# GigaDevice Semiconductor Inc.

# GD32E11x
# Arm® Cortex®-M4 32-bit MCU

适用于 **GD32E113xx**

# 用户手册

1.5 版本

(2026 年 7 月)

# 目 录

目 录 ..... 2
图索引 ..... 15
表索引 ..... 21
1. 系统及存储器架构 ..... 23
1.1. ARM Cortex-M4 处理器 ..... 23
1.2. 系统架构 ..... 24
1.3. 存储器映射 ..... 26
1.3.1. 位带操作 ..... 30
1.3.2. 片上 SRAM 存储器 ..... 31
1.3.3. 片上 FLASH 存储器概述 ..... 31
1.4. 引导配置 ..... 31
1.5. 设备电子签名 ..... 32
1.5.1. 存储器容量信息 ..... 32
1.5.2. 设备唯一 ID(96 位) ..... 32
1.6. 系统配置寄存器 ..... 33
2. 闪存控制器(FMC) ..... 35
2.1. 简介 ..... 35
2.2. 主要特征 ..... 35
2.3. 功能说明 ..... 35
2.3.1. 闪存结构 ..... 35
2.3.2. 读操作 ..... 36
2.3.3. FMC_CTL 寄存器解锁 ..... 37
2.3.4. 页擦除 ..... 38
2.3.5. 整片擦除 ..... 39
2.3.6. 主存储闪存块编程 ..... 40
2.3.7. OTP 编程 ..... 42
2.3.8. 选项字节擦除 ..... 42

2.3.9. 选项字节编程 ..... 43
2.3.10. 选项字节说明 ..... 43
2.3.11. 页擦除/编程保护 ..... 44
2.3.12. 安全保护 ..... 44
**2.4. FMC 寄存器 ..... 46**
2.4.1. 等待状态寄存器 (FMC_WS) ..... 46
2.4.2. 解锁寄存器 (FMC_KEY) ..... 47
2.4.3. 选项字节操作解锁寄存器(FMC_OBKEY) ..... 47
2.4.4. 状态寄存器 (FMC_STAT) ..... 48
2.4.5. 控制寄存器 (FMC_CTL) ..... 48
2.4.6. 地址寄存器 (FMC_ADDR) ..... 50
2.4.7. 选项字节状态寄存器 (FMC_OBSTAT) ..... 50
2.4.8. 擦除/编程保护寄存器 (FMC_WP) ..... 51
2.4.9. 产品 ID 寄存器 (FMC_PID) ..... 51
**3. 电源管理单元(PMU) ..... 52**
**3.1. 简介 ..... 52**
**3.2. 主要特征 ..... 52**
**3.3. 功能说明 ..... 53**
3.3.1. 电池备份域 ..... 53
3.3.2. $V_{DD}$ / $V_{DDA}$ 电源域 ..... 54
3.3.3. $V_{CORE}$ 电源域 ..... 56
3.3.4. 省电模式 ..... 56
**3.4. PMU 寄存器 ..... 58**
3.4.1. 控制寄存器(PMU_CTL) ..... 58
3.4.2. 电源控制和状态寄存器(PMU_CS) ..... 59
**4. 备份寄存器(BKP) ..... 61**
**4.1. 简介 ..... 61**
**4.2. 主要特征特性 ..... 61**
**4.3. 功能说明 ..... 61**
4.3.1. RTC 时钟校准 ..... 61
4.3.2. 侵入检测 ..... 61
**4.4. BKP 寄存器 ..... 62**
4.4.1. 备份数据寄存器 (BKP_DATAx) (x= 0..41) ..... 62

4.4.2. RTC 信号输出控制寄存器 (BKP_OCTL) ..... 62
4.4.3. 侵入引脚控制寄存器 (BKP_TPCTL) ..... 63
4.4.4. 侵入控制状态寄存器 (BKP_TPCS) ..... 63

**5. 复位和时钟单元(RCU) ..... 65**

**5.1. 复位控制单元(RCTL) ..... 65**

5.1.1. 简介 ..... 65
5.1.2. 功能说明 ..... 65

**5.2. 时钟控制单元(CCTL) ..... 66**

5.2.1. 简介 ..... 66
5.2.2. 主要特征 ..... 68
5.2.3. 功能说明 ..... 68

**5.3. RCU 寄存器 ..... 72**

5.3.1. 控制寄存器(RCU_CTL) ..... 72
5.3.2. 时钟配置寄存器 0(RCU_CFG0) ..... 74
5.3.3. 时钟中断寄存器(RCU_INT) ..... 77
5.3.4. APB2 复位寄存器(RCU_APB2RST) ..... 80
5.3.5. APB1 复位寄存器(RCU_APB1RST) ..... 82
5.3.6. AHB 使能寄存器(RCU_AHBEN) ..... 85
5.3.7. APB2 使能寄存器(RCU_APB2EN) ..... 86
5.3.8. APB1 使能寄存器(RCU_APB1EN) ..... 88
5.3.9. 备份域控制寄存器(RCU_BDCTL) ..... 91
5.3.10. 复位源/时钟寄存器(RCU_RSTSCK) ..... 92
5.3.11. AHB 复位寄存器(RCU_AHBRST) ..... 94
5.3.12. 时钟配置寄存器 1(RCU_CFG1) ..... 94
5.3.13. 深度睡眠模式电压寄存器(RCU_DSV) ..... 97
5.3.14. 附加时钟控制寄存器(RCU_ADDCTL) ..... 97
5.3.15. 附加时钟中断寄存器(RCU_ADDINT) ..... 98
5.3.16. APB1 附加复位寄存器(RCU_ADDAPB1RST) ..... 99
5.3.17. APB1 附加使能寄存器(RCU_ADDAPB1EN) ..... 99

**6. 时钟校准控制器(CTC) ..... 101**

**6.1. 简介 ..... 101**

**6.2. 主要特征 ..... 101**

**6.3. 功能描述 ..... 101**

6.3.1. 参考同步脉冲发生器 ..... 102
6.3.2. CTC 校准计数器 ..... 102
6.3.3. 频率评估和自动校准过程 ..... 103

6.3.4. 软件编程指南 ..... 104
6.4. CTC 寄存器 ..... 105
6.4.1. 控制寄存器 0(CTC_CTL0) ..... 105
6.4.2. 控制寄存器 1(CTC_CTL1) ..... 106
6.4.3. 状态寄存器(CTC_STAT) ..... 107
6.4.4. 中断清除寄存器(CTC_INTC) ..... 109
7. 中断/事件控制器(EXTI) ..... 110
7.1. 简介 ..... 110
7.2. 主要特征 ..... 110
7.3. 功能说明 ..... 110
7.4. 外部中断及事件框图 ..... 113
7.5. 外部中断及事件功能概述 ..... 113
7.6. EXTI 寄存器 ..... 115
7.6.1. 中断使能寄存器(EXTI_INTEN) ..... 115
7.6.2. 事件使能寄存器(EXTI_EVEN) ..... 115
7.6.3. 上升沿触发使能寄存器(EXTI_RTEN) ..... 116
7.6.4. 下降沿触发使能寄存器(EXTI_FTEN) ..... 116
7.6.5. 软件中断事件寄存器(EXTI_SWIEV) ..... 116
7.6.6. 挂起寄存器(EXTI_PD) ..... 117
8. 通用和备用输入/输出接口(GPIO 和 AFIO) ..... 118
8.1. 简介 ..... 118
8.2. 主要特征 ..... 118
8.3. 功能说明 ..... 118
8.3.1. GPIO 引脚配置 ..... 119
8.3.2. 外部中断/事件线 ..... 120
8.3.3. 备用功能(AF) ..... 120
8.3.4. 输入配置 ..... 120
8.3.5. 输出配置 ..... 120
8.3.6. 模拟配置 ..... 121
8.3.7. 备用功能(AF)配置 ..... 121
8.3.8. GPIO 锁定功能 ..... 122
8.3.9. GPIO I/O 补偿单元 ..... 122

8.4. I/O 重映射功能和调试配置 ..... 122
8.4.1. 介绍 ..... 122
8.4.2. 主要特征 ..... 122
8.4.3. JTAG/SWD 备用功能重映射 ..... 123
8.4.4. ADC AF 重映射 ..... 123
8.4.5. TIMER AF 重映射 ..... 124
8.4.6. USART AF 重映射 ..... 125
8.4.7. I2C0 备用功能重映射 ..... 126
8.4.8. SPI0/SPI2/I2S 备用功能重映射 ..... 126
8.4.9. CTC 备用功能重映射 ..... 127
8.4.10. CLK 引脚 AF 重映射 ..... 127
8.5. GPIO 寄存器 ..... 129
8.5.1. 端口控制寄存器 0 (GPIOx_CTL0, x=A..E) ..... 129
8.5.2. 端口控制寄存器 1 (GPIOx_CTL1, x=A..E) ..... 131
8.5.3. 端口输入状态寄存器 (GPIOx_ISTAT, x=A..E) ..... 132
8.5.4. 端口输出控制寄存器 (GPIOx_OCTL, x=A..E) ..... 133
8.5.5. 端口位操作寄存器 (GPIOx_BOP, x=A..E) ..... 133
8.5.6. 位清除寄存器 (GPIOx_BC, x=A..E) ..... 134
8.5.7. 端口配置锁定寄存器 (GPIOx_LOCK, x=A..E) ..... 134
8.5.8. 端口位速度寄存器 (GPIOx_ SPD, x=A..E) ..... 135
8.5.9. 事件控制寄存器 (AFIO_EC) ..... 136
8.5.10. AFIO 端口配置寄存器 0 (AFIO_PCF0) ..... 136
8.5.11. EXTI 源选择寄存器 0 寄存器 (AFIO_ EXTISS0) ..... 139
8.5.12. EXTI 源选择寄存器 1 寄存器 (AFIO_ EXTISS1) ..... 140
8.5.13. EXTI 源选择寄存器 2 寄存器 (AFIO_ EXTISS2) ..... 142
8.5.14. EXTI 源选择寄存器 3 寄存器 (AFIO_ EXTISS3) ..... 143
8.5.15. AFIO 端口配置寄存器 1 (AFIO_ PCF1) ..... 144
8.5.16. IO 补偿控制寄存器 (AFIO_CPSCTL) ..... 144
9. 循环冗余校验计算单元(CRC) ..... 146
9.1. 简介 ..... 146
9.2. 主要特征 ..... 146
9.3. 功能说明 ..... 147
9.4. CRC 寄存器 ..... 148
9.4.1. 数据寄存器(CRC_DATA) ..... 148
9.4.2. 独立数据寄存器(CRC_FDATA) ..... 148
9.4.3. 控制寄存器(CRC_CTL) ..... 149
10. 直接存储器访问控制器(DMA) ..... 150

10.1. 简介 ..... 150
10.2. 主要特征 ..... 150
10.3. 结构框图 ..... 151
10.4. 功能说明 ..... 151
10.4.1. DMA 操作 ..... 151
10.4.2. 外设握手 ..... 152
10.4.3. 仲裁 ..... 153
10.4.4. 地址生成 ..... 153
10.4.5. 循环模式 ..... 153
10.4.6. 存储器到存储器模式 ..... 154
10.4.7. 通道配置 ..... 154
10.4.8. 中断 ..... 154
10.4.9. DMA 请求映射 ..... 155
10.5. DMA 寄存器 ..... 158
10.5.1. 中断标志位寄存器(DMA_INTF) ..... 158
10.5.2. 中断标志位清除寄存器(DMA_INTC) ..... 159
10.5.3. 通道 x 控制寄存器(DMA_CHxCTL) ..... 159
10.5.4. 通道 x 计数寄存器(DMA_CHxCNT) ..... 161
10.5.5. 通道 x 外设基地址寄存器(DMA_CHxPADDR) ..... 162
10.5.6. 通道 x 存储器基地址寄存器(DMA_CHxMADDR) ..... 162
11. 调试(DBG) ..... 164
11.1. 简介 ..... 164
11.2. JTAG/SW 功能说明 ..... 164
11.2.1. 切换 JTAG/ SW 接口 ..... 164
11.2.2. 引脚分配 ..... 164
11.2.3. JTAG 链状结构 ..... 165
11.2.4. 调试复位 ..... 165
11.2.5. JEDEC-106 ID 代码 ..... 165
11.3. 调试保持功能说明 ..... 165
11.3.1. 低功耗模式调试支持 ..... 165
11.3.2. TIMER, I2C, WWDGT 和 FWDGT 外设调试支持 ..... 166
11.4. DBG 寄存器 ..... 167
11.4.1. ID 寄存器 (DBG_ID) ..... 167
11.4.2. 控制寄存器 (DBG_CTL) ..... 167

12. 模数转换器(ADC) ..... 171
12.1. 简介 ..... 171
12.2. 主要特征 ..... 171
12.3. 引脚和内部信号 ..... 172
12.4. 功能说明 ..... 173
12.4.1. 前置校准功能 ..... 173
12.4.2. ADC 时钟 ..... 174
12.4.3. ADCON 使能 ..... 174
12.4.4. 常规序列 ..... 174
12.4.5. 运行模式 ..... 174
12.4.6. 转换结果阈值监测功能 ..... 177
12.4.7. 数据存储模式 ..... 177
12.4.8. 采样时间配置 ..... 177
12.4.9. 外部触发配置 ..... 178
12.4.10. DMA 请求 ..... 178
12.4.11. ADC 内部通道 ..... 178
12.4.12. 可编程分辨率(DRES) ..... 179
12.4.13. 片上硬件过采样 ..... 179
12.5. ADC 同步模式 ..... 181
12.5.1. 独立模式 ..... 182
12.5.2. 常规并行模式 ..... 182
12.5.3. 常规快速交叉模式 ..... 183
12.5.4. 常规慢速交叉模式 ..... 183
12.6. 中断 ..... 184
12.7. ADC 寄存器 ..... 185
12.7.1. 状态寄存器 (ADC_STAT) ..... 185
12.7.2. 控制寄存器 0 (ADC_CTL0) ..... 185
12.7.3. 控制寄存器 1 (ADC_CTL1) ..... 187
12.7.4. 采样时间寄存器 0 (ADC_SAMPT0) ..... 189
12.7.5. 采样时间寄存器 1 (ADC_SAMPT1) ..... 189
12.7.6. 看门狗高阈值寄存器 (ADC_WDHT) ..... 190
12.7.7. 看门狗低阈值寄存器 (ADC_WDLT) ..... 191
12.7.8. 常规序列寄存器 0 (ADC_RSQ0) ..... 191
12.7.9. 常规序列寄存器 1 (ADC_RSQ1) ..... 192
12.7.10. 常规序列寄存器 2 (ADC_RSQ2) ..... 192
12.7.11. 常规数据寄存器 (ADC_RDATA) ..... 193

12.7.12. 过采样控制寄存器 (ADC_OVSAMPCTL) ..... 193
13. 数模转换器(DAC) ..... 196
13.1. 简介 ..... 196
13.2. 主要特征 ..... 196
13.3. 功能描述 ..... 197
13.3.1. DAC 使能 ..... 197
13.3.2. DAC 输出缓冲 ..... 197
13.3.3. DAC 数据配置 ..... 197
13.3.4. DAC 触发 ..... 198
13.3.5. DAC 转换 ..... 198
13.3.6. DAC 噪声波 ..... 198
13.3.7. DAC 输出电压 ..... 199
13.3.8. DMA 请求 ..... 199
13.3.9. DAC 并发转换 ..... 200
13.4. DAC 寄存器 ..... 201
13.4.1. DACx 控制寄存器(DAC_CTL0) ..... 201
13.4.2. DACx 软件触发寄存器(DAC_SWT) ..... 203
13.4.3. DACx_OUT0 12 位右对齐数据保持寄存器(DAC_OUT0_R12DH) ..... 204
13.4.4. DACx_OUT0 12 位左对齐数据保持寄存器(DAC_OUT0_L12DH) ..... 204
13.4.5. DACx_OUT0 8 位右对齐数据保持寄存器(DAC_OUT0_R8DH) ..... 205
13.4.6. DACx_OUT1 12 位右对齐数据保持寄存器(DAC_OUT1_R12DH) ..... 205
13.4.7. DACx_OUT1 12 位左对齐数据保持寄存器(DAC_OUT1_L12DH) ..... 206
13.4.8. DACx_OUT1 8 位右对齐数据保持寄存器(DAC_OUT1_R8DH) ..... 206
13.4.9. DACx 并发模式 12 位右对齐数据保持寄存器(DACC_R12DH) ..... 206
13.4.10. DACx 并发模式 12 位左对齐数据保持寄存器(DACC_L12DH) ..... 207
13.4.11. DACx 并发模式 8 位右对齐数据保持寄存器(DACC_R8DH) ..... 208
13.4.12. DACx_OUT0 数据输出寄存器(DAC_OUT0_DO) ..... 208
13.4.13. DACx_OUT1 数据输出寄存器(DAC_OUT1_DO) ..... 208
14. 看门狗定时器(WDGT) ..... 210
14.1. 独立看门狗定时器(FWDGT) ..... 210
14.1.1. 简介 ..... 210
14.1.2. 主要特征 ..... 210
14.1.3. 功能说明 ..... 210
14.1.4. FWDGT 寄存器 ..... 213
14.2. 窗口看门狗定时器(WWDGT) ..... 216
14.2.1. 简介 ..... 216

14.2.2. 主要特征 ..... 216
14.2.3. 功能说明 ..... 216
14.2.4. WWDGT 寄存器 ..... 218

**15. 实时时钟(RTC) ..... 220**

**15.1. 简介 ..... 220**

**15.2. 主要特征 ..... 220**

**15.3. 功能说明 ..... 220**

15.3.1. RTC 复位 ..... 221
15.3.2. RTC 读取 ..... 221
15.3.3. RTC 配置 ..... 221
15.3.4. RTC 标志位 ..... 222

**15.4. RTC 寄存器 ..... 223**

15.4.1. RTC 中断使能寄存器 (RTC_INTEN) ..... 223
15.4.2. RTC 控制寄存器 (RTC_CTL) ..... 223
15.4.3. RTC 预分频寄存器高位 (RTC_PSCH) ..... 224
15.4.4. RTC 预分频寄存器低位 (RTC_PSCL) ..... 224
15.4.5. RTC 分频器高位 (RTC_DIVH) ..... 225
15.4.6. RTC 分频器低位 (RTC_DIVL) ..... 225
15.4.7. RTC 计数寄存器高位 (RTC_CNTH) ..... 225
15.4.8. RTC 计数寄存器低位 (RTC_CNTL) ..... 226
15.4.9. RTC 闹钟寄存器高位 (RTC_ALRMH) ..... 226
15.4.10. RTC 闹钟寄存器低位 (RTC_ALRML) ..... 226

**16. 定时器(TIMER) ..... 227**

**16.1. 高级定时器(TIMERx,x=0,7) ..... 228**

16.1.1. 简介 ..... 228
16.1.2. 主要特征 ..... 228
16.1.3. 结构框图 ..... 228
16.1.4. 功能描述 ..... 229
16.1.5. TIMERx 寄存器(x=0,7) ..... 254

**16.2. 通用定时器 L0(TIMERx, x=1,2,3,4) ..... 279**

16.2.1. 简介 ..... 279
16.2.2. 主要特征 ..... 279
16.2.3. 结构框图 ..... 279
16.2.4. 功能描述 ..... 280
16.2.5. TIMERx 寄存器(x=1,2,3,4) ..... 294

**16.3. 通用定时器 L1(TIMERx, x=8,11) ..... 315**

16.3.1. 简介 ..... 315
16.3.2. 主要特征 ..... 315
16.3.3. 结构框图 ..... 315
16.3.4. 功能描述 ..... 316
16.3.5. TIMERx 寄存器(x=8,11) ..... 326
**16.4. 通用定时器 L2(TIMERx, x=9,10,12,13) ..... 338**
16.4.1. 简介 ..... 338
16.4.2. 主要特征 ..... 338
16.4.3. 结构框图 ..... 338
16.4.4. 功能描述 ..... 338
16.4.5. TIMERx 寄存器(x=9,10,12,13) ..... 345
**16.5. 基本定时器(TIMERx, x=5,6) ..... 355**
16.5.1. 简介 ..... 355
16.5.2. 主要特征 ..... 355
16.5.3. 结构框图 ..... 355
16.5.4. 功能描述 ..... 355
16.5.5. TIMERx 寄存器(x=5,6) ..... 359
**17. 通用同步异步收发器(USART) ..... 364**
**17.1. 简介 ..... 364**
**17.2. 主要特征 ..... 364**
**17.3. 功能说明 ..... 365**
17.3.1. USART 帧格式 ..... 366
17.3.2. 波特率发生 ..... 367
17.3.3. USART 发送器 ..... 367
17.3.4. USART 接收器 ..... 368
17.3.5. DMA 方式访问数据缓冲区 ..... 369
17.3.6. 硬件流控制 ..... 371
17.3.7. 多处理器通信 ..... 372
17.3.8. LIN 模式 ..... 372
17.3.9. 同步通信模式 ..... 373
17.3.10. 串行红外(IrDA SIR)编解码功能模块 ..... 374
17.3.11. 半双工通信模式 ..... 375
17.3.12. 智能卡(ISO7816-3)模式 ..... 375
17.3.13. USART 中断 ..... 377
**17.4. USART 寄存器 ..... 378**
17.4.1. 状态寄存器 0(USART_STAT0) ..... 378
17.4.2. 数据寄存器(USART_DATA) ..... 380

17.4.3. 波特率寄存器(USART_BAUD) ..... 380
17.4.4. 控制寄存器 0(USART_CTL0) ..... 381
17.4.5. 控制寄存器 1(USART_CTL1) ..... 382
17.4.6. 控制寄存器 2(USART_CTL2) ..... 384
17.4.7. 保护时间和预分频器寄存器(USART_GP) ..... 385
17.4.8. 控制寄存器 3(USART_CTL3) ..... 386
17.4.9. 接收超时寄存器(USART_RT) ..... 388
17.4.10. 状态寄存器 1(USART_STAT1) ..... 388
17.4.11. 兼容性控制寄存器(USART_CHC) ..... 389

**18. 内部集成电路总线接口(I2C) ..... 391**

**18.1. 简介 ..... 391**

**18.2. 主要特征 ..... 391**

**18.3. 功能说明 ..... 392**

18.3.1. SDA 线和 SCL 线 ..... 392
18.3.2. 数据有效性 ..... 393
18.3.3. 开始和停止信号 ..... 393
18.3.4. 时钟同步 ..... 393
18.3.5. 仲裁 ..... 394
18.3.6. I2C 通讯流程 ..... 394
18.3.7. 软件编程模型 ..... 395
18.3.8. SCL 线控制 ..... 403
18.3.9. DMA 模式下数据传输 ..... 404
18.3.10. 报文错误校验 ..... 404
18.3.11. SMBus 支持 ..... 404
18.3.12. SAM_V 支持 ..... 406
18.3.13. 状态、错误和中断 ..... 406

**18.4. I2C 寄存器 ..... 407**

18.4.1. 控制寄存器 0(I2C_CTL0) ..... 407
18.4.2. 控制寄存器 1(I2C_CTL1) ..... 408
18.4.3. 从机地址寄存器 0(I2C_SADDR0) ..... 410
18.4.4. 从机地址寄存器 1(I2C_SADDR1) ..... 410
18.4.5. 传输缓冲区寄存器(I2C_DATA) ..... 411
18.4.6. 传输状态寄存器 0(I2C_STAT0) ..... 411
18.4.7. 传输状态寄存器 1(I2C_STAT1) ..... 413
18.4.8. 时钟配置寄存器(I2C_CKCFG) ..... 414
18.4.9. 上升时间寄存器(I2C_RT) ..... 415
18.4.10. SAM 控制状态寄存器(I2C_SAMCS) ..... 416
18.4.11. 快速+模式配置寄存器(I2C_FMPCFG) ..... 417

**19. 串行外设接口/片上音频接口(SPI/I2S) ..... 418**

19.1. 简介 ..... 418
19.2. 主要特性 ..... 418
19.2.1. SPI 主要特性 ..... 418
19.2.2. I2S 主要特性 ..... 418
19.3. SPI 功能说明 ..... 419
19.3.1. SPI 结构框图 ..... 419
19.3.2. SPI 信号线描述 ..... 419
19.3.3. SPI 时序和数据帧格式 ..... 420
19.3.4. NSS 功能 ..... 421
19.3.5. SPI 运行模式 ..... 422
19.3.6. DMA 功能 ..... 430
19.3.7. CRC 功能 ..... 430
19.3.8. SPI 中断 ..... 431
19.4. I2S 功能说明 ..... 432
19.4.1. I2S 结构框图 ..... 432
19.4.2. I2S 信号线描述 ..... 432
19.4.3. I2S 音频标准 ..... 433
19.4.4. I2S 时钟 ..... 440
19.4.5. 运行 ..... 441
19.4.6. DMA 功能 ..... 445
19.4.7. I2S 中断 ..... 445
19.5. SPI/I2S 寄存器 ..... 447
19.5.1. 控制寄存器 0 (SPI_CTL0) ..... 447
19.5.2. 控制寄存器 1 (SPI_CTL1) ..... 449
19.5.3. 状态寄存器 (SPI_STAT) ..... 450
19.5.4. 数据寄存器 (SPI_DATA) ..... 451
19.5.5. CRC 多项式寄存器 (SPI_CRCPOLY) ..... 452
19.5.6. 接收 CRC 寄存器 (SPI_RCRC) ..... 452
19.5.7. 发送 CRC 寄存器 (SPI_TCRC) ..... 453
19.5.8. I2S 控制寄存器 (SPI_I2SCTL) ..... 453
19.5.9. I2S 时钟预分频寄存器 (SPI_I2SPSC) ..... 455
19.5.10. SPI0 四线 SPI 控制寄存器 (SPI_QCTL) ..... 455
20. 外部存储器控制器(EXMC) ..... 457
20.1. 简介 ..... 457
20.2. 主要特征 ..... 457
20.3. 功能说明 ..... 457

20.3.1. 结构框图 ..... 457
20.3.2. EXMC 访问基本规范 ..... 458
20.3.3. NOR/PSRAM 控制器 ..... 459
**20.4. EXMC 寄存器 ..... 478**
20.4.1. SRAM/NOR Flash 控制寄存器（EXMC_SNCTL) ..... 478
20.4.2. SRAM/NOR Flash 时序配置寄存器（EXMC_SNTCFG) ..... 479
20.4.3. SRAM/NOR Flash 写时序寄存器（EXMC_SNWTCFG) ..... 481
**21. 通用串行总线全速接口(USBFS) ..... 483**
**21.1. 概述 ..... 483**
**21.2. 主要特征 ..... 483**
**21.3. 结构框图 ..... 484**
**21.4. 信号线描述 ..... 484**
**21.5. 功能说明 ..... 484**
21.5.1. USBFS 时钟及工作模式 ..... 484
21.5.2. USB 主机功能 ..... 486
21.5.3. USB 设备功能 ..... 488
21.5.4. OTG 功能概述 ..... 489
21.5.5. 数据 FIFO ..... 489
21.5.6. 操作手册 ..... 492
**21.6. 中断 ..... 495**
**21.7. USBFS 寄存器 ..... 497**
21.7.1. 全局控制与状态寄存器组 ..... 497
21.7.2. 主机控制和状态寄存器 ..... 518
21.7.3. 设备控制和状态寄存器 ..... 529
21.7.4. 电源和时钟控制寄存器（USBFS_PWRCLKCTL) ..... 551
**22. 附录 ..... 553**
**22.1. 寄存器表中使用的缩写列表 ..... 553**
**22.2. 术语表 ..... 553**
**22.3. 可用外设 ..... 553**
**23. 版本历史 ..... 554**

# 图索引

图 1-1. Cortex®-M4 结构框图 ..... 24
图 1-2. GD32E11x 器件的系统架构示意图 ..... 26
图 2-1. 页擦除操作流程 ..... 39
图 2-2. 整片擦除操作流程 ..... 40
图 2-3. 字编程操作流程 ..... 42
图 3-1. 电源域概览 ..... 53
图 3-2. 上电 / 掉电复位波形图 ..... 55
图 3-3. LVD 阈值波形图 ..... 55
图 5-1. 系统复位电路 ..... 66
图 5-2. 时钟树 ..... 67
图 5-3. HXTAL 时钟源 ..... 68
图 5-4. 旁路模式下 HXTAL 时钟源 ..... 69
图 6-1. CTC 简介 ..... 102
图 6-2. CTC 校准计数器 ..... 103
图 7-1. EXTI 框图 ..... 113
图 8-1. GPIO 端口位的基本结构 ..... 119
图 8-2. 输入配置的基本结构 ..... 120
图 8-3. 输出配置的基本结构 ..... 121
图 8-4. 模拟配置的基本结构 ..... 121
图 8-5. 备用功能配置的基本结构 ..... 122
图 9-1. CRC 计算单元框图 ..... 147
图 10-1. DMA 结构框图 ..... 151
图 10-2. 握手机制 ..... 153
图 10-3. DMA 中断逻辑图 ..... 155
图 10-4. DMA0 请求映射 ..... 156
图 10-5. DMA1 请求映射 ..... 157
图 12-1. ADC 模块框图 ..... 173
图 12-2. 单次运行模式 ..... 174
图 12-3. 连续运行模式 ..... 175
图 12-4. 扫描运行模式，且连续转换模式失能 ..... 176
图 12-5. 扫描运行模式，连续运行模式使能 ..... 176
图 12-6. 间断转换模式 ..... 176
图 12-7. 12 位数据存储模式 ..... 177
图 12-8. 6 位数据存储模式 ..... 177
图 12-9. 20 位到 16 位的结果截断 ..... 180
图 12-10. 右移 5 位和取整的数例 ..... 180
图 12-11. ADC 同步框图 ..... 182
图 12-12. 基于 10 个通道的常规并行模式 ..... 183
图 12-13. 常规序列上的快速交叉模式(两个 ADC 的 CTN=1) ..... 183
图 12-14. 常规序列上的慢速交叉模式 ..... 184
图 13-1. DAC 结构框图 ..... 196

图 13-2. DAC LFSR 算法 ..... 199
图 13-3. DAC 三角噪声模式生成的波形 ..... 199
图 14-1. 独立看门狗定时器框图 ..... 211
图 14-2. 窗口看门狗定时器框图 ..... 216
图 14-3. 窗口看门狗定时器时序图 ..... 217
图 15-1. RTC 框图 ..... 221
图 15-2. RTC 秒信号及闹钟信号的波形 (RTC_PSC = 3, RTC_ALRM = 2) ..... 222
图 15-3. RTC 秒信号及溢出信号的波形(RTC_PSC = 3) ..... 222
图 16-1. 高级定时器结构框图 ..... 229
图 16-2. 内部时钟分频为 1 时，计数器的时序图 ..... 230
图 16-3. 当 PSC 数值从 0 变到 2 时，计数器的时序图 ..... 231
图 16-4. 向上计数时序图，PSC=0/2 ..... 232
图 16-5. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 ..... 232
图 16-6. 向下计数时序图，PSC=0/2 ..... 233
图 16-7. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值 ..... 234
图 16-8. 中央计数模式计数器时序图 ..... 235
图 16-9. 中央计数模式下计数器重复时序图 ..... 236
图 16-10. 在向上计数模式下计数器重复时序图 ..... 236
图 16-11. 在向下计数模式下计数器重复时序图 ..... 237
图 16-12. 通道输入捕获原理 ..... 238
图 16-13. 三种输出比较模式 ..... 240
图 16-14. EAPWM 时序图 ..... 241
图 16-15. CAPWM 时序图 ..... 241
图 16-16. 带死区时间的互补输出 ..... 244
图 16-17. 通道响应中止输入(高电平有效)时，输出信号的行为 ..... 245
图 16-18. 在编码器模式 2 且 CI0FE0 极性不反相时计数器行为 ..... 246
图 16-19. 在编码器模式 2 且 CI0FE0 极性反相时计数器行为 ..... 246
图 16-20. 霍尔传感器用在 BLDC 电机控制 ..... 247
图 16-21. 两个定时器之间的霍尔传感器时序图 ..... 248
图 16-22. 复位模式下的控制电路 ..... 249
图 16-23. 暂停模式下的控制电路 ..... 249
图 16-24. 事件模式下的控制电路 ..... 250
图 16-25. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99 ..... 250
图 16-26. 定时器 0 主/从模式的例子 ..... 251
图 16-27. 用定时器 2 的使能信号触发定时器 0 ..... 252
图 16-28. 用定时器 2 的 CI0 输入来触发定时器 0 和定时器 2 ..... 253
图 16-29. 通用定时器 L0 结构框图 ..... 280
图 16-30. 内部时钟分频为 1 时，计数器的时序图 ..... 281
图 16-31. 当 PSC 数值从 0 变到 2 时，计数器的时序图 ..... 282
图 16-32. 向上计数时序图，PSC=0/2 ..... 283
图 16-33. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 ..... 283
图 16-34. 向下计数时序图，PSC=0/2 ..... 284
图 16-35. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值 ..... 285
图 16-36. 中央计数模式计数器时序图 ..... 286

图 16-37. 通道输入捕获原理 ..... 287
图 16-38. 三种输出比较模式 ..... 289
图 16-39. EAPWM 时序图 ..... 290
图 16-40. CAPWM 时序图 ..... 290
图 16-41. 复位模式下的控制电路 ..... 292
图 16-42. 暂停模式下的控制电路 ..... 292
图 16-43. 事件模式下的控制电路 ..... 293
图 16-44. 通用定时器 L1 结构框图 ..... 315
图 16-45. 内部时钟分频为 1 时，计数器的时序图 ..... 316
图 16-46. 当 PSC 数值从 0 变到 2 时，计数器的时序图 ..... 317
图 16-47. 向上计数时序图，PSC=0/2 ..... 318
图 16-48. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 ..... 318
图 16-49. 输入捕获原理 ..... 319
图 16-50. 三种输出比较模式 ..... 321
图 16-51. EAPWM 时序图 ..... 322
图 16-52. CAPWM 时序图 ..... 322
图 16-53. 复位模式下的控制电路 ..... 323
图 16-54. 暂停模式下的控制电路 ..... 324
图 16-55. 事件模式下的控制电路 ..... 324
图 16-56. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99 ..... 325
图 16-57. 通用定时器 L2 结构框图 ..... 338
图 16-58. 内部时钟分频为 1 时，计数器的时序图 ..... 339
图 16-59. 当 PSC 数值从 0 变到 2 时，计数器的时序图 ..... 340
图 16-60. 向上计数时序图，PSC=0/2 ..... 341
图 16-61. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 ..... 341
图 16-62. 通道输入捕获原理 ..... 342
图 16-63. 三种输出比较模式 ..... 344
图 16-64. 基本定时器结构框图 ..... 355
图 16-65. 内部时钟分频为 1 时，计数器的时序图 ..... 356
图 16-66. 当 PSC 数值从 0 变到 2 时，计数器的时序图 ..... 356
图 16-67. 向上计数时序图，PSC=0/2 ..... 357
图 16-68. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 ..... 358
图 17-1. USART 模块内部框图 ..... 366
图 17-2. USART 字符帧(8 数据位和 1 停止位) ..... 366
图 17-3. USART 发送步骤 ..... 368
图 17-4. 过采样方式接收一个数据位 ..... 369
图 17-5. 采用 DMA 方式实现 USART 数据发送配置步骤 ..... 370
图 17-6. 采用 DMA 方式实现 USART 数据接收配置步骤 ..... 370
图 17-7. 两个 USART 之间的硬件流控制 ..... 371
图 17-8. 硬件流控制 ..... 371
图 17-9. 空闲状态下检测断开帧 ..... 372
图 17-10. 数据传输过程中检测断开帧 ..... 373
图 17-11. 同步模式下的 USART 示例 ..... 373
图 17-12. 8-bit 格式的 USART 同步通信波形(CLEN=1) ..... 373

图 17-13. IrDA SIR ENDEC 模块 ..... 374
图 17-14. IrDA 数据调制 ..... 374
图 17-15. ISO7816-3 数据帧格式 ..... 375
图 17-16. USART 中断映射框图 ..... 377
图 17-17. I2C 模块框图 ..... 392
图 17-18. 数据有效性 ..... 393
图 17-19. 起始和停止信号 ..... 393
图 17-20. 时钟同步 ..... 394
图 17-21. SDA 线仲裁 ..... 394
图 17-22. 7 位地址的 I2C 通讯流程 ..... 394
图 17-23. 10 位地址的 I2C 通讯流程(主机发送) ..... 395
图 17-24. 10 位地址的 I2C 通讯流程(主机接收) ..... 395
图 17-25. 从机发送模式(10 位地址模式) ..... 396
图 17-26. 从机接收模式(10 位地址模式) ..... 397
图 17-27. 主机发送模式(10 位地址模式) ..... 399
图 17-28. 主机接收使用方案 A 模式(10 位地址模式) ..... 401
图 17-29. 主机接收使用方案 B 模式(10 位地址模式) ..... 403
图 19-1. SPI 结构框图 ..... 419
图 19-2. 常规模式下的 SPI 时序图 ..... 420
图 19-3. SPI 四线模式下的 SPI 时序图(CKPL=1, CKPH=1, LF=0) ..... 421
图 19-4. 典型的全双工模式连接 ..... 423
图 19-5. 典型的单工模式连接(主机：接收，从机：发送) ..... 424
图 19-6. 典型的单工模式连接(主机：只发送，从机：接收) ..... 424
图 19-7. 典型的双向线连接 ..... 424
图 19-8. 主机 TI 模式在不连续发送时的时序图 ..... 426
图 19-9. 主机 TI 模式在连续发送时的时序图 ..... 426
图 19-10. 从机 TI 模式时序图 ..... 426
图 19-11. NSS 脉冲模式时序图(主机连续发送) ..... 427
图 19-12. SPI 四线模式四线写操作时序图 ..... 428
图 19-13. SPI 四线模式四线读操作时序图 ..... 429
图 19-14. I2S 结构框图 ..... 432
图 19-15. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=0) ..... 433
图 19-16. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=1) ..... 433
图 19-17. I2S 飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=0) ..... 433
图 19-18. I2S 飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=1) ..... 434
图 19-19. I2S 飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=0) ..... 434
图 19-20. I2S 飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=1) ..... 434
图 19-21. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=0) ..... 434
图 19-22. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=1) ..... 434
图 19-23. MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=0) ..... 435
图 19-24. MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=1) ..... 435
图 19-25. MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=0) ..... 435
图 19-26. MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=1) ..... 435
图 19-27. MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0) ..... 435

图 19-28. MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1) ..... 436
图 19-29. MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0) ..... 436
图 19-30. MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1) ..... 436
图 19-31. LSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0) ..... 436
图 19-32. LSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1) ..... 436
图 19-33. LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0) ..... 437
图 19-34. LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1) ..... 437
图 19-35. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=0) ..... 437
图 19-36. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=1) ..... 437
图 19-37. PCM 标准短帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=0) ..... 437
图 19-38. PCM 标准短帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=1) ..... 438
图 19-39. PCM 标准短帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=0) ..... 438
图 19-40. PCM 标准短帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=1) ..... 438
图 19-41. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=0) ..... 438
图 19-42. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=1) ..... 438
图 19-43. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=0) ..... 438
图 19-44. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=1) ..... 439
图 19-45. PCM 标准长帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=0) ..... 439
图 19-46. PCM 标准长帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=1) ..... 439
图 19-47. PCM 标准长帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=0) ..... 439
图 19-48. PCM 标准长帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=1) ..... 439
图 19-49. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=0) ..... 439
图 19-50. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=1) ..... 440
图 19-51. I2S 时钟生成结构框图 ..... 440
图 19-52. I2S 初始化流程 ..... 442
图 19-53. I2S 主机接收禁能流程 ..... 444
图 20-1. EXMC 结构框图 ..... 458
图 20-2. EXMC Bank 划分 ..... 459
图 20-3. 模式 1 读访问 ..... 462
图 20-4. 模式 1 写访问 ..... 463
图 20-5. 模式 A 读访问 ..... 464
图 20-6. 模式 A 写访问 ..... 464
图 20-7. 模式 2/B 读访问 ..... 465
图 20-8. 模式 2 写访问 ..... 466
图 20-9. 模式 B 写访问 ..... 466
图 20-10. 模式 C 读访问 ..... 467
图 20-11. 模式 C 写访问 ..... 468
图 20-12. 模式 D 读访问 ..... 469
图 20-13. 模式 D 写访问 ..... 469
图 20-14. 复用模式读访问 ..... 471
图 20-15. 复用模式写访问 ..... 471
图 20-16. 异步等待有效时的读时序 ..... 473
图 20-17. 异步等待有效时的写时序 ..... 473
图 20-18. 同步复用突发传输读时序 ..... 475

图 20-19. 同步复用突发传输写时序 ..... 476
图 21-1. USBFS 结构框图 ..... 484
图 21-2. 在主机或设备模式下连接示意图 ..... 485
图 21-3. OTG 模式下连接示意图 ..... 486
图 21-4. 主机端口状态转移图 ..... 486
图 21-5. 主机模式 FIFO 空间 ..... 490
图 21-6. 主机模式 FIFO 访问寄存器映射表 ..... 491
图 21-7. 设备模式 FIFO 空间 ..... 491
图 21-8. 设备模式 FIFO 访问寄存器映射表 ..... 492

# 表索引

表 1-1. AHB 互联矩阵的互联关系列表 ..... 24
表 1-2. GD32E11x 系列器件的存储器映射表 ..... 27
表 1-3. 引导模式 ..... 31
表 2-1. GD32E11x 闪存基地址和构成 ..... 35
表 2-2. WSCNT 与 AHB 时钟频率对应关系 ..... 36
表 2-3. 选项字节 ..... 43
表 3-1. 节电模式总结 ..... 57
表 5-1. 时钟输出 0 的时钟源选择 ..... 71
表 5-2. 深度睡眠模式下 1.2V 域电压选择 ..... 71
表 7-1. Cortex®-M4 中的 NVIC 异常类型 ..... 110
表 7-2. 中断向量表 ..... 111
表 7-3. EXTI 触发源 ..... 114
表 8-1. GPIO 配置表 ..... 118
表 8-2. 调试接口信号 ..... 123
表 8-3. 调试端口和引脚可用性 ..... 123
表 8-4. ADC0/1 常规转换外部触发备用功能重映射(1) ..... 123
表 8-5. TIMER 备用功能重映射 ..... 124
表 8-6. TMER4 备用功能重映射[1] ..... 125
表 8-7. USART0/1 备用功能重映射 ..... 125
表 8-8. I2C0 备用功能重映射 ..... 126
表 8-9. SPI0/SPI2/I2S 备用功能重映射 ..... 126
表 8-10. CTC 备用功能重映射 ..... 127
表 8-11. OSC32 引脚配置 ..... 127
表 8-12. OSC 引脚配置 ..... 128
表 10-1. DMA 传输操作 ..... 151
表 10-2. 中断事件 ..... 154
表 10-3. DMA0 各通道请求表 ..... 156
表 10-4. DMA1 各通道请求表 ..... 157
表 12-1. ADC 内部输入信号 ..... 172
表 12-2. ADC 输入引脚定义 ..... 172
表 12-3. ADC0 和 ADC1 的外部触发源 ..... 178
表 12-4. 不同分辨率对应的 $t_{CONV}$ 时间 ..... 179
表 12-5. 不同 N 和 M 组合的最大输出值(灰色值表示截断) ..... 180
表 12-6. ADC 同步模式表 ..... 181
表 13-1. DAC 引脚 ..... 197
表 13-2. DAC 触发与输出 ..... 197
表 13-3. DAC 外部触发 ..... 198
表 14-1. 独立看门狗定时器在 40kHz(IRC40K)时的最小/最大超时周期 ..... 211
表 14-2. 在 60MHz($f_{PCLK1}$)时的最大/最小超时值 ..... 217
表 16-1. 定时器(TIMERx)分为五种类型 ..... 227
表 16-2. 由参数控制的互补输出表 ..... 242

表 16-3. 不同编码器模式下的计数方向 ..... 245
表 16-4. 从模式例子列表 ..... 248
表 16-5. 从模式列表和举例 ..... 291
表 16-6. 从机模式列表和举例 ..... 323
表 17-1. USART 重要引脚描述 ..... 365
表 17-2. 停止位配置 ..... 366
表 17-3. USART 中断请求 ..... 377
表 17-4. I2C 总线术语说明(参考飞利浦 I2C 规范) ..... 392
表 17-5. 事件状态标志位 ..... 406
表 17-6. 错误标志位 ..... 406
表 19-1. SPI 信号描述 ..... 419
表 19-2. SPI 四线信号描述 ..... 420
表 19-3. 从机模式 NSS 功能 ..... 421
表 19-4. 主机模式 NSS 功能 ..... 422
表 19-5. SPI 运行模式 ..... 422
表 19-6. SPI 中断请求 ..... 432
表 19-7. I2S 比特率计算公式 ..... 440
表 19-8. 音频采样频率计算公式 ..... 441
表 19-9. 各种运行模式下 I2S 接口信号的方向 ..... 441
表 19-10. I2S 中断 ..... 446
表 20-1. NOR Flash 接口信号描述 ..... 459
表 20-2. PSRAM 非复用接口信号描述 ..... 460
表 20-3. EXMC 的 Bank0 支持的所有传输 ..... 460
表 20-4. NOR/PSRAM 控制时序参数 ..... 461
表 20-5. EXMC 时序模型 ..... 461
表 20-6. 模式 1 相关寄存器配置 ..... 463
表 20-7. 模式 A 相关寄存器配置 ..... 464
表 20-8. 模式 2/B 相关寄存器配置 ..... 466
表 20-9. 模式 C 相关寄存器配置 ..... 468
表 20-10. 模式 D 相关寄存器配置 ..... 470
表 20-11. 复用模式相关寄存器配置 ..... 471
表 20-12. 同步复用模式读时序配置 ..... 475
表 20-13. 同步复用模式写时序配置 ..... 476
表 21-1. USBFS 信号线描述 ..... 484
表 21-2. USBFS 全局中断 ..... 495
表 22-1. 寄存器功能位访问属性 ..... 553
表 22-2. 术语 ..... 553
表 23-1. 版本历史 ..... 554

# 1. 系统及存储器架构

GD32E11x系列器件是基于ARM® Cortex™-M4处理器的32位通用微控制器。ARM® Cortex®-M4处理器包括三条AHB总线分别称为I-CODE总线、D-Code总线和系统总线。Cortex®-M4处理器的所有存储访问，根据不同的目的和目标存储空间，都会在这三条总线上执行。存储器的组织采用了哈佛结构，预先定义的存储器映射和高达4 GB的存储空间，充分保证了系统的灵活性和可扩展性。

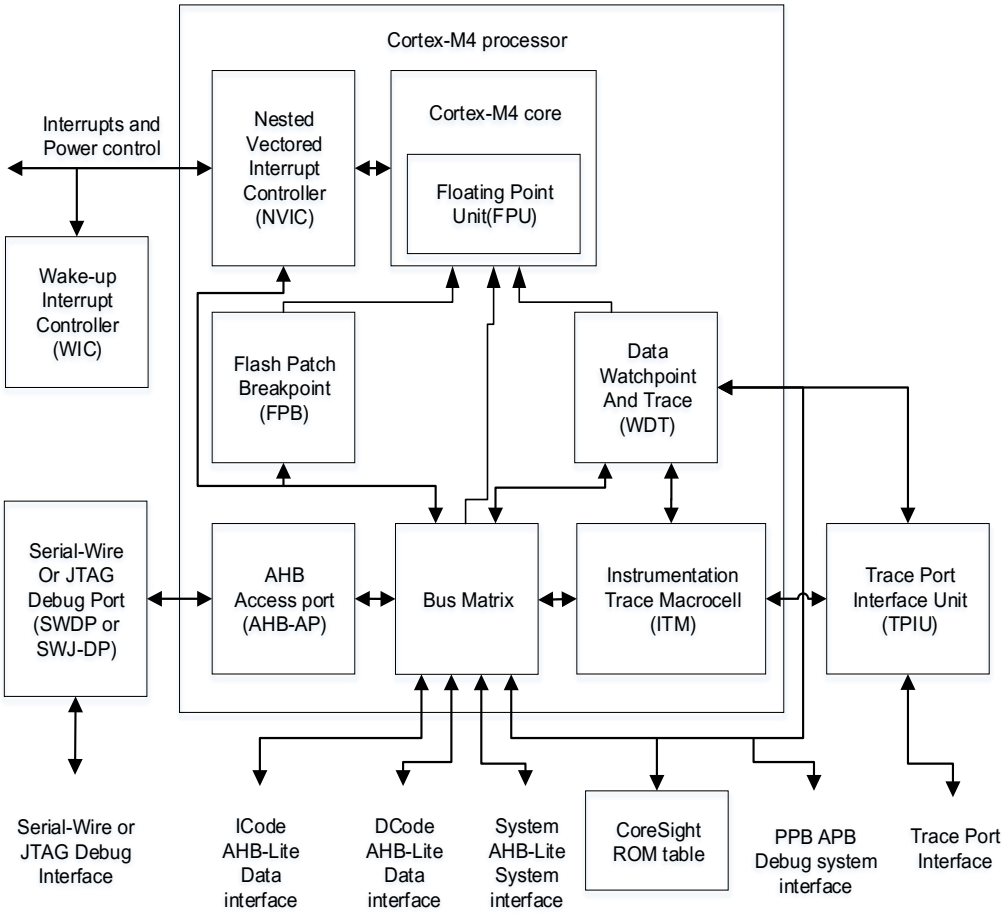
## 1.1. ARM Cortex-M4 处理器

Cortex®-M4处理器是一个具有浮点运算功能、低中断延迟时间和低成本调试特性的32位处理器。高集成度和增强的特性使Cortex®-M4处理器适合于那些需要高性能和低功耗微控制器的市场领域。Cortex®-M4处理器基于ARMv7架构，并且支持一个强大且可扩展的指令集，包括通用数据处理I/O控制任务、增强的数据处理位域操作、DSP(数字信号处理)和浮点运算指令。下面列出由Cortex®-M4提供的一些系统外设：

- 内部总线矩阵，用于实现I-Code总线、D-Code总线、系统总线、专用总线(PPB)以及调试专用总线(AHB-AP)的互联；
- 嵌套式向量型中断控制器（NVIC)；
- 闪存地址重载及断点单元（FPB)；
- 数据观测点及跟踪单元（DWT)；
- 指令跟踪宏单元（ITM)；
- 串行线和JTAG调试接口（SWJ-DP)；
- 跟踪端口接口单元（TPIU)；
- 浮点运算单元（FPU)。

*图1-1. Cortex®-M4结构框图*显示了Cortex®-M4处理器结构框图。欲了解更多信息，请参阅ARM® Cortex®-M4技术参考手册。

图 1-1. Cortex®-M4 结构框图




## 1.2. 系统架构

GD32E11x系列器件采用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。AHB互联矩阵的互联关系接下来将进行说明。在*表1-1. AHB互联矩阵的互联关系列表*中，"1"表示相应的主机可以通过AHB互联矩阵访问对应的从机，空白的单元格表示相应的主机不可以通过AHB互联矩阵访问对应的从机。

表 1-1. AHB 互联矩阵的互联关系列表

| | IBUS | DBUS | SBUS | DMA0 | DMA1 |
|---|---|---|---|---|---|
| FMC-I | 1 | | | | |
| FMC-D | | 1 | | 1 | 1 |
| SRAM | 1 | 1 | 1 | 1 | 1 |
| EXMC | 1 | 1 | 1 | 1 | 1 |
| AHB | | | 1 | 1 | 1 |

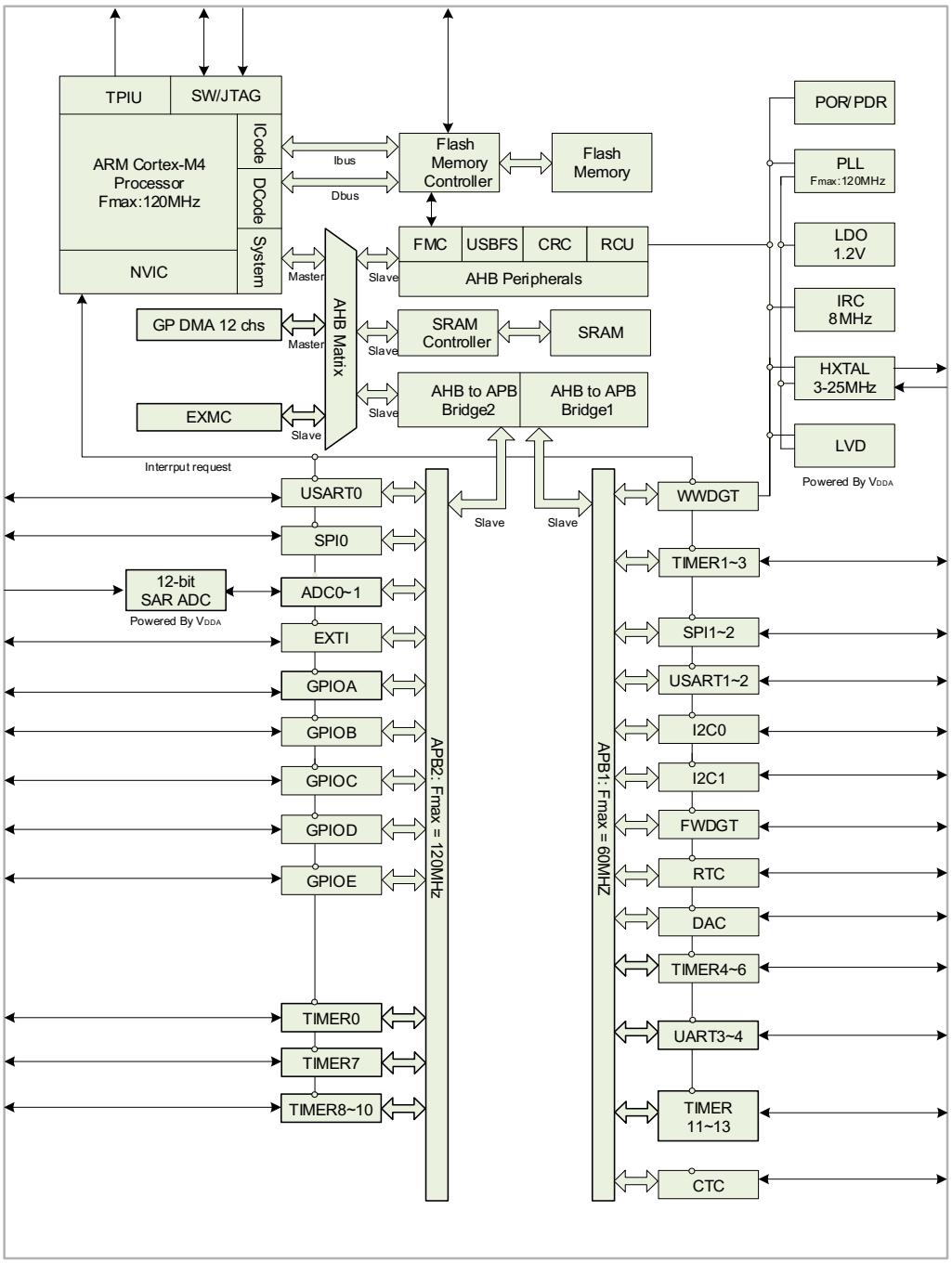
| | IBUS | DBUS | SBUS | DMA0 | DMA1 |
|---|---|---|---|---|---|
| **APB1** | | | 1 | 1 | 1 |
| **APB2** | | | 1 | 1 | 1 |

如上表所示，AHB互联矩阵共连接11个主机，分别为：IBUS、DBUS、SBUS、DMA0、DMA1。IBUS是Cortex®-M4内核的指令总线，用于从代码区域(0x0000 0000-0x1FFF FFFF)中取指令和向量。DBUS是Cortex®-M4内核的数据总线，用于加载和存储数据，以及代码区域的调试访问。同样，SBUS是Cortex®-M4内核的系统总线，用于指令和向量获取、数据加载和存储以及系统区域的调试访问。系统区域包括内部SRAM区域和外设区域。DMA0和DMA1分别是DMA0和DMA1的存储器总线。

AHB互联矩阵也连接了12个从机，分别为：FMC-I、FMC-D、SRAM、EXMC、AHB、 APB1和APB2。FMC-I是闪存存储器控制器的指令总线，而FMC-D是闪存存储器的数据总线。SRAM是片上静态随机存取存储器。EXMC是外部存储器控制器。AHB是连接所有AHB从机的AHB总线，APB1和APB2是连接所有APB从机的两条APB总线。APB1速度限制为60MHz，APB2可以全速运行(最高可到120MHz)。

GD32E11x系列器件的系统架构如*图1-2. GD32E11x器件的系统架构示意图*所示。

**图 1-2. GD32E11x 器件的系统架构示意图**




## 1.3. 存储器映射

ARM® Cortex®-M4处理器采用哈佛结构，可以使用相互独立的总线来读取指令和加载/存储数据。指令代码和数据都位于相同的存储器地址空间，但在不同的地址范围。程序存储器，数据存储器，寄存器和I/O端口都在同一个线性的4 GB的地址空间之内。这是Cortex®-M4的最大地址范围，因为它的地址总线宽度是32位。此外，为了降低不同客户在相同应用时的软件复杂度，存储映射是按Cortex®-M4处理器提供的规则预先定义的。在存储器映射表中，一部分地址空间

由ARM® Cortex®-M4的系统外设所占用，且不可更改。此外，其余部分地址空间可由芯片供应商定义使用。*表1-2. GD32E11x系列器件的存储器映射表*显示了GD32E11x系列器件的存储器映射，包括代码、SRAM、外设和其他预先定义的区域。几乎每个外设都分配了1KB的地址空间，这样可以简化每个外设的地址译码。

**表 1-2. GD32E11x 系列器件的存储器映射表**

| 预先定义的地址空间 | 总线 | 地址范围 | 外设 |
|---|---|---|---|
| 外部设备 | AHB3 | 0xA000 0000 - 0xA000 0FFF | EXMC - SWREG |
| 外部 RAM | | 0x9000 0000 - 0x9FFF FFFF | 保留 |
| | | 0x7000 0000 - 0x8FFF FFFF | 保留 |
| | | 0x6000 0000 - 0x63FF FFFF | EXMC - NOR/PSRAM/SRAM |
| Peripheral | AHB1 | 0x5000 0000 - 0x5003 FFFF | USBFS |
| | | 0x4008 0000 - 0x4FFF FFFF | 保留 |
| | | 0x4004 0000 - 0x4007 FFFF | 保留 |
| | | 0x4002 BC00 - 0x4003 FFFF | 保留 |
| | | 0x4002 B000 - 0x4002 BBFF | 保留 |
| | | 0x4002 A000 - 0x4002 AFFF | 保留 |
| | | 0x4002 8000 - 0x4002 9FFF | 保留 |
| | | 0x4002 6800 - 0x4002 7FFF | 保留 |
| | | 0x4002 6400 - 0x4002 67FF | 保留 |
| | | 0x4002 6000 - 0x4002 63FF | 保留 |
| | | 0x4002 5000 - 0x4002 5FFF | 保留 |
| | | 0x4002 4000 - 0x4002 4FFF | 保留 |
| | | 0x4002 3C00 - 0x4002 3FFF | 保留 |
| | | 0x4002 3800 - 0x4002 3BFF | 保留 |
| | | 0x4002 3400 - 0x4002 37FF | 保留 |
| | | 0x4002 3000 - 0x4002 33FF | CRC |
| | | 0x4002 2C00 - 0x4002 2FFF | 保留 |
| | | 0x4002 2800 - 0x4002 2BFF | 保留 |
| | | 0x4002 2400 - 0x4002 27FF | 保留 |
| | | 0x4002 2000 - 0x4002 23FF | FMC |
| | | 0x4002 1C00 - 0x4002 1FFF | 保留 |
| | | 0x4002 1800 - 0x4002 1BFF | 保留 |
| | | 0x4002 1400 - 0x4002 17FF | 保留 |
| | | 0x4002 1000 - 0x4002 13FF | RCU |
| | | 0x4002 0C00 - 0x4002 0FFF | 保留 |
| | | 0x4002 0800 - 0x4002 0BFF | 保留 |
| | | 0x4002 0400 - 0x4002 07FF | DMA1 |
| | | 0x4002 0000 - 0x4002 03FF | DMA0 |
| | | 0x4001 8400 - 0x4001 FFFF | 保留 |
| | | 0x4001 8000 - 0x4001 83FF | 保留 |
| | APB2 | 0x4001 7C00 - 0x4001 7FFF | 保留 |

| 预先定义的地址空间 | 总线 | 地址范围 | 外设 |
|---|---|---|---|
| | | 0x4001 7800 - 0x4001 7BFF | 保留 |
| | | 0x4001 7400 - 0x4001 77FF | 保留 |
| | | 0x4001 7000 - 0x4001 73FF | 保留 |
| | | 0x4001 6C00 - 0x4001 6FFF | 保留 |
| | | 0x4001 6800 - 0x4001 6BFF | 保留 |
| | | 0x4001 5C00 - 0x4001 67FF | 保留 |
| | | 0x4001 5800 - 0x4001 5BFF | 保留 |
| | | 0x4001 5400 - 0x4001 57FF | TIMER10 |
| | | 0x4001 5000 - 0x4001 53FF | TIMER9 |
| | | 0x4001 4C00 - 0x4001 4FFF | TIMER8 |
| | | 0x4001 4800 - 0x4001 4BFF | 保留 |
| | | 0x4001 4400 - 0x4001 47FF | 保留 |
| | | 0x4001 4000 - 0x4001 43FF | 保留 |
| | | 0x4001 3C00 - 0x4001 3FFF | 保留 |
| | | 0x4001 3800 - 0x4001 3BFF | USART0 |
| | | 0x4001 3400 - 0x4001 37FF | TIMER7 |
| | | 0x4001 3000 - 0x4001 33FF | SPI0 |
| | | 0x4001 2C00 - 0x4001 2FFF | TIMER0 |
| | | 0x4001 2800 - 0x4001 2BFF | ADC1 |
| | | 0x4001 2400 - 0x4001 27FF | ADC0 |
| | | 0x4001 2000 - 0x4001 23FF | 保留 |
| | | 0x4001 1C00 - 0x4001 1FFF | 保留 |
| | | 0x4001 1800 - 0x4001 1BFF | GPIOE |
| | | 0x4001 1400 - 0x4001 17FF | GPIOD |
| | | 0x4001 1000 - 0x4001 13FF | GPIOC |
| | | 0x4001 0C00 - 0x4001 0FFF | GPIOB |
| | | 0x4001 0800 - 0x4001 0BFF | GPIOA |
| | | 0x4001 0400 - 0x4001 07FF | EXTI |
| | | 0x4001 0000 - 0x4001 03FF | AFIO |
| | APB1 | 0x4000 CC00 - 0x4000 FFFF | 保留 |
| | | 0x4000 C800 - 0x4000 CBFF | CTC |
| | | 0x4000 C400 - 0x4000 C7FF | 保留 |
| | | 0x4000 C000 - 0x4000 C3FF | 保留 |
| | | 0x4000 8000 - 0x4000 BFFF | 保留 |
| | | 0x4000 7C00 - 0x4000 7FFF | 保留 |
| | | 0x4000 7800 - 0x4000 7BFF | 保留 |
| | | 0x4000 7400 - 0x4000 77FF | DAC |
| | | 0x4000 7000 - 0x4000 73FF | PMU |
| | | 0x4000 6C00 - 0x4000 6FFF | BKP |
| | | 0x4000 6800 - 0x4000 6BFF | 保留 |

| 预先定义的地址空间 | 总线 | 地址范围 | 外设 |
|---|---|---|---|
| | | 0x4000 6400 - 0x4000 67FF | 保留 |
| | | 0x4000 6000 - 0x4000 63FF | 保留 |
| | | 0x4000 5C00 - 0x4000 5FFF | 保留 |
| | | 0x4000 5800 - 0x4000 5BFF | I2C1 |
| | | 0x4000 5400 - 0x4000 57FF | I2C0 |
| | | 0x4000 5000 - 0x4000 53FF | UART4 |
| | | 0x4000 4C00 - 0x4000 4FFF | UART3 |
| | | 0x4000 4800 - 0x4000 4BFF | USART2 |
| | | 0x4000 4400 - 0x4000 47FF | USART1 |
| | | 0x4000 4000 - 0x4000 43FF | 保留 |
| | | 0x4000 3C00 - 0x4000 3FFF | SPI2/I2S2 |
| | | 0x4000 3800 - 0x4000 3BFF | SPI1/I2S1 |
| | | 0x4000 3400 - 0x4000 37FF | 保留 |
| | | 0x4000 3000 - 0x4000 33FF | FWDGT |
| | | 0x4000 2C00 - 0x4000 2FFF | WWDGT |
| | | 0x4000 2800 - 0x4000 2BFF | RTC |
| | | 0x4000 2400 - 0x4000 27FF | 保留 |
| | | 0x4000 2000 - 0x4000 23FF | TIMER13 |
| | | 0x4000 1C00 - 0x4000 1FFF | TIMER12 |
| | | 0x4000 1800 - 0x4000 1BFF | TIMER11 |
| | | 0x4000 1400 - 0x4000 17FF | TIMER6 |
| | | 0x4000 1000 - 0x4000 13FF | TIMER5 |
| | | 0x4000 0C00 - 0x4000 0FFF | TIMER4 |
| | | 0x4000 0800 - 0x4000 0BFF | TIMER3 |
| | | 0x4000 0400 - 0x4000 07FF | TIMER2 |
| | | 0x4000 0000 - 0x4000 03FF | TIMER1 |
| SRAM | AHB | 0x2007 0000 - 0x3FFF FFFF | 保留 |
| | | 0x2006 0000 - 0x2006 FFFF | 保留 |
| | | 0x2003 0000 - 0x2005 FFFF | 保留 |
| | | 0x2002 0000 - 0x2002 FFFF | 保留 |
| | | 0x2001 C000 - 0x2001 FFFF | 保留 |
| | | 0x2001 8000 - 0x2001 BFFF | 保留 |
| | | 0x2000 8000 - 0x2001 7FFF | 保留 |
| | | 0x2000 0000 - 0x2000 7FFF | SRAM |
| Code | AHB | 0x1FFF F810 - 0x1FFF FFFF | 保留 |
| | | 0x1FFF F800 - 0x1FFF F80F | Option Bytes |
| | | 0x1FFF F000 - 0x1FFF F7FF | Boot loader |
| | | 0x1FFF C010 - 0x1FFF EFFF | |
| | | 0x1FFF C000 - 0x1FFF C00F | |
| | | 0x1FFF B000 - 0x1FFF BFFF | |

| 预先定义的地址空间 | 总线 | 地址范围 | 外设 |
|---|---|---|---|
| | | 0x1FFF 7A10 - 0x1FFF AFFF | 保留 |
| | | 0x1FFF 7800 - 0x1FFF 7A0F | 保留 |
| | | 0x1FFF 0000 - 0x1FFF 77FF | 保留 |
| | | 0x1FFE C010 - 0x1FFE FFFF | 保留 |
| | | 0x1FFE C000 - 0x1FFE C00F | 保留 |
| | | 0x1001 0000 - 0x1FFE BFFF | 保留 |
| | | 0x1000 0000 - 0x1000 FFFF | 保留 |
| | | 0x083C 0000 - 0x0FFF FFFF | 保留 |
| | | 0x0830 0000 - 0x083B FFFF | 保留 |
| | | 0x0810 0000 - 0x082F FFFF | 保留 |
| | | 0x0802 0000 - 0x080F FFFF | 保留 |
| | | 0x0800 0000 - 0x0801 FFFF | Main Flash |
| | | 0x0030 0000 - 0x07FF FFFF | 保留 |
| | | 0x0010 0000 - 0x002F FFFF | 别名为 Main Flash 或 Boot loader |
| | | 0x0002 0000 - 0x000F FFFF | |
| | | 0x0000 0000 - 0x0001 FFFF | |

### 1.3.1. 位带操作

为了减少"读-改-写"操作的次数，Cortex®-M4处理器提供了一个可以执行单原子比特操作的位带功能。存储器映射包含了两个支持位带操作的区域。一个是SRAM，另一个是片内外设区。这两个区域中的地址除了普通应用外，还有自己的"位带别名区"。位带别名区把每个比特扩展成一个32位的字。

下面的公式表明了位带别名区中的每个字如何对应位带区的相应比特或目标比特。

$$bit_word_addr = bit_band_base + (byte_offset \times 32) + (bit_number \times 4) \quad (1\text{-}1)$$

其中：

- bit_word_addr指的是位带区目标比特对应在位带别名区的地址；
- bit_band_base指的是位带别名区的起始地址；
- byte_offset指的是位带区目标比特所在的字节的字节地址偏移量；
- bit_number指的是目标比特在对应字节中的位置(0-7)。

例如，要想访问0x2000 0200地址的第7位，可访问的位带别名区地址是0x2000 401C：

$$bit_word_addr = 0x2200\ 0000 + (0x200 * 32) + (7 * 4) = 0x2200\ 401C \quad (1\text{-}2)$$

如果对0x2200 401C进行写操作，那么0x2000 0200的第7位将会相应变化；如果对0x2200 401C进行读操作，那么视0x2000 0200的第7位状态而返回0x01或0x00。

### 1.3.2. 片上 SRAM 存储器

GD32E11x系列微控制器含有高达32KB片上SRAM(起始地址为0x2000 0000)。支持字节、半字(16比特)和整字(32比特)访问。

### 1.3.3. 片上 FLASH 存储器概述

GD32E11x系列微控制器可以提供高密度片上FLASH存储器，按以下分类进行组织：

- 高达128KB主FLASH存储器；
- 高达18KB引导装载程序(boot loader)信息块存储器；
- 器件配置的选项字节。

更多详细说明请参考*闪存控制器(FMC)*章节。

## 1.4. 引导配置

GD32E11x系列微控制器提供了三种引导源，可以通过BOOT0和BOOT1引脚来进行选择，详细说明见*表1-3. 引导模式*。该两个引脚的电平状态会在复位后的第四个CK_SYS(系统时钟)的上升沿进行锁存。用户可自行选择所需要的引导源，通过设置上电复位和系统复位后的BOOT0和BOOT1的引脚电平。一旦这两个引脚电平被采样，它们可以被释放并用于其他用途。

**表 1-3. 引导模式**

| 引导源选择 | 启动模式选择引脚 | |
|---|---|---|
| | Boot1 | Boot0 |
| 主FLASH存储器 | x | 0 |
| 引导装载程序 | 0 | 1 |
| 片上SRAM | 1 | 1 |

上电序列或系统复位后，ARM®Cortex®-M4处理器先从0x0000 0000地址获取栈顶值，再从0x0000 0004地址获得引导代码的基地址，然后从引导代码的基地址开始执行程序。

根据所选择的引导源，主FLASH存储器(开始于0x0800 0000的原始存储空间)或系统存储器(开始于0x1FFF F000的原始存储空间)被映射到引导存储空间(起始于0x0000 0000)。片上SRAM存储空间的起始地址是0x2000 0000，当它被选择为引导源时，在应用初始化代码中，你必须使用NVIC异常表和偏移寄存器来将向量表重定向到SRAM中。

