锁定/解锁命令之前定义。

位设置如下所示：

- ERASE：设置后会强制执行擦除操作。所有其它位都必须为零，并且仅发送命令字节；
- LOCK_UNLOCK：设置后会锁定卡。LOCK_UNLOCK 可以与 SET_PWD 一起设置，但不能与 CLR_PWD 一起设置；
- CLR_PWD：设置后会清除密码数据；
- SET_PWD：设置后会将密码数据保存到存储器；

- PWD_LEN：它定义密码的长度（以字节为单位）；
- PWD：密码（新的或当前使用的，取决于命令）；

以下各节列出了有关设置/复位密码、锁定/解锁卡以及强制擦除的命令。

**设置密码**

- 如果未选择任何卡，请选择一个 (SELECT/DESELECT_CARD, CMD7)。
- 定义要发送的块长度 (SET_BLOCKLEN, CMD16)（由 8 位卡锁定/解锁模式来确定）、 8 位 PWD_LEN 以及新密码的字节数。完成了命令替换后，块大小必须考虑到如下情况：无论是旧密码还是新密码都会与命令一起发送。
- 在数据线上，与相应的数据块大小一起发送 LOCK/UNLOCK (CMD42)，包括 16 位 CRC。数据块指示模式 (SET_PWD = 1)、长度 (PWD_LEN) 和密码自身 (PWD)。完成密码替换后，长度值 (PWD_LEN) 会包括新旧两个密码的长度，PWD 字段包括旧密码（当前使用的）后跟新密码。
- 如果密码匹配，则新密码及其大小会分别保存到 PWD 和 PWD_LEN 字段。如果发送的旧密码与预期的密码不符（大小和/或内容），会在卡状态寄存器中将 LOCK_UNLOCK_FAILED 错误位置 1，而密码不会更改。

密码长度字段 (PWD_LEN) 指示当前是否设置了密码。如果此字段非零，则表示设置了密码并且卡在上电后锁定自身。通过将 LOCK_UNLOCK 位置 1（在设置密码时）或通过发送用于卡锁定的其它命令，可以在当前上电会话中立即锁定卡。

**复位密码**

- 如果未选择任何卡，请选择一个 (SELECT/DESELECT_CARD, CMD7)。
- 定义要发送的块长度 (SET_BLOCKLEN, CMD16)（由 8 位卡锁定/解锁模式来确定）、8 位 PWD_LEN 以及当前使用的密码的字节数。
- 在数据线上，与相应的数据块大小一起发送 LOCK/UNLOCK (CMD42)，包括 16 位 CRC。数据块指示模式 (CLR_PWD = 1)、长度 (PWD_LEN) 和密码自身 (PWD)。将忽略 LOCK_UNLOCK 位。
- 如果密码匹配，会清除 PWD 字段并将 PWD_LEN 置 0。如果发送的密码与预期的密码不符（大小和/或内容），会在卡状态寄存器中将 LOCK_UNLOCK_FAILED 错误位置 1，而密码不会更改。

**锁定卡**

- 如果未选择任何卡，请选择一个 (SELECT/DESELECT_CARD, CMD7)。
- 定义要发送的块长度 (SET_BLOCKLEN, CMD16)、8 位 PWD_LEN 以及当前密码的字节数。
- 在数据线上，与相应的数据块大小一起发送 LOCK/UNLOCK (CMD42)，包括 16 位 CRC。数据块指示模式 (LOCK_UNLOCK = 1)、长度 (PWD_LEN) 和密码自身 (PWD)。
- 如果密码匹配，则卡被锁定并且卡状态寄存器中 CARD_IS_LOCKED 状态位会置 1。如果发送的密码与预期的密码不符（大小和/或内容），会在卡状态寄存器中将 LOCK_UNLOCK_FAILED 错误位置 1，且锁定会失败。

可以按照相同的顺序来设置密码和锁定卡。在此情况下，SDIO 卡主机模块会执行用于设置密码的所有必需步骤，但是发送新密码命令时，需要在步骤 3 中将 LOCK_UNLOCK 位置 1。

如果之前设置过密码（PWD_LEN 不为 0），则卡在上电复位后会自动锁定。尝试锁定已锁定的卡或者尝试锁定没有密码的卡都会失败，且会将卡状态寄存器 LOCK_UNLOCK_FAILED 错误位置 1。

**解锁卡**

- 如果未选择任何卡，请选择一个 (SELECT/DESELECT_CARD, CMD7)。
- 定义要发送的块长度 (SET_BLOCKLEN, CMD16)（由 8 位卡锁定/解锁模式来确定）、8 位 PWD_LEN 以及当前密码的字节数。
- 在数据线上，与相应的数据块大小一起发送 LOCK/UNLOCK (CMD42)，包括 16 位 CRC。数据块指示模式 (LOCK_UNLOCK = 0)、长度 (PWD_LEN) 和密码自身 (PWD)。
- 如果密码匹配，则卡被解锁并且卡状态寄存器中 CARD_IS_LOCKED 状态位会清零。如果发送的密码在大小和 / 或内容方面不正确并且与预期的密码不符，会在卡状态寄存器中将 LOCK_UNLOCK_FAILED 错误位置 1，且卡保持锁定。

解锁功能仅对当前上电会话有效。如果 PWD 字段未清零，则卡会在下一次上电时自动锁定。

尝试解锁已解锁的卡会导致失败，并且会在卡状态寄存器中将 LOCK_UNLOCK_FAILED 错误位置 1。

**强制擦除**

如果用户忘记了密码（PWD 内容），可以在清除卡上所有数据之后访问卡。这种强制擦除操作将擦除所有卡数据和所有密码数据。

- 如果未选择任何卡，请选择一个 (SELECT/DESELECT_CARD, CMD7)。
- 将块长度 (SET_BLOCKLEN, CMD16) 设置为 1 个字节。仅发送 8 位卡锁定/解锁字节。
- 在数据线上，与相应的数据字节一起发送 LOCK/UNLOCK (CMD42)，包括 16 位 CRC。数据块指示模式 (ERASE = 1)。所有其它位都必须为零。
- 如果数据字段中仅设置了 ERASE 位，会擦除所有卡内容，包括 PWD 和 PWD_LEN 字段，并且卡不再锁定。如果设置了任何其它位，则会在卡状态寄存器中将 LOCK_UNLOCK_FAILED 错误位置 1，而卡会保留其所有数据并保持锁定。

尝试在锁定的卡上使用强制擦除会导致失败，且会将卡状态寄存器 LOCK_UNLOCK_FAILED 错误位置 1。

### 23.4.11. 卡状态寄存器

响应格式 R1 包含一个由 32 位字段命名的卡状态。此字段旨在将卡状态信息（可能存储在本地状态寄存器中）传输到主机。除非另有说明，否则状态条目始终与先前发出的命令有关。

下表定义了状态的不同条目。表中的类型和清除条件字段以如下方式缩写：

类型：

- E：错误位
- S：状态位
- R：已检测到并已针对实际命令响应进行了设置
- X：已检测到并在命令执行期间进行了设置。SDIO 卡主机要对卡进行轮询必须发出状态命令来读取这些位。

清除条件：

- A：根据卡的当前状态
- B：始终与先前命令有关。收到有效命令后会将其清零（延迟一个命令）
- C：通过读取清零

表 23-9 卡状态

| 位 | 标识符 | 类型 | 值 | 说明 | 清除条件 |
|---|---|---|---|---|---|
| 31 | ADDRESS_ OUT_OF_RANGE | ERX | “0”= 无错误<br>“1”= 有错误 | 命令地址参数超出了此卡允许的范围。<br>多个块或流读取/写入操作正在尝试进行超出卡容量的读取或写入（尽管是从有效地址开始）。 | C |
| 30 | ADDRESS_MISALIGN | | “0”= 无错误<br>“1”= 有错误 | 命令地址参数（与当前设置的块长度一致）会将第一个偏离的数据块定位到卡的物理块。<br>多个块读取/写入操作正在尝试读取或写入与卡的物理块未对齐的数据块（尽管是以有效的地址/块长度组合开始） | C |
| 29 | BLOCK_LEN_ERROR | | “0”= 无错误<br>“1”= 有错误 | 可能是 SET_BLOCKLEN 命令的参数超过了卡的最大允许值，或者先前定义的块长度对当前命令而言非法（例如，主机发出了写入命令，而当前块长度小于卡的最大允许值并且不允许写入不完整的块）。 | C |
| 28 | ERASE_SEQ_ERROR | | “0”= 无错误<br>“1”= 有错误 | 擦除命令序列中发生错误。 | C |
| 27 | ERASE_PARAM | EX | “0”= 无错误<br>“1”= 有错误 | 为擦除选择的擦除组无效。 | C |
| 26 | WP_VIOLATION | EX | “0”= 无错误<br>“1”= 有错误 | 尝试对写保护块进行编程。 | A |
| 25 | CARD_IS_LOCKED | SR | “0”= 表示卡已解锁<br>“1”= 卡已锁定 | 如果设置，则表示卡已由主机锁定 | C |
| 24 | LOCK_UNLOCK_ FAILED | EX | “0”= 无错误<br>“1”= 有错误 | 如果在锁定/解锁卡命令中检测到序列或密码错误，则会设置 | B |
| 23 | COM_CRC_ERROR | ER | “0”= 无错误<br>“1”= 有错误 | 先前命令的 CRC 校验失败。 | B |
| 22 | ILLEGAL_COMMAND | ER | “0”= 无错误<br>“1”= 有错误 | 命令对于卡状态不合法 | B |
| 21 | CARD_ECC_FAILED | EX | “0”= 成功<br>“1”= 失败 | 应用了卡内部 ECC，但未能更正数据。 | C |
| 20 | CC_ERROR | | “0”= 无错误<br>“1”= 有错误 | （标准中未定义）发生了卡错误，该错误与主机命令无关。 | C |
| 19 | ERROR | | “0”= 无错误<br>“1”= 有错误 | （标准中未定义）与最后一个主机命令的执行（以及在此期间的检测）有关的一般卡错误（例如，读取或写入失败）。 | C |
| 18 | 保留 | | | | |
| 17 | 保留 | | | | |
| 16 | CID/CSD_OVERWRITE | EX | “0”= 无错误，<br>“1”= 有错误 | 可以是以下错误之一：<br>– CID 寄存器已写入且无法覆盖<br>– CSD 的只读部分与卡内容不匹 | C |

| 位 | 标识符 | 类型 | 值 | 说明 | 清除条件 |
|---|---|---|---|---|---|
| | | | | 配<br>– 尝试恢复已设置的副本（设置为原始值）或永久 WP（未受保护）位 | |
| 15 | WP_ERASE_SKIP | EX | “0”= 未受保护<br>“1”= 受保护 | 仅当由于现有写入而导致部分地址空间被擦除时才会设置 | C |
| 14 | CARD_ECC_DISABLED | SX | “0”= 使能<br>“1”= 禁用 | 命令已在未使用内部 ECC 的情况下执行 | A |
| 13 | ERASE_RESET | | “0”= 已清零<br>“1”= 已置 1 | 在执行之前已清除了擦除序列，因为收到的不是擦除序列命令（CMD35、CMD36、CMD38 或 CMD13 之外的命令） | C |
| 12:9 | CURRENT_STATE | SR | 0 = 空闲<br>1 = 就绪<br>2 = 识别<br>3 = 待机<br>4 = 传输<br>5 = 数据<br>6 = 接收<br>7 = 进行中<br>8 = 断开<br>9 = 突发<br>10-15 = 保留 | 接收命令时卡的状态。如果命令执行导致了状态变化，主机可以在下一个命令的响应中看到。这四个位将被解释为 0 到 15 之间的二进制数字。 | B |
| 8 | READY_FOR_DATA | SR | “0”= 未就绪，<br>“1”= 就绪 | 对应于总线上的缓冲器空信令 | |
| 7 | SWITCH_ERROR | EX | “0”= 无错误<br>“1”= 开关错误 | 如果设置，则卡不会根据 SWITCH 命令的请求切换到预期模式 | B |
| 6 | 保留 | | | | |
| 5 | APP_CMD | SR | “0”= 禁用<br>“1”= 使能 | 卡将预期收到 ACMD，或者命令已解释为 ACMD 的指示 | C |
| 4 | 保留，用于 SD I/O 卡 | | | | |
| 3 | AKE_SEQ_ERROR | ER | “0”= 无错误<br>“1”= 有错误 | 认证过程序列中出现错误 | C |
| 2 | 保留，用于应用程序特定的命令 | | | | |
| 1 | 保留，用于制造商测试模式 | | | | |

### 23.4.12. SD 状态寄存器

SD 状态包含与 SD 存储卡专有功能相关的状态位，在将来可用于特定于应用的场合中。SD 状态的大小是一个 512 位的数据块。如果发送了 ACMD13（CMD55，后面带有 CMD13），此寄存器的内容将传输到 SDIO 卡主机。ACMD13 只能发送到处于传输状态的卡（卡已被选定）。

下表定义了 SD 状态寄存器的不同条目。表中的类型和清除条件字段以如下方式缩写：

类型：

- E：错误位
- S：状态位

- R：已检测到并已针对实际命令响应进行了设置
- X：已检测到并在命令执行期间进行了设置。SDIO 卡主机要对卡进行轮询必须发出状态命令来读取这些位。

清除条件：

- A：根据卡的当前状态
- B：始终与先前命令有关。收到有效命令后会将其清零（延迟一个命令）
- C：通过读取清零

表 23-10 SD 状态

| 位 | 标识符 | 类型 | 值 | 说明 | 清除条件 |
|---|---|---|---|---|---|
| 511:510 | DAT_BUS_WIDTH | S R | “00” = 1（默认值）<br>“01” = 保留<br>“10” = 4 位宽度<br>“11” = 保留 | 显示了当前定义的数据总线<br>宽度（由 SET_BUS_WIDTH 命令定义） | A |
| 509 | SECURED_MODE | S R | “0” = 未处于安全模式<br>“1” = 处于安全模式 | 卡处于安全操作模式（请参<br>见“SD 安全规范”）。 | A |
| 508:496 | 保留 | | | | |
| 495:480 | SD_CARD_TYPE | S R | “00xxh” = SD 存储卡，按照物理规范版本 1.01-2.00 中所定义（“x” = 无关）。目前定义了以下卡：<br>“0000” = 常规 SD RD/WR 卡。<br>“0001” = SD ROM 卡 | 将来，8 个 LSB 将用于定义<br>SD 存储卡的不同变体（每个位将定义不同的 SD 类型）。<br>8 个 MSB 将用于定<br>义不符合当前 SD 物理层规范的 SD 卡。 | A |
| 479:448 | SIZE_OF_PROTECT ED_AREA | S R | 受保护区域的大小（请参见以下内容） | （请参见以下内容） | A |
| 447:440 | SPEED_CLASS | S R | 卡的速度等级（请参见以下内容） | （请参见以下内容） | |
| 439:432 | PERFORMANCE_MOVE | S R | 按（请参见以下内容）1 [MB/s] 步长表示的移动性能。 | （请参见以下内容） | A |
| 431:428 | AU_SIZE | S R | AU 大小（请参见以下内容） | （请参见以下内容） | A |
| 427:424 | 保留 | | | | |
| 423:408 | ERASE_SIZE | S R | 一次要擦除的 AU 数 | （请参见以下内容） | A |
| 407:402 | ERASE_TIMEOUT | S R | 擦除 UNIT_OF_ERASE_AU 指定<br>的区域时的超时值 | （请参见以下内容） | A |
| 401:400 | ERASE_OFFSET | S R | 为擦除时间增加的固定偏移值。 | （请参见以下内容） | A |
| 399:312 | 保留 | | | | |
| 311:0 | 保留，用于制造商 | | | | |

**SIZE_OF_PROTECTED_AREA**

此字段的设置在标准容量卡和高容量卡之间有所不同。如果是标准容量卡，受保护区域的容量计算方式如下：

- 受保护区域 = SIZE_OF_PROTECTED_AREA_* MULT * BLOCK_LEN；
- SIZE_OF_PROTECTED_AREA 是以 MULT*BLOCK_LEN 为单位来指定。

如果是高容量卡，受保护区域的容量在以下字段中指定：

- 受保护区域 = SIZE_OF_PROTECTED_AREA；
- SIZE_OF_PROTECTED_AREA 是以字节为单位来指定。

**SPEED_CLASS**

此 8 位字段指示速度等级，该值可以通过 PW/2 来计算（其中 PW 是写入性能）。

表 23-11 速度等级代码字段

| SPEED_CLASS | 值定义 |
|---|---|
| 00h | 等级 0 |
| 01h | 等级 2 |
| 02h | 等级 4 |
| 03h | 等级 6 |
| 04h — FFh | 保留 |

**PERFORMANCE_MOVE**

此 8 位字段指示 Pm（移动性能），该值可通过 1 [MB/s] 步长进行设置。如果卡不移动已使用的 RU（记录单元），则 Pm 应被视为无穷大。将该字段设置为 FFh 即表示无穷大。

表 23-12 移动性能字段

| PERFORMANCE_MOVE | 值定义 |
|---|---|
| 00h | 未定义 |
| 01h | 1 [MB/s] |
| 02h | 02h 2 [MB/s] |
| --------- | --------- |
| FEh | 254 [MB/s] |
| FFh | 无穷大 |

**AU_SIZE**

该 4 位域指示 AU 大小，其取值范围是 2 的 n 次方（起始值为 16KB）。

表 23-13 AU_SIZE 字段

| AU_SIZE | 值定义 |
|---|---|
| 00h | 未定义 |
| 01h | 16 KB |
| 02h | 32 KB |
| 03h | 64 KB |
| 04h | 128 KB |
| 05h | 256 KB |
| 06h | 512 KB |
| 07h | 1 MB |

| AU_SIZE | 值定义 |
|---|---|
| 08h | 2 MB |
| 09h | 4 MB |
| Ah – Fh | 保留 |

AU 的最大大小，具体取决于卡的容量（在下表中进行了定义）。可将卡的容量大小设置为介于 RU 大小和最大 AU 大小之间的任何 AU 大小。

表 23-14 最大 AU 大小

| 容量 | 16 MB-64 MB | 128 MB-256 MB | 512 MB | 1 GB-32 GB |
|---|---|---|---|---|
| 最大 AU 大小 | 512 KB | 1 MB | 2 MB | 4 MB |

**ERASE_SIZE**

该 16 位域指示 NERASE。擦除 AU 的 NERASE 数量后，将由 ERASE_TIMEOUT 指定超时值（请参见 ERASE_TIMEOUT）。主机应确定要在一个操作中擦除的正确 AU 数量，以便主机可以显示擦除操作的进度。如果将该字段设置为 0，则不支持擦除超时计算。

表 23-15 擦除大小字段

| ERASE_SIZE | 值定义 |
|---|---|
| 0000h | 不支持擦除超时计算。 |
| 0001h | 1 个 AU |
| 0002h | 2 个 AU |
| 0003h | 3 个 AU |
| --------- | --------- |
| FFFFh | 65535 个 AU |

**ERASE_TIMEOUT**

该 6 位域指示 TERASE，该值指示当擦除 ERASE_SIZE 所指定的多个 AU 时，与偏移量之间的擦除超时。ERASE_TIMEOUT 的取值范围最大可定义为 63 秒，卡制造商可以选择 ERASE_SIZE 和 ERASE_TIMEOUT 的任意组合，具体取决于实现情况。确定 ERASE_TIMEOUT 即确定了 ERASE_SIZE。

表 23-16 擦除超时字段

| ERASE_TIMEOUT | 值定义 |
|---|---|
| 0 | 不支持擦除超时计算。 |
| 1 | 1 [秒] |
| 2 | 2 [秒] |
| 3 | 3 [秒] |
| --------- | --------- |
| 63 | 63 [秒] |

**ERASE_OFFSET**

该 2 位域指示 TOFFSET，可选择四个值之一。如果 ERASE_SIZE 和 ERASE_TIMEOUT 字段设置为 0，则该字段无意义。

表 23-17 擦除偏移字段

| ERASE_OFFSET | 值定义 |
|---|---|
| 0h | 0 [秒] |
| 1h | 1 [秒] |
| 2h | 2 [秒] |

| ERASE_OFFSET | 值定义 |
| --- | --- |
| 3h | 3 [秒] |

### 23.4.13. SD I/O 模式

#### SD I/O 中断

为使 SD I/O 卡能够中断多媒体卡/SD 模块，在 SD 接口的一个引脚上提供了一个中断功能。引脚 8（当以 4 位 SD 模式运行时用作 SDIO_D1）可以向多媒体卡/SD 模块发出卡中断信号。每个卡或者卡中的功能都可以使用中断。SD I/O 中断为电平敏感型，这意味着，在多媒体卡/SD 模块识别出中断线并对其采取操作或者由于中断周期结束而使其失效之前，中断线必须保持有效状态（低电平）。在多媒体卡/SD 模块处理完中断后，将通过在 SD I/O 卡内部寄存器中的相应位中进行 I/O 写入来清除中断状态位。所有 SD I/O 卡的中断输出均为低电平有效，并且应用程序必须在所有数据线 (SDIO_D[3:0]) 上提供外部上拉电阻。只有在中断周期内，多媒体卡/SD 模块才会将引脚 8 (SDIO_D/IRQ) 的电平仅采样到中断检测器中。而在所有其他时候，多媒体卡/SD 模块将忽略此值。

中断周期适用于存储器和 I/O 操作。对于具有单个块的操作和多块数据传输，二者的中断周期定义有所不同。

#### SD I/O 挂起和恢复

在多功能 SD I/O 中或者在同时具备 I/O 和存储器功能的卡中，有多个设备（I/O 和存储器）共享对 MMC/SD 总线的访问权限。要在多个设备之间共享对 MMC/SD 模块的访问权限，SD I/O 和组合卡可以有选择性地实施“挂起/恢复”这一概念。如果卡支持挂起/恢复，则 MMC/SD 模块可以临时停止对某一功能或存储器的数据传输操作（挂起）以释放总线，从而 将总线用于其他功能或存储器的更高优先级传输。此更高优先级传输完成后，原始传输将从中断位置恢复（重新开始）。是否支持“挂起/恢复”功能具体取决于卡。要在 MMC/SD 总 线上执行挂起/恢复，MMC/SD 模块将执行以下步骤：

- 确定当前使用 SDIO_D [3:0] 线的功能；
- 请求将优先级较低或速度较慢的事务挂起；
- 等待事务挂起完成；
- 开始更高优先级的事务；
- 等待更高优先级事务完成；
- 恢复挂起的事务。

#### SD I/O 读取等待

可选的读取等待 (RW) 操作仅针对 SD 的 1 位和 4 位模式进行了定义。“读取等待”操作允许 MMC/SD 模块发出如下信号：卡正在读取多个寄存器 (IO_RW_EXTENDED, CMD53) 以临时停止数据传输，同时允许 MMC/SD 模块向 SD I/O 设备中的任何功能发送命令。要确定卡是否支持“读取等待”协议，MMC/SD 模块必须测试内部卡寄存器中的功能位。“读取等待”的计时是基于中断周期的。

### 23.4.14. 命令和响应

#### 应用程序特定的命令和常规命令

SD 卡主机模块系统旨在为各种应用程序类型提供一个标准接口。在此环境中，需要有特定的客户/应用程序功能。为实现这些功能，在标准中定义了两种类型的通用命令：应用程序特定的命令 (ACMD) 和常规命

令 (GEN_CMD)。

当卡接收到 APP_CMD (CMD55) 命令时，卡所需的下一个命令是应用程序特定的命令。ACMD 与常规多媒体卡命令的结构相同，且 CMD 编号也可以相同。卡将其识别 ACMD 的依据是：它出现在 APP_CMD (CMD55) 的后面。如果后面紧跟 APP_CMD (CMD55) 的命令不是已定义的应用程序特定的命令，则会使用标准命令。例如，如果卡具有 SD_STATUS (ACMD13) 的定义并且接收到了后跟 APP_CMD (CMD55) 的 CMD13，则此命令被解释为 SD_STATUS (ACMD13)。但是，如果卡接收到后跟 APP_CMD (CMD55) 的 CMD7，但是卡没有 ACMD7 的定义，则此命令被解释为标准的 (SELECT/DESELECT_CARD) CMD7。

要使用制造商特定的 ACMD，SD 卡主机必须执行以下步骤：

- 发送 APP_CMD (CMD55)。
- 卡对多媒体卡/SD 模块进行响应，指示已将 APP_CMD 位置 1，并指示当前需要的是 ACMD。
- 发送所需的 ACMD。
- 卡对多媒体卡/SD 模块进行响应，指示已将 APP_CMD 位置 1，并指示接受的命令被解释为 ACMD。如果发送非 ACMD，则卡会将其作为普通多媒体卡命令进行处理，卡状态寄存器中的 APP_CMD 位保持清零。

如果发送了无效命令（既不是 ACMD 也不是 CMD），则将作为标准的多媒体卡非法命令错误来处理。

GEN_CMD 的总线事务与单块读/写命令（WRITE_BLOCK, CMD24 或 READ_SINGLE_BLOCK,CMD17）的总线事务相同。在此情况下，参数表示数据传输的方向而非地址，数据块具有供应商特定的格式和含义。

发送 GEN_CMD (CMD56) 之前，必须选择卡（处于传输状态）。数据块大小由 SET_BLOCKLEN (CMD16) 定义。对 GEN_CMD (CMD56) 的响应为 R1b 格式。

**命令类型**

应用程序特定的命令和常规命令均具有以下四种类型：

- 广播命令 (BC)：发送到所有卡；不返回任何响应。
- 具有响应的广播命令 (BCR)：发送到所有卡；同时接收来自所有卡的响应。
- 编址（点到点）命令 (AC)：发送到选定的卡；不包含 SDIO_D 线上的数据传输。
- 编址（点到点）数据传输命令 (ADTC)：发送到选定的卡；包含 SDIO_D 线上的数据传输。

**命令格式**

请参见表 23-2 命令格式以了解命令格式。

**多媒体卡/SD 模块的命令**

表 23-18 面向块的写入命令

| CMD 索引 | 类型 | 参数 | 响应格式 | 缩写 | 说明 |
|---|---|---|---|---|---|
| CMD23 | ac | [31:16] 设置为 0<br>[15:0] 块数量 | R1 | SET_BLOCK_COUNT | 用于定义将在后面的多块读/写命令中传输的块数量。 |
| CMD24 | adtc | [31:0] 数据地址 | R1 | WRITE_BLOCK | 写入一个块，可通过 命令选择其大小。SET_BLOCKLEN |
| CMD25 | adtc | [31:0] 数据地址 | R1 | WRITE_MULTIPLE_BLOCK | 连续写入数据块，直至出现 STOP_TRANSMISSION，或者直至已接收到所请求的块数量。 |
| CMD26 | adtc | [31:0] 填充位 | R1 | PROGRAM_CID | 对卡标识寄存器进行编程。每个卡只能 |

| CMD 索引 | 类型 | 参数 | 响应格式 | 缩写 | 说明 |
|---|---|---|---|---|---|
| | | | | | 发出一次该命令。卡包含的硬件可防止在首次编程后执行此操作。通常该命令留给供制造商使用。 |
| CMD27 | adtc | [31:0] 填充位 | R1 | PROGRAM_CSD | 对 CSD 的可编程位进行编程。 |

表 23-19 面向块的写保护命令

| CMD 索引 | 类型 | 参数 | 响应格式 | 缩写 | 说明 |
|---|---|---|---|---|---|
| CMD28 | ac | [31:0] 数据地址 | R1b | SET_WRITE_PROT | 如果卡具有写保护功能，此命令会将编址组的写保护位置 1。写保护的属性以卡特定的数据 (WP_GRP_SIZE) 进行编码。 |
| CMD29 | ac | [31:0] 数据地址 | R1b | CLR_WRITE_PROT | 如果卡提供了写保护功能，此命令会将编址组的写保护位清零。 |
| CMD30 | adtc | [31:0] 写保护数据地址 | R1 | SEND_WRITE_PROT | 如果卡提供了写保护功能，此命令将要求卡发送写保护位的状态。 |
| CMD31 | 保留 | | | | |

表 23-20 擦除命令

| CMD 索引 | 类型 | 参数 | 响应格式 | 缩写 | 说明 |
|---|---|---|---|---|---|
| CMD32 ... CMD34 | 保留。不能使用这些命令索引来保持与较旧版本多媒体卡的向后兼容性。 | | | | |
| CMD35 | ac | [31:0] 数据地址 | R1 | ERASE_GROUP_START | 在要选择进行擦除的某个范围内设置第一个擦除组的地址。 |
| CMD36 | ac | [31:0] 数据地址 | R1 | ERASE_GROUP_END | 在要选择进行擦除的某个连续范围内设置最后一个擦除组的地址。 |
| CMD37 | 保留。不能使用此命令索引来保持与较旧版本多媒体卡的向后兼容性。 | | | | |
| CMD38 | ac | [31:0] 填充位 | R1 | ERASE | 擦除所有先前选择的写入块。 |

表 23-21 I/O 模式命令

| CMD 索引 | 类型 | 参数 | 响应格式 | 缩写 | 说明 |
|---|---|---|---|---|---|
| CMD39 | ac | [31:16] RCA<br>[15:15] 寄存器写入标志<br>[14:8] 寄存器地址<br>[7:0] 寄存器数据 | R4 | FAST_IO | 用于写入和读取 8 位（寄存器）数据字段。该命令用于对卡和寄存器寻址，如果将写入标志置 1，则提供要写入的数据。R4 响应包含从寻址寄存器中读取的数据。该命令可访问与应用程序相关、未在多媒体卡标准中定义的寄存器。 |
| CMD40 | bcr | [31:0] 填充位 | R5 | GO_IRQ_STATE | 使系统处于中断模式。 |
| CMD41 | 保留 | | | | |

表 23-22 锁定卡

| CMD 索引 | 类型 | 参数 | 响应格式 | 缩写 | 说明 |
|---|---|---|---|---|---|
| CMD42 | adtc | [31:0] 填充位 | R1b | LOCK_UNLOCK | 设置/重置密码或锁定/解锁卡。数据块的大小由 SET_BLOCK_LEN 命令设置。 |
| CMD43 ... CMD54 | 保留 | | | | |

表 23-23 应用程序特定的命令

| CMD 索引 | 类型 | 参数 | 响应格式 | 缩写 | 说明 |
|---|---|---|---|---|---|
| CMD55 | ac | [31:16] RCA [15:0] 填充位 | R1 | APP_CMD | 告知卡下一个命令位是应用程序特定的命令，而非标准命令 |
| CMD56 | adtc | [31:1] 填充位 [0]：RD/WR | | | 用于将数据块传输到卡，或从卡中获取数据块以用于常规命令/应用程序特定的命令。数据块的大小由 SET_BLOCK_LEN 命令设置。 |
| CMD57 ...CMD59 | 保留。 | | | | |
| CMD60 ...CMD63 | 留给制造商使用。 | | | | |

## 23.5. 响应格式

所有响应都是通过 MCCMD 命令线 SDIO_CMD 来发送。响应传输始终从对应于响应代码字的位字符串的左侧位开始。代码长度取决于响应类型。

响应始终从起始位（始终为 0）开始，后跟用于指示传输方向的位 (card = 0)。下表中标有 x 的值表示变量项。除 R3 响应类型之外的所有响应均受 CRC 保护。每个命令代码字由结束位（始终为 1）来终止。

有五种类型的响应。其格式定义如下:

### 23.5.1. R1(正常响应命令)

代码长度 = 48 位。45:40 位用于指示待响应命令的索引，该值被解释为二进制编码的数字（介于 0 和 63 之间）。卡的状态采用 32 位进行编码。

表 23-24 R1 响应

| 位的位置 | 宽度(位) | 值 | 说明 |
|---|---|---|---|
| 47 | 1 | 0 | 起始位 |
| 46 | 1 | 0 | 传输位 |
| [45:40] | 3 | X | 命令索引 |
| [39:8] | 32 | X | 卡状态 |
| [7:1] | 7 | X | CRC7 |
| 0 | 1 | 1 | 结束位 |

### 23.5.2. R1b

同 R1，并且有一个可选的繁忙信号在数据线上传输。根据卡在接收命令之前所处的状态，卡在接收到这些命令之后可能会变为繁忙状态。

### 23.5.3. R2(CID 和 CSD 寄存器)

代码长度 = 136 位。发送 CID 寄存器的内容以响应 CMD2 和 CMD10 命令。发送 CSD 寄存器的内容以响应 CMD9。仅传输 CID 和 CSD 的位 [127...1], 这些寄存器的保留位 [0] 将被替换为响应的结束位。卡通过将 MCDAT 保持低电平来指示擦除正在进行中。实际擦除时间可能很长，主机可以发出 CMD7 来取消选择卡。

表 23-25 R2 响应

| 位的位置 | 宽度(位) | 值 | 说明 |
|---|---|---|---|
| 135 | 1 | 0 | 起始位 |
| 134 | 1 | 0 | 传输位 |
| [133:128] | 6 | "111111" | 命令索引 |
| [127:1] | 127 | X | 卡状态 |
| 0 | 1 | 1 | 结束位 |

### 23.5.4. R3(OCR 寄存器)

代码长度: 48 位。发送 OCR 寄存器的内容以响应 CMD1。电平编码如下所示: 受限电压窗口 = 低电平, 卡繁忙 = 低电平。

表 23-26 R3 响应

| 位的位置 | 宽度(位) | 值 | 说明 |
|---|---|---|---|
| 47 | 1 | 0 | 起始位 |
| 46 | 1 | 0 | 传输位 |
| [45:40] | 6 | "111111" | 命令索引 |
| [39:8] | 32 | X | OCR 寄存器 |
| [7:1] | 7 | "1111111" | 保留 |
| 0 | 1 | 1 | 结束位 |

### 23.5.5. R4(快速 I/O)

代码长度：48 位。参数字段包含寻址卡的 RCA、要读出或写入的寄存器地址及其内容。

表 23-27 R4 响应

| 位的位置 | | 宽度(位) | 值 | 说明 |
|---|---|---|---|---|
| 47 | | 1 | 0 | 起始位 |
| 46 | | 1 | 0 | 传输位 |
| [45:40] | | 6 | "100111" | 命令索引 |
| [39:8]参数字段 | [31:16] | 32 | X | OCR 寄存器 |
| | [15:8] | 8 | X | 寄存器地址 |
| | [7:0] | 8 | X | 读取寄存器内容 |
| [7:1] | | 7 | "1111111" | 保留 |
| 0 | | 1 | 1 | 结束位 |

### 23.5.6. R4b

仅限 SD I/O：接收 CMD5 的 SDIO 卡将以唯一的 SDIO 响应 R4 进行响应。格式为:

表 23-28 R4b 响应

| 位的位置 | 宽度(位) | 值 | 说明 |
|---|---|---|---|
| 47 | 1 | 0 | 起始位 |
| 46 | 1 | 0 | 传输位 |

| 位的位置 | | 宽度(位) | 值 | 说明 |
|---|---|---|---|---|
| [45:40] | | 6 | X | 命令索引 |
| [39:8]参数字段 | 39 | 1 | X | 卡就绪 |
| | [38:36] | 3 | X | I/O 功能的数量 |
| | 35 | 1 | X | 存在存储器 |
| | [34:32] | 3 | X | 填充位 |
| | [31:8] | 24 | X | I/O OCR |
| [7:1] | | 7 | X | 保留 |
| 0 | | 1 | 1 | 结束位 |

一旦 SD I/O 卡接收到 CMD5，即会使能该卡的 I/O 部分以正常响应所有后续命令。I/O 卡中的此功能 I/O 使能将始终保持有效，直至卡接收到复位、电源重启或对 I/O 复位执行写入操作的 CMD52。请注意，仅具存储器的 SD 卡可对 CMD5 进行响应。仅具存储器的卡的正确响应将为：存在存储器 = 1， I/O 功能的数量 = 0。仅具存储器的卡（其设计标准符合 SD 存储卡规范版本 1.0）会将 CMD5 视为非法命令，并且不予响应。可感知 I/O 的主机将发送 CMD5。如果卡以响应 R4 进行响应，则主机会根据 R4 响应中包含的数据来确定卡的配置。

### 23.5.7. R5(中断请求)

仅用于多媒体卡。代码长度：48 位。如果响应由主机生成，则参数中的 RCA 字段将为 0x0。

表 23-29 R5 响应

| 位的位置 | | 宽度(位) | 值 | 说明 |
|---|---|---|---|---|
| 47 | | 1 | 0 | 起始位 |
| 46 | | 1 | 0 | 传输位 |
| [45:40] | | 6 | "101000" | CMD40 |
| [39:8] 参数字段 | [39:16] | 16 | X | 胜出的卡或主机的 RCA[31:16] |
| | [15:0] | 16 | X | 未定义，可用于 irq 数据 |
| [7:1] | | 7 | X | CRC7 |
| 0 | | 1 | 1 | 结束位 |

### 23.5.8. R6

仅用于 SD I/O。存储设备对 CMD3 的正常响应。

表 23-30 R6 响应

| 位的位置 | | 宽度(位) | 值 | 说明 |
|---|---|---|---|---|
| 47 | | 1 | 0 | 起始位 |
| 46 | | 1 | 0 | 传输位 |
| [45:40] | | 6 | "101000" | CMD40 |
| [39:8] 参数字段 | [39:16] | 16 | X | 胜出的卡或主机的 RCA[31:16] |
| | [15:0] | 16 | X | 未定义，可用于 irq 数据 |
| [7:1] | | 7 | X | CRC7 |
| 0 | | 1 | 1 | 结束位 |

当 CMD3 发送到 I/O 专用卡时，卡的状态位 [23:8] 将更改。在此情况下，响应的 16 位代表仅限 SD I/O 的值：

- 位 [15] COM_CRC_ERROR；
- 位 [14] ILLEGAL_COMMAND；

- 位 [13] ERROR;
- 位 [12:0] 保留。

## 23.6. SDIO I/O 卡专用操作

以下功能是专用于 SD I/O 的操作:

- 通过触发 SDIO_D2 执行的 SDIO 读取等待操作;
- 通过停止时钟执行的 SDIO 读取等待操作;
- SDIO 挂起/恢复操作（写入和读取挂起）;
- SDIO 中断。

仅当 SDIO_DCTRL[11] 位置 1 时，SDIO 才支持这些操作，但读取挂起操作除外，该操作不需要专门的硬件实现。

### 23.6.1. 通过触发 SDIO_D2 执行的 SDIO I/O 读取等待操作

可以在收到第一个数据块前启动读取等待间隔：如果使能数据路径(SDIO_DCTRL[0]位置 1)并 使能 SDIO 专用操作(SDIO_DCTRL[11] 位 置 1)，则当读取等待开始(SDI0_DCTRL[10] =0 且 SDI_DCTRL[8] =1）并且数据方向为从卡到 SDIO(SDIO_DCTRL[1] = 1) 时，DPSM 直接从空闲状态进入读取等待状态。在读取等待状态下, DPSM 在 2 个 SDIO_CK 时钟周期后将 SDIO_D2 驱动为 0。在此状态下, 如果将 RWSTOP 位 (SDIO_DCTRL[9]) 置 1，DPSM 保持等待状态的时间将多出两个 SDIO_CK 时钟周期, 以将 SDIO_D2 驱动为 1 并保持一个时钟周期（根据 SDIO 规范)。之后, DPSM 重新开始等待, 直至从卡中接收到数据。当 DPSM 接收到数据块时，即使读取等待开始位置 1，也不会启动读取等待间隔：读取等待间隔将在接收到 CRC 后启动。必须将 RWSTOP 位清零，才能开始新的读取等待操作。在读取等待间隔内，SDIO 可以在 SDIO_D1 上检测 SDIO 中断。

### 23.6.2. 通过停止 SDIO_CK 执行的 SDIO 读取等待操作

如果 SDIO 卡不支持上述读取等待方法，SDIO 可以通过停止 SDIO_CK 来执行读取等待（SDIO_DCTRL 置 1，这一点与上一节所述方法相同，但 SDIO_DCTRL[10] =1)：达到当前接收数据块的结束位后，DSPM 将在两个 SDIO_CK 周期后停止时钟，并在读取等待开始位置 1 后再次启动时钟。

SDIO_CK 停止时，可以向卡发出任何命令。在读取/等待间隔期间，SDIO 可以在 SDIO_D1 上检测 SDIO 中断。

### 23.6.3. SDIO 挂起/恢复操作

向卡发送数据时，SDIO 可以挂起写操作。SDIO_CMD[11] 位将置 1 并向 CPSM 指示当前命令为挂起命令。CPSM 将分析响应，一旦从卡接收到 ACK（接受挂起）信号，即确认 CPSM 在接收到当前数据块的 CRC 令牌后将变为空闲状态。

硬件不会保存要完成挂起的操作（恢复）还需发送的剩余数据块数量。

写操作可以由软件挂起，只需在从卡接收到挂起命令的 ACK 信号时禁止 DPSM (SDIO_DCTRL[0] =0)。DPSM 随即进入空闲状态。

要挂起读取操作：DPSM 将以 Wait_r 状态进行等待，因为要挂起的功能会在停止数据事务前发送一个完整的数据包。应用程序将继续读取 RxFIFO 直至 FIF0 为空，然后 DPSM 自动进入空闲状态。

### 23.6.4. SDIO 中断

一旦 SDIO_DCTRL[11] 位置 1，即会在 SDIO_D1 线路上检测到 SDIO 中断。

## 23.7. CE-ATA 专用操作

以下功能是专用于 CE-ATA 的操作：

- 向 CE-ATA 器件发送命令完成信号禁止；
- 从 CE-ATA 器件接收命令完成信号；
- 使用状态位和/或中断，向 CPU 发出 CE-ATA 命令完成信号。

SDIO 仅对 CE-ATA CMD61 命令支持这些操作，也就是说，应将 SDIO_CMD[14] 置 1。

### 23.7.1. 命令完成信号禁止

如果“使能 CMD 完成”位 SDIO_CMD[12] 未置 1，而“非中断使能”位 SDIO_CMD[13] 置 1，则在接收到短响应后，再经过 8 个位周期，将发送命令完成信号禁止。

CPSM 进入挂起状态，禁止序列“00001”将加载到命令移位寄存器中，43 将加载到命令计数器中。八个周期后，触发器将 CPSM 变为发送状态。当命令计数器达到 48 时，CPSM 将变为空闲状态，因为不再等待任何响应。

### 23.7.2. 命令完成信号使能

如果“使能 CMD 完成”位 SDIO_CMD[12] 置 1 并且“非中断使能”位 SDIO_CMD[13] 也置 1，则 CPSM 将以 Waitcpl 状态等待命令完成信号。

如果在 CMD 线路上接收到“0”，CPSM 将进入空闲状态。在 7 个位周期内将不能再发送任何新命令。在上述 7 个周期内，CMD 线路会在后 5 个周期以推挽模式驱动为“1”。

### 23.7.3. CE-ATA 中断

通过状态位 SDIO_STA[23] 向 CPU 发出命令完成信号。此静态位可通过清零位 SDIO_ICR[23]来清零。

SDIO_STA[23] 状态位可在每个中断线上产生一个中断，具体取决于屏蔽位 SDIO_MASKx[23]。

### 23.7.4. 中止 CMD61

如果尚未发送命令完成禁止信号并且需要中止 CMD61，则必须禁止命令状态机。CMD61 随后进入空闲状态，并且可发送 CMD12 命令。在此操作期间，不会发送任何命令完成禁止信号。

## 23.8. 硬件流控制

硬件流控制功能用于避免 FIFO 下溢（发送模式）和上溢（接收模式）错误。

该功能可停止 SDIO_CK 并冻结 SDIO 状态机。数据传输将停止，而 FIFO 无法发送或接收数据。只有由 SDIOCLK 提供时钟的状态机才会冻结，APB2 接口仍保持活动状态。因此，即使激活流控制，仍可填充或清空 FIFO。

要使能硬件流控制，SDIO_CLKCR[14] 寄存器位必须置 1。复位后，流控制将禁止。

## 23.9. SDIO 寄存器

器件通过 32 位控制寄存器(可通过 APB2 访问)与系统通信。

外设寄存器必须按字(32 位)进行访问。

### 23.9.1. SDIO 电源控制寄存器(SDIO_POWER)

偏移地址：0x00

复位值：0x00000000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:2 | 保留 | 必须保持复位值 |
| 位 1:0 | MODE | 电源控制位<br>这些位用于定义卡时钟的当前功能状态:<br>00：掉电：停止为卡提供时钟<br>01：保留<br>10：保留，上电<br>11：通电：为卡提供时钟 |

*注意：此寄存器的两次写访问至少需要相隔 7 个 HCLK 时钟周期。*

*注意：写入数据后，在三个 SDIOCK(48MHz)时钟周期以及 PCLK2 时钟周期内无法向此寄存器写入数据。*

### 23.9.2. SDIO 时钟控制寄存器(SDIO_CLKCR)

偏移地址：0x04

复位值：0x00000000

SDIO_CLKCR 寄存器控制 SDIO_CK 输出时钟。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | HWFC_EN | NEGEDGE | WIDBUS | | BYPASS | PWRSAV | CLKEN | CLKDIV | | | | | | | |
| | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:15 | 保留 | 必须保持复位值 |
| 位 14 | HWFC_EN | 硬件流控制使能<br>0：禁止硬件流控制 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 1：使能硬件流控制 |
| 位 13 | NEGEDGE | SDIO_CK 移相选择位<br>0：在主时钟 SDIOCLK 的上升沿产生 SDIO_CK<br>1：在主时钟 SDIOCLK 的下降沿产生 SDIO_CK |
| 位 12:11 | WIDBUS | 宽总线模式使能位<br>00：默认总线模式：使用 SDIO_D0<br>01：4 位宽总线模式：使用 SDIO_D[3:0]<br>10：8 位宽总线模式：使用 SDIO_D[7:0]<br>其他无意义 |
| 位 10 | BYPASS | 时钟分频器旁路使能位<br>0: 禁止旁路: 在驱动 SDIO_CK 输出信号前, 根据 CLKDIV 值对 SDIOCLK 进行分频。<br>1：使能旁路：SDIOCLK 直接驱动 SDIO_CK 输出信号。 |
| 位 9 | PWRSAV | 节能模式配置位<br>要实现节能模式, 可在总线空闲时通过将 PWRSAV 置 1 来禁止 SDIO_CK 时钟输出:<br>0: 始终使能 SDIO_CK 时钟<br>1: 仅在总线激活时使能 SDIO_CK |
| 位 8 | CLKEN | 时钟使能位<br>0：禁止 SDIO_CK<br>1：使能 SDIO_CK |
| 位 7:0 | CLKDIV | 时钟分频系数<br>该字段定义输入时钟 (SDIOCLK) 与输出时钟 (SDIO_CK) 之间的分频系数: SDIO_CK 频率 = SDIOCLK / [CLKDIV + 2]。 |

### 23.9.3. SDIO 参数寄存器(SDIO_ARG)

偏移地址：0x08

复位值：0x00000000

SDIO_ARG 寄存器包含一个 32 位命令参数，该参数作为命令消息的一部分发送到卡。

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:0 | CMDARG | 命令参数<br>作为命令消息的一部分发送给卡的命令参数。如果命令包含参数，则在将命令写入到命令寄 存器之前，必须将参数加载到此寄存器中。 |

### 23.9.4. SDIO 命令寄存器(SDIO_CMD)

偏移地址：0x0C

复位值：0x00000000

SDIO_CMD 寄存器包含命令索引和命令类型位。命令索引作为命令消息的一部分而发送给卡。命令类型位控制命令路径状态机 (CPSM)。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | ATACMD | nIEN | ENCMDcompl | SDIOSuspend | CPSMEN | WAITPEND | WAITINT |
| | rw | rw | rw | rw | rw | rw | rw |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WAITRESP | | CMDINDEX | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:15 | 保留 | 必须保持复位值 |
| 位 14 | ATACMD | CE-ATA 命令<br>如果 ATACMD 置 1，则 CPSM 将传输 CMD61 |
| 位 13 | nIEN | 非中断使能<br>如果该位为 0，则使能 CE-ATA 设备中的中断。 |
| 位 12 | ENCMDcompl | 使能 CMD 完成<br>如果此位置 1，则使能命令完成信号。 |
| 位 11 | SDIOSuspend | SD I/O 挂起命令<br>如果此位置 1，则要发送的命令为挂起命令（仅用于 SDIO 卡）。 |
| 位 10 | CPSMEN | 命令路径状态机 (CPSM) 使能位<br>如果此位置 1，则使能 CPSM。 |
| 位 9 | WAITPEND | CPSM 等待数据传输结束（CmdPend 内部信号）<br>如果此位置 1，则 CPSM 将等到数据传输结束后才开始发送命令。 |
| 位 8 | WAITINT | 必须保持复位值 |
| 位 7:6 | WAITRESP | 等待响应位<br>这些位用于配置 CPSM 是否等待响应，如果等待，将等待哪种类型的响应。<br>00：无响应，但 CMDSENT 标志除外<br>01：短响应，但 CMDREND 或 CCRCFAIL 标志除外<br>10：无响应，但 CMDSENT 标志除外<br>11：长响应，但 CMDREND 或 CCRCFAIL 标志除外 |
| 位 5:0 | CMDINDEX | 命令索引<br>命令索引作为命令消息的一部分发送给卡。 |

### 23.9.5. SDIO 命令响应寄存器(SDIO_RESPCMD)

偏移地址：0x20

复位值：0x00000000

SDIO_RESPCMD 寄存器包含接收到的最后一个命令响应的命令索引字段。如果命令响应传输不包含命令索引字段（长响应或 OCR 响应），则 RESPCMD 字段为未知，但该字段必须 包含 111111b（响应中保

留字段的值)。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:6 | 保留 | 必须保持复位值 |
| 位 5:0 | RESPCMD | 响应命令索引<br>只读位域。包含接收到的最后一个命令响应的命令索引。 |

### 23.9.6. SDIO 响应 1..4 寄存器(SDIO_RESPx)

偏移地址：(0x10 + (4 * x)), x = 1..4

复位值：0x00000000

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

卡状态为 32 位或 127 位，具体取决于响应类型。

| 寄存器 | 短响应 | 长响应 |
|---|---|---|
| SDIO_RESP1 | 卡状态[31:0] | 卡状态[127:96] |
| SDIO_RESP2 | 未使用 | 卡状态[95:64] |
| SDIO_RESP3 | 未使用 | 卡状态[63:32] |
| SDIO_RESP4 | 未使用 | 卡状态[31:1]0b |

首先接收卡状态的最高有效位。 SDIO_RESP3 寄存器 LSB 始终为 0b。

### 23.9.7. SDIO 数据定时寄存器(SDIO_DTIMER)

偏移地址：0x24

复位值：0x00000000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | DATATIME | 数据超时周期<br>以卡总线时钟周期表示的数据超时周期. |

### 23.9.8. SDIO 数据长度寄存器(SDIO_DLEN)

偏移地址：0x28

复位值：0x00000000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:25 | 保留 | 必须保持复位值 |
| 位 24:0 | DATALENGTH | 数据长度值<br>要传输的数据字节数量。 |

### 23.9.9. SDIO 数据控制寄存器(SDIO_DCTRL)

偏移地址：0x2C

复位值：0x00000000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | SDIOEN | RWMOD | RWSTOP | RWSTART | DBLOCKSIZE | | | | DMAEN | DTMODE | DTDIR | DTEN |
| | | | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:12 | 保留 | 必须保持复位值 |
| 位 11 | SDIOEN | SD I/O 使能功能<br>如果将该位置 1，则 DPSM 执行特定于 SD I/O 卡的操作。 |
| 位 10 | RWMOD | 读取等待模式<br>0：通过停止 SDIO_D2 进行读取等待控制<br>1：使用 SDIO_CK 进行读取等待控制 |
| 位 9 | RWSTOP | 读取等待停止<br>0：如果将 RWSTART 位置 1，则读取等待正在进行中<br>1：如果将 RWSTART 位置 1，则使能读取等待停止 |
| 位 8 | RWSTART | 读取等待开始<br>如果将该位置 1，则读取等待操作开始。 |
| 位 7:4 | DBLOCKSIZE | 数据块大小<br>定义在选择了块数据传输模式时数据块的长度： |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 0000：（十进制数 0）块长度 = 1 字节<br>0001：（十进制数 1）块长度 = 2 字节<br>0010：（十进制数 2）块长度 = 4 字节<br>0011：（十进制数 3）块长度 = 8 字节<br>0100：（十进制数 4）块长度 = 16 字节<br>0101：（十进制数 5）块长度 = 32 字节<br>0110：（十进制数 6）块长度 = 64 字节<br>0111：（十进制数 7）块长度 = 128 字节<br>1000：（十进制数 8）块长度 = 256 字节<br>1001：（十进制数 9）块长度 = 512 字节<br>1010：（十进制数 10）块长度 = 1024 字节<br>1011：（十进制数 11）块长度 = 2048 字节<br>1100：（十进制数 12）块长度 = 4096 字节<br>1101：（十进制数 13）块长度 = 8192 字节<br>1110：（十进制数 14）块长度 = 16384 字节<br>1111：（十进制数 15）保留 |
| 位 3 | DMAEN | DMA 使能位<br>0：禁止 DMA。<br>1：使能 DMA。 |
| 位 2 | DTMODE | 数据传输模式选择<br>0：块数据传输<br>1：流或 SDIO 多字节数据传输 |
| 位 1 | DTDIRT | 数据传输方向选择<br>0：从控制器到卡。<br>1：从卡到控制器。 |
| 位 0 | DTEN | 数据传输使能位<br>如果 1 写入到 DTEN 位，则数据传输开始。根据方向位 DTDIR，如果在传输开始时立即将 RW 置 1 开始，则 DPSM 变为 Wait_S 状态、Wait_R 状态或读取等待状态。在数据传输结束后不需要将使能位清零，但必须更新 SDIO_DCTRL 以使能新的数据传输 |

### 23.9.10. SDIO 数据计数寄存器(SDIO_DCOUNT)

偏移地址：0x30

复位值：0x00000000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:25 | 保留 | 必须保持复位值 |
| 位 24:0 | DATACOUNT | 数据计数值<br>读取该位时，将返回要传输的剩余数据字节数量。写入没有任何效果。 |

### 23.9.11. SDIO 状态寄存器(SDIO_STA)

偏移地址：0x34

复位值：0x00000000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | CEATAEND | SDIOIT | RXDAVL | TXDAVL | RXFIFOE | TXFIFOE | RXFIFOF | TXFIFOF |
| | | | | | | | | r | r | r | r | r | r | r | r |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RXFIFOHF | TXFIFOHE | RXACT | TXACT | CMDACT | DBCKEND | STBITERR | DATAEND | CMDSENT | CMDREND | RXOVERR | TXUNDERR | DTIMEOUT | CTIMEOUT | DCRCFAIL | CCRCFAIL |
| r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:24 | 保留 | 必须保持复位值 |
| 位 23 | CEATAEND | 针对 CMD61 收到了 CE-ATA 命令完成信号 |
| 位 22 | SDIOIT | 收到了 SDIO 中断 |
| 位 21 | RXDAVL | 接收 FIFO 中有数据可用 |
| 位 20 | TXDAVL | 传输 FIFO 中有数据可用 |
| 位 19 | RXFIFOE | 接收 FIFO 为空 |
| 位 18 | TXFIFOE | 发送 FIFO 为空<br>如果使能了硬件流控制，则 TXFIFOE 信号在 FIFO 包含 2 个字时激活。 |
| 位 17 | RXFIFOF | 接收 FIFO 已满<br>如果使能了硬件流控制，则 RXFIFOF 信号在 FIFO 差 2 个字便变满之前激活。 |
| 位 16 | TXFIFOF | 传输 FIFO 已满 |
| 位 15 | RXFIFOHF | 接收 FIFO 半满：FIFO 中至少有 8 个字 |
| 位 14 | TXFIFOHE | 传输 FIFO 半空：至少可以写入 8 个字到 FIFO |
| 位 13 | RXACT | 数据接收正在进行中 |
| 位 12 | TXACT | 数据传输正在进行中 |
| 位 11 | CMDACT | 命令传输正在进行中 |
| 位 10 | DBCKEND | 已发送/接收数据块（CRC 校验通过） |
| 位 9 | STBITERR | 在宽总线模式下，并非在所有数据信号上都检测到了起始位 |
| 位 8 | DATAEND | 数据结束（数据计数器 SDIDCOUNT 为零） |
| 位 7 | CMDSENT | 命令已发送（不需要响应） |
| 位 6 | CMDREND | 已接收命令响应（CRC 校验通过） |
| 位 5 | RXOVERR | 收到了 FIFO 上溢错误 |
| 位 4 | TXUNDERR | 传输 FIFO 下溢错误 |
| 位 3 | DTIMEOUT | 数据超时 |
| 位 2 | CTIMEOUT | 命令响应超时<br>命令超时周期为固定值 64 个 SDIO_CK 时钟周期。 |
| 位 1 | DCRCFAIL | 已发送/接收数据块（CRC 校验失败） |
| 位 0 | CCRCFAIL | 已接收命令响应（CRC 校验失败） |

### 23.9.12. SDIO 中断清零寄存器(SDIO_ICR)

偏移地址：0x38

复位值：0x00000000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| CEATAENDC | SDIOITC | Reserved | | | | | |
| rw | rw | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | DBCKENDC | STBITERRC | DATAENDC |
| | | | | | rw | rw | rw |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CMDSENTC | CMDRENDC | RXOVERRC | TXUNDERRC | DTIMEOUTC | CTIMEOUTC | DCRCFAILC | CCRCFAILC |
| rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:24 | 保留 | 必须保持复位值 |
| 位 23 | CEATAENDC | CEATAEND 标志清零位<br>由软件置 1，用于将 CEATAEND 标志清零。<br>0：未将 CEATAEND 清零<br>1：已将 CEATAEND 清零 |
| 位 22 | SDIOITC | SDIOIT 标志清零位<br>由软件置 1，用于将 SDIOIT 标志清零。<br>0：未将 SDIOIT 清零<br>1：已将 SDIOIT 清零 |
| 位 21:11 | 保留 | 必须保持复位值 |
| 位 10 | DBCKENDC | DBCKEND 标志清零位<br>由软件置 1，用于将 DBCKEND 标志清零。<br>0：未将 DBCKEND 清零 1：已将 DBCKEND 清零 |
| 位 9 | STBITERR | STBITERR 标志清零位<br>由软件置 1，用于将 STBITERR 标志清零。<br>0：未将 STBITERR 清零<br>1：已将 STBITERR 清零 |
| 位 8 | DATAEND | DATAEND 标志清零位<br>由软件置 1，用于将 DATAEND 标志清零。<br>0：未将 DATAEND 清零<br>1：已将 DATAEND 清零 |
| 位 7 | CMDSENT | CMDSENT 标志清零位<br>由软件置 1，用于将 CMDSENT 标志清零。<br>0：未将 CMDSENT 清零<br>1：已将 CMDSENT 清零 |
| 位 6 | CMDREND | CMDREND 标志清零位<br>由软件置 1，用于将 CMDREND 标志清零。<br>0：未将 CMDREND 清零 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 1：已将 CMDREND 清零 |
| 位 5 | RXOVERR | RXOVERR 标志清零位<br>由软件置 1，用于将 RXOVERR 标志清零。<br>0：未将 RXOVERR 清零<br>1：已将 RXOVERR 清零 |
| 位 4 | TXUNDERR | TXUNDERR 标志清零位<br>由软件置 1，用于将 TXUNDERR 标志清零。<br>0：未将 TXUNDERR 清零<br>1：已将 TXUNDERR 清零 |
| 位 3 | DTIMEOUT | DTIMEOUT 标志清零位<br>由软件置 1，用于将 DTIMEOUT 标志清零。<br>0：未将 DTIMEOUT 清零<br>1：已将 DTIMEOUT 清零 |
| 位 2 | CTIMEOUT | CTIMEOUT 标志清零位<br>由软件置 1，用于将 CTIMEOUT 标志清零。<br>0：未将 CTIMEOUT 清零<br>1：已将 CTIMEOUT 清零 |
| 位 1 | DCRCFAIL | DCRCFAIL 标志清零位<br>由软件置 1，用于将 DCRCFAIL 标志清零。<br>0：未将 DCRCFAIL 清零<br>1：已将 DCRCFAIL 清零 |
| 位 0 | CCRCFAIL | CCRCFAIL 标志清零位<br>由软件置 1，用于将 CCRCFAIL 标志清零。<br>0：未将 CCRCFAIL 清零<br>1：已将 CCRCFAIL 清零 |

### 23.9.13. SDIO 屏蔽寄存器(SDIO_MASK)

偏移地址：0x38

复位值：0x00000000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | CEATAENDIE | SDIOITIE | RXDAVLIE | TXDAVLIE | RXFIFOEIE | TXFIFOEIE | RXFIFOFIE | TXFIFOFIE |
| | | | | | | | | r | r | r | r | r | r | r | r |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RXFIFOHFIE | TXFIFOHEIE | RXACTIE | TXACTIE | CMDACTIE | DBCKENDIE | STBITERRIE | DATAENDIE | CMDSENTIE | CMDRENDIE | RXOVERRIE | TXUNDERRIE | DTIMEOUTIE | CTIMEOUTIE | DCRCFAILIE | CCRCFAILIE |
| r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:24 | 保留 | 必须保持复位值 |
| 位 23 | CEATAENDIE | 接收到 CE-ATA 命令完成信号时中断使能<br>由软件置 1 和清零，用于使能/禁止收到 CE-ATA 命令完成信号时生成的中断。<br>0：禁止接收到 CE-ATA 命令完成信号时中断<br>1：使能接收到 CE-ATA 命令完成信号时中断 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 22 | SDIOITIE | 接受到 SDIO 模式中断时中断使能<br>由软件置 1 和清零，用于使能/禁止收到 SDIO 模式中断时生成中断。<br>0：禁止接收到 SDIO 模式中断时中断<br>1：使能接收到 SDIO 模式中断时中断 |
| 位 21 | RXDAVLIE | Rx FIFO 中数据可用时中断使能<br>由软件置 1 和清零，用于使能/禁止由于 Rx FIFO 中存在数据而生成的中断。<br>0：禁止 Rx FIFO 中有数据时中断<br>1：使能 Rx FIFO 中有数据时中断 |
| 位 20 | TXDAVLIE | Tx FIFO 中数据可用时中断使能<br>由软件置 1 和清零，用于使能/禁止由于 Tx FIFO 中存在数据而生成的中断。<br>0：禁止 Tx FIFO 中数据可用时中断<br>1：使能 Tx FIFO 中数据可用时中断 |
| 位 19 | RXFIFOEIE | Rx FIFO 为空时中断使能<br>由软件置 1 和清零，用于使能/禁止由 Rx FIFO 为空引起的中断。<br>0：禁止 Rx FIFO 为空时中断<br>1：使能 Rx FIFO 为空时中断 |
| 位 18 | TXFIFOEIE | Tx FIFO 为空时中断使能<br>由软件置 1 和清零，用于使能/禁止由 Tx FIFO 为空引起的中断。<br>0：禁止 Tx FIFO 为空时中断<br>1：使能 Tx FIFO 为空时中断 |
| 位 17 | RXFIFOFIE | Rx FIFO 变满时中断使能<br>由软件置 1 和清零，用于使能/禁止由 Rx FIFO 变满引起的中断。<br>0：禁止 Rx FIFO 变满时中断<br>1：使能 Rx FIFO 变满时中断 |
| 位 16 | TXFIFOFIE | Tx FIFO 变满时中断使能<br>由软件置 1 和清零，用于使能/禁止由 Tx FIFO 变满引起的中断。<br>0：禁止 Tx FIFO 变满时中断<br>1：使能 Tx FIFO 变满时中断 |
| 位 15 | RXFIFOHFIE | Rx FIFO 半满时中断使能<br>由软件置 1 和清零，用于使能/禁止由 Rx FIFO 半满引起的中断。<br>0：禁止 Rx FIFO 半满时中断<br>1：使能 Rx FIFO 半满时中断 |
| 位 14 | TXFIFOHEIE | Tx FIFO 半空时中断使能<br>由软件置 1 和清零，用于使能/禁止由 Tx FIFO 半空引起的中断。<br>0：禁止 Tx FIFO 半空时中断<br>1：使能 Tx FIFO 半空时中断 |
| 位 13 | RXACTIE | 数据接收操作中断使能<br>由软件置 1 和清零，用于使能/禁止由正在接收的数据（数据接收操作）导致的中断。<br>0：禁止数据接收操作中断<br>1：使能数据接收操作中断 |
| 位 12 | TXACTIE | 数据传输操作中断使能<br>由软件置 1 和清零，用于使能/禁止由正在传输的数据（数据传输操作）导致的中断。<br>0：禁止数据传输操作中断<br>1：使能数据传输操作中断 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 11 | CMDACTIE | 命令操作中断使能<br>由软件置 1 和清零，用于使能/禁止由正在传输的命令（命令操作）导致的中断。<br>0：禁止命令操作中断<br>1：使能命令操作中断 |
| 位 10 | DBCKENDIE | 数据块结束中断使能<br>由软件置 1 和清零，用于使能/禁止由数据块结束引起的中断。<br>0：禁止数据块结束中断<br>1：使能数据块结束中断 |
| 位 9 | STBITERRIE | 起始位错误中断使能<br>由软件置 1 和清零，用于使能/禁止由起始位错误引起的中断。<br>0：禁止起始位错误中断<br>1：使能起始位错误中断 |
| 位 8 | DATAENDIE | 数据结束中断使能<br>由软件置 1 和清零，用于使能/禁止由数据结束引起的中断。<br>0：禁止数据结束中断<br>1：使能数据结束中断 |
| 位 7 | CMDSENTIE | 命令发送中断使能<br>由软件置 1 和清零，用于使能/禁止由发送命令引起的中断。<br>0：禁止命令发送中断<br>1：使能命令发送中断 |
| 位 6 | CMDRENDIE | 命令响应接收中断使能<br>由软件置 1 和清零，用于使能/禁止由接收命令响应引起的中断。<br>0：禁止命令响应接收中断<br>1：使能命令响应接收中断 |
| 位 5 | RXOVERRIE | Rx FIFO 上溢错误中断使能<br>由软件置 1 和清零，用于使能/禁止由 Rx FIFO 上溢错误引起的中断。<br>0：禁止 Rx FIFO 上溢错误中断<br>1：使能 Rx FIFO 上溢错误中断 |
| 位 4 | TXUNDERRIE | Tx FIFO 下溢错误中断使能<br>由软件置 1 和清零，用于使能/禁止由 Tx FIFO 下溢错误引起的中断。<br>0：禁止 Tx FIFO 下溢错误中断<br>1：使能 Tx FIFO 下溢错误中断 |
| 位 3 | DTIMEOUTIE | 数据超时中断使能<br>由软件置 1 和清零，用于使能/禁止由数据超时引起的中断。<br>0：禁止数据超时中断<br>1：使能数据超时中断 |
| 位 2 | CTIMEOUTIE | 命令超时中断使能<br>由软件置 1 和清零，用于使能/禁止由命令超时引起的中断。<br>0：禁止命令超时中断<br>1：使能命令超时中断 |
| 位 1 | DCRCFAILIE | 数据 CRC 失败中断使能<br>由软件置 1 和清零，用于使能/禁止由数据 CRC 失败引起的中断。<br>0：禁止数据 CRC 失败中断<br>1：使能数据 CRC 失败中断 |
| 位 0 | CCRCFAILIE | 命令 CRC 失败中断使能<br>由软件置 1 和清零，用于使能/禁止由命令 CRC 失败引起的中断。<br>0：禁止命令 CRC 失败中断 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 1：使能命令 CRC 失败中断 |

### 23.9.14. SDIO FIFO 计数寄存器(SDIO_FIFOCNT)

偏移地址：0x48

复位值：0x00000000

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:24 | 保留 | 必须保持复位值 |
| 位 23:0 | FIFO_COUNT | 要在 FIFO 中写入或读取的剩余字数。 |

### 23.9.15. SDIO 数据 FIFO 寄存器(SDIO_FIFO)

偏移地址：0x48

复位值：0x00000000

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:0 | FIFO_DATA | 接收和传输 FIFO 数据<br>FIFO 数据占用 32 位字的 32 个入口，从地址： SDIO 基址 + 0x080 到 SDIO 基址 + 0xFC。 |

# 24. 控制器区域网络(bxCAN)

## 24.1. bxCAN 简介

基本扩展 CAN 外设又称 bxCAN, 可与 CAN 网络进行交互。该外设支持 2.0A 和 B 版本的 CAN 协议, 旨在以最少的 CPU 负载高效管理大量的传入消息，并可按需要的优先级实现消息发送。

在关系安全性的应用中，CAN 控制器提供所有必要的硬件功能来支持 CAN 时间触发通信方案。

## 24.2. bxCAN 主要特性

- 支持 2.0 A 及 2.0 B Active 版本 CAN 协议;
- 比特率高达 1 Mb/s;
- 支持时间触发通信方案。

**发送**

- 三个发送邮箱;
- 可配置的发送优先级;
- SOF 发送时间戳。

**接收**

- 两个具有三级深度的接收 FIFO;
- 可调整的筛选器组:
  - CAN1 和 CAN2 之间共享 28 个筛选器组;
- 标识符列表功能;
- 可配置的 FIFO 上溢;
- SOF 接收时间戳。

**时间触发通信方案**

- 禁止自动重发送模式;
- 16 位自由运行定时器;
- 在最后两个数据字节发送时间戳。

**管理**

- 可屏蔽中断;
- 在唯一地址空间通过软件实现高效的邮箱映射。

**双 CAN**

- CAN1：主 bxCAN，用于管理 bxCAN 与 512 字节 SRAM 存储器之间的通信。
- CAN2：从 bxCAN，无法直接访问 SRAM 存储器。
- 两个 bxCAN 单元共享 512 字节 SRAM 存储器。

## 24.3. bxCAN 一般说明

在如今的 CAN 应用中，网络节点数量日益增多，经常需要通过网关将数个网络连接在一起。系统中的消

息数量（以及各个节点需要处理的消息）也有了显著增加。除应用程序消息外，还引入了网络管理和诊断消息。

- 各种类型的消息需要一个增强的筛选机制进行处理。

  此外，应用程序任务需要更多的 CPU 时间，因此必须减少因消息接收而对实时处理造成的限制。
- 接收 FIFO 方案使 CPU 能够长时间专门处理应用程序任务，而又不致丢失消息。

  基于标准 CAN 驱动程序的标准 HLP（更高层协议）需要一个高效接口来与 CAN 控制器连接。

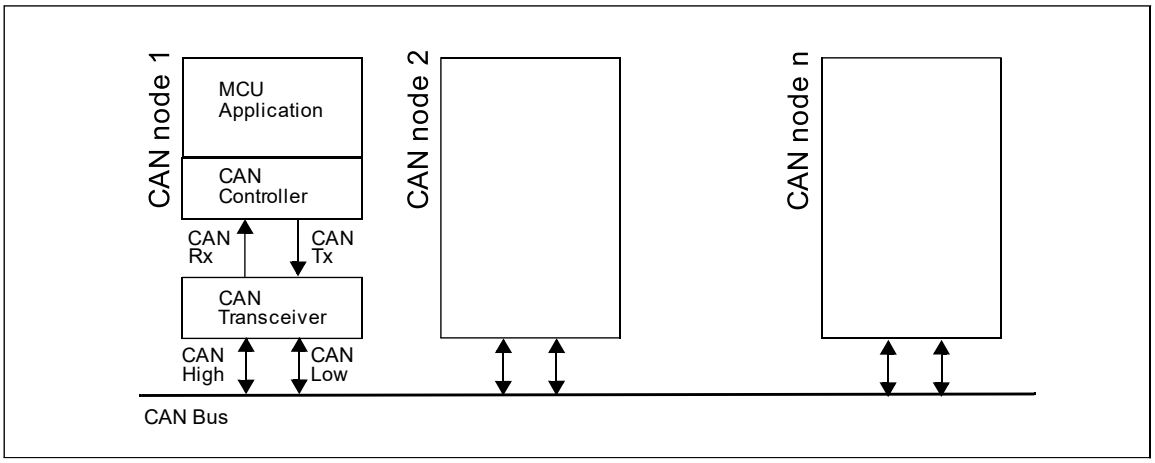



图 24-1 CAN 网络拓扑结构

### 24.3.1. CAN 2.0B 主动内核

bxCAN 模块可完全自主地处理 CAN 消息的发送和接收。标准标识符（11 位）和扩展标识符（29 位）完全由硬件支持。

### 24.3.2. 控制、状态和配置寄存器

应用程序使用这些寄存器进行以下操作：

- 配置 CAN 参数，例如波特率；
- 请求发送；
- 处理接收；
- 管理中断；
- 获取诊断信息。

### 24.3.3. 发送邮箱

软件可通过三个发送邮箱设置消息。发送调度程序负责决定首先发送哪个邮箱的内容。

### 24.3.4. 验收筛选器

bxCAN 提供了 28 个可调整/可配置的标识符筛选器组，用于选择软件所需的传入消息并丢弃其余消息。其他器件中，共有 14 个可调整/可配置的标识符筛选器组。

**接收 FIFO**

硬件使用两个接收 FIFO 来存储传入消息。每个 FIFO 中可以存储三条完整消息。 FIFO 完全由硬件管理。

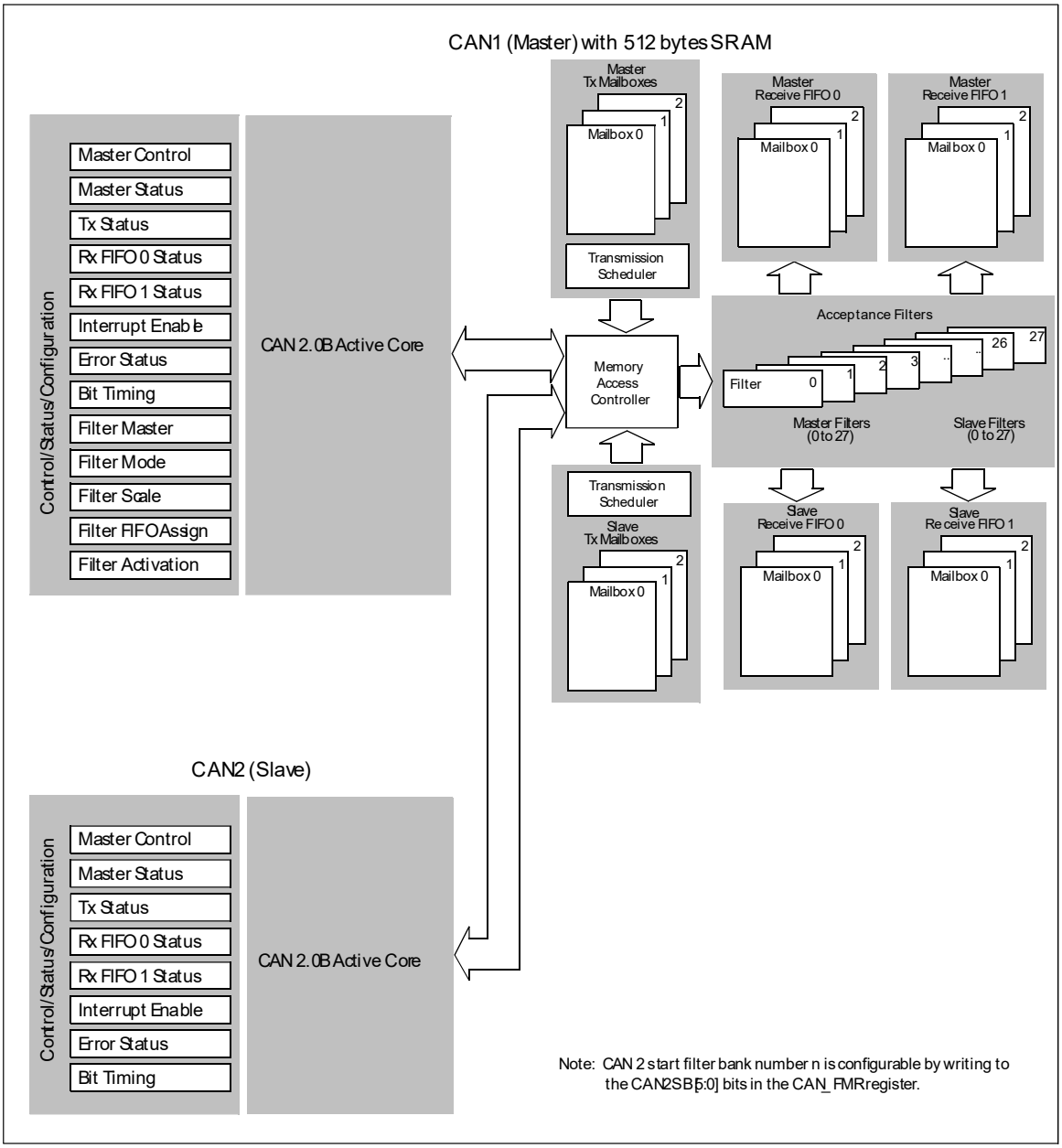



图 24-2 双 CAN 框图

## 24.4. bxCAN 工作模式

bxCAN 有三种主要的工作模式：初始化、正常和睡眠。硬件复位后，bxCAN 进入睡眠模式以降低功耗，同时 CANTX 上的内部上拉电阻激活。软件将 CAN_MCR 寄存器的 INRQ 或 SLEEP 位置 1，以请求 bxCAN 进入初始化或睡眠模式。一旦进入该模式， bxCAN 即将 CAN_MSR 寄存器的 INAK 或 SLAK 位置 1，以确认该模式，同时禁止内部上拉电阻。如果 INAK 和 SLAK 均未置 1，则 bxCAN 将处于正常模式。进入正常模式之前， bxCAN 必须始终在 CAN 总线上实现同步。为了进行同步， bxCAN 将等待 CAN 总线空闲（即，已监测到 CANRX 上的 11 个隐性位）。

### 24.4.1. 初始化模式

当硬件处于初始化模式时，可以进行软件初始化。为进入该模式，软件将 CAN_MCR 寄存器的 INRQ 位

置 1，并等待硬件通过将 CAN_MCR 寄存器的 INAK 位置 1 来确认请求。

为退出初始化模式，软件将 INQR 位清零。一旦硬件将 INAK 位清零，bxCAN 即退出初始化模式。

在初始化模式下，所有从 CAN 总线传入和传出的消息都将停止，并且 CAN 总线输出 CANTX 的状态为隐性（高）。

进入初始化模式不会更改任何配置寄存器。

为初始化 CAN 控制器，软件必须设置位定时 (CAN_BTR) 和 CAN 选项 (CAN_MCR) 寄存器。

为初始化与 CAN 筛选器组相关的寄存器（模式、尺度、 FIFO 分配、激活和筛选器值），软件必须将 FINIT 位 (CAN_FMR) 置 1。筛选器的初始化也可以在初始化模式之外进行。

*注意： FINIT=1 时， CAN 接收停用。*

*筛选器值也可通过停用（CAN_FA1R 寄存器的）相关筛选器激活位来修改。*

*如果某个筛选器组未使用，建议将其保持未激活状态（将相应 FACT 位保持清零）。*

### 24.4.2. 正常模式

一旦初始化完成，软件必须向硬件请求进入正常模式，这样才能在 CAN 总线上进行同步，并开始接收和发送。

进入正常模式的请求可通过将 CAN_MCR 寄存器的 INRQ 位清零来发出。bxCAN 进入正常模式，并与 CAN 总线上的数据传输实现同步后，即可参与总线活动。执行这一步时，需要等待出现一个由 11 个连续隐性位（总线空闲状态）组成的序列。硬件通过将 CAN_MSR 寄存器的 INAK 位清零，来确认切换到正常模式。

筛选器值的初始化与初始化模式无关，但必须要在筛选器处于未激活状态（相应 FACTx 位清零）时进行。筛选器尺度和模式配置必须在进入正常模式之前完成。

### 24.4.3. 睡眠模式（低功耗）

为降低能耗功耗， bxCAN 具有低功耗模式，称为睡眠模式。软件通过将 CAN_MCR 寄存器的 SLEEP 位置 1 而发出请求后，即可进入该模式。该模式下，bxCAN 时钟停止，但软件仍可访问 bxCAN 邮箱。

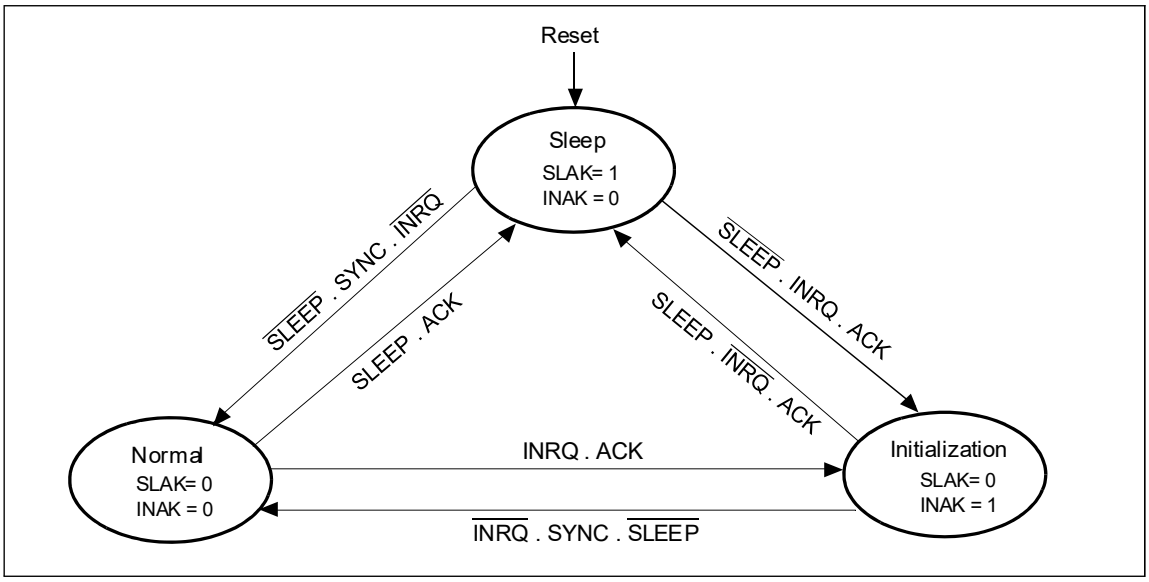
在 bxCAN 处于睡眠模式时，如果软件通过将 INRQ 位置 1 来请求进入初始化模式，则必须同时将 SLEEP 位清零。

软件将 SLEEP 位清零或是检测到 CAN 总线活动时， bxCAN 即被唤醒（退出睡眠模式）。

检测到 CAN 总线活动后，如果 CAN_MCR 寄存器的 AWUM 位置 1，硬件将通过清零 SLEEP 位来自动执行唤醒序列。如果 AWUM 位清零，在发生唤醒中断时，软件必须将 SLEEP 位清零才能退出睡眠模式。

*注意： 如果使能唤醒中断 (CAN_IER 寄存器的 WKUIE 位置 1)，即使 bxCAN 自动执行唤醒序列，一旦检测到 CAN 总线活动，也会发生唤醒中断。*

*SLEEP 位清零后，一旦 bxCAN 与 CAN 总线同步，即会退出睡眠模式，请参见下图 bxCAN 工作模式。一旦硬件将 SLAK 位清零，即会退出睡眠模式。*




*1. ACK = 硬件通过将 CAN_MSR 寄存器的 INAK 或 SLAK 位置 1 来确认请求的等待状态。*

*2. SYNC = bxCAN 等待 CAN 总线变为空闲（即在 CANRX 上监测到连续 11 个隐性位）的状态。*

图 24-3 bxCAN 工作模式

## 24.5. 测试模式

可以通过 CAN_BTR 寄存器中的 SILM 和 LBKM 位来选择测试模式。这些位必须在 bxCAN 处于初始化模式时进行配置。选择测试模式后，必须复位 CAN_MCR 寄存器中的 INRQ 位才能进入正常模式。

### 24.5.1. 静默模式

可以通过将 CAN_BTR 寄存器的 SILM 位置 1，将 bxCAN 置于静默模式。

在静默模式下，bxCAN 可以接收有效数据帧和有效遥控帧，但仅在 CAN 总线上发送隐性位，并且无法启动发送。如果 bxCAN 必须发送一个显性位（ ACK 位、溢出标志、活动错误标志），该位将在内部被改道发送，以便 CAN 内核可以监视该显性位，但 CAN 总线可以保持隐性状态。静默模式可用于分析 CAN 总线上的流量，同时又不会因发送显性位（确认位、错误帧）对其造成影响。

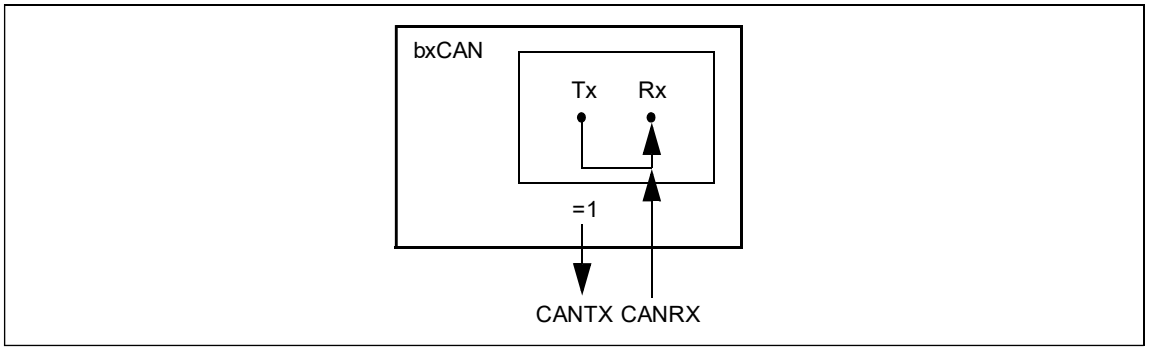



图 24-4 静默模式下的 bxCAN

### 24.5.2. 环回模式

可以通过将 CAN_BTR 寄存器的 LBKM 位置 1，将 bxCAN 置于环回模式。在环回模式下，bxCAN 将其自身发送的消息作为接收的消息来处理并存储（如果这些消息通过了验收筛选）在接收邮箱中。

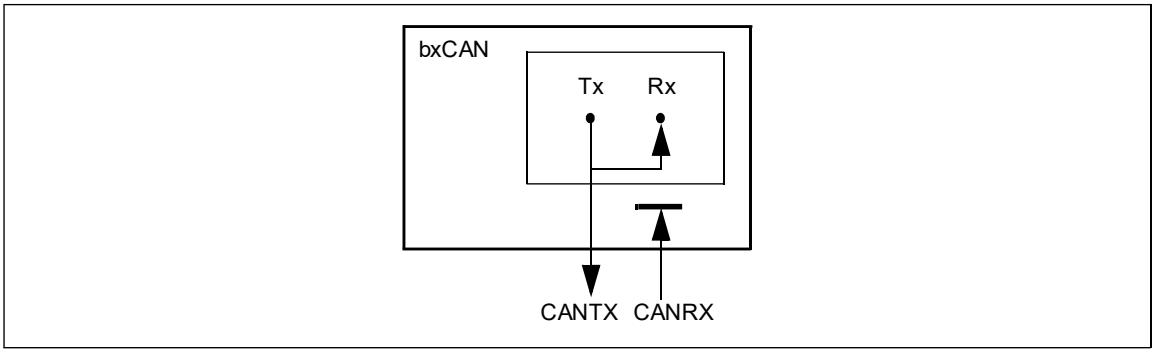



图 24-5 环回模式下的 bxCAN

该模式为自检功能提供。为了不受外部事件的影响， CAN 内核在环回模式下将忽略确认错误（在数据/远程帧的确认时隙不对显性位采样）。在此模式下， bxCAN 将执行从发送输出到接收输入的内部反馈。bxCAN 将忽略 CANRX 输入引脚的实际值。从 CANTX 引脚可以监视发送的消息。

### 24.5.3. 环回与静默模式组合

可以通过将 CAN_BTR 寄存器的 LBKM 和 SILM 位置 1，将环回模式和静默模式组合起来。该模式可用于“热自检”，也就是说， bxCAN 可以像在环回模式下一样进行检测，同时又不会影响与 CANTX 和 CANRX 引脚相连接的运行中的 CAN 系统。在此模式下， CANRX 引脚与 bxCAN 断开连接， CANTX 引脚则保持隐性。

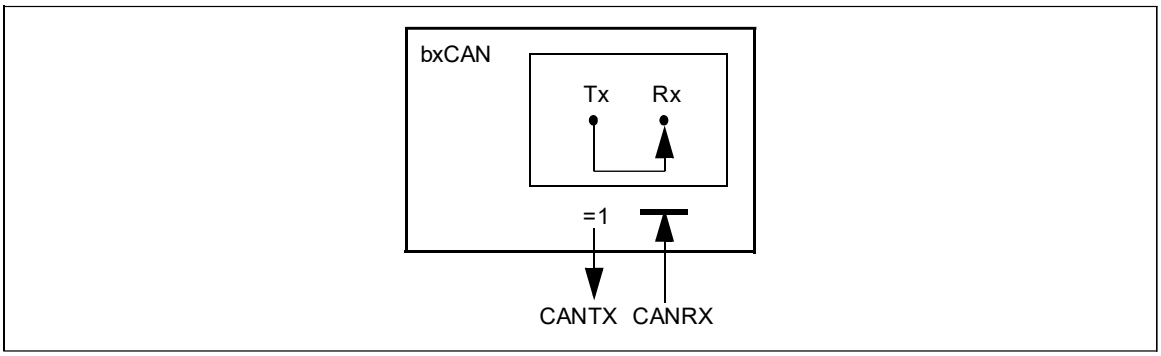



图 24-6 组合模式下的 bxCAN

## 24.6. 调试模式

当微控制器进入调试模式(内核停止)时， bxCAN 可以继续正常工作，也可以停止工作，具体取决于如下条件：

- DBG 模块中用于 CAN1 的 DBG_CAN1_STOP 位或者用于 CAN2 的 DBG_CAN2_STOP 位。
- CAN_MCR 中的 DBF 位。

## 24.7. bxCAN 功能说明

### 24.7.1. 发送处理

为了发送消息，应用程序必须在请求发送前，通过将 CAN_TIxR 寄存器的相应 TXRQ 位置 1，选择一个空发送邮箱，并设置标识符、数据长度代码 (DLC) 和数据。一旦邮箱退出空状态，软件即不再具有对邮箱寄存器的写访问权限。TXRQ 位置 1 后，邮箱立即进入挂起状态，等待成为优先级最高的邮箱，请参见发送优先级。一旦邮箱拥有最高优先级，即被安排发送。CAN 总线变为空闲后，被安排好的邮箱中的消息即开始发送（进入发送状态）。邮箱一旦发送成功，即恢复空状态。硬件通过将 CAN_TSR 寄存器的 RQCP 和 TXOK 位置 1，来表示发送成功。

如果发送失败，失败原因将由 CAN_TSR 寄存器的 ALST 位（仲裁丢失）和/或 TERR 位（检测到发送错误）指示。

**发送优先级**

**按标识符**

当多个发送邮箱挂起时，发送顺序由邮箱中所存储消息的标识符来确定。根据 CAN 协议的仲裁，标识符值最低的消息具有最高的优先级。如果标识符值相等，则首先安排发送编号较小的邮箱。

**按发送请求顺序**

可以通过设置 CAN_MCR 寄存器中的 TXFP 位，将发送邮箱配置为发送 FIFO。在此模式下，优先级顺序按照发送请求顺序来确定。

该模式对分段发送非常有用。

**中止**

可以通过将 CAN_TSR 寄存器的 ABRQ 位置 1，来中止发送请求。在挂起或已安排状态下，邮箱立即中止。如果在邮箱处于发送状态时请求中止，则会出现两种结果。如果邮箱发送成功，将变为空状态，同时 CAN_TSR 寄存器的 TXOK 位置 1。如果发送失败，邮箱变为已安排状态，发送中止并变为空状态，同时 TXOK 位清零。在所有情况下，邮箱至少在当前发送结束时都会恢复空状态。

**禁止自动重发送模式**

该模式旨在满足 CAN 标准的时间触发通信方案的要求。要将硬件配置为此模式，必须将 CAN_MCR 寄存器的 NART 位置 1。

在此模式下，每个发送仅启动一次。如果第一次尝试失败，由于仲裁丢失或错误，硬件将不会自动重新启动消息发送。

第一次发送尝试结束时，硬件将认为请求已完成，并将 CAN_TSR 寄存器的 RQCP 位置 1。发送结果由 CAN_TSR 寄存器的 TXOK、 ALST 和 TERR 位来指示。

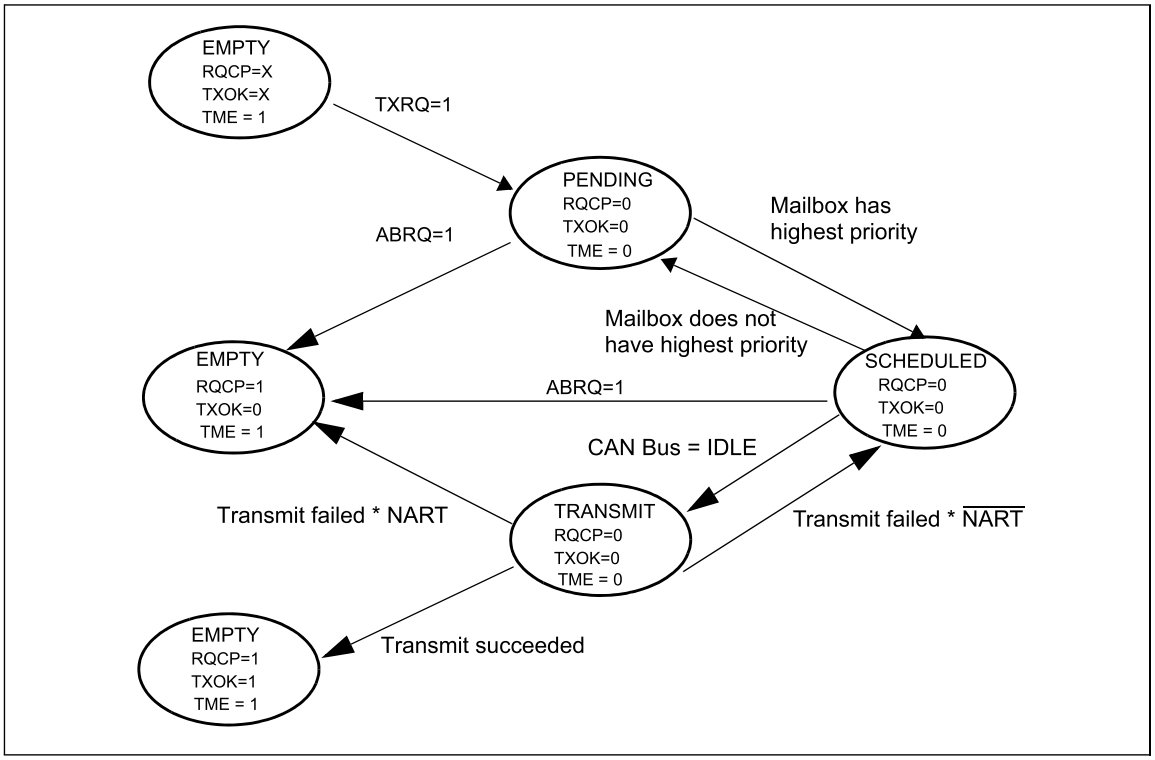



图 24-7 发送邮箱状态

### 24.7.2. 时间触发通信模式

在此模式下，CAN 硬件的内部计数器激活，用于为接收和发送邮箱生成时间戳值，这些值分别存储在 CAN_RDTxR/CAN_TDTxR 寄存器中。内部计数器在每个 CAN 位时间递增。在接收和发送时，都会在帧起始位的采样点捕获内部计数器。

### 24.7.3. 接收处理

为了接收 CAN 消息，提供了构成 FIFO 的三个邮箱。为了节约 CPU 负载，简化软件并保证数据一致性，FIFO 完全由硬件进行管理。应用程序通过 FIFO 输出邮箱访问 FIFO 中所存储的消息。