嵌入式的Bootloader存放在系统存储空间，用于对FLASH存储器进行重新编程。在GD32E11x设备中，Bootloader可以通过USART0接口和外界交互。

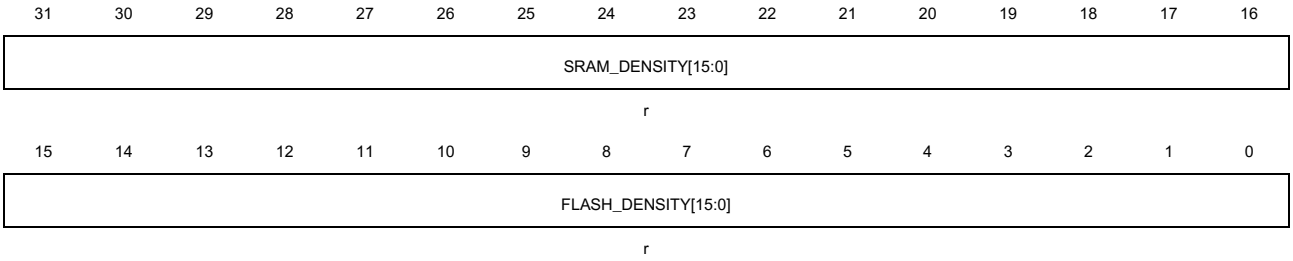
**注意：**为确保系统的可靠性和稳定性，在硬件设计阶段，应查阅对应型号的数据手册的Boot mode章节，确认Bootloader所使用的引脚上避免连接可能产生干扰信号的外设（如传感器输出、看门狗芯片、按键等），以免影响Bootloader的正常功能，和对所连接的外设产生异常输出。

## 1.5. 设备电子签名

设备的电子签名中包含的存储容量信息和96位的唯一设备ID，它位于Flash存储器的信息块中。96位设备唯一ID对任何设备来说都是独一无二的，可以用作序列号，或秘钥的一部分，等等。

### 1.5.1. 存储器容量信息

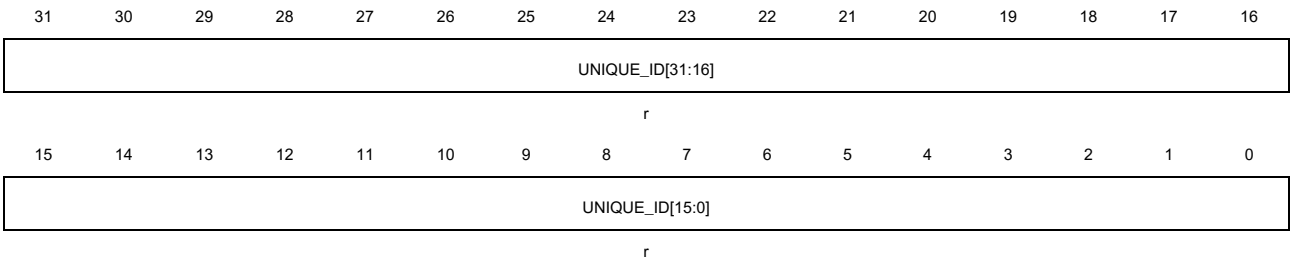
基地址：0x1FFF F7E0
其值出厂已设置，不可由用户更改。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | SRAM_DENSITY[15:0] | SRAM容量<br>该值表示器件的片上SRAM容量，以Kbytes为单位。<br>例如：0x0008表示8Kbytes。 |
| 15:0 | FLASH_DENSITY[15:0] | FLASH存储器容量<br>该值表示器件的FLASH存储器容量，以Kbytes为单位。<br>例如：0x0020表示32Kbytes。 |

### 1.5.2. 设备唯一 ID(96 位)

基地址：0x1FFF F7E8
其值出厂已设置，不可由用户更改。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | UNIQUE_ID[31:0] | 设备唯一ID |

基地址：0x1FFF F7EC
其值出厂已设置，不可由用户更改。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UNIQUE_ID[63:48] | | | | | | | | | | | | | | | |
| r | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UNIQUE_ID[47:32] | | | | | | | | | | | | | | | |
| r | | | | | | | | | | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | UNIQUE_ID[63:32] | 设备唯一ID |

基地址：0x1FFF F7F0
其值出厂已设置，不可由用户更改。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UNIQUE_ID[95:80] | | | | | | | | | | | | | | | |
| r | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UNIQUE_ID[79:64] | | | | | | | | | | | | | | | |
| r | | | | | | | | | | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | UNIQUE_ID[95:64] | 设备唯一ID |

## 1.6. 系统配置寄存器

基地址：0x4002 103C
复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 7 | CEE | 代码执行效率<br>0：默认的代码执行效率<br>1：代码执行效率降低 |

**注意：**只有bit[7]可以被读-改-写，其他的位都不允许。

# 2. 闪存控制器(FMC)

## 2.1. 简介

闪存控制器(FMC)，提供了片上闪存需要的所有功能。在闪存的前128K字节空间内，CPU执行指令需要少量等待时间。FMC也提供了页擦除，整片擦除，以及编程操作。

## 2.2. 主要特征

- 高达128KB的片上闪存可用于存储指令或数据；
- 在闪存的前128K字节空间内，CPU执行指令需要0~3个等待时间；
- 闪存支持64位宽数据读取；
- 预取缓冲区以加速读操作；
- IBUS高速缓存区为512字节，由32条缓存线组成，每条缓存线为2*64位；
- DBUS高速缓存区为256字节，由8条缓存线组成，每条缓存线为4*64位；
- 闪存页大小为1KB；
- 支持32位整字或双16位半字编程，页擦除和整片擦除操作；
- 大小为16字节的选项字节可根据用户需求配置；
- 512字节OTP块(一次性编程)，用于存储用户数据；
- 当系统复位时，选项字节被上载到选项字节控制寄存器；
- 具有安全保护状态，可阻止对代码或数据的非法读访问；
- 具有擦除和编程保护状态，可阻止意外写操作。

## 2.3. 功能说明

### 2.3.1. 闪存结构

闪存页大小为1KB。主存储闪存的每页都可以单独擦除。闪存结构细节见下表。

**表 2-1. GD32E11x 闪存基地址和构成**

| 闪存块 | 名称 | 地址范围 | 大小(字节) |
|---|---|---|---|
| 主存储闪存块 | 第0页 | 0x0800 0000 - 0x0800 03FF | 1KB |
| | 第1页 | 0x0800 0400 - 0x0800 07FF | 1KB |
| | 第2页 | 0x0800 0800 - 0x0800 0BFF | 1KB |
| | . | . | . |
| | . | . | . |

| | | | |
|---|---|---|---|
| | . | . | . |
| | 第127页 | 0x0801 FC00 - 0x0801 FFFF | 1KB |
| 信息块 | 引导装载程序 | 0x1FFF B000- 0x1FFF F7FF | 18KB |
| 选项字节块 | 选项字节 | 0x1FFF F800 - 0x1FFF F80F | 16B |
| 一次性编程块 | OTP字节 | 0x1FFF_7000~0x1FFF_71FF | 512B |

**注意:** 信息块存储了引导装载程序(boot loader)，不能被用户编程或擦除。

### 2.3.2. 读操作

闪存可以像普通存储空间一样直接寻址访问。对闪存取指令和取数据分别使用CPU的IBUS或DBUS总线。

**增加等待状态:**

根据AHB时钟频率，正确配置FMC_WS寄存器中的WSCNT位。WSCNT位和AHB时钟频率对应关系见下表。

**表 2-2. WSCNT 与 AHB 时钟频率对应关系**

| AHB 时钟频率 | WSCNT 配置 |
|---|---|
| <= 30MHz | 0 (0等待状态增加) |
| <= 60MHz | 1 (1等待状态增加) |
| <= 90MHz | 2 (2等待状态增加) |
| <= 120MHz | 3 (3等待状态增加) |

如果发生系统复位，AHB时钟频率为8MHz，此时WSCNT置为0。

**注意:**

1. 如果希望增加AHB时钟频率。首先，参考*表2-2. WSCNT与AHB时钟频率对应关系*，根据目标AHB时钟频率配置WSCNT位。然后，增加AHB时钟频率至目标频率。禁止在配置WSCNT位之前增加AHB时钟频率。

2. 如果希望降低AHB时钟频率。首先，降低AHB时钟频率至目标频率。然后，参考*表2-2. WSCNT与AHB时钟频率对应关系*，根据目标AHB时钟频率配置WSCNT位。禁止在降低AHB时钟频率之前配置WSCNT位。

**当前缓存区：**

当前缓存区总是被使能的。每次从flash闪存中读取数据时，当前缓存区可以缓存64位数据。因为CPU每次读操作只需要32位或16位数据。因此在顺序代码下，CPU所需数据可以从当前缓存区获取而不必重复从flash闪存中获取。

**预取缓存区：**

置位FMC_WS寄存器中PFEN位来使能预取缓存区。预取缓存区仅在IBUS上执行。在顺序代码下，当CPU执行来自当前缓存区的数据时(64位)，按32位执行时需要至少2个时钟周期，按16位执行时需要至少4个时钟周期。在这种情况下，从flash闪存中预取下一个双字地址的数据并存储在预取缓存区。当CPU执行完当前缓存区的数据时，预取缓存区提供下次需要执行的数据。

**IBUS 高速缓存区：**

置位FMC_WS寄存器中ICEN位来使能IBUS高速缓存区。IBUS高速缓存区仅在IBUS取数据时使用。IBUS快速缓存区为512字节，由32条缓存线组成，每条缓存线为2*64位。

当IBUS数据存在于IBUS高速缓存区时，CPU从IBUS高速缓存区读取数据时无延迟。当IBUS数据不存在于IBUS高速缓存区并且也不存在于当前缓存区/预取缓存区时，高速缓存线从flash闪存中读取数据并复制到IBUS高速缓存区。当所有高速缓存线被填充，LRU(最近最少使用)策略被用于转移高速缓存线中数据。

**DBUS 高速缓存区：**

置位FMC_WS寄存器中DCEN位来使能IBUS高速缓存区。DBUS高速缓存区仅在CPU通过DBUS取数据(不是通过DMA)时使用，此时选项字节不可缓存。DBUS快速缓存区为256字节，由8条缓存线组成，每条缓存线为4*64位。

当DBUS数据存在于DBUS高速缓存区时，CPU从DBUS高速缓存区读取数据时无延迟。当DBUS数据不存在于DBUS高速缓存区并且也不存在于当前缓存区时，高速缓存线从flash闪存中读取数据并复制到DBUS高速缓存区。当所有高速缓存线被填充，LRU(最近最少使用)策略被用于转移高速缓存线中数据。

### 2.3.3. FMC_CTL 寄存器解锁

复位后，FMC_CTL寄存器进入写操作锁定状态，LK位复位后置为1。通过先后向FMC_KEY寄存器写入0x45670123和0xCDEF89AB，可以使得FMC_CTL寄存器解锁。两次写操作后，FMC_CTL寄存器的LK位被硬件清0。可以通过软件设置FMC_CTL寄存器的LK位为1再次锁定FMC_CTL寄存器。任何对FMC_KEY寄存器的错误操作都会将LK位置1，从而锁定FMC_CTL寄存器，并引发一个总线错误。

FMC_CTL寄存器的OBPG位和OBER位在FMC_CTL寄存器解锁后，仍然被保护。向FMC_OBKEY寄存器先后写入0x45670123和0xCDEF89AB，然后硬件将FMC_CTL寄存器的

OBWEN位置1。软件可以将FMC_CTL的OBWEN位清0来锁定FMC_CTL的OBPG位和OBER位。

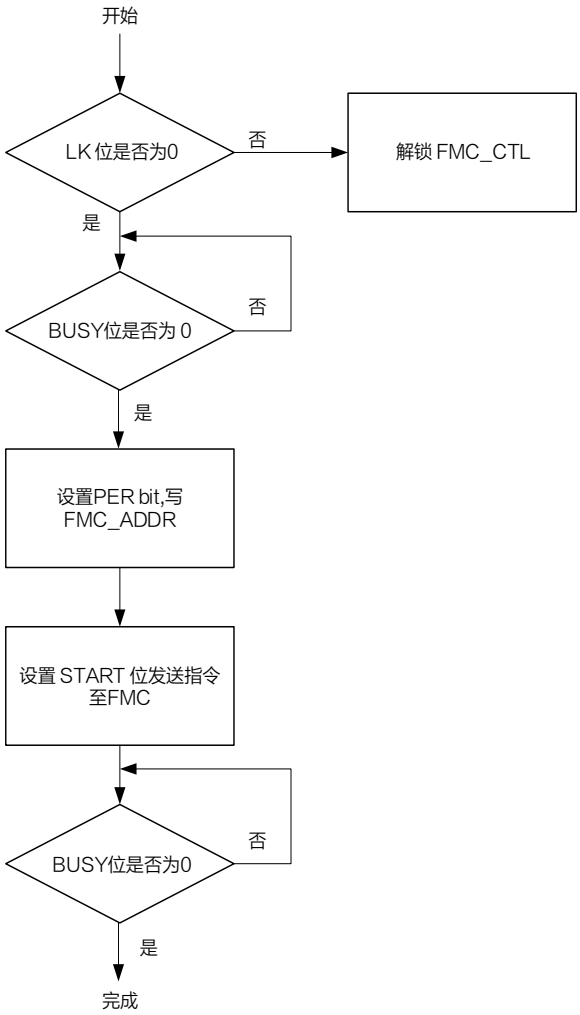
### 2.3.4. 页擦除

FMC的页擦除功能使得主存储闪存的页内容初始化为高电平。每一页都可以被独立擦除，而不影响其他页内容。页擦除页操作，寄存器设置具体步骤如下：

- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；
- 置位FMC_CTL寄存器的PER位；
- 将待擦除页的绝对地址(0x08XX XXXX)写到FMC_ADDR寄存器；
- 通过将FMC_CTL寄存器的START位置1来发送页擦除命令到FMC；
- 等待擦除指令执行完毕，FMC_STAT寄存器的BUSY位清0；
- 如果需要，使用DBUS读并验证该页是否擦除成功。

当页擦除成功执行，FMC_STAT寄存器的ENDF位将置位。若FMC_CTL寄存器的ENDIE位被置1，则FMC将触发一个中断。需要注意的是，用户需确保写入的是正确的擦除地址。否则当待擦除页的地址被用来取指令或访问数据时，软件将会"跑飞"。该情况下，FMC不会提供任何出错通知。另一方面，对擦写保护的页进行擦除操作将无效。如果FMC_CTL寄存器的ERRIE位被置位，该操作将触发操作出错中断。中断服务程序可通过检测FMC_STAT寄存器的WPERR位来判断该中断是否发生。下图显示了页擦除操作流程。

**图 2-1. 页擦除操作流程**




### 2.3.5. 整片擦除

FMC提供了整片擦除功能可以初始化主存储闪存块的内容。当设置FMC_CTL寄存器中MER为1时，擦除过程作用于整片闪存。整片擦除操作，寄存器设置具体步骤如下：

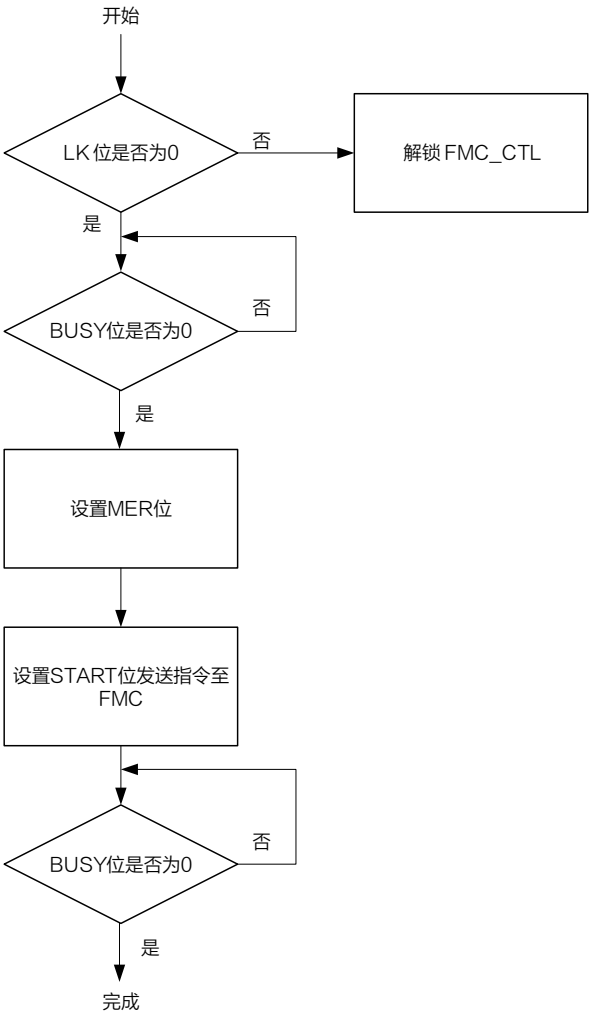
- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；
- 如果整片擦除闪存，置位FMC_CTL寄存器的MER位；
- 通过将FMC_CTL寄存器的START位置1来发送整片擦除命令到FMC；
- 等待擦除指令执行完毕，FMC_STAT寄存器的BUSY位清0；
- 如果需要，使用DBUS读并验证是否擦除成功。

当整片擦除成功执行，FMC_STAT寄存器的ENDF位置位。若FMC_CTL寄存器的ENDIE位被置1，FMC将触发一个中断。由于所有的闪存数据都将被复位为0xFFFF_FFFF，可以通过运行在SRAM中的程序或使用调试工具直接访问FMC寄存器来实现整片擦除操作。此外，如果任何闪存页处于擦除/编程保护下，整片擦除操作会被忽略。在这种情况下，如果FMC_CTL寄存器

的ERRIE位被置位，该操作将触发操作出错中断。在中断服务程序中，软件可以通过检查FMC_STAT寄存器中的WPERR位来检测这种情况。

下图显示了整片擦除操作流程。

**图 2-2. 整片擦除操作流程**




### 2.3.6. 主存储闪存块编程

FMC提供了一个通过DBUS修改主存储闪存内容的32位整字/16位半字编程功能。实际上，主存储闪存编程为32位或64位，由FMC_WS寄存器中的PGW位定义。

编程操作，寄存器设置具体步骤如下：

- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；
- 设置FMC_WS寄存器的PGW位；
- 置位FMC_CTL寄存器的PG位；
- DBUS写一个32位整字/16位半字到目的绝对地址(0x08XX XXXX)；

如果DBUS按32位字编程并且PGW位设置为0(32位编程)，DBUS写1次即可将数据编程入闪存。待编程数据必须字对齐。
如果DBUS按32位字编程并且PGW位设置为1(64位编程)，DBUS写2次即可将数据编程入闪存。待编程数据必须双字对齐。
如果DBUS按16位字编程并且PGW位设置为0(32位编程)，DBUS写2次即可将数据编程入闪存。待编程数据必须字对齐。
如果DBUS按16位字编程并且PGW位设置为1(64位编程)，DBUS写4次即可将数据编程入闪存。待编程数据必须双字对齐。
为了缩短编程时间，建议DBUS采用按32位编程。当待编程数据按双字对齐时，设置PGW位为1。当待编程数据按字对齐时，设置PGW位为0。

- 等待编程指令执行完毕，FMC_STAT寄存器的BUSY位清0；
- 如果需要，使用DBUS读并验证是否编程成功。

当主存储块编程成功执行，FMC_STAT寄存器的ENDF位置位。若FMC_CTL寄存器的ENDIE位被置1，FMC将触发一个中断。有一些编程错误需要注意：

编程操作时需要检查目的地址是否已经被擦除。如果该地址没有被擦除，即使对该地址写0x0值，FMC_STAT寄存器的PGERR位也将被置1。每个字在擦除后和下次擦除前只能编程一次。注意PG位必须在字/半字编程操作前被置位。

此外，在正被擦除/保护页上的编程操作会被忽略，FMC_STAT寄存器中的WPERR位会被置位。

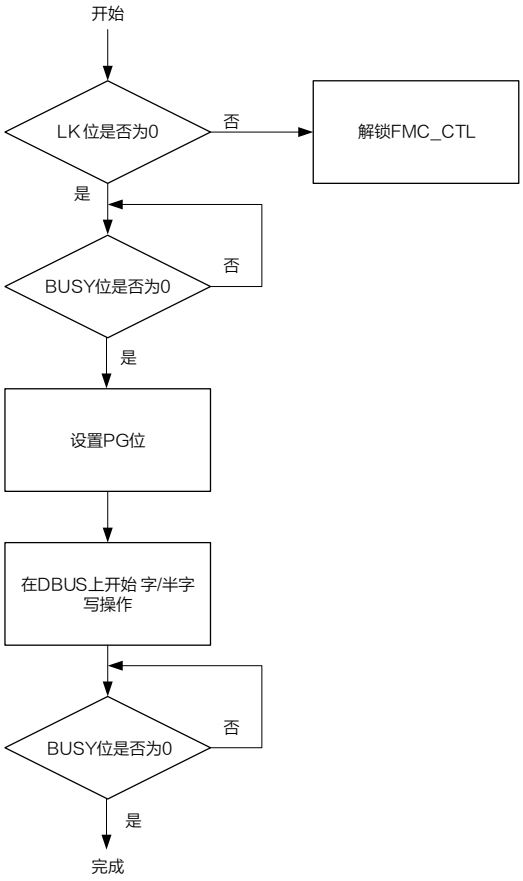
在下列情况下，FMC_STAT寄存器中的PGAERR位会被置位。

- DUBS按字节编程(非32位或16位)
- DBUS混合编程。不允许混合32位和16位编程。
- DBUS编程未对齐。如果DBUS按32位编程并且PGW设置为1(64位编程)，下一次DBUS必须在对应的双字地址按双字对齐写操作。如果DBUS按16位编程并且PGW设置为0(32位编程)，下一次DBUS必须在对应的字地址按字对齐写操作。如果DBUS按16位编程并且PGW设置为1(64位编程)，第2$^{nd}$/3$^{rd}$/4$^{th}$ DBUS写操作必须在对应的双字地址按双字对齐。

**注意：**如果编程数据未能写满64位/32位，这些数据不会被编程入flash闪存，并且不会有任何提示。

在这些情况下，如果FMC_CTL寄存器的ERRIE位被置1，FMC将触发一次闪存操作错误中断。在中断服务程序中，软件可以通过检查FMC_STAT寄存器中的PGERR位，PGAERR位或者WPERR位来检测发生了哪种错误。下图显示了主存储块字编程操作流程。

图 2-3. 字编程操作流程




**注意**：避免在同一个bank中既进行读操作，又进行擦除/编程操作。当CPU进入省电模式时，对闪存的操作将失败。

### 2.3.7. OTP 编程

OTP编程方法与主储存闪存编程相同。OTP块只能被编程一次并且不能被擦除。

**注意：**必须确保在OTP编程操作时不会发生任何意外中断，例如系统复位或掉电。如果发生意外中断，flash闪存中的数据有很小可能性会出错。

### 2.3.8. 选项字节擦除

FMC提供了一个擦除功能用来初始化闪存中的选项字节。选项字节擦除过程如下所示。

- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；
- 解锁FMC_CTL寄存器的选项字节操作位；
- 等待FMC_CTL寄存器的OBWEN位置1；
- 置位FMC_CTL寄存器的OBER位；

- 通过将FMC_CTL寄存器的START位置1来发送选项字节擦除命令到FMC；
- 等待擦除指令执行完毕，FMC_STAT寄存器的BUSY位清0；
- 如果需要，使用DBUS读并验证是否擦除成功。

当选项字节擦除成功执行，FMC_STAT寄存器的ENDF位置位。若FMC_CTL寄存器的ENDIE位被置1，FMC将触发一个中断。

### 2.3.9. 选项字节编程

FMC提供了擦除/编程功能，可用来修改选项字节内容。选项字节共有8对选项字节。每对选项字节的高字节是低字节的补字节。当选项字节被修改时，FMC自动生成该选项字节的高字节。擦除操作过程如下。

- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；
- 解锁FMC_CTL寄存器的选项字节操作位；
- 等待FMC_CTL寄存器的OBWEN位置1；
- 置位FMC_CTL寄存器的OBPG位；
- DBUS写一个32位整字/16位半字到目的地址。根据不同的DBUS写数据大小和FMC_WS寄存器中的PGW位数值，可能需要写1次，2次或4次。写操作方法与主存储闪存编程操作类似；
- 等待编程指令执行完毕，FMC_STAT寄存器的BUSY位清0；
- 如果需要，使用DBUS读并验证是否编程成功。

当选项字节编程成功执行，FMC_STAT寄存器的ENDF位置位。若FMC_CTL寄存器的ENDIE位被置1，FMC将触发一个中断。注意可能发生编程错误，类似主存储闪存编程操作，PGERR位和PGAERR位可能被置位。

选项字节修改后只有在系统复位后才生效。

### 2.3.10. 选项字节说明

每次系统复位后，选项字节被重加载到FMC_OBSTAT和FMC_WP寄存器后，选项字节生效。选项字节的补字节具体为选项字节取反。当选项字节被重装载时，如果选项字节的补字节和选项字节不匹配，FMC_OBSTAT寄存器的OBERR位将被置1，选项字节被强制设置为0xFF。若选项字节和其补字节同为0xFF，则OBERR位不置位。选项字节详情见下表。

**表 2-3. 选项字节**

| 地址 | 名称 | 说明 |
|---|---|---|
| 0x1fff f800 | SPC | 选项字节安全保护值<br>0xA5：未保护状态<br>除0xA5外的任何值：已保护状态 |
| 0x1fff f801 | SPC_N | SPC补字节 |
| 0x1fff f802 | USER | [7:3]：保留<br>[2]：nRST_STDBY |

| 地址 | 名称 | 说明 |
| --- | --- | --- |
| | | 0：设置待机模式时产生复位而不是进入待机模式<br>1：设置待机模式时进入待机模式而不产生复位<br>[1]：nRST_DPSLP<br>0：设置深度睡眠模式时产生复位而不进入深度睡眠模式<br>1：设置深度睡眠模式时进入深度睡眠模式而不产生复位<br>[0]：nWDG_HW<br>0：硬件使能独立看门狗功能<br>1：软件使能独立看门狗功能 |
| 0x1fff f803 | USER_N | USER补字节值 |
| 0x1fff f804 | DATA[7:0] | 用户定义数据7到0位 |
| 0x1fff f805 | DATA_N[7:0] | DATA补字节值的7到0位 |
| 0x1fff f806 | DATA[15:8] | 用户定义数据15到8位 |
| 0x1fff f807 | DATA_N[15:8] | DATA补字节值的15到8位 |
| 0x1fff f808 | WP[7:0] | 页擦除/编程保护值的7到0位<br>0：保护生效<br>1：未保护 |
| 0x1fff f809 | WP_N[7:0] | WP补字节值的7到0位 |
| 0x1fff f80a | WP[15:8] | 页擦除/编程的保护值的15到8位 |
| 0x1fff f80b | WP_N[15:8] | WP补字节值的15到8位 |
| 0x1fff f80c | WP[23:16] | 页擦除/编程的保护值的23到16位 |
| 0x1fff f80d | WP_N[23:16] | WP补字节值的23到16位 |
| 0x1fff f80e | WP[31:24] | 页擦除/编程的保护值的31到24位<br>WP[30:24]：每个bit可设置4KB闪存的保护状态。第0位设置前4KB闪存的保护状态，以此类推。这31位总计可设置前124KB的闪存保护状态。<br>WP[31]：第31位可设置闪存剩下部分的保护状态。 |
| 0x1fff f80f | WP_N[31:24] | WP补字节值的31到24位 |

### 2.3.11. 页擦除/编程保护

FMC的页擦除/编程保护功能可以阻止对闪存的意外操作。当FMC对被保护页进行页擦除或编程操作时，操作本身无效且FMC_STAT寄存器的WPERR位将被置1。如果WPERR位被置1且FMC_CTL寄存器的ERRIE位也被置1来使能相应的中断，FMC将触发闪存操作出错中断，等待CPU处理。配置选项字节的WP [31:0]某位为0可以单独使能某几页的保护功能。如果在选项字节块执行了擦除操作，所有的闪存页擦除和编程保护功能都将失效。当选项字节的WP被改变时，需要系统复位使之生效。

### 2.3.12. 安全保护

FMC提供了一个安全保护功能来阻止非法读取闪存。此功能可以很好地保护软件和固件免受非法的用户操作。

未保护状态：当将SPC字节和它的补字节被设置为0x5AA5，系统掉电复位以后，闪存将处于

非安全保护状态。主存储块和选项字节可以被所有操作模式访问。

已保护状态：当设置SPC字节和它的补字节值为任何除0x5AA5外的值，系统掉电复位以后，安全保护状态生效。需要注意的是，若该修改过程中，MCU的调试模块依然和外部JTAG/SWD设备相连，需要用上电复位代替系统复位以使得修改后的保护状态生效。在安全保护状态下，主存储闪存块仅能被用户代码访问且前4KB的闪存自动处于页擦除/编程保护状态下。在调试模式下，或从SRAM中启动时，以及从boot loader区启动时，这些模式下对主存储块的操作都被禁止。如果在这些模式下读主存储块，将产生总线错误。如果在这些模式下，对主存储块进行编程或擦除操作，FMC_STAT寄存器的WPERR位将被置1。但这些模式下都可以对选项字节进行操作，从而可以通过该方式失能安全保护功能。如果将SPC字节和它的补字节设置为0x5AA5，安全保护功能将失效，并自动触发一次整片擦除操作。

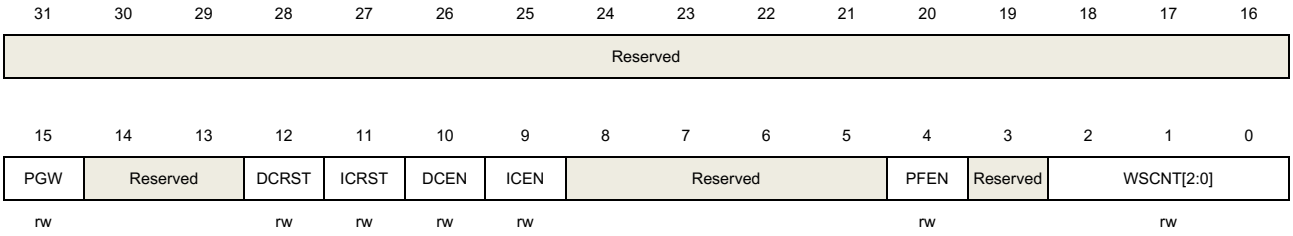
## 2.4. FMC 寄存器

FMC基地址：0x4002 2000

### 2.4.1. 等待状态寄存器 (FMC_WS)

地址偏移：0x00
复位值：0x0000 0630

该寄存器只能按字(32位)访问




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15 | PGW | 编程宽度<br>0：32位编程<br>1：64位编程 |
| 14:13 | 保留 | 必须保持复位值 |
| 12 | DCRST | 复位DBUS高速缓存区。该位仅可在DCEN位置0时可写。<br>0：无效果<br>1：复位DBUS高速缓存区 |
| 11 | ICRST | 复位IBUS高速缓存区。该位仅可在ICEN位置0时可写。<br>0：无效果<br>1：复位IBUS高速缓存区 |
| 10 | DCEN | DBUS高速缓存区使能位<br>0：失能DBUS高速缓存区<br>1：使能DBUS高速缓存区 |
| 9 | ICEN | IBUS高速缓存区使能位<br>0：失能IBUS高速缓存区<br>1：使能IBUS高速缓存区 |
| 8:5 | 保留 | 必须保持复位值。 |
| 4 | PFEN | 预取功能使能位<br>0：失能预取功能<br>1：使能预取功能 |

| | | |
|---|---|---|
| 3 | 保留 | 必须保持复位值 |
| 2:0 | WSCNT[2:0] | 等待状态计数寄存器<br>软件置1和清0。<br>000：不增加等待状态<br>001：增加1个等待状态<br>010：增加2个等待状态<br>011：增加3个等待状态<br>100 ~ 111：保留 |

### 2.4.2. 解锁寄存器 (FMC_KEY)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32位)访问

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | KEY[31:0] | FMC_CTL解锁寄存器<br>这些位仅能被软件写。<br>写解锁值到KEY[31:0]可以解锁 FMC_CTL寄存器。 |

### 2.4.3. 选项字节操作解锁寄存器(FMC_OBKEY)

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字(32位)访问

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | OBKEY [31:0] | FMC_CTL选项字节操作解锁寄存器 |

这些位仅能被软件写

写解锁值到OBKEY[31:0]解锁FMC_CTL寄存器的选项字节命令。

### 2.4.4. 状态寄存器 (FMC_STAT)

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能按字(32位)访问

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | | | ENDF | WPERR | PGAERR | PGERR | 保留 | BUSY |
| | | | | | | | | | | rc_w1 | rc_w1 | rc_w1 | rc_w1 | | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:6 | 保留 | 必须保持复位值 |
| 5 | ENDF | 操作结束标志位<br>操作成功执行后，此位被硬件置1。软件写1清0。 |
| 4 | WPERR | 擦除/编程保护错误标志位<br>在受保护的页上擦除/编程操作时，此位被硬件置1。软件写1清0。 |
| 3 | PGAERR | 编程对齐错误标志位<br>当DBUS写数据不对齐时，此位被硬件置1。软件写1清0。 |
| 2 | PGERR | 编程错误标志位<br>当被编程区域状态不为0xFFFF时，对闪存编程，此位被硬件置1。软件写1清0。 |
| 1 | 保留 | 必须保持复位值 |
| 0 | BUSY | 闪存忙标志<br>当闪存操作正在进行时，此位被置1。当操作结束或者出错，此位被清0。 |

### 2.4.5. 控制寄存器 (FMC_CTL)

地址偏移：0x10

复位值：0x0000 0080

该寄存器只能按字(32位)访问

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

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

| 保留 | ENDIE | 保留 | ERRIE | OBWEN | 保留 | LK | START | OBER | OBPG | 保留 | MER | PER | PG |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | rw | | rw | rw | | rs | rs | rw | rw | | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:13 | 保留 | 必须保持复位值 |
| 12 | ENDIE | 操作结束中断使能位<br>软件置1和清0<br>0：无硬件中断产生<br>1：使能操作结束中断 |
| 11 | 保留 | 必须保持复位值 |
| 10 | ERRIE | 出错中断使能位<br>软件置1和清0<br>0：无硬件中断产生.<br>1：使能出错中断 |
| 9 | OBWEN | 选项字节擦除/编程使能位<br>当正确的序列写入FMC_OBKEY寄存器，此位由硬件置1。此位可以被软件清0。 |
| 8 | 保留 | 必须保持复位值 |
| 7 | LK | FMC_CTL0寄存器锁定标志位<br>当正确的序列写入FMC_KEY0寄存器，此位由硬件清0。此位可以由软件置1。 |
| 6 | START | 向FMC发送擦除命令位<br>软件置1可以发送擦除命令到FMC。当BUSY位被清0时，此位由硬件清0。 |
| 5 | OBER | 选项字节擦除命令位<br>软件置1和清0<br>0：无作用<br>1：选项字节擦除命令 |
| 4 | OBPG | 选项字节编程命令位<br>软件置1和清0<br>0：无作用<br>1：选项字节编程命令 |
| 3 | 保留 | 必须保持复位值 |
| 2 | MER | 主存储块整片擦除命令位<br>软件置1和清0<br>0：无作用<br>1：主存储块整片擦除命令 |
| 1 | PER | 主存储块页擦除命令位<br>软件置1和清0<br>0：无作用<br>1：主存储块页擦除命令 |

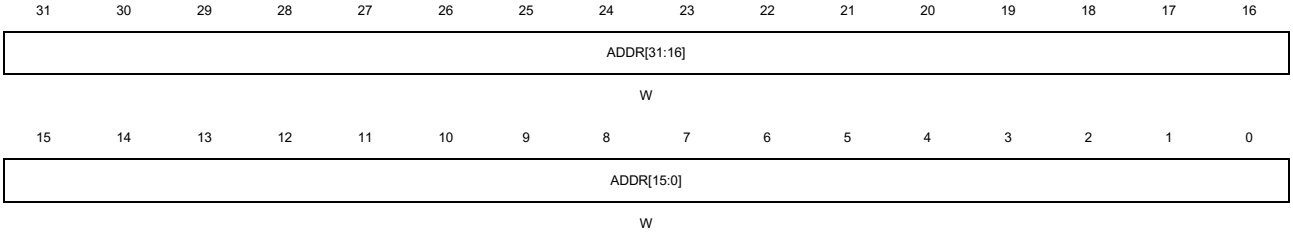
| | | |
|---|---|---|
| 0 | PG | 主存储块编程命令位<br>软件置1和清0<br>0：无作用<br>1：主存储块编程命令 |

**注意：**当相应闪存操作完成后，该寄存器需处于复位状态。

### 2.4.6. 地址寄存器 (FMC_ADDR)

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字(32位)访问

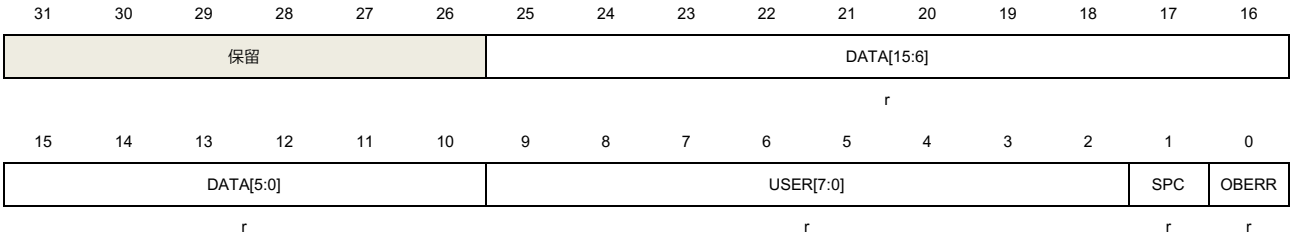



| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | ADDR[31:0] | 闪存擦除或编程地址<br>该位通过软件设置。<br>ADDR位是闪存擦除/编程命令的地址 |

### 2.4.7. 选项字节状态寄存器 (FMC_OBSTAT)

地址偏移：0x1C
复位值：0x0XXX XXXX

该寄存器只能按字(32位)访问




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:26 | 保留 | 必须保持复位值 |
| 25:10 | DATA[15:0] | 系统复位后保存选项字节的DATA[15:0]部分 |
| 9:2 | USER[7:0] | 系统复位后保存选项字节块的USER字节 |

| | | |
|---|---|---|
| 1 | SPC | 安全保护状态<br>0：未保护<br>1：已保护 |
| 0 | OBERR | 选项字节读错误位<br>当选项字节和它的补字节不匹配时此位由硬件置1，选项字节被强制设置为0xFF。 |

### 2.4.8. 擦除/编程保护寄存器 (FMC_WP)

地址偏移：0x20
复位值：0xXXXX XXXX

该寄存器只能按字(32位)访问

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | WP[31:0] | 系统复位后保存选项字节块的WP[31:0]部分 |

### 2.4.9. 产品 ID 寄存器 (FMC_PID)

地址偏移：0x100
复位值：0xXXXX XXXX

该寄存器只能按字(32位)访问

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | PID[31:0] | 产品保留 ID寄存器<br>该寄存器为只读<br>上电后这些位始终不会改变，该寄存器在生产过程中被一次性编程。 |

# 3. 电源管理单元(PMU)

## 3.1. 简介

功耗设计是GD32E11x系列产品最注重的问题之一。电源管理单元提供了三种省电模式，包括睡眠模式、深度睡眠模式和待机模式。这些模式能减少电源能耗，且使得应用程序可以在CPU运行时间要求、速度和功耗的相互冲突中获得最佳折衷。如*图3-1. 电源域概览*所示，GD32E11x系列设备有三个电源域，包括$V_{DD}$ / $V_{DDA}$域、$V_{CORE}$域和备份域。$V_{DD}$ / $V_{DDA}$域由电源直接供电。在$V_{DD}$ / $V_{DDA}$域中嵌入了一个LDO，用来为$V_{CORE}$域供电。在备份域中有一个电源切换器，当VDD电源关闭时，电源切换器可以将备份域的电源切换到VBAT引脚，此时备份域由VBAT引脚(电池)供电。

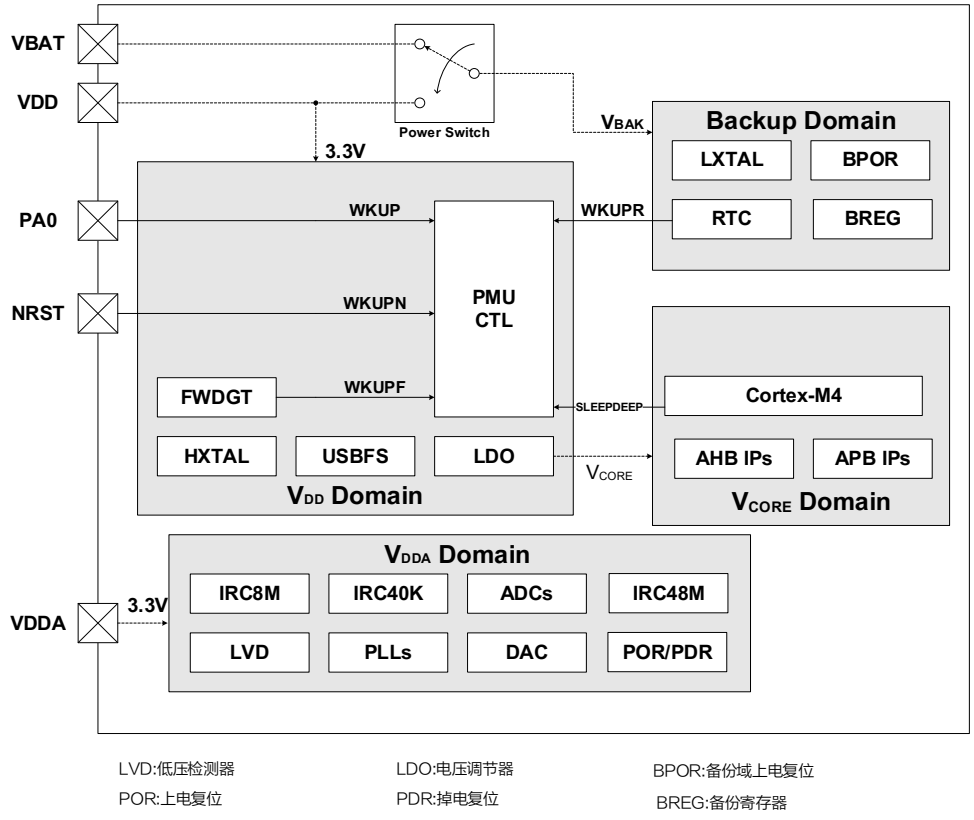
## 3.2. 主要特征

- 三个电源域：备份域、$V_{DD}$ / $V_{DDA}$域和$V_{CORE}$电源域；
- 三种省电模式：睡眠模式、深度睡眠模式和待机模式；
- 内部电压调节器(LDO)提供$V_{CORE}$电源；
- 提供低电压检测器(LVD)，当电压低于所设定的阈值时能发出中断或事件；
- 当VDD供电关闭时，由VBAT(电池)为备份域供电；
- LDO输出电压用于节约能耗。

## 3.3. 功能说明

*图3-1. 电源域概览*提供了PMU及相关电源域的内部结构框图。

**图3-1. 电源域概览**




### 3.3.1. 电池备份域

电池备份域由内部电源切换器来选择VDD供电或VBAT(电池)供电，然后由$V_{BAK}$为备份域供电，该备份域包含RTC(实时时钟)、LXTAL(低速外部晶体振荡器)、BPOR(备份域上电复位)、BREG(备份寄存器)，以及PC13至PC15共3个BKP PAD。为了确保备份域中寄存器的内容及RTC正常工作，当VDD关闭时，VBAT引脚可以连接至电池或其他电源等备份源供电。电源切换器是由$V_{DD}$ / $V_{DDA}$域掉电复位电路控制的。对于没有外部电池的应用，建议将VBAT引脚通过100nF的外部陶瓷去耦电容连接到VDD引脚上。

备份域的复位源包括备份域上电复位和备份域软件复位。在$V_{BAK}$没有完全上电前，BPOR信号强制设备处于复位状态。应用软件可以通过设置RCU_BDCTL寄存器BKPRST位来触发备份域软件复位。

RTC的时钟源可以是低速内部RC振荡器(IRC40K)或低速外部晶体振荡器(LXTAL)，或高速外部晶体振荡器(HXTAL)时钟的128分频。当$V_{DD}$被关闭时，RTC只能选择LXTAL作为时钟源。在通过WFI / WFE指令进入省电模式之前，Cortex®-M4需要通过RTC寄存器设置预期的唤醒时间并启用唤醒功能，以实现RTC定时器唤醒事件。进入省电模式一定时间之后，当经过的时间与预设的唤醒时间匹配时，RTC将唤醒设备。RTC的配置和操作的细节将在*实时时钟(RTC)*章节来

描述。

当备份域由VDD供电($V_{BAK}$连接至$V_{DD}$)时，以下功能可用：

- PC13可以作为通用I / O口或RTC功能引脚(参见*实时时钟(RTC)*)；
- PC14和PC15可以作为通用I / O口或LXTAL晶振引脚。

当备份域由VBAT电源供电时($V_{BAK}$连接至$V_{BAT}$)，以下功能可用：

- PC13仅可以作为RTC功能引脚(参见*实时时钟(RTC)*)；
- PC14和PC15仅可作为LXTAL晶振引脚。

**注意：**由于PC13至PC15引脚是通过电源切换器供电的，电源切换器仅可通过小电流，因此当PC13至PC15的GPIO口在输出模式时，其工作的速度不能超过2MHz(最大负载为30pF)。

### 3.3.2. $V_{DD}$ / $V_{DDA}$ 电源域

$V_{DD}$ / $V_{DDA}$ 域包括 $V_{DD}$ 域和 $V_{DDA}$ 域两部分。$V_{DD}$ 域包括 HXTAL(高速外部晶体振荡器)、LDO(电压调节器)、USBFS(通用串行总线全速接口)、FWDGT(独立看门狗定时器)和除 PC13、PC14 和 PC15 之外的所有 PAD 等等。$V_{DDA}$ 域包括 ADC / DAC(AD / DA 转换器)、IRC8M(内部 8M RC 振荡器)、IRC48M(内部 48M RC 振荡器)、IRC40K(内部 40KHz RC 振荡器)、POR / PDR(上电 / 掉电复位)、PLLs(锁相环)和 LVD(低电压检测器)等等。

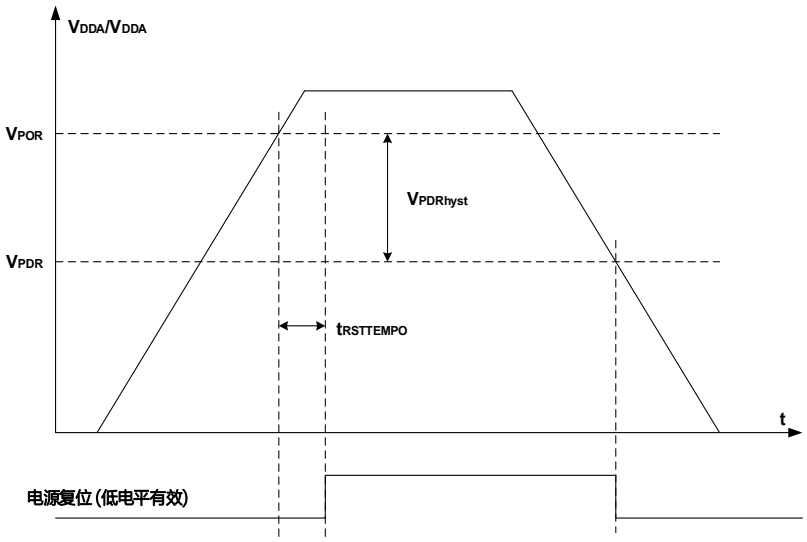
#### $V_{DD}$ 域

为 $V_{CORE}$ 域供电的 LDO(电压调节器)，其复位后保持使能。可以被配置为三种不同的工作状态：包括睡眠模式(全供电状态)、深度睡眠模式(全供电或低功耗状态)和待机模式(关闭状态)。

#### $V_{DDA}$ 域

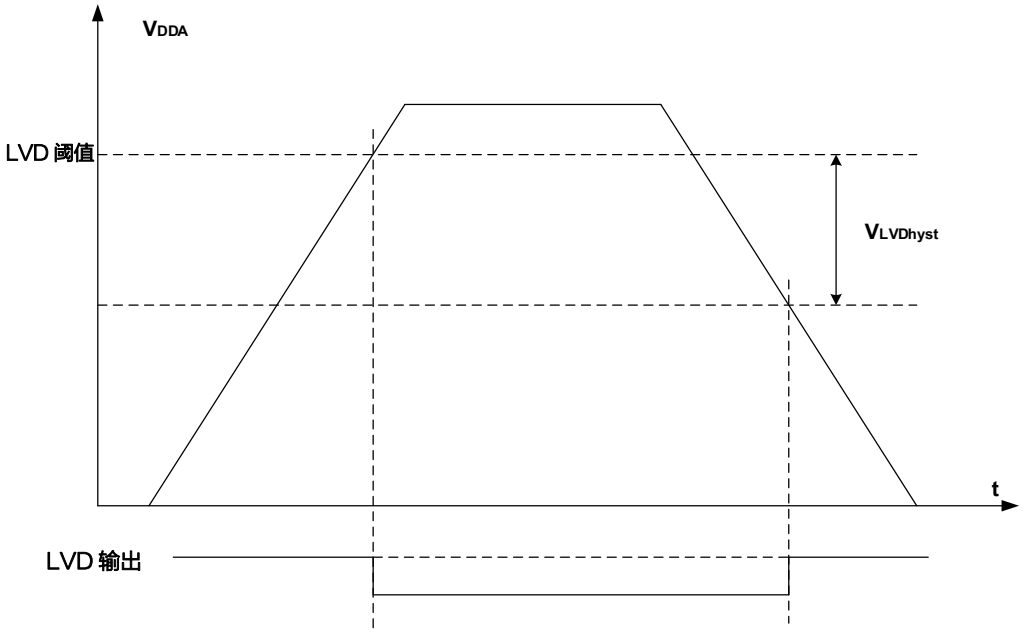
POR / PDR(上电 / 掉电复位)电路检测 VDDA / VDDA 并在电压低于特定阈值时产生电源复位信号复位除备份域之外的整个芯片。*图 3-2. 上电 / 掉电复位波形图*显示了供电电压和电源复位信号之间的关系。$V_{POR}$ 表示上电复位的阈值电压，$V_{PDR}$ 表示掉电复位的阈值电压，$V_{PDRhyst}$ 表示迟滞电压值，以上参数值请参见数据手册。

**图3-2. 上电 / 掉电复位波形图**




LVD 的功能是检测 VDDA 供电电压是否低于低电压检测阈值，该阈值由电源控制寄存器(PMU_CTL)中的 LVDT[2:0]位进行配置。LVD 通过 LVDEN 置位使能，位于电源控制和状态寄存器(PMU_CS)中的 LVDF 位表示低电压事件是否出现，该事件连接至 EXTI 的第 16 线，用户可以通过配置 EXTI 的第 16 线产生相应的中断。*图 3-3. LVD 阈值波形图*显示了 $V_{DDA}$ 供电电压和 LVD 输出信号的关系(LVD 中断信号依赖于 EXTI 第 16 线的上升或下降沿配置)。迟滞电压 $V_{LVDhyst}$ 值请参见数据手册。

**图3-3. LVD阈值波形图**




一般来说，数字电路由 VDD 供电，而大多数的模拟电路由 VDDA 供电。为了提高 ADC 和 DAC 的转换精度，为 VDDA 独立供电可使模拟电路达到更好的特性。为避免噪声，VDDA 通过外部滤波电路连接至 VDD，相应的 VSSA 通过特定电路连接至 VSS。否则，当 VDD 和 VDDA 不是同一个电源提供时，在上电和运行过程中 VDD 与 VDDA 差值不超过 0.3V。

为提高 ADC 和 DAC 的精度，可将独立的外部参考电压连接至 ADC / DAC 引脚 VREFP / VREFN。根据不同的封装，VREFP 可被连接至 VDDA 引脚，或者外部参考电压，外部参考电压的范围请参考*表 12-2. ADC 输入引脚定义*和*表 13-1. DAC 引脚*,VREFN 须被连接至 VSSA 引脚，VREFP 引脚仅存在于不小于 100-pin 的封装上，而在 64-pin 或更少引脚的封装上不存在，因其内部已经连接至 VDDA。VREFN 内部直接连接至 VSSA。

### 3.3.3. $V_{CORE}$ 电源域

$V_{CORE}$电源域为Cortex®-M4内核逻辑、AHB / APB外设、备份域和$V_{DD}$ / $V_{DDA}$域的APB接口等供电。当$V_{CORE}$电压上电后，POR将在$V_{CORE}$域中产生一个复位序列，复位完成后，如果要进入指定的省电模式，须先配置相关的控制位，之后一旦执行WFI或WFE指令，设备便进入该省电模式。关于这方面的详细内容，将在以下章节予以说明。$V_{CORE}$电压值请参考数据手册。

### 3.3.4. 省电模式

系统复位或电源复位后，GD32E11x MCU处于全功能状态且电源域全部处于供电状态。实现较低的功耗的方法有三种：减慢系统时钟(HCLK，PCLK1，PCLK2)，关闭未使用的外设的时钟或通过PMU_CTL寄存器的LDOVS来配置LDO输出电压。LDOVS只有在PLL关闭情况下才可以配置，在PLL打开时，被配置的LDO输出电压才会被用来驱动$V_{CORE}$电源域。此外，三种省电模式可以实现更低的功耗，它们是睡眠模式、深度睡眠模式和待机模式。

#### 睡眠模式

睡眠模式与Cortex®-M4的SLEEPING模式相对应。在睡眠模式下，仅关闭Cortex®-M4的时钟。如需进入睡眠模式，只要清除Cortex®-M4系统控制寄存器中的SLEEPDEEP位，并执行一条WFI或WFE指令即可。如果睡眠模式是通过执行WFI指令进入的，任何中断都可以唤醒系统。如果睡眠模式是通过执行WFE指令进入的，任何唤醒事件都可以唤醒系统(如果SEVONPEND为1，任何中断都可以唤醒系统，请参考Cortex®-M4技术手册)。由于无需在进入或退出中断上消耗时间，该模式所需的唤醒时间最短。

根据Cortex®-M4中SCR(系统控制寄存器)的SLEEPONEXIT位，有两种睡眠进入机制可选：

- Sleep-now：如果SLEEPONEXIT位被清零，一旦执行WFI或WFE指令，MCU立即进入睡眠模式；
- Sleep-on-exit：如果SLEEPONEXIT位被置位，当系统从最低优先级的中断处理程序离开后，MCU立即进入睡眠模式。

#### 深度睡眠模式

深度睡眠模式与 Cortex®-M4 的 SLEEPDEEP 模式相对应。在深度睡眠模式下，$V_{CORE}$ 域中的所有时钟全部关闭，IRC8M、IRC48M、HXTAL 及 PLLs 也全部被禁用。SRAM 和寄存器中的内容被保留。根据 PMU_CTL 寄存器的 LDOLP 位的配置，可控制 LDO 工作在正常模式或低功耗模式。进入深度睡眠模式之前，先将 Cortex®-M4 系统控制寄存器的 SLEEPDEEP 位置 1，

再清除 PMU_CTL 寄存器的 STBMOD 位，然后执行 WFI 或 WFE 指令即可进入深度睡眠模式。如果睡眠模式是通过执行 WFI 指令进入的，任何来自 EXTI 的中断可以将系统从深度睡眠模式中唤醒。如果睡眠模式是通过执行 WFE 指令进入的，任何来自 EXTI 的事件可以将系统从深度睡眠模式中唤醒(如果 SEVONPEND 为 1，任何来自 EXTI 的中断都可以唤醒系统，请参考 Cortex®-M4 技术手册)。刚退出深度睡眠模式时，IRC8M 被选中作为系统时钟。请注意，如果 LDO 工作在低功耗模式，那么唤醒时需额外的延时时间。

**注意：**为了顺利进入深度睡眠模式，所有 EXTI 线上的挂起状态(在 EXTI_PD 寄存器中)和 RTC 闹钟标志必须被复位，参考*表 7-3. EXTI 触发源*。否则，程序将直接跳过深度睡眠模式进入过程而继续执行下面的程序。

### 待机模式

待机模式也是基于 Cortex®-M4 的 SLEEPDEEP 模式实现的。在待机模式下，整个 $V_{CORE}$ 域全部停止供电，同时 LDO 和包括 IRC8M、IRC48M、HXTAL 和 PLL 也会被关闭。进入待机模式前，先将 Cortex®-M4 系统控制寄存器的 SLEEPDEEP 位置 1，再将 PMU_CTL 寄存器的 STBMOD 位置 1，再清除 PMU_CS 寄存器的 WUF 位，然后执行 WFI 或 WFE 指令，系统进入待机模式，PMU_CS 寄存器的 STBF 位状态表示 MCU 是否已进入待机模式。待机模式有四个唤醒源，包括来自 NRST 引脚的外部复位，RTC 闹钟，FWDGT 复位，WKUP 引脚的上升沿。待机模式可以达到最低的功耗，但唤醒时间最长。另外，一旦进入待机模式，SRAM 和 $V_{CORE}$ 电源域寄存器的内容都会丢失。退出待机模式时，会发生上电复位，复位之后 Cortex®-M4 将从 0x0000 0000 地址开始执行指令代码。

**表3-1. 节电模式总结**

| 模式 | 睡眠 | 深度睡眠 | 待机 |
|---|---|---|---|
| **描述** | 仅关闭 CPU 时钟 | 1、关闭 $V_{CORE}$ 电源域的所有时钟<br>2、关闭 IRC8M、IRC48M、HXTAL 和 PLL | 1、关闭 $V_{CORE}$ 电源域的供电<br>2、关闭 IRC8M、IRC48M、HXTAL 和 PLL |
| **LDO 状态** | 开启(正常功耗模式) | 开启(正常功耗模式或者低功耗模式) | 关闭 |
| **配置** | SLEEPDEEP = 0 | SLEEPDEEP = 1<br>STBMOD = 0 | SLEEPDEEP = 1<br>STBMOD = 1, WURST=1 |
| **进入指令** | WFI 或 WFE | WFI 或 WFE | WFI 或 WFE |
| **唤醒** | 若通过 WFI 进入，则任何中断均可唤醒；<br>若通过 WFE 进入，则任何事件(或 SEVONPEND=1 时的中断)均可唤醒 | 若通过 WFI 进入，来自 EXTI 的任何中断可唤醒；<br>若通过 WFE 进入，来自 EXTI 的任何事件(或 SEVONPEND=1 时的中断)可唤醒 | 1. NRST 引脚<br>2. WKUP 引脚<br>3. FWDGT 复位<br>4. RTC 闹钟 |
| **唤醒延迟** | 无 | IRC8M 唤醒时间<br>如果 LDO 处于低功耗模式，需增加 LDO 唤醒时间 | 上电序列 |

**注意：**在待机模式下，除了 NRST 引脚，配置为 RTC 功能的 PC13，用作 LXTAL 晶振引脚的

PC14 和 PC15，使能的 WKUP 引脚，其他所有 I / O 都处于高阻态。

## 3.4. PMU 寄存器

PMU基地址：0x4000 7000

### 3.4.1. 控制寄存器(PMU_CTL)

地址偏移：0x00
复位值：0x0000 4000 (从待机模式唤醒后复位)

该寄存器可以按半字(16位)或字(32位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LDOVS[1:0] | | 保留 | | | | | BKPWEN | LVDT[2:0] | | | LVDEN | STBRST | WURST | STBMOD | LDOLP |
| rs | | | | | | | rw | rw | | | rw | rc_w1 | rc_w1 | rw | rw |

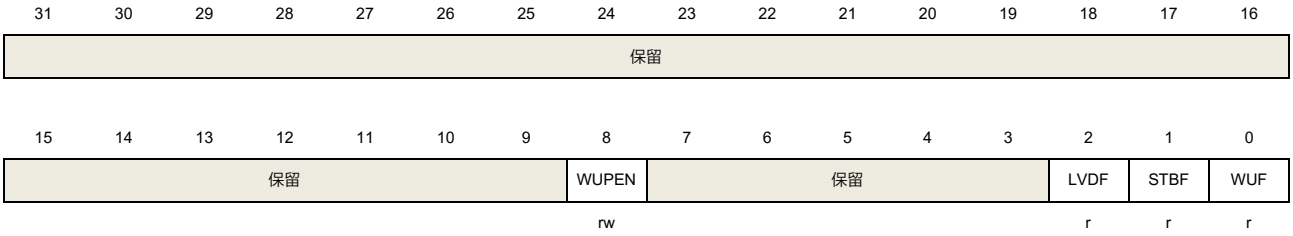
| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15:14 | LDOVS[1:0] | 选择LDO输出<br>在主PLL关闭时，这些位由软件配置。在主PLL使能后，LDOVS设置的值生效。如果主PLL关闭，LDO输出低电压模式被选中。<br>00：保留(LDO输出正常电压模式)<br>01：LDO输出正常电压模式<br>10：保留(LDO输出低电压模式)<br>11：LDO输出低电压模式 |
| 13:9 | 保留 | 必须保持复位值。 |
| 8 | BKPWEN | 备份域写使能<br>0：禁止对备份域寄存器的写访问<br>1：允许对备份域寄存器的写访问<br>复位之后，任何对备份域寄存器的写访问都将被禁止。如需对备份域寄存器做写访问，需先将该位置1。 |
| 7:5 | LVDT[2:0] | 低电压检测器阈值<br>000：2.1V<br>001：2.3V<br>010：2.4V<br>011：2.6V<br>100：2.7V<br>101：2.9V<br>110：3.0V |

| | | |
|---|---|---|
| | | 111：3.1V |
| 4 | LVDEN | 低电压检测器使能<br>0：关闭低电压检测器<br>1：开启低电压检测器 |
| 3 | STBRST | 待机标志复位<br>0：无影响<br>1：复位待机标志<br>读该位，始终返回0 |
| 2 | WURST | 唤醒标志复位<br>0：无影响<br>1：复位唤醒标志<br>读该位，始终返回0 |
| 1 | STBMOD | 待机模式<br>0：当Cortex®-M4进入SLEEPDEEP模式时，系统进入深度睡眠模式<br>1：当Cortex®-M4进入SLEEPDEEP模式时，系统进入待机模式 |
| 0 | LDOLP | LDO低功耗模式<br>0：当系统进入深度睡眠模式时，LDO仍正常工作<br>1：当系统进入深度睡眠模式时，LDO进入低功耗模式 |

### 3.4.2. 电源控制和状态寄存器(PMU_CS)

地址偏移：0x04
复位值：0x0000 0000 (从待机模式唤醒后不复位)

该寄存器可以按半字(16位)或字(32位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:9 | 保留 | 必须保持复位值。 |
| 8 | WUPEN | WKUP引脚唤醒使能<br>0：关闭WKUP引脚唤醒功能<br>1：开启WKUP引脚唤醒功能<br>如果WUPEN在进入待机模式之前置1，WKUP引脚的上升沿会将系统从待机模式唤醒。由于WKUP引脚为高电平有效，WKUP引脚内部被配置为输入下拉模式。当置位该控制位后，当输入为高的时候，将会触发一个唤醒事件。 |

| | | |
|---|---|---|
| 7:3 | 保留 | 必须保持复位值。 |
| 2 | LVDF | 低电压检测器状态标志<br>0：低电压事件没出现($V_{DD}$高于设定的LVD阈值)<br>1：低电压事件出现($V_{DD}$等于或低于LVD阈值)<br>**注意：**LVD功能在待机模式被禁用。 |
| 1 | STBF | 待机标志<br>0：设备没进入过待机模式<br>1：设备曾进入过待机模式<br>该位只能由POR/PDR或通过设置PMU_CTL寄存器的STBRST位来清零。 |
| 0 | WUF | 唤醒标志<br>0：没有收到唤醒事件<br>1：收到来自WKUP引脚或RTC闹钟事件。<br>该位只能由POR / PDR或通过设置PMU_CTL寄存器的WURST位来清零。 |

# 4. 备份寄存器(BKP)

## 4.1. 简介

位于备份域中的备份寄存器可在 $V_{DD}$ 电源关闭时由 $V_{BAT}$ 供电，备份寄存器有 42 个 16 位(84 字节)寄存器可用来存储并保护用户应用数据，从待机模式唤醒或系统复位也不会对这些寄存器造成影响。

此外，BKP 寄存器也可实现侵入检测和 RTC 校准功能。

在复位之后，任何对备份域寄存器的写操作都将被禁止，也就是说，备份寄存器和 RTC 不允许写访问。为使能对备份寄存器和 RTC 的写访问，首先通过设置 RCU_APB1EN 寄存器的 PMUEN 和 BKPIEN 位来打开电源和备份接口时钟，然后再通过设置 PMU_CTL 寄存器的 BKPWEN 位来使能对备份域中寄存器的写访问。

## 4.2. 主要特征特性

- 84字节备份寄存器用来在省电模式下保护数据。如果侵入事件发生，备份寄存器会被复位
- 侵入源(PC13)的有效电平可配置
- RTC时钟校准寄存器可提供RTC闹钟或秒输出选择，及设置校准值的功能
- 侵入控制状态寄存器(BKP_TPCS)可实现对侵入检测的中断或事件的控制

## 4.3. 功能说明

### 4.3.1. RTC 时钟校准

为提高RTC时钟精度，MCU提供时钟输出校准功能。RTC时钟或者RTC时钟经64分频后作为输出至PC13。通过设置BKP_OCTL寄存器中的COEN位来使能此功能。

校准值通过BKP_OCTL寄存器中的RCCV[6:0]设置，校准功能可实现以1000000/2^20ppm的比例减慢RTC时钟。

### 4.3.2. 侵入检测

MCU提供侵入检测功能以保护重要的用户数据，可通过设置BKP_TPCTL寄存器中的TPEN位来使能TAMPER引脚对应的功能。为防止侵入事件的丢失，边沿检测信号与TPEN位的逻辑与作为侵入检测信号的输入，因此在TAMPER引脚使能之前，侵入检测应该被配置。当侵入事件被检测到，对应的BKP_TPCS寄存器中的TEF位被置位。如果侵入中断被使能，侵入事件可以产生一个中断。任何侵入事件将会复位所有备份数据寄存器。

**注意：**当TPAL=0/1，如果TAMPER引脚在使能(通过设置TPEN位)之前已经为高/低，尽管TAMPER引脚上没有上升/下降沿信号，一个额外的侵入事件将会发生。

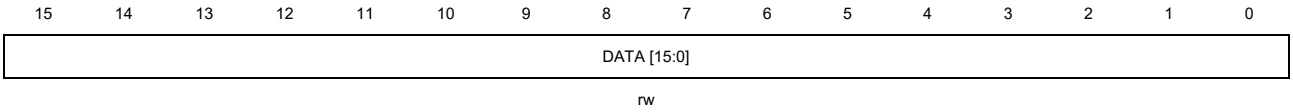
## 4.4. BKP 寄存器

BKP基地址：0x4000 6C00

### 4.4.1. 备份数据寄存器 (BKP_DATAx) (x= 0..41)

地址偏移：0x04到0x28，0x40到0xBC
复位值：0x0000

该寄存器可以按半字(16位)或字(32位)访问

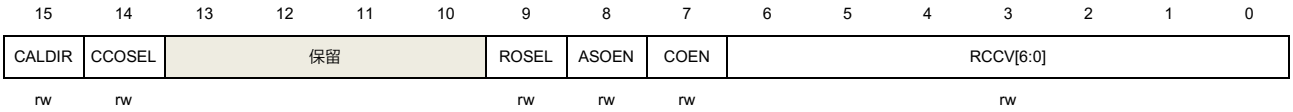



| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:0 | DATA[15:0] | 备份数据<br>这些位用来存储一般用户数据。即使从待机模式唤醒或系统复位或电源复位后，BKP_DATAx寄存器的内容仍旧不会丢失。 |

### 4.4.2. RTC 信号输出控制寄存器 (BKP_OCTL)

地址偏移：0x2C
复位值：0x0000

该寄存器可以按半字(16位)或字(32位)访问




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15 | CALDIR | RTC时钟校准方向<br>0：变慢<br>1：变快<br>该位只能被备份域复位清除。 |
| 14 | CCOSEL | RTC时钟输出选择<br>0：RTC时钟64分频<br>1：RTC时钟<br>该位只能被上电复位(POR)清除。 |
| 13:10 | 保留 | 必须保持复位值 |
| 9 | ROSEL | RTC输出选择<br>0：RTC输出为闹钟脉冲 |

| | | |
|---|---|---|
| | | 1：RTC输出为秒脉冲<br>该位只能被备份域复位清除。 |
| 8 | ASOEN | RTC闹钟或秒信号输出使能<br>0：RTC闹钟或秒信号输出禁止<br>1：RTC闹钟或秒信号输出使能<br>使能后，TAMPER引脚可作为RTC输出。<br>该位只能被备份域复位清除。 |
| 7 | COEN | RTC时钟校准输出使能<br>0：RTC时钟校准输出禁止<br>1：RTC时钟校准输出使能<br>使能后，TAMPER引脚输出RTC时钟或RTC时钟的64分频。ASOEN位优先于COEN位，当ASOEN位置位时，不管COEN置位与否，TAMPER引脚作为RTC闹钟或秒信号输出。<br>该位只能被上电复位(POR)清除。 |
| 6:0 | RCCV[6:0] | RTC时钟校准值<br>该值表示在每2^20个时钟脉冲内将有多少个时钟脉冲被忽略。<br>该位只能被备份域复位清除。 |

### 4.4.3. 侵入引脚控制寄存器 (BKP_TPCTL)

地址偏移：0x30
复位值：0x0000

该寄存器可以按半字(16位)或字(32位)访问

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

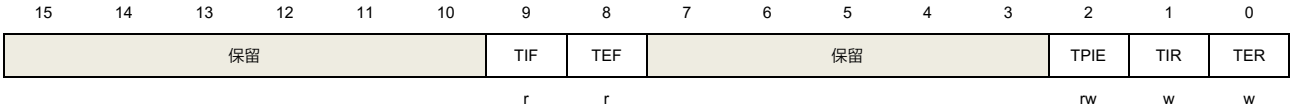
| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:2 | 保留 | 必须保持复位值 |
| 1 | TPAL | TAMPER引脚有效电平<br>0：TAMPER引脚高电平有效<br>1：TAMPER引脚低电平有效 |
| 0 | TPEN | TAMPER引脚使能<br>0：TAMPER引脚作为GPIO口使用<br>1： TAMPER引脚可实现备份复位功能。TAMPER引脚上的有效电平将复位BKP_DATAx寄存器中所有数据。 |