**有效消息**

当消息依据 CAN 协议正确接收（直到 EOF 字段的倒数第二位都没有发送错误） 并且成功通过标识符筛选后，该消息将视为有效，请参见：标识符筛选。

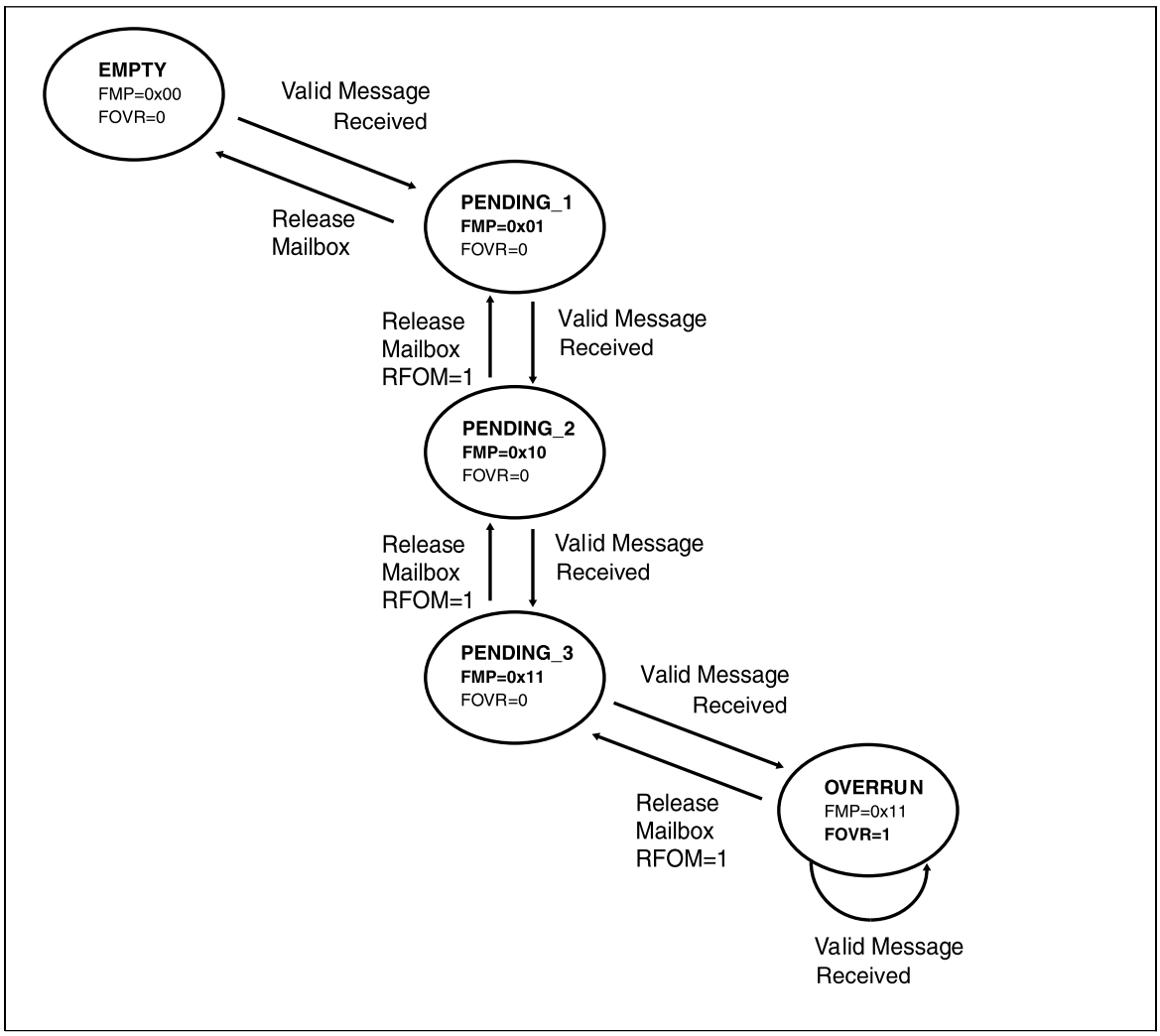



图 24-8 接收 FIFO 状态

### FIFO 管理

FIFO 开始时处于空状态，在接收的第一条有效消息存储在其中后，变为 Pending_1 状态。硬件通过将 CAN_RFR 寄存器的 FMP[1:0] 位置为 01b 来指示该事件。消息将在 FIFO 输出邮箱中供读取。软件将读取邮箱内容，并通过将 CAN_RFR 寄存器的 RFOM 位置 1，来将邮箱释放。 FIFO 随即恢复空状态。如果同时接收到新的有效消息，FIFO 将保持 Pending_1 状态，新消息将在输出邮箱中供读取。

如果应用程序未释放邮箱，下一条有效消息将存储在 FIFO 中，使其进入 Pending_2 状态(FMP[1:0] = 10b)。下一条有效消息会重复该存储过程，同时将 FIFO 变为 Pending_3 状态(FMP[1:0] = 11b)。此时，软件必须通过将 RFOM 位置 1 来释放输出邮箱，从而留出一个空邮箱来存储下一条有效消息。否则，下一次接收到有效消息时，将导致消息丢失。

### 上溢

一旦 FIFO 处于 Pending_3 状态（即三个邮箱均已满），则下一次接收到有效消息时，将导致上溢并丢失一条消息。硬件通过将 CAN_RFR 寄存器的 FOVR 位置 1 来指示上溢状况。丢失的消息取决于 FIFO 的配置：

- 如果禁止 FIFO 锁定功能（CAN_MCR 寄存器的 RFLM 位清零），则新传入的消息将覆盖 FIFO 中存储的最后一条消息。在这种情况下，应用程序将始终能访问到最新的消息
- 如果使能 FIFO 锁定功能（CAN_MCR 寄存器的 RFLM 位置 1），则将丢弃最新的消息，软件将提供 FIFO 中最早的三条消息。

与接收相关的中断消息存储到 FIFO 中后，FMP[1:0] 位即会更新，如果 CAN_IER 寄存器的 FMPIE 位置 1，将产生中断请求。

FIFO 存满消息（即存储了第三条消息）后，CAN_RFR 寄存器的 FULL 位置 1，如果 CAN_IER 寄存器的 FFIE 位置 1，将产生中断。

出现上溢时，FOVR 位将置 1，如果 CAN_IER 寄存器的 FOVIE 位置 1，将产生中断。

### 24.7.4. 标识符筛选

在 CAN 协议中，消息的标识符与节点地址无关，但与消息内容有关。因此，发送器将消息广播给所有接收器。在接收到消息时，接收器节点会根据标识符的值来确定软件是否需要该消息。如果需要，该消息将复制到 SRAM 中。如果不需要，则必须在无软件干预的情况下丢弃该消息。

为了满足这一要求， bxCAN 控制器为应用程序提供了 28 个可配置且可调整的筛选器组 (27-0)。在其他器件中， bxCAN 控制器为应用程序提供了 14 个可配置且可调整的筛选器组 (13-0)，以便仅接收软件需要的消息。此硬件筛选功能可以节省软件筛选所需的 CPU 资源。每个筛选器组 x 均包含两个 32 位寄存器，分别是 CAN_FxR0 和 CAN_FxR1。

**可调整的宽度**

为了根据应用程序的需求来优化和调整筛选器，每个筛选器组可分别进行伸缩调整。根据筛选器尺度不同，一个筛选器组可以：

- 为 STDID[10:0]、EXTID[17:0]、IDE 和 RTR 位提供一个 32 位筛选器。
- 为 STDID[10:0]、RTR、IDE 和 EXTID[17:15] 位提供两个 16 位筛选器。

此外，筛选器还可配置为掩码模式或标识符列表模式。

**掩码模式**

在掩码模式下，标识符寄存器与掩码寄存器关联，用以指示标识符的哪些位“必须匹配”，哪些位“无关”。

**标识符列表模式**

在标识符列表模式下，掩码寄存器用作标识符寄存器。这时，不会定义一个标识符和一个掩码，而是指定两个标识符，从而使单个标识符的数量加倍。传入标识符的所有位都必须与筛选器寄存器中指定的位匹配。

**筛选器组尺度和模式配置**

筛选器组通过相应的 CAN_FMR 寄存器进行配置。为了配置筛选器组，必须通过将 CAN_FAR 寄存器的 FACT 位清零而将其停用。筛选器尺度通过 CAN_FS1R 寄存器的相应 FSCx 位进行配置。相应掩码/标识符寄存器的标识符列表或标识符掩码模式通过 CAN_FMR 寄存器的 FBMx 位进行配置。

要筛选一组标识符，应将掩码/标识符寄存器配置为掩码模式。

要选择单个标识符，应将掩码/标识符寄存器配置为标识符列表模式。

未由应用程序使用的筛选器应保持停用。

筛选器组中的每个筛选器将按从 0 到最大值的顺序进行编号（称为筛选器编号），具体取决于每个筛选器组的模式和尺度。

有关筛选器配置，请参见下图。

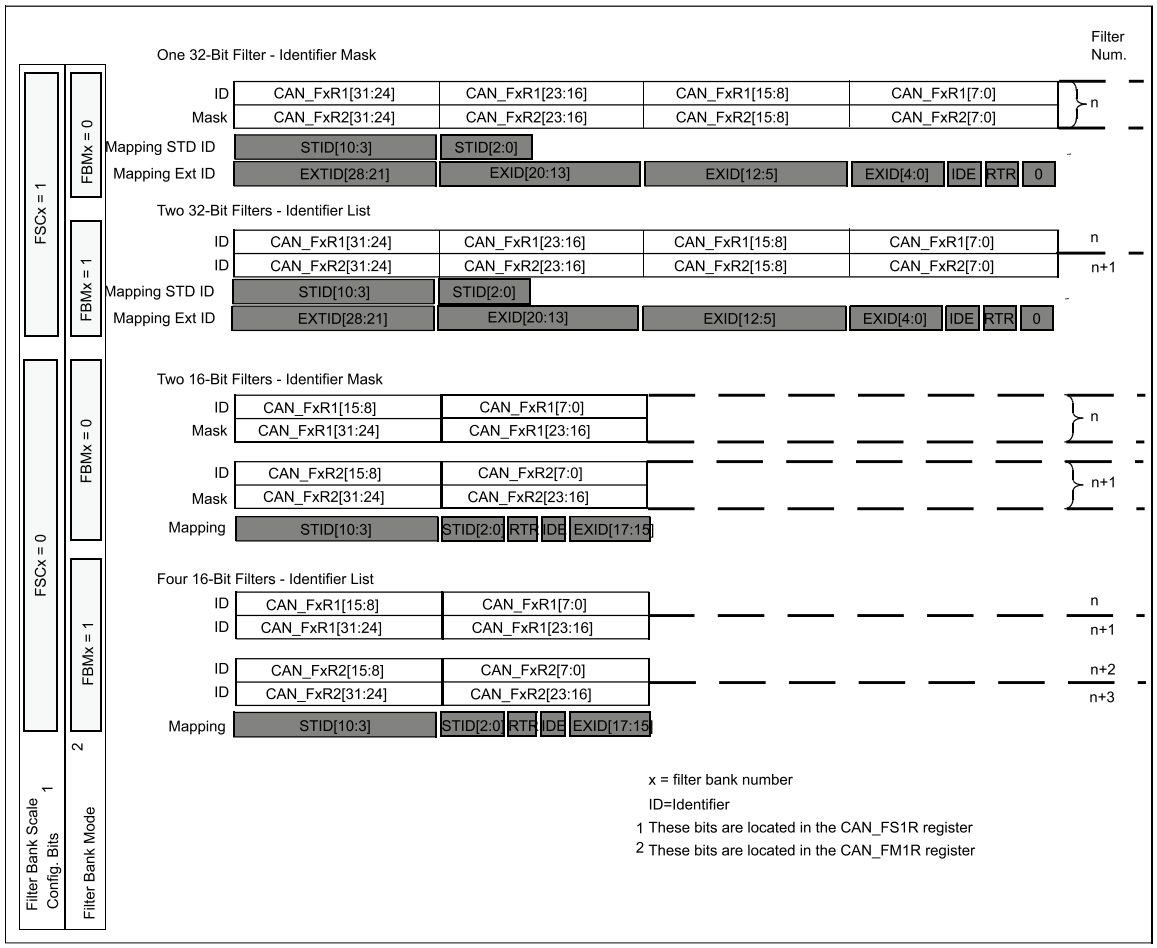



图 24-9 筛选器组尺度配置 - 寄存器构成

### 筛选器匹配索引

消息接收到 FIFO 中后，即可供应用程序使用。应用程序数据通常会复制到 SRAM 中的位置。为了将数据复制到正确的位置，应用程序必须通过标识符来识别数据。为了避免这种情况，方便访问 SRAM 位置，CAN 控制器提供了一个筛选器匹配索引。

该索引根据筛选器优先级规则与消息一同存储在邮箱中。因此，每条收到的消息都有相关联的筛选器匹配索引。

筛选器匹配索引的使用方法有两种：

- 将筛选器匹配索引与预期值列表进行比较。
- 将筛选器匹配索引用作阵列索引，以访问数据目标位置。

对于非屏蔽筛选器，软件不再需要比较标识符。

如果筛选器有屏蔽，软件则只需比较屏蔽位。

筛选器编号的索引值与筛选器组的激活状态无关。此外，还将使用两个独立的编号方案，每个 FIFO 各一个。有关示例，请参见下图。

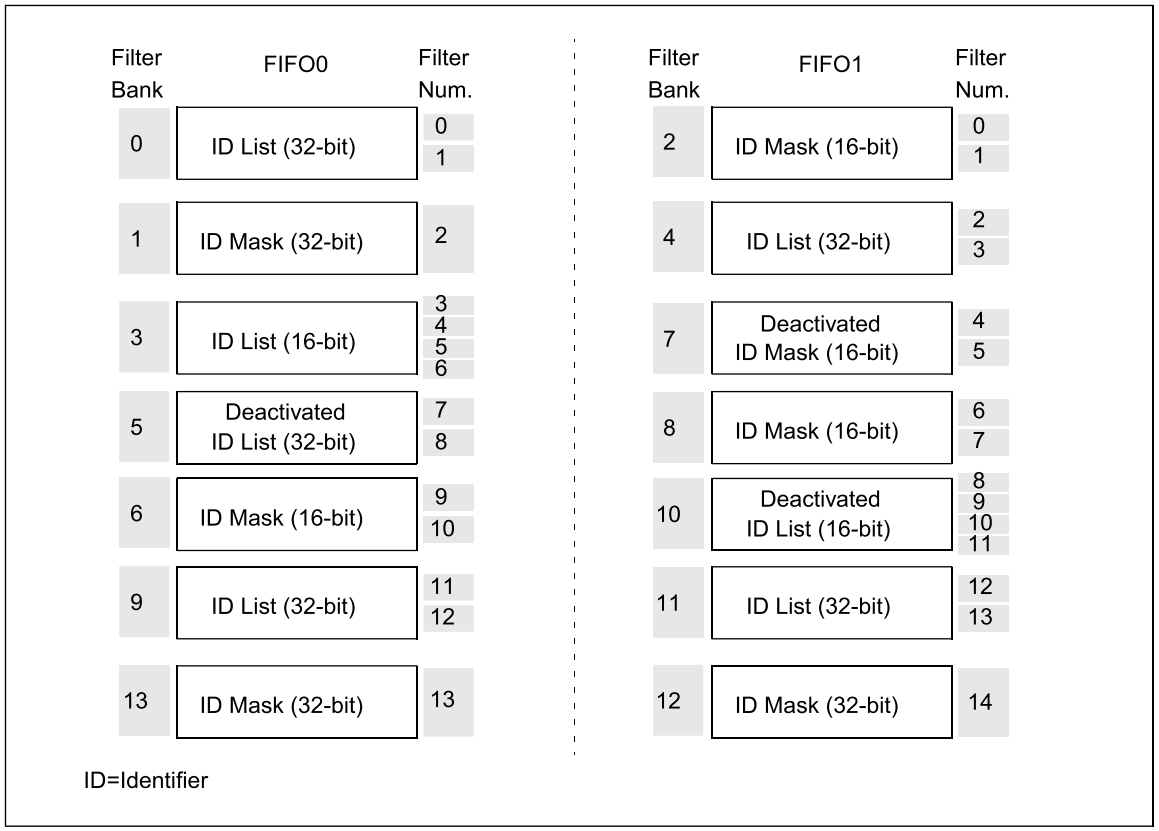



图 24-10 筛选器编号示例

**筛选器优先级规则**

根据筛选器组合，可能会出现一个标识符成功通过数个筛选器的情况。这种情况下，将根据以下优先级规则选择接收邮箱中存储的筛选器匹配值：

- 32 位筛选器优先于 16 位筛选器。
- 对于尺度相等的筛选器，标识符列表模式优先于标识符掩码模式。
- 对于尺度和模式均相等的筛选器，则按筛选器编号确定优先级（编号越低，优先级越高）。

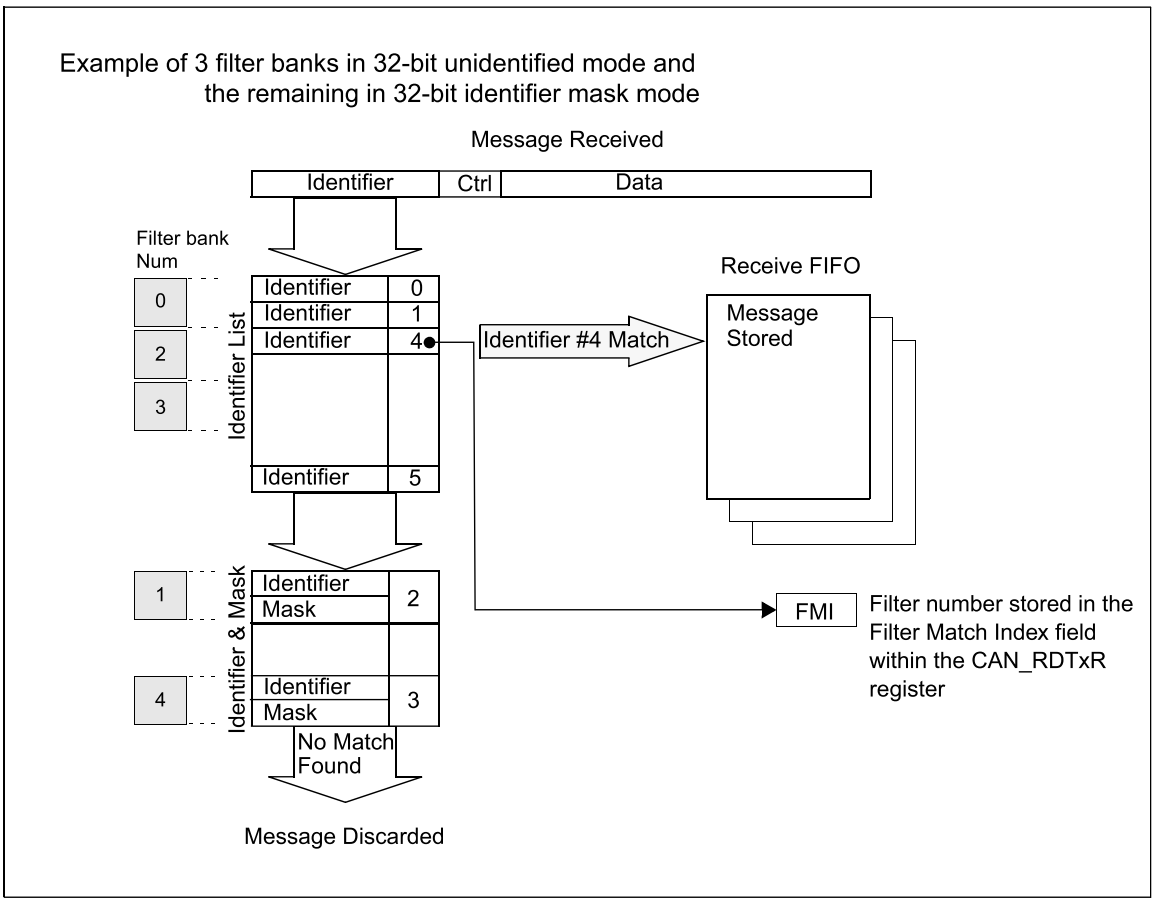



图 24-11 筛选机制-示例

以上示例说明了 bxCAN 的筛选原则。接收到消息后，首先将标识符与标识符列表模式中配置的筛选器进行比较。如果匹配，消息将存储在相应 FIFO 中，匹配筛选器的索引则存储在筛选器匹配索引中。如本例所示，标识符与标识符 #2 匹配，因此消息内容和 FMI 2 存储到该 FIFO 中。

如果不匹配，则将传入标识符与掩码模式中配置的筛选器进行比较。

如果标识符与筛选器中配置的任何标识符均不匹配，硬件会在不干扰软件的情况下丢弃该消息。

### 24.7.5. 消息存储

CAN 消息软件与硬件之间的接口通过邮箱实现。邮箱中包含所有与消息相关的信息：标识符、数据、控制、状态和时间戳信息。

**发送邮箱**

软件在空发送邮箱中设置将要发送的消息。发送状态由硬件在 CAN_TSR 寄存器中进行指示。

表 24-1 发送邮箱映射

| 与发送邮箱基址之间的偏移 | 寄存器名称 |
|---|---|
| 0 | CAN_TIxR |
| 4 | CAN_TDTxR |
| 8 | CAN_TDLxR |
| 12 | CAN_TDHxR |

**接收邮箱**

消息在接收到后，将在 FIFO 输出邮箱中供软件使用。一旦软件对消息进行了处理（例如读取），则必须通过 CAN_RFR 寄存器的 RFOM 位释放 FIFO 输出邮箱，以接收下一条传入消息。筛选器匹配索引存储在 CAN_RDTxR 寄存器的 MFMI 字段中。16 位时间戳值则存储在 CAN_RDTxR 的 TIME[15:0] 字段中。

表 24-2 接收邮箱映射

| 与接收邮箱基址之间的偏移（字节） | 寄存器名称 |
|---|---|
| 0 | CAN_RIxR |
| 4 | CAN_RDTxR |
| 8 | CAN_RDLxR |
| 12 | CAN_RDHxR |

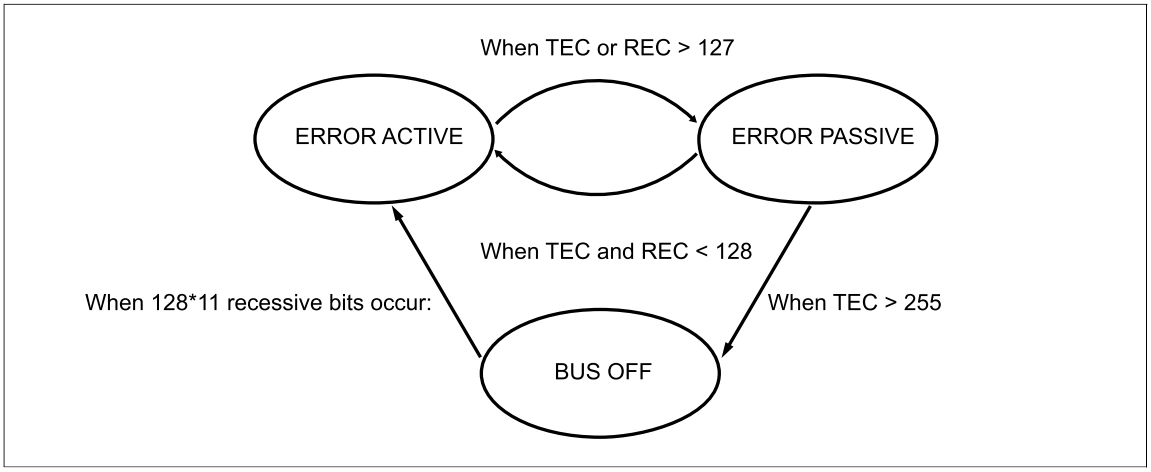



图 24-12 CAN 错误状态图

### 24.7.6. 错误管理

如 CAN 协议所述，错误管理完全由硬件通过发送错误计数器（CAN_ESR 寄存器中的 TEC 值）和接收错误计数器（CAN_ESR 寄存器中的 REC 值）来处理，这两个计数器根据错误状况进行递增或递减。有关 TEC 和 REC 管理的详细信息，请参见 CAN 标准。

两者均可由软件读取，用以确定网络的稳定性。此外，CAN 硬件还将在 CAN_ESR 寄存器中提供当前错误状态的详细信息。通过 CAN_IER 寄存器（ERRIE 位等），软件可以非常灵活地配置在检测到错误时生成的中断。

**总线关闭恢复**

当 TEC 大于 255 时，达到总线关闭状态，该状态由 CAN_ESR 寄存器的 BOFF 位指示。在总线关闭状态下，bxCAN 不能再发送和接收消息。

bxCAN 可以自动或者应软件请求而从总线关闭状态中恢复（恢复错误主动状态），具体取决于 CAN_MCR 寄存器的 ABOM 位。但在两种情况下，bxCAN 都必须至少等待 CAN 标准中指定的恢复序列完成（在 CANRX 上监测到 128 次 11 个连续隐性位）。

如果 ABOM 位置 1，bxCAN 将在进入总线关闭状态后自动启动恢复序列。

如果 ABOM 位清零，则软件必须请求 bxCAN 先进入再退出初始化模式，从而启动恢复序列。

*注意： 在初始化模式下， bxCAN 不会监视 CANRX 信号，因此无法完成恢复序列。 要进行恢复，bxCAN 必须处于正常模式。*

### 24.7.7. 位时序

位时序逻辑将监视串行总线，执行采样并调整采样点，在调整采样点时，需要在起始位边沿进行同步并后续的边沿进行再同步。

通过将标称位时间划分为以下三段，即可解释其工作过程：

**同步段 (SYNC_SEG)**：位变化应该在此时间段内发生。它只有一个时间片的固定长度(1 x tCAN)。

**位段 1 (BS1)**：定义采样点的位置。它包括 CAN 标准的 PROP_SEG 和 PHASE_SEG1。其持续长度可以在 1 到 16 个时间片之间调整，但也可以自动加长，以补偿不同网络节点的频率差异所导致的正相位漂移。

**位段 2 (BS2)**：定义发送点的位置。它代表 CAN 标准的 PHASE_SEG2。其持续长度可以在 1 到 8 个时间片之间调整，但也可以自动缩短，以补偿负相位漂移。

再同步跳转宽度 (SJW) 定义位段加长或缩短的上限。它可以在 1 到 4 个时间片之间调整。

有效边沿是指一个位时间内总线电平从显性到隐性的第一次转换（前提是控制器本身不发送隐性位）。

如果在 BS1 而不是 SYNC_SEG 中检测到有效边沿，则 BS1 会延长最多 SJW，以便延迟采样点。

相反地，如果在 BS2 而不是 SYNC_SEG 中检测到有效边沿，则 BS2 会缩短最多 SJW，以便提前发送点。

为了避免编程错误，位时序寄存器 (CAN_BTR) 只能在器件处于待机模式时进行配置。

*注意：有关 CAN 位时序和再同步机制的详细说明，请参见 ISO 11898 标准。*

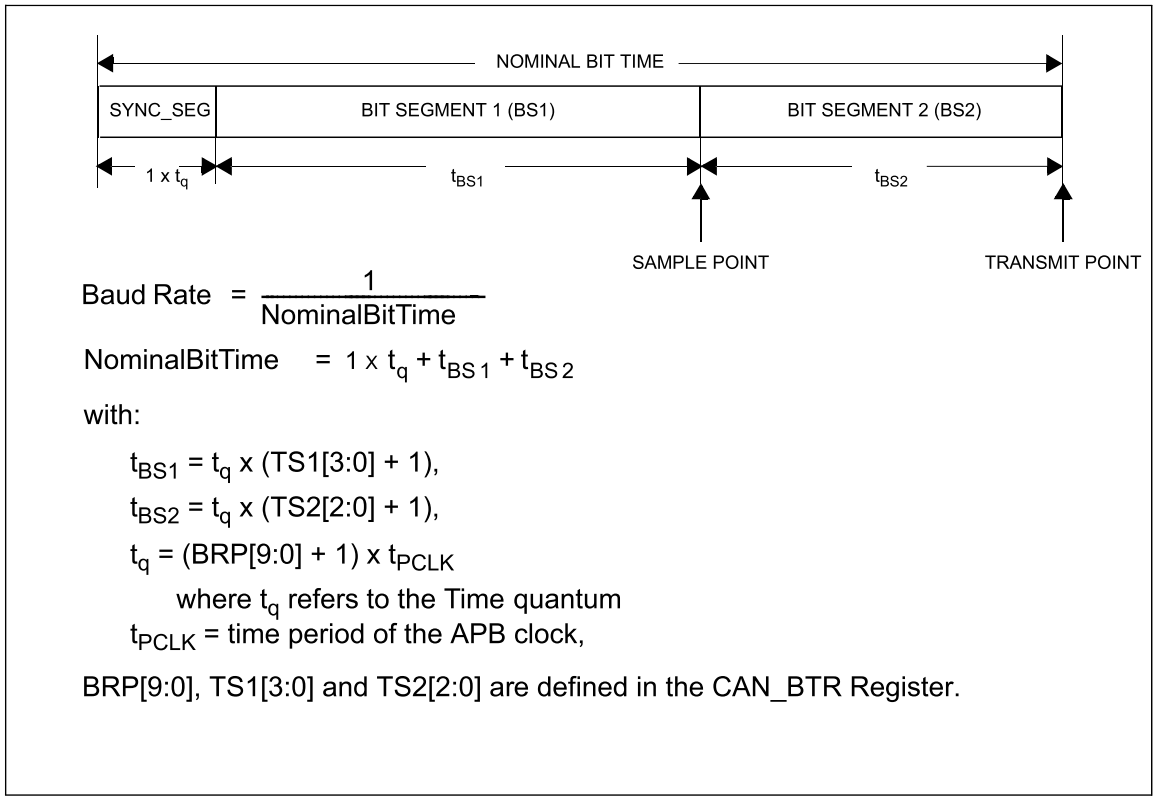



图 24-13 位时序

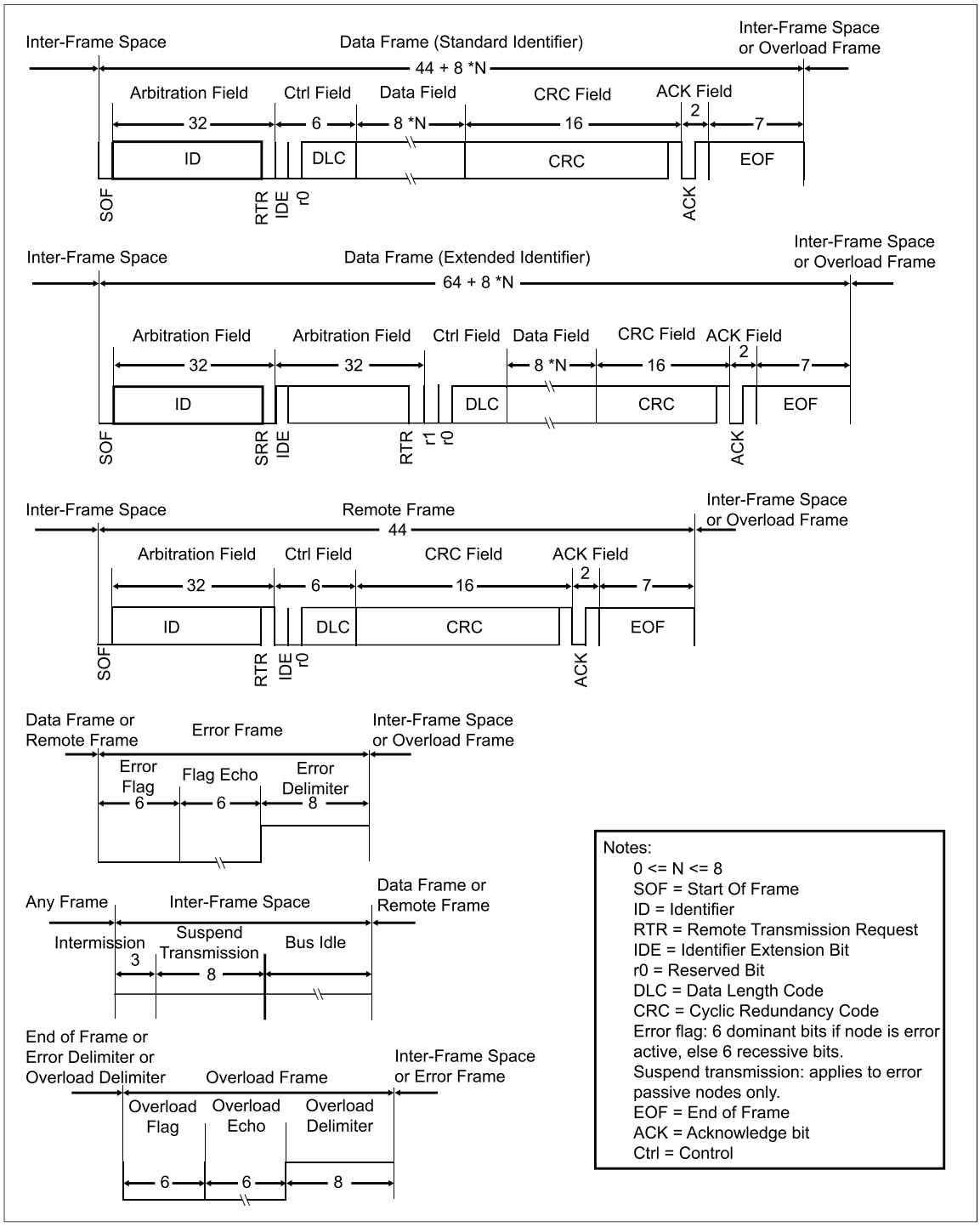



图 24-14 CAN 帧

## 24.8. bxCAN 中断

bxCAN 共有四个专用的中断向量。每个中断源均可通过 CAN 中断使能寄存器 (CAN_IER)来单独地使能或禁止。

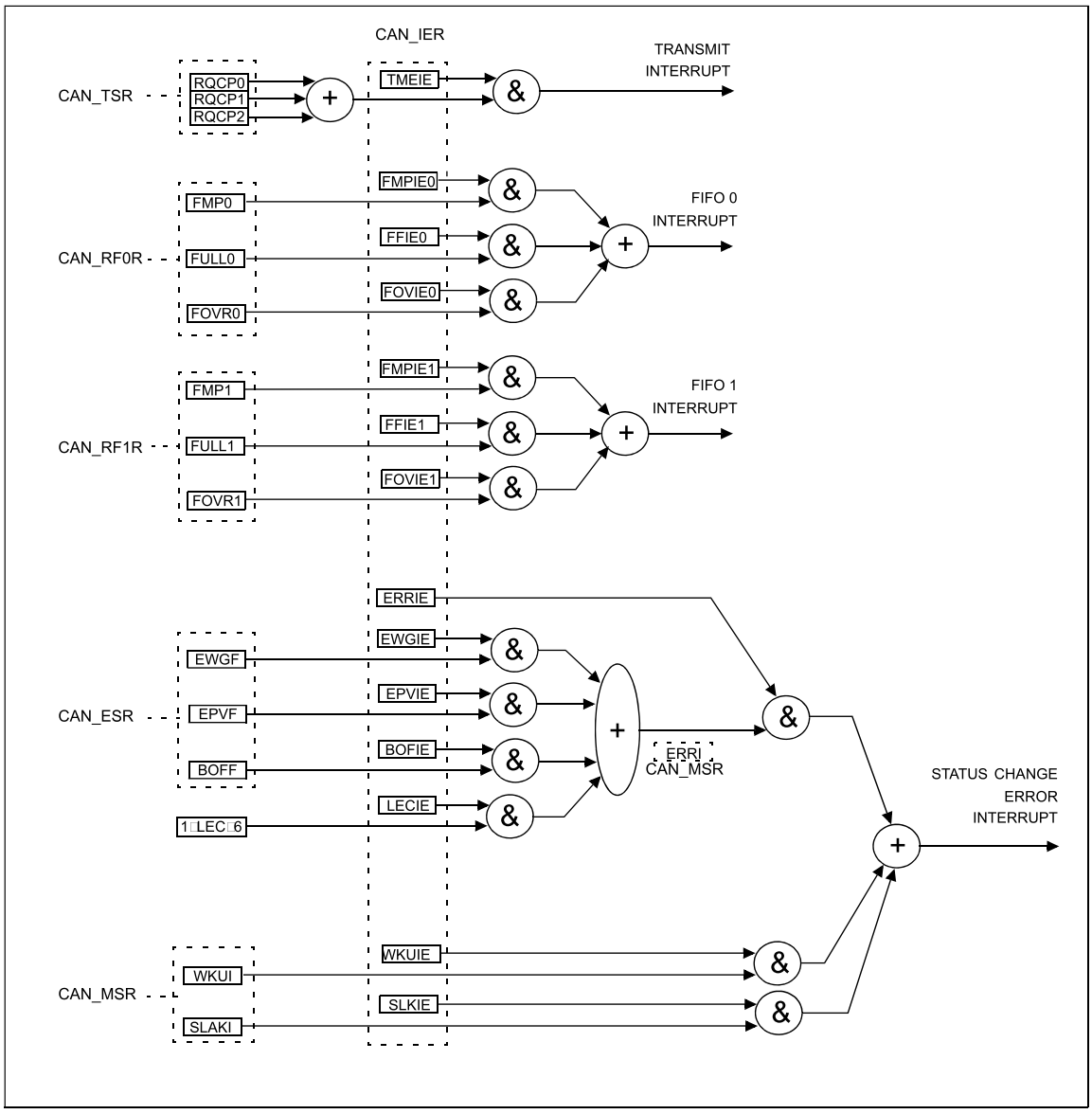



图 24-15 事件标志与中断产生

- 发送中断可由以下事件产生：
  - 发送邮箱 0 变为空，CAN_TSR 寄存器的 RQCP0 位置 1。
  - 发送邮箱 1 变为空，CAN_TSR 寄存器的 RQCP1 位置 1。
  - 发送邮箱 2 变为空，CAN_TSR 寄存器的 RQCP2 位置 1。
- FIFO 0 中断可由以下事件产生：
  - 接收到新消息，CAN_RF0R 寄存器的 FMP0 位不是“00”。
  - FIFO0 满，CAN_RF0R 寄存器的 FULL0 位置 1。
  - FIFO0 上溢，CAN_RF0R 寄存器的 FOVR0 位置 1。
- FIFO 1 中断可由以下事件产生：
  - 接收到新消息，CAN_RF1R 寄存器的 FMP1 位不是“00”。
  - FIFO1 满，CAN_RF1R 寄存器的 FULL1 位置 1。
  - FIFO1 上溢，CAN_RF1R 寄存器的 FOVR1 位置 1。

- 错误和状态改变中断可由以下事件产生：
  - 错误状况，有关错误状况的更多详细信息，请参见 CAN 错误状态寄存器(CAN_ESR)。
  - 唤醒状况， CAN Rx 信号上监测到 SOF。
  - 进入睡眠模式。

## 24.9. CAN 寄存器

外设寄存器必须按字（ 32 位）进行访问。

### 24.9.1. 寄存器访问保护

错误访问某些配置寄存器可能会导致硬件暂时性地干扰整个 CAN 网络。因此，只有在 CAN 硬件处于初始化模式时，才可以通过软件修改 CAN_BTR 寄存器。

尽管传输错误的数据不会引发 CAN 网络级别的故障，但可能会对应用程序造成严重干扰。发送邮箱只能在处于空状态时通过软件进行修改。

可以通过停用相应筛选器组或将 FINIT 位置 1 来修改筛选器值。此外，只有在 CAN_FMR 寄存器的筛选器初始化模式位置 1 (FINIT=1) 时，才能对 CAN_FMxR、 CAN_FSxR 和 CAN_FFAR 寄存器中的筛选器配置（大小、模式和 FIFO 分配）进行修改。

### 24.9.2. CAN 控制和状态寄存器

有关寄存器说明中使用的缩写，请参见第 1.1 节：寄存器相关缩写词列表。

#### CAN 主控制寄存器(CAN_MCR)

偏移地址： 0x00

复位值：0x0001 0002

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESET | Reserved | | | | | | | TTCM | ABOM | AWUM | NART | RFLM | TXFP | SLEEP | INRQ |
| rs | | | | | | | | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:17 | 保留 | 必须保持复位值 |
| 位 16 | DBF | 调试冻结<br>0：调试期间 CAN 处于工作状态。<br>1：调试期间 CAN 处于接收/发送冻结状态。接收 FIFO 仍可正常访问/控制。 |
| 位 15 | RESET | bxCAN 软件主复位<br>0：正常工作。<br>1：强制 bxCAN 进行主复位 -> 复位后激活睡眠模式(FMP 位和 CAN_MCR 寄存器初始化为复位值)。此位自动复位为 0。 |
| 位 14:8 | 保留 | 必须保持复位值 |
| 位 7 | TTCM | 时间触发通信模式 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 0：禁止时间触发通信模式。<br>1：使能时间触发通信模式。 |
| 位 6 | ABOM | 自动的总线关闭管理<br>此位控制 CAN 硬件在退出总线关闭状态时的行为。<br>0：在软件发出请求后，一旦监测到 128 次连续 11 个隐性位，并且软件将 CAN_MCR 寄存器的 INRQ 位先置 1 再清零，即退出总线关闭状态。<br>1：一旦监测到 128 次连续 11 个隐性位，即通过硬件自动退出总线关闭状态。 |
| 位 5 | AWUM | 自动唤醒模式<br>此位控制 CAN 硬件在睡眠模式下接收到消息时的行为。<br>0：在软件通过将 CAN_MCR 寄存器的 SLEEP 位清零发出请求后，退出睡眠模式。<br>1：一旦监测到 CAN 消息，即通过硬件自动退出睡眠模式。<br>CAN_MCR 寄存器的 SLEEP 位和 CAN_MCR 寄存器的 SLAK 位由硬件清零。 |
| 位 4 | NART | 禁止自动重发送<br>0： CAN 硬件将自动重发送消息，直到根据 CAN 标准消息发送成功。<br>1：无论发送结果如何（成功、错误或仲裁丢失），消息均只发送一次。 |
| 位 3 | RFLM | 接收 FIFO 锁定模式<br>0：接收 FIFO 上溢后不锁定。接收 FIFO 装满后，下一条传入消息将覆盖前一条消息。<br>1：接收 FIFO 上溢后锁定。接收 FIFO 装满后，下一条传入消息将被丢弃。 |
| 位 2 | TXFP | 发送 FIFO 优先级<br>此位用于控制在几个邮箱同时挂起时的发送顺序。<br>0：优先级由消息标识符确定<br>1：优先级由请求顺序（时间顺序）确定 |
| 位 1 | SLEEP | 睡眠模式请求<br>此位由软件置 1，用于请求 CAN 硬件进入睡眠模式。一旦当前 CAN 活动（发送或接收 CAN 帧）结束，即进入睡眠模式。<br>此位由软件清零时，将退出睡眠模式。<br>当 AWUM 位置 1 以及在 CAN RX 信号上检测到 SOF 位时，硬件即将此位清零。<br>复位后，此位将置 1 - CAN 启动睡眠模式。 |
| 位 0 | INRQ | 初始化请求<br>软件通过将此位清零，来将硬件切换到正常模式。一旦在 Rx 信号上监测到连续 11 个隐性位， CAN 硬件即完成同步并准备进行发送和接收。硬件通过将 CAN_MSR 寄存器的 INAK 位清零来指示此事件。<br>软件通过将此位置 1 来请求 CAN 硬件进入初始化模式。一旦软件将 INRQ 位置 1， CAN 硬件将等待当前 CAN 活动（发送或接收）结束，然后进入初始化模式。硬件通过将 CAN_MSR 寄存器的 INAK 位置 1 来指示此事件。 |

**CAN 主状态寄存器(CAN_MSR)**

偏移地址： 0x04

复位值：0x0000 0C02

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | RX | SAMP | RXM | TXM | Reserved | | | SLAKI | WKUI | ERRI | SLAK | INAK |
| | | | | r | r | r | r | | | | rc_w1 | rc_w1 | rc_w1 | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:12 | 保留 | 必须保持复位值。 |
| 位 11 | RX | CAN Rx 信号<br>监视 CAN_RX 引脚的实际值。 |
| 位 10 | SAMP | 上一个采样点<br>上一个采样点的 RX 值（当前接收的位值）。 |
| 位 9 | RXM | 接收模式<br>CAN 硬件当前为接收器。 |
| 位 8 | TXM | 发送模式<br>CAN 硬件当前为发送器。 |
| 位 7:5 | 保留 | 必须保持复位值。 |
| 位 4 | SLAKI | 睡眠确认中断<br>当 SLKIE=1 时，硬件将此位置 1，以指示 bxCAN 已经进入睡眠模式。如果 CAN_IER 寄存器的 SLKIE 位置 1，则当此位置 1 后将产生状态改变中断。<br>SLAK 清零后，此位由软件或硬件清零。<br>*注意：SLKIE=0 时，无法对 SLAKI 进行轮询。在这种情况下，可以轮询 SLAK 位。* |
| 位 3 | WKUI | 唤醒中断<br>此位由硬件置 1，用于指示在 CAN 硬件处于睡眠模式期间检测到一个 SOF 位。如果 CAN_IER 寄存器的 WKUIE 位置 1，则当此位置 1 后将产生状态改变中断。<br>此位由软件清零。 |
| 位 2 | ERRI | 错误中断<br>如果在检测到错误时 CAN_ESR 的一个位置 1，并且使能 CAN_IER 中的相应中断，则硬件将此位置 1。如果 CAN_IER 寄存器的 ERRIE 位置 1，则当此位置 1 后将产生状态改变中断。<br>此位由软件清零。 |
| 位 1 | SLAK | 睡眠确认<br>此位由硬件置 1，用于向软件指示 CAN 硬件此时处于睡眠模式。此位可确认软件的睡眠模式请求（CAN_MCR 寄存器的 SLEEP 位置 1）。<br>CAN 硬件退出睡眠模式（以在 CAN 总线上进行同步）时，此位由硬件清零。要进行同步，硬件必须在 CAN RX 信号上监测到由 11 个连续隐性位组成的一个序列。<br>*注意：CAN_MCR 寄存器的 SLEEP 位清零时，将触发退出睡眠模式程序。有关 SLEEP 位清零的详细信息，请参阅 CAN_MCR 寄存器 AWUM 位的说明。* |
| 位 0 | INAK | 初始化确认<br>此位由硬件置 1，用于向软件指示 CAN 硬件此时处于初始化模式。此位可确认软件的初始化请求（CAN_MCR 寄存器的 INRQ 位置 1）。<br>CAN 硬件退出初始化模式（以在 CAN 总线上进行同步）时，此位由硬件清零。要进行同步，硬件必须在 CAN RX 信号上监测到由 11 个连续隐性位组成的一个序列。 |

### CAN 发送状态寄存器(CAN_TSR)

偏移地址： 0x08

复位值：0x1C00 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LOW2 | LOW1 | LOW0 | TME2 | TME1 | TME0 | CODE[1:0] | | ABRQ2 | Reserved | | | TERR2 | ALST2 | TXOK2 | RQCP2 |
| r | r | r | r | r | r | r | r | rs | | | | rc_w1 | rc_w1 | rc_w1 | rc_w1 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ABRQ1 | Reserved | | | TERR1 | ALST1 | TXOK1 | RQCP1 | ABRQ0 | Reserved | | | TERR0 | ALST0 | TXOK0 | RQCP0 |
| rs | | | | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rs | | | | rc_w1 | rc_w1 | rc_w1 | rc_w1 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31 | LOW2 | 邮箱 2 最低优先级标志<br>当多个邮箱挂起等待发送且邮箱 2 优先级最低时，此位由硬件置 1。 |
| 位 30 | LOW1 | 邮箱 1 最低优先级标志<br>当多个邮箱挂起等待发送且邮箱 1 优先级最低时，此位由硬件置 1。 |
| 位 29 | LOW0 | 邮箱 0 最低优先级标志<br>当多个邮箱挂起等待发送且邮箱 0 优先级最低时，此位由硬件置 1。<br>注意： 当只有一个邮箱挂起时，LOW[2:0] 位置为零。 |
| 位 28 | TME2 | 发送邮箱 2 空<br>当邮箱 2 没有挂起的发送请求时，此位由硬件置 1。 |
| 位 27 | TME1 | 发送邮箱 1 空<br>当邮箱 1 没有挂起的发送请求时，此位由硬件置 1。 |
| 位 26 | TME0 | 发送邮箱 0 空<br>当邮箱 0 没有挂起的发送请求时，此位由硬件置 1。 |
| 位 25:24 | CODE[1:0] | 邮箱代码<br>如果至少一个发送邮箱空闲，代码值等于下一个空闲发送邮箱的编号。<br>如果所有发送邮箱均挂起，则代码值等于优先级最低的发送邮箱的编号。 |
| 位 23 | ABRQ2 | 邮箱 2 中止请求<br>由软件置 1，用于中止相应邮箱的发送请求。<br>邮箱变为空后，此位由硬件清零。<br>邮箱未挂起等待发送时，将此位置 1 没有任何作用。 |
| 位 22:20 | 保留 | 必须保持复位值。 |
| 位 19 | TERR2 | 邮箱 2 发送错误<br>如果上一次发送因错误而失败，此位将置 1。 |
| 位 18 | ALST2 | 邮箱 2 仲裁丢失<br>如果上一次发送因仲裁丢失而失败，此位将置 1。 |
| 位 17 | TXOK2 | 邮箱 2 发送成功<br>每次发送尝试后，硬件都将更新此位。<br>0：上一次发送失败<br>1：上一次发送成功<br>当邮箱 2 的发送请求成功完成时，此位由硬件置 1。 |
| 位 16 | RQCP2 | 邮箱 2 请求完成<br>最后一个请求（发送或中止）执行完毕时，由硬件置 1。<br>由软件通过写入“1”清零，或是在发生发送请求（CAN_TMID2R 寄存器中的 TXRQ2 位置 1）时由硬件清零。 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 如果将此位清零，邮箱 2 的所有状态位（TXOK2、ALST2 和 TERR2）都将清零。 |
| 位 15 | ABRQ1 | 邮箱 1 中止请求<br>由软件置 1，用于中止相应邮箱的发送请求。<br>邮箱变为空后，此位由硬件清零。<br>邮箱未挂起等待发送时，将此位置 1 没有任何作用。 |
| 位 14:12 | 保留 | 必须保持复位值。 |
| 位 11 | TERR1 | 邮箱 1 发送错误<br>如果上一次发送因错误而失败，此位将置 1。 |
| 位 10 | ALST1 | 邮箱 1 仲裁丢失<br>如果上一次发送因仲裁丢失而失败，此位将置 1。 |
| 位 9 | TXOK1 | 邮箱 1 发送成功<br>每次发送尝试后，硬件都将更新此位。<br>0：上一次发送失败<br>1：上一次发送成功<br>当邮箱 1 的发送请求成功完成时，此位由硬件置 1。 |
| 位 8 | RQCP1 | 邮箱 1 请求完成<br>最后一个请求（发送或中止）执行完毕时，由硬件置 1。<br>由软件通过写入“1”清零，或是在发生发送请求（CAN_TI1R 寄存器中的 TXRQ1 位）时由硬件清零。<br>如果将此位清零，邮箱 1 的所有状态位（TXOK1、ALST1 和 TERR1）都将清零。 |
| 位 7 | ABRQ0 | 邮箱 0 中止请求<br>由软件置 1，用于中止相应邮箱的发送请求。<br>邮箱变为空后，此位由硬件清零。<br>邮箱未挂起等待发送时，将此位置 1 没有任何作用。 |
| 位 6:4 | 保留 | 必须保持复位值。 |
| 位 3 | TERR0 | 邮箱 0 发送错误<br>如果上一次发送因错误而失败，此位将置 1。 |
| 位 2 | ALST0 | 邮箱 0 仲裁丢失<br>如果上一次发送因仲裁丢失而失败，此位将置 1。 |
| 位 1 | TXOK0 | 邮箱 0 发送成功<br>每次发送尝试后，硬件都将更新此位。<br>0：上一次发送失败<br>1：上一次发送成功<br>当邮箱 1 的发送请求成功完成时，此位由硬件置 1。 |
| 位 0 | RQCP0 | 邮箱 0 请求完成<br>最后一个请求（发送或中止）执行完毕时，由硬件置 1。<br>由软件通过写入“1”清零，或是在发生发送请求（CAN_TI0R 寄存器的 TXRQ0 位置 1）时由硬件清零。<br>如果将此位清零，邮箱 0 的所有状态位（TXOK0、ALST0 和 TERR0）都将清零。 |

**CAN 接收 FIFO 0 寄存器(CAN_RF0R)**

偏移地址： 0x0C

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | RFOM0 | FOVR0 | FULL0 | Reserved | FMP0[1:0] | |
| | | | | | | | | | | rs | rc_w1 | rc_w1 | | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:6 | 保留 | 必须保持复位值。 |
| 位 5 | RFOM0 | 释放 FIFO 0 输出邮箱<br>由软件置 1，用于释放 FIFO 的输出邮箱。FIFO 中至少有一条消息挂起时，才能释放输出邮箱。FIFO 为空时，将此位置 1 没有任何作用。如果 FIFO 中至少有两条消息挂起，软件必须释放输出邮箱，才能访问下一条消息。<br>输出邮箱释放后，此位由硬件清零。 |
| 位 4 | FOVR0 | FIFO 0 上溢<br>FIFO 填满时，如果接收到新消息并且通过筛选器，此位将由硬件置 1。<br>此位由软件清零。 |
| 位 3 | FULL0 | FIFO 0 满<br>FIFO 中存储三条消息后，由硬件置 1。<br>此位由软件清零。 |
| 位 2 | 保留 | 必须保持复位值。 |
| 位 1:0 | FMP0[1:0] | FIFO 0 消息挂起<br>这些位用于指示接收 FIFO 中挂起的消息数。<br>硬件每向 FIFO 存储一条新消息，FMP 即会增加。软件每次通过将 RFOM0 位置 1 来释放输出邮箱，FMP 即会减小。 |

### CAN 接收 FIFO 1 寄存器(CAN_RF1R)

偏移地址： 0x10

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | RFOM1 | FOVR1 | FULL1 | Reserved | FMP1[1:0] | |
| | | | | | | | | | | rs | rc_w1 | rc_w1 | | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:6 | 保留 | 必须保持复位值。 |
| 位 5 | RFOM1 | 释放 FIFO 1 输出邮箱<br>由软件置 1，用于释放 FIFO 的输出邮箱。FIFO 中至少有一条消息挂起时，才能释放输出邮箱。FIFO 为空时，将此位置 1 没有任何作用。如果 FIFO 中至少有两条消息挂起，软件必须释放输出邮箱，才能访问下一条消息。<br>输出邮箱释放后，此位由硬件清零。 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 4 | FOVR1 | FIFO 1 上溢<br>FIFO 填满时，如果接收到新消息并且通过筛选器，此位将由硬件置 1。<br>此位由软件清零。 |
| 位 3 | FULL1 | FIFO 1 满<br>FIFO 中存储三条消息后，由硬件置 1。<br>此位由软件清零。 |
| 位 2 | 保留 | 必须保持复位值。 |
| 位 1:0 | FMP1[1:0] | FIFO 1 消息挂起<br>这些位用于指示接收 FIFO 中挂起的消息数。<br>硬件每向 FIFO 存储一条新消息, FMP 即会增加。软件每次通过将 RFOM1 位置 1 来释放输出邮箱，FMP 即会减小。 |

**CAN 中断使能寄存器(CAN_IER)**

偏移地址： 0x14

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ERRIE | Reserved | | | IECIE | BOFIE | EPVIE | EWGIE | Reserved | FOVIE1 | FFIE1 | FMPIE1 | FOVIE0 | FFIE0 | FMPIE0 | TMEIE |
| rw | | | | rw | rw | rw | rw | | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:18 | 保留 | 必须保持复位值。 |
| 位 17 | SLKIE | 睡眠中断使能<br>0：SLAKI 位置 1 时不产生中断。<br>1：SLAKI 位置 1 时产生中断。 |
| 位 16 | WKUIE | 唤醒中断使能<br>0：WKUI 置 1 时不产生中断。<br>1：WKUI 位置 1 时产生中断。 |
| 位 15 | ERRIE | 错误中断使能<br>0：CAN_ESR 中有挂起的错误状况时，不会产生中断。<br>1：CAN_ESR 中有挂起的错误状况时，会产生中断。 |
| 位 14:12 | 保留 | 必须保持复位值。 |
| 位 11 | LECIE | 上一个错误代码中断使能<br>0：如果在检测到错误后硬件将 LEC[2:0] 中的错误代码置 1，则不会将 ERRI 位置 1。<br>1: 如果在检测到错误后硬件将 LEC[2:0] 中的错误代码置 1, 会将 ERRI 位置 1。 |
| 位 10 | BOFIE | 总线关闭中断使能<br>0：BOFF 置 1 时，不会将 ERRI 位置 1。<br>1：BOFF 置 1 时，将 ERRI 位置 1。 |
| 位 9 | EPVIE | 错误被动中断使能<br>0：EPVF 置 1 时，不会将 ERRI 位置 1。 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 1：EPVF 置 1 时，将 ERRI 位置 1。 |
| 位 8 | EWGIE | 错误警告中断使能<br>0：EWGF 置 1 时，不会将 ERRI 位置 1。<br>1：EWGF 置 1 时，将 ERRI 位置 1。 |
| 位 7 | 保留 | 必须保持复位值。 |
| 位 6 | FOVIE1 | FIFO 上溢中断使能<br>0：FOVR 置 1 时不产生中断。<br>1：FOVR 置 1 时产生中断。 |
| 位 5 | FFIE1 | FIFO 满中断使能<br>0：FULL 位置 1 时不产生中断。<br>1：FULL 位置 1 时产生中断。 |
| 位 4 | FMPIE1 | FIFO 消息挂起中断使能<br>0：FMP[1:0] 位的状态不是 00b 时，不产生中断。<br>1：FMP[1:0] 位的状态不是 00b 时，产生中断。 |
| 位 3 | FOVIE0 | FIFO 上溢中断使能<br>0：FOVR 位置 1 时不产生中断。<br>1：FOVR 位置 1 时产生中断。 |
| 位 2 | FFIE0 | FIFO 满中断使能<br>0：FULL 位置 1 时不产生中断。<br>1：FULL 位置 1 时产生中断。 |
| 位 1 | FMPIE0 | FIFO 消息挂起中断使能<br>0：FMP[1:0] 位的状态不是 00b 时，不产生中断。<br>1：FMP[1:0] 位的状态不是 00b 时，产生中断。 |
| 位 0 | TMEIE | 发送邮箱空中断使能<br>0：RQCPx 位置 1 时不产生中断。<br>1：RQCPx 位置 1 时产生中断。 |

### CAN 错误状态寄存器(CAN_ESR)

偏移地址： 0x18

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| REC[7:0] | | | | | | | | TEC[7:0] | | | | | | | |
| r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | | LEC[2:0] | | | Reserved | BOFF | EPVF | EWGF | |
| | | | | | | | | rw | rw | rw | | r | r | r | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:24 | REC[7:0] | 接收错误计数器<br>CAN 协议故障隔离机制的实施部分。如果接收期间发生错误，该计数器按 1 或 8 递增，具体取决于 CAN 标准所定义的错误状况。每次成功接收后，该计数器按 1 递减，如果其数值大于 128，则复位为 120。计数器值超过 127 时，CAN 控制器进入错误被动状态。 |
| 位 23:16 | TEC[7:0] | 9 位发送错误计数器最低有效字节 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | CAN 协议故障隔离机制的实施部分。 |
| 位 15:7 | 保留 | 必须保持复位值。 |
| 位 6:4 | LEC[2:0] | 上一个错误代码<br>该字段由硬件置 1，其中的代码指示 CAN 总线上检测到的上一个错误的错误状况。如果消息成功传送（接收或发送）且未发生错误，该字段将清为“0”。LEC[2:0] 位可由软件置为 0b111 值。这些位由硬件更新，以指示当前通信状态。<br>000：无错误<br>001：填充错误<br>010：格式错误<br>011：确认错误<br>100：位隐性错误<br>101：位显性错误<br>110：CRC 错误<br>111：由软件置 1 |
| 位 3 | 保留 | 必须保持复位值。 |
| 位 2 | BOFF | 总线关闭标志<br>此位由硬件在进入睡眠状态时置 1。TEC 上溢（超过 255）时，进入总线关闭状态 |
| 位 1 | EPVF | 错误被动标志<br>达到错误被动极限（接收错误计数器或发送错误计数器 > 127）时，此位由硬件置 1。 |
| 位 0 | EWGF | 错误警告标志<br>达到警告极限时，此位由硬件置 1（接收错误计数器或发送错误计数器³96）。 |

**CAN 错误状态寄存器(CAN_BTR)**

偏移地址： 0x1C

复位值：0x0123 0000

只有 CAN 硬件处于初始化模式时，才能由软件访问此寄存器。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SILM | LBKM | Reserved | | | | SJW[1:0] | | Reserved | TS2[2:0] | | | TS1[3:0] | | | |
| rw | rw | | | | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31 | SILM | 静默模式（调试）<br>0：正常工作<br>1：静默模式 |
| 位 30 | LBKM | 环回模式（调试）<br>0：禁止环回模式<br>1：使能环回模式 |
| 位 29:26 | 保留 | 必须保持复位值。 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 25:24 | SJW[1:0] | 再同步跳转宽度<br>这些位定义 CAN 硬件在执行再同步时最多可以将位加长或缩短的时间片数目。<br>$t_{RJW} = t_{CAN} \times (SJW[1:0] + 1)$ |
| 位 23 | 保留 | 必须保持复位值。 |
| 位 22:20 | TS2[2:0] | 时间段 2<br>这些位定义时间段 2 中的时间片数目。<br>$t_{BS2} = t_{CAN} \times (TS2[2:0] + 1)$ |
| 位 19:16 | TS1[3:0]: | 时间段 1<br>这些位定义时间段 1 中的时间片数目。<br>$t_{BS1} = t_{CAN} \times (TS1[3:0] + 1)$ |
| 位 15:10 | 保留 | 必须保持复位值。 |
| 位 9:0 | BRP[9:0] | 波特率预分频器<br>这些位定义一个时间片的长度。<br>$t_q = (BRP[9:0]+1) \times T_{pclk}$ |

### 24.9.3. CAN 邮箱寄存器

本章介绍发送和接收邮箱的寄存器。

除了以下情况外，发送邮箱和接收邮箱使用相同的寄存器：

- CAN_RDTxR 寄存器中的 FMI 字段。
- 接收邮箱始终处于写保护状态。
- 发送邮箱仅在为空时才处于可写状态（CAN_TSR 寄存器中的相应 TME 位置 1）。

发送邮箱共有 3 个，接收邮箱共有 2 个。每个接收邮箱允许访问一个深度为 3 级的 FIFO，访问仅限于 FIFO 中最早接收到的消息。

每个邮箱由 4 个寄存器组成。

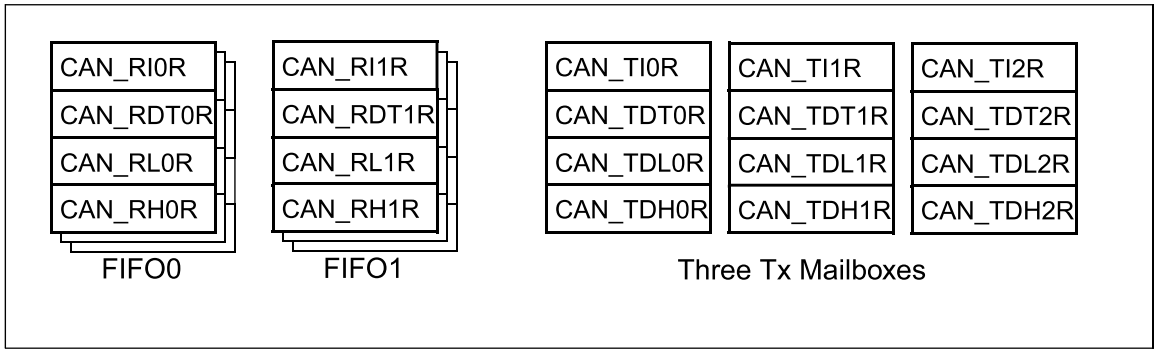



图 24-16 每个邮箱由 4 个寄存器组成

#### CAN 发送邮箱标识符寄存器 (CAN_TIxR) (x=0..2)

偏移地址： 0x180、 0x190、 0x1A0

复位值： 0xXXXX XXXX（位 0 除外， TXRQ = 0）

当邮箱处于发送挂起状态（ TMEx 复位）时，所有发送寄存器均为写保护状态。

该寄存器还会进行发送请求控制（位 0） - 复位值 0。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:21 | STID[10:0]/<br>EXID[28:18] | 标准标识符或扩展标识符<br>标准标识符或扩展标识符的 MSB（取决于 IDE 位的值）。 |
| 位 20:3 | EXID[17:0] | 扩展标识符<br>扩展标识符的 LSB。 |
| 位 2 | IDE | 标识符扩展<br>此位用于定义邮箱中消息的标识符类型。<br>0：标准标识符。<br>1：扩展标识符。 |
| 位 1 | RTR | 远程发送请求<br>0：数据帧<br>1：遥控帧 |
| 位 0 | TXRQ | 发送邮箱请求<br>由软件置 1，用于请求发送相应邮箱的内容。<br>邮箱变为空后，此位由硬件清零。 |

**CAN 邮箱数据长度控制和时间戳寄存器 (CAN_TDTxR) (x=0..2)**

偏移地址： 0x184、 0x194、 0x1A4

复位值： 0xXXXX XXXX

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:16 | TIME[15:0] | 消息时间戳<br>此字段包含在进行 SOF 发送时所捕获的 16 位定时器值。 |
| 位 15:9 | 保留 | 必须保持复位值。 |
| 位 8 | TGT | 发送全局时间<br>只有硬件处于时间触发通信模式（CAN_MCR 寄存器的 TTCM 位置 1）时，此位才会激活。<br>0：不发送时间戳 TIME[15:0]。<br>1：在 8 字节消息的最后两个数据字节中发送时间戳 TIME[15:0] 的值。数据字节 7 对应 TIME[7:0]，数据字节 6 对应 TIME[15:8]，该值将替换 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | CAN_TDHxR[31:16] 寄存器<br>（DATA6[7:0] 和 DATA7[7:0]）中写入的数据。DLC 必须编程为 8，才能通过 CAN 总线发送这两个字节。 |
| 位 7:4 | 保留 | 必须保持复位值。 |
| 位 3:0 | DLC[3:0] | 数据长度代码<br>该字段定义数据帧或遥控帧请求中的数据字节数。<br>一条消息可以包含 0 到 8 个数据字节，具体取决于 DLC 字段的值。 |

### CAN 邮箱数据低位寄存器 (CAN_TDLxR) (x=0..2)

偏移地址： 0x188、 0x198、 0x1A8

复位值： 0xXXXX XXXX

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA3[7:0] | | | | | | | | DATA2[7:0] | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA1[7:0] | | | | | | | | DATA0[7:0] | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:24 | DATA3[7:0] | 数据字节 3<br>消息的数据字节 3。 |
| 位 23:16 | DATA2[7:0] | 数据字节 2<br>消息的数据字节 2。 |
| 位 15:8 | DATA1[7:0] | 数据字节 1<br>消息的数据字节 1。 |
| 位 7:0 | DATA0[7:0] | 数据字节 0<br>消息的数据字节 0。<br>一条消息可以包含 0 到 8 个数据字节，从字节 0 开始。 |

### CAN 邮箱数据高位寄存器 (CAN_TDHxR) (x=0..2)

偏移地址： 0x18C、 0x19C、 0x1AC

复位值： 0xXXXX XXXX

当邮箱未处于空状态时，该寄存器的所有位均为写保护状态。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA7[7:0] | | | | | | | | DATA6[7:0] | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA5[7:0] | | | | | | | | DATA4[7:0] | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:24 | DATA7[7:0] | 数据字节 7<br>消息的数据字节 7。 |
| 位 23:16 | DATA6[7:0] | 数据字节 6<br>消息的数据字节 6。 |
| 位 15:8 | DATA5[7:0] | 数据字节 5<br>消息的数据字节 5。 |
| 位 7:0 | DATA4[7:0] | 数据字节 4<br>消息的数据字节 4。<br>一条消息可以包含 0 到 8 个数据字节，从字节 0 开始。 |

### CAN 接收 FIFO 邮箱标识符寄存器 (CAN_RIxR) (x=0..1)

偏移地址： 0x1B0 、0x1C0

复位值： 0xXXXX XXXX

所有接收寄存器均受到写保护。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STID[10:0]/EXID[28:18] | | | | | | | | | | | EXID[17:13] | | | | |
| r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:21 | STID[10:0]/<br>EXID[28:18] | 标准标识符或扩展标识符<br>标准标识符或扩展标识符的 MSB（取决于 IDE 位的值）。 |
| 位 20:3 | EXID[17:0] | 扩展标识符<br>扩展标识符的 LSB。 |
| 位 2 | IDE | 标识符扩展<br>此位用于定义邮箱中消息的标识符类型。<br>0：标准标识符。<br>1：扩展标识符。 |
| 位 1 | RTR | 远程发送请求<br>0：数据帧<br>1：遥控帧 |
| 位 0 | 保留 | 必须保持复位值。 |

### CAN 接收 FIFO 邮箱数据长度控制和时间戳寄存器 (CAN_RDTxR) (x=0..1)

偏移地址： 0x1B4 、0x1C4

复位值： 0xXXXX XXXX

所有接收寄存器均受到写保护。

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:16 | TIME[15:0] | 消息时间戳<br>此字段包含在进行 SOF 检测时所捕获的 16 位定时器值。 |
| 位 15:8 | FMI[7:0] | 筛选器匹配索引<br>该寄存器包含筛选器索引，邮箱中存储的消息需要经过该筛选器。 |
| 位 7:4 | 保留 | 必须保持复位值。 |
| 位 3:0 | DLC[3:0] | 数据长度代码<br>该字段定义一个数据帧所包含的数据字节数（0 到 8）。如果是远程帧请求，则为 0。 |

### CAN 接收 FIFO 邮箱数据低位寄存器 (CAN_RDLxR) (x=0..1)

偏移地址： 0x1B8 、0x1C8

复位值： 0xXXXX XXXX

当邮箱未处于空状态时，该寄存器的所有位均为写保护状态。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA3[7:0] | | | | | | | | DATA2[7:0] | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA1[7:0] | | | | | | | | DATA0[7:0] | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:24 | DATA3[7:0] | 数据字节 3<br>消息的数据字节 3。 |
| 位 23:16 | DATA2[7:0] | 数据字节 2<br>消息的数据字节 2。 |
| 位 15:8 | DATA1[7:0] | 数据字节 1<br>消息的数据字节 1。 |
| 位 7:0 | DATA0[7:0] | 数据字节 0<br>消息的数据字节 0。<br>一条消息可以包含 0 到 8 个数据字节，从字节 0 开始。 |

### CAN 接收 FIFO 邮箱数据高位寄存器 (CAN_RDHxR) (x=0..1)

偏移地址： 0x1BC、0x1CC

复位值： 0xXXXX XXXX

当邮箱未处于空状态时，该寄存器的所有位均为写保护状态。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA7[7:0] | | | | | | | | DATA6[7:0] | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA5[7:0] | | | | | | | | DATA4[7:0] | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:24 | DATA7[7:0] | 数据字节 7<br>消息的数据字节 7。 |
| 位 23:16 | DATA6[7:0] | 数据字节 6<br>消息的数据字节 6。 |
| 位 15:8 | DATA5[7:0] | 数据字节 5<br>消息的数据字节 5。 |
| 位 7:0 | DATA4[7:0] | 数据字节 4<br>消息的数据字节 4。<br>一条消息可以包含 0 到 8 个数据字节，从字节 0 开始。 |

### 24.9.4. CAN 筛选器寄存器

#### CAN 筛选器主寄存器 (CAN_FMR)

偏移地址： 0x200

复位值：0x2A1C 0E01

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | CAN2SB[5:0] | | | | | | Reserved | | | | | | | |
| | | rw | rw | rw | rw | rw | rw | | | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:14 | 保留 | 必须保持复位值。 |
| 位 13:8 | CAN2SB[5:0] | CAN2 起始存储区<br>这些位将由软件置 1 和清零。它们为处于 0 到 27 范围内的 CAN2 接口（从模式）定义起始存储区。<br>*注意： CAN2SB[5:0] = 28d 时，可以使用 CAN1 的所有筛选器。*<br>*CAN2SB[5:0] 设置为 0 时，不会为 CAN1 分配任何筛选器。* |
| 位 7:1 | 保留 | 必须保持复位值。 |
| 位 0 | FINIT | 筛选器初始化模式<br>筛选器组的初始化模式<br>0：筛选器工作模式。<br>1：筛选器初始化模式。 |

### CAN 筛选器模式寄存器 (CAN_FM1R)

偏移地址： 0x204

复位值：0x00000000

仅当 CAN_FMR 寄存器中设置了筛选器初始化模式 (FINIT=1) 时，才能对此寄存器执行操作

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | FBM27 | FBM26 | FBM25 | FBM24 | FBM23 | FBM22 | FBM21 | FBM20 | FBM19 | FBM18 | FBM17 | FBM16 |
| | | | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FBM15 | FBM14 | FBM13 | FBM12 | FBM11 | FBM10 | FBM9 | FBM8 | FBM7 | FBM6 | FBM5 | FBM4 | FBM3 | FBM2 | FBM1 | FBM0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:28 | 保留 | 必须保持复位值。 |
| 位 27:0 | FBMx | 筛选器模式<br>筛选器 x 的寄存器的模式<br>0：筛选器存储区 x 的两个 32 位寄存器处于标识符屏蔽模式。<br>1：筛选器存储区 x 的两个 32 位寄存器处于标识符列表模式。 |

### CAN 筛选器尺度寄存器 (CAN_FS1R)

偏移地址： 0x20C

复位值：0x00000000

仅当 CAN_FMR 寄存器中设置了筛选器初始化模式 (FINIT=1) 时，才能对此寄存器执行操作

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | FSC27 | FSC26 | FSC25 | FSC24 | FSC23 | FSC22 | FSC21 | FSC20 | FSC19 | FSC18 | FSC17 | FSC16 |
| | | | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FSC15 | FSC14 | FSC13 | FSC12 | FSC11 | FSC10 | FSC9 | FSC8 | FSC7 | FSC6 | FSC5 | FSC4 | FSC3 | FSC2 | FSC1 | FSC0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:28 | 保留 | 必须保持复位值。 |
| 位 27:0 | FSCx | 筛选器尺度配置<br>这些位定义了筛选器 13-0 的尺度配置。<br>0：双 16 位尺度配置<br>1：单 32 位尺度配置 |

### CAN 筛选器 FIFO 分配寄存器 (CAN_FFA1R)

偏移地址： 0x214

复位值：0x00000000

仅当 CAN_FMR 寄存器中设置了筛选器初始化模式 (FINIT=1) 时，才能对此寄存器执行操作

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | FFA27 | FFA26 | FFA25 | FFA24 | FFA23 | FFA22 | FFA21 | FFA20 | FFA19 | FFA18 | FFA17 | FFA16 |
| | | | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FFA15 | FFA14 | FFA13 | FFA12 | FFA11 | FFA10 | FFA9 | FFA8 | FFA7 | FFA6 | FFA5 | FFA4 | FFA3 | FFA2 | FFA1 | FFA0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:28 | 保留 | 必须保持复位值。 |
| 位 27:0 | FFAx | 筛选器 x 的筛选器 FIFO 分配<br>通过此筛选器的消息将存储在指定的 FIFO 中。<br>0：筛选器分配到 FIFO 0<br>1：筛选器分配到 FIFO 1 |

**CAN 筛选器激活寄存器 (CAN_FA1R)**

偏移地址： 0x21C

复位值：0x00000000

仅当 CAN_FMR 寄存器中设置了筛选器初始化模式 (FINIT=1) 时，才能对此寄存器执行操作

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | FACT27 | FACT26 | FACT25 | FACT24 | FACT23 | FACT22 | FACT21 | FACT20 | FACT19 | FACT18 | FACT17 | FACT16 |
| | | | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FACT15 | FACT14 | FACT13 | FACT12 | FACT11 | FACT10 | FACT9 | FACT8 | FACT7 | FACT6 | FACT5 | FACT4 | FACT3 | FACT2 | FACT1 | FACT0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:28 | 保留 | 必须保持复位值。 |
| 位 27:0 | FACTx | 筛选器激活<br>软件将此位置 1 可激活筛选器 x。要修改筛选器 x 的寄存器 (CAN_FxR[0:7]), 必须将 FACTx 位清零或将 CAN_FMR 寄存器的 FINIT 位置 1。<br>0：筛选器 x 未激活<br>1：筛选器 x 激活 |

**CAN 筛选器组 i 寄存器 x (CAN_FiRx)（i=0..27 ， x=1 ， 2）**

偏移地址： 0x240..0x31C

复位值：0x00000000

共有 28 个筛选器组， i=0 ..27。每个筛选器组 i 由两个 32 位寄存器组成，即 CAN_FiR[2:1]。

仅当 CAN_FAxR 寄存器的 FACTx 位清零，或者 CAN_FMR 寄存器的 FINIT 位置 1 时，才能修改此寄存器

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FB31 | FB30 | FB29 | FB28 | FB27 | FB26 | FB25 | FB24 | FB23 | FB22 | FB21 | FB20 | FB19 | FB18 | FB17 | FB16 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FB15 | FB14 | FB13 | FB12 | FB11 | FB10 | FB9 | FB8 | FB7 | FB6 | FB5 | FB4 | FB3 | FB2 | FB1 | FB0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | FB[31:0] | 筛选器位<br>**标识符**<br>寄存器的每一位用于指定预期标识符相应位的级别。<br>0：需要显性位<br>1：需要隐性位<br>**掩码**<br>寄存器的每一位用于指定相关标识符寄存器的位是否必须与预期标识符的相应位匹配。<br>0：无关，不使用此位进行比较。<br>1：必须匹配，传入标识符的此位必须与筛选器相应标识符寄存器中指定的级别相同。<br><br>*注意：*<br>*每个寄存器的功能可能因筛选器的尺度和模式配置而异。*<br>*掩码模式中，掩码/标识符寄存器的位映射与标识符列表模式中的相同。* |

# 25. 以太网(ETH)

## 25.1. 以太网简介

以太网提供了可配置、灵活的外设，用以满足客户的各种应用需求。它支持与外部物理层(PHY) 相连的两个工业标准接口：默认情况下使用的介质独立接口 (MII)（在 IEEE 802.3 规范中定义）和简化介质独立接口 (RMII)。它有多种应用领域，例如交换机、网络接口卡等。

以太网遵守以下标准：

- IEEE 802.3-2002，用于以太网 MAC。
- IEEE 1588-2008 标准，用于规定联网时钟同步的精度。
- AMBA 2.0，用于 AHB 主/从端口。
- RMII 联盟的 RMII 规范。

## 25.2. 以太网主要特性

以太网 (ETH) 外设包括以下特性（按类列出）：

### 25.2.1. MAC 内核特性

- 支持外部 PHY 接口实现 10/100 Mbit/s 数据传输速率；
- 通过符合 IEEE 802.3 的 MII 接口与外部快速以太网 PHY 进行通信；
- 支持全双工和半双工操作：
  - 支持适用于半双工操作的 CSMA/CD 协议；
  - 支持适用于全双工操作的 IEEE 802.3x 流量控制；
  - 全双工操作时可以将接收的暂停控制帧转发到用户应用程序；
  - 半双工操作时提供背压流量控制；
  - 全双工操作中如果流量控制输入信号消失，将自动发送零时间片暂停帧；
- 报头和帧起始数据 (SFD) 在发送路径中插入、在接收路径中删除；
- 可逐帧控制 CRC 和 pad 自动生成；
- 接收帧时可自动去除 pad/CRC；
- 可编程帧长度，支持高达 16 KB 的巨型帧；
- 可编程帧间隔（40-96 位时间，以 8 为步长）
- 支持多种灵活的地址过滤模式：
  - 高达四个 48 位完美 (DA) 地址过滤器，对每个字节进行掩码操作；
  - 高达三次 48 位 SA 地址比较检查，对每个字节进行掩码操作；
  - 64 位 Hash 滤波器（可选），适用于多播和单播 (DA) 地址；
  - 可传送所有多播地址帧；
  - 支持混合模式，因此可传送所有帧，无需为网络监视进行过滤；
  - 传送所有传入数据包时（每次过滤时）均附有一份状态报告；

- 为发送和接收数据包分别返回 32 位状态；
- 支持对接收帧进行 IEEE 802.1Q VLAN 变量检测；
- 为应用程序提供单独的发送、接收和控制接口；
- 支持通过 RMON/MIB 计数器 (RFC2819/RFC2665) 进行强制网络统计；
- 使用 MDIO 接口配置和管理 PHY 设备；
- 检测 LAN 唤醒帧和 AMD Magic Packet™ 帧；
- 在接收功能中支持对接收到的由以太网帧封装的 IPv4 和 TCP 数据包进行校验和卸载；
- 在增强型接收功能中支持检查 IPv4 头校验和以及在 IPv4 或 IPv6 数据包中封装的 TCP、UDP 或 ICMP 校验和；
- 支持以太网帧时间戳（请参见 IEEE 1588-2008）。每个帧的发送或接收状态下给出 64 位时间戳；
- 两组 FIFO：一个具有可编程阈值功能的 2 KB 发送 FIFO 和一个具有可配置阈值（默认为 64 个字节）功能的 2 KB 接收 FIFO；
- 接收 FIFO 进行多帧存储时，通过在 EOF 传输后向接收 FIFO 插入接收状态矢量，从而使得接收 FIFO 无需存储这些帧的接收状态；
- 在存储转发模式下，可以在接收时过滤所有的错误帧，但不将这些错误帧转发给应用程序；
- 可以转发过小的好帧；
- 为接收 FIFO 中丢失或损坏的帧（由于溢出）生成脉冲，借此支持数据统计；
- 向 MAC 内核发送数据时支持存储转发机制；
- 根据接收 FIFO 填充（阈值可配置）级别自动生成要发送至 MAC 内核的暂停帧控制或背压信号；
- 发送时处理冲突帧的自动重新发送；
- 丢弃延迟冲突、过度冲突、过度延迟和下溢条件下的帧；
- 通过软件控制刷新 Tx FIFO；
- 计算 IPv4 头校验和与 TCP、 UDP 或 ICMP 校验和并将其插入在存储转发模式下发送的帧中；
- 支持调试时通过 MII 进行内部回送。

### 25.2.2. DMA 特性

- AHB 从接口中支持所有 AHB 突发类型；
- 软件可以在 AHB 主接口中选择 AHB 突发类型（固定或不确定突发）；
- 可以从 AHB 主端口选择地址对齐突发；
- 通过帧定界符优化以数据包为导向的 DMA 传输；
- 支持对数据缓冲区进行字节对齐寻址；
- 双缓冲区（环）或链表（链接）描述符链接；
- 采用描述符架构可以在 CPU 几乎不干预的情况传输大型数据块；
- 每个描述符可传输高达 8 KB 的数据；
- 报告正常工作和传输错误时的综合状态；
- 可为发送和接收 DMA 引擎单独编程突发大小，以充分利用主总线；
- 可编程中断选项，适用于不同的工作条件；
- 按帧控制发送/接收完成中断；
- 接收引擎和发送引擎间采用循环调度仲裁或固定优先级仲裁；

- 启动/停止模式；
- 当前 Tx/Rx 缓冲区指针作为状态寄存器；
- 当前 Tx/Rx 描述符指针作为状态寄存器。

### 25.2.3. PTP 特性

- 接收帧和发送帧时间戳；
- 粗略校准法和精细校准法；
- 系统时间大于目标时间时触发中断；
- 输出秒脉冲（产品复用功能输出）。

## 25.3. 以太网引脚

下表显示了 MAC 信号和相应的 MII/RMII 信号映射。所有 MAC 信号均映射到 AF11，一些信号映射到不同的 I/O 引脚，这些应在复用功能模式下进行配置。

表 25-1 复用功能映射

| 端口 | AF11 |
|---|---|
| | ETH |
| PA0 | ETH_MII_CRS |
| PA1 | ETH_MII_RX_CLK/ETH_RMII_REF_CLK |
| PA2 | ETH_MDIO |
| PA3 | ETH_MII_COL |
| PA7 | ETH_MII_RX_DV/ETH_RMII_CRS_DV |
| PB0 | ETH_MII_RXD2 |
| PB1 | ETH_MII_RXD3 |
| PB5 | ETH_PPS_OUT |
| PB8 | ETH_MII_TXD3 |
| PB10 | ETH_MII_RX_ER |
| PB11 | ETH_MII_TX_EN/ETH_RMII_TX_EN |
| PB12 | ETH_MII_TXD0/ETH_RMII_TXD0 |
| PB13 | ETH_MII_TXD1/ETH_RMII_TXD1 |
| PB14 | ETH_MII_RX_DV/ETH_RMII_CRS_DV |
| PC1 | ETH_MDC |
| PC2 | ETH_MII_TXD2 |
| PC3 | ETH_MII_TX_CLK |
| PC4 | ETH_MII_RXD0/ETH_RMII_RXD0 |
| PC5 | ETH_MII_RXD1/ETH_RMII_RXD1 |
| PE2 | ETH_MII_TXD3 |
| PG8 | ETH_PPS_OUT |
| PG11 | ETH_MII_TX_EN/ETH_RMII_TX_EN |
| PG13 | ETH_MII_TXD0/ETH_RMII_TXD0 |
| PG14 | ETH_MII_TXD1/ETH_RMII_TXD1 |
| PH2 | ETH_MII_CRS |
| PH3 | ETH_MII_COL |
| PH4 | ETH_MDIO |
| PH6 | ETH_MII_RXD2 |

| 端口 | AF11 |
| --- | --- |
| | ETH |
| PH7 | ETH_MII_RXD3 |
| PI10 | ETH_MII_RX_ER |

## 25.4. 以太网功能说明：SMI、MII 和 RMII

以太网外设包括带专用 DMA 控制器的 MAC 802.3（介质访问控制）。它支持默认情况下使用的介质独立接口 (MII) 和简化介质独立接口 (RMII)，并通过一个选择位在两个接口间进行切换。

DMA 控制器通过 AHB 主从接口与内核和存储器相连。AHB 主接口用于控制数据传输，而 AHB 从接口则用于访问“控制和状态寄存器” (CSR) 的空间。

在进行数据发送时，首先将数据由系统存储器以 DMA 的方式送至发送 FIFO (Tx FIFO) 进行缓冲，再通过 MAC 内核发送。同样，接收 FIFO (Rx FIFO) 则存储通过线路接收的以太网帧，直到这些帧通过 DMA 传送到系统存储器。

以太网外设还包括用于与外部 PHY 通信的 SMI。通过一组配置寄存器，用户可以为 MAC 控制器和 DMA 控制器选择所需模式和功能。

*注意： 当使用以太网时， AHB 时钟频率必须至少为 25 MHz。*

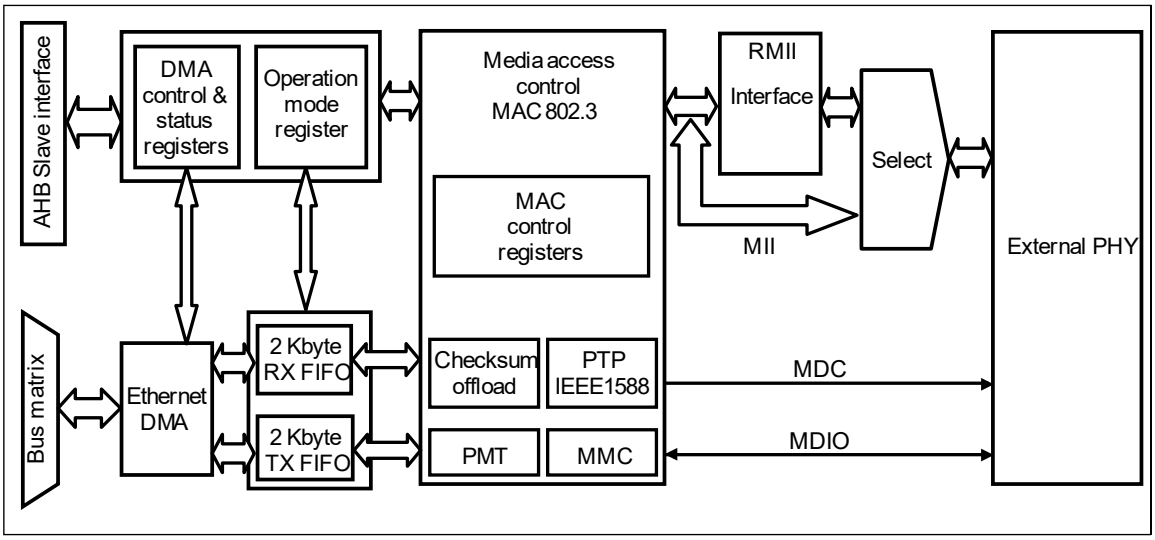



图 25-1 ETH 框图

### 25.4.1. 站管理接口：SMI

站管理接口 (SMI) 允许应用程序通过 2 线时钟和数据线访问任意 PHY 寄存器。该接口支持访问多达 32 个 PHY。

应用程序可以从 32 个 PHY 中选择一个 PHY， 然后从任意 PHY 包含的 32 个寄存器中选择一个寄存器，发送控制数据或接收状态信息。任意给定时间内只能对一个 PHY 中的一个寄存器进行寻址。

MDC 时钟线和 MDIO 数据线在微控制器中均用作复用功能 I/O：

- MDC：周期性时钟，提供以最大频率 2.5 MHz 传输数据时的参考时序。MDC 的最短高电平时间和最短低电平时间必须均为 160 ns。MDC 的最小周期必须为 400 ns。在空闲状态下，SMI 管理接口将 MDC 时钟信号驱动为低电平。
- MDIO：数据输入/输出比特流，用于通过 MDC 时钟信号向/从 PHY 设备同步传输状态信息。

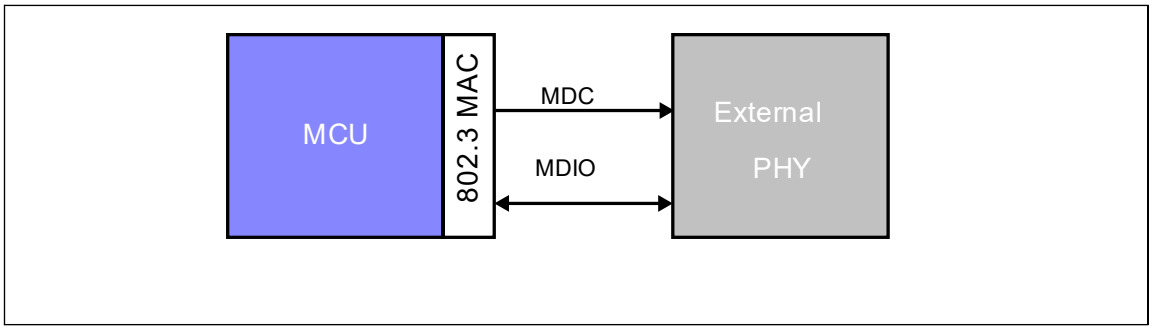



图 25-2 SMI 接口信号

### SMI 帧格式

下表给出了与读操作或写操作有关的帧结构，位传输顺序必须从左到右。

表 25-2 管理帧格式

| | 管理帧字段 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 报头（32 位） | 起始 | 操作 | PADDR | RADDR | TA | 数据（16 位） | 空闲 |
| 读取 | 1...1 | 1 | 10 | ppppp | rrrrr | Z0 | dddddddddddddddd | Z |
| 写入 | 1...1 | 1 | 1 | ppppp | rrrrr | 10 | dddddddddddddddd | Z |

管理帧包括八个字段：

- 报头：每个事务（读取或写入）均可通过报头字段启动，报头字段对应于 MDIO 线上 32 个连续的逻辑“1”位以及 MDC 上的 32 个周期。该字段用于与 PHY 设备建立同步。
- 起始：帧起始由 <01> 模式定义，用于验证线路从默认逻辑“1”状态变为逻辑“0”状态，然后再从逻辑“0”状态变为逻辑“1”状态。
- 操作：定义正在发生的事务（读取或写入）的类型。
- PADDR：PHY 地址有 5 位，可构成 32 个唯一 PHY 地址。最先发送和接收地址的 MSB 位。
- RADDR：寄存器地址有 5 位，从而可在所选 PHY 设备中对 32 个不同的寄存器进行寻址。最先发送和接收地址的 MSB 位。
- TA：周转字段在 RADDR 和 DATA 字段间定义了一个 2 位模式，以避免在读取事务期间出现竞争现象。读取事务时，MAC 控制器将 TA 的 2 个位驱动为 MDIO 线上的高阻态。
- PHY 设备必须将 TA 的第一位驱动为高阻态，将 TA 的第二位驱动为“0”。
- 写入事务时，MAC 控制器针对 TA 字段驱动 <10> 模式。PHY 设备必须将 TA 的 2 个位驱动为高阻态。
- 数据：数据字段为 16 位。最先发送和接收的位必须为 ETH_MIID 寄存器的位 15。
- 空闲： MDIO 线驱动为高阻态。三态驱动器必须禁止，PHY 的上拉电阻使线路保持逻辑“1”状态。

### SMI 写操作

当应用程序将 MII 写入位和繁忙位（在以太网 MAC MII 地址寄存器 (ETH_MACMIIAR) 中）置 1 时，SMI 将通过传输 PHY 地址、PHY 中的寄存器地址以及写入数据（在以太网 MAC MII 数据寄存器 (ETH_MACMIIDR) 中）来触发对 PHY 寄存器进行写操作。事务进行期间，应用程序不应更改 MII 地址寄存器的内容或 MII 数据寄存器。在此期间对 MII 地址寄存器或 MII 数据寄存器执行的写操作将会忽略（繁忙位处于高电平状态），事务将无错完成。写操作完成后，SMI 将通过复位繁忙位进行指示。

下图显示了写操作的帧格式。

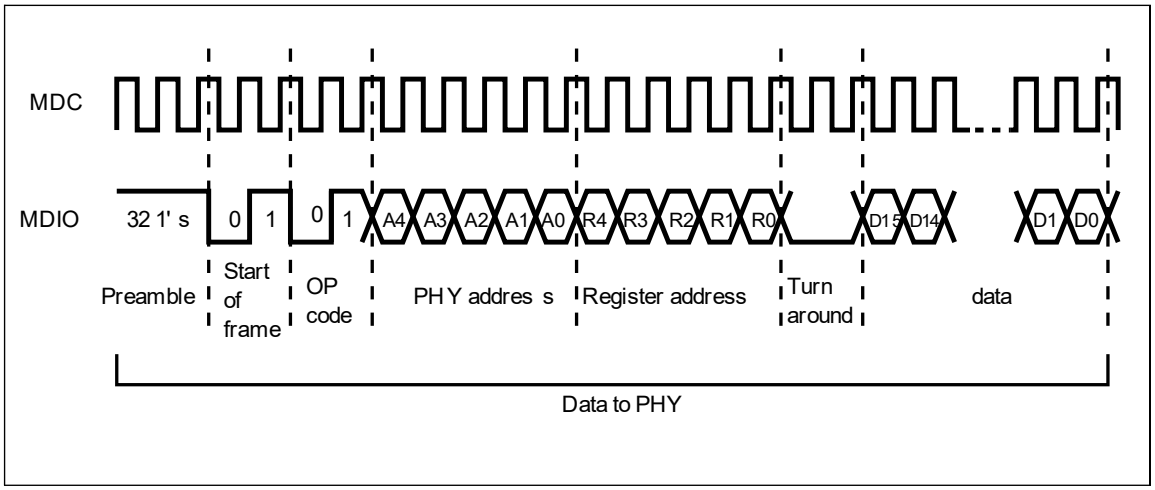



图 25-3 MDIO 时序和帧结构 - 写周期

### SMI 读操作

当用户将以太网 MAC MII 地址寄存器 (ETH_MACMIIAR) 中的 MII 繁忙位置 1、MII 写入位清零时, SMI 将通过传输 PHY 地址和 PHY 中的寄存器地址在 PHY 寄存器中触发读操作。事务进行期间, 应用程序不应更改 MII 地址寄存器的内容或 MII 数据寄存器。在此期间对 MII 地址寄存器或 MII 数据寄存器执行的写操作将会忽略（繁忙位处于高电平状态），事务将无错完成。读操作完成后，SMI 将复位繁忙位，然后用从 PHY 中读取的数据更新 MII 数据寄存器。

下图显示了读操作的帧格式。

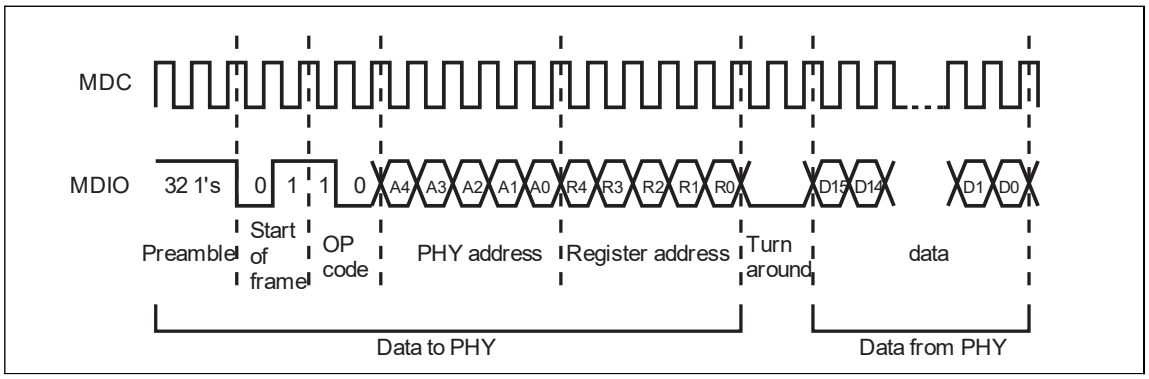



图 25-4 MDIO 时序和帧结构 – 读周期

### SMI 时钟选择

MAC 启动管理写/读操作。 SMI 时钟是一个分频时钟，其时钟源为应用时钟（ AHB 时钟）。分频系数取决于 MII 地址寄存器中设置的时钟范围。

下表显示了如何设置时钟范围。

表 25-3 时钟选择

| 选择 | HCLK 时钟 | MDC 时钟 |
|---|---|---|
| 000 | 60-100 MHz | AHB 时钟/42 |
| 001 | 100-150 MHz | AHB 时钟/62 |
| 010 | 20-35 MHz | AHB 时钟/16 |
| 011 | 35-60 MHz | AHB 时钟/26 |
| 100 | 150~168(or more) MHz | AHB 时钟/102 |
| 101, 110, 111 | 保留 | - |

### 25.4.2. 介质独立接口：MII

介质独立接口 (MII) 定义了 10 Mbit/s 和 100 Mbit/s 的数据传输速率下 MAC 子层与 PHY 之间的互连。

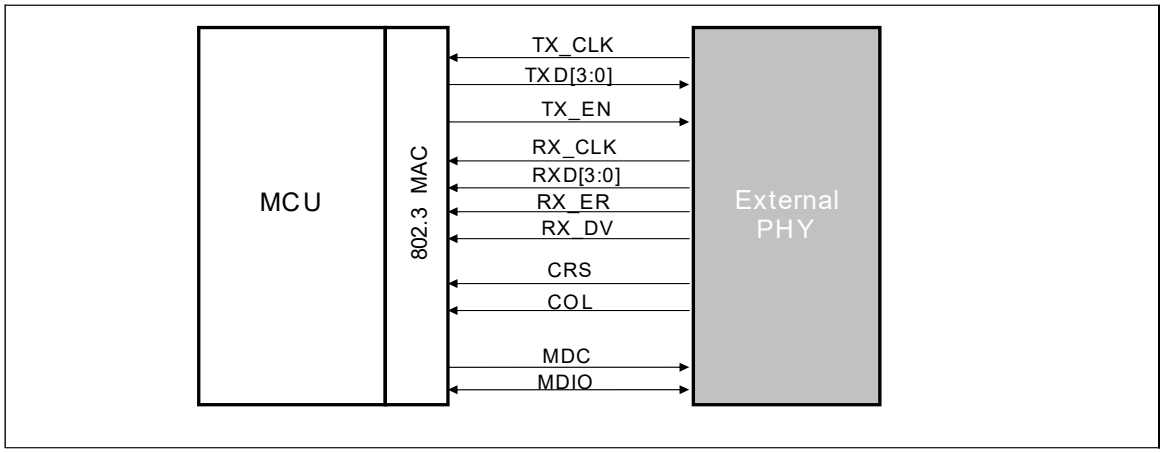



图 25-5 介质独立接口信号

- MII_TX_CLK：连续时钟信号。该信号提供进行 TX 数据传输时的参考时序。标称频率为：速率为 10 Mbit/s 时为 2.5 MHz；速率为 100 Mbit/s 时为 25 MHz。
- MII_RX_CLK：连续时钟信号。该信号提供进行 RX 数据传输时的参考时序。标称频率为：速率为 10 Mbit/s 时为 2.5 MHz；速率为 100 Mbit/s 时为 25 MHz。
- MII_TX_EN：发送使能信号。该信号表示 MAC 当前正针对 MII 发送半字节。该信号必须与报头的前半字节进行同步 (MII_TX_CLK)，并在所有待发送的半字节均发送到 MII 时必须保持同步。
- MII_TXD[3:0]：数据发送信号。该信号是 4 个一组的数据信号，由 MAC 子层同步驱动，在 MII_TX_EN 信号有效时才为有效信号（有效数据）。MII_TXD[0] 为最低有效位，MII_TXD[3] 为最高有效位。禁止 MII_TX_EN 时，发送数据不会对 PHY 产生任何影响。
- MII_CRS：载波侦听信号。当发送或接收介质处于非空闲状态时，由 PHY 使能该信号。发送和接收介质均处于空闲状态时，由 PHY 禁止该信号。PHY 必须确保 MII_CS 信号在冲突条件下保持有效状态。该信号无需与 TX 和 RX 时钟保持同步。在全双工模式下，该信号没意义。
- MII_COL：冲突检测信号。检测到介质上存在冲突后，PHY 必须立即使能冲突检测信号，并且只要存在冲突条件，冲突检测信号必须保持有效状态。该信号无需与 TX 和 RX 时钟保持同步。在全双工模式下，该信号没意义。
- MII_RXD[3:0]：数据接收信号。该信号是 4 个一组的数据信号，由 PHY 同步驱动，在 MII_RX_DV 信号有效时才为有效信号（有效数据）。MII_RXD[0] 为最低有效位，MII_RXD[3] 为最高有效位。当 MII_RX_DV 禁止、MII_RX_ER 使能时，特定的 MII_RXD[3:0] 值用于传输来自 PHY 的特定信息（请参见下表）。
- MII_RX_DV：接收数据有效信号。该信号表示 PHY 当前正针对 MII 接收已恢复并解码的半字节。该信号必须与恢复帧的头半字节进行同步 (MII_RX_CLK)，并且一直保持同步到恢复帧的最后半字节。该信号必须在最后半字节随后的第一个时钟周期之前禁止。为了正确地接收帧， MII_RX_DV 信号必须在时间范围上涵盖要接收的帧，其开始时间不得迟于 SFD 字段出现的时间。
- MII_RX_ER：接收错误信号。该信号必须保持一个或多个周期 (MII_RX_CLK)，从而向 MAC 子层指示在帧的某处检测到错误。该错误条件必须通过 MII_RX_DV 验证，如下表所示。

表 25-4 TX 接口信号编码

| II_TX_EN | MII_TXD[3:0] | 说明 |
|---|---|---|
| 0 | 0000 到 1111 | 正常帧间 |

| II_TX_EN | MII_TXD[3:0] | 说明 |
|---|---|---|
| 1 | 0000 到 1111 | 正常数据发送 |

表 25-5 RX 接口信号编码

| MII_RX_DV | MII_RX_ER | MII_RXD[3:0] | 说明 |
|---|---|---|---|
| 0 | 0 | 0000 到 1111 | 正常帧间 |
| 0 | 1 | 0 | 正常帧间 |
| 0 | 1 | 0001 到 1101 | 保留 |
| 0 | 1 | 1110 | 错误载波检测 |
| 0 | 1 | 1111 | 保留 |
| 1 | 0 | 0000 到 1111 | 正常数据接收 |
| 1 | 1 | 0000 到 1111 | 数据接收出现错误 |

**MII 时钟源**

要生成 TX_CLK 和 RX_CLK 时钟信号，必须向外部 PHY 提供 25MHz 时钟，或使用嵌入式 PLL 生成 25 MHz 频率。

### 25.4.3. 精简介质独立接口：RMII

精简介质独立接口 (RMII) 规范降低了 10/100 Mbit/s 下微控制器以太网外设与外部 PHY 间的引脚数。根据 IEEE 802.3u 标准，MII 包括 16 个数据和控制信号的引脚。RMII 规范将引脚数减少为 7 个（引脚数减少 62.5%）。

RMII 接口是 MAC 和 PHY 之间的实例化对象。这有助于将 MAC 的 MII 转换为 RMII。RMII 具有以下特性：

- 支持 10-Mbit/s 和 100-Mbit/s 的运行速率；
- 参考时钟必须是 50 MHz；
- 相同的参考时钟必须从外部提供给 MAC 和外部以太网 PHY；
- 它提供了独立的 2 位宽（双位）的发送和接收数据路径。

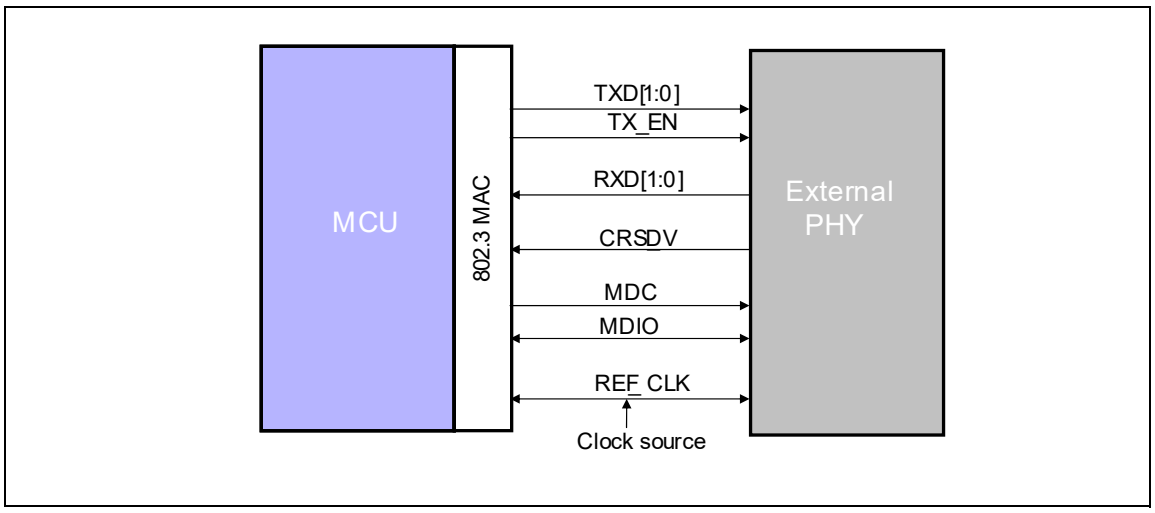



图 25-6 精简介质独立接口信号

**RMII 时钟源**

使用外部 50 MHz 时钟驱动 PHY 或使用嵌入式 PLL 生成 50 MHz 频率信号来驱动 PHY。

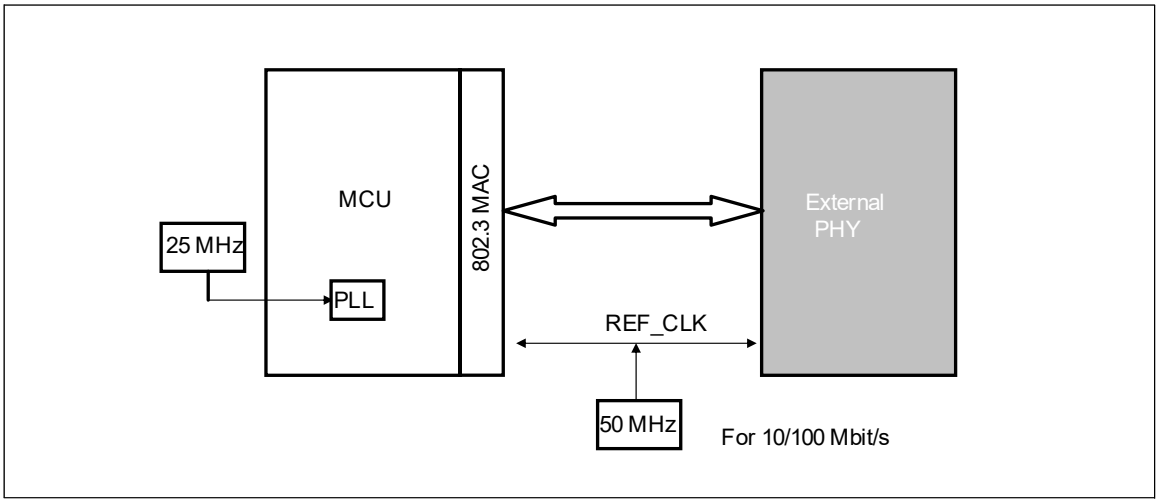



图 25-7 RMII 时钟源

## 25.4.4. MII/RMII 选择

使用 SYSCFG_PMC 寄存器中的配置位 23 MII_RMII_SEL 选择 MII 或 RMII 模式。以太网控制器处于复位模式或使能时钟前，应用程序必须设置 MII/RMII 模式。

**MII/RMII 内部时钟方案**

支持 MII 和 RMII 以及 10 和 100 Mbit/s 运行所需的时钟方案如下图所示。

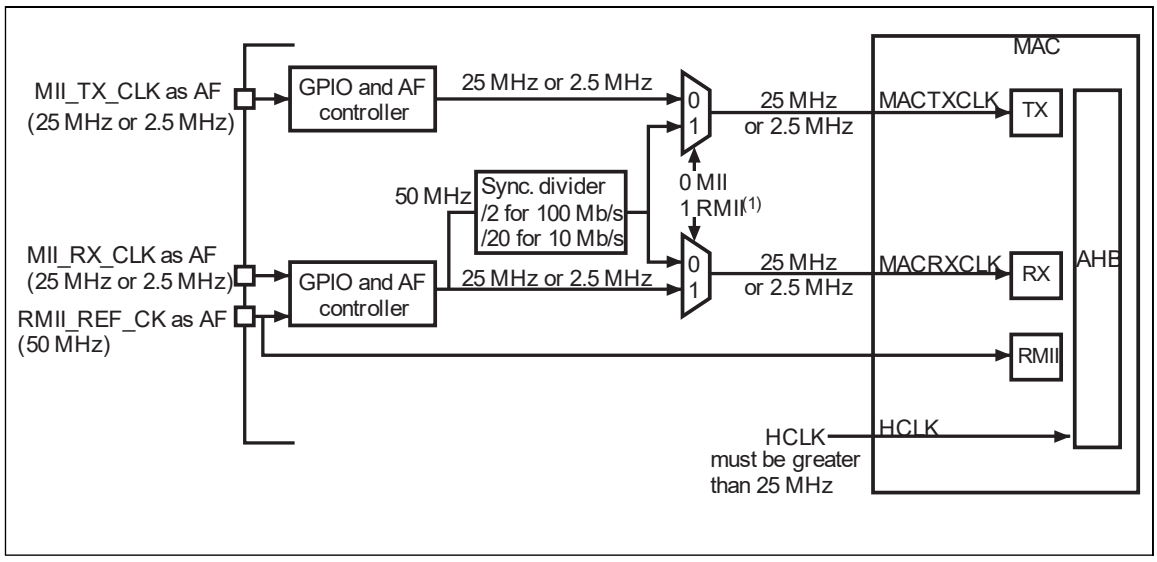



图 25-8 时钟方案

*1. 通过 SYSCFG_PMC 寄存器中的位 23 MII_RMII_SEL 控制 MII/RMII 选择。*

要节省引脚，需在同一个 GPIO 引脚上复用 RMII_REF_CK 和 MII_RX_CLK 这两个输入时钟信号。

## 25.5. 以太网功能说明：MAC 802.3

适用于局域网 (LAN) 的 IEEE 802.3 国际标准将 CSMA/CD（带有冲突检测的载波侦听多路访问）用作访问方法。

以太网外设包括一个带介质独立接口 (MII) 的 MAC 802.3（介质访问控制）控制器和一个专用 DMA 控制器。

MAC 模块对以下系列的系统使用 LAN CSMA/CD 子层：数据速率为 10 Mbit/s 和 100 Mbit/s 的基带系统和宽带系统。支持半双工和全双工工作模式。冲突检测访问方法仅适用于半双工工作模式。支持 MAC 控制帧子层。

MAC 子层执行以下与数据链路控制步骤相关的功能：

- 数据封装（发送和接收）：
  - 组帧（帧边界定界、帧同步）；
  - 寻址（处理源地址和目标地址）；
  - 错误检测；
- 介质访问管理：
  - 介质分配（冲突避免）；
  - 竞争解决（冲突处理）。

MAC 子层主要有两个工作模式：

- 半双工模式：站点使用 CSMA/CD 算法争用物理介质。
- 全双工模式：满足以下条件时，无需解决竞争问题（CSMA/CD 算法不是必需的）便可同时发送和接收数据：
  - 物理介质能够支持同步发送和接收；
  - 正好有 2 个站点与 LAN 相连；
  - 两个站均配置为全双工工作模式。

### 25.5.1. MAC 802.3 帧格式

正如 IEEE 802.3-2002 标准规定，MAC 块使用 MAC 子层和可选 MAC 控制子层(10/100 Mbit/s)。

为使用 CSMA/CD MAC 的数据通信系统指定了两个帧格式：

- 基本 MAC 帧格式；
- 标记 MAC 帧格式（扩展了基本 MAC 帧格式）。

下面图介绍了包含以下字段的帧结构（未标记和标记）：

- 报头： 7 字节字段，用于进行同步（ PLS 电路）

  十六进制值： 55-55-55-55-55-55-55

  位模式：01010101 01010101 01010101 01010101 01010101 01010101 01010101（从右到左进行位传输）
- 起始帧定界符 (SFD)：1 字节字段，用于指示帧的开始。十六进制值： D5 位模式：11010101（从右到左进行位传输）
- 目标地址和源地址字段：MAC 地址字段（ 6 字节），指示目标站和源站地址：
  - 每个地址的长度为 48 位。
  - 目标地址字段中的第一个 LSB 位 (I/G) 用于指示单个地址 (I/G = 0) 或组地址(I/G = 1)。一个组

地址可以标识“无”、一个或多个，或所有连接至 LAN 的站。在源地址中，第一位保留并复位为 0。

- 第二位 (U/L) 用于区分局部 (U/L = 1) 或全局 (U/L = 0) 管理地址。对于广播地址，该位同样为 1。
- 每个地址字段的各个字节必须最先发送最低有效位。

地址指定基于以下类型：

- 单个地址：与网络中的特殊站关联的物理地址。
- 组地址。与给定网络中一个或多个站关联的多目标地址。有两种多播地址：
  - 多播组地址：与一组逻辑相关站关联的地址。
  - 广播地址：一个特殊的预定义多播地址（目标地址字段中全为“1”），该地址总是表示给定 LAN 上的所有站。

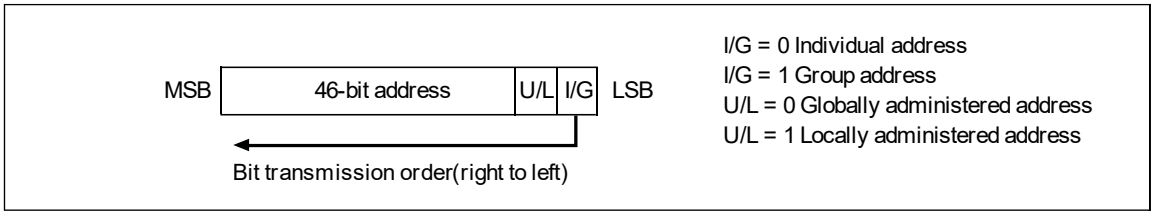



图 25-9 地址字段格式

- QTag 前缀：在源地址字段和 MAC 客户端长度/类型字段中插入的 4 字节字段。该字段是对基本帧（未标记）的扩展，用于获得标记的 MAC 帧。未标记的 MAC 帧不包括该字段。扩展的标记如下：
  - 2 字节常量长度/类型字段值（符合“类型”解析，大于 0x0600），等于 802.1Q 变量协议类型的值（十六进制 0x8100）。该常量字段用于区分标记和未标记的 MAC 帧。
  - 包含变量控制信息的 2 字节字段可以再分为：一个 3 位用户优先级、一个 1 位标准格式指示符 (CFI) 和一个 12 位 VLAN 标识符。标记 MAC 帧的长度通过 QTag 前缀扩展 4 个字节。
- MAC 客户端长度/类型： 2 字节字段，具有不同含义（互斥），具体取决于其值：
  - 如果该值小于或等于 maxValidFrame (0d1500)，则该字段表示 802.3 帧的后续数据字段中所包含的 MAC 客户端数据字节的数量（长度解析）。
  - 如果该值大于或等于 MinTypeValue（十进制 0d1536，0x0600），则该字段表示与以太网帧相关的 MAC 客户端协议的种类（类型解析）。

无论长度/类型字段的解析结果为何，如果数据字段的长度小于协议正确运行所需的最小长度，则将在数据字段之后、 FCS（帧检查序列）字段之前添加一个 PAD 字段。发送和接收长度/类型字段时，高位字节在前。

对于在 maxValidLength 到 minTypeValue 范围内（不包括边界）的长度/类型字段值，未指定 MAC 子层的行为：MAC 子层可能传递、也可能不传递这些值。

- 数据和 PAD 字段：n 字节数据字段。其数据完全透明，这意味着数据字段中可能出现任意顺序的字节值。PAD 的大小（如果存在）由数据字段的大小决定。数据和 PAD 字段的最大和最小长度为：
  - 最大长度 = 1500 字节；
  - 无标记的 MAC 帧的最小长度 = 46 字节；
  - 带标记的 MAC 帧的最小长度 = 42 字节。

当数据字段的长度小于要求的最小长度时，将添加 PAD 字段以匹配最小长度（带标记的帧为 42 字节，无标记的帧为 46 字节）。

- 帧检查序列：包含循环冗余校验 (CRC) 值的 4 字节字段。CRC 计算基于下列字段：源地址、目标地

址、QTag 前缀、长度/类型、LLC 数据和 PAD（即，除报头、SFD 字段以外的所有字段）。生成的多项式如下：

$$\mathbf{G(x) = X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X + 1}$$

帧的 CRC 值按如下方式计算：

- 帧的前 2 位是互补位；
- 帧的 n 个位是 (n − 1) 次多项式 M(x) 的系数。目标地址的第一位对应于 $x^{n-1}$ 项，数据字段的最后一位对应于 $x^0$ 项；
- M(x) 乘以 $x^{32}$ 再除以 G(x)，得到 ≤ 31 次的余数 R(x)；
- 将 R(x) 的系数视为一个 32 位序列；
- 对位序列进行互补，结果为 CRC；
- 32 位的 CRC 值放置在帧检查序列中。首先发送 $x^{32}$ 项，最后发送 $x^0$ 项。

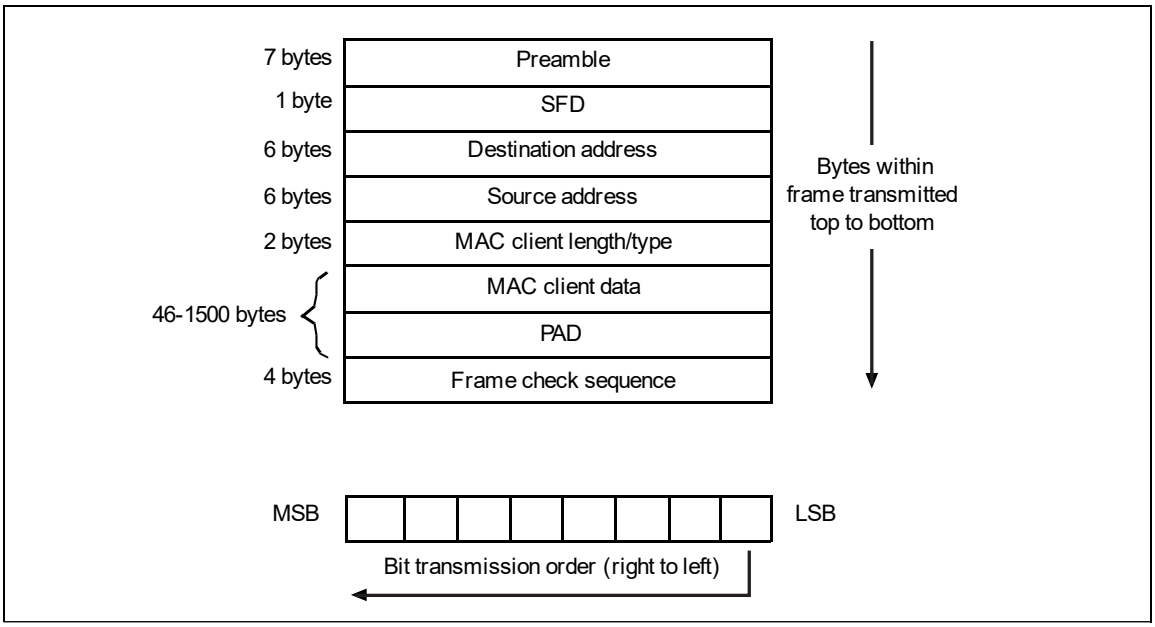



图 25-10 MAC 帧格式

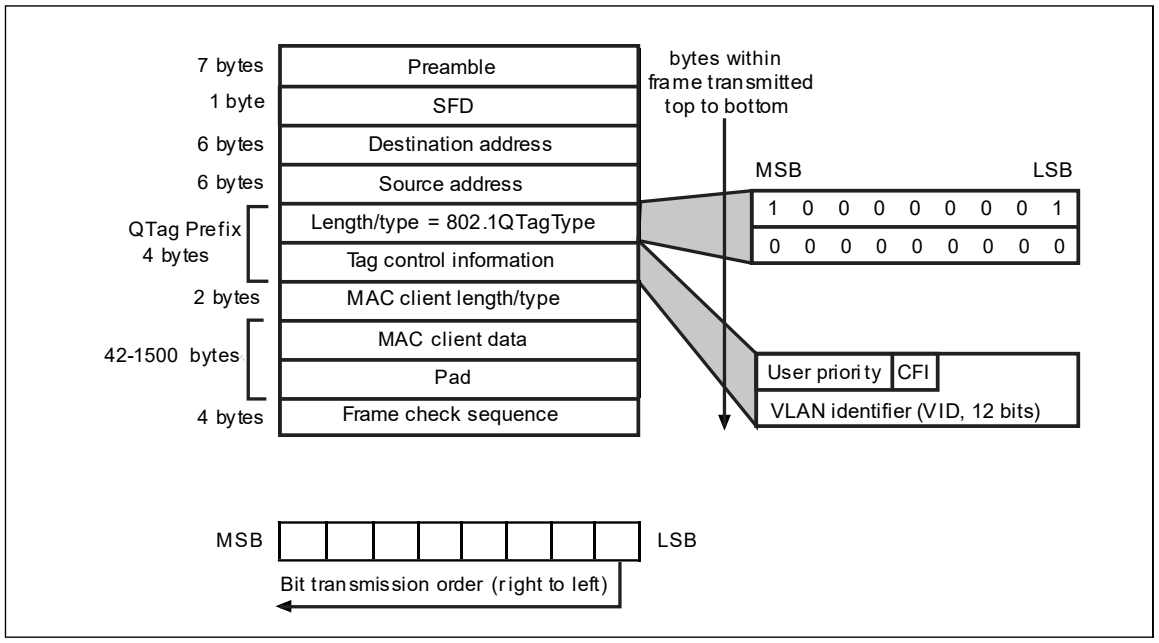



图 25-11 带标记的 MAC 帧格式

发送 MAC 帧的每个字节（除 FSC 字段外）时，低位在前。

通过以下条件之一定义无效 MAC 帧：

- 帧长度与长度/类型字段指定的预期值不一致。如果长度/类型字段包含类型值，则认为帧长度与此字段一致（没有无效帧）；
- 帧长度不是字节的整数倍（额外位）；
- 根据传入帧计算出的 CRC 值与包含的 FCS 不匹配。

### 25.5.2. MAC 帧发送

DMA 控制发送路径的所有事务。从系统存储器读取的以太网帧由 DMA 推入 FIFO。然后将帧弹出并传输到 MAC 内核。帧传输结束时，从 MAC 内核获取发送状态并传回 DMA。发送 FIFO 的深度为 2 KB。FIFO 填充级别将指示给 DMA，以便 DMA 可通过 AHB 接口在所需的系统存储器突发中启动数据获取。来自 AHB 主接口的数据将推入 FIFO。

检测到 SOF 时，MAC 接受数据并开始向 MII 发送。在应用程序启动发送后，向 MII 发送帧数据所需的时间是可变的，具体取决于 IFG 延迟、发送报头/SFD 的时间以及半双工模式的任意回退延迟等延迟因素。EOF 传输到 MAC 内核后，内核将完成正常的发送，然后将发送的状态返回给 DMA。如果在发送过程中发生常规冲突（在半双工模式下），MAC 内核将使发送状态有效，然后接受并丢弃所有后续数据，直至收到下一 SOF。检测到来自 MAC 的重试请求（在状态中）时，应从 SOF 重新发送同一帧。如果发送期间未连续提供数据，MAC 将发出下溢状态。在帧的正常传输期间，如果 MAC 在未获得前一帧的 EOF 的情况下接收到 SOF，则将忽略该 SOF 并将新的帧视为前一帧的延续。

向 MAC 内核弹出数据有两种操作模式：

在阈值模式下，只要 FIFO 中的字节数超过配置的阈值（或在超过阈值前写入帧结束），数据就准备好弹出并转发到 MAC 内核。使用 ETH_DMABMR 的 TTC 位配置阈值。

在存储转发模式下，仅当在 FIFO 中存储完整的帧后，才会向 MAC 内核弹出帧。如果 Tx FIFO 的大小小于要发送的以太网帧，则在 Tx FIFO 接近填满时向 MAC 内核弹出帧。

应用可通过将 FTF 位（ETH_DMAOMR 寄存器 [20]）置 1 来清空发送 FIFO 的所有内容。此位自行清零，并将 FIFO 指针初始化为默认状态。如果向 MAC 内核传输帧时将 FTF 位置 1，则传输将停止，因为

此时 FIFO 被视为空。因此， MAC 发送器将出现下溢事件并且相应的状态字将转发给 DMA。

**自动 CRC 和 pad 生成**

当从应用程序接收的字节数少于 60（ DA+SA+LT+数据）时，会向发送帧附加零，使数据长度正好为 46 字节，以满足 IEEE 802.3 的最小数据字段要求。可将 MAC 编程为不附加任何填充值。计算帧检查序列 (FCS) 字段的循环冗余校验 (CRC) 并将其附加到正在发送的数据。如果将 MAC 编程为不将 CRC 值附加到以太网帧的末尾，则不发送计算出的 CRC。此规则有一种例外情况，即，将 MAC 编程为向小于 60 字节（ DA+SA+LT+数据）的帧附加填充时， CRC 将附加在填充帧的末尾。

CRC 发生器计算以太网帧的 FCS 字段的 32 位 CRC。编码由下面的多项式定义。

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

**发送协议**

MAC 控制以太网帧的发送操作。它执行下列功能以满足 IEEE 802.3/802.3z 规范。包括:

- 生成报头和 SFD;
- 在半双工模式下生成阻塞信号;
- 控制 Jabber 超时;
- 控制半双工模式下的流量（背压）;
- 生成发送帧状态;
- 包含符合 IEEE 1588 的时间戳快照逻辑。

当请求发送新的帧时，MAC 将发送报头和 SFD，紧接着发送数据。报头定义为 0b10101010 模式的 7 个字节， SFD 定义为 0b10101011 模式的 1 个字节。冲突窗口定义为 1 个时隙（对于 10/100 Mb/s 以太网，为 512 个位时间）。阻塞信号生成仅适用于半双工模式，不适用于全双工模式。

在 MII 模式下，如果在开始传输帧到 CRC 字段结束之间的任何时间发生冲突，MAC 将在 MII 上发送 0x5555 5555 的 32 位阻塞信号，通知所有其它站已发生冲突。如果在报头发送阶段发生冲突，MAC 将完成报头和 SFD 的发送，然后发送阻塞信号。

系统使用一个 jabber 定时器，用于在传输的字节超过 2048 字节（默认值）时切断以太网帧的发送。在半双工模式下， MAC 使用延迟机制进行流量控制（背压）。当应用程序请求停止接收帧时，如果已使能发送流量控制，则只要 MAC 检测到接收帧，就会发送一个 32 字节的 JAM 信号。这会导致冲突并使远程站回退。应用程序通过将 ETH_MACFCR 寄存器中的 BPA 位（位 0）置 1 来请求流量控制。如果应用程序请求发送帧，则即使激活背压功能，也将按调度计划发送。请注意，如果背压功能长时间保持激活（发生的连续冲突事件超过 16 个），则远程站将由于冲突过多而中止发送。如果针对发送帧使能 IEEE 1588 时间戳功能，则将 SFD 置于发送 MII 总线时，此模块会获取系统时间的快照。

**发送调度程序**

MAC 负责调度 MII 上的帧发送。它可保持两个发送帧之间的帧间隔，并在半双工模式下遵循截断二进制指数回退算法。在满足 IFG 和回退延迟条件后， MAC 使能发送。它可确保两个发送帧之间的空闲期段，即配置的帧间隔（ETH_MACCR 寄存器中的 IFG 位）。如果要发送的帧在配置的 IFG 时间之前到达，则 MII 会等待来自 MAC 的使能信号，然后再开始发送。只要 MII 的载波信号进入无效状态，MAC 就会启动其 IFG 计数器。在编程的 IFG 值的末尾，MAC 将以全双工模式使能发送。在半双工模式下，当 IFG 配置为 96 个位时间时，MAC 将遵循 IEEE 802.3 规范第 4.2.3.2.1 节中指定的顺从规则。如果在 IFG 间隔的前三分之二时间内（对于所有 IFG 值都为 64 位时间）检测到载波，MAC 将复位其 IFG 计数器。如果在 IFG 间隔的后三分之一时间内检测到载波，MAC 将继续执行 IFG 计数并在 IFG 间隔结束后使能发送器。MAC 在半双工模式下工作时，实施截断二进制指数回退算法。

**发送流量控制**

在全双工模式下，当发送流量控制使能位（ETH_MACFCR 中的 TFE 位）置 1 时， MAC 将生成暂停帧并根据需要发送暂停帧。暂停帧与计算出的 CRC 附加在一起并发送。可以通过两种方式启动暂停帧的生

成。

当应用程序将 ETH_MACFCR 寄存器中的 FCB 位置 1 或接收 FIFO 已满（数据包缓冲区）时，将发送暂停帧。

- 如果应用程序已通过将 ETH_MACFCR 寄存器中的 FCB 位置 1 来请求流量控制，则 MAC 将生成并发送单个暂停帧。生成的帧中的暂停时间值为 ETH_MACFCR 中编程的暂停时间值。要在先前发送的暂停帧中指定的时间之前延长或结束暂停时间，应用程序必须以适当的值编程暂停时间值（ETH_MACFCR 寄存器中的 PT），然后请求另一次暂停帧发送。
- 如果接收 FIFO 填满时应用程序已请求流量控制，则 MAC 将生成并发送暂停帧。生成的帧中的暂停时间值为 ETH_MACFCR 中编程的暂停时间值。如果在此暂停时间结束前，接收 FIFO 在可配置的时隙数（ETH_MACFCR 中的 PLT 位）期间保持填满状态，将发送第二个暂停帧。只要接收 FIFO 保持填满状态，该过程将一直重复下去。如果在采样时间之前不再满足此条件， MAC 将发送一个暂停时间为零的暂停帧，向远程端表明接收缓冲区已准备好接收新数据帧。

**单数据包发送操作**

发送操作的常规事件序列如下：

- 如果系统有要传输的数据，则 DMA 控制器将通过 AHB 主接口从存储器中获取这些数据并将其转发给 FIFO。它将继续接收数据直到传输帧结束。
- 当超过阈值或 FIFO 接收到完整的数据包时，直到将完整的数据包传输到 MAC 为止。DMA 继续从 FIFO 传输数据，数据会被弹出并被传输到 MAC 内核。帧传输完成后，来自 MAC 的状态将通知 DMA 控制器。

**发送操作 - 缓冲区中有两个数据包**

- 由于 DMA 必须先更新描述符状态才能将其释放给主机，因此一个发送 FIFO 内最多只能有两个帧。仅当 OSF（对第二个帧起作用）位置 1 时， DMA 才会获取第二个帧并将其置于 FIFO 中。如果此位未置 1，则仅当 MAC 完全处理该帧且 DMA 已释放描述符后，DMA 才会从存储器获取下一帧。
- 如果 OSF 已置 1，则 DMA 将第一个帧传输到 FIFO 后将立即开始获取第二个帧。不会等待状态更新。同时，在发送第一个帧时，FIFO 接收到第二个帧。第一帧的数据传输完成后，来自 MAC 的状态会通知 DMA。如果 DMA 已向 FIFO 发送第二个数据包，则第二次发送必须等待第一个数据包的状态，才能继续下一帧。

**冲突期间的重新发送**

在半双工模式下，向 MAC 传输帧时，可能在 MAC 线接口上发生冲突事件。MAC 甚至会在接收到帧结束之前就给出状态来指示重试。然后将使能重新发送并再次将帧从 FIFO 中弹出。当超过 96 个字节弹向 MAC 内核后，FIFO 控制器将释放该空间，使 DMA 可推入更多数据。这意味着超过阈值后或 MAC 内核指示延迟冲突事件时，无法重新发送。

**发送 FIFO 刷新操作**

MAC 为软件提供了一个控制权，使其可通过操作模式寄存器中的位 20 来清空发送 FIFO。清空操作是立即操作，即使 Tx FIFO 正在向 MAC 内核传输帧，Tx FIFO 和相应的指针也会清零到初始状态。这将导致 MAC 发送器中生成下溢事件，并且帧发送将中止。此类帧的状态将同时标记下溢和帧清空事件（ TDES0 位 13 和 1）。清空操作期间，没有数据从应用程序 (DMA) 传输到 FIFO。根据清空的帧数（包括局部帧），将相应数量的传输发送状态字传输到应用程序。完全清空的帧的帧清空状态位 (TDES0 13) 将置 1。当应用程序 (DMA) 接受所有已清空的帧的状态字后，清空操作完成。发送 FIFO 清空控制寄存器位随后将清零。此时，将接受来自应用程序 (DMA) 的新帧。清空操作完成后，将丢弃所有为发送提交的数据，除非数据以 SOF 标记开头。

**发送状态字**

在向 MAC 内核传输以太网帧结束时以及内核完成帧的发送后，发送状态将提供给应用程序。发送状态的详细说明与 TDES0 中的位 [23:0] 相同。如果使能 IEEE 1588 时间戳功能，将返回特定帧的 64 位时间戳以及发送状态。

**发送校验和减荷**

通信协议（例如 TCP 和 UDP）将实施校验和字段，这有助于确定通过网络发送的数据的完整性。由于以太网最广泛的用途是通过 IP 数据报封装 TCP 和 UDP，因此以太网控制器具有发送校验和减荷功能，该功能支持校验和计算、发送路径中的校验和插入以及接收路径中的错误检测。本节将介绍发送帧的校验和减荷功能的操作。

*注意：通过完整的帧来计算 TCP、UDP 或 ICMP 的校验和，然后将其插入相应的报头字段。由于此要求，仅当发送 FIFO 配置为存储转发模式（即，ETH_ETH_DMAOMR 寄存器中的 TSF 位置 1）时，才使能此功能。如果内核配置为阈值（直通）模式，则将绕过发送校验和减荷。*

*必须确保发送 FIFO 足够深，使其能存储一个完整的帧，以便将该帧传输到 MAC 内核发送器。如果 FIFO 的深度小于输入以太网帧的太小，则将绕过有效负载 (TCP/UDP/ICMP) 校验和插入功能，并且仅修改帧的 IPv4 报头校验和，即使在存储转发模式下也是如此。*

发送校验和减荷支持两种类型的校验和计算和插入。可通过将 CIC 位（TDES1 中的位 28:27）置 1 来控制每个帧的校验和。

有关 IPv4、TCP、UDP、ICMP、IPv6 和 ICMPv6 数据包头规范的信息，请分别参见 IETF 规范 RFC 791、RFC 793、RFC 768、RFC 792、RFC 2460 和 RFC 4443。

- IP 报头校验和

  在 IPv4 数据报中，报头字段的完整性由 16 位头校验和字段（ IPv4 数据报的第十一个字节和第十二个字节）指示。当以太网帧的类型字段值为 0x0800 且 IP 数据报的版本字段值为 0x4 时，校验和减荷将检测到 IPv4 数据报。计算期间，将忽略输入帧的校验和字段并将其替换为计算出的值。 IPv6 报头没有校验和字段；因此，校验和减荷不修改 IPv6 报头字段。此 IP 报头校验和计算的结果由发送状态中的 IP 报头错误状态位（位 16）指示。只要以太网类型字段的值和 IP 报头版本字段的值不一致，或当以太网帧没有足够的数据（如 IP 报头长度字段所指示）时，此状态位将置 1。换言之，在以下情况下发生 IP 报头错误时，此位将置 1：

- 对于 IPv4 数据报：
  - 接收的以太网类型为 0x0800，但 IP 报头的版本字段不等于 0x4；
  - IPv4 报头长度字段指示小于 0x5（ 20 字节）的值；
  - 总的帧长度小于 IPv4 报头长度字段给定的值。
- 对于 IPv6 数据报：
  - 以太网类型为 0x86DD，但 IP 报头版本字段不等于 0x6；
  - 帧在 IPv6 报头（ 40 字节）之前结束，或已完全接收到扩展报头（如扩展报头中相应的报头长度字段中给定）。如果以太网类型字段指示 IPv4 有效负载，即使校验和减荷检测到 IP 报头错误，也会插入 IPv4 报头校验和。
  - TCP/UDP/ICMP 校验和

    TCP/UDP/ICMP 校验和对 IPv4 或 IPv6 报头（包括扩展报头）进行处理，并确定封装的有效负载是 TCP、UDP 还是 ICMP。
- 请注意：
  - 对于非 TCP、非 UDP 或非 ICMP/ICMPv6 的有效负载，将绕过此校验和且不会对帧进行修改。
  - 将绕过分段的 IP 帧（ IPv4 或 IPv6）、带安全功能（例如，验证报头或封装的安全有效负载）的 IP 帧和带路由报头的 IPv6 帧，并且校验和不对这些帧进行处理。

计算 TCP、 UDP 或 ICMP 有效负载的校验和，然后将其插入报头中相应的字段。它可工作在以下两种模式：

- 在第一种模式下， TCP、UDP 或 ICMPv6 伪报头并未包含在校验和计算中，并假定其存在于输入帧的校验和字段中。校验和字段包含在校验和计算中，然后替换为最终计算出的校验和。
- 在第二种模式下，将忽略校验和字段，TCP、UDP 或 ICMPv6 伪报头数据包含在校验和计算中，并使用最终计算出的值覆盖校验和字段。

请注意：对于 ICMP-over-IPv4 数据包，由于没有为其定义伪报头，因此在这两种模式下 ICMP 数据包中的校验和字段都必须始终为 0x0000。如果不等于 0x0000，可能向数据包插入不正确的校验和。

此操作的结果由发送状态向量中的有效负载校验和错误状态位（位 12）指示。当检测到下列情况之一时，有效负载校验和错误状态位置 1：

- 在存储转发模式下，帧已转发到 MAC 发送器，但帧结束未写入 FIFO；
- 在接收到 IP 报头中的有效负载长度字段指示的字节数前，数据包已结束。

当数据包的长度大于指示的有效负载长度时，将字节作为填充字节忽略，并且不报告错误。检测到第一种类型的错误时，不修改 TCP、UDP 或 ICMP 报头。对于第二种错误类型，计算的校验和仍将插入相应的报头字段。

**MII/RMII 发送位序**

来自 MII 的每个半字节都在 RMII 上发送，一次发送双位，双位的发送顺序如下图所示。首先发送位序较低的位（D1 和 D0），再发送位序较高的位（D2 和 D3）。

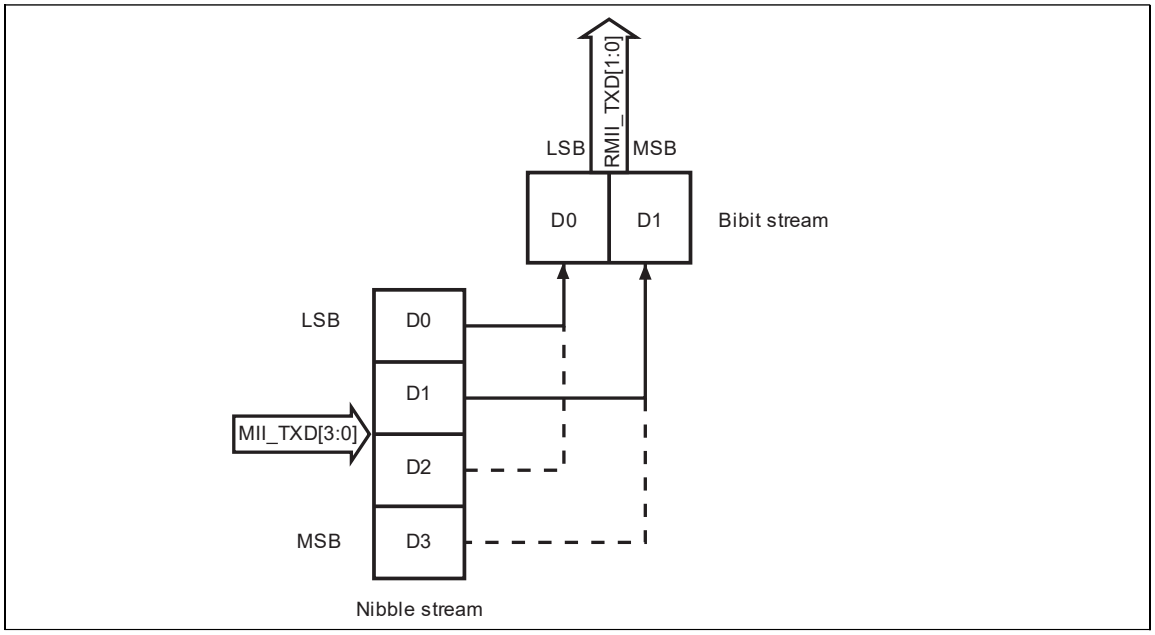



图 25-12 发送位序

**MII/RMII 发送时序图**

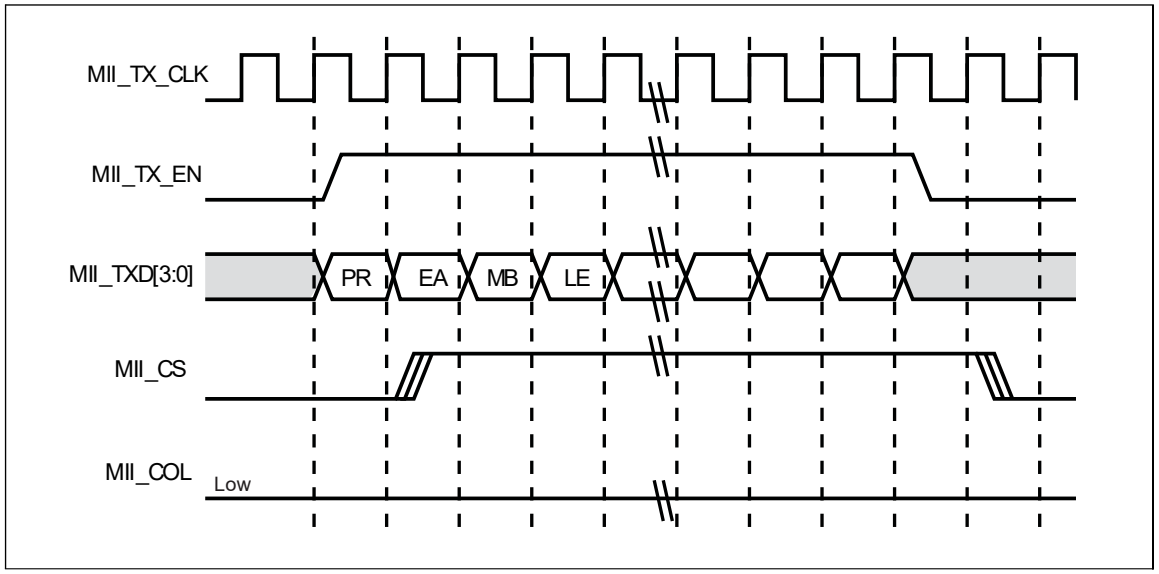



图 25-13 无冲突发送

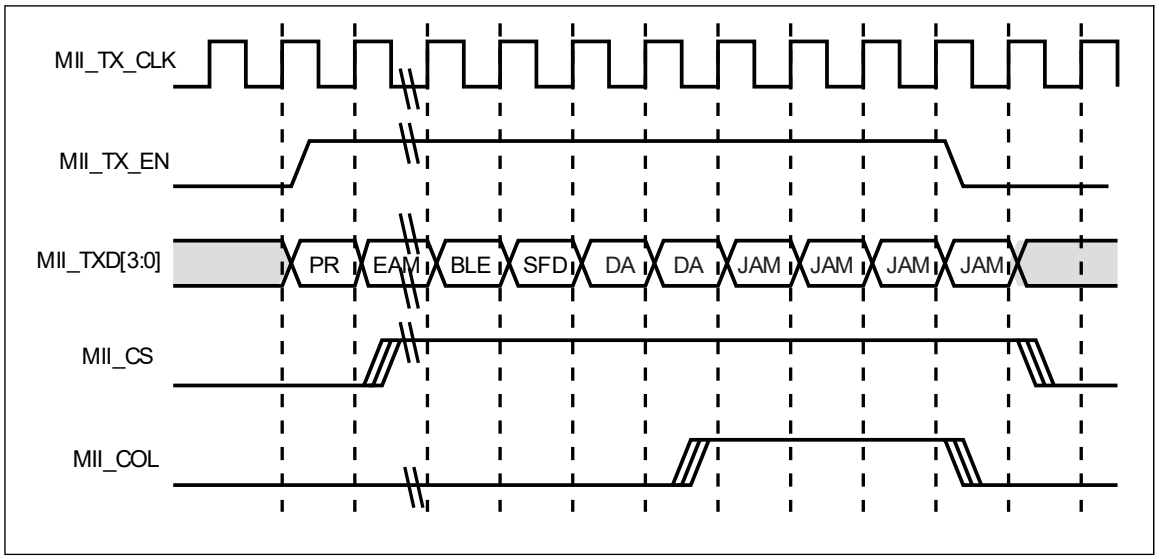



图 25-14 有冲突发送

下图显示了 MII 和 RMII 中的帧发送。

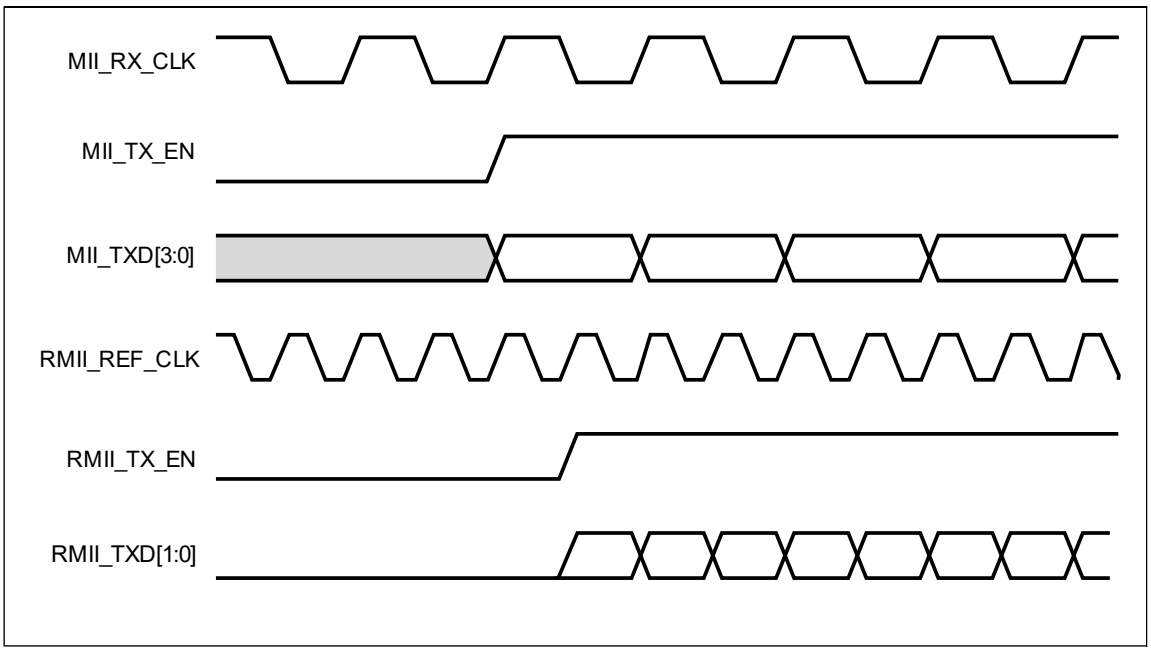



图 25-15 MII 和 RMII 模式下的帧发送

### 25.5.3. MAC 帧接收

MAC 接收的帧将推入 Rx FIFO。此 FIFO 的状态（填充级别）一旦超过配置的接收阈值(ETH_DMAOMR 寄存器中的 RTC)，就会将其指示给 DMA，这样 DMA 可向 AHB 接口发起预配置的突发传输。

在默认直通模式下，当 FIFO 接收到 64 个字节（使用 ETH_DMAOMR 寄存器中的 RTC 位配置）或完整的数据包时，数据将弹出，其可用性将通知给 DMA。 DMA 向 AHB 接口发起传输后，数据传输将从 FIFO 持续进行，直到传输完整个数据包。完成 EOF 帧的传输后，状态字将弹出并发送到 DMA 控制器。

在 Rx FIFO 存储转发模式（通过 ETH_DMAOMR 寄存器中的 RSF 位配置）下，仅在帧完全写入接收 FIFO 后才可读出帧。在此模式下，将丢弃所有错误帧（如果内核配置为执行此操作），这样只会读出有效帧并将其转发到应用程序。在直通模式下，某些错误帧不会被丢弃，因为在帧结束时接收到错误状态，而此时已从 FIFO 读出帧开始。

当 MAC 在 MII 上检测到 SFD 时，将启动接收操作。MAC 内核将去除报头和 SFD，然后再继续处理帧。检查报头字段以进行过滤，FCS 字段用于验证帧的 CRC。如果帧未通过地址滤波器，则在内核中丢弃该帧。

**接收协议**

去除接收的帧的报头和 SFD。检测到 SFD 后，MAC 开始向接收 FIFO 发送以太网帧数据，从 SFD 后面的第一个字节（目标地址）开始发送。如果使能 IEEE 1588 时间戳功能，则在 MII 上检测到任何帧的 SFD 时，都将获取系统时间的快照。除非 MAC 滤出并丢弃帧，否则此时间戳将传递给应用程序。

如果接收的帧长度/ 类型字段小于 0x600 并且为 MAC 编程了自动去除 CRC/pad 选项，则 MAC 将向接收 FIFO 发送帧数据（数据量不超过长度/类型字段中指定的数量），然后开始丢弃字节（包括 FCS 字段）。如果长度/ 类型字段大于或等于 0x600，则不管编程的自动 CRC 去除选项的值如何，MAC 都会向 Rx FIFO 发送所有接收到的以太网帧数据。默认情况下，使能 MAC 看门狗定时器，即，超过 2048 个字节（ DA + SA + LT + 数据 + pad + FCS）的帧会被切断。可通过对 MAC 配置寄存器中的看门狗禁止 (WD) 位编程来禁止此功能。但是，即使禁止看门狗定时器，仍将切断大于 16 KB 的帧并给出看门狗超时状态。

**接收 CRC：自动 CRC 和 pad 去除**

MAC 将检查接收帧中的任何 CRC 错误。它将计算接收的帧（包括目标地址字段到 FCS 字段）的 32 位 CRC。编码由下面的多项式定义。

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

不管是否自动去除 pad/CRC， MAC 都将接收整个帧来计算所接收帧的 CRC 校验。

**接收校验和减荷**

对接收的以太网帧中的 IPv4 和 IPv6 帧进行检测和处理，以确保数据完整性。可通过将 ETH_MACCR 寄存器中的 IPCO 位置 1 来使能接收校验和减荷。MAC 通过检查所接收的以太网帧的类型字段中是否存在值 0x0800 或 0x86DD，来识别 IPv4 或 IPv6 帧。此识别方法也适用于带 VLAN 标记的帧。接收校验和减荷将计算 IPv4 报头校验和，并检查其是否与接收的 IPv4 报头校验和匹配。如果指示的有效负载类型（以太网类型字段）与 IP 报头版本之间有任何不匹配，或接收的帧的字节数小于 IPv4 报头的长度字段中指示的数量（IPv4 或 IPv6 报头中的可用字节数少于 20），IP 报头错误位将置 1。接收校验和减荷还可以识别接收的 IP 数据报（IPv4 或 IPv6）中的 TCP、UDP 或 ICMP 有效负载，并正确计算此类有效负载的校验和，如 TCP、 UDP 或 ICMP 规范中所定义。它包括用于校验和计算的 TCP/UDP/ICMPv6 伪报头字节，并检查接收的校验和字段是否与计算的值匹配。此操作的结果由接收状态字中的有效负载校验和错误位给出。如果 TCP、UDP 或 ICMP 有效负载的长度与 IP 报头中给出的预期有效负载长度不匹配，此状态位也将置 1。如 TCP/UDP/ICMP 校验和中所述，接收校验和减荷将绕过分段 IP 数据报的有效负载、带安全功能的 IP 数据报、 IPv6 路由报头以及除 TCP、UDP 或 ICMP 以外的有效负载。此信息在接收状态中给出（无论是否绕过校验和），如 RDES0：接收描述符字 0 一节所述。在此配置中，内核不会向接收的以太网帧附加任何有效负载校验和字节。

如 RDES0：接收描述符字 0 中所述，某些寄存器位的含义会发生变化，如下表所示。

表 25-6 帧状态

| 位 18：以太网帧 | 位 27：报头校验和错误 | 位 28：有效负载校验和错误 | 帧状态 |
|---|---|---|---|
| 0 | 0 | 0 | 帧为 IEEE 802.3 帧（长度字段值小于 0x0600）。 |
| 1 | 0 | 0 | IPv4/IPv6 类型帧，在其中未检测到校验和错误。 |
| 1 | 0 | 1 | IPv4/IPv6 类型帧，在其中检测到有效负载校验和错误（参见 PCE 的说明）。 |
| 1 | 1 | 0 | IPv4/IPv6 类型帧，在其中检测到 IP 报头校验和错误（参见 IPCO HCE 的说明）。 |
| 1 | 1 | 1 | IPv4/IPv6 类型帧，在其中检测到 PCE 和 IPCO HCE。 |
| 0 | 0 | 1 | IPv4/IPv6 类型帧，其中不存在 IP HCE，并且由于不支持的有效负载而绕过有效负载检查。 |
| 0 | 1 | 1 | 既不是 IPv4 也不是 IPv6 的类型帧（校验和减荷完全绕过校验和的校验） |
| 0 | 1 | 0 | 保留 |

**接收帧控制器**

如果复位 MAC CSR 帧过滤寄存器中的 RA 位，则 MAC 将根据目标/源地址执行帧过滤（如果应用程序决定不接收任何不良帧，如矮帧、CRC 错误帧等，则仍需要执行另一等级的过滤）。检测到过滤失败时，帧将被丢弃且不会传输到应用程序。当过滤参数动态改变时，如果 (DA-SA) 过滤失败，则剩余的帧将被丢弃并且接收状态字将立即更新（零帧长度位、 CRC 错误位和矮帧错误位将置 1），指示过滤失败。在以太网掉电模式下，所有接收到的帧都将被丢弃且不会转发给应用程序。

**接收流量控制**

MAC 将检测接收暂停帧并暂停帧发送，暂停时间为接收的暂停帧内指定的延迟（仅限全双工模式）。可通

过 ETH_MACFCR 中的 RFCE 位使能或禁止暂停帧检测功能。使能接收流量控制后，将开始监视接收帧的目标地址是否与控制帧的多播地址 (0x0180 C200 0001) 匹配。如果检测到匹配（接收的帧的目标地址与保留的控制帧的目标地址匹配），MAC 将根据 ETH_MACFFR 中的 PCF 位来决定是否将接收的控制帧传输到应用程序。

MAC 还将对接收控制帧的类型、操作码和暂停定时器字段进行解码。如果状态的字节计数指示 64 个字节，并且不存在任何 CRC 错误，则 MAC 发送器将暂停任何数据帧的发送，暂停时间为解码的暂停时间值乘以时隙（对于 10/100 Mb/s 模式，均为 64 字节时间）。同时，如果检测到另一个零暂停时间值的暂停帧，MAC 将复位暂停时间并管理新的暂停请求。

如果接收的控制帧与类型字段 (0x8808)、操作码 (0x00001) 以及字节长度（64 字节）均不匹配，或存在 CRC 错误，则 MAC 不会生成暂停。

如果暂停帧具有多播目标地址，MAC 将根据地址匹配来过滤帧。

对于具有单播目标地址的暂停帧，MAC 将根据 DA 是否与 MAC 地址 0 寄存器的内容匹配以及 ETH_MACFCR 中的 UPDF 位是否置 1（检测具有单播目标地址的暂停帧）来进行过滤。PCF 寄存器位（ETH_MACFFR 中的位 [7:6]）可对控制帧的过滤以及地址过滤进行控制。

**接收操作多帧处理**

由于接收数据后状态立即可用，因此只要 FIFO 未满，就可以向其中存储帧。

**错误处理**

如果在从 MAC 接收 EOF 数据之前 Rx FIFO 已满，则将声明上溢并丢弃整个帧，（ETH_DMAMFBOCR 寄存器）中的上溢计数器将递增。由于上溢，状态将指示这是一个部分帧。如果（使用 ETH_DMAOMR 中的 FEF 和 FUGF 位）使能相应功能， Rx FIFO 可过滤错误帧和过小帧。

如果将接收 FIFO 配置为在存储转发模式下工作，则可过滤并丢弃所有错误帧。

在直通模式下，如果在从 Rx FIFO 读取帧的 SOF 时，该帧的状态和长度可用，则可丢弃整个错误帧。DMA 可通过使能接收帧清空位来清空正在从 FIFO 读取的错误帧。然后到应用(DMA) 的数据传输将停止，其余帧将从内部读取并丢弃。如果可用，则可以启动下一帧传输。

**接收状态字**

以太网帧接收结束时， MAC 向应用 (DMA) 输出接收状态。接收状态的详细说明与 RDES0 中的位 [31:0] 相同，请参见 RDES0：接收描述符字 0。

**帧长度接口**

对于开关应用，应用和 MAC 之间的数据发送和接收以传输完整帧的形式进行。为了向出站端口传输帧，应用层应知道从入站端口接收的帧的长度。在每次帧接收结束时， MAC 内核在状态内提供每个接收的帧的长度。

*注意： 对于由于上溢而写入 Rx FIFO 的部分帧，将指定值为 0 的帧长度。*

**MII/RMII 接收位序**

每个半字节都从接收自 RMII 的双位发送到 MII，半字节发送顺序如下图所示。先接收位序较低的位（D0 和 D1），再接收位序较高的位（D2 和 D3）。

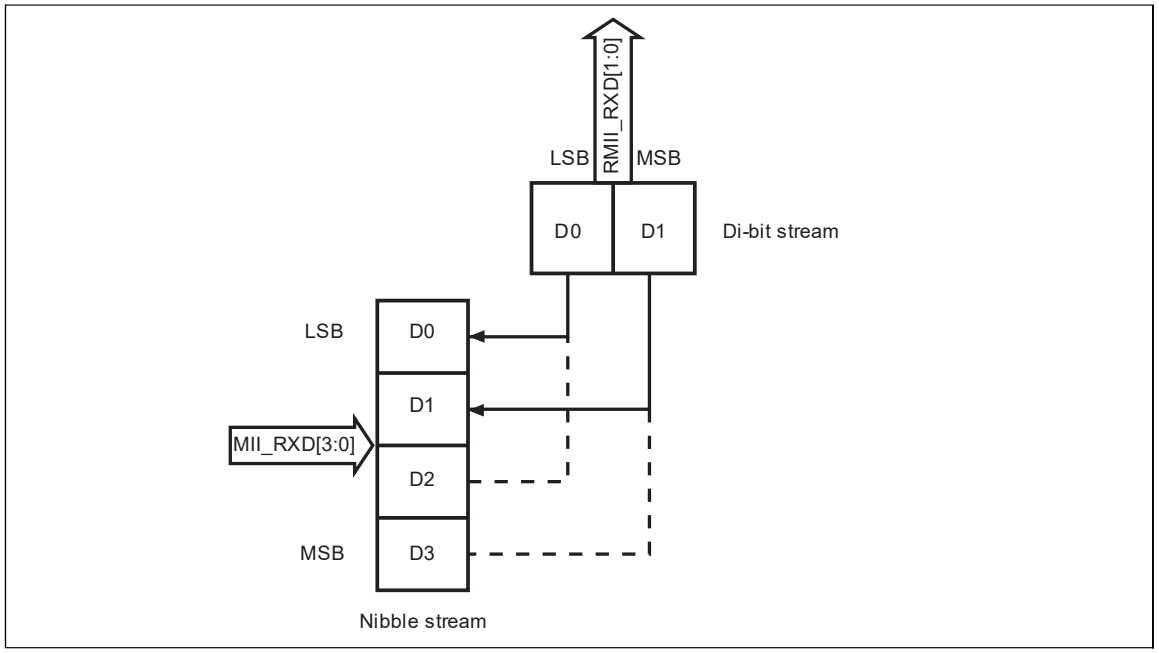



图 25-16 接收位序

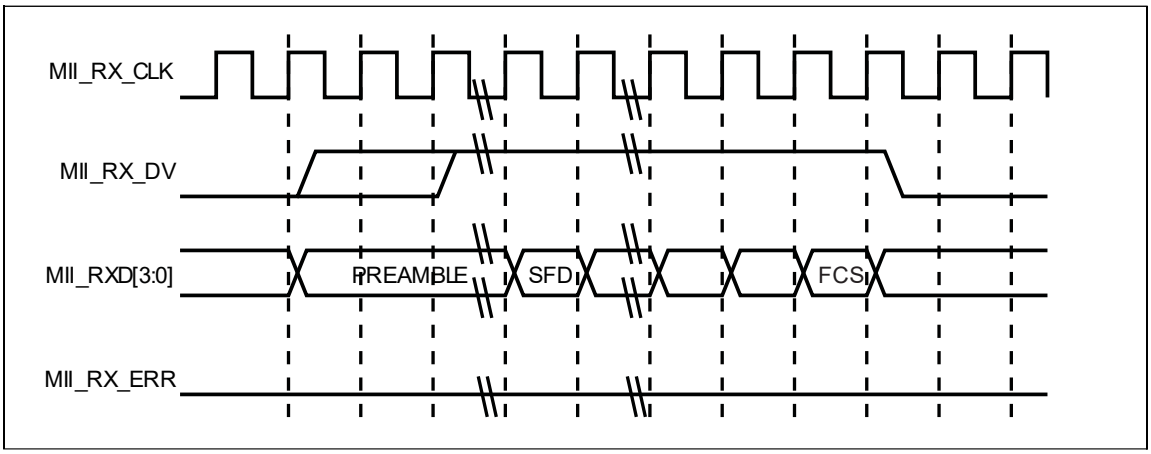



图 25-17 无错误接收

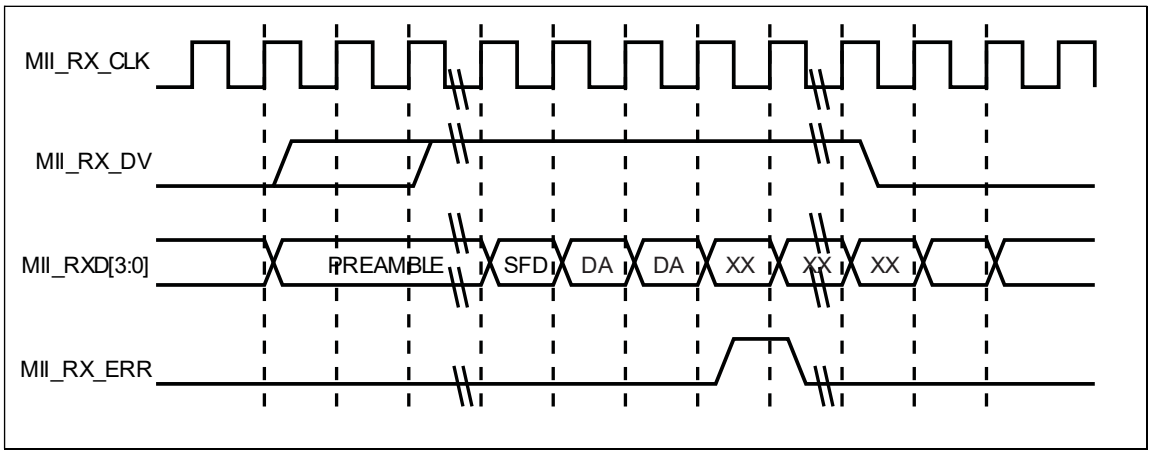



图 25-18 有错误接收

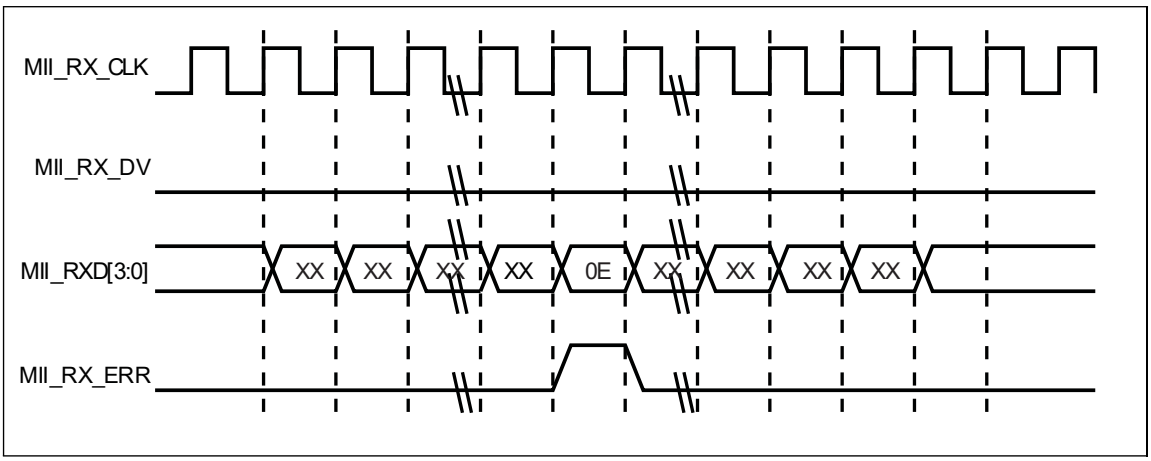



图 25-19 出现假载波指示的接收

## 25.5.4. MAC 中断

MAC 内核可由于各种事件而生成中断。

ETH_MACSR 寄存器描述了可导致 MAC 内核生成中断的事件。可通过将中断屏蔽寄存器中相应的屏蔽位置 1 来阻止各事件生成中断。

中断寄存器位仅指示报告事件的模块。必须读取相应的状态寄存器和其它寄存器才能清除中断。例如，如果中断寄存器的位 3 设置为高电平，则指示在掉电模式下接收到魔术数据包或网络唤醒帧。必须读取 ETH_MACPMTCSR 寄存器才能清除此中断事件。

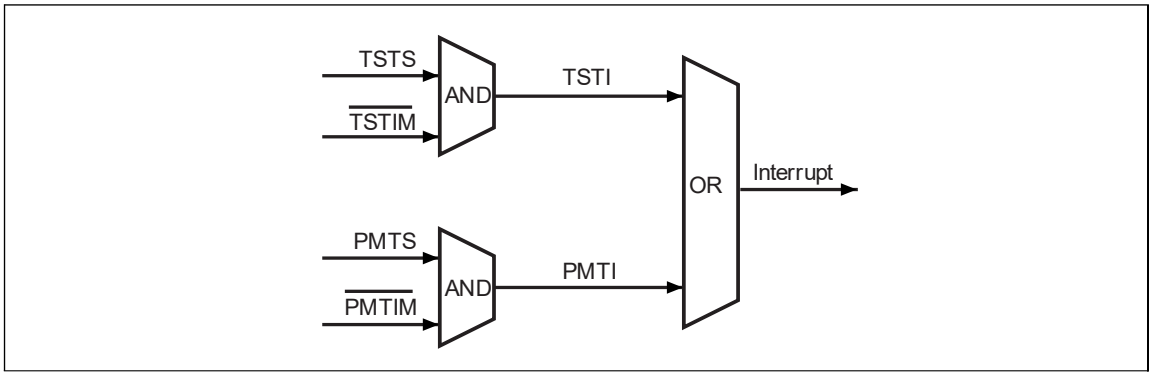



图 25-20 MAC 内核中断屏蔽方案

## 25.5.5. MAC 过滤

### 地址过滤

地址过滤将检查所有接收的帧的目标地址和源地址并相应报告地址过滤状态。地址检查基于应用程序选择的不同参数（帧过滤寄存器）。 还可以识别过滤的帧：多播帧或广播帧。地址过滤使用站的物理 (MAC) 地址和多播散列表进行地址检查。

### 单播目标地址过滤

MAC 支持多达 4 个用于单播完美过滤的 MAC 地址。如果选择完美过滤（复位帧过滤寄存器中的 HU 位），MAC 会将接收的单播地址的所有 48 位与编程的 MAC 地址进行比较来确定是否匹配。默认情况下，始终使能 MacAddr0，其它地址 MacAddr1—MacAddr3 则通过单独的使能位进行选择。将其它地址 (MacAddr1—MacAddr3) 的各个字节与接收的相应 DA 字节进行比较时，可以将寄存器中相应的屏蔽字节

控制位置 1 来屏蔽该字节。这有助于 DA 的组地址过滤。在散列过滤模式（HU 位置 1）下，MAC 将使用 64 位散列表执行对单播地址的不完美过滤。对于散列过滤，MAC 将使用接收的目标地址的 6 个高 CRC 位来索引散列表的内容。值为 000000 时，选取所选寄存器中的位 0；值为 111111 时，选取散列表寄存器中的位 63。如果相应位（由 6 位 CRC 指示）已置 1，将认为单播帧已通过散列过滤，否则认为帧未能通过散列过滤。

*注意： CRC 是使用下列多项式编码的 32 位值（有关详细信息，请参见 MAC 帧接收）：*

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

**多播目标地址过滤**

可通过将帧过滤寄存器中的 PAM 位置 1，将 MAC 编程为通过所有多播帧。如果 PAM 位复位， MAC 将根据帧过滤寄存器中的 HM 位执行对多播地址的过滤。在完美过滤模式下，将多播地址与编程的 MAC 目标地址寄存器 (1–3) 进行比较。组地址过滤也受到支持。在散列过滤模式下，MAC 将使用 64 位散列表执行不完美过滤。对于散列过滤， MAC 将使用接收的多播地址的 6 个高 CRC 位来索引散列表的内容。值为 000000 时，选取所选寄存器中的位 0；值为 111111 时，选取散列表寄存器中的位 63。如果相应位已置 1，将认为多播帧已通过散列过滤，否则认为帧未能通过散列过滤。

*注意： CRC 是使用下列多项式编码的 32 位值（有关详细信息，请参见 MAC 帧接收）：*

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

**散列或完美地址过滤**

可通过将帧过滤寄存器中的 HPF 位置 1 并将相应的 HU 或 HM 位置 1，来将 DA 过滤配置为在其 DA 与散列过滤或完美过滤匹配时允许帧通过。此配置适用于单播帧和多播帧。如果 HPF 位复位，则只有一种过滤方式（散列或完美）应用于接收的帧。

**广播地址过滤**

在默认模式下，MAC 不过滤任何广播帧。但是，如果将帧过滤寄存器中的 BFD 位置 1 来将 MAC 编程为拒绝所有广播帧，则会丢弃任何广播帧。

**单播源地址过滤**

MAC 还可以根据接收的帧的源地址字段来执行完美过滤。默认情况下，MAC 将 SA 字段与 SA 寄存器中编程的值进行比较。可通过将相应寄存器中的位 30 置 1，来将 MAC 地址寄存器 [1:3] 配置为包含 SA 而不是 DA 进行比较。带 SA 的组地址过滤也受到支持。如果帧过滤寄存器中的 SAF 位置 1，则 MAC 将丢弃未通过 SA 过滤的帧。否则，SA 过滤的结果将通过接收状态字中的状态位给出（参见 RDES0：接收描述符字 0）。

SAF 位置 1 时，对 SA 过滤和 DA 过滤的结果进行与运算，以决定是否需要转发帧。这意味着任何一个过滤未通过都将丢弃帧。两个过滤必须都通过，才能将帧转发到应用。

**反向过滤操作**

对于目标地址和源地址过滤，可在最终输出时选择反转过滤匹配结果。这分别由帧过滤寄存器中的 DAIF 和 SAIF 位控制。DAIF 位同时适用于单播和多播 DA 帧。在此模式下，将反转单播/多播目标地址过滤的结果。类似地，当 SAIF 位置 1 时，将反转单播 SA 过滤的结果。下表按照接收帧的类型汇总了目标地址和源地址过滤。

表 25-7 目标地址过滤

| 帧类型 | PM | HPF | HU | DAIF | HM | PAM | DB | DA 过滤操作 |
|---|---|---|---|---|---|---|---|---|
| 广播 | 1 | X | X | X | X | X | X | 通过 |
| | 0 | X | X | X | X | X | 0 | 通过 |
| | 0 | X | X | X | X | X | 1 | 不通过 |
| 单播 | 1 | X | X | X | X | X | X | 通过所有帧 |
| | 0 | X | 0 | 0 | X | X | X | 完美/组过滤匹配时通过 |
| | 0 | X | 0 | 1 | X | X | X | 完美/组过滤匹配时不通过 |

| 帧类型 | PM | HPF | HU | DAIF | HM | PAM | DB | DA 过滤操作 |
|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 1 | 0 | X | X | X | 散列过滤匹配时通过 |
| | 0 | 0 | 1 | 1 | X | X | X | 散列过滤匹配时不通过 |
| | 0 | 1 | 1 | 0 | X | X | X | 散列或完美/组过滤匹配时通过 |
| | 0 | 1 | 1 | 1 | X | X | X | 散列或完美/组过滤匹配时不通过 |
| 多播 | 1 | X | X | X | X | X | X | 通过所有帧 |
| | X | X | X | X | X | 1 | X | 通过所有帧 |
| | 0 | X | X | 0 | 0 | 0 | X | 完美/组过滤匹配时通过，并在 PCF = 0x 时丢弃暂停控制帧 |
| | 0 | 0 | X | 0 | 1 | 0 | X | 散列过滤匹配时通过，并在 PCF = 0x 时丢弃暂停控制帧 |
| | 0 | 1 | X | 0 | 1 | 0 | X | 散列或完美/组过滤匹配时通过，并在 PCF = 0x 时丢弃暂停控制帧 |
| | 0 | X | X | 1 | 0 | 0 | X | 完美/组过滤匹配时不通过，并在 PCF = 0x 时丢弃暂停控制帧 |
| | 0 | 0 | X | 1 | 1 | 0 | X | 散列过滤匹配时不通过，并在 PCF = 0x 时丢弃暂停控制帧 |
| | 0 | 1 | X | 1 | 1 | 0 | X | 散列或完美/ 组过滤匹配时不通过，并在 PCF = 0x 时丢弃暂停控制帧 |

表 25-8 源地址过滤

| 帧类型 | PM | SAIF | SAF | SA 过滤操作 |
|---|---|---|---|---|
| 单播 | 1 | X | X | 通过所有帧 |
| | 0 | 0 | 0 | 完美/组过滤匹配时通过，但不丢弃未通过的帧 |
| | 0 | 1 | 0 | 完美/组过滤匹配时不通过，但不丢弃帧 |
| | 0 | 0 | 1 | 完美/组过滤器匹配时通过并将不通过的帧丢弃 |
| | 0 | 1 | 1 | 完美/组过滤器匹配时不通过并将不通过的帧丢弃 |

### 25.5.6. MAC 回环模式

MAC 支持对发送到其接收器的帧进行回送。默认情况下，禁止 MAC 回送功能，可通过编程 MAC ETH_MACCR 寄存器中的 Loopback 位使能该功能。

### 25.5.7. MAC 管理计数器：MMC

MAC 管理计数器 (MMC) 保持一组寄存器来收集有关已接收帧和已发送帧的统计信息。其中包括一个用于控制各寄存器行为的控制寄存器、两个包含当前中断状态的 32 位寄存器（接收和发送），以及两个包含中断寄存器掩码的 32 位寄存器（接收和发送）。这些寄存器均可从应用程序中访问。每个寄存器均为 32 位宽。

以太网寄存器说明介绍了各个计数器，并列出了各统计计数器的地址。该地址用于对所需发送/接收计数器进行读/写访问。

接收 MMC 计数器针对通过地址过滤的帧进行更新。已丢弃帧的统计信息不会进行更新，除非丢弃的帧为小于 6 字节的矮帧（ DA 字节不能完全接收）。

**好的发送帧与接收帧**

如果帧成功发送，则将发送的帧视为“好帧”。换句话说，如果帧发送过程没有因为以下错误而中止，则认为发送的帧是“好帧”：

+ Jabber 超时

+ 无载波/载波丢失
+ 延迟冲突
+ 帧下溢
+ 过度延迟
+ 过度冲突

如果不存在以下错误，则认为接收帧是“好帧”：

+ CRC 错误
+ 矮帧（短于 64 字节）
+ 对齐错误（仅限 10/100 Mb/s）
+ 长度错误（仅限非类型帧）
+ 超出范围（仅限非类型帧，超过最大大小）
+ MII_RXER 输入错误

最大帧大小取决于帧类型，如下：

+ 无标记帧的最大大小 = 1518
+ VLAN 帧的最大大小 = 1522

### 25.5.8. 电源管理：PMT

本部分介绍 MAC 支持的电源管理 (PMT) 机制。PMT 支持接收网络（远程）唤醒帧和魔术数据包帧。PMT 可为 MAC 接收到的唤醒帧和魔术数据包帧生成中断。可通过远程唤醒帧使能位以及魔术数据包使能位使能 PMT 模块。这些使能位（WFE 和 MPE）位于 ETH_MACPMTCSR 寄存器中，可由应用编程。在 PMT 中使能掉电模式时，MAC 将丢弃所有接收到的帧并且不会将这些帧转发给应用。仅当接收到魔术数据包或远程唤醒帧且使能相应检测时，MAC 才会退出掉电模式。

**远程唤醒帧过滤寄存器**

有八个唤醒帧过滤寄存器。要对每个寄存器执行写操作，需要逐个值加载唤醒帧过滤寄存器。连续加载唤醒帧过滤寄存器八次，便可对唤醒帧过滤寄存器加载所需的值。读操作与写操作相同。要读取八个值，必须读唤醒帧过滤寄存器八次后，才能到达最后一个寄存器。每次读/写操作都会将唤醒帧过滤寄存器指向下一个过滤寄存器。

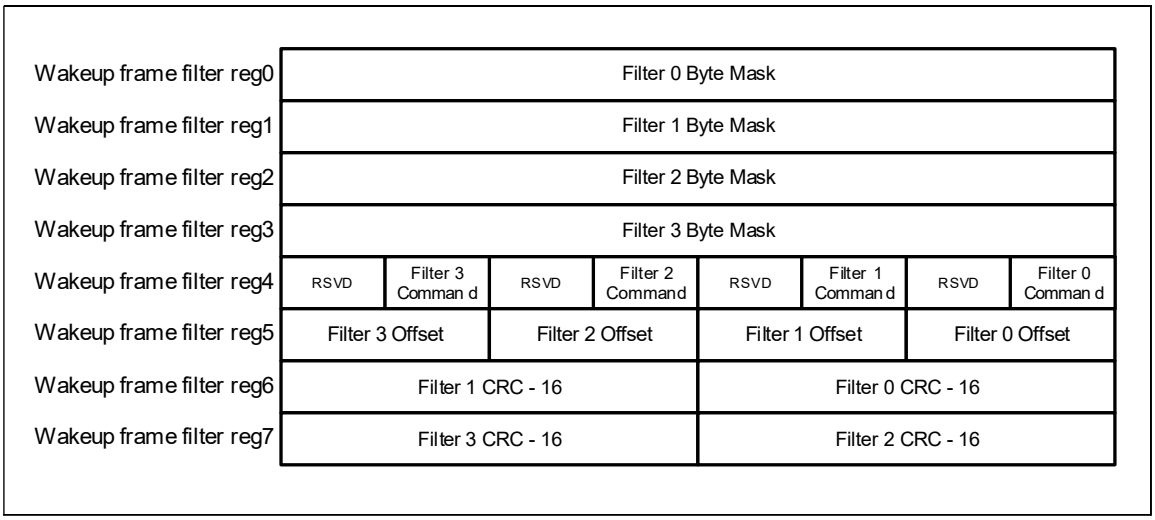



图 25-21 唤醒帧过滤寄存器

- 过滤器 i 字节屏蔽

  该寄存器定义过滤器 i（0、 1、 2 和 3）检测帧的哪些字节来确定帧是否为唤醒帧。MSB（第 31 位）必须为零。位 j[30:0] 为字节屏蔽。如果将字节屏蔽的位 j（字节数）置 1，则传入帧的过滤器 i 偏移 + j 由 CRC 模块处理；否则将忽略过滤器 i 偏移 + j。

- 过滤器 i 命令

  该 4 位命令控制过滤器 i 操作。位 3 指定地址类型，定义模式的目标地址类型。当该位置 1 时，模式只适用于多播帧。当该位复位时，模式只适用于单播帧。位 2 和位 1 为保留位。位 0 为过滤器 i 的使能位；如果位 0 未置 1，则将禁止过滤器 i。

- 过滤器 i 偏移

  该寄存器定义过滤器 i 要检测的帧的偏移（在帧范围内）。该 8 位模式偏移是要检测的过滤器 i 第一个字节的偏移。允许的最小值为 12，表示帧的第 13 个字节（偏移值 0 表示帧的第一个字节）。

- 过滤器 i CRC-16

  该寄存器包含根据模式计算的 CRC_16 值，以及对唤醒过滤寄存器模块编程的字节屏蔽。

**远程唤醒帧检测**

当 MAC 处于睡眠模式并已使能 ETH_MACPMTCSR 寄存器中的远程唤醒位时，MAC 可在接收到远程唤醒帧后恢复正常工作。应用可通过连续对唤醒帧过滤寄存器地址执行写操作，来写入全部八个唤醒过滤寄存器。应用可通过向 ETH_MACPMTCSR 寄存器中的位 2 写入 1 来使能远程唤醒。PMT 支持四种可编程过滤器，这些过滤器提供不同的接收帧模式。如果传入帧通过过滤命令的地址过滤，并且过滤器 CRC-16 与传入的检测模式相匹配，则可接收唤醒帧。Filter_offset（最小值 12，表示帧的第 13 个字节）确定要检测的帧的偏移。过滤器字节屏蔽确定必须检测帧的哪些字节。必须将字节屏蔽的第 31 位置 0。只需要检查唤醒帧是否存在长度错误、FCS 错误、帧尾错误、MII 错误、冲突，确保其不是矮帧。即使唤醒帧长度超过 512 字节，但只要帧的 CRC 值有效，帧便有效。ETH_MACPMTCSR 寄存器中的唤醒帧检测会针对每个接收到的远程唤醒帧进行更新。此外，还会生成 PMT 中断（如果已使能）来指示已接收到远程唤醒帧。

**魔术数据包检测**

魔术数据包帧基于一种特殊的方法，这种方法使用 AMD 公司的魔术数据包技术对网络上处于睡眠模式下的器件上电。MAC 接收称为魔术数据包的特定信息包，此信息包的目标地址是网络上的节点。只对目标地址为器件或多播地址的魔术数据包进行检查，以确定这些数据包是否满足唤醒要求。对通过地址过滤（单播或多播）的魔术数据包进行检测，确定其是否符合远程唤醒数据格式，即 6 个字节的数据所有位为全‘1’的数据包加上重复出现 16 次的 MAC 地址。应用通过向 ETH_MACPMTCSR 寄存器中的位 1 写入 1 来使能魔术数据包唤醒。PMT 模块持续监视目标地址为对应于指定魔术数据包模式的节点的每个帧。检查每个接收帧的目标地址和源地址字段之后是否为 0xFFFF FFFF FFFF 形式。之后，PMT 模块检查帧是否存在重复 16 次且没有任何断开或中断的 MAC 地址。如果重复 16 次的地址中存在中断，则会再次扫描传入帧中是否存在 0xFFFF FFFF FFFF 形式。这 16 次重复可位于帧中的任何位置，但必须在同步数据流后面 (0xFFFF FFFF FFFF)。此外，只要检测到重复 16 次的 MAC 地址，器件就可以接收多播帧。如果节点的 MAC 地址为 0x0011 2233 4455，则 MAC 扫描的数据序列为：

目标地址源地址 .....FFFF FFFF FFFF

0011 2233 4455 0011 2233 4455 0011 2233 4455 0011 2233 4455

0011 2233 4455 0011 2233 4455 0011 2233 4455 0011 2233 4455

0011 2233 4455 0011 2233 4455 0011 2233 4455 0011 2233 4455

0011 2233 4455 0011 2233 4455 0011 2233 4455 0011 2233 4455

...CRC

ETH_MACPMTCSR 寄存器中的魔术数据包检测会针对接收到的魔术数据包进行更新。此外，还会生成 PMT 中断（如果已使能）来指示已接收到魔术数据包。

**掉电期间的系统的注意事项**

如果使能 EXTI 中断线 19，以太网 PMT 模块能够在系统处于停止模式时检测帧。

MAC 接收器状态机在掉电模式期间应保持使能状态。这意味着，由于 ETH_MACCR 寄存器中的 RE 位与魔术数据包/网络唤醒帧检测有关，因此必须始终将该位置 1。但是，在掉电模式期间，应通过将 ETH_MACCR 寄存器中的 TE 位清零来关闭发送器状态机。此外，在掉电模式期间，应禁止以太网 DMA，原因是此时无需将魔术数据包/ 网络唤醒帧复制到 SRAM。要禁止以太网 DMA，需将 ETH_DMAOMR 寄存器中的 ST 位和 SR 位（分别对应发送 DMA 和接收 DMA）清零。

推荐的掉电和唤醒顺序如下：

- 禁止发送 DMA，并等待所有之前的帧发送完成。接收到发送中断 ETH_DMASR 寄存器[0] 时，便可检测到这些帧发送。
- 通过将 ETH_MACCR 配置寄存器中的 RE 位和 TE 位清零来禁止 MAC 发送器和 MAC 接收器。
- 等待接收 DMA 清空 Rx FIFO 中的所有帧。
- 禁止接收 DMA。
- 配置并使能 EXTI 中断线 19，以生成一个事件或中断。
- 如果将 EXTI 中断线 19 配置为生成中断，还必须正确配置 ETH_WKUP_IRQ 处理程序功能，此功能用于清零 EXTI 中断线 19 的挂起位。
- 通过将 ETH_MACPMTCSR 寄存器中的 MFE/WFE 位置 1 来使能魔术数据包/网络唤醒帧检测。
- 通过将 ETH_MACPMTCSR 寄存器中的 PD 位置 1 来使能 MAC 掉电模式。
- 通过将 ETH_MACCR 寄存器中的 RE 位置 1 来使能 MAC 接收器。
- 进入系统的停止模式。
- 接收到有效唤醒帧时，以太网外设会退出掉电模式。
- 读取 ETH_MACPMTCSR 以清零电源管理事件标志，使能 MAC 发送器状态机并接收和发送 DMA。
- 配置系统时钟：使能 HSE 并设置时钟。

### 25.5.9. 精密时间协议(IEEE1588 PTP)

IEEE 1588 标准定义了一种协议，此协议支持使用网络通信、局域计算和分布式对象等技术实现的测量和控制系统中的精密时钟同步。该协议适用于利用支持多播消息传送的局域网（包括但不限于以太网）进行通信的系统。该协议用于对非均匀系统进行同步，这类系统包含固有精度、分辨率和稳定性都不断变化的时钟。该协议支持亚秒范围的系统级同步精度，并且需要极少的网络和本地时钟计算资源。这种基于消息的协议称为精密时间协议 (PTP)，它通过 UDP/IP 传送。系统或网络归类为主节点和从节点，用于分配时序/时钟信息。该协议通过交换 PTP 消息来同步从节点与主节点，下图中对此有详细说明。

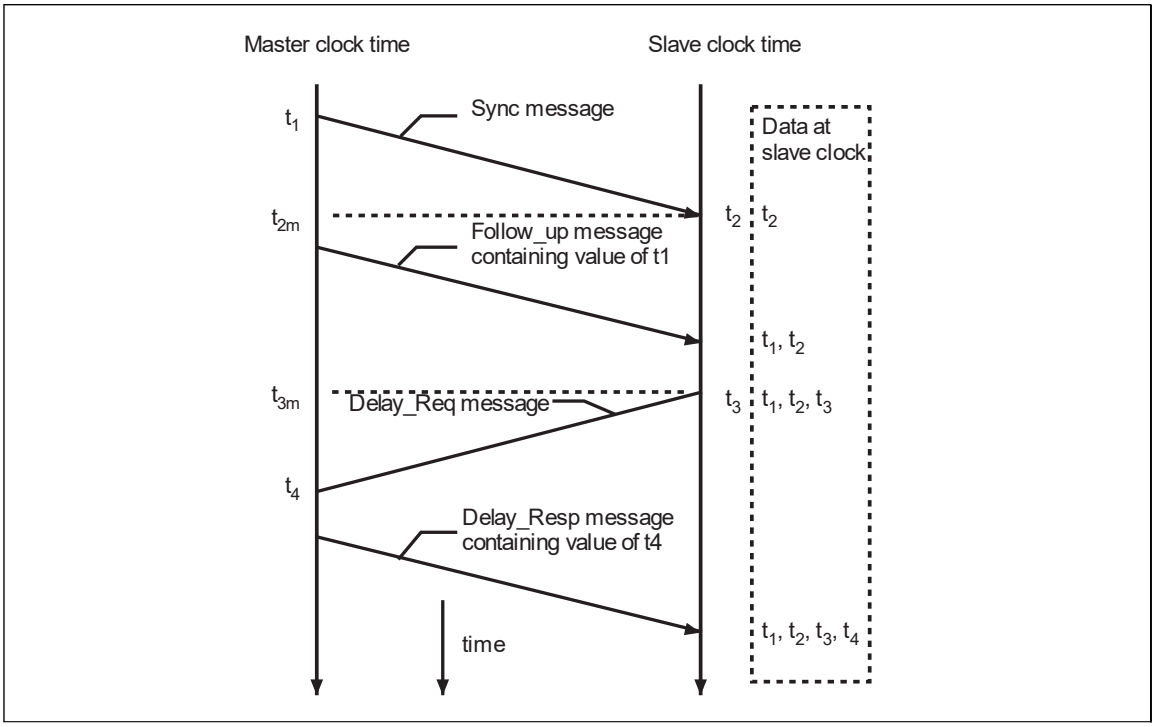



图 25-22 网络时间同步

- 主节点向其所有节点广播 PTP 同步消息。同步消息包含主节点的参考时间信息。消息离开主节点系统的时间为 $t_1$。对于以太网端口，必须通过 MII 接口来捕获该时间。
- 从节点接收同步消息并利用其参考时序捕获准确时间 $t_2$。
- 主节点随后会向从节点发送一个包含 $t_1$ 信息的 Follow_up 消息以备后用。
- 从节点向主节点发送一个 Delay_Req 消息，标出该帧离开 MII 的准确时间 $t_3$。
- 主节点接收该消息，捕获消息进入其系统的准确时间 $t_4$。
- 主节点将 Delay_Resp 消息中的 $t_4$ 信息发送给从节点。
- 从节点使用 $t_1$、$t_2$、$t_3$ 和 $t_4$ 这四个值来同步其本地参考时序与主节点的参考时序。

大多数协议都在 UDP 层之上通过软件实现。但如上所述，要通过 MII 接口来捕获特定 PTP 包进入或离开以太网端口时的准确时间，需要硬件支持。必须捕获该时序信息并将其返回给软件，以实现高精度的 PTP。

**参考时序源**

根据 IEEE 1588 规范的定义，要获取时间快照，内核需要一个 64 位格式的参考时间（分成两个 32 位通道，高 32 位表示时间的秒数，低 32 位表示时间的纳秒数）。

PTP 参考时钟输入用于在内部生成参考时间（亦称作系统时间）以及捕获时间戳。该参考时钟的频率必须大于等于时间戳计数器的分辨率。主节点与各从节点之间的同步精度目标约为 100 ns。

系统时间校准方法一节中介绍了系统时间的生成、更新与修改。

精度取决于 PTP 参考时钟输入周期、振荡器的特性（漂移）以及同步过程的频率。

由于从 Tx 和 Rx 时钟输入域到 PTP 参考时钟域的同步，时间戳锁存值的不确定度为 1 个参考时钟周期。如果加上分辨率导致的不确定度，则会增加一半的时间戳周期。

**使用 PTP 功能发送帧**

在 MII 上输出帧的 SFD 时，将捕获时间戳。对于需要捕获时间戳的帧，可按帧为单位进行控制。换句话说，可对每个发送帧进行标记以指示是否有必要捕获该帧的时间戳。无需处理发送帧即可识别 PTP 帧。可通过发送描述符中的控制位执行帧控制。捕获到的时间戳返回给应用的方式与提供帧状态的方式相同。时

间戳会随帧的发送状态一同发送回相应的发送描述符内，从而自动将时间戳与特定 PTP 帧相连。64 位时间戳信息会写回 TDES2 和 TDES3 字段，其中 TDES2 会保持时间戳的 32 个最低有效位。

**使用 PTP 功能接收帧**

使能 IEEE 1588 时间戳功能时，以太网 MAC 将捕获 MII 上接收到的所有帧的时间戳。MAC 会在完成帧接收过程时提供时间戳。捕获到的时间戳返回给应用的方式与提供帧状态的方式相同。时间戳会随帧的接收状态一同发送回相应的接收描述符内。64 位时间戳信息会写回 RDES2 和 RDES3 字段，其中 RDES2 会保持时间戳的 32 个最低有效位。

**系统时间校准方法**

使用 PTP 输入参考时钟 HCLK 更新 64 位 PTP 时间。该 PTP 时间可用作时钟源，以获取 MII 上发送或接收的以太网帧的快照（时间戳）。可使用粗略校准方法或精密校准方法对系统时间定时器进行初始化或校准。

使用粗略校准方法时，初始值或偏移值会写入时间戳更新寄存器（请参见 IEEE 1588 时间戳寄存器）。对于初始化，会将时间戳更新寄存器中的值写入系统时间计数器；对于系统时间校准，会将偏移值（ 时间戳更新寄存器）加到系统时间中或从系统时间中减去。

使用精密校准方法时，从时钟（参考时钟）频率相对于主时钟（如 IEEE 1588 中定义）的偏移会在一段时间内进行校准，而不像粗略校准方法中那样，在单个时钟周期内进行校准。校准时间越长，越有助于保持线性时间，并且不会导致各 PTP 同步消息间隔之间的参考时间发生剧烈变化（或者大型抖动）。在此方法中，会使用一个累加器对加数寄存器中的内容求和，如下图所示。累加器生成的算数进位将用作使系统时间计数器递增的脉冲。累加器和加数寄存器均为 32 位寄存器。此处，累加器用作高精度频率乘法器或除法器。下图给出了该算法。