### 4.4.4. 侵入控制状态寄存器 (BKP_TPCS)

地址偏移：0x34

复位值：0x0000

该寄存器可以按半字(16位)或字(32位)访问




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:10 | 保留 | 必须保持复位值 |
| 9 | TIF | 侵入中断标志<br>0：没有侵入中断发生<br>1：有侵入中断发生<br>该位可通过TIR位置1或TPIE位置0来清零。 |
| 8 | TEF | 侵入事件标志<br>0：没有侵入事件发生<br>1：有侵入事件发生<br>该位可通过对TER为写1来清零。 |
| 7:3 | 保留 | 必须保持复位值 |
| 2 | TPIE | 侵入中断使能<br>0：侵入中断禁用<br>1：侵入中断使能<br>该位仅可通过系统复位或待机模式唤醒后复位。 |
| 1 | TIR | 侵入中断复位<br>0：不影响<br>1：复位TIF位<br>该位一直读为0。 |
| 0 | TER | 侵入事件复位<br>0：不影响<br>1：复位TEF位<br>该位一直读为0。 |

# 5. 复位和时钟单元(RCU)

## 5.1. 复位控制单元(RCTL)

### 5.1.1. 简介

复位控制包括三种控制方式：电源复位、系统复位和备份域复位。电源复位又称为冷复位，将复位除了备份域以外的所有系统。系统复位将复位除了SW-DP控制器和备份域之外的其余部分，包括处理器内核和外设IP。备份域复位将复位备份区域。复位能够被外部信号、内部事件和复位发生器触发。后续章节将介绍关于这些复位的详细信息。

### 5.1.2. 功能说明

#### 电源复位

当发生以下任一事件时，产生电源复位：上电/掉电复位(POR / PDR复位)，从待机模式中返回后由内部复位发生器产生。电源复位复位所有的寄存器除了备份域。电源复位为低电平有效，当内部LDO电源基准准备好提供1.2V电压时，电源复位电平将变为无效。复位入口向量被固定在存储器映射的地址0x0000 0004。

#### 系统复位

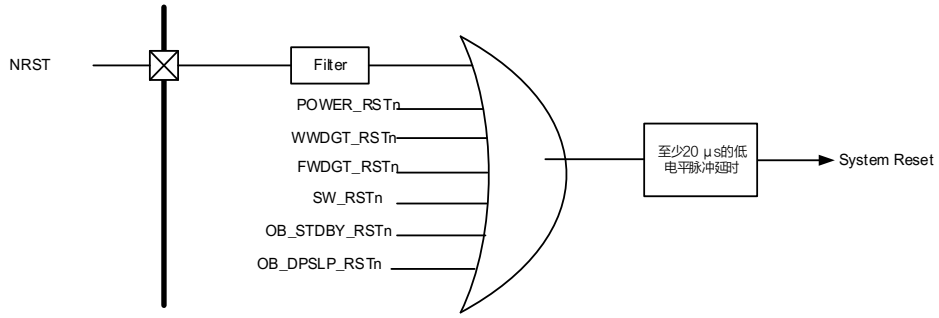
当发生以下任一事件时，产生一个系统复位：

- 上电复位(POWER_RSTn)
- 外部引脚复位(NRST)
- 窗口看门狗计数终止(WWDGT_RSTn)
- 独立看门狗计数终止(FWDGT_RSTn)
- Cortex®-M4的中断应用和复位控制寄存器中的SYSRESETREQ位置'1'(SW_RSTn)
- 用户选择字节寄存器nRST_STDBY设置为0，并且进入待机模式时将产生复位(OB_STDBY_RSTn)
- 用户选择字节寄存器nRST_DPSLP设置为0，并且进入深度睡眠模式时将产生复位(OB_DPSLP_RSTn)

系统复位将复位除了SW-DP控制器和备份域之外的其余部分，包括处理器内核和外设IP。

系统复位脉冲发生器保证每一个复位源(外部或内部)都能有至少20μs的低电平脉冲延时。

图5-1. 系统复位电路




### 备份域复位

以下事件之一发生时，产生备份域复位：1、设置备份域控制寄存器中的BKPRST位为'1'；2、备份域电源上电复位(在$V_{DD}$和$V_{BAT}$两者都掉电的前提下，$V_{DD}$或$V_{BAT}$上电)。

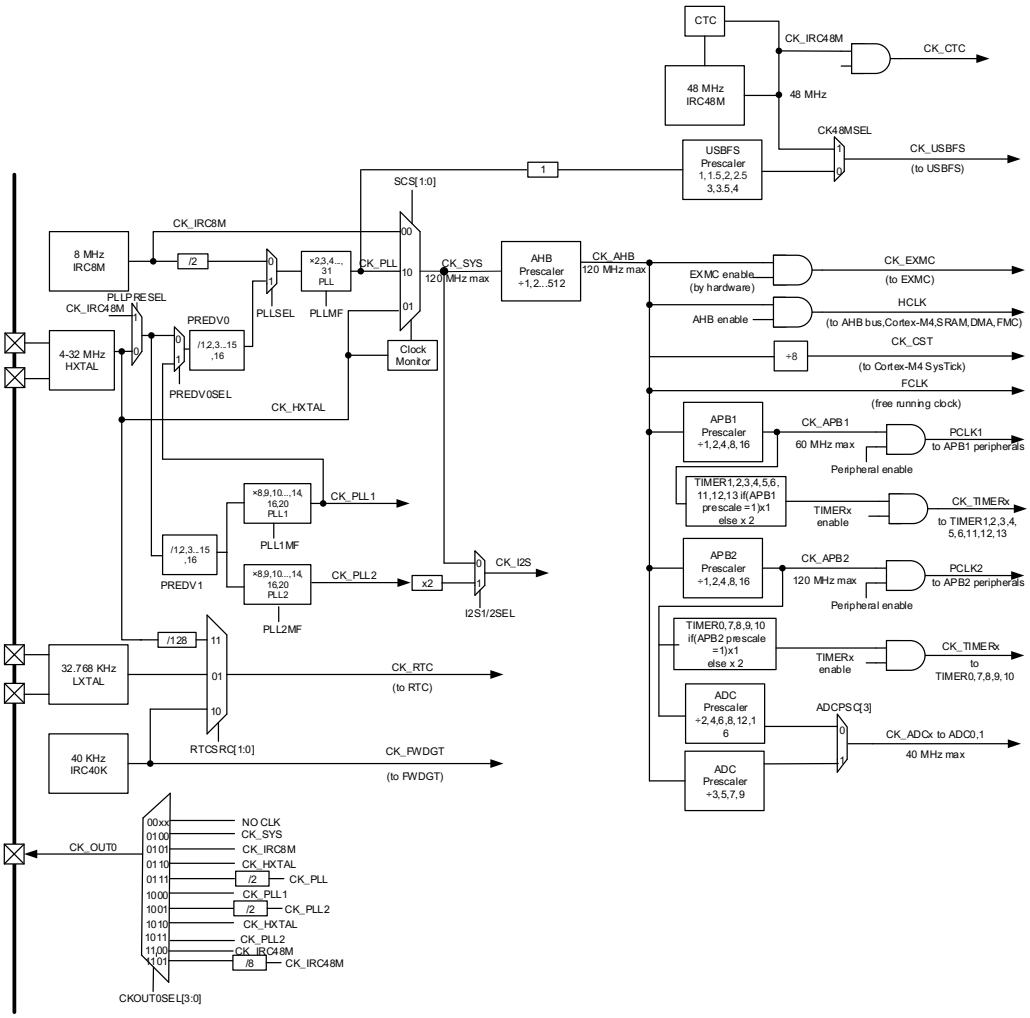
## 5.2. 时钟控制单元(CCTL)

### 5.2.1. 简介

时钟控制单元提供了一系列频率和时钟功能，包括一个内部8M RC振荡器时钟(IRC8M)、一个内部48M RC 振荡器时钟(IRC48M)、一个外部高速晶体振荡器时钟(HXTAL)、一个内部40K RC振荡器时钟(IRC40K)、一个外部低速晶体振荡器时钟(LXTAL)、三个锁相环(PLL)、一个HXTAL时钟监视器、时钟预分频器、时钟多路复用器和时钟门控电路。

AHB、APB和Cortex®-M4时钟都源自系统时钟(CK_SYS)，系统时钟的时钟源可以选择IRC8M、HXTAL或PLL。系统时钟的最大运行时钟频率可以达到120MHz。

**图5-2. 时钟树**




预分频器可以配置AHB、APB2和APB1域的时钟频率。AHB、APB2、APB1域的最高时钟频率分别为120MHz、120MHz、60MHz。RCU通过AHB时钟(HCLK)8分频后作为Cortex系统定时器(SysTick)的外部时钟。通过对SysTick控制和状态寄存器的设置，可选择上述时钟或AHB(HCLK)时钟作为SysTick时钟。

ADC时钟由APB2时钟经2、4、6、8、12、16分频或由AHB时钟经3、5、7、9分频获得，它们是通过设置RCU_CFG0和RCU_CFG1寄存器的ADCPSC位来选择。

TIMER时钟由CK_APB1和CK_APB2时钟分频获得，如果APBx(x=1，2)的分频系数不为1，则TIMER时钟为CK_APBx(x=1，2)的两倍，如果APBx(x=1，2)的分频系数为1，则TIMER时钟等于CK_APBx(x=1，2)。

USBFS的时钟由CK48M时钟提供。通过配置RCU_ADDCTL寄存器的CK48MSEL位可以选择CK_PLL时钟或IRC48M时钟做为CK48M的时钟源。

CTC时钟由IRC48M时钟提供，通过CTC单元，可以实现IRC48M时钟精度的自动调整。

I2S的时钟由CK_SYS或PLL2*2提供，通过配置RCU_CFG1寄存器的I2SxSEL来选择。

通过配置RCU_BDCTL寄存器的RTCSRC位，RTC时钟可以选择由LXTAL时钟、IRC40K时钟

或HXTAL时钟的128分频提供。RTC时钟选择HXTAL时钟的128分频做为时钟源后，当1.2V内核电压域掉电时，时钟将停止。RTC时钟选择IRC40K时钟做为时钟源后，当$V_{DD}$掉电时，时钟将停止。RTC时钟选择LXTAL时钟做为时钟源后，当$V_{DD}$和$V_{BAT}$都掉电时，时钟将停止。

当FWDGT启动时，FWDGT时钟被强制选择由IRC40K时钟做为时钟源。

### 5.2.2. 主要特征

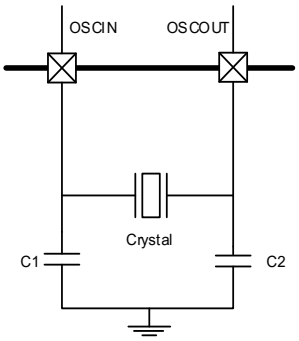
- 4到32MHz外部高速晶体振荡器(HXTAL)；
- 内部8MHz RC振荡器(IRC8M)；
- 内部48MHz RC振荡器；
- 32,768 Hz外部低速晶体振荡器(LXTAL)；
- 内部40KHz RC振荡器(IRC40K)；
- PLL时钟源可选HXTAL、IRC8M或IRC48M；
- HXTAL时钟监视器。

### 5.2.3. 功能说明

#### 外部高速晶体振荡时钟(HXTAL)

4到32M的外部高速晶体振荡器可为系统时钟提供更为精确时钟源。带有特定频率的晶体必须靠近两个HXTAL的引脚连接。和晶体连接的外部电阻和电容必须根据所选择的振荡器来调整。

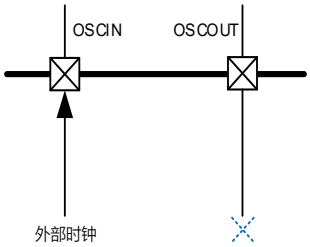
**图5-3. HXTAL时钟源**




HXTAL晶体振荡器可以通过设置控制寄存器RCU_CTL的HXTALEN位来启动或关闭，在控制寄存器RCU_CTL中的HXTALSTB位用来指示外部高速振荡器是否已稳定。在启动时，直到这一位被硬件置'1'，时钟才被释放出来。这个特定的延迟时间被称为振荡器的启动时间。当HXTAL时钟稳定后，如果在中断寄存器RCU_INT中的相应中断使能位HXTALSTBIE位被置'1'，将会产生相应中断。此时，HXTAL时钟可以被直接用作系统时钟源或者PLL输入时钟。

将控制寄存器RCU_CTL的HXTALBPS和HXTALEN位置'1'可以设置外部时钟旁路模式。CK_HXTAL等于驱动OSCIN管脚的外部时钟。旁路输入时，信号接至OSCIN，OSCOUT保持悬空状态，如*图5-4. 旁路模式下HXTAL时钟源*所示。此时，CK_HXTAL等于驱动OSCIN管脚的外部时钟。

**图5-4. 旁路模式下HXTAL时钟源**




### 内部 8M RC 振荡器时钟(IRC8M)

内部8MHz RC振荡器时钟，简称IRC8M时钟，拥有8MHz的固定频率，设备上电后CPU默认选择其做为系统时钟源。IRC8M RC振荡器能够在不需要任何外部器件的条件下为用户提供更低成本类型的时钟源。IRC8M RC振荡器可以通过设置控制寄存器(RCU_CTL)中的IRC8MEN位被启动和关闭。控制寄存器RCU_CTL中的IRC8MSTB位用来指示IRC8M内部RC振荡器是否稳定。IRC8M振荡器的启动时间比HXTAL晶体振荡器要更短。如果中断寄存器RCU_INT中的相应中断使能位IRC8MSTBIE被置'1'，在IRC8M稳定以后，将产生一个中断。IRC8M时钟也可用作系统时钟源或PLL输入时钟。

工厂会校准IRC8M时钟频率的精度，但是它的精度仍然比HXTAL时钟要差。用户可以根据需求、环境条件和成本决定选择哪个时钟作为系统时钟源。

如果HXTAL或者PLL被选择为系统时钟源，为了最大程度减小系统从深度睡眠模式恢复的时间，当系统从深度睡眠模式初始唤醒时，硬件会强制IRC8M时钟作为系统时钟。

### 内部 48M RC 振荡器时钟(IRC48M)

内部48MHz RC振荡器时钟，简称IRC48M时钟，拥有48MHz的固定频率，当使用USBFS模块时，IRC48M振荡器在不需要任何外部器件的条件下为用户提供了一种成本更低的时钟源选择。IRC48M RC振荡器可以通过设置RCU_ADDCTL寄存器中的IRC48MEN位被启动和关闭。RCU_ADDCTL寄存器中的IRC48MSTB位用来指示内部48MHz RC振荡器是否稳定。如果RCU_ADDINT寄存器中的相应中断使能位IRC48MSTBIE被置'1'，在IRC48M稳定以后，将产生一个中断。IRC48M时钟可做为USBFS的系统时钟。

工厂会校准IRC48M时钟频率的精度，但是它的精度仍然不够精准。因为USB模块需要的时钟频率必须满足48MHz(500ppm)。CTC单元提供了一种硬件自动执行动态调整的功能将IRC48M时钟调整到需要的频率。

### 锁相环(PLL)

内部有三个锁相环，PLL，PLL1，PLL2。

PLL可以通过设置RCU_CTL寄存器中的PLLEN位被启动和关闭。RCU_CTL寄存器中的PLLSTB位用来指示PLL时钟是否稳定。如果RCU_INT寄存器中的相应中断使能位PLLSTBIE被置'1'，在PLL稳定以后，将产生一个中断。

PLL1可以通过设置RCU_CTL寄存器中的PLL1EN位被启动和关闭。RCU_CTL寄存器中的PLL1STB位用来指示PLL1时钟是否稳定。如果RCU_INT寄存器中的相应中断使能位PLL1STBIE被置'1'，在PLL1稳定以后，将产生一个中断。

PLL2可以通过设置RCU_CTL寄存器中的PLL2EN位被启动和关闭。RCU_CTL寄存器中的PLL2STB位用来指示PLL2时钟是否稳定。如果RCU_INT寄存器中的相应中断使能位PLL2STBIE被置'1'，在PLL2稳定以后，将产生一个中断。

当进入深度睡眠模式或待机模式或者HXTAL监视器检测到时钟阻塞时(HXTAL做为锁相环的输入时钟)，三个PLL将被关闭。

### 外部低速晶体振荡器时钟(LXTAL)

LXTAL是一个频率为32.768kHz的外部低速晶体或陶瓷谐振器。它为实时时钟电路提供一个低功耗且高精准的时钟源。LXTAL振荡器可以通过设置备份域控制寄存器(RCU_BDCTL)中的LXTALEN位被启动和关闭。备份域控制寄存器RCU_BDCTL中的LXTALSTB位用来指示LXTAL时钟是否稳定。如果中断寄存器RCU_INT中的相应中断使能位LXTALSTBIE被置'1'，在LXTAL稳定以后，将产生一个中断。

将备份域控制寄存器RCU_BDCTL的LXTALBPS和LXTALEN位置'1'可以选择外部时钟旁路模式。CK_LXTAL与连到OSC32IN脚上外部时钟信号一致。

### 内部 40K RC 振荡器时钟(IRC40K)

IRC40K内部RC振荡器时钟担当一个低功耗时钟源的角色，不需要外部器件，它的时钟频率大约40kHz，为独立看门狗和实时时钟电路提供时钟。IRC40K RC振荡器可以通过设置复位源/时钟寄存器RCU_RSTSCK中的IRC40KEN位被启动和关闭。复位源/时钟寄存器RCU_RSTSCK中的IRC40KSTB位用来指示IRC40K时钟是否已稳定。如果复位源/时钟寄存器RCU_RSTSCK中的相应中断使能位IRC40KSTBIE被置'1'，在IRC40K稳定以后，将产生一个中断。

TIMER4_CH3可以捕获IRC40K的时钟，进而对RTC和FWDGT的计数器进行校准，详细的信息可以参考AFIO_PCF0寄存器的位TIMER4CH3_IREMAP。

### 系统时钟(CK_SYS)选择

系统复位后，IRC8M时钟默认做为CK_SYS的时钟源，改变配置寄存器0(RCU_CFG0)中的系统时钟变换位SCS可以切换系统时钟源为HXTAL或CK_PLL。当SCS的值被改变，系统时钟将使用原来的时钟源继续运行直到转换的目标时钟源稳定。当一个时钟源被直接或通过PLL间接作为系统时钟时，它将不能被停止。

### HXTAL 时钟监视器(CKM)

设置控制寄存器RCU_CTL中的HXTAL时钟监视使能位CKMEN，HXTAL可以使能时钟监视功能。该功能必须在HXTAL启动延迟完毕后使能，在HXTAL停止后禁止。一旦监测到HXTAL故障，HXTAL将自动被禁止，中断寄存器RCU_INT中的HXTAL时钟阻塞中断标志位CKMIF将被

置'1'，产生HXTAL故障事件。这个故障引发的中断和Cortex-M4的不可屏蔽中断NMI相连。如果HXTAL被选作系统，PLL或是RTC的时钟源，HXTAL故障将促使选择IRC8M为系统时钟源，PLL将被自动禁止，RTC的时钟源需要重新配置。

### 时钟输出能力

时钟输出功能输出从0.09375MHz到120MHz的时钟。通过设置时钟配置寄存器0(RCU_CFG0)中的CK_OUT0时钟源选择位域CKOUT0SEL能够选择不同的时钟信号。相应的GPIO引脚应该被配置成备用功能I/O(AFIO)模式来输出选择的时钟信号。

**表5-1. 时钟输出0的时钟源选择**

| 时钟输出 0 的时钟源选择位域 | 时钟源 |
|---|---|
| 00xx | NO CLK |
| 0100 | CK_SYS |
| 0101 | CK_IRC8M |
| 0110 | CK_HXTAL |
| 0111 | CK_PLL/2 |
| 1000 | CK_PLL1 |
| 1001 | CK_PLL2/2 |
| 1010 | CK_HXTAL |
| 1011 | CK_PLL2 |

### 电压控制

深度睡眠模式电压寄存器(RCU_DSV)中的DSLPVS[1:0]位域可以控制1.2V域在深度睡眠模式下的电压。

**表5-2. 深度睡眠模式下1.2V域电压选择**

| DSLPVS[1:0] | 深度睡眠模式电压(V) |
|---|---|
| 00 | 1.0 |
| 01 | 0.9 |
| 10 | 0.8 |
| 11 | 1.2 |

## 5.3. RCU 寄存器

RCU基地址：0x4002 1000

### 5.3.1. 控制寄存器(RCU_CTL)

地址偏移：0x00
复位值：0x0000 xx83 x表示未定义

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | PLL2STB | PLL2EN | PLL1STB | PLL1EN | PLLSTB | PLLEN | 保留 | | | | CKMEN | HXTALBPS | HXTALSTB | HXTALEN |
| | | r | rw | r | rw | r | rw | | | | | rw | rw | r | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IRC8MCALIB[7:0] | | | | | | | | IRC8MADJ[4:0] | | | | | 保留 | IRC8MSTB | IRC8MEN |
| r | | | | | | | | rw | | | | | | r | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:30 | 保留 | 必须保持复位值。 |
| 29 | PLL2STB | PLL2时钟稳定标志位<br>硬件置1来表示PLL2输出时钟是否稳定待用<br>0：PLL2未稳定<br>1：PLL2已稳定 |
| 28 | PLL2EN | PLL2使能<br>软件置位或复位，当PLL2时钟做为系统时钟时该位不能被复位。当进入深度睡眠或待机模式时由硬件复位<br>0：PLL2被关闭<br>1：PLL2被打开 |
| 27 | PLL1STB | PLL1时钟稳定标志位<br>硬件置1来表示PLL1输出时钟是否稳定待用<br>0：PLL1未稳定<br>1：PLL1已稳定 |
| 26 | PLL1EN | PLL1使能<br>软件置位或复位，当PLL1时钟做为系统时钟时该位不能被复位。当进入深度睡眠或待机模式时由硬件复位<br>0：PLL1被关闭<br>1：PLL1被打开 |
| 25 | PLLSTB | PLL时钟稳定标志位<br>硬件置1来表示PLL输出时钟是否稳定待用 |

| | | |
|---|---|---|
| | | 0：PLL未稳定<br>1：PLL已稳定 |
| 24 | PLLEN | PLL使能<br>软件置位或复位，当PLL时钟做为系统时钟时该位不能被复位。当进入深度睡眠或待机模式时由硬件复位<br>0：PLL被关闭<br>1：PLL被打开 |
| 23:20 | 保留 | 必须保持复位值 |
| 19 | CKMEN | HXTAL时钟监视器使能<br>0：禁止高速4 ~ 32 MHz晶体振荡器(HXTAL)时钟监视器<br>1：使能高速4 ~ 32 MHz晶体振荡器(HXTAL)时钟监视器<br>当硬件检测到HXTAL时钟被阻塞在低或高状态时，内部硬件自动切换系统时钟到IRC8M时钟。恢复原来系统时钟的方式有以下几种：外部复位，上电复位，软件清CKMIF位。<br>**注意：**使能HXTAL时钟监视器以后，硬件无视控制位IRC8MEN的状态，自动使能IRC8M时钟。 |
| 18 | HXTALBPS | 高速晶体振荡器(HXTAL)时钟旁路模式使能<br>只有在HXTALEN位为0时HXTALBPS位才可写<br>0：禁止HXTAL旁路模式<br>1：使能HXTAL旁路模式 HXTAL输出时钟等于输入时钟 |
| 17 | HXTALSTB | 高速晶体振荡器(HXTAL)时钟稳定标志位<br>硬件置'1'来指示HXTAL振荡器时钟是否稳定待用<br>0：HXTAL振荡器未稳定<br>1：HXTAL振荡器已稳定 |
| 16 | HXTALEN | 高速晶体振荡器(HXTAL)使能<br>软件置位或复位，如果HXTAL时钟作为系统时钟或者当PLL时钟做为系统时钟时，其做为PLL的输入时钟，该位不能被复位。进入深度睡眠或待机模式时硬件自动复位<br>0：高速4 ~ 32 MHz晶体振荡器被关闭<br>1：高速4 ~ 32 MHz晶体振荡器被打开 |
| 15:8 | IRC8MCALIB[7:0] | 内部8MHz RC振荡器校准值寄存器<br>上电时自动加载这些位 |
| 7:3 | IRC8MADJ[4:0] | 内部8MHz RC振荡器时钟调整值<br>这些位由软件置位，最终调整值为IRC8MADJ[4:0]位域的当前值加上IRC8MCALIB[7:0]位域的值。最终调整值应该调整IRC8M到8MHz +/- 1% |
| 2 | 保留 | 必须保持复位值 |
| 1 | IRC8MSTB | IRC8M内部8MHz RC振荡器稳定标志位<br>硬件置'1'来指示IRC8M振荡器时钟是否稳定待用<br>0：IRC8M振荡器未稳定<br>1：IRC8M振荡器已稳定 |

| | | |
|---|---|---|
| 0 | IRC8MEN | 内部8MHz RC振荡器使能<br>软件置位或复位，如果IRC8M时钟做为系统时钟时，该位不能被复位。当从深度睡眠或待机模式返回，或当CKMEN置位同时用作系统时钟的HXTAL振荡器发生故障时，该位由硬件置1来启动IRC8M振荡器。<br>0：内部8MHz RC振荡器被关闭<br>1：内部8MHz RC振荡器被打开 |

### 5.3.2. 时钟配置寄存器 0(RCU_CFG0)

地址偏移：0x04

复位值：0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| USBFSPSC[2] | 保留 | PLLMF[4] | ADCPSC[2] | CKOUT0SEL[3:0] | | | | USBFSPSC[1:0] | | PLLMF[3:0] | | | | PREDV0_LSB | PLLSEL |
| rw | | rw | rw | rw | | | | rw | | rw | | | | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADCPSC[1:0] | | APB2PSC[2:0] | | | APB1PSC[2:0] | | | AHBPSC[3:0] | | | | SCSS[1:0] | | SCS[1:0] | |
| rw | | rw | | | rw | | | rw | | | | r | | rw | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | USBFSPSC[2] | USBFSPSC的第2位<br>参考寄存器RCU_CFG0的22到23位 |
| 30 | 保留 | 必须保持复位值。 |
| 29 | PLLMF[4] | PLLMF的第4位<br>参考寄存器RCU_CFG0的18到21位 |
| 28 | ADCPSC[2] | ADCPSC的第2位<br>参考寄存器RCU_CFG0的14到15位 |
| 27:24 | CKOUT0SEL[3:0] | CKOUT0时钟源选择<br>由软件置位或清零<br>00xx：无时钟输出<br>0100：选择系统时钟CK_SYS<br>0101：选择内部8M RC振荡器时钟<br>0110：选择高速晶体振荡器时钟 (HXTAL)<br>0111：选择(CK_PLL / 2)时钟<br>1000：选择CK_PLL1时钟<br>1001：选择(CK_PLL2 / 2)时钟<br>1010：选择CK_HXTAL时钟<br>1011：选择CK_PLL2时钟<br>1100：选择CK_IRC48M时钟<br>1101：选择(CK_IRC48M / 8)时钟 |

| | | |
|---|---|---|
| 23:22 | USBFSPSC[1:0] | USBFS的时钟分频系数<br>由软件置位或清零。USBFS的时钟必须为48MHz，当USBFS时钟使能的时候，这些位无法修改<br>000：CK_USBFS = CK_PLL / 1.5<br>001：CK_USBFS = CK_PLL<br>010：CK_USBFS = CK_PLL / 2.5<br>011：CK_USBFS = CK_PLL / 2<br>100：CK_USBFS = CK_PLL / 3<br>101：CK_USBFS = CK_PLL / 3.5<br>11x：CK_USBFS = CK_PLL / 4 |
| 21:18 | PLLMF[3:0] | PLL时钟倍频因子<br>与寄存器RCU_CFG0的29位共同构成倍频因子，由软件置位或清零<br>**注意：** PLL输出时钟频率不能超过120MHz<br>00000：(PLL源时钟 x 2)<br>00001：(PLL源时钟 x 3)<br>00010：(PLL源时钟 x 4)<br>00011：(PLL源时钟 x 5)<br>00100：(PLL源时钟 x 6)<br>00101：(PLL源时钟 x 7)<br>00110：(PLL源时钟 x 8)<br>00111：(PLL源时钟 x 9)<br>01000：(PLL源时钟 x 10)<br>01001：(PLL源时钟 x 11)<br>01010：(PLL源时钟 x 12)<br>01011：(PLL源时钟 x 13)<br>01100：(PLL源时钟 x 14)<br>01101：(PLL源时钟 x 6.5)<br>01110：(PLL源时钟 x 16)<br>01111：(PLL源时钟 x 16)<br>10000：(PLL源时钟 x 17)<br>10001：(PLL源时钟 x 18)<br>10010：(PLL源时钟 x 19)<br>10011：(PLL源时钟 x 20)<br>10100：(PLL源时钟 x 21)<br>10101：(PLL源时钟 x 22)<br>10110：(PLL源时钟 x 23)<br>10111：(PLL源时钟 x 24)<br>11000：(PLL源时钟 x 25)<br>11001：(PLL源时钟 x 26)<br>11010：(PLL源时钟 x 27)<br>11011：(PLL源时钟 x 28)<br>11100：(PLL源时钟 x 29)<br>11101：(PLL源时钟 x 30)<br>11110：(PLL源时钟 x 31) |

| | | |
|---|---|---|
| | | 11111：(PLL源时钟 x 31) |
| 17 | PREDV0_LSB | PREDV0分频因子的最低位<br>与寄存器RCU_CFG1位PREDV0第0位相同，通过寄存器RCU_CFG1来改变PREDV0的值，此位也会一同改。当PREDV0的第1到3位未修改时，此位决定PREDV0的输入时钟是否二分频。 |
| 16 | PLLSEL | PLL时钟源选择<br>由软件置位或复位，控制PLL时钟源<br>0：(IRC8M / 2)被选择为PLL时钟的时钟源<br>1：HXTAL时钟或者IRC48M时钟(寄存器RCU_CFG1位PLLPRESEL决定)被选择为PLL时钟的时钟源 |
| 15:14 | ADCPSC[1:0] | ADC的时钟分频系数<br>与寄存器RCU_CFG0的28位，寄存器RCU_CFG1的29位共同构成分频因子，由软件置位或清零<br>0000：CK_ADC = CK_APB2 / 2<br>0001：CK_ADC = CK_APB2 / 4<br>0010：CK_ADC = CK_APB2 / 6<br>0011：CK_ADC = CK_APB2 / 8<br>0100：CK_ADC = CK_APB2 / 2<br>0101：CK_ADC = CK_APB2 / 12<br>0110：CK_ADC = CK_APB2 / 8<br>0111：CK_ADC = CK_APB2 / 16<br>1x00：CK_ADC = CK_AHB / 3<br>1x01：CK_ADC = CK_AHB / 5<br>1x10：CK_ADC = CK_AHB / 7<br>1x11：CK_ADC = CK_AHB / 9 |
| 13:11 | APB2PSC[2:0] | APB2预分频选择<br>由软件置位或清零，控制APB2时钟分频因子<br>0xx：CK_APB2 = CK_AHB<br>100：CK_APB2 = CK_AHB / 2<br>101：CK_APB2 = CK_AHB / 4<br>110：CK_APB2 = CK_AHB / 8<br>111：CK_APB2 = CK_AHB / 16 |
| 10:8 | APB1PSC[2:0] | APB1预分频选择<br>由软件置位或清零，控制APB1时钟分频因子.<br>0xx：CK_APB1 = CK_AHB<br>100：CK_APB1 = CK_AHB / 2<br>101：CK_APB1 = CK_AHB / 4<br>110：CK_APB1 = CK_AHB / 8<br>111：CK_APB1 = CK_AHB / 16 |
| 7:4 | AHBPSC[3:0] | AHB预分频选择<br>由软件置位或清零，控制AHB时钟分频因子 |

0xxx：CK_AHB = CK_SYS
1000：CK_AHB = CK_SYS / 2
1001：CK_AHB = CK_SYS / 4
1010：CK_AHB = CK_SYS / 8
1011：CK_AHB = CK_SYS / 16
1100：CK_AHB = CK_SYS / 64
1101：CK_AHB = CK_SYS / 128
1110：CK_AHB = CK_SYS / 256
1111：CK_AHB = CK_SYS / 512

| | | |
|---|---|---|
| 3:2 | SCSS[1:0] | 系统时钟选择状态<br>由硬件置位或清零，标识当前系统时钟的时钟源<br>00：选择CK_IRC8M时钟作为CK_SYS时钟源<br>01：选择CK_HXTAL时钟作为CK_SYS时钟源<br>10：选择CK_PLL时钟作为CK_SYS时钟源<br>11：保留 |
| 1:0 | SCS[1:0] | 系统时钟选择<br>由软件配置选择系统时钟源。由于CK_SYS的改变存在固有的延迟，因此软件应当读SCSS位来确保时钟源切换是否结束。在从深度睡眠或待机模式中返回时，以及当HXTAL直接或间接作为系统时钟同时HXTAL时钟监视器检测到HXTAL故障时，强制选择IRC8M作为系统时钟。<br>00：选择CK_IRC8M时钟作为CK_SYS时钟源<br>01：选择CK_HXTAL时钟作为CK_SYS时钟源<br>10：选择CK_PLL时钟作为CK_SYS时钟源<br>11：保留 |

### 5.3.3. 时钟中断寄存器(RCU_INT)

地址偏移：0x08
复位值：0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | CKMIC | PLL2 STBIC | PLL1 STBIC | PLL STBIC | HXTAL STBIC | IRC8M STBIC | LXTAL STBIC | IRC40K STBIC |
| | | | | | | | | w | w | w | w | w | w | w | w |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | PLL2 STBIE | PLL1 STBIE | PLL STBIE | HXTAL STBIE | IRC8M STBIE | LXTAL STBIE | IRC40K STBIE | CKMIF | PLL2 STBIF | PLL1 STBIF | PLL STBIF | HXTAL STBIF | IRC8M STBIF | LXTAL STBIF | IRC40K STBIF |
| | rw | rw | rw | rw | rw | rw | rw | r | r | r | r | r | r | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:24 | 保留 | 必须保持复位值 |

| | | |
|---|---|---|
| 23 | CKMIC | HXTAL时钟阻塞中断清零<br>软件写1复位CKMIF标志位.<br>0：不复位CKMIF标志位<br>1：复位CKMIF标志位 |
| 22 | PLL2STBIC | PLL2时钟稳定中断清零<br>软件写1复位PLL2STBIF标志位<br>0：不复位PLL2STBIF标志位<br>1：复位PLL2STBIF标志位 |
| 21 | PLL1STBIC | PLL1时钟稳定中断清零<br>软件写1复位PLL1STBIF标志位<br>0：不复位PLL1STBIF标志位<br>1：复位PLL1STBIF标志位 |
| 20 | PLLSTBIC | PLL时钟稳定中断清零<br>软件写1复位PLLSTBIF标志位<br>0：不复位PLLSTBIF标志位<br>1：复位PLLSTBIF标志位 |
| 19 | HXTALSTBIC | HXTAL时钟稳定中断清零<br>软件写1复位HXTALSTBIF标志位<br>0：不复位HXTALSTBIF标志位<br>1：复位HXTALSTBIF标志位 |
| 18 | IRC8MSTBIC | IRC8M时钟稳定中断清零<br>软件写1复位IRC8MSTBIF标志位<br>0：不复位IRC8MSTBIF标志位<br>1：复位IRC8MSTBIF标志位 |
| 17 | LXTALSTBIC | LXTAL时钟稳定中断清零<br>软件写1复位LXTALSTBIF标志位<br>0：不复位LXTALSTBIF标志位<br>1：复位LXTALSTBIF标志位 |
| 16 | IRC40KSTBIC | IRC40K时钟稳定中断清零<br>软件写1复位IRC40KSTBIF标志位<br>0：不复位IRC40KSTBIF标志位<br>1：复位IRC40KSTBIF标志位 |
| 15 | 保留 | 必须保持复位值 |
| 14 | PLL2STBIE | PLL2时钟稳定中断使能<br>软件置位和复位来使能/禁止PLL2时钟稳定中断<br>0：禁止PLL2时钟稳定中断<br>1：使能PLL2时钟稳定中断 |
| 13 | PLL1STBIE | PLL1时钟稳定中断使能<br>软件置位和复位来使能/禁止PLL1时钟稳定中断 |

| | | |
|---|---|---|
| | | 0：禁止PLL1时钟稳定中断<br>1：使能PLL1时钟稳定中断 |
| 12 | PLLSTBIE | PLL时钟稳定中断使能<br>软件置位和复位来使能/禁止PLL时钟稳定中断<br>0：禁止PLL时钟稳定中断<br>1：使能PLL时钟稳定中断 |
| 11 | HXTALSTBIE | HXTAL时钟稳定中断使能<br>软件置位和复位来使能/禁止HXTAL时钟稳定中断<br>0：禁止HXTAL时钟稳定中断<br>1：使能HXTAL时钟稳定中断 |
| 10 | IRC8MSTBIE | IRC8M时钟稳定中断使能<br>软件置位和复位来使能/禁止IRC8M时钟稳定中断<br>0：禁止IRC8M时钟稳定中断<br>1：使能IRC8M时钟稳定中断 |
| 9 | LXTALSTBIE | LXTAL时钟稳定中断使能<br>软件置位和复位来使能/禁止LXTAL时钟稳定中断<br>0：禁止LXTAL时钟稳定中断<br>1：使能LXTAL时钟稳定中断 |
| 8 | IRC40KSTBIE | IRC40K时钟稳定中断使能<br>软件置位和复位来使能/禁止IRC40K时钟稳定中断<br>0：禁止IRC40K时钟稳定中断<br>1：使能IRC40K时钟稳定中断 |
| 7 | CKMIF | HXTAL时钟阻塞中断标志位<br>当HXTAL时钟被阻塞时由硬件置位.<br>软件置位CKMIC位时清除该位<br>0：时钟正常运行<br>1：HXTAL时钟阻塞 |
| 6 | PLL2STBIF | PLL2时钟稳定中断标志位<br>当PLL时钟稳定且PLL2STBIE位被置1时由硬件置1<br>软件置位PLL2STBIC位时清除该位<br>0：无PLL2时钟稳定中断产生<br>1：产生PLL2时钟稳定中断 |
| 5 | PLL1STBIF | PLL1时钟稳定中断标志位<br>当PLL时钟稳定且PLL1STBIE位被置1时由硬件置1<br>软件置位PLL1STBIC位时清除该位<br>0：无PLL1时钟稳定中断产生<br>1：产生PLL1时钟稳定中断 |
| 4 | PLLSTBIF | PLL时钟稳定中断标志位<br>当PLL时钟稳定且PLLSTBIE位被置1时由硬件置1<br>软件置位PLLSTBIC位时清除该位 |

| | | |
|---|---|---|
| | | 0：无PLL时钟稳定中断产生<br>1：产生PLL时钟稳定中断 |
| 3 | HXTALSTBIF | HXTAL时钟稳定中断标志位<br>当高速4~32 MHz晶体振荡器时钟稳定且HXTALSTBIE位被置1时由硬件置1<br>软件置位HXTALSTBIC位时清除该位<br>0：无HXTAL时钟稳定中断产生<br>1：产生HXTAL时钟稳定中断 |
| 2 | IRC8MSTBIF | IRC8M时钟稳定中断标志位<br>当内部8MHz RC振荡器时钟稳定且IRC8MSTBIE位被置1时由硬件置1<br>软件置位IRC8MSTBIC位时清除该位<br>0：无IRC8M时钟稳定中断产生<br>1：产生IRC8M时钟稳定中断 |
| 1 | LXTALSTBIF | LXTAL时钟稳定中断标志位<br>当低速晶体振荡器时钟稳定且LXTALSTBIE位被置1时由硬件置1<br>软件置位LXTALSTBIC位时清除该位<br>0：无LXTAL时钟稳定中断产生<br>1：产生LXTAL时钟稳定中断 |
| 0 | IRC40KSTBIF | IRC40K时钟稳定中断标志位<br>当内部40kHz RC振荡器时钟稳定且IRC40KSTBIE位被置1时由硬件置1<br>软件置位IRC40KSTBIC位时清除该位<br>0：无IRC40K时钟稳定中断产生<br>1：产生IRC40K时钟稳定中断 |

### 5.3.4. APB2 复位寄存器(RCU_APB2RST)

地址偏移：0x0C

复位值：0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | | | TIMER10RST | TIMER9RST | TIMER8RST | 保留 | | |
| | | | | | | | | | | rw | rw | rw | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | USART0RST | TIMER7RST | SPI0RST | TIMER0RST | ADC1RST | ADC0RST | 保留 | | PERST | PDRST | PCRST | PBRST | PARST | 保留 | AFRST |
| | 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 | TIMER10RST | TIMER10复位 |

| | | |
|---|---|---|
| | | 由软件置位或复位<br>0：无作用<br>1：复位TIMER10 |
| 20 | TIMER9RST | TIMER9复位<br>由软件置位或复位<br>0：无作用<br>1：复位TIMER9 |
| 19 | TIMER8RST | TIMER8复位<br>由软件置位或复位<br>0：无作用<br>1：复位TIMER8 |
| 18:15 | 保留 | 必须保持复位值。 |
| 14 | USART0RST | USART0复位<br>由软件置位或复位<br>0：无作用<br>1：复位USART0 |
| 13 | TIMER7RST | TIMER7复位<br>由软件置位或复位<br>0：无作用<br>1：复位TIMER7 |
| 12 | SPI0RST | SPI0复位<br>由软件置位或复位<br>0：无作用<br>1：复位SPI0 |
| 11 | TIMER0RST | TIMER0复位<br>由软件置位或复位<br>0：无作用<br>1：复位TIMER0 |
| 10 | ADC1RST | ADC1复位<br>由软件置位或复位<br>0：无作用<br>1：复位所有ADC1 |
| 9 | ADC0RST | ADC0复位<br>由软件置位或复位<br>0：无作用<br>1：复位所有ADC0 |
| 8:7 | 保留 | 必须保持复位值。 |
| 6 | PERST | GPIO端口E复位<br>由软件置位或复位 |

| | | |
|---|---|---|
| | | 0：无作用<br>1：复位GPIO端口E |
| 5 | PDRST | GPIO端口D复位<br>由软件置位或复位<br>0：无作用<br>1：复位GPIO端口D |
| 4 | PCRST | GPIO端口C复位<br>由软件置位或复位<br>0：无作用<br>1：复位GPIO端口C |
| 3 | PBRST | GPIO端口B复位<br>由软件置位或复位<br>0：无作用<br>1：复位GPIO端口B |
| 2 | PARST | GPIO端口A复位<br>由软件置位或复位<br>0：无作用<br>1：复位GPIO端口A |
| 1 | 保留 | 必须保持复位值。 |
| 0 | AFRST | 复用功能I/O复位<br>由软件置位或复位<br>0：无作用<br>1：复位复用功能I/O |

### 5.3.5. APB1 复位寄存器(RCU_APB1RST)

地址偏移：0x10
复位值：0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | DACRST | PMURST | BKPIRST | 保留 | | | | I2C1RST | I2C0RST | UART4RST | UART3RST | USART2RST | USART1RST | 保留 |
| | | rw | rw | rw | | | | | rw | rw | rw | rw | rw | rw | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SPI2RST | SPI1RST | 保留 | | WWDGTRST | 保留 | | TIMER13RST | TIMER12RST | TIMER11RST | TIMER6RST | TIMER5RST | TIMER4RST | TIMER3RST | TIMER2RST | TIMER1RST |
| 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 | DACRST | DAC复位<br>由软件置位或复位<br>0：无作用<br>1：复位DAC |
| 28 | PMURST | PMU复位<br>由软件置位或复位<br>0：无作用<br>1：复位PMU |
| 27 | BKPIRST | BKPI复位<br>由软件置位或复位<br>0：无作用<br>1：复位BKP |
| 26:23 | 保留 | 必须保持复位值。 |
| 22 | I2C1RST | I2C1复位<br>由软件置位或复位<br>0：无作用<br>1：复位I2C1 |
| 21 | I2C0RST | I2C0复位<br>由软件置位或复位<br>0：无作用<br>1：复位I2C0 |
| 20 | UART4RST | UART4复位<br>由软件置位或复位<br>0：无作用<br>1：复位UART4 |
| 19 | UART3RST | UART3复位<br>由软件置位或复位<br>0：无作用<br>1：复位UART3 |
| 18 | USART2RST | USART2复位<br>由软件置位或复位<br>0：无作用<br>1：复位USART2 |
| 17 | USART1RST | USART1复位<br>由软件置位或复位<br>0：无作用<br>1：复位USART1 |
| 16 | 保留 | 必须保持复位值。 |
| 15 | SPI2RST | SPI2复位 |

| | | |
|---|---|---|
| | | 由软件置位或复位<br>0：无作用<br>1：复位SPI2 |
| 14 | SPI1RST | SPI1复位<br>由软件置位或复位<br>0：无作用<br>1：复位SPI1 |
| 13:12 | 保留 | 必须保持复位值。 |
| 11 | WWDGTRST | WWDGT复位<br>由软件置位或复位<br>0：无作用<br>1：复位WWDGT |
| 10:9 | 保留 | 必须保持复位值。 |
| 8 | TIMER13RST | TIMER13复位<br>由软件置位或复位<br>0：无作用<br>1：复位TIMER13 |
| 7 | TIMER12RST | TIMER12复位<br>由软件置位或复位<br>0：无作用<br>1：复位TIMER12 |
| 6 | TIMER11RST | TIMER11复位<br>由软件置位或复位<br>0：无作用<br>1：复位TIMER11 |
| 5 | TIMER6RST | TIMER6复位<br>由软件置位或复位<br>0：无作用<br>1：复位TIMER6 |
| 4 | TIMER5RST | TIMER5复位<br>由软件置位或复位<br>0：无作用<br>1：复位TIMER5 |
| 3 | TIMER4RST | TIMER4复位<br>由软件置位或复位<br>0：无作用<br>1：复位TIMER4 |
| 2 | TIMER3RST | TIMER3复位<br>由软件置位或复位 |

| | | |
|---|---|---|
| | | 0：无作用<br>1：复位TIMER3 |
| 1 | TIMER2RST | TIMER2复位<br>由软件置位或复位<br>0：无作用<br>1：复位TIMER2 |
| 0 | TIMER1RST | TIMER1复位<br>由软件置位或复位<br>0：无作用<br>1：复位TIMER1 |

### 5.3.6. AHB 使能寄存器(RCU_AHBEN)

地址偏移：0x14
复位值：0x0000 0014

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | USBFSEN | 保留 | | | EXMCEN | 保留 | CRCEN | 保留 | FMCSPEN | 保留 | SRAMSPEN | DMA1EN | DMA0EN |
| | | | 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 | USBFSEN | USBFS时钟使能<br>由软件置位或复位<br>0：关闭USBFS时钟<br>1：开启USBFS时钟 |
| 11:9 | 保留 | 必须保持复位值。 |
| 8 | EXMCEN | EXMC时钟使能<br>由软件置位或复位<br>0：关闭EXMC时钟<br>1：开启EXMC时钟 |
| 7 | 保留 | 必须保持复位值。 |
| 6 | CRCEN | CRC时钟使能<br>由软件置位或复位<br>0：关闭CRC时钟 |

| | | |
|---|---|---|
| | | 1：开启CRC时钟 |
| 5 | 保留 | 必须保持复位值。 |
| 4 | FMCSPEN | 在睡眠模式下FMC时钟使能<br>由软件置位或复位<br>0：在睡眠模式下关闭FMC时钟<br>1：在睡眠模式下开启FMC时钟 |
| 3 | 保留 | 必须保持复位值。 |
| 2 | SRAMSPEN | 在睡眠模式下SRAM时钟使能<br>由软件置位或复位<br>0：在睡眠模式下关闭SRAM时钟<br>1：在睡眠模式下开启SRAM时钟 |
| 1 | DMA1EN | DMA1时钟使能<br>由软件置位或复位<br>0：关闭DMA1时钟<br>1：开启DMA1时钟 |
| 0 | DMA0EN | DMA0时钟使能<br>由软件置位或复位<br>0：关闭DMA0时钟<br>1：开启DMA0时钟 |

### 5.3.7. APB2 使能寄存器(RCU_APB2EN)

地址偏移：0x18
复位值：0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | | | TIMER10EN | TIMER9EN | TIMER8EN | 保留 | | |
| | | | | | | | | | | rw | rw | rw | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | USART0EN | TIMER7EN | SPI0EN | TIMER0EN | ADC1EN | ADC0EN | 保留 | | PEEN | PDEN | PCEN | PBEN | PAEN | 保留 | AFEN |
| 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 | TIMER10EN | TIMER10时钟使能<br>由软件置位或复位<br>0：关闭TIMER10时钟<br>1：开启TIMER10时钟 |

| | | |
|---|---|---|
| 20 | TIMER9EN | TIMER9时钟使能<br>由软件置位或复位<br>0：关闭TIMER9时钟<br>1：开启TIMER9时钟 |
| 19 | TIMER8EN | TIMER8时钟使能<br>由软件置位或复位<br>0：关闭TIMER8时钟<br>1：开启TIMER8时钟 |
| 18:15 | 保留 | 必须保持复位值。 |
| 14 | USART0EN | USART0时钟使能<br>由软件置位或复位<br>0：关闭USART0时钟<br>1：开启USART0时钟 |
| 13 | TIMER7EN | TIMER7复位<br>由软件置位或复位<br>0：无作用<br>1：复位TIMER7 |
| 12 | SPI0EN | SPI0复位<br>由软件置位或复位<br>0：无作用<br>1：复位SPI0 |
| 11 | TIMER0EN | TIMER0复位<br>由软件置位或复位<br>0：无作用<br>1：复位TIMER0 |
| 10 | ADC1EN | ADC1时钟使能<br>由软件置位或复位<br>0：关闭ADC1时钟<br>1：开启ADC1时钟 |
| 9 | ADC0EN | ADC0时钟使能<br>由软件置位或复位<br>0：关闭ADC0时钟<br>1：开启ADC0时钟 |
| 8:7 | 保留 | 必须保持复位值。 |
| 6 | PEEN | GPIO端口E时钟使能<br>由软件置位或复位<br>0：关闭GPIO端口E时钟<br>1：开启GPIO端口E时钟 |
| 5 | PDEN | GPIO端口D时钟使能 |

由软件置位或复位
0：关闭GPIO端口D时钟
1：开启GPIO端口D时钟

| | | |
|---|---|---|
| 4 | PCEN | GPIO端口C时钟使能<br>由软件置位或复位<br>0：关闭GPIO端口C时钟<br>1：开启GPIO端口C时钟 |
| 3 | PBEN | GPIO端口B时钟使能<br>由软件置位或复位<br>0：关闭GPIO端口B时钟<br>1：开启GPIO端口B时钟 |
| 2 | PAEN | GPIO端口A时钟使能<br>由软件置位或复位<br>0：关闭GPIO端口A时钟<br>1：开启GPIO端口A时钟 |
| 1 | 保留 | 必须保持复位值。 |
| 0 | AFEN | 复用功能IO时钟使能<br>由软件置位或复位<br>0：关闭复用功能IO时钟<br>1：开启复用功能IO时钟 |

## 5.3.8. APB1 使能寄存器(RCU_APB1EN)

地址偏移：0x1C
复位值：0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | DACEN | PMUEN | BKPIEN | 保留 | | | | I2C1EN | I2C0EN | UART4EN | UART3EN | USART2EN | USART1EN | 保留 |
| | | rw | rw | rw | | | | | rw | rw | rw | rw | rw | rw | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SPI2EN | SPI1EN | 保留 | | WWDGTEN | 保留 | | TIMER13EN | TIMER12EN | TIMER11EN | TIMER6EN | TIMER5EN | TIMER4EN | TIMER3EN | TIMER2EN | TIMER1EN |
| 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 | DACEN | DAC时钟使能<br>由软件置位或复位<br>0：关闭DAC时钟 |

| | | |
|---|---|---|
| | | 1：开启DAC时钟 |
| 28 | PMUEN | PMU时钟使能<br>由软件置位或复位<br>0：关闭PMU时钟<br>1：开启PMU时钟 |
| 27 | BKPIEN | BKP时钟使能<br>由软件置位或复位<br>0：关闭BKP时钟<br>1：开启BKP时钟 |
| 26:23 | 保留 | 必须保持复位值。 |
| 22 | I2C1EN | I2C1时钟使能<br>由软件置位或复位<br>0：关闭I2C1时钟<br>1：开启I2C1时钟 |
| 21 | I2C0EN | I2C0时钟使能<br>由软件置位或复位<br>0：关闭I2C0时钟<br>1：开启I2C0时钟 |
| 20 | UART4EN | UART4时钟使能<br>由软件置位或复位<br>0：关闭UART4时钟<br>1：开启UART4时钟 |
| 19 | UART3EN | UART3时钟使能<br>由软件置位或复位<br>0：关闭UART3时钟<br>1：开启UART3时钟 |
| 18 | USART2EN | USART2时钟使能<br>由软件置位或复位<br>0：关闭USART2时钟<br>1：开启USART2时钟 |
| 17 | USART1EN | USART1时钟使能<br>由软件置位或复位<br>0：关闭USART1时钟<br>1：开启USART1时钟 |
| 16 | 保留 | 必须保持复位值。 |
| 15 | SPI2EN | SPI2时钟使能<br>由软件置位或复位<br>0：关闭SPI2时钟 |

| | | |
|---|---|---|
| | | 1：开启SPI2时钟 |
| 14 | SPI1EN | SPI1时钟使能<br>由软件置位或复位<br>0：关闭SPI1时钟<br>1：开启SPI1时钟 |
| 13:12 | 保留 | 必须保持复位值。 |
| 11 | WWDGTEN | WWDGT时钟使能<br>由软件置位或复位<br>0：关闭WWDGT时钟<br>1：开启WWDGT时钟 |
| 10:9 | 保留 | 必须保持复位值。 |
| 8 | TIMER13EN | TIMER13时钟使能<br>由软件置位或复位<br>0：关闭TIMER13时钟<br>1：开启TIMER13时钟 |
| 7 | TIMER12EN | TIMER12时钟使能<br>由软件置位或复位<br>0：关闭TIMER12时钟<br>1：开启TIMER12时钟 |
| 6 | TIMER11EN | TIMER11时钟使能<br>由软件置位或复位<br>0：关闭TIMER11时钟<br>1：开启TIMER11时钟 |
| 5 | TIMER6EN | TIMER6时钟使能<br>由软件置位或复位<br>0：关闭TIMER6时钟<br>1：开启TIMER6时钟 |
| 4 | TIMER5EN | TIMER5时钟使能<br>由软件置位或复位<br>0：关闭TIMER5时钟<br>1：开启TIMER5时钟 |
| 3 | TIMER4EN | TIMER4时钟使能<br>由软件置位或复位<br>0：关闭TIMER4时钟<br>1：开启TIMER4时钟 |
| 2 | TIMER3EN | TIMER3时钟使能<br>由软件置位或复位<br>0：关闭TIMER3时钟 |

| | | |
|---|---|---|
| | | 1：开启TIMER3时钟 |
| 1 | TIMER2EN | TIMER2时钟使能<br>由软件置位或复位<br>0：关闭TIMER2时钟<br>1：开启TIMER2时钟 |
| 0 | TIMER1EN | TIMER1时钟使能<br>由软件置位或复位<br>0：关闭TIMER1时钟<br>1：开启TIMER1时钟 |

### 5.3.9. 备份域控制寄存器(RCU_BDCTL)

地址偏移：0x20
复位值：0x0000 0018，只能由备份域复位进行复位

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问。

**注意：**备份域控制寄存器(RCU_BDCTL)的LXTALEN、LXTALBPS、RTCSRC和RTCEN位仅在备份域复位后才清0。只有在电源控制寄存器(PMU_CTL)中的BKPWEN位置1后才能对这些位进行改动。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:17 | 保留 | 必须保持复位值。 |
| 16 | BKPRST | 备份域复位<br>由软件置位或复位<br>0：无作用<br>1：复位备份域 |
| 15 | RTCEN | RTC时钟使能<br>由软件置位或复位<br>0：关闭RTC时钟<br>1：开启RTC时钟 |
| 14:10 | 保留 | 必须保持复位值。 |
| 9:8 | RTCSRC[1:0] | RTC时钟源选择<br>由软件置位或清零来控制RTC的时钟源。一旦RTC的时钟源选择后，除了将备份域复位否则时钟源不能被改变。 |

| | | |
|---|---|---|
| | | 00：没有时钟<br>01：选择CK_LXTAL时钟作为RTC的时钟源<br>10：选择CK_IRC40K时钟作为RTC的时钟源<br>11：选择CK_HXTAL / 128时钟作为RTC的时钟源 |
| 7:5 | 保留 | 必须保持复位值。 |
| 4:3 | LXTALDRI[1:0] | LXTAL驱动能力<br>由软件置位或复位。当备份域复位时将复位该值<br>00：弱驱动能力<br>01：中低驱动能力<br>10：中高驱动能力<br>11：强驱动能力（复位后的缺省值）<br>**注意：** LXTALDRI位在旁路模式下无效 |
| 2 | LXTALBPS | LXTAL旁路模式使能<br>由软件置位或复位<br>0：禁止LXTAL旁路模式<br>1：使能LXTAL旁路模式 |
| 1 | LXTALSTB | 低速晶体振荡器稳定标志位<br>硬件置'1'来指示LXTAL振荡器时钟是否稳定待用<br>0：LXTAL未稳定<br>1：LXTAL已稳定 |
| 0 | LXTALEN | LXTAL时钟使能<br>由软件置位或复位<br>0：关闭LXTAL时钟<br>1：使能LXTAL时钟 |

## 5.3.10. 复位源/时钟寄存器(RCU_RSTSCK)

地址偏移：0x24

复位值：0x0C00 0000，所有复位标志位仅在电源复位时被清零，RSTFC/IRC40KEN在系统复位时被清零。

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LP RSTF | WWDGT RSTF | FWDGT RSTF | SW RSTF | POR RSTF | EP RSTF | 保留 | RSTFC | 保留 | | | | | | | |
| r | r | r | r | r | r | | rw | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | | | | | | | | IRC40K STB | IRC40KE N |
| | | | | | | | | | | | | | | r | rw |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 31 | LPRSTF | 低功耗复位标志位<br>深度睡眠/待机复位发生时由硬件置位<br>向RSTFC位写1来清除该位<br>0：无低功耗管理复位发生<br>1：发生低功耗管理复位 |
| 30 | WWDGTRSTF | 窗口看门狗定时器复位标志位<br>窗口看门狗定时器复位发生时由硬件置1<br>向RSTFC位写1来清除该位<br>0：无窗口看门狗复位发生<br>1：发生窗口看门狗复位 |
| 29 | FWDGTRSTF | 独立看门狗定时器复位标志位<br>独立看门狗复位发生时由硬件置1<br>向RSTFC位写1来清除该位<br>0：无独立看门狗定时器复位发生<br>1：发生独立看门狗定时器复位 |
| 28 | SWRSTF | 软件复位标志位<br>软件复位发生时由硬件置1<br>向RSTFC位写1来清除该位<br>0：无软件复位发生<br>1：发生软件复位 |
| 27 | PORRSTF | 电源复位标志位<br>电源复位发生时由硬件置1<br>向RSTFC位写1来清除该位<br>0：无电源复位发生<br>1：发生电源复位 |
| 26 | EPRSTF | 外部引脚复位标志位<br>外部引脚复位发生时由硬件置1<br>向RSTFC位写1来清除该位<br>0：无外部引脚复位发生<br>1：发生外部引脚复位 |
| 25 | 保留 | 必须保持复位值。 |
| 24 | RSTFC | 清除复位标志位<br>由软件置1来清除所有复位标志位<br>0：无作用<br>1：清除所有复位标志位 |
| 23:2 | 保留 | 必须保持复位值。 |
| 1 | IRC40KSTB | IRC40K时钟稳定标志位<br>该位由硬件置1指示IRC40K输出时钟是否稳定待用<br>0：IRC40K时钟未稳定 |

| | | |
|---|---|---|
| | | 1：IRC40K已稳定 |
| 0 | IRC40KEN | IRC40K使能<br>由软件置位和复位<br>0：关闭IRC40K时钟<br>1：开启IRC40K时钟 |

### 5.3.11. AHB 复位寄存器(RCU_AHBRST)

地址偏移：0x28
复位值：0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问。

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:13 | 保留 | 必须保持复位值。 |
| 12 | USBFSRST | USBFS复位<br>由软件置位或复位<br>0：无作用<br>1：复位USBFS |
| 11:0 | 保留 | 必须保持复位值。 |

### 5.3.12. 时钟配置寄存器 1(RCU_CFG1)

地址偏移：0x2C
复位值：0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | PLLREPSEL | ADCPSC[3] | 保留 | | | | | | | | | | I2S2SEL | I2S1SEL | PREDV0SEL |
| | rw | rw | | | | | | | | | | | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PLL2MF[3:0] | | | | PLL1MF[3:0] | | | | PREDV1[3:0] | | | | PREDV0[3:0] | | | |
| rw | | | | rw | | | | rw | | | | rw | | | |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 31 | 保留 | 必须保持复位值。 |
| 30 | PLLPRESEL | PLL时钟源预选择<br>由软件置位或复位，控制PLL时钟源<br>0：HXTAL被选择为PLL时钟的时钟源<br>1：CK_IRC48M被选择为PLL时钟的时钟源 |
| 29 | ADCPSC[3] | ADCPSC的第3位<br>参考寄存器RCU_CFG0的14到15位 |
| 28:19 | 保留 | 必须保持复位值。 |
| 18 | I2S2SEL | I2S2时钟源选择<br>由软件置位或复位，控制I2S2时钟源<br>0：系统时钟被选择为I2S2时钟的时钟源<br>1：(CK_PLL2 x 2) 被选择为I2S2时钟的时钟源 |
| 17 | I2S1SEL | I2S1时钟源选择<br>由软件置位或复位，控制I2S1时钟源<br>0：系统时钟被选择为I2S1时钟的时钟源<br>1：(CK_PLL2 x 2) 被选择为I2S1时钟的时钟源 |
| 16 | PREDV0SEL | PREDV0时钟源选择<br>由软件置位或复位<br>0：HXTAL或IRC48M被选择为PREDV0的时钟源<br>1：CK_PLL1被选择为PREDV0的时钟源 |
| 15:12 | PLL2MF[3:0] | PLL2时钟倍频因子<br>倍频因子，由软件置位或清零<br>00xx：保留<br>010x：保留<br>0110：(PLL2源时钟 x 8)<br>0111：(PLL2源时钟 x 9)<br>1000：(PLL2源时钟 x 10)<br>1001：(PLL2源时钟 x 11)<br>1010：(PLL2源时钟 x 12)<br>1011：(PLL2源时钟 x 13)<br>1100：(PLL2源时钟 x 14)<br>1101：保留<br>1110：(PLL2源时钟 x 16)<br>1111：(PLL2源时钟 x 20) |
| 11:8 | PLL1MF[3:0] | PLL1时钟倍频因子<br>由软件置位或清零<br>00xx：保留<br>010x：保留 |

0110：(PLL1源时钟 x 8)
0111：(PLL1源时钟 x 9)
1000：(PLL1源时钟 x 10)
1001：(PLL1源时钟 x 11)
1010：(PLL1源时钟 x 12)
1011：(PLL1源时钟 x 13)
1100：(PLL1源时钟 x 14)
1101：保留
1110：(PLL1源时钟 x 16)
1111：(PLL1源时钟 x 20)

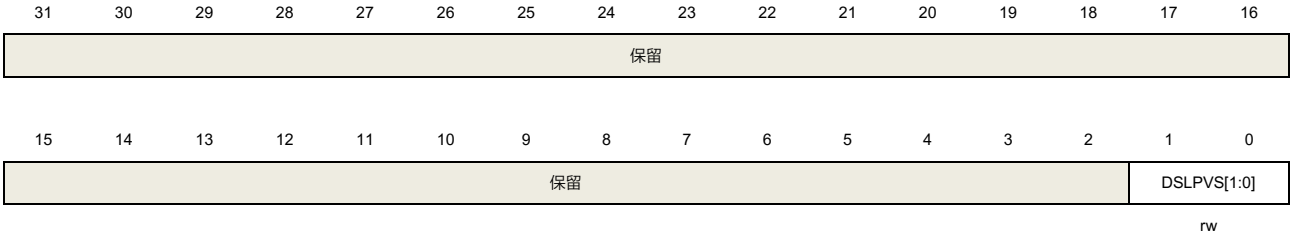
| | | |
|---|---|---|
| 7:4 | PREDV1[3:0] | PREDV1分频因子<br>由软件置位或清零，PLL1和PLL2未使能时，可以修改这些位<br>0000：PREDV1输入源时钟未分频<br>0001：PREDV1输入源时钟2分频<br>0010：PREDV1输入源时钟3分频<br>0011：PREDV1输入源时钟4分频<br>0100：PREDV1输入源时钟5分频<br>0101：PREDV1输入源时钟6分频<br>0110：PREDV1输入源时钟7分频<br>0111：PREDV1输入源时钟8分频<br>1000：PREDV1输入源时钟9分频<br>1001：PREDV1输入源时钟10分频<br>1010：PREDV1输入源时钟11分频<br>1011：PREDV1输入源时钟12分频<br>1100：PREDV1输入源时钟13分频<br>1101：PREDV1输入源时钟14分频<br>1110：PREDV1输入源时钟15分频<br>1111：PREDV1输入源时钟16分频 |
| 3:0 | PREDV0[3:0] | PREDV0分频因子<br>由软件置位或清零，PLL未使能时，可以修改这些位<br>**注意：**PREDV0的第0位与RCU_CFG0寄存器的17位相同，修改RCU_CFG0寄存器的17位，PREDV0的第0位也会进行相同的修改<br>0000：PREDV0输入源时钟未分频<br>0001：PREDV0输入源时钟2分频<br>0010：PREDV0输入源时钟3分频<br>0011：PREDV0输入源时钟4分频<br>0100：PREDV0输入源时钟5分频<br>0101：PREDV0输入源时钟6分频<br>0110：PREDV0输入源时钟7分频<br>0111：PREDV0输入源时钟8分频<br>1000：PREDV0输入源时钟9分频<br>1001：PREDV0输入源时钟10分频<br>1010：PREDV0输入源时钟11分频 |

1011：PREDV0输入源时钟12分频
1100：PREDV0输入源时钟13分频
1101：PREDV0输入源时钟14分频
1110：PREDV0输入源时钟15分频
1111：PREDV0输入源时钟16分频

### 5.3.13. 深度睡眠模式电压寄存器(RCU_DSV)

地址偏移：0x34
复位值：0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问。

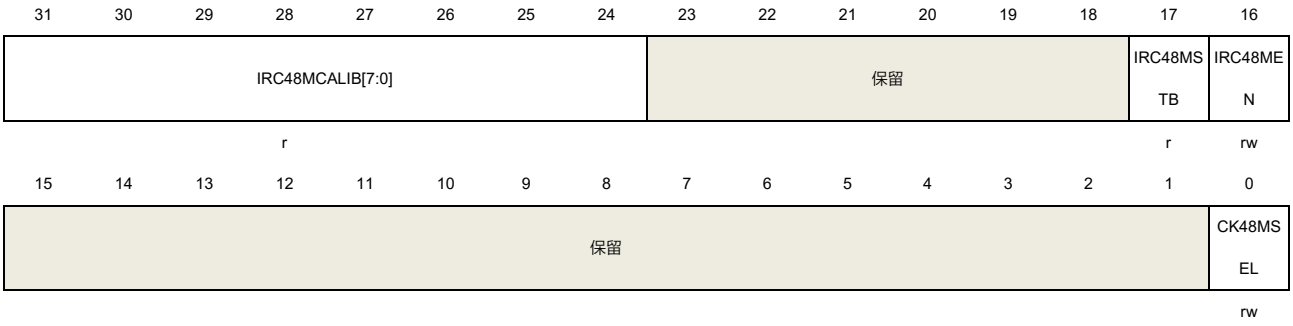



| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:2 | 保留 | 必须保持复位值 |
| 1:0 | DSLPVS[1:0] | 深度睡眠模式电压选择<br>由软件置位和清零这些位<br>000：在深度睡眠模式下内核电压为1.0V<br>001：在深度睡眠模式下内核电压为0.9V<br>010：在深度睡眠模式下内核电压为0.8V<br>011：在深度睡眠模式下内核电压为1.2V |

### 5.3.14. 附加时钟控制寄存器(RCU_ADDCTL)

地址偏移：0xC0
复位值：0x8000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|

| | | |
|---|---|---|
| 31:24 | IRC48MCALIB [7:0] | 内部48MHz RC振荡器校准值寄存器<br>上电时自动加载这些位 |
| 23:18 | 保留 | 必须保持复位值。 |
| 17 | IRC48MSTB | 内部48MHz RC振荡器时钟稳定标志位<br>硬件置'1'来指示IRC48M振荡器时钟是否稳定待用<br>0：IRC48M未稳定<br>1：IRC48M已稳定 |
| 16 | IRC48MEN | 内部48MHz RC 振荡器使能<br>由软件置位和复位。当进入深度睡眠或待机模式后由硬件复位<br>0：关闭IRC48M时钟<br>1：打开IRC48M时钟 |
| 15:1 | 保留 | 必须保持复位值。 |
| 0 | CK48MSEL | 48MHz时钟源选择<br>由软件置位和复位。该位用于选择IRC48M时钟或PLL48M时钟作为CK48M时钟源。<br>CK48M时钟用于：<br>0：不选择IRC48M时钟(使用CK_PLL/USBFSPSC时钟)<br>1：选择IRC48M时钟 |

## 5.3.15. 附加时钟中断寄存器(RCU_ADDINT)

地址偏移：0xCC
复位值：0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | | IRC48MSTBIC | 保留 | | | | | |
| | | | | | | | | | w | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | IRC48MSTBIE | 保留 | | | | | | | IRC48MSTBIF | 保留 | | | | | |
| | rw | | | | | | | | r | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:23 | 保留 | 必须保持复位值。 |
| 22 | IRC48MSTBIC | 内部48 MHz RC振荡器稳定中断清零<br>软件写1复位IRC48MSTBIF标志位<br>0：不复位IRC48MSTBIF标志位<br>1：复位IRC48MSTBIF标志位 |
| 21:15 | 保留 | 必须保持复位值。 |
| 14 | IRC48MSTBIE | 内部48 MHz RC振荡器稳定中断使能 |

| | | |
|---|---|---|
| | | 由软件置位和复位来使能/禁止IRC48M时钟稳定中断<br>0：禁止IRC48M时钟稳定中断<br>1：使能IRC48M时钟稳定中断 |
| 13:7 | 保留 | 必须保持复位值。 |
| 6 | IRC48MSTBIF | IRC48M时钟稳定中断标志位<br>当内部48 MHz RC振荡器时钟稳定且IRC48MSTBIE位被置1时由硬件置1<br>软件置位IRC48MSTBIC位时清除该位<br>0：无IRC48M时钟稳定中断产生<br>1：产生IRC48M时钟稳定中断 |
| 5:0 | 保留 | 必须保持复位值。 |

## 5.3.16. APB1 附加复位寄存器(RCU_ADDAPB1RST)

地址偏移：0xE0
复位值：0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问。

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:28 | 保留 | 必须保持复位值。 |
| 27 | CTCRST | CTC复位<br>由软件置位或复位<br>0：无作用<br>1：复位CTC |
| 26:0 | 保留 | 必须保持复位值。 |