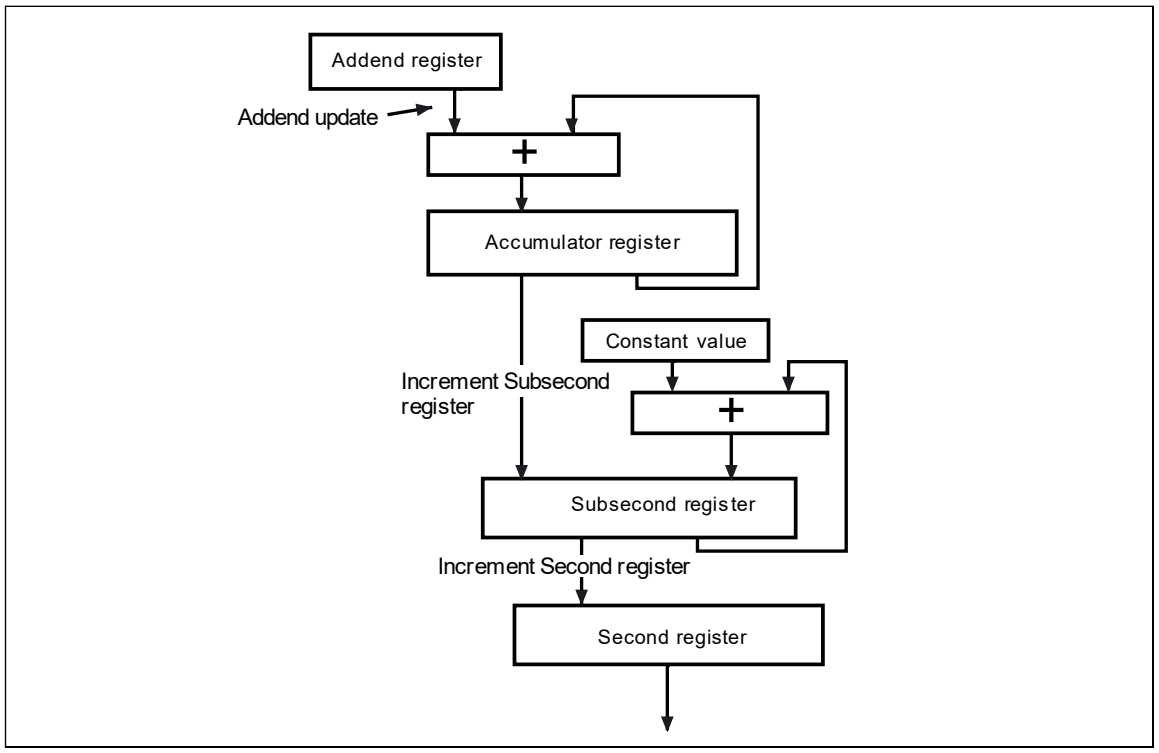



图 25-23 使用精密校准方法更新系统时间

系统时间更新逻辑需要使用 50 MHz 的时钟频率，以达到 20 ns 的精度。分频是指参考时钟频率与所需时钟频率的比率。因此，如果我们说参考时钟 (HCLK) 的频率为 66MHz，则通过计算可知分频比为 66 MHz/50 MHz = 1.32。故此，寄存器中设置的默认加数值为 232/1.32，相当于 0xC1F0 7C1F。

如果参考时钟向下偏移，例如降至 65 MHz，则分频比为 65/50 或 1.3，加数寄存器中要设置的值为 $2^{32}$/1.30，相当于 0xC4EC 4EC4。如果时钟向上偏移，例如升至 67 MHz，则必须将加数寄存器设置为

0xBF0 B7672。当时钟偏移为零时，应编程的默认加法值为 0xC1F07C1F ($2^{32}$/1.32)。

在上图中，用于使亚秒寄存器递增的常数值为 0d43。这可使系统时间的精度达到 20 ns（换句话说，增量步长时间为 20 ns）。

软件必须根据同步消息对频率偏移进行计算，并相应更新加数寄存器。首先，使用加数寄存器中的 FreqCompensationValue0 设置从时钟。该值的计算公式如下：

$$\mathbf{FreqCompensationValue0} = \mathbf{2^{32} / FreqDivisionRatio}$$

如果起初假定 MasterToSlaveDelay 对于连续的同步消息是相同的，则必须应用下述算法。数个同步周期之后，频率会锁定。从时钟随后可确定 MasterToSlaveDelay 的精确值，并使用新值重新与主时钟同步。

该算法如下：

- 在 MasterSyncTime (n) 时刻，主时钟向从时钟发送同步消息。从时钟在其本地时钟为 SlaveClockTime (n) 时接收到该消息，并用如下公式计算 MasterClockTime (n)：MasterClockTime (n) = MasterSyncTime (n) + MasterToSlaveDelay (n)；
- 当前同步周期的主时钟计数 MasterClockCount (n) 的计算公式如下：MasterClockCount (n) = MasterClockTime (n) - MasterClockTime (n - 1)（假定 MasterToSlaveDelay 对于同步周期 n 和 n - 1 是相同的）；
- 当前同步周期的从时钟计数 SlaveClockCount (n) 的计算公式如下：SlaveClockCount (n) = SlaveClockTime (n) - SlaveClockTime (n - 1)；
- 当前同步周期的主从时钟计数差值 SlaveClockCount (n) 的计算公式如下：ClockDiffCount (n) = MasterClockCount (n) - SlaveClockCount (n)；
- 从时钟的分频系数 FreqScaleFactor (n) 的计算公式如下：FreqScaleFactor (n) = (MasterClockCount (n) + ClockDiffCount (n)) / SlaveClockCount (n)；
- 加数寄存器的频率补偿值 FreqCompensationValue (n) 的计算公式如下：FreqCompensationValue (n) = FreqScaleFactor (n) × FreqCompensationValue (n - 1)。

理论上，该算法可在一个同步周期内实现锁定；但是，由于网络传播延迟和工作条件会不断变化，因此可能需要多个周期。

该算法可进行自校准：如果出于某些原因导致最初通过主时钟设置的从时钟不正确，则该算法会花费更多同步周期将其校准。

**系统时间生成初始化的编程步骤**

通过将时间戳控制寄存器 (ETH__PTPTSCR) 中的位 0 置 1 来使能时间戳功能。但之后必须对时间戳计数器进行初始化才能启动时间戳操作。适当的顺序如下：

- 通过将 MACIMR 寄存器中的位 9 置 1 以屏蔽时间戳触发中断。
- 编程时间戳寄存器位 0 以使能时间戳。
- 根据 PTP 时钟频率编程亚秒递增计数器。
- 如果使用精密校准方法，编程时间戳加数寄存器并将时间戳控制寄存器的位 5 置 1（加数寄存器更新）。
- 轮询时间戳控制寄存器，直到位 5 清零。
- 要选择精密校准方法（根据需要），编程时间戳控制寄存器位 1。
- 用适当的时间值编程时间戳高位更新寄存器和时间戳低位更新寄存器。
- 将时间戳控制寄存器位 2 置 1（时间戳初始化）。
- 用时间戳更新寄存器中写入的值初始化时间戳计数器后，时间戳计数器便开始运行。
- 使能 MAC 接收器和发送器以使时间戳功能正常运行。

*注意：如果将 ETH_PTPTSCR 寄存器中的位 0 清零来禁止时间戳操作，必须重复执行以上步骤以重新启动时间戳操作。*

**使用粗略校准方法更新系统时间的编程步骤**

要在一个过程（粗略校准方法）中同步或更新系统时间，请按照以下步骤执行：

- 将偏移（正偏移或负偏移）写入时间戳更新高位和低位寄存器。
- 将时间戳控制寄存器中的位 3 (TSSTU) 置 1。
- 当 TSSTU 位清零时，时间戳更新寄存器中的值将加到系统时间中或者从系统时间中减去。

**使用精密校准方法更新系统时间的编程步骤**

要同步或更新系统时间以减少系统时间抖动（精密校准方法），请按照以下步骤执行：

- 借助系统时间校准方法一节中介绍的算法，计算想要增加或减少系统时间增量时采用的速率。
- 更新时间戳。
- 等待加数寄存器中的新值生效。这可以通过在系统时间达到目标值后，激活时间戳触发中断来实现。
- 在目标时间高位和低位寄存器中编程所需目标时间。通过将 ETH_MACIMR 寄存器中的位 9 清零以取消屏蔽时间戳中断。
- 将时间戳控制寄存器位 4 (TSARU) 置 1。
- 当该寄存器产生中断时，读取 ETH_MACSR 寄存器。
- 用旧值重新编程时间戳加数寄存器并再次将 ETH_TPTSCR 位 5 置 1。

**与 TIM2 内部相连的 PTP 触发**

当系统时间大于目标时间时，MAC 会提供触发中断。使用中断会引起已知延迟，并导致命令执行时间出现不确定性。

为避免这种不确定性，可在系统时间大于目标时间时将 PTP 触发输出信号置为高电平。该信号内部连接到 TIM2 输入触发。通过该信号，由已同步的 PTP 系统时间触发的输入捕获功能、输出比较功能以及定时器的波形均可使用。这时不会引起不确定性，因为定时器的时钟（PCLK1： TIM2 APB1 时钟）以及 PTP 参考时钟 (HCLK) 是同步的。

该 PTP 触发信号会连接到可用软件选择的 TIM2 ITR1 输入。该连接可通过 TIM2 选项寄存器 (TIM2_OR) 中的位 11 和 10 使能。 下图显示了该连接。

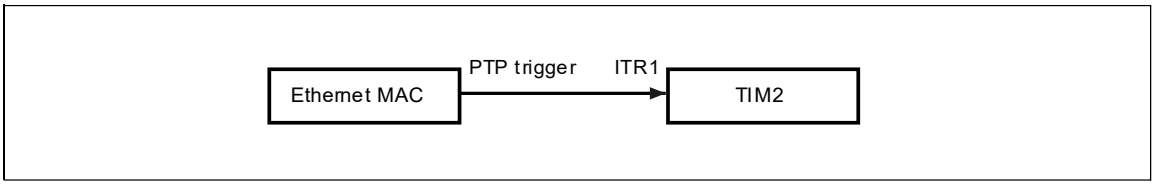



图 25-24 PTP 触发输出与 TIM2 ITR1 的连接

**PTP 每秒脉冲数输出信号**

该 PTP 脉冲输出用于检查网络中所有节点之间的同步性。要能够测试本地从时钟与主参考时钟之间的差值，可为两个时钟提供每秒脉冲数 (PPS) 输出信号（根据需要，可将该信号连接到示波器）。这样便可测量两个信号间的偏差。 PPS 输出的脉冲宽度为 125 ms。

PPS 输出可通过 TIM2 选项寄存器 (TIM2_OR) 中的位 11 和 10 使能。

PPS 输出的默认频率为 1 Hz。可使用 PPSFREQ[3:0]（位于 ETH_PTPPPSCR 中）将 PPS 输出的频率设置为 $2^{PPSFREQ}$ Hz。

如果设置为 1 Hz，则使用二进制翻转（TSSSR=0，ETH_PTPTSCR 寄存器中的位 9）时，PPS 脉冲宽度为 125 ms；使用数字翻转 (TSSSR=1) 时，PPS 脉冲宽度为 100 ms。如果设置为 2 Hz 或更高频率，则使用二进制翻转时，PPS 输出的占空比为 50%。

使用数字翻转 (TSSSR=1) 时，建议不要使用频率为 1 Hz 以外的 PPS 输出，因为波形可能不规则（尽管其平均频率可能在任意一秒窗口期间始终正确）。

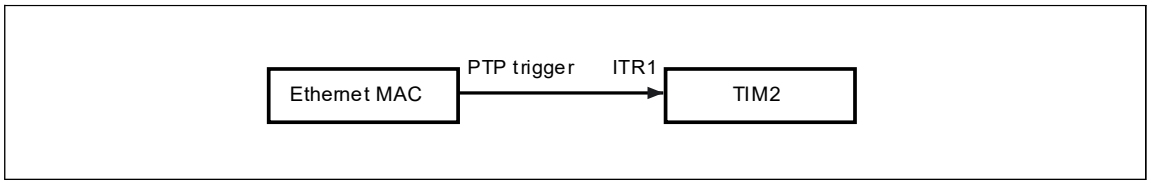



图 25-25 PPS 输出

## 25.6. 以太网功能说明：DMA 控制器操作

DMA 具有独立的发送和接收引擎以及相应的 CSR（控制和状态寄存器）空间。发送引擎将数据从系统存储器传送到 Tx FIFO，而接收引擎将数据从 Rx FIFO 传送到系统存储器。DMA 可以在 CPU 完全不干预的情况下，通过描述符有效地将数据从源传送到目标。DMA 专为面向包的数据传送（如以太网中的帧）而设计。该控制器经过编程后，可在完成帧发送和接收传送操作时以及其它正常/错误条件下产生 CPU 中断。DMA 和 MCU 通过以下两种数据结构进行通信：

- 控制和状态寄存器 (CSR);
- 描述符列表和数据缓冲区。

DMA 既可将接收到的数据帧传送到 MCU 存储器中的接收缓冲区，也可以传送存储器的发送缓冲区中的数据帧。位于存储器中的描述符用作指向这些缓冲区的指针。共有两个描述符列表：一个用于接收，一个用于发送。两个列表的基址分别写入 DMA 寄存器 3 和寄存器 4。描述符列表是一种前向链表（无论是隐式还是显式）。最后一个描述符会指回第一个描述符以构成环形结构。可通过配置接收和发送描述符（RDES1[14] 和 TDES0[20]）中链接的第二个地址来完成描述符的显式链接。描述符列表位于主机的物理存储空间。每个描述符最多可指向两个缓冲区。这样便能使用两个物理寻址的缓冲区替代存储器中两个连续的缓冲区。数据缓冲区位于主机的物理存储空间，通常由整个帧或部分帧组成，但不会超过单个帧。缓冲区中仅包含数据。其状态保存在描述符中。数据链接是指跨越多个数据缓冲区的帧。但是单个描述符不能跨越多个帧。检测到帧结束时，DMA 会跳到下一个帧缓冲区。可以使能或禁止数据链接。描述符的环形结构与链接结构如下图所示。

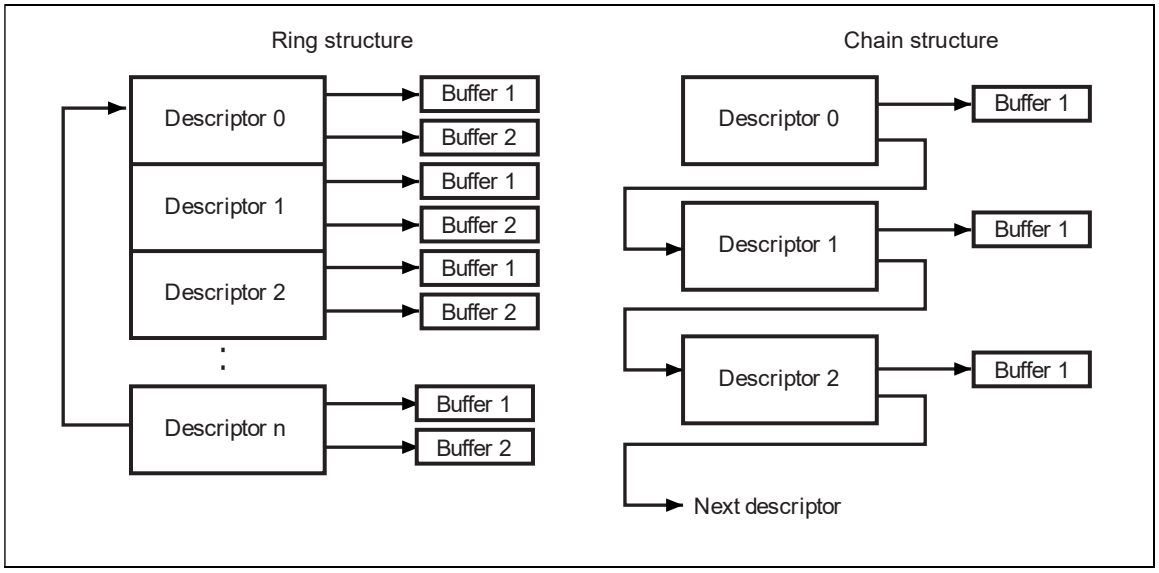



图 25-26 描述符环形结构与链接结构

### 25.6.1. 使用 DMA 进行传送的初始化

MAC 的初始化步骤如下：

- 对 ETH_DMABMR 执行写操作以设置 MCU 总线访问参数。
- 对 ETH_DMAIER 寄存器执行写操作以屏蔽不必要的中断源。
- 软件驱动器创建发送和接收描述符列表。随后对 ETH_DMARDLAR 和 ETH_DMATDLAR 寄存器执行写操作，为 DMA 提供各列表的起始地址。
- 对 MAC 寄存器 1、2 和 3 执行写操作以选择所需的过滤选项。
- 对 MAC ETH_MACCR 寄存器执行写操作以配置和使能发送与接收工作模式。可根据自动协商结果将 PS 位与 DM 位置 1（读取自 PHY）。
- 对 ETH_DMAOMR 寄存器执行写操作，将位 13 和位 1 置 1 以启动发送和接收。
- 发送与接收引擎进入运行状态，并尝试从相应描述符列表中获取描述符。这两个引擎随后开始处理接收和发送操作。发送和接收处理过程彼此独立，可单独进行启动或停止。

### 25.6.2. 主机总线突发访问

DMA 会尝试在 AHB 主接口上执行固定长度的突发传送（如果已使用 ETH_DMABMR 中的 FB 位进行相应配置）。突发的最大长度由 PBL 字段 (ETH_DMABMR [13:8]) 指示和限制。对于要读取的 16 个字节，接收和发送描述符始终采用可能的最大突发大小（由 PBL 限制）进行访问。

仅当发送 FIFO 中的空间足以容纳配置的突发或帧结束之前的字节数时（当帧短于配置的突发长度时），发送 DMA 才会启动数据传送。 DMA 会向 AHB 主接口指示起始地址和所需的传送次数。当 AHB 接口配置为固定长度突发时，将使用 INCR4、INCR8、INCR16 和单独事务的最佳搭配来传送数据。否则（非固定长度突发）会使用 INCR（未定义长度）与单独事务传送数据。

仅当接收 FIFO 中有足够的数据用于配置的突发时，或者在接收 FIFO 中检测到帧结束时（当帧短于配置的突发长度时），接收 DMA 才会启动数据传送。DMA 会向 AHB 主接口指示起始地址和所需的传送次数。当 AHB 接口配置为固定长度突发时，将使用 INCR4、INCR8、INCR16 和单独事务的最佳搭配来传送数据。如果在 AHB 接口上的固定突发结束前已到达帧结束，将执行空传送以完成固定长度的突发传送。否则（复位 ETH_DMABMR 中的 FB 位）会使用 INCR（未定义长度）和单独事务传送数据。

当 AHB 接口配置为地址对齐的节拍时，两个 DMA 引擎会确保 AHB 启动的第一次突发传送小于或者等于已配置 PBL 的大小。这样，后续的所有节拍都会从与已配置 PBL 对齐的地址开始。由于 AHB 接口多于 INCR16 的传送，DMA 只能对齐节拍最大为 16（对于 PBL > 16）的地址。

### 25.6.3. 主机数据缓冲区对齐

发送和接收数据缓冲区在起始地址对齐方面没有任何限制。在我们的系统（ 32 位寻址空间）中，缓冲区的起始地址可与四个字节中的任意一个对齐。但是，DMA 始终在地址与总线宽度对齐时启动传输，并且在不需要的字节通道上传输空数据。这通常发生在以太网帧的开始或结束传送期间。

- 缓冲区读操作示例：

  如果发送缓冲区地址为 0x0000 0FF2，并且需要传送 15 个字节，则 DMA 将从地址 0x0000 0FF0 读取 5 个全字，但在将数据传送到发送 FIFO 时，将丢弃或忽略额外的字节（前两个字节）。同样地，还将忽略最后一次传送的最后 3 个字节。 DMA 始终可确保向发送 FIFO 传送的是全 32 位数据项，除非是帧结束。

- 缓冲区写操作示例：

  如果接收缓冲区地址为 0x0000 0FF2，并且需要传送接收到的帧的 16 个字节，则 DMA 将从地址

0x0000 0FF0 开始写入 5 个全 32 位数据项。但是，第一次传送的前 2 个字节与第三次传送的最后 2 个字节将包含空数据

### 25.6.4. 缓冲区大小计算

DMA 不会更新发送和接收描述符中的大小字段。DMA 只更新描述符的状态字段 (xDES0)。必须用驱动程序计算大小。发送 DMA 会向 MAC 内核传送准确的字节数（由 TDES1 中的缓冲区大小字段指示）。如果将描述符标记为第一个描述符（将 TDES0 中的 FS 位置 1），则 DMA 会将缓冲区的第一次传送标记为帧起始。如果将描述符标记为最后一个描述符（TDES0 中的 LS 位），则 DMA 会将数据缓冲区的最后一次传送标记为帧结束。接收 DMA 持续将数据传送至缓冲区，直到缓冲区已满或接收到帧结束为止。当描述符的 FS 位置 1 时，如果未将描述符标记为最后一个描述符（RDES0 中的 LS 位），则该描述符对应的缓冲区会填满，并且缓冲区中的有效数据量将通过缓冲区大小字段减去数据缓冲指针偏移得到的结果表示。当数据缓冲区指针与数据总线宽度对齐时，偏移为零。如果将描述符标记为最后一个描述符，则缓冲区不会填满（由 RDES1 中的缓冲区大小指示）。要计算该缓冲区中最终的有效数据量，驱动程序必须读取帧长度（RDES0[29:16] 中的 FL 位），并减去该帧中先前缓冲区的缓冲区大小之和。接收 DMA 始终使用新的描述符传送下一个帧的帧起始。

*注意： 即使当接收缓冲区的起始地址与系统数据总线宽度未对齐时，系统也应分配一个大小与系统总线宽度对齐的接收缓冲区。例如，如果系统分配一个起始地址为 0x1000、大小为 1024 字节 (1 KB) 的接收缓冲区，则软件可将接收描述符中的缓冲区起始地址编程为具有 0x1002 偏移。接收 DMA 将帧写入该缓冲区，其中前两个单元（0x1000 和 0x1001）中为空数据。实际帧从单元 0x1002 开始写入。因此，尽管已将缓冲区大小编程为 1024 字节，但由于存在起始偏移地址，因此该缓冲区的实际有用空间为 1022 字节。*

### 25.6.5. DMA 仲裁器

DMA 内的仲裁器会在发送和接收通道对 AHB 主接口进行的访问之间进行仲裁。可以使用两类仲裁：循环调度和固定优先级。如果选择循环调度仲裁（复位 ETH_DMABMR 中的 DA 位），仲裁器会在发送和接收 DMA 同时请求访问时，按照 ETH_DMABMR 中的 PM 位设置的比率分配数据总线。当 DA 位置 1 时，接收 DMA 进行数据访问的优先级始终高于发送 DMA。

### 25.6.6. 对 DMA 的错误响应

对于由 DMA 通道发起的任何数据传送，如果从机给出错误响应，则相应 DMA 将停止所有操作并更新状态寄存器（ETH_DMASR 寄存器）中的错误位和致命总线错误位。此外，该 DMA 控制器只能在软复位或硬复位外设以及重新初始化 DMA 之后才能恢复操作。

### 25.6.7. Tx DMA 配置

#### TxDMA 操作：默认(非 OSF)模式

发送 DMA 引擎在默认模式下的操作顺序如下：

- 在为数据缓冲区设置好相应的以太网帧数据后，用户配置发送描述符 (TDES0-TDES3)并将 OWN 位 (TDES0[31]) 置 1。
- ST 位（ETH_DMAOMR 寄存器 [13]）置 1 后， DMA 会立即进入运行状态。
- 在运行状态下，DMA 会为需要传送的帧轮询发送描述符列表。轮询启动后，DMA 会以连续的描述符环形顺序或链接顺序持续进行。如果 DMA 检测到标记为 CPU 所有的描述符，或者发生错误条件，则会挂起传送并将发送缓冲区不可用位（ETH_DMASR 寄存器 [2]）与正常中断汇总使能位（ETH_DMASR 寄存器 [16]）置 1。发送引擎继续执行步骤 9。
- 如果获得的描述符标记为 DMA 所有（TDES0[31] 置 1），则 DMA 将根据取得的描述符对发送数据

缓冲区地址进行解码。

- DMA 从存储器中获取发送数据并传送这些数据。
- 如果以太网帧保存在多个描述符的数据缓冲区中，则 DMA 将关闭中间描述符并获取下一个描述符。重复执行步骤 3、4 和 5，直到完成以太网帧数据的传送。
- 完成帧传送后，如果已为帧使能 IEEE 1588 时间戳（按发送状态中的指示），则时间戳值将写入包含帧结束缓冲区的发送描述符（TDES2 和 TDES3）随后，状态信息将写入该发送描述符 (TDES0)。由于此步骤中会清零 OWN 位，因此该描述符现在由 CPU 所有。如果没有为该帧使能时间戳，DMA 不会更改 TDES2 和 TDES3 的内容。
- 发送完在其最后一个描述符中将完成时中断 (TDES1[31]) 置 1 的帧后，发送中断（ETH_DMASR 寄存器 [0]）将置 1。随后，DMA 引擎返回步骤 3。
- 在挂起状态下，DMA 会在接收到发送轮询要求时尝试重新获取描述符（因此返回步骤 3），并将下溢中断状态位清零。

下图给出了默认模式下 TxDMA 发送过程的流程图。

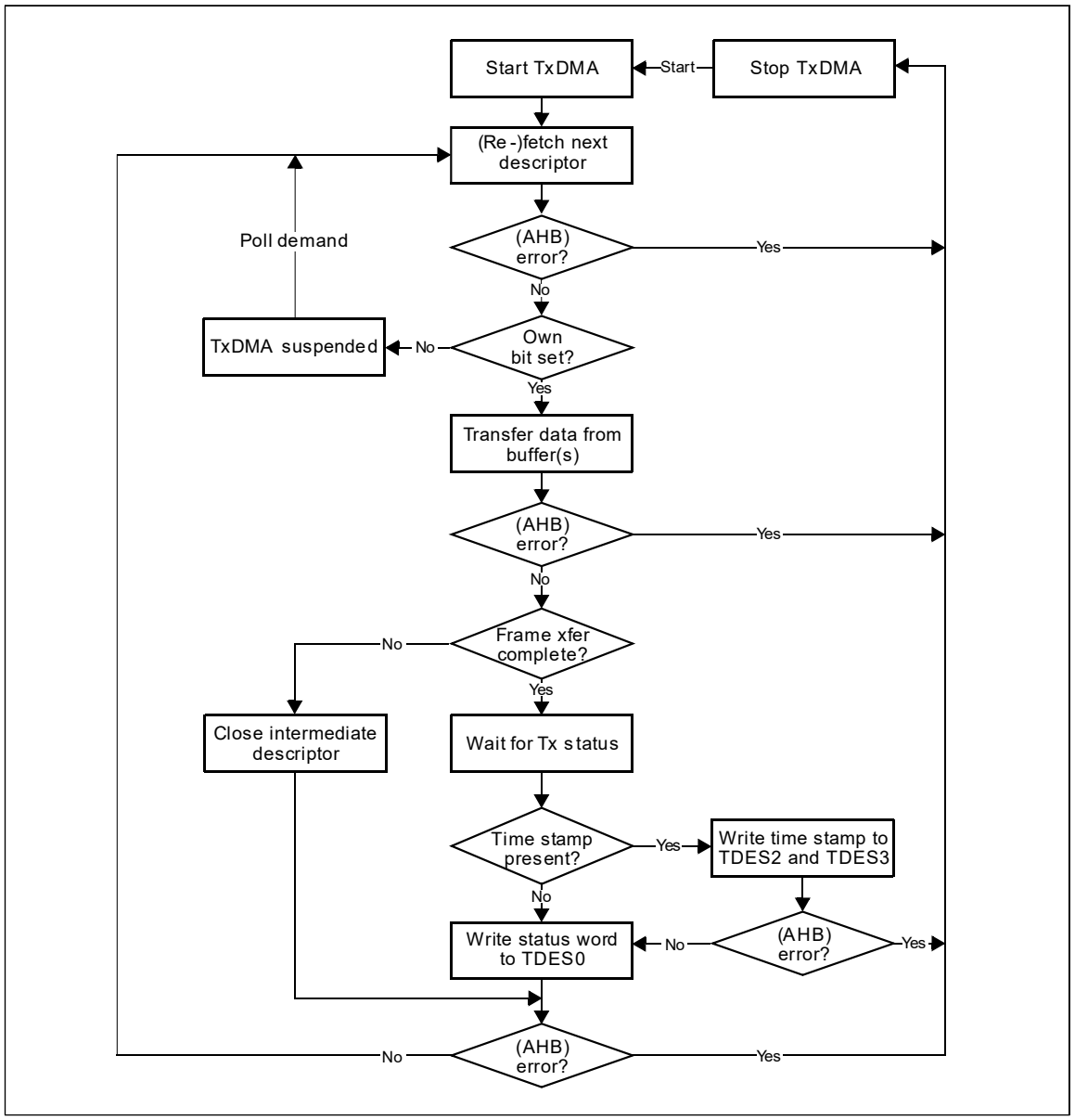



图 25-27 默认模式下的 TxDMA 操作

### TxDMA 操作：OSF 模式

在运行状态下，发送过程无需关闭第一个帧的状态描述符便可同时获取两个帧（如果 ETH_DMAOMR 寄存器 [2] 中的 OSF 位置 1）。发送过程完成第一个帧的传送后，会立即轮询发第二个帧的发送描述符列表。如果第二个帧有效，发送过程会在写入第一个帧的状态信息前发送此帧。在 OSF 模式下，运行状态发送 DMA 按照以下顺序进行操作：

- DMA 的操作过程如默认模式下 TxDMA 的步骤 1—6 中所述。
- 无需关闭前一帧的最后一个描述符，DMA 便可获取下一个描述符。
- 如果 DMA 拥有所需描述符，便会对该描述符中的发送缓冲区地址进行解码。 如果 DMA 未拥有描述符，则会进入挂起模式并跳到步骤 7。
- DMA 从存储器中取得发送帧并发送该帧，直到帧数据发送结束，如果该帧拆分到多个描述符中，则会同时关闭中间描述符。
- DMA 等待前一个帧的发送状态和时间戳。当状态可用时，如果捕获到时间戳（由状态位指示），DMA 会将这些时间戳写入 TDES2 和 TDES3。之后，OWN 位清零， DMA 将状态写入相应的 TDES0，进而关闭描述符。如果没有为前一个帧使能时间戳，DMA 不会更改 TDES2 和 TDES3 的内容。
- 如果已使能时间戳，发送中断将置 1，DMA 获取下一个描述符，然后继续执行步骤 3（状态正常时）。如果上一个发送状态显示下溢错误，则 DMA 会进入挂起模式（步骤 7）。
- 在挂起模式下，如果 DMA 接收到挂起状态和时间戳，则会将时间戳（如果已为当前帧使能时间戳）写入 TDES2 和 TDES3，随后将状态写入相应的 TDES0。之后，将相关中断置 1 并返回挂起模式。
- 只有在接收到发送轮询要求（ETH_DMATPDR 寄存器）后， DMA 才会退出挂起模式并进入运行状态（转到步骤 1 或步骤 2，具体取决于挂起状态）。

下图显示了 OSF 模式下的基本流程图。

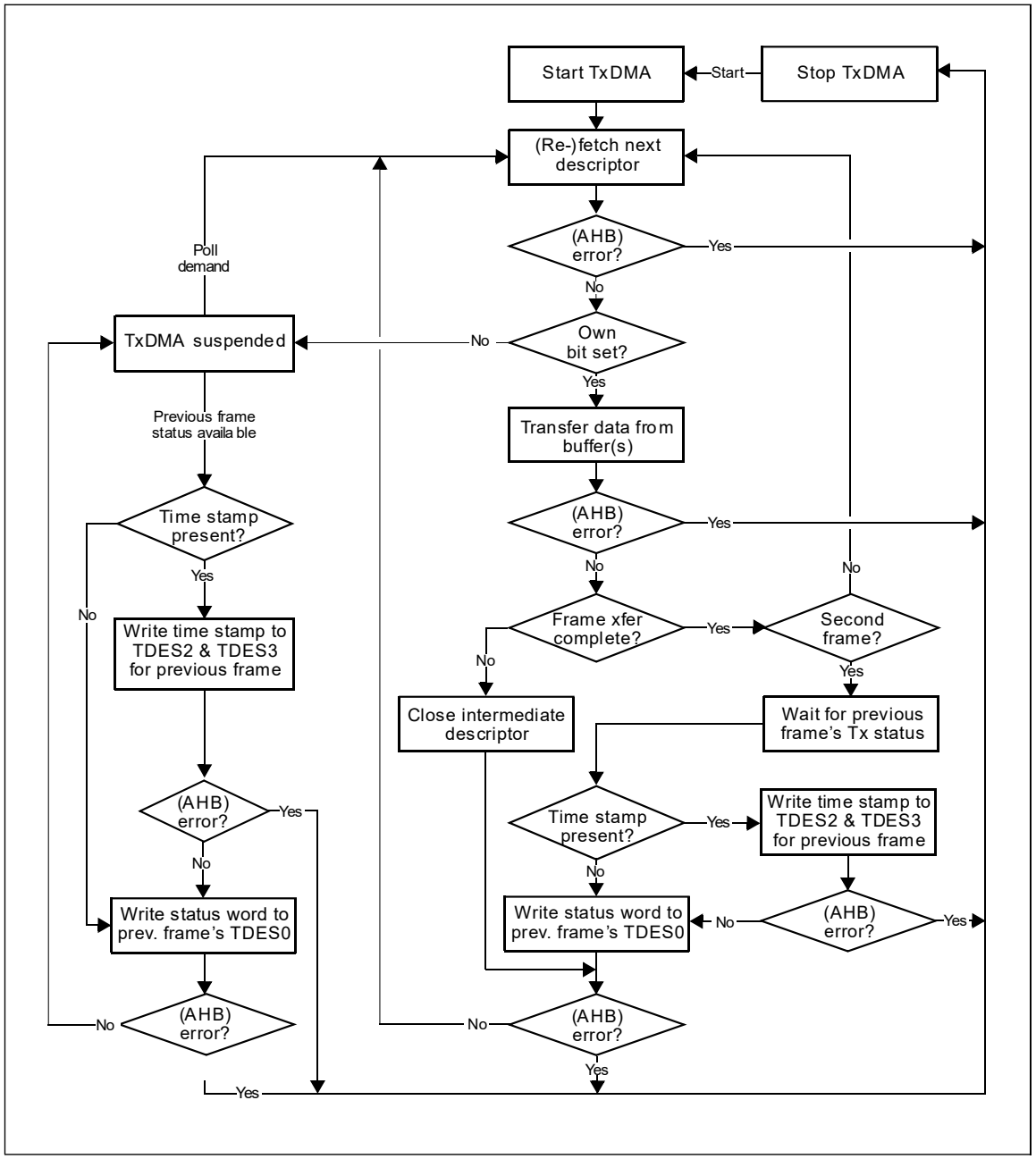



图 25-28 OSF 模式下的 TxDMA 操作

### 发送帧处理

发送 DMA 期望数据缓冲区包含完整的以太网帧，不包括报头、填充字节和 FCS 字段。DA、 SA 和类型/长度字段包含有效数据。如果发送描述符指示 MAC 内核必须禁止插入 CRC 或填充字节，则缓冲区必须具有包括 CRC 字节的完整以太网帧（不包括报头）。帧可以采用数据链接形式，也可以跨越多个缓冲区。帧必须由第一个描述符 (TDES0[28]) 和最后一个描述符 (TDES0[29]) 定界。传送启动时，必须将第一个描述符的 TDES0[28] 置 1。之后，帧数据便从存储器缓冲区传送到发送 FIFO。与此同时，如果当前帧的最后一个描述符(TDES0[29]) 清零，发送过程将尝试获取下一个描述符。发送过程期望 TDES0[28] 在此描述符中清零。如果 TDES0[29] 清零，则指示中间缓冲区。如果 TDES0[29] 置 1，则指示帧的最后一个缓冲区。当帧的最后一个缓冲区完成传送后， DMA 会将最终状态信息写回描述符的发送描述符 0 (TDES0) 字，该描述符在发送描述符 0 (TDES0[29]) 中将末段置 1。此时，如果完成时中断 (TDES0[30]) 置 1，则发送

中断（ETH_DMASR 寄存器 [0] 中）将置 1，并获取下一个描述符，然后重复执行该过程。实际的帧传送过程在发送 FIFO 达到可编程的发送阈值时（ETH_DMAOMR 寄存器 [16:14]），或者 FIFO 中包含完整的帧时才会启动。此外，还可以选择存储转发模式（ ETH_DMAOMR 寄存器 [21]）。DMA 完成帧的传送后会释放描述符（清零 OWN 位 TDES0[31]）。

### 发送轮询挂起

可通过以下任意条件暂停发送轮询：

- DMA 检测到 CPU 所有的描述符 (TDES0[31]=0)，并且发送缓冲区不可用标志置 1（ETH_DMASR 寄存器 [2]）。如要恢复，驱动程序必须将描述符的所有权交给 DMA，然后发出轮询查询要求命令。
- 检测到由下溢导致的传送错误时，将中止帧的传送。相应的发送描述符 0 (TDES0) 位将置 1。如果发生第二个条件，异常中断汇总位（ETH_DMASR 寄存器 [15] 中）与发送下溢位（ETH_DMASR 寄存器 [5] 中）都将置 1，并且信息将写入发送描述符 0，从而导致挂起。如果 DMA 由于第一个条件而进入挂起状态，则正常中断汇总位（ETH_DMASR 寄存器 [16]）与发送缓冲区不可用位（ETH_DMASR 寄存器 [2]）均置 1。两种情况下，发送列表中的位置都会保留。保留的位置是 DMA 关闭的最后一个描述符后面的描述符位置。纠正挂起原因后，驱动程序必须明确发出发送轮询要求命令。

### 常规 Tx DMA 描述符

常规发送描述符结构由四个 32 位字组成，如下图所示。下文给出了 TDES0、TDES1、TDES2 和 TDES3 的位说明。

请注意，如果时间戳已激活（ETH_PTPTSCR 位 0，TSE=1）或 IPv4 校验和减荷已激活（ETH_MACCR 位 10，IPCO=1），则必须使用增强的描述符。

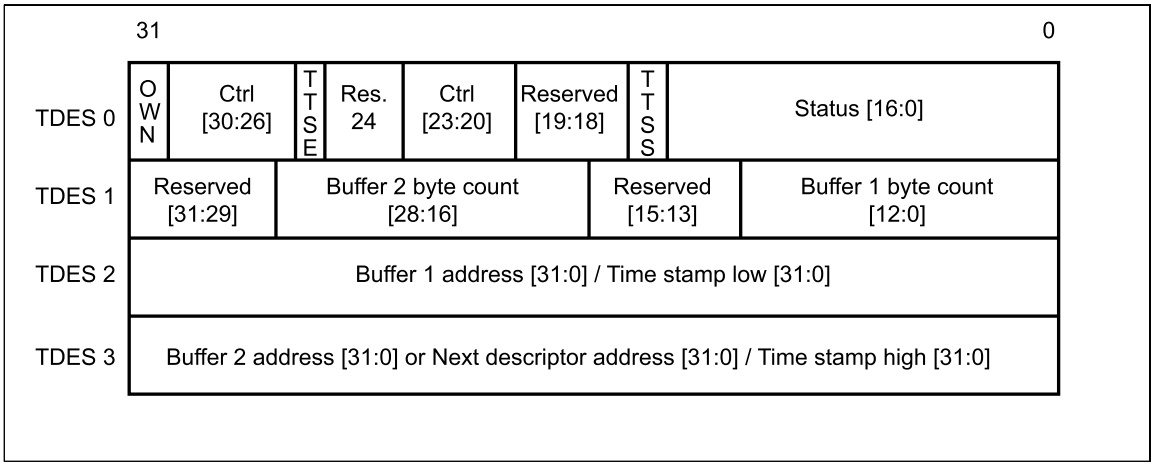



图 25-29 常规发送描述符

- TDES0：发送描述符字 0

  应用软件必须在描述符初始化期间编程控制位 [30:26]+[23:20] 以及 OWN 位 [31]。当 DMA 更新描述符（或将其写回）时，将复位所有控制位与 OWN 位，并且只报告状态信息。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OWN | IC | LS | FS | DC | DP | TTSE | Res | CIC | | TER | TCH | Res | | TTSS | IHE |
| rw | rw | rw | rw | rw | rw | rw | | rw | rw | rw | rw | | | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ES | JT | FF | IPE | LCA | NC | LCO | EC | VF | CC | | | | ED | UF | DB |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31 | OWN | 所有关系位<br>该位置 1 时, 表示描述符为 DMA 所有。该位复位时, 表示描述符为 CPU 所有。DMA 可在 完成帧发送时或在描述符分配的缓冲区全部读取完成后将该位清零。将属于同一个帧的所有 后续描述符置 1 后，必须将帧的第一个描述符的所有位置 1。 |
| 位 30 | IC | 完成时中断<br>该位置 1 时，当前帧发送完毕后，发送中断位（寄存器 5[0]）置 1。 |
| 位 29 | LS | 末段<br>该位置 1 时，指示缓冲区中包含帧的末段。 |
| 位 28 | FS | 首段<br>该位置 1 时，指示缓冲区中包含帧的首段。 |
| 位 27 | DC | 禁止 CRC<br>该位置 1 时， MAC 不会将循环冗余校验 (CRC) 附加到所发送帧的末尾。该位仅在首段位 (TDES0[28]) 置 1 的情况下有效。 |
| 位 26 | DP | 禁止 pad<br>该位置 1 时， MAC 不会自动为不足 64 字节的帧添加补位项。该位复位时，DMA 会自动为 不足 64 字节的帧添加补位项和 CRC，是否添加 CRC 字段与 DC (TDES0[27]) 位状态无关。<br>该位仅在首段位 (TDES0[28]) 置 1 的情况下有效。 |
| 位 25 | TTSE | 发送时间戳使能<br>TTSE 位置 1 且 TSE 置 1（ETH_PTPTSCR 位为 0）时，将针对描述符所描述的发送帧激活 IEEE1588 硬件时间戳功能。该字段仅在首段控制位 (TDES0[28]) 置 1 的情况下有效。 |
| 位 24 | 保留 | 必须保持复位值 |
| 位 23:22 | CIC | 校验和插入控制<br>这些位控制校验和的计算与插入。位编码如下所示:<br>00：禁止插入校验和<br>01：仅使能 IP 报头校验和的计算与插入<br>10：使能 IP 报头校验和以及有效负载校验和的计算与插入，但不会在硬件中计算伪报头校验和<br>11：使能 IP 报头校验和以及有效负载校验和的计算与插入，并在硬件中计算伪报头校验和。 |
| 位 21 | TER | 环发送结束<br>该位置 1 时，表示描述符列表已到达其最后一个描述符。DMA 会返回描述符列表的基址，并形成一个描述符环。 |
| 位 20 | TCH | 链接的第二个地址<br>该位置 1 时，表示描述符中的第二个地址是下一个描述符地址，而非第二个缓冲区地址。TDES0[20] 置 1 时，TBS2 (TDES1[28:16]) 为“无关”值。TDES0[21] 优先级高于 TDES0[20]。 |
| 位 19:18 | 保留 | 必须保持复位值 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 17 | TTSS | 发送时间戳状态<br>此字段用作状态位，指示已为所描述的发送帧捕获到时间戳。该位置 1 时， TDES2 和 TDES3 将保存为发送帧捕获到的时间戳值。该字段仅在描述符末段控制位 (TDES0[29]) 置 1 的情况下有效。<br>注意，使能增强的描述符时（ETH_DMABMR 中的 EDFE=1），TTSS=1 指示 TDES6 和 TDES7 中存有时间戳值。 |
| 位 16 | IHE | IP 报头错误<br>该位置 1 时，指示 MAC 发送器在 IP 数据报头中检测到错误。发送器将对照从应用程序接收的报头字节数检查 IPv4 数据包的报头长度，如果出现不匹配，则指示错误状态。对于 IPv6 帧，如果主报头长度不是 40 个字节，则报告一个报头错误。此外，IPv4 或 IPv6 帧的以太网 长度/类型字段值必须与随数据包接收的 IP 报头版本匹配。对于 IPv4 帧，如果报头长度字段的值小于 0x5，也会指示一个错误状态。 |
| 位 15 | ES | 错误汇总<br>指示以下位的逻辑或运算结果：<br>TDES0[14]：Jabber 超时 (Jabber timeout)<br>TDES0[13]：帧刷新 (Frame flush)<br>TDES0[11]：载波丢失 (Loss of carrier)<br>TDES0[10]：无载波 (No carrier)<br>TDES0[9]：延迟冲突 (Late collision)<br>TDES0[8]：过度冲突 (Excessive collision)<br>TDES0[2]：过度延迟 (Excessive deferral)<br>TDES0[1]：下溢错误 (Underflow error)<br>TDES0[16]：IP 报头错误 (IP header error)<br>TDES0[12]：IP 有效负载错误 (IP payload error) |
| 位 14 | JT | Jabber 超时<br>该位置 1 时，指示 MAC 发送器经历了 jabber 超时。只有在 MAC 配置寄存器的 JD 位未置 1 时，该位才会置 1。 |
| 位 13 | FF | 帧已刷新<br>该位置 1 时，指示 DMA/MTL 已依照 CPU 发出的软件刷新命令刷新帧。 |
| 位 12 | IPE | IP 有效负载错误<br>该位置 1 时，指示 MAC 发送器在 TCP、UDP 或 ICMP IP 数据报有效负载中检测到错误。发送器将对照从应用程序接收的 TCP、UDP 或 ICMP IP 数据包实际字节数检查 IPv4 或 IPv6 报头中接收的有效负载长度，如果出现不匹配，则指示错误状态。 |
| 位 11 | LCA | 载波丢失<br>该位置 1 时，指示帧发送期间丢失载波（即，帧发送期间有一个或多个发送时钟周期的 MII_CRS 信号无效）。该位仅对在 MAC 处于半双工模式时实现无冲突发送的帧有效。 |
| 位 10 | NC | 无载波<br>该位置 1 时，指示在发送期间未由 PHY 触发载波侦听信号。 |
| 位 9 | LCO | 延迟冲突<br>该位置 1 时，指示帧发送过程因冲突窗口（MII 模式下为 64 个字节时间，含报头）后出现冲 突而中止。如果下溢错误位置 1，则该位无效。 |
| 位 8 | EC | 过度冲突<br>该位置 1 时，指示尝试发送当前帧时因出现 16 个连续冲突而中止发送。如果 MAC 配置寄存 器中的 RD（禁止重试）位置 1，则此位会在出现首 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 个冲突后置 1 并且帧发送过程将中止。 |
| 位 7 | VF | VLAN 帧<br>该位置 1 时，指示所发送帧为 VLAN 类型的帧。 |
| 位 6:3 | CC | 冲突计数<br>此 4 位计数器值指示发送帧之前出现的冲突个数。过度冲突位 (TDES0[8]) 置 1 时，该计数无效。 |
| 位 2 | ED | 过度延迟<br>该位置 1 时， 如果 MAC 控制寄存器中的延迟检查 (DC) 位为高电平，则此位指示因出现超过 24288 个位时间的过度延迟而中止发送。 |
| 位 1 | UF | 下溢错误<br>该位置 1 时，指示 MAC 因 RAM 存储器的数据未及时到达而中止了帧发送。下溢错误表示 DMA 在帧发送期间遇到发送缓冲区为空的情况。发送过程将进入挂起状态并将发送下溢（寄存器 5[5]）和发送中断（寄存器 5[0]）位置 1。 |
| 位 0 | DB | 延迟位<br>该位置 1 时，指示 MAC 在发送前因存在载波而延迟。该位仅在半双工模式下有效。 |

TDES1：发送描述符字 1

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Reserved | | | TBS2 | | | | | | | | | | | | |
| | | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | TBS1 | | | | | | | | | | | | |
| | | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:29 | 保留 | 必须保持复位值 |
| 位 28:16 | TBS2 | 发送缓冲区 2 大小<br>这些位以字节为单位指示第二个数据缓冲区的大小。如果 TDES0[20] 位置 1，则该字段无效。 |
| 位 15:13 | 保留 | 必须保持复位值 |
| 位 12:0 | TBS1 | 发送缓冲区 1 大小<br>这些位以字节为单位指示第一个数据缓冲区的大小。如果该字段为 0，DMA 将忽略该缓冲区并使用缓冲区 2 或下一个描述符，具体取决于 TCH (TDES0[20]) 的值。 |

TDES2：发送描述符字 2

TDES2 包含描述符第一个缓冲区的地址指针，或者包含时间戳数据。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TBAP1/TBAP/TTSL | | | | | | | | | | | | | | | |
| rw | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TBAP1/TBAP/TTSL | | | | | | | | | | | | | | | |
| rw | | | | | | | | | | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | TBAP1 | 发送缓冲区 1 地址指针/发送帧时间戳低位<br>这些位有两种不同的功能：这些位向 DMA 指示数据在存储器中的位置，而当所有数据传输完 毕后， DMA 可利用这些位传回时间戳数据。 |
| | TBAP | 当软件为 DMA 提供此描述符（TDES0 中的 OWN 位置 1）时，这些位将指示缓冲区 1 的物理地址。缓冲区地址对齐方式不限。 |
| | TTSL | 在将 TDES0 中的 OWN 位清零前，DMA 将使用为相应发送帧所捕获的时间戳的 32 个最低有效位更新该字段（覆盖 TBAP1 的值）。只有在激活该帧的时间戳功能（请参见 TDES0 的位 25 TTSE）并且描述符中的末段控制位 (LS) 置 1 时，该字段才包含时间戳。 |

TDES3：发送描述符字 3

TDES3 包含描述符中第二个缓冲区或下一个描述符的地址指针， 或者包含时间戳数据。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TBAP2/TBAP2/TTSH | | | | | | | | | | | | | | | |
| rw | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TBAP2/TBAP2/TTSH | | | | | | | | | | | | | | | |
| rw | | | | | | | | | | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | TBAP2/TBAP2/TTSH | **TBAP2：**发送缓冲区 2 地址指针（下一个描述符地址）/ 发送帧时间戳高位<br>这些位有两种不同的功能：这些位向 DMA 指示数据在存储器中的位置，而当所有数据传输完毕后，DMA 可利用这些位传回时间戳数据。<br>**TBAP2：**当软件为 DMA 提供此描述符（TDES0 中的 OWN 位置 1）并且使用描述符环结构 时，这些位将指示缓冲区 2 的物理地址。如果链接第二个地址位 (TDES1 [24]) 置 1，则该地址包含下一个描述符所在物理寄存器的指针。只有在 TDES1 [24] 位置 1 时，缓冲区地址指针才必须与总线宽度相符。(将在内部忽略 LSB)<br>**TTSH：**在将 TDES0 中的 OWN 位清零前，DMA 将使用为相应发送帧所捕获的时间戳的 32 个最高有效位更新该字段（覆盖 TBAP2 的值）。只有在激活该帧的时间戳功能（请参见 TDES0 的位 25 TTSE）并且描述符中的末段控制位 (LS) 置 1 时，该字段才包含时间戳。 |

### 增强 Tx DMA 描述符

如果时间戳已激活（TSE=1，ETH_PTPTSCR 位 0）或 IPv4 校验和减荷已激活（IPCO=1，ETH_MACCR 位 10），则必须使用增强的描述符（通过 ETHDMABMR 位 7 EDFE=1 使能）。

增强的描述符由 8 个 32 位字组成，即描述符正常大小的两倍。TDES0、TDES1、TDES2 和 TDES3 的定义与常规发送描述符相同（请参见常规 Tx DMA 描述符）。TDES6 和 TDES7 包含时间戳。 TDES4、TDES5、TDES6 和 TDES7 的定义如下。

选择增强的描述符模式时，需要利用软件为每个描述符分配 32 个字节（8 个双字）的内存。如果未使用时间戳或 IPv4 校验和减荷，则可以禁止增强的描述符格式，而软件可使用默认大小为 16 字节的常规描述符。

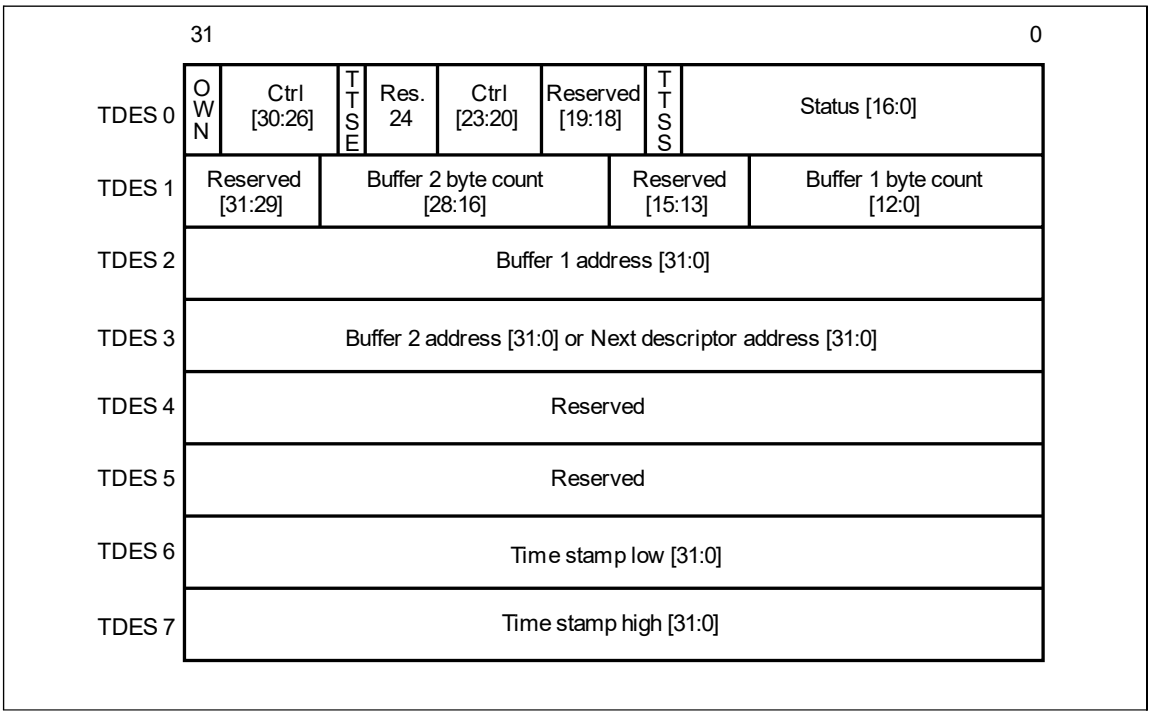



图 25-30 增强的发送描述符

TDES4：发送描述符字 4

保留。

TDES5：发送描述符字 5

保留。

TDES6：发送描述符字 6

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | TTSL | 发送帧时间戳低位<br>DMA 使用为相应发送帧所捕获的时间戳的 32 个最低有效位更新该字段。只有在描述符的末段控制位 (LS) 置 1 时，该字段才包含时间戳。 |

TDES7：发送描述符字 7

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | TTSH | 发送帧时间戳高位<br>DMA 使用为相应发送帧所捕获的时间戳的 32 个最高有效位更新该字段。只有在描述符的末段控制位 (LS) 置 1 时，该字段才包含时间戳。 |

### 25.6.8. Rx DMA 配置

下图所示为接收 DMA 引擎的接收序列，具体说明如下：

- CPU 设置接收描述符 (RDES0-RDES3) 并将 OWN 位 (RDES0[31]) 置 1。
- SR 位（ETH_DMAOMR 寄存器 [1]）置 1 后，DMA 即进入运行状态。在运行状态下，DMA 轮询接收描述符列表，尝试获取空闲描述符。如果获得的描述符不空闲（由 CPU 所拥有），则 DMA 进入挂起状态并跳转到第 9 步。
- DMA 将所获取描述符中的接收数据缓冲区地址解码。
- 处理传入帧并将其放入所获取描述符的数据缓冲区。
- 当缓冲区已满或帧传输完成时，接收引擎将获取下一个描述符。
- 如果当前的帧传输完成， DMA 将继续执行第 7 步。如果 DMA 未拥有下一个获取的描述符且帧传输尚未完成（尚未传输 EOF），则 DMA 会将 RDES0 中的描述符错误位置 1（除非禁止刷新）。DMA 关闭当前描述符（将 OWN 位清零）并通过在 RDES1 值中将末段位 (LS) 清零来将其标记为中间描述符（如果禁止刷新，则将其标记为最后一个描述符），随后继续执行第 8 步。如果 DMA 已拥有下一个描述符，但当前的帧传输尚未完成，则 DMA 将关闭当前描述符作为中间值并返回第 4 步。
- 如果已使能 IEEE 1588 时间戳功能， DMA 会将时间戳（如果可用）写入当前描述符的 RDES2 和 RDES3。 DMA 随后获取所接收帧的状态并将该状态字写入当前描述符的 RDES0，同时 OWN 位清零且末段位置 1。
- 接收引擎检查最新描述符的 OWN 位。如果 CPU 拥有该描述符（OWN 位为 0），则接收缓冲区不

可用位（ETH_DMASR 寄存器 [7]）置 1 且 DMA 接收引擎进入挂起状态（第 9 步）。如果 DMA 拥有该描述符，引擎将返回第 4 步并等待下一个帧。

- 在接收引擎进入挂起状态前，将从接收 FIFO 中刷新部分帧（可使用 ETH_DMAOMR 寄存器的位 24 控制刷新）。
- 当收到接收轮询要求命令或者可以从接收 FIFO 获得下一个帧的起点时，接收 DMA 将退出挂起状态。引擎继续执行第 2 步并重新获取下一个描述符。

直到完成时间戳回写并准备将状态回写到描述符时，DMA 才会确认接收状态。如果软件已通过 CSR 使能时间戳功能，则当无法获得帧的有效时间戳值时（例如，在写入时间戳前接收 FIFO 已满），DMA 会向 RDES2 和 RDES3 的所有位写入 1。否则（即，未使能时间戳功能）， RDES2 和 RDES3 保持不变。

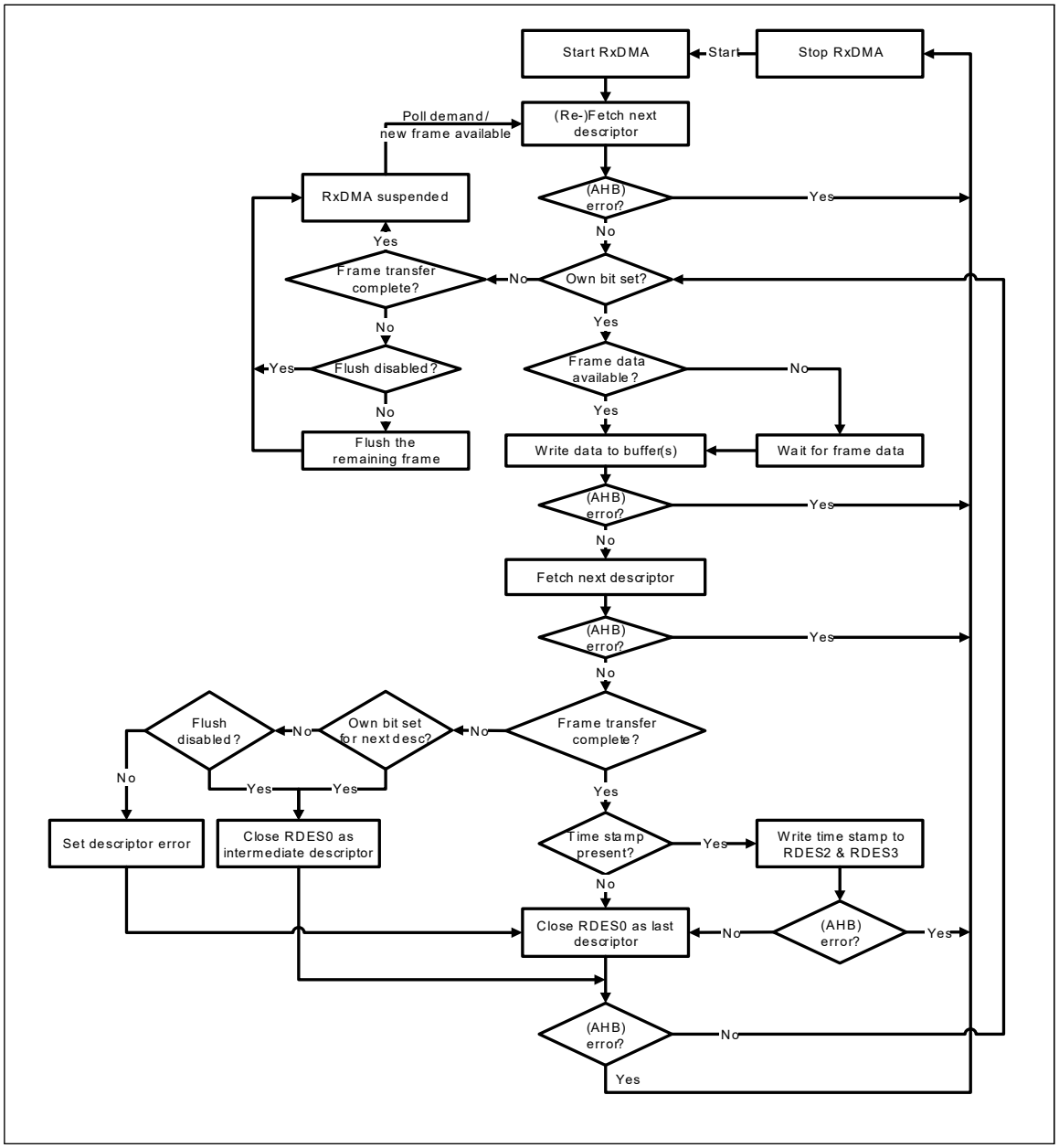



图 25-31 DMA 接收操作

**接收描述符获取**

接收引擎始终会尝试为即将传入的帧获取一个额外的描述符。只要满足以下任一条件，即尝试获取描述符：

- DMA 进入运行状态后，接收启动/停止位（ ETH_DMAOMR 寄存器 [1]）立即置 1。
- 在当前传输的帧结束前，当前描述符的数据缓冲区已满。
- 控制器已完成数据帧接收，但当前的接收描述符尚未关闭。
- 接收过程因缓冲区由 CPU 所拥有 (RDES0[31] = 0) 而挂起，并且接收到新帧。
- 已发出接收轮询要求命令。

**接收帧处理**

只有当帧通过地址过滤且其大小大于或等于为接收 FIFO 所设置的可配置阈值字节数时，或者在存储转发模式下将整个帧写入 FIFO 时， MAC 才会将接收的帧传输到存储器。如果帧未通过地址过滤，将自行进入 MAC 块中（除非接收所有位 ETH_MACFFR [31]置 1）。不足 64 字节的帧由于会发生冲突或过早结束，因而可从接收 FIFO 中清除。接收 64（可配置阈值）个字节后， DMA 块开始将帧数据传输到当前描述符所指定的接收缓冲区中。 DMA AHB 接口准备好接收数据传输后，如果 DMA 当前未从存储器获取发送数据，则将第一个描述符位 (RDES0[9]) 置 1，以分隔该帧。数据缓冲区已满或者帧的末段已传输到接收缓冲区时，OWN (RDES0[31]) 位将复位为 0，此时将释放描述符。如果帧包含在单个描述符中，则最后一个描述符位 (RDES0[8]) 和第一个描述符位 (RDES0[9]) 均置 1。DMA 获取下一个描述符，将最后一个描述符位 (RDES0[8]) 置 1，并释放前一个帧描述符中的 RDES0 状态位。然后， DMA 将接收中断位（ ETH_DMASR 寄存器 [6]）置 1。这一过程将重复执行，直至 DMA 遇到被标记为由 CPU 所有的描述符。如果出现这种情况，接收过程将接收缓冲区不可用位（ETH_DMASR 寄存器 [7]）置 1，随后进入挂起状态。但其在接收列表中的位置仍然保留。

**接收过程挂起**

如果在接收过程处于挂起状态时有新的接收帧到达，则 DMA 将重新获取 MCU 存储器中的当前描述符。如果该描述符现在由 DMA 所拥有，接收过程将重新进入运行状态并开始接收帧。如果该描述符仍由主机所拥有，则默认情况下，DMA 将丢弃 Rx FIFO 顶部的当前帧并将丢失帧计数器递增。如果 Rx FIFO 中存储了多个帧，将重复执行该过程。将 DMA 工作模式寄存器的位 24 (DFRF) 置 1 后，可避免丢弃或刷新 Rx FIFO 顶部的帧。在这种情况下，接收过程将接收缓冲区不可用状态位置 1 并返回到挂起状态。

**常规 Rx DMA 描述符**

常规接收描述符结构由四个 32 位字（16 字节）组成，如下图所示。下文将介绍 RDES0、RDES1、RDES2 和 RDES3 的各个位。

请注意，如果已激活时间戳功能（ETH_PTPTSCR 位 0 TSE=1）或 IPv4 校验和减荷（ETH_MACCR 位 10 IPCO=1），则必须使用增强的描述符。

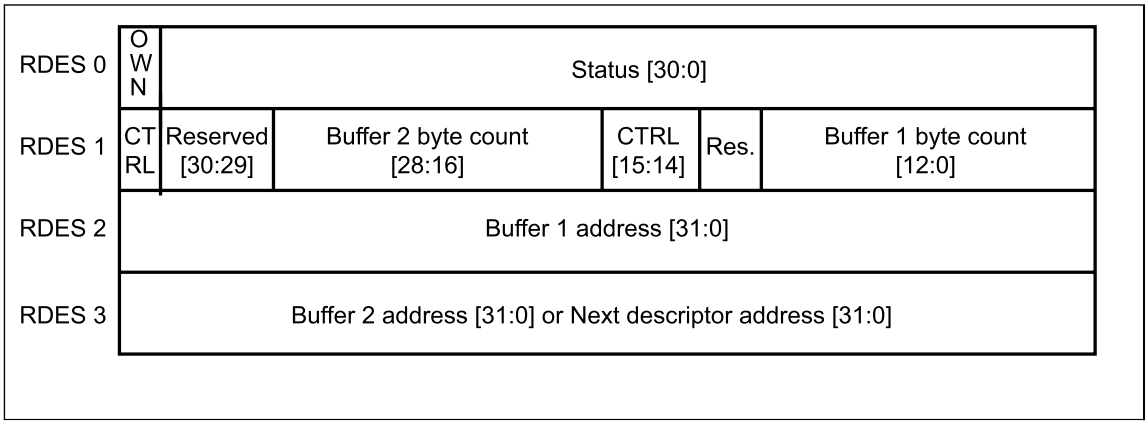



图 25-32 常规 Rx DMA 描述符结构

- RDES0：接收描述符字 0
- REDS0 包含接收帧状态、帧长度和描述符所有关系信息。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ES | DE | SAF | LE | OE | VLAN | FS | LS | IPHCE/TSV | LCO | FT | RWT | RE | DE | CE | PCE/ESA |
| rw | | | | | | | | | | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31 | OWN | 所有关系位<br>该位置 1 时，指示描述符由 MAC 子系统的 DMA 所拥有。该位复位时，指示描述符由主机所 拥有。DMA 在帧接收完成或此描述符的关联缓冲区已满时将该位清零。 |
| 位 30 | AFM | 目标地址过滤失败<br>该位置 1 时，指示帧未能通过 MAC 内核中的 DA 过滤。 |
| 位 29:16 | FL | 帧长度<br>这些位指示传输到主机存储器中的接收帧字节长度（含 CRC）。只有在最后一个描述符位 (RDES0[8]) 置 1 且描述符错误位 (RDES0[14]) 复位时，该字段才有效。<br>在最后一个描述符位 (RDES0[8]) 置 1 时，该字段有效。当最后一个描述符位和错误汇总位均未置 1 时，该字段指示已为当前帧传输的累计字节数。 |
| 位 15 | ES | 错误汇总<br>指示以下位的逻辑或运算结果：<br>RDES0[1]：CRC 错误 (CRC error)<br>RDES0[3]：接收错误 (Receive error)<br>RDES0[4]：看门狗超时 (Watchdog timeout)<br>RDES0[6]：延迟冲突 (Late collision)<br>RDES0[7]：巨帧 (Giant frame)（当 RDES0[7] 指示 IPV4 报头校验和错误时，该位不适用）<br>RDES0[11]：上溢错误 (Overflow error)<br>RDES0[14]：描述符错误 (Descriptor error)。 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 只有最后一个描述符 (RDES0[8]) 置 1 时，该字段才有效。 |
| 位 14 | DE | 描述符错误<br>该位置 1 时，指示某个帧因超过当前描述符缓冲区的大小而被截断以及 DMA 未拥有下一个描述符。帧被截断。只有最后一个描述符 (RDES0[8]) 置 1 时，该字段才有效。 |
| 位 13 | SAF | 源地址过滤失败<br>该位置 1 时，指示帧的 SA 字段未能通过 MAC 内核中的 SA 过滤。 |
| 位 12 | LE | 长度错误<br>该位置 1 时，指示接收帧的实际长度与长度/ 类型字段的值不符。该字段仅在帧类型位 (RDES0[5]) 复位后有效。 |
| 位 11 | OE | 上溢错误<br>该位置 1 时，指示接收帧因缓冲区上溢而损坏。 |
| 位 10 | VLAN | VLAN 标记 (VLAN tag)<br>该位置 1 时，指示描述符所指向的帧是由 MAC 内核标记的 VLAN 帧。 |
| 位 9 | FS | 第一个描述符<br>该位置 1 时，指示此描述符包含帧的第一个缓冲区。如果第一个缓冲区的大小为 0, 则第二个缓冲区将包含帧的帧头。如果第二个缓冲区的大小为 0, 则下一个描述符将包含帧的帧头。 |
| 位 8 | LS | 最后一个描述符<br>该位置 1 时，指示此描述符指向的缓冲区为帧的最后一个缓冲区。 |
| 位 7 | IPHCE/TSV | IPv 报头校验和错误/时间戳有效<br>IPHCE 位置 1 时，指示 IPv4 或 IPv6 报头中存在错误。导致此错误的原因可能是: 以太网类 型字段与 IP 报头版本字段值不一致, 或者与 IPv4 中报头的校验和不匹配，或者以太网帧缺少所需的 IP 报头字节数。<br>使能增强的描述符格式（EDFE=1，ETH_DMABMR 的位 7）后，该位具有 TSV 功能（否则其为 IPHCE）。TSV 位置 1 时，表示将在描述符字 6 (RDES6) 和 7 (RDES7) 中写入时间戳快照。只有最后一个描述符位 (RDES0[8]) 置 1 时， TSV 才有效。 |
| 位 6 | LCO | 延迟冲突<br>该位置 1 时，表示在以半双工模式接收帧时发生了延迟冲突。 |
| 位 5 | FT | 帧类型<br>该位置 1 时，表示接收帧为以太网类型帧（LT 字段大于或等于 0x0600）。该位复位时，表示接收的帧为 IEEE802.3 帧。该位不适用于少于 14 个字节的矮帧。使用正常描述符格式 (ETH_DMABMR EDFE=0) 时， FT 可具有特殊的含义。 |
| 位 4 | RWT | 接收看门狗超时<br>该位置 1 时，表示接收看门狗计时器在接收当前帧时超时，且当前帧在看门狗超时后被截断了。 |
| 位 3 | RE | 接收错误<br>该位置 1 时, 表示在帧接收期间, 当发出 RX_DV 信号时, 会发出 RX_ERR 信号。 |
| 位 2 | DE | Dribble 位错误<br>该位置 1 时，表示接收的帧具有非整数倍数的字节（奇数半字节）。该位仅在 MII 模式下有效。 |
| 位 1 | CE | CRC 错误<br>该位置 1 时，表示接收的帧发生循环冗余校验 (CRC) 错误。只有最后一个描述符 (RDES0[8]) 置 1 时，该字段才有效。 |
| 位 0 | PCE/ESA | 有效负载校验和错误/扩展状态可用 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 该位置 1 时，表示由内核计算的 TCP、UDP 或 ICMP 校验和与接收到的封装 TCP、UDP 或 ICMP 段校验和字段不匹配。当接收到的有效负载字节数与接收到的以太网帧中封装 IPv4 或 IPv6 数据图的长度字段中所指示的值不匹配时，该位也会被置 1。该位可具有特殊的含义。<br>使能增强的描述符格式（EDFE=1，ETH_DMABMR 的位 7）后，该位具有 ESA 功能（否则 其为 PCE）。 ESA 置 1 时，表示描述符字 4(RDES4) 中存在扩展状态。只有最后一个描述 符位 (RDES0[8]) 置 1 时， ESA 才有效。 |

位 5 、7 和 0 用于指示表 xxx 中讨论的情况。

表 25-9 接收描述符 0 – 位 7、5 和 0 的编码（仅正常描述符格式，EDFE=0）

| 位 5: 帧 类 型 (frame type) | 位 7: IPC 校验和错误 (IPC checksum error) | 位 0: 有效负载校验和错误 (payload checksum error) | 帧状态 |
| --- | --- | --- | --- |
| 0 | 0 | 0 | IEEE 802.3 类型帧（长度字段的值小于 0x0600。） |
| 1 | 0 | 0 | IPv4/IPv6 类型帧，未检测到校验和错误 |
| 1 | 0 | 1 | IPv4/IPv6 类型帧, 检测到有效负载校验和错误(请参见 PCE 的相关说明) |
| 1 | 1 | 0 | IPv4/IPv6 类型帧, 检测到 IP 报头校验和错误（请参见 IPC CE 的相关说明） |
| 1 | 1 | 1 | IPv4/IPv6 类型帧，同时检测到 IP 报头和有效负载校验和错误 |
| 0 | 0 | 1 | IPv4/IPv6 类型帧, 无 IP 报头校验和错误, 有效负载检查因有效负载不受支持而被绕过 |
| 0 | 1 | 1 | 既不是 IPv4 也不是 IPv6 的类型帧（校验和减荷引擎完全绕 过校验和。） |
| 0 | 1 | 0 | 保留 |

RDES1：接收描述符字 1

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| DIC | Reserved | | RBS2 | | | | | | | | | | | | |
| rw | | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RER | RCH | Reserved | RBS | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31 | DIC | 完成时禁止中断<br>置 1 时，该位会阻止状态寄存器的 RS 位 (CSR5[6]) 置 1，使得接收的帧在由该描述符所指示的缓冲区中结束, 从而会因该帧的 RS 而禁止引发主机的中断。 |
| 位 30:29 | 保留 | 必须保持复位值 |
| 位 28:16 | RBS2 | 接收缓冲区 2 大小<br>这些位以字节为单位指示第二个数据缓冲区的大小。即使 RDES3（缓冲 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 区 2 地址指针）的值未与总线宽度对齐，缓冲区大小也必须为 4、8 或 16 的倍数，具体取决于总线宽度（分别为 32、64 或 128）。如果缓冲区大小不是 4、8 或 16 的倍数，则不会定义产生的行为。如果将 RDES1[14] 置 1，则该字段无效。 |
| 位 15 | RER | 接收环结束<br>该位置 1 时，表示描述符列表已到达其最后一个描述符。DMA 会返回描述符列表的基址，并形成一个描述符环。 |
| 位 14 | RCH | 链接的第二个地址<br>该位置 1 时，表示描述符中的第二个地址是下一个描述符地址，而非第二个缓冲区地址。该位置 1 时， RBS2 (RDES1[28:16]) 为“无关”值。RDES1[15] 优先级高于 RDES1[14]。 |
| 位 13 | 保留 | 必须保持复位值 |
| 位 12:0 | RBS | 接收缓冲区 1 大小<br>以字节为单位表示第一个数据缓冲区的大小。即使 RDES2（缓冲区 1 地址指针）的值未对齐，缓冲区大小也必须为 4、8 或 16 的倍数，具体取决于总线宽度（32、64 或 128）。如果 缓冲区大小不是 4、8 或 16 的倍数，则不会定义产生的行为。如果该字段为 0，则 DMA 会忽 略该缓冲区并使用缓冲区 2 或下一个描述符，具体取决于 RCH（位 14）的值。 |

RDES2：接收描述符字 2

RDES2 包含描述符中第一个数据缓冲区的地址指针，或者包含时间戳数据。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RBAP1/RTSL | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RBAP1/RTSL | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:0 | RBAP1/RTSL | 接收缓冲区 1 地址指针/接收帧时间戳低电平<br>这些位有两种不同的功能：应用程序使用它们向 DMA 指示数据在存储器中的存储位置，而当所有数据均传输完成后，DMA 可以使用这些位来传回时间戳数据。<br>**RBAP1：**当软件为 DMA 提供此描述符时（当 RDES0 中的 OWN 位置 1 时），这些位会指示缓冲区 1 的物理地址。在缓冲区地址对齐方面没有任何限制，但以下情况除外：当使用 RDES2 值存储帧起点时，DMA 会使用所配置的值生成其地址。请注意，在传输帧开始期间，DMA 将在 RDES2[3/2/1:0] 位为 0 的情况下执行写操作，但帧数据会根据实际的缓冲区地址指针进行移位。当地址指针指向存储有帧中间部分或最后部分的缓冲区时，DMA 将忽略 RDES2[3/2/1:0]（与总线宽度 128/64/32 相对应）。<br>**RTSL：**在将 RDES0 中的 OWN 位清零之前，DMA 会使用为对应接收帧捕获的时间戳的 32 个最低有效位更新该字段（覆盖 RBAP1 的值）。只有此帧的时间戳已激活且描述符中的末段控制位 (LS) 已置 1 时，该字段才具有时间戳。 |

RDES3：接收描述符字 3

RDES3 包含描述符中第二个数据缓冲区或下一个描述符的地址指针，或者包含时间戳数据。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RBAP2/RTSH | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RBAP2/RTSH | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | RBAP2/RTSH | 接收缓冲区 2 地址指针（下一个描述符地址）/ 接收帧时间戳高电平<br>这些位有两种不同的功能：应用程序使用它们向 DMA 指示数据在存储器中的存储位置，而当所有数据均传输完成后，DMA 可以使用这些位来传回时间戳数据。<br>**RBAP1：**当软件为 DMA 提供此描述符时（当 RDES0 中的 OWN 位置 1 时），这些位会指示当使用描述符环结构时缓冲区 2 的物理地址。如果链接的第二个地址 (RDES1 [24]) 位置 1，则该地址包含指向下一个描述符所在物理寄存器的指针。如果 RDES1 [24] 置 1，则缓冲区（下一个描述符）地址指针必须是总线宽度对齐形式（RDES3[3、2 或 1:0] = 0，与总线宽度 128、64 或 32 相对应，LSB 被内部忽略。）然而，当 RDES1 [24] 复位时，RDES3 值无任何限制，但以下情况除外：当使用 RDES3 值存储帧起点时，DMA 会使用所配置的值生成 其缓冲区地址。当地址指针指向存储有帧中间部分或最后部分的缓冲区时，DMA 将忽略 RDES3[3、2 或 1:0]（与总线宽度 128、64 或 32 相对应）。<br>**RTSH：**在将 RDES0 中的 OWN 位清零之前，DMA 会使用为对应接收帧捕获的时间戳的 32 个最高有效位更新该字段（覆盖 RBAP2 的值）。只有时间戳已激活且描述符中的末段控制位 (LS) 已置 1 时，该字段才具有时间戳。 |

**带有 IEEE1588 时间戳的增强的 Rx DMA 描述符**

如果时间戳已激活（TSE=1， ETH_PTPTSCR 位 0）或 IPv4 校验和减荷已激活（IPCO=1，ETH_MACCR 位 10），则必须使用增强的描述符（通过 ETHDMABMR 位 7 EDFE=1 使能）。

增强的描述符由 8 个 32 位字组成，即描述符正常大小的两倍。RDES0、RDES1、RDES2 和 RDES3 的定义与常规接收描述符相同（请参见常规 Rx DMA 描述符）。RDES4 包含扩展状态，而 RDES6 和 RDES7 包含时间戳。RDES4、RDES5、RDES6 和 RDES7 的定义如下。

选择增强的描述符模式时，需要利用软件为每个描述符分配 32 个字节（8 个双字）内存。如果未使用时间戳或 IPv4 校验和减荷，则可以禁止增强的描述符格式，而软件可使用默认大小为 16 字节的常规描述符。

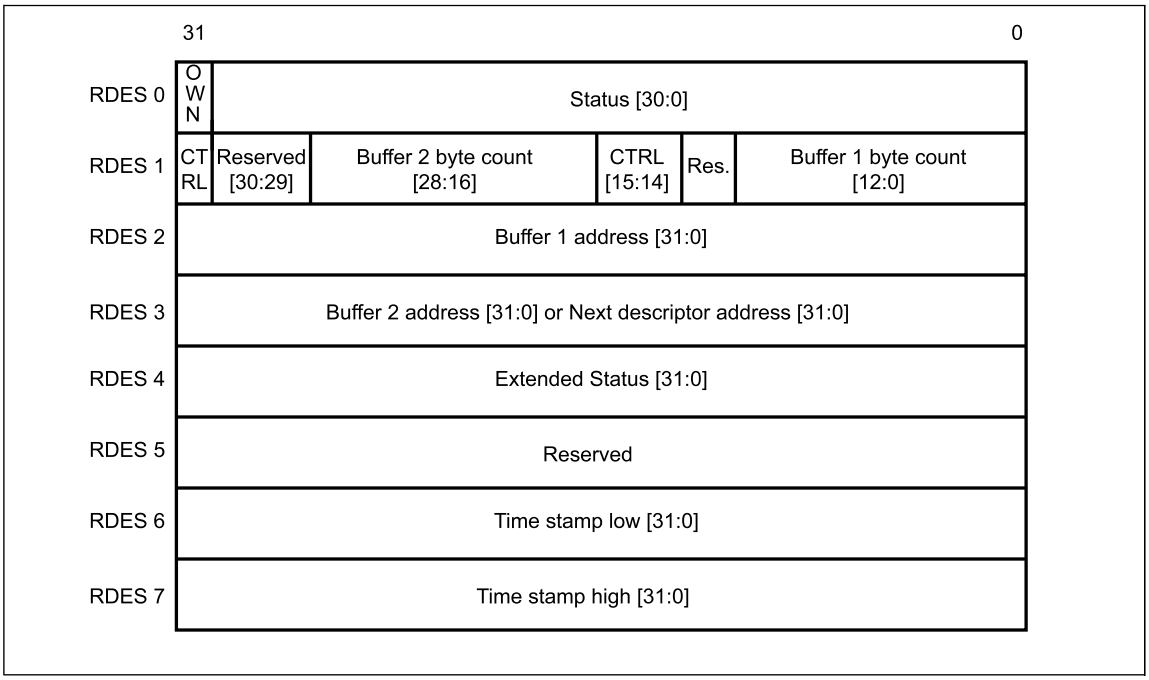



图 25-33 使能带有 IEEE1588 时间戳的增强的接收描述符字段格式

- RDES4：接收描述符字 4。
- 仅当存在与 IPv4 校验和或时间戳（由 RDES0 中的位 0 加以指示）相关的状态时，如下所示的扩展状态才有效。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | PV | PFT | PMT | | | | IPV6PR | IPV4PR | IPCB | IPPE | IPHE | IPPT | | |
| | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:14 | 保留 | 必须保持复位值 |
| 位 13 | PV | PTP 版本<br>置 1 时，表示接收的 PTP 消息使用 IEEE 1588 的版本 2 格式。清零时，将使用版本 1 格式。仅当消息类型为非零时才有效。 |
| 位 12 | PFT | PTP 帧类型<br>该位置 1 时，表示 PTP 消息直接通过以太网发送。该位清零且消息类型为非零时，表示 PTP 消息通过 UDP-IPv4 或 UDP-IPv6 发送。可通过位 6 或位 7 获取有关 IPv4 或 IPv6 的信息。 |
| 位 11:8 | PMT | PTP 消息类型<br>将对这些位进行编码，以给出所接收消息的类型。<br>0000：未接收到任何 PTP 消息<br>0001：SYNC（所有时钟类型）<br>0010：Follow_Up（所有时钟类型）<br>0011：Delay_Req（所有时钟类型） |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 0100：Delay_Resp（所有时钟类型）<br>0101：Pdelay_Req（在点对点透明时钟中）或 Announce（在普通时钟或边界时钟中）<br>0110：Pdelay_Resp（在点对点透明时钟中）或 Management（在普通时钟或边界时钟中）<br>0111：Pdelay_Resp_Follow_Up（在点对点透明时钟中）或 Signaling（在普通时钟或边界时钟中）<br>1xxx - 保留 |
| 位 7 | IPV6PR | 接收到 IPv6 数据包<br>该位置 1 时，表示接收到的数据包为 IPv6 数据包。 |
| 位 6 | IPV4PR | 接收到 IPv4 数据包<br>该位置 1 时，表示接收到的数据包为 IPv4 数据包。 |
| 位 5 | IPCB | 绕过 IP 校验和<br>该位置 1 时，表示绕过校验和减荷引擎。 |
| 位 4 | IPPE | IP 有效负载错误<br>该位置 1 时，表示由内核计算的 16 位 IP 有效负载校验和（即 TCP、UDP 或 ICMP 校验和）与接收段中对应的校验和字段不匹配。当 TCP、UDP 或 ICMP 段长度与 IP 报头字段中的有 效负载长度值不匹配时，它同样会置 1。 |
| 位 3 | IPHE | IP 报头错误<br>该位置 1 时，表示由内核计算的 16 位 IPv4 报头校验和与接收的校验和字节不匹配，或 IP 数据报版本与以太网类型值不一致。 |
| 位 2:0 | IPPT | IP 有效负载字节<br>如果 IPv4 校验和减荷被激活（IPCO=1 ，ETH_MACCR 位 10），则这些位表示 IP 数据报中封装的有效负载类型。当 IP 报头出错或存在分段的 IP 时，这些位为“00”。<br>000：未知，或未处理 IP 有效负载<br>001：UDP<br>010：TCP<br>011：ICMP<br>1xx：保留 |

RDES5：接收描述符字 5

保留。

RDES6：接收描述符字 6

下表介绍了当接收描述符关闭且使能时间戳时对 RDES6 而言具有不同含义的字段。

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:0 | RTSL | 接收帧时间戳低电平<br>DMA 使用为对应接收帧捕获的时间戳的 32 最低有效位更新该字段。<br>DMA 仅为最后描述符状态位 (RDES0[8]) 指示的接收帧的最后描述符更新该字段。当该字段和 RDES7 中的 RTSH 字段均显示 1 时，必须视为时间戳已损坏。 |

RDES7：接收描述符字 7

下表介绍了当接收描述符关闭且使能时间戳时对 RDES7 而言具有不同含义的字段。

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:0 | RTSH | 接收帧时间戳高电平<br>DMA 使用为对应接收帧捕获的时间戳的 32 最高有效位更新该字段。<br>DMA 仅为最后描述符状态位 (RDES0[8]) 指示的接收帧的最后描述符更新该字段。<br>当该字段和 RDES7 的 RTSL 字段都显示 1 时，必须将时间戳视为已损坏。 |

### 25.6.9. DMA 中断

中断可以作为各种事件的结果而生成。ETH_DMASR 寄存器包含可能引起中断的所有位。ETH_DMAIER 寄存器对可能引起中断的每一个事件都包含一个使能位。

如 ETH_DMASR 寄存器中所描述，存在两组中断：正常中断和异常中断。通过向对应位位置写入 1 来清除中断。当清除了组内所有使能的中断时，对应的汇总位亦被清零。如果 MAC 内核是引发中断的原因，则 ETH_DMASR 寄存器中的任何 TSTS 或 PMTS 位都将置为高电平。

中断不会排队积压，如果中断事件在驱动程序对其做出响应之前发生，则不会再生成中断。例如，接收中断位（ETH_DMASR 寄存器 [6]）指示一个或多个帧被传输到缓冲器。驱动程序必须扫描所有描述符，从 DMA 拥有的最后记录的位置到第一个位置。

对于同时发生的多个事件，一个中断只生成一次。驱动程序必须扫描 ETH_DMASR 寄存器，查找中断的原因。不会再次生成中断，除非在驱动程序已将 ETH_DMASR 寄存器中的适当位清零后发生新的中断事件。例如，控制器生成接收中断（ETH_DMASR 寄存器[6]）而驱动程序开始读取 ETH_DMASR 寄存器。然后，发生接收缓冲器不可用（ETH_DMASR 寄存器[7]）事件。驱动程序清除接收中断。即使是这时，也会由于活动或挂起的接收缓冲器不可用中断而生成一个新的中断。

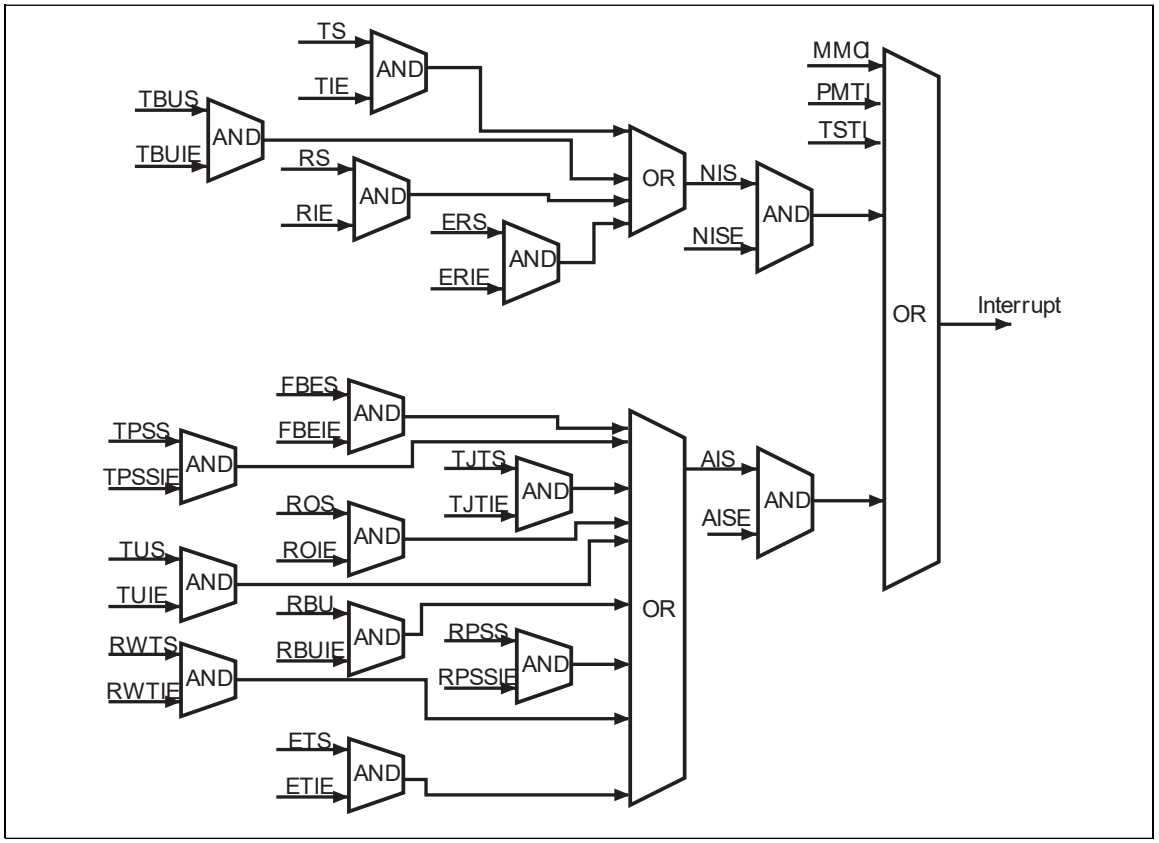



图 25-34 中断方案

## 25.7. 以太网中断

以太网控制器有两个中断向量：一个专用于正常以太网操作，另一个仅当映射到 EXTI 线路 19 时用于以太网唤醒事件（带有唤醒帧或魔术数据包检测）。

如 MAC 中断和 DMA 中断章节所列，第一个以太网向量为 MAC 和 DMA 生成的中断而保留。

第二个向量为发生唤醒事件时 PMT 生成的中断而保留。唤醒事件对 EXTI 线路 19 的映射使 MCU 退出低功耗模式并生成一个中断。

当映射到 EXTI 线路 19 的以太网唤醒事件发生、使能了 MAC PMT 中断并且还使能了带上升沿检测的 EXTI 线路 19 中断时，会生成这两个中断。

可使用看门狗定时器（请参见 ETH_DMARSWTR 寄存器）灵活控制 RS 位（ETH_DMASR 寄存器）。当此看门狗定时器使用非零值编程时，只要 RxDMA 完成将接收的帧传输到系统存储器并且未触发接收状态（因未在相应接收描述符 (RDES1[31]) 使能接收状态），看门狗定时器即会被激活。当定时器按照编程值到时时，如果使能了 ETH_DMAIER 寄存器中的相应 RIE，则 RS 位会置 1 并会引发中断。当由于为该描述符将其使能而将一帧传输到存储器并且 RS 置 1 时，会在到时前禁止定时器。

*注意：读取 PMT 控制和状态寄存器会自动将接收的唤醒帧和接收的魔术数据包 PMT 中断标志清零。但是，由于用于这些标志的寄存器位于 CLK_RX 域，因此在固件能发现此更新前可能有显著的延迟。当 RX 时钟很慢（在 10 Mbit 模式）和当 AHB 总线为高频时，该延迟会特别长。由于从 PMT 到 CPU 的中断请求基于 CLK_RX 域中的相同寄存器，所以即使在读取 PMT_CSR 之后，CPU 也可能错误地第二次调用中断例程。因此，可能需要固件轮询接收的唤醒帧和接收的魔术数据包位，并仅在发现它们都为'0'时退出中断服务程序。*

## 25.8. 以太网寄存器

外设寄存器可支持字节（ 8 位)、半字（ 16 位）或字（ 32 位）访问。

### 25.8.1. MAC 寄存器

#### 以太网 MAC 配置寄存器 (ETH_MACCR)

偏移地址： 0x0000

复位值： 0x0000 8000

MAC 配置寄存器是 MAC 的工作模式寄存器。它建立了接收和发送工作模式。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | CSTF | Reserved | WD | JD | Reserved | | IFG | | | CSD |
| | | | | | | rw | | rw | rw | | | | | | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | FES | ROD | LM | DM | IPCO | RD | Reserved | APCS | BL | | DC | TE | RE | Reserved | |
| | rw | rw | rw | rw | rw | rw | | rw | rw | | rw | rw | rw | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:26 | 保留 | 必须保持复位值 |
| 位 25 | CSTF | 类型帧的 CRC 去除<br>该位置 1 时，在将帧转发到应用之前，将去除并丢弃所有以太类型（类型字段大于 0x0600） 帧的最后 4 个字节 (FCS)。 |
| 位 24 | 保留 | 必须保持复位值 |
| 位 23 | WD | 禁止看门狗<br>当该位置 1 时，MAC 禁止接收器上的看门狗定时器并可接收多达 16384 字节的帧。<br>当该位复位时， MAC 允许接收的帧不超过 2048 字节并会截断超过此限制接收的任何字节。 |
| 位 22 | JD | 禁止 Jabber<br>当该位置 1 时，MAC 禁止发送器上的 jabber 定时器并可发送多达 16384 字节的帧。<br>当该位复位时，如果发送期间应用程序发送超过 2 048 字节的数据，MAC 会截断发送器。 |
| 位 21:20 | 保留 | 必须保持复位值 |
| 位 19:17 | IFG | 帧间间隙<br>这些位控制发送期间帧间的最小间隙。<br>000：96 位时间<br>001：88 位时间<br>010：80 位时间<br>.....<br>111：40 位时间<br>*注意： 在半双工模式下，最小 IFG 仅能配置为 64 位时间(IFG = 100)。不考虑更低的值。* |
| 位 16 | CSD | 禁止载波侦听<br>值设置得很高时，该位会使 MAC 发送器忽略半双工模式下帧发送期间的 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | MII CRS 信号。不会因在此发送期间载波丢失或无载波而生成错误。<br>当该位为低电平时，MAC 发送器会生成载波侦听错误，甚至中止发送。 |
| 位 15 | 保留 | 必须保持复位值 |
| 位 14 | FES | 快速以太网速度<br>指示快速以太网 (MII) 模式下的速度：<br>0：10 Mbit/s<br>1：100 Mbit/s |
| 位 13 | ROD | 禁止接收自身<br>当该位置 1 时，MAC 禁止在半双工模式下接收帧。<br>当该位复位时，MAC 接收发送时 PHY 提供的所有包。<br>如果 MAC 在全双工模式下工作，该位不适用。 |
| 位 12 | LM | 回送模式<br>当该位置 1 时，MAC 在 MII 回送模式下工作。回送正确工作需要 MII 接收时钟输入 (RX_CLK)，因为发送时钟不是内部回送的。 |
| 位 11 | DM | 双工模式<br>当该位置 1 时，MAC 在全双工模式下工作，此时它可以同时发送和接收。 |
| 位 10 | IPCO | IPv4 校验和减荷<br>置 1 时，该位使能接收的帧有效载荷的 TCP/UDP/ICMP 标头的 IPv4 校验和检查。当该位复位时，禁止接收器中的校验和减荷功能并始终将相应的 PCE 和 IP HCE 状态位清零。 |
| 位 9 | RD | 禁止重试<br>当该位置 1 时，MAC 仅尝试一次发送。当在 MII 模式下发生冲突时，MAC 会忽略当前帧发送并以发送帧状态下过度冲突错误报告帧中止。<br>当该位复位时，MAC 会尝试根据 BL 的设置进行重试。<br>*注意： 该位仅在半双工模式下适用。* |
| 位 8 | 保留 | 必须保持复位值 |
| 位 7 | APCS | 去除自动 pad/CRC<br>当该位置 1 时，MAC 仅在长度字段值小于或等于 1500 字节时去除传入帧上的 Pad/FCS 字段。传送给应用程序的所有接收的帧，如果长度字段大于或等于 1501 字节，将都不去除 Pad/FCS 字段。<br>当该位复位时，MAC 会不加修改地传送所有传入的帧。 |
| 位 6:5 | BL | 后退限制<br>后退限制决定发生冲突后重试期间 MAC 在重新安排一次发送之前等待的随机整数 (r) 个时间片延迟（对于 1000 Mbit/s 为 4096 位时间，对于 10/100 Mbit/s 为 512 位时间）。<br>*注意： 该位仅在半双工模式下适用。*<br>00: k = min (n, 10)<br>01: k = min (n, 8)<br>10: k = min (n, 4)<br>11: k = min (n, 1),<br>其中 n = 重新发送尝试的次数。随机整数 r 的取值范围为 $0 < r < 2^k$ |
| 位 4 | DC | 检查延迟<br>当该位置 1 时，便在 MAC 中使能了检查延迟功能。当发送状态机在 10/100-Mbit/s 模式下延迟超过 24288 位时间时，MAC 将指示帧中止状态，同时在发送帧状态中将过度延迟错误位 置 1。当发送器准备好发送但因 MII 上有有效 CRS（载波侦听）信号而被阻止时延迟开始。延迟时间是非累积性的。如果发送器延迟 10 000 位时间，然后发送、冲突、 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 后退，然后在 完成后退后不得不再次延迟，则延迟定时器复位为 0 并重新开始。<br>当该位复位时，禁止延迟检查功能，MAC 发生延迟，直到 CRS 信号变为无效信号。该位仅在半双工模式下适用。 |
| 位 3 | TE | 发送器使能<br>当该位置 1 时，针对 MII 上的发送使能 MAC 的发送状态机。当该位复位时，在当前帧的发送完成后禁止 MAC 发送状态机，并且不再发送任何帧。 |
| 位 2 | RE | 接收器使能<br>当该位置 1 时，使能 MAC 的接收器状态机，以便从 MII 接收帧。当该位复位时，在当前帧接收完成后禁止 MAC 接收状态机，并且不再从 MII 接收任何帧。 |
| 位 1:0 | 保留 | 必须保持复位值 |