## 5.3.17. APB1 附加使能寄存器(RCU_ADDAPB1EN)

地址偏移：0xE4
复位值：0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问。

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:28 | 保留 | 必须保持复位值。 |
| 27 | CTCEN | CTC时钟使能<br>由软件置位或复位<br>0：关闭CTC时钟<br>1：开启CTC时钟 |
| 26:0 | 保留 | 必须保持复位值。 |

# 6. 时钟校准控制器(CTC)

## 6.1. 简介

时钟校准控制器(CTC)采用硬件的方式，自动校准内部48MHz RC晶振(IRC48M)。CTC模块基于外部高精度的参考信号源来校准IRC48M的时钟频率，通过自动的或手动的调整校准值，以得到一个精准的IRC48M时钟。

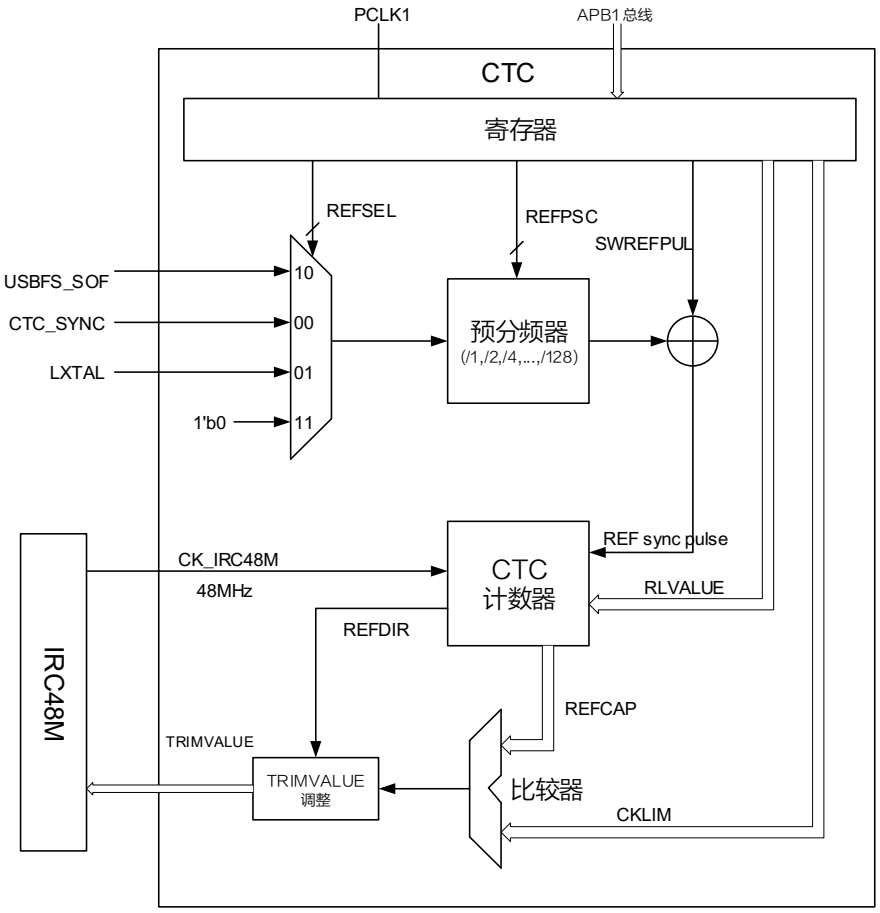
## 6.2. 主要特征

- 两个外部参考信号源：GPIO(CTC_SYNC)，LXTAL时钟；
- 提供软件参考同步脉冲；
- 硬件自动校准，无需软件操作；
- 具有参考信号源捕获和重载功能的16位校准计数器；
- 用于频率评估和自动校准的8位时钟校准基值；
- 标志位和中断，用于指示时钟校准的状态：校准成功状态(CKOKIF)，警告状态(CKWARNIF)和错误状态(ERRIF)。

## 6.3. 功能描述

CTC模块的内部结构图如*错误!书签自引用无效。*。

图 6-1. CTC 简介




### 6.3.1. 参考同步脉冲发生器

首先，通过设置CTC_CTL1寄存器中的REFSEL位来选择参考信号源：GPIO(CTC_SYNC)或者LXTAL时钟输出。

然后，可以通过设置CTC_CTL1寄存器中的REFPOL位来配置参考信号源同步时的信号极性，并且通过设置CTC_CTL1寄存器中的REFPSC位进行分频从而产生一个合适的同步时钟频率。

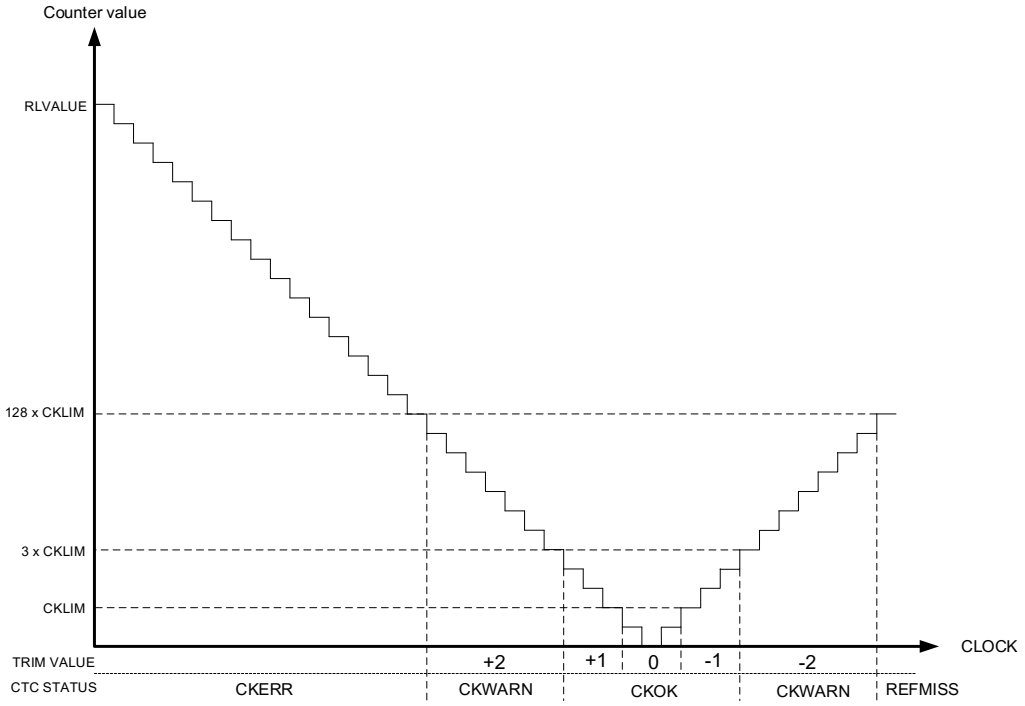
如果需要使用软件参考脉冲信号，则需要设置CTC_CTL0寄存器中的SWREFPUL位为1。软件参考脉冲信号与外部参考脉冲信号最后进行逻辑或操作。

### 6.3.2. CTC 校准计数器

CTC时钟校准计数器由CK_IRC48M提供时钟。在CTC_CTL0寄存器中的CNTEN置位后，当检测到第一个参考同步脉冲信号，计数器开始从RLVALUE值(RLVALUE在CTC_CTL1寄存器中定义)向下计数。每次检测到参考同步脉冲信号时，计数器重载RLVALUE值，同时重新开始向下计数。如果始终检测不到参考同步脉冲信号，计数器会向下计数到零，然后再向上计数到128 x CKLIM(CKLIM在CTC_CTL1中定义)，最后停止，直到检测到下一个参考同步脉冲信号。一旦检测到参考同步脉冲信号，当前CTC校准计数器的计数值被捕获存入CTC_STAT寄存器中的

REFCAP位，同时，当前计数器的计数方向被存入CTC_STAT寄存器中的REFDIR位。详细内容如*错误!未找到引用源。*所示。

**图 6-2. CTC 校准计数器**




## 6.3.3. 频率评估和自动校准过程

当参考同步脉冲信号出现时，时钟频率评估功能开始执行。如果参考同步脉冲信号出现在计数器向下计数的过程中，说明当前时钟频率比期望时钟频率(频率为48M)慢，需要增大CTC_CTL0中的TRIMVALUE值(时钟校准值)。如果参考同步脉冲信号出现在计数器向上计数的过程中，说明当前时钟频率比期望时钟频率快，需要减小TRIMVALUE值。CTC_STAT寄存器中的CKOKIF位，CKWARNIF位，CKERR位和REFMISS位反映了频率评估的状态。

如果CTC_CTL0中的AUTOTRIM(硬件自动校准模式)位置1，硬件自动校准模式使能。在这个模式中，如果参考同步脉冲信号出现在计数器向下计数的过程中，说明当前时钟频率比期望时钟频率慢，CTC_CTL0中的TRIMVALUE值会自动增大，来提高当前的时钟频率。反之，如果参考同步脉冲信号出现在计数器向上计数的过程中，说明当前时钟频率比期望时钟频率快，TRIMVALUE值会自动减小，从而减小当前的时钟频率。

- Counter < CKLIM时，检测到参考同步脉冲信号；

  CTC_STAT中的CKOKIF位(时钟校准成功标志位)被置位，同时，如果CTC_CTL0中的CKOKIE位(时钟校准完成中断使能位)置1，将会产生一个中断。
  如果CTC_CTL0中的AUTOTRIM置1，CTC_CTL0中的TRIMVALUE值不变。

- CKLIM <= Counter < 3 x CKLIM时，检测到参考同步脉冲信号；

  CTC_STAT中的CKOKIF位被置位，同时，如果CTC_CTL0中的CKOKIE位置1，将会产生一个中断。

如果CTC_CTL0中的AUTOTRIM位置1，在计数器向下计数过程中，CTC_CTL0中的TRIMVALUE值将加1，而在向上计数过程中将减1。

- 3 x CKLIM <= Counter < 128 x CKLIM时，检测到参考同步脉冲信号；

  CTC_STAT中的CKWARNIF位(时钟校准警告中断位)被置位，同时，如果CTC_CTL0中的CKWARNIE位(时钟校准警告中断使能位)置1，将会产生一个中断。
  如果CTC_CTL0中的AUTOTRIM位置1，在计数器向下计数过程中，CTC_CTL0中的TRIMVALUE值将加2，而在向上计数过程中将减2。

- Counter >= 128 x CKLIM，计数器在向下计数过程中，检测到参考同步脉冲信号；

  CTC_STAT中的CKERR位(时钟校准错误位)被置位，同时，如果CTC_CTL0中的ERRIE位(错误中断使能位)置1，将会产生一个中断。
  CTC_CTL0中的TRIMVALUE值不变。

- Counter = 128 x CKLIM，计数器在向上计数过程中；

  CTC_STAT中的REFMISS位(参考同步脉冲丢失位)被置位，同时，如果CTC_CTL0中的ERRIE位置1，将会产生一个中断。
  CTC_CTL0中的TRIMVALUE值不变。

如果CTC_CTL0中的TRIMVALUE的校准值大于63，将会发生上溢事件，同时，若TRIMVALUE的校准值小于0，将会发生下溢事件。TRIMVALUE的取值范围为0~63(上溢事件发生时，TRIMVALUE值为63；下溢事件发生时，TRIMVALUE值为0)。然后，CTC_STAT中的TRIMERR位(校准值错误位)将会被置位，如果CTC_CTL0中的ERRIE位置1，将会产生一个中断。

### 6.3.4. 软件编程指南

CTC_CTL1中RLVALUE位和CKLIM位是时钟频率评估和硬件自动校准的关键。它们的数值由期望时钟的频率(IRC48M：48 MHz)和参考同步脉冲信号的频率计算得到。理想状态是参考同步脉冲信号在CTC计数器计数到零时出现，所以RLVALUE的值为：

$$RLVALUE=(F_{clock}/ F_{REF})-1 \quad (6\text{-}1)$$

CKLIM的值由用户根据时钟的精度来设置，一般建议设为步长的一半，所以CKLIM的值为：

$$CKLIM=(F_{clock}/ F_{REF}) \times 0.12\% /2 \quad (6\text{-}2)$$

典型的步长值是0.12%，$F_{clock}$是期望时钟的频率(IRC48M)，$F_{REF}$是参考同步脉冲信号的频率。

## 6.4. CTC 寄存器

CTC基地址：0x4000 C800

### 6.4.1. 控制寄存器 0(CTC_CTL0)

地址偏移：0x00
复位值：0x0000 2000

该寄存器只能按字(32 位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | TRIMVALUE[5:0] | | | | | | SWREFPUL | AUTOTRIM | CNTEN | 保留 | EREFIE | ERRIE | CKWARNIE | CKOKIE |
| | | rw | | | | | | w | rw | rw | | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:14 | 保留 | 必须保持复位值。 |
| 13:8 | TRIMVALUE[5:0] | IRC48M校准值<br>当 CTC_CTL0 中的 AUTOTRIM 值为 0 时，该位由软件置位和清除，该模式用于软件校准过程。<br>当 CTC_CTL0 中的 AUTOTRIM 值为 1 时，该位只读，由硬件自动修改，该模式用于硬件校准过程。<br>TRIMVALUE 的中间值是 32，当 TRIMVALUE 值加 1 时，IRC48M 时钟频率增加大约 57KHz。当 TRIMVALUE 值减 1 时，IRC48M 时钟频率的减少大约 57KHz。 |
| 7 | SWREFPUL | 软件生成参考同步脉冲信号<br>该位由软件置位，并为CTC计数器提供一个参考同步脉冲信号。该位由硬件自动清除，读操作时返回0。<br>0：没有影响<br>1：软件产生一个同步参考脉冲信号 |
| 6 | AUTOTRIM | 硬件自动校准模式<br>该位由软件置位或清除。当该位置1时，硬件自动校准模式使能，通过硬件不断的自动修改CTC_CTL0中的TRIMVALUE值，直到IRC48M的时钟频率达到48MHz。<br>0：禁止硬件自动校准模式<br>1：使能硬件自动校准模式 |
| 5 | CNTEN | CTC计数器使能<br>该位由软件置位或清除，用于使能或禁止CTC计数器。当该位置1时，不能修改CTC_CTL1的值。<br>0：禁止CTC计数器 |

| | | |
|---|---|---|
| | | 1：使能 CTC 计数器 |
| 4 | 保留 | 必须保持复位值。 |
| 3 | EREFIE | 期望参考信号中断使能<br>0：禁止期望参考信号产生中断<br>1：使能期望参考信号产生中断 |
| 2 | ERRIE | 错误中断使能<br>0：禁止错误中断<br>1：使能错误中断 |
| 1 | CKWARNIE | 时钟校准警告中断使能<br>0：禁止时钟校准警告中断<br>1：使能时钟校准警告中断 |
| 0 | CKOKIE | 时钟校准完成中断使能<br>0：禁止时钟校准完成中断<br>1：使能时钟校准完成中断 |

### 6.4.2. 控制寄存器 1(CTC_CTL1)

地址偏移：0x04
复位值：0x2022 BB7F

该寄存器只能按字(32位)访问。
**注意：**当CNTEN为1时，不能修改该寄存器的值。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| REFPOL | 保留 | REFSEL[1:0] | | 保留 | REFPSC[2:0] | | | CKLIM[7:0] | | | | | | | |
| rw | | rw | | | rw | | | rw | | | | | | | |

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | REFPOL | 参考信号源极性<br>该位由软件置位或清除，用于选择参考信号源的同步极性<br>0：选择上升沿<br>1：选择下降沿 |
| 30 | 保留 | 必须保持复位值。 |
| 29:28 | REFSEL[1:0] | 参考信号源选择<br>该位由软件置位或清除，用于选择参考信号源<br>00：选择GPIO(CTC_SYNC)输入信号<br>01：选择LXTAL时钟<br>10：保留 |

| | | |
|---|---|---|
| | | 11：保留 |
| 27 | 保留 | 必须保持复位值。 |
| 26:24 | REFPSC[2:0] | 参考信号源预分频<br>该位由软件置位或清除<br>000：参考信号不分频<br>001：参考信号2分频<br>010：参考信号4分频<br>011：参考信号8分频<br>100：参考信号16分频<br>101：参考信号32分频<br>110：参考信号64分频<br>111：参考信号128分频 |
| 23:16 | CKLIM[7:0] | 时钟校准时基限值<br>该位由软件置位或清除，用于定义时钟校准时基限值。该位用于频率评估和自动校准过程，详细情况请参考*频率评估和自动校准过程*。 |
| 15:0 | RLVALUE[15:0] | CTC计数器重载值<br>该位由软件置位或清除，用于定义CTC计数器的重载值，当检测到一个参考同步脉冲时，该值将重载到CTC校准计数器中。 |

### 6.4.3. 状态寄存器(CTC_STAT)

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| REFDIR | 保留 | | | | TRIMERR | REFMISS | CKERR | 保留 | | | | EREFIF | ERRIF | CKWARNIF | CKOKIF |
| r | | | | | r | r | r | | | | | r | r | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | REFCAP[15:0] | CTC计数器捕获值<br>当检测到一个同步参考脉冲信号时，CTC 校准计数器中的计数值被存入到 REFCAP 位中。 |
| 15 | REFDIR | CTC校准时钟计数方向<br>当检测到一个同步参考脉冲信号时，CTC校准计数器的计数方向被存入REFDIR位中。<br>0：向上计数 |

| | | |
|---|---|---|
| | | 1：向下计数 |
| 14:11 | 保留 | 必须保持复位值 |
| 10 | TRIMERR | 校准值错误位<br>当CTC_CTL0中的TRIMVALUE值发生上溢或下溢时，该位由硬件置位。若CTC_CTL0中的ERRIE位置1，则会产生一个中断。通过写1到CTC_INTC中的ERRIC位，可以将TRIMERR位清零。<br>0：无校准值错误发生<br>1：发生校准值错误 |
| 9 | REFMISS | 参考同步脉冲信号丢失<br>当参考同步脉冲信号丢失时，该位由硬件置位。当CTC校准计数器在向上增加计数的过程中计数到128 x CKLIM都没有检测到同步参考脉冲信号时，REFMISS位置位。说明当前时钟太快，无法校准到期望频率值，或者有其他错误产生。通过写1到CTC_INTC中的ERRIC位，可以将REFMISS位清零。<br>0：无参考同步脉冲信号丢失<br>1：参考同步脉冲信号丢失 |
| 8 | CKERR | 时钟校准错误位<br>当时钟校准错误产生时，该位由硬件置位。当CTC校准计数器计数值在向下减计数的过程中大于或等于128 x CKLIM，并检测到同步参考脉冲信号时，CKERR置位，说明当前时钟太慢，无法校准到期望频率值。当CTC_CTL0中的ERRIE置1时，产生一个中断。通过写1到CTC_INTC中的ERRIC位，可以将CKERR位清零。<br>0：无时钟校准错误发生<br>1：发生时钟校准错误 |
| 7:4 | 保留 | 必须保持复位值 |
| 3 | EREFIF | 期望参考中断标志位<br>当CTC校准时钟计数器计数到0时，该位由硬件置位。当CTC_CTL0中的EREFIE置1时，产生一个中断。通过写1到CTC_INTC中的EREFIC位，可以将EREFIF位清零。<br>0：无期望参考信号产生<br>1：期望参考信号产生 |
| 2 | ERRIF | 错误中断标志位<br>当发生一个错误时，该位由硬件置位。只要有TRIMERR，REFMISS或者CKERR错误发生时，该位置位。当CTC_CTL0中的ERRIE置位时，产生一个中断。通过写1到CTC_INTC中的ERRIC位，可以将ERRIF位清零。<br>0：无错误发生<br>1：发生错误 |
| 1 | CKWARNIF | 时钟校准警告中断标志位<br>当时钟校准警告产生时，该位由硬件置位。当CTC校准计数器计数值大于或等于3 x CKLIM且小于128 x CKLIM，并检测到同步参考脉冲信号时，CKWARNIF置位。这说明当前时钟频率太慢或者太快，但可以通过校准达到期望频率值。当时钟校准警告产生时，TRIMVALUE值加2或者减2。当CTC_CTL0中的CKWARNIE置1时，产生一个中断。通过写1到CTC_INTC中的CKWARNIC位，可以将CKWARNIF位清零。 |

0：无时钟校准警告发生
1：有时钟校准警告发生

| | | |
|---|---|---|
| 0 | CKOKIF | 时钟校准成功中断标志位<br>当时钟校准成功时，该位由硬件置位。若在CTC校准计数器计数值小于3 x CKLIM时，检测当同步参考脉冲信号，CKOKIF置位。说明当前时钟频率正常，可以使用，不需要通过TRIMVALUE值进行时钟校准。当CTC_CTL0中的CKOKIE置1时，产生一个中断。通过写1到CTC_INTC中的CKOKIC位，可以将CKOKIF位清零。<br>0：时钟校准未成功<br>1：时钟校准成功 |

### 6.4.4. 中断清除寄存器(CTC_INTC)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | | | | | EREFIC | ERRIC | CKWARNIC | CKOKIC |
| | | | | | | | | | | | | w | w | w | w |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:4 | 保留 | 必须保持复位值。 |
| 3 | EREFIC | EREFIF中断清除位<br>该位只能由软件写，读操作返回 0。写 1 可以清除 CTC_STAT 中的 EREFIF 位，写 0 没影响。 |
| 2 | ERRIC | ERRIF 中断清除位<br>该位只能由软件写，读操作返回 0。写 1 可以清除 CTC_STAT 中的 ERRIF 位，TRIMERR 位，REFMISS 位和 CKERR 位，写 0 没影响。 |
| 1 | CKWARNIC | CKWARNIF中断清除位<br>该位只能由软件写，读操作返回 0。写 1 可以清除 CTC_STAT 中的 CKWARNIF 位，写 0 没影响。 |
| 0 | CKOKIC | CKOKIF中断清除位<br>该位只能由软件写，读操作返回 0。写 1 可以清除 CTC_STAT 中的 CKOKIF 位，写 0 没影响。 |

# 7. 中断/事件控制器(EXTI)

## 7.1. 简介

Cortex®-M4集成了嵌套式矢量型中断控制器(Nested Vectored Interrupt Controller(NVIC))来实现高效的异常和中断处理。NVIC实现了低延迟的异常和中断处理，以及电源管理控制。它和内核是紧密耦合的。更多关于NVIC的说明请参考《Cortex®-M4技术参考手册》。

EXTI(中断/事件控制器)包括19个相互独立的边沿检测电路并且能够向处理器内核产生中断请求或唤醒事件。EXTI有三种触发类型：上升沿触发、下降沿触发和任意沿触发。EXTI中的每一个边沿检测电路都可以独立配置和屏蔽。

## 7.2. 主要特征

- Cortex®-M4 系统异常；
- 多达 66 种可屏蔽的外设中断；
- 4 位中断优先级配置位-16 个中断优先等级；
- 高效的中断处理；
- 支持异常抢占和咬尾中断；
- 将系统从省电模式唤醒；
- EXTI 中有多达 19 个相互独立的边沿检测电路；
- 3 种触发类型：上升沿触发，下降沿触发和任意沿触发；
- 软件中断或事件触发；
- 可配置的触发源。

## 7.3. 功能说明

ARM Cortex®-M4处理器和嵌套式矢量型中断控制器(NVIC)在处理(Handler)模式下对所有异常进行优先级区分以及处理。当异常发生时，系统自动将当前处理器工作状态压栈，在执行完中断服务子程序(ISR)后自动将其出栈。

取向量是和当前工作态压栈并行进行的，从而提高了中断入口效率。处理器支持咬尾中断，可实现背靠背中断，大大削减了反复切换工作态所带来的开销。下表列出了Cortex®-M4中的NVIC异常类型。

**表 7-1. Cortex®-M4 中的 NVIC 异常类型**

| 异常类型 | 向量编号 | 优先级(a) | 向量地址 | 描述 |
|---|---|---|---|---|
| **-** | 0 | - | 0x0000_0000 | 保留 |
| **复位** | 1 | -3 | 0x0000_0004 | 复位 |
| **NMI** | 2 | -2 | 0x0000_0008 | 不可屏蔽中断 |
| **硬件故障** | 3 | -1 | 0x0000_000C | 各种硬件级别的故障 |
| **存储器管理** | 4 | 可编程设置 | 0x0000_0010 | 存储器管理 |

| 异常类型 | 向量编号 | 优先级(a) | 向量地址 | 描述 |
|---|---|---|---|---|
| **总线故障** | 5 | 可编程设置 | 0x0000_0014 | 预取指故障，存储器访问故障 |
| **用法故障** | 6 | 可编程设置 | 0x0000_0018 | 未定义的指令或非法状态 |
| **-** | 7-10 | - | 0x0000_001C - 0x0000_002B | 保留 |
| **SVCall 服务调用** | 11 | 可编程设置 | 0x0000_002C | 通过 SWI 指令实现系统服务调用 |
| **调试监控** | 12 | 可编程设置 | 0x0000_0030 | 调试监视器 |
| **-** | 13 | - | 0x0000_0034 | 保留 |
| **PendSV 挂起服务** | 14 | 可编程设置 | 0x0000_0038 | 可挂起的系统服务请求 |
| **SysTick** | 15 | 可编程设置 | 0x0000_003C | 系统节拍定时器 |

**表 7-2. 中断向量表**

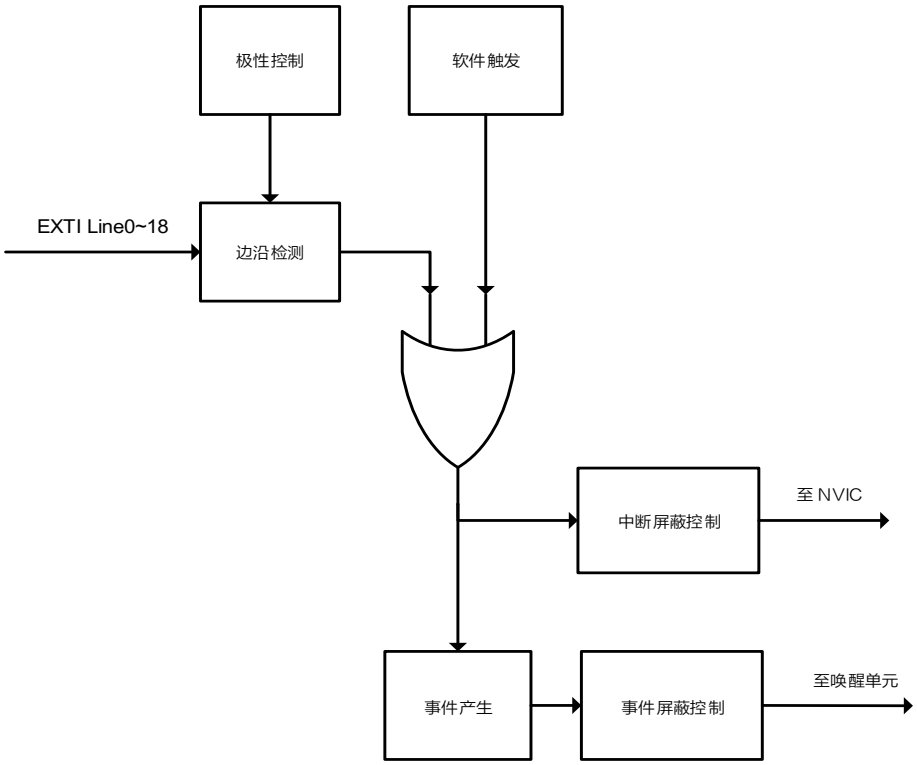
| 中断编号 | 向量编号 | 外设中断描述 | 向量地址 |
|---|---|---|---|
| **IRQ 0** | 16 | 窗口看门狗定时器中断 | 0x0000_0040 |
| **IRQ 1** | 17 | 连接到 EXTI 线的 LVD 中断 | 0x0000_0044 |
| **IRQ 2** | 18 | 侵入检测中断 | 0x0000_0048 |
| **IRQ 3** | 19 | RTC 全局中断 | 0x0000_004C |
| **IRQ 4** | 20 | FMC 全局中断 | 0x0000_0050 |
| **IRQ 5** | 21 | RCU 和 CTC 中断 | 0x0000_0054 |
| **IRQ 6** | 22 | EXTI 线 0 中断 | 0x0000_0058 |
| **IRQ 7** | 23 | EXTI 线 1 中断 | 0x0000_005C |
| **IRQ 8** | 24 | EXTI 线 2 中断 | 0x0000_0060 |
| **IRQ 9** | 25 | EXTI 线 3 中断 | 0x0000_0064 |
| **IRQ 10** | 26 | EXTI 线 4 中断 | 0x0000_0068 |
| **IRQ 11** | 27 | DMA0 通道 0 全局中断 | 0x0000_006C |
| **IRQ 12** | 28 | DMA0 通道 1 全局中断 | 0x0000_0070 |
| **IRQ 13** | 29 | DMA0 通道 2 全局中断 | 0x0000_0074 |
| **IRQ 14** | 30 | DMA0 通道 3 全局中断 | 0x0000_0078 |
| **IRQ 15** | 31 | DMA0 通道 4 全局中断 | 0x0000_007C |
| **IRQ 16** | 32 | DMA0 通道 5 全局中断 | 0x0000_0080 |
| **IRQ 17** | 33 | DMA0 通道 6 全局中断 | 0x0000_0084 |
| **IRQ 18** | 34 | ADC0 和 ADC1 全局中断 | 0x0000_0088 |
| **IRQ 19-22** | 35-38 | 保留 | 0x0000_008C-0x0000_0098 |
| **IRQ 23** | 39 | EXTI 线[9:5]中断 | 0x0000_009C |
| **IRQ 24** | 40 | TIMER0 中止中断和 TIMER8 全局中断 | 0x0000_00A0 |
| **IRQ 25** | 41 | TIMER0 更新中断和 TIMER9 全局中断 | 0x0000_00A4 |

| 中断编号 | 向量编号 | 外设中断描述 | 向量地址 |
|---|---|---|---|
| **IRQ 26** | 42 | TIMER0 触发与通道换相中断和 TIMER10 全局中断 | 0x0000_00A8 |
| **IRQ 27** | 43 | TIMER0 通道捕获比较中断 | 0x0000_00AC |
| **IRQ 28** | 44 | TIMER1 全局中断 | 0x0000_00B0 |
| **IRQ 29** | 45 | TIMER2 全局中断 | 0x0000_00B4 |
| **IRQ 30** | 46 | TIMER3 全局中断 | 0x0000_00B8 |
| **IRQ 31** | 47 | I2C0 事件中断 | 0x0000_00BC |
| **IRQ 32** | 48 | I2C0 错误中断 | 0x0000_00C0 |
| **IRQ 33** | 49 | I2C1 事件中断 | 0x0000_00C4 |
| **IRQ 34** | 50 | I2C1 错误中断 | 0x0000_00C8 |
| **IRQ 35** | 51 | SPI0 全局中断 | 0x0000_00CC |
| **IRQ 36** | 52 | SPI1 全局中断 | 0x0000_00D0 |
| **IRQ 37** | 53 | USART0 全局中断 | 0x0000_00D4 |
| **IRQ 38** | 54 | USART1 全局中断 | 0x0000_00D8 |
| **IRQ 39** | 55 | USART2 全局中断 | 0x0000_00DC |
| **IRQ 40** | 56 | EXTI 线[15:10]中断 | 0x0000_00E0 |
| **IRQ 41** | 57 | 连接到 EXTI 线的 RTC 闹钟中断 | 0x0000_00E4 |
| **IRQ 42** | 58 | 连接到 EXTI 线的 USBFS 唤醒中断 | 0x0000_00E8 |
| **IRQ 43** | 59 | TIMER7 中止中断和 TIMER11 全局中断 | 0x0000_00EC |
| **IRQ 44** | 60 | TIMER7 更新中断和 TIMER12 全局中断 | 0x0000_00F0 |
| **IRQ 45** | 61 | TIMER7 触发与通道换相中断和 TIMER13 全局中断 | 0x0000_00F4 |
| **IRQ 46** | 62 | TIMER7 通道捕获比较中断 | 0x0000_00F8 |
| **IRQ 47** | 63 | 保留 | 0x0000_00FC |
| **IRQ 48** | 64 | EXMC 全局中断 | 0x0000_0100 |
| **IRQ 49** | 65 | 保留 | 0x0000_0104 |
| **IRQ 50** | 66 | TIMER4 全局中断 | 0x0000_0108 |
| **IRQ 51** | 67 | SPI2 全局中断 | 0x0000_010C |
| **IRQ 52** | 68 | UART3 全局中断 | 0x0000_0110 |
| **IRQ 53** | 69 | UART4 全局中断 | 0x0000_0114 |
| **IRQ 54** | 70 | TIMER5 全局中断 | 0x0000_0118 |
| **IRQ 55** | 71 | TIMER6 全局中断 | 0x0000_011C |
| **IRQ 56** | 72 | DMA1 通道 0 全局中断 | 0x0000_0120 |
| **IRQ 57** | 73 | DMA1 通道 1 全局中断 | 0x0000_0124 |
| **IRQ 58** | 74 | DMA1 通道 2 全局中断 | 0x0000_0128 |
| **IRQ 59** | 75 | DMA1 通道 3 全局中断 | 0x0000_012C |
| **IRQ 60** | 76 | DMA1 通道 4 全局中断 | 0x0000_0130 |

| 中断编号 | 向量编号 | 外设中断描述 | 向量地址 |
|---|---|---|---|
| **IRQ 61-66** | 77-82 | 保留 | 0x0000_0134-0x0000_0148 |
| **IRQ 67** | 83 | USBFS 全局中断 | 0x0000_014C |

## 7.4. 外部中断及事件框图

**图 7-1. EXTI 框图**




## 7.5. 外部中断及事件功能概述

EXTI包含多达19个相互独立的边沿检测电路并且可以向处理器产生中断请求或事件唤醒。EXTI提供3种触发类型：上升沿触发，下降沿触发和任意沿触发。EXTI中每个边沿检测电路都可以分别予以配置或屏蔽。

EXTI触发源包括来自I/O管脚的16根线以及来自内部模块的3根线，具体细节参考*表7-3. EXTI触发源*。通过配置GPIO模块的AFIO_EXTISSx寄存器，所有的GPIO管脚都可以被选作EXTI的触发源，具体细节请参考*通用和备用输入/输出接口(GPIO和AFIO)*章节。

除了中断，EXTI还可以向处理器提供事件信号。Cortex®-M4内核完全支持等待中断(WFI)，等待事件(WFE)和发送事件(SEV)指令。芯片内部有一个唤醒中断控制器(WIC)，用户可以放心的让处理器和NVIC进入功耗极低的省电模式，由WIC来识别中断和事件以及判断优先级。当某些预期的事件发生时，EXTI能唤醒处理器及整个系统，例如一个特定的I/O管脚电平翻转或者RTC闹钟动作。

**硬件触发**

硬件触发被用来检测外部或内部信号的电压变化。软件需要按如下步骤配置来使用这项功能：

1. 根据应用需要配置 AFIO 模块中的 EXTI 触发源；
2. 配置 EXTI_RTEN 寄存器和 EXTI_FTEN 寄存器以使能相应引脚的上升沿或下降沿检测(软件应当同时配置引脚对应的 RTENx 和 FTENx 位以检测该引脚上升沿和下降沿的变化)；
3. 通过配置引脚对应的 EXTI_INTEN 或 EXTI_EVEN 位，使能中断或事件；
4. EXTI 开始检测被配置的引脚上的电平变化，当这些引脚上期望的变化被检测到时，使能的中断或事件将被触发。如果为中断触发，则对应的 PD 位将立刻被置 1；如果为事件触发，则对应的 PD 位不被置 1。软件需要响应该中断或事件并清除相应 PDx 位。

**软件触发**

按照如下步骤软件也可以触发 EXTI 中断或事件：

1. 配置对应的 EXTI_INTEN 或 EXTI_EVEN 位使能中断或事件；
2. 配置EXTI_SWIEV寄存器的对应SWIEVx位，使能的中断或事件将被立即触发。如果为中断触发，则对应的PD位将立刻被置1；如果为事件触发，则对应的PD位不被置1。软件需要响应该中断或事件并清除相应PDx位。

**表 7-3. EXTI 触发源**

| EXTI 线编号 | 触发源 |
|---|---|
| 0 | PA0 / PB0 / PC0 / PD0 / PE0 |
| 1 | PA1 / PB1 / PC1 / PD1 / PE1 |
| 2 | PA2 / PB2 / PC2 / PD2 / PE2 |
| 3 | PA3 / PB3 / PC3 / PD3 / PE3 |
| 4 | PA4 / PB4 / PC4 / PD4 / PE4 |
| 5 | PA5 / PB5 / PC5 / PD5 / PE5 |
| 6 | PA6 / PB6 / PC6 / PD6 / PE6 |
| 7 | PA7 / PB7 / PC7 / PD7 / PE7 |
| 8 | PA8 / PB8 / PC8 / PD8 / PE8 |
| 9 | PA9 / PB9 / PC9 / PD9 / PE9 |
| 10 | PA10 / PB10 / PC10 / PD10 / PE10 |
| 11 | PA11 / PB11 / PC11 / PD11 / PE11 |
| 12 | PA12 / PB12 / PC12 / PD12 / PE12 |
| 13 | PA13 / PB13 / PC13 / PD13 / PE13 |
| 14 | PA14 / PB14 / PC14 / PD14 / PE14 |
| 15 | PA15 / PB15 / PC15 / PD15 / PE15 |
| 16 | LVD |
| 17 | RTC 闹钟 |
| 18 | USBFS 唤醒 |

## 7.6. EXTI 寄存器

EXTI基地址：0x4001 0400

### 7.6.1. 中断使能寄存器(EXTI_INTEN)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | | | | | | INTEN18 | INTEN17 | INTEN16 |
| | | | | | | | | | | | | | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| INTEN15 | INTEN14 | INTEN13 | INTEN12 | INTEN11 | INTEN10 | INTEN9 | INTEN8 | INTEN7 | INTEN6 | INTEN5 | INTEN4 | INTEN3 | INTEN2 | INTEN1 | INTEN0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:19 | 保留 | 必须保持复位值。 |
| 18: 0 | INTENx | 中断使能位x(x = 0…18)<br>0：第x线中断被禁用<br>1：第x线中断被使能 |

### 7.6.2. 事件使能寄存器(EXTI_EVEN)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | | | | | | EVEN18 | EVEN17 | EVEN16 |
| | | | | | | | | | | | | | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EVEN15 | EVEN14 | EVEN13 | EVEN12 | EVEN11 | EVEN10 | EVEN9 | EVEN8 | EVEN7 | EVEN6 | EVEN5 | EVEN4 | EVEN3 | EVEN2 | EVEN1 | EVEN0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:19 | 保留 | 必须保持复位值。 |
| 18: 0 | EVENx | 事件使能位x(x = 0…18)<br>0：第x线事件被禁用.<br>1：第x线事件被使能 |

### 7.6.3. 上升沿触发使能寄存器(EXTI_RTEN)

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | | | | | | RTEN18 | RTEN17 | RTEN16 |
| | | | | | | | | | | | | | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RTEN15 | RTEN14 | RTEN13 | RTEN12 | RTEN11 | RTEN10 | RTEN9 | RTEN8 | RTEN7 | RTEN6 | RTEN5 | RTEN4 | RTEN3 | RTEN2 | RTEN1 | RTEN0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:19 | 保留 | 必须保持复位值。 |
| 18:0 | RTENx | 上升沿触发使能x(x = 0…18)<br>0：第x线上升沿触发无效<br>1：第x线上升沿触发有效(中断/事件请求) |

### 7.6.4. 下降沿触发使能寄存器(EXTI_FTEN)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | | | | | | FTEN18 | FTEN17 | FTEN16 |
| | | | | | | | | | | | | | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FTEN15 | FTEN14 | FTEN13 | FTEN12 | FTEN11 | FTEN10 | FTEN9 | FTEN8 | FTEN7 | FTEN6 | FTEN5 | FTEN4 | FTEN3 | FTEN2 | FTEN1 | FTEN0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31: 19 | 保留 | 必须保持复位值。 |
| 18: 0 | FTENx | 下降沿触发使能x(x = 0…18)<br>0：第x线下降沿触发无效<br>1：第x线下降沿触发有效(中断/事件请求) |

### 7.6.5. 软件中断事件寄存器(EXTI_SWIEV)

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | | | | | | SWIEV18 | SWIEV17 | SWIEV16 |
| | | | | | | | | | | | | | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SWIEV15 | SWIEV14 | SWIEV13 | SWIEV12 | SWIEV11 | SWIEV10 | SWIEV9 | SWIEV8 | SWIEV7 | SWIEV6 | SWIEV5 | SWIEV4 | SWIEV3 | SWIEV2 | SWIEV1 | SWIEV0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31: 19 | 保留 | 必须保持复位值。 |
| 18: 0 | SWIEVx | 中断/事件软件触发x(x = 0…18)<br>0：禁用EXTI线x软件中断/事件请求<br>1：激活EXTI线x软件中断/事件请求 |

## 7.6.6. 挂起寄存器(EXTI_PD)

地址偏移：0x14
复位值：未定义

该寄存器只能按字(32 位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | | | | | | PD18 | PD17 | PD16 |
| | | | | | | | | | | | | | rc_w1 | rc_w1 | rc_w1 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PD15 | PD14 | PD13 | PD12 | PD11 | PD10 | PD9 | PD8 | PD7 | PD6 | PD5 | PD4 | PD3 | PD2 | PD1 | PD0 |
| rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31: 19 | 保留 | 必须保持复位值。 |
| 18: 0 | PDx | 中断挂起状态x(x = 0…18)<br>0：EXTI线x没有被触发<br>1：EXTI线x被触发<br>对这些位写1，可将其清0。 |

# 8. 通用和备用输入/输出接口(GPIO 和 AFIO)

## 8.1. 简介

最多可支持80个通用I/O引脚(GPIO)，分别为PA0 ~ PA15，PB0 ~ PB15，PC0 ~ PC15，PD0 ~ PD15，PE0 ~ PE15，各片上设备用其来实现逻辑输入/输出功能。每个GPIO端口有相关的控制和配置寄存器以满足特定应用的需求。GPIO引脚上的外部中断在中断/事件控制器(EXTI)中有相关的控制和配置寄存器。

GPIO端口和其他的备用功能(AFs)共用引脚，在特定的封装下获得最大的灵活性。GPIO引脚通过配置相关的寄存器可以用作备用功能输入/输出引脚。

每个GPIO引脚可以由软件配置为输出(推挽或开漏)、输入、外设备用功能或者模拟模式。每个GPIO引脚都可以配置为上拉、下拉或无上拉/下拉。除模拟模式外，所有的GPIO引脚都具备大电流驱动能力。

## 8.2. 主要特征

- 输入/输出方向控制；
- 施密特触发器输入功能使能控制；
- 每个引脚都具有弱上拉/下拉功能；
- 推挽/开漏输出使能控制；
- 置位/复位输出使能；
- 可编程触发沿的外部中断-使用EXTI配置寄存器
- 模拟输入/输出配置；
- 备用功能输入/输出配置；
- 端口配置锁定。

## 8.3. 功能说明

每个通用I/O端口都可以通过两个32位的控制寄存器(GPIOx_CTL0/ GPIOx_CTL1)和一个32位寄存器(GPIOx_OCTL)配置为8种模式：模拟输入，浮空输入，上拉输入，下拉输入，GPIO推挽输出，GPIO开漏输出，备用推挽输出和备用开漏输出。详情请见*表8-1. GPIO配置表*。

**表 8-1. GPIO 配置表**

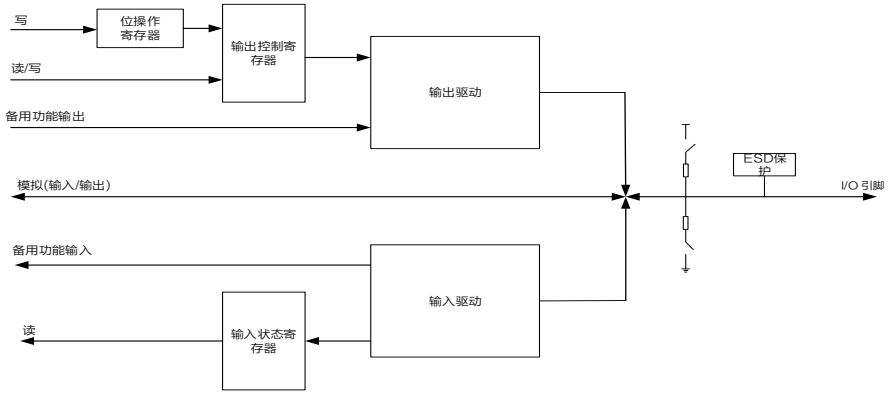
| 配置模式 | | CTL[1:0] | MD[1:0] | OCTL |
|---|---|---|---|---|
| 输入 | 模拟 | 00 | 00 | 不使用 |
| | 浮空输入 | 01 | | 不使用 |
| | 下拉输入 | 10 | | 0 |

| | | | | |
|---|---|---|---|---|
| | 上拉输入 | 10 | | 1 |
| **配置模式** | | **CTL[1:0]** | **SPDy: MD[1:0]** | **OCTL** |
| 普通输出(GPIO) | 推挽 | 00 | 00: 保留<br>01: 最大速度到 10MHz<br>10: 最大速度到 2MHz<br>11: 最大速度到 50MHz<br>11: 最大速度到 120MHz(1)(同时设置SPDy值为1) | 0 或 1 |
| | 开漏 | 01 | | 0 或 1 |
| 备用功能输出(AFIO) | 推挽 | 10 | | 不使用 |
| | 开漏 | 11 | | 不使用 |

1、当GPIO输出速度超过50MHz时，需要使能GPIO的补偿单元，参考IO补偿控制寄存器(AFIO_CPSCTL)。

*图 8-1. GPIO 端口位的基本结构*为标准 I/O 端口位的基本结构图。

**图 8-1. GPIO 端口位的基本结构**




## 8.3.1. GPIO 引脚配置

在复位期间或复位之后，备用功能并未激活，所有GPIO端口都被配置成输入浮空模式，这种输入模式禁用上拉(PU)/下拉(PD)电阻。但是复位后，串行线调试端口(JTAG/Serial-Wired Debug pins)为输入PU/PD模式：

PA15：JTDI为上拉模式；
PA14：JTCK / SWCLK为下拉模式；
PA13：JTMS / SWDIO为上拉模式；
PB4：NJTRST为上拉模式；
PB3：JTDO为浮空模式。

GPIO引脚可以配置为输入或输出模式，当GPIO引脚可配置为输入引脚时，所有的GPIO引脚内部都有一个可选择的弱上拉和弱下拉电阻。外部引脚上的数据在每个APB2时钟周期时都会装载到数据输入寄存器(GPIOx_ISTAT)。

当GPIO引脚配置为输出引脚，用户可以配置端口的输出速度和选择输出驱动模式：推挽或开漏模式，输出寄存器(GPIOx_OCTL)的值将会从相应I/O引脚上输出。

当对GPIOx_OCTL进行位操作时，不需要先读再写，用户可以通过写'1'到位操作寄存器

(GPIOx_BOP，或用于清0的GPIOx_BC)修改一位或几位，该过程仅需要一个最小的APB2写访问周期，而其他位不受影响。

### 8.3.2. 外部中断/事件线

只有在输入模式下配置，端口才能使用外部中断/事件线。

### 8.3.3. 备用功能(AF)

当端口配置为AFIO(设置GPIOx_CTL0/GPIOx_CTL1寄存器中的CTLy值为"0b10"或"0b11"，MDy位值为"0b01"，"0b10"或"0b11")时，该端口用作外设备用功能。端口备用功能分配的详细介绍见芯片数据手册。

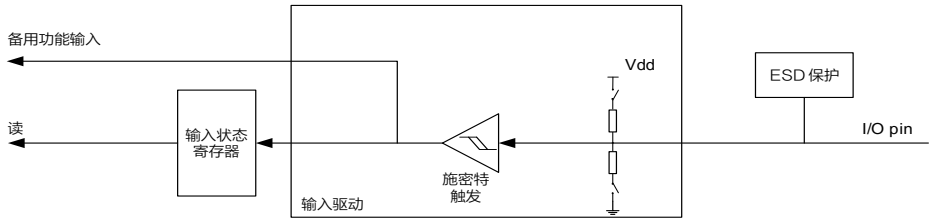
### 8.3.4. 输入配置

当GPIO引脚配置为输入时：

- 施密特触发输入使能；
- 可选择的弱上拉和下拉电阻；
- 当前I/O引脚上的数据在每个APB2时钟周期都会被采样并存入端口输入状态寄存器；
- 输出缓冲器禁用。

*图 8-2. 输入配置的基本结构*显示 I/O 引脚的输入配置。

**图 8-2. 输入配置的基本结构**



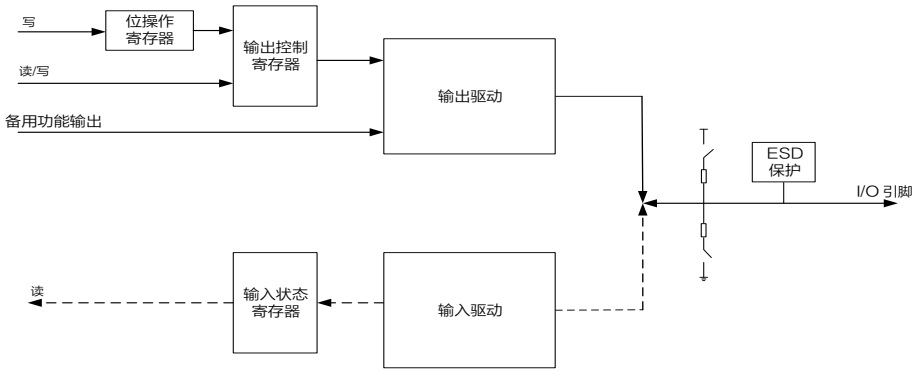

### 8.3.5. 输出配置

当GPIO配置为输出时：

- 施密特触发输入使能；
- 弱上拉和下拉电阻禁用；
- 输出缓冲器使能；
- 开漏模式：输出控制寄存器设置为"0"时，相应引脚输出低电平；输出控制寄存器设置为"1"，相应管脚处于高阻状态；
- 推挽模式：输出控制寄存器设置为"0"时，相应引脚输出低电平；输出控制寄存器设置为"1"，相应引脚输出高电平；
- 对端口输出控制寄存器进行读操作，将返回上次写入的值；
- 对端口输入状态寄存器进行读操作，将获得当前I/O口的状态。

*图 8-3. 输出配置的基本结构*是 I/O 端口的输出配置。

**图 8-3. 输出配置的基本结构**




### 8.3.6. 模拟配置

当GPIO引脚用于模拟模式时：

- 弱上拉和下拉电阻禁用；
- 输出缓冲器禁用；
- 施密特触发输入禁用；
- 端口输入状态寄存器相应位读取为"0"。

*图 8-4. 模拟配置的基本结构*是 I/O 端口的模拟模式配置。

**图 8-4. 模拟配置的基本结构**




### 8.3.7. 备用功能(AF)配置

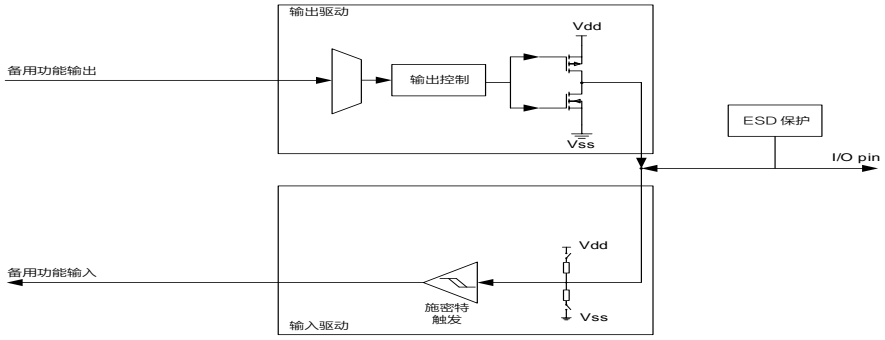
为了适应不同的器件封装，GPIO端口支持软件配置将一些备用功能应用到其他引脚上。

当引脚配置为备用功能时：

- 使用开漏或推挽功能时，可使能输出缓冲器；
- 输出缓冲器由外设驱动；
- 施密特触发输入使能；
- 在输入配置时，可选择弱上拉/下拉电阻；
- I/O引脚上的数据在每个APB2时钟周期采样并存入端口输入状态寄存器；
- 对端口输入状态寄存器进行读操作，将获得I/O口的状态；
- 对端口输出控制寄存器进行读操作，将返回上次写入的值。

*图 8-5. 备用功能配置的基本结构*是 I/O 端口备用功能配置图。

**图 8-5. 备用功能配置的基本结构**




### 8.3.8. GPIO 锁定功能

GPIO的锁定机制可以保护I/O端口的配置。

被保护的寄存器有GPIOx_CTL0和GPIOx_CTL1。通过配置32位锁定寄存器(GPIOx_LOCK)可以锁定I/O端口的配置。通过特定的锁定序列配置GPIOx_LOCK中的LKK位和LKy位，相应的端口位被锁定，直到下一个复位前，相应端口位的配置都不能修改。建议在电源驱动模块的配置中使用锁定功能。

### 8.3.9. GPIO I/O 补偿单元

当I/O端口输出速度大于50MHz时，建议使用I/O补偿单元对I/O端口进行斜率控制，从而降低I/O端口噪声对工作电源的影响。

I/O补偿单元在系统复位后，默认是关闭状态，需要用户根据需要开启。

在使能I/O补偿单元后，将产生一个准备完成标志位CPS_RDY，用于指示补偿单元已经准备好，可以使用。如果电源电压超过2.4V~3.6V，不能使用 I/O 补偿单元，必须关闭该功能。

## 8.4. I/O 重映射功能和调试配置

### 8.4.1. 介绍

为了扩展GPIO的灵活性或外设功能使用，通过配置AFIO端口配置寄存器(AFIO_PCF0/AFIO_PCF1)，每个I/O引脚都可以配置多达4种不同的功能。通过使用外设IO的重映射功能可以选择合适的引脚位置。另外，通过配置相应的EXTI源选择寄存器(AFIO_EXTISSx)选择触发中断或事件，GPIO引脚可以用作EXTI中断线。

### 8.4.2. 主要特征

- EXTI 源选择
- 每个引脚具有多达4种备用功能的配置

### 8.4.3. JTAG/SWD 备用功能重映射

调试接口信号映射在GPIO端口的情况如下表所示。

**表 8-2. 调试接口信号**

| 引脚 | 功能描述 |
|---|---|
| PA13 | JTMS / SWDIO |
| PA14 | JTCK / SWCLK |
| PA15 | JTDI |
| PB3 | JTDO / TRACESWO |
| PB4 | NJTRST |

为了减少用于调试的GPIO端口，用户可以配置AFIO_PCF0寄存器中的SWJ_CFG [2:0]位为不同的值。具体情况参照下表。

**表 8-3. 调试端口和引脚可用性**

| SWJ_CFG[2:0] | 可用的调试端口 | 引脚可用性 | | | | |
|---|---|---|---|---|---|---|
| | | PA13 | PA14 | PA15 | PB3 | PB4 |
| 000 | JTAG-DP 开启<br>SW-DP 开启<br>(复位状态) | X | X | X | X | X |
| 001 | JTAG-DP 开启<br>SW-DP 开启<br>没有 NJTRST | X | X | X | X | sqrt |
| 010 | JTAG-DP 关闭<br>SW-DP 开启 | X | X | sqrt | sqrt(1) | sqrt |
| 100 | JTAG-DP 关闭<br>SW-DP 关闭 | sqrt | sqrt | sqrt | sqrt | sqrt |
| 其他 | 禁用 | | | | | |

1. 只有在不使用异步跟踪时，I/O才能使用。
2. "sqrt"表示对应的引脚作为普通 I/O 引脚使用。
3. "X"表示对应的引脚不能作为普通 I/O 引脚使用。
4. SWJ(串行线JTAG)支持JTAG或SWD访问Cortex调试端口。系统复位后的默认状态是启用SWJ但没有跟踪功能，这种状态下，可以通过在JTMS/JTCK引脚上的发送特定的信号使能JTAG或SW(串行线)模式。

### 8.4.4. ADC AF 重映射

参考AFIO端口配置寄存器0(AFIO_PCF0)。

**表 8-4. ADC0/1 常规转换外部触发备用功能重映射(1)**

| Register | ADC0 | ADC1 |
|---|---|---|
| ADC0_ETRGRT_REMAP = 0 | 连接 ADC0 常规转换外部触发与 EXTI11 | - |

| Register | ADC0 | ADC1 |
|---|---|---|
| ADC0_ETRGRT_REMAP = 1 | 连接 ADC0 常规转换外部触发与 TIMER7_TRGO | |
| ADC1_ETRGRT_REMAP = 0 | - | 连接 ADC1 常规转换外部触发与 EXTI11 |
| ADC1_ETRGRT_REMAP = 1 | - | 连接 ADC1 常规转换外部触发与 TIMER7_TRGO |

1. 重映射仅仅适用于高密度和超高密度的产品。

### 8.4.5. TIMER AF 重映射

**表 8-5. TIMER 备用功能重映射**

| 备用功能 | TIMERX_REMAP [1:0](X = 0，1，2) | | | |
|---|---|---|---|---|
| | TIMERX_REMAP(X = 3，8) | | - | |
| | "0"/"00" (没有映射) | "1"/"01" (部分映射) | "10" (部分映射) | "11" (全映射) |
| TIMER0_ETI | PA12 | | - | PE7 |
| TIMER0_CH0 | PA8 | | - | PE9 |
| TIMER0_CH1 | PA9 | | - | PE11 |
| TIMER0_CH2 | PA10 | | - | PE13 |
| TIMER0_CH3 | PA11 | | - | PE14 |
| TIMER0_BRKIN | PB12(2) | PA6 | - | PE15 |
| TIMER0_CH0_ON | PB13(2) | PA7 | - | PE8 |
| TIMER0_CH1_ON | PB14(2) | PB0 | - | PE10 |
| TIMER0_CH2_ON | PB15(2) | PB1 | - | PE12 |
| TIMER1_CH0/TIMER1_ETI(3) | PA0 | PA15 | PA0 | PA15 |
| TIMER1_CH1 | PA1 | PB3 | PA1 | PB3 |
| TIMER1_CH2 | PA2 | | PB10 | |
| TIMER1_CH3 | PA3 | | PB11 | |
| TIMER2_CH0 | PA6 | - | PB4 | PC6 |
| TIMER2_CH1 | PA7 | - | PB5 | PC7 |
| TIMER2_CH2 | PB0 | - | PB0 | PC8 |
| TIMER2_CH3 | PB1 | - | PB1 | PC9 |
| TIMER3_CH0 | PB6 | PD12 | - | - |

| 备用功能 | TIMERX_REMAP [1:0](X = 0，1，2) | | | |
|---|---|---|---|---|
| | TIMERX_REMAP(X = 3，8) | | - | |
| | "0"/"00" (没有映射) | "1"/"01" (部分映射) | "10" (部分映射) | "11" (全映射) |
| TIMER3_CH1 | PB7 | PD13 | - | - |
| TIMER3_CH2 | PB8 | PD14 | - | - |
| TIMER3_CH3 | PB9 | PD15 | - | - |
| TIMER8_CH0 | PA2 | PE5 | - | - |
| TIMER8_CH1 | PA3 | PE6 | - | - |

1. TIMER0 重映射(全映射)仅仅适用于100引脚的封装。
2. TIMER0 重映射不适用于36引脚封装。
3. TIMER1_CH0 和 TIMER1_ETI 共用一个引脚，但不能同时使用。
4. TIMER1 重映射(全映射)不适用于36引脚封装。
5. TIMER2 重映射(全映射)仅仅适用于64引脚，100引脚的封装。
6. TIMER3 重映射(全映射)仅仅适用于100引脚的封装。
7. TIMER8 参考备用功能映射和调试I/O配置寄存器1(AFIO_PCF1)。

**表 8-6. TMER4 备用功能重映射[1]**

| 备用功能 | TIMER4CH3_IREMAP = 0 | TIMER4CH3_IREMAP = 1 |
|---|---|---|
| TIMER4_CH3 | TIMER4_CH3 与 PA3 相连 | IRC40K 内部时钟与 TIMER4_CH3 输入相连，用于校正 |

1. 重映射适用于高密度、超高密度和互联型的产品。

### 8.4.6. USART AF 重映射

参考 AFIO 端口配置寄存器 0 (AFIO_PCF0)。

**表 8-7. USART0/1 备用功能重映射**

| 寄存器 | USART0 | USART1 | USART2 |
|---|---|---|---|
| USART0_REMAP = 0 | PA9(USART0_TX)<br>PA10(USART0_RX) | | - |
| USART0_REMAP = 1 | PB6(USART0_TX)<br>PB7(USART0_RX) | | - |
| USART1_REMAP = 0 | - | PA0(USART1_CTS)<br>PA1(USART1_RTS)<br>PA2(USART1_TX)<br>PA3(USART1_RX)<br>PA4(USART1_CK) | - |

| 寄存器 | USART0 | USART1 | USART2 |
|---|---|---|---|
| USART1_REMAP = 1 (1) | - | PD3(USART1_CTS)<br>PD4(USART1_RTS)<br>PD5(USART1_TX)<br>PD6(USART1_RX)<br>PD7(USART1_CK) | - |
| USART2_REMAP [1:0] = "00" (没有映射) | - | - | PB10(USART2_TX)<br>PB11(USART2_RX)<br>PB12(USART2_CK)<br>PB13(USART2_CTS)<br>PB14(USART2_RTS) |
| USART2_REMAP [1:0] ="01" (部分映射) (2) | - | - | PC10(USART2_TX)<br>PC11(USART2_RX)<br>PC12(USART2_CK)<br>PB13(USART2_CTS)<br>PB14(USART2_RTS) |
| USART2_REMAP [1:0] ="11" (全映射) (3) | - | - | PD8(USART2_TX)<br>PD9(USART2_RX)<br>PD10(USART2_CK)<br>PD11(USART2_CTS)<br>PD12(USART2_RTS) |

1. 重映射仅仅适用于 100 引脚的封装。
2. 重映射仅仅适用于 64 引脚，100 引脚的封装。
3. 重映射仅仅适用于 100 引脚的封装。

### 8.4.7. I2C0 备用功能重映射

参考 AFIO 端口配置寄存器 0 (AFIO_PCF0)。

**表 8-8. I2C0 备用功能重映射**

| 寄存器 | I2C0_SCL | I2C0_SDA |
|---|---|---|
| I2C0_REMAP = 0 | PB6 | PB7 |
| I2C0_REMAP = 1 | PB8 | PB9 |

### 8.4.8. SPI0/SPI2/I2S 备用功能重映射

参考 AFIO 端口配置寄存器 0 (AFIO_PCF0)。

**表 8-9. SPI0/SPI2/I2S 备用功能重映射**

| 寄存器 | SPI0 | SPI2/I2S |
|---|---|---|
| SPI0_REMAP = 0 | PA4(SPI0_NSS)<br>PA5(SPI0_SCK)<br>PA6(SPI0_MISO)<br>PA7(SPI0_MOSI) | - |

| 寄存器 | SPI0 | SPI2/I2S |
|---|---|---|
| | PA2(SPI0_IO2)<br>PA3(SPI0_IO3) | |
| SPI0_REMAP = 1 | PA15(SPI0_NSS)<br>PB3(SPI0_SCK)<br>PB4(SPI0_MISO)<br>PB5(SPI0_MOSI)<br>PB6(SPI0_IO2)<br>PB7(SPI0_IO3) | - |
| SPI2_REMAP = 0 | - | PA15(SPI2_NSS/ I2S2_WS)<br>PB3(SPI2_SCK/ I2S2_CK)<br>PB4(SPI2_MISO)<br>PB5(SPI2_MOSI/I2S2_SD) |
| SPI2_REMAP = 1 | - | PA4(SPI2_NSS/ I2S2_WS)<br>PC10(SPI2_SCK/ I2S2_CK)<br>PC11(SPI2_MISO)<br>PC12(SPI2_MOSI/I2S2_SD) |

### 8.4.9. CTC 备用功能重映射

参考AFIO端口配置寄存器1(AFIO_PCF1)。

**表 8-10. CTC 备用功能重映射**

| 备用功能 | CTC_REMAP [1:0] = "00" | CTC_REMAP [1:0] = "01" |
|---|---|---|
| CTC_SYNC | PA8 | PD15 |

### 8.4.10. CLK 引脚 AF 重映射

当LXTAL关闭的时候，OSC32_IN和OSC32_OUT分别可以用做普通的I/O端口PC14和PC15。LXTAL的优先级比其他普通IO功能高。

注意：1. 当1.8V区域关掉(进入待机模式)或备份区域由VBAT供电(不使用VDD供电)，PC14/PC15 不能用于普通IO功能，将会被设置为模拟模式。

2. 参考3.3.1电池备份域章节中的IO口用法。

**表 8-11. OSC32 引脚配置**

| 备用功能 | LXTAL= ON | LXTAL= OFF |
|---|---|---|
| PC14 | OSC32_IN | PC14 |
| PC15 | OSC32_OUT | PC15 |

HXTAL晶振引脚OSC_IN/OSC_OUT可以用做普通的I/O端口PD0/PD1。对于36、48和64脚的封装，PD0和PD1不能用来产生外部中断/事件。

表 8-12. OSC 引脚配置

| 备用功能 | HXTAL= ON | HXTAL= OFF |
|---|---|---|
| PD0 | OSC_IN | PD0 |
| PD1 | OSC_OUT | PD1 |

## 8.5. GPIO 寄存器

GPIOA基地址：0x4001 0800
GPIOB基地址：0x4001 0C00
GPIOC基地址：0x4001 1000
GPIOD基地址：0x4001 1400
GPIOE基地址：0x4001 1800
AFIO基地址：0x4001 0000

### 8.5.1. 端口控制寄存器 0 (GPIOx_CTL0, x=A..E)

地址偏移：0x00
复位值：0x4444 4444(x= A, C..E)/ 0x4448 4444(x=B)

该寄存器只能按字(32 位)访问。

| 31 30 | 29 28 | 27 26 | 25 24 | 23 22 | 21 20 | 19 18 | 17 16 |
|---|---|---|---|---|---|---|---|
| CTL7[1:0] | MD7[1:0] | CTL6[1:0] | MD6[1:0] | CTL5[1:0] | MD5[1:0] | CTL4[1:0] | MD4[1:0] |
| rw | rw | rw | rw | rw | rw | rw | rw |

| 15 14 | 13 12 | 11 10 | 9 8 | 7 6 | 5 4 | 3 2 | 1 0 |
|---|---|---|---|---|---|---|---|
| CTL3[1:0] | MD3[1:0] | CTL2[1:0] | MD2[1:0] | CTL1[1:0] | MD1[1:0] | CTL0[1:0] | MD0[1:0] |
| rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:30 | CTL7[1:0] | Port 7配置位<br>该位由软件置位和清除。<br>参考CTL0[1:0]的描述 |
| 29:28 | MD7[1:0] | Port 7 模式位<br>该位由软件置位和清除。<br>参考MD0 [1:0]的描述 |
| 27:26 | CTL6[1:0] | Port 6配置位<br>该位由软件置位和清除。<br>参考CTL0[1:0]的描述 |
| 25:24 | MD6[1:0] | Port 6模式位<br>该位由软件置位和清除。<br>参考MD0 [1:0]的描述 |
| 23:22 | CTL5[1:0] | Port 5配置位<br>该位由软件置位和清除。<br>参考CTL0[1:0]的描述 |
| 21:20 | MD5[1:0] | Port 5模式位<br>该位由软件置位和清除。 |

| | | |
|---|---|---|
| | | 参考MD0 [1:0]的描述 |
| 19:18 | CTL4[1:0] | Port 4配置位<br>该位由软件置位和清除。<br>参考CTL0[1:0]的描述 |
| 17:16 | MD4[1:0] | Port 4模式位<br>该位由软件置位和清除。<br>参考MD0 [1:0]的描述 |
| 15:14 | CTL3[1:0] | Port 3配置位<br>该位由软件置位和清除。<br>参考CTL0[1:0]的描述 |
| 13:12 | MD3[1:0] | Port 3模式位<br>该位由软件置位和清除。<br>参考MD0 [1:0]的描述 |
| 11:10 | CTL2[1:0] | Port 2配置位<br>该位由软件置位和清除。<br>参考CTL0[1:0]的描述 |
| 9:8 | MD2[1:0] | Port 2模式位<br>该位由软件置位和清除。<br>参考MD0 [1:0]的描述 |
| 7:6 | CTL1[1:0] | Port 1配置位<br>该位由软件置位和清除。<br>参考CTL0[1:0]的描述 |
| 5:4 | MD1[1:0] | Port 1模式位<br>该位由软件置位和清除。<br>参考MD0 [1:0]的描述 |
| 3:2 | CTL0[1:0] | Pin 0配置位<br>该位由软件置位和清除。<br>输入模式（MD[1:0] =00)<br>00： 模拟输入<br>01： 浮空输入<br>10： 上拉输入 /下拉输入<br>11： 保留<br><br>输出模式（MD[1:0] >00)<br>00： GPIO 推挽输出<br>01： GPIO 开漏输出<br>10： AFIO 推挽输出<br>11： AFIO 开漏输出 |
| 1:0 | MD0[1:0] | Port 0模式位 |

该位由软件置位和清除。
00：输入模式 (复位状态)
01：输出模式(10MHz)
10：输出模式(2MHz)
11：输出模式(50MHz)

### 8.5.2. 端口控制寄存器 1 (GPIOx_CTL1, x=A..E)

地址偏移：0x04
复位值：0x4444 4444(x= B..E)/ 0x8884 4444(x=A)

该寄存器只能按字(32 位)访问。

| 31 30 | 29 28 | 27 26 | 25 24 | 23 22 | 21 20 | 19 18 | 17 16 |
|---|---|---|---|---|---|---|---|
| CTL15[1:0] | MD15[1:0] | CTL14[1:0] | MD14[1:0] | CTL13[1:0] | MD13[1:0] | CTL12[1:0] | MD12[1:0] |
| rw | rw | rw | rw | rw | rw | rw | rw |

| 15 14 | 13 12 | 11 10 | 9 8 | 7 6 | 5 4 | 3 2 | 1 0 |
|---|---|---|---|---|---|---|---|
| CTL11[1:0] | MD11[1:0] | CTL10[1:0] | MD10[1:0] | CTL9[1:0] | MD9[1:0] | CTL8[1:0] | MD8[1:0] |
| rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:30 | CTL15[1:0] | Port 15配置位<br>该位由软件置位和清除。<br>参考CTL0[1:0]的描述 |
| 29:28 | MD15[1:0] | Port 15模式位<br>该位由软件置位和清除。<br>参考MD0 [1:0]的描述 |
| 27:26 | CTL14[1:0] | Port 14配置位<br>该位由软件置位和清除。<br>参考CTL0[1:0]的描述 |
| 25:24 | MD14[1:0] | Port 14模式位<br>该位由软件置位和清除。<br>参考MD0 [1:0]的描述 |
| 23:22 | CTL13[1:0] | Port13配置位<br>该位由软件置位和清除。<br>参考CTL0[1:0]的描述 |
| 21:20 | MD13[1:0] | Port 13模式位<br>该位由软件置位和清除。<br>参考MD0 [1:0]的描述 |
| 19:18 | CTL12[1:0] | Port 12配置位<br>该位由软件置位和清除。 |

| | | |
|---|---|---|
| | | 参考CTL0[1:0]的描述 |
| 17:16 | MD12[1:0] | Port 12模式位<br>该位由软件置位和清除。<br>参考MD0 [1:0]的描述 |
| 15:14 | CTL11[1:0] | Port 11配置位<br>该位由软件置位和清除。<br>参考CTL0[1:0]的描述 |
| 13:12 | MD11[1:0] | Port 11模式位<br>该位由软件置位和清除。<br>参考MD0 [1:0]的描述 |
| 11:10 | CTL10[1:0] | Port 10配置位<br>该位由软件置位和清除。<br>参考CTL0[1:0]的描述 |
| 9:8 | MD10[1:0] | Port10模式位<br>该位由软件置位和清除。<br>参考MD0 [1:0]的描述 |
| 7:6 | CTL9[1:0] | Port 9配置位<br>该位由软件置位和清除。<br>参考CTL0[1:0]的描述 |
| 5:4 | MD9[1:0] | Port 9模式位<br>该位由软件置位和清除。<br>参考MD0 [1:0]的描述 |
| 3:2 | CTL8[1:0] | Pin8配置位<br>该位由软件置位和清除。<br>参考CTL0[1:0]的描述 |
| 1:0 | MD8[1:0] | Port 8模式位<br>该位由软件置位和清除。<br>参考MD0 [1:0]的描述 |

### 8.5.3. 端口输入状态寄存器 (GPIOx_ISTAT, x=A..E)

地址偏移：0x08
复位值：0x0000 XXXX

该寄存器只能按字(32 位)访问。

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

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

| ISTAT15 | ISTAT14 | ISTAT13 | ISTAT12 | ISTAT11 | ISTAT10 | ISTAT 9 | ISTAT 8 | ISTAT 7 | ISTAT 6 | ISTAT 5 | ISTAT 4 | ISTAT 3 | ISTAT 2 | ISTAT 1 | ISTAT 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 | ISTATy | 端口输入状态位(y=0..15)<br>这些位由硬件置位和清除。<br>0：引脚输入信号为低电平<br>1：引脚输入信号为高电平 |

### 8.5.4. 端口输出控制寄存器 (GPIOx_OCTL, x=A..E)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OCTL15 | OCTL14 | OCTL13 | OCTL12 | OCTL11 | OCTL10 | OCTL9 | OCTL8 | OCTL7 | OCTL6 | OCTL5 | OCTL4 | OCTL3 | OCTL2 | OCTL1 | OCTL0 |
| 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 | OCTLy | 端口输出控制位(y=0..15)<br>这些位由软件置位和清除。<br>0：引脚输出低电平<br>1：引脚输出高电平 |

### 8.5.5. 端口位操作寄存器 (GPIOx_BOP, x=A..E)

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CR15 | CR14 | CR13 | CR12 | CR11 | CR10 | CR9 | CR8 | CR7 | CR6 | CR5 | CR4 | CR3 | CR2 | CR1 | CR0 |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BOP15 | BOP14 | BOP13 | BOP12 | BOP11 | BOP10 | BOP9 | BOP8 | BOP7 | BOP6 | BOP5 | BOP4 | BOP3 | BOP2 | BOP1 | BOP0 |

| 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 | CRy | 端口清除位y(y=0..15)<br>这些位由软件置位和清除。<br>0：相应的OCTLy位没有改变<br>1：清除相应的OCTLy位为0 |
| 15:0 | BOPy | 端口置位位y(y=0..15)<br>这些位由软件置位和清除。<br>0：相应的OCTLy位没有改变<br>1：设置相应的OCTLy位为1 |

### 8.5.6. 位清除寄存器 (GPIOx_BC, x=A..E)

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CR15 | CR14 | CR13 | CR12 | CR11 | CR10 | CR9 | CR8 | CR7 | CR6 | CR5 | CR4 | CR3 | CR2 | CR1 | CR0 |
| 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 | CRy | 端口清除位y(y=0..15)<br>这些位由软件置位和清除。<br>0：相应OCTLy位没有改变<br>1：清除相应的OCTLy位 |

### 8.5.7. 端口配置锁定寄存器 (GPIOx_LOCK, x=A..E)

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LK15 | LK14 | LK13 | LK12 | LK11 | LK10 | LK9 | LK8 | LK7 | LK6 | LK5 | LK4 | LK3 | LK2 | LK1 | LK0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:17 | 保留 | 必须保持复位值 |
| 16 | LKK | 锁定序列键<br>该位只能通过使用Lock Key写序列设置，始终可读。<br>0：GPIO_LOCK寄存器和端口配置没有锁定<br>1：直到下一次MCU复位前，GPIO_LOCK寄存器被锁定<br>LOCK Key写序列：<br>写1->写0->写1-> 读0-> 读1<br>注意：在LOCK Key写序列期间，LK[15:0]的值必须保持。 |
| 15:0 | LKy | 端口锁定位y(y=0..15)<br>这些位由软件置位和清除。<br>0：相应的端口位配置没有锁定<br>1：当LKK位置1时，相应的端口位配置被锁定 |

### 8.5.8. 端口位速度寄存器 (GPIOx_ SPD, x=A..E)

地址偏移：0x3C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SPD15 | SPD 14 | SPD 13 | SPD 12 | SPD 11 | SPD 10 | SPD 9 | SPD 8 | SPD 7 | SPD 6 | SPD 5 | SPD 4 | SPD 3 | SPD 2 | SPD 1 | SPD 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 | SPDy | 当MDx值为0b11时，设置相应端口速度为高速(120MHz)。<br>如果端口输出速度大于50MHz，该位置1，同时设置MDx值为0b11。这些位由软件置位和清除。<br>0: 没有影响<br>1: 最大输出速度大于50MHz(同时，需要设置MDx值为0b11)<br>注意：当端口输出速度大于50MHz时，需要使能I/O补偿单元。详见AFIO_CPSCTL寄存器中的CPS_EN位说明。 |

### 8.5.9. 事件控制寄存器 (AFIO_EC)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | EOE | PORT[2:0] | | | PIN[3:0] | | | |
| | | | | | | | | rw | rw | | | rw | | | |

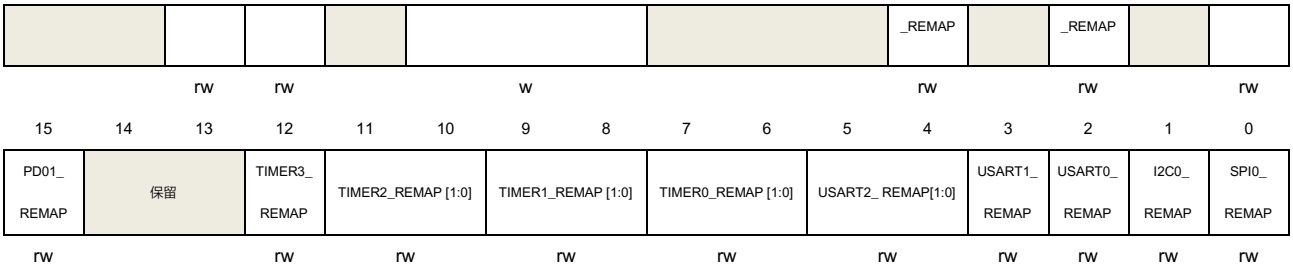
| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 必须保持复位值 |
| 7 | EOE | 事件输出使能<br>该位由软件置位和清除。当设置该位后，Cortex 的EVENTOUT输出将连接到由PORT[2:0]和PIN[3:0]位选择的I/O口。 |
| 6:4 | PORT[2:0] | 事件输出端口选择<br>这些位由软件置位和清除。选择用于输出Cortex的EVENTOUT信号的端口。<br>000： 选择端口A<br>001： 选择端口B<br>010： 选择端口C<br>011： 选择端口D<br>100： 选择端口E |
| 3:0 | PIN[3:0] | 事件输出引脚选择<br>这些位由软件置位和清除。选择用于输出Cortex的EVENTOUT信号的引脚。<br>0000： 选择引脚0<br>0001： 选择引脚1<br>0010： 选择引脚2<br>...<br>1111： 选择引脚15 |

### 8.5.10. AFIO 端口配置寄存器 0 (AFIO_PCF0)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | TIMER1ITR0_REMAP | SPI2_REMAP | 保留 | SWJ_CFG[2:0] | | | 保留 | | | ADC1_ETRGRT | 保留 | ADC0_ETRGRT | 保留 | TIMER4CH3_IREMAP |



| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | _REMAP | | _REMAP | | | |
| | | rw | rw | | | w | | | | rw | | rw | | rw | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PD01_ REMAP | 保留 | | TIMER3_ REMAP | TIMER2_REMAP [1:0] | | TIMER1_REMAP [1:0] | | TIMER0_REMAP [1:0] | | USART2_ REMAP[1:0] | | USART1_ REMAP | USART0_ REMAP | I2C0_ REMAP | SPI0_ REMAP |
| 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 | TIMER1ITI1_REMAP | TIMER1内部触发1重映射<br>该位由软件置位和清除，用于控制TMER1_ITI1内部重映射。<br>0：关闭重映射功能<br>1：TIMER1_ITI1 内部连接到USBFS SOF (起始帧) 输出，用于校准 |
| 28 | SPI2_REMAP | SPI2/I2S2重映射<br>该位由软件置位和清除。<br>0：关闭重映射功能(SPI2_NSS-I2S2_WS/PA15， SPI2_SCK-I2S2_CK/PB3，SPI2_MISO/PB4，SPI2_MOSI-I2S_SD/PB5)<br>1：完全开启重映射功能(SPI2_NSS-I2S2_WS/PA4，SPI2_SCK-I2S2_CK/PC10，SPI2_MISO/PC11，SPI2_MOSI-I2S_SD/PC12) |
| 27 | 保留 | 必须保持复位值 |
| 26:24 | SWJ_CFG[2:0] | 串行线 JTAG 配置<br>这些位只写(读这些位，将返回未定义值)。<br>000：JTAG-DP 使能和 SW-DP 使能(复位状态)<br>001：JTAG-DP 使能和 SW-DP 使能但没有 NJTRST<br>010：JTAG-DP 禁用和 SW-DP 使能<br>100：JTAG-DP 禁用和 SW-DP 禁用<br>其他：未定义<br>注意：每次只能设置三个比特中的一个。 |
| 23:21 | 保留 | 必须保持复位值 |
| 20 | ADC1_ETRGRT _REMAP | ADC 1 常规转换外部触发重映射<br>0：连接 ADC1 常规转换外部触发与 EXTI11<br>1：连接ADC1常规转换外部触发与TIM7_TRGO |
| 19 | 保留 | 必须保持复位值 |
| 18 | ADC0_ETRGRT_RE MAP | ADC 0 常规转换外部触发重映射<br>该位由软件置位和清除。<br>0：连接 ADC0 常规转换外部触发与 EXTI11<br>1：连接ADC0常规转换外部触发与TIM7_TRGO |
| 17 | 保留 | 必须保持复位值 |
| 16 | TIMER4CH3_IREMA | TIMER4 通道 3 内部重映射 |

| | | |
|---|---|---|
| | P | 该位由软件置位和清除。<br>0：连接 TIMER4_CH3 与 PA3<br>1：连接 TMER4_CH3 与 IRC40K 内部时钟，用于对 IRC40K 进行校准。<br>注意：该位在高密度和超高密度产品线中可用。 |
| 15 | PD01_REMAP | OSC_IN/OSC_OUT 重映射到 Port D0/Port D1<br>该位由软件置位和清除。<br>0：关闭重映射功能<br>1：OSC_IN重映射到PD0，OSC_OUT重映射到PD1 |
| 14:13 | 保留 | 必须保持复位值 |
| 12 | TIMER3_REMAP | TIMER3 重映射<br>该位由软件置位和清除。<br>0：关闭重映射功能(TIMER3_CH0/PB6，TIMER3_CH1/PB7，TIMER3_CH2/PB8，TIMER3_CH3/PB9)<br>1：完全开启重映射功能(TIMER3_CH0/PD12，TIMER3_CH1/PD13，TIMER3_CH2/PD14， TIMER3_CH3/PD15) |
| 11:10 | TIMER2_REMAP [1:0] | TIMER2 重映射<br>这些位由软件置位和清除。<br>00：关闭重映射功能 (TIMER2_CH0/PA6，TIMER2_CH1/PA7，TIMER2_CH2/PB0，TIMER2_CH3/PB1)<br>01：没有使用<br>10：开启重映射部分功能 (TIMER2_CH0/PB4， TIMER2_CH1/PB5，TIMER2_CH2/PB0， TIMER2_CH3/PB1)<br>11：完全开启重映射功能 (TIMER2_CH0/PC6，TIMER2_CH1/PC7，TIMER2_CH2/PC8， TIMER2_CH3/PC9) |
| 9:8 | TIMER1_REMAP [1:0] | TIMER1 重映射<br>这些位由软件置位和清除。<br>00：关闭重映射功能 (TIMER1_CH0/TIMER1_ETI/PA0， TIMER1_CH1/PA1，TIMER1_CH2/PA2，TIMER1_CH3/PA3)<br>01：开启重映射部分功能 (TIMER1_CH0/TIMER1_ETI/PA15，TIMER1_CH1/PB3，TIMER1_CH2/PA2，TIMER1_CH3/PA3)<br>10：开启重映射部分功能 (TIMER1_CH0/TIMER1_ETI/PA0，TIMER1_CH1/PA1，TIMER1_CH2/PB10，TIMER1_CH3/PB11)<br>11：完全开启重映射功能 (TIMER1_CH0/ TIMER1_ETI/PA15，TIMER1_CH1/PB3， TIMER1_CH2/PB10，TIMER1_CH3/PB11) |
| 7:6 | TIMER0_REMAP [1:0] | TIMER0 重映射<br>这些位由软件置位和清除。<br>00：关闭重映射功能(TIMER0_ETI/PA12，TIMER0_CH0/ PA8，TIMER0_CH1/PA9，TIMER0_CH2/PA10 ， TIMER0_CH3/PA11 ， TIMER0_BRKIN /PB12 ，TIMER0_CH0_ON/PB13，TIMER0_CH1_ON/PB14，TIMER0_CH2_ON /PB15)<br>01：开启重映射部分功能 (TIMER0_ETI/PA12， TIMER0_CH0/ PA8，TIMER0_CH1/PA9， TIMER0_CH2/PA10，TIMER0_CH3/PA11，TIMER0_BRKIN |

| | | |
|---|---|---|
| | | /PA6，TIMER0_CH0_ON/PA7，TIMER0_CH1_ON/PB0，TIMER0_CH2_ON/PB1)<br>10：没有使用<br>11：完全开启重映射功能(TIMER0_ETI/PE7，TIMER0_CH0/PE9，TIMER0_CH1/PE11，TIMER0_CH2/PE13，TIMER0_CH3/PE14，TIMER0_BRKIN/PE15，TIMER0_CH0_ON/PE8，TIMER0_CH1_ON/PE10，TIMER0_CH2_ON/PE12) |
| 5:4 | USART2_REMAP [1:0] | USART2 重映射<br>这些位由软件置位和清除。<br>00：关闭重映射功能(USART2_TX/PB10，USART2_RX /PB11，USART2_CK/PB12，USART2_CTS/PB13，USART2_RTS/PB14)<br>01：开启重映射部分功能(USART2_TX/PC10，USART2_RX /PC11，USART2_CK/PC12，USART2_CTS/PB13，USART2_RTS/PB14)<br>10：没有使用<br>11：完全开启重映射功能(USART2_TX/PD8，USART2_RX /PD9，USART2_CK /PD10，USART2_CTS/PD11，USART2_RTS/PD12) |
| 3 | USART1_REMAP | USART1 重映射<br>该位由软件置位和清除。<br>0：关闭重映射功能(USART1_CTS/PA0，USART1_RTS/PA1，USART1_TX/PA2，USART1_RX /PA3，USART1_CK/PA4)<br>1：开启重映射功能(USART1_CTS/PD3，USART1_RTS/PD4，USART1_TX/PD5，USART1_RX/PD6，USART1_CK/PD7) |
| 2 | USART0_REMAP | USART0 重映射<br>该位由软件置位和清除。<br>0：关闭重映射功能(USART0_TX/PA9，USART0_RX/PA10)<br>1：开启重映射功能(USART0_TX/PB6，USART0_RX/PB7) |
| 1 | I2C0_REMAP | I2C0重映射<br>该位由软件置位和清除。<br>0：关闭重映射功能(I2C0_SCL/PB6, I2C0_SDA/PB7)1<br>1：开启重映射功能(I2C0_SCL/PB8, I2C0_SDA/PB9) |
| 0 | SPI0_REMAP | SPI0重映射<br>该位由软件置位和清除。<br>0：关闭重映射功能(SPI0_NSS/PA4, SPI0_SCK/PA5, SPI0_MISO/PA6, SPI0_MOSI /PA7, SPI0_IO2 /PA2, SPI0_IO3 /PA3)<br>1：开启重映射功能(SPI0_NSS/PA15, SPI0_SCK/PB3, SPI0_MISO/PB4, SPI0_MOSI/PB5, SPI0_IO2 /PB6, SPI0_IO3 /PB7) |

### 8.5.11. EXTI 源选择寄存器 0 寄存器 (AFIO_ EXTISS0)

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

| 保留 |
|---|

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EXTI3_SS [3:0] | | | | EXTI2_SS [3:0] | | | | EXTI1_SS [3:0] | | | | EXTI0_SS [3:0] | | | |
| rw | | | | rw | | | | rw | | | | rw | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:12 | EXTI3_SS[3:0] | EXTI 3 源选择<br>0000： PA3 引脚<br>0001： PB3 引脚<br>0010： PC3 引脚<br>0011： PD3 引脚<br>0100： PE3 引脚<br>其他配置保留。 |
| 11:8 | EXTI2_SS[3:0] | EXTI 2 源选择<br>0000： PA2 引脚<br>0001： PB2 引脚<br>0010： PC2 引脚<br>0011： PD2 引脚<br>0100： PE2 引脚<br>其他配置保留。 |
| 7:4 | EXTI1_SS[3:0] | EXTI 1 源选择<br>0000： PA1 引脚<br>0001： PB1 引脚<br>0010： PC1 引脚<br>0011： PD1 引脚<br>0100： PE1 引脚<br>其他配置保留。 |
| 3:0 | EXTI0_SS[3:0] | EXTI 0 源选择<br>0000： PA0 引脚<br>0001： PB0 引脚<br>0010： PC0 引脚<br>0011： PD0 引脚<br>0100： PE0 引脚<br>其他配置保留。 |

### 8.5.12. EXTI 源选择寄存器 1 寄存器（AFIO_ EXTISS1）

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

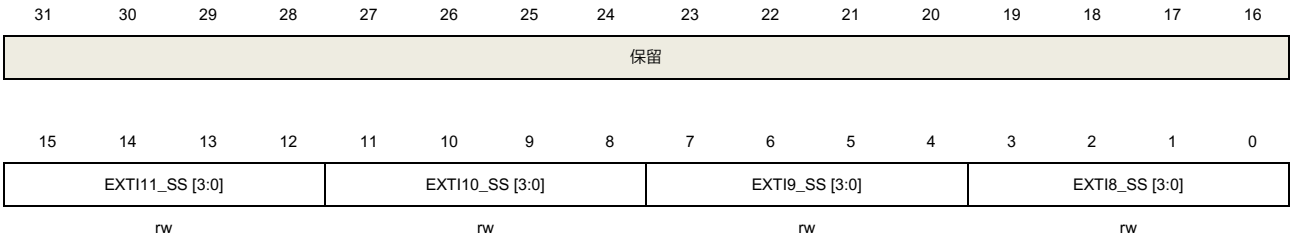
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EXTI7_SS [3:0] | | | | EXTI6_SS [3:0] | | | | EXTI5_SS [3:0] | | | | EXTI4_SS [3:0] | | | |
| rw | | | | rw | | | | rw | | | | rw | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:12 | EXTI7_SS[3:0] | EXTI 7 源选择<br>0000： PA7 引脚<br>0001： PB7 引脚<br>0010： PC7 引脚<br>0011： PD7 引脚<br>0100： PE7 引脚<br>其他配置保留。 |
| 11:8 | EXTI6_SS[3:0] | EXTI 6 源选择<br>0000： PA6 引脚<br>0001： PB6 引脚<br>0010： PC6 引脚<br>0011： PD6 引脚<br>0100： PE6 引脚<br>其他配置保留。 |
| 7:4 | EXTI5_SS[3:0] | EXTI 5 源选择<br>0000： PA5 引脚<br>0001： PB5 引脚<br>0010： PC5 引脚<br>0011： PD5 引脚<br>0100： PE5 引脚<br>其他配置保留。 |
| 3:0 | EXTI4_SS[3:0] | EXTI 4 源选择<br>0000： PA4 引脚<br>0001： PB4 引脚<br>0010： PC4 引脚<br>0011： PD4 引脚<br>0100： PE4 引脚<br>其他配置保留。 |

### 8.5.13. EXTI 源选择寄存器 2 寄存器 (AFIO_ EXTISS2)

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



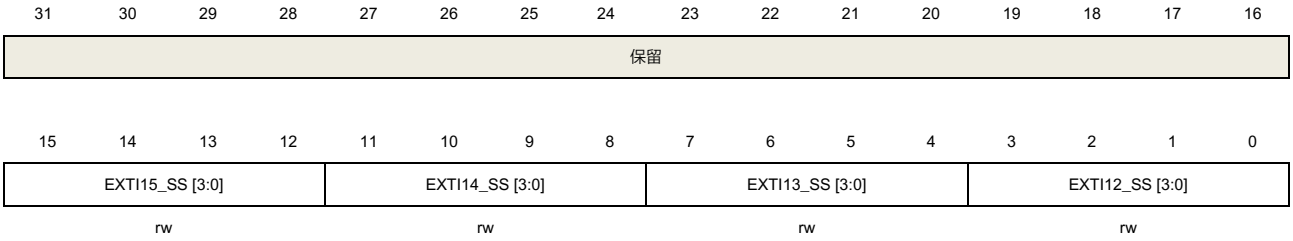

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:12 | EXTI11_SS[3:0] | EXTI 11 源选择<br>0000： PA11 引脚<br>0001： PB11 引脚<br>0010： PC11 引脚<br>0011： PD11 引脚<br>0100： PE11 引脚<br>其他配置保留。 |
| 11:8 | EXTI10_SS[3:0] | EXTI 10 源选择<br>0000： PA10 引脚<br>0001： PB10 引脚<br>0010： PC10 引脚<br>0011： PD10 引脚<br>0100: PE10 引脚<br>其他配置保留。 |
| 7:4 | EXTI9_SS[3:0] | EXTI 9 源选择<br>0000： PA9 引脚<br>0001： PB9 引脚<br>0010： PC9 引脚<br>0011： PD9 引脚<br>0100： PE9 引脚<br>其他配置保留。 |
| 3:0 | EXTI8_SS[3:0] | EXTI 8 源选择<br>0000： PA8 引脚<br>0001： PB8 引脚<br>0010： PC8 引脚<br>0011： PD8 引脚<br>0100： PE8 引脚 |

其他配置保留。

### 8.5.14. EXTI 源选择寄存器 3 寄存器 (AFIO_ EXTISS3)

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:12 | EXTI15_SS[3:0] | EXTI 15 源选择<br>0000： PA15 引脚<br>0001： PB15 引脚<br>0010： PC15 引脚<br>0011： PD15 引脚<br>0100： PE15 引脚<br>其他配置保留。 |
| 11:8 | EXTI14_SS[3:0] | EXTI 14 源选择<br>0000： PA14 引脚<br>0001： PB14 引脚<br>0010： PC14 引脚<br>0011： PD14 引脚<br>0100： PE14 引脚<br>其他配置保留。 |
| 7:4 | EXTI13_SS[3:0] | EXTI 13 源选择<br>0000： PA13 引脚<br>0001： PB13 引脚<br>0010： PC13 引脚<br>0011： PD13 引脚<br>0100： PE13 引脚<br>其他配置保留。 |
| 3:0 | EXTI12_SS[3:0] | EXTI 12 源选择<br>0000： PA12 引脚<br>0001： PB12 引脚<br>0010： PC12 引脚 |

0011： PD12 引脚
0100： PE12 引脚
其他配置保留。

## 8.5.15. AFIO 端口配置寄存器 1 (AFIO_ PCF1)

地址偏移：0x1C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

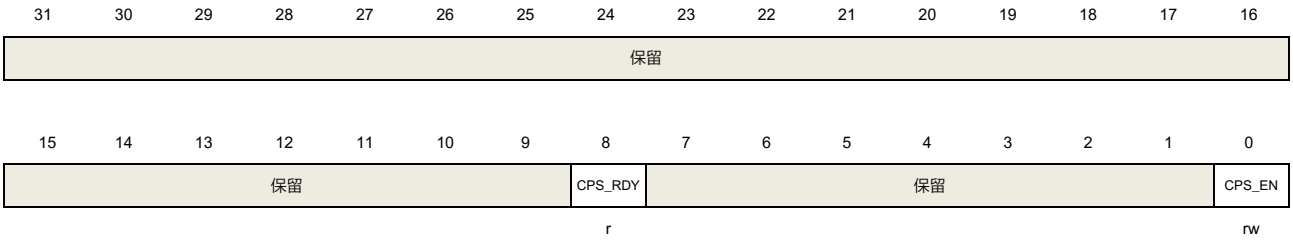



| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:13 | 保留 | 必须保持复位值 |
| 12:11 | CTC_REMAP [1:0] | CTC重映射<br>这些位由软件置位和清除，控制着将CTC_SYNC备用功能重映射到GPIO端口。<br>00：关闭重映射功能(PA8)<br>01：开启重映射功能(PD15)<br>10/11: 保留 |
| 10 | EXMC_NADV | EXMC_NADV 连接/不连接<br>该位由软件置位和清除，控制着可选的EXMC_NADV信号<br>0：NADV信号连接到输出(默认值)<br>1：NADV信号没有连接，I/O引脚可以用于其他外设。 |
| 9:6 | 保留 | 必须保持复位值 |
| 5 | TIMER8_REMAP | TIMER8 重映射<br>该位由软件置位和清除，控制着将TIMER8_CH0和TIMER8_CH1备用功能重映射到GPIO端口。<br>0：关闭重映射功能(TIMER8_CH0连接到PA2和TIMER8_CH1连接到 PA3)<br>1：开启重映射功能(TIMER8_CH0重映射到PE5和TIMER8_CH1重映射到PE6) |
| 4:0 | 保留 | 必须保持复位值 |

## 8.5.16. IO 补偿控制寄存器 (AFIO_CPSCTL)

地址偏移：0x20
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:9 | 保留 | 必须保持复位值 |
| 8 | CPS_RDY | I/O补偿单元是否准备好，该位只读。<br>0： I/O补偿单元没有准备好<br>1： I/O补偿单元准备好 |
| 7:1 | 保留 | 必须保持复位值 |
| 0 | CPS_EN | I/O补偿单元使能<br>当端口输出速度大于50MHz时，需要使能I/O补偿单元。<br>0： I/O补偿单元掉电<br>1： I/O补偿单元使能 |

# 9. 循环冗余校验计算单元(CRC)

## 9.1. 简介

循环冗余校验码是一种用在数字网络和存储设备上的差错校验码，可以校验原始数据的偶然差错。

CRC 计算单元使用固定多项式计算 32 位 CRC 校验码。

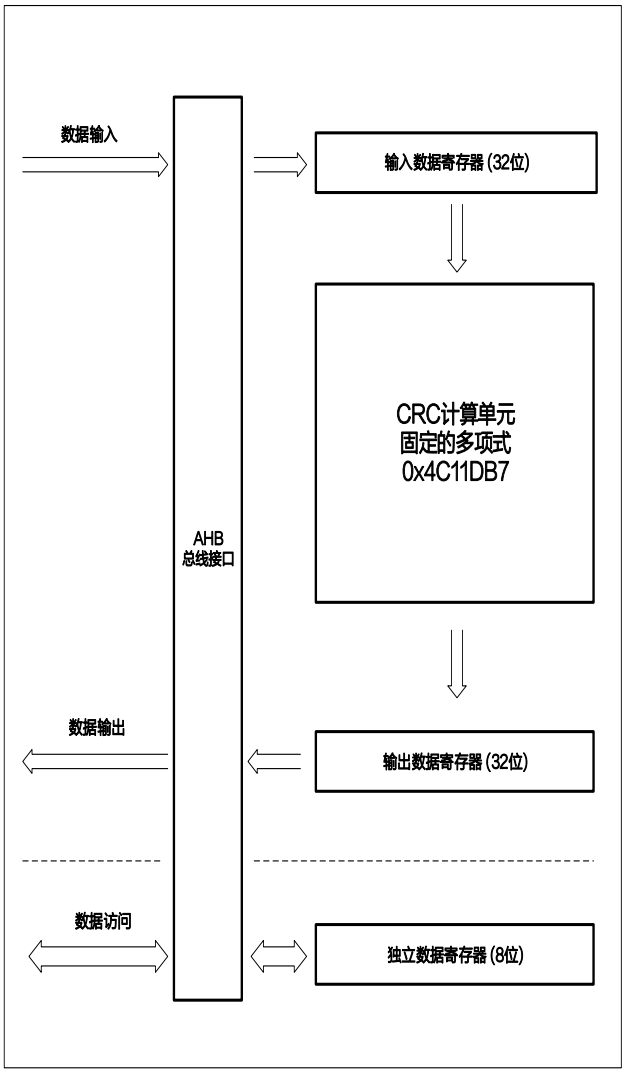
## 9.2. 主要特征

- 32位数据输入/输出寄存器。对于32位的输入数据，从数据输入到得出计算结果，需要4个AHB的时钟周期；
- 配有与计算无关的独立8位寄存器，可以供其他任何外设使用；
- 固定的计算多项式：0x4C11DB7：

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

该 32 位 CRC 多项式与以太网 CRC 计算多项式相同。

图 9-1. CRC 计算单元框图




## 9.3. 功能说明

- CRC计算单元可以用来计算32位的原始数据，CRC_DATA寄存器接收原始数据并存储计算结果。

  如果不通过软件设置CRC_CTL寄存器的方式来清除CRC_DATA寄存器，CRC计算单元将基于新输入的原始数据和前一次CRC_DATA寄存器中的结果进行计算。

  对于32位的数据，CRC的计算需要4个AHB的时钟周期。在此期间，因为32位输入缓存的存在，AHB总线将不会被挂起。

- 此模块提供了一个8位的独立数据寄存器CRC_FDATA。

  CRC_FDATA与CRC计算无关，任何时候都可以进行独立的读写操作。

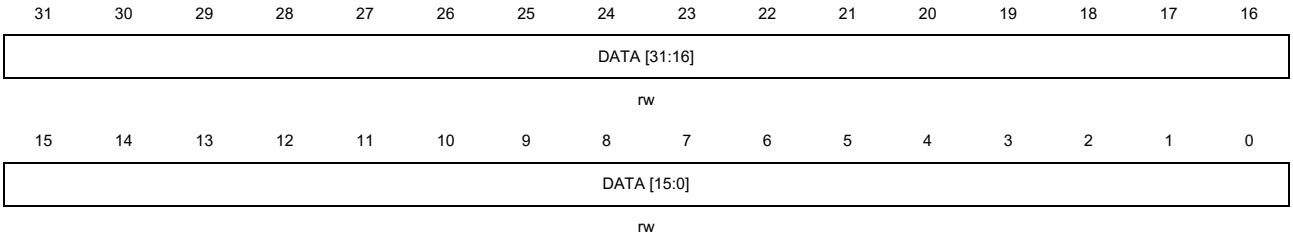
## 9.4. CRC 寄存器

CRC基地址：0x4002 3000

### 9.4.1. 数据寄存器(CRC_DATA)

地址偏移：0x00
复位值：0xFFFF FFFF

该寄存器只能按字(32位)访问

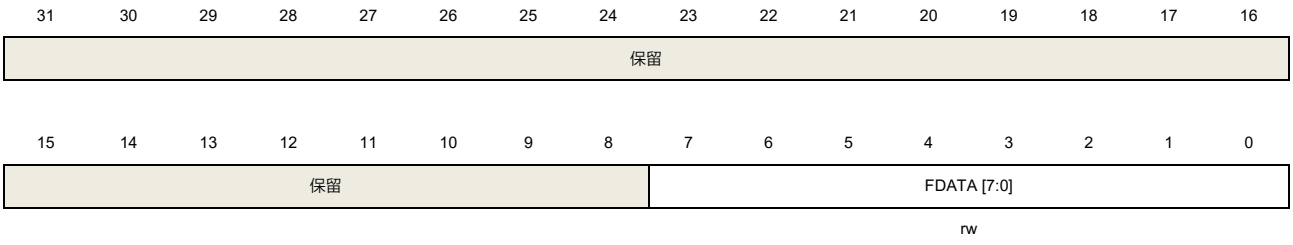



| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | DATA[31:0] | CRC计算结果位<br>软件可读可写<br>该寄存器用于接收待计算的新数据，直接将其写入即可。刚写入的数据不能被读出来，因为读取该寄存器得到的是上次CRC计算的结果。 |

### 9.4.2. 独立数据寄存器(CRC_FDATA)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32位)访问



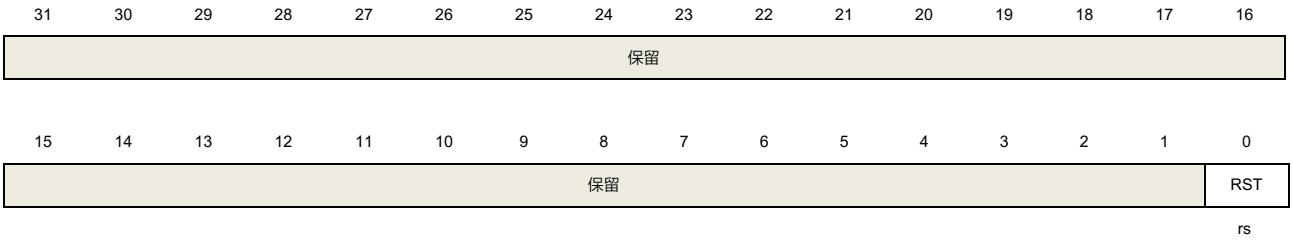

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 保持复位值 |
| 7:0 | FDATA[7:0] | 独立数据寄存器位<br>软件可读可写<br>这些位与CRC计算无关。该字节能被任何其他外设用于其他任何目的。该字节不受CRC_CTL |

寄存器的影响。

### 9.4.3. 控制寄存器(CRC_CTL)

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字(32位)访问




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:1 | 保留 | 保持复位值 |
| 0 | RST | 将该位置1可以复位CRC_DATA寄存器，并设置其值为0xFFFFFFFF，然后该位被硬件自动清零。该位对CRC_FDATA寄存器没有影响。<br>软件可读可写。 |

# 10. 直接存储器访问控制器(DMA)

## 10.1. 简介

DMA 控制器提供了一种硬件的方式在外设和存储器之间或者存储器和存储器之间传输数据，而无需 CPU 的介入，从而释放带宽，使 CPU 可以专注在处理其他系统功能上。数据可以通过 DMA 在外设和内存以及内存与内存之间快速移动，而无需任何 CPU 操作。DMA 控制器有 12 个通道(DMA0 有 7 个通道，DMA1 有 5 个通道)。每个通道都是专门用来处理一个或多个外设的存储器访问请求的。DMA 控制器内部实现了一个仲裁器，用来仲裁多个 DMA 请求的优先级。

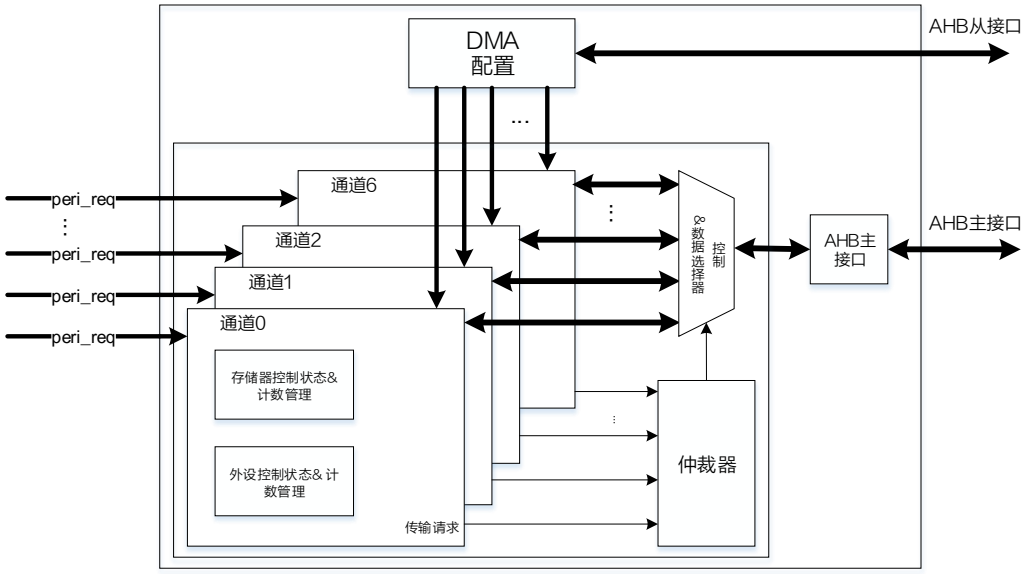
DMA 控制器和 Cortex®-M4 内核共享系统总线。当 DMA 和 CPU 访问同样的地址空间时，DMA 访问可能会阻挡 CPU 访问系统总线几个总线周期。总线矩阵中实现了循环仲裁算法来分配 DMA 与 CPU 的访问权，它可以确保 CPU 得到至少一半的系统总线带宽。

## 10.2. 主要特征

- 传输数据长度可编程配置，最大到 65536；
- 12 个通道，并且每个通道都可配置(DMA0 有 7 个通道，DMA1 有 5 个通道)；
- AHB 和 APB 外设，片上闪存和 SRAM 都可以作为访问的源端和目的端；
- 每个通道连接固定的硬件 DMA 请求；
- 支持软件优先级(低、中、高、极高)和硬件优先级(通道号越低，优先级越高)；
- 存储器和外设的数据传输宽度可配置：字节，半字，字；
- 存储器和外设的数据传输支持固定寻址和增量式寻址；
- 支持循环传输模式；
- 支持外设到存储器，存储器到外设，存储器到存储器的数据传输；
- 每个通道有 3 种类型的事件标志和独立的中断；
- 支持中断的使能和清除。

## 10.3. 结构框图

图 10-1. DMA 结构框图




由*图 10-1. DMA 结构框图*所示，DMA 控制器由 4 部分组成：

- AHB 从接口配置 DMA
- AHB 主接口进行数据传输，用于存储器访问和外设访问
- 仲裁器 DMA 请求的优先级管理
- 通道管理用于控制数据/地址选择和数据计数

## 10.4. 功能说明

### 10.4.1. DMA 操作

DMA 传输分为两步操作：从源地址读取数据，之后将读取的数据存储到目的地址。DMA 控制器基于 DMA_CHxPADDR、DMA_CHxMADDR、DMA_CHxCTL 寄存器的值计算下一次操作的源/目的地址。DMA_CHxCNT 寄存器用于控制传输的次数。DMA_CHxCTL 寄存器的 PWIDTH 和 MWIDTH 位域决定每次发送和接收的字节数(字节/半字/字)。

假设 DMA_CHxCNT 寄存器的值为 4，并且 PNAGA 和 MNAGA 位均置位。结合 PWIDTH 和 MWIDTH 的各种配置，DMA 传输的操作详见*表 10-1. DMA 传输操作*。

表 10-1. DMA 传输操作

| 传输宽度 | | 传输操作 | |
|---|---|---|---|
| 源 | 目标 | 源 | 目标 |
| 32 bits | 32 bits | 1: Read B3B2B1B0[31:0] @0x0<br>2: Read B7B6B5B4[31:0] @0x4<br>3: Read BBBAB9B8[31:0] @0x8<br>4: Read BFBEBDBC[31:0] @0xC | 1: Write B3B2B1B0[31:0] @0x0<br>2: Write B7B6B5B4[31:0] @0x4<br>3: Write BBBAB9B8[31:0] @0x8<br>4: Write BFBEBDBC[31:0] @0xC |

| 传输宽度 | | 传输操作 | |
|---|---|---|---|
| 源 | 目标 | 源 | 目标 |
| 32 bits | 16 bits | 1: Read B3B2B1B0[31:0] @0x0<br>2: Read B7B6B5B4[31:0] @0x4<br>3: Read BBBAB9B8[31:0] @0x8<br>4: Read BFBEBDBC[31:0] @0xC | 1: Write B1B0[7:0] @0x0<br>2: Write B5B4[7:0] @0x2<br>3: Write B9B8[7:0] @0x4<br>4: Write BDBC[7:0] @0x6 |
| 32 bits | 8 bits | 1: Read B3B2B1B0[31:0] @0x0<br>2: Read B7B6B5B4[31:0] @0x4<br>3: Read BBBAB9B8[31:0] @0x8<br>4: Read BFBEBDBC[31:0] @0xC | 1: Write B0[7:0] @0x0<br>2: Write B4[7:0] @0x1<br>3: Write B8[7:0] @0x2<br>4: Write BC[7:0] @0x3 |
| 16 bits | 32 bits | 1: Read B1B0[15:0] @0x0<br>2: Read B3B2[15:0] @0x2<br>3: Read B5B4[15:0] @0x4<br>4: Read B7B6[15:0] @0x6 | 1: Write 0000B1B0[31:0] @0x0<br>2: Write 0000B3B2[31:0] @0x4<br>3: Write 0000B5B4[31:0] @0x8<br>4: Write 0000B7B6[31:0] @0xC |
| 16 bits | 16 bits | 1: Read B1B0[15:0] @0x0<br>2: Read B3B2[15:0] @0x2<br>3: Read B5B4[15:0] @0x4<br>4: Read B7B6[15:0] @0x6 | 1: Write B1B0[15:0] @0x0<br>2: Write B3B2[15:0] @0x2<br>3: Write B5B4[15:0] @0x4<br>4: Write B7B6[15:0] @0x6 |
| 16 bits | 8 bits | 1: Read B1B0[15:0] @0x0<br>2: Read B3B2[15:0] @0x2<br>3: Read B5B4[15:0] @0x4<br>4: Read B7B6[15:0] @0x6 | 1: Write B0[7:0] @0x0<br>2: Write B2[7:0] @0x1<br>3: Write B4[7:0] @0x2<br>4: Write B6[7:0] @0x3 |
| 8 bits | 32 bits | 1: Read B0[7:0] @0x0<br>2: Read B1[7:0] @0x1<br>3: Read B2[7:0] @0x2<br>4: Read B3[7:0] @0x3 | 1: Write 000000B0[31:0] @0x0<br>2: Write 000000B1[31:0] @0x4<br>3: Write 000000B2[31:0] @0x8<br>4: Write 000000B3[31:0] @0xC |
| 8 bits | 16 bits | 1: Read B0[7:0] @0x0<br>2: Read B1[7:0] @0x1<br>3: Read B2[7:0] @0x2<br>4: Read B3[7:0] @0x3 | 1, Write 00B0[15:0] @0x0<br>2, Write 00B1[15:0] @0x2<br>3, Write 00B2[15:0] @0x4<br>4, Write 00B3[15:0] @0x6 |
| 8 bits | 8 bits | 1: Read B0[7:0] @0x0<br>2: Read B1[7:0] @0x1<br>3: Read B2[7:0] @0x2<br>4: Read B3[7:0] @0x3 | 1, Write B0[7:0] @0x0<br>2, Write B1[7:0] @0x1<br>3, Write B2[7:0] @0x2<br>4, Write B3[7:0] @0x3 |

DMA_CHxCNT寄存器的CNT位域必须在CHEN位置位前被配置，其控制传输的次数。在传输过程中，CNT位域的值表示还有多少次数据传输将被执行。

将 DMA_CHxCTL 寄存器的 CHEN 位清零，可以停止 DMA 传输

- 若 CHEN 位被清零时 DMA 传输还未完成，重新使能 CHEN 位将分两种情况：
  - 在重新使能 DMA 通道前，未对该通道的相关寄存器进行操作，则 DMA 将继续完成上次的传输。
  - 在重新使能 DMA 通道前，对任意配置相关寄存器(对应通道的 DMA_CHxCNT / DMA_CHxPADDR / DMA_CHxMADDR 寄存器)进行了操作，则 DMA 将开始一次新的传输。
- 若清零 CHEN 位时，DMA 传输已经完成，之后未对任意寄存器(对应通道的 DMA_CHxCNT / DMA_CHxPADDR / DMA_CHxMADDR 寄存器)进行操作前便使能 DMA 通道，则不会触发任何 DMA 传输。

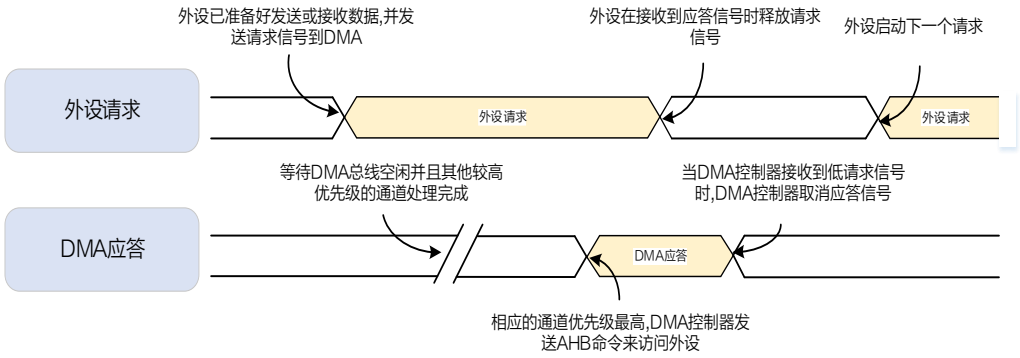
### 10.4.2. 外设握手

为了保证数据的有效传输，DMA控制器中引入了外设和存储器的握手机制，包括请求信号和应答信号：

- 请求信号：由外设发出，表明外设已经准备好发送或接收数据；
- 应答信号：由 DMA 控制器响应，表明 DMA 控制器已经发送 AHB 命令去访问外设。

*图10-2. 握手机制*中详细描述了DMA控制器与外设之间的握手机制。

**图 10-2. 握手机制**




### 10.4.3. 仲裁

当DMA控制器在同一时间接收到多个外设请求时，仲裁器将根据外设请求的优先级来决定响应哪一个外设请求。优先级包括软件优先级和硬件优先级，优先级规则如下：

- 软件优先级：分为4级，低，中，高和极高。可以通过寄存器DMA_CHxCTL的PRIO位域来配置。
- 硬件优先级：当通道具有相同的软件优先级时，编号低的通道优先级高。例：通道0和通道2配置为相同的软件优先级时，通道0的优先级高于通道2。

### 10.4.4. 地址生成

存储器和外设都独立的支持两种地址生成算法：固定模式和增量模式。寄存器DMA_CHxCTL的PNAGA和MNAGA位用来设置存储器和外设的地址生成算法。

在固定模式中，地址一直固定为初始化的基地址(DMA_CHxPADDR，DMA_CHxMADDR)。

在增量模式中，下一次传输数据的地址是当前地址加1(或者2，4)，这个值取决于数据传输宽度。

### 10.4.5. 循环模式

循环模式用来处理连续的外设请求(如ADC扫描模式)。将DMA_CHxCTL寄存器的CMEN位置位可以使能循环模式。

在循环模式中，当每次DMA传输完成后，CNT值会被重新载入，且传输完成标志位会被置1。DMA会一直响应外设的请求，直到通道使能位(DMA_CHxCTL寄存器的CHEN位)被清0。

### 10.4.6. 存储器到存储器模式

将DMA_CHxCTL寄存器的M2M位置位可以使能存储器到存储器模式。在此模式下，DMA通道传输数据时不依赖外设的请求信号。一旦DMA_CHxCTL寄存器的CHEN位被置1，DMA通道就立即开始传输数据，直到DMA_CHxCNT寄存器达到0，DMA通道才会停止。

### 10.4.7. 通道配置

要启动一次新的 DMA 数据传输，建议遵循以下步骤进行操作：

1. 读取 CHEN 位，判断通道是否使能。如果为 1(通道已使能)，清零该位。当 CHEN 为 0 时，请按照下列步骤配置 DMA，启动新的传输。
2. 配置 DMA_CHxCTL 寄存器的 M2M 及 DIR 位，选择传输模式。
3. 配置 DMA_CHxCTL 寄存器的 CMEN 位，选择是否使能循环模式。
4. 配置 DMA_CHxCTL 寄存器的 PRIO 位域，选择该通道的软件优先级。
5. 通过 DMA_CHxCTL 寄存器配置存储器和外设的传输宽度以及存储器和外设地址生成算法。
6. 通过 DMA_CHxCTL 寄存器配置传输完成中断，半传输完成中断，传输错误中断的使能位。
7. 通过 DMA_CHxPADDR 寄存器配置外设基地址。
8. 通过 DMA_CHxMADDR 寄存器配置存储器基地址。
9. 通过 DMA_CHxCNT 寄存器配置数据传输总量。
10. 将 DMA_CHxCTL 寄存器的 CHEN 位置 1，使能 DMA 通道。

### 10.4.8. 中断

每个DMA通道都有一个专用的中断。中断事件有三种类型：传输完成，半传输完成和传输错误。

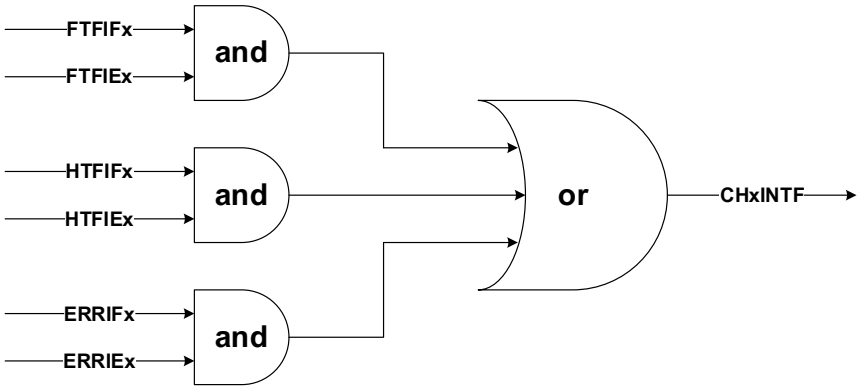
每一个中断事件在DMA_INTF寄存器中有专用的标志位，在DMA_INTC寄存器中有专用的清除位，在DMA_CHxCTL寄存器中有专用的使能位。*表10-2. 中断事件*描述了其对应关系。

**表 10-2. 中断事件**

| 中断事件 | 标志位 | 清除位 | 使能位 |
|---|---|---|---|
| | DMA_INTF | DMA_INTC | DMA_CHxCTL |
| 传输完成(Full transfer finish) | FTFIF | FTFIFC | FTFIE |
| 半传输完成(Half transfer finish) | HTFIF | HTFIFC | HTFIE |
| 传输错误(Transfer error) | ERRIF | ERRIFC | ERRIE |

DMA中断逻辑如*图10-3. DMA中断逻辑图*所示，任何类型中断使能时，产生了相应中断事件均会产生中断。

图 10-3. DMA 中断逻辑图




**注意：**"x"表示通道数(DMA0对应x=0…6；DMA1对应x=0…4)

### 10.4.9. DMA 请求映射

多个外设请求被映射到同一个 DMA 通道。这些请求信号在经过逻辑或后进入 DMA。详情可见*图 10-4. DMA0 请求映射*和*图 10-5. DMA1 请求映射*。通过配置对应外设的寄存器，每个外设的请求均可以独立的开启或关闭。用户必须确保同一时间，在同一个通道上仅有一个外设的请求被开启。*表 10-3. DMA0 各通道请求表*列举了 DMA0 的每个通道所支持的外设请求，*表 10-4. DMA1 各通道请求表*列举了 DMA1 的每个通道所支持的外设请求。

**图 10-4. DMA0 请求映射**

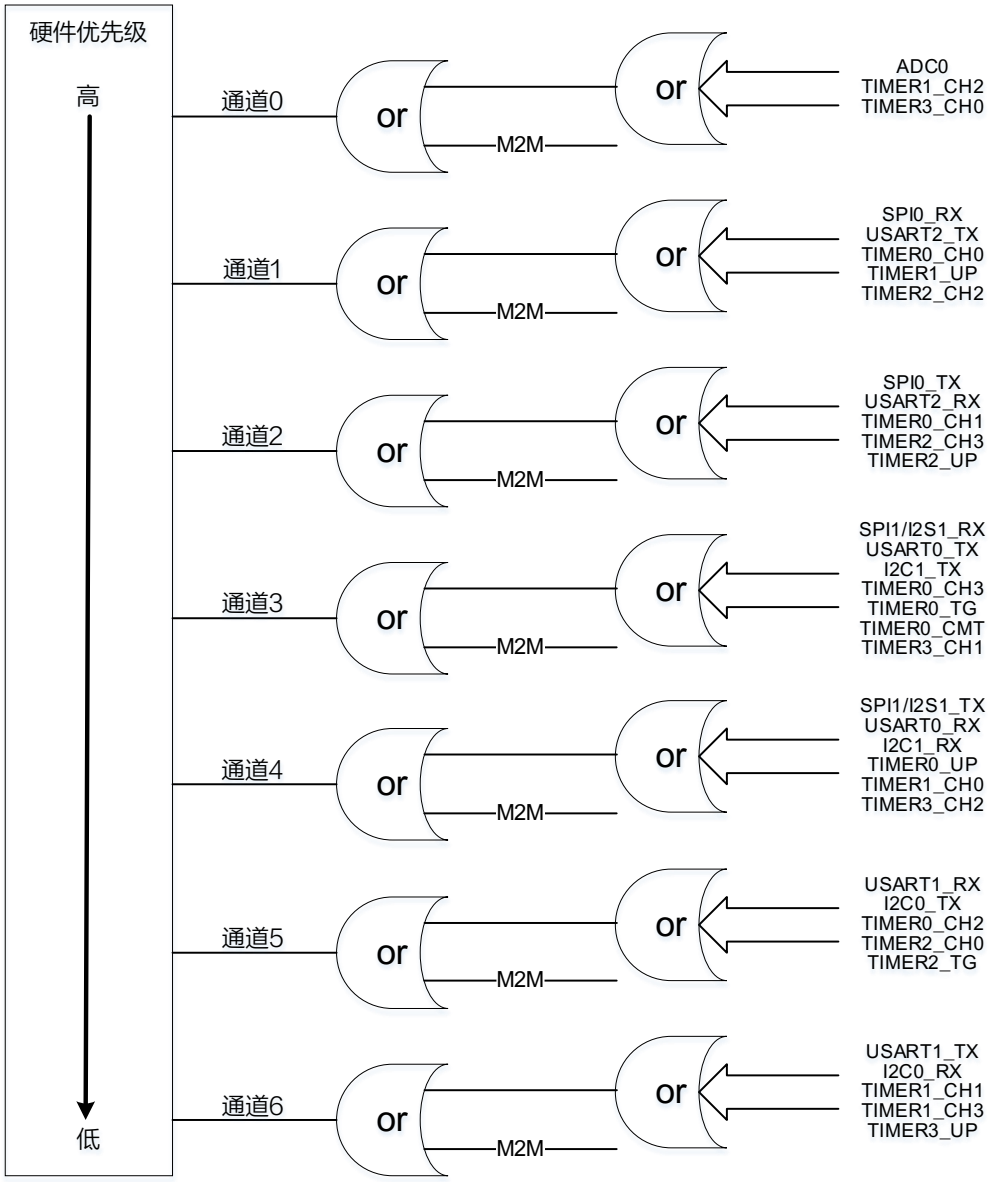



**表10-3. DMA0各通道请求表**

| 外设 | 通道0 | 通道1 | 通道2 | 通道3 | 通道4 | 通道5 | 通道6 |
|---|---|---|---|---|---|---|---|
| TIMER0 | • | TIMER0_CH0 | TIMER0_CH1 | TIMER0_CH3<br>TIMER0_TG<br>TIMER0_CMT | TIMER0_UP | TIMER0_CH2 | • |
| TIMER1 | TIMER1_CH2 | TIMER1_UP | • | • | TIMER1_CH0 | • | TIMER1_CH1<br>TIMER1_CH3 |
| TIMER2 | • | TIMER2_CH2 | TIMER2_CH3<br>TIMER2_UP | • | • | TIMER2_CH0<br>TIMER2_TG | • |
| TIMER3 | TIMER3_CH0 | • | • | TIMER3_CH1 | TIMER3_CH2 | • | TIMER3_UP |
| ADC0 | ADC0 | • | • | • | • | • | • |

| 外设 | 通道0 | 通道1 | 通道2 | 通道3 | 通道4 | 通道5 | 通道6 |
|---|---|---|---|---|---|---|---|
| SPI/I2S | • | SPI0_RX | SPI0_TX | SPI1/I2S1_RX | SPI1/I2S1_TX | • | • |
| USART | • | USART2_TX | USART2_RX | USART0_TX | USART0_RX | USART1_RX | USART1_TX |
| I2C | • | • | • | I2C1_TX | I2C1_RX | I2C0_TX | I2C0_RX |

**图10-5. DMA1请求映射**

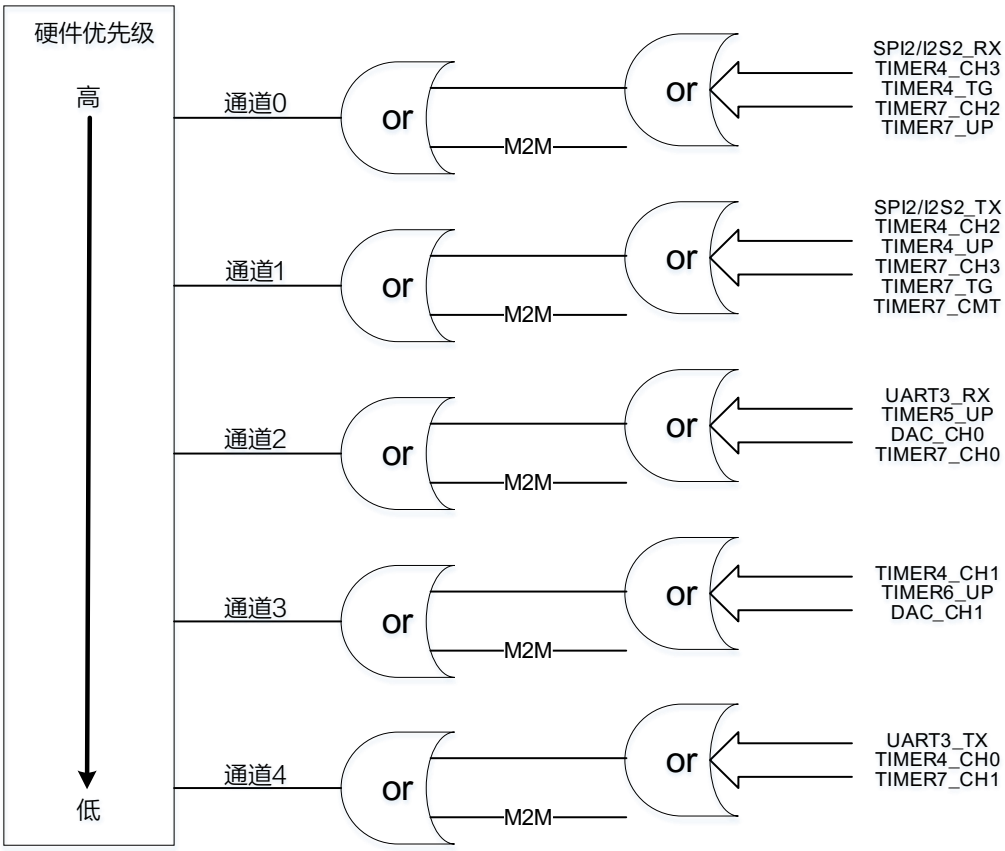



**表10-4. DMA1各通道请求表**

| 外设 | 通道0 | 通道1 | 通道2 | 通道3 | 通道4 |
|---|---|---|---|---|---|
| TIMER4 | TIMER4_CH3<br>TIMER4_TG | TIMER4_CH2<br>TIMER4_UP | • | TIMER4_CH1 | TIMER4_CH0 |
| TIMER5 | • | • | TIMER5_UP | • | • |
| TIMER6 | • | • | • | TIMER6_UP | • |
| TIMER7 | TIMER7_CH2<br>TIMER7_UP | TIMER7_CH3<br>TIMER7_TG<br>TIMER7_CMT | TIMER7_CH0 | • | TIMER7_CH1 |
| DAC | • | • | DAC_CH0 | DAC_CH1 | • |
| SPI/I2S | SPI2/I2S2_RX | SPI2/I2S2_TX | • | • | • |
| USART | • | • | UART3_RX | • | UART3_TX |

## 10.5. DMA 寄存器

DMA0 基地址：0x4002 0000
DMA1 基地址：0x4002 0400

**注意：**DMA1 仅有五个通道(0 到 4 通道)，所有相关寄存器中通道 5 和通道 6 相关标志位不适用于 DMA1。

### 10.5.1. 中断标志位寄存器(DMA_INTF)

地址偏移：0x00
复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | ERRIF6 | HTFIF6 | FTFIF6 | GIF6 | ERRIF5 | HTFIF5 | FTFIF5 | GIF5 | ERRIF4 | HTFIF4 | FTFIF4 | GIF4 |
| | | | | 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 |
| ERRIF3 | HTFIF3 | FTFIF3 | GIF3 | ERRIF2 | HTFIF2 | FTFIF2 | GIF2 | ERRIF1 | HTFIF1 | FTFIF1 | GIF1 | ERRIF0 | HTFIF0 | FTFIF0 | GIF0 |
| r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:28 | 保留 | 必须保持复位值 |
| 27/23/19/<br>15/11/7/3 | ERRIFx | 通道x错误标志位(x=0…6)<br>硬件置位，软件写DMA_INTC相应位为1清零<br>0：通道x未发生传输错误<br>1：通道x发生传输错误 |
| 26/22/18/<br>14/10/6/2 | HTFIFx | 通道x半传输完成标志位(x=0…6)<br>硬件置位，软件写DMA_INTC相应位为1清零<br>0：通道x半传输未完成<br>1：通道x半传输完成 |
| 25/21/17/<br>13/9/5/1 | FTFIFx | 通道x传输完成标志位(x=0…6)<br>硬件置位，软件写DMA_INTC相应位为1清零<br>0：通道x传输未完成<br>1：通道x传输完成 |
| 24/20/16/<br>12/8/4/0 | GIFx | 通道x全局中断标志位(x=0…6)<br>硬件置位，软件写DMA_INTC相应位为1清零<br>0：通道x ERRIF, HTFIF或FTFIF标志位未置位<br>1：通道x至少发生ERRIF, HTFIF或FTFIF之一置位 |

### 10.5.2. 中断标志位清除寄存器(DMA_INTC)

地址偏移：0x04
复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | ERRIFC6 | HTFIFC6 | FTFIFC6 | GIFC6 | ERRIFC5 | HTFIFC5 | FTFIFC5 | GIFC5 | ERRIFC4 | HTFIFC4 | FTFIFC4 | GIFC4 |
| | | | | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ERRIFC3 | HTFIFC3 | FTFIFC3 | GIFC3 | ERRIFC2 | HTFIC2 | FTFIFC2 | GIFC2 | ERRIFC1 | HTFIFC1 | FTFIFC1 | GIFC1 | ERRIFC0 | HTFIFC0 | FTFIFC0 | GIFC0 |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:28 | 保留 | 必须保持复位值 |
| 27/23/19/<br>15/11/7/3 | ERRIFCx | 清除通道x(x=0…6)的错误标志位<br>0：无影响<br>1：清零DMA_INTF寄存器的ERRIFx位 |
| 26/22/18/<br>14/10/6/2 | HTFIFCx | 清除通道x(x=0…6)的半传输完成标志位<br>0：无影响<br>1：清零DMA_INTF寄存器的HTFIFx位 |
| 25/21/17/<br>13/9/5/1 | FTFIFCx | 清除通道x(x=0…6)的传输完成标志位<br>0：无影响<br>1：清零DMA_INTF寄存器的FTFIFx位 |
| 24/20/16/<br>12/8/4/0 | GIFCx | 清除通道x(x=0…6)的全局中断标志位<br>0：无影响<br>1：清零DMA_INTF寄存器的GIFx，ERRIFx，HTFIFx和FTFIFx位 |

### 10.5.3. 通道 x 控制寄存器(DMA_CHxCTL)

x = 0...6, x 为通道序号

地址偏移：0x08 + 0x14 × x
复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | M2M | PRIO[1:0] | | MWIDTH[1:0] | | PWIDTH[1:0] | | MNAGA | PNAGA | CMEN | DIR | ERRIE | HTFIE | FTFIE | CHEN |
| | 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 | M2M | 存储器到存储器模式<br>软件置位和清零<br>0：禁止存储器到存储器模式<br>1：使能存储器到存储器模式<br>CHEN位为1时，该位不能被配置 |
| 13:12 | PRIO[1:0] | 软件优先级<br>软件置位和清零<br>00：低<br>01：中<br>10：高<br>11：极高<br>CHEN位为1时，该位域不能被配置 |
| 11:10 | MWIDTH[1:0] | 存储器的传输数据宽度<br>软件置位和清零<br>00：8-bit<br>01：16-bit<br>10：32-bit<br>11：保留<br>CHEN位为1时，该位域不能被配置 |
| 9:8 | PWIDTH[1:0] | 外设的传输数据宽度<br>软件置位和清零<br>00：8-bit<br>01：16-bit<br>10：32-bit<br>11：保留<br>CHEN位为1时，该位域不能被配置 |
| 7 | MNAGA | 存储器的地址生成算法<br>软件置位和清零<br>0：固定地址模式<br>1：增量地址模式<br>CHEN位为1时，该位不能被配置 |
| 6 | PNAGA | 外设的地址生成算法<br>软件置位和清零<br>0：固定地址模式<br>1：增量地址模式<br>CHEN位为1时，该位不能被配置 |
| 5 | CMEN | 循环模式使能<br>软件置位和清零<br>0：禁止循环模式 |

| | | |
|---|---|---|
| | | 1：使能循环模式<br>CHEN位为1时，该位不能被配置 |
| 4 | DIR | 传输方向<br>软件置位和清零<br>0：从外设读出并写入存储器<br>1：从存储器读出并写入外设<br>CHEN位为1时，该位不能被配置 |
| 3 | ERRIE | 通道错误中断使能位<br>软件置位和清零<br>0：禁止通道错误中断<br>1：使能通道错误中断 |
| 2 | HTFIE | 通道半传输完成中断使能位<br>软件置位和清零<br>0：禁止通道半传输完成中断<br>1：使能通道半传输完成中断 |
| 1 | FTFIE | 通道传输完成中断使能位<br>软件置位和清零<br>0：禁止通道传输完成中断<br>1：使能通道传输完成中断 |
| 0 | CHEN | 通道使能<br>软件置位和清零<br>0：禁止该通道<br>1：使能该通道 |

### 10.5.4. 通道 x 计数寄存器(DMA_CHxCNT)

x = 0...6, x 为通道序号

地址偏移：0x0C + 0x14 × x

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | CNT[15:0] | 传输计数 |

CHEN位为1时，该位域不能被配置

该寄存器标明还有多少数据等待被传输。一旦通道使能，该寄存器为只读的，并在每个DMA传输之后值减1。如果该寄存器的值为0，无论通道开启与否，都不会有数据传输。如果该通道工作在循环模式下，一旦通道的传输任务完成，该寄存器会被自动重装载为初始设置值。

### 10.5.5. 通道 x 外设基地址寄存器(DMA_CHxPADDR)

x = 0...6, x 为通道序号

地址偏移：0x10 + 0x14 × x

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | PADDR[31:0] | 外设基地址<br>CHEN位为1时，该位域不能被配置<br>当PWIDTH位域的值为01 (16-bit),，PADDR[0]被忽略，访问自动与16位地址对齐。<br>当PWIDTH位域的值为10 (32-bit)，PADDR [1:0]被忽略，访问自动与32位地址对齐。 |

### 10.5.6. 通道 x 存储器基地址寄存器(DMA_CHxMADDR)

x = 0...6, x 为通道序号

地址偏移：0x14 + 0x14 × x

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | MADDR[31:0] | 存储器基地址<br>CHEN位为1时，该位域不能被配置<br>当MWIDTH位域的值为01 (16-bit)时，MADDR [0]被忽略，访问自动与16位地址对齐。 |

当MWIDTH位域的值为10 (32-bit)时，MADDR [1:0]被忽略，访问自动与32位地址对齐。

# 11. 调试(DBG)

## 11.1. 简介

GD32E11x系列产品提供了各种各样的调试、跟踪和测试功能。这些功能通过ARM CoreSight™组件的标准配置和链状连接的TAP控制器来实现的。调试和跟踪功能集成在ARM Cortex-M4内核中。调试系统支持串行(SW)调试和跟踪功能，也支持JTAG调试。调试和跟踪功能请参考下列文档：

- Cortex-M4技术参考手册；
- ARM调试接口v5结构规范。

调试系统帮助调试者实现低功耗模式调试和TIMER、I2C、WWDGT和FWDGT等一些外设调试。当相应的位被置1，调试系统会在低功耗模式下提供时钟，或者为一些外设保持当前状态，这些外设包括：TIMER、I2C、WWDGT和FWDGT。

## 11.2. JTAG/SW 功能说明

调试工具可以通过串行(SW)调试接口或者JTAG调试接口来访问调试功能。

### 11.2.1. 切换 JTAG/ SW 接口

默认使用JTAG调试接口，可以通过下列软件序列从JTAG调试切换到SW调试：

- 发送50个以上TCK周期的TMS=1信号；
- 发送16位TMS = 1110011110011110 (0xE79E LSB)信号；
- 发送50个以上TCK周期的TMS=1信号。

切换SW调试到JTAG调试的软件序列：

- 发送50个以上TCK周期的TMS=1信号；
- 发送16位TMS = 1110011100111100 (0xE73C LSB)信号；
- 发送50个以上TCK周期的TMS=1信号。

### 11.2.2. 引脚分配

JTAG调试提供五个引脚的接口：JTAG时钟引脚(JTCK)，JTAG 模式选择引脚(JTMS)，JTAG数据输入引脚(JTDI)，JTAG数据输出引脚(JTDO)，JTAG复位引脚(NJTRST,低电平有效)。串行调试(SWD)提供两个引脚的接口：数据输入输出引脚(SWDIO)和时钟引脚(SWCLK)。SW调试接口的两个引脚与JTAG调试接口的两个引脚复用，SWDIO和JTMS复用，SWCLK和JTCK复用。