### 以太网 MAC 帧过滤寄存器 (ETH_MACFFR)

偏移地址： 0x0004

复位值： 0x0000 0000

MAC 帧过滤寄存器包含用于接收帧的过滤控件。该寄存器的某些控件转到 MAC 的地址检查块，以执行第一级地址过滤。第二级过滤基于其它控件（例如传送不良帧和传送控制帧）对传入帧过滤。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Reserved | | | | | HPF | SAF | SAIF | PCF | | BFD | PAM | DAIF | HM | HU | PM |
| | | | | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31 | RA | 接收所有<br>当该位置 1 时，MAC 接收器将所有接收的帧传送到应用程序，不管它们是否已通过地址过滤。SA/DA 过滤的结果在接收状态字的相应位更新（通过或失败）。当该位复位时，MAC 接收器传送给应用程序的只是通过了 SA/DA 地址过滤的那些帧。 |
| 位 30:11 | 保留 | 必须保持复位值。 |
| 位 10 | HPF | 散列或完美过滤器<br>当该位置 1 时，如果 HM 或 HU 位置 1，则地址过滤器传送与完美过滤或散列过滤匹配的帧。<br>当该位清零时，如果 HU 或 HM 位置 1，则仅传送与散列过滤器匹配的帧。 |
| 位 9 | SAF | 源地址过滤器<br>MAC 内核将接收的帧的 SA 字段与使能的 SA 寄存器中编程的值进行比较。如果比较结果匹配，则将 RxStatus 字中的 SAMatch 位设为高电平。当该位设为高电平且 SA 过滤失败时，MAC 放弃该帧。<br>当该位复位时，MAC 内核将接收的帧转发给应用程序。它还根据 SA 地址比较结果转发 RxStatus 中更新的 SA 匹配位。 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 8 | SAIF | 源地址反向过滤<br>当该位置 1 时，地址检查块在反向过滤模式下进行 SA 地址比较。SA 与 SA 寄存器匹配的帧被标记为 SA 地址过滤失败。<br>当该位复位时，SA 与 SA 寄存器不匹配的帧被标记为 SA 地址过滤失败。 |
| 位 7:6 | PCF | 传送控制帧<br>这些位控制所有控制帧（包括单播和多播暂停帧）的转发。请注意暂停控制帧的处理只取决于流控制寄存器[2] 中的 RFCE。<br>00：MAC 阻止所有控制帧到达应用程序<br>01：MAC 将除了暂停控制帧以外的所有控制帧转发到应用程序<br>10：即使所有控制帧通过地址过滤失败，MAC 仍将它们转发给应用程序<br>11：MAC 转发通过地址过滤的控制帧。<br>这些位控制所有控制帧（包括单播和多播暂停帧）的转发。请注意暂停控制帧的处理只取决于流控制寄存器[2] 中的 RFCE。<br>00 或 01：MAC 阻止所有控制帧到达应用程序<br>10：即使所有控制帧通过地址过滤失败，MAC 仍将它们转发给应用程序<br>11：MAC 转发通过地址过滤的控制帧。 |
| 位 5 | BFD | 禁止广播帧<br>当该位置 1 时，地址过滤器对所有传入的广播帧进行过滤。<br>当该位复位时，地址过滤器传送所有接收到的广播帧。 |
| 位 4 | PAM | 传送所有多播<br>置 1 时，该位指示传送接收到的带多播目标地址（目标地址字段的第一位是 1）的所有帧。<br>复位时，多播帧的过滤取决于 HM 位。 |
| 位 3 | DAIF | 目标地址反向过滤<br>当该位置 1 时，地址检查块在反向过滤模式下对单播和多播帧都进行 DA 地址比较。<br>复位时，正常过滤帧。 |
| 位 2 | HM | 散列多播<br>置 1 时，MAC 根据散列表对接收到的多播帧执行目标地址过滤。<br>复位时，MAC 对多播帧执行完美目标地址过滤，即将 DA 字段与 DA 寄存器中编程的值进行比较。 |
| 位 1 | HU | 散列单播<br>置 1 时，MAC 根据散列表对单播帧执行目标地址过滤。<br>复位时，MAC 对单播帧执行完美目标地址过滤，即将 DA 字段与 DA 寄存器中编程的值进行比较。 |
| 位 0 | PM | 混合模式<br>当该位置 1 时，不论其目标或源地址为何，地址过滤器都传送所有传入的帧。PM 置 1 时，始终将接收状态字中 SA/DA 过滤失败状态位清零。 |

**以太网 MAC 散列表高位寄存器 (ETH_MACHTHR)**

偏移地址： 0x0008

复位值： 0x0000 0000

64 位散列表用于组地址过滤。对于散列过滤，传入帧中目标地址的内容通过 CRC 逻辑传送，CRC 寄存器中的高 6 位用于对散列表内容进行索引。CRC 是使用下列多项式编码的 32 位值：

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

最高有效位决定要使用的寄存器（散列表高位/散列表低位），其它 5 位决定寄存器中的位。散列值 0b0 0000 选取所选寄存器中的位 0，值 0b1 1111 选取所选寄存器中的位 31。

例如，如果传入帧的 DA 接收为 0x1F52 419C B6AF（0x1F 是 MII 接口上接收的第一个字节），则内部计算出的 6 位散列值为 0x2C 且为了过滤要检查 HTH 寄存器位[12]。如果传入帧的 DA 接收为 0xA00A 9800 0045，则计算出的 6 位散列值为 0x07 且为了过滤要检查 HTL 寄存器位[7]。

如果寄存器中对应的位值为 1，则接受该帧。否则拒绝该帧。如果 ETH_MACFFR 寄存器中的 PAM（传送所有多播）位置 1，则不论多播散列值为何都接受所有多播帧。

散列表高位寄存器包含多播散列表的高 32 位。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | HTH | 散列表高位<br>该字段包含散列表的高 32 位。 |

### 以太网 MAC 散列表低位寄存器 (ETH_MACHTLR)

偏移地址： 0x000C

复位值： 0x0000 0000

散列表低位寄存器包含多播散列表的低 32 位。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | HTL | 散列表低位<br>该字段包含散列表的低 32 位。 |

### 以太网 MAC MII 地址寄存器 (ETH_MACMIIAR)

偏移地址： 0x0010

复位值： 0x0000 0000

MII 地址寄存器通过管理接口控制外部 PHY 的管理周期。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PA | | | | | MR | | | | | Reserved | CR | | | MW | MB |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | | rw | rw | rw | rw | rc_w1 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:16 | 保留 | 必须保持复位值。 |
| 位 15:11 | PA | PHY 地址<br>该字段指示正在访问 32 个可能的 PHY 器件中的哪一个。 |
| 位 10:6 | MR | MII 寄存器<br>这些位在所选 PHY 器件中选择需要的 MII 寄存器。 |
| 位 5 | 保留 | 必须保持复位值。 |
| 位 4:2 | CR | 时钟范围<br>CR 时钟范围选项可确定 HCLK 频率并用于决定 MDC 时钟频率:<br>选项 HCLK MDC 时钟<br>000 60-100 MHz HCLK/42<br>001 100-150 MHz HCLK/62<br>010 20-35 MHz HCLK/16<br>011 35-60 MHz HCLK/26<br>100 150-168 MHz HCLK/102<br>101、110、111 保留 - |
| 位 1 | MW | MII 写<br>此位置 1 是在告知 PHY，将要启动一个使用 MII 数据寄存器的写操作。如果此位未置 1, 则表示会启动一个读操作, 将数据放入 MII 数据寄存器。 |
| 位 0 | MB | MII 忙碌<br>向 ETH_MACMIIAR 和 ETH_MACMIIDR 写入前, 此位应读取逻辑 0。向 ETH_MACMIIAR 写入过程中, 此位也必须复位为 0。在 PHY 寄存器访问过程中，此位由应用程序设为 0b1，指 示读或写访问正在进行中。在对 PHY 进行写操作过程中，ETH_MACMIIDR（MII 数据）应始终保持有效，直到 MAC 将此位清零。在对 PHY 进行读操作过程中, ETH_MACMIIDR 始终无效，直到 MAC 将此位清零。在此位清零后，才可以向 ETH_MACMIIAR（MII 地址） 写入。 |

**以太网 MAC MII 数据寄存器 (ETH_MACMIIDR)**

偏移地址： 0x0014

复位值： 0x0000 0000

MAC MII 数据寄存器存储要写入 PHY 寄存器的数据，该寄存器地址在 ETH_MACMIIAR 中指定。ETH_MACMIIDR 也将存储从 PHY 寄存器读取的数据，该寄存器地址由 ETH_MACMIIAR 指定。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:16 | 保留 | 必须保持复位值。 |
| 位 15:0 | MD | MII 数据<br>其中包含在某次管理读操作之后从 PHY 中读取的 16 位数据值，或在某次管理写操作之前要 写入 PHY 的 16 位数据值。 |

### 以太网 MAC 流控制寄存器 (ETH_MACFCR)

偏移地址： 0x0018

复位值： 0x0000 0000

流控制寄存器通过 MAC 来管理控制（暂停命令）帧的生成和接收。忙碌位置“1”时对寄存器的写操作将导致 MAC 生成暂停控制帧。控制帧的字段根据 802.3x 规范中的指定进行选择，此寄存器中的暂停时间值被用在控制帧的暂停时间字段中。忙碌位将保持置 1，直到将控制帧发送到电缆。主机必须确保向该寄存器写入前已将忙碌位清零。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PT | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | | ZQPD | Reserved | PLT | | UPFD | RFCE | TFCE | FCB/BPA |
| | | | | | | | | rw | | rw | rw | rw | rw | rw | Rc_w1/rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:16 | PT | 暂停时间<br>此字段保留发送控制帧中暂停时间字段要使用的值。如果将暂停时间位配置为与 MII 时钟域双同步，则只能在目标时钟域的至少 4 个时钟周期后执行对此寄存器的连续写操作。 |
| 位 15:8 | 保留 | 必须保持复位值。 |
| 位 7 | ZQPD | 零时间片暂停禁止<br>如果置 1，当来自 FIFO 层的流控制信号去断言后，此位会禁止自动生成零时间片暂停控 制帧。<br>此位复位时，会使能自动零时间片暂停控制帧生成的正常操作 |
| 位 6 | 保留 | 必须保持复位值。 |
| 位 5:4 | PLT | 暂停阈值下限<br>此字段配置暂停定时器的阈值，达到该值时，会自动重新传输暂停帧。该阈值应始终小于位 [31:16] 配置的暂停时间。例如，如果 PT = 100H（256 个时隙），而 PLT = 01，则在第一个 PAUSE 帧发送完成后，第二个 PAUSE 帧会在 228 (256 - 28) 个时隙启动时自动发送。<br>选项 阈值 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 00 暂停时间减去 4 个时隙<br>01 暂停时间减去 28 个时隙<br>10 暂停时间减去 144 个时隙<br>11 暂停时间减去 256 个时隙<br>时隙定义为 MII 接口每发送 512 位（64 字节）所需的时间。 |
| 位 3 | UPFD | 单播暂停帧检测<br>当此位置 1 时，MAC 除了检测具有唯一多播地址的暂停帧外，还会检测具有 ETH_MACA0HR 和 ETH_MACA0LR 寄存器所指定的站单播地址的暂停帧。<br>当此位复位时，MAC 仅检测具有 802.3x 标准中指定的唯一多播地址的暂停帧。 |
| 位 2 | RFCE | 接收流控制使能<br>当此位置 1 时，MAC 对接收到的暂停帧进行解码，并禁止其在指定时间（暂停时间）内发送。<br>当此位复位时，将禁止暂停帧的解码功能。 |
| 位 1 | TFCE | 发送流控制使能<br>在全双工模式下，当此位置 1 时，MAC 将使能流控制操作来发送暂停帧。当此位复位时，将禁止 MAC 中的流控制操作，MAC 不会传送任何暂停帧。<br>在半双工模式下，当此位置 1 时，MAC 将使能背压操作。当此位复位时，将禁止背压功能。 |
| 位 0 | FCB/BPA | 流控制忙碌/背压激活<br>此位在全双工模式下启动暂停控制帧，在半双工模式下，TFCE 位置 1 时，会激活背压功能。<br>全双工模式下，向流控制寄存器写入数据前此位应读为 0。要启动暂停控制帧，应用程序必 须将此位置 1。在控制帧发送过程中，此位保持置 1 以指示帧发送正在进行中。当暂停控制帧发送完成后，MAC 会将此位复位为 0。将此位清零后，才可以对流控制寄存器执行写 操作。<br>在半双工模式下，当此位置 1（TFCE 也置 1）时， MAC 内核将置位背压功能。在背压操作期间，当 MAC 接收到新帧时，发送器会开始发送一个导致冲突的 JAM 模式。当 MAC 配置 为全双工模式时，将自动禁止 BPA。 |

**以太网 MAC VLAN 标记寄存器 (ETH_MACVLANTR)**

偏移地址： 0x001C

复位值： 0x0000 0000

VLAN 标记寄存器包含用于标识 VLAN 帧的 IEEE 802.1Q VLAN 标记。MAC 将所接收帧的第十三及第十四字节（长度/类型）与 0x8100 比较，随后的 2 个字节与 VLAN 标记比较，如果存在匹配，则将所接收帧状态中收到的 VLAN 位置 1。帧的合法长度从 1518 字节增加到 1522 字节。

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:17 | 保留 | 必须保持复位值。 |
| 位 16 | VLANTC | 12 位 VLAN 标记比较<br>当此位置 1 时，使用 12 位 VLAN 标识符而不是完整的 16 位 VLAN 标记进行比较和过滤。VLAN 标记的位 [11:0] 与收到的 VLAN 标记帧的相应字段进行比较。<br>当此位复位时，收到的 VLAN 帧的第十五和第十六字节的全部 16 位都要进行比较。 |
| 位 15:0 | VLANTI | VLAN 标记标识符（对应接收帧）<br>此字段包含用于标识 VLAN 帧的 802.1Q VLAN 标记，并与正在接收的 VLAN 帧的第十五和 第十六字节进行比较。位 [15:13] 是用户优先级，位 [12] 是标准格式指示符 (CFI)，位 [11:0] 是 VLAN 标记的 VLAN 标识符 (VID) 字段。 VLANTC 位置 1 时，仅使用 VID（位 [11:0]）进 行比较。如果 VLANTI（VLANTI[11:0]，如 果 VLANTC 位置 1）全部为零，则 MAC 不会检查用于 VLAN 标记比较的第十五和第十六字节，而是将类型字段值为 0x8100 的所有帧均声明为 VLAN 帧。 |

**以太网 MAC 远程唤醒帧过滤寄存器 (ETH_MACRWUFFR)**

偏移地址： 0x0028

复位值： 0x0000 0000

应用程序将通过此地址对远程唤醒帧过滤寄存器进行写/读操作。实际上，唤醒帧过滤寄存器是八个（不透明的）此类唤醒帧过滤寄存器。对偏移 (0x0028) 的这个地址所进行的八个连续写操作会写入所有唤醒帧过滤寄存器。对偏移 (0x0028) 的这个地址所进行的八个连续读操作会读取所有唤醒帧过滤寄存器。此寄存器包含第七个 MAC 地址的高 16 位。

有关详细信息，请参见远程唤醒帧过滤寄存器。

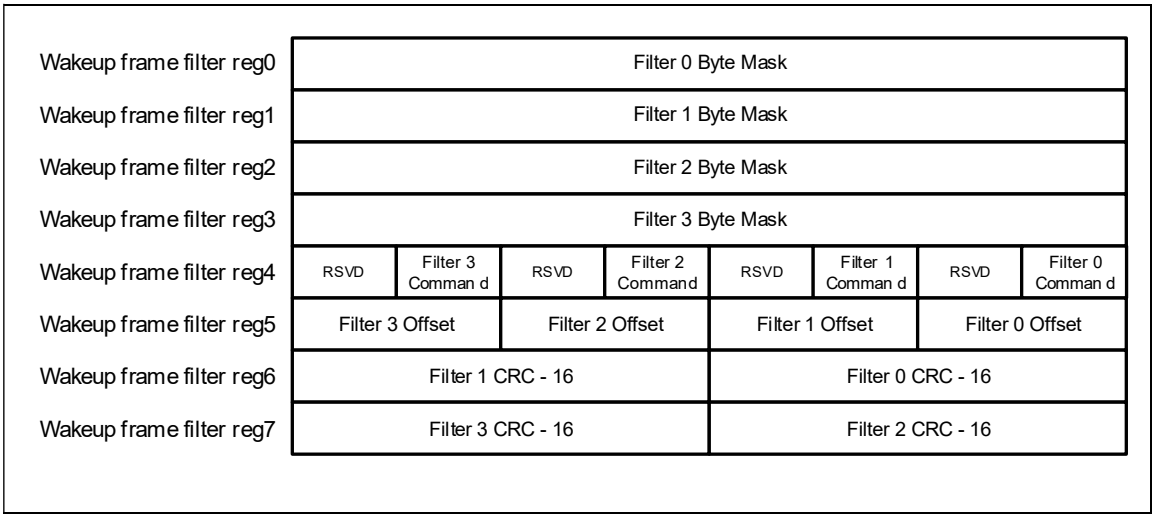



图 25-35 以太网 MAC 远程唤醒帧过滤寄存器(ETH_MACRWUFFR)

### 以太网 MAC PMT 控制和状态寄存器 (ETH_MACPMTCSR)

偏移地址： 0x002C

复位值： 0x0000 0000

ETH_MACPMTCSR 会配置唤醒事件请求并监视唤醒事件。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | GU | Reserved | | WFR | MPR | Reserved | | WFE | MPE | PD |
| | | | | | | rw | | | rc_r | rc_r | | | rw | rw | rs |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31 | WFFRPR | 唤醒帧过滤寄存器指针复位<br>当此位置 1 时，会将远程唤醒帧过滤寄存器指针复位为 0b000。它会在 1 个时钟周期后自动清零。 |
| 位 30:10 | 保留 | 必须保持复位值。 |
| 位 9 | GU | 全局单播<br>当此位置 1 时，它会将 MAC (DAF) 地址确认所过滤的任意单播包使能为唤醒帧。 |
| 位 8:7 | 保留 | 必须保持复位值。 |
| 位 6 | WFR | 接收到唤醒帧<br>当此位置 1 时，表示因接收唤醒帧而生成了电源管理事件。通过对此寄存器执行读操作可将此位清零。 |
| 位 5 | MPR | 接收到魔术数据包<br>当此位置 1 时，表示因接收魔术数据包而生成了电源管理事件。通过对此寄存器执行读操作可将此位清零。 |
| 位 4:3 | 保留 | 必须保持复位值。 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 2 | WFE | 唤醒帧使能<br>当此位置 1 时，会因为接收到唤醒帧而使能电源管理事件的生成。 |
| 位 1 | MPE | 魔术数据包使能<br>当此位置 1 时，会因为接收到魔术数据包而使能电源管理事件的生成。 |
| 位 0 | PD | 掉电<br>当此位置 1 时，所有接收到的帧都将丢弃。接收到魔术数据包或唤醒帧时，此位会自动清零，同时禁止掉电模式。此位清零后收到的帧会转发到应用程序。只有魔术数据包使能 (Magic Packet Enable) 或唤醒帧使能 (Wakeup Frame Enable) 位置 1 时，才可以将此位 置 1。 |

### 以太网 MAC 调试寄存器 (ETH_MACDBGR)

偏移地址： 0x0034

复位值： 0x0000 0000

此调试寄存器给出了发送及接收数据路径和 FIFO 的所有主要模块的状态。全零状态表示 MAC 内核处于空闲状态（FIFO 为空）并且数据路径中没有进行任何活动。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | TFF | TFNE |
| | | | | | | ro | ro |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | TFWA | TFRS | | MTP | MTFCS | | MMTEA |
| | ro | ro | ro | ro | ro | ro | ro |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | RFFL | |
| | | | | | | ro | ro |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | RFRCS | | RFWRA | Reserved | MSFRWCS | | MMRPEA |
| | ro | ro | ro | | ro | ro | ro |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:26 | 保留 | 必须保持复位值。 |
| 位 25 | TFF | Tx FIFO 已满<br>当它为高电平时，表示 Tx FIFO 已满，因此无法再接受发送的帧。 |
| 位 24 | TFNE | Tx FIFO 非空<br>当它为高电平时，表示 TxFIFO 不为空，还留有一些数据要发送。 |
| 位 23 | 保留 | 必须保持复位值。 |
| 位 22 | TFWA | Tx FIFO 写有效<br>当它为高电平时，表示 TxFIFO 写控制器有效且正在将数据传输到 TxFIFO。 |
| 位 21:20 | TFRS | Tx FIFO 读状态<br>此字段指示 TxFIFO 读控制器的状态：<br>00：空闲状态 (Idle state)<br>01：读状态（将数据传输到 MAC 发送器）<br>10：等待来自 MAC 发送器的 TxStatus |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 11：写入收到的 TxStatus 或刷新 TxFIFO |
| 位 19 | MTP | MAC 发送器处于暂停<br>当它为高电平时，表示 MAC 发送器处于暂停条件（仅在全双工模式下），因此不会安排发送任何帧。 |
| 位 18:17 | MTFCS | MAC 发送帧控制器状态<br>此字段指示 MAC 发送帧控制器的状态：<br>00：空闲<br>01：等待前一个帧的状态或 IFG/回退阶段结束<br>10：生成并发送暂停控制帧（在全双工模式下）<br>11：传输要发送的输入帧 |
| 位 16 | MMTEA | MAC MII 发送引擎有效<br>当它为高电平时，表示 MAC MII 发送引擎正在主动发送数据而未处于空闲状态。 |
| 位 15:10 | 保留 | 必须保持复位值。 |
| 位 9:8 | RFFL | Rx FIFO 填充级别<br>它指示 Rx FIFO 填充级别的状态：<br>00：RxFIFO 为空<br>01：RxFIFO 填充级别低于流控制取消激活阈值<br>10：RxFIFO 填充级别高于流控制激活阈值<br>11：RxFIFO 已满 |
| 位 7 | 保留 | 必须保持复位值。 |
| 位 6:5 | RFRCS | Rx FIFO 读控制器状态<br>它指示 Rx FIFO 读控制器的状态：<br>00：IDLE 状态<br>01：读取帧数据<br>10：读取帧状态（或时间戳）<br>11：刷新帧数据和状态 |
| 位 4 | RFWRA | Rx FIFO 写控制器有效<br>当它为高电平时，表示 Rx FIFO 写控制器有效并正在将收到的帧传输到 FIFO。 |
| 位 3 | 保留 | 必须保持复位值。 |
| 位 2:1 | MSFRWCS | MAC 小 FIFO 读/写控制器状态<br>当这些位为高电平时，表示 MAC 接收帧控制器模块的各个小 FIFO 读和写控制器的有效状态。 |
| 位 0 | MMRPEA | MAC MII 接收协议引擎有效<br>当它为高电平时，表示 MAC MII 接收协议引擎正在主动接收数据而未处于空闲状态。 |

**以太网 MAC 中断状态寄存器 (ETH_MACSR)**

偏移地址： 0x0038

复位值： 0x0000 0000

ETH_MACSR 寄存器内容在 MAC 中标识可生成中断的事件。

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | TSTS | Reserved | | MMCTS | MMCRS | MMCS | PMTS | Reserved | | |
| | | | | | | rc_r | | | r | r | r | r | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 15:10 | 保留 | 必须保持复位值。 |
| 位 9 | TSTS | 时间戳触发状态<br>当系统时间值等于或大于目标时间高位与低位寄存器中指定的值时，此位置 1。对此寄存器执行读操作时此位清零。 |
| 位 8:7 | 保留 | 必须保持复位值。 |
| 位 6 | MMCTS | MMC 发送状态<br>只要 ETH_MMCTIR Register 寄存器中生成中断，此位就为高电平。当此中断寄存器 (ETH_MMCTIR) 中的所有位都清零时，此位清零。 |
| 位 5 | MMCRS | MMC 接收状态<br>只要 ETH_MMCRIR 寄存器中生成中断，此位就为高电平。当此中断寄存器 (ETH_MMCRIR) 中的所有位都清零时，此位清零。 |
| 位 4 | MMCS | MMC 状态<br>只要位 6:5 中的任何一个位被设置为高电平，此位即会被设置为高电平。仅当这两个位均为低电平时，此位才被清零。 |
| 位 3 | PMTS | PMT 状态<br>只要在掉电模式下接收到魔术数据包或局域网唤醒帧，此位即被置 1（请参见 ETH_MACPMTCSR 寄存器中的位 5 和 6 第 xxx 页的以太网 MAC PMT 控制和状态寄存器 (ETH_MACPMTCSR)）。当此最后一个寄存器的两个位 [6:5] 因对 ETH_MACPMTCSR 寄存器执行读取操作而被清零时，此位也将会清零。 |
| 位 2:0 | 保留 | 必须保持复位值。 |

**以太网 MAC 中断屏蔽寄存器 (ETH_MACIMR)**

偏移地址： 0x003C

复位值： 0x0000 0000

通过使用 ETH_MACIMR 寄存器位，可屏蔽因 ETH_MACSR 寄存器中的相应事件导致的中断信号。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 15:10 | 保留 | 必须保持复位值。 |
| 位 9 | TSTIM | 时间戳触发中断屏蔽<br>如果此位置 1，则会禁止生成时间戳中断。 |
| 位 8:4 | 保留 | 必须保持复位值。 |
| 位 3 | PMTIM | PMT 中断屏蔽<br>如果此位置 1，则会禁止因将 ETH_MACSR 中的 PMT 状态位置 1 而触发中断信号。 |
| 位 2:0 | 保留 | 必须保持复位值。 |

### 以太网 MAC 地址 0 高位寄存器 (ETH_MACA0HR)

偏移地址： 0x0040

复位值： 0x0010 FFFF

MAC 地址 0 高位寄存器可保存站的第一个 6 字节 MAC 地址的高 16 位。请注意，在 MII 接口上接收到的第一个 DA 字节与 MAC 地址低位寄存器的 LS 字节（位 [7:0]）相对应。例如，如果在 MII 上将 0x1122 3344 5566（0x11 为第一个字节）作为目标地址进行接收，则 MAC 地址 0 寄存器 [47:0] 将与 0x6655 4433 2211 进行比较。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MO | Reserved | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MACA0H | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31 | MO | 始终为 1 |
| 位 30:16 | 保留 | 必须保持复位值。 |
| 位 15:0 | MACA0H | MAC 地址 0 高位 [47:32]<br>此字段包含 6 字节 MAC 地址 0 的高 16 位 (47:32) 。MAC 使用此字段过滤所接收的帧以及将 MAC 地址插入到发送流控制（暂停）帧中。 |

### 以太网 MAC 地址 0 低位寄存器 (ETH_MACA0LR)

偏移地址： 0x0044

复位值： 0xFFFF FFFF

MAC 地址 0 低位寄存器可保存站的第一个 6 字节 MAC 地址的低 32 位。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MACA0L | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MACA0L | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | MACA0L | MAC 地址 0 低位 [31:0]<br>此字段包含 6 字节 MAC 地址 0 的低 32 位。 MAC 使用此字段过滤所接收的帧以及将 MAC 地址插入到发送流控制（暂停）帧中。 |

### 以太网 MAC 地址 1 高位寄存器 (ETH_MACA1HR)

偏移地址： 0x0048

复位值： 0x0000 FFFF

MAC 地址 1 高位寄存器可保存站的第二个 6 字节 MAC 地址的高 16 位。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AE | SA | MBC | | | | | | Reserved | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MACA1H | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31 | AE | 地址使能<br>此位置 1 时, 地址过滤器使用 MAC 地址 1 实现完美过滤。此位清零时, 地址过滤器会忽略 用于过滤的地址。 |
| 位 30 | SA | 源地址<br>此位置 1 时, 将使用 MAC 地址 1 [47:0] 与所接收帧的 SA 字段进行比较。 此位清零时, 将使用 MAC 地址 1 [47:0] 与所接收帧的 DA 字段进行比较。 |
| 位 29:24 | MBC | 屏蔽字节控制<br>这些位是用于比较每个 MAC 地址 1 字节的屏蔽控制位。当将它们设为高电平时， MAC 内核不会将所接收到的 DA/SA 的相应字节与 MAC 地址 1 寄存器的内容进行比较。每个位都用于 控制字节的屏蔽，如下所示：<br>位 29：ETH_MACA1HR [15:8]<br>位 28：ETH_MACA1HR [7:0]<br>位 27：ETH_MACA1LR [31:24]<br>...<br>位 24：ETH_MACA1LR [7:0] |
| 位 23:16 | 保留 | 必须保持复位值。 |
| 位 15:0 | MACA1H | MAC 地址 1 高位 [47:32]<br>此字段包含第二个 6 字节 MAC 地址的高 16 位 (47:32)。 |

### 以太网 MAC 地址 1 低位寄存器 (ETH_MACA1LR)

偏移地址： 0x004C

复位值： 0xFFFF FFFF

MAC 地址 1 低位寄存器可保存站的第二个 6 字节 MAC 地址的低 32 位。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MACA1L | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MACA1L | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:0 | MACA1L | MAC 地址 1 低位 [31:0]<br>此字段包含 6 字节 MAC 地址 1 的低 32 位。如果在初始化过程之后，应用程序未加载此字段的内容，则该内容将不会被定义。 |

### 以太网 MAC 地址 2 高位寄存器 (ETH_MACA2HR)

偏移地址： 0x0050

复位值： 0x0000 FFFF

MAC 地址 2 高位寄存器可保存站的第二个 6 字节 MAC 地址的高 16 位。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| AE | SA | MBC | | | | | | Reserved | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MACA2H | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31 | AE | 地址使能<br>此位置 1 时，地址过滤器使用 MAC 地址 2 实现完美过滤。复位后，地址过滤器会忽略用于过滤的地址。 |
| 位 30 | SA | 源地址<br>此位置 1 时，将使用 MAC 地址 2 [47:0] 与所接收帧的 SA 字段进行比较。此位被复位时，将使用 MAC 地址 2 [47:0] 与所接收帧的 DA 字段进行比较。 |
| 位 29:24 | MBC | 屏蔽字节控制<br>这些位是用于比较每个 MAC 地址 2 字节的屏蔽控制位。当将其设为高电平时，MAC 内核不会将所接收到的 DA/SA 的相应字节与 MAC 地址 2 寄存器的内容进行比较。每个位都用于控制字节的屏蔽，如下所示：<br>位 29：ETH_MACA2HR [15:8]<br>位 28：ETH_MACA2HR [7:0]<br>位 27：ETH_MACA2LR [31:24]<br>…<br>位 24：ETH_MACA2LR [7:0] |
| 位 23:16 | 保留 | 必须保持复位值。 |
| 位 15:0 | MACA2H | MAC 地址 2 高位 [47:32]<br>此字段包含 6 字节 MAC 地址 2 的高 16 位 (47:32)。 |

### 以太网 MAC 地址 2 低位寄存器 (ETH_MACA2LR)

偏移地址： 0x0054

复位值： 0xFFFF FFFF

MAC 地址 2 低位寄存器可保存站的第二个 6 字节 MAC 地址的低 32 位。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MACA2L | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MACA2L | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | MACA2L | MAC 地址 2 低位 [31:0]<br>此字段包含第二个 6 字节 MAC 地址 2 的低 32 位。如果在初始化过程之后，应用程序未加载此字段的内容，则该内容将不会被定义。 |

**以太网 MAC 地址 3 高位寄存器 (ETH_MACA3HR)**

偏移地址： 0x0058

复位值： 0x0000 FFFF

MAC 地址 3 高位寄存器可保存站的第二个 6 字节 MAC 地址的高 16 位。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AE | SA | MBC | | | | | | Reserved | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MACA3H | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31 | AE | 地址使能<br>此位置 1 时, 地址过滤器使用 MAC 地址 3 实现完美过滤。此位清零时，地址过滤器会忽略用于过滤的地址。 |
| 位 30 | SA | 源地址<br>此位置 1 时, 将使用 MAC 地址 3 [47:0] 与所接收帧的 SA 字段进行比较。此位被清零时，将使用 MAC 地址 3 [47:0] 与所接收帧的 DA 字段进行比较。 |
| 位 29:24 | MBC | 屏蔽字节控制<br>这些位是用于比较每个 MAC 地址 3 字节的屏蔽控制位。当将这些位设为高电平时，MAC 内核不会将所接收到的 DA/SA 的相应字节与 MAC 地址 3 寄存器的内容进行比较。每个位都用于控制字节的屏蔽，如下所示：<br>位 29：ETH_MACA3HR [15:8]<br>位 28：ETH_MACA3HR [7:0]<br>位 27：ETH_MACA3LR [31:24]<br>...<br>位 24：ETH_MACA3LR [7:0] |
| 位 23:16 | 保留 | 必须保持复位值。 |
| 位 15:0 | MACA1H | MAC 地址 3 高位 [47:32]<br>该字段包含 6 字节 MAC 地址 3 的高 16 位 (47:32)。 |

## 25.8.2. MMC 寄存器

### 以太网 MMC 控制寄存器 (ETH_MMCCR)

偏移地址： 0x0100

复位值： 0x0000 0000

以太网 MMC 控制寄存器用于建立管理计数器的工作模式。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | | | | MCFHP | MCP | MCF | ROR | CSR | CR |
| | | | | | | | | | | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:6 | 保留 | 必须保持复位值。 |
| 位 5 | MCFHP | MMC 计数器全- 半预设置<br>当 MCFHP 为低电平且位 4 置 1 时，所有 MMC 计数器均预设为几乎一半值。所有帧计数器均预设为 0x7FFF_FFF0（一半值 - 16）<br>当 MCFHP 为高电平且位 4 置 1 时，所有 MMC 计数器均预设为几乎全值。所有帧计数器均预设为 0xFFFF_FFF0（全值 - 16） |
| 位 4 | MCP | MMC 计数器预设<br>该位置 1 时，所有计数器都将根据上面的位 5 初始化或预设为几乎全值或几乎一半值。该位将在 1 个时钟周期后自动清零。该位连同位 5 用于调试和测试由于 MMC 计数器变为半全值 或全值而导致的中断产生。 |
| 位 3 | MCF | MMC 计数器冻结<br>该位置 1 时，会冻结所有 MMC 计数器，使其保持为当前值。(该位清零后，才会因存在发送帧或接收帧而对 MMC 计数器进行更新。在此模式下，“读取时复位”位置 1 时，如果读 取任何 MMC 计数器，则该计数器也会清零。) |
| 位 2 | ROR | 读取时复位<br>该位置 1 时，读取 MMC 计数器后，该计数器会复位为零（复位后自清）。读取最低有效字节通道（位 [7:0]）后，计数器会清零。 |
| 位 1 | CSR | 计数器停止翻转<br>该位置 1 时，计数器达到其最大值后，不会返回到零。 |
| 位 0 | CR | 计数器复位<br>该位置 1 时，所有计数器都将复位。该位在 1 个时钟周期后自动清零。 |

### 以太网 MMC 接收中断寄存器 (ETH_MMCRIR)

偏移地址： 0x0104

复位值： 0x0000 0000

以太网 MMC 接收中断寄存器用于保持当接收统计计数器达到其最大值的一半时所生成的中断(计数器的 MSB 置 1)。 该寄存器是一个 32 位宽的寄存器。当读取引发中断的各个 MMC 计数器时，会将中断位清零。必须读取各个计数器的最低有效字节通道(位 [7:0])，才能将中断位清零。

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:18 | 保留 | 必须保持复位值。 |
| 位 17 | RGUFS | 接收的良好单播帧状态<br>当接收到的良好单播帧计数器达到其最大值的一半时，该位置 1。 |
| 位 16:7 | 保留 | 必须保持复位值。 |
| 位 6 | RFAES | 接收的帧对齐错误状态<br>当存在对齐错误的接收帧计数器达到其最大值的一半时，该位置 1。 |
| 位 5 | RFCES | 接收的帧 CRC 错误状态<br>当存在 CRC 错误的接收帧计数器达到其最大值的一半时，该位置 1。 |
| 位 4:0 | 保留 | 必须保持复位值。 |

### 以太网 MMC 发送中断寄存器 (ETH_MMCTIR)

偏移地址： 0x0108

复位值： 0x0000 0000

以太网 MMC 发送中断寄存器用于保持当发送统计计数器达到其最大值的一半时所生成的中断(计数器的 MSB 置 1)。该寄存器是一个 32 位宽的寄存器。当读取引发中断的各个 MMC 计数器时，会将中断位清零。必须读取各个计数器的最低有效字节通道(位 [7:0])，才能将中断位清零。

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:22 | 保留 | 必须保持复位值。 |
| 位 21 | TGFS | 发送的良好帧状态<br>当发送的良好帧计数器达到其最大值的一半时，该位置 1。 |
| 位 20:16 | 保留 | 必须保持复位值。 |
| 位 15 | TGFMSCS | 多个冲突后发送的良好帧状态<br>多个冲突后发送的良好帧计数器达到其最大值的一半时，该位置 1。 |
| 位 14 | TGFSCS | 单个冲突后发送的良好帧状态<br>单个冲突后发送的良好帧计数器达到其最大值的一半时，该位置 1。 |
| 位 13:0 | 保留 | 必须保持复位值。 |

### 以太网 MMC 接收中断屏蔽寄存器 (ETH_MMCRIMR)

偏移地址： 0x010C

复位值： 0x0000 0000

以太网 MMC 接收中断屏蔽寄存器用于保持当接收统计计数器达到其最大值的一半时所生成中断的屏蔽(计数器的 MSB 置 1)。该寄存器是一个 32 位宽的寄存器。

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:18 | 保留 | 必须保持复位值。 |
| 位 17 | RGUFM | 接收的良好单播帧屏蔽<br>如果该位置 1，当接收到的良好单播帧计数器达到其最大值的一半时，会屏蔽中断。 |
| 位 16:7 | 保留 | 必须保持复位值。 |
| 位 6 | RFAEM | 接收的帧对齐错误屏蔽<br>如果该位置 1，当存在对齐错误的接收帧计数器达到其最大值的一半时，会屏蔽中断。 |
| 位 5 | RFCEM | 接收的帧 CRC 错误屏蔽<br>如果该位置 1，当存在 CRC 错误的接收帧计数器达到其最大值的一半时，会屏蔽中断。 |
| 位 4:0 | 保留 | 必须保持复位值。 |

### 以太网 MMC 发送中断屏蔽寄存器 (ETH_MMCTIMR)

偏移地址： 0x0110

复位值： 0x0000 0000

以太网 MMC 发送中断屏蔽寄存器用于保持当发送统计计数器达到其最大值的一半时所生成中断的屏蔽。（计数器的 MSB 置 1）。该寄存器是一个 32 位宽的寄存器。

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:22 | 保留 | 必须保持复位值。 |
| 位 21 | TGFM | 发送的良好帧屏蔽 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 如果该位置 1，当发送的良好帧计数器达到其最大值的一半时，会屏蔽中断。 |
| 位 20:16 | 保留 | 必须保持复位值。 |
| 位 15 | TGFMSCM | 多个冲突后发送的良好帧屏蔽<br>如果该位置 1，当多个冲突后发送的良好帧计数器达到其最大值的一半时，会屏蔽中断。 |
| 位 14 | TGFSCM | 单个冲突后发送的良好帧屏蔽<br>如果该位置 1，当单个冲突后发送的良好帧计数器达到其最大值的一半时，会屏蔽中断。 |
| 位 13:0 | 保留 | 必须保持复位值。 |

### 以太网 MMC 在单个冲突后发送的良好帧计数器寄存器 (ETH_MMCTGFSCCR)

偏移地址： 0x014C

复位值： 0x0000 0000

该寄存器包含在半双工模式下于单个冲突后成功发送的帧的数量。

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

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:02 | TGFSCC | 单个冲突后发送的良好帧计数器 |

### 以太网 MMC 在多个冲突后发送的良好帧计数器寄存器 (ETH_MMCTGFMSCCR)

偏移地址： 0x0150

复位值： 0x0000 0000

该寄存器包含在半双工模式下于多个冲突后成功发送的帧的数量。

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

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:0 | TGFMSCC | 多个冲突后发送的良好帧计数器 |

### 以太网 MMC 发送的良好帧计数器寄存器 (ETH_MMCTGFCR)

偏移地址： 0x0168

复位值： 0x0000 0000

该寄存器包含发送的良好帧的数量。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | TGFC | 发送的良好帧计数器 |

### 带有 CRC 错误计数器寄存器的以太网 MMC 接收帧 (ETH_MMCRFCECR)

偏移地址： 0x0194

复位值： 0x0000 0000

此寄存器包含接收的含 CRC 错误的帧数。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | RFCEC | 接收的帧 CRC 错误计数器<br>接收的带有 CRC 错误计数器的帧 |

### 带有对齐错误计数器寄存器的以太网 MMC 接收帧 (ETH_MMCRFAECR)

偏移地址： 0x0198

复位值： 0x0000 0000

此寄存器包含接收的含对齐 (dribble) 错误的帧数。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | RFAEC | 接收的帧对齐错误计数器<br>接收的带有对齐错误计数器的帧 |

**MMC 接收的良好单播帧计数器寄存器 (ETH_MMCRGUFCR)**

偏移地址： 0x01C4

复位值： 0x0000 0000

此寄存器包含接收的良好单播帧数。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | RGUFC | 接收的良好单播帧计数器 |

## 25.8.3. IEEE 1588 时间戳寄存器

本节介绍用于支持符合 IEEE 1588 标准的精确网络时钟同步功能的寄存器。

**以太网 PTP 时间戳控制寄存器 (ETH_PTPTSCR)**

偏移地址： 0x0700

复位值： 0x0000 00002000

此寄存器用于控制时间戳生成和更新逻辑。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | TSPFFMAE | TSCNT | |
| | | | | | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TSSMRME | TSSEME | TSSIPV4FE | TSSIPV6FE | TSSPTPOEFE | TSSPTPOEFE | TSSSR | TSSARFE |
| rw | rw | rw | rw | rw | rw | rw | rw |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | TSARU | TSITE | TSSTU | TSSTI | TSFCU | TSE |
| | | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:19 | 保留 | 必须保持复位值。 |
| 位 18 | TSPFFMAE | 时间戳 PTP 帧过滤 MAC 地址使能<br>如果置 1，则当 PTP 直接通过以太网发送时，此位将使用 MAC 地址（除 MAC 地址 0）来过滤 PTP 帧。 |
| 位 17:16 | TSCNT | 时间戳时钟节点类型<br>可用的时钟节点类型如下:<br>00：普通时钟<br>01：边界时钟<br>10：端对端透明时钟<br>11：点对点透明时钟 |
| 位 15 | TSSMRME | 主节点相关消息时间戳快照使能<br>当此位置 1 时，仅拍摄主节点相关消息的快照。当此位清零时，仅拍摄从节点相关消息的快照。这仅适用于普通时钟和边界时钟节点。 |
| 位 14 | TSSEME | 事件消息时间戳快照使能<br>当此位置 1 时，仅拍摄事件消息的时间戳快照（SYNC、Delay_ Re、Pdelay_Req 或 Pdelay_Resp）。当此位清零时，将拍摄除 Announce、Management 和 Signaling 以外其它所有消息的快照。 |
| 位 13 | TSSIPV4FE | IPv4 帧时间戳快照使能<br>当此位置 1 时，将拍摄 IPv4 帧的时间戳快照。 |
| 位 12 | TSSIPV6FE | IPv6 帧时间戳快照使能<br>当此位置 1 时，将拍摄 IPv6 帧的时间戳快照。 |
| 位 11 | TSSPTPOEFE | PTP over ethernet 帧时间戳快照使能<br>当此位置 1 时，将拍摄在以太网帧中也含有 PTP 消息 (PTP over Ethernet) 的帧的时间戳快照。默认情况下, 将拍摄 UDP-IPEthernet PTP 数据包的快照。 |
| 位 10 | TSPTPPSV2E | 监听版本 2 格式时间戳 PTP 数据包使能<br>当此位置 1 时, 将使用版本 2 格式对 PTP 数据包进行监听。当此位清零时，将使用版本 1 格 式对 PTP 数据包进行监听。<br>*注意：IEEE 1588 版本 1 和版本 2 格式符合 IEEE 标准 1588-2008 (版本为 IEEE STD.1588- 2002).* |
| 位 9 | TSSSR | 时间戳亚秒翻转：数字或二进制翻转控制<br>当此位置 1 时，如果亚秒计数器达到值 0x3B9A C9FF（十进制 999 999 999）并递增了时间戳（高位）秒数，则时间戳低位寄存器将翻转。当此位清零时，亚秒寄存器的翻转值将达到 0x7FFF FFFF。亚秒增量必 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 须根据 PTP 参考时钟频率和此位的值正确进行编程。 |
| 位 8 | TSSARFE | 所有接收帧的时间戳快照使能<br>当此位置 1 时，时间戳快照会针对内核所接收的全部帧处于使能状态。 |
| 位 7:6 | 保留 | 必须保持复位值。 |
| 位 5 | TSARU | 时间戳加数寄存器更新<br>当此位置 1 时，时间戳加数寄存器的内容会被更新到 PTP 进行精密校准。更新结束时此位会清零。此寄存器位必须在读为零后，才可以将其置 1。 |
| 位 4 | TSITE | 时间戳中断触发使能<br>当此位置 1 时，如果系统时间大于在目标时间寄存器中写入的值，将产生时间戳中断。产生时间戳触发中断时，此位会清零。 |
| 位 3 | TSSTU | 时间戳系统时间更新<br>当此位置 1 时，系统时间将通过在时间戳高位更新寄存器和时间戳低位更新寄存器中指定的值进行更新（增至其中或从中减去）。必须将 TSSTU 和 TSSTI 两位均读为零，才可以将此位置 1。硬件中的更新完成后，此位会清零。 |
| 位 2 | TSSTI | 时间戳系统时间初始化<br>当此位置 1 时，系统时间将通过在时间戳高位更新寄存器和时间戳低位更新寄存器中指定的值进行初始化（覆盖）。此位必须在读为零后，才可以将其置 1。初始化完成后，此位会 清零。 |
| 位 1 | TSFCU | 时间戳精密更新或粗略更新<br>置 1 时，此位表示将使用精密更新方法更新系统时间戳。清零时，此位表示将使用粗略方法更新系统时间戳。 |
| 位 0 | TSE | 时间戳使能<br>当此位置 1 时，时间戳针对发送和接收帧处于使能状态。当此位清零时，时间戳功能将挂起，不会为发送和接收帧添加时间戳。由于保持的系统时间处于挂起状态，所以在将此位置为高电平后，始终都需要初始化时间戳功能（系统时间）。 |

下表指出了拍摄快照时的各种消息（取决于事件消息寄存器设置的时钟、使能主节点和使能快照）。

表 25-10 时间戳快照对寄存器位的相依性

| TSCNT（位 17:16） | TSSMRME（位 15）(1) | TSSEME（位 14） | 拍摄快照时的消息 |
|---|---|---|---|
| 00 或 01 | X(2) | 0 | SYNC、Follow_Up、Delay_Req、Delay_Resp |
| 00 或 01 | 1 | 1 | Delay_Req |
| 00 或 01 | 0 | 1 | SYNC |
| 10 | N/A | 0 | SYNC、Follow_Up、Delay_Req、Delay_Resp |
| 10 | N/A | 1 | SYNC、Follow_Up |
| 11 | N/A | 0 | SYNC、Follow_Up、Delay_Req、Delay_Resp、Pdelay_Req、Pdelay_Resp |
| 11 | N/A | 1 | SYNC、Pdelay_Req、Pdelay_Resp |

1. N/A = 不适用。
2. X = 无关。

**以太网 PTP 亚秒递增寄存器 (ETH_PTPSSIR)**

偏移地址： 0x0704

复位值： 0x0000 0000

此寄存器包含亚秒寄存器递增时使用的 8 位值。在粗略更新模式下（ETH_PTPTSCR 中的 TSFCU 位），此寄存器中的值在每个 HCLK 时钟周期后都会被添加到系统时间。在精密更新模式下，此寄存器中的值将在累加器溢出时被添加到系统时间。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:8 | 保留 | 必须保持复位值。 |
| 位 7:0 | STSSI | 系统时间亚秒增量<br>此寄存器中编程的值在每次更新时都会被增加到系统时间亚秒值的内容中。例如，要得到 20 ns 的精度，该值为：20 / 0.467 = ~ 43（或 0x2A）。 |

**以太网 PTP 时间戳高位寄存器 (ETH_PTPTSHR)**

偏移地址： 0x0708

复位值： 0x0000 0000

此寄存器包含最高有效（高位）32 个时间位。这个只读寄存器包含秒系统时间值。时间戳高位寄存器与时间戳低位寄存器可指示由 MAC 保持的系统时间的当前值，但是，它可以连续进行更新。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | STS | 系统时间秒<br>此字段中的值表示由内核维持的以秒为单位的系统时间当前值。 |

**以太网 PTP 时间戳低位寄存器 (ETH_PTPTSLR)**

偏移地址： 0x070C

复位值： 0x0000 0000

此寄存器包含最低有效（低位） 32 个时间位。这个只读寄存器包含亚秒系统时间值。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31 | STPNS | 系统时间正负号<br>该位指示正负时间值。置 1 时，该位指示时间表示为负值。清零时，该位指示时间表示为正值。由于系统时间始终应为正值，因此该位通常为零。 |
| 位 30:0 | STSS | 系统时间亚秒<br>该字段中的值包含亚秒时间表示，精度为 0.46 ns。 |

**以太网 PTP 时间戳高位更新寄存器 (ETH_PTPTSHUR)**

偏移地址： 0x0710

复位值： 0x0000 0000

该寄存器包含要写入、增加到系统时间值或从系统时间值中减去的最高有效（高）32 位时间。时间戳高位更新寄存器与时间戳低位更新寄存器一起用于初始化或更新 MAC 维护的系统时间。在将时间戳控制寄存器中的 TSSTI 或 TSSTU 位置 1 前，必须写入这两个寄存器。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | TSUS | 时间戳更新秒<br>该字段中的值指示要初始化或增加到系统时间中的时间，单位为秒。 |

**以太网 PTP 时间戳低位更新寄存器 (ETH_PTPTSLUR)**

偏移地址： 0x0714

复位值： 0x0000 0000

该寄存器包含要写入、增加到系统时间值或从系统时间值中减去的最低有效（低） 32 位时间。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31 | TSUPNS | 时间戳更新正负号<br>该位指示正负时间值。置 1 时，该位指示时间表示为负值。清零时，该位指示时间表示为正 值。当 TSSTI 置 1（系统时间初始化）时，该位应为零。如果当 TSSTU 置 1 时该位也置 1，则时间戳更新寄存器中的值要从系统时间中减去。否则，它要增加到系统时间中。 |
| 位 30:0 | TSUSS | 时间戳更新亚秒<br>该字段中的值指示要初始化或增加到系统时间中的亚秒时间。该值的精度为 0.46 ns（换句 话说，值 0x0000_0001 即为 0.46 ns）。 |

**以太网 PTP 时间戳加数寄存器 (ETH_PTPTSAR)**

偏移地址： 0x0718

复位值： 0x0000 0000

该寄存器由软件用来重新线性调节时钟频率，使其与主时钟频率匹配。仅当系统时间配置为微调更新模式（ETH_PTPTSCR 中的 TSFCU 位)时使用该寄存器值。该寄存器的内容在每个时钟周期添加到一个 32 位累加器中，系统时间在该累加器发生上溢时更新。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | TSA | 时间戳加数<br>该寄存器指示要添加到累加器寄存器以实现时间同步的 32 位时间值。 |

**以太网 PTP 目标时间高位寄存器 (ETH_PTPTTHR)**

偏移地址： 0x071C

复位值： 0x0000 0000

该寄存器包含要与用于中断事件生成的系统时间进行比较的高 32 位时间。目标时间高位寄存器与目标时间低位寄存器一起，用于在系统时间超过这些寄存器中编程的值时规划中断事件（ETH_PTPTSCR 中的 TSARU 位)。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | TTSH | 目标时间戳高位<br>该寄存器存储以秒为单位的时间。当时间戳的值匹配或同时超过两个目标时间戳寄存器时，MAC（如果已使能）会生成中断。 |

### 以太网 PTP 目标时间低位寄存器 (ETH_PTPTTLR)

偏移地址： 0x0720

复位值： 0x0000 0000

该寄存器包含要与用于中断事件生成的系统时间进行比较的低 32 位时间。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | TTSL | 目标时间戳低位<br>该寄存器存储以纳秒为单位的时间（带符号）。当时间戳的值匹配或同时超过两个目标时间戳寄存器时，MAC（如果已使能）会生成中断。 |

### 以太网 PTP 时间戳状态寄存器 (ETH_PTPTSSR)

偏移地址： 0x0728

复位值： 0x0000 0000

该寄存器包含时间戳状态寄存器。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:2 | 保留 | 必须保持复位值。 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 1 | TSTTR | 达到时间戳目标时间置 1 时，该位指示系统时间值大于或等于在目标时间高位和低位寄存器中指定的值。 |
| 位 0 | TSSO | 时间戳秒上溢<br>置 1 时，该位指示时间戳的秒值已上溢，超过 0xFFFF FFFF。 |

**以太网 PTP PPS 控制寄存器 (ETH_PTPPPSCR)**

偏移地址： 0x072C

复位值： 0x0000 0000

该寄存器控制 PPS 输出的频率。

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:4 | 保留 | 必须保持复位值。 |
| 位 3:0 | PPSFREQ | PPS 频率选择<br>PPS 输出频率设置为 2PPSFREQ Hz。<br>0000：1 Hz，使用二进制翻转时，脉冲宽度为 125 ms，使用数字翻转时，脉冲宽度为 100 ms<br>0001：2 Hz，使用二进制翻转时，占空比为 50%（不建议使用数字翻转）<br>0010：4 Hz，使用二进制翻转时，占空比为 50%（不建议使用数字翻转）<br>0011：8 Hz，使用二进制翻转时，占空比为 50%（不建议使用数字翻转）<br>0100：16 Hz，使用二进制翻转时，占空比为 50%（不建议使用数字翻转）<br>...<br>1111：32768 Hz，使用二进制翻转时，占空比为 50%（不建议使用数字翻转）<br>*注意： 如果使用数字翻转（TSSSR = 1，ETH_PTPTSCR 中的位 9），建议不要使用频率不是 1 Hz 的 PPS 输出。否则，使用数字翻转时，PPS 输出在更高的频率下的波形会不规则（尽管其平均频率在任何一秒窗口期间始终正确）。* |

## 25.8.4. DMA 寄存器

本节定义了每个 DMA 寄存器的位。只要地址是字对齐的，就允许非 32 位访问。

**以太网 DMA 总线模式寄存器 (ETH_DMABMR)**

偏移地址： 0x1000

复位值： 0x0000 2101

总线模式寄存器为 DMA 建立总线工作模式。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | MB | AAB | FPM | USP | RDP | | | | | | FB |
| | | | | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PM | | PBL | | | | | | EDFE | DSL | | | | | DA | SR |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rs |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:27 | 保留 | 必须保持复位值。 |
| 位 26 | MB | 混合突发<br>该位置为高电平且 FB 位置为低电平时，AHB 主接口将以 INCR（未定义突发）启动所有长度大于 16 的突发传输。该位清零时，对于长度等于或小于 16 的突发，将恢复为固定突发传输（INCRx 和 SINGLE）。 |
| 位 25 | AAB | 地址对齐的节拍<br>当该位设置为高电平并且 FB 位等于 1 时， AHB 接口会生成与起始地址 LS 位对齐的所有突发。如果 FB 位等于 0，则第一个突发（访问数据缓冲器的起始地址）不对齐，但后续的突发 与地址对齐。 |
| 位 24 | FPM | 4xPBL 模式<br>当设置为高电平时，该位会将编程的 PBL 值（位 [22:17] 和位 [13:8]）乘以四倍。因此，DMA 根据 PBL 值以最大 4、8、16、32、64 和 128 个节拍传输数据。 |
| 位 23 | USP | 使用单独的 PBL<br>设置为高电平时，它会配置 RxDMA，将位 [22:17] 中配置的值用作 PBL，而位 [13:8] 中的 PBL 值仅适用于 TxDMA 操作。当该位清零时，位 [13:8] 中的 PBL 值同时适用于两个 DMA 引擎。 |
| 位 22:17 | RDP | Rx DMA PBL<br>这些位指示要在一个 RxDMA 事务中传输的最大节拍数。这是在单个块读/写操作中使用的最大值。RxDMA 每次在主机总线上开始突发传输时，始终尝试按 RDP 中指定的方式进行突发。允许使用值 1、2、4、8、16 和 32 对 RDP 进行编程。任何其它值都会产生未定义的行为。<br>仅在 USP 设置为高电平时这些位才有效且适用。 |
| 位 16 | FB | 固定突发<br>该位控制 AHB 主接口是否执行固定突发传输。置 1 时， AHB 在正常突发传输开始期间仅使用 SINGLE、INCR4、INCR8 或 INCR16。复位时，AHB 使用 SINGLE 和 INCR 突发传输 操作。 |
| 位 15:14 | PM | Rx Tx 优先级比<br>RxDMA 请求优先于 TxDMA 请求，优先级比如下：<br>00：1:1<br>01：2:1<br>10：3:1<br>11：4:1<br>这仅在 DA 位清零时有效。 |
| 位 13:8 | PBL | 可编程突发长度<br>这些位指示要在一个 DMA 事务中传输的最大节拍数。这是在单个块读/写操作中使用的最大值。DMA 每次在主机总线上开始突发传输时，始终尝试按 PBL 中指定的方式进行突发。允许使用值 1、2、4、8、16 和 32 对 PBL 进行编程。任何其它值都会产生未定义的行为。当 USP 置 1 时，此 PBL 值仅适用于 TxDMA 事务。 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | PBL 值有以下限制：<br>可能的最大节拍数 (PBL) 受 Tx FIFO 和 Rx FIFO 大小的限制。<br>FIFO 有一个限制，即支持的最大节拍数等于 FIFO 深度的一半。<br>如果 PBL 由发送和接收 DMA 共用，则必须考虑 Rx FIFO 和 Tx FIFO 的最小深度。<br>请不要编程超出范围的 PBL 值，因为系统可能发生异常行为。 |
| 位 7 | EDFE | 增强描述符格式使能<br>该位置 1 时，使能增强描述符格式，并将描述符大小增加至 32 字节（8 个 DWORD）。如果已激活时间戳功能（ETH_PTPTSCR 位 0 TSE=1）或 IPv4 校验和减荷（ETH_MACCR 位 10 IPCO=1），则必须使用此增强描述符。 |
| 位 6:2 | DSL | 述符跳过长度<br>该位指定两个未链接描述符之间跳过的字数。地址从当前描述符结束处开始跳到下一个描述符起始处。当 DSL 值等于零时，在环形模式下，DMA 会将描述符表视为连续的。 |
| 位 1 | DA | DMA 仲裁<br>0：循环调度，Rx:Tx 优先级比在位 [15:14] 中给出<br>1：Rx 优先于 Tx |
| 位 0 | SR | 软件复位<br>当该位置 1 时，MAC DMA 控制器会复位所有 MAC 子系统的内部寄存器和逻辑。在所有内核时钟域完成复位操作后，该位自动清零。重新编程任何内核寄存器之前，在该位中读取 0 值。 |

**以太网 DMA 发送轮询要求寄存器 (ETH_DMATPDR)**

偏移地址： 0x1004

复位值： 0x0000 0000

应用程序使用此寄存器来指示 DMA 轮询发送描述符列表。发送轮询要求寄存器使能发送 DMA 来检查当前描述符是否为 DMA 所有。如果 TxDMA 处于挂起模式，则发出发送轮询要求命令将其唤醒。如果发送帧中出现下溢错误或发送 DMA 所拥有的描述符不可用，则 TxDMA 会进入挂起模式。用户可以随时发出此命令，当该命令开始重新获取主机存储器的当前描述符后，TxDMA 即会对其进行复位。

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:0 | TPD | 发送轮询要求<br>向这些位写入任何值时，DMA 都会读取 ETH_DMACHTDR 寄存器指向的当前描述符。如果该描述符不可用（由主机所有），则发送会返回到挂起状态，并将 ETH_DMASR 寄存器位 2 进行置位。如果该描述符可用，则发送会继续进行。 |

### 以太网 DMA 接收轮询要求寄存器 (ETH_DMARPDR)

偏移地址： 0x1008

复位值： 0x0000 0000

应用程序使用此寄存器来指示 DMA 轮询接收描述符列表。接收轮询要求寄存器会使能接收 DMA 来检查新描述符。此命令用于将 RxDMA 从挂起状态唤醒。仅当 RxDMA 所拥有的描述符不可用时，它才会进入挂起状态。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | RPD | 接收轮询要求<br>向这些位写入任何值时，DMA 都会读取 ETH_DMACHRDR 寄存器指向的当前描述符。如果该描述符不可用（由主机所有），则接收会返回到挂起状态，且不会将 ETH_DMASR 寄存器位 7 进行置位。如果该描述符可用，接收 DMA 将返回到活动状态。 |

### 以太网 DMA 接收描述符列表地址寄存器 (ETH_DMARDLAR)

偏移地址： 0x100C

复位值： 0x0000 0000

接收描述符列表地址寄存器会指向接收描述符列表的起始处。描述符列表位于物理存储器空间，且必须为字对齐。DMA 会将描述符列表的对应 LS 位置为低电平，进而在内部将其转换为总线宽度对齐地址。仅当接收停止时，才允许对 ETH_DMARDLAR 寄存 器执行写操作。停止后，必须先对 ETH_DMARDLAR 寄存器执行写操作，然后才能发出接收启动命令。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | SRL | 接收列表的起始处<br>此字段包含接收描述符列表中的首个描述符的基址。DMA 会在内部将 LSB 位 [1/2/3:0]（对应于 32/64/128 位的总线宽度）忽略，并将其值均视为零。因此，这些 LSB 位为只读。 |

### 以太网 DMA 发送描述符列表地址寄存器 (ETH_DMATDLAR)

偏移地址： 0x1010

复位值： 0x0000 0000

发送描述符列表地址寄存器会指向发送描述符列表的起始处。描述符列表位于物理存储器空间，且必须为字对齐。DMA 会将描述符列表的对应 LSB 位置为低电平，进而在内部将其转换为总线宽度对齐地址。仅当发送停止时，才允许对 ETH_DMATDLAR 寄存器执行写操作。发送停止后，可以先对 ETH_DMATDLAR 寄存器执行写操作，然后再发出发送启动命令。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | STL | 发送列表的起始处<br>此字段包含发送描述符列表中的首个描述符的基址。DMA 会在内部将 LSB 位 [1/2/3:0]（对应于 32/64/128 位的总线宽度）忽略，并将其值均视为零。因此，这些 LSB 位为只读。 |

### 以太网 DMA 状态寄存器 (ETH_DMASR)

偏移地址： 0x1014

复位值： 0x0000 0000

状态寄存器包含所有 DMA 向应用程序报告的状态位。软件驱动程序通常在中断服务例程或轮询期间读取 ETH_DMASR 寄存器。此寄存器中的大部分字段会导致主机中断。

读取时 ETH_DMASR 寄存器位不会清零。向 ETH_DMASR 寄存器 [16:0] 中的（未保留）位写入 1 会将其清零，写入 0 则不起作用。通过屏蔽 ETH_DMAIER 寄存器中的位，可以屏蔽各个对 应字段（位 [16:0]）。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | TSTS | PMTS | MMCS | Reserved | EBS | | | TPS | | | RPS | | | NIS |
| | | r | r | r | | r | r | r | r | r | r | r | r | r | rc_w1 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| AIS | ERS | FBES | Reserved | | ETS | RWTS | RPSS | RBUS | RS | TUS | ROS | TJTS | TBUS | TPSS | TS |
| rc_w1 | rc_w1 | rc_w1 | | | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:30 | 保留 | 必须保持复位值。 |
| 位 29 | TSTS | 时间戳触发状态<br>此位指示 MAC 内核的时间戳发生器块中发生的中断事件。软件必须读取 MAC 内核的状态寄存器，将其源（位 9）清零，才能将此位复位为 0。当此位处于高电平时，使能后会产生中断。 |
| 位 28 | PMTS | PMT 状态 |

<table>
<tr><th>位/位域</th><th>名称</th><th>描述</th></tr>
<tr><td></td><td></td><td>此位指示 MAC 内核的 PMT 中发生的事件。软件必须读取 MAC 内核中对应的寄存器，以获取产生中断的具体原因，然后将其源清零，才能将此位复位为 0。当此位处于高电平时，使能后会产生中断。</td></tr>
<tr><td>位 27</td><td>MMCS</td><td>MMC 状态<br>此位反映 MAC 内核的 MMC 中发生的事件。软件必须读取 MAC 内核中对应的寄存器，以获取产生中断的具体原因，然后将中断源清零，才能将此位置 0。当此位处于高电平时，使能后会产生中断。</td></tr>
<tr><td>位 26</td><td>保留</td><td>必须保持复位值。</td></tr>
<tr><td>位 25:23</td><td>EBS</td><td>错误位状态<br>这些位指示导致总线错误（AHB 接口上的错误响应）的错误类型。仅在致命总线错误位（ETH_DMASR 寄存器 [13]）置 1 时有效。此字段不会产生中断。
<table>
<tr><td rowspan="2">位 23</td><td>1</td><td>TxDMA 传输数据时出错 (Error during data transfer by TxDMA)</td></tr>
<tr><td>0</td><td>RxDMA 传输数据时出错 (Error during data transfer by RxDMA)</td></tr>
<tr><td rowspan="2">位 24</td><td>1</td><td>读取传输时出错 (Error during read transfer)</td></tr>
<tr><td>0</td><td>写入传输时出错 (Error during write transfer)</td></tr>
<tr><td rowspan="2">位 25</td><td>1</td><td>访问描述符时出错 (Error during descriptor access)</td></tr>
<tr><td>0</td><td>访问数据缓冲区时出错 (Error during data buffer access)</td></tr>
</table></td></tr>
<tr><td>位 22:20</td><td>TPS</td><td>发送过程状态<br>这些位指示发送 DMA FSM 的状态。此字段不会产生中断。<br>000：停止；发出复位或停止发送命令<br>001：运行中；正在获取发送传输描述符<br>010：运行中；正在等待状态<br>011：运行中；正在读取主机存储器缓冲区中的数据并将其加入发送缓冲区 (Tx FIFO) 队列 100、101：预留给将来使用<br>110：挂起；发送描述符不可用或发送缓冲区下溢<br>111：运行中；正在关闭发送描述符</td></tr>
<tr><td>位 19:17</td><td>RPS</td><td>接收过程状态<br>这些位指示接收 DMA FSM 的状态。此字段不会产生中断。<br>000：停止：发出复位或停止接收命令<br>001：运行中：正在获取接收传输描述符<br>010：预留给将来使用<br>011：运行中：正在等待接收数据包<br>100：挂起：接收描述符不可用<br>101：运行中：正在关闭接收描述符<br>110：预留给将来使用<br>111：运行中：将接收数据包的数据从接收缓冲区传输到主机存储器</td></tr>
<tr><td>位 16</td><td>NIS</td><td>正常中断汇总<br>当使能 ETH_DMAIER 寄存器中对应的中断位时，正常中断汇总位的值是以下位的逻辑或运算结果：<br>ETH_DMASR [0]：发送中断 (Transmit interrupt)<br>ETH_DMASR [2]：发送缓冲区不可用 (Transmit buffer unavailable)<br>ETH_DMASR [6]：接收中断 (Receive interrupt)<br>ETH_DMASR [14]：提前接收中断 (Early receive interrupt)<br>只有未屏蔽的位会影响正常中断汇总位。</td></tr>
</table>

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 它是黏着位，每当导致 NIS 位置 1 的对应位被清零时，必须同时将它也清零（通过向此位写 入 1）。 |
| 位 15 | AIS | 异常中断汇总<br>当使能 ETH_DMAIER 寄存器中对应的中断位时，异常中断汇总位的值是以下位的逻辑或运算结果：<br>ETH_DMASR [1]：发送过程停止 (Transmit process stopped)<br>ETH_DMASR [3]：发送 jabber 超时 (Transmit jabber timeout)<br>ETH_DMASR [4]：接收 FIFO 上溢 (Receive FIFO overflow)<br>ETH_DMASR [5]：发送下溢 (Transmit underflow)<br>ETH_DMASR [7]：接收缓冲区不可用 (Receive buffer unavailable)<br>ETH_DMASR [8]：接收过程停止 (Receive process stopped)<br>ETH_DMASR [9]：接收看门狗超时 (Receive watchdog timeout)<br>ETH_DMASR [10]：提前发送中断 (Early transmit interrupt)<br>ETH_DMASR [13]：致命总线错误 (Fatal bus error)<br>只有未屏蔽的位会影响异常中断汇总位。<br>它是黏着位，每当导致 AIS 位置 1 的对应位被清零时，必须同时将它也清零。 |
| 位 14 | ERS | 提前接收状态<br>此位指示 DMA 已填满数据包的首个数据缓冲区。接收中断 ETH_DMASR [6] 会自动将此位 清零。 |
| 位 13 | FBES | 致命总线错误状态<br>此位指示发生了总线错误，具体信息参见位 [25:23]。此位置 1 后，对应的 DMA 引擎会禁止其所有的总线访问。 |
| 位 12:11 | 保留 | 必须保持复位值。 |
| 位 10 | ETS | 提前发送状态<br>此位指示要发送的帧已完全传输到发送 FIFO。 |
| 位 9 | RWTS | 接收看门狗超时状态<br>当接收到的帧的长度大于 2048 个字节时，会对此位进行置位。 |
| 位 8 | RPSS | 接收过程停止状态<br>当接收过程进入停止状态时，会对此位进行置位。 |
| 位 7 | RBUS | 接收缓冲区不可用状态<br>此位指示接收列表中的下一个描述符由主机所拥有，DMA 无法获取。接收过程进入挂起状态。要恢复处理接收描述符，主机应更改描述符的拥有关系，然后发出接收轮询要求命令。如果未发出接收轮询要求命令，则当接收到下一个识别的传入帧时，接收过程会恢复。仅当 上一接收描述符由 DMA 所拥有时，才能将 ETH_DMASR [7] 置 1。 |
| 位 6 | RS | 接收状态<br>此位指示帧接收已完成。特定的帧状态信息已发布在描述符中。接收保持运行状态。 |
| 位 5 | TUS | 发送下溢状态<br>此位指示在帧发送期间，发送缓冲区发生下溢。发送会进入挂起状态，且下溢错误 TDES0[1] 置 1。 |
| 位 4 | ROS | 接收上溢状态<br>此位指示在帧接收期间，接收缓冲区发生上溢。如果部分帧已传输到应用程序，则 RDES0[11] 中的上溢状态位置 1。 |
| 位 3 | TJTS | 发送 jabber 超时状态<br>此位指示发送 jabber 定时器已过期，这意味着发送器过度有效。发送过程会中止并将其置于停止状态。这会导致对发送 jabber 超时 TDES0[14] 标志进行置位。 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 2 | TBUS | 发送缓冲区不可用状态<br>此位指示发送列表中的下一个描述符由主机所拥有，DMA 无法获取。发送会进入挂起状态。位 [22:20] 用于解释发送过程状态转换。要恢复处理发送描述符，主机应更改描述符位的拥有关系，然后发出发送轮询要求命令。 |
| 位 1 | TPSS | 发送过程停止状态<br>当发送停止时，此位置 1。 |
| 位 0 | TS | 发送状态<br>此位指示帧发送已完成，且首个描述符中的 TDES1[31] 位已置 1。 |

**以太网 DMA 工作模式寄存器 (ETH_DMAOMR)**

偏移地址： 0x1018

复位值： 0x0000 0000

工作模式寄存器将建立发送和接收工作模式及命令。作为 DMA 初始化过程的一部分，应将 ETH_DMAOMR 寄存器作为最后的 CSR 进行写操作。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | DTCEFD | RSF | DFRF | Reserved | | TSF | FTF | Reserved | | | TTC |
| | | | | | rw | rw | rw | | | rw | rs | | | | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TTC | | ST | Reserved | | | | | FEF | FUGF | Reserved | RTC | | OSF | SR | Reserved |
| rw | rw | rw | | | | | | rw | rw | | rw | rw | rw | rw | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:27 | 保留 | 必须保持复位值。 |
| 位 26 | DTCEFD | 禁止丢弃 TCP/IP 校验和错误帧<br>当此位置 1 时，如果帧中仅存在由接收校验和减荷引擎检测出来的错误，则内核不会丢弃它。这类帧在 MAC 接收到的以太网帧中没有任何错误（包括 FCS 错误），而仅在封装的有效负载中有错误。当此位清零时，如果 FEF 位进行了复位，则会丢弃所有错误帧。 |
| 位 25 | RSF | 接收存储并转发<br>当此位置 1 时，向 Rx FIFO 写入完整帧后，可从中读取一个帧，同时忽略 RTC 位。当此位清零时，Rx FIFO 在直通模式下工作，具体取决于 RTC 位指定的阈值。 |
| 位 24 | DFRF | 禁止刷新接收帧<br>当此位置 1 时，RxDMA 不会因为接收描述符/缓冲区不可用而刷新任何帧（通常情况下，当此位清零时它会这样做）。 |
| 位 23:22 | 保留 | 必须保持复位值。 |
| 位 21 | TSF | 发送存储并转发<br>当此位置 1 时，如果发送 FIFO 中有一个完整帧，则发送会启动。当此位置 1 时，会忽略由 ETH_DMAOMR 寄存器位 [16:14] 指定的 TTC 值。<br>该位清零后，ETH_DMAOMR 寄存器位 [16:14] 指定的 TTC 值才会有效。<br>该位只有在已停止传输时才能更改。 |
| 位 20 | FTF | 刷新发送 FIFO<br>该位置 1 时，发送 FIFO 控制器逻辑会复位为默认值，因此，Tx FIFO 中的所有数据均会丢失/刷新。刷新操作结束时该位在内部清零。此位清零之前不 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 得对工作模式寄存器执行写操作。 |
| 位 19:17 | 保留 | 必须保持复位值。 |
| 位 16:14 | TTC | 发送阈值控制<br>这三个位用于控制发送 FIFO 的阈值级别。当发送 FIFO 中的帧大小大于阈值时启动发送。此外，还会发送长度小于阈值的全帧。这些位只有在 TSF 位（位 21）清零后才能使用。<br>000：64<br>001：128<br>010：192<br>011：256<br>100：40<br>101：32<br>110：24<br>111：16 |
| 位 13 | ST | 启动/停止发送<br>该位置 1 时，发送过程会进入运行状态，DMA 会检查当前位置的发送列表查找待发送的帧。可通过发送列表中的当前位置，即由 ETH_DMATDLAR 寄存器设定的基址，或者上一次停止发送时的保留位置尝试获取描述符。如果当前描述符不属于 DMA，则发送过程会进入挂起状态，且发送缓冲区不可用位 (ETH_DMASR [2]) 置 1。启动发送命令只有在传输已停止时才有效。如果该命令在设置 DMA ETH_DMATDLAR 寄存器之前发出，则 DMA 行为无法预知。<br>该位清零时，发送过程会在完成发送当前帧的任务之后进入停止状态，并保存发送列表中的下一个描述符位置，该位置在重启发送后会成为当前位置。停止发送命令只有在当前帧发送过程结束或发送过程处于挂起状态时才有效。 |
| 位 12:8 | 保留 | 必须保持复位值。 |
| 位 7 | FEF | 转发错误帧<br>该位置 1 时，除短错误帧之外的所有帧都会转发到 DMA。<br>该位清零时，Rx FIFO 会丢弃带有错误状态（CRC 错误、冲突错误、巨帧、看门狗超时、上溢）的帧。不过，如果某个帧的起始字节（写）指针已传输到读控制器端（阈值模式下），则不会丢弃该帧。如果 ARI 总线上未传输（输出）帧的起始字节，则 Rx FIFO 会丢弃此错误帧。 |
| 位 6 | FUGF | 转发过小的好帧<br>该位置 1 时，Rx FIFO 会转发包含 pad 字节和 CRC 的过小帧（无错误但长度不足 64 字节的帧）。<br>该位清零时，Rx FIFO 会丢弃所有不足 64 字节的帧，除非因接收阈值下限更低（例如 RTC = 01）而导致此类帧已传输。 |
| 位 5 | 保留 | 必须保持复位值。 |
| 位 4:3 | RTC | 接收阈值控制<br>这两个位用于控制接收 FIFO 的阈值级别。当接收 FIFO 中的帧大小大于阈值时启动 DMA 传输（请求）。此外，长度小于阈值的全帧会自动传输。<br>*注意： 如果配置的接收 FIFO 大小为 128 字节，则不适合使用 11 作为阈值。*<br>*注意： 这些位只有在 RSF 位为 0 时才有效，而当 RSF 位置 1 后会忽略这些位。*<br>00：64<br>01：32<br>10：96<br>11：128 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 2 | OSF | 处理第二个帧<br>该位置 1 时会命令 DMA 处理第二个发送数据帧，即使尚未获得首个帧的状态。 |
| 位 1 | SR | 启动/停止接收<br>该位置 1 时，接收过程会进入运行状态。DMA 尝试从接收列表中获取描述符并处理传入帧。可通过列表中的当前位置，由 DMA ETH_DMARDLAR 寄存器设定的地址，或者上一次停止接收时的保留位置尝试获取描述符。如果 DMA 未占据任何描述符，则接收过程会进入挂起状态，且接收缓冲区不可用位 (ETH_DMASR [7]) 置 1。启动接收命令只有在已停止接收时才有效。如果该命令在设置 DMA ETH_DMARDLAR 寄存器之前发出，则 DMA 行为无法预知。<br>该位清零时，RxDMA 会在传输当前帧之后停止操作，并保存接收列表中的下一个描述符位置，该位置在接收过程重启后会成为当前位置。停止接收命令只有在当前接收过程处于运行（等待接收数据包）或挂起状态时才有效。 |
| 位 0 | 保留 | 必须保持复位值。 |

**以太网 DMA 中断使能寄存器 (ETH_DMAIER)**

偏移地址： 0x101C

复位值： 0x0000 0000

中断使能寄存器可使能 ETH_DMASR 报告的中断。将一个位置 1 可使能相应的中断。

完成硬件或软件复位之后，会禁止所有中断。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AISE | ERIE | FBEIE | Reserved | | ETIE | RWTIE | RPSIE | RBUIE | RIE | TUIE | ROIE | TJTIE | TBUIE | TPSIE | TIE |
| rw | rw | rw | | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:17 | 保留 | 必须保持复位值。 |
| 位 16 | NISE | 正常中断汇总使能<br>该位置 1 时，会使能一个正常中断。该位清零时，会禁止一个正常中断。<br>该位可使能以 下位：<br>ETH_DMASR [0]：发送中断<br>ETH_DMASR [2]：发送缓冲区不可用<br>ETH_DMASR [6]：接收中断<br>ETH_DMASR [14]：提前接收中断 |
| 位 15 | AISE | 异常中断汇总使能<br>该位置 1 时，会使能一个异常中断。该位清零时，会禁止一个异常中断。<br>该位可使能以 下位：<br>ETH_DMASR [1]：发送过程停止<br>ETH_DMASR [3]：发送 jabber 超时<br>ETH_DMASR [4]：接收上溢<br>ETH_DMASR [5]：发送下溢 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | ETH_DMASR [7]：接收缓冲区不可用<br>ETH_DMASR [8]：接收过程停止<br>ETH_DMASR [9]：接收看门狗超时<br>ETH_DMASR [10]：提前发送中断<br>ETH_DMASR [13]：致命总线错误 |
| 位 14 | ERIE | 提前接收中断使能<br>当该位通过正常中断汇总使能位（ETH_DMAIER 寄存器 [16]）置 1 时，可使能提前接收中断。<br>该位清零时，会禁止提前接收中断。 |
| 位 13 | FBEIE | 致命总线错误中断使能<br>当该位通过异常中断汇总使能位（ETH_DMAIER 寄存器 [15]）置 1 时，可使能致命总线错误中断。<br>该位清零时，会禁止致命总线使能中断。 |
| 位 12:11 | 保留 | 必须保持复位值。 |
| 位 10 | ETIE | 提前发送中断使能<br>当该位通过异常中断汇总使能位（ETH_DMAIER 寄存器 [15]）置 1 时，可使能提前发送中断。<br>该位清零时，会禁止提前发送中断。 |
| 位 9 | RWTIE | 接收看门狗超时中断使能<br>当该位通过异常中断汇总使能位（ETH_DMAIER 寄存器 [15]）置 1 时，可使能接收看门狗超时中断。<br>该位清零时，会禁止接收看门狗超时中断。 |
| 位 8 | RPSIE | 接收过程停止中断使能<br>当该位通过异常中断汇总使能位（ETH_DMAIER 寄存器 [15]）置 1 时，可使能接收停止中断。该位清零时，会禁止接收停止中断。 |
| 位 7 | RBUIE | 接收缓冲区不可用中断使能<br>当该位通过异常中断汇总使能位（ETH_DMAIER 寄存器 [15]）置 1 时，可使能接收缓冲区不可用中断。<br>该位清零时，会禁止接收缓冲区不可用中断。 |
| 位 6 | RIE | 接收中断使能<br>当该位通过正常中断汇总使能位（ETH_DMAIER 寄存器 [16]）置 1 时，可使能接收中断。该位清零时，会禁止接收中断。 |
| 位 5 | TUIE | 下溢中断使能<br>当该位通过异常中断汇总使能位（ETH_DMAIER 寄存器 [15]）置 1 时，可使能发送下溢中断。<br>该位清零时，会禁止下溢中断。 |
| 位 4 | ROIE | 上溢中断使能<br>当该位通过异常中断汇总使能位（ETH_DMAIER 寄存器 [15]）置 1 时，可使能接收上溢中断。<br>该位清零时，会禁止上溢中断。 |
| 位 3 | TJTIE | 发送 jabber 超时中断使能<br>当该位通过异常中断汇总使能位（ETH_DMAIER 寄存器 [15]）置 1 时，可使能发送 jabber 超时中断。<br>该位清零时，会禁止发送 jabber 超时中断。 |
| 位 2 | TBUIE | 发送缓冲区不可用中断使能<br>当该位通过正常中断汇总使能位（ETH_DMAIER 寄存器 [16]）置 1 时，可使能发送缓冲区不可用中断。<br>该位清零时，会禁止发送缓冲区不可用中断。 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 1 | TPSIE | 发送过程停止中断使能<br>当该位通过异常中断汇总使能位（ETH_DMAIER 寄存器 [15]）置 1 时，可使能发送停止中断。<br>该位清零时，会禁止发送停止中断。 |
| 位 0 | TIE | 发送中断使能<br>当该位通过正常中断汇总使能位（ETH_DMAIER 寄存器 [16]）置 1 时，可使能发送中断。该位清零时，会禁止发送中断。 |

只有 DMA 状态寄存器的 TSTS 或 PMTS 位已置位但相应的中断均未带标记时，或者 NIS/AIS 状态位已置位并且相应的中断使能位 (NISE/AISE)已使能时，才会生成以太网中断。

**以太网 DMA 丢失帧和缓冲区上溢计数器寄存器 (ETH_DMAMFBOCR)**

偏移地址： 0x1020

复位值： 0x0000 0000

DMA 可维护两个计数器在接收过程中对丢失帧数目进行计数。该寄存器会报告计数器的当前值。计数器用于进行诊断。位 [15:0] 指示因缓冲区不可用（无可用的接收描述符）而丢失的帧。位 [27:17] 指示因 Rx FIFO 存在上溢情况以及帧长度过短（不足 64 字节的好帧）而丢失的帧。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Reserved | | | OFOC | MFA | | | | | | | | | | | OMFC |
| | | | rc_r | rc_r | rc_r | rc_r | rc_r | rc_r | rc_r | rc_r | rc_r | rc_r | rc_r | rc_r | rc_r |

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:29 | 保留 | 必须保持复位值。 |
| 位 28 | OFOC | FIFO 上溢计数器的上溢位 |
| 位 27:17 | MFA | 应用程序所丢失的帧<br>指示应用程序所丢失的帧数 |
| 位 16 | OMFC | 丢失帧计数器的上溢位 |
| 位 15:0 | MFC | 控制器所丢失的帧<br>指示控制器因主机接收缓冲区不可用而丢失的帧数。DMA 每丢弃一个传入帧，此计数器值 加 1。 |

**以太网 DMA 接收状态看门狗定时器寄存器 (ETH_DMARSWTR)**

偏移地址： 0x1024

复位值： 0x0000 0000

向该寄存器写入一个非零值时，可使能针对接收状态 (RS, ETH_DMASR[6]) 的看门狗定时器。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:8 | 保留 | 必须保持复位值。 |
| 位 7:0 | RSWTC | 接收状态 (RS) 看门狗定时器计数<br>指示 HCLK 时钟周期数乘以 256 之后的时间值，即看门狗定时器的设定时间值。看门狗定时器会在 RxDMA 结束帧传输之后由编程设定的值触发，但相应的 RS 状态位因相应描述符中的 RDES1[31] 已置 1 而未置 1。当看门狗定时器计时结束时，RS 位置 1 且定时器停止。由于 RS 根据每个接收帧的 RDES1[31] 进行自动设置而使 RS 位设为高电平时，看门狗定时器复位。 |

### 以太网 DMA 当前主机发送描述符寄存器 (ETH_DMACHTDR)

偏移地址： 0x1048

复位值： 0x0000 0000

当前主机发送描述符寄存器会指向 DMA 所读取的当前发送描述符的起始地址。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | HTDAP | 主机发送描述符地址指针<br>复位时清零。该指针在运行期间由 DMA 更新。 |

### 以太网 DMA 当前主机接收描述符寄存器 (ETH_DMACHRDR)

偏移地址： 0x104C

复位值： 0x0000 0000

当前主机接收描述符寄存器会指向 DMA 所读取的当前接收描述符的起始地址。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | HRDAP | 主机接收描述符地址指针<br>复位时清零。该指针在运行期间由 DMA 更新。 |

**以太网 DMA 当前主机发送缓冲区地址寄存器 (ETH_DMACHTBAR)**

偏移地址： 0x1050

复位值： 0x0000 0000

当前主机发送缓冲区地址寄存器会指向 DMA 所读取的当前发送缓冲区地址。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | HTBAP | 主机发送缓冲区地址指针<br>复位时清零。该指针在运行期间由 DMA 更新。 |

**以太网 DMA 当前主机接收缓冲区地址寄存器 (ETH_DMACHRBAR)**

偏移地址： 0x1054

复位值： 0x0000 0000

当前主机接收缓冲区地址寄存器会指向 DMA 所读取的当前接收缓冲区地址。

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | HRBAP | 主机接收缓冲区地址指针<br>复位时清零。该指针在运行期间由 DMA 更新。 |

# 26. 通用串行总线 (USB)

本章节适用于片上两个 USB 模块 USB1 和 USB2，对于模块的差异，请注意针对模块单独的说明。

## 26.1. USB 简介

本节介绍了 USB 控制器的架构和编程模型。

### 26.1.1. 缩写描述

本节中使用以下缩写：

表 26-1 缩写描述

| 缩写 | 描述 |
|---|---|
| USB | 通用串行总线 |
| FS | 全速 |
| HS | 高速 |
| LS | 低速 |
| OTG | On-the-go |
| PHY | 物理层 |
| MAC | 介质访问控制器 |
| PFC | 数据包 FIFO 控制器 |
| UTMI+ | USB 收发器接口 |
| UTMIFS | USB 全速收发器接口 |

### 26.1.2. 模块简介

芯片包含两个 USB 模块 USB1 和 USB2，两个模块都是双角色设备 (DRD) 控制器，同时支持设备和主机功能，并且完全符合 USB 2.0 规范的 On-The-Go 补充标准。

此外，控制器可配置为仅主机或仅设备控制器，完全符合 USB 2.0 规范。

在主机模式中，控制器支持高速 (HS, 480 Mbits/s **仅 USB2 支持**) 和全速 (FS, 12 Mbits/s) 传输。

在设备模式中，控制器支持高速 (HS, 480 Mbits/s **仅 USB2 支持**) 和全速 (FS, 12 Mbits/s) 传输。

OTG 模式下需要的唯一外部设备是提供 VBUS 的电荷泵。

### 26.1.3. 参考文档

- **On-The-Go Supplement to the USB 2.0 Specification** Revision 1.3
- **Universal Serial Bus Specification** Revision 2.0

## 26.2. USB 主要特性

主要特性可分为三类：通用特性、主机模式特性和设备模式特性。

### 26.2.1. 通用特性

USB 接口的通用特性如下：

- 经 USB-IF 认证，符合通用串行总线规范 2.0 版本；
- USB1 内嵌 UTMIFS 全速 PHY；
- USB2 内嵌 UTMI+ **高速** PHY；
- 在 OTG 应用中允许主机关闭 VBUS 以节省功耗，而不需要外部组件；
- 允许使用内部比较器来监视 VBUS 电平；
- 支持主机和设备之间的动态角色切换；
- 可通过软件配置为以下角色：
  - USB OTG FS 双角色设备；
- USB1 不含有 DMA；USB2 内嵌 DMA，并可软件配置 AHB 的突发传输类型；
- 具备省电功能，例如在 USB 挂起期间停止系统时钟；
- 具有包含高级 FIFO 管理的专用 RAM **(USB1: 1280 Byte, USB2: 4096 Byte)**：
  - 可以将存储区配置为不同 FIFO，以便灵活高效地使用 RAM；
  - 每个 FIFO 可包含多个数据包；
  - 动态地进行存储器分配；
  - FIFO 大小可配置为 2 的幂以外的值，以便连续使用存储区；
- 一帧之内可以无需要应用程序干预，以达到最大 USB 带宽。

### 26.2.2. 主机模式特性

主机模式下的 USB 接口特征如下：

- 需要外部电荷泵来生成 VBUS
- 多个主机通道，每个通道可动态地进行重新配置，可支持任何类型的 USB 传输 **(USB1: 8 通道, USB2: 12 通道)；**
- 内置硬件调度器：
  - 在周期性硬件队列中存储多达 8 个中断加同步传输请求；
  - 在非周期性硬件队列中存储多达 8 个控制加批量传输请求；
- 管理一个共享 RX FIFO、一个周期性 TX FIFO 和一个非周期性 TX FIFO，以有效使用 USB 数据 RAM；
- 在主机模式下具备对 SOF 帧周期进行动态调校的功能。

### 26.2.3. 设备模式特性

USB 接口在设备模式下具有以下特性：

- 具有多对双向 IN / OUT 端点（含控制端点），可配置为支持批量、中断或同步传输 **(USB1: 4 对, USB2: 6 对)；**
- 管理一个共享 Rx FIFO 和一个 Tx-OUT FIFO，可高效使用 USB 数据 RAM；
- 管理多个专用 Tx-IN FIFO（每个 IN 配置的 EP 使用一个）以降低应用程序负载 **(USB1: 4 个, USB2: 6 个)；**

- 具备软断开功能。

## 26.3. USB 功能说明

### 26.3.1. USB 结构框图

USB 模块结构框图如下所示：

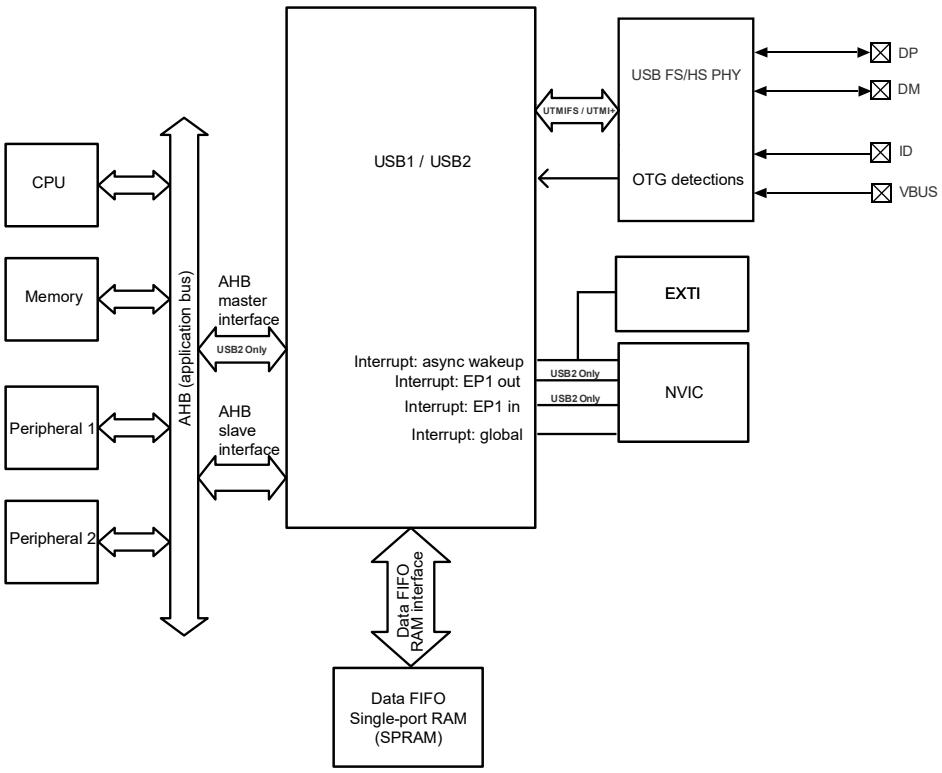



图 26-1 USB-BlockDiagram

### 26.3.2. USB PHY

USB1 嵌入了一个支持全速模式的 FS OTG PHY。

USB2 嵌入了一个支持高速/全速模式的 HS OTG PHY。

**注意：为确保 USB 模块正常工作，AHB 频率应大于 30 MHz。**

## 26.4. OTG 双角色设备

### 26.4.1. ID 线检测

主机或设备（默认设置）角色取决于 ID 输入线的电平。角色是在插入 USB 电缆时确定，并且取决于 USB 电缆的哪一端连接到 micro-AB 插座：

- 如果 USB 电缆的 B 端连入，其 ID 线悬空，则由于设备在 ID 线上的集成上拉电阻设备将检测到 ID 高电平并确认采取默认的设备角色。在此配置中，USB 符合 USB On- The-Go 补充标准第 1.3 版中

第 6.8.2 章节 On-The-Go B 器件中所述的 FSM 标准。

- 如果 USB 电缆的 A 端连入，且 ID 线接地，则 USB 将发出 ID 线状态更改中断（USB_GINTSTS.CIDSCHG 位）以初始化主机软件，且自动切换为主机角色。在此配置中，USB 符合 USB On-The-Go 补充标准第 1.3 版中第 6.8.1 章节 On-The-Go A 器件中所述的 FSM 标准。

## 26.5. 设备模式下的 USB 功能说明

在以下情形下， USB 作为 USB 设备运行：

- OTG B 器件
  OTG B 器件在插入 USB 电缆 B 端时的默认状态
- OTG A 器件
  OTG A 器件被 HNP 切换为外设角色后的状态
- B 器件
  如果 ID 线有效，设备与 USB 电缆的 B 端相连，而且全局 USB 配置寄存器中的 HNP 功能位（USB_GUSBCFG.HNPCAP 位）清零（请参见 On-The-Go 规范第 1.3 版的第 6.8.3 节）。
- 仅作设备
  全局 USB 配置寄存器中的强制外设模式位（USB_GUSBCFG.FDMOD）置 1，可强制 USB 模块以 USB 外设模式运行（请参见 On-The-Go 规范第 1.3 版中的第 6.8.3 节）。这种情况下，即使 USB 连接器上存在 ID 线，也会将该 ID 线忽略。