当异步跟踪功能开启时，JTDO引脚也用作异步跟踪数据输出(TRACESWO)。

调试引脚分配：

PA15 ： JTDI

PA14 : JTCK/SWCLK
PA13 : JTMS/SWDIO
PB4 : NJTRST
PB3 : JTDO

默认复位后使用五个引脚的JTAG调试，用户可以在不使用NJTRST引脚情况下正常使用JTAG功能，此时PB4可以用作普通GPIO功能(NJTRST硬件拉高)。如果切换到SW调试模式，PA15/PB4/PB3释放作为普通GPIO功能。如果JTAG和SW调试功能都没有使用，这五个引脚都释放作为普通GPIO功能。五个引脚具体配置请参考*GPIO引脚配置*。

### 11.2.3. JTAG 链状结构

Cortex-M4内核的JTAG TAP和MCU JTAG TAP串行连接。MCU JTAG的IR(指令寄存器)是5位，而Cortex-M4内核的JTAG的IR(指令寄存器)是4位。所以当JTAG进行IR移位输入时，首先移位5位BYPASS指令给MCU JTAG，然后移位4位标准指令给Cortext-M4 JTAG。当进行数据移位时，数据链只需要额外添加一位，因为MCU JTAG已处在BYPASS模式。

MCU JTAG ID代码是0x790007A3。

### 11.2.4. 调试复位

JTAG-DP和SW-DP寄存器位于上电复位域。系统复位初始化了Cortex-M4的绝大部分组件，除了NVIC和调试逻辑(FPB，DWT，ITM)。NJTRST能复位JTAG TAP控制器。所以，可以在系统复位下实现调试功能。例如：复位后停止，用户在系统复位后配置相应停止位，系统复位释放后处理器会立即停止。

### 11.2.5. JEDEC-106 ID 代码

Cortex-M4集成了JEDEC-106 ID代码。位于ROM表中，映射地址为0xE00FF000_0xE00FFFFF。

## 11.3. 调试保持功能说明

### 11.3.1. 低功耗模式调试支持

当DBG控制寄存器(DBG_CTL)的STB_HOLD位置1并且进入待机模式，AHB总线时钟和系统时钟由CK_IRC8M提供，可以在待机模式下调试。当退出待机模式后，产生系统复位。

当DBG控制寄存器(DBG_CTL)的DSLP_HOLD位置1并且进入深度睡眠模式，AHB总线时钟和系统时钟由CK_IRC8M提供，可以在深度睡眠模式下调试。

当DBG控制寄存器(DBG_CTL)的SLP_HOLD位置1并且进入睡眠模式，AHB总线时钟没有关闭，可以在睡眠模式下调试。

### 11.3.2. TIMER, I2C, WWDGT 和 FWDGT 外设调试支持

当内核停止，并且DBG控制寄存器(DBG_CTL)中的相应位置1。对于不同外设，有不同动作发生：

对于TIMER外设，TIMER计数器停止并进行调试；

对于I2C外设，SMBUS保持状态并进行调试；

对于WWDGT或者FWDGT外设，计数器时钟停止并进行调试；

## 11.4. DBG 寄存器

DBG基地址：0xE004 2000

### 11.4.1. ID 寄存器 (DBG_ID)

地址： 0xE004 2000
只读寄存器

该寄存器只能按字(32位)访问

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | ID_CODE[31:0] | DBG ID 寄存器<br>这些位由软件读取，这些位是不变的常数 |

### 11.4.2. 控制寄存器 (DBG_CTL)

地址偏移：0x04
复位值： 0x0000 0000，仅上电复位

该寄存器只能按字(32 位)访问

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留. | TIMER10_HOLD | TIMER9_HOLD | TIMER8_HOLD | TIMER13_HOLD | TIMER12_HOLD | TIMER11_HOLD | 保留 | | | | TIMER7_HOLD | TIMER6_HOLD | TIMER5_HOLD | TIMER4_HOLD | I2C1_HOLD |
| | rw | rw | rw | rw | rw | rw | | | | | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| I2C0_HOLD | 保留 | TIMER3_HOLD | TIMER2_HOLD | TIMER1_HOLD | TIMER0_HOLD | WWDGT_HOLD | FWDGT_HOLD | 保留 | | TRACE_IOEN | 保留 | | STB_HOLD | DSLP_HOLD | SLP_HOLD |
| rw | | rw | rw | rw | rw | rw | rw | | | rw | | | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | 保留 | 必须保持复位值 |
| 30 | TIMER10_HOLD | TIMER10 保持寄存器<br>该位由软件置位和复位<br>0： 无影响<br>1： 当内核停止时保持定时器 10 计数器不变，用于调试 |

| | | |
|---|---|---|
| 29 | TIMER9_HOLD | TIMER9 保持寄存器<br>该位由软件置位和复位<br>0： 无影响<br>1： 当内核停止时保持定时器 9 计数器不变，用于调试 |
| 28 | TIMER8_HOLD | TIMER8 保持寄存器<br>该位由软件置位和复位<br>0： 无影响<br>1： 当内核停止时保持定时器 8 计数器不变，用于调试 |
| 27 | TIMER13_HOLD | TIMER13 保持寄存器<br>该位由软件置位和复位<br>0： 无影响<br>1： 当内核停止时保持定时器 13 计数器不变，用于调试 |
| 26 | TIMER12_HOLD | TIMER12 保持寄存器<br>该位由软件置位和复位<br>0： 无影响<br>1： 当内核停止时保持定时器 13 计数器不变，用于调试 |
| 25 | TIMER11_HOLD | TIMER11 保持寄存器<br>该位由软件置位和复位<br>0： 无影响<br>1： 当内核停止时保持定时器 13 计数器不变，用于调试 |
| 24:21 | 保留 | 必须保持复位值 |
| 20 | TIMER7_HOLD | TIMER7 保持寄存器<br>该位由软件置位和复位<br>0： 无影响<br>1： 当内核停止时保持定时器 7 计数器不变，用于调试 |
| 19 | TIMER6_HOLD | TIMER 6 保持寄存器<br>该位由软件置位和复位<br>0： 无影响<br>1： 当内核停止时保持定时器 6 计数器不变，用于调试 |
| 18 | TIMER5_HOLD | TIMER 5 保持寄存器<br>该位由软件置位和复位<br>0： 无影响<br>1： 当内核停止时保持定时器 5 计数器不变，用于调试 |
| 17 | TIMER4_HOLD | TIMER 4 保持寄存器<br>该位由软件置位和复位<br>0： 无影响<br>1： 当内核停止时保持定时器 4 计数器不变，用于调试 |
| 16 | I2C1_HOLD | I2C1 保持寄存器<br>该位由软件置位和复位 |

| | | |
|---|---|---|
| | | 0： 无影响<br>1： 当内核停止时保持 I2C1 的状态不变以避免 SMBUS 超时，用于调试 |
| 15 | I2C0_HOLD | I2C0 保持寄存器<br>该位由软件置位和复位<br>0： 无影响<br>1： 当内核停止时保持 I2C0 的状态不变以避免 SMBUS 超时，用于调试 |
| 14 | 保留 | 必须保持复位值 |
| 13 | TIMER3_HOLD | TIMER 3 保持寄存器<br>该位由软件置位和复位<br>0： 无影响<br>1： 当内核停止时保持定时器 3 计数器不变，用于调试 |
| 12 | TIMER2_HOLD | TIMER 2 保持寄存器<br>该位由软件置位和复位<br>0： 无影响<br>1： 当内核停止时保持定时器 2 计数器不变，用于调试 |
| 11 | TIMER1_HOLD | TIMER 1 保持寄存器<br>该位由软件置位和复位<br>0： 无影响<br>1： 当内核停止时保持定时器 1 计数器不变，用于调试 |
| 10 | TIMER0_HOLD | TIMER 0 保持寄存器<br>该位由软件置位和复位<br>0： 无影响<br>1： 当内核停止时保持定时器 0 计数器不变，用于调试 |
| 9 | WWDGT_HOLD | WWDG 保持寄存器<br>该位由软件置位和复位<br>0： 无影响<br>1： 当内核停止时保持WWDGT计数器时钟，用于调试 |
| 8 | FWDGT_HOLD | FWDGT 保持寄存器<br>该位由软件置位和复位<br>0： 无影响<br>1： 当内核停止时保持FWDGT计数器时钟，用于调试 |
| 7:6 | 保留 | 必须保持复位值 |
| 5 | TRACE_IOEN | 跟踪引脚分配使能<br>该位由软件置位和复位<br>0： 跟踪引脚分配禁用<br>1： 跟踪引脚分配使能 |
| 4:3 | 保留 | 必须保持复位值 |
| 2 | STB_HOLD | 待机模式保持寄存器 |

| | | |
|---|---|---|
| | | 该位由软件置位和复位<br>0： 无影响<br>1： 在待机模式下，系统时钟和 AHB 时钟由 CK_IRC8M 提供，当退出待机模式时，产生系统复位 |
| 1 | DSLP_HOLD | 深度睡眠模式保持寄存器<br>该位由软件置位和复位<br>0： 无影响<br>1： 在深度睡眠模式下，系统时钟和 AHB 时钟由 CK_IRC8M 提供 |
| 0 | SLP_HOLD | 睡眠模式保持寄存器<br>该位由软件置位和复位<br>0： 无影响<br>1： 在睡眠模式下, AHB 时钟继续运行 |

# 12. 模数转换器(ADC)

## 12.1. 简介

MCU片上集成了12位逐次逼近式模数转换器模块(ADC)，可以采样来自于16个外部通道和2个内部通道上的模拟信号。这18个ADC采样通道都支持多种运行模式，采样转换后，转换结果可以按照最低有效位对齐或最高有效位对齐的方式保存在相应的数据寄存器中。片上的硬件过采样机制可以通过减少来自MCU的相关计算负担来提高性能。

## 12.2. 主要特征

- 高性能：
  - ADC采样分辨率：12位、10位、8位、或者6位分辨率；
  - 前置校准功能；
  - 可编程采样时间；
  - 数据存储模式：最高有效位对齐和最低有效位对齐；
  - DMA请求。
- 模拟输入通道：
  - 16个外部模拟输入通道；
  - 1个内部温度传感器输入通道($V_{SENSE}$)；
  - 1个内部参考电压输入通道($V_{REFINT}$)。
- 转换开始的发起：
  - 软件；
  - 硬件触发。
- 运行模式：
  - 转换单个通道，或者扫描一组通道；
  - 单次运行模式，每次触发转换一次选择的输入通道；
  - 连续运行模式，连续转换所选择的输入通道；
  - 间断运行模式；
  - 同步模式(适用于具有两个或多个ADC的设备)。
- 转换结果阈值监测器功能：模拟看门狗。
- 中断的产生：
  - 常规序列转换结束；
  - 模拟看门狗事件。
- 过采样：
  - 16位的数据寄存器；
  - 可调整的过采样率，从2x到256x；
  - 高达8位的可编程数据移位。
- 通道输入范围：$V_{REF-} <= V_{IN} <= V_{REF+}$。

## 12.3. 引脚和内部信号

*图12-1. ADC模块框图* 给出了ADC模块框图。*表12-2. ADC输入引脚定义*给出了ADC引脚说明。

**表 12-1. ADC 内部输入信号**

| 内部信号名称 | 说明 |
|---|---|
| $V_{SENSE}$ | 内部温度传感器电压输出 |
| $V_{REFINT}$ | 内部参考电压输出 |

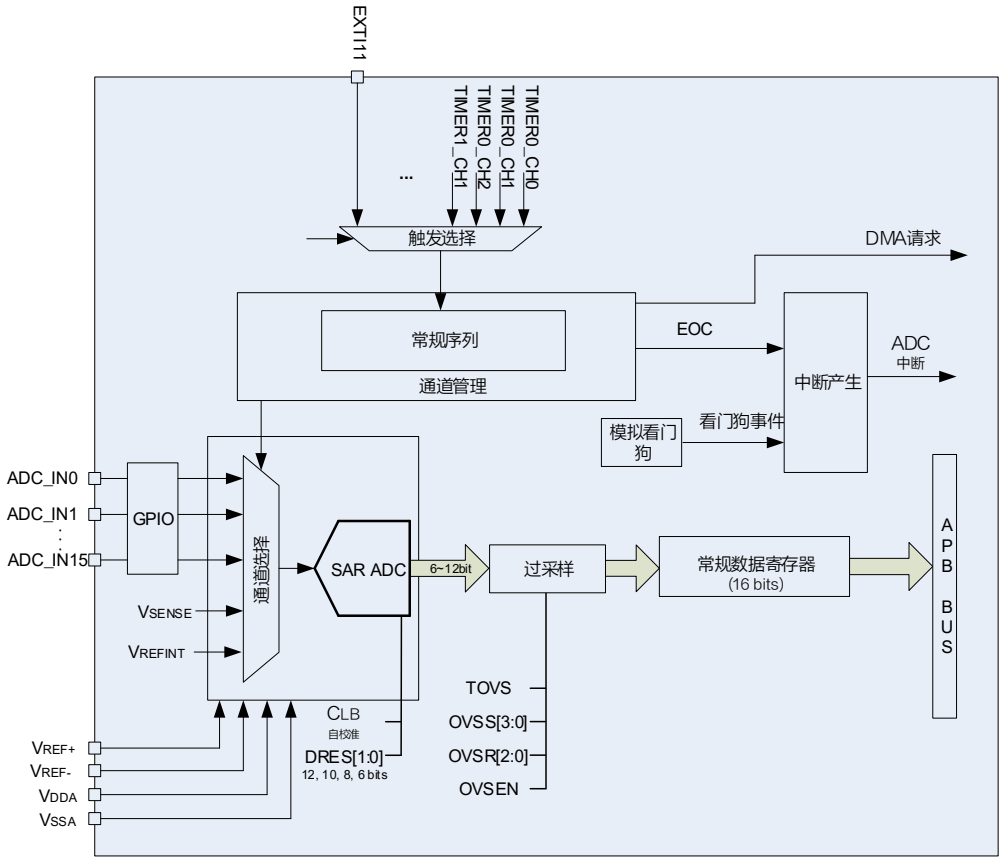
**表 12-2. ADC 输入引脚定义**

| 名称 | 注释 |
|---|---|
| $V_{DDA}$ | 模拟电源输入等于$V_{DD}$ |
| $V_{SSA}$ | 模拟地，等于$V_{SS}$ |
| $V_{REF+}$ | ADC正参考电压 |
| $V_{REF-}$ | ADC负参考电压，$V_{REF-} = V_{SSA}$ |
| ADCx_IN[15:0] | 多达16路外部通道 |

**注意：** $V_{DDA}$和$V_{SSA}$必须分别连接到$V_{DD}$和$V_{SS}$。

## 12.4. 功能说明

图 12-1. ADC 模块框图




### 12.4.1. 前置校准功能

在前置校准期间，ADC计算一个校准系数，这个系数是应用于ADC内部的，它直到ADC下次掉电才无效。在校准期间，应用不能使用ADC，它必须等到校准完成。在A/D转换前应执行校准操作。通过软件设置CLB=1来对校准进行初始化，在校准期间CLB位会一直保持1，直到校准完成，该位由硬件清0。

当ADC运行条件改变(例如，$V_{DDA}$、$V_{REF+}$ 以及温度等)，建议重新执行一次校准操作。

内部的模拟校准可以通过设置ADC_CTL1寄存器的RSTCLB位来重置。

软件校准过程：

1. 确保ADCON=1；
2. 延迟14个CK_ADC以等待ADC稳定；
3. 设置RSTCLB (可选的)；
4. 设置CLB=1；
5. 等待直到CLB=0。

### 12.4.2. ADC 时钟

CK_ADC时钟是由时钟控制器提供的，它和AHB、APB2时钟保持同步。ADC时钟可以在RCU时钟控制器中进行分配和配置。

### 12.4.3. ADCON 使能

ADC_CTL1寄存器中的ADCON位是ADC模块的使能开关。如果该位为0，则ADC模块保持复位状态。为了省电，当ADCON位为0时，ADC模拟子模块将会进入掉电模式。ADC使能后需等待$t_{su}$时间后才能采样，$t_{su}$数值详见芯片数据手册。

### 12.4.4. 常规序列

通道管理电路可以将采样通道组织成一个序列：常规序列。常规序列支持最多16个通道，每个通道称为常规通道。

ADC_RSQ0寄存器的RL[3:0]位规定了整个常规序列的长度。ADC_RSQ0~ADC_RSQ2寄存器规定了常规序列的通道选择。

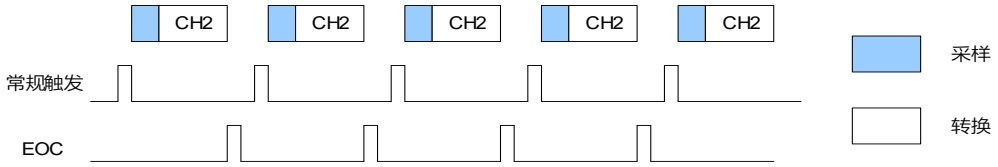
**注意：**尽管ADC支持18个通道，但常规序列一次最多转换16个通道。

### 12.4.5. 运行模式

#### 单次运行模式

单次运行模式下，ADC_RSQ2 寄存器的 RSQ0[4:0]位规定了 ADC 的转换通道。当 ADCON 位被置 1，一旦相应软件触发或者外部触发发生，ADC 就会采样和转换一个通道。

**图 12-2. 单次运行模式**




常规通道单次转换结束后，转换数据将被存放于 ADC_RDATA 寄存器中，EOC 将会置 1。如果 EOCIE 位被置 1，将产生一个中断。

常规序列单次运行模式的软件流程：

1. 确保ADC_CTL0寄存器的DISRC和SM位以及ADC_CTL1寄存器的CTN位为0；
2. 用模拟通道编号来配置RSQ0；
3. 配置ADC_SAMPTx寄存器；
4. 如果有需要，可以配置ADC_CTL1寄存器的ETERC和ETSRC位；
5. 设置SWRCST位，或者为常规序列产生一个外部触发信号；
6. 等到EOC置1；

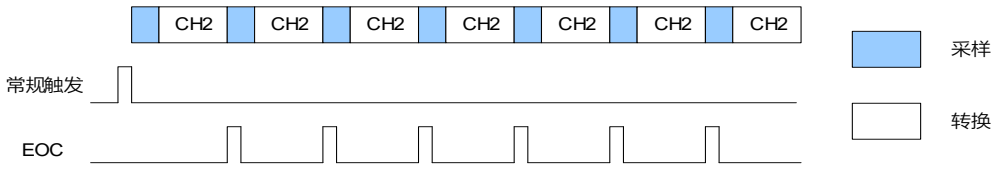
7. 延迟一个CK_ADC后，从ADC_RDATA寄存器中读ADC转换结果；
8. 写0清除EOC标志位。

**注意：**当EOC置1后，需延迟一个CK_ADC再读取ADC转换结果。

### 连续运行模式

对 ADC_CTL1 寄存器的 CTN 位置 1 可以使能连续运行模式。在此模式下，ADC 执行由 RSQ0[4:0]规定的转换通道。当 ADCON 位被置 1，一旦相应软件触发或者外部触发产生，ADC 就会采样和转换规定的通道。转换数据保存在 ADC_RDATA 寄存器中。

**图 12-3. 连续运行模式**




常规序列连续运行模式的软件流程：

1. 设置ADC_CTL1寄存器的CTN位为1；
2. 根据模拟通道编号配置RSQ0；
3. 配置ADC_SAMPTx寄存器；
4. 如果有需要，配置ADC_CTL1寄存器的ETERC和ETSRC位；
5. 设置SWRCST位，或者给常规序列产生一个外部触发信号；
6. 等待EOC标志位置1；
7. 延迟一个CK_ADC后，从ADC_RDATA寄存器中读ADC转换结果；
8. 写0清除EOC标志位；
9. 只要还需要进行连续转换，重复步骤6~8。

**注意：**当EOC置1后，需延迟一个CK_ADC再读取ADC转换结果。

由于要循环查询 EOC 标志位，DMA 可以被用来传输转换数据，软件流程如下：

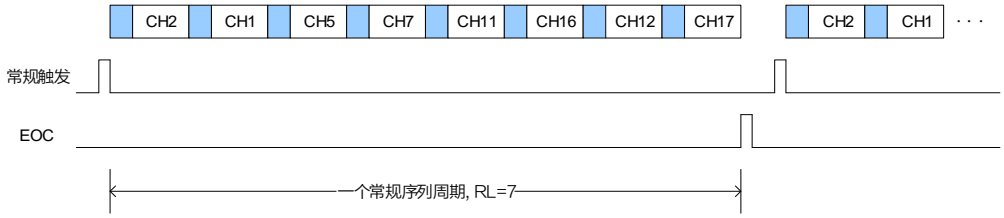
1. 设置ADC_CTL1寄存器的CTN位为1；
2. 根据模拟通道编号配置RSQ0；
3. 配置ADC_SAMPTx寄存器；
4. 如果有需要，配置ADC_CTL1寄存器的ETERC和ETSRC位；
5. 准备*直接存储器访问控制器(DMA)*模块，用于传输来自ADC_RDATA的数据；
6. 设置SWRCST位，或者给常规序列产生一个外部触发。

### 扫描运行模式

扫描运行模式可以通过将 ADC_CTL0 寄存器的 SM 位置 1 来使能。在此模式下，ADC 扫描转换所有被 ADC_RSQ0~ADC_RSQ2 寄存器选中的所有通道。一旦 ADCON 位被置 1，当相应软件触发或者外部触发产生，ADC 就会一个接一个的采样和转换常规序列通道。转换数据存储在 ADC_RDATA 寄存器中。常规序列转换结束后，EOC 位将被置 1。如果 EOCIE 位被置 1，将产生中断。当常规序列工作在扫描模式下时，ADC_CTL1 寄存器的 DMA 位必须设置为 1。

如果 ADC_CTL1 寄存器的 CTN 位也被置 1，则在常规序列转换完之后，这个转换自动重新开始。

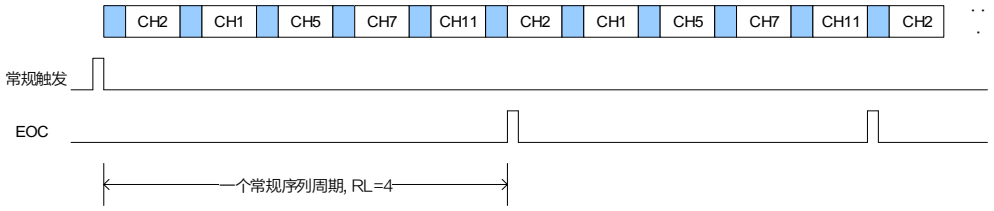
**图 12-4. 扫描运行模式，且连续转换模式失能**




常规序列扫描运行模式的软件流程：

1. 设置 ADC_CTL0 寄存器的 SM 位和 ADC_CTL1 寄存器的 DMA 位为 1；
2. 配置 ADC_RSQx 和 ADC_SAMPTx 寄存器；
3. 如果有需要，配置 ADC_CTL1 寄存器中的 ETERC 和 ETSRC 位；
4. 准备*直接存储器访问控制器(DMA)*模块，用于传输来自 ADC_RDATA 的数据；
5. 设置 SWRCST 位，或者给常规序列产生一个外部触发；
6. 等待 EOC 标志位置 1；
7. 写 0 清除 EOC 标志位。

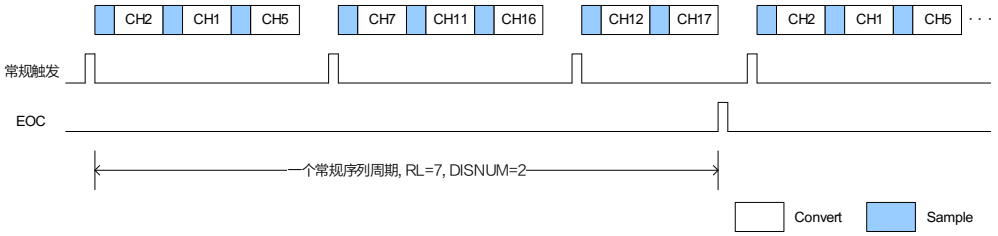
**图 12-5. 扫描运行模式，连续运行模式使能**




## 间断运行模式

当 ADC_CTL0 寄存器的 DISRC 位置 1 时，常规序列使能间断运行模式。该模式下可以执行一次 n 个通道的短序列转换(n 不超过 8)，该序列是 ADC_RSQ0~RSQ2 寄存器所选择的序列的一部分。数值 n 由 ADC_CTL0 寄存器的 DISCNUM[2:0]位配置。当相应的软件触发或外部触发发生，ADC 就会采样和转换在 ADC_RSQ0~RSQ2 寄存器所配置通道中接下来的 n 个通道，直到常规序列中所有的通道转换完成。每个常规序列转换周期结束后，EOC 位将被置 1。如果 EOCIE 位被置 1 将产生一个中断。

**图 12-6. 间断转换模式**




常规序列间断运行模式的软件流程：

1. 设置 ADC_CTL0 寄存器的 DISRC 位和 ADC_CTL1 寄存器的 DMA 位为 1；

2. 配置 ADC_CTL0 寄存器的 DISNUM[2:0]位；
3. 配置 ADC_RSQx 和 ADC_SAMPTx 寄存器；
4. 如果有需要，配置 ADC_CTL1 寄存器中的 ETERC 和 ETSRC 位；
5. 准备*直接存储器访问控制器(DMA)*模块，用于传输来自 ADC_RDATA 的数据；
6. 设置 SWRCST 位，或者给常规序列产生一个外部触发；
7. 如果需要，重复步骤 6；
8. 等待 EOC 标志位置 1；
9. 写 0 清除 EOC 标志位。

### 12.4.6. 转换结果阈值监测功能

ADC_CTL0 寄存器的 RWDEN 位置 1 将使能常规序列的模拟看门狗功能。该功能用于监测转换结果是否超过设定的阈值。如果 ADC 的模拟转换电压低于低阈值或高于高阈值时，ADC_STAT 状态寄存器的 WDE 位将被置 1。如果 WDEIE 位被置 1，将产生中断。ADC_WDHT 和 ADC_WDLT 寄存器用来设定高低阈值。内部数据的比较在对齐之前完成，因此阀值与 ADC_CTL1 寄存器的 DAL 位确定的对齐方式无关。ADC_CTL0 寄存器的 RWDEN，WDSC 和 WDCHSEL[4:0]位可以用来选择模拟看门狗监控单一通道或者多通道。

### 12.4.7. 数据存储模式

ADC_CTL1 寄存器的 DAL 位确定转换后数据存储的对齐方式。

**图 12-7. 12 位数据存储模式**

常规通道数据

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

DAL=0

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

DAL=1

6 位分辨率的数据存储模式不同于 12 位/10 位/8 位分辨率数据存储模式，如*图 12-8. 6 位数据存储模式*所示。

**图 12-8. 6 位数据存储模式**

常规通道数据

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

DAL=0

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | D5 | D4 | D3 | D2 | D1 | D0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

DAL=1

### 12.4.8. 采样时间配置

ADC 使用多个 CK_ADC 周期对输入电压采样，采样周期数目可以通过 ADC_SAMPT0 和 ADC_SAMPT1 寄存器的 SPTn[2:0]位配置。每个通道可以用不同的采样时间。在 12 位分辨率

的情况下，总转换时间=采样时间+12.5 个 CK_ADC 周期。

**例如：**

CK_ADC = 30MHz，采样时间为 1.5 个周期，那么总的转换时间为："1.5+12.5"个 CK_ADC 周期，即 0.467us。

### 12.4.9. 外部触发配置

外部触发输入的上升沿可以触发常规序列的转换。常规序列的外部触发源由 ADC_CTL1 寄存器的 ETSRC[2:0]位控制。

**表 12-3. ADC0 和 ADC1 的外部触发源**

| ETSRC[2:0] | 触发源 | 触发类型 |
|---|---|---|
| 000 | TIMER0_CH0 | 硬件触发 |
| 001 | TIMER0_CH1 | |
| 010 | TIMER0_CH2 | |
| 011 | TIMER1_CH1 | |
| 100 | TIMER2_TRGO | |
| 101 | TIMER3_CH3 | |
| 110 | EXTI11/ TIMER7_TRGO | |
| 111 | SWRCST | 软件触发 |

### 12.4.10. DMA 请求

DMA 请求，可以通过设置 ADC_CTL1 寄存器的 DMA 位来使能，它用于常规序列多个通道的转换结果。ADC 在常规序列一个通道转换结束后产生一个 DMA 请求，DMA 接受到请求后可以将转换的数据从 ADC_RDATA 寄存器传输到用户指定的目的地址。

### 12.4.11. ADC 内部通道

将 ADC_CTL1 寄存器的 TSVREN 位置 1 可以使能温度传感器通道(ADC0_IN16)和 $V_{REFINT}$ 通道(ADC0_IN17)。温度传感器可以用来测量器件周围的温度。传感器输出电压能被 ADC 转换成数字量。建议温度传感器的采样时间至少设置为 ts_temp μs(具体数值请参考 datasheet 文档)。温度传感器不用时，复位 TSVREN 位可以将其置于掉电模式。

温度传感器的输出电压随温度线性变化，由于生产过程的多样化，温度变化曲线的偏移在不同的芯片上会有不同(最多相差 45°C)。内部温度传感器更适合于检测温度的变化，而不是测量绝对温度。如果需要测量精确的温度，应该使用一个外置的温度传感器来校准这个偏移错误。

内部电压参考($V_{REFINT}$)提供了一个稳定的(带隙基准)电压输出给 ADC 和比较器。$V_{REFINT}$ 内部连接到 ADC0_IN17 输入通道。

使用温度传感器：

1. 配置温度传感器通道(ADC0_IN16)的转换序列和采样时间为 ts_temp μs
2. 置位 ADC_CTL1 寄存器中的 TSVREN 位，使能温度传感器

3. 置位 ADC_CTL1 寄存器的 ADCON 位，或者由外部触发启动 ADC 转换
4. 读取内部温度传感器输出电压 $V_{temperature}$，并由下面公式计算出实际温度：

$$温度\ (°C) = \{(V_{25} - V_{temperature}) / Avg_Slope\} + 25$$

$V_{25}$：内部温度传感器在 25°C 下的电压，典型值请参考相关型号 datasheet。

Avg_Slope：温度与内部温度传感器输出电压曲线的均值斜率，典型值请参考相关型号 datasheet。

### 12.4.12. 可编程分辨率(DRES)

ADC 分辨率可以通过寄存器 ADC_OVSAMPCTL 中的 DRES[1:0]位进行配置。对于那些不需要高精度数据的应用，可以使用较低的分辨率来实现更快速地转换。只有在 ADCON 比特为 0 时，才能修改 DRES[1:0]的值。较低的分辨率能够减少转换时间。如*表 12-4. 不同分辨率对应的 tCONV 时间*所示，较低的分辨率能够减少逐次逼近步骤所需的转换时间 $t_{ADC}$。

**表 12-4. 不同分辨率对应的 $t_{CONV}$ 时间**

| DRES[1:0] bits | $t_{CONV}$ (ADC clock cycles) | $t_{CONV}$(ns) at $f_{ADC}$=30MHz | $t_{SMPL}$(min) (ADC clock cycles) | $t_{ADC}$ (ADC clock cycles) | $t_{ADC}$(ns) at $f_{ADC}$=30MHz |
|---|---|---|---|---|---|
| 12 | 12.5 | 417 ns | 1.5 | 14 | 467 ns |
| 10 | 10.5 | 350 ns | 1.5 | 12 | 400 ns |
| 8 | 8.5 | 283 ns | 1.5 | 10 | 333 ns |
| 6 | 6.5 | 217 ns | 1.5 | 8 | 267 ns |

### 12.4.13. 片上硬件过采样

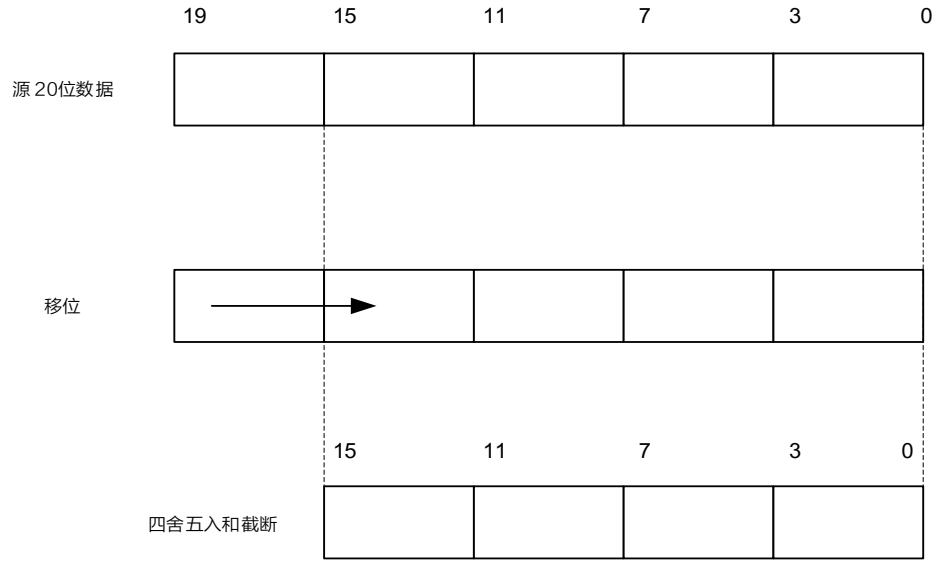
片上硬件过采样单元执行数据预处理以减轻 CPU 负担。它能够处理多个转换，并将多个转换的结果取平均，得出一个 16 位宽的数据。其结果根据如下公式计算得出，其中 N 和 M 的值可以被调整，过采样单元可以通过设置 ADC_OVSAMPCTL 寄存器的 OVSE 位来使能，它是以降低数据输出率为代价，换取较高的数据分辨率。$D_{out}(n)$是指 ADC 输出的第 n 个数字信号：

$$Result = \frac{1}{M} * \sum_{n=0}^{N-1} D_{out}(n) \qquad (12\text{-}1)$$

片上硬件过采样单元执行两个功能：求和和位右移。过采样率 N 是在 ADC_OVSAMPCTL 寄存器的 OVSR[2:0]位定义，它的取值范围为 2x 到 256x。除法系数 M 定义一个多达 8 位的右移，它通过 ADC_OVSAMPCTL 寄存器 OVSS[3:0]位进行配置。

求和单元能够生成一个多达 20 位(256*12 位)的值。首先，将这个值要进行右移，将移位后剩余的部分再通过取整转化一个近似值，最后将高位会被截断，仅保留最低 16 位有效位作为最终值传入对应的数据寄存器中。

**图 12-9. 20 位到 16 位的结果截断**




**注意：**如果移位后的中间结果还是超过 16 位，那么该结果的高位就会被直接截掉。

*图 12-10. 右移 5 位和取整的数例*描述一个从原始 20 位的累积数值处理成 16 位结果值的例子。

**图 12-10. 右移 5 位和取整的数例**

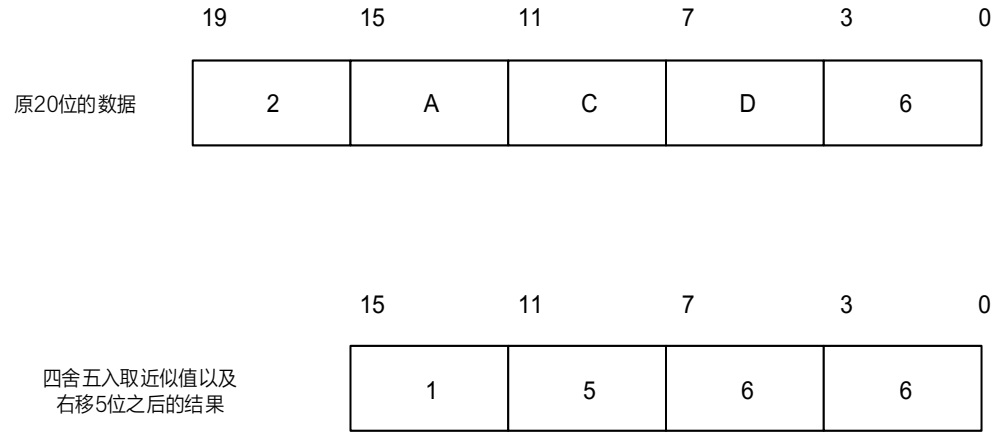



*表 12-5. 不同 N 和 M 组合的最大输出值(灰色值表示截断)*给出了 N 和 M 各种组合的数据格式，初始转换值为 0xFFF。

**表 12-5. 不同 N 和 M 组合的最大输出值(灰色值表示截断)**

| Oversampling ratio | Max Raw data | No-shift OVSS=0000 | 1-bit shift OVSS=0001 | 2-bit shift OVSS=0010 | 3-bit shift OVSS=0011 | 4-bit shift OVSS=0100 | 5-bit shift OVSS=0101 | 6-bit shift OVSS=0110 | 7-bit shift OVSS=0111 | 8-bit shift OVSS=1000 |
|---|---|---|---|---|---|---|---|---|---|---|
| 2x | 0x1FFE | 0x1FFE | 0x0FFF | 0x07FF | 0x03FF | 0x01FF | 0x00FF | 0x007F | 0x003F | 0x001F |
| 4x | 0x3FFC | 0x3FFC | 0x1FFE | 0x0FFF | 0x07FF | 0x03FF | 0x01FF | 0x00FF | 0x007F | 0x003F |
| 8x | 0x7FF8 | 0x7FF8 | 0x3FFC | 0x1FFE | 0x0FFF | 0x07FF | 0x03FF | 0x01FF | 0x00FF | 0x007F |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 16x | 0xFFF0 | 0xFFF0 | 0x7FF8 | 0x3FFC | 0x1FFE | 0x0FFF | 0x07FF | 0x03FF | 0x01FF | 0x00FF |
| 32x | 0x1FFE0 | 0xFFE0 | 0xFFF0 | 0x7FF8 | 0x3FFC | 0x1FFE | 0x0FFF | 0x07FF | 0x03FF | 0x01FF |
| 64x | 0x3FFC0 | 0xFFC0 | 0xFFE0 | 0xFFF0 | 0x7FF8 | 0x3FFC | 0x1FFE | 0x0FFF | 0x07FF | 0x03FF |
| 128x | 0x7FF80 | 0xFF80 | 0xFFC0 | 0xFFE0 | 0xFFF0 | 0x7FF8 | 0x3FFC | 0x1FFE | 0x0FFF | 0x07FF |
| 256x | 0xFFF00 | 0xFF00 | 0xFF80 | 0xFFC0 | 0xFFE0 | 0xFFF0 | 0x7FF8 | 0x3FFC | 0x1FFE | 0x0FFF |

和标准的转换模式相比，过采样模式的转换时间不会改变，在整个过采样序列的过程中采样时间仍然保持相等。每 N 个转换就会产生一个新的数据，一个等价的延迟为：

$$N \times t_{ADC} = N \times (t_{SMPL} + t_{CONV}) \quad (12\text{-}2)$$

## 12.5. ADC 同步模式

在有多个 ADC 模块的产品中，可以使用 ADC 同步模式。在 ADC 同步模式下，根据 ADC_CTL0 寄存器中 SYNCM[3:0]位所选的模式，转换的启动可以是 ADC0 和 ADC1 的交替触发或同步触发。

在同步模式下，当配置由外部事件触发的转换时，ADC1 必须通过软件来配置触发来，从而避免错误的触发引起不必要的转换。此外，对于 ADC0 和 ADC1 的外部触发必须被使能。

ADC同步模式如*表12-6. ADC同步模式表*所示。

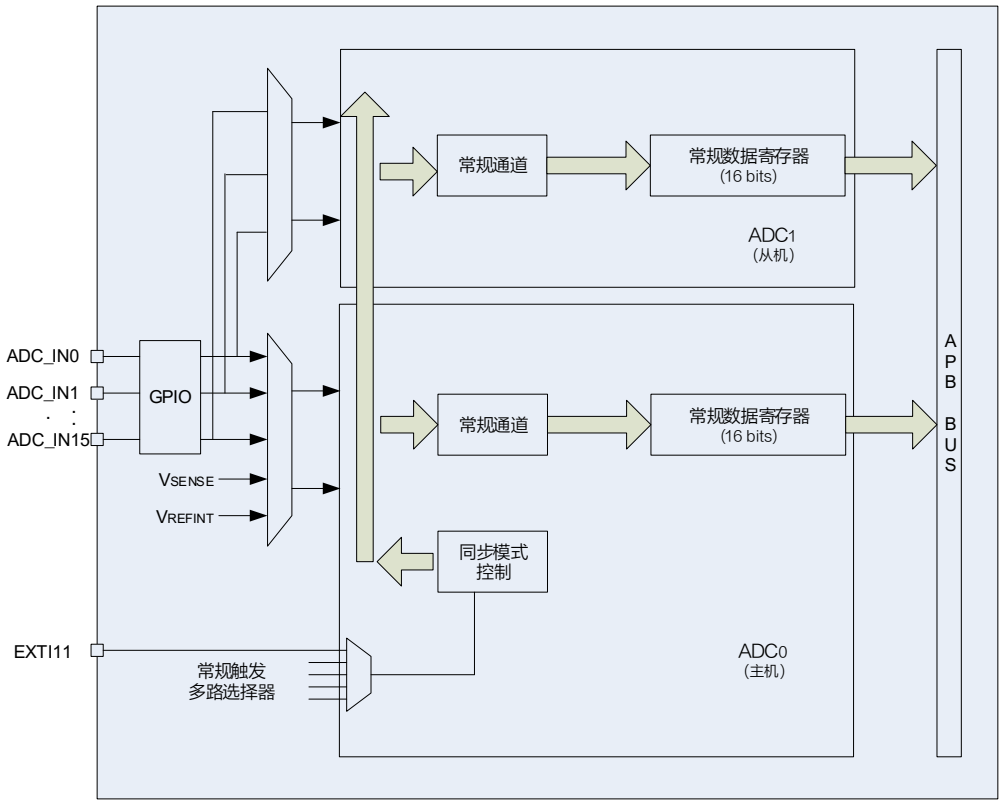
**表 12-6. ADC 同步模式表**

| SYNCM[3: 0] | 模式 |
|---|---|
| 0000 | 独立模式 |
| 0110 | 常规并行模式 |
| 0111 | 常规快速交叉模式 |
| 1000 | 常规慢速交叉模式 |

在ADC同步模式下，即使DMA不用，也要将DMA置位，ADC1的转换数据可以通过ADC0数据寄存器读取。

ADC 同步框图如*图 12-11. ADC 同步框图*所示。

图 12-11. ADC 同步框图




### 12.5.1. 独立模式

在这种模式下，每个 ADC 都独立工作，互不干扰。

### 12.5.2. 常规并行模式

此模式可并行转换常规序列，外部触发来源于 ADC0 常规序列触发(由 ADC_CTL1 寄存器的 ETSRC[2:0]决定)，ADC1 常规序列配置为软件触发模式。

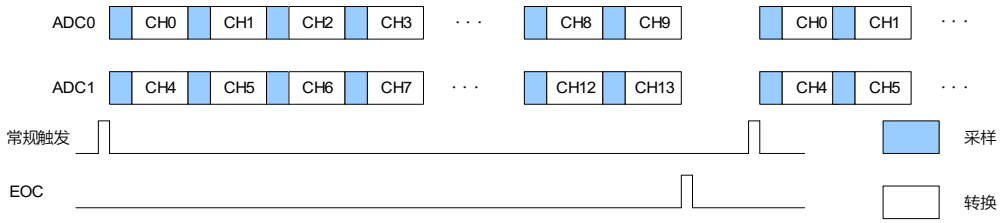
在 ADC0 或 ADC1 的转换事件结束时，即 ADC0 或 ADC1 的常规序列转换完毕，会产生一个 EOC 中断(如果某个 ADC 中断使能)。常规并行模式请参考*图 12-12. 基于 10 个通道的常规并行模式*。

32 位 ADC_RDATA 寄存器( [15: 0]位域用于保存 ADC0 常规通道采样数据，[31: 16]位域用于保存 ADC1 常规通道采样数据),32 位的 DMA 被用来将 ADC_RDATA 中的数据传送到 SRAM。

**注意**:

1. 若两个 ADC 模块使用了相同的采样通道，应保证不在同一时间使用该通道。
2. 两个 ADC 在同一时刻采样的两个通道，应该配置相同的采样时间。

**图 12-12. 基于 10 个通道的常规并行模式**




### 12.5.3. 常规快速交叉模式

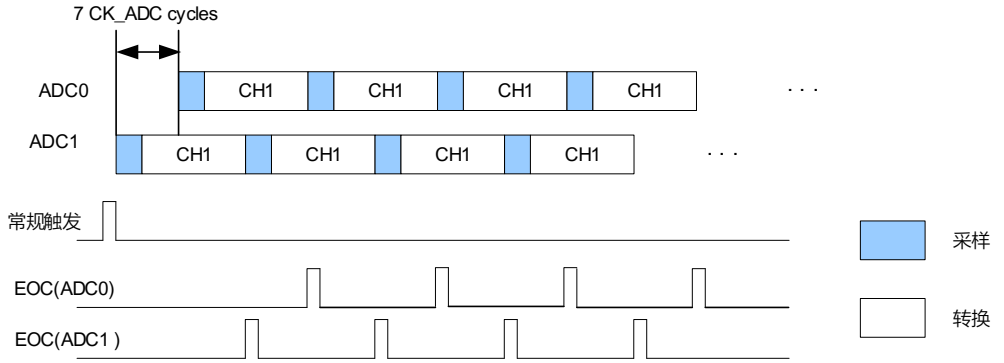
快速交叉模式适用于两个 ADC 的常规序列采样同一个通道，外部触发来源于 ADC0 常规序列(由 ADC_CTL1 寄存器的 ETSRC[2:0]决定)。当触发产生时，ADC1 立刻启动，而 ADC0 在 7 个 ADC 时钟周期后启动。

如果 ADC0 和 ADC1 的 CTN 位被置位，所选的常规序列在两个 ADC 中被不停的转换。如*图 12-13. 常规序列上的快速交叉模式*所示。

32 位 ADC_RDATA 寄存器([15: 0]位域用于保存 ADC0 常规通道采样数据，[31: 16]位域用于保存 ADC1 常规通道采样数据)。在 ADC0 产生 EOC 中断后(可通过置位 EOCIE 位)，可通过 32 位 DMA 将 ADC_RDATA 中数据传送到 SRAM。

**注意：** 两个 ADC 模块常规组通道的采样时间都应小于 7 个 ADC 时钟周期。

**图 12-13. 常规序列上的快速交叉模式(两个 ADC 的 CTN=1)**




### 12.5.4. 常规慢速交叉模式

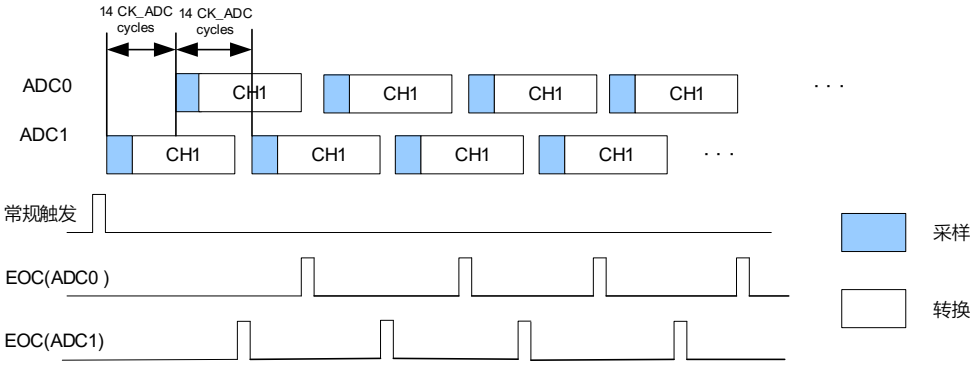
此模式应用于两个 ADC 的常规序列(通常一个常规通道)，外部触发来源于 ADC0 常规序列(由 ADC_CTL1 寄存器的 ETSRC[2:0]决定)。当触发产生时，ADC1 立刻启动，而 ADC0 在 14 个 ADC 时钟周期后启动，在 ADC0 启动后的 14 个时钟周期，ADC1 再次启动。

在这种模式下，不能使用连续转换模式，因为在这种模式下所选的常规通道组在两个 ADC 中被不停的转换，如*图 12-14. 常规序列上的慢速交叉模式*所示。

32 位 ADC_RDATA 寄存器([15: 0]位域用于保存 ADC0 常规通道采样数据，[31: 16]位域用于保存 ADC1 常规通道采样数据)。在 ADC0 产生 EOC 中断后(可通过置位 EOCIE 位)，可通过 32 位 DMA 将 ADC_RDATA 中数据传送到 SRAM。

**注意：**可允许的最大采样时间必须小于 14 个 CK_ADC 采样时钟，从而避免 ADC0 和 ADC1 在转换相同通道时出现采样时钟重叠。

**图 12-14. 常规序列上的慢速交叉模式**




## 12.6. 中断

以下任一个事件发生都可以产生中断：

- 常规序列转换结束；
- 模拟看门狗事件。

ADC0、ADC1都被映射到同一个中断向量IRQ18。

## 12.7. ADC 寄存器

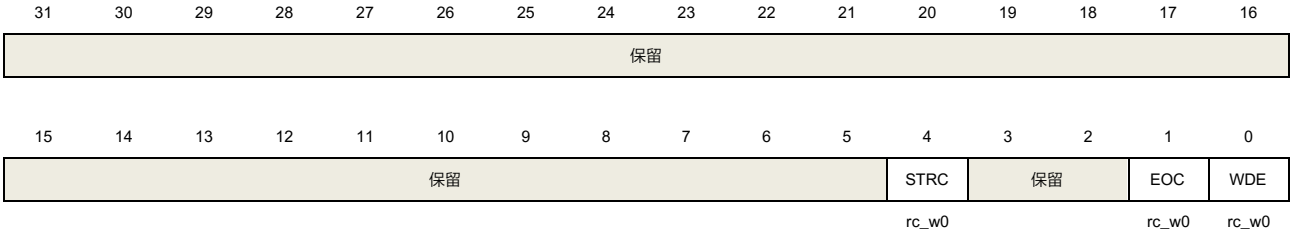
ADC0 基地址：0x4001 2400

ADC1 基地址：0x4001 2800

### 12.7.1. 状态寄存器 (ADC_STAT)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32位)访问。




| 位/位域 | 名称 | 说明 |
|---|---|---|
| 31:5 | 保留 | 必须保持复位值。 |
| 4 | STRC | 常规序列转换开始标志<br>0：转换没有开始<br>1：转换开始<br>常规序列转换开始时硬件置位，软件写0清除。 |
| 3:2 | 保留 | 必须保持复位值。 |
| 1 | EOC | 常规序列转换结束标志<br>0：转换没有结束<br>1：转换结束<br>常规序列转换结束时硬件置位，软件写 0 或读 ADC_RDATA 寄存器清除。 |
| 0 | WDE | 模拟看门狗事件标志<br>0： 没有模拟看门狗事件<br>1： 产生模拟看门狗事件<br>转换电压超过 ADC_WDLT 和 ADC_WDHT 寄存器设定的阈值时由硬件置 1，软件写 0 清除。 |

### 12.7.2. 控制寄存器 0 (ADC_CTL0)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | RWDEN | 保留 | | | SYNCM[3:0] | | | |
| | | | | | | | | rw | | | | rw | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DISNUM[2:0] | | | 保留 | DISRC | 保留 | WDSC | SM | 保留 | WDEIE | EOCIE | WDCHSEL[4:0] | | | | |
| rw | | | | rw | | rw | rw | | rw | rw | | rw | | | |

| 位/位域 | 名称 | 说明 |
|---|---|---|
| 31:24 | 保留 | 必须保持复位值。 |
| 23 | RWDEN | 常规序列看门狗使能<br>0：常规序列看门狗禁止<br>1：常规序列看门狗使能 |
| 22:20 | 保留 | 必须保持复位值 |
| 19：16 | SYNCM[3: 0] | 同步模式选择<br>这些位用于运行模式选择<br>0000：独立模式<br>0001~0101：保留<br>0110：常规并行模式<br>0111：常规快速交叉模式<br>1000：常规慢速交叉模式<br>1001~1111：保留<br>**注意：** 1)这些位只用于 ADC0；2)建议用户在任何配置之前关闭同步模式。 |
| 15:13 | DISNUM[2:0] | 间断模式下的转换数目<br>触发后即将被转换的通道数目将变成 DISNUM[2:0]+1 |
| 12 | 保留 | 必须保持复位值。 |
| 11 | DISRC | 常规序列间断模式<br>0：间断运行模式禁止<br>1：间断运行模式使能 |
| 10 | 保留 | 必须保持复位值。 |
| 9 | WDSC | 扫描模式下，模拟看门狗在通道配置<br>0：模拟看门狗在所有通道有效<br>1：模拟看门狗在单通道有效 |
| 8 | SM | 扫描模式<br>0：扫描运行模式禁止<br>1：扫描运行模式使能 |
| 7 | 保留 | 必须保持复位值。 |
| 6 | WDEIE | WDE 中断使能<br>0：中断禁止 |

| | | |
|---|---|---|
| | | 1：中断使能 |
| 5 | EOCIE | EOC 中断使能<br>0：中断禁止<br>1：中断使能 |
| 4:0 | WDCHSEL[4:0] | 模拟看门狗通道选择<br>00000: ADC 通道 0<br>00001: ADC 通道 1<br>00010: ADC 通道 2<br>00011: ADC 通道 3<br>00100: ADC 通道 4<br>00101: ADC 通道 5<br>00110: ADC 通道 6<br>00111: ADC 通道 7<br>01000: ADC 通道 8<br>01001: ADC 通道 9<br>01010: ADC 通道 10<br>01011: ADC 通道 11<br>01100: ADC 通道 12<br>01101: ADC 通道 13<br>01110: ADC 通道 14<br>01111: ADC 通道 15<br>10000: ADC 通道 16<br>10001: ADC 通道 17<br>其他值保留。<br>注意： ADC0 的模拟输入通道 16 和通道 17 分别连接到温度传感器和 $V_{REFINT}$。<br>ADC1 的模拟输入通道 16 和通道 17 内部都连接到 $V_{SSA}$。 |

### 12.7.3. 控制寄存器 1 (ADC_CTL1)

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | TSVREN | SWRCST | 保留 | ETERC | ETSRC[2：0] | | | 保留. |
| | | | | | | | | rw | rw | | rw | | rw | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | DAL | 保留. | | DMA | 保留 | | | | RSTCLB | CLB | CTN | ADCON |
| | | | | rw | | | rw | | | | | rw | rw | rw | rw |

| 位/位域 | 名称 | 说明 |
|---|---|---|
| 31：24 | 保留 | 必须保持复位值。 |
| 23 | TSVREN | ADC0 的通道 16 和 17 使能 |

| | | |
|---|---|---|
| | | 0：ADC0 的通道 16 和 17 禁止<br>1：ADC0 的通道 16 和 17 使能 |
| 22 | SWRCST | 软件触发常规序列转换开始<br>如果 ETSRC 是 111，该位置'1'开启常规序列转换。软件置位，软件清零，或转换开始后，由硬件清零。 |
| 21 | 保留 | 必须保持复位值。 |
| 20 | ETERC | 常规序列外部触发使能<br>0：常规序列外部触发禁止<br>1：常规序列外部触发使能 |
| 19：17 | ETSRC[2：0] | 常规序列外部触发选择<br>对于 ADC0 与 ADC1：<br>000：定时器 0 CH0<br>001：定时器 0 CH1<br>010：定时器 0 CH2<br>011：定时器 1 CH1<br>100：定时器 2 TRGO<br>101：定时器 3 CH3<br>110：中断线 11/定时器 7 TRGO<br>111：软件触发 |
| 16:12 | 保留 | 必须保持复位值。 |
| 11 | DAL | 数据对齐<br>0：最低有效位对齐<br>1：最高有效位对齐 |
| 10：9 | 保留 | 必须保持复位值。 |
| 8 | DMA | DMA 请求使能<br>0：DMA 请求禁止<br>1：DMA 请求使能 |
| 7：4 | 保留 | 必须保持复位值。 |
| 3 | RSTCLB | 校准复位<br>软件置位，在校准寄存器初始化后该位硬件清零。<br>0：校准寄存器初始化结束.<br>1：校准寄存器初始化开始 |
| 2 | CLB | ADC 校准<br>0：校准结束<br>1：校准开始 |
| 1 | CTN | 连续模式<br>0：禁止连续运行模式<br>1：使能连续运行模式 |

| | | |
|---|---|---|
| 0 | ADCON | 开启 ADC。该位从'0'变成'1'将在稳定时间结束后唤醒 ADC。当该位被置位以后，不改变寄存器的其他位仅仅对该位写'1'， 将开启转换。<br>0：禁止 ADC 关闭电源<br>1：使能 ADC |

### 12.7.4. 采样时间寄存器 0 (ADC_SAMPT0)

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能按字(32位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | SPT17[2:0] | | | SPT16[2:0] | | | SPT15[2:1] | |
| | | | | | | | | rw | | | rw | | | rw | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SPT15[0] | SPT14[2:0] | | | SPT13[2:0] | | | SPT12[2:0] | | | SPT11[2:0] | | | SPT10[2:0] | | |
| rw | rw | | | rw | | | rw | | | rw | | | rw | | |

| 位/位域 | 名称 | 说明 |
|---|---|---|
| 31:24 | 保留 | 必须保持复位值。 |
| 23:21 | SPT17[2:0] | 参考 SPT10[2:0]的描述 |
| 20:18 | SPT16[2:0] | 参考 SPT10[2:0]的描述 |
| 17:15 | SPT15[2:0] | 参考 SPT10[2:0]的描述 |
| 14:12 | SPT14[2:0] | 参考 SPT10[2:0]的描述 |
| 11:9 | SPT13[2:0] | 参考 SPT10[2:0]的描述 |
| 8:6 | SPT12[2:0] | 参考 SPT10[2:0]的描述 |
| 5:3 | SPT11[2:0] | 参考 SPT10[2:0]的描述 |
| 2:0 | SPT10[2:0] | 通道采样时间<br>000：通道采样时间为1.5周期<br>001：通道采样时间为7.5周期<br>010：通道采样时间为13.5周期<br>011：通道采样时间为28.5周期<br>100：通道采样时间为41.5周期<br>101：通道采样时间为55.5周期<br>110：通道采样时间为71.5周期<br>111：通道采样时间为 239.5 周期 |

### 12.7.5. 采样时间寄存器 1 (ADC_SAMPT1)

地址偏移：0x10

复位值：0x0000 0000

该寄存器只能按字(32位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | SPT9[2:0] | | | SPT8[2:0] | | | SPT7[2:0] | | | SPT6[2:0] | | | SPT5[2:1] | |
| | | rw | | | rw | | | rw | | | rw | | | rw | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SPT5[0] | | SPT4[2:0] | | | SPT3[2:0] | | | SPT2[2:0] | | | SPT1[2:0] | | | SPT0[2:0] | |
| rw | | rw | | | rw | | | rw | | | rw | | | rw | |

| 位/位域 | 名称 | 说明 |
|---|---|---|
| 31:30 | 保留 | 必须保持复位值。 |
| 29:27 | SPT9[2:0] | 参考 SPT0[2:0]的描述 |
| 26:24 | SPT8[2:0] | 参考 SPT0[2:0]的描述 |
| 23:21 | SPT7[2:0] | 参考 SPT0[2:0]的描述 |
| 20:18 | SPT6[2:0] | 参考 SPT0[2:0]的描述 |
| 17:15 | SPT5[2:0] | 参考 SPT0[2:0]的描述 |
| 14:12 | SPT4[2:0] | 参考 SPT0[2:0]的描述 |
| 11:9 | SPT3[2:0] | 参考 SPT0[2:0]的描述 |
| 8:6 | SPT2[2:0] | 参考 SPT0[2:0]的描述 |
| 5:3 | SPT2[2:0] | 参考 SPT0[2:0]的描述 |
| 2:0 | SPT0[2:0] | 通道采样时间<br>000：通道采样时间为1.5周期<br>001：通道采样时间为7.5周期<br>010：通道采样时间为13.5周期<br>011：通道采样时间为28.5周期<br>100：通道采样时间为41.5周期<br>101：通道采样时间为55.5周期<br>110：通道采样时间为71.5周期<br>111：通道采样时间为 239.5 周期 |

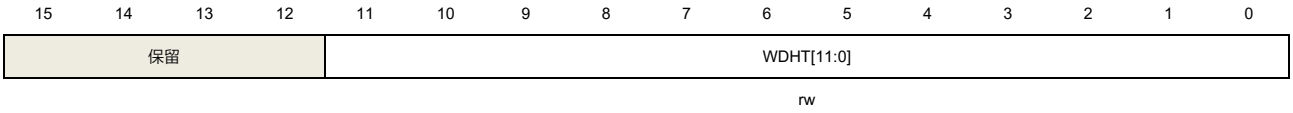
## 12.7.6. 看门狗高阈值寄存器 (ADC_WDHT)

地址偏移：0x24

复位值：0x0000 0FFF

该寄存器只能按字(32位)访问。

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

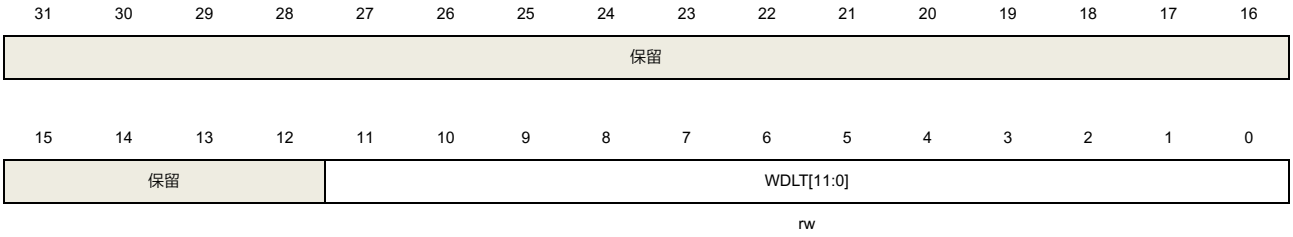



| 位/位域 | 名称 | 说明 |
|---|---|---|
| 31:12 | 保留 | 必须保持复位值。 |
| 11:0 | WDHT[11:0] | 模拟看门狗高侧阈值<br>这些位定义了模拟看门狗的高侧阈值。 |

### 12.7.7. 看门狗低阈值寄存器 (ADC_WDLT)

地址偏移：0x28
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

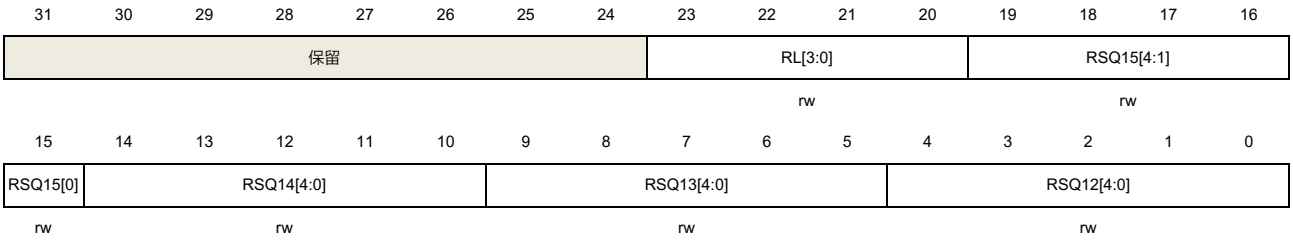



| 位/位域 | 名称 | 说明 |
|---|---|---|
| 31:12 | 保留 | 必须保持复位值。 |
| 11:0 | WDLT[11:0] | 模拟看门狗低侧阈值<br>这些位定义了模拟看门狗的低侧阈值。 |

### 12.7.8. 常规序列寄存器 0 (ADC_RSQ0)

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。




| 位/位域 | 名称 | 说明 |
|---|---|---|
| 31:24 | 保留 | 必须保持复位值。 |

| | | |
|---|---|---|
| 23:20 | RL[3:0] | 常规序列长度<br>常规通道转换序列中的总的通道数目为 RL[3:0]+1。 |
| 19:15 | RSQ15[4:0] | 参考 RSQ0[4:0]的描述 |
| 14:10 | RSQ14[4:0] | 参考 RSQ0[4:0]的描述 |
| 9:5 | RSQ13[4:0] | 参考 RSQ0[4:0]的描述 |
| 4:0 | RSQ12[4:0] | 参考 RSQ0[4:0]的描述 |

### 12.7.9. 常规序列寄存器 1 (ADC_RSQ1)

地址偏移：0x30
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | RSQ11[4:0] | | | | | RSQ10[4:0] | | | | | RSQ9[4:1] | | | |
| | | rw | | | | | rw | | | | | rw | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSQ9[0] | | RSQ8[4:0] | | | | | RSQ7[4:0] | | | | | RSQ6[4:0] | | | |
| rw | | rw | | | | | rw | | | | | rw | | | |

| 位/位域 | 名称 | 说明 |
|---|---|---|
| 31:30 | 保留 | 必须保持复位值。 |
| 29:25 | RSQ11[4:0] | 参考 RSQ0[4:0]的描述 |
| 24:20 | RSQ10[4:0] | 参考 RSQ0[4:0]的描述 |
| 19:15 | RSQ9[4:0] | 参考 RSQ0[4:0]的描述 |
| 14:10 | RSQ8[4:0] | 参考 RSQ0[4:0]的描述 |
| 9:5 | RSQ7[4:0] | 参考 RSQ0[4:0]的描述 |
| 4:0 | RSQ6[4:0] | 参考 RSQ0[4:0]的描述 |

### 12.7.10. 常规序列寄存器 2 (ADC_RSQ2)

地址偏移：0x34
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | RSQ5[4:0] | | | | | RSQ4[4:0] | | | | | RSQ3[4:1] | | | |
| | | rw | | | | | rw | | | | | rw | | | |

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

| RSQ3[0] | RSQ2[4:0] | RSQ1[4:0] | RSQ0[4:0] |
|---|---|---|---|
| rw | rw | rw | rw |

| 位/位域 | 名称 | 说明 |
|---|---|---|
| 31:30 | 保留 | 必须保持复位值。 |
| 29:25 | RSQ5[4:0] | 参考 RSQ0[4:0]的描述 |
| 24:20 | RSQ4[4:0] | 参考 RSQ0[4:0]的描述 |
| 19:15 | RSQ3[4:0] | 参考 RSQ0[4:0]的描述 |
| 14:10 | RSQ2[4:0] | 参考 RSQ0[4:0]的描述 |
| 9:5 | RSQ1[4:0] | 参考 RSQ0[4:0]的描述 |
| 4:0 | RSQ0[4:0] | 通道编号(0..17)写入这些位来选择常规通道的第 n 个转换的通道 |

### 12.7.11. 常规数据寄存器 (ADC_RDATA)

地址偏移：0x4C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADC1RDTR[15:0] | | | | | | | | | | | | | | | |
| r | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RDATA[15:0] | | | | | | | | | | | | | | | |
| r | | | | | | | | | | | | | | | |

| 位/位域 | 名称 | 说明 |
|---|---|---|
| 31:16 | ADC1RDTR[15:0] | ADC1 常规通道数据<br>在同步模式下，这些位包含着 ADC1 的常规通道数据<br>这些位只在 ADC0 中使用。 |
| 15:0 | RDATA[15:0] | 常规通道数据<br>这些位包含了常规通道的转换结果，只读。 |

### 12.7.12. 过采样控制寄存器 (ADC_OVSAMPCTL)

地址偏移：0x80
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | DRES[1:0] | | 保留 | | TOVS | OVSS[3:0] | | | | OVSR[2:0] | | | 保留 | OVSEN |
| | | Rw | | | | rw | rw | | | | rw | | | | rw |

| 位/位域 | 名称 | 说明 |
|---|---|---|
| 31:14 | 保留 | 必须保持复位值。 |
| 13:12 | DRES[1：0] | ADC 分辨率<br>00：12 位<br>01：10 位<br>10：8 位<br>11：6 位 |
| 11:10 | 保留 | 必须保持复位值。 |
| 9 | TOVS | 触发过滤采样<br>该位通过软件设置和清除.<br>0：所有的过滤采样连续转换完成一个触发后<br>1：对于过采样通道的每次转换都需要一次触发，触发次数由过采样率(OVSR[2:0])决定。<br>注意：当 ADCON= 0 时软件才允许写该位(确定没有转换正在进行). |
| 8:5 | OVSS[3:0] | 过滤采样移位<br>该位通过软件设置和清除.<br>0000：不移位<br>0001：移 1 位<br>0010：移 2 位<br>0011：移 3 位<br>0100：移 4 位<br>0101：移 5 位<br>0110：移 6 位<br>0111：移 7 位<br>1000：移 8 位<br>其他保留<br>注意:当 ADCON=0 时软件才允许写该位(确定没有转换正在进行). |
| 4:2 | OVSR[2:0] | 过采样率<br>这些位定义了过采样率的大小.<br>000：2x<br>001：4x<br>010：8x<br>011：16x<br>100：32x<br>101：64x<br>110：128x<br>111：256x |

| | | |
|---|---|---|
| | | 注意:当 ADCON=0 时软件才允许写该位(确定没有转换正在进行) |
| 1 | 保留 | 必须保持复位值。 |
| 0 | OVSEN | 过滤采样使能<br>该位通过软件和设置和清除<br>0：过滤采样失能<br>1：过滤采样使能<br>注意:当 ADCON=0 时软件才允许写该位(确定没有转换正在进行) |

# 13. 数模转换器(DAC)

## 13.1. 简介

数字/模拟转换器可以将 12 位的数字数据转换为外部引脚上的电压输出。数据可以采用 8 位或 12 位模式，左对齐或右对齐模式。当使能了外部触发，DMA 可被用于更新输入端数字数据。

在输出电压时，可以利用 DAC 输出缓冲区来获得更高的驱动能力。

每个 DAC 的两个通道可以独立或并发工作。

## 13.2. 主要特征

DAC 的主要特征如下：

- 8 位或 12 位分辨率；
- 数据左对齐或右对齐；
- DMA 功能；
- 同步更新转换；
- 外部事件触发转换；
- 可配置的内部缓冲区；
- 输入参考电压 $V_{REFP}$；
- 噪声波生成(LSFR 噪声模式和三角噪声模式)；
- DACx 双通道并发模式。