*注意： 在 B 设备模式和仅作外设模式下，若要构建总线供电设备，必须添加一个外部调压器以通过 VBUS 来生成 VDD 电源电压。*

### 26.5.1. USB 设备状态

#### 供电状态

VBUS 输入检测到 B 会话有效电压，就会使 USB 设备进入供电状态（请参见 USB 2.0 规范 第 9.1 节）。然后， USB 自动连接 DP 上拉电阻，发出全速设备与主机相连的信号并生成会话请求中断（USB_GINTSTS.SRQINT 位），指示进入供电状态。此外，VBUS 输入还可确保主机在 USB 操作期间提供有效的 VBUS 电平。如果 VBUS 降至 B 会话有效电压以下（例如，由于发生电源干扰或者主机端口已关闭），则 USB 自动断开连接，并且生成检测到会话结束中断（USB_GOTGINT.SEDET 位），指示 USB 已退出供电状态。

在供电状态下，USB 会收到来自主机的复位信号。其它 USB 操作则无法执行。收到复位信号，将生成检测到复位中断（USB_GINTSTS.USBRST）。复位完成时，将生成枚举已完成中断（USB_GINTSTS.ENUMDNE 位），并且 USB 进入默认状态。

#### 软断开

供电状态可借助软断开功能通过软件退出。将设备控制寄存器中的软断开位（USB_DCTL.SDIS 位）置 1 即可移除 DP 上拉电阻，此时尽管没有从主机端口实际拔出 USB 电缆， 但主机端仍会生成设备断开检测中断。

#### 默认状态

默认状态下，USB 会从主机收到 SET_ADDRESS 命令。其它 USB 操作则无法执行。当 USB 上解码出有效 SET_ADDRESS 命令时，应用程序会将相应的数值写入设备配置寄存器中的设备地址字段（USB_DCFG.DAD 位）。USB 随即进入地址状态，并准备好以所配置的 USB 地址对主机事务进行应答。

#### 挂起状态

USB 设备持续监视 USB 活动。如果 USB 处于空闲状态 3ms，则发出早期挂起中断（USB_GINTSTS.ESUSP 位）。并在 3ms 后通过挂起中断（USB_GINTSTS.USBSUSP 位）来确认设备进入挂起状态。然后，设备状态寄存器中的设备挂起位（USB_DSTS.SUSPSTS 位）自动置 1，USB 随即进入挂起状态。
设备可自行从挂起状态退出。这种情况下，应用程序应主动将设备控制寄存器中的远程唤醒信号位（USB_DCTL.RWUSIG 位）置 1，并在 1ms 到 15ms 后将其清零。
但若设备检测到主机发出的恢复信号时，将生成恢复中断（USB_GINTSTS.WKUPINT 位），设备挂起位自动清零。

### 26.5.2. USB 设备端点

USB 模块实现了以下 USB 端点：

- 控制端点 0
  - 该端点是双向的，仅处理控制消息。
  - 该端点具有单独一组寄存器来处理 IN 和 OUT 事务，还具有专用的控制寄存器 (USB_DI/OEPCTL0)、传输配置寄存器 (USB_DI/OEPTSIZ0) 和状态中断寄存器 (USB_DI/OEPINT0)。
  - 控制和传输大小寄存器中可用的位与其它端点略有不同。
- IN 端点
  - 可以将其配置为支持同步、批量或中断传输类型。
  - 它们具备专用的控制寄存器 (USB_DIEPCTLn)、传输配置寄存器 (USB_DIEPTSIZn) 和状态中断寄存器 (USB_DIEPINTn)。
  - 设备 IN 端点通用中断屏蔽寄存器 (USB_DIEPMSK) 可以在所有 IN 端点（包括 EP0）上使能/禁止单个端点中断源。
  - 支持不完整的同步 IN 传输中断（USB_GINTSTS.IISOIXFR 位）。只要有一个同步 IN 端点，其传输未在当前帧内完成，则触发该中断。该中断和周期帧结束中断 (USB_GINTSTS.EOPF) 一起触发。
- OUT 端点
  - 可以将其配置为支持同步、批量或中断传输类型。
  - 它们具备专用的控制寄存器 (USB_DOEPCTLn)、传输配置寄存器 (USB_DOEPTSIZn) 和状态中断寄存器 (USB_DOEPTSIZn)。
  - 设备 OUT 端点通用中断屏蔽寄存器 (USB_DOEPMSK) 可以在所有 OUT 端点（包括 EP0）上使能/禁止单个端点中断源。
  - 支持不完整的同步 OUT 传输中断（USB_GINTSTS.INCOMPISOOUT 位）。只要有一个同步 OUT 端点，其传输未在当前帧内完成，则触发该中断。该中断和周期帧结束中断 (USB_GINTSTS.EOPF) 一起触发。

#### 端点控制

通过设备端点 IN/OUT 控制寄存器 n (DIEPCTLn/DOEPCTLn) 可以对端点进行以下控制：

- 端点使能/禁止；
- 在当前配置下激活端点；
- 设置 USB 传输类型（同步、批量和中断）；

- 设置支持的数据包大小；
- 设置与 IN 端点相关的 Tx-FIFO 编号；
- 设置希望收到的或发送时要使用到的 data0/data1 PID（仅限批量/中断传输）；
- 设置接收或发送事务时所对应的奇/偶帧（仅限同步传输）；
- 可以设置 NAK 位，从而不论此时 FIFO 的状态如何，都对主机的请求回复 NAK；
- 可以设置 STALL 位，使得主机对该端点的令牌都被硬件回复 STALL；
- 可以将 OUT 端点设置为侦听模式，即对接收到的数据不进行 CRC 检查。

**端点传输**

设备端点传输大小寄存器 n (DI/OEPTSIZn) 允许应用程序对传输大小参数进行编程并读取传输状态。
必须在端点控制寄存器中的端点使能位置 1 之前完成对此寄存器的设置。
使能端点后，这些字段立即变为只读状态，同时 OTG FS 模块根据当前传输状态对这些字段进行更新。
可对以下传输参数进行编程：

- 以字节为单位的传输大小；
- 构成整个传输大小的数据包个数。

**端点状态/中断**

设备端点中断寄存器 n (DI/OEPINTn) 指示端点在出现 USB 和 AHB 相关事件时的状态。当模块中断寄存器中的 OUT 端点中断位或 IN 端点中断位（分别为 USB_GINTSTS.OEPINT 位或 USB_GINTSTS.IEPINT 位）置 1 时，应用程序必须读取这些寄存器以获得详细信息。在应用程序读取这些寄存器之前，必须先读取设备全体端点中断 (USB_DAINT) 寄存器，以获取设备端点中断寄存器 n 的具体端点编号。应用程序必须将此寄存器中的相应位清零，才能将 DAINT 和 GINTSTS 寄存器中的相应位清零。

模块提供以下状态检查和中断产生功能：

- 传输完成中断，指示应用程序 (AHB) 和 USB 端均已完成数据传输；
- Setup 阶段已完成（仅针对控制传输类型的 OUT 端点）；
- 相关的发送 FIFO 为半空或全空状态（IN 端点）；
- NAK 应答已发送到主机（仅针对同步传输的 IN 端点）；
- TxFIFO 为空时接收到 IN 令牌（仅针对批量和中断传输类型的 IN 端点）；
- 尚未使能端点时接收到 OUT 令牌；
- 检测到 babble 错误；
- 应用程序关闭端点生效；
- 应用程序对端点设置 NAK 生效（仅针对同步传输类型的 IN 端点）；
- 接收到 3 个以上连续 setup 数据包（仅针对控制类型的 OUT 端点）；
- 检测到超时状况（仅针对控制传输类型的 IN 端点）；
- 同步传输类型的数据包未产生中断而丢失。

## 26.6. 主机模式下的 USB 功能说明

本节介绍了 USB 在 USB 主机模式下所具有的功能。在以下情形下， USB 用作 USB 主机：

- **OTG A 主机**
  OTG A 器件在插入 USB 电缆 A 端时的默认状态

- **OTG B 主机**
  OTG B 器件被 HNP 切换为主机角色后的状态
- **A 器件**
  如果 ID 线有效，设备与 USB 电缆的 A 端相连，则全局 USB 配置寄存器中的 HNP 功能位（USB_GUSBCFG.HNPCAP 位）清零。DP/DM 线上的集成下拉电阻自动使能。
- **仅作主机**
  全局 USB 配置寄存器中的强制模式位（USB_GUSBCFG.FHMOD 位）将强制 USB 模块作为 USB 主机运行。这种情况下，即使 USB 连接器上存在 ID 线，也会将该 ID 线忽略。USB_FS_DP/USB_FS_DM 线上的集成下拉电阻自动使能。

*注意：微控制器不能输出 5 V 以提供 VBUS。因此，必须在微控制器以外添加电荷泵或电源开关（如果应用电路板提供 5V 电源）来驱动 5V VBUS 线。外部电荷泵可通过任何 GPIO 输出驱动。OTG A 主机、A 器件和仅作主机配置都需要使用电荷泵。*

*VBUS 输入可确保电荷泵在 USB 操作期间提供有效的 VBUS 电平，同时电荷泵过流输出可连接到任意配置成外部中断的 GPIO 引脚。在过流 ISR 中必须立即关闭 VBUS。*

### 26.6.1. 支持 SRP 功能的 USB 主机

全局 USB 配置寄存器中的 SRP 使能位（USB_GUSBCFG.SRPCAP 位）可提供 SRP 支持。使能 SRP 功能后，主机可在 USB 会话挂起时通过关闭 VBUS 电源来节省电能。A 器件会话请求协议一节中详细介绍了 SRP 主机的编程模型。

### 26.6.2. USB 主机状态

#### 给主机端口供电

微控制器不能输出 5V 以提供 VBUS。因此，必须在微控制器以外添加电荷泵或电源开关（如果应用电路板提供 5V 电源）来驱动 5V VBUS 线。外部电荷泵可通过任何 GPIO 输出驱动。如果应用通过所选 GPIO 来为 VBUS 供电，则必须将主机端口控制和状态寄存器中的端口电源位（USB_HPRT.PPWR 位）置 1。

#### VBUS 有效

在使能 HNP 或 SRP 的情况下，应将 VBUS 感应引脚连接到 VBUS。VBUS 输入可确保电荷泵在 USB 操作期间提供有效的 VBUS 电平。如果 VBUS 电压意外降至 VBUS 有效阈值 (4.25V) 以下，将通过会话结束检测位（USB_GOTGINT.SEDET 位）触发 OTG 中断。在中断处理程序中，应用程序必须关闭 VBUS 电源并将端口电源位清零。

在 HNP 和 SRP 同时关闭的情况下，应断开 VBUS 感应引脚与 VBUS 之间的连接。该引脚可用作 GPIO。

电荷泵过流标志也可用来防止电气损坏。将电荷泵的过流标志输出连接到任意 GPIO 输入，然后将其配置为出现有效电平时生成端口中断。过流 ISR 必须立即关闭 VBUS 并清零端口电源位。

#### 主机检测到设备连接

如果使能 SRP 或 HNP，即使可以随时连接 USB 外设或 B 器件，但是 USB 也只有在 VBUS 有效后 (4.75V) 才能检测到设备的连接。

当 VBUS 处于有效电平且已连接远程 B 器件时，USB 模块将发出主机端口中断信号，该中断由主机端口控制和状态寄存器中的设备连接位（USB_HPRT.PCDET 位）触发。

在 HNP 和 SRP 同时关闭的情况下，USB 设备或 B 器件将在连接后立即被检测到。USB 模块将发出主机端口中断信号，该中断由主机端口控制和状态寄存器中的设备连接位（USB_HPRT.PCDET 位）触发。

#### 主机检测到设备断开连接

设备断开事件将触发断开连接中断（USB_GINTSTS.DISCINT 位）。

#### 主机枚举

检测到设备连接后，主机必须通过向新的设备发出 USB 复位和配置命令来启动枚举过程。

发送 USB 复位前，应用程序将等待由“去抖动完成”位（USB_GOTGINT.DBCDNE 位）触发的 OTG 中断，这表明在 USB_FS_DP（全速）或 USB_FS_DM（低速）上连接上拉电阻所导致的电气抖动之后，总线再次稳定。

应用程序将主机端口控制和状态寄存器中的“端口复位”位（USB_HPRT.PRST 位） 置 1，保持 10ms 到 20ms，然后将“端口复位”位清零，以发出 USB 复位信号（单端零）。

USB 复位序列完成后，端口使能/禁止更改位（USB_HPRT.PENCHNG 位）将触发主机端口中断来通知应用程序，指示可以从主机端口控制和状态寄存器中的端口速度字段 （USB_HPRT.PSPD 位）读取所枚举设备的速度，以及主机已经开始驱动 SOF（全速）或“keep-alive”令牌（低速）。此时主机已就绪，可通过对设备发送命令来完成对设备的枚举。

#### 主机挂起

应用程序可通过将主机端口控制和状态寄存器中的端口挂起位（USB_HPRT.PSUSP）置 1 来挂起 USB 活动。USB 模块停止发送 SOF 并进入挂起状态。
可由远程设备的自主活动（远程唤醒）使总线退出挂起状态。在此情况下，远程唤醒事件将触发远程唤醒中断（USB_GINTSTS.WKUPINT 位），把主机端口控制和状态寄存器中的端口恢复位（USB_HPRT.PRES 位）置 1，并且在 USB 上自动发出恢复信号。应用程序必须监视恢复窗口持续时间，随后必须将端口恢复位清零以退出挂起状态并重新启动 SOF 发送。
如果由主机退出挂起状态，则应用程序必须将端口恢复位置 1 以在主机端口上启动恢复信号，监视恢复窗口持续时间并随后将端口恢复位清零。

### 26.6.3. 主机通道

USB 模块实现了多个主机通道。每个主机通道均可用于 USB 主机传输（USB 管道）。主机最多能同时处理 8 个传输请求。如果应用程序有 8 个以上的传输请求挂起，则通道可用后（也就是在收到“传输完成”和“通道停止”中断后），主机控制器驱动程序 (HCD) 必须为未处理的传输请求重新分配通道。

每个主机通道都可配置为支持输入/输出以及周期性/非周期性事务。每个主机通道都具有专用的控制寄存器 (HCCHARx)、传输配置寄存器 (HCTSIZx) 状态/中断寄存器 (HCINTn) 以及和其相关的中断屏蔽寄存器 (HCINTMSKx)。

#### 主机通道控制

通过主机通道 x 特性寄存器 (HCCHARx) 可以对主机通道作以下控制：

- 通道使能/禁止；
- 设置目标 USB 设备的速度：HS/FS/LS；
- 设置目标 USB 设备的地址；
- 设置与该通道通信的目标 USB 设备上的端点的编号；
- 设置该通道上的传输方向：IN/OUT；

- 设置该通道上的 USB 传输的类型：控制/批量/中断/同步；
- 设置与该通道通信的设备端点的最大包长；
- 设置要进行周期传输的帧：奇帧/偶帧。

**主机通道传输**

主机通道传输大小寄存器 n (HCTSIZn) 允许应用程序对传输大小参数进行编程并读取传输状态。

必须在主机通道特性控制寄存器中的通道使能位置 1 之前完成对此寄存器的设置。使能端点后，数据包计数字段立即变为只读状态，同时 OTG HS 模块根据当前传输状态对该字段进行更新。

可对以下传输参数进行编程：

- 以字节为单位的传输大小；
- 构成整个传输大小的数据包个数；
- 初始数据 PID。

**主机通道状态/中断**

主机通道 x 中断寄存器 (HCINTn) 指示端点在出现 USB 和 AHB 相关事件时的状态。当中断寄存器中的主机通道中断位（USB_GINTSTS.HCINT）置 1 时，应用程序必须读取这些寄存器以获得详细信息。在读取这些寄存器之前，应用程序必须先读取主机全体通道中断 (HCAINT) 寄存器，以获取主机通道 x 中断寄存器的通道编号。应用程序必须将此寄存器中的相应位清零，才能将 HAINT 和 GINTSTS 寄存器中的相应位清零。USB_HCINTMSKn 寄存器还提供每个通道各中断源的屏蔽位。

主机模块提供以下状态检查和中断产生功能：

- 传输完成中断，指示应用程序 (AHB) 和 USB 端均已完成数据传输；
- 通道因传输完成、USB 事务错误或应用程序发出禁止命令而停止；
- 相关的发送 FIFO 为半空或全空状态（IN 端点）；
- 接收到 ACK 响应；
- 接收到 NAK 响应；
- 接收到 STALL 响应；
- 由于 CRC 校验失败、超时、位填充错误和错误的 EOP 导致 USB 事务错误；
- 串扰错误；
- 帧上溢；
- 用于数据同步的翻转位出错。

### 26.6.4. 主机调度器

主机模块内置硬件调度器，可自主对应用程序发出的 USB 事务请求重新排序和管理。每一帧开始时，主机都先执行周期性（同步和中断）事务，然后执行非周期性（控制和批量）事务，以符合 USB 规范对同步和中断传输高优先级的保证。

主机通过请求队列（一个周期性请求队列和一个非周期请求队列）处理 USB 事务。每个请求队列最多可存储 8 个条目。每个条目代表一个应用程序发起但还未得到响应的 USB 事务请求，并存储了执行该 USB 事务所用到的 IN 或 OUT 通道的编号，以及其它相关信息。USB 事务请求在队列中的写入顺序决定了事务在 USB 接口上的执行顺序。

每一帧开始时，主机都先处理周期性请求队列，然后处理非周期性请求队列。如果当前帧结束时，计划在当

前帧执行的同步或中断类型的 USB 传输事务请求仍处于挂起状态，则主机将发出未完成周期性传输中断（USB_GINTSTS.IPXFR 位）。OTG HS 模块负责对周期性和非周期性请求队列的管理。周期性发送 FIFO 和队列状态寄存器 (HPTXSTS) 与非周期性发送 FIFO 和队列状态寄存器 (HNPTXSTS) 都为只读寄存器，应用程序可使用它们来读取各请求队列的状态。其中包括：

- 周期性（非周期性）请求队列中当前可用的空闲条目数（最多 8 个）；
- 周期性（非周期性）Tx-FIFO（OUT 事务）中当前可用的空闲空间；
- IN/OUT 令牌、主机通道编号和其它状态信息。

由于每个请求队列最多可存储 8 个 USB 事务请求，因此应用程序可以把主机 USB 事务请求提前发送给调度器；实际的通信最晚会在调度器处理完已挂起的 8 个周期事务和 8 个非周期事务完成之后出现在 USB 总线上。

要向主机调度器（队列）发出事务请求，应用程序必须读取 USB_HNPTXSTS.PTXQSAV 位或 USB_HNPTXSTS.NPTQXSAV 位，确保周期性（非周期性）请求队列中至少有一个可用空间来存储当前请求。

## 26.7. USB 功耗模式

OTG PHY 的功耗由通用模块配置寄存器中的以下三个位控制：

- PHY 掉电 (GCCFG.PWRDWN)
  该位可开启/关闭 PHY 全速收发器模块。先置位后才允许后续的 USB 操作。
- A-VBUS 感应使能 (GCCFG.VBUSASEN)
  该位可开启/关闭与 A 器件操作相关联的的 VBUS 比较器。必须在 A 器件（USB 主机）模式和 HNP 期间进行设置。
- B-VBUS 感应使能 (GCCFG.VBUSBSEN)
  该位可开启/关闭与 B 器件操作相关联的的 VBUS 比较器。必须在 B 器件（USB 设备）模式和 HNP 期间进行设置。
  USB 会话没有开始或设备未连接时，可以在 USB 挂起状态下使用功率降低技术。
- 停止 PHY 时钟（USB_PCGCCTL.STPPCLK 位）
  - 将时钟门控控制寄存器中的停止 PHY 时钟位置 1 时，OTG 高速模块的大部分内部时钟域都将由时钟门控关闭。即使应用程序仍提供时钟输入，也会节省掉模块由于时钟信号翻转带来的动态功耗。
  - 还会关掉收发器的大部分单元，只有负责检测异步恢复事件或远程唤醒事件的部分还保持工作状态。
- HCLK 门控（USB_PCGCCTL.GATEHCLK 位）
  将时钟门控控制寄存器中的 Gate HCLK 位置 1 时，USB 模块内部的大多数系统时钟域均由时钟门控关闭。只有寄存器读取和写入接口保持活动状态。即使应用程序仍提供时钟输入，也会节省掉模块由于时钟信号翻转带来的动态功耗。
- USB 系统停止
  - 当 USB 处于 USB 挂起状态时，应用程序可通过关闭 USB 系统中的所有时钟源来大幅降低总体功耗。USB 系统停止可通过以下方式激活：首先将停止 PHY 时钟位置 1，然后在电源控制系统模块 (PWR) 中将系统配置为深度睡眠模式。
  - USB 模块通过对 USB 上的远程唤醒（作为主机）或恢复（作为设备）信号进行异步检测，自动重新激活系统时钟和 USB 时钟。

## 26.8. FIFO RAM 分配

### 26.8.1. 设备模式

#### 接收 FIFO RAM

对于接收 FIFO RAM，应用程序应为 SETUP 数据包分配 RAM：接收 FIFO 中必须保留 10 个位置以在控制端点上接收 SETUP 数据包。这些位置将为 SETUP 数据包保留，OTG 模块不会向这些位置来写入任何其它数据。

必须为全局 OUT NAK 分配一个位置。状态信息也将与收到的每个数据包一起写入到 FIFO 中。因此，必须至少为接收数据包分配（最大数据包大小 / 4）+ 1 的空间。如果使能了高带宽端点或多个同步端点，则必须至少分配两个（最大数据包的四分之一）+ 1 的空间，以接收连续数据包。通常，推荐的空间为（最大数据包 / 4 + 1）的两倍，这样当上一个数据包向 AHB 传送时，USB 可同时接收后续的数据包。

传输完成状态信息和该端点收到的最后一个数据包会一起被推入 FIFO。通常情况下，推荐为每个 OUT 端点分配一个位置。

#### 发送 FIFO RAM

对于发送 FIFO RAM，每个 IN 端点发送 FIFO 需要的最小 RAM 空间是该 IN 端点的最大数据包大小。

*注意：为发送 IN 端点 FIFO 分配的空间越多，USB 的性能就越高。*

### 26.8.2. 主机模式

#### 接收 FIFO RAM

对于接收 FIFO RAM 分配，状态信息将与收到的每个数据包一起写入到 FIFO。因此，必须至少为接收数据包分配（最大数据包大小 / 4）+ 1 的空间。如果使能了高带宽通道或多个同步通道，则必须至少分配两个（最大数据包的四分之一）+ 1 的空间，以接收连续数据包。通常，推荐的空间为（最大数据包 / 4 + 1）的两倍，这样当上一个数据包向 AHB 传送时，USB 可同时接收后续的数据包。

传输完成状态信息和该通道收到的最后一个数据包会一起被推入 FIFO。因此，必须分配一个位置以存储此数据。

#### 发送 FIFO RAM

对于发送 FIFO RAM 分配，主机非周期发送 FIFO 所需要的最小 RAM 量是所有非周期性 OUT 通道中的最大数据包大小。通常，推荐的空间为最大数据包大小的两倍，这样当 USB 正在发送当前数据包的同时，AHB 可以往发送 FIFO 填入下一个数据包。

主机周期性发送 FIFO 所需要的最小 RAM 为所有周期性 OUT 通道中的最大数据包大小。如果至少有一个高带宽同步 OUT 端点，则空间必须至少为该通道中最大数据包大小的两倍。

*注意：为非周期性发送 FIFO 分配的空间越多，USB 的性能就越高。以 DMA 模式运行时，每个主机通道的 DMA 地址寄存器 (HCDMAn) 将存储在 SPRAM (FIFO) 中。为此需要在 FIFO 中为每个通道保留一个位置存放其地址寄存器。*

## 26.9. USB 中断

当 USB 控制器在一种模式（设备模式或主机模式）下工作时，应用程序不得以另一种角色模式访问寄存器。如果发生了非法访问，将会产生模式不匹配中断并在模块中断寄存器 （USB_GINTSTS.MMIS 位）中反映。当模块从一种角色模式切换到另一种角色模式时，新工作模式下的寄存器必须重新编程为上电复位后的状态。

下图为中断结构示意图：

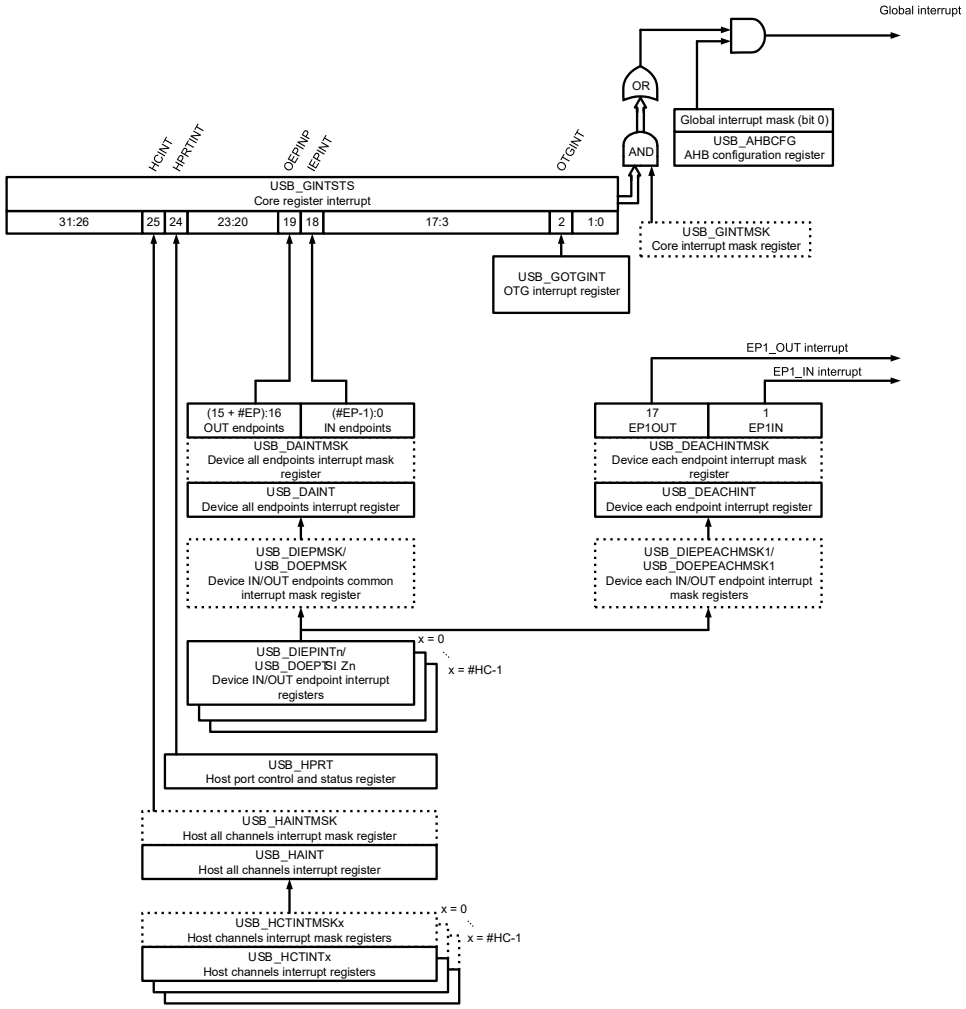



图 26-2 USB Interrupt hierarchy

## 26.10. USB 寄存器

应用程序通过 AHB 从接口对控制和状态寄存器 (CSR) 进行读写操作，以此来控制 USB 模块。这些都是 32 位寄存器，其地址按 32 位对齐，因此只能以 32 位的方式访问。CSR 分为以下几类：

- 模块全局寄存器；

- 主机模式寄存器;
- 主机全局寄存器;
- 主机端口 CSR;
- 主机通道相关寄存器;
- 设备模式寄存器;
- 设备全局寄存器;
- 设备端点相关寄存器;
- 电源和时钟门控寄存器;
- 数据 FIFO (DFIFO) 访问寄存器。

只有模块全局寄存器、电源和时钟门控寄存器、数据 FIFO 访问寄存器以及主机端口控制和状态寄存器，可在主机模式和设备模式下进行访问。当 USB 控制器在一种模式（设备模式或主机模式）下工作时，应用程序不得以另一种角色模式访问寄存器。如果发生了非法访问，将会产生模式不匹配中断并在模块中断寄存器（USB_GINTSTS.MMIS 位）中反映。当模块从一种角色模式切换到另一种角色模式时，新工作模式下的寄存器必须重新编程为上电复位后的状态。

### 26.10.1. USB 控制和状态寄存器 (GOTGCTL)

USB control and status register

**偏移地址：** 0x0000

**复位值：** 0x00000800

OTG 控制和状态寄存器控制模块的 OTG 功能并反映其状态

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | | | | | | | | | BSVLD | ASVLD | DBCT | CIDSTS |
| | | | | | | | | | | | | | r | r | r |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | DHNPEN | HSHNPEN | HNPRQ | HNGSCS | res. | | | | | | SRQ | SRQSCS |
| r | | | | rw | rw | rw | r | | | | | | | rw | rw |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:20 | 保留 | 必须保持复位值。 |
| 19 | BSVLD | B 会话有效 (B-session valid)<br>指示设备模式下收发器的状态。<br>0：B 会话无效。<br>1：B 会话有效。<br>在 OTG 模式下，该位可用于确定设备处于连接状态还是断开状态。<br>*注意： 仅可在设备模式下访问。* |
| 18 | ASVLD | A 会话有效 (A-session valid)<br>指示主机模式下收发器的状态。<br>0：A 会话无效<br>1：A 会话有效<br>*注意： 仅可在主机模式下访问。* |
| 17 | DBCT | 长/短去抖动时间 (Long/short debounce time) |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 指示已检测到连接的去抖动时间。<br>0：长去抖动时间，用于物理连接 (100 ms + 2.5 µs)<br>1：短去抖动时间，用于软连接 (2.5 µs)<br>*注意： 仅可在主机模式下访问。* |
| 16 | CIDSTS | 连接器 ID 状态 (Connector ID status)<br>指示发生连接事件时的连接器 ID 状态。<br>0：USB 控制器处于 A 器件模式<br>1：USB 控制器处于 B 器件模式<br>*注意： 在设备模式和主机模式均可访问。* |
| 15:12 | 保留 | 必须保持复位值。 |
| 11 | DHNPEN | 使能设备 HNP 特性 (Device HNP enabled)<br>从所连 USB 主机处成功接收到 SetFeature.SetHNPEnable 命令时，应用程序将该位置 1。<br>0：未在应用程序中使能 HNP<br>1：已在应用程序中使能 HNP<br>*注意： 仅可在设备模式下访问。* |
| 10 | HSHNPEN | 主机设置 HNP 使能 (Host set HNP enable)<br>（使用 SetFeature.SetHNPEnable 命令）在所连接设备上成功使能 HNP 后，应用程序将该位置 1。<br>0：未使能主机设置 HNP<br>1：已使能主机设置 HNP<br>*注意： 仅可在主机模式下访问。* |
| 9 | HNPRQ | HNP 请求 (HNP request)<br>应用程序将该位置 1 时，将对所连接 USB 主机发起 HNP 请求。当 USB_GOTGINT 寄存器中的主机协商成功状态更改位（USB_GOTGINT 中的 HNSSCHG 位）置 1 时，应用程序可通过写 0 将该位清零。HNSSCHG 位清零时，模块会将该位清零。<br>0：不发送 HNP 请求<br>1：发送 HNP 请求<br>*注意： 仅可在设备模式下访问。* |
| 8 | HNGSCS | 主机协商成功 (Host negotiation success)<br>当主机协商成功时，模块会将该位置 1。当该寄存器中的 HNP 请求位 (HNPRQ) 置 1 时，模块会将该位清零。<br>0：主机协商失败<br>1：主机协商成功<br>*注意： 仅可在设备模式下访问。* |
| 7:2 | 保留 | 必须保持复位值。 |
| 1 | SRQ | 会话请求 (Session request)<br>应用程序将该位置 1 时，将在 USB 上发起会话请求。当 USB_GOTGINT 寄存器中的主机协商成功状态更改位（USB_GOTGINT 中的 HNSSCHG 位）置 1 时，应用程序可通过写 0 将该位清零。HNSSCHG 位清零时，模块会将该位清零。<br>如要使用 USB 1.1 全速串行收发器接口来启动会话请求，则应用程序必须在该寄存器中的 B 会话有效位（USB_GOTGCTL 中的 BSVLD 位）清零后，等待 VBUS 放电至 0.2 V。该放电时间因 PHY 不同而异，可从 PHY 供应商处了解相关信息。<br>0：无会话请求<br>1：会话请求 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | *注意： 仅可在设备模式下访问。* |
| 0 | SRQSCS | 会话请求成功 (Session request success)<br>当会话请求成功时，模块会将该位置 1。<br>0：会话请求失败<br>1：会话请求成功<br>*注意： 仅可在设备模式下访问。* |

### 26.10.2. USB 中断寄存器 (GOTGINT)

USB interrupt register

**偏移地址：** 0x0004

**复位值：** 0x00000000

应用程序会在出现 OTG 中断时读取该寄存器，并通过将该寄存器中的位清零来清除 OTG 中断。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | | | | | | | | | DBCDNE | ADTOCHG | HNGDET | res. |
| | | | | | | | | | | | | rw1c | rw1c | rw1c | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | | | HNSSCHG | SRSSCHG | res. | | | | | SEDET | res. | |
| | | | | | | rw1c | rw1c | | | | | | rw1c | | |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:20 | 保留 | 必须保持复位值。 |
| 19 | DBCDNE | 去抖动完成 (Debounce done)<br>模块会在设备连接后且去抖动结束时将该位置 1。应用程序会在看见该中断后，开始驱动 USB 复位。该位仅在 USB_GUSBCFG 寄存器中的 HNP 功能位或 SRP 功能位（分别为 USB_GUSBCFG 中的 HNPCAP 位或 SRPCAP 位）置 1 时有效。<br>*注意： 仅可在主机模式下访问。* |
| 18 | ADTOCHG | A 器件超时更改 (A-device timeout change)<br>模块将该位置 1 时，指示 A 器件在等待 B 器件连接时超时。<br>*注意： 在设备模式和主机模式均可访问。* |
| 17 | HNGDET | 检测到主机协商 (Host negotiation detected)<br>当检测到 USB 上的主机协商请求时，模块会将该位置 1。<br>*注意： 在设备模式和主机模式均可访问。* |
| 16:10 | 保留 | 必须保持复位值。 |
| 9 | HNSSCHG | 主机协商成功状态更改 (Host negotiation success status change)<br>模块将在 USB 主机协商请求成功或失败时将此位置 1。应用程序必须读取 USB_GOTGCTL 寄存器中的主机协商成功位（USB_GOTGCTL 中的 HNGSCS）来检查请求成功还是失败。<br>*注意： 在设备模式和主机模式均可访问。* |
| 8 | SRSSCHG | 会话请求成功状态更改 (Session request success status change)<br>模块将在会话请求成功或失败时将此位置 1。应用程序必须读取 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | USB_GOTGCTL 寄存器中的会话请求成功位（USB_GOTGCTL 中的 SRQSCS 位）来检查请求成功还是失败。<br>*注意： 在设备模式和主机模式均可访问。* |
| 2 | SEDET | 检测到会话结束 (Session end detected)<br>模块将该位置 1 时，指示 VBUS < 0.8 V，VBUS 上的电压不再适用于 B 器件会话。 |

### 26.10.3. USB 配置寄存器 (GAHBCFG)

USB AHB configuration register

**偏移地址：** 0x0008

**复位值：** 0x00000000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | | | | PTXFELVL | TXFELVL | res. | DMAEN | BURST | | | | GINTMSK |
| | | | | | | | rw | rw | | rw | rw | | | | rw |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:20 | 保留 | 必须保持复位值。 |
| 8 | PTXFELVL | 周期性 TxFIFO 空门限 (Periodic TxFIFO empty level)<br>指示 USB_GINTSTS 寄存器中的周期性 TxFIFO 空中断位（USB_GINTSTS 中的 PTXFE 位）已触发。<br>0：PTXFE（位于 USB_GINTSTS）中断指示周期性 TxFIFO 为半空状态<br>1：PTXFE（位于 USB_GINTSTS）中断指示周期性 TxFIFO 为全空状态<br>*注意： 仅可在主机模式下访问。* |
| 7 | TXFELVL | TxFIFO 空级别 (TxFIFO empty level)<br>在设备模式下，该位指示 IN 端点发送 FIFO 空中断（USB_DIEPINTn 中的 TXFE）已触发。<br>TXFE（位于 USB_DIEPINTn）中断指示 IN 端点 TxFIFO 为半空状态<br>TXFE（位于 USB_DIEPINTn）中断指示 IN 端点 TxFIFO 为全空状态<br>在主机模式下，该位指示非周期性 Tx FIFO 空中断（USB_GINTSTS 中的 NPTXFE）已触发。<br>NPTXFE（位于 USB_GINTSTS）中断指示非周期性 Tx FIFO 为半空状态<br>NPTXFE（位于 USB_GINTSTS）中断指示非周期性 Tx FIFO 为全空状态 |
| 5 | DMAEN | DMA 使能 (DMA Enable)<br>0: 不使能 DMA<br>1: 使能 DMA |
| 4:1 | BURST | DMA Burst 类型 (DMA Burst Length/Type)<br>0: Single<br>1: INCR<br>3: INCR4<br>5: INCR8<br>7: INCR16<br>Others: Reserved |
| 0 | GINTMSK | 全局中断屏蔽 (Global interrupt mask)<br>该位用于屏蔽全局中断或对全局中断取消屏蔽。中断状态寄存器由模块进行 |

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 更新，与此位的设置无关。<br>0：屏蔽应用程序触发的中断。<br>1：取消对应用程序触发的中断的屏蔽。<br>*注意： 在设备模式和主机模式均可访问。* |

### 26.10.4. USB 配置寄存器 (GUSBCFG)

USB configuration register

**偏移地址：** 0x00C

**复位值：** 0x00000A00

该寄存器可用于在上电或更改角色模式后对模块进行配置。其中包含与 USB 和 USB-PHY 相关的配置参数。应用程序必须在开始任何 AHB 或 USB 事务前对该寄存器进行编程。请勿在初始编程后更改该寄存器。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| res. | | TRDT | | | | HNPCAP | SRPCAP | res. | PHYSEL | res. | | | TOCAL | | |
| | | rw | | | | rw | rw | | rw | | | | rw | | |

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| 30 | FDMOD | 强制设备模式 (Force device mode)<br>向该位写入 1 时，可将模块强制为设备模式，而无需考虑 USB_ID 输入引脚。<br>0：正常模式<br>1：强制设备模式<br>将强制位置 1 后，应用程序必须等待至少 25 ms 后更改方可生效。<br>*注意： 在设备模式和主机模式均可访问。* |
| 29 | FHMOD | 强制主机模式 (Force host mode)<br>向该位写入 1 时，可将模块强制为主机模式，而无需考虑 USB_ID 输入引脚。<br>0：正常模式<br>1：强制主机模式<br>将强制位置 1 后，应用程序必须等待至少 25 ms 后更改方可生效。<br>*注意： 在设备模式和主机模式均可访问。* |
| 28:14 | 保留 | 必须保持复位值。 |
| 13:10 | TRDT | USB 周转时间 (USB turnaround time)<br>以 PHY 时钟数为单位设置周转时间。<br>要计算 TRDT 的值，请使用如下公式：<br>TRDT = 4 * AHB 时钟 + 1 个 PHY 时钟<br>例如：<br>1. 如果 AHB 时钟频率 = 72 MHz（PHY 时钟频率 = 48 MHz），则 TRDT 设置为 9。<br>2. 如果 AHB 时钟频率 = 48 MHz（PHY 时钟频率 = 48 MHz），则 TRDT |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 设置为 5。<br>*注意： 仅可在设备模式下访问。* |
| 9 | HNPCAP | HNP 使能 (HNP-capable)<br>应用程序使用该位控制 USB 控制器的 HNP 功能。<br>0：不使能 HNP 功能。<br>1：使能 HNP 功能。<br>*注意： 在设备模式和主机模式均可访问。* |
| 8 | SRPCAP | SRP 使能 (SRP-capable)<br>应用程序使用该位控制 USB 控制器的 SRP 功能。如果模块作为无 SRP 功能的 B 器件，则无法请求连接的 A 器件（主机）激活 VBUS 并开始会话。<br>0：不使能 SRP 功能。<br>1：使能 SRP 功能。<br>*注意： 在设备模式和主机模式均可访问。* |
| 6 | PHYSEL | 全速系列收发器选择 (Full Speed serial transceiver select)<br>0: USB 2.0 HS PHY<br>1: USB 1.1 FS PHY。<br>*注意: 仅当前控制器支持 PHY 切换的情况下可写* |
| 5:3 | 保留 | 必须保持复位值。 |
| 2:0 | TOCAL | FS 超时校准 (FS timeout calibration)<br>PHY 引入的额外延迟包括应用程序在该字段中设置的 PHY 时钟数，以及模块的全速数据包间超时间隔。不同 PHY 引入的延迟对数据线状态的影响是不同的。<br>全速操作的 USB 标准超时值为 16 到 18（含）个位时间。应用程序必须根据枚举速度编程该字段。每个 PHY 时钟增加的位时间数为 0.25 个位时间。 |

### 26.10.5. USB 复位寄存器 (GRSTCTL)

USB reset register

**偏移地址：** 0x010

**复位值：** 0x20000000

应用程序通过此寄存器复位模块中的各项硬件特性。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | | TXFNUM | | | | | TXFFLSH | RXFFLSH | res. | FCRST | HSRST | CSRST |
| | | | | | rw | | | | | rw1s | rw1s | | rw1s | rw1s | rw1s |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31 | AHBIDL | AHB 主器件空闲 (AHB master idle)<br>指示 AHB 主器件状态机处于空闲情况。<br>*注意： 在设备模式和主机模式均可访问。* |

| 位域 | 名称 | 描述 |
|---|---|---|
| 30:11 | 保留 | 必须保持复位值。 |
| 10:6 | TXFNUM | TxFIFO 编号 (TxFIFO number)<br>使用 TxFIFO 刷新位进行 FIFO 刷新的 FIFO 编号。只有在模块将 TxFIFO 刷新位清零后，方可更改此字段。<br>00000:<br>- 主机模式下刷新非周期性 TxFIFO<br>- 设备模式下刷新 Tx FIFO 0<br>00001:<br>- 主机模式下刷新周期性 TxFIFO<br>- 设备模式下刷新 TXFIFO 1<br>00010：设备模式下刷新 TXFIFO 2<br>...<br>00101：设备模式下刷新 TXFIFO 15<br>10000：在设备模式或主机模式下刷新所有的发送 FIFO。<br>*注意： 在设备模式和主机模式均可访问。* |
| 5 | TXFFLSH | TxFIFO 刷新 (TxFIFO flush)<br>此位选择性地刷新一个或所有的发送 FIFO，但当模块处理通信事务时无法执行该操作。<br>只有在确认模块当前未对 TxFIFO 执行读写操作后，应用程序方可对此位执行写操作。使用以下寄存器进行确认：<br>读 – NAK 有效中断可确保模块当前未对 FIFO 执行读操作<br>写 – USB_GRSTCTL 中的 AHBIDL 位可确保模块当前未对 FIFO 执行任何写操作。<br>*注意： 在设备模式和主机模式均可访问。* |
| 4 | RXFFLSH | RxFIFO 刷新 (RxFIFO flush)<br>应用程序可使用此位刷新整个 RxFIFO，但必须首先确保模块当前未在处理通信事务。<br>只有在确认模块当前未对 RxFIFO 执行读写操作后，应用程序方可对此位执行写操作。<br>应用程序必须等到此位清零后，方可执行其它操作。通常需要等待 8 个时钟周期（以 PHY 或 AHB 时钟中最慢的为准）。<br>*注意： 在设备模式和主机模式均可访问。* |
| 3 | 保留 | 必须保持复位值。 |
| 2 | FCRST | 主机帧计数器复位 (Host frame counter reset)<br>应用程序对该位执行写操作时，模块中的帧数计数器复位。帧计数器复位后，由模块发送的下一个 SOF 的帧号为 0。<br>*注意： 仅可在主机模式下访问。* |
| 1 | HSRST | HCLK 软复位 (HCLK soft reset)<br>应用程序使用此位来刷新 AHB 时钟域中的控制逻辑。仅复位 AHB 时钟域流水线。<br>FIFO 不通过此位来刷新。<br>遵照协议终止 AHB 上的事务后，AHB 时钟域中的所有状态机均复位至空闲状态。<br>AHB 时钟域状态机所使用的 CSR 控制位清零。<br>要清零该中断，需要将由 AHB 时钟域状态机生成并用于控制中断状态的状态屏蔽位清零。<br>由于中断状态位并未清零，因此应用程序可以获取在该位置 1 后所发生的所有模块事件的状态。 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 此位为自清零位，模块将在其中所有必要逻辑复位后将该位清零。该过程需要若干个时钟的时间，具体取决于模块的当前状态。<br>*注意： 在设备模式和主机模式均可访问。* |
| 0 | CSRST | 模块软复位 (Core soft reset)<br>按如下所述将 HCLK 和 PCLK 域复位：<br>除以下各位外，将各个中断和所有 CSR 寄存器位清零：<br>- USB_PCGCCTL 中的 RSTPDMODL 位<br>- USB_PCGCCTL 中的 GAYEHCLK 位<br>- USB_PCGCCTL 中的 PWRCLMP 位<br>- USB_PCGCCTL 中的 STPPCLK 位<br>- USB_HCFG 中的 FSLSPCS 位<br>- USB_DCFG 中的 DSPD 位<br>将所有模块状态机（AHB 从器件除外）复位至空闲状态，并清空所有发送 FIFO 和接收 FIFO。<br>在 AHB 传输的最后数据阶段结束后，尽快终止 AHB 主器件上的所有事务。立即终止 USB 上 的所有事务。<br>应用程序可在需要复位模块时随时对该位执行写操作。该位为自清零位，模块将在其中所有必要逻辑复位后将该位清零，该过程需要若干个时钟的时间，具体取决于模块的当前状态。<br>该位一旦清零，软件必须等待至少 3 个 PHY 时钟后才可以访问 PHY 域（同步延迟）。此外，软件还必须在确定该寄存器中的位 31 置 1（AHB 主器件空闲）后方可开始运行。<br>软件复位通常在两种情况下使用，一是软件开发期间，二是用户动态更改以上所列 USB 配置寄存器中的 PHY 选择位后。用户更改 PHY 时，将为 PHY 选择相应的时钟并用于 PHY 域中。一旦选择了新的时钟，则必须复位 PHY 域，才能保证正常运行。<br>*注意： 在设备模式和主机模式均可访问。* |

### 26.10.6. USB 模块中断寄存器 (GINTSTS)

USB core interrupt register

**偏移地址：** 0x014

**复位值：** 0x04000020

该寄存器用于在当前模式（设备模式或主机模式）下借助系统级别的事件来中断应用程序。该寄存器中的某些位仅在主机模式下有效，而其它位则仅在设备模式下有效。此外，该寄存器还可指示当前模式。要将 rc_w1 类型中断状态位清零，应用程序必须向该位写入 1。FIFO 状态中断为只读；如果软件在处理这些中断期间对 FIFO 执行读写操作，则 FIFO 中断标志将自动清零。使能中断位前，应用程序必须在初始化时将 USB_GINTSTS 寄存器清零，才可以避免在初始化前产生任何中断。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WKUPINT | SRQINT | DISCINT | CIDSCHG | res. | PTXFE | HCINT | HPRTINT | res. | | IPXFR | IISOIXFR | OEPINT | IEPINT | res. | |
| rw1c | rw1c | rw1c | rw1c | | r | r | r | | | rw1c | r | r | r | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EOPF | ISOODRP | ENUMDNE | USBRST | USBSUSP | ESUSP | res. | | GONAKEFF | GINAKEFF | NPTXFE | RXFLVL | SOF | OTGINT | MMIS | CMOD |
| rw1c | rw1c | rw1c | rw1c | rw1c | rw1c | | | r | r | r | r | rw1c | r | rw1c | r |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31 | WKUPINT | 检测到恢复/远程唤醒中断 (Resume/remote wakeup detected interrupt)<br>在设备模式下，当 USB 总线上检测到恢复信号时，将触发该中断。在主机模式下，当 USB 上检测到远程唤醒时，将触发该中断。<br>*注意： 在设备模式和主机模式均可访问。* |
| 30 | SRQINT | 检测到会话请求/新会话中断 (Session request/new session detected interrupt)<br>在主机模式下，当检测到来自设备的会话请求时，将触发该中断。在设备模式下，当 VBUS<br>在 B 外设设备的有效范围内时，将触发该中断。在设备模式和主机模式均可访问。 |
| 29 | DISCINT | 检测到断开连接中断 (Disconnect detected interrupt)<br>当检测到设备断开连接时触发该中断。<br>*注意： 仅可在主机模式下访问。* |
| 28 | CIDSCHG | 连接器 ID 线状态更改 (Connector ID status change)<br>当连接器 ID 线状态发生更改时，模块将该位置 1。<br>*注意： 在设备模式和主机模式均可访问。* |
| 27 | 保留 | 必须保持复位值。 |
| 26 | PTXFE | 周期性 TxFIFO 空 (Periodic TxFIFO empty)<br>当周期性发送 FIFO 为半空或全空状态，且周期性请求队列中存在可写入至少一个条目的空间时，将触发该中断。该 FIFO 为半空状态还是全空状态由 USB_GAHBCFG 寄存器中的周期性 TxFIFO 空级别位（USB_GAHBCFG 中的 PTXFELVL 位）决定。<br>*注意： 仅可在主机模式下访问。* |
| 25 | HCINT | 主机通道中断 (Host channels interrupt)<br>模块将该位置 1 时，指示模块中一个通道上存在挂起的中断（在主机模式下）。应用程序必须读取主机 USB_HAINT 寄存器，以确定发生中断的通道的准确编号，然后读取相应的 USB_HCINTn 寄存器，以确定引发中断的确切原因。应用程序必须先将 USB_HCINTn 寄存器的相应状态位清零，之后才能将该位清零。<br>*注意： 仅可在主机模式下访问。* |
| 24 | HPRTINT | 主机端口中断 (Host port interrupt)<br>模块将该位置 1 时，指示主机模式下 USB 控制器端口的状态发生变化。应用程序必须读取 USB_HPRT 寄存器，以确定引发此中断的确切事件。应用程序必须先将 USB_HPRT 寄存器的相应状态位清零，之后才能将该位清零。<br>*注意： 仅可在主机模式下访问。* |
| 23:22 | 保留 | 必须保持复位值。 |
| 21 | IPXFR | 未完成周期性传输 (Incomplete periodic transfer)<br>在主机模式下，如果存在仍处于挂起状态的未完成周期性事务，而这些事务计划在当前帧期间完成，则模块会将该中断位置 1。<br>INCOMPISOOUT：未完成 OUT 同步传输 (Incomplete isochronous OUT transfer 在设备模式下，模块将该中断置 1 时，指示当前帧中至少有一个同步 OUT 端点上的传输未完成。该中断随该寄存器中的周期性帧结束中断 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | (EOPF) 位一同触发。 |
| 20 | IISOIXFR | 未完成 IN 同步传输 (Incomplete isochronous IN transfer)<br>模块将该中断置 1 时，指示当前帧中至少有一个同步 IN 端点上的传输未完成。该中断随该寄存器中的周期性帧结束中断 (EOPF) 位一同触发。<br>*注意： 仅可在设备模式下访问。* |
| 19 | OEPINT | OUT 端点中断 (OUT endpoint interrupt)<br>模块将该位置 1 时，指示模块中一个 OUT 端点上存在挂起的中断（在设备模式下）。应用程序必须读取主机 USB_DAINT 寄存器，以确定发生中断的 OUT 端点的准确编号，然后读取相应的 USB_DOEPTSIZn 寄存器，以确定引发中断的确切原因。应用程序必须先将相应 USB_DOEPTSIZn 寄存器的相应状态位清零，之后才能将该位清零。<br>*注意： 仅可在设备模式下访问。* |
| 18 | IEPINT | IN 端点中断 (IN endpoint interrupt)<br>模块将该位置 1 时，指示模块中一个 IN 端点上存在挂起的中断（在设备模式下）。应用程序必须读取主机 USB_DAINT 寄存器，以确定发生中断的 IN 端点的准确编号，然后读取相应的 USB_DIEPINTn 寄存器，以确定引发中断的确切原因。应用程序必须先将相应 USB_DIEPINTn 寄存器的相应状态位清零，之后才能将该位清零。<br>*注意： 仅可在设备模式下访问。* |
| 17:16 | 保留 | 必须保持复位值。 |
| 15 | EOPF | 周期性帧结束中断 (End of periodic frame interrupt)<br>指示当前帧已达到 USB_DCFG 寄存器中周期性帧间隔字段（USB_DCFG 中的 PFIVL 位）所指定的周期。<br>*注意： 仅可在设备模式下访问。* |
| 14 | ISOODRP | 丢弃同步 OUT 数据包中断 (Isochronous OUT packet dropped interrupt)<br>如果由于 RxFIFO 空间不足，无法容纳同步 OUT 端点的最大数据包，从而导致模块无法向 RxFIFO 写入同步 OUT 数据包，模块会将该位置 1。<br>*注意： 仅可在设备模式下访问。* |
| 13 | ENUMDNE | 枚举完成 (Enumeration done)<br>模块将该位置 1 时，指示速度枚举已完成。应用程序必须读取 USB_DSTS 寄存器来获取枚举速度。<br>*注意： 仅可在设备模式下访问。* |
| 12 | USBRST | USB 复位 (USB reset)<br>模块将该位置 1 时，指示在 USB 上检测到复位信号。<br>*注意： 仅可在设备模式下访问。* |
| 11 | USBSUSP | USB 挂起 (USB suspend)<br>模块将该位置 1 时，指示在 USB 上检测到挂起状态。当 USB 总线上的空闲状态保持 3 ms，模块便会进入挂起状态。<br>*注意： 仅可在设备模式下访问。* |
| 10 | ESUSP | 早期挂起 (Early suspend)<br>模块将该位置 1 时，指示已检测到 USB 处于空闲状态的时间达到 3 ms。<br>*注意： 仅可在设备模式下访问。* |
| 9:8 | 保留 | 必须保持复位值。 |
| 7 | GONAKEFF | 全局 OUT NAK 有效 (Global OUT NAK effective)<br>指示 USB_DCTL 寄存器中由应用程序设置的“将全局 OUT NAK 置 1”位（USB_DCTL 中的 SGONAK 位）已在模块中生效。通过写入 USB_DCTL 寄存器中的“将全局 OUT NAK 清零”位（USB_DCTL 中的 CGONAK 位），可将该位清零。 |

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| | | *注意： 仅可在设备模式下访问。* |
| 6 | GINAKEFF | 全局非周期性 IN NAK 有效 (Global IN nonperiodic NAK effective)<br>指示 USB_DCTL 寄存器中由应用程序设置的“将全局非周期性 IN NAK 置 1”位（USB_DCTL 中的 SGINAK 位）已在模块中生效。也就是说，模块已对应用程序设置的全局 IN NAK 位进行采样，结果已生效。通过清零 USB_DCTL 寄存器中的“将全局非周期性 IN NAK 清零”位（USB_DCTL 中的 CGINAK 位），可将该位清零。<br>此中断不一定表示 USB 上已发送了一个 NAK 握手信号。STALL 位优先级高于 NAK 位。<br>*注意： 仅可在设备模式下访问。* |
| 5 | NPTXFE | 非周期性 TxFIFO 空 (Non-periodic TxFIFO empty)<br>当非周期性 TxFIFO 为半空或全空状态，且非周期性发送请求队列中至少存在可写入一个条目的空间时，将触发该中断。该 FIFO 为半空状态还是全空状态由 USB_GAHBCFG 寄存器中的非周期性 TxFIFO 空级别位（USB_GAHBCFG 中的 TXFELVL 位）决定。<br>*注意： 仅可在主机模式下访问。* |
| 4 | RXFLVL | RxFIFO 非空 (RxFIFO non-empty)<br>指示 RxFIFO 中至少有一个数据包等待读取。<br>*注意： 在主机模式和设备模式均可访问。* |
| 3 | SOF | 帧起始 (Start of frame)<br>在主机模式下，模块将该位置 1 时，指示 USB 上已发送一个 SOF (FS) 或 Keep-Alive (LS)信号。应用程序必须将此位置 1 才可清除该中断。<br>在设备模式下，模块将该位置 1 时，指示 USB 上已接收到一个 SOF 令牌。应用程序可通过读取设备状态寄存器来获得当前的帧编号。只有在模块以 FS 模式运行时，才会出现此中断。<br>*注意： 在主机模式和设备模式均可访问。* |
| 2 | OTGINT | OTG 中断 (OTG interrupt)<br>模块将该位置 1 时，指示出现 OTG 协议事件。应用程序必须读取 OTG 中断状态(USB_GOTGINT) 寄存器，以确定引发此中断的确切事件。应用程序必须先将<br>USB_GOTGINT 寄存器的相应状态位清零，之后才能将该位清零。<br>*注意： 在主机模式和设备模式均可访问。* |
| 1 | MMIS | 模式不匹配中断 (Mode mismatch interrupt)<br>当应用程序尝试访问以下寄存器时，模块将该位置 1：<br>- 模块运行在设备模式下访问主机模式寄存器<br>- 模块运行在主机模式下访问设备模式寄存器<br>寄存器访问在 AHB 上以 OKAY 响应结束，但该访问在内部被模块忽略并且不会影响模块运行。<br>*注意： 在主机模式和设备模式均可访问。* |
| 0 | CMOD | 当前工作模式 (Current mode of operation)<br>指示当前模式。<br>0：设备模式<br>1：主机模式<br>*注意： 在主机模式和设备模式均可访问。* |

### 26.10.7. USB 中断屏蔽寄存器 (GINTMSK)

USB interrupt mask register

**偏移地址：** 0x018

**复位值：** 0x00000000

该寄存器与模块中断寄存器结合使用，以中断应用程序。如果将某个中断位屏蔽，则不会产生与该位相关的中断。但是，与该中断相对应的模块中断 (USB_GINTSTS) 寄存器位仍会置 1。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WUIM | SRQIM | DISCINT | CIDSCHGM | res. | PTXFEM | HCIM | PRTIM | res. | | IPXFRM | IISOIXFRM | OEPINT | IEPINT | EPMISM | res. |
| rw | rw | rw | rw | | rw | rw | rw | | | rw | rw | rw | rw | rw | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EOPFM | ISOODRPM | ENUMDNEM | USBRST | USBSUSPM | ESUSPM | res. | | GONAKEFFM | GINAKEFFM | NPTXFEM | RXFLVLM | SOFM | OTGINT | MMISM | res. |
| rw | rw | rw | rw | rw | rw | | | rw | rw | rw | rw | rw | rw | rw | |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31 | WUIM | 检测到恢复/远程唤醒中断屏蔽 (Resume/remote wakeup detected interrupt mask)<br>0：屏蔽中断<br>1：使能中断<br>*注意： 在主机模式和设备模式均可访问。* |
| 30 | SRQIM | 检测到会话请求/ 新会话中断屏蔽 (Session request/new session detected interrupt mask)<br>0：屏蔽中断<br>1：使能中断<br>*注意： 在主机模式和设备模式均可访问。* |
| 29 | DISCINT | 检测到断开连接中断屏蔽 (Disconnect detected interrupt mask)<br>0：屏蔽中断<br>1：使能中断<br>*注意： 仅可在设备模式下访问。* |
| 28 | CIDSCHGM | 连接器 ID 状态更改屏蔽 (Connector ID status change mask)<br>0：屏蔽中断<br>1：使能中断<br>*注意： 在主机模式和设备模式均可访问。* |
| 27 | 保留 | 必须保持复位值。 |
| 26 | PTXFEM | 周期性 TxFIFO 空屏蔽 (Periodic TxFIFO empty mask)<br>0：屏蔽中断<br>1：使能中断<br>*注意： 仅可在主机模式下访问。* |
| 25 | HCIM | 主机通道中断屏蔽 (Host channels interrupt mask)<br>0：屏蔽中断<br>1：使能中断<br>*注意： 仅可在主机模式下访问。* |
| 24 | PRTIM | 主机端口中断屏蔽 (Host port interrupt mask)<br>0：屏蔽中断<br>1：使能中断<br>*注意： 仅可在主机模式下访问。* |

| 位域 | 名称 | 描述 |
|---|---|---|
| 23:22 | 保留 | 必须保持复位值。 |
| 21 | IPXFRM | 未完成周期性传输中断屏蔽 (Incomplete periodic transfer mask)<br>0：屏蔽中断<br>1：使能中断<br>*注意： 仅可在主机模式下访问。*<br>IISOOXFRM：未完成 OUT 同步传输中断屏蔽 (Incomplete isochronous OUT transfer mask)<br>0：屏蔽中断<br>1：使能中断<br>*注意： 仅可在设备模式下访问。* |
| 20 | IISOIXFRM | 未完成 IN 同步传输中断屏蔽 (Incomplete isochronous IN transfer mask)<br>0：屏蔽中断<br>1：使能中断<br>*注意： 仅可在设备模式下访问。* |
| 19 | OEPINT | OUT 端点中断屏蔽 (OUT endpoints interrupt mask)<br>0：屏蔽中断<br>1：使能中断<br>*注意： 仅可在设备模式下访问。* |
| 18 | IEPINT | IN 端点中断屏蔽 (IN endpoints interrupt mask)<br>0：屏蔽中断<br>1：使能中断<br>*注意： 仅可在设备模式下访问。* |
| 17 | EPMISM | 端点不匹配中断屏蔽 (Endpoint mismatch interrupt mask)<br>0：屏蔽中断<br>1：使能中断<br>*注意： 仅可在设备模式下访问。* |
| 16 | 保留 | 必须保持复位值。 |
| 15 | EOPFM | 周期性帧结束中断屏蔽 (End of periodic frame interrupt mask)<br>0：屏蔽中断<br>1：使能中断<br>*注意： 仅可在设备模式下访问。* |
| 14 | ISOODRPM | 丢弃同步 OUT 数据包中断屏蔽 (Isochronous OUT packet dropped interrupt mask)<br>0：屏蔽中断<br>1：使能中断<br>*注意： 仅可在设备模式下访问。* |
| 13 | ENUMDNEM | 枚举完成中断屏蔽 (Enumeration done mask)<br>0：屏蔽中断<br>1：使能中断<br>*注意： 仅可在设备模式下访问。* |
| 12 | USBRST | USB 复位中断屏蔽 (USB reset mask)<br>0：屏蔽中断<br>1：使能中断<br>*注意： 仅可在设备模式下访问。* |
| 11 | USBSUSPM | USB 挂起中断屏蔽 (USB suspend mask)<br>0：屏蔽中断<br>1：使能中断<br>*注意： 仅可在设备模式下访问。* |

| 位域 | 名称 | 描述 |
|---|---|---|
| 10 | ESUSPM | 早期挂起中断屏蔽 (Early suspend mask)<br>0：屏蔽中断<br>1：使能中断<br>*注意： 仅可在设备模式下访问。* |
| 9:8 | 保留 | 必须保持复位值。 |
| 7 | GONAKEFFM | 全局 OUT NAK 生效中断屏蔽 (Global OUT NAK effective mask)<br>0：屏蔽中断<br>1：使能中断<br>*注意： 仅可在设备模式下访问。* |
| 6 | GINAKEFFM | 全局非周期性 IN NAK 生效中断屏蔽 (Global non-periodic IN NAK effective mask)<br>0：屏蔽中断<br>1：使能中断<br>*注意： 仅可在设备模式下访问。* |
| 5 | NPTXFEM | 非周期性 TxFIFO 空中断屏蔽 (Non-periodic TxFIFO empty mask)<br>0：屏蔽中断<br>1：使能中断<br>*注意： 仅可在主机模式下访问。* |
| 4 | RXFLVLM | 接收 FIFO 非空中断屏蔽 (Receive FIFO nonempty mask)<br>0：屏蔽中断<br>1：使能中断<br>*注意： 在设备模式和主机模式均可访问。* |
| 3 | SOFM | 帧起始中断屏蔽 (Start of frame mask)<br>0：屏蔽中断<br>1：使能中断<br>*注意： 在设备模式和主机模式均可访问。* |
| 2 | OTGINT | OTG 中断屏蔽 (OTG interrupt mask)<br>0：屏蔽中断<br>1：使能中断<br>*注意： 在设备模式和主机模式均可访问。* |
| 1 | MMISM | 模式不匹配中断屏蔽 (Mode mismatch interrupt mask)<br>0：屏蔽中断<br>1：使能中断<br>*注意： 在设备模式和主机模式均可访问。* |

### 26.10.8. USB 接收状态读取寄存器 (GRXSTSR)

USB Receive status read registers

**偏移地址：** 0x01C

**复位值：** 0x00000000

读取接收状态调试读取寄存器将返回接收 FIFO 顶部的内容。读取接收状态读取和出栈寄存器将额外弹出 RxFIFO 顶部的数据条目。接收状态内容在主机模式和设备模式下的解释不同。当接收 FIFO 为空时，模块会忽略对该寄存器的读取或出栈操作，并返回值 0x0000 0000。仅当模块中断寄存器的接收 FIFO 非空位（OTG_FS_GINTSTS 中的 RXFLVL 位）置位时，应用程序才能弹出接收状态 FIFO。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | | | | FRMNUM | | | | PKTSTS | | | | DPID |
| | | | | | | | r | | | | r | | | | r |

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 24:21 | FRMNUM | 帧编号 (Frame number)<br>这是在 USB 上接收的数据包帧编号的 4 个最低有效位。只有当支持同步 OUT 端点时才支持此字段。 |
| 20:17 | PKTSTS | 数据包状态 (Packet status)<br>指示接收的数据包的状态<br>0001：全局 OUT NAK（触发中断）<br>0010：接收到 OUT 数据包<br>0011：OUT 传输完成（触发中断）<br>0100：SETUP 事务完成（触发中断）<br>0110：接收到 SETUP 数据包<br>其它值：保留 |
| 16:15 | DPID | 数据 PID (Data PID)<br>指示接收的 OUT 数据包的数据 PID<br>00：DATA0<br>10：DATA1<br>01：DATA2<br>11：MDATA |
| 14:4 | BCNT | 字节计数 (Byte count)<br>指示接收的数据包的字节数。 |
| 3:0 | EPNUM | 端点编号 (Endpoint number)<br>指示当前接收的数据包所属的端点编号。 |

### 26.10.9. USB 接收状态出栈寄存器 (GRXSTSP)

USB Receive status pop registers

**偏移地址：** 0x020

寄存器结构与 GRXSTSR 一致，差别仅为读取寄存器意味着从 RX FIFO 中对栈顶状态包进行出栈操作。

### 26.10.10. USB 接收 FIFO 大小寄存器 (GRXFSIZ)

USB Receive FIFO size register

**偏移地址：** 0x024

**复位值：** -

此应用程序可以对必须分配给 RxFIFO 的 RAM 大小进行编程。

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 15:0 | RXFDEP | 接收 FIFO 深度 (RxFIFO Depth)<br>以四字节为单位进行配置，实际可配置区间受限于 USB 控制器总 FIFO 大小。 |

### 26.10.11. USB 非周期发送 FIFO 配置寄存器 (GNPTXFSIZ)

USB non-periodic transmit FIFO size register

**偏移地址：** 0x028

**复位值：** -

非周期 FIFO 配置寄存器，用于配置发送 FIFO 0 的起始地址与深度，FIFO 0 一般与端点 0 进行绑定。

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

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | TXFDEP | 发送 FIFO 深度 (TxFIFO depth)<br>以 32 位字为单位。<br>以四字节为单位进行配置，实际可配置区间受限于 USB 控制器总 FIFO 大小。 |
| 15:0 | TXFADDR | 发送 FIFO 起始地址 (TxFIFO start address)<br>以四字节为单位的起始地址。 |

### 26.10.12. USB 非周期性发送 FIFO/队列状态寄存器 (GNPTXSTS)

USB nonperiodic transmit FIFO/queue status register

**偏移地址：** 0x02C

**复位值：** 0x00080200

此只读寄存器包含非周期性 TxFIFO 和非周期性发送请求队列的空闲空间信息

### 26.10.13. USB 通用模块配置寄存器 (GCFG)

USB global configuration register

**偏移地址：** 0x038

**复位值：** 0x00000000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | | | | | | | NOVBUSSENS | SOFOUTEN | VBUSBSEN | VBUSASEN | res. | PWRDWN |
| | | | | | | | | | | rw | rw | rw | rw | | rw |

| 位域 | 名称 | 描述 |
|---|---|---|
| 21 | NOVBUSSENS | VBUS 感应禁止选项 (VBUS sensing disable option)<br>该位置 1 时，在内部将 VBUS 视为始终位于 VBUS 有效电平 (5 V)。该选项消除了对 VBUS 引脚的需求，使该引脚可自由用于其它用途，例如该引脚上的其他功能。VBUS 连接可重映射到其它通用输入引脚，并由软件监视。此选项仅适用于只作主机或只作设备。<br>0：硬件支持 VBUS 感应<br>1：硬件不支持 VBUS 感应 |
| 20 | SOFOUTEN | SOF 输出使能 (SOF output enable)<br>0：SOF 脉冲不从引脚输出<br>1：SOF 脉冲可从引脚输出 |
| 19 | VBUSBSEN | 使能“B”器件的 VBUS 感应功能 (Enable the VBUS sensing “B” device)<br>0：禁止“B”器件的 VBUS 感应功能<br>1：使能“B”器件的 VBUS 感应功能 |
| 18 | VBUSASEN | 使能“A”器件的 VBUS 感应功能 (Enable the VBUS sensing “A” device)<br>0：禁止“A”器件的 VBUS 感应功能<br>1：使能“A”器件的 VBUS 感应功能 |
| 16 | PWRDWN | 掉电 (Power down)<br>用于在发送/接收时激活收发器<br>0：掉电激活<br>1：掉电停用（“收发器激活”） |

### 26.10.14. USB 主机周期性发送 FIFO 大小寄存器 (HPTXFSIZ)

USB Host periodic transmit FIFO size register

**偏移地址：** 0x100

**复位值：** 0x02000600

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

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | HPTFD | 主机周期性 TxFIFO 深度 (Host periodic TxFIFO depth)<br>以 32 位字为单位。<br>最小值为 16 |
| 15:0 | HPTFA | 主机周期性 TxFIFO 起始地址 (Host periodic TxFIFO start address)<br>上电复位值是最大 Rx 数据 FIFO 深度与最大非周期性 Tx 数据 FIFO 深 |

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 度之和。 |

### 26.10.15. USB 设备 IN 端点发送 FIFO 大小寄存器 (DIEPTXFn)

USB device IN endpoint transmit FIFO size register

**偏移地址：** 0x104 + n * 0x04 (n: TxFIFO index - 1)

**复位值：** 0x02000400

发送端点 TxFIFO 1 ~ n 的配置由此寄存器进行配置，TxFIFO 0 的配置见 GNPTXFSIZ。

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

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

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:16 | EPTXFD | 设备端点 TxFIFO 深度 (Device Endpoint TxFIFO n Depth)<br>以四字节为单位。 |
| 15:0 | EPTXA | 设备端点 TxFIFO 起始地址 (Device Endpoint TxFIFO n start address)<br>以四字节为单位。 |

### 26.10.16. USB 主机配置寄存器 (HCFG)

USB host configuration register

**偏移地址：** 0x400

**复位值：** 0x00000000

此寄存器将在上电后对模块进行配置。请勿在初始化主机后更改此寄存器。

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

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:3 | 保留 | 必须保持复位值。 |
| 2 | FSLSS | 仅支持 FS 和 LS (FS- and LS-only support)<br>应用程序使用此位控制模块的枚举速度。使用此位，应用程序可使模块工作为 FS 主机，即使所连接的设备支持 HS 通信也是如此。请勿在初始编程后更改此字段。<br>1：仅限 FS/LS，即使所连接设备可支持 HS（只读） |
| 1:0 | FSLSPCS | FS/LS PHY 时钟选择 (FS/LS PHY clock select)<br>当模块处于 FS 主机模式时<br>01：PHY 时钟以 48 MHz 运行<br>其它值：保留<br>当模块处于 LS 主机模式时 |

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 00：保留<br>01：选择 48 MHz PHY 时钟频率<br>10：选择 6 MHz PHY 时钟频率<br>11：保留<br>注意： 当设备连上主机时，必须依照所连接设备的速度设置 FSLSPCS（更改此位后，必须进行<br>软件复位）。 |

### 26.10.17. USB 主机帧间隔寄存器 (HFIR)

USB Host frame interval register

**偏移地址：** 0x404

**复位值：** 0x0000EA60

此寄存器用于存储 USB 控制器对已连接设备当前速度所设定的帧间隔信息。

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

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| 15:0 | FRIVL | 帧间隔 (Frame interval)<br>应用程序在此字段编程的值用于指定两个连续 SOF (FS) 或 Keep-Alive 令牌 (LS) 之间的时间间隔。此字段包含构成所需帧间隔的 PHY 时钟数。只有将主机端口控制和状态寄存器的端口使能位（USB_HPRT 的 PENA 位）置 1 后，应用程序才能向此寄存器中写入值。如果未对值进行编程，模块将根据在主机配置寄存器的 FS/LS PHY 时钟选择字段（USB_HCFG 中的 FSLSPCS）中指定的 PHY 时钟来计算。请勿在初始配置后更改此字段的值。<br>1 ms *（PHY 时钟频率） |

### 26.10.18. USB 主机帧编号/帧剩余时间寄存器 (HFNUM)

USB host frame number/frame time remaining register

**偏移地址：** 0x408

**复位值：** 0x00003FFF

此寄存器用于指示当前帧编号。它还指示当前帧的剩余时间（以 PHY 时钟数为单位）。

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

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

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:16 | FTREM | 帧剩余时间 (Frame time remaining) |

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 指示当前帧的剩余时间（以 PHY 时钟数为单位）。每过去 1 个 PHY 时钟，此字段递减 1。当值达到零时，此字段将重新装载帧间隔寄存器中的值，并由模块在 USB 上发送一个新 SOF。 |
| 15:0 | FRNUM | 帧编号 (Frame number)<br>当在 USB 上发送 1 个新 SOF 时此字段的值将递增 1，当达到 0x3FFF 时会清零。 |

### 26.10.19. USB Host 周期性发送 FIFO/队列状态寄存器 (HPTXSTS)

USB Host periodic transmit FIFO/queue status register

**偏移地址：** 0x410

**复位值：** 0x00080100

此只读寄存器包含周期性 TxFIFO 和周期性发送请求队列的空闲空间信息。

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

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

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:24 | PTXQTOP | 周期性发送请求队列顶部 (Top of the periodic transmit request queue)<br>指示周期性 Tx 请求队列中 MAC 当前正在处理的项。<br>该寄存器用于调试。<br><br>31：奇数/偶数帧 (Odd/Even frame)<br>- 0：以偶数帧发送<br>- 1：以奇数帧发送<br><br>30:27：通道/端点编号 (Channel/endpoint number)<br><br>26:25：类型 (Type)<br>- 00：输入/输出<br>- 01：零长度数据包<br>- 11：禁止通道命令<br><br>24：结束（所选通道/端点的最后一个条目）(Terminate (last entry for the selected<br>channel/endpoint)) |
| 23:16 | PTXQSAV | 周期性发送请求队列可用空间 (Periodic transmit request queue space available)<br>指示可供写入的周期性发送请求队列的空闲位置的数量。该队列既包含 IN 请求，又包含 OUT 请求。<br>00：周期性发送请求队列已满<br>01：1 个位置可用 |

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 10：2 个位置可用<br>bxn：n 个位置可用 (0 ≤ n ≤ 8)<br>其它值：保留 |
| 15:0 | PTXFSAVL | 周期性发送数据 FIFO 可用空间 (Periodic transmit data FIFO space available)<br>指示可供写入的周期性 TxFIFO 的空闲位置的数量。<br>以 32 位字为单位<br>0000：周期性 TxFIFO 已满<br>0001：1 个字可用<br>0010：2 个字可用<br>bxn：n 个字可用（其中 0 ≤ n ≤ PTXFD）<br>其它值：保留 |

## 26.10.20. USB 主机全体通道中断寄存器 (HAINT)

USB Host all channels interrupt register

**偏移地址：** 0x414

**复位值：** 0x00000000

当通道上有事件发生时，主机全体通道中断寄存器会使用模块中断寄存器中的主机通道中断位（USB_GINTSTS 中的 HCINT 位）中断应用程序。相关内容如图 383 所示。每个通道对应 1 个中断位，最多有 16 个位。当应用程序通过相应主机通道中断寄存器 n 清零中断时，该寄存器中的位也会清零。

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

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:16 | | 保留，必须保持复位值。 |
| 15:0 | HAINT | 通道中断 (Channel interrupt)<br>每个通道对应一位：通道 0 对应位 0，通道 15 对应位 15 |

## 26.10.21. USB 主机全体通道中断屏蔽寄存器 (HAINTMSK)

USB host all channels interrupt mask register

**偏移地址：** 0x418

**复位值：** 0x00000000

主机全体通道中断屏蔽寄存器与主机全体通道中断寄存器结合使用，进而在通道上发生事件时中断应用程序。每个通道对应 1 个中断屏蔽位，最多有 16 个位。

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

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:16 | | 保留，必须保持复位值。 |
| 15:0 | HAINTM | 通道中断屏蔽 (Channel interrupt mask)<br>0：屏蔽中断 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 1：使能中断<br>每个通道对应一位：通道 0 对应位 0，通道 15 对应位 15 |

### 26.10.22. USB 主机端口控制和状态寄存器 (HPRT)

USB host port control and status register

**偏移地址：** 0x440

**复位值：** 0x00000000

该寄存器仅在主机模式下可用。当前，USB 主机仅支持一个端口。该寄存器包含与 USB 端口相关的信息，例如 USB 复位、使能、挂起、恢复、连接状态以及测试模式。该寄存器中的 rc_w1 位可通过模块中断寄存器中的主机端口中断位（USB_GINTSTS 中的 HPRTINT 位）触发应用程序中断。发生端口中断时，应用程序必须读取该寄存器，并将引起中断的位清零。对于 rc_w1 位，应用程序必须向该位写入 1 以清除该中断。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | | | | | | | | | | PSPD | | PTCTL |
| | | | | | | | | | | | | | r | | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PTCTL | | | PPWR | PLSTS | | res. | PRST | PSUSP | PRES | POCCHNG | POCA | PENCHNG | PENA | PCDET | PCSTS |
| rw | | | rw | r | | r | rw | rw1s | rw | rw1c | r | rw1c | rw1c | rw1c | r |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:19 | | 保留，必须保持复位值。 |
| 18:17 | PSPD | 端口速度 (Port speed)<br>指示连接到该端口的设备的速度。<br>01：全速<br>10：低速<br>11：保留 |
| 16:13 | PTCTL | 端口测试控制 (Port test control)<br>应用程序向该字段写入一个非零值，以将端口置于测试模式，同时端口上会产生对应模式的信号。<br>0000：测试模式禁止<br>0001：Test_J 模式<br>0010：Test_K 模式<br>0011：Test_SE0_NAK 模式<br>0100：Test_Packet 模式<br>0101：Test_Force_Enable<br>其它值：保留 |
| 12 | PPWR | 端口电源 (Port power)<br>应用程序使用该字段控制该端口的电源，且发生过流情况时，模块会将该位清零。<br>0：掉电<br>1：通电 |
| 11:10 | PLSTS | 端口线状态 (Port line status) |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 指示 USB 数据线的当前逻辑电平<br>10 : USB_FS_DP 的逻辑电平 (Logic level of USB_FS_DP)<br>11 : USB_FS_DM 的逻辑电平 (Logic level of USB_FS_DM) |
| 9 | | 保留，必须保持复位值。 |
| 8 | PRST | 端口复位 (Port reset)<br>应用程序将该位置 1 时，会在该端口上启动复位序列。应用程序必须为复位周期定时，并在复位序列完成后将该位清零。<br>0：端口未处于复位状态<br>1：端口处于复位状态<br>应用程序必须将该位置 1 并最少保持 10 ms，以在端口上启动复位。除所需的最少持续时间之外，在将该位清零前，应用程序可将该位置 1 的状态再保持 10 ms，即使 USB 标准并没有设置最大限制。 |
| 7 | PSUSP | 端口挂起 (Port suspend)<br>应用程序将此位置 1 以将此端口置于挂起模式。只有此位置 1 时，模块才会停止发送 SOF。要停止 PHY 时钟，应用程序必须将端口时钟停止位置 1，这会使能 PHY 的挂起输入引脚。<br>此位的读取值反映该端口的当前挂起状态。检测到远程唤醒信号，或者应用程序将此寄存器中的端口复位位或端口恢复位置 1 后，模块可将此位清零；或应用程序将模块中断寄存器中的恢复/ 远程唤醒检测中断位或断开连接检测中断位（分别为 USB_GINTSTS 中的 WKUINT 或 DISCINT）置 1，模块也可将此位清零。<br>0：端口未处于挂起模式<br>1：端口处于挂起模式 |
| 6 | PRES | 端口恢复 (Port resume)<br>应用程序将此位置 1 以在该端口上驱动恢复信号。模块会持续驱动恢复信号直到应用程序将此位清零。<br>如模块中断寄存器中的端口恢复/ 远程唤醒检测中断位（USB_GINTSTS 中 WKUINT 位）指示，如果模块检测到 USB 远程唤醒序列，则开始驱动恢复信号，而无需应用程序进行干预；如果模块检测到断开连接的情况，则将此位清零。此位的读取值指示当前模块是否正在驱动恢复信号。<br>0：不驱动恢复信号<br>1：驱动恢复信号 |
| 5 | POCCHNG | 端口过流变化 (Port overcurrent change)<br>该寄存器中端口过流激活位（位 4）状态发生变化时，模块将此位置 1。 |
| 4 | POCA | 端口过流激活 (Port overcurrent active)<br>此位指示端口的过流状况。<br>0：无过流状况<br>1：有过流状况 |
| 3 | PENCHNG | 端口使能/禁止变化 (Port enable/disable change)<br>该寄存器中的端口使能位 2 的状态发生变化时，模块将此位置 1。 |
| 2 | PENA | 端口使能 (Port enable)<br>端口执行复位序列后，只能由模块使能，并且可以由过流状况、断开连接状况或应用程序将此位清零来禁止。应用程序无法通过对寄存器执行写操作将此位置 1。只能将此位清零来禁止端口。对此位的操作不会触发应用程序的任何中断。<br>0：禁止端口<br>1：使能端口 |
| 1 | PCDET | 检测到端口连接 (Port connect detected) |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 当检测到设备连接时，模块将此位置 1，以使用模块中断寄存器中的主机端口中断位（USB_GINTSTS 中的 HPRTINT 位）触发应用程序的中断。应用程序必须将此位置 1 才可清除该中断。 |
| 0 | PCSTS | 端口连接状态 (Port connect status)<br>0：端口未连接设备<br>1：端口已连接设备 |

### 26.10.23. USB 主机通道特性寄存器 n (HCCHARn)

USB host channel characteristics register n

**偏移地址：** 0x500 + (ChannelIndex * 0x20)

**复位值：** 0x00000000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CHENA | CHDIS | ODDFRM | DAD | | | | | | | MCNT | | EPTYP | | LSDEV | res. |
| rw1s | rw1s | rw | rw | | | | | | | rw | | rw | | rw | |

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 31 | CHENA | 通道使能 (Channel enable)<br>此字段由应用程序软件置 1，并由 OTG 主机硬件清零。<br>0：禁止通道 (Channel disabled)<br>1：使能通道 |
| 30 | CHDIS | 通道禁止 (Channel disable)<br>应用程序将此位置 1 以停止通过通道发送/接收数据，即使通过该通道的传输还未完成，停止操作仍然生效。应用程序必须等待禁止通道的中断以确认通道已经被禁止。 |
| 29 | ODDFRM | 奇数帧 (Odd frame)<br>此字段由应用程序置位或复位，以分别指示 OTG 主机必须传输奇数帧或偶数帧。此字段只适用于周期性（同步和中断）事务。<br>0：偶数帧<br>1：奇数帧 (Odd frame) |
| 28:22 | DAD | 设备地址 (Device address)<br>此字段用于指定要与该主机通信的特定设备。 |
| 21:20 | MCNT | 多重计数 (Multicount)<br>此字段向主机指示该周期性端点每帧必须执行的事务数。该字段不用于非周期性传输。<br>00：保留。对该字段的操作会产生不明确的结果<br>01：1 个事务<br>10：该端点每帧需要发出 2 个事务<br>11：该端点每帧需要发出 3 个事务<br>*注意： 此字段至少须置为 01。* |
| 19:18 | EPTYP | 端点类型 (Endpoint type) |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 指示选择的传输类型。<br>00：控制<br>01：同步<br>10：批量<br>11：中断 |
| 17 | LSDEV | 低速设备 (Low-speed device)<br>此字段由应用程序置 1，表示此通道正在与一个低速设备进行通信。 |
| 16 | | 保留，必须保持复位值。 |
| 15 | EPDIR | 端点方向 (Endpoint direction)<br>指示通信事务的方向是输入还是输出。<br>0：输出<br>1：输入 |
| 14:11 | EPNUM | 端点编号 (Endpoint number)<br>指示要与该主机通道通信的 USB 设备的端点号。 |
| 10:0 | MPSIZ | 最大数据包大小 (Maximum packet size)<br>指示与该主机通道通信的设备端点的最大数据包大小。 |

### 26.10.24. USB 主机通道分离控制寄存器 n (HCSPLTn)

USB host channel split control register n

**偏移地址：** 0x504 + (ChannelIndex * 0x20)

**复位值：** 0x00000000

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 31 | SPLTEN | 通道分离使能控制寄存器 (Split Enable)<br>0：禁止分离 (Split not enabled)<br>1：使能分离 |

### 26.10.25. USB 主机通道中断寄存器 n (HCINTn)

USB host channel interrupt register n

**偏移地址：** 0x508 + (ChannelIndex * 0x20)

**复位值：** 0x00000000

该寄存器指示在出现 USB 和 AHB 相关事件时通道的状态。具体如图 383 中所示。当模块中断寄存器中的主机通道中断位（USB_GINTSTS 中的 HCINT 位）置 1 时，应用程序必须读取该寄存器。在对寄存器执行读操作之前，应用程序必须先读取主机全体通道中断(USB_HAINT) 寄存器，以获取主机通道中断寄存器 n 的准确 ChannelIndex。应用程序必须将该寄存器中的相应位清零，才能将 USB_HAINT 和 USB_GINTSTS 寄存器中的对应位清零。

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | | DTERR | FRMOR | BBERR | TXERR | res. | ACK | NAK | STALL | res. | CHH | XFRC |
| | | | | | rw1c | rw1c | rw1c | rw1c | | rw1c | rw1c | rw1c | rw1c | rw1c | rw1c |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:11 | | 保留，必须保持复位值。 |
| 10 | DTERR | 数据同步错误 (Data toggle error) |
| 9 | FRMOR | 帧溢出错误 (Frame overrun) |
| 8 | BBERR | 串扰错误 (Babble error) |
| 7 | TXERR | 通信事务错误 (Transaction error)<br>指示 USB 上发生下列错误之一:<br>CRC 校验失败<br>超时<br>位填充错误<br>错误的 EOP |
| 6 | | 保留，必须保持复位值。 |
| 5 | ACK | 收到/发出 ACK 响应 (ACK response received/transmitted interrupt) |
| 4 | NAK | 收到 NAK 响应 (NAK response received interrupt) |
| 3 | STALL | 收到 STALL 响应 (STALL response received interrupt) |
| 2 | | 保留，必须保持复位值。 |
| 1 | CHH | 通道停止 (Channel halted)<br>因任意 USB 事务错误或为响应应用程序的禁止请求而导致传输非正常结束。 |
| 0 | XFRC | 传输完成 (Transfer completed)<br>未出现任何错误，正常完成传输。 |

### 26.10.26. USB 主机通道中断屏蔽寄存器 n (HCINTMSKn)

USB host channel interrupt mask register n

**偏移地址：** 0x50C + (ChannelIndex * 0x20)

**复位值：** 0x00000000

此寄存器反映了先前部分中介绍的各通道状态的屏蔽情况。

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | | DTERRM | FRMORM | BBERRM | TXERRM | NYET | ACKM | NAKM | STALLM | res. | CHHM | XFRCM |
| | | | | | rw | rw | rw | rw | rw | rw | rw | rw | | rw | rw |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:11 | | 保留，必须保持复位值。 |
| 10 | DTERRM | 数据同步错误屏蔽 (Data toggle error mask)<br>0：屏蔽中断<br>1：使能中断 |
| 9 | FRMORM | 帧溢出屏蔽 (Frame overrun mask)<br>0：屏蔽中断<br>1：使能中断 |
| 8 | BBERRM | 串扰错误屏蔽 (Babble error mask)<br>0：屏蔽中断<br>1：使能中断 |
| 7 | TXERRM | 通信事务错误屏蔽 (Transaction error mask)<br>0：屏蔽中断<br>1：使能中断 |

| 位域 | 名称 | 描述 |
|---|---|---|
| 6 | NYET | NYET 响应接收中断屏蔽 (response received interrupt mask)<br>0：屏蔽中断<br>1：使能中断 |
| 5 | ACKM | ACK 响应接收/发送中断屏蔽 (ACK response received/transmitted interrupt mask)<br>0：屏蔽中断<br>1：使能中断 |
| 4 | NAKM | NAK 响应接收中断屏蔽 (NAK response received interrupt mask)<br>0：屏蔽中断<br>1：使能中断 |
| 3 | STALLM | STALL 响应接收中断屏蔽 (STALL response received interrupt mask)<br>0：屏蔽中断<br>1：使能中断 |
| 2 | | 保留，必须保持复位值。 |
| 1 | CHHM | 通道停止中断屏蔽 (Channel halted mask)<br>0：屏蔽中断<br>1：使能中断 |
| 0 | XFRCM | 传输完成中断屏蔽 (Transfer completed mask)<br>0：屏蔽中断<br>1：使能中断 |

### 26.10.27. USB 主机通道传输大小寄存器 n (HCTSIZn)

USB host channel transfer size register n

**偏移地址：** 0x510 + (ChannelIndex * 0x20)

**复位值：** 0x00000000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | DPID | | PKTCNT | | | | | | | | | | XFRSIZ | | |
| | rw | | rw | | | | | | | | | | rw | | |

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 31 | | 保留，必须保持复位值。 |
| 30:29 | DPID | 数据 PID (Data PID)<br>应用程序在此字段设置数据通信的初始同步 PID。主机在此次传输事务过程中保留该字段的<br>设置。<br>00：DATA0<br>01：DATA2<br>10：DATA1<br>11：MDATA（控制传输）/SETUP（非控制传输） |
| 28:19 | PKTCNT | 数据包计数 (Packet count)<br>应用程序在此字段中设置将要发送 (OUT) 或接收 (IN) 的数据包数。 |

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 主机每成功发送或接收一个 OUT/IN 数据包便递减一次计数值。此值达到 0 后，将中断应用程序来指示操作正常完成。 |
| 18:0 | XFRSIZ | 传输大小 (Transfer size)<br>对于 OUT 操作，此字段为传输期间主机发送的数据字节数。<br>对于 IN 操作，此字段为应用程序保留给传输的缓冲区大小。对于 IN 事务（周期性和非周期性），应用程序会将此字段编程为最大数据包大小的整数倍。 |

### 26.10.28. USB 主机通道 DMA 地址寄存器 n (HCDMAn)

USB host channel DMA address register n

**偏移地址：** 0x514 + (ChannelIndex * 0x20)

**复位值：** 0x00000000

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

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

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:0 | DMAAddr | 配置 DMA 地址 |

### 26.10.29. USB 设备配置寄存器 (DCFG)

USB device configuration register

**偏移地址：** 0x800

**复位值：** 0x02200000

此寄存器在上电、执行某些控制命令或枚举后，会将模块配置为设备模式。请勿在初始编程后更改该寄存器。

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| res. | | | PFIVL | | DAD | | | | | | | res. | NZLSOHSK | DSPD | |
| | | | rw | | rw | | | | | | | | rw | rw | |

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:13 | | 保留，必须保持复位值。 |
| 12:11 | PFIVL | 周期性帧间隔 (Periodic frame interval)<br>指示一帧内必须使用周期性帧中断通知应用程序的时间点。此功能可用于确定该帧的所有同<br>步通信是否完成。<br>00：80% 帧间隔<br>01：85% 帧间隔 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 10：90% 帧间隔<br>11：95% 帧间隔 |
| 10:4 | DAD | 设备地址 (Device address)<br>应用程序必须在执行每个 SetAddress 控制命令后根据命令参数对该字段进行设置。 |
| 3 | | 保留，必须保持复位值。 |
| 2 | NZLSOHSK | 非零长度状态 OUT 握手信号 (Non-zero-length status OUT handshake)<br>在控制传输状态阶段的 OUT 事务期间，当模块收到非零长度数据包后，应用程序可以使用此字段选择要发送的握手信号。<br>1：收到非零长度状态 OUT 事务时，回复 STALL 握手信号，收到的 OUT 数据包不发送给应用程序。<br>0：将收到的 OUT 数据包（零长度或非零长度）发送给应用程序，并基于设备端点控制寄存器中端点的 NAK 和 STALL 位回复握手信号。 |
| 1:0 | DSPD | 设备速度 (Device speed)<br>指示应用程序要求模块进行枚举所采用的速度，或应用程序支持的最大速度。但是，实际总<br>线速度只有在完成 chirp 序列后才能确定，同时此速度基于与模块连接的 USB 主机的速度。<br>00：保留<br>01：保留<br>10：保留<br>11：全速（USB 1.1 收发器时钟为 48 MHz） |

### 26.10.30. USB 设备控制寄存器 (DCTL)

USB device control register

**偏移地址：** 0x804

**复位值：** 0x00000000

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | POPRGDNE | CGONAK | SGONAK | CGINAK | SGINAK | TCTL | | | GONSTS | GINSTS | SDIS | RWUSIG |
| | | | | rw | w | w | w | w | rw | | | r | r | rw | rw |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:12 | | 保留，必须保持复位值。 |
| 11 | POPRGDNE | 上电编程完成 (Power-on programming done)<br>应用程序使用此位指示寄存器从掉电模式唤醒后已完成编程。 |
| 10 | CGONAK | 将全局 OUT NAK 清零 (Clear global OUT NAK)<br>对此位执行写操作会将全局 OUT NAK 清零。 |
| 9 | SGONAK | 将全局 OUT NAK 置 1 (Set global OUT NAK)<br>对此位执行写操作会将全局 OUT NAK 置 1。<br>应用程序使用此位在所有 OUT 端点发送 NAK 握手信号。<br>应用程序只有确定模块中断寄存器中全局 OUT NAK 有效位（USB_GINTSTS 中 GONAKEFF 位）已清零时，才可以将此位置 1。 |
| 8 | CGINAK | 将全局 IN NAK 清零 (Clear global IN NAK)<br>对此位执行写操作会将全局 IN NAK 清零。 |

| 位域 | 名称 | 描述 |
|---|---|---|
| 7 | SGINAK | 将全局 IN NAK 置 1 (Set global IN NAK)<br>对此字段执行写操作会将全局非周期性 IN NAK 置 1。应用程序使用此位使所有非周期性 IN 端点发送 NAK 握手信号。<br>应用程序只有确定模块中断寄存器中全局 IN NAK 有效位（USB_GINTSTS 中 GINAKEFF 位）已清零时，才可以将此位置 1。 |
| 6:4 | TCTL | 测试控制 (Test control)<br>000：测试模式禁止<br>001：Test_J 模式<br>010：Test_K 模式<br>011：Test_SE0_NAK 模式<br>100：Test_Packet 模式<br>101：Test_Force_Enable<br>其它值：保留 |
| 3 | GONSTS | 全局 OUT NAK 状态 (Global OUT NAK status)<br>0：将根据 FIFO 状态和 NAK 和 STALL 位设置发送握手信号。<br>1：无论 RxFIFO 中是否还有空闲空间都不接收数据。除 SETUP 事务之外，对所有收到的数据包回复 NAK 握手信号。所有同步类型的 OUT 数据包都将被丢弃。 |
| 2 | GINSTS | 全局 IN NAK 状态 (Global IN NAK status)<br>0：将根据发送 FIFO 中的数据可用性回复握手信号。<br>1：使所有非周期性 IN 端点回复 NAK 握手信号，无需考虑发送 FIFO 中的数据可用性。 |
| 1 | SDIS | 软断开 (Soft disconnect)<br>应用程序使用该位向 USB OTG 模块发出执行软断开的信号。该位置 1 时，主机不会看到设备已连接，且该设备也不会接收 USB 上的信号。在应用程序将此位清零之前，模块会保持断开状态。<br>0：正常工作。此位在软断开之后清零，会使主机收到设备已连接的事件。重新连接设备之后，USB 主机会重新启动设备枚举。<br>1：使主机收到设备断开连接的事件。 |
| 0 | RWUSIG | 发送远程唤醒信号 (Remote wakeup signaling)<br>应用程序将此位置 1 时，模块会启动远程发送信号，以唤醒 USB 主机。应用程序必须将此位置 1 以使模块退出挂起状态。根据 USB 2.0 规范，应用程序必须在将此位置 1 之后的 1 ms 到 15 ms 内将其清零。 |

### 26.10.31. USB 设备状态寄存器 (DSTS)

USB device status register

**偏移地址：** 0x808

**复位值：** 0x00000010

此寄存器指示模块在出现 USB 相关事件时的状态。发生中断时，必须从设备全体中断 (USB_DAINT) 寄存器读取发生中断的端点信息。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FNSOF | | | | | | | | res. | | | | EERR | ENUMSPD | | SUSPSTS |
| r | | | | | | | | | | | | r | r | | r |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:22 | | 保留，必须保持复位值。 |
| 21:8 | FNSOF | 接收 SOF 的帧编号 (Frame number of the received SOF) |
| 7:4 | | 保留，必须保持复位值。 |
| 3 | EERR | 不定错误 (Erratic error)<br>模块将该位置 1 以报告任何不定错误。<br>由于不定错误，USB 控制器会进入挂起状态，并且会以 USB_GINTSTS 寄存器的早期挂起位（USB_GINTSTS 中的 ESUSP 位）生成一个中断。如果早期挂起中断是由不定错误触发，则应用程序只能执行软断开以恢复通信。 |
| 2:1 | ENUMSPD | 枚举速度 (Enumerated speed)<br>指示 USB 控制器通过 chirp 序列检测速度后被枚举成的速度。<br>01：保留<br>10：保留<br>11：全速（PHY 时钟运行频率为 48 MHz）<br>其它值：保留 |
| 0 | SUSPSTS | 挂起状态 (Suspend status)<br>在设备模式下，只要在 USB 上检测到挂起状态，该位就会置 1。当 USB 总线上的空闲状态保持 3ms，模块便会进入挂起状态。出现以下情况时，模块会退出挂起状态：<br>- USB 数据线上有活动<br>- 应用程序对 USB_DCTL 寄存器的远程唤醒信号位（USB_DCTL 中的 RWUSIG 位）<br>执行写操作。 |

### 26.10.32. USB 设备 IN 端点通用中断屏蔽寄存器 (DIEPMSK)

USB device IN endpoint common interrupt mask register

**偏移地址：** 0x810

**复位值：** 0x00000000

此寄存器与设备 IN 端点中断 (USB_DIEPINTn) 寄存器配合使用，以便在每个 IN 端点上生成中断。通过对此寄存器的相应位执行写操作，可屏蔽 USB_DIEPINTn 寄存器中的 IN 端点中断。默认情况下，状态中断都被屏蔽。

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | | | | | | INEPNEM | INEPNMM | ITTXFEMSK | TOM | res. | EPDM | XFRCM |
| | | | | | | | | | rw | rw | rw | rw | | rw | rw |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:7 | | 保留，必须保持复位值。 |
| 6 | INEPNEM | IN 端点 NAK 有效中断屏蔽 (IN endpoint NAK effective mask)<br>0：屏蔽中断 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 1：使能中断 |
| 5 | INEPNMM | EP 不匹配时接收到 IN 令牌中断屏蔽 (IN token received with EP mismatch mask)<br>0：屏蔽中断<br>1：使能中断 |
| 4 | ITTXFEMSK | TxFIFO 为空时接收到 IN 令牌中断屏蔽 (IN token received when TxFIFO empty mask)<br>0：屏蔽中断<br>1：使能中断 |
| 3 | TOM | 超时中断屏蔽（非同步端点）(Timeout condition mask (Non-isochronous endpoints))<br>0：屏蔽中断<br>1：使能中断 |
| 2 | | 保留，必须保持复位值。 |
| 1 | EPDM | 端点禁止中断屏蔽 (Endpoint disabled interrupt mask)<br>0：屏蔽中断<br>1：使能中断 |
| 0 | XFRCM | 传输完成中断屏蔽 (Transfer completed interrupt mask)<br>0：屏蔽中断<br>1：使能中断 |

### 26.10.33. USB 设备 OUT 端点通用中断屏蔽寄存器 (DOEPMSK)

USB device OUT endpoint common interrupt mask register

**偏移地址：** 0x814

**复位值：** 0x00000000

此寄存器与设备 OUT 端点中断 (USB_DOEPTSIZn) 寄存器配合使用，以便可以在每个 OUT 端点上生成中断。通过对此寄存器的相应位执行写操作，可屏蔽 USB_DOEPTSIZn 寄存器中的 OUT 端点中断。默认情况下，状态中断都被屏蔽。

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | | | | | | | | OTEPDM | STUPM | res. | EPDM | XFRCM |
| | | | | | | | | | | | rw | rw | | rw | rw |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:5 | | 保留，必须保持复位值。 |
| 4 | OTEPDM | 端点禁止时接收到 OUT 令牌中断屏蔽 (OUT token received when endpoint disabled mask)<br>仅适用于控制 OUT 端点。<br>0：屏蔽中断<br>1：使能中断 |
| 3 | STUPM | SETUP 阶段完成中断屏蔽 (SETUP phase done mask)<br>仅适用于控制端点。<br>0：屏蔽中断<br>1：使能中断 |
| 2 | | 保留，必须保持复位值。 |

| 位域 | 名称 | 描述 |
|---|---|---|
| 1 | EPDM | 端点禁止中断屏蔽 (Endpoint disabled interrupt mask)<br>0：屏蔽中断<br>1：使能中断 |
| 0 | XFRCM | 传输完成中断屏蔽 (Transfer completed interrupt mask)<br>0：屏蔽中断<br>1：使能中断 |

### 26.10.34. USB 设备全体端点中断寄存器 (DAINT)

USB device all endpoints interrupt register

**偏移地址：** 0x818

**复位值：** 0x00000000

当端点上发生有效事件时，设备全体端点中断寄存器将通过模块中断寄存器中的设备 OUT 端点中断位或设备 IN 端点中断位（分别为 USB_GINTSTS 中的 OEPINT 或 IEPINT 位）来中断应用程序。每个端点对应一个中断位，OUT 端点和 IN 端点均最多有 16 个中断 位。双向端点将使用相应的 IN 和 OUT 中断位。当应用程序将相应设备端点中断寄存器 n (USB_DIEPINTn/USB_DOEPTSIZn) 中的位置 1 和清零时，此寄存器中的相应位也 将置 1 和清零。

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

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | OEPINT | OUT 端点中断位 (OUT endpoint interrupt bits)<br>每个 OUT 端点对应一位：<br>OUT 端点 0 对应位 16，而 OUT 端点 3 对应位 18。 |
| 15:0 | IEPINT | IN 端点中断位 (IN endpoint interrupt bits)<br>每个 IN 端点对应一位：<br>IN 端点 0 对应位 0，而 IN 端点 3 对应位 3。 |

### 26.10.35. USB 全体端点中断屏蔽寄存器 (DAINTMSK)

USB all endpoints interrupt mask register

**偏移地址：** 0x81C

**复位值：** 0x00000000

设备端点中断屏蔽寄存器与设备端点中断寄存器结合使用，进而在设备端点上发生事件时中断应用程序。不过，与该中断对应的设备全体端点中断 (USB_DAINT) 寄存器位仍会置 1。

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

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | OEPM | OUT EP 中断屏蔽位 (OUT EP interrupt mask bits)<br>每个 OUT 端点对应一位:<br>OUT EP 0 对应位 16，而 OUT EP 3 对应位 18<br>0：屏蔽中断<br>1：使能中断 |
| 15:0 | IEPM | IN EP 中断屏蔽位 (IN EP interrupt mask bits)<br>每个 IN 端点对应一位:<br>IN EP 0 对应位 0，而 IN EP 3 对应位 3<br>0：屏蔽中断<br>1：使能中断 |

## 26.10.36. USB 设备 VBUS 放电时间寄存器 (DVBUSDIS)

USB device VBUS discharge time register

**偏移地址：** 0x0828

**复位值：** 0x000017D7

该寄存器指定 SRP 期间 VBUS 发出脉冲之后的 VBUS 放电时间。

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | | 保留，必须保持复位值。 |
| 15:0 | VBUSDT | 设备 VBUS 放电时间 (Device VBUS discharge time)<br>指定 SRP 期间 VBUS 发出脉冲之后的 VBUS 放电时间。该时间值等于:<br>VBUS 放电时间（PHY 时钟数）/1024<br>该值可基于不同的 VBUS 负载根据需要进行调整。 |

## 26.10.37. USB 设备 VBUS 脉冲时间寄存器 (DVBUSPULSE)

USB device VBUS pulsing time register

**偏移地址：** 0x082C

**复位值：** 0x000005B8

该寄存器指定 SRP 期间的 VBUS 脉冲时间。

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:12 | | 保留，必须保持复位值。 |

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| 11:0 | DVBUSP | 设备 VBUS 脉冲时间 (Device VBUS pulsing time)<br>指定 SRP 期间的 VBUS 脉冲时间。该时间值等于:<br>VBUS 脉冲时间（PHY 时钟数）/1024<br>USB 设备 IN 端点 FIFO 空中断屏蔽寄存器：USB_DIEPEMPMSK)<br>USB device IN endpoint FIFO empty interrupt mask register<br>偏移地址：0x834<br>复位值：0x0000 0000<br>此寄存器用于控制 IN 端点 FIFO 空中断的生成 (TXFE_USB_DIEPINTn)。 |

### 26.10.38. USB 设备 IN 端点 FIFO 空中断屏蔽寄存器 (DIEPEMPMSK)

USB device IN endpoint FIFO empty interrupt mask register

**偏移地址：** 0x834

**复位值：** 0x00000000

此寄存器用于控制 IN 端点 FIFO 空中断的生成 (TXFE_USB_DIEPINTn)。

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

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:16 | | 保留，必须保持复位值。 |
| 15:0 | INEPTXFEM | IN EP Tx FIFO 空中断屏蔽位 (IN EP Tx FIFO empty interrupt mask bits)<br>这些位用作 USB_DIEPINTn 的屏蔽位。<br>每个位对应一个 IN 端点的 TXFE 中断:<br>IN 端点 0 对应位 0，而 IN 端点 3 对应位 3<br>0：屏蔽中断<br>1：使能中断 |

### 26.10.39. USB 设备单个端点中断寄存器 (DEACHINT)

USB device each endpoint interrupt register

**偏移地址：** 0x0838

**复位值：** 0x00000000

IN 端点 1 对应一个中断位，OUT 端点 1 对应一个中断位。

### 26.10.40. USB 设备单个端点中断屏蔽寄存器 (DEACHINTMSK)

USB device each endpoint interrupt register mask

**偏移地址：** 0x083C

**复位值：** 0x00000000

IN 端点 1 对应一个中断位，OUT 端点 1 对应一个中断位。

### 26.10.41. USB 设备 IN 端点 1 中断屏蔽寄存器 (DIEPEACHMSK1)

USB device each in endpoint-1 interrupt register

**偏移地址：** 0x844

**复位值：** 0x00000000

### 26.10.42. USB 设备 OUT 端点 1 中断屏蔽寄存器 (DOEPEACHMSK1)

USB device each OUT endpoint-1 interrupt register

**偏移地址：** 0x884

**复位值：** 0x00000000

### 26.10.43. USB 设备端点控制寄存器 n (DIEPCTLn)

USB device endpoint control register n

**偏移地址：** 0x900 + (EPIndex * 0x20)

**复位值：** 0x00000000

应用程序使用此寄存器控制各个逻辑端点（端点 0 除外）的行为。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EPENA | EPDIS | SODDFRM | SD0PID | SNAK | CNAK | TXFNUM | | | | STALL | res. | EPTYP | | NAKSTS | EONUM |
| rw1s | rw1s | w | w | w | w | rw | | | | rw1s | | rw | | r | r |

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 31 | EPENA | 端点使能 (Endpoint enable)<br>应用程序将此位置 1 以在端点上启动数据发送。<br>在此端点上触发以下任一中断之前，模块会将此位清零：<br>- SETUP 阶段完成<br>- 端点禁止<br>- 传输完成 |
| 30 | EPDIS | 端点禁止 (Endpoint disable)<br>即使在该端点上的传送完成之前，应用程序也可将此位置 1，以停止端点上的数据发送/接<br>收。应用程序必须等到发生端点禁止中断后，才能将端点视为禁止端点。在端点禁止中断<br>位置 1 前，模块会将此位清零。只有在该端点的端点使能位置 1 后，应用程序才可将该位置 1。 |
| 29 | SODDFRM | 设置奇数帧 (Set odd frame)<br>仅适用于同步 IN 和 OUT 端点。<br>对此字段进行写操作会将偶数/奇数帧 (EONUM) 字段设置为奇数帧。 |
| 28 | SD0PID | 设置 DATA0 PID (Set DATA0 PID)<br>仅适用于中断/批量 IN 端点。<br>对此字段进行写操作会将此寄存器中的端点数据 PID (DPID) 字段设置为 DATA0。 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | SEVNFRM：设置偶数帧 (Set even frame)<br>仅适用于同步 IN 端点。<br>对此字段进行写操作会将偶数/奇数帧 (EONUM) 字段设置为偶数帧。 |
| 27 | SNAK | 将 NAK 置 1 (Set NAK)<br>对此位进行写操作会将端点的 NAK 位置 1。<br>通过此位，应用程序可以控制端点上 NAK 握手信号的发送。发生传输完成中断时或端点上接收到 SETUP 后，模块也可以将 OUT 端点的这个位置 1。 |
| 26 | CNAK | 将 NAK 清零 (Clear NAK)<br>对此位进行写操作会将端点的 NAK 位清零。 |
| 25:22 | TXFNUM | TxFIFO 编号 (TxFIFO number)<br>这些位用于指定与此端点相关联的 FIFO 编号。必须为每个有效的 IN 端点设置单独的 FIFO 编号。<br>此字段仅针对 IN 端点有效。 |
| 21 | STALL | STALL 握手 (STALL handshake)<br>仅适用于非控制、非同步 IN 端点（访问类型为 rw）。<br>应用程序将此位置 1 使得设备对来自 USB 主机的所有令牌都回复 STALL。如果 NAK 位、全局 IN NAK 或全局 OUT NAK 与此位同时置 1，则 STALL 位优先。只有应用程序能够将此位清零，而模块则不能。<br>仅适用于控制端点（访问类型为 rs）<br>此端点接收到 SETUP 令牌时，应用程序只能将此位置 1，而模块会将其清零。如果 NAK 位、全局 IN NAK 或全局 OUT NAK 与此位同时置 1，则 STALL 位优先。无论此位如何设置，模块总是通过 ACK 握手响应 SETUP 数据包。 |
| 20 | | 保留，必须保持复位值。 |
| 19:18 | EPTYP | 端点类型 (Endpoint type)<br>以下是这个逻辑端点支持的传输类型。<br>00：控制<br>01：同步<br>10：批量<br>11：中断 |
| 17 | NAKSTS | NAK 状态 (NAK status)<br>它指示以下结果：<br>0：模块根据 FIFO 状态回复非 NAK 握手。<br>1：模块在此端点上回复 NAK 握手。<br>当应用程序或模块将此位置 1 时：<br>对于非同步 IN 端点：即使 TxFIFO 中存在可用数据，模块也会停止通过 IN 端点发送任何数据。<br>对于同步 IN 端点：即使 TxFIFO 中存在可用数据，模块也会发送长度为零的数据包。<br>无论此位如何设置，模块总是通过 ACK 握手响应 SETUP 数据包。 |
| 16 | EONUM | 偶数/奇数帧 (Even/odd frame)<br>仅适用于同步 IN 端点。<br>指示模块为此端点发送/ 接收同步的数据所在的帧的编号。应用程序必须通过此寄存器中<br>的 SEVNFRM 和 SODDFRM 字段对偶数/奇数帧编号进行编程，以便此端点发送/接收同步数据。<br>0：偶数帧 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 1：奇数帧<br>DPID：端点数据 PID (Endpoint data PID)<br>仅适用于中断/批量 IN 端点。<br>包含此端点上将要接收或发送的数据包的 PID。端点激活后，应用程序必须对要在此端点上接收或发送的首个数据包的 PID 进行编程。应用程序使用 SD0PID 寄存器字段对 DATA0 或 DATA1 PID 进行编程。<br>0：DATA0<br>1：DATA1 |
| 15 | USBAEP | USB 活动端点 (USB active endpoint)<br>指示此端点在当前配置和接口中是否激活。检测到 USB 复位后，模块会为所有端点（端点 0 除外）将此位清零。接收到 SetConfiguration 和 SetInterface 命令后，应用程序必须相应地对端点寄存器进行编程并将此位置 1。 |
| 14:11 | | 保留，必须保持复位值。 |
| 10:0 | MPSIZ | 最大数据包大小 (Maximum packet size)<br>应用程序必须将此字段编程为当前逻辑端点的最大数据包大小。此值以字节为单位。 |

### 26.10.44. USB 设备端点中断寄存器 n (DIEPINTn)

USB device endpoint interrupt register n

**偏移地址：** 0x908 + (EPIndex * 0x20)

**复位值：** 0x00000080

此寄存器指示端点在出现 USB 和 AHB 相关事件时的状态。具体如图 383 中所示。当模块中断寄存器中的 IN 端点中断位（USB_GINTSTS 中的 IEPINT 位）置 1 时，应用程序必须读取此寄存器。应用程序必须先读取设备全体端点中断 (USB_DAINT) 寄存器，以获取设备端点中断寄存器 n 的准确端点编号，然后才能读取此寄存器。应用程序必须将该寄存器中的相应位清零，才能将 USB_DAINT 和 USB_GINTSTS 寄存器中的对应位清零。

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | | | | | TXFE | INEPNE | res. | ITTXFE | TOC | res. | EPDISD | XFRC |
| | | | | | | | | | | | | | | | |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | | 保留，必须保持复位值。 |
| 7 | TXFE | 发送 FIFO 为空 (Transmit FIFO empty)<br>当此端点的 TxFIFO 为半空或全空时，此中断被置位。TxFIFO 为半空还是全空状态由 USB_GAHBCFG 寄存器中的 TxFIFO 空白级别位（USB_GAHBCFG 中的 TXFELVL 位）决定。 |
| 6 | INEPNE | IN 端点 NAK 有效 (IN endpoint NAK effective)<br>当应用程序通过向 USB_DIEPCTLn 中的 CNAK 位写入数据来将 IN 端点 NAK 清零时，此位可被清零。<br>该中断指示模块已对（由应用程序或模块）置 1 的 NAK 采样，结果已生效。该中断指示由应用程序置 1 的 IN 端点 NAK 位已在模块中起作用。<br>此中断不保证在 USB 上发送了 NAK 握手信号。STALL 位的优先级高于 NAK 位。 |
| 5 | | 保留，必须保持复位值。 |

| 位域 | 名称 | 描述 |
|---|---|---|
| 4 | ITTXFE | TxFIFO 为空时接收到 IN 令牌 (IN token received when TxFIFO is empty)<br>仅适用于非周期性 IN 端点。<br>当和该端点相对应的 TxFIFO（周期性/非周期性）为空时，接收到 IN 令牌，从而产生中断。 |
| 3 | TOC | 超时条件 (Timeout condition)<br>仅适用于控制 IN 端点。<br>指示该端点对最近收到的 IN 令牌响应超时。 |
| 2 | | 保留，必须保持复位值。 |
| 1 | EPDISD | 端点禁止中断 (Endpoint disabled interrupt)<br>此位指示该端点已经由应用程序禁止掉。 |
| 0 | XFRC | 传输完成中断 (Transfer completed interrupt)<br>此字段指示在此端点上设置的传输已经在 USB 和 AHB 上传输完成。 |

## 26.10.45. USB 设备 IN 端点 0 传输大小寄存器 (DIEPTSIZ0)

USB device IN endpoint 0 transfer size register

**偏移地址：** 0x910

**复位值：** 0x00000000

在使能端点 0 之前，应用程序必须修改此寄存器。通过设备控制端点 0 控制寄存器中的端点使能位（USB_DIEPCTL0 中的 EPENA）使能端点 0 后，模块对此寄存器进行修改。仅当模块将端点使能位清零后，应用程序才能读取此寄存器。非零端点使用端点 1-15 的寄存器。

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

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:21 | | 保留，必须保持复位值。 |
| 20:19 | PKTCNT | 数据包计数 (Packet count)<br>指示端点 0 的一次数据传输包含的数据包个数。<br>每次从 TxFIFO 读取数据包（最大大小或短数据包）时，此字段将递减。 |
| 18:7 | | 保留，必须保持复位值。 |
| 6:0 | XFRSIZ | 传输大小 (Transfer size)<br>指示端点 0 的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据<br>包结束时中断。<br>每次向 TxFIFO 写入来自外部存储器的数据包时，模块会使此字段递减。 |

## 26.10.46. USB 设备端点传输大小寄存器 n (DIEPTSIZn)

USB device endpoint transfer size register n

**偏移地址：** 0x910 + (EPIndex * 0x20)

**复位值：** 0x00000000

在使能该端点之前，应用程序必须修改此寄存器。通过设备端点控制寄存器 n 中的端点使能位（USB_DIEPCTLn 中的 EPENA 位）使能该端点后，模块对此寄存器进行修改。仅当模块将端点使能位清零后，应用程序才能读取此寄存器。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | MCNT | | PKTCNT | | | | | | | | | | XFRSIZ | | |
| | | | | | | | | | | | | | | | |

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 31 | | 保留，必须保持复位值。 |
| 30:29 | MCNT | 多重计数 (Multi count)<br>对于周期性 IN 端点，此字段指示在 USB 上每帧必须发送的数据包数。模块使用此字段计算同步 IN 端点的数据 PID。<br>01：1 个数据包<br>10：2 个数据包<br>11：3 个数据包 |
| 28:19 | PKTCNT | 数据包计数 (Packet count)<br>指示该端点上的一次数据传输包含的数据包个数。<br>每次从 TxFIFO 读取数据包（最大大小或短数据包）时，此字段将递减。 |
| 18:0 | XFRSIZ | 传输大小 (Transfer size)<br>此字段包含当前端点的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这<br>些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每<br>个数据包结束时中断。<br>每次向 TxFIFO 写入来自外部存储器的数据包时，模块会使此字段递减。 |

## 26.10.47. USB 设备 IN 端点 DMA 地址寄存器 n (DIEPDMAn)

USB device IN endpoint DMA address register n

**偏移地址：** 0x914 + (EPIndex * 0x20)

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

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | DMAA | DMA 起始地址 (DMA Start Address) |

### 26.10.48. USB 设备 IN 端点发送 FIFO 状态寄存器 (DTXFSTSn)

USB device IN endpoint transmit FIFO status register

**偏移地址：** 0x918 + (EPIndex * 0x20)

此只读寄存器包含设备 IN 端点 TxFIFO 的空闲空间信息。

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 15:0 | IEPTAVA | 发送端点 TxFIFO 可用空间 (IN Endpoint TxFIFO Space Avail)<br>以四字节为单位，指示发送端点 TxFIFO 可用空间。 |

### 26.10.49. USB 设备控制 OUT 端点 0 控制寄存器 (DOEPCTL0)

USB device control OUT endpoint 0 control register

**偏移地址：** 0xB00

**复位值：** 0x00008000

本节介绍设备控制 OUT 端点 0 控制寄存器。非零控制端点使用编号 1-15 的寄存器。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EPENA | EPDIS | res. | | SNAK | CNAK | res. | | | | STALL | res. | EPTYP | | NAKSTS | res. |
| rw1s | r | | | w | w | | | | | rw1s | | r | | r | |

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 31 | EPENA | 端点使能 (Endpoint enable)<br>应用程序将此位置 1 以在端点 0 上启动数据发送。<br>在此端点上触发以下任一中断之前，模块会将此位清零：<br>- SETUP 阶段完成<br>- 端点禁止<br>- 传输完成 |
| 30 | EPDIS | 端点禁止 (Endpoint disable)<br>应用程序无法禁止控制 OUT 端点 0。 |
| 29:28 | | 保留，必须保持复位值。 |
| 27 | SNAK | 将 NAK 置 1 (Set NAK)<br>对此位进行写操作会将端点的 NAK 位置 1。<br>通过此位，应用程序可以控制端点上 NAK 握手信号的发送。发生传输完成中断时或端点上接收到 SETUP 后，模块也可以将此位置 1。 |
| 26 | CNAK | 将 NAK 清零 (Clear NAK) |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 对此位进行写操作会将端点的 NAK 位清零。 |
| 25:22 | | 保留，必须保持复位值。 |
| 21 | STALL | STALL 握手 (STALL handshake)<br>此端点接收到 SETUP 令牌时，应用程序只能将此位置 1，而模块会将其清零。如果 NAK 位、全局 OUT NAK 与此位同时置 1，则 STALL 位优先。<br>无论此位如何设置，模块总是通过 ACK 握手响应 SETUP 数据包。 |
| 19:18 | EPTYP | 端点类型 (Endpoint type)<br>硬件固定为二进制 00，表示端点为控制传输类型。 |
| 17 | NAKSTS | NAK 状态 (NAK status)<br>指示以下结果：<br>0：模块根据 FIFO 状态回复非 NAK 握手。<br>1：模块在此端点上回复 NAK 握手。<br>当应用程序或模块将此位置 1 时，即使 RxFIFO 中存在空间可继续容纳收到的数据包，模块也会停止接收数据。无论此位如何设置，模块总是通过 ACK 握手响应 SETUP 数据包。 |
| 16 | | 保留，必须保持复位值。 |
| 15 | USBAEP | USB 活动端点 (USB active endpoint)<br>此位总是置 1，指示在所有配置和接口中控制端点 0 始终处于激活状态。 |
| 14:2 | | 保留，必须保持复位值。 |
| 1:0 | MPSIZ | 最大数据包大小 (Maximum packet size)<br>控制 OUT 端点 0 的最大数据包大小与在控制 IN 端点 0 中进行编程的值相同。<br>00：64 字节<br>01：32 字节<br>10：16 字节<br>11：8 字节 |

### 26.10.50. USB 设备端点控制寄存器 (DOEPCTLn)

USB device endpoint control register n

**偏移地址：** 0xB00 + (EPIndex * 0x20)

**复位值：** 0x00000000

应用程序使用此寄存器控制各个逻辑端点（端点 0 除外）的行为。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EPENA | EPDIS | SD1PID | SD0PID | SNAK | CNAK | res. | | | | STALL | res. | EPTYP | | NAKSTS | EONUM |
| rw1s | rw1s | w | w | w | w | | | | | rw1s | | rw | | r | r |

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 31 | EPENA | 端点使能 (Endpoint enable)<br>适用于 IN 和 OUT 端点。<br>应用程序将此位置 1 以在端点上启动数据发送。 |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 在此端点上触发以下任一中断之前，模块会将此位清零：<br>- SETUP 阶段完成<br>- 端点禁止<br>- 传输完成 |
| 30 | EPDIS | 端点禁止 (Endpoint disable)<br>即使在该端点上的传送完成之前，应用程序也可将此位置 1，以停止端点上的数据发送/接收。应用程序必须等到发生端点禁止中断后，才能将端点视为禁止端点。在端点禁止中断位<br>置 1 前，模块会将此位清零。只有在该端点的端点使能位置 1 后，应用程序才可将该位置 1。 |
| 29 | SD1PID | 设置 DATA1 PID (Set DATA1 PID)<br>仅适用于中断/批量 IN 和 OUT 端点。对此字段进行写操作会将此寄存器中的端点数据 PID (DPID) 字段设置为 DATA1。<br>SODDFRM：设置奇数帧 (Set odd frame)<br>仅适用于同步 IN 和 OUT 端点。对此字段进行写操作会将偶数/奇数帧 (EONUM) 字段设置为奇数帧。 |
| 28 | SD0PID | 设置 DATA0 PID (Set DATA0 PID)<br>仅适用于中断/批量 OUT 端点。<br>对此字段进行写操作会将此寄存器中的端点数据 PID (DPID) 字段设置为 DATA0。<br>SEVNFRM：设置偶数帧 (Set even frame)<br>仅适用于同步 OUT 端点。<br>对此字段进行写操作会将偶数/奇数帧 (EONUM) 字段设置为偶数帧。 |
| 27 | SNAK | 将 NAK 置 1 (Set NAK)<br>对此位进行写操作会将端点的 NAK 位置 1。<br>通过此位，应用程序可以控制端点上 NAK 握手信号的发送。发生传输完成中断时或端点上接收到 SETUP 后，模块也可以将 OUT 端点的这个位置 1。 |
| 26 | CNAK | 将 NAK 清零 (Clear NAK)<br>对此位进行写操作会将端点的 NAK 位清零。 |
| 25:22 | | 保留，必须保持复位值。 |
| 21 | STALL | STALL 握手 (STALL handshake)<br>仅适用于非控制、非同步 OUT 端点（访问类型为 rw）。<br>应用程序将此位置 1 使得设备对来自 USB 主机的所有令牌都回复 STALL。如果 NAK 位、全局 IN NAK 或全局 OUT NAK 与此位同时置 1，则 STALL 位优先。只有应用程序能够将此位清零，而模块则不能。<br>仅适用于控制端点（访问类型为 rs）<br>此端点接收到 SETUP 令牌时，应用程序只能将此位置 1，而模块会将其清零。如果 NAK 位、全局 IN NAK 或全局 OUT NAK 与此位同时置 1，则 STALL 位优先。无论此位如何设置，模块总是通过 ACK 握手响应 SETUP 数据包。 |
| 19:18 | EPTYP | 端点类型 (Endpoint type)<br>以下是这个逻辑端点支持的传输类型。<br>00：控制<br>01：同步<br>10：批量<br>11：中断 |
| 17 | NAKSTS | NAK 状态 (NAK status) |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 指示以下结果：<br>0：模块根据 FIFO 状态回复非 NAK 握手。<br>1：模块在此端点上回复 NAK 握手。<br>当应用程序或模块将此位置 1 时：<br>即使 RxFIFO 存在空间可容纳传入数据包，模块也会停止在 OUT 端点上接收任何数据。<br>无论此位如何设置，模块总是通过 ACK 握手响应 SETUP 数据包。 |
| 16 | EONUM | 偶数/奇数帧 (Even/odd frame)<br>仅适用于同步 IN 和 OUT 端点。<br>指示模块为此端点发送/接收同步的数据所在的帧的编号。应用程序必须通过此寄存器中的<br>SEVNFRM 和 SODDFRM 字段对偶数/奇数帧编号进行编程，以便此端点发送/接收同步数据。<br>0：偶数帧<br>1：奇数帧<br>DPID：端点数据 PID (Endpoint data PID)<br>仅适用于中断/批量 OUT 端点。<br>包含此端点上将要接收或发送的数据包的 PID。端点激活后，应用程序必须对要在此端点上接收或发送的首个数据包的 PID 进行编程。应用程序使用 SD0PID 寄存器字段对 DATA0 或 DATA1 PID 进行编程。<br>0：DATA0<br>1：DATA1 |
| 15 | USBAEP | USB 活动端点 (USB active endpoint)<br>指示此端点在当前配置和接口中是否激活。检测到 USB 复位后，模块会为所有端点（端点 0 除外）将此位清零。接收到 SetConfiguration 和 SetInterface 命令后，应用程序必须相应地对端点寄存器进行编程并将此位置 1。 |
| 14:11 | | 保留，必须保持复位值。 |
| 10:0 | MPSIZ | 最大数据包大小 (Maximum packet size)<br>应用程序必须将此字段编程为当前逻辑端点的最大数据包大小。此值以字节为单位。 |

### 26.10.51. USB 设备端点中断寄存器 n (DOEPINTn)

USB device endpoint interrupt register n

**偏移地址：** 0xB08 + (EPIndex * 0x20)

**复位值：** 0x00000080

此寄存器指示端点在出现 USB 和 AHB 相关事件时的状态。具体如图 383 中所示。当模块中断寄存器中的 OUT 端点中断位（USB_GINTSTS 中的 OEPINT 位）置 1 时，应用程序必须读取此寄存器。应用程序必须先读取设备全体端点中断 (USB_DAINT) 寄存器，以获取设备端点中断寄存器 n 的准确端点编号，然后才能读取此寄存器。应用程序必须将该寄存器中的相应位清零，才能将 USB_DAINT 和 USB_GINTSTS 寄存器中的对应位清零。

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | | | | | | | | | B2BSTUP | res. | OTEPDIS | STUP | res. | EPDISD | XFRC |
| | | | | | | | | | rw1c | | rw1c | rw1c | | rw1c | rw1c |

| 位域 | 名称 | 描述 |
|---|---|---|
| 31:7 | | 保留，必须保持复位值。 |
| 6 | B2BSTUP | 接收到连续的 SETUP 数据包 (Back-to-back SETUP packets received)<br>仅适用于控制 OUT 端点。<br>此位指示该端点已接收到三个以上的连续 SETUP 数据包。 |
| 5 | | 保留，必须保持复位值。 |
| 4 | OTEPDIS | 端点禁止时接收到 OUT 令牌 (OUT token received when endpoint disabled)<br>仅适用于控制 OUT 端点。<br>指示在尚未使能端点时接收到 OUT 令牌，从而产生中断。 |
| 3 | STUP | SETUP 阶段完成 (SETUP phase done)<br>仅适用于控制 OUT 端点。<br>指示控制端点的 SETUP 阶段已完成，当前控制传输中不再接收到连续的 SETUP 数据包。在此中断上，应用程序可以对接收到的 SETUP 数据包进行解码。 |
| 2 | | 保留，必须保持复位值。 |
| 1 | EPDISD | 端点禁止中断 (Endpoint disabled interrupt)<br>此位指示该端点已经由应用程序禁止掉。 |
| 0 | XFRC | 传输完成中断 (Transfer completed interrupt)<br>此字段指示在此端点上设置的传输已经在 USB 和 AHB 上传输完成。 |

## 26.10.52. USB 设备 OUT 端点 0 传输大小寄存器 (DOEPTSIZ0)

USB device OUT endpoint 0 transfer size register

**偏移地址：** 0xB10

**复位值：** 0x00000000

在使能端点 0 之前，应用程序必须修改此寄存器。通过设备控制端点 0 控制寄存器中的端点使能位（USB_DOEPCTL0 中的 EPENA 位）使能端点 0 后，模块对此寄存器进行修改。仅当模块将端点使能位清零后，应用程序才能读取此寄存器。非零端点使用端点 1-15 的寄存器。

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

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 31 | | 保留，必须保持复位值。 |
| 30:29 | STUPCNT | SETUP 数据包计数 (SETUP packet count)<br>此字段指定端点能连续接收的 SETUP 数据包数量。<br>01：1 个数据包<br>10：2 个数据包<br>11：3 个数据包 |
| 28:20 | | 保留，必须保持复位值。 |
| 19 | PKTCNT | 数据包计数 (Packet count) |

| 位域 | 名称 | 描述 |
|---|---|---|
| | | 每向 RxFIFO 写入一个数据包，此字段递减。 |
| 18:7 | | 保留，必须保持复位值。 |
| 6:0 | XFRSIZ | 传输大小 (Transfer size)<br>指示端点 0 的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据<br>包结束时中断。<br>每次从 RxFIFO 读取数据包并将其写入外部存储器时，模块会使此字段递减。 |

## 26.10.53. USB 设备端点传输大小寄存器 n (DOEPTSIZn)

USB device endpoint transfer size register n

**偏移地址：** 0xB10 + (EPIndex * 0x20)

**复位值：** 0x00000000

在使能该端点之前，应用程序必须修改此寄存器。通过设备端点控制寄存器 n 中的端点使能位（USB_DOEPCTLn 中的 EPENA 位）使能该端点后，模块对此寄存器进行修改。仅当模块将端点使能位清零后，应用程序才能读取此寄存器。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| res. | RXDPID | | PKTCNT | | | | | | | | | | XFRSIZ | | |
| | rw | | rw | | | | | | | | | | rw | | |

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

| 位域 | 名称 | 描述 |
|---|---|---|
| 31 | | 保留，必须保持复位值。 |
| 30:29 | RXDPID | 接收到的数据 PID (Received data PID)<br>仅适用于同步 OUT 端点。<br>这是此端点收到的上一个数据包的 PID。<br>00：DATA0<br>01：DATA2<br>10：DATA1<br>11：MDATA<br>STUPCNT：SETUP 数据包计数 (SETUP packet count)<br>仅适用于控制 OUT 端点。<br>此字段指定端点能连续接收的 SETUP 数据包数量。<br>01：1 个数据包<br>10：2 个数据包<br>11：3 个数据包 |
| 28:19 | PKTCNT | 数据包计数 (Packet count)<br>指示该端点上的一次数据传输包含的数据包个数。<br>每次向 RxFIFO 写入数据包（最大大小或短数据包）后，此字段将递减。 |
| 18:0 | XFRSIZ | 传输大小 (Transfer size)<br>此字段包含当前端点的一次数据传输包含的数据量，以字节为单位。仅当应 |

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 用程序传输完这<br>些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每<br>个数据包结束时中断。<br>每次从 RxFIFO 读取数据包并将其写入外部存储器时，模块会使此字段递减。 |

### 26.10.54. USB 设备 OUT 端点 DMA 地址寄存器 n (DOEPDMAn)

USB device OUT endpoint DMA address register n

**偏移地址：** 0xB14 + (EPIndex * 0x20)

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

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

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:0 | DMAA | DMA 起始地址 (DMA Start Address) |

### 26.10.55. USB 电源和时钟门控控制寄存器 (PCGCCTL)

USB power and clock gating control register

**偏移地址：** 0xE00

**复位值：** 0x00000000

此寄存器在主机模式和设备模式下均可用。

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

| 位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:5 | | 保留，必须保持复位值。 |
| 1 | GATEHCLK | 门控 HCLK (Gate HCLK)<br>当 USB 通信挂起或会话无效时，应用程序会将此位置 1，以停止对除 AHB 总线从接口、主接口和唤醒逻辑之外的模块提供时钟。当 USB 恢复通信或新会话启动时，应用程序将此位清零。 |
| 0 | STPPCLK | 停止 PHY 时钟 (Stop PHY clock)<br>当 USB 通信挂起、会话无效或设备断开连接时，应用程序将此位置 1 以停止 PHY 时钟。当 USB 恢复通信或新会话启动时，应用程序将此位清零。 |

# 27. OTP 控制器(OTP)

## 27.1. OTP 简介

OTP 是 1 块具有单次写操作的特殊存储器，OTP 出厂时内部数据经过初始化后 bit 位均为“0”，OTP 写操作只能将内部 bit 位由“0”写为“1”，而不能由“1”改为“0”。

OTP 开放空间为 0x50076380~0x500763FF，共 128 字节。

## 27.2. OTP 功能描述

### 27.2.1. OTP 只读锁定

OTP 提供区域写保护和区域写保护锁定功能。

- 区域写保护：
  - OTP 区域写保护 bit 位为“0”时，对应区域可以进行编程/擦除操作，为“1”时，对应区域只能进行读操作。
- 区域写保护锁定：
  - OTP 区域写保护锁定 bit 位为“0”时，对应区域写保护 bit 位可以修改，为“1”时，对应区域写保护 bit 位保持已有状态不可修改。

### 27.2.2. OTP 编程操作保护

为防止用户程序对 OTP 的误操作，OTP 在启动编程/擦除操作时，需要进行固定的寄存器操作后，再启动编程/擦除操作使能。

编程/擦除操作前需要对 OTP_PROT 进行 2 次连续写操作，第 1 次写入：0xABCD00A5，第 2 次写入：0x1234005A。2 次写入操作完成后，紧接着进行启动编程/擦除操作使能，如果期间有其他 FCU 操作则启动编程/擦除操作使能视为无效。

操作流程如下：

OTP_PROT = 0xABCD00A5;

OTP_PROT = 0x1234005A;

完成以上 2 次寄存器操作后，启动编程/擦除操作使能。

## 27.3. OTP_CTRL 寄存器

### 27.3.1. OTP 控制状态寄存器（OTP_CS）

偏移地址： 0x0404

复位值： 0x8000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31 | READY | 1：可进行 OTP 读操作<br>0：OTP 处于编程状态，不可进行读操作 |
| 位 30:4 | 保留 | 必须保持默认值 |
| 位 3:1 | ILLEGAL | 操作完成状态，硬件置位软件清 0：<br>000：此次操作无异常；<br>001：在编程状态下对 OTP 进行读操作<br>010：对只读区域进行编程<br>011：编程范围超出 OTP 地址范围 |
| 位 0 | START | 启动对 OTP 的操作，软件置 1 硬件清 0，软件可通过该位查询操作是否完成。 |

### 27.3.2. OTP 启动保护寄存器（OTP_PROT）

偏移地址： 0x0408

复位值： 0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | PROT | 启动保护控制位。<br>要启动 OTP 编程，需进行 3 次连续的写操作，顺序如下：写 PROT 0xabcd_00a5，写 PROT 0x1234_005a，将 START 置 1。 |

### 27.3.3. OTP 编程擦除地址寄存器（OTP_ADDR）

偏移地址： 0x040C

复位值： 0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | ADDR | 编程地址 |

### 27.3.4. OTP 编程数据寄存器（OTP_PDATA）

偏移地址： 0x0410

复位值： 0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:8 | 保留 | 必须保持默认值 |
| 位 7:0 | PDATA | 用于存放 8bit 数据编程数据。<br>只能在 START 为 0 时可写。 |

### 27.3.5. OTP 时序寄存器（OTP_TIM）

偏移地址： 0x0420

复位值： 0x0000 010D

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | | CYCLES_1US | | | | | | | |
| | | | | | | | | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 预留 | 保留位 |
| 7:0 | CYCLES_1US | 决定 1us 所用的时钟周期，比如 60M 时，周期为 16.67ns，为保险起见减去 1ns 作为余量，所以该寄存器配置为 1000ns/15ns 有余数则加 1，得 67。<br>只能在 START 为 0 时可写。 |

### 27.3.6. OTP 主存区域只读区域寄存器 3（OTP_RO3）

偏移地址： 0x0438

复位值： 0xFFFF FFFF

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | RO | 按 64bit 将 128 字节 OTP 划分成 32 个区域，每个区域由 RO 的 1bit 控制。<br>0：可编程。<br>1：只读。 |

### 27.3.7. OTP 主存区只读锁定寄存器 3（OTP_ROL3）

偏移地址： 0x043C

复位值： 0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | RO_LOCK | 用于锁定 OTP_RO3 寄存器的配置，一旦锁定，OTP_RO3 对应位不能修改，复位后锁定解除。<br>0：不锁定。<br>1：锁定，置 1 后软件无法清 0，只有复位后硬件清 0。<br>RO_LOCK[0]对应 RO[0]，RO_LOCK[1]对应 RO[1]，以此类推。 |

# 28. 真随机数发生器(TRNG)

## 28.1. TRNG 简介

TRNG 单元用于产生真随机数序列。

## 28.2. TRNG 主要特性

- 一次工作产生 128-bit 真随机数序列;
- 可配置随机数生成后产生 CPU 中断请求。

## 28.3. TRNG 功能描述

- 配置控制状态寄存器（TRNG_CSR）及模拟控制寄存器（TRNG_ANA）。
- 将 TRNG_CSR[0]清 “0”，TRNG 单元开始产生随机数。
- 轮询 TRNG_CSR[0]，硬件置“1”表示随机数生成完成（中断模式中，随机数生成后产生中断）。
- 连续读 TRNG_DATA 4 次获取 128bit 随机数。
- 循环步骤 2~4 获取更多随机数。

## 28.4. TRNG 寄存器

### 28.4.1. 控制状态寄存器（RNG_CSR）

偏移地址： 0x00

复位值： 0x0000 0060

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | INTP_EN | Reserved | | CHI_FAIL | TOT_FAIL | TOTD_FAIL | RNG_S128 |
| | | | | | | | | | rw | | | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:7 | 保留 | 必须保持默认值 |
| 位 6 | INTP_EN | 0：不产生中断;<br>1：产生中断。 |
| 位 5:4 | 保留 | 必须保持默认值 |
| 位 3 | CHI_FAIL | 1：真随机发生器卡方检测出错<br>0：无卡方错误 |
| 位 2 | TOT_FAIL | 1：真随机发生器 TOT 检测出错<br>0：无 TOT 错误 |
| 位 1 | TOTD_FAIL | 1：伪随机发生器 TOT 检测出错 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 0：无 TOT 错误 |
| 位 0 | RNG_S128 | RNG0,128-bit 随机数状态位，硬件置 1 软件清 0，清 0 后自动开始新的 128-bit 随机数生成：<br>0：128-bit 随机数未生成；<br>1：128-bit 随机数已生成。 |

### 28.4.2. 数据寄存器（RNG_DATA）

偏移地址： 0x04

复位值： 0x0000 0000

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:0 | DATA | 32-bit 随机数，需在 RNG_S128 置 1 后读取，连续 4 次读取该寄存器，来获取 128-bit 随机数。 |

### 28.4.3. 模拟控制寄存器（RNG_AMA）

偏移地址： 0x0C

复位值： 0x0000 F4A2

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:16 | 保留 | 必须保持默认值 |
| 位 15:12 | PN_TRNG | 分别对应 4 个模拟随机源的 PD 信号；<br>12bit 对应随机源 0<br>13bit 对应随机源 1<br>14bit 对应随机源 2 ... 以此类推 |
| 位 11:0 | 保留 | 必须保持默认值 |

### 28.4.4. RNG FIFO Index（RNG_INDEX）

偏移地址： 0x10

复位值： 0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31 | FIFO_RDOV | FIFO 读溢出标示，当读取完新生成的 128bit 随机数后，再对 fifo 进行读操作，该位置 1，写操作清 0 |
| 位 30:2 | 保留 | 必须保持默认值 |
| 位 1:0 | RNG_INDEX | FIFO 已读深度 |

### 28.4.5. 伪随机序列寄存器（RNG_PN0）

偏移地址： 0x14

复位值： 0x69D8 4C18

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | PN | 128-bit 伪随机序列[31:0]位，1 个系统 cycle 更新一次。<br>该寄存器初始值可配。 |

### 28.4.6. 伪随机序列寄存器（RNG_PN1）

偏移地址： 0x18

复位值： 0xD447 4AFE

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PN1 | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PN1 | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | PN1 | 128-bit 伪随机序列[63:32]位，1 个系统 cycle 更新一次。<br>该寄存器初始值可配。 |

### 28.4.7. 伪随机序列寄存器（RNG_PN2）

偏移地址： 0x1C

复位值： 0xABD8 4C18

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PN2 | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PN2 | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | PN2 | 128-bit 伪随机序列[95:64]位，1 个系统 cycle 更新一次。<br>该寄存器初始值可配。 |

### 28.4.8. 伪随机序列寄存器（RNG_PN3）

偏移地址： 0x20

复位值： 0xACC7 326C

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PN3 | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PN3 | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | PN3 | 128-bit 伪随机序列[127:96]位，1 个系统 cycle 更新一次。<br>该寄存器初始值可配。 |

# 29. 实时时钟(RTC)

## 29.1. RTC 简介

实时时钟是一个独立的定时器。RTC 模块拥有一组连续计数的计数器。RTC 模块和 RTC 相关配置寄存器都处于电池电源域，即主电源掉电对 RTC 没有任何影响，RTC 依旧保持正常计数。

## 29.2. RTC 特性

- 以秒作为计时单位（通过配置产生秒中断）;
- CPU 独立中断源;
- 32 位闹钟设置寄存器，用于在设定时间处产生中断信号或唤醒 CPU;
- RTC 实时时钟单元，支持报警中断生成;
- 可从芯片管脚输出 32K 时钟。

## 29.3. RTC 寄存器

### 29.3.1. RTC 控制状态寄存器（RTC_CS）

偏移地址：0xA0

复位值：0x0000 0008

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | | | | | CLR | RDY | INEN | RDLOCK | INTP |
| | | | | | | | | | | | rw | ro | rw | rw | ro |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:5 | 保留 | 必须保持复位值 |
| 位 4 | CLR | 0：RTC 正常计数，可从 RTC_TIM 寄存器读取当前计数值;<br>1：RTC 计数器清 0，即 RTC_TIM 寄存器清 0。<br>软件置 1 硬件清 0，即上电后 RTC 就处于计数状态。 |
| 位 3 | RDY | 每次“电池电源域上电”软件对 RTC 进行操作前，需查询 RDY 状态，<br>0：表明 RTC 正在复位，无法对 RTC 进行操作;<br>1：表明 RTC 复位结束，可对 RTC 进行读写操作, |
| 位 2 | INEN | RTC 中断使能。<br>0：关闭中断;<br>1：使能中断。 |
| 位 1 | RDLOCK | RTC 当前值读取锁定位，当 CPU 要读取当前 RTC 的计数值时，需要先将该为置 1，读取当前值后需要将该位置 0。 |
| 位 0 | INTP | RTC 中断标识，只有中断使能后，该位才能置 1<br>0：无中断; |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 1：有中断。 |

### 29.3.2. RTC 计数初始值寄存器（RTC_REF）

偏移地址：0xA4

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:0 | REF | RTC 计时初始值寄存器，仅用于保存计时初始值，不参加计时。<br>实际计时值= REF + RTC_TIM; |

### 29.3.3. RTC 闹钟设置寄存器（RTC_ARM）

偏移地址：0xA8

复位值：0x0000 FFFF

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:0 | ALARM | RTC 闹钟设置寄存器，当 RTC_TIM = RTC_ALARM 时产生中断; |

### 29.3.4. RTC 当前计数值寄存器（RTC_TIM）

偏移地址：0xAC

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:0 | TIM | RTC 32bit 计数器的当前计数值。读取该寄存器前需将 RDLOCK 置 1，读取完后需置 0。<br>实际计时值 = RTC_REF + RTC_TIM。 |

### 29.3.5. RTC 中断清除寄存器（RTC_INTCLR）

偏移地址：0xB0

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:1 | 保留 | 必须保持默认值 |
| 位 0 | INTCLR | 对该寄存器进行写操作可清除中断。 |

# 30. XIP Flash 控制器(XFC)

## 30.1. XFC 简介

芯片集成可串行访问外部 FLASH 器件的控制器。
XFC 控制器支持标准 SPI，Dual SPI 以及 Quad SPI 通讯。

## 30.2. XFC 主要特性

- 支持单线（Single），两线（Double）和四线（Quad）I/O 命令；
- 最高频率支持 120MHz，支持分频；
- 可配置 DMA 控制器进行访问；
- 命令参数/格式、极性/相位等均可配置，具有良好的兼容性。

## 30.3. XFC 寄存器

### 30.3.1. XFC 命令寄存器(XFC_CMDR)

地址偏移： 0x00
复位值： 0x003F0800

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CMD_CODE | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CACHE_READY | Reserved | | | | | | |
| rw | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | ADDR_MODE | BUS_MODE | |
| | | | | | rw | rw | rw |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CMD_FORMAT | | | | DONE | BUSY | Reserved | REQ |
| rw | rw | rw | rw | ro | ro | | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:24 | CMD_CODE | 命令字 |
| 位 23:11 | 保留 | 必须保持复位值 |
| 位 10 | ADDR_MODE | 0：使用 24bit 地址模式访问 Flash<br>1：使用 32bit 地址模式访问 Flash |
| 位 9:8 | BUS_MODE | 00：SPI mode，cmd-addr-data = 1-1-1<br>01：Q-out mode，cmd-addr-data = 1-1-4<br>10：Q-IO mode, cmd-addr-data = 1-4-4 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 11：QPI mode，cmd-addr-data = 4-4-4 |
| 位 7:4 | CMD_FORMAT | 命令格式：<br>0000：8bit cmd<br>0001：8bit cmd + 8bit read reg data<br>0010：8bit cmd + 16bit read reg data<br>0011：8bit cmd + 24bit read reg data<br>0100：8bit cmd + 24bit dummy + 8bit write reg data<br>0101：8bit cmd + addr + 8bit read reg data<br>0110：8bit cmd + addr + 16bit read reg data<br>0111：8bit cmd + 8bit write reg data<br>1000：8bit cmd + 16bit write reg data<br>1001：8bit cmd + addr<br>1010：8bit cmd + addr + read data...<br>1011：8bit cmd + addr + dummy + read data...<br>1100：8bit cmd + addr + 8bit M + dummy + read data...<br>1101：8bit cmd + addr + program data... |
| 位 3 | DONE | 当前命令完成标志 |
| 位 2 | BUSY | 当前命令正在进行 |
| 位 1 | 保留 | 必须保持复位值 |
| 位 0 | REQ | 请求访问 Flash |

### 30.3.2. XFC 地址寄存器 (XFC_ADDRR)

地址偏移： 0x04

复位值： 0x00000000

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:0 | ADDR | 命令执行对应的 Flash 地址 |

### 30.3.3. XFC 读写数据量寄存器(XFC_BNR)

地址偏移： 0x08

复位值： 0x00000000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:29 | 保留 | 必须保持复位值 |
| 位 28:16 | WRITE_BN | 写命令后的写数据字节数量， 1~4096 字节，写‘0’ 表示 4096 |
| 位 15:13 | 保留 | 必须保持复位值 |
| 位 12:0 | READ_BN | 读命令后的读数据字节数量， 1~4096 字节，写‘0’ 表示 4096 |

### 30.3.4. XFC 写数据 FIFO 寄存器(XFC_WFIFOR)

地址偏移： 0x0C

复位值： 0x00000000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | WR_DATA | 写数据 FIFO，按 WRITE_BN 的实际数据量发送到 Flash |

### 30.3.5. XFC 读数据 FIFO 寄存器(XFC_RFIFOR)

地址偏移： 0x10

复位值： 0x00000000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | RD_DATA | 读数据 FIFO，如果 READ_BN 不是 4 的整数倍，LSB 补 0 |

### 30.3.6. XFC 配置寄存器(XFC_CFGR)

地址偏移： 0x14

复位值： 0x00A80087

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| COUNT_1US | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| COUNT_1US | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| SAMPLE_DLY | SAMPLE_PHA | Reserved | | READY | Reserved | PROTOCOL | |
| rw | rw | | | ro | | rw | rw |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DUMMY_CYCLE | | | | FREQ_DIV | | | |
| rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:16 | COUNT_1US | 时钟周期计数 1us 的数值 |
| 位 15 | SAMPLE_DLY | 采样延迟时间：<br>0：延迟 1 个周期采样返回的 Flash 读数据<br>1：延迟 2 个周期采样返回的 Flash 读数据 |
| 位 14 | SAMPLE_PHA | 采样时钟相位选择：<br>0：使用非反相时钟的上升沿采样 Flash 读数据<br>1：使用反相时钟的上升沿采样 Flash 读数据 |
| 位 13:12 | 保留 | 必须保持复位值 |
| 位 11 | READY | 0：Flash 器件初始化未完成，Cache 读操作被暂停<br>1：Flash 器件初始化完成，Cache 可以读 Flash |
| 位 10 | 保留 | 必须保持复位值 |
| 位 9:8 | PROTOCOL | 00：CLPL（clock inactive low）<br>11：CHPH（clock inactive high）<br>01,10：保留 |
| 位 7:4 | DUMMY_CYCLE | DUMMY CYCLE 数量<br>包括了地址后面的 8bit M-code |
| 位 3:0 | FREQ_DIV | QSPI 频率选择<br>QSPI CLK = CPU CLK/(FREQ_DIV+2)，支持 2~16 分频 |

### 30.3.7. XFC 写 REG 寄存器(XFC_WREGR)

地址偏移： 0x18

复位值： 0x00000000

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:0 | WREG_DATA | 写 Flash 寄存器数据 |

### 30.3.8. XFC 读 REG 寄存器(XFC_RREGR)

地址偏移： 0x1C

复位值： 0x00000000

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:0 | RREG_DATA | 读 Flash 寄存器数据 |

### 30.3.9. XFC 状态寄存器(XFC_SR)

地址偏移： 0x30

复位值： 0x00000000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Reserved | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Reserved | TXFIFO_EMPTY | RXFIFO_NEMPTY | TXFIFO_OFLOW | TXFIFO_UFLOW | RXFIFO_OFLOW | RXFIFO_UFLOW | CMD_DONE |
| | ro | ro | ro | ro | ro | ro | ro |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:7 | 保留 | 必须保持复位值 |
| 位 6 | TXFIFO_EMPTY | 当前为编程命令，且 TX FIFO 为空指示<br>0：TX FIFO 非空<br>1：TX FIFO 空 |
| 位 5 | RXFIFO_NEMPTY | 当前为读命令，且 RX FIFO 非空指示<br>0：RX FIFO 空<br>1：RX FIFO 非空 |
| 位 4 | TXFIFO_OFLOW | TX FIFO 上溢指示<br>0：TX FIFO 发生上溢<br>1：TX FIFO 未发生上溢 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 3 | TXFIFO_UFLOW | TX FIFO 下溢指示<br>0：TX FIFO 发生下溢<br>1：TX FIFO 未发生下溢 |
| 位 2 | RXFIFO_OFLOW | RX FIFO 上溢指示<br>0：RX FIFO 发生上溢<br>1：RX FIFO 未发生上溢 |
| 位 1 | RXFIFO_UFLOW | RX FIFO 下溢指示<br>0：RX FIFO 发生下溢<br>1：RX FIFO 未发生下溢 |
| 位 0 | CMD_DONE | 命令完成指示 |

### 30.3.10. XFC 状态清除寄存器(XFC_SCR)

地址偏移： 0x34

复位值： 0x00000000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | TXFIFO_ECLR | RXFIFO_NECLR | TXFIFO_OFCLR | TXFIFO_UFCLR | RXFIFO_OFCLR | RXFIFO_UFCLR | CMD_DONE_CLR |
| | wo | wo | wo | wo | wo | wo | wo |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:7 | 保留 | 必须保持复位值 |
| 位 6 | TXFIFO_ECLR | 清除 TX FIFO 空指示 |
| 位 5 | RXFIFO_NECLR | 清除 RX FIFO 非空指示 |
| 位 4 | TXFIFO_OFCLR | 清除 TX FIFO 上溢指示 |
| 位 3 | TXFIFO_UFCLR | 清除 TX FIFO 下溢指示 |
| 位 2 | RXFIFO_OFCLR | 清除 RX FIFO 上溢指示 |
| 位 1 | RXFIFO_UFCLR | 清除 RX FIFO 下溢指示 |
| 位 0 | CMD_DONE_CLR | 清除命令完成指示 |

### 30.3.11. XFC 休眠配置寄存器(XFC_LPCFGR)

地址偏移： 0x38

复位值： 0xABB92BEB

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:24 | RLS_DP_CMD | Flash Release Power down 命令 |
| 位 23:16 | ENTER_DP_CMD | Flash Deep Power-down 命令 |
| 位 15 | FDPD | 0：系统休眠时 XFC 接口自动发 Deep Power-down 命令，Flash 也进入低功耗模式，对应唤醒后 XFC 接口自动发 Release Power down 命令<br>1：系统休眠时 XFC 接口不自动发 Deep Power-down 命令，Flash 不进入低功耗模式 |
| 位 14:0 | 保留 | 必须保持复位值 |

### 30.3.12. XFC DMA 配置寄存器(XFC_DMACR)

地址偏移： 0x3C

复位值： 0x00004400

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:17 | 保留 | 必须保持复位值 |
| 位 16:12 | DMA_RXTHR | DMA 接收门限阈值 |
| 位 11:8 | DMA_TXTHR | DMA 发送门限阈值 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 7:2 | 保留 | 必须保持复位值 |
| 位 1 | DMA_RXEN | 使能 RX DMA，写‘1’后如果 RX FIFO 数据量大于等于 DMA_RXTHR 则产生 DMA 请求 |
| 位 0 | DMA_TXEN | 使能 TX DMA，写‘1’后如果 TX FIFO 数据量小于 DMA_TXTHR 则产生 DMA 请求 |

### 30.3.13. XFC FIFO 配置寄存器(XFC_FCR)

地址偏移： 0x40

复位值： 0x00400020

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| TX_FIFO_FLUSH | Reserved | | | RX_FIFO_NUM | | | |
| wc | | | | ro | ro | ro | ro |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| RX_FIFO_NUM | TX_FIFO_EMPTY | TX_FIFO_FULL | TX_FIFO_NUM | | | | |
| ro | ro | ro | ro | ro | ro | ro | ro |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RX_FIFO_FLUSH | Reserved | | | | | | |
| wc | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | RX_FIFO_EMPTY | RX_FIFO_FULL | Reserved | | | |
| | | ro | ro | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31 | TX_FIFO_FLUSH | 写‘1’ 清空 TX FIFO |
| 位 30:28 | 保留 | 必须保持复位值 |
| 位 27:23 | RX_FIFO_NUM | RX FIFO 中的数据量 |
| 位 22 | TX_FIFO_EMPTY | TX FIFO 空指示<br>0：TX FIFO 非空<br>1：TX FIFO 为空 |
| 位 21 | TX_FIFO_FULL | TX FIFO 满指示<br>0：TX FIFO 未满<br>1：TX FIFO 满 |
| 位 20:16 | TX_FIFO_NUM | TX FIFO 中的数据量 |
| 位 15 | RX_FIFO_FLUSH | 写‘1’ 清空 RX FIFO |
| 位 14:6 | 保留 | 必须保持复位值 |
| 位 5 | RX_FIFO_EMPTY | RX FIFO 空指示<br>0：RX FIFO 非空<br>1：RX FIFO 为空 |
| 位 4 | RX_FIFO_FULL | RX FIFO 满指示<br>0：RX FIFO 未满<br>1：RX FIFO 满 |
| 位 3:0 | 保留 | 必须保持复位值 |

# 31. SDRAM 控制器(SDRAM)

## 31.1. SDRAM 简介

SDRAM 控制器支持外接两个 SDR-SDRAM 器件。

## 31.2. SDRAM 主要特性

- 支持 12bits 地址线;
- 支持 16bits 数据位宽的 SDR-SDRAM;
- 可编程的行大小;
- 可编程的列大小;
- 可编程的内部存储区域数量;
- 支持最多 2 个片选。

## 31.3. SDRAM 的功能描述

### 31.3.1. SDRAM 上电初始化

当系统上电复位或者置位 SDRAM_SCTLR 寄存器的 INIT 位时(初始化过程结束后,此位硬件清 0), SDRAM 控制器会执行初始化过程。

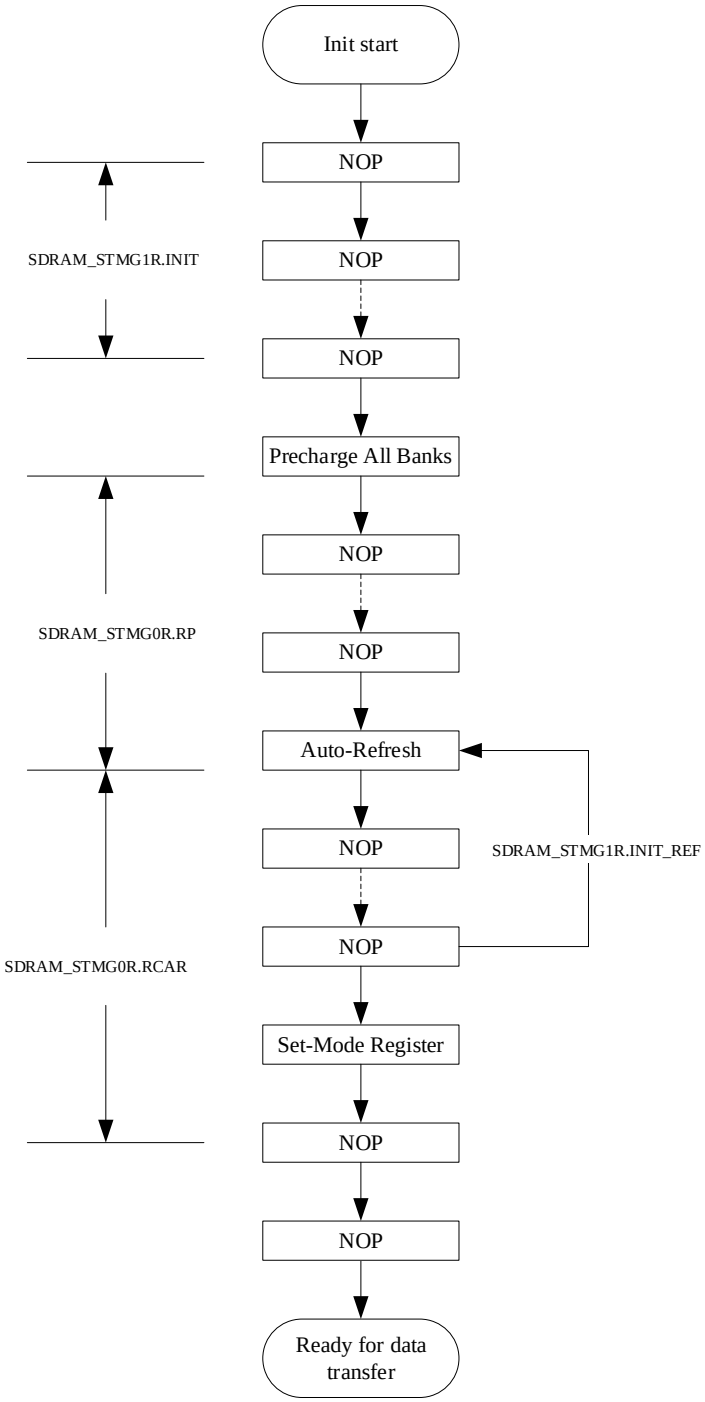



图 31-1 软件初始化流程

软件初始化流程:

- 将存储器设备的特性编程到 SDRAM_SCONR、SDRAM_STMG0R 以及 SDRAM_STMG1R 寄存器中。
- 编程 SDRAM_SREFR 刷新间隔寄存器，其值必须与存储器设备相适应。
- 根据存储器的大小以及所使用的片选个数编程 SDRAM_SCSLR0 和 SDRAM_SCSLR1 寄存器，用于配置存储器映射的内存基地址。
- 置位 SDRAM_SCTLR.INIT 位,使控制器进行初始化并设置存储器的 Mode 寄存器。
- 等待控制器初始化完成(SDRAM_SCTLR.INIT 位清零)。

### 31.3.2. SDRAM 的模式寄存器配置

当控制器上电复位或者置位 SDRAM_SCTLR.INIT 时，自动配置存储器设备的模式寄存器，控制器会将模式寄存器的突发长度固定配置为 0b010(burst length 4)，寻址模式固定为 0b0(sequential)。其他的字段会根据 SDRAM 控制寄存器和 SDRAM 时序寄存器相关字段确定。

### 31.3.3. 自动刷新

**SDRAM_SREFR = (SDRAM 刷新周期 / 行数) – 50**

例如：存储器设备$t_{REF}$=64ms，8K 行，控制器频率为 90MHz

SDRAM 的刷新周期 = 64ms / 8192 行 = 7.81us

SDRAM_SREFR = 7.81us × 90MHz = 702.9

在此基础上预留 50 个周期的裕量，所以：

SDRAM_SREFR = 702 – 50 = 652

### 31.3.4. 低功耗模式

支持两种低功耗模式：

- 自刷新模式

  由存储器设备自身执行自动刷新以保留数据,无需外部时钟。

- 掉电模式

  由控制器执行自动刷新。

**自刷新模式**

置位 SDRAM_SCTLR.SREF 后 SDRAM 会进入自刷新模式，软件清零 SDRAM_SCTLR.SREF 后 SDRAM 会退出自刷新模式；SDRAM_SCTLR.SREFS 位指示当前 SDRAM 的自刷新模式状态；SDRAM_SCTLR.FREFAS 和 SDRAM_SCTLR.FREFBS 位可以选择 SDRAM 进入自刷新和退出自刷新模式时行刷新模式：仅刷新一行或者刷新所有的行。

**掉电模式**

置位 SDRAM_SCTLR.PD 后 SDRAM 会进入掉电模式，清零 SDRAM_SCTLR.PD 后 SDRAM 会退出掉电模式。

掉电模式下控制器仍会自动刷新存储器设备，自刷新后仍会回到掉电模式。此时仍然可以读写 SDRAM，当读写完成后设备仍会回到掉电模式。

### 31.3.5. SDRAM 的地址映射

MH2435 为控制器分配的内存地址范围是：0x60000000 – 0x67FFFFFF。则 SDRAM_SCSLRx(x=0/1)寄存器范围必须设置在此范围内。

通常 SDRAM_SCSLR0 设置为 0x60000000,SDRAM_SCSLR1 设置为紧接着 CS0 的存储器设备的内存大小的地址处。

必须正确设置 SDRAM_SCSLRx(x=0/1)寄存器。

## 31.4. SDRAM 寄存器

本节对 SDRAM 寄存器详细介绍

仅可通过字（32 位）对 SDRAM 寄存器进行访问。

### 31.4.1. SDRAM 配置寄存器(SDRAM_SCONR)

偏移地址： 0x00

复位值:

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | DWID[1:0] | | CWID[3:0] | | | | RWID[3:0] | | | | BWID[1:0] | | Reserved | | |
| | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:15 | 保留 | 必须保持复位值。 |
| 位 14:13 | DWID[1:0] | 数据总线宽度<br>00：16bits<br>01：保留<br>10：保留<br>11：保留 |
| 位 12:9 | CWID[3:0] | 列地址位数<br>15：保留<br>7-14：8-15bits<br>0-6：保留 |
| 位 8:5 | RWID[3:0] | 行地址位数<br>10-15: 11-16bits<br>0-10: 保留 |
| 位 4:3 | BWID[1:0] | 内部存储区域数量<br>0: 1bits<br>1: 2bits<br>2-3: 保留 |
| 位 2:0 | 保留 | 必须保持复位值。 |

### 31.4.2. SDRAM 时序寄存器 0(SDRAM_STMG0R)

偏移地址： 0x04

复位值:

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| XSR[8:4] | | | | | CAS[2] | RC[3:0] | | | | XSR[3:0] | | | | RCAR[3:2] | |
| rw | rw | rw | rw | rw | Rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RCAR[1:0] | | WR | | RP[2:0] | | | RCD[2:0] | | | RAS[3:0] | | | | CAS[1:0] | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 25:22 | RC | 激活命令之间的延迟，以存储器时钟周期数表示<br>0：1 个周期<br>1：2 个周期<br>…<br>15：16 个周期 |
| 位 31:27<br>位 21:18 | XSR[8:0] | 退出自刷新到激活命令或者退出自刷新命令到自动刷新命令之间的延时，以存储器时钟周期数表示<br>0：1 个周期<br>1：2 个周期<br>…<br>511: 512 个周期 |
| 位 17:14 | RCAR[3:0] | 两个自动刷新命令之间的延时，以存储器时钟周期数表示<br>0：1 个周期<br>1：2 个周期<br>…<br>15：16 个周期 |
| 位 13:12 | WR[1:0] | 写命令与预充电命令的延时，以存储器时钟周期数表示<br>0：1 个周期<br>1：2 个周期<br>2：3 个周期<br>3：4 个周期 |
| 位 11:9 | RP[2:0] | 两个预充电命令之间的延时，以存储器时钟周期数表示<br>0：1 个周期<br>1：2 个周期<br>…<br>7：8 个周期 |
| 位 8:6 | RCD[2:0] | 激活命令与读/写命令之间的延时，以存储器时钟周期数表示<br>0：1 个周期<br>1：2 个周期<br>…<br>7：8 个周期 |
| 位 5:2 | RAS[3:0]: | 激活命令到预充电命令之间的延时，以存储器时钟周期数表示<br>0：1 个周期<br>1：2 个周期<br>…<br>15： 16 个周期 |
| 位 1:0 | CAS[2:0] | CAS 延迟，读命令到第一个有效数据之间的延时，以存储器时钟周期数表示<br>0：1 个周期<br>1：2 个周期 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 2：3 个周期<br>3：4 个周期<br>4-7：保留 |

### 31.4.3. SDRAM 时序寄存器 1(SDRAM_STMG1R)

偏移地址： 0x08

复位值:

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:20 | 保留 | 必须保持复位值。 |
| 位 19:16 | INIT_REF[3:0] | 初始化期间的自动刷新命令个数<br>0：1 个<br>1：2 个<br>…<br>15: 16 个 |
| 位 15:0 | INIT[15:0] | 上电后执行命令之前 SDRAM 保持输入稳定的时钟周期数 |

### 31.4.4. SDRAM 控制寄存器(SDRAM_SCTLR)

偏移地址： 0x0C

复位值:

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | SREFS | Reserved | SMD | RPIPE[2:0] | | | FREFAS | FREFBS | PRCAL | PD | SREF | INIT |
| | | | | r | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:12 | 保留 | 必须保持复位值。 |
| 位 11 | SREFS | 只读.<br>1：SDRAM 处于自刷新模式<br>0：SDRAM 未处于自刷新模式 |
| 位 10 | 保留 | 必须保持复位值。 |
| 位 9 | SMD | 1：控制器更新 SDRAM 的模式寄存器<br>当更新完成时,此位由硬件清 0 |
| 位 8:6 | RPIPE[2:0] | 在 CAS 延迟后多少个时钟周期读取数据<br>0：0 个周期 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 1：1 个周期<br>…<br>7：7 个周期 |
| 位 5 | FREFAS | 退出自刷新模式的刷新配置<br>0：退出自刷新模式后,仅刷新一行<br>1：退出自刷新模式后,刷新所有的行 |
| 位 4 | FREFBS | 进入自刷新模式的刷新配置<br>0：进入自刷新模式前,仅刷新一行<br>1：进入自刷新模式前,刷新所有的行 |
| 位 3 | PRCAL | 预充电模式<br>0：立即预充电；读写命令后立即行预充电<br>1：延时预充电；读写命令后保持行打开 |
| 位 2 | PD | 掉电模式配置<br>0：不进入掉电模式<br>1：进入掉电模式 |
| 位 1 | SREF | 自刷新模式配置<br>0：不进入自刷新模式<br>1：进入自刷新模式 |
| 位 0 | INIT | 初始化命令配置<br>1：控制器发送初始化命令；当初始化序列完成后此位由硬件清 0 |

### 31.4.5. SDRAM 刷新间隔寄存器(SDRAM_SREFR)

偏移地址： 0x10

复位值:

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:16 | 保留 | 必须保持复位值。 |
| 位 15:0 | REF | 刷新时间间隔，以存储器时钟周期数表示<br>REF = (SDRAM 刷新周期 / 行数) - 50 |

### 31.4.6. SDRAM 片选寄存器 x(SDRAM_SCSLRx)(x=0,1)

偏移地址： 0x14，0x18

复位值:

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:16 | CSRx | CSx(x=0,1)对应的基地址<br>范围必须在 0x6000:0x67FF 之间 |
| 位 15:0 | 保留 | 必须保持复位值。 |

# 32. 图像处理加速器(DMA2D)

## 32.1. DMA2D 简介

DMA2D 是专用于图像处理的专业 DMA。它可以执行下列操作:

- 用特定颜色填充目标图像的一部分或全部。
- 将源图像的一部分（或全部）复制到目标图像的一部分（或全部）中。
- 通过像素格式转换将源图像的一部分或全部复制到目标图像的一部分或全部中。
- 将像素格式不同的两个源图像部分和/ 或全部混合，再将结果复制到颜色格式不同的部分或整个目标图像中。

在索引颜色模式或直接颜色模式下，支持所有经典颜色编码方案，包括处理 JPEG 解码器输出的基于块的 YCbCr，并支持每像素 4 位到最高 32 位。 DMA2D 自身具有专门的 CLUT（颜色查找表）存储器。

## 32.2. DMA2D 主要特性

DMA2D 的主要特性有:

- 采用单 AHB 主设备总线架构。
- AHB 从设备编程接口支持 8/16/32 位访问 （32 位的 CLUT 访问除外）。
- 用户可编程工作区大小。
- 用户可编程源区域和目标区域的偏移。
- 用户可编程整个存储空间的源地址和目标地址。
- 最多支持 2 个源的混合操作。
- Alpha 值可修改（源值、固定值或调制的值）。
- 用户可编程源颜色格式和目标的颜色格式。
- 采用间接或直接颜色编码时，支持多达 11 种颜色格式，且支持每像素 4 位到最高 32 位。
- 支持基于块的（8x8）YCbCr，具有 4:4:4、4:2:2 和 4:2:0 色度抽样。
- 间接颜色模式下使用 2 个内部存储器存储 CLUT。
- 通过 CPU 自动加载 CLUT 或对 CLUT 进行编程。
- 用户可编程 CLUT 大小。
- 使用内部定时器控制 AHB 带宽。
- 支持 6 种工作模式：寄存器到存储器、存储器到存储器、存储器到存储器且支持像素格式转换、存储器到存储器且支持像素格式转换和混合、存储器到存储器前景固定且支持背景像素格式转换和混合、存储器到存储器背景固定且支持前景像素格式转换和混合。
- 可使用固定颜色进行区域填充。
- 可从一个区域复制到另一个区域。
- 在源图像和目标图像之间进行复制时进行像素格式转换。
- 支持从颜色格式不同的两幅源图像复制并混合。
- 输出缓冲区字节交换，支持通过并行接口刷新显示器。
- 可中止并挂起 DMA2D 操作。

- 支持在传输用户可编程的目标行时生成水印中断。
- 支持发生总线错误或访问冲突时生成中断。
- 支持处理完成时生成中断。
- 支持图像 0°、90°、180°和 270°旋转。

## 32.3. DMA2D 功能说明

### 32.3.1. 概述

DMA2D 控制器执行直接存储器传输。作为一个 AHB 主设备，它可以控制 AHB 总线矩阵来启动 AHB 事务。

DMA2D 可在以下六种模式下工作：

- 寄存器到存储器；
- 存储器到存储器；
- 存储器到存储器并执行像素格式转换；
- 存储器到存储器并执行像素格式转换和混合；
- 存储器到存储器前景固定并执行背景像素格式转换和混合；
- 存储器到存储器背景固定并执行前景像素格式转换和混合；
- AHB 从设备端口用于编程 DMA2D 控制器。

### 32.3.2. DMA2D 控制

通过 DMA2D 控制寄存器 (DMA2D_CR) 配置 DMA2D 控制器，允许选择：

用户应用可以执行下列操作：

- 选择工作模式；
- 使能/禁止 DMA2D 中断；
- 启动/挂起/中止进行中的数据传输。

### 32.3.3. DMA2D 前景层 FIFO 和背景层 FIFO

DMA2D 前景层 (FG) FG FIFO 和背景层 (BG) FIFO 获取要复制和/或处理的输入数据。这些 FIFO 根据相应像素格式转换器 (PFC) 中定义的颜色格式获取像素。

通过如下一组寄存器对它们进行编程：

- DMA2D 前景层存储器地址寄存器 (DMA2D_FGMAR)；
- DMA2D 前景层偏移寄存器 (DMA2D_FGOR)；
- DMA2D 背景层存储器地址寄存器 (DMA2D_BGMAR)；
- DMA2D 背景层偏移寄存器 (DMA2D_BGBOR)；
- DMA2D 行数寄存器（行数和每行像素数）(DMA2D_NLR)；
- DMA2D 在寄存器到存储器模式下工作时，不激活任何 FIFO。

DMA2D 在存储器到存储器模式下工作时（无像素格式转换和混合操作），仅激活 FG FIFO，并将其用作缓冲区。

DMA2D 在存储器到存储器模式下工作时并支持像素格式转换时（无混合操作），不会激活 BG FIFO。

### 32.3.4. DMA2D 前景层和背景层像素格式转换器 (PFC)

DMA2D 前景层和背景层像素格式转换器 (PFC) 执行像素格式转换，以生成每像素 32 位的值。PFC 还能够修改 alpha 通道。

转换器在第一阶段转换颜色格式。前景层像素和背景层像素的原始颜色格式分别通过 DMA2D_FGPFCCR 和 DMA2D_BGPFCCR 的 CM[3:0] 位来配置。

下表给出了支持的输入格式。

表 32-1 输入时支持的颜色模式

| CM[3:0] | 颜色模式 |
|---|---|
| 0 | ARGB8888 |
| 1 | RGB888 |
| 10 | RGB565 |
| 11 | ARGB1555 |
| 100 | ARGB4444 |
| 101 | L8 |
| 110 | AL44 |
| 111 | AL88 |
| 1000 | L4 |
| 1001 | A8 |
| 1010 | A4 |
| 1011 | YCbCr(只支持前景) |

颜色格式的编码方式如下：

- Alpha 值字段：透明
- 0xFF 值对应不透明像素，0x00 对应透明像素。
- R 字段代表红色
- G 字段代表绿色
- B 字段代表蓝色
- L 字段：亮度

该字段是 CLUT 的索引，用于检索三个/四个 RGB/ARGB 分量。

如果原始格式为直接颜色模式，则通过将 MSB 复制到 LSB 扩展为每通道 8 位。这可以确保转换具有良好的线性。

如果原始格式不包括 alpha 通道，则会自动将 alpha 值设为 0xFF（不透明）。

如果原始格式为间接颜色模式，则需要使用 CLUT，并且每个像素格式转换器与一个 256 个 32 位条目的 CLUT 相关联。

对于特定的 alpha 模式 A4 和 A8，既不存储颜色信息，也不编制索引。用于生成图像的颜色是固定的，并且在 DMA2D_FGCOLR 寄存器中定义前景层像素的颜色，在 DMA2D_BGCOLR 寄存器中定义背景层像素的颜色。

系统存储器中的字段顺序如下表所示。

表 32-2 存储器中的数据顺序

| 颜色模式 | @ + 3 | @ + 2 | @ + 1 | @ + 0 |
|---|---|---|---|---|
| ARGB8888 | $A_0$[7:0] | $R_0$[7:0] | $G_0$[7:0] | $B_0$[7:0] |
| | $B_1$[7:0] | $R_0$[7:0] | $G_0$[7:0] | $B_0$[7:0] |

| 颜色模式 | @ + 3 | @ + 2 | @ + 1 | @ + 0 |
|---|---|---|---|---|
| RGB888 | $G_2$[7:0] | $B_2$[7:0] | $R_1$[7:0] | $G_1$[7:0] |
| | $R_3$[7:0] | $G_3$[7:0] | $B_3$[7:0] | $R_2$[7:0] |
| RGB565 | $R_1$[4:0]$G_1$[5:3] | $G_1$[2:0]$B_1$[4:0] | $R_0$[4:0]$G_0$[5:3] | $G_0$[2:0]$B_0$[4:0] |
| ARGB1555 | $A_1$[0]$R_1$[4:0]$G_1$[4:3] | $G_1$[2:0]$B_1$[4:0] | $A_0$[0]$R_0$[4:0]$G_0$[4:3] | $G_0$[2:0]$B_0$[4:0] |
| ARGB4444 | $A_1$[3:0]$R_1$[3:0] | $G_1$[3:0]$B_1$[3:0] | $A_0$[3:0]$R_0$[3:0] | $G_0$[3:0]$B_0$[3:0] |
| L8 | $L_3$[7:0] | $L_2$[7:0] | $L_1$[7:0] | $L_0$[7:0] |
| AL44 | $A_3$[3:0]$L_3$[3:0] | $A_2$[3:0]$L_2$[3:0] | $A_1$[3:0]$L_1$[3:0] | $A_0$[3:0]$L_0$[3:0] |
| AL88 | $A_1$[7:0] | $L_1$[7:0] | $A_0$[7:0] | $L_0$[7:0] |
| L4 | $L_7$[3:0]$L_6$[3:0] | $L_5$[3:0]$L_4$[3:0] | $L_3$[3:0]$L_2$[3:0] | $L_1$[3:0]$L_0$[3:0] |
| A8 | $A_3$[7:0] | $A_2$[7:0] | $A_1$[7:0] | $A_0$[7:0] |
| A4 | $A_7$[3:0]$A_6$[3:0] | $A_5$[3:0]$A_4$[3:0] | $A_3$[3:0]$A_2$[3:0] | $A_1$[3:0]$A_0$[3:0] |

通过 ARGB8888 模式支持按 32 位对齐 24 位 RGB888。

生成 32 位值后，即可根据 DMA2D_FGPFCCR/DMA2D_BGPFCCR 寄存器的 AM[1:0] 字段修改 alpha 通道，如下表：Alpha 模式配置所示。

Alpha 通道可以：

- 保持不变（不做修改），
- 替换为 DMA2D_FGPFCCR/DMA2D_BGPFCCR 的 ALPHA[7:0] 值。
- 或替换为原始 alpha 值与 DMA2D_FGPFCCR/DMA2D_BGPFCCR 的 ALPHA [7:0] 值除以 255 所得商的乘积。

表 32-3 Alpha 模式配置

| AM[1:0] | Alpha 模式 |
|---|---|
| 0 | 不做修改 |
| 1 | 替换为 DMA2D_xxPFCCR 中的值 |
| 10 | 替换为原始值与 DMA2D_xxPFCCR 中的值/255 所得商的乘积 |
| 11 | 保留 |

### 32.3.5. DMA2D 前景层 FIFO 和背景层 CLUT 接口

CLUT 接口可管理对 CLUT 存储器的访问以及 CLUT 的自动加载。

支持如下三种访问：

- PFC 在像素格式转换期间读取 CLUT
- CPU 对 CLUT 进行数据的读取或写入时，通过 AHB 从设备端口访问 CLUT
- 执行自动加载 CLUT 时，通过 AHB 主设备端口进行 CLUT 写入

可通过两种不同方法执行 CLUT 存储器加载：

- 自动加载

  加载 CLUT 时应遵守以下顺序：

  - 将 CLUT 地址编程到 DMA2D_FGCMAR 寄存器（前景层 CLUT）或 DMA2D_BGCMAR 寄存器（背景层 CLUT）。
  - 将 CLUT 大小编程到 DMA2D_FGPFCCR 寄存器（前景层 CLUT）或 MA2D_BGPFCCR 寄存器（背景层 CLUT）的 S[7:0] 字段。
  - 将 MA2D_FGPFCCR 寄存器（前景层 CLUT）或 DMA2D_BGPFCCR 寄存器（背景层 CLUT）的 START 位置 1 以启动传输。自动加载过程期间，不可通过 CPU 访问 CLUT。如果出现冲突，若 DMA2D_CR 中 CAEIE 被置 1，则发生 CLUT 访问错误中断。

- 手动加载

  应用程序必须通过 DMA2D AHB 从设备端口手动编程本地 CLUT 存储器映射到的 CLUT。前景层 CLUT 从偏移地址 0x0400 开始，背景层 CLUT 从偏移地址 0x0800 开始。

  CLUT 格式可以是 24 位或 32 位。通过 DMA2D_FGPFCCR 寄存器（前景层 CLUT）或 DMA2D_BGPFCCR 寄存器（背景层 CLUT）的 CCM 位配置格式，如下表：支持的 CLUT 颜色模式所示。

表 32-4 支持的 CLUT 颜色模式

| CCM | CLUT 颜色模式 |
|---|---|
| 0 | 32 位 ARGB8888 |
| 1 | 24 位 RGB888 |

下表：存储器中的 CLUT 数据顺序给出了系统存储器中 CLUT 数据的组织方式。

表 32-5 存储器中的 CLUT 数据顺序

| CLUT 颜色模式 | @ + 3 | @ + 2 | @ + 1 | @ + 0 |
|---|---|---|---|---|
| ARGB8888 | $A_0$[7:0] | $R_0$[7:0] | $G_0$[7:0] | $B_0$[7:0] |
| RGB888 | $B_1$[7:0] | $R_0$[7:0] | $G_0$[7:0] | $B_0$[7:0] |
| | $G_2$[7:0] | $B_2$[7:0] | $R_1$[7:0] | $G_1$[7:0] |
| | $R_3$[7:0] | $G_3$[7:0] | $B_3$[7:0] | $R_2$[7:0] |

### 32.3.6. DMA2D 混合器

DMA2D 混合器成对混合源像素以计算结果像素。

混合将按以下公式执行：

$$with\ \alpha_{Mult} = \frac{\alpha_{FG} \times \alpha_{BG}}{255}$$

$$\alpha_{OUT} = \alpha_{FG} + \alpha_{BG} - \alpha_{Mult}$$

with C = R or G or B

$$C_{OUT} = \frac{C_{FG} \times \alpha_{FG} + C_{BG} \times \alpha_{BG} - C_{BG} \times \alpha_{Mult}}{\alpha_{OUT}}$$

*Division is rounded to the nearest lower integer*

混合器不需要任何配置寄存器。是否使用混合器取决于 DMA2D_CR 寄存器的 MODE[1:0] 字段中定义的 DMA2D 工作模式。

### 32.3.7. DMA2D 输出 PFC

输出 PFC 将像素格式从 32 位转换为指定的输出格式，输出格式在 DMA2D 输出像素格式转换器配置寄存器 (DMA2D_OPFCCR) 的 CM[2:0] 字段中定义。

下表给出了支持的输出格式。

表 32-6 输出时支持的颜色模式

| CM[2:0] | 颜色模式 |
|---|---|
| 0 | ARGB8888 |
| 1 | RGB888 |
| 10 | RGB565 |
| 11 | ARGB1555 |
| 100 | ARGB4444 |

### 32.3.8. DMA2D 输出 FIFO

输出 FIFO 根据输出 PFC 中定义的颜色格式对像素进行编程。

通过如下一组寄存器定义目标区域：

- DMA2D 输出存储器地址寄存器 (DMA2D_OMAR)；
- DMA2D 输出偏移寄存器 (DMA2D_OOR)；
- DMA2D 行数寄存器（行数和每行像素数） (DMA2D_NLR)。

如果 DMA2D 在寄存器到存储器模式下工作，则配置的输出矩形将以 DMA2D 输出颜色寄存器 (DMA2D_OCOLR) 中指定的颜色填充，该寄存器中包含固定的 32 位、24 位或 16 位值。通过 DMA2D_OPFCCR 寄存器的 CM[2:0] 字段选择格式。

将按照下表中定义的顺序在存储器中存储数据。

表 32-7 存储器中的数据顺序

| 颜色模式 | @ + 3 | @ + 2 | @ + 1 | @ + 0 |
|---|---|---|---|---|
| ARGB8888 | $A_0$[7:0] | $R_0$[7:0] | $G_0$[7:0] | $B_0$[7:0] |
| RGB888 | $B_1$[7:0] | $R_0$[7:0] | $G_0$[7:0] | $B_0$[7:0] |
| | $G_2$[7:0] | $B_2$[7:0] | $R_1$[7:0] | $G_1$[7:0] |
| | $R_3$[7:0] | $G_3$[7:0] | $B_3$[7:0] | $R_2$[7:0] |
| RGB565 | $R_1$[4:0]$G_1$[5:3] | $G_1$[2:0]$B_1$[4:0] | $R_0$[4:0]$G_0$[5:3] | $G_0$[2:0]$B_0$[4:0] |
| ARGB1555 | $A_1$[0]$R_1$[4:0]$G_1$[4:3] | $G_1$[2:0]$B_1$[4:0] | $A_0$[0]$R_0$[4:0]$G_0$[4:3] | $G_0$[2:0]$B_0$[4:0] |
| ARGB4444 | $A_1$[3:0]$R_1$[3:0] | $G_1$[3:0]$B_1$[3:0] | $A_0$[3:0]$R_0$[3:0] | $G_0$[3:0]$B_0$[3:0] |

通过 ARGB8888 模式支持按 32 位对齐 RGB888。

### 32.3.9. DMA2D 输出 FIFO 字节重新排序

输出 FIFO 字节可以重新排序，以支持通过并行接口（F（S）MC）。

字节的重新排序可以使用以下方法完成：

- RBS 位用于交换红色和蓝色；
- SB 位在输出 FIFO 中逐字节交换。

当字节交换被激活时（DMA2D_OPFCR 的 SB 字段被设置）每行像素（DMA2D_NLR 的 PL 字段）必须是偶数，并且输出存储器地址（DMA2D_OMAR 的 MA 字段）必须是偶数，并且输出行偏移量以字节为单位计算（由 DMA2D_CR 的 LOM 字段和 DMA2D_OOR 值的 LO 字段产生）必须为偶数。如果没有将检测到配置错误。

### 32.3.10. DMA2D AHB 主设备端口定时器

AHB 主设备端口内嵌一个 8 位定时器，以便可选择限制交叉开关矩阵的带宽。

此定时器由 AHB 时钟驱动，对两个连续访问之间的死区进行计数。这样可限制带宽的使用。

通过 AHB 主设备端口定时器配置寄存器 (DMA2D_AMPTCR) 配置定时器使能和死区值。

### 32.3.11. DMA2D 事务

DMA2D 事务由给定数目的数据传输序列组成。可通过软件对数据数目和宽度进行编程。

每个 DMA2D 数据传输最多需要 4 个步骤：

- 从 DMA2D_FGMAR 寄存器寻址的存储单元加载数据并按照 DMA2D_FGCR 中的定义进行像素格

式转换。

- 从 DMA2D_BGMAR 寄存器寻址的存储单元加载数据并按照 DMA2D_BGCR 中的定义进行像素格式转换。
- 根据对 alpha 值进行 PFC 操作所得到的 alpha 通道，将所有检索到的像素混合。
- 根据 DMA2D_OCR 寄存器对合成像素进行像素格式转换，然后将数据编程到通过 DMA2D_OMAR 寄存器寻址的存储单元。

### 32.3.12. DMA2D 配置

源和目标数据传输在整个 4 GB 区域（地址范围在 0x0000 0000 和 0xFFFF FFFF 之间）都可以寻址外设和存储器。

DMA2D 可在以下六种模式下工作，通过 DMA2D_CR 寄存器的 MODE[2:0] 位选择工作模式：

- 寄存器到存储器；
- 存储器到存储器；
- 存储器到存储器并执行 PFC；
- 存储器到存储器并执行 PFC 和混合；
- 存储器到存储器前景固定并执行背景像素格式转换和混合；
- 存储器到存储器背景固定并执行前景像素格式转换和混合。

**寄存器到存储器**

寄存器到存储器模式用于以预定义颜色填充用户自定义区域。

颜色格式在 DMA2D_OPFCCR 中设置。

DMA2D 不从任何源获取数据。它只将 DMA2D_OCOLR 寄存器中定义的颜色写入通过 DMA2D_OMAR 寻址以及 DMA2D_NLR 和 DMA2D_OOR 定义的区域。

**存储器到存储器**

在存储器到存储器模式下，DMA2D 不执行任何图形数据转换。前景层输入 FIFO 充当缓冲区，数据从 DMA2D_FGMAR 中定义的源存储单元传输到 DMA2D_OMAR 寻址的目标存储单元。

DMA2D_FGPFCCR 寄存器的 CM[3:0] 位中编程的颜色模式决定输入和输出的每像素位数。

对于要传输的区域大小，源区域大小由 DMA2D_NLR 和 DMA2D_FGOR 寄存器定义，目标区域大小则由 DMA2D_NLR 和 DMA2D_OOR 寄存器定义。

**存储器到存储器并执行 PFC**

此模式下，DMA2D 对源数据执行像素格式转换并将结果存储在目标存储单元。

对于要传输的区域大小，源区域大小由 DMA2D_NLR 和 DMA2D_FGOR 寄存器定义，目标区域大小则由 DMA2D_NLR 和 DMA2D_OOR 寄存器定义。

从 DMA2D_FGMAR 寄存器定义的位置获取数据，并由前景层 PFC 进行处理。原始像素格式通过 DMA2D_FGPFCCR 寄存器配置。

如果原始像素格式是直接颜色模式，则所有颜色通道都扩展到 8 位。

如果像素格式是间接颜色模式，则必须将相关 CLUT 加载到 CLUT 存储器中。

CLUT 加载可按如下顺序自动完成：

- 在 DMA2D_FGCMAR 中设置 CLUT 地址。
- 在 DMA2D_FGPFCCR 寄存器的 CS[7:0] 位设置 CLUT 大小。
- 在 DMA2D_FGPFCCR 寄存器的 CCM 位设置 CLUT 格式（24 或 32 位）。

- 将 DMA2D_FGPFCCR 寄存器的 START 位置 1 启动 CLUT 加载。

CLUT 加载完成时，DMA2D_ISR 寄存器将置位 CTCIF 标志；如果 DMA2D_CR 中的 CTCIE 位置 1，还将产生中断。CLUT 自动加载过程无法与传统的 DMA2D 传输同时进行。

CLUT 还可由 CPU 填充，或由任意其它主设备通过 APB 端口填充。在 DMA2D 传输进行期间和使用 CLUT（间接颜色格式）时无法访问 CLUT。

在颜色转换执行期间，可根据 DMA2D_FGPFCCR 寄存器中编程的值添加或更改 alpha 值。如果原始图像没有 alpha 通道，则会自动添加一个默认的 alpha 值 0xFF 以获得完全不透明的像素。可根据 DMA2D_FGPFCCR 寄存器的 AM[1:0] 位修改 alpha 值：

- 保持不变。
- 替换为 DMA2D_FGPFCCR 寄存器的 ALPHA[7:0] 值中定义的值。
- 替换为原始值与 DMA2D_FGPFCCR 寄存器的 ALPHA[7:0] 值除以 255 所得商的乘积。

结果得到的 32 位数据由 OUT PFC 编码成 DMA2D_OPFCCR 寄存器的 CM[2:0] 字段所指定的格式。输出像素格式不可是间接模式，原因是输出时不支持 CLUT 生成过程。

数据经处理后，将写入 DMA2D_OMAR 寻址的目标存储单元。

**存储器到存储器并执行 PFC 和混合**

此模式下，将在前景层 FIFO 和背景层 FIFO（分别在 DMA2D_FGMAR 和 DMA2D_BGMAR 中定义）获取 2 个源图像。

必须按存储器到存储器模式中所述配置两个像素格式转换器。由于这两个像素格式转换器各自独立且自身具有 CLUT 存储器，因此其配置可以不同。

在每个像素都通过相应的 PFC 转换为 32 位后，将根据以下公式进行混合：

$$with\ \alpha_{Mult} = \frac{\alpha_{FG} \times \alpha_{BG}}{255}$$

$$\alpha_{OUT} = \alpha_{FG} + \alpha_{BG} - \alpha_{Mult}$$

with C = R or G or B

$$C_{OUT} = \frac{C_{FG} \times \alpha_{FG} + C_{BG} \times \alpha_{BG} - C_{BG} \times \alpha_{Mult}}{\alpha_{OUT}}$$

*Division is rounded to the nearest lower integer*

输出 PFC 将根据指定的输出格式对得到的 32 位像素值进行编码，并且编码数据将写入 DMA2D_OMAR 寻址的目标存储单元。

**存储器到存储器前景固定并执行 PFC 和混合**

此模式下，只有 1 个源在后台 FIFO 中从 DMA2D_BGMAR 定义的存储器位置取出。前景色的值由 DMA2D_FGCOLR 寄存器给出，阿尔法值设置为 0xFF（不透明）。