*图 13-1. DAC 结构框图*为 DAC 的结构框图，*表 13-1. DAC 引脚*给出了引脚描述。

**图 13-1. DAC 结构框图**

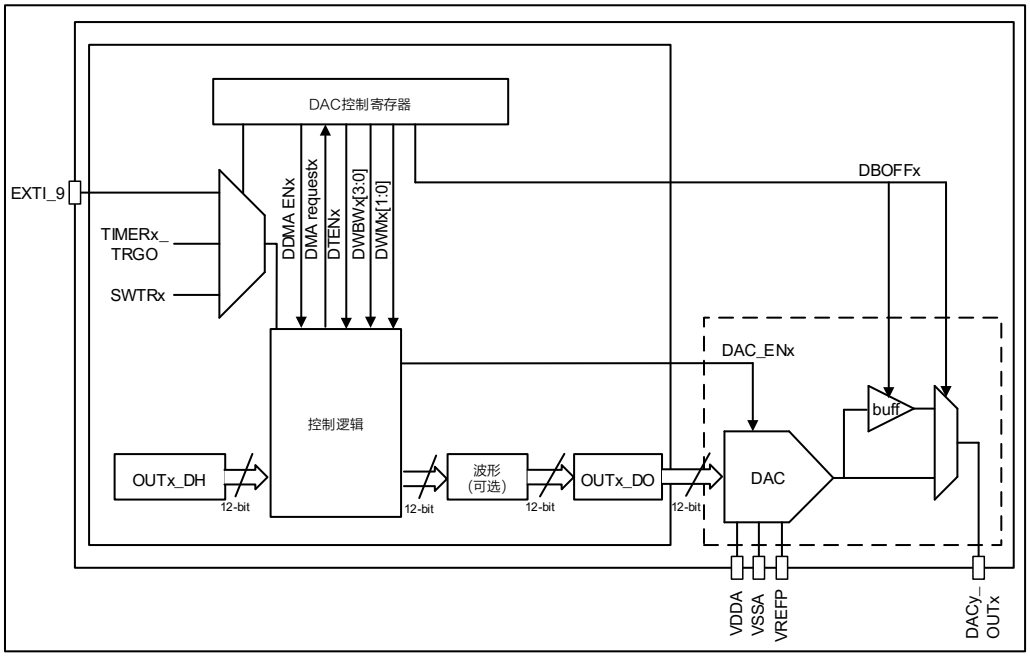


表 13-1. DAC 引脚

| 名称 | 描述 | 信号类型 |
|---|---|---|
| $V_{DDA}$ | 模拟电源 | 输入，模拟电源 |
| $V_{SSA}$ | 模拟电源地 | 输入，模拟电源地 |
| $V_{REFP}$ | DAC 正参考电压 | 输入，模拟正参考电压 |
| DACy_OUTx | DAC 模拟输出 | 模拟输出信号 |

下表详细列出了 DAC 的触发与输出信号。

表 13-2. DAC 触发与输出

| | DAC0 | |
|---|---|---|
| **通道** | 通道 0 | 通道 1 |
| **DAC 输出 I/O** | PA4 | PA5 |
| **DAC 输出 BUFFER 功能** | • | • |
| **软件触发功能** | • | |
| **EXTI 触发信号** | EXTI_9 | |
| **TIMER 触发信号** | TIMER1_TRGO<br>TIMER2_TRGO<br>TIMER3_TRGO<br>TIMER4_TRGO<br>TIMER5_TRGO<br>TIMER6_TRGO | |

**注意：**在使能 DAC 模块前，GPIO 口(DAC 输出 I/O)应配置为模拟模式。

## 13.3. 功能描述

### 13.3.1. DAC 使能

将 DAC_CTL0 寄存器中的 DENx 位置 1，可以给 DAC 模块上电，DAC 子模块完全启动需要等待 $t_{WAKEUP}$ 时间。

### 13.3.2. DAC 输出缓冲

为了降低输出阻抗，并在没有外部运算放大器的情况下驱动外部负载，每个 DAC 模块内部各集成了一个输出缓冲区。

缺省情况下，输出缓冲区是开启的，可以通过设置 DAC_CTL0 寄存器的 DBOFFx 位来开启或关闭缓冲区。

### 13.3.3. DAC 数据配置

对于 12 位的 DAC 保持数据(OUTx_DH)，可以通过对 DAC_OUTx_R12DH 、

DAC_OUTx_L12DH 和 DAC_OUTx_R8DH 中的任意一个寄存器写入数据来配置。当数据被加载到 DAC_OUTx_R8DH 寄存器时，只有 8 位最高有效位是可配置，4 位最低有效位被强制置为 4'b0000。

### 13.3.4. DAC 触发

DAC 可以通过软件或者外部信号的上升沿触发。外部触发可以通过设置 DAC_CTL0 寄存器中 DTENx 位来使能。触发源可以通过 DAC_CTL0 寄存器中 DTSELx 位来进行选择，如*表 13-3. DAC 外部触发*所示。

**表 13-3. DAC 外部触发**

| DTSELx[2:0] | 触发源 | 触发类型 |
|---|---|---|
| 3b'000 | TIMER5_TRGO | 硬件触发 |
| 3b'001 | TIMER2_TRGO | |
| 3b'010 | TIMER6_TRGO | |
| 3b'011 | TIMER4_TRGO | |
| 3b'100 | TIMER1_TRGO | |
| 3b'101 | TIMER3_TRGO | |
| 3b'110 | EXTI_9 | |
| 3b'111 | SWTR | 软件触发 |

TIMERx_TRGO 信号是由定时器生成的，而软件触发是通过设置 DAC_SWT 寄存器的 SWTRx 位生成的。

### 13.3.5. DAC 转换

如果使能了外部触发(通过设置 DAC_CTL0 寄存器的 DTENx 位)，当已经选择的触发事件发生，DAC 保持数据(OUTx_DH)会被转移到 DAC 数据输出寄存器(DAC_OUTx_DO)。而在外部触发未使能的情况下，DAC 保持数据(OUTx_DH)会被自动转移到 DAC 数据输出寄存器(DAC_OUTx_DO)。

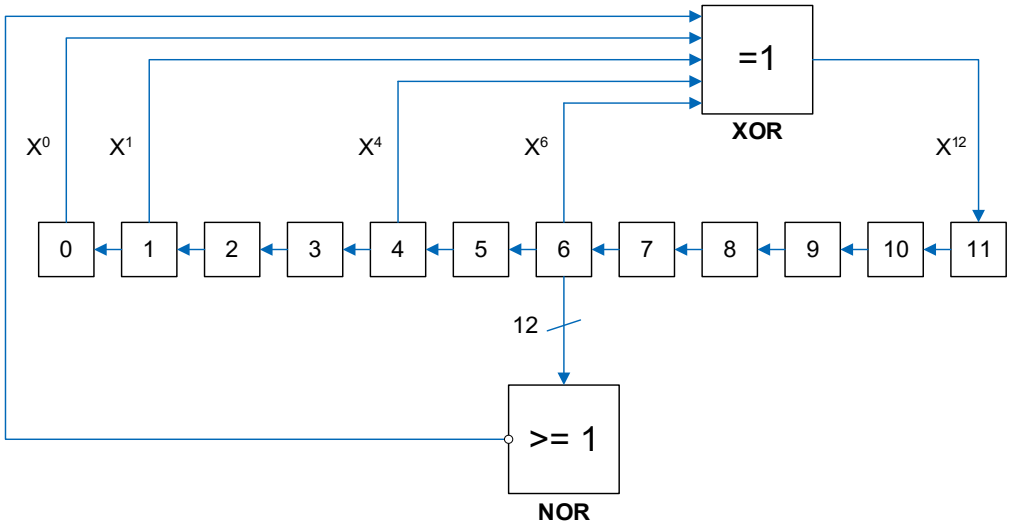
当 DAC 保持数据(OUTx_DH)加载到 DAC_OUTx_DO 寄存器时，经过 $t_{SETTLING}$ 时间之后，模拟输出变得有效，$t_{SETTLING}$ 的值与电源电压和模拟输出负载有关。

### 13.3.6. DAC 噪声波

有两种方式可以将噪声波加载到 DAC 输出数据：LFSR 噪声波和三角波。噪声波模式可以通过 DAC_CTL0 寄存器的 DWMx 位来进行选择。噪声的幅值可以通过配置 DAC_CTL0 寄存器的 DAC 噪声波位宽(DWBWx)位来进行设置。

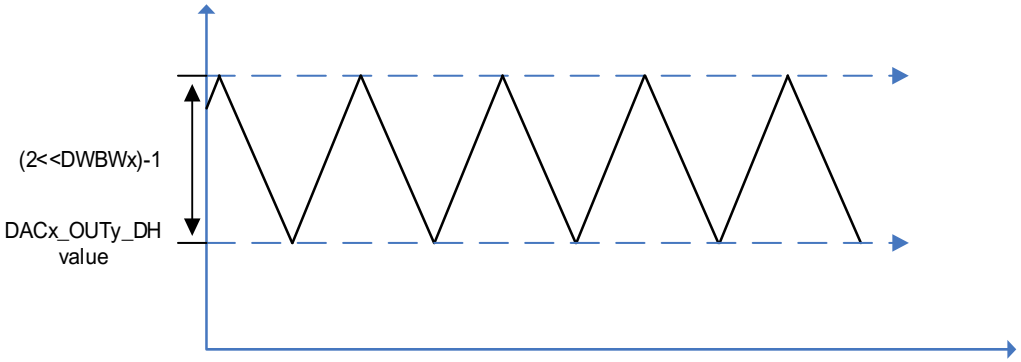
LFSR 噪声模式：在 DAC 控制逻辑中有一个线性反馈移位寄存器(LFSR)。在此模式下，LFSR 的值与 OUTx_DH 值相加后，被写入到 DAC 数据输出寄存器(DAC_OUTx_DO)。当配置的 DAC 噪声波位宽小于 12 时，LFSR 的值等于 LFSR 寄存器最低的 DWBWx 位，高位被屏蔽。

**图 13-2. DAC LFSR 算法**




三角噪声模式：三角波幅值与 OUTx_DH 值相加后，被写入到 DAC 数据输出寄存器(DAC_OUTx_DO)。三角波幅值的最小值为 0，最大值为(2 << DWBWx) - 1。

**图 13-3. DAC 三角噪声模式生成的波形**




## 13.3.7. DAC 输出电压

DAC 引脚上的模拟输出电压取决于下面的等式：

$$V_{DAC_OUT}=V_{REFP}*OUTx_DO/4096 \quad (13\text{-}1)$$

数字输入被线性地转换成模拟输出电压，输出范围为 0 到 $V_{REFP}$。

## 13.3.8. DMA 请求

在外部触发使能的情况下，通过设置 DAC_CTL0 寄存器的 DDMAENx 位来使能 DMA 请求。当有外部硬件触发的时候(不是软件触发)，则产生一个 DMA 请求。

### 13.3.9. DAC 并发转换

当 DAC 的两个通道同时工作时，为了在特定应用中最大限度利用总线带宽，DAC 的两个通道可以被配置为并发模式。在并发模式中，DAC 的 OUTx_DH 和 OUTx_DO 值将同时被更新。

有 3 个并发寄存器可用于加载 OUTx_DH 的值，分别是：DACC_R8DH、DACC_R12DH 和 DACC_L12DH 寄存器，配置其中的任意一个寄存器都可实现同时驱动 DAC 的两个通道。

当使能了外部触发时，DAC 两个通道的 DTENx 位都需要置 1，需要配置 DTSEL0/1 相同来保证同时触发。

当使能了 DMA 功能时，DAC 任一通道的 DDMAENx 位置 1 即可。

噪声模式和噪声位宽可以根据使用情况配置为相同或不同。

## 13.4. DAC 寄存器

DAC0 基地址：0x4000 7400

### 13.4.1. DACx 控制寄存器(DAC_CTL0)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | DDMAEN1 | DWBW1[3:0] | | | | DWM1[1:0] | | DTSEL1[2:0] | | | DTEN1 | DBOFF1 | DEN1 |
| | | | rw | rw | | | | rw | | rw | | | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | DDMAEN0 | DWBW0[3:0] | | | | DWM0[1:0] | | DTSEL0[2:0] | | | DTEN0 | DBOFF0 | DEN0 |
| | | | 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 | DDMAEN1 | DACx_OUT1 DMA 使能<br>0：DACx_OUT1 DMA 模式禁能<br>1：DACx_OUT1 DMA模式使能 |
| 27:24 | DWBW1[3:0] | DACx_OUT1 噪声波位宽<br>这些位指定了 DACx_OUT1 的噪声波信号的位宽。LFSR 噪声模式下，这些位表示不屏蔽 LFSR 的位[n-1,0]；三角噪声模式下，这些位表示三角波幅值为(2<<(n-1))-1。其中，n 为噪声波位宽。<br>0000：波形信号的位宽为 1<br>0001：波形信号的位宽为 2<br>0010：波形信号的位宽为 3<br>0011：波形信号的位宽为 4<br>0100：波形信号的位宽为 5<br>0101：波形信号的位宽为 6<br>0110：波形信号的位宽为 7<br>0111：波形信号的位宽为 8<br>1000：波形信号的位宽为 9<br>1001：波形信号的位宽为 10<br>1010：波形信号的位宽为 11<br>>=1011：波形信号的位宽为12 |
| 23:22 | DWM1[1:0] | DACx_OUT1 噪声波模式<br>这些位指定了在 DACx_OUT1 外部触发使能(DTEN1=1)的情况下，DACx_OUT1 的 |

| | | |
|---|---|---|
| | | 噪声波模式的选择。<br>00：波形生成禁能<br>01：LFSR 噪声模式<br>1x：三角噪声模式 |
| 21:19 | DTSEL1[2:0] | DACx_OUT1 触发选择<br>这些位仅在 DTEN1=1 并选择用于触发 DAC 的外部事件时使用。<br>000：TIMER5 TRGO<br>001：TIMER2 TRGO<br>010：TIMER6 TRGO<br>011：TIMER4 TRGO<br>100：TIMER1 TRGO<br>101：TIMER3 TRGO<br>110：外部中断线9<br>111：软件触发 |
| 18 | DTEN1 | DACx_OUT1 触发使能<br>0：DACx_OUT1 触发禁能<br>1：DACx_OUT1触发使能 |
| 17 | DBOFF1 | DACx_OUT1输出缓冲区关闭<br>0：DACx_OUT1输出缓冲区打开，以降低输出阻抗，提高驱动能力<br>1：DACx_OUT1输出缓冲区关闭 |
| 16 | DEN1 | DACx_OUT1 使能<br>0：DACx_OUT1 禁能<br>1：DACx_OUT1使能 |
| 15:13 | 保留 | 必须保持复位值。 |
| 12 | DDMAEN0 | DACx_OUT0 DMA 使能<br>0：DACx_OUT0 DMA 模式禁能<br>1：DACx_OUT0 DMA模式使能 |
| 11:8 | DWBW0[3:0] | DACx_OUT0 噪声波位宽<br>这些位指定了 DACx_OUT0 的噪声波信号的位宽。LFSR 噪声模式下，这些位表示不屏蔽 LFSR 的位[n-1,0]；三角噪声模式下，这些位表示三角波幅值为(2<<(n-1))-1。其中，n 为噪声波位宽。<br>0000：波形信号的位宽为 1<br>0001：波形信号的位宽为 2<br>0010：波形信号的位宽为 3<br>0011：波形信号的位宽为 4<br>0100：波形信号的位宽为 5<br>0101：波形信号的位宽为 6<br>0110：波形信号的位宽为 7<br>0111：波形信号的位宽为 8<br>1000：波形信号的位宽为 9<br>1001：波形信号的位宽为 10 |

| | | |
|---|---|---|
| | | 1010：波形信号的位宽为 11<br>>=1011：波形信号的位宽为12 |
| 7:6 | DWM0[1:0] | DACx_OUT0 噪声波模式<br>这些位指定了在 DACx_OUT0 外部触发使能(DTEN0=1)的情况下，DACx_OUT0 的噪声波模式的选择。<br>00：波形生成禁能<br>01：LFSR 噪声模式<br>1x：三角噪声模式 |
| 5:3 | DTSEL0[2:0] | DACx_OUT0 触发选择<br>这些位仅在 DTEN0=1 并选择用于触发 DAC 的外部事件时使用。<br>000：TIMER5 TRGO<br>001：TIMER2 TRGO<br>010：TIMER6 TRGO<br>011：TIMER4 TRGO<br>100：TIMER1 TRGO<br>101：TIMER3 TRGO<br>110：外部中断线9<br>111：软件触发 |
| 2 | DTEN0 | DACx_OUT0 触发使能<br>0：DACx_OUT0 触发禁能<br>1：DACx_OUT0触发使能 |
| 1 | DBOFF0 | DACx_OUT0输出缓冲区关闭<br>0：DACx_OUT0输出缓冲区打开，以降低输出阻抗，提高驱动能力<br>1：DACx_OUT0输出缓冲区关闭 |
| 0 | DEN0 | DACx_OUT0 使能<br>0：DACx_OUT0 禁能<br>1：DACx_OUT0使能 |

### 13.4.2. DACx 软件触发寄存器(DAC_SWT)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | | | | | | | SWTR1 | SWTR0 |
| | | | | | | | | | | | | | | w | w |

| 位/位域 | 名称 | 描述 |
|---|---|---|

| | | |
|---|---|---|
| 31:2 | 保留 | 必须保持复位值。 |
| 1 | SWTR1 | DACx_OUT1 软件触发，由硬件清除。<br>0：软件触发禁能<br>1：软件触发使能 |
| 0 | SWTR0 | DACx_OUT0 软件触发，由硬件清除。<br>0：软件触发禁能<br>1：软件触发使能 |

### 13.4.3. DACx_OUT0 12 位右对齐数据保持寄存器(DAC_OUT0_R12DH)

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | OUT0_DH[11:0] | | | | | | | | | | | |
| | | | | rw | | | | | | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:12 | 保留 | 必须保持复位值。 |
| 11:0 | OUT0_DH[11:0] | DACx_OUT0 12 位右对齐数据<br>这些位指定了将由 DACx_OUT0 转换的数据。 |

### 13.4.4. DACx_OUT0 12 位左对齐数据保持寄存器(DAC_OUT0_L12DH)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OUT0_DH[11:0] | | | | | | | | | | | | 保留 | | | |
| rw | | | | | | | | | | | | | | | |

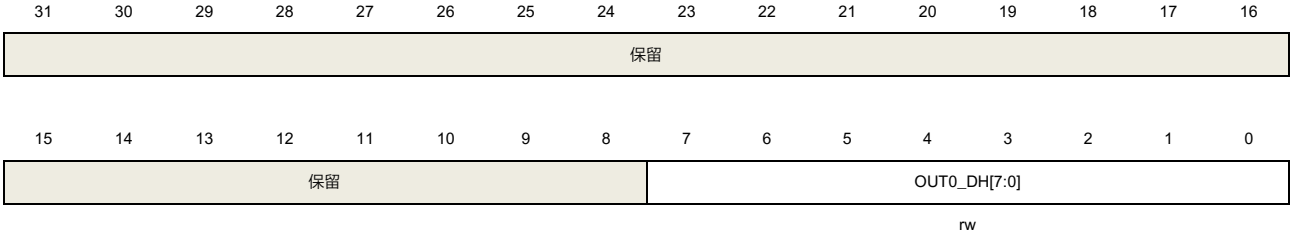
| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |

| | | |
|---|---|---|
| 15:4 | OUT0_DH[11:0] | DACx_OUT0 12 位左对齐数据<br>这些位指定了将由 DACx_OUT0 转换的数据。 |
| 3:0 | 保留 | 必须保持复位值。 |

### 13.4.5. DACx_OUT0 8 位右对齐数据保持寄存器(DAC_OUT0_R8DH)

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

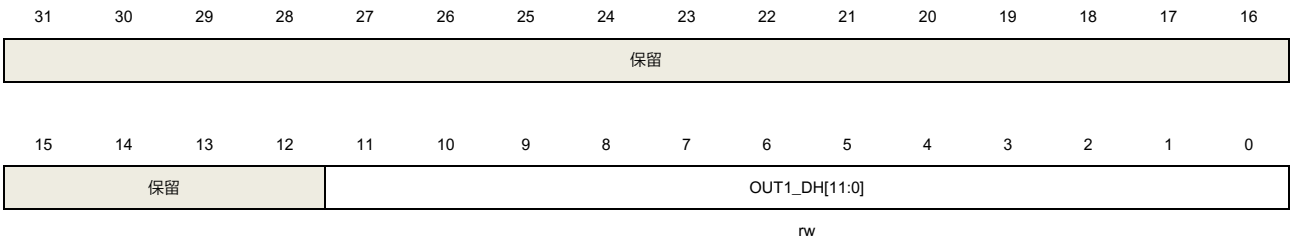



| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 必须保持复位值。 |
| 7:0 | OUT0_DH[7:0] | DACx_OUT0 8 位右对齐数据<br>这些位指定了将由 DACx_OUT0 转换的数据的最高 8 位有效位。 |

### 13.4.6. DACx_OUT1 12 位右对齐数据保持寄存器(DAC_OUT1_R12DH)

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

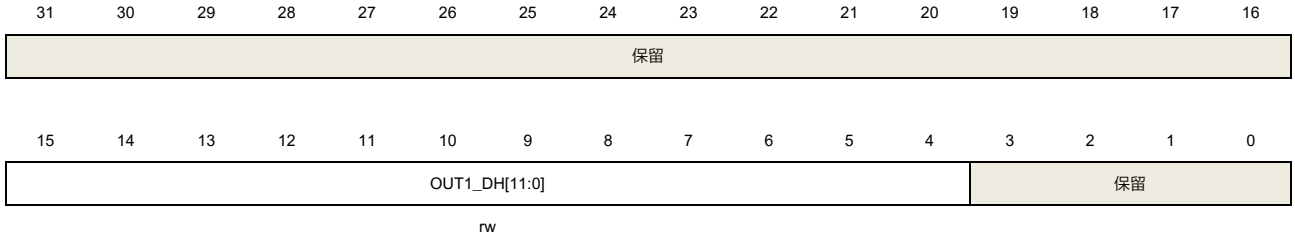



| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:12 | 保留 | 必须保持复位值。 |
| 11:0 | OUT1_DH[11:0] | DACx_OUT1 12 位右对齐数据<br>这些位指定了将由 DACx_OUT1 转换的数据。 |

### 13.4.7. DACx_OUT1 12 位左对齐数据保持寄存器(DAC_OUT1_L12DH)

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

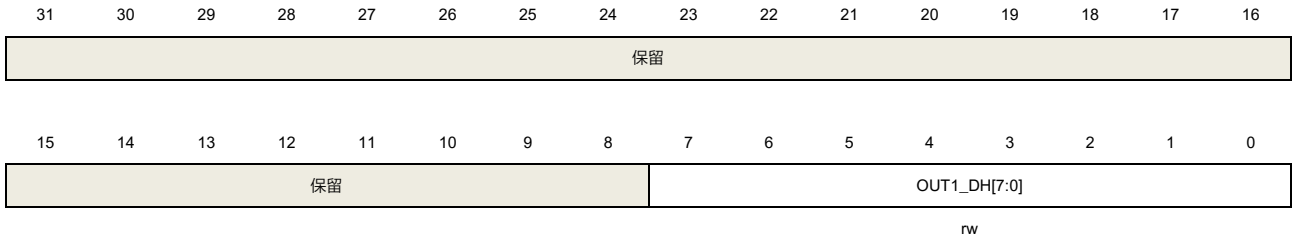



| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15:4 | OUT1_DH[11:0] | DACx_OUT1 12 位左对齐数据<br>这些位指定了将由 DACx_OUT1 转换的数据。 |
| 3:0 | 保留 | 必须保持复位值。 |

### 13.4.8. DACx_OUT1 8 位右对齐数据保持寄存器(DAC_OUT1_R8DH)

地址偏移：0x1C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 必须保持复位值。 |
| 7:0 | OUT1_DH[7:0] | DACx_OUT1 8 位右对齐数据<br>这些位指定了将由 DACx_OUT1 转换的数据的 8 位最高有效位。 |

### 13.4.9. DACx 并发模式 12 位右对齐数据保持寄存器(DACC_R12DH)

地址偏移：0x20
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | OUT1_DH[11:0] | | | | | | | | | | | |
| | | | | rw | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | OUT0_DH[11:0] | | | | | | | | | | | |
| | | | | rw | | | | | | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:28 | 保留 | 必须保持复位值。 |
| 27:16 | OUT1_DH[11:0] | DACx_OUT1 12 位右对齐数据<br>这些位指定了将由 DACx_OUT1 转换的数据。 |
| 15:12 | 保留 | 必须保持复位值。 |
| 11:0 | OUT0_DH[11:0] | DACx_OUT0 12 位右对齐数据<br>这些位指定了将由 DACx_OUT0 转换的数据。 |

## 13.4.10. DACx 并发模式 12 位左对齐数据保持寄存器(DACC_L12DH)

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OUT1_DH[11:0] | | | | | | | | | | | | 保留 | | | |
| rw | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OUT0_DH[11:0] | | | | | | | | | | | | 保留 | | | |
| rw | | | | | | | | | | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:20 | OUT1_DH[11:0] | DACx_OUT1 12 位左对齐数据<br>这些位指定了将由 DACx_OUT1 转换的数据。 |
| 19:16 | 保留 | 必须保持复位值。 |
| 15:4 | OUT0_DH[11:0] | DACx_OUT0 12 位左对齐数据<br>这些位指定了将由 DACx_OUT0 转换的数据。 |
| 3:0 | 保留 | 必须保持复位值。 |

### 13.4.11. DACx 并发模式 8 位右对齐数据保持寄存器(DACC_R8DH)

地址偏移：0x28
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

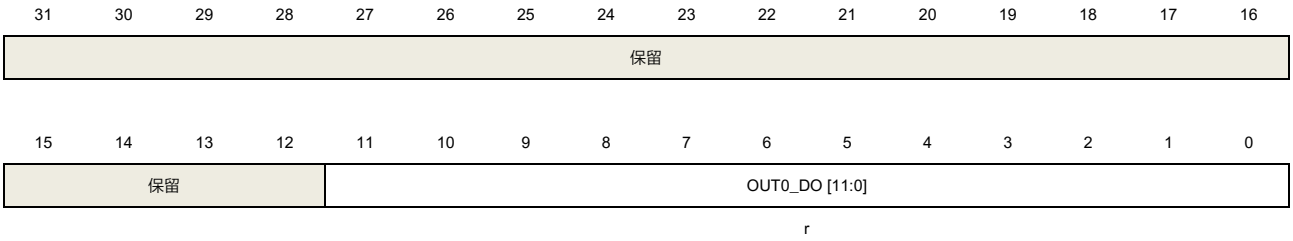



| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15:8 | OUT1_DH[7:0] | DACx_OUT1 8 位右对齐数据<br>这些位指定了将由 DACx_OUT1 转换的数据的 8 位最高有效位。 |
| 7:0 | OUT0_DH[7:0] | DACx_OUT0 8 位右对齐数据<br>这些位指定了将由 DACx_OUT0 转换的数据的 8 位最高有效位。 |

### 13.4.12. DACx_OUT0 数据输出寄存器(DAC_OUT0_DO)

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:12 | 保留 | 必须保持复位值。 |
| 11:0 | OUT0_DO [11:0] | DACx_OUT0 数据输出。<br>这些位为只读类型，存储由 DACx_OUT0 转换的数据。 |

### 13.4.13. DACx_OUT1 数据输出寄存器(DAC_OUT1_DO)

地址偏移：0x30

复位值：0x0000 0000

该寄存器只能按字(32 位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:12 | 保留 | 必须保持复位值。 |
| 11:0 | OUT1_DO [11:0] | DACx_OUT1 数据输出。<br>这些位为只读类型，存储由 DACx_OUT1 转换的数据。 |

# 14. 看门狗定时器(WDGT)

看门狗定时器(WDGT)是一个硬件计时电路，用来监测由软件故障导致的系统故障。片上有两个看门狗定时器外设，独立看门狗定时器(FWDGT)和窗口看门狗定时器(WWDGT)。它们使用灵活，并提供了很高的安全水平和精准的时间控制。两个看门狗定时器都是用来解决软件故障问题的。

看门狗定时器在内部计数值达到预设门限的时候，会触发一个复位。当处理器工作在调试模式的时候看门狗定时器定时计数器可以停止计数。

## 14.1. 独立看门狗定时器(FWDGT)

### 14.1.1. 简介

独立看门狗定时器(FWDGT)有独立的时钟源(IRC40K)。即使主时钟失效，FWDGT依然能保持正常工作状态，适用于需要独立环境且对计时精度要求不高的场合。

当内部向下计数器的计数值达到0，独立看门狗会产生一个系统复位。使能独立看门狗的寄存器写保护功能可以避免寄存器的值被意外的配置篡改。

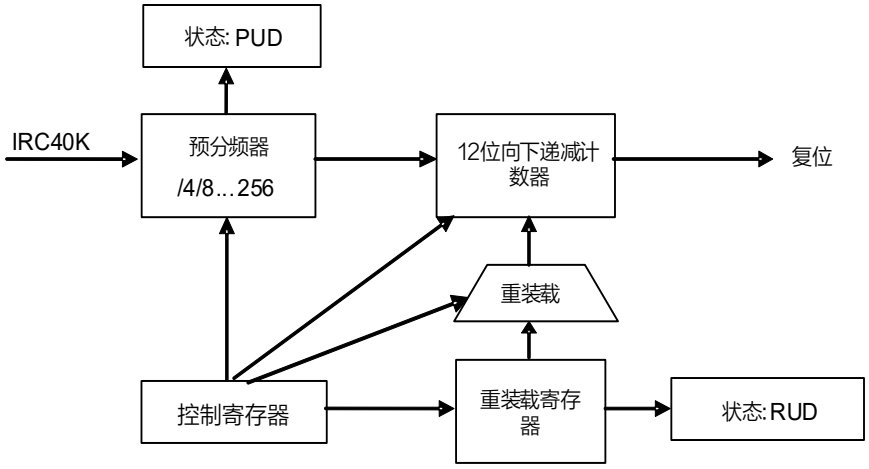
### 14.1.2. 主要特征

- 自由运行的12位向下计数器；
- 如果看门狗定时器被使能，那么当向下计数器的值达到0时产生系统复位；
- 独立时钟源，独立看门狗定时器在主时钟故障(例如待机和深度睡眠模式下)时仍能工作；
- 独立看门狗定时器硬件控制位，用来控制是否在上电时自动启动独立看门狗定时器；
- 可以配置独立看门狗定时器在调试模式下选择停止还是继续工作。

### 14.1.3. 功能说明

独立看门狗定时器带有一个8级预分频器和一个12位的向下递减计数器。*图14-1. 独立看门狗定时器框图*的独立看门狗定时器的功能模块。

**图 14-1. 独立看门狗定时器框图**




向控制寄存器(FWDGT_CTL)中写0xCCCC可以开启独立看门狗定时器，计数器开始向下计数。当计数器记到0x000，产生一次系统复位。

在任何时候向控制寄存器(FWDGT_CTL)中写0xAAAA都可以重装载计数器，重装载值来源于重装载寄存器(FWDGT_RLD)。软件可以在计数器计数值达到0x000之前可以通过重装载计数器来阻止看门狗定时器产生系统复位。

如果在选项字节中打开了"硬件看门狗定时器"功能，那么在上电的时候看门狗定时器就被自动打开。为了避免系统复位，软件应该在计数器达到0x000之前重装载计数器。

预分频寄存器(FWDGT_PSC)和FWDGT_RLD寄存器都有写保护功能。在写数据到这些寄存器之前，需要写0x5555到FWDGT_CTL中。写其他任何值到FWDGT_CTL中将会再次启动对这些寄存器的写保护。当FWDGT_PSC或者FWDGT_RLD更新时，FWDGT_STAT寄存器相应的状态位被置1。

如果DBG控制寄存器0(DBG_CTL0)中的FWDGT_HOLD位被清0，即使Cortex®-M4内核停止(调试模式下)独立看门狗定时器依然工作。如果FWDGT_HOLD位置1，独立看门狗定时器将在调试模式下停止工作。

**表 14-1. 独立看门狗定时器在 40kHz(IRC40K)时的最小/最大超时周期**

| 预分频系数 | PSC[2:0] 位 | 最小超时(ms) RLD[11:0]=0x000 | 最大超时(ms) RLD[11:0]=0xFFF |
|---|---|---|---|
| 1 / 4 | 000 | 0.025 | 409.525 |
| 1 / 8 | 001 | 0.025 | 819.025 |
| 1 / 16 | 010 | 0.025 | 1638.025 |
| 1 / 32 | 011 | 0.025 | 3276.025 |
| 1 / 64 | 100 | 0.025 | 6552.025 |
| 1 / 128 | 101 | 0.025 | 13104.025 |
| 1 / 256 | 110 or 111 | 0.025 | 26208.025 |

通过校准IRC40K可以使独立看门狗定时器超时更加精确。

**注意**：1.当执行完喂狗reload操作之后，如需要立即进入deepsleep / standby模式时，必须通

过软件设置，在reload命令及deepsleep / standby模式命令中间插入(3个以上)IRC40K时钟间隔。2.两次连续喂狗之间需插入7个及以上IRC40K时钟间隔。

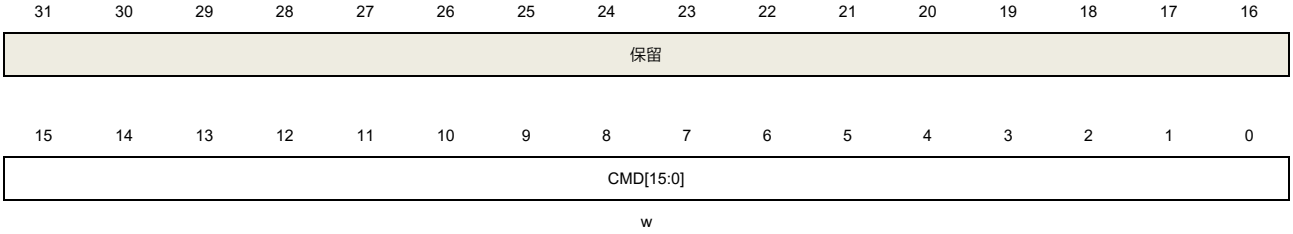
## 14.1.4. FWDGT 寄存器

FWDGT基地址：0x4000 3000

### 控制寄存器 (FWDGT_CTL)

地址偏移：0x00
复位值：0x0000 0000

该寄存器可以按半字(16 位)或字(32 位)访问。

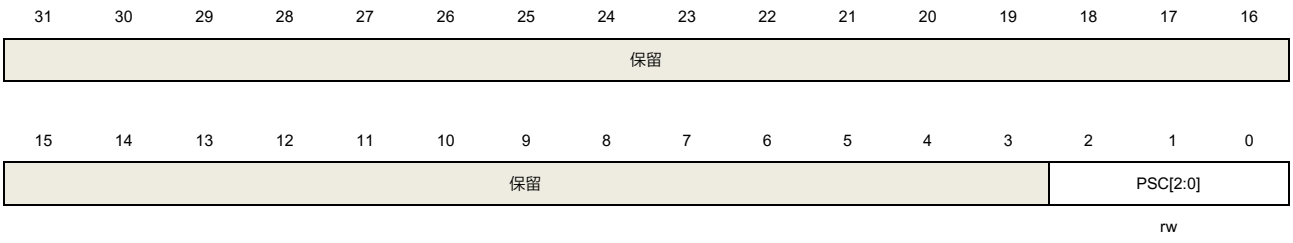



| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15:0 | CMD[15:0] | 只可写，写入不同的值来产生不同的功能<br>0x5555：关闭FWDGT_PSC和FWDGT_RLD的写保护<br>0xCCCC：开启独立看门狗定时器定时计数器。计数减到0时产生复位<br>0xAAAA：重装载计数器 |

### 预分频寄存器 (FWDGT_PSC)

地址偏移：0x04
复位值：0x0000 0000

该寄存器可以按半字(16 位)或字(32 位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:3 | 保留 | 必须保持复位值。 |
| 2:0 | PSC[2:0] | 独立看门狗定时器计时预分频选择。写这些位之前要通过向FWDGT_CTL寄存器写0x5555去除写保护。在改写这个寄存器的过程中，FWDGT_STAT寄存器的PUD位被置1，此时读取此寄存器的值都是无效的。<br>000：1 / 4 |

001：1 / 8
010：1 / 16
011：1 / 32
100：1 / 64
101：1 / 128
110：1 / 256
111：1 / 256

如果应用需要使用几个预分频系数，改变预分频值之前必须等到PUD位被清0。更新预分频寄存器中的值后，在代码持续执行之前不必等待PUD值被清零(在进入省电模式前需等待PUD值清零)。

### 重装载寄存器 (FWDGT_RLD)

地址偏移：0x08
复位值：0x0000 0FFF

该寄存器可以按半字(16 位)或字(32 位)访问。

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:12 | 保留 | 必须保持复位值。 |
| 11:0 | RLD[11:0] | 独立看门狗定时器定时计数器重装载值，向FWDGT_CTL寄存器写入0xAAAA的时候，这个值会被更新到看门狗定时器计数器中。<br>这些位有写保护功能。在写这些位之前需向FWDGT_CTL寄存器中写0x5555。在改写这个寄存器的过程中，FWDGT_STAT寄存器的RUD位被置1，从此寄存器中读取的任何值都是无效的。<br>如果应用需要使用几个重装载值，改变重装载值之前必须等到RUD位被清0。更新了重装载寄存器的值后，在代码持续执行之前不必等待RUD值被清零(在进入省电模式前需等待RUD值清零)。 |

### 状态寄存器 (FWDGT_STAT)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器可以按半字(16 位)或字(32 位)访问。

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

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

| 保留 | RUD | PUD |
|---|---|---|
| | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:2 | 保留 | 必须保持复位值。 |
| 1 | RUD | 独立看门狗定时器计数器重装载值更新<br>FWDGT_RLD寄存器写操作时，该位被置1，此时读取FWDGT_RLD寄存器的任何值都是无效的。在FWDGT_RLD寄存器更新后，该位由硬件清零。 |
| 0 | PUD | 独立看门狗定时器预分频值更新<br>FWDGT_PSC寄存器写操作时，该位被置1，此时读取FWDGT_PSC寄存器的任何值都是无效的。在FWDGT_PSC寄存器更新后，该位由硬件清零。 |

# 14.2. 窗口看门狗定时器(WWDGT)

## 14.2.1. 简介

窗口看门狗定时器(WWDGT)用来监测由软件故障导致的系统故障。窗口看门狗定时器开启后，7位向下递减计数器值逐渐减小。计数值达到0x3F时会产生系统复位(CNT[6]位被清0)。在计数器计数值达到窗口寄存器值之前，计数器的更新也会产生系统复位。因此软件需要在给定的区间内更新计数器。窗口看门狗定时器在计数器计数值达到0x40，会产生一个提前唤醒标志，如果使能中断将会产生提前唤醒中断。

窗口看门狗定时器时钟是由APB1时钟预分频而来。窗口看门狗定时器适用于需要精确计时的场合。

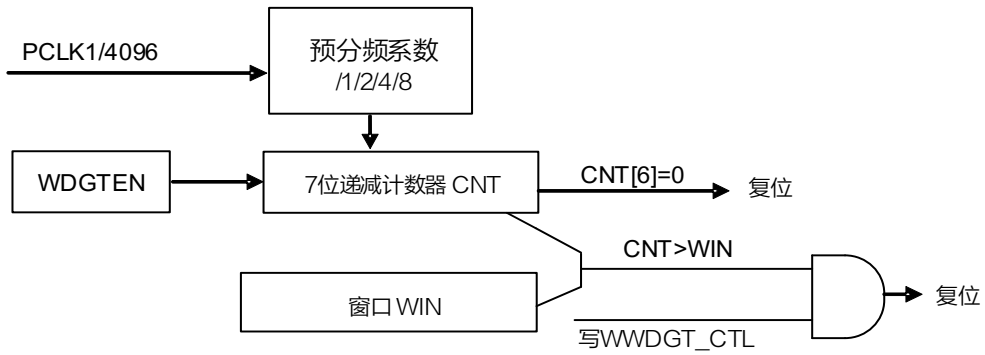
## 14.2.2. 主要特征

- 可编程的7位自由运行向下递减计数器。
- 当窗口看门狗使能后，有以下两种情况会产生复位：
  - 当计数器达到0x3F时产生复位；
  - 当计数器的值大于窗口寄存器的值时，更新计数器会产生复位。
- 提前唤醒中断(EWI)：看门狗定时器打开，中断使能，计数值达到0x40时会产生中断。
- 可以配置窗口看门狗定时器在调试模式下选择停止还是继续工作。

## 14.2.3. 功能说明

如果窗口看门狗定时器使能(将WWDGT_CTL寄存器的WDGTEN位置1)，计数值达到0x3F的时候产生系统复位(CNT[6]位被清0)。或者在计数值达到窗口寄存器值之前，更新计数器也会产生系统复位。

**图 14-2. 窗口看门狗定时器框图**



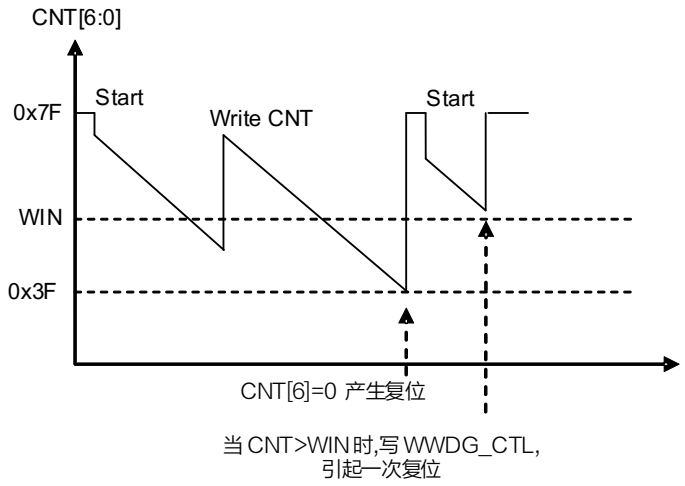

上电复位之后窗口看门狗定时器总是关闭的。软件可以向WWDGT_CTL的WDGTEN写1开启窗口看门狗定时器。窗口看门狗定时器打开后，计数器始终递减计数，计数器配置的值应该大于0x3F，也就是说CNT[6]位应该被置1。CNT[5:0]决定了两次重装载之间的最大间隔时间。计数器的递减速度取决于APB1时钟和预分频器(WWDGT_CFG寄存器的PSC[1:0]位)。

配置寄存器(WWDGT_CFG)中的WIN[6:0]位用来设定窗口值。当计数器的值小于窗口值，且大于0x3F的时候，重装载向下计数器可以避免复位，否则在其他时候进行重加载就会引起复位。

对WWDGT_CFG寄存器的EWIE位置1可以使能提前唤醒中断(EWI)，当计数值达到0x40的时候该中断产生。同时可以用相应的中断服务程序(ISR)来触发特定的行为(例如通信或数据记录)，来分析软件故障的原因以及在器件复位的时候挽救重要数据。此外，在ISR中软件可以重装载计数器来管理软件系统检查等。在这种情况下，窗口看门狗定时器将永远不会复位但是可以用于其他地方。

通过将WWDGT_STAT寄存器的EWIF位写0可以清除EWI中断。

**图 14-3. 窗口看门狗定时器时序图**




窗口看门狗定时器超时的计算公式如下：

$$t_{WWDGT}=t_{PCLK1} \times 4096 \times 2^{PSC} \times (CNT[5:0]+1) \quad (ms) \qquad (14\text{-}1)$$

其中：

$t_{WWDGT}$：窗口看门狗定时器的超时时间

$t_{PCLK1}$： APB1以ms为单位的时钟周期

$t_{WWDGT}$的最大值和最小值请参考*表14-2. 在60MHz(fPCLK1)时的最大/最小超时值*。

**表 14-2. 在 60MHz($f_{PCLK1}$)时的最大/最小超时值**

| 预分频系数 | PSC[1:0] | 最小超时<br>CNT[6:0] =0x40 | 最大超时<br>CNT[6:0]=0x7F |
|---|---|---|---|
| 1 / 1 | 00 | 68.2 μs | 4.3ms |
| 1 / 2 | 01 | 136.4 μs | 8.6 ms |
| 1 / 4 | 10 | 272.8μs | 17.2 ms |
| 1 / 8 | 11 | 545.6 μs | 34.4 ms |

如果MCU调试模块中的WWDGT_HOLD位被清0，即使Cortex®-M4内核停止工作(调试模式下)，窗口看门狗定时器也可以继续工作。当WWDGT_HOLD位被置1时，窗口看门狗定时器在调试模式下停止。

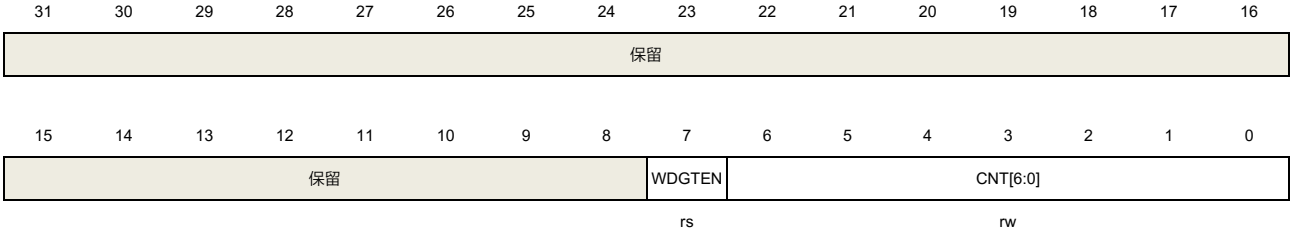
## 14.2.4. WWDGT 寄存器

WWDGT基地址：0x4000 2C00

### 控制寄存器 (WWDGT_CTL)

地址偏移：0x00
复位值：0x0000 007F

该寄存器可以按半字(16 位)或字(32 位)访问。

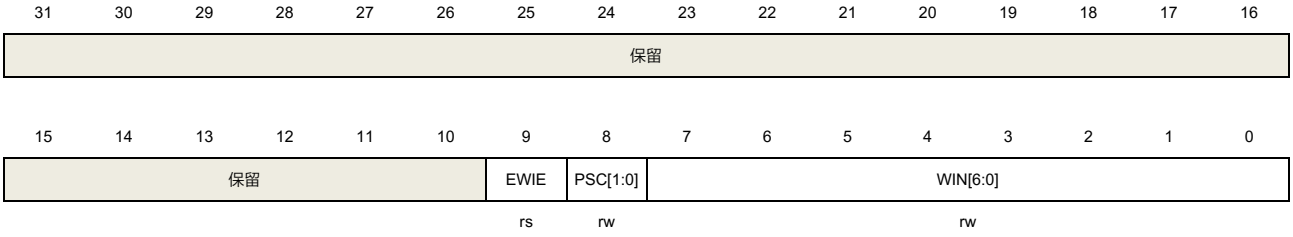



| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 必须保持复位值。 |
| 7 | WDGTEN | 开启窗口看门狗定时器，硬件复位的时候清0，写0无效。<br>0：关闭窗口看门狗定时器<br>1：开启窗口看门狗定时器 |
| 6:0 | CNT[6:0] | 看门狗定时器计数器的值。当计数值从0x40降到0x3F时，产生看门狗定时器复位。<br>当计数器值高于窗口值的时候，写计数器可以产生看门狗定时器复位。 |

### 配置寄存器 (WWDGT_CFG)

地址偏移：0x04
复位值：0x0000 007F

该寄存器可以按半字(16 位)或字(32 位)访问。



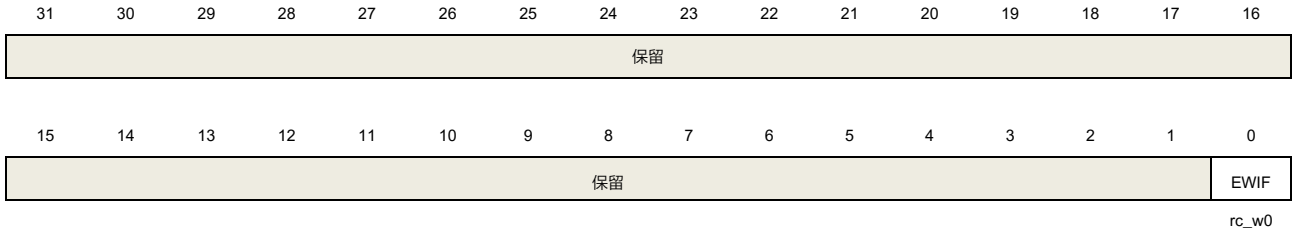

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:10 | 保留 | 必须保持复位值。 |
| 9 | EWIE | 提前唤醒中断使能。如果该位被置1，计数值达到0x40时触发中断，或者在计数值达到窗口寄存器之前更新计数器也能触发中断。该位由硬件复位清0，或通过RCU模块 |

| | | |
|---|---|---|
| | | 的WWDGT软件复位来清0。写0没有任何作用。 |
| 8:7 | PSC[1:0] | 预分频器，看门狗定时器计数器的时间基准<br>00：PCLK1 / 4096 / 1<br>01：PCLK1 / 4096 / 2<br>10：PCLK1 / 4096 / 4<br>11：PCLK1 / 4096 / 8 |
| 6:0 | WIN[6:0] | 窗口值，当看门狗定时器计数器的值大于窗口值时，写看门狗定时器计数器(WWDGT_CTL的CNT位)会产生复位。 |

### 状态寄存器 (WWDGT_STAT)

地址偏移： 0x08
复位值： 0x0000 0000

该寄存器可以按半字(16 位)或字(32 位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:1 | 保留 | 必须保持复位值 |
| 0 | EWIF | 提前唤醒中断标志位。当计数值达到0x40或者在计数值达到窗口寄存器之前更新计数器，即使中断没有被使能(WWDGT_CFG中的EWIE位为0)该位也会被硬件置1。这个位可以通过写0清零，写1无效。 |

# 15. 实时时钟(RTC)

## 15.1. 简介

实时时钟RTC通常用于日历时钟。RTC电路分属于两个电源域。一部分位于备份域中，该部分包括一个32位的累加计数器、一个闹钟、一个预分频器、一个分频器以及RTC时钟配置寄存器。这表明系统复位或者从待机模式唤醒时，RTC的设置和时间都保持不变。另一部分位于$V_{DD}$电源域中，该部分只包括APB接口以及一组控制寄存器。在本章接下来的部分，将详细介绍RTC的功能。

## 15.2. 主要特征

- 32位可编程计数器，用于计数运行时间
  - 可编程的预分频器：分频系数最高可达$2^{20}$
- 独立时钟域：
  - PCLK1时钟域
  - RTC时钟域(该时钟必须比PCLK1时钟至少慢4倍)
- RTC时钟源：
  - HXTAL时钟除以128
  - LXTAL振荡电路时钟
  - IRC40K振荡电路时钟
- 可屏蔽的中断源：
  - 闹钟中断
  - 秒中断
  - 溢出中断

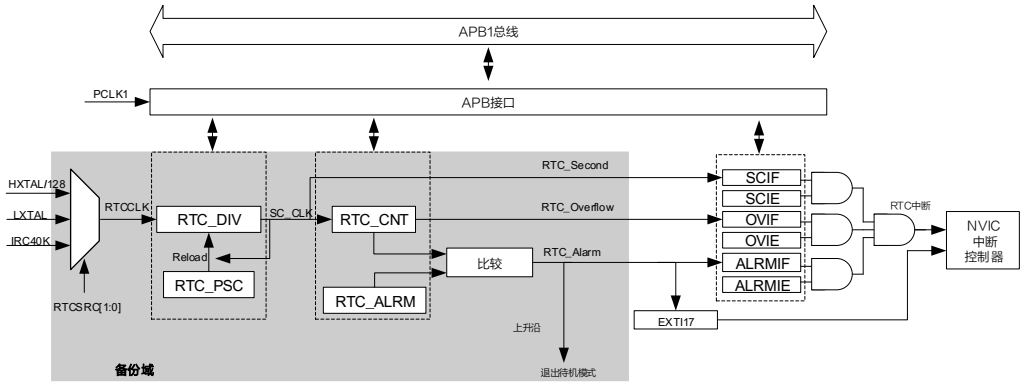
## 15.3. 功能说明

RTC电路包含两个主要部分，位于PCLK1时钟域的APB接口和位于RTC时钟域的RTC内核。

APB接口与APB1总线相连，包含一组寄存器，通过APB1总线可以对其进行读或写。

RTC内核包含两个主要模块。一个是RTC预分频模块，用来产生RTC时间基准SC_CLK，RTC预分频模块包含一个20位可编程分频器(RTC预分频器)。该分频器可以通过对RTC时钟源分频产生SC_CLK。如果对RTC_INTEN寄存器中秒中断位进行使能，RTC会在每个SC_CLK上升沿产生一个秒中断。另外一个模块是一个32位可编程计数器，其数值可以被初始化为当前系统时间。如果对RTC_INTEN寄存器的闹钟中断位进行使能，RTC会在系统时间等于闹钟时间(存储于RTC_ALRMH/L寄存器)时产生一个闹钟中断。

图 15-1. RTC 框图




### 15.3.1. RTC 复位

APB接口和RTC_INTEN寄存器会随着系统复位进行复位。RTC内核(预分频器、分频器、计数器以及闹钟)只会随备份域复位进行复位。

通过下面的步骤，可以在复位后访问备份域寄存器以及RTC寄存器：

1. 通过对RCU_APB1EN寄存器中的PMUEN和BKPEN位进行置位，使能电源以及备份接口时钟。

2. 通过对PMU_CTL中的BKPWEN位进行置位，使能对备份域寄存器和RTC的访问。

### 15.3.2. RTC 读取

APB接口和RTC内核分属于两个不同电源域。

在RTC内核中，只有计数器和分频器寄存器为可读寄存器。这两个寄存器的值以及RTC标志会在每个RTC时钟的上升沿进行内部更新，并与APB1时钟进行重新同步。

当APB接口从禁用状态使能后，建议不要立即进行读操作，因为这些寄存器的首次内部更新可能尚未完成。这表明，在系统复位、电源复位、从待机/深度睡眠模式下唤醒时，APB接口是被禁用的，但是RTC内核仍然保持运行。在这类情况下，正确的读操作应该先将RTC_CTL寄存器的RSYNF清零并等待其被硬件置位。WFI和WFE指令对于RTC的APB接口没有影响。

### 15.3.3. RTC 配置

RTC内核中的RTC_PSC、RTC_CNT和RTC_ALRM寄存器都是可写的。只有在外设进入配置模式后，这些寄存器的值才能进行设置。通过查询RTC_CTL寄存器的CMF位，可以检测配置模式的状态。只有在外设退出配置模式后，之前对这些寄存器的写操作才能生效，且至少需要三个RTCCLK周期才能完成。当写操作完成后，RTC_CTL寄存器中的LWOFF位的值变为'1'。下一个写操作必须等待上次写操作完成之后才能进行。

配置过程如下：

1. 等待RTC_CTL寄存器中的LWOFF位的值变为1；

2. 通过设置RTC_CTL寄存器中的CMF位来进入配置模式；
3. 对RTC寄存器进行写操作；
4. 通过将RTC_CTL寄存器中的CMF清零来退出配置模式；
5. 等待RTC_CTL寄存器中的LWOFF位的值变为1。

### 15.3.4. RTC 标志位

RTC秒中断标志(SCIF)在RTC计数器更新之前的最后一个RTCCLK周期进行置位。

RTC闹钟中断标志(ALRMIF)在计数器达到存储于闹钟寄存器中的RTC闹钟值加1的前一个RTCCLK周期进行置位。

RTC溢出中断标志(OVIF)在计数器值达到0x00的前一个RTCCLK周期置位。

RTC闹钟的写操作需要按照下列方法之一来进行，以达到和秒中断标志保持同步：

■ 启用RTC闹钟中断，在RTC内部中断服务程序内更新RTC闹钟寄存器及/或RTC计数器寄存器的内容。

■ RTC闹钟寄存器及/或RTC计数器寄存器的内容必须等待RTC控制寄存器中SCIF置位后才能更新。

**图 15-2. RTC 秒信号及闹钟信号的波形 (RTC_PSC = 3, RTC_ALRM = 2)**

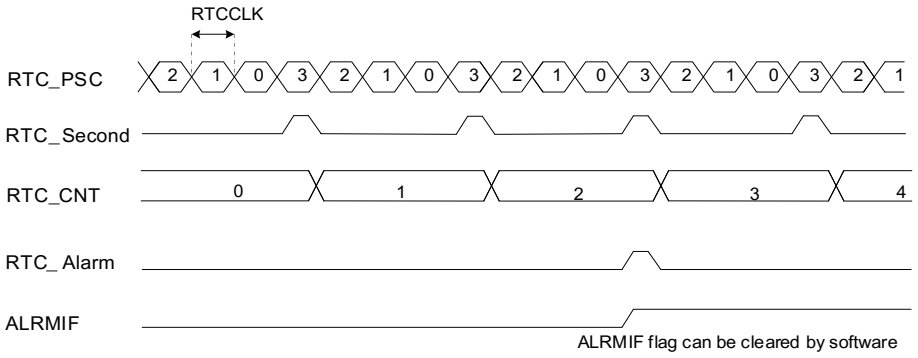
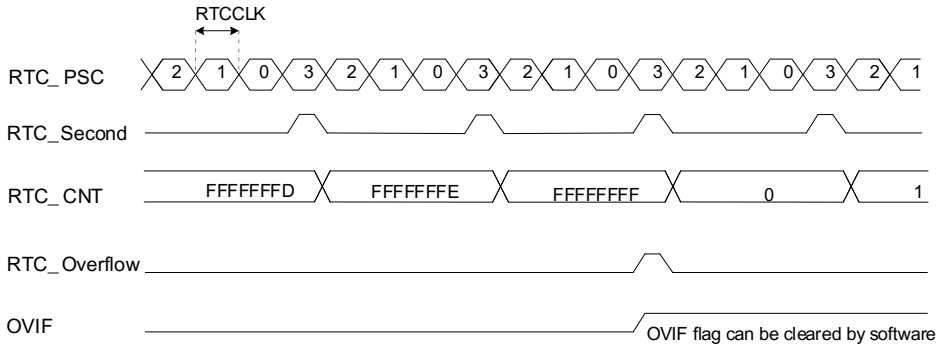



**图 15-3. RTC 秒信号及溢出信号的波形(RTC_PSC = 3)**



## 15.4. RTC 寄存器

RTC基地址：0x4000 2800

### 15.4.1. RTC 中断使能寄存器 (RTC_INTEN)

偏移地址：0x00
复位值：0x0000

该寄存器可以按半字(16 位)或字(32 位)访问

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | | | | | | OVIE | ALRMIE | SCIE |
| | | | | | | | | | | | | | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:3 | 保留 | 必须保持复位值 |
| 2 | OVIE | 溢出中断使能<br>0：禁用溢出中断<br>1：使能溢出中断 |
| 1 | ALRMIE | 闹钟中断使能<br>0：禁用闹钟中断<br>1：使能闹钟中断 |
| 0 | SCIE | 秒中断使能<br>0：禁用秒中断<br>1：使能秒中断 |

### 15.4.2. RTC 控制寄存器 (RTC_CTL)

偏移地址：0x04
复位值：0x0020

该寄存器可以按半字(16 位)或字(32 位)访问

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | | | LWOFF | CMF | RSYNF | OVIF | ALRMIF | SCIF |
| | | | | | | | | | | r | rw | rc_w0 | rc_w0 | rc_w0 | rc_w0 |

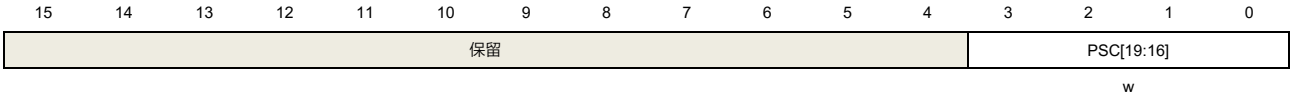
| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:6 | 保留 | 必须保持复位值 |
| 5 | LWOFF | 上次对RTC寄存器写操作标志<br>0：上次对RTC寄存器写操作没有完成<br>1：上次对RTC寄存器写操作已经完成 |

| | | |
|---|---|---|
| 4 | CMF | 配置模式标志<br>0：退出配置模式<br>1：进入配置模式 |
| 3 | RSYNF | 寄存器同步标志<br>0：寄存器没有与APB1时钟同步<br>1：寄存器已经与APB1时钟同步 |
| 2 | OVIF | 溢出中断标志<br>0：没有检测到溢出事件<br>1：检测到溢出事件。当RTC_INTEN寄存器的OVIE位被置1，中断发生。 |
| 1 | ALRMIF | 闹钟中断标志<br>0：没有检测到闹钟事件<br>1：检测到闹钟事件。当RTC_INTEN寄存器的ALRMIE位被置1，RTC全局中断发生。并且当EXTI17被使能中断模式，发生RTC闹钟中断。 |
| 0 | SCIF | 秒中断标志<br>0：没有检测到秒事件<br>1：检测到秒事件。当RTC_INTEN 寄存器的SCIE位被置1，中断发生。<br>当分频器重加载RTC_PSC值时，硬件将该位置1，从而累加RTC计数器。 |

### 15.4.3. RTC 预分频寄存器高位 (RTC_PSCH)

偏移地址： 0x08
复位值： 0x0000

该寄存器可以按半字(16 位)或字(32 位)访问

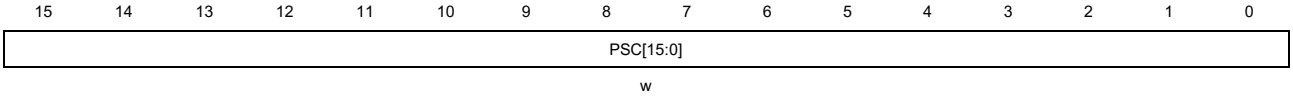



| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15:4 | 保留 | 必须保持复位值 |
| 3:0 | PSC[19:16] | RTC预分频器高位值 |

### 15.4.4. RTC 预分频寄存器低位 (RTC_PSCL)

偏移地址： 0x0C
复位值： 0x8000

该寄存器可以按半字(16 位)或字(32 位)访问

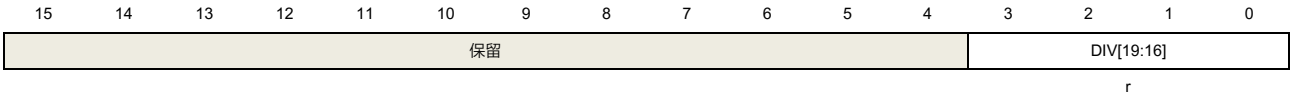


| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 15:0 | PSC[15:0] | RTC预分频器低位值<br>SC_CLK的频率是RTCCLK的频率除以(PSC[19:0]+1) |

### 15.4.5. RTC 分频器高位 (RTC_DIVH)

偏移地址： 0x10
复位值： 0x0000

该寄存器可以按半字(16 位)或字(32 位)访问

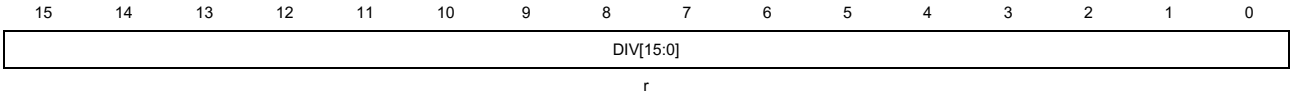



| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 15:4 | 保留 | 必须保持复位值 |
| 3:0 | DIV[19:16] | RTC分频器高位 |

### 15.4.6. RTC 分频器低位 (RTC_DIVL)

偏移地址： 0x14
复位值： 0x8000

该寄存器可以按半字(16 位)或字(32 位)访问

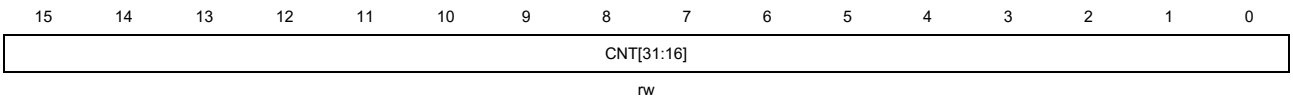



| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 15:0 | DIV[15:0] | RTC分频器低位<br>当RTC预分频寄存器或者RTC计数寄存器更新时，RTC分频器寄存器会由硬件自动加载 |

### 15.4.7. RTC 计数寄存器高位 (RTC_CNTH)

偏移地址： 0x18
复位值： 0x0000

该寄存器可以按半字(16 位)或字(32 位)访问



| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 15:0 | CNT[31:16] | RTC计数寄存器高位 |

### 15.4.8. RTC 计数寄存器低位 (RTC_CNTL)

偏移地址： 0x1C
复位值： 0x0000

该寄存器可以按半字(16 位)或字(32 位)访问

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 15:0 | CNT[15:0] | RTC计数寄存器低位 |

### 15.4.9. RTC 闹钟寄存器高位 (RTC_ALRMH)

偏移地址： 0x20
复位值： 0xFFFF

该寄存器可以按半字(16 位)或字(32 位)访问

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ALRM[31:16] | | | | | | | | | | | | | | | |
| w | | | | | | | | | | | | | | | |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 15:0 | ALRM[31:16] | RTC闹钟值高位 |

### 15.4.10. RTC 闹钟寄存器低位 (RTC_ALRML)

偏移地址： 0x24
复位值： 0xFFFF

该寄存器可以按半字(16 位)或字(32 位)访问

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 15:0 | ALRM[15:0] | RTC闹钟值低位 |

# 16. 定时器(TIMER)

**表 16-1. 定时器(TIMERx)分为五种类型**

| 定时器 | 定时器 0/7 | 定时器 1/2/3/4 | 定时器 8/11 | 定时器 9/10/12/13 | 定时器 5/6 |
|---|---|---|---|---|---|
| 类型 | 高级 | 通用(L0) | 通用(L1) | 通用(L2) | 基本 |
| 预分频器 | 16 位 | 16 位 | 16 位 | 16 位 | 16 位 |
| 计数器 | 16 位 | 16 位 | 16 位 | 16 位 | 16 位 |
| 计数模式 | 向上，向下，中央对齐 | 向上，向下，中央对齐 | 只有向上 | 只有向上 | 只有向上 |
| 可重复性 | ● | × | × | × | × |
| 捕获/比较通道数 | 4 | 4 | 2 | 1 | 0 |
| 互补和死区时间 | ● | × | × | × | × |
| 中止输入 | ● | × | × | × | × |
| 单脉冲 | ● | ● | ● | × | ● |
| 正交译码器 | ● | ● | × | × | × |
| 主-从管理 | ● | ● | ● | × | × |
| 内部连接 | ●(1) | ●(2) | ●(3) | × | TRGO TO DAC |
| DMA | ● | ● | × | × | ●(4) |
| Debug 模式 | ● | ● | ● | ● | ● |

(1)

| | | | | |
|---|---|---|---|---|
| TIMER0 | **ITI0:** TIMER4_TRGO | **ITI1:** TIMER1_TRGO | **ITI2:** TIMER2_TRGO | **ITI3:** TIMER3_TRGO |
| TIMER7 | **ITI0:** TIMER0_TRGO | **ITI1:** TIMER1_TRGO | **ITI2:** TIMER3_TRGO | **ITI3:** TIMER4_TRGO |

(2)

| | | | | |
|---|---|---|---|---|
| TIMER1 | **ITI0:** TIMER0_TRGO | **ITI1:** 0 | **ITI2:** TIMER2_TRGO | **ITI3:** TIMER3_TRGO |
| TIMER2 | **ITI0:** TIMER0_TRGO | **ITI1:** TIMER1_TRGO | **ITI2:** TIMER4_TRGO | **ITI3:** TIMER3_TRGO |
| TIMER3 | **ITI0:** TIMER0_TRGO | **ITI1:** TIMER1_TRGO | **ITI2:** TIMER2_TRGO | **ITI3:** TIMER7_TRGO |
| TIMER4 | **ITI0:** TIMER1_TRGO | **ITI1:** TIMER2_TRGO | **ITI2:** TIMER3_TRGO | **ITI3:** TIMER7_TRGO |

(3)

| | | | | |
|---|---|---|---|---|
| TIMER8 | **ITI0:** TIMER1_TRGO | **ITI1:** TIMER2_TRGO | **ITI2:** TIMER9_TRGO | **ITI3:** TIMER10_TRGO |
| TIMER11 | **ITI0:** TIMER3_TRGO | **ITI1:** TIMER4_TRGO | **ITI2:** TIMER12_TRGO | **ITI3:** TIMER13_TRGO |

(4) 只有更新事件可以产生 DMA 请求，定时器 5 和定时 6 中没有 DMAS 位(DMA 请求源选择位)。

# 16.1. 高级定时器(TIMERx,x=0,7)

## 16.1.1. 简介

高级定时器(TIMER0和TIMER7)是四通道定时器，支持输入捕获和输出比较。可以产生PWM信号控制电机和电源管理。高级定时器含有一个16位无符号计数器。

高级定时器是可编程的，可以用来计数，其外部事件可以驱动其他定时器。

高级定时器包含了一个死区时间插入模块，非常适合电机控制。

定时器和定时器之间是相互独立，但是它们的计数器可以被同步在一起形成一个更大的定时器。

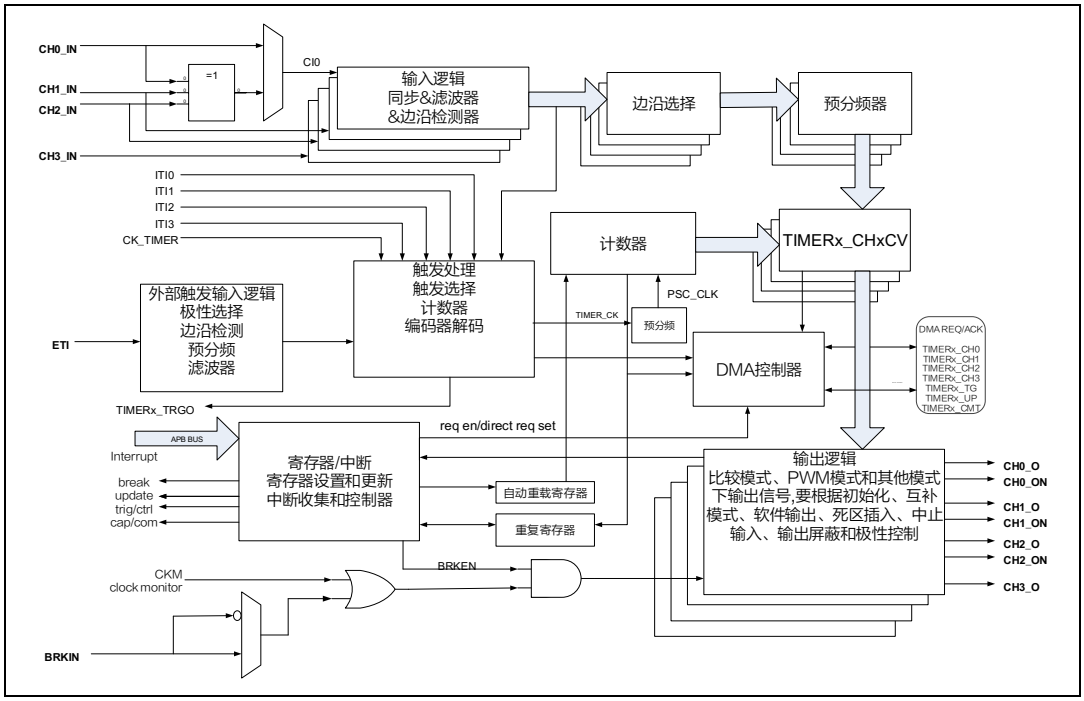
## 16.1.2. 主要特征

- 总通道数：4；
- 计数器宽度：16位；
- 定时器时钟源可选：内部时钟，内部触发，外部输入，外部触发；
- 多种计数模式：向上计数，向下计数和中央计数；
- 正交编码器接口：用来追踪运动和分辨旋转方向和位置；
- 霍尔传感器接口：用来做三相电机控制；
- 可编程的预分频器：16位。运行时可以被改变；
- 每个通道可配置：输入捕获模式，输出比较模式，可编程的PWM模式，单脉冲模式；
- 可编程的死区时间；
- 自动重装载功能；
- 可编程的计数器重复功能；
- 中止输入功能；
- 中断输出和DMA请求：更新事件，触发事件，比较/捕获事件和中止事件；
- 多个定时器的菊链使得一个定时器可以同时启动多个定时器；
- 定时器的同步允许被选择的定时器在同一个时钟周期开始计数；
- 定时器主-从管理。