阿尔法值可以根据 DMA2D_FGPFCCR 的 AM[1:0]和 alpha[7:0]字段来替换或修改。

两个像素格式转换器必须按照存储器到存储器模式中的描述进行配置。

它们的配置可能不同，因为每个像素格式转换器独立且有自己的 CLUT 存储器。

一旦每个像素被它们各自的 PFC 转换成 32 位，它们就被混合一起，并且输出 PFC 根据指定的输出格式对得到的 32 位像素值进行编码，并且将数据写入 DMA2D_OMAR 所指向的目的存储器位置。

**存储器到存储器背景固定并执行 PFC 和混合**

此模式中，在前景 FIFO 中，从 DMA2D_FGMAR 定义的存储器位置仅提取 1 个源。

背景色的值由 DMA2D_BGCOLR 寄存器给出，阿尔法值设置为 0xFF（不透明）。

阿尔法值可以根据 DMA2D_BGPFCCR 的 AM[1:0]和 alpha[7:0]字段来替换或修改。

两个像素格式转换器必须按照存储器到存储器模式中的描述进行配置。它们的配置可以不同，因为每个像素格式转换器是独立的，并且具有自己的 CLUT 存储器。

一旦每个像素被它们各自的 PFC 转换成 32 位，它们就被混合在一起，并且输出 PFC 根据指定的输出格式对得到的 32 位像素值进行编码，并且数据被写入 DMA2D_OMAR 所指向的目的存储器位置。

**配置错误检测**

DMA2D 将在每次执行传输前检查配置是否正确。开始新的传输/自动加载时，如果检测到配置错误，硬件将设置配置错误中断标志。如果 DMA2D_CR 寄存器的 CEIE 位置 1，还将产生中断。

可检测到的错误配置如下：

- 前景层 CLUT 自动加载：DMA2D_FGCMAR 的 MA 位与 DMA2D_FGPFCCR 的 CCM 位不匹配；
- 背景层 CLUT 自动加载：DMA2D_BGCMAR 的 MA 位与 DMA2D_BGPFCCR 的 CCM 位不匹配；
- 存储器传输（寄存器到存储器模式除外）：DMA2D_FGMAR 的 MA 位与 DMA2D_FGPFCCR 的 CM 位不匹配；
- 存储器传输（寄存器到存储器模式除外）：DMA2D_FGPFCCR 中的 CM 位无效；
- 存储器传输（寄存器到存储器模式除外）：DMA2D_NLR 的 PL 位为奇，而 DMA2D_FGPFCCR 的 CM 位为 A4 或 L4；
- 存储器传输（寄存器到存储器模式除外）：DMA2D_FGOR 中的 LO 位为奇，而 DMA2D_FGPFCCR 的 CM 位为 A4 或 L4；
- 存储器传输（仅限混合模式）：DMA2D_BGMAR 中的 MA 位与 DMA2D_BGPFCCR 的 CM 位不一致；
- 存储器传输：DMA2D_BGPFCCR 的 CM 无效（仅限混合模式）；
- 存储器传输（仅限混合模式）：DMA2D_NLR 的 PL 位为奇，而 DMA2D_BGPFCCR 的 CM 位为 A4 或 L4；
- 存储器传输（仅限混合模式）：DMA2D_BGOR 的 LO 位为奇，而 DMA2D_BGPFCCR 的 CM 位为 A4 或 L4；
- 存储器传输（存储器到存储器模式除外）：DMA2D_OMAR 的 MA 位与 DMA2D_OPFCCR 的 CM 位不匹配；
- 存储器传输（存储器到存储器模式除外）：DMA2D_OPFCCR 中的 CM 位无效；
- 存储器传输：DMA2D_NLR 中的 NL 位 = 0；
- 存储器传输：DMA2D_NLR 中的 PL 位 = 0；
- 存储器传输：DMA2D_CR 中的 MODE 位无效；
- YCbCr 格式：当 CLUT 开始加载时设置 MDMA2D_FGPFCCR 的 START 位；
- YCbCr 格式：当选择内存到内存模式时；
- YCbCr 格式：当选择 YCbCr4:4:4 并且像素（PL）和行偏移 LO 之和不是像素 8 的倍数时；
- YCbCr 格式：当选择 YCbCr4:2:2 和 YCbCr4:2:0 并且像素（PL）和行偏移 LO 之和不是像素的 16 倍数时。

### 32.3.13. DMA2D 传输控制（启动、挂起、中止和完成）

配置 DMA2D 后，通过将 DMA2D_CR 寄存器的 START 位置 1 可启动传输。传输完成时，START 位自动复位且 DMA2D_ISR 寄存器置位 TCIF 标志。如果 DMA2D_CR 寄存器中的 TCIE 位置 1，还将产生中断。

用户应用程序随时都可通过将 DMA2D_CR 寄存器的 SUSP 位置 1 来挂起 DMA2D。随即可通过将

DMA2D_CR 寄存器的 ABORT 位置 1 中止事务，或通过将 DMA2D_CR 寄存器的 SUSP 位复位重新启动事务。

用户应用程序随时都可通过将 DMA2D_CR 寄存器的 ABORT 位置 1 来中止处理中的事务。在这种情况下，不会置位 TCIF 标志。

还可通过 DMA2D_FGPFCCR 和 DDM12D1_BGPFCCR 寄存器自身的 START 位中止或挂起 CLUT 自动传输过程。

### 32.3.14. 水印

可对水印编程，以在指定行的最后一个像素写入目标存储区域时产生中断。行号在 DMA2D_LWR 寄存器的 LW[15:0] 字段中定义。

在该行的最后一个像素传输完成时，DMA2D_ISR 将置位 TWIF 标志；在 DMA2D_CR 的 TWIE 位置 1 的情况下，还将产生中断。

### 32.3.15. 错误管理

可触发两种错误：

- AHB 主设备端口错误， 通过 DMA2D_ISR 寄存器的 TEIF 标志指示。
- CLUT 访问引发的冲突（CPU 在 CLUT 加载或 DMA2D 传输执行期间尝试访问 CLUT），通过 DMA2D_ISR 寄存器的 CAEIF 标志指示。

这两个标志都与其在 DMA2D_CR 寄存器中的中断使能标志（控制在需要时产生中断的 TEIE 和 CAEIE）相关。

### 32.3.16. AHB 死区

要限制 AHB 带宽的使用，可在两个连续的 AHB 访问之间编程一个死区。可通过将 DMA2D_AMTCR 寄存器中的 EN 位置 1 使能此特性。

死区值存储在 DMA2D_AMTCR 寄存器的 DT[7:0] 字段中。该值表示 AHB 总线上两个连续事务之间允许占用的最少周期数。

在 DMA2D 运行过程中对死区值所做的更新将在下一次 AHB 传输时生效。

## 32.4. DMA2D 中断

发生如下事件时可生成中断：

- 配置错误；
- CLUT 传输完成；
- CLUT 访问错误；
- 到达传输水印；
- 传输完成；
- 传输错误。

可以使用单独的中断使能位以提高灵活性。

表 32-8 DMA2D 中断请求

| 中断事件 | 事件标志 | 使能控制位 |
| --- | --- | --- |
| 配置错误 | CEIF | CEIE |

| 中断事件 | 事件标志 | 使能控制位 |
|---|---|---|
| CLUT 传输完成 | CTCIF | CTCIE |
| CLUT 访问错误 | CAEIF | CAEIE |
| 传输水印 | TWF | TWIE |
| 传输完成 | TCIF | TCIE |
| 传输错误 | TEIF | TEIE |

## 32.5. DMA2D 寄存器

### 32.5.1. DMA2D 控制寄存器 (DMA2D_CR)

偏移地址： 0x0000

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | CEIE | CTCIE | CAEIE | TWIE | TCIE | TEIE | Reserved | LOM | Reserved | | | ABORT | SUSP | START |
| | | rw | rw | rw | rw | rw | rw | | rw | | | | rs | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:19 | 保留 | 必须保持复位值 |
| 位 18:16 | MODE | DMA2D 模式<br>此位由软件置 1 和清零。无法在传输进行时修改此位。<br>000：存储器到存储器（仅限 FG 获取）<br>001：存储器到存储器并执行 PFC（仅限 FG PFC 激活时的 FG 获取）<br>010：存储器到存储器并执行混合（执行 PFC 和混合时的 FG 和 BG 获）<br>011：寄存器到存储器（无 FG 和 BG，仅输出阶段激活）<br>100：存储器到存储器前景固定并执行混合<br>101：存储器到存储器前景固定并执行混合 |
| 位 15:14 | 保留 | 必须保持复位值 |
| 位 13 | CEIE | 配置错误中断使能<br>此位由软件置 1 和清零。<br>0：CE 中断禁止<br>1：CE 中断使能 |
| 位 12 | CTCIE | CLUT 传输完成中断使能<br>此位由软件置 1 和清零。<br>0：CTC 中断禁止<br>1：CTC 中断使能 |
| 位 11 | CAEIE | CLUT 访问错误中断使能<br>此位由软件置 1 和清零。<br>0：CAE 中断禁止<br>1：CAE 中断使能 |
| 位 10 | TWIE | 传输水印中断使能<br>此位由软件置 1 和清零。<br>0：TW 中断禁止 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 1：TW 中断使能 |
| 位 9 | TCIE | 传输完成中断使能<br>此位由软件置 1 和清零。<br>0：TC 中断禁止<br>1：TC 中断使能 |
| 位 8 | TEIE | 传输错误中断使能<br>此位由软件置 1 和清零。<br>0：TE 中断禁止<br>1：TE 中断使能 |
| 位 7 | 保留 | 必须保持复位值 |
| 位 6 | LOM | 行偏移模式<br>此位由软件置 1 和清零。<br>0：行偏移以像素表示<br>1：行偏移以字节表示 |
| 位 5:3 | 保留 | 必须保持复位值 |
| 位 2 | ABORT | 中止<br>此位可用于中止当前传输。此位可通过软件置 1 并在 START 位复位时由硬件自动复位。<br>0：不请求传输中止<br>1：请求传输中止 |
| 位 1 | SUSP | 挂起<br>此位可用于挂起当前传输。此位由软件置 1 和复位。此位在 START 位复位时由硬件自动复位。<br>0：传输不挂起<br>1：传输挂起 |
| 位 0 | START | 此位可用于根据各种配置寄存器中加载的参数启动 DMA2D。在下列情况下将自动复位此位：<br>传输结束时<br>通过用户应用程序将 DMA2D_CR 中的 ABORT 位置 1 中止数据传输时<br>数据传输出错时<br>因配置错误或已经在进行其它传输操作（CLUT 自动加载）导致数据传输未启动时。 |

### 32.5.2. DMA2D 中断状态寄存器 (DMA2D_ISR)

偏移地址： 0x0004

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | | | | CEIF | CTCIF | CAEIF | TWIF | TCIF | TEIF |
| | | | | | | | | | | r | r | r | r | r | r |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:6 | 保留 | 必须保持复位值 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 5 | CEIF | 配置错误中断标志<br>DMA2D_CR、DMA2D_FGPFCCR 或 DMA2D_BGPFCCR 的 START 位置 1 以及编程了错误的配置时，此为置 1。 |
| 位 4 | CTCIF | CLUT 传输完成中断标志<br>完成将 CLUT 从系统存储区复制到 DMA2D 内部存储器时，此位置 1。 |
| 位 3 | CAEIF | CLUT 访问错误中断标志<br>在系统存储器自动将 CLUT 复制到 DMA2D 内部存储器期间，CPU 若访问 CLUT， 此位将置 1 |
| 位 2 | TWIF | 传输水印中断标志<br>带水印行的最后一个像素完成传输时，此位置 1 |
| 位 1 | TCIF | 传输完成中断标志<br>DMA2D 传输操作完成(仅限数据传输)时，此位置 1 |
| 位 0 | TEIF | 传输错误中断标志<br>DMA 传输期间(数据传输或 CLUT 自动加载)出错时，此位置 1 |

### 32.5.3. DMA2D 中断标志清零寄存器(DMA2D_IFCR)

偏移地址： 0x0008

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | CCEIF | CCTCIF | CCAEIF | CTWIF | CTCIF | CTEIF |
| | | | | | | | | | | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:6 | 保留 | 必须保持复位值 |
| 位 5 | CCEIF | 清除配置错误中断标志<br>将此位编程为 1 可清除 DMA2D_ISR 寄存器中的 CEIF 标志。 |
| 位 4 | CCTCIF | 清除 CLUT 传输完成中断标志<br>将此为编程为 1 可清除 DMA2D_ISR 寄存器中的 CTCIF 标志。 |
| 位 3 | CAEIF | 清除 CLUT 访问错误中断标志<br>将此位编程为 1 可清除 DMA2D_ISR 寄存器中的 CAEIF 标志。 |
| 位 2 | CTWIF | 清除传输水印中断标志<br>将此位编程为 1 可清除 DMA2D_ISR 寄存器中的 TWIF 标志。 |
| 位 1 | CTCIF | 清除传输完成中断标志<br>将此位编程为 1 可清除 DMA2D_ISR 寄存器中的 TCIF 标志。 |
| 位 0 | CTEIF | 清除传输错误中断标志<br>将此位编程为 1 可清除 DMA2D_ISR 寄存器中的 TEIF 标志。 |

### 32.5.4. DMA2D 前景层存储器地址寄存器(DMA2D_FGMAR)

偏移地址： 0x000C

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | MA[31:0] | 存储器地址<br>前景层图像所用数据的地址。只有在禁止数据传输的情况下才能写入该寄存器。数据传入一旦启动，此寄存器即变为只读。<br>地址对齐必须与所选图像格式相匹配, 例如每像素 32 位格式必须为 32 位对齐，每像素 16 位格式必须为 16 位对齐，而每像素 4 位格式必须为 8 位对齐。 |

### 32.5.5. DMA2D 前景层偏移寄存器(DMA2D_FGOR)

偏移地址： 0x00010

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:16 | 保留 | 必须保持复位值 |
| 位 15:0 | LO[15:0] | 行偏移<br>用于前景层图像的行偏移。当 LOM 位为 0 时以像素表示，当 LOM 位为 1 时以字节表示。<br>当以像素表示时，只考虑 LO[13:0]，忽略 LO[15:14]。<br>此值用于生成地址，行偏移将添加到各行末尾，用于确定下一行的起始地址。只有在禁止数据传输的情况下才能写入这些位，数据传输一旦启动，这些位将变成只读。<br>如果像素格式为每个像素 4 位，则行偏移值必须为偶数。 |

### 32.5.6. DMA2D 背景层存储器地址寄存器(DMA2D_BGMAR)

偏移地址： 0x00014

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | MA[31:0] | 存储器地址<br>背景层图像所用数据的地址。只有在禁止数据传输的情况下才能写入该寄存器。数据传入一旦启动，此寄存器即变为只读。<br>地址对齐必须与所选图像格式相匹配, 例如每像素 32 位格式必须为 32 位对齐，每像素 16 位格式必须为 16 位对齐，而每像素 4 位格式必须为 8 位对齐。 |

### 32.5.7. DMA2D 背景层偏移寄存器(DMA2D_BGOR)

偏移地址： 0x00018

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:16 | 保留 | 必须保持复位值 |
| 位 15:0 | LO[15:0] | 行偏移<br>用于前景层图像的行偏移。当 LOM 位为 0 时以像素表示，当 LOM 位为 1 时以字节表示。<br>当以像素表示时，只考虑 LO[13:0]，忽略 LO[15:14]。<br>此值用于生成地址，行偏移将添加到各行末尾，用于确定下一行的起始地址。只有在禁止数据传输的情况下才能写入这些位，数据传输一旦启动，这些位将变成只读。<br>如果像素格式为每个像素 4 位，则行偏移值必须为偶数。 |

### 32.5.8. DMA2D 前景层 PFC 控制寄存器(DMA2D_FGPFCCR)

偏移地址： 0x0001C

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ALPHA[7:0] | | | | | | | | Reserved | | RBS | AI | CSS | | AM[1:0] | |
| rw | rw | rw | rw | rw | rw | rw | rw | | | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CS[7:0] | | | | | | | | Reserved | | START | CCM | CM[3:0] | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | | | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:24 | ALPHA[7:0] | Alpha 值<br>这些位定义固定 alpha 通道值，该值可替换原始的 alpha 值或与原始的 alpha 值相乘，具体取决于 AM[1:0]位选择的 alpha 模式。只有在禁止数据传输的情况下才能写入这些位，传输一旦启动，这些位将变成只读。 |
| 位 23:22 | 保留 | 必须保持复位值 |
| 位 21 | RBS | 红蓝交换<br>此位用于交换 R&B 以支持 BGR 和 ABGR 颜色模式。只有在禁止数据传输的情况下才能写入此位。数据传输一旦启动，此位将变成只读。<br>0：常规模式(RGB 或 ARGB)<br>1：交换模式(BGR 或 ABGR) |
| 位 20 | AI | Alpha 反转<br>此位用于反转 Alpha 值。只有在禁止数据传输的情况下才能写入此位。数据传输一旦启动，此位将变成只读。<br>0：常规模式<br>1：Alpha 反转模式 |
| 位 19:18 | CSS[1:0] | 色度抽样<br>这些位定义了 YCbCr 颜色模式的色度抽样模式。只有在禁止数据传输的情况下才能写入这些位。数据传输一旦启动，这些位将变成只读。<br>00：4:4:4<br>01：4:2:2<br>10：4:2:0<br>其他无意义 |
| 位 17:16 | AM[1:0] | Alpha 模式<br>这些位用于前景层图像的 alpha 通道值。只有在禁止数据传输的情况下才能写入这些位。数据传输一旦启动，这些位将变成只读。<br>00：不修改前景层 alpha 通道值<br>01：原始前景层的 alpha 通道值替换为 ALPHA[7:0]<br>10：原始前景层的 alpha 通道值替换为 ALPHA[7:0]与原始的 alpha 通道值的乘积<br>其他无意义 |
| 位 15:8 | CS[7:0] | CLUT 大小<br>这些位定义用于前景层图像所用的 CLUT 大小。<br>CLUT 传输一旦启动，此字段将变成只读。<br>CLUT 条目数等于 CS[7:0]+1。 |
| 位 7:6 | 保留 | 必须保持复位值 |
| 位 5 | START | 启动<br>可将此位置 1 以启动 CLUT 的自动加载过程。该位在以下情况下自动复位：<br>传输结束时<br>通过用户应用程序将 DMA2D_CR 中的 ABORT 位置 1 中止传输时<br>传输出错时 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 因配置错误或已经在进行其它传输操作(数据传输或自动背景 CLUT 传输)导致传输未启动时 |
| 位 4 | CCM | CLUT 颜色模式<br>这些位定义 CLUT 的颜色格式。只有在禁止数据传输的情况下才能写入这些位。数据传输一旦启动，这些位将变成只读。<br>0：ARGB8888<br>1：RGB888 |
| 位 3:0 | CM[3:0] | 颜色模式<br>这些位定义前景层图像的颜色模式。只有在禁止数据传输的情况下才能写入这些位。数据传输一旦启动，这些位将变成只读。<br>0000: ARGB8888<br>0001: RGB888<br>0010: RGB565<br>0011: ARGB1555<br>0100: ARGB4444<br>0101: L8<br>0110: AL44<br>0111: AL88<br>1000: L4<br>1001: A8<br>1010: A4<br>1011: YCbCr<br>其它无意义 |

### 32.5.9. DMA2D 前景层颜色寄存器(DMA2D_FGCOLR)

偏移地址： 0x00020

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:24 | 保留 | 必须保持复位值 |
| 位 23:16 | RED[7:0] | 红色值<br>这些位定义前景层图像 A4 或 A8 模式的红色值。只有在禁止数据传输的情况下才能写入这些位。数据传输一旦启动，这些位将变成只读。 |
| 位 15:8 | GREEN[7:0] | 绿色值<br>这些位定义前景层图像 A4 或 A8 模式的绿色值。只有在禁止数据传输的情况下才能写入这些位。数据传输一旦启动，这些位将变成只读。 |
| 位 7:0 | BLUE[7:0] | 蓝色值<br>这些位定义前景层图像 A4 或 A8 模式的蓝色值。只有在禁止数据传输的情况下才能写入这些位。数据传输一旦启动，这些位将变成只读。 |

### 32.5.10. DMA2D 背景层 PFC 控制寄存器(DMA2D_BGPFCCR)

偏移地址： 0x00024

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ALPHA[7:0] | | | | | | | | Reserved | | RBS | AI | CSS | | AM[1:0] | |
| rw | rw | rw | rw | rw | rw | rw | rw | | | rw | rw | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CS[7:0] | | | | | | | | Reserved | | START | CCM | CM[3:0] | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | | | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:24 | ALPHA[7:0] | Alpha 值<br>这些位定义固定 alpha 通道值, 该值可替换原始的 alpha 值或与原始的 alpha 值相乘，具体取决于 AM[1:0]位选择的 alpha 模式。只有在禁止数据传输的情况下才能写入这些位，传输一旦启动，这些位将变成只读。 |
| 位 23:22 | 保留 | 必须保持复位值 |
| 位 21 | RBS | 红蓝交换<br>此位用于交换 R&B 以支持 BGR 和 ABGR 颜色模式。只有在禁止数据传输的情况下才能写入此位。数据传输一旦启动，此位将变成只读。<br>0：常规模式(RGB 或 ARGB)<br>1：交换模式(BGR 或 ABGR) |
| 位 20 | AI | Alpha 反转<br>此位用于反转 Alpha 值。只有在禁止数据传输的情况下才能写入此位。数据传输一旦启动，此位将变成只读。<br>0：常规模式<br>1：Alpha 反转模式 |
| 位 19:18 | CSS[1:0] | 色度抽样<br>这些位定义了 YCbCr 颜色模式的色度抽样模式。只有在禁止数据传输的情况下才能写入这些位。数据传输一旦启动，这些位将变成只读。<br>00：4:4:4<br>01：4:2:2<br>10：4:2:0<br>其他无意义 |
| 位 17:16 | AM[1:0] | Alpha 模式<br>这些位用于背景层图像的 alpha 通道值。只有在禁止数据传输的情况下才能写入这些位。数据传输一旦启动，这些位将变成只读。<br>00：不修改背景层 alpha 通道值<br>01：原始背景层的 alpha 通道值替换为 ALPHA[7:0]<br>10：原始背景层的 alpha 通道值替换为 ALPHA[7:0]与原始的 alpha 通道值的乘积<br>其他无意义 |
| 位 15:8 | CS[7:0] | CLUT 大小<br>这些位定义用于背景层图像所用的 CLUT 大小。<br>CLUT 传输一旦启动，此字段将变成只读。<br>CLUT 条目数等于 CS[7:0]+1。 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 7:6 | 保留 | 必须保持复位值 |
| 位 5 | START | 启动<br>可将此位置 1 以启动 CLUT 的自动加载过程。该位在以下情况下自动复位：<br>传输结束时<br>通过用户应用程序将 DMA2D_CR 中的 ABORT 位置 1 中止传输时<br>传输出错时<br>因配置错误或已经在进行其它传输操作(数据传输或自动背景 CLUT 传输)导致传输未启动时 |
| 位 4 | CCM | CLUT 颜色模式<br>这些位定义 CLUT 的颜色格式。只有在禁止数据传输的情况下才能写入这些位。数据传输一旦启动，这些位将变成只读。<br>0：ARGB8888<br>1：RGB888 |
| 位 3:0 | CM[3:0] | 颜色模式<br>这些位定义背景层图像的颜色模式。只有在禁止数据传输的情况下才能写入这些位。数据传输一旦启动，这些位将变成只读。<br>0000: ARGB8888<br>0001: RGB888<br>0010: RGB565<br>0011: ARGB1555<br>0100: ARGB4444<br>0101: L8<br>0110: AL44<br>0111: AL88<br>1000: L4<br>1001: A8<br>1010: A4<br>1011: YCbCr(只适用前景)<br>其它无意义 |

### 32.5.11. DMA2D 背景层颜色寄存器(DMA2D_BGCOLR)

偏移地址： 0x00028

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:24 | 保留 | 必须保持复位值 |
| 位 23:16 | RED[7:0] | 红色值<br>这些位定义背景层图像 A4 或 A8 模式的红色值。只有在禁止数据传输的情况下才能写入这些位。数据传输一旦启动，这些位将变成只读。 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 15:8 | GREEN[7:0] | 绿色值<br>这些位定义背景层图像 A4 或 A8 模式的绿色值。只有在禁止数据传输的情况下才能写入这些位。数据传输一旦启动，这些位将变成只读。 |
| 位 7:0 | BLUE[7:0] | 蓝色值<br>这些位定义背景层图像 A4 或 A8 模式的蓝色值。只有在禁止数据传输的情况下才能写入这些位。数据传输一旦启动，这些位将变成只读。 |

### 32.5.12. DMA2D 前景层 CLUT 存储器地址寄存器(DMA2D_FGCMAR)

偏移地址： 0x0002C

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:0 | MA[31:0] | 存储器地址<br>专用于前景层图像的 CLUT 地址所使用的数据地址。只有在禁止数据传输的情况下才能写入这些位。CLUT 传输一旦启动，这些位将变成只读。<br>如果前景层 CLUT 格式时 32 位，则地址必须 32 位对齐。 |

### 32.5.13. DMA2D 背景层 CLUT 存储器地址寄存器(DMA2D_BGCMAR)

偏移地址： 0x00030

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 位 31:0 | MA[31:0] | 存储器地址<br>专用于背景层图像的 CLUT 地址所使用的数据地址。只有在禁止数据传输的情况下才能写入这些位。CLUT 传输一旦启动，这些位将变成只读。<br>如果背景层 CLUT 格式时 32 位，则地址必须 32 位对齐。 |

### 32.5.14. DMA2D 输出 PFC 控制寄存器(DMA2D_OPFCCR)

偏移地址： 0x00034

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:22 | 保留 | 必须保持复位值 |
| 位 21 | RBS | 红蓝交换<br>此位用于交换 R&B 以支持 BGR 和 ABGR 颜色模式。只有在禁止数据传输的情况下才能写入此位。数据传输一旦启动，此位将变成只读。<br>0：常规模式(RGB 或 ARGB)<br>1：交换模式(BGR 或 ABGR) |
| 位 20 | AI | Alpha 反转<br>此位用于反转 Alpha 值。只有在禁止数据传输的情况下才能写入此位。数据传输一旦启动，此位将变成只读。<br>0：常规模式<br>1：Alpha 反转模式 |
| 位 19:11 | 保留 | 必须保持复位值 |
| 位 10:9 | 保留 | 图像旋转<br>这些位定义图像旋转。只有在禁止数据传输的情况下才能写入这些位。数据传输一旦启动，这些位将变成只读。<br>00: 旋转 0°<br>01: 旋转 90°<br>10: 旋转 180°<br>11: 旋转 270° |
| 位 8 | SB[1:0] | 交换字节<br>这些位定义用于 FIFO 中的字节逐个交换。只有在禁止数据传输的情况下才能写入这些位。数据传输一旦启动，这些位将变成只读。<br>当此位被设置后，每行像素(PL)必须为偶数，并且输出寄存器地址(OMAR)必须为偶数。<br>0：输出 FIFO 中按规则顺序排列的字节<br>1：字节在输出 FIFO 中逐个交换 |
| 位 7:3 | 保留 | 必须保持复位值 |
| 位 2:0 | CM[2:0] | 颜色模式<br>这些位定义输出图像的颜色模式。只有在禁止数据传输的情况下才能写入这些位。数据传输一旦启动，这些位将变成只读。<br>0000: ARGB8888<br>0001: RGB888<br>0010: RGB565<br>0011: ARGB1555<br>0100: ARGB4444<br>其它无意义 |

### 32.5.15. DMA2D 输出颜色寄存器(DMA2D_OCOLR)

偏移地址： 0x00038

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:24 | ALPHA[7:0] | Alpha 通道值<br>这些位定义输出颜色的 alpha 通道。只有在禁止数据传输的情况下才能写入此位。数据传输一旦启动，此位将变成只读。 |
| 位 23:16 | RED[7:0] | 红色值<br>这些位定义输出图像的红色值。只有在禁止数据传输的情况下才能写入此位。数据传输一旦启动，此位将变成只读。 |
| 位 15:8 | GREEN[7:0] | 绿色值<br>这些位定义输出图像的绿色值。只有在禁止数据传输的情况下才能写入此位。数据传输一旦启动，此位将变成只读。 |
| 位 7:0 | BLUE[7:0] | 蓝色值<br>这些位定义输出图像的蓝色值。只有在禁止数据传输的情况下才能写入此位。数据传输一旦启动，此位将变成只读。 |

### 32.5.16. DMA2D 输出存储器地址寄存器(DMA2D_OMAR)

偏移地址： 0x0003C

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:0 | MA[31:0] | 存储器地址<br>输出 FIFO 所用的数据的地址。有在禁止数据传输的情况下才能写入此位。数据传输一旦启动，此位将变成只读。<br>地址对齐必须与所选图像的格式相匹配，例如每像素 32 位必须为 32 位对齐，每像素 16 位格式必须为 16 位对齐。 |

### 32.5.17. DMA2D 输出偏移寄存器(DMA2D_OOR)

偏移地址： 0x00040

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:16 | 保留 | 必须保持复位值 |
| 位 15:0 | LO[15:0] | 行偏移<br>用于前景层图像的行偏移。当 LOM 位为 0 时以像素表示，当 LOM 位为 1 时以字节表示。<br>当以像素表示时，只考虑 LO[13:0]，忽略 LO[15:14]。<br>此值用于生成地址，行偏移将添加到各行末尾，用于确定下一行的起始地址。只有在禁止数据传输的情况下才能写入这些位，数据传输一旦启动，这些位将变成只读。<br>如果像素格式为每个像素 4 位，则行偏移值必须为偶数。 |

### 32.5.18. DMA2D 行数寄存器(DMA2D_NLR)

偏移地址： 0x00044

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:30 | 保留 | 必须保持复位值 |
| 位 29:16 | PL[13:0] | 每行像素数<br>待传输区域的每行像素数。只有在禁止数据传输的情况下才能写入这些位，数据传输一旦启动，这些位将变成只读。<br>如果任何一个输入图像格式为每像素 4 位，则每行像素数必须为偶数。 |
| 位 15:0 | NL[15:0] | 行数<br>待传输区域的行数。只有在禁止数据传输的情况下才能写入这些位，数据传输一旦启动，这些位将变成只读。 |

### 32.5.19. DMA2D 行水印寄存器(DMA2D_LWR)

偏移地址： 0x00048

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:16 | 保留 | 必须保持复位值 |
| 位 15:0 | LW[13:0] | 行水印<br>这些位可用于配置可产生中断的行水印。<br>在带水印的最后一行像素传输完成时产生中断。只有在禁止数据传输的情况下才能写入这些位，数据传输一旦启动，这些位将变成只读。 |

### 32.5.20. DMA2D AHB 主设备定时器配置寄存器(DMA2D_AMTCR)

偏移地址： 0x00048

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:16 | 保留 | 必须保持复位值 |
| 位 15:8 | DT[7:0] | 死区<br>在 AHB 主设备端口上两个连续访问之间所插入的死区值，以 AHB 时钟周期数表示。这些位表示两个连续 AHB 访问之间允许所占用的最少周期数。 |
| 位 7:1 | 保留 | 必须保持复位值 |
| 位 0 | EN | 使能<br>使能死区功能。 |

# 33. JPEG 解码控制器(JPEGD)

## 33.1. JPEGD 简介

JPEGD 控制器是遵循 ISO/IEC 10918-1 标准的 JPEG 解码器。

## 33.2. JPEGD 的功能描述

JPEGD 控制器根据输入的 JPEG 数据流实时解码后输出，控制器会自动解析 JPEG 数据流中的参数,无需用户配置。

## 33.3. JPEGD 寄存器

### 33.3.1. JPEGD 配置寄存器 0(JPEGD_CONFR0)

偏移地址： 0x00

复位值：0x00000000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | LP | SWR | CLR | Reserved | START |
| | | | | | | | | | | | rw | rw | rw | | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:5 | 保留 | 必须保持复位值 |
| 位 4 | LP | JPEGD 控制器的低功耗控制<br>0：退出低功耗模式<br>1：进入低功耗模式 |
| 位 3 | SWR | JPEGD 控制器的软件复位<br>0：释放<br>1：复位 |
| 位 2 | CLR | JPEGD 控制器的同步复位<br>0：释放<br>1：复位 |
| 位 1 | 保留 | 必须保持复位值 |
| 位 0 | START | JPEGD 控制器的启动控制<br>0：关闭<br>1：启动 |

### 33.3.2. JPEGD 配置寄存器 1(JPEGD_CONFR1)

偏移地址： 0x04

复位值：0x00000000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | NS | | COLSPACE | | DE | Reserved | NF | |
| | | | | | | | | ro | ro | ro | ro | rw | | ro | ro |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:16 | YSIZE | 原始图像的行数 |
| 位 15:8 | 保留 | 必须保持复位值 |
| 位 7:6 | NS | 扫描的分量表个数 |
| 位 5:4 | COLSPACE | 颜色空间<br>00：灰度(1 个量化表)<br>01：YUV(2 个量化表)<br>10：RGB(3 个量化表)<br>11：CMYK(4 个量化表) |
| 位 3 | DE | 解码使能<br>0：解码未使能<br>1：解码使能 |
| 位 2 | 保留 | 必须保持复位值 |
| 位 1:0 | NF | 颜色分量的个数<br>00：灰度(1 个)<br>01：保留(2 个)<br>10：YUV 或 RGB(3 个)<br>11：CMYK(4 个) |

### 33.3.3. JPEGD 配置寄存器 2(JPEGD_CONFR2)

偏移地址： 0x08

复位值：0x00000000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | NMCU[25:16] | | | | | | | | | |
| | | | | | | ro | ro | ro | ro | ro | ro | ro | ro | ro | ro |

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:26 | 保留 | 必须保持复位值 |
| 位 25:0 | NMCU | 当前扫描中每个 MCU 的块数 |

### 33.3.4. JPEGD 配置寄存器 3(JPEGD_CONFR3)

偏移地址： 0x0C

复位值：0x00000000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:16 | XSIZE | 每行的像素数 |
| 位 15:0 | 保留 | 必须保持复位值 |

### 33.3.5. JPEGD 配置寄存器 4-7(JPEGD_CONFR4-7)

偏移地址： 0x10 + 0x04*i, i 为图像分量, i=0...3

复位值：0x00000000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| HSF | | | | VSF | | | | NB | | | | QT | | Reserved | |
| ro | ro | ro | ro | ro | ro | ro | ro | ro | ro | ro | ro | ro | ro | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:16 | 保留 | 必须保持复位值 |
| 位 15:12 | HSF | 分量 i 的水平采样因子 |
| 位 11:8 | VSF | 分量 i 的垂直采样因子 |
| 位 7:4 | NB | 分量 i 的 MCU 块数: |
| 位 3:2 | QT | 分量 i 的量化表 id |
| 位 1:0 | 保留 | 必须保持复位值 |

### 33.3.6. JPEGD 输入 FIFO 阈值寄存器 (JPEGD_INTH)

偏移地址： 0x20

复位值：0x00000000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:8 | 保留 | 必须保持复位值 |
| 位 7:0 | TH | 输入 FIFO 阈值，1~31 |

### 33.3.7. JPEGD 输出 FIFO 阈值寄存器 (JPEGD_OUTTH)

偏移地址： 0x24

复位值：0x00000000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:8 | 保留 | 必须保持复位值 |
| 位 7:0 | TH | 输出 FIFO 阈值，1~31 |

### 33.3.8. JPEGD 控制寄存器 (JPEGD_CR)

偏移地址： 0x30

复位值：0x00000000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | OFF | IFF | Reserved | | | | | | HPDIE | EOCIE | OFNEIE | OFTIE | IFNFIE | IFTIE | JCEN |
| | rw | rw | | | | | | | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:15 | 保留 | 必须保持复位值 |
| 位 14 | OFF | 刷新输出 FIFO<br>0：无影响<br>1：刷新输出 FIFO |
| 位 13 | IFF | 刷新输入 FIFO<br>0：无影响<br>1：刷新输入 FIFO |
| 位 12:7 | 保留 | 必须保持复位值 |
| 位 6 | HPDIE | 数据头解析完成中断使能<br>0：不使能<br>1：使能 |
| 位 5 | EOCIE | 解码完成中断使能<br>0：不使能<br>1：使能 |
| 位 4 | OFNEIE | 输出 FIFO 非空中断使能<br>0：不使能<br>1：使能 |
| 位 3 | OFTIE | 输出 FIFO 的有效数据个数大于等于输出阈值时，中断使能 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 0：不使能<br>1：使能 |
| 位 2 | IFNFIE | 输入 FIFO 非空中断使能<br>0：不使能<br>1：使能 |
| 位 1 | IFTIE | 输入 FIFO 的有效数据个数小于输入阈值时，中断使能<br>0：不使能<br>1：使能 |
| 位 0 | JCEN | JPEGD 模块使能<br>0：不使能<br>1：使能 |

### 33.3.9. JPEGD 控制寄存器 (JPEGD_SR)

偏移地址： 0x34

复位值：0x00000000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | | COF | HPDF | EOCF | OFNEF | OFTF | IFNFF | IFTF | CFGE |
| | | | | | | | | ro | ro | ro | ro | ro | ro | ro | ro |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:8 | 保留 | 必须保持复位值 |
| 位 7 | COF | 解码模块工作状态<br>0：未工作<br>1：正在工作 |
| 位 6 | HPDF | 数据头解析状态<br>0：未完成<br>1：完成 |
| 位 5 | EOCF | 解码完成以及数据全部传输到输出 FIFO 中<br>0：未完成<br>1：完成 |
| 位 4 | OFNEF | 输出 FIFO 数据有效状态，此位不能在 DMA 模式中使用<br>0：输出 FIFO 空<br>1：输出 FIFO 非空 |
| 位 3 | OFTF | 输出 FIFO 的有效数据与输出 FIFO 阈值的关系，此位不能在 DMA 模式中使用<br>0：小于阈值<br>1：大于等于阈值 |
| 位 2 | IFNFF | 输入 FIFO 未满状态，此位不能在 DMA 模式中使用<br>0：输入 FIFO 满<br>1：输入 FIFO 未满 |
| 位 1 | IFTF | 输入 FIFO 的有效数据与输入 FIFO 阈值的关系，此位不能在 DMA 模式中使用 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 0：大于等于阈值<br>1：小于阈值 |
| 位 0 | CFGE | 解码模块错误状态（此状态无法清除,需要复位模块）:<br>0：未发生错误<br>1：发送错误 |

### 33.3.10. JPEGD 数据状态清除寄存器 (JPEGD_CFR)

偏移地址： 0x38

复位值：0x00000000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:7 | 保留 | 必须保持复位值 |
| 位 6 | CHPDF | 写 1 清除 JPEGD_SR 中的 HPDF 位<br>0：无影响<br>1：清除 |
| 位 5 | CEOCF | 写 1 清除 JPEGD_SR 中的 EOCF 位<br>0：无影响<br>1：清除 |
| 位 4:0 | 保留 | 必须保持复位值 |

### 33.3.11. JPEGD 数据输入寄存器 (JPEGD_DIR)

偏移地址： 0x40

复位值：0x00000000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:8 | 保留 | 必须保持复位值 |
| 位 7:0 | DIR | 数据输入 |

### 33.3.12. JPEGD 数据输出寄存器 (JPEGD_DOR)

偏移地址： 0x44

复位值：0x00000000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:8 | 保留 | 必须保持复位值 |
| 位 7:0 | DOR | 数据输出 |

# 34. LCD-TFT 控制器(LTDC)

## 34.1. LTDC 简介

LCD-TFT 显示器控制器提供并行数字 RGB（红色、绿色、蓝色）以及水平同步、垂直同步、像素时钟和数据使能信号，这些信号直接输出到不同 LCD 和 TFT 面板的接口。

## 34.2. LTDC 主要特性

- 24 位 RGB 并行像素输出；每像素 8 位 (RGB888);
- 2 个带有专用 FIFO 的显示层（FIFO 深度 64x32 位）;
- 查色表 (CLUT)，每层高达 256 种颜色（256x24 位）;
- 支持高达 WXGA (1280 x 800) 的分辨率;
- 可针对不同显示面板编程时序;
- 可编程背景色;
- 可编程 HSync、VSync 和数据使能信号的极性;
- 每层有多达 8 个输入颜色格式可供选择:
  - ARGB8888
  - RGB888
  - RGB565
  - ARGB1555
  - ARGB4444
  - L8（8 位 Luminance 或 CLUT）
  - AL44（4 位 alpha + 4 位 luminance）
  - AL88（8 位 alpha + 8 位 luminance）
- 每通道的低位采用伪随机抖动输出:
  - 红色、绿色、蓝色的抖动宽度为 2 位;
- 使用 alpha 值（每像素或常数）在两层之间灵活混合;
- 色键（透明颜色）;
- 可编程窗口位置和大小;
- 支持薄膜晶体管 (TFT) 彩色显示器;
- AHB 主接口支持 16 个字的突发;
- 高达 4 个可编程中断事件。

## 34.3. LTDC 功能说明

### 34.3.1. LTDC 框图

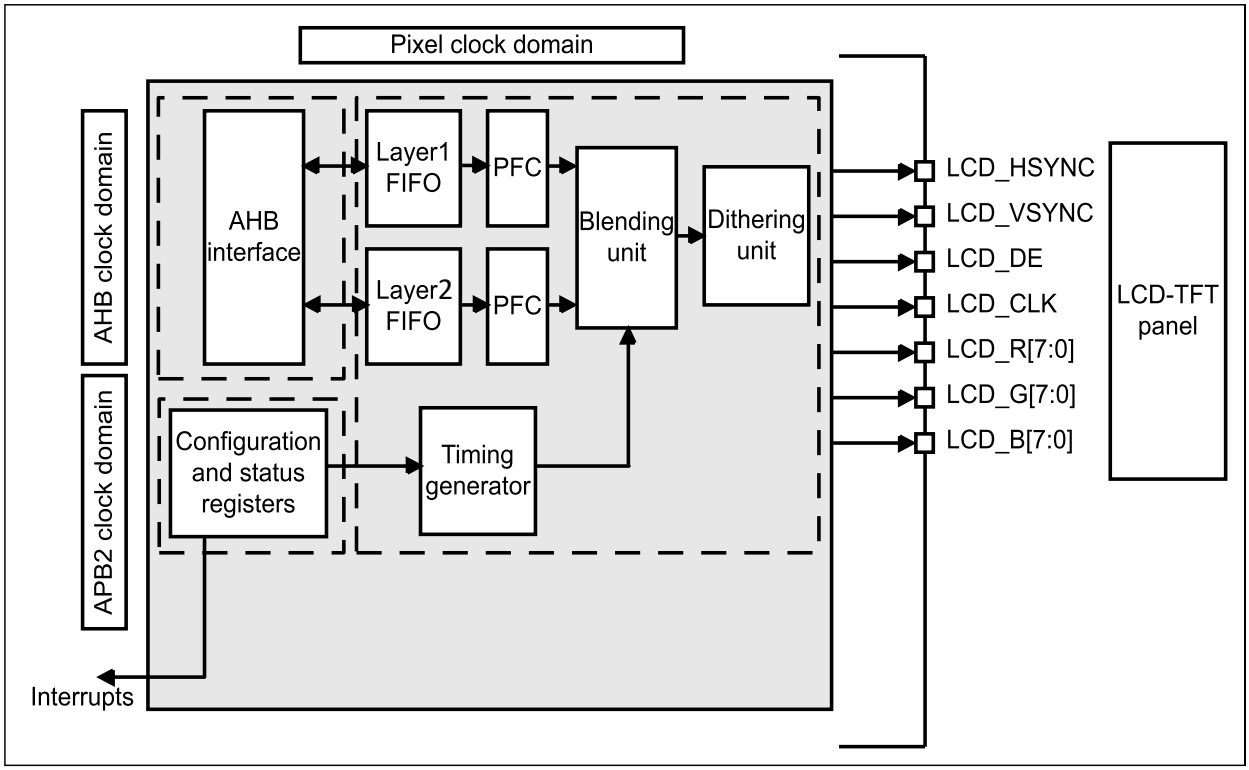



图 34-1 LTDC 框图

层 FIFO：每层一个 64x32 位 FIFO。

PFC：执行像素格式转换的像素格式转换器，从层的所选输入像素格式转换为字。

AHB 接口：用于将数据从存储器传输到 FIFO。

### 34.3.2. LTDC 复位和时钟

LCD-TFT 控制器外设使用 3 个时钟域：

- AHB 时钟域 (HCLK)：用于将数据从存储器传输到 FIFO 层；
- APB2 时钟域 (PCLK2)：用于配置寄存器；
- 像素时钟域 (LCD_CLK)：用于生成 LCD-TFT 接口信号。LCD_CLK 输出应按照面板要求配置。

通过将 RCC_AHB4RSTR 寄存器中的相应位置 1 可将 LCD 控制器复位。这将复位三个时钟域。

### 34.3.3. LTDC 引脚和信号接口

表 34-1 LTDC 引脚和信号接口

| LCD-TFT 信号 | I/O | 说明 |
|---|---|---|
| LCD_CLK | O | 时钟输出 |
| LCD_HSYNC | O | 水平同步 |

| LCD-TFT 信号 | I/O | 说明 |
|---|---|---|
| LCD_VSYNC | O | 垂直同步 |
| LCD_DE | O | 数据使能 |
| LCD_R[7:0] | O | 数据： 8 位红色数据 |
| LCD_G[7:0] | O | 数据： 8 位绿色数据 |
| LCD_B[7:0] | O | 数据： 8 位蓝色数据 |

必须通过用户程序配置 LCD-TFT 控制器引脚。未使用的引脚可用于其他功能。

对于高达 24 位 (RGB888) 的 LTDC 输出，如果使用低于 8bpp 的像素深度将 RGB565 或 RGB666 输出到 16 位或 18 位显示器，则 RGB 显示数据线必须连接到 LCD-TFT 控制器 RGB 数据线的 MSB。例如，当 LCD-TFT 控制器与 RGB565 16 位显示器相连时，LCD 显示器的 R[4:0]、G[5:0] 和 B[4:0] 数据线引脚必须连接至 LCD-TFT 控制器的 LCD_R[7:3]、LCD_G[7:2] 和 LCD_B[7:3]。

## 34.4. LTDC 可编程参数

LCD-TFT 控制器提供灵活的可配置参数。其可通过 LTDC_GCR 寄存器使能或禁止。

### 34.4.1. LTDC 全局配置参数

**同步时序：**

下图显示了同步时序发生器模块生成的可配置时序参数。该模块生成水平和垂直同步时序面板信号、像素时钟和数据使能信号。

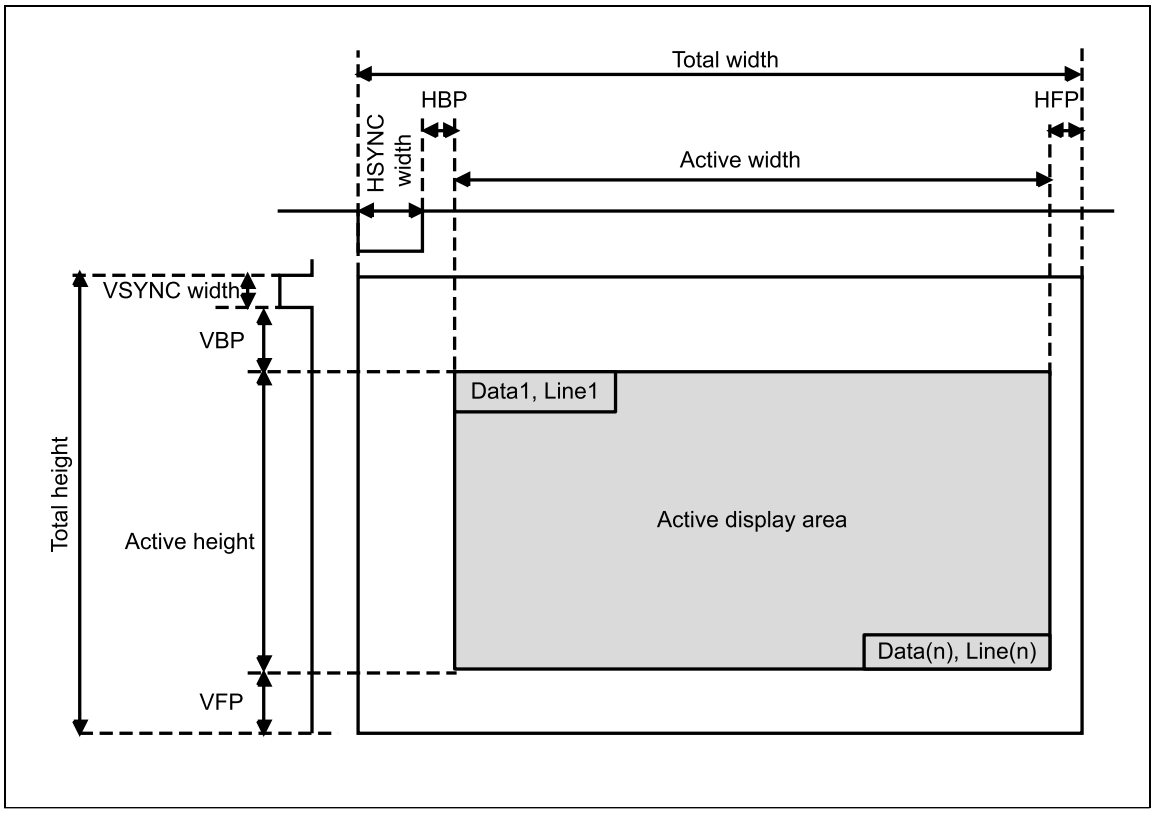



图 34-2 LTDC 同步时序

*注：HBP 和 HFP 分别为水平后沿周期和水平前沿周期。*

VBP 和 VFP 分别为垂直后沿周期和垂直前沿周期。

LCD-TFT 可编程同步时序包括：

- HSYNC 和 VSYNC 宽度：水平和垂直同步宽度，通过编程 LTDC_SSCR 寄存器中的 HSYNC Width - 1 和 VSYNC Width - 1 的值进行配置。
- HBP 和 VBP：水平和垂直同步后沿宽度，通过编程 LTDC_BPCR 寄存器中的累加值 HSYNC Width + HBP - 1 和累加值 VSYNC Width + VBP - 1 进行配置。
- 有效宽度和有效高度：有效宽度和有效高度通过编程 LTDC_AWCR 寄存器中的累加值 HSYNC Width + HBP + Active Width - 1 和累加值 VSYNC Width + VBP + Active Height - 1 进行配置（支持最高 1280 x 800）。
- 总宽度：总宽度通过编程 LTDC_TWCR 寄存器中的累加值 HSYNC Width + HBP + Active Width + HFP - 1 进行配置。HFP 为水平前沿周期。
- 总高度：总高度通过编程 LTDC_TWCR 寄存器中的累加值 VSYNC Height + VBP + Active Height + VFP - 1 进行配置。VFP 为垂直前沿周期。

*注：使能 LTDC 时，产生的时序以 X/Y=0/0 位置作为垂直同步区域中的第一个水平同步像素，随后是后沿、有效数据显示区域和前沿。*

*禁止 LTDC 时，时序发生器模块复位为 X= 总宽度- 1、Y= 总高度- 1，并在垂直同步阶段和 FIFO 刷新前保持上一个像素。因此，仅连续输出消隐数据。*

**同步时序配置示例：**

TFT-LCD 时序（应从面板数据手册中提取）：

- 水平和垂直同步宽度：0x8 像素，0x4 行；
- 水平和垂直后沿：0x7 像素， 0x2 行；
- 有效宽度和有效高度：0x280 像素，0x1E0 行 (640x480)；
- 水平前沿：0x6 像素；
- 垂直前沿：0x2 行。

LTDC 时序寄存器中编程的值将为：

- LTDC_SSCR 寄存器：将编程为 0x00070001。(HSW[11:0] 为 0x7 且 VSH[10:0] 为 0x3)；
- LTDC_BPCR 寄存器：将编程为 0x000E0005。(AHBP[11:0] 为 0xE(0x8 + 0x6) 且 AVBP[10:0] 为 0x5(0x4 + 0x1))；
- LTDC_AWCR 寄存器：将编程为 0x028E01E5。(AAW[11:0] 为 0x28E(0x8 + 0x7 + 0x27F) 且 AAH[10:0] 为 0x1E5(0x4 + 0x2 + 0x1DF))；
- LTDC_TWCR 寄存器：将编程为 0x00000294。(TOTALW[11:0] 为 0x294(0x8 + 0x7 + 0x280 + 0x5))；
- LTDC_THCR 寄存器：将编程为 0x000001E7。(TOTALH[10:0] 为 0x1E7(0x4 + 0x2 + 0x1E0 + 1))。

**可编程极性**

水平和垂直同步、数据使能和像素时钟输出信号的极性可通过 LTDC_GCR 寄存器编程为高电平有效或低电平有效。

**背景色**

恒定的背景色 (RGB888) 可通过 LTDC_BCCR 寄存器编程。它用于与底层混合。

**抖动**

使用 LFSR 的伪随机抖动技术用于向各个像素颜色通道值（R、G 或 B）添加小的随机值（阈值），从而当 18 位显示器上显示 24 位数据时，可在某些情况下对 MSB 进行舍入操作。因此，抖动技术用于对各帧中不同的数据进行舍入操作。

伪随机抖动技术的过程是将 LSB 与阈值比较，并在 LSB 部分 >= 阈值时，仅向 MSB 部分加 1。一旦应用抖动技术，通常会减少 LSB。

添加的伪随机值的宽度为每个颜色通道 2 位；红色 2 位、绿色 2 位及蓝色 2 位。

使能 LCD-TFT 控制器后，LFSR 以第一个有效像素开始运行，并且即使在消隐周期内和抖动关闭时也保持运行状态。如果禁止 LTDC，LFSR 将复位。

可通过 LTDC_GCR 寄存器实时开启和关闭抖动。

**重载影子寄存器**

一些配置寄存器执行影子操作。对活动寄存器执行写操作时，或在 **LTDC_SRCR** 寄存器配置阶段之后的垂直消隐周期开始时，可将影子寄存器值立即重载到活动寄存器中。如果选择了立即重载配置，则只应在所有新寄存器完成写操作后激活重载。

不应在重载完成前再次修改影子寄存器。读取影子寄存器将返回实际有效值。新写入的值只能在重载发生后读取。

如果在 LTDC_IER 寄存器中相应使能，则可产生寄存器重载中断。

影子寄存器均为第 1 层和第 2 层寄存器，但 LTDC_LxCLUTWR 寄存器除外。

**中断产生事件**

有关中断配置，请参见 LTDC 中断。

### 34.4.2. 层可编程参数

最多可单独使能、禁止和配置两个层。层显示顺序固定，即自下而上。如果使能两个层，则层 2 为顶部显示窗口。

**窗口**

可为每个层定位和调整大小，各个层必须位于有效显示区域内。

窗口位置和大小通过左上和右下的 X/Y 位置以及包含同步、后沿大小和有效数据区域的内部时序发生器配置。请参见 LTDC_LxWHPCR 和 LTDC_WVPCR 寄存器。

可编程层位置和大小定义了一行中的第一个/最后一个可见像素和窗口中的第一个/最后一个可见行。它允许显示完整的图像帧，也允许只显示图像帧的一部分。请参见下图。

- 层中的第一个和最后一个可见像素通过配置 LTDC_LxWHPCR 寄存器的 WHSTPOS[11:0] 和 WHSPPOS[11:0] 进行设置。
- 层中的第一个和最后一个可见行通过配置 LTDC_LxWVPCR 寄存器的 WVSTPOS[10:0] 和 WVSPPOS[10:0] 进行设置。

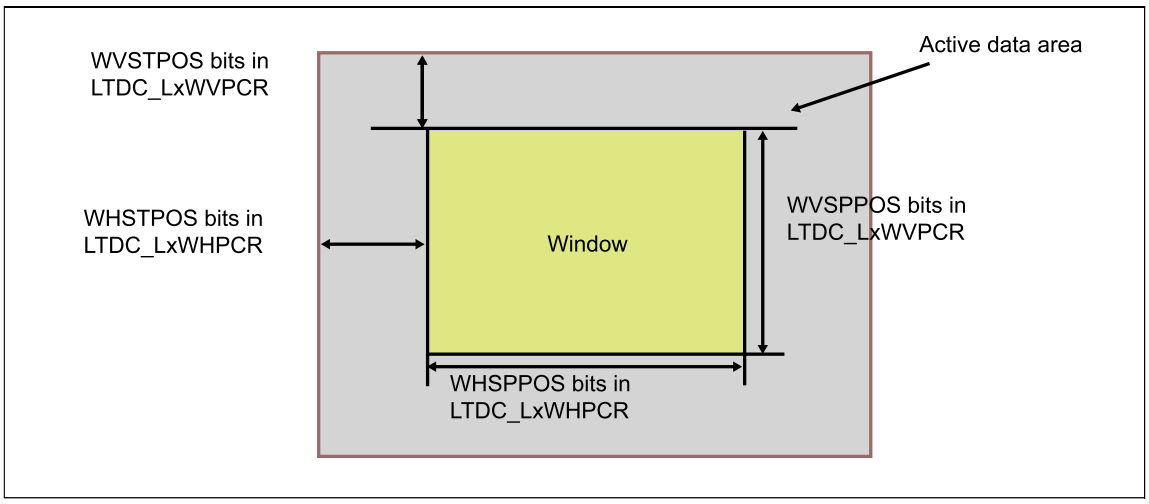



图 34-3 层窗口可编程参数

### 像素输入格式

可编程像素格式用于层的帧缓冲区中存储的数据。

可通过 LTDC_LxPFCR 寄存器为每个层配置多达 8 个输入像素格式。

像素数据从帧缓冲区中读取，随后按照以下方式转换为内部 8888 (ARGB) 格式：

宽度低于 8 位的分量通过位重复扩展到 8 位。所选位范围多次拼接，直至其超过 8 位。在得到的向量中，选择高 8 位。例如：5 位 RGB565 红色通道将变为（位位置）：43210432（低 3 位由 5 位中的高 3 位填充）

下图说明了像素数据映射与所选格式的关系。

表 34-2 像素数据映射与颜色格式的关系

| ARGB8888 | | | |
|---|---|---|---|
| @+3 | @+2 | @+1 | @ |
| $A_x$[7:0] | $R_x$[7:0] | $G_x$[7:0] | $B_x$[7:0] |
| @+7 | @+6 | @+5 | @+4 |
| $A_{x+1}$[7:0] | $R_{x+1}$[7:0] | $G_{x+1}$[7:0] | $B_{x+1}$[7:0] |
| RGB888 | | | |
| @+3 | @+2 | @+1 | @ |
| $B_{x+1}$[7:0] | $R_x$[7:0] | $G_x$[7:0] | $B_x$[7:0] |
| @+7 | @+6 | @+5 | @+4 |
| $G_{x+2}$[7:0] | $B_{x+2}$[7:0] | $R_{x+1}$[7:0] | $G_{x+1}$[7:0] |
| RGB565 | | | |
| @+3 | @+2 | @+1 | @ |
| $R_{x+1}$[4:0] $G_{x+1}$[5:3] | $G_{x+1}$[2:0] $B_{x+1}$[4:0] | $R_x$[4:0] $G_x$[5:3] | $G_x$[2:0] $B_x$[4:0] |
| @+7 | @+6 | @+5 | @+4 |
| $R_{x+3}$[4:0] $G_{x+3}$[5:3] | $G_{x+3}$[2:0] $B_{x+3}$[4:0] | $R_{x+2}$[4:0] $G_{x+2}$[5:3] | $G_{x+2}$[2:0] $B_{x+2}$[4:0] |
| ARGB1555 | | | |
| @+3 | @+2 | @+1 | @ |
| $A_{x+1}$[0]$R_{x+1}$[4:0] | $G_{x+1}$[2:0] $B_{x+1}$[4:0] | $A_x$[0] $R_x$[4:0] $G_x$[4:3] | $G_x$[2:0] $B_x$[4:0] |
| $G_{x+1}$[4:3] | | | |
| @+7 | @+6 | @+5 | @+4 |
| $A_{x+3}$[0]$R_{x+3}$[4:0] | $G_{x+3}$[2:0] $B_{x+3}$[4:0] | $A_{x+2}$[0]$R_{x+2}$[4:0]$G_{x+2}$[4: | $G_{x+2}$[2:0] $B_{x+2}$[4:0] |
| $G_{x+3}$[4:3] | | 3] | |

| ARGB8888 | | | |
|---|---|---|---|
| ARGB4444 | | | |
| @+3 | @+2 | @+1 | @ |
| $A_{x+1}[3:0]R_{x+1}[3:0]$ | $G_{x+1}[3:0]\ B_{x+1}[3:0]$ | $A_x[3:0]\ R_x[3:0]$ | $G_x[3:0]\ B_x[3:0]$ |
| @+7 | @+6 | @+5 | @+4 |
| $A_{x+3}[3:0]R_{x+3}[3:0]$ | $G_{x+3}[3:0]\ B_{x+3}[3:0]$ | $A_{x+2}[3:0]R_{x+2}[3:0]$ | $G_{x+2}[3:0]\ B_{x+2}[3:0]$ |
| ARGB8888 | | | |
| L8 | | | |
| @+3 | @+2 | @+1 | @ |
| $L_{x+3}[7:0]$ | $L_{x+2}[7:0]$ | $L_{x+1}[7:0]$ | $L_x[7:0]$ |
| @+7 | @+6 | @+5 | @+4 |
| $L_{x+7}[7:0]$ | $L_{x+6}[7:0]$ | $L_{x+5}[7:0]$ | $L_{x+4}[7:0]$ |
| AL44 | | | |
| @+3 | @+2 | @+1 | @ |
| $A_{x+3}[3:0]\ L_{x+3}[3:0]$ | $A_{x+2}[3:0]\ L_{x+2}[3:0]$ | $A_{x+1}[3:0]\ L_{x+1}[3:0]$ | $A_x[3:0]\ L_x[3:0]$ |
| @+7 | @+6 | @+5 | @+4 |
| $A_{x+7}[3:0]\ L_{x+7}[3:0]$ | $A_{x+6}[3:0]\ L_{x+6}[3:0]$ | $A_{x+5}[3:0]\ L_{x+5}[3:0]$ | $A_{x+4}[3:0]\ L_{x+4}[3:0]$ |
| AL88 | | | |
| @+3 | @+2 | @+1 | @ |
| $A_{x+1}[7:0]$ | $L_{x+1}[7:0]$ | $A_x[7:0]$ | $L_x[7:0]$ |
| @+7 | @+6 | @+5 | @+4 |
| $A_{x+3}[7:0]$ | $L_{x+3}[7:0]$ | $A_{x+2}[7:0]$ | $L_{x+2}[7:0]$ |

**查色表 (CLUT)**

可在运行时通过 LTDC_LxCR 寄存器为每个层使能 CLUT，CLUT 仅在使用 L8、AL44 和 AL88 输入像素格式时适用于索引色。

首先, CLUT 必须加载用于替换相应像素（索引色）的原始 R、G 和 B 值的 R、G 和 B 值。每个颜色（RGB 值）在 CLUT 内都有自己对应的地址。

R、G 和 B 值及其各自的地址均通过 LTDC_LxCLUTWR 寄存器编程。

- 在使用 L8 和 AL88 输入像素格式时，CLUT 必须加载 256 个颜色。各颜色的地址在 LTDC_LxCLUTWR 寄存器的 CLUTADD 位中配置。
- 在使用 AL44 输入像素格式时，CLUT 必须仅加载 16 个颜色。各颜色的地址必须通过将 4 位 L 通道重复为 8 位进行填充，具体如下:
  - L0（索引色 0），地址 0x00 处
  - L1，地址 0x11 处
  - L2，地址 0x22 处
  - .....
  - L15，地址 0xFF 处

**颜色帧缓冲区地址**

每个层的颜色帧缓冲区均有一个起始地址，该地址通过 LTDC_LxCFBAR 寄存器进行配置。 当使能某个层时，将从颜色帧缓冲区中获取该数据。

**颜色帧缓冲区长度**

每层均设置颜色帧缓冲区的总行长(单位为字节)和行数，二者可分别通过 LTDC_LxCFBLR 和

LTDC_LxCFBLNR 寄存器进行配置。

行长和行数的设置用于阻止超出帧缓存结尾的数据被预取到层对应的 FIFO 中。

- 如果设置为低于所需字节，则会产生 FIFO 下溢中断（如果之前已使能）。
- 如果设置为高于实际所需字节，则将丢弃从 FIFO 中读取的无用数据，无用数据不会显示。

**颜色帧缓冲区间距**

每层的颜色帧缓冲区均具有可配置间距，此间距是一行的开始与下一行开始的距离（以字节为单位）。它通过 LTDC_LxCFBLR 寄存器配置。

**层混合**

混合操作始终有效，两层可按照 LTDC_LxBFCR 寄存器中配置的混合系数进行混合。

混合顺序固定，即由下至上。如果使能了两层，首先第 1 层将与背景色混合，随后第 2 层与 第 1 层和背景的混合颜色结果再次混合。请参见下图。

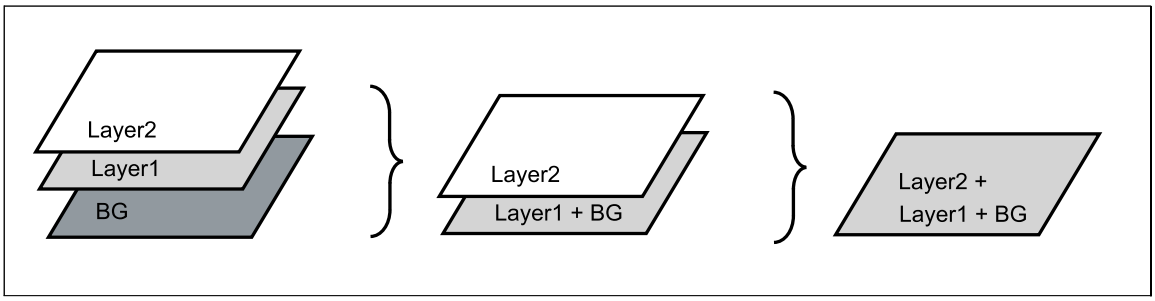



图 34-4 两层与背景混合

**默认颜色**

每层可具有 ARGB 格式的默认颜色，该颜色在定义的层窗口外使用或在层禁止时使用。默认颜色通过 LTDC_LxDCCR 寄存器配置。

始终在两层间执行混合操作，即便其中一层禁止也是如此。要避免层禁止时显示默认颜色，需将 LTDC_LxBFCR 寄存器中此层的混合系数设置为其复位值。

**色键**

色键 (RGB) 可配置为代表透明像素。

使能色键后，当前像素（格式转换后、混合前的像素）将与色键进行比较。如果当前像素与编程的 RGB 值相匹配，则该像素的所有通道 (ARGB) 均设置为 0。

运行时，可配置色键值并用其替换像素 RGB 值。

色键通过 LTDC_LxCKCR 寄存器配置。

## 34.5. LTDC 中断

LTDC 提供四个可屏蔽中断，这些中断经逻辑或运算后产生两个中断向量。

中断源可通过 LRDC_IER 寄存器单独使能或禁止。将相应的屏蔽位置 1 可使能相应中断。

发生如下事件时会产生两个中断：

- 行中断：达到编程的行时产生。行中断的位置在 LTDC_LIPCR 寄存器中编程
- 寄存器重载中断：在垂直消隐周期内执行影子寄存器重载时产生
- FIFO 下溢中断：从空层 FIFO 中请求像素时产生

- 传输错误中断：数据传输期间出现 AHB 总线错误时产生这些中断事件与 NVIC 控制器相连，如下图所示。

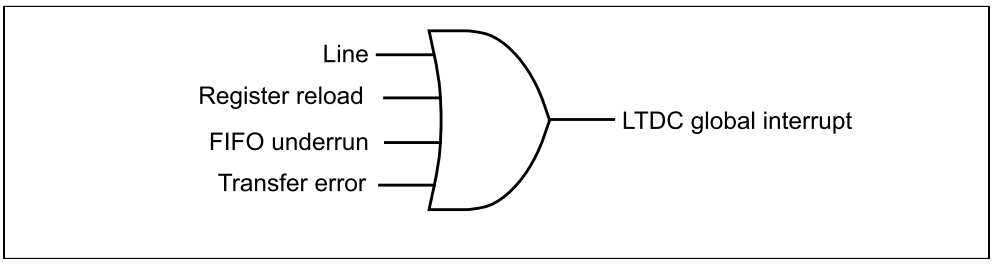



图 34-5 中断事件

表 34-3 LTDC 中断请求

| 中断事件 | 事件标志 | 使能控制位 |
|---|---|---|
| 行 | LIF | LIE |
| 寄存器重载 | RRIF | RRIEN |
| FIFO 下溢 | FUDERRIF | FUDERRIE |
| 传输错误 | TERRIF | TERRIE |

## 34.6. LTDC 编程步骤

- 在 RCC 寄存器中使能 LTDC 时钟。
- 按照面板数据表配置所需像素时钟。
- 按照 LTDC 全局配置参数中所述的面板数据表配置同步时序：VSYNC、HSYNC、垂直和水平后沿、有效数据区域以及前沿时序。
- 配置 LTDC_GCR 寄存器中的同步信号和时钟极性。
- 必要时配置 LTDC_BCCR 寄存器中的背景色。
- 配置 LTDC_IER 和 LTDC_LIPCR 寄存器中的所需中断。
- 通过执行以下编程操作配置第 1/2 层的参数：
  - 编程 LTDC_LxWHPCR 和 LTDC_WVPCR 寄存器中的层窗口的水平和垂直位置。层窗口必须位于有效数据区域；
  - 编程 LTDC_LxPFCR 寄存器中的像素输入格式；
  - 编程 LTDC_LxCFBAR 寄存器中的颜色帧起始地址；
  - 编程 LTDC_LxCFBLR 寄存器中的颜色帧缓冲区的行长和间距；
  - 编程 LTDC_LxCFBLNR 寄存器中的颜色帧缓冲区的行数；
  - 必要时在 LTDC_LxCLUTWR 寄存器中为 CLUT 加载 RGB 值及其地址；
  - 必要时分别在 LTDC_LxDCCR 和 LTDC_LxBFCR 寄存器中配置默认颜色和混合系数。
- 使能 LTDC_LxCR 寄存器中的第 1/2 层，必要时使能 CLUT。
- 必要时可分别在 LTDC_GCR 和 LTDC_LxCKCR 寄存器中使能抖动和色键。也可以实时使能这两个功能。
- 通过 LTDC_SRCR 寄存器将影子寄存器重载到活动寄存器中。
- 使能 LTDC_GCR 寄存器中的 LCD-TFT 控制器。

- 除 CLUT 外，所有层参数均可实时修改。新配置必须通过配置 LTDC_SRCR 寄存器立即重载或在垂直消隐周期内重载。

*注：所有层的寄存器均执行影子操作。一旦对某个寄存器执行写操作，便不应在重载完成前再次进行修改。因此，如果在尚未重载时对同一寄存器执行新的写操作，则将覆盖之前的配置。*

# 34.7. LTDC 寄存器

## 34.7.1. LTDC 同步大小配置寄存器 (LTDC_SSCR)

此寄存器定义水平同步像素数减 1 以及垂直同步行数减 1。

偏移地址： 0x08

复位值： 0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:28 | 保留 | 必须保持复位值 |
| 位 27:16 | HSW[11:0] | 水平同步宽度（以像素时钟周期为单位）<br>这些位定义水平同步像素数减 1 |
| 位 15:11 | 保留 | 必须保持复位值 |
| 位 10:0 | VSH[10:0] | 垂直同步高度（以水平扫描行为单位）<br>这些位定义垂直同步高度减 1。它代表水平同步行的数量。 |

## 34.7.2. LTDC 后沿配置寄存器 (LTDC_BPCR)

此寄存器定义水平同步像素加水平后沿像素的累加数减 1（HSYNC 宽度 + HBP- 1）以及垂直同步行加垂直后沿行的累加数减 1（VSYNC 高度 + VBP - 1）。

偏移地址： 0x0C

复位值： 0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 位 31:28 | 保留 | 必须保持复位值 |
| 位 27:16 | AHBP[11:0] | 累加水平后沿（以像素时钟周期为单位）<br>这些位定义累加水平后沿宽度（水平同步像素加水平后沿像素减 1）。 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 水平后沿是水平同步信号变为无效到下一扫描行的有效显示开始之间的间隔。 |
| 位 15:11 | 保留 | 必须保持复位值 |
| 位 10:0 | AVBP[10:0] | 累加垂直后沿（以水平扫描行为单位）<br>这些位定义累加垂直后沿宽度（垂直同步行加垂直后沿行减 1）。<br>垂直后沿是帧开始到下一帧的首个有效扫描行开始所包含的水平扫描行的数量。 |

### 34.7.3. LTDC 有效宽度配置寄存器 (LTDC_AWCR)

此此寄存器定义水平同步像素加水平后沿像素加有效像素的累加数减 1（HSYNC 宽度 + HBP + 有效宽度 - 1）以及垂直同步行加垂直后沿行加有效行的累加数减 1（VSYNC 高度+ BVBP + 有效高度 - 1）。

偏移地址： 0x10

复位值： 0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:28 | 保留 | 必须保持复位值。 |
| 27:16 | AAW[11:0] | 累加有效宽度（以像素时钟周期为单位）<br>这些位定义累加有效宽度（水平同步像素加水平后沿像素加有效像素减 1）。<br>有效宽度是面板扫描行的有效显示区中的像素数。支持的最大有效宽度为 0x400。 |
| 15:11 | 保留 | 必须保持复位值。 |
| 10:0 | AAH[10:0] | 累加有效高度（以水平扫描行为单位）<br>这些位定义累加高度（垂直同步行加垂直后沿行加有效高度行减 1）。有效高度是面板中 的有效行数。支持的最大有效高度为 0x300。 |

### 34.7.4. LTDC 总宽度配置寄存器 (LTDC_TWCR)

此寄存器定义水平 同步像素加水平后沿像素加有效像素加水平前沿像素的累加数减 1 （HSYNC 宽度 + HBP + 有效宽度 + HFP - 1） 以及垂直同步行加垂直后沿行加有效行加垂直前沿行的累加数减 1（VSYNC 高度 + BVBP + 有效高度 + VFP - 1）。

偏移地址： 0x14

复位值： 0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:28 | 保留 | 必须保持复位值。 |
| 27:16 | TOTALW[11:0] | 总宽度（以像素时钟周期为单位）<br>这些位定义累加总宽度（水平同步像素加水平后沿像素加有效宽度加水平前沿像素 减 1）。 |
| 15:11 | 保留 | 必须保持复位值。 |
| 10:0 | TOTALH[10:0] | 总高度（以水平扫描行为单位）<br>这些位定义累加高度（垂直同步行加垂直后沿行加有效高度加垂直前沿行减 1）。 |

### 34.7.5. LTDC 全局控制寄存器 (LTDC_GCR)

此寄存器定义 LCD_TFT 控制器的全局配置。

偏移地址： 0x18

复位值：0x0000 2220

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| HSPOL | VSPOL | DEPOL | PCPOL | Reserved | | | | | | | | | | | DEN |
| rw | rw | rw | rw | | | | | | | | | | | | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | DRW[2:0] | | | Reserved | DGW[2:0] | | | Reserved | DBW[2:0] | | | Reserved | | | LTDCEN |
| | rw | rw | rw | | rw | rw | rw | | rw | rw | rw | | | | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | HSPOL | 水平同步极性<br>此位由软件置 1 和清零。<br>0：水平同步极性低电平有效<br>1：水平同步极性高电平有效 |
| 30 | VSPOL | 垂直同步极性<br>此位由软件置 1 和清零。<br>0：垂直同步低电平有效<br>1：垂直同步高电平有效 |
| 29 | DEPOL | 数据使能极性<br>此位由软件置 1 和清零。<br>0：数据使能极性低电平有效<br>1：数据使能极性高电平有效 |
| 28 | PCPOL | 像素时钟极性<br>此位由软件置 1 和清零。<br>0：输入像素时钟<br>1：反相输入像素时钟 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 27:17 | 保留 | 必须保持复位值。 |
| 16 | DEN | 抖动使能<br>此位由软件置 1 和清零。<br>0：禁止抖动<br>1：使能抖动 |
| 15 | 保留 | 必须保持复位值。 |
| 14:12 | DRW[2:0] | 抖动红色宽度<br>这些位返回抖动红色位 |
| 11 | 保留 | 必须保持复位值。 |
| 10:8 | DGW[2:0] | 抖动绿色宽度<br>这些位返回抖动绿色位 |
| 7 | 保留 | 必须保持复位值。 |
| 6:4 | DBW[2:0] | 抖动蓝色宽度<br>这些位返回抖动蓝色位 |
| 3:1 | 保留 | 必须保持复位值。 |
| 0 | LTDCEN | LCD-TFT 控制器使能位<br>此位由软件置 1 和清零。<br>0：禁止 LTDC<br>1：使能 LTDC |

### 34.7.6. LTDC 影子重载配置寄存器 (LTDC_SRCR)

此寄存器允许立即或在垂直消隐周期内将影子寄存器的值重载到活动寄存器中。影子寄存器均为第 1 层和第 2 层寄存器，但 LTDC_L1CLUTWR 和 LTDC_L2CLUTWR 除外。

偏移地址： 0x24

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:2 | 保留 | 必须保持复位值。 |
| 1 | VBR | 垂直消隐重载<br>此位由软件置 1，只有重载后才由硬件清零。（一旦置 1，便无法通过寄存器写操作清零）<br>0：无影响<br>1：影子寄存器在垂直消隐周期（有效显示区后的第一行开始时）内重载 |
| 0 | IMR | 立即重载<br>此位由软件置 1，只有重载后才由硬件清零。<br>0：无影响<br>1：影子寄存器立即重载 |

注： 影子寄存器回读有效值。直至重载完成，才会读取“旧”值

### 34.7.7. LTDC 背景色配置寄存器 (LTDC_BCCR)

此寄存器定义背景色 (RGB888)。

偏移地址： 0x2C

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | BCRED[7:0] | | | | | | | |
| | | | | | | | | rw | rw | rw | rw | rw | rw | rw | rw |

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:24 | 保留 | 必须保持复位值。 |
| 23:16 | BCRED[7:0] | 背景红色值<br>这些位配置背景红色值 |
| 15:8 | BCGREEN[7:0] | 背景绿色值<br>这些位配置背景绿色值 |
| 7:0 | BCBLUE[7:0] | 背景蓝色值<br>这些位配置背景蓝色值 |

### 34.7.8. LTDC 中断使能寄存器 (LTDC_IER)

此寄存器通过将对应的位置 1 来确定哪一个状态标志位产生中断请求。

偏移地址： 0x34

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | RRIE | TERRIE | FUIE | LIE |
| | | | | | | | | | | | | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:4 | 保留 | 必须保持复位值。 |
| 3 | RRIE | 寄存器重载中断使能<br>此位由软件置 1 和清零<br>0：禁止寄存器重载中断<br>1：使能寄存器重载中断 |
| 2 | TERRIE | 传输错误中断使能<br>此位由软件置 1 和清零<br>0：禁止传输错误中断<br>1：使能传输错误中断 |
| 1 | FUIE | FIFO 下溢中断使能 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| | | 此位由软件置 1 和清零<br>0：禁止 FIFO 下溢中断<br>1：使能 FIFO 下溢中断 |
| 0 | LIE | 行中断使能<br>此位由软件置 1 和清零<br>0：禁止行中断<br>1：使能行中断 |

### 34.7.9. LTDC 中断状态寄存器 (LTDC_ISR)

此寄存器返回中断状态标志。

偏移地址： 0x38

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Reserved | | | | | | | | | | | | RRIF | TERRIF | FUIF | LIF |
| | | | | | | | | | | | | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:24 | 保留 | 必须保持复位值。 |
| 3 | RRIF | 寄存器重载中断标志<br>0：未产生寄存器重载中断<br>1：发生垂直消隐重载时（以及到达有效区域后的第一行时）产生寄存器重载中断 |
| 2 | TERRIF | 传输错误中断标志<br>0：未产生传输错误中断<br>1：出现总线错误时产生传输错误中断 |
| 1 | FUIF | FIFO 下溢中断标志<br>0：未产生 FIFO 下溢中断<br>1：当其中一个层 FIFO 为空并从 FIFO 读取像素数据时，将产生 FIFO 下溢中断 |
| 0 | LIF | 行中断标志<br>0：未产生行中断<br>1：到达编程的行时产生行中断 |

### 34.7.10. LTDC 中断清零寄存器 (LTDC_ICR)

偏移地址： 0x3C

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | CRRIF | CTERRIF | CFUIF | CLIF |
| | | | | | | | | | | | | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:24 | 保留 | 必须保持复位值。 |
| 3 | CRRIF | 寄存器重载中断清零标志<br>0：无影响<br>1：将 LTDC_ISR 寄存器中的 RRIF 标志清零 |
| 2 | CTERRIF | 传输错误中断清零标志<br>0：无影响<br>1：将 LTDC_ISR 寄存器中的 TERRIF 标志清零 |
| 1 | CFUIF | FIFO 下溢中断清零标志<br>0：无影响<br>1：将 LTDC_ISR 寄存器中的 FUDERRIF 标志清零 |
| 0 | CLIF | 行中断清零标志<br>0：无影响<br>1：将 LTDC_ISR 寄存器中的 LIF 标志清零 |

### 34.7.11. LTDC 行中断位置配置寄存器 (LTDC_LIPCR)

此寄存器定义行中断的位置。要编程的行值取决于时序参数。

偏移地址： 0x40

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:11 | 保留 | 必须保持复位值。 |
| 10:0 | LIPOS[10:0] | 行中断位置<br>这些位配置行中断位置 |

### 34.7.12. LTDC 当前位置状态寄存器 (LTDC_CPSR)

偏移地址： 0x44

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | CXPOS[15:0] | 当前 X 位置<br>这些位返回当前 X 位置 |
| 15:0 | CYPOS[15:0] | 当前 Y 位置<br>这些位返回当前 Y 位置 |

### 34.7.13. LTDC 当前显示状态寄存器 (LTDC_CDSR)

此寄存器返回由 HSYNC、VSYNC 和水平/垂直 DE 信号控制的当前显示阶段的状态。示例：如果当前显示阶段为垂直同步阶段，则 VSYNCS 位置 1（高电平有效）。示例：如果当前显示阶段为水平同步阶段，则 HSYNCS 位高电平有效。

偏移地址： 0x48

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:4 | 保留 | 必须保持复位值。 |
| 3 | HSYNCS | 水平同步显示状态<br>0：低电平有效<br>1：高电平有效 |
| 2 | VSYNCS | 垂直同步显示状态<br>0：低电平有效<br>1：高电平有效 |
| 1 | HDES | 水平数据使能显示状态<br>0：低电平有效<br>1：高电平有效 |
| 0 | VDES | 垂直数据使能显示状态<br>0：低电平有效<br>1：高电平有效 |

*注：返回的状态不取决于LTDC_GCR 寄存器中配置的极性，而是返回当前的有效显示阶段。*

### 34.7.14. LTDC 第 x 层控制寄存器 (LTDC_LxCR)（其中， x = 1..2)

此寄存器返回由 HSYNC、VSYNC 和水平/垂直 DE 信号控制的当前显示阶段的状态。示例：

如果当前显示阶段为垂直同步阶段，则 VSYNCS 位置 1（高电平有效）。示例：如果当前显示阶段为水平同步阶段，则 HSYNCS 位高电平有效。

偏移地址：0x84 + 0x80 x (第 x 层 - 1) ， x = 1 或 2

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:6 | 保留 | 必须保持复位值。 |
| 5 | MINC | 数据帧缓冲的地址增加模式<br>此位由软件置 1 和清零。<br>0：DMA 时地址递增<br>1：DMA 时地址固定 |
| 4 | CLUTEN | 查色表使能<br>此位由软件置 1 和清零。<br>0：禁止查色表<br>1：使能查色表<br>CLUT 仅对 L8、AL44 和 AL88 像素格式有意义。 |
| 3:2 | 保留 | 必须保持复位值。 |
| 1 | COLKEN | 色键使能<br>此位由软件置 1 和清零。<br>0：禁止色键<br>1：使能色键 |
| 0 | LEN | 层使能<br>此位由软件置 1 和清零。<br>0：禁止层<br>1：使能层 |

### 34.7.15. LTDC 第 x 层窗口水平位置配置寄存器 (LTDC_LxWHPCR)（其中 x=1..2)

此寄存器定义第 1 层或第 2 层窗口的水平位置（第一个和最后一个像素）。一行的第一个可见像素是在 LTDC_BPCR 寄存器中编程的 AHBP[10:0] bits + 1 的值。一行的最后一个可见像素是在 LTDC_AWCR 寄存器中编程的 AAW[10:0] bits 的值。此范围内的所有值均为允许值。

偏移地址：0x88+ 0x80 x (第 x 层 - 1) ， x = 1 或 2

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:28 | 保留 | 必须保持复位值。 |
| 27:16 | WHSPPOS[11:0] | 窗口水平结束位置<br>这些位配置层窗口的一行的最后一个可见像素。 |
| 15:12 | 保留 | 必须保持复位值。 |
| 11:0 | WHSTPOS[11:0] | 窗口水平起始位置<br>这些位配置层窗口的一行的第一个可见像素。 |

示例：

LTDC_BPCR 寄存器配置为 0x000E0005（AHBP[11:0] 为 0xE），LTDC_AWCR 寄存器配置为 0x028E01E5（AAW[11:0] 为 0x28E)。要配置大小为 630x460 的窗口的水平位置（有效数据区域中的水平起始偏移为 5 个像素)。

1. 层窗口的第一个像素：WHSTPOS[11:0] 应编程为 0x14 (0xE+1+0x5)
2. 层窗口的最后一个像素：WHSPPOS[11:0] 应编程为 0x28A

### 34.7.16. LTDC 第 x 层窗口垂直位置配置寄存器 (LTDC_LxWVPCR)（其中 x=1..2)

此寄存器定义第 1 层或第 2 层窗口的垂直位置（第一行或最后一行)。一个帧的第一个可见行是在 LTDC_BPCR 寄存器中编程的 AVBP[10:0] bits + 1 的值。一个帧的最后一个可见行是在 LTDC_AWCR 寄存器中编程的 AAH[10:0] bits 的值。此范围内的所有值均为允许值。

偏移地址：0x8C + 0x80 x (第 x 层-1) ， x = 1 或 2

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | WVSPPOS[10:0] | | | | | | | | | | |
| | | | | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:27 | 保留 | 必须保持复位值。 |
| 26:16 | WVSPPOS[10:0] | 窗口垂直结束位置<br>这些位配置层窗口的最后一个可见行。 |
| 15:11 | 保留 | 必须保持复位值。 |
| 10:0 | WVSTPOS[10:0] | 窗口垂直起始位置<br>这些位配置层窗口的第一个可见行。 |

示例：

LTDC_BPCR 寄存器配置为 0x000E0005（AVBP[10:0] 为 0x5)，LTDC_AWCR 寄存器配置为 0x028E01E5（AAH[10:0] 为 0x1E5)。要配置大小为 630x460 的窗口的垂直位置（有效数据区域中的垂直起始偏移为 8 行)：

1. 层窗口的第一行：WVSTPOS[10:0] 应编程为 0xE (0x5 + 1 + 0x8)
2. 层窗口的最后一行：WVSPPOS[10:0] 应编程为 0x1DA

### 34.7.17. LTDC 第 x 层色键配置寄存器 (LTDC_LxCKCR)（其中 x=1..2)

此寄存器定义色键使用的色键值 (RGB)。

偏移地址：0x90 + 0x80 x (第 x 层 - 1) ， x = 1 或 2

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | CKRED[7:0] | | | | | | | |
| | | | | | | | | rw | rw | rw | rw | rw | rw | rw | rw |

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:24 | 保留 | 必须保持复位值。 |
| 23:16 | CKRED[7:0] | 色键红色值 |
| 15:8 | CKGREEN[7:0] | 色键绿色值 |
| 7:0 | CKBLUE[7:0] | 色键蓝色值 |

### 34.7.18. LTDC 中断使能寄存器 (LTDC_IER)

此寄存器定义层的帧缓冲区中存储的数据所使用的像素格式。像素数据从帧缓冲区读取，随后转换为内部格式 8888 (ARGB)。

偏移地址：0x94 + 0x80 x (第 x 层 - 1) ， x = 1 或 2

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:3 | 保留 | 必须保持复位值。 |
| 2:0 | PF[2:0] | 像素格式<br>这些位配置像素格式<br>000：ARGB8888<br>001：RGB888<br>010：RGB565<br>011：ARGB1555<br>100：ARGB4444<br>101：L8（8 位 Luminance）<br>110：AL44（4 位 Alpha，4 位 Luminance） |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 111：AL88（8 位 Alpha，8 位 Luminance） |

### 34.7.19. LTDC 第 x 层常数 Alpha 配置寄存器 (LTDC_LxCACR)（其中 x=1..2)

此寄存器定义在 alpha 混合中使用的常数 alpha 值（由硬件实现 255 等分)。请参见 LTDC_LxBFCR 寄存器。

偏移地址：0x98 + 0x80 x (第 x 层 - 1)，x = 1 或 2

复位值：(第 x 层 - 1) 0x0000 00FF

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 必须保持复位值。 |
| 7:0 | CONSTA[7:0] | 常数 Alpha<br>这些位配置混合时使用的常数 Alpha。常数 Alpha 由硬件实现 255 等分。<br>示例：如果编程的常数 Alpha 为 0xFF，则常数 Alpha 值为 255/255=1 |

### 34.7.20. LTDC 第 x 层默认颜色配置寄存器 (LTDC_LxDCCR)（其中 x=1..2)

此寄存器定义采用 ARGB 格式的层的默认颜色。默认颜色在定义的层窗口外使用或在层禁止时使用。

复位值 0x00000000 定义了透明黑色。

偏移地址：0x9C + 0x80 x (第 x 层 - 1)，x = 1 或 2

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:24 | DCALPHA[7:0] | 默认 Alpha 值<br>这些位配置默认 alpha 值 |
| 23:16 | DCRED[7:0] | 默认颜色红色<br>这些位配置默认红色值 |
| 15:8 | DCGREEN[7:0] | 默认颜色绿色<br>这些位配置默认绿色值 |
| 7:0 | DCBLUE[7:0] | 默认颜色蓝色 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 这些位配置默认蓝色值 |

### 34.7.21. LTDC 第 x 层混合系数配置寄存器 (LTDC_LxBFCR)（其中 x=1..2)

此寄存器定义混合系数 F1 和 F2。

通用混合公式为： BC = BF1 x C + BF2 x Cs

- BC = 混合后的颜色
- BF1 = 混合系数 1
- C = 当前层颜色
- BF2 = 混合系数 2
- Cs = 底层混合后的颜色

偏移地址：0xA0 + 0x80 x (第 x 层 - 1) ， x = 1 或 2

复位值：0x0000 0607

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | BF1[2:0] | | | Reserved | | | | | BF2[2:0] | | |
| | | | | | rw | rw | rw | | | | | | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:11 | 保留 | 必须保持复位值。 |
| 10:8 | BF1[2:0] | 混合系数 1<br>这些位选择混合系数 F1<br>000：保留<br>001：保留<br>010：保留<br>011：保留<br>100：常数 Alpha<br>101：保留<br>110：像素 Alpha x 常数 Alpha<br>111：保留 |
| 7:3 | 保留 | 必须保持复位值。 |
| 2:0 | BF2[2:0] | 混合系数 2<br>这些位选择混合系数 F2<br>000：保留<br>001：保留<br>010：保留<br>011：保留<br>100：保留<br>101：1 - 常数 Alpha<br>110：保留<br>111：1 -（像素 Alpha x 常数 Alpha） |

*注： 常数 Alpha 值是在寄存器中编程的值，由硬件实现 255 等分。*

示例：仅使能第 1 层，BF1 配置为常数 Alpha

BF2 配置为 1 — 常数 Alpha

常数 Alpha：在 LxCACR 寄存器中编程的常数 Alpha 为 240 (0xF0)。因此，常数 Alpha 值为 240/255 = 0.94

C：当前层颜色为 128

Cs：背景色为 48

第 1 层与背景色混合。

BC = 常数 Alpha x C + (1 - 常数 Alpha) x Cs = 0.94 x 128 + (1- 0.94) x 48 = 123。

### 34.7.22. LTDC 第 x 层颜色帧缓冲区地址寄存器 (LTDC_LxCFBAR)（其中 x=1..2)

此寄存器定义颜色帧缓冲区的起始地址，该地址必须指向帧缓冲区中存储的层的左上角像素的像素数据地址。

偏移地址： 0xAC + 0x80 x (第 x 层 - 1) ， x = 1 或 2

复位值： 0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | CFBADD[31:0] | 颜色帧缓冲区起始地址<br>这些位定义颜色帧缓冲区的起始地址。 |

### 34.7.23. LTDC 第 x 层颜色帧缓冲区长度寄存器 (LTDC_LxCFBLR)（其中 x=1..2)

此寄存器定义颜色帧缓冲区的行长和行间距。

偏移地址： 0xB0 + 0x80 (第 x 层 - 1) ， x = 1 或 2

复位值： 0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:29 | 保留 | 必须保持复位值。 |
| 28:16 | CFBP[17:0] | 颜色帧缓冲区间距（以字节为单位）<br>这些位定义从像素某行的起始处到下一行的起始处的增量（以字节为单位）。 |
| 15:13 | 保留 | 必须保持复位值。 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 12:0 | CFBLL[12:0] | 颜色帧缓冲区行长<br>这些位定义一行像素的长度（以字节为单位） 。<br>行长的计算方法为：有效宽度 x 每像素的字节数。 |

示例：

采用 RGB565（每像素 2 个字节）格式且宽度为 256 像素的帧缓冲区（每行总字节数为 256x2=512）需要向此寄存器写入值 0x02000200（其中， 间距 = 行长）。

采用 RGB888（每像素 3 个字节）格式且宽度为 320 像素的帧缓冲区（每行总字节数为 320x3=960）需要向此寄存器写入值 0x03C003C0（其中，间距 = 行长）。

### 34.7.24. LTDC 第 x 层颜色帧缓冲区长度寄存器 (LTDC_LxCFBLR)（其中 x=1..2)

此寄存器定义颜色帧缓冲区中的行数。

偏移地址： 0xB4 + 0x80 x (第 x 层 - 1) ， x = 1 或 2

复位值： 0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:11 | 保留 | 必须保持复位值。 |
| 10:0 | CFBLNBR[10:0] | 帧缓冲区行数<br>这些位用来定义缓冲区中的行数，行数跟有效高度对应。 |

*注：*

*行数和行长设置定义针对每层从每个帧中获取的数据量。如果配置为低于所需字节，则会产生 FIFO 下溢中断（如果使能）。*

*另一方面，起始地址和间距设置定义存储器中每行的正确起始位置。*

### 34.7.25. LTDC 第 x 层 CLUT 写寄存器 (LTDC_LxCLUTWR) （其中 x=1..2)

此寄存器定义 CLUT 地址和 RGB 值。

偏移地址： 0xC4 + 0x80 x (第 x 层 - 1) ， x = 1 或 2

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CLUTADD[7:0] | | | | | | | | RED[7:0] | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:24 | CLUTADD[7:0] | CLUT 地址 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| | | 这些位配置每个 RGB 值的 CLUT 地址（CLUT 内的颜色位置） |
| 23:16 | RED[7:0] | 红色值<br>这些位配置红色值 |
| 15:8 | GREEN[7:0] | 绿色值<br>这些位配置绿色值 |
| 7:0 | BLUE[7:0] | 蓝色值<br>这些位配置蓝色值 |

*注：*

*CLUT 写寄存器只应在消隐周期内或在层禁止时配置。CLUT 可通过 LTDC_LxCR 寄存器使能和禁止。*

*CLUT 仅对 L8、AL44 和 AL88 像素格式有意义。*

### 34.7.26. LTDC AHB BURST 配置寄存器 (LTDC_CFG0)

此寄存器配置 AHB 突发传输的长度。

偏移地址： 0x60

复位值： 0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:3 | 保留 | 必须保持复位值。 |
| 2:0 | BURST_TYPE[2:0] | AHB 使用的 burst 类型<br>此位由软件置 1，只有重载后才由硬件清零。<br>011：4<br>101：8<br>111：16 |

# 35. 修订历史

表 35-1 修订历史

| 日期 | 版本 | 描述 |
| --- | --- | --- |
| 2024-08-07 | 1.00 | 初稿完成 |
| 2024-11-04 | 1.10 | 更新 NVIC 章节向量表描述 |
| 2025-02-13 | 1.20 | 更新存储器映射表 |
| 2025-05-28 | 1.30 | 增加 LTDC、JPEGD 章节内容 |
| 2025-12-25 | 1.40 | 更新模板和优化排版 |
| | | |
| | | |

# DISCLAIMER

The information in this document is current as of the published date. The information is subject to change without notice. For actual design-in, refer to the latest publications of MEGAHUNT Datasheets, and other official documents, for the most up-to-date specifications of MEGAHUNT products. Not all products and/or types are available in every country. Please contact a MEGAHUNT sales representative for availability and additional information.

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of MEGAHUNT. MEGAHUNT assumes no responsibility for any errors or inaccuracies that may appear in this document.

MEGAHUNT disclaims any liability for patent, copyright or other intellectual property rights infringements of third parties by or arising from the use of MEGAHUNT products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of MEGAHUNT or others. Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. MEGAHUNT assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.

The fonts, images, and other logos used by our company are legally authorized or legally used free of charge without infringing any third-party's rights.

Our company reserves the right to modify or revoke this disclaimer.

If you have any doubts, please contact us promptly and we will actively resolve them.

For more information about MEGAHUNT products and solutions, visit our website at www.megahuntmicro.com