## 16.1.3. 结构框图

*图16-1. 高级定时器结构框图*提供了高级定时器的内部配置细节。

图 16-1. 高级定时器结构框图




## 16.1.4. 功能描述

### 时钟源配置

高级定时器可以由内部时钟源 CK_TIMER 或者由 SMC(TIMERx_SMCFG 寄存器位[2:0])控制的复用时钟源驱动。

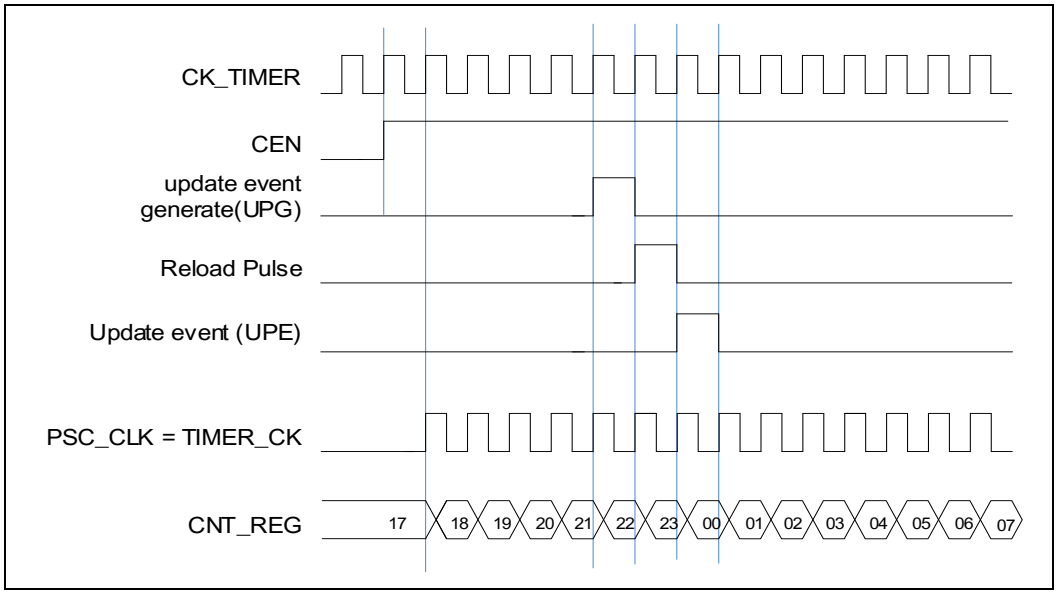
■ SMC[2:0]==3'b000，定时器选择内部时钟源(连接到RCU模块的CK_TIMER)

如果 SMC[2:0]==3'b000，默认用来驱动计数器预分频器的是内部时钟源 CK_TIMER。当 CEN 置位，CK_TIMER 经过预分频器(预分频值由 TIMERx_PSC 寄存器确定)产生 PSC_CLK。

这种模式下，驱动预分频器计数的 TIMER_CK 等于来自于 RCU 模块的 CK_TIMER

如果将 TIMERx_SMCFG 寄存器的 SMC[2:0]设置为 0x1、0x2、0x3 和 0x7，预分频器被其他时钟源(由 TIMERx_SMCFG 寄存器的 TRGS [2:0]区域选择)驱动，在下文说明。当 SMC 位被设置为 0x4、0x5 和 0x6，计数器预分频器时钟源由内部时钟 CK_TIMER 驱动。

**图 16-2. 内部时钟分频为 1 时，计数器的时序图**




■ SMC[2:0]==3'b111(外部时钟模式0)，定时器选择外部输入引脚作为时钟源

计数器预分频器可以在 TIMERx_CH0/ TIMERx_CH1 引脚的每个上升沿或下降沿计数。这种模式可以通过设置 SMC [2:0]为 0x7 同时设置 TRGS[2:0]为 0x4，0x5 或 0x6 来选择。

计数器预分频器也可以在内部触发信号 ITI0/1/2/3 的上升沿计数。这种模式可以通过设置 SMC [2:0]为 0x7 同时设置 TRGS [2:0]为 0x0, 0x1, 0x2 或者 0x3。

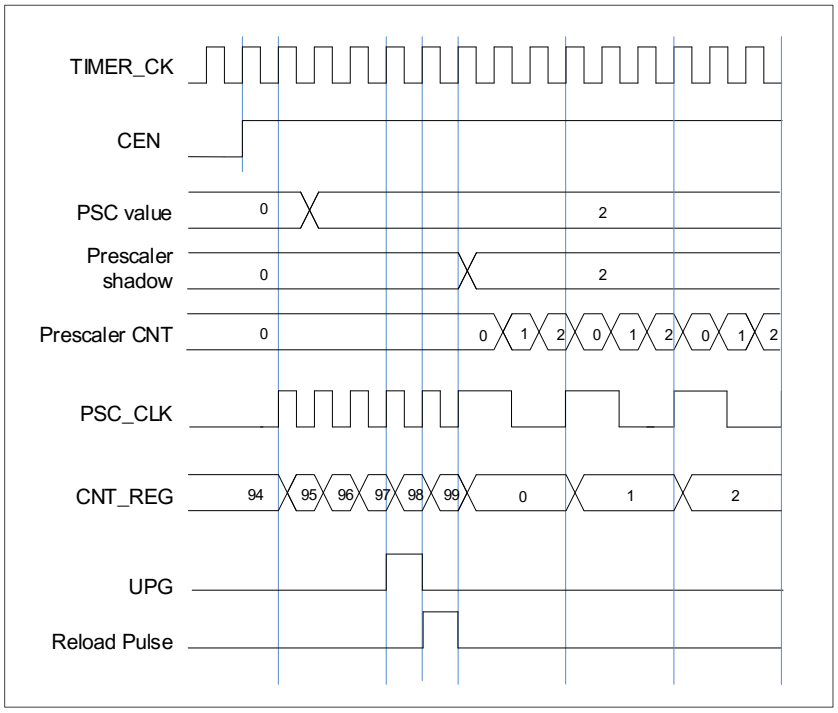
■ SMC1==1'b1(外部时钟模式1)，定时器选择外部输入引脚ETI作为时钟源

计数器预分频器可以在外部引脚 ETI 的每个上升沿或下降沿计数。这种模式可以通过设置 TIMERx_SMCFG 寄存器中的 SMC1 位为 1 来选择。另一种选择 ETI 信号作为时钟源方式是，设置 SMC [2:0]为 0x7 同时设置 TRGS [2:0]为 0x7。注意 ETI 信号是通过数字滤波器采样 ETI 引脚得到的。如果选择 ETI 信号为时钟源，触发控制器包括边沿监测电路将在每个 ETI 信号上升沿产生一个时钟脉冲来为计数器预分频器提供时钟。

### 时钟预分频器

预分频器可以将定时器的时钟(TIMER_CK)频率按 1 到 65536 之间的任意值分频，分频后的时钟 PSC_CLK 驱动计数器计数。分频系数受预分频寄存器 TIMERx_PSC 控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 16-3. 当 PSC 数值从 0 变到 2 时，计数器的时序图




### 计数器向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从 0 开始向上连续计数到自动加载值(定义在 TIMERx_CAR 寄存器中)，一旦计数器计数到自动加载值，会重新从 0 开始向上计数。如果设置了重复计数器，在(TIMERx_CREP+1)次上溢后产生更新事件，否则在每次上溢时都会产生更新事件。在向上计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 0。

当通过 TIMERx_SWEVG 寄存器的 UPG 位置 1 来设置更新事件时，计数值会被清 0，并产生更新事件。

如果 TIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(重复计数器，计数器自动重载寄存器，预分频寄存器)都将被更新。

下面这些图给出了一些例子，当 TIMERx_CAR=0x99 时，计数器在不同预分频因子下的行为。

图 16-4. 向上计数时序图，PSC=0/2

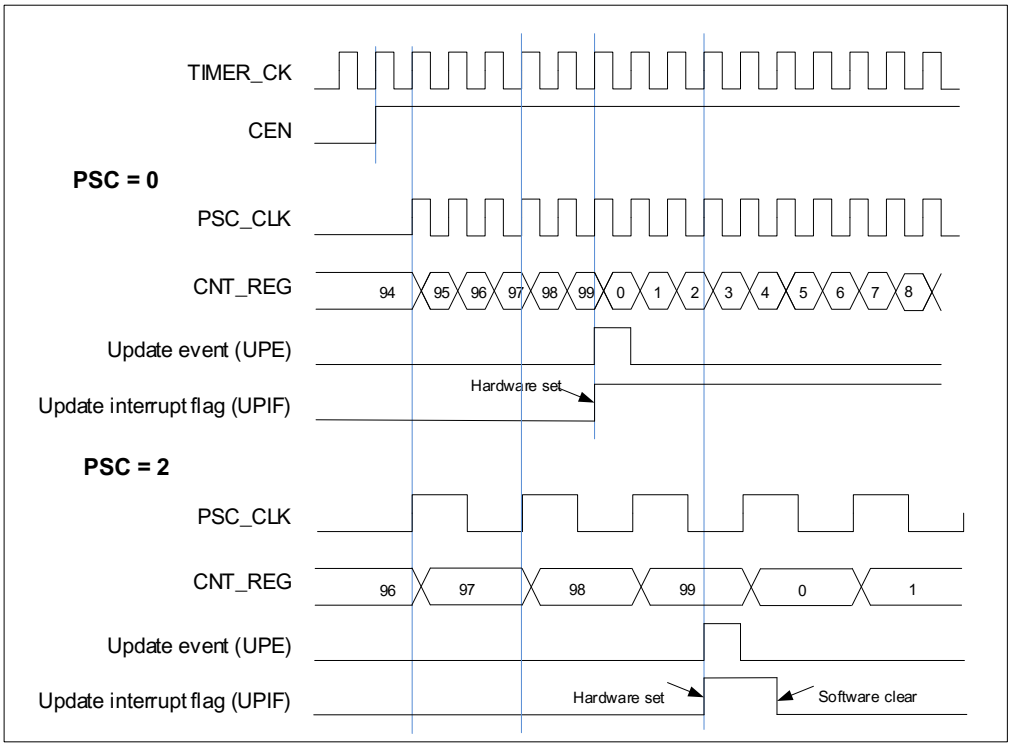
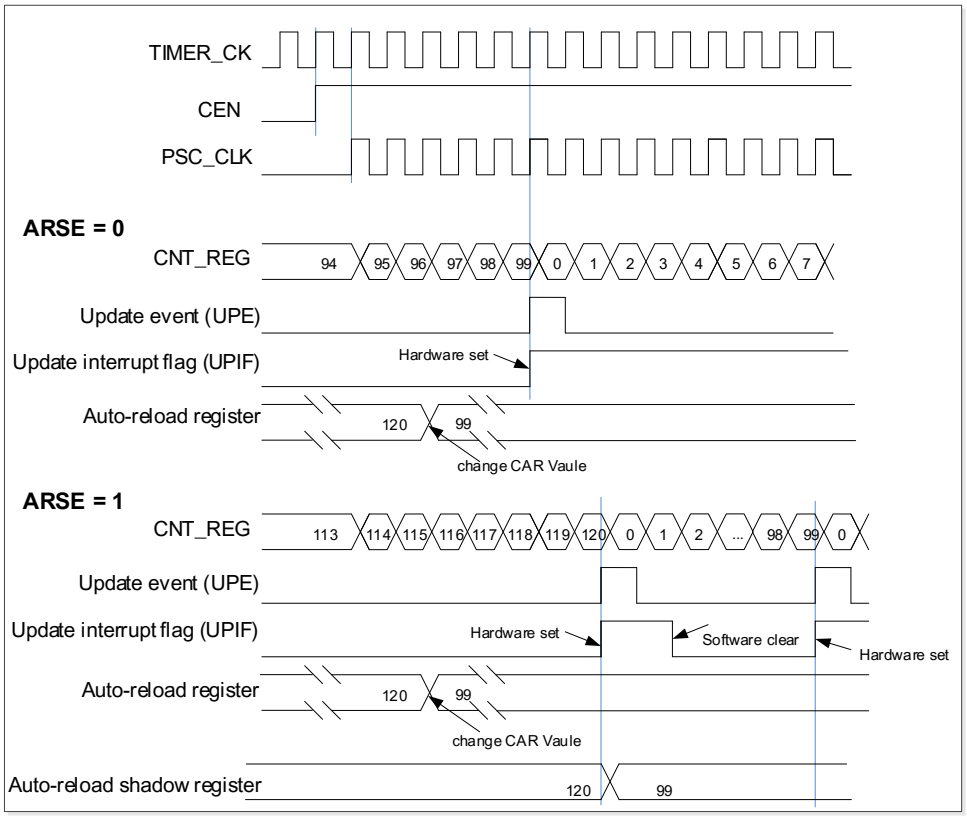



图 16-5. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



### 计数器向下计数模式

在这种模式，计数器的计数方向是向下计数。计数器从自动加载值(定义在 TIMERx_CAR 寄存器中)向下连续计数到 0。一旦计数器计数到 0，计数器会重新从自动加载值开始计数。如果设置了重复计数器，在(TIMERx_CREP+1)次下溢后产生更新事件，否则在每次下溢时都会产生更新事件。在向下计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 1。

当通过 TIMERx_SWEVG 寄存器的 UPG 位置 1 来设置更新事件时，计数值会被初始化为自动加载值，并产生更新事件。

如果 TIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(重复计数器，计数器自动重载寄存器，预分频寄存器)都将被更新。

下面这些图给出了一些例子，当 TIMERx_CAR=0x99 时，计数器在不同时钟频率下的行为。

**图 16-6. 向下计数时序图，PSC=0/2**

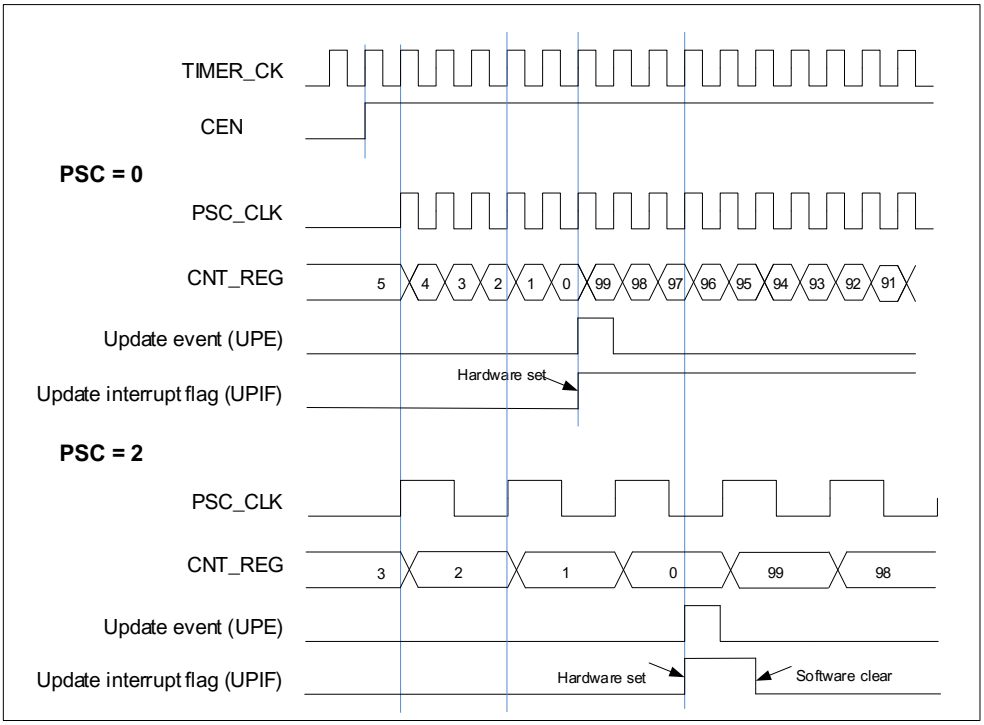
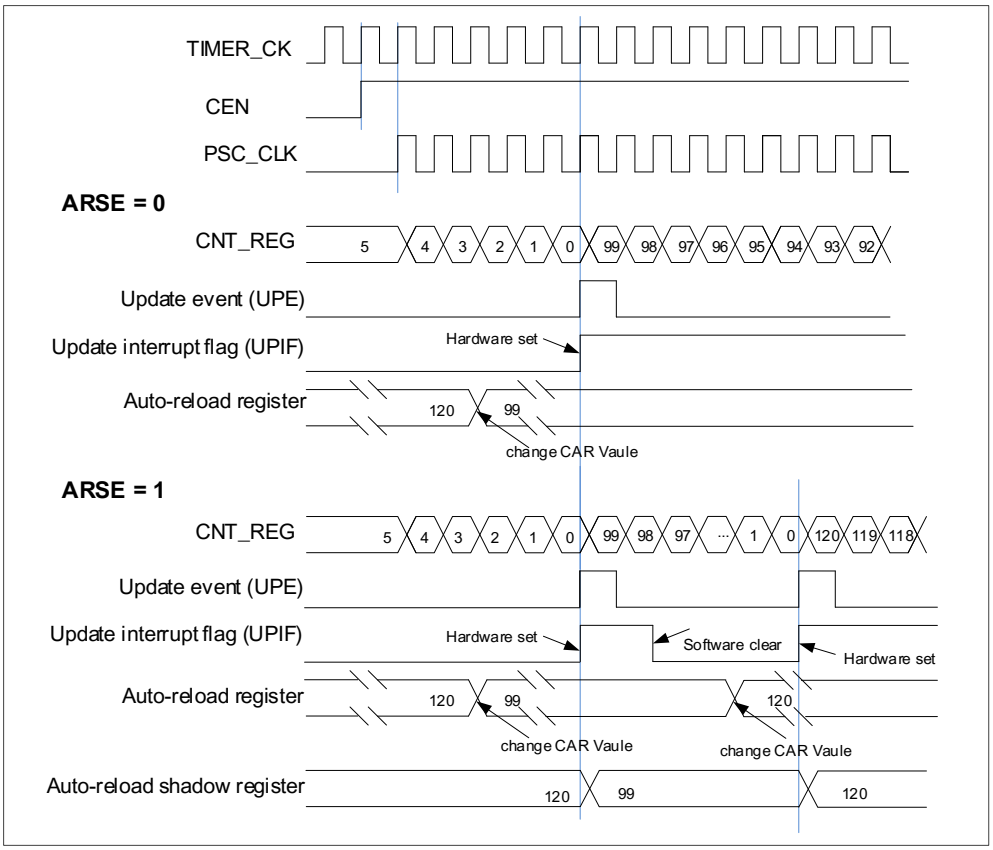


**图 16-7. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值**




### 计数器中央对齐模式

在中央对齐模式下，计数器交替的从 0 开始向上计数到自动加载值，然后再向下计数到 0。。向上计数模式中，定时器模块在计数器计数到自动加载值-1 产生一个上溢事件；向下计数模式中，定时器模块在计数器计数到 1 时产生一个下溢事件。在中央计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 只读，表明了的计数方向。

将 TIMERx_SWEVG 寄存器的 UPG 位置 1 可以初始化计数值为 0，并产生一个更新事件，而无需考虑计数器在中央模式下是向上计数还是向下计数。

上溢或者下溢时，TIMERx_INTF 寄存器中的 UPIF 位都会被置 1，然而 CHxIF 位置 1 与 TIMERx_CTL0 寄存器中 CAM 的值有关。具体细节参考*图 16-8. 中央计数模式计数器时序图*。

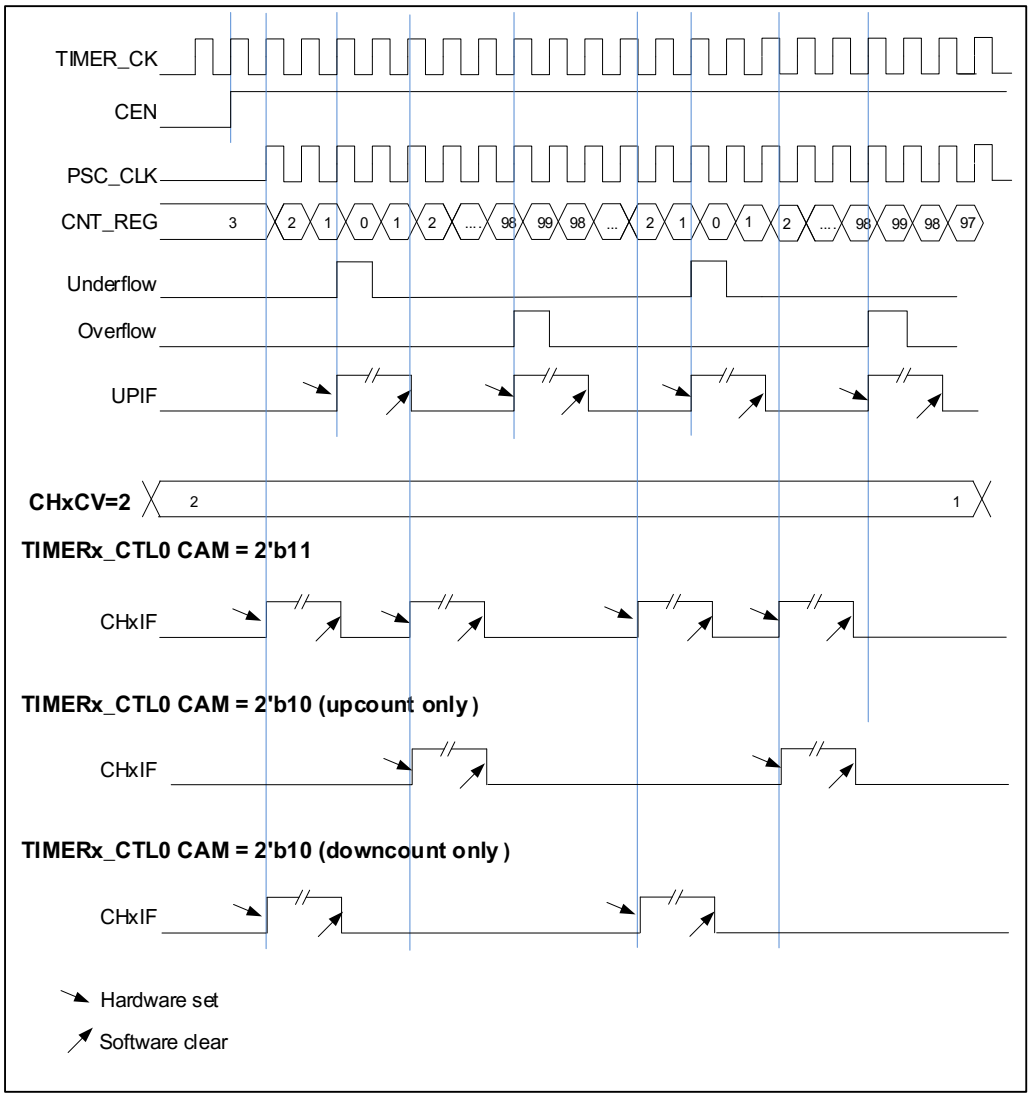
如果 TIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(重复计数器，计数器自动重载寄存器，预分频寄存器)都将被更新。

*图 16-8. 中央计数模式计数器时序图*给出了一些例子，当 TIMERx_CAR=0x99，

TIMERx_PSC=0x0时，计数器的行为

**图 16-8. 中央计数模式计数器时序图**




## 更新事件(来自上溢/下溢)频率配置

更新事件的生成频率(来自上溢和下溢事件)可以通过TIMERx_CREP寄存器进行配置。重复计数器是用来在N+1个计数周期之后产生更新事件，更新定时器的寄存器，N为TIMERx_CREP寄存器的CREP。重复计数器在每次计数器上溢和下溢时递减(向上计数模式中不存在下溢事件；向下计数模式中不存在上溢事件)。

将TIMERx_SWEVG寄存器的UPG位置1可以重载TIMERx_CREP寄存器中CREP的值并产生一个更新事件。

新写入的CREP值将在下一次更新事件到来时生效。当CREP的值为奇数，并且计数器在中央对齐模式下计数时，更新事件发生在上溢或下溢取决于写入的CREP值何时生效。如果在写入

奇数到CREP寄存器后由软件生成更新事件(UPG位置1)，则在下溢时产生更新事件。如果在写入奇数到CREP寄存器后下一个更新事件发生在上溢，此后将在上溢时产生更新事件。

**图 16-9. 中央计数模式下计数器重复时序图**

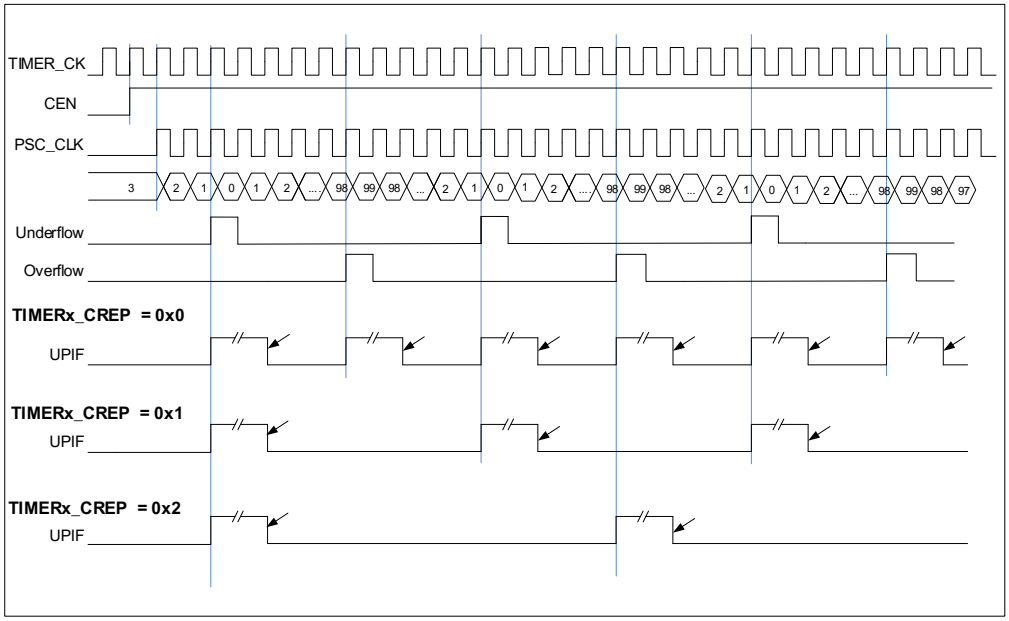



**图 16-10. 在向上计数模式下计数器重复时序图**

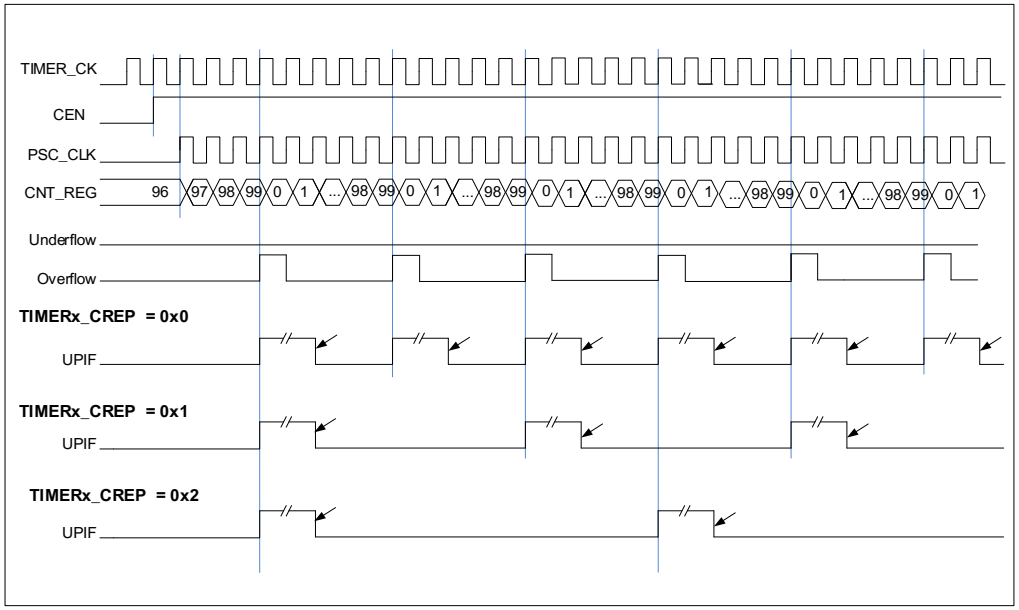
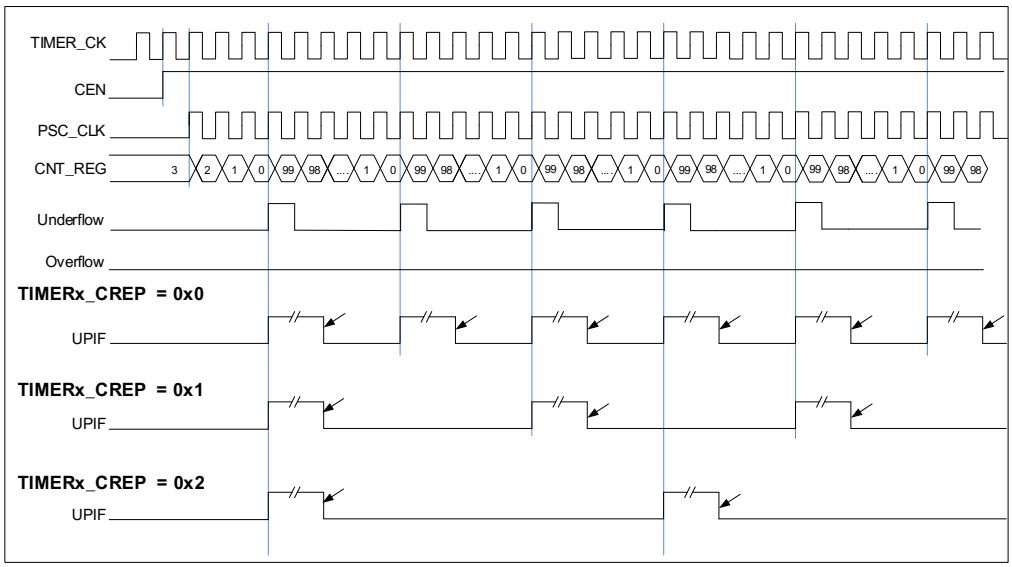


图 16-11. 在向下计数模式下计数器重复时序图



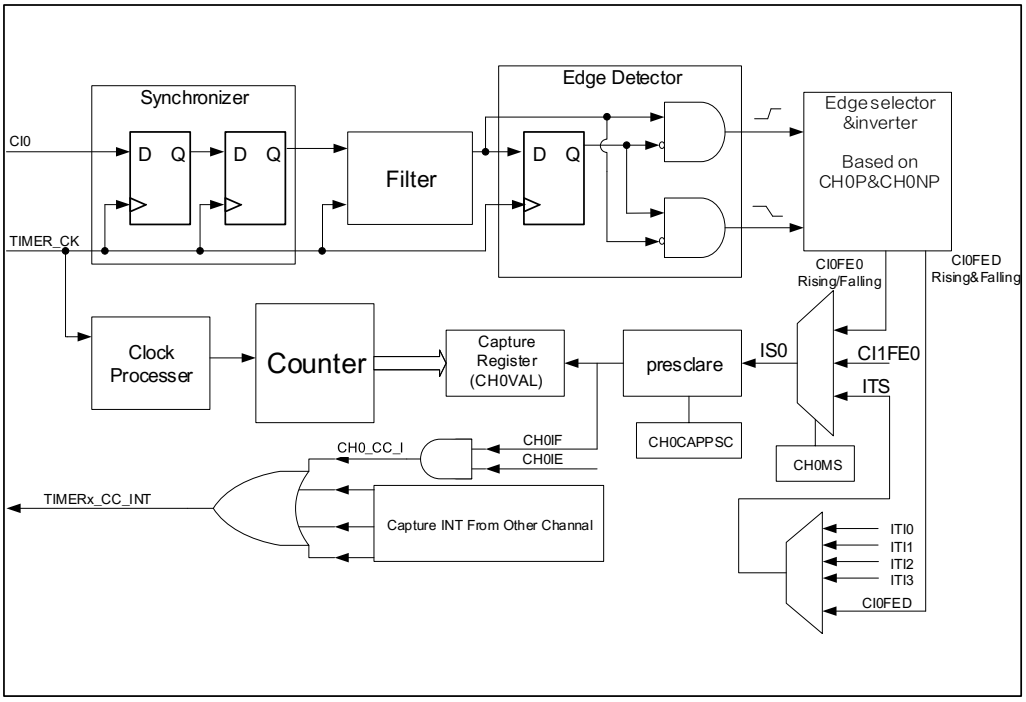

## 输入捕获和输出比较通道

高级定时器拥有四个独立的通道用于捕获输入或比较输出是否匹配。每个通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

■ 通道输入捕获功能

通道输入捕获功能允许通道测量一个波形时序，频率，周期，占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，TIMERx_CHxCV 寄存器会捕获计数器当前的值，同时 CHxIF 位被置 1，如果 CHxIE = 1 则产生通道中断。

**图 16-12. 通道输入捕获原理**




通道输入信号 CIx 有两种选择，一种是 TIMERx_CHx 信号，另一种是 TIMERx_CH0,TIMERx_CH1 和 TIMERx_CH2 异或之后的信号。通道输入信号 CIx 先被 TIMER_CK 信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置 CHxP 选择使用上升沿或者下降沿。配置 CHxMS.，可以选择其他通道的输入信号，内部触发信号。配置 IC 预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生，TIMERx_CHxCV 存储计数器的值。

配置步骤如下：

**第一步：** 滤波器配置(TIMERx_CHCTL0寄存器中CHxCAPFLT)：
根据输入信号和请求信号的质量，配置相应的CHxCAPFLT。

**第二步：** 边沿选择(TIMERx_CHCTL2寄存器中CHxP/CHxNP)：
配置CHxP/CHxNP选择上升沿或者下降沿。

**第三步：** 捕获源选择(TIMERx_CHCTL0寄存器中CHxMS)：
一旦通过配置CHxMS选择输入捕获源，必须确保通道配置在输入模式(CHxMS!=0x0)，而且TIMERx_CHxCV寄存器不能再被写。

**第四步：** 中断使能(TIMERx_DMAINTEN寄存器中CHxIE 和 CHxDEN)：
使能相应中断，可以获得中断和DMA请求。

**第五步：** 捕获使能(TIMERx_CHCTL2寄存器中CHxEN)。

**结果：** 当期望的输入信号发生时，TIMERx_CHxCV被设置成当前计数器的值，CHxIF为置1。如果CHxIF位已经为1，则CHxOF位置1。根据TIMERx_DMAINTEN寄存器中CHxIE和CHxDEN的配置，相应的中断和DMA请求会被提出。

**直接产生：** 软件设置CHxG位，会直接产生中断和DMA请求。

通道输入捕获功能也可用来测量 TIMERx_CHx 引脚上信号的脉冲波宽度。例如，一个 PWM 波连接到 CI0。配置 TIMERx_CHCTL0 寄存器中 CH0MS 为 2'b01，选择通道 0 的捕获信号为 CI0 并设置上升沿捕获。配置 TIMERx_CHCTL0 寄存器中 CH1MS 为 2'b10，选择通道 1 捕获信号为 CI0 并设置下降沿捕获。计数器配置为复位模式，在通道 0 的上升沿复位。TIMERX_CH0CV 寄存器测量 PWM 的周期值，TIMERx_CH1CV 寄存器测量 PWM 占空比值。

■ 通道输出比较功能

在通道输出比较功能，TIMERx 可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的 TIMERx_CHxCV 寄存器与计数器的值匹配时，根据 CHxCOMCTL 的配置，这个通道的输出可以被置高电平，被置低电平或者反转。当计数器的值与 TIMERx_CHxCV 寄存器的值匹配时，CHxIF 位被置 1，如果 CHxIE = 1 则会产生中断，如果 CxCDE=1 则会产生 DMA 请求。

配置步骤如下：

**第一步：** 时钟配置：
配置定时器时钟源，预分频器等。

**第二步：** 比较模式配置：
设置CHxCOMSEN位来配置输出比较影子寄存器；
设置CHxCOMCTL位来配置输出模式(置高电平/置低电平/反转)；
设置CHxP/CHxNP位来选择有效电平的极性；
设置CHxEN使能输出。

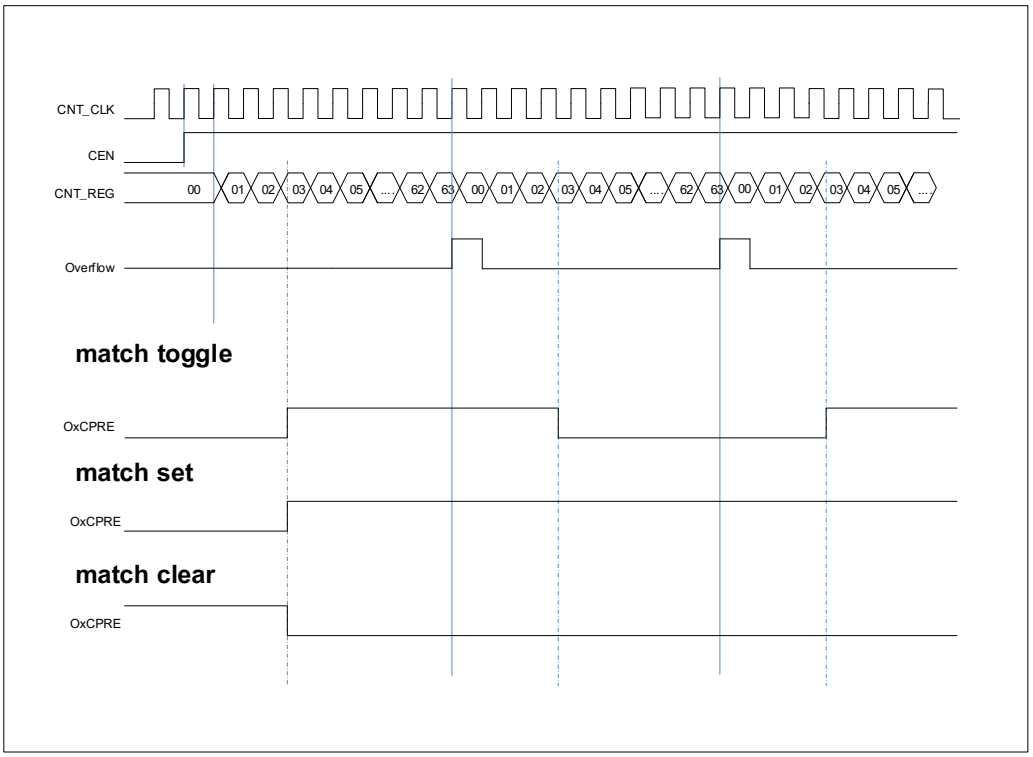
**第三步：** 通过CHxIE/CxCDE位配置中断/DMA请求使能。

**第四步：** 通过TIMERx_CAR寄存器和TIMERx_CHxCV寄存器配置输出比较时基：
TIMERx_CHxCV可以在运行时根据你所期望的波形而改变。

**第五步：** 设置CEN位使能定时器。

*图16-13. 三种输出比较模式* 显示了三种比较输出模式：反转/置高电平/置低电平，CAR=0x63, CHxVAL=0x3。

图 16-13. 三种输出比较模式




## 输出 PWM 功能

在 PWM 输出模式下(PWM 模式 0 是配置 CHxCOMCTL 为 3'b110，PWM 模式 1 是配置 CHxCOMCTL 为 3'b111)，通道根据 TIMERx_CAR 寄存器和 TIMERx_CHxCV 寄存器的值，输出 PWM 波形。

根据计数模式，我们可以分为两种PWM波：EAPWM(边沿对齐PWM)和CAPWM(中央对齐PWM)。

EAPWM的周期由TIMERx_CAR寄存器值决定，占空比由TIMERx_CHxCV寄存器值决定。*图 16-14. EAPWM时序图* 显示了EAPWM的输出波形和中断。

CAPWM的周期由(2*TIMERx_CAR寄存器值)决定，占空比由(2*TIMERx_CHxCV寄存器值)决定。*图16-15. CAPWM时序图* 显示了CAPWM的输出波形和中断。

在 PWM0 模式下 (CHxCOMCTL==3'b110)，如果 TIMERx_CHxCV 寄存器的值大于 TIMERx_CAR寄存器的值，通道输出一直为有效电平。

在PWM0模式下(CHxCOMCTL==3'b110)，如果TIMERx_CHxCV寄存器的值等于0，通道输出一直为无效电平。

**图 16-14. EAPWM 时序图**

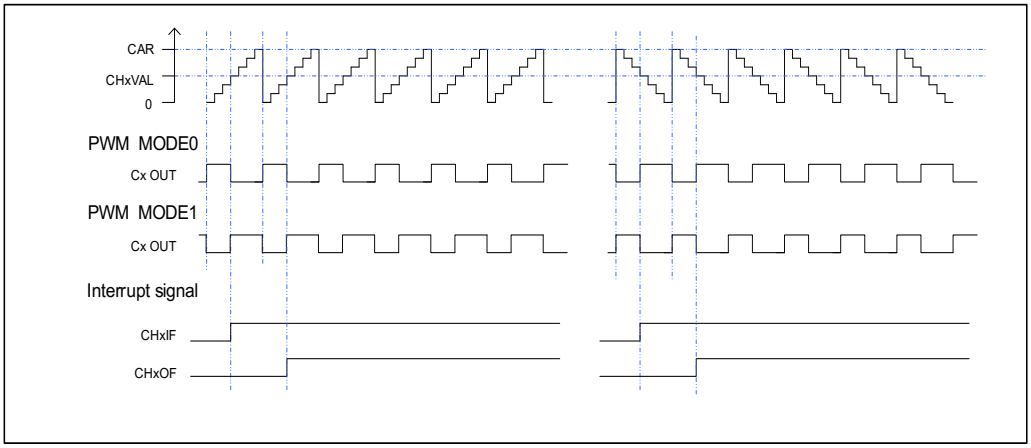
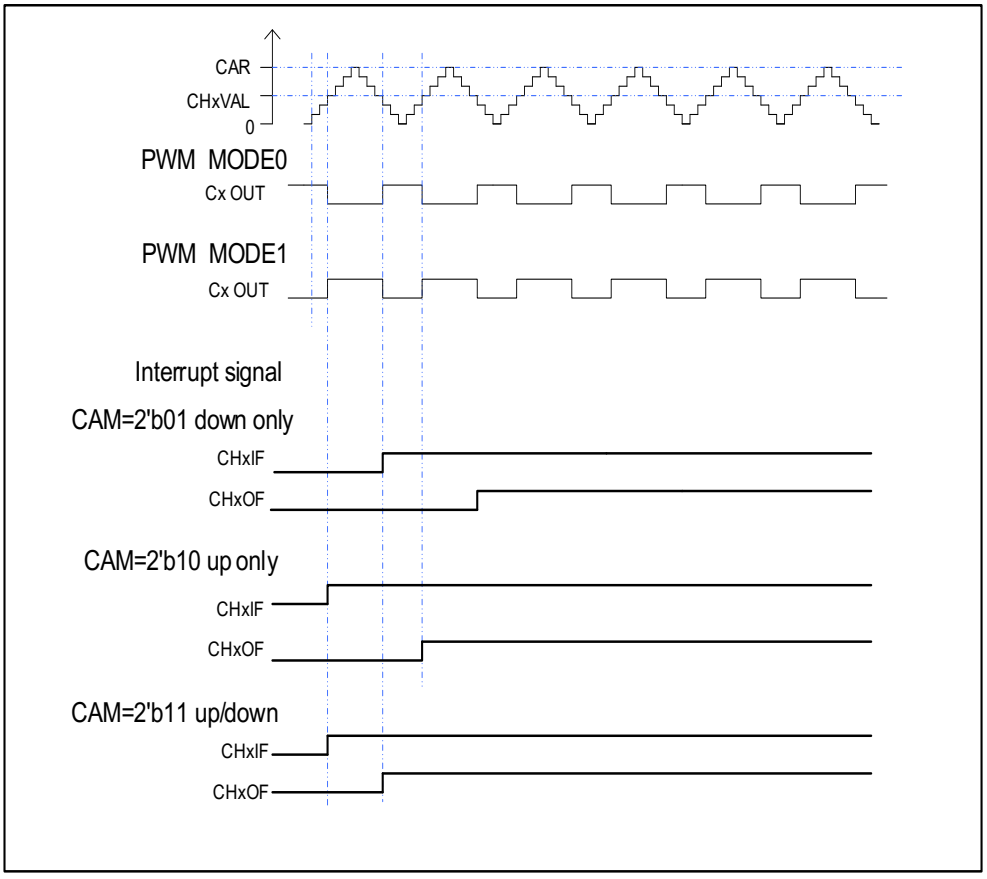



**图 16-15. CAPWM 时序图**




### 通道输出准备信号

当 TIMERx 用于输出匹配比较模式下，设置 CHxCOMCTL 位可以定义 OxCPRE 信号(通道 x 准备信号)类型。OxCPRE 信号有若干类型的输出功能，包括，设置 CHxCOMCTL=0x00 可以保持原始电平；设置 CHxCOMCTL=0x01 可以将 OxCPRE 信号设置为高电平；设置 CHxCOMCTL=0x02 可以将 OxCPRE 信号设置为低电平；设置 CHxCOMCTL=0x03，在计数

器值和 TIMERx_CHxCV 寄存器的值匹配时，可以翻转输出信号。

PWM 模式 0 和 PWM 模式 1 是 OxCPRE 的另一种输出类型，设置 CHxCOMCTL 位域位 0x06 或 0x07 可以配置 PWM 模式 0/PWM 模式 1。在这些模式中，根据计数器值和 TIMERx_CHxCV 寄存器值的关系以及计数方向，OxCPRE 信号改变其电平。具体细节描述，请参考相应的位。

设置 CHxCOMCTL =0x04 或 0x05 可以实现 OxCPRE 信号的强制输出功能。输出比较信号能够直接由软件强置为有效或无效状态，而不依赖于 TIMERx_CHxCV 的值和计数器值之间的比较结果。

设置 CHxCOMCEN=1，当由外部 ETI 引脚信号产生的 ETIFE 信号为高电平时，OxCPRE 被强制为低电平。在下一次更新事件到来时，OxCPRE 信号才会回到有效电平状态。

### 通道输出互补 PWM

CHx_O 和 CHx_ON 是一对互补输出通道，这两个信号不能同时有效。TIMERx 有四路通道，只有前三路有互补输出通道。互补信号 CHx_O 和 CHx_ON 是由一组参数来决定：TIMERx_CHCTL2 寄存器中的 CHxEN 和 CHxNEN 位，TIMERx_CCHP 寄存器中和 TIMERx_CTL1 寄存器中的 POEN，ROS，IOS，ISOx 和 ISOxN 位。输出极性由 TIMERx_CHCTL2 寄存器中的 CHxP 和 CHxNP 位来决定。

**表 16-2. 由参数控制的互补输出表**

<table>
<tr><th colspan="5">互补参数</th><th colspan="2">输出状态</th></tr>
<tr><th>POEN</th><th>ROS</th><th>IOS</th><th>CHxEN</th><th>CHxNEN</th><th>CHx_O</th><th>CHx_ON</th></tr>
<tr><td rowspan="4">0</td><td rowspan="4">0/1</td><td rowspan="2">0</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O / CHx_ON = LOW<br>CHx_O / CHx_ON 输出禁能(1)</td></tr>
<tr><td>1</td><td colspan="2">CHx_O/CHx_ON输出关闭状态(2):<br>通道先输出无效电平：CHx_O = CHxP，CHx_ON = CHxNP)；如果死区产生时钟未失效，在死区时间之后：<br>CHx_O = ISOx，CHx_ON = ISOxN (3)</td></tr>
<tr><td rowspan="2">1</td><td>0</td></tr>
<tr><td>1</td></tr>
<tr><td>1</td><td>x</td><td>x</td><td colspan="2">CHx_O/CHx_ON输出关闭状态:<br>通道先输出无效电平：CHx_O = CHxP，CHx_ON = CHxNP)；如果死区产生时钟未失效，在死区时间之后：<br>CHx_O = ISOx，CHx_ON = ISOxN</td></tr>
<tr><td rowspan="3">1</td><td rowspan="3">0</td><td rowspan="3">0/1</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O/CHx_ON = LOW<br>CHx_O/CHx_ON输出禁能</td></tr>
<tr><td>1</td><td>CHx_O = LOW<br>CHx_O输出禁能</td><td>CHx_ON=OxCPRE⊕(4)CHxNP<br>CHx_ON输出使能</td></tr>
<tr><td>1</td><td>0</td><td>CHx_O=OxCPRE⊕CHxP<br>CHx_O输出使能</td><td>CHx_ON = LOW<br>CHx_ON输出禁能</td></tr>
</table>

| 互补参数 | | | | | 输出状态 | |
|---|---|---|---|---|---|---|
| POEN | ROS | IOS | CHxEN | CHxNEN | CHx_O | CHx_ON |
| | | | | 1 | CHx_O=OxCPRE⊕CHxP<br>CHx_O输出使能 | CHx_ON=(!OxCPRE)(5)⊕CHxNP<br>CHx_ON输出使能 |
| | 1 | | 0 | 0 | CHx_O = CHxP<br>CHx_O输出关闭状态 | CHx_ON = CHxNP<br>CHx_ON输出关闭状态 |
| | | | | 1 | CHx_O = CHxP<br>CHx_O输出关闭状态 | CHx_O=OxCPRE⊕CHxNP<br>CHx_ON输出使能 |
| | | | 1 | 0 | CHx_O=OxCPRE⊕CHxP<br>CHx_O输出使能 | CHx_ON = CHxNP<br>CHx_ON输出关闭状态 |
| | | | | 1 | CHx_O=OxCPRE⊕CHxP<br>CHx_O输出使能 | CHx_ON=(!OxCPRE)⊕CHxNP<br>CHx_ON输出使能 |

**注意**:

（1） 输出禁能：CHx_O / CHx_ON 输出与对应引脚断开，对应引脚电平受 GPIO 上下拉配置控制，无上下拉时为悬空高阻态；

（2） 输出关闭状态：CHx_O / CHx_ON 输出无效电平(CHx_O = 0⊕CHxP = CHxP)；

（3） 详情见中止模式章节。

（4） ⊕：异或操作；

（5） (!OxCPRE)：OxCPRE 信号的互补信号。

### 互补 PWM 插入死区时间

设置 CHxEN 和 CHxNEN 为 1'b1 同时设置 POEN，死区插入就会被使能。DTCFG 位域定义了死区时间，死区时间对除了通道 3 以外通道有效。死区时间的细节，请参考 TIMERx_CCHP 寄存器。

死区时间的插入，确保了通道互补的两路信号不会同时有效。

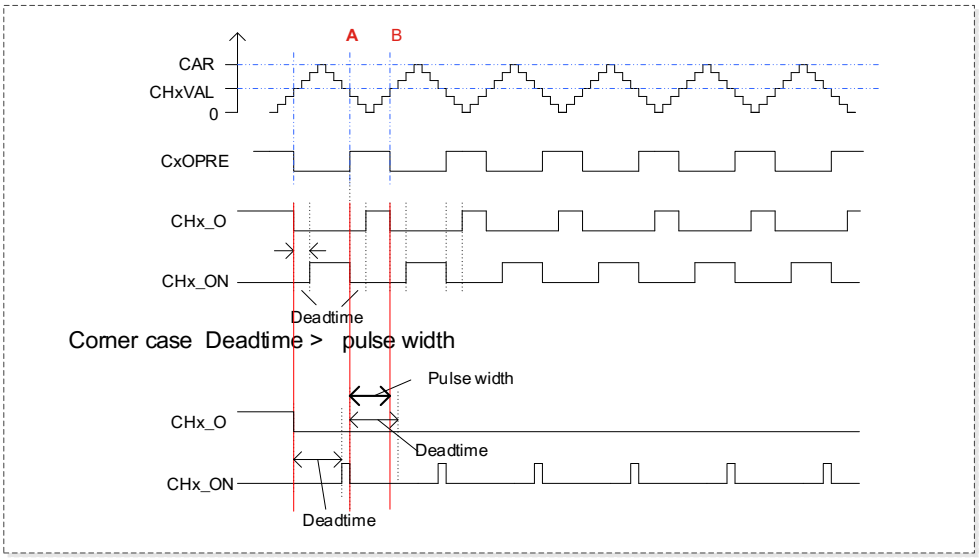
在PWM0模式，当通道x匹配发生时(TIMERx计数器=TIMERx_CHxCV)，OxCPRE反转。在*图*

*16-16. 带死区时间的互补输出*中的A点，CHx_O信号在死区时间内为低电平，直到死区时间过后才变为高电平，而CHx_ON信号立刻变为低电平。同样，在B点，计数器再次匹配(TIMERx计数器= TIMERx_CHxCV)，OxCPRE信号被清0，CHx_O信号被立即清零，CHx_ON信号在死区时间内仍然是低电平，在死区时间过后才变为高电平。

有时会有一些死角事件发生，例如：

■ 如果死区延时大于或者等于CHx_ON信号的占空比，CHx_ON信号一直为无效值。

**图 16-16. 带死区时间的互补输出**



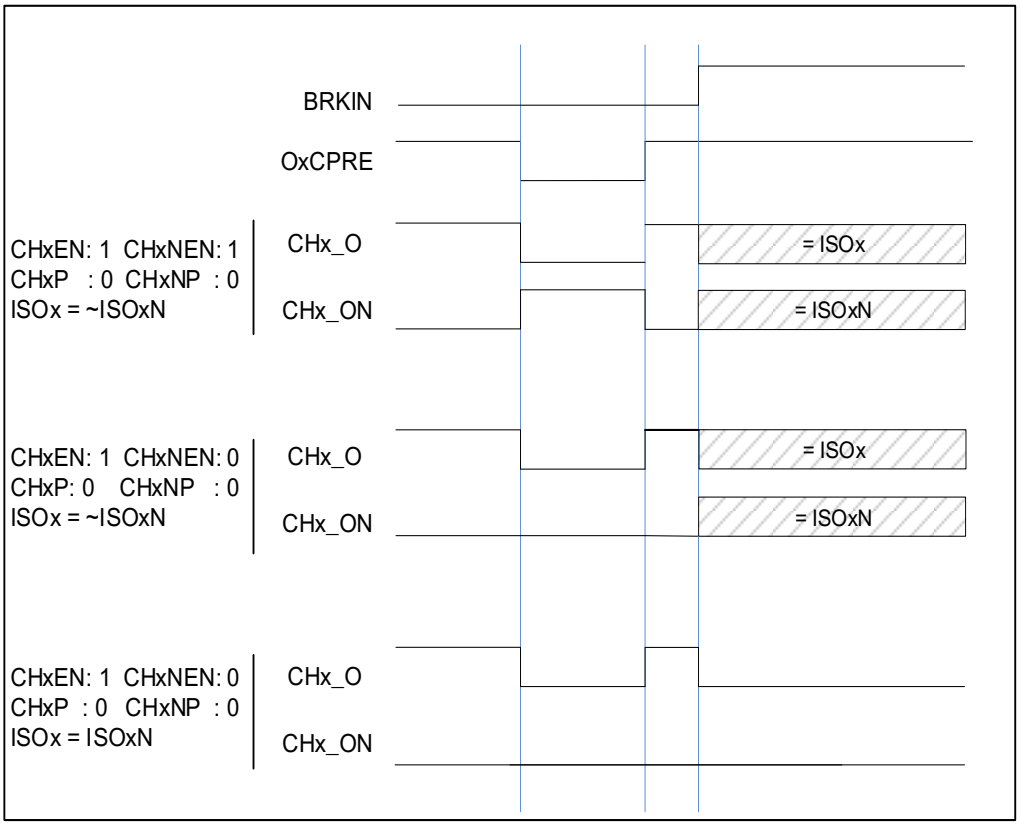

## 中止模式

使用中止模式时，输出 CHx_O 和 CHx_ON 信号电平被以下位控制，TIMERx_CCHP 寄存器的 POEN, IOS 和 ROS 位，TIMERx_CTL1 寄存器的 ISOx 和 ISOxN 位。当中止事件发生时，CHx_O 和 CHx_ON 信号输出不能同时设置为有效电平。中止源可以选择中止输入引脚，也可以选择 HXTAL 时钟失效事件，时钟失败事件由 RCU 中的时钟监视器(CKM)产生。将 TIMERx_CCHP 寄存器的 BRKEN 位置 1 可以使能中止功能。TIMERx_CCHP 寄存器的 BRKP 位决定了中止输入极性。

发生中止时，POEN 位被异步清除，一旦 POEN 位为 0，CHx_O 和 CHx_ON 被 TIMERx_CTL1 寄存器中的 ISOx 位和 ISOxN 驱动。如果 IOS=0，定时器释放输出使能，否则输出使能仍然为高。起初互补输出被置于复位状态，然后死区时间产生器重新被激活，以便在一个死区时间后驱动输出，输出电平由 ISOx 和 ISOxN 位配置。

发生中止时，TIMERx_INTF 寄存器的 BRKIF 位被置 1。如果 BRKIE=1，中断产生。

图 16-17. 通道响应中止输入(高电平有效)时，输出信号的行为




## 正交译码器

正交译码器功能使用由TIMERx_CH0和TIMERx_CH1引脚生成的CI0FE0和CI1FE1正交信号各自相互作用产生计数值。在每个输入源改变期间，DIR位会发生改变。输入源可以是只有CI0FE0，可以只有CI1FE1，或着可以同时有CI0FE0和CI1FE1，通过设置SMC=0x01, 0x02或0x03来选择使用哪种模式。计数器计数方向改变的机制如*表16-3. 不同编码器模式下的计数方向*所示。正交译码器可以当作一个带有方向选择的外部时钟，这意味着计数器会在0和自动加载值之间连续的计数。因此，用户必须在计数器开始计数前配置TIMERx_CAR寄存器。

表 16-3. 不同编码器模式下的计数方向

| 计数模式 | 电平 | CI0FE0 | | CI1FE1 | |
|---|---|---|---|---|---|
| | | 上升 | 下降 | 上升 | 下降 |
| 编码器模式0 SMC[2:0]=3'b001 | CI1FE1=1 | 向下 | 向上 | - | - |
| | CI1FE1=0 | 向上 | 向下 | - | - |
| 编码器模式1 SMC [2:0]=3'b010 | CI0FE0=1 | - | - | 向上 | 向下 |
| | CI0FE0=0 | - | - | 向下 | 向上 |
| 编码器模式2 SMC [2:0]=3'b011 | CI1FE1=1 | 向下 | 向上 | X | X |
| | CI1FE1=0 | 向上 | 向下 | X | X |
| | CI0FE0=1 | X | X | 向上 | 向下 |

| 计数模式 | 电平 | CI0FE0 | | CI1FE1 | |
|---|---|---|---|---|---|
| | | 上升 | 下降 | 上升 | 下降 |
| | CI0FE0=0 | X | X | 向下 | 向上 |

**注意:** "-" 意思是"无计数"; "X" 意思是不可能。"0" 意思是低电平, "1" 意思是高电平。

**图 16-18. 在编码器模式 2 且 CI0FE0 极性不反相时计数器行为**

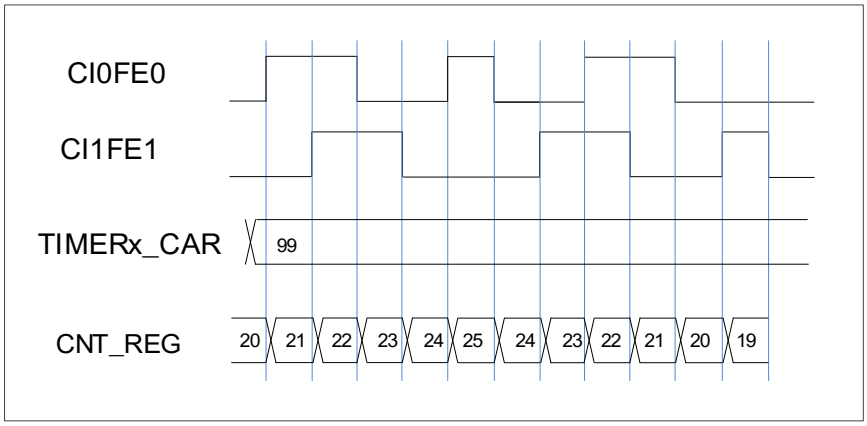
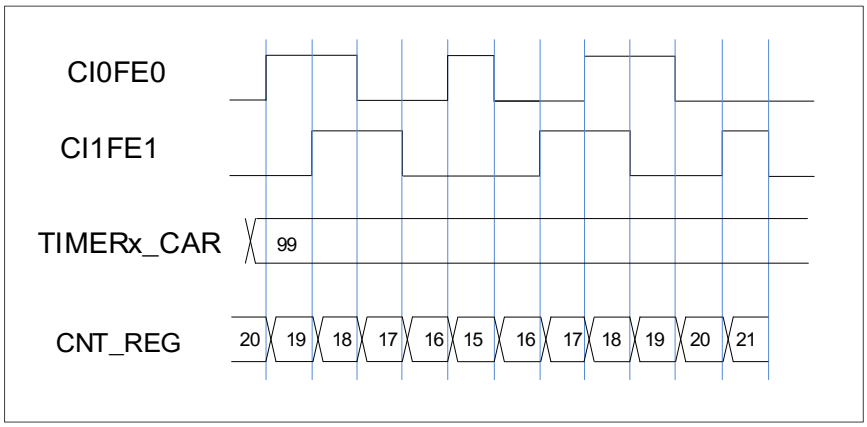



**图 16-19. 在编码器模式 2 且 CI0FE0 极性反相时计数器行为**




### 霍尔传感器接口功能

高级定时器支持霍尔传感器接口功能，该功能可以用来控制 BLDC 电机。

*图 16-20. 霍尔传感器用在 BLDC 电机控制*是定时器和电机的连接示意图。众所周知，我们要两个定时器。TIMER_in 定时器(可以是高级定时器或者通用 L0 定时器)接收霍尔传感器的三路信号。

三个霍尔传感器信号与 TIMER_in 定时器的三路输入捕获引脚一一对应连接，每个霍尔传感器输入一路波形到输入引脚，分析三路霍尔信号可以计算出转子的位置和速度。

通过定时器内部连接，例如 TRGO-ITIx，TIMER_in 定时器和 TIMER_out 定时器可以连接在一起。TIMER_out 定时器根据 ITIx 触发信号输出 PWM 波，驱动 BLDC 电机，控制 BLDC 电

机的速度。这样，TIMER_in 定时器和 TIMER_out 定时器的连接形成了一个反馈电路，可以根据需求改变配置。

TIMER_in 定时器需要具备输入异或功能，所以可以选择高级定时器和通用 L0 定时器。

TIMER_out 定时器需要具备互补输出和死区插入功能，所以可以选择高级定时器。另外，根据定时器的内部互连关系，可以选择成对的互连定时器，例如:

TIMER_in (TIMER0) -> TIMER_out (TIMER7 ITI0)

TIMER_in (TIMER1) -> TIMER_out (TIMER0 ITI1)

等等。

选择好合适的互连定时器，定时器和 BLDC 的线路也已经连接好，我们就可以配置定时器了。有以下关键配置：

- 设置TI0S，使能异或功能。三路输入信号的任何一路发生变化，CI0都会反转，CH0VAL此时会捕获计数器的当前值。
- 设置CCUC和CCSE，使能ITIx直接连接到换相功能。
- 根据需求配置PWM参数。

**图 16-20. 霍尔传感器用在 BLDC 电机控制**

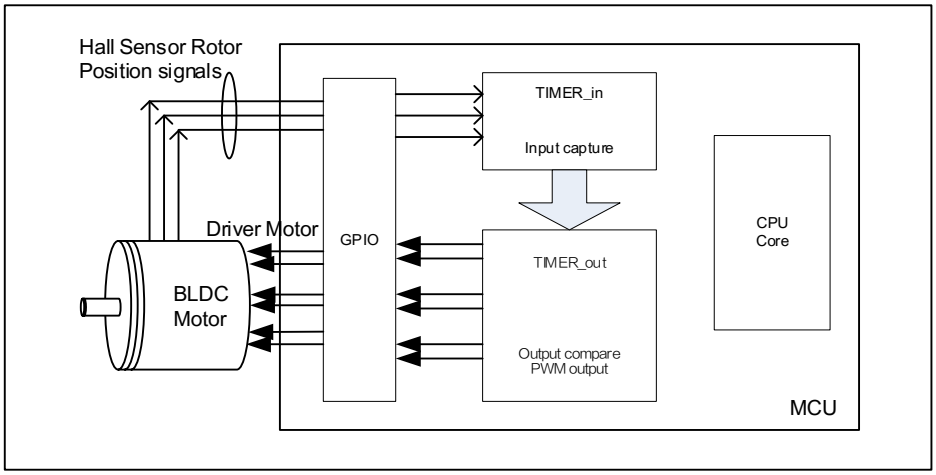
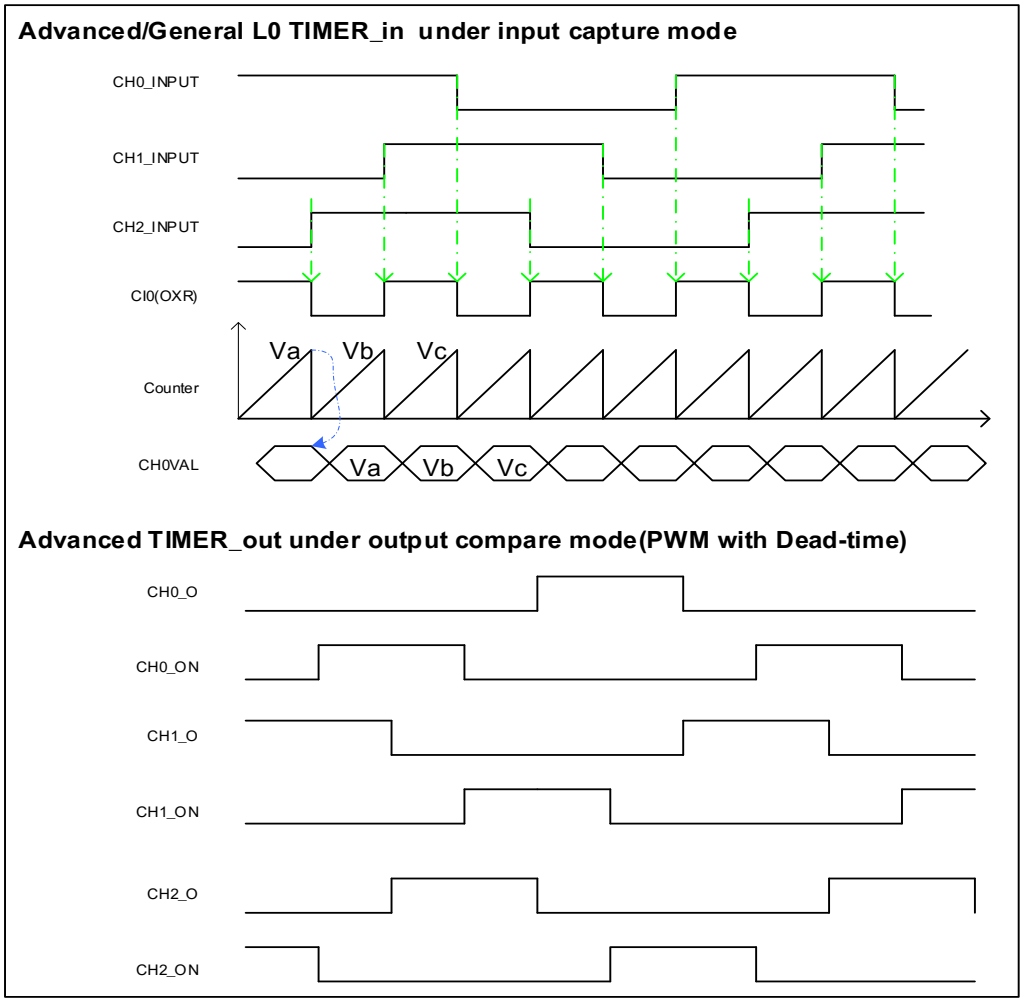


图 16-21. 两个定时器之间的霍尔传感器时序图

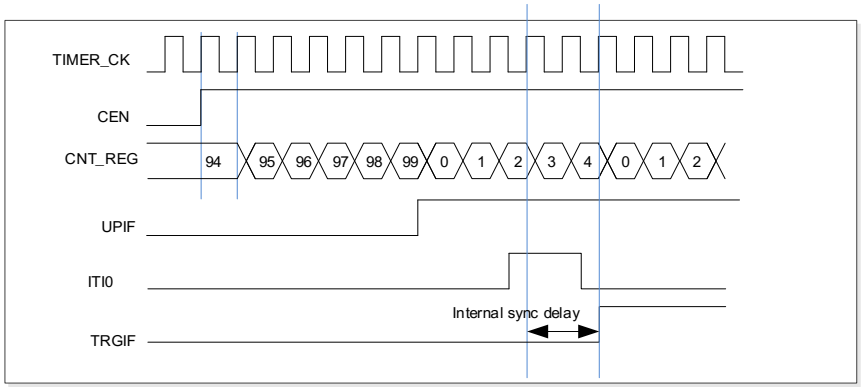
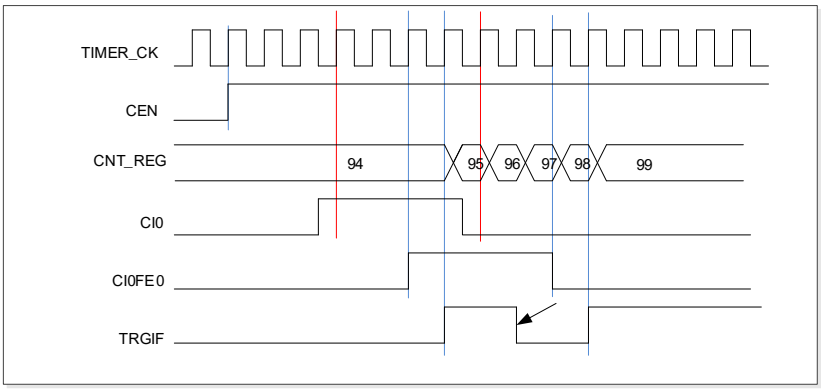



## 主-从管理

TIMERx 能在多种模式下同步外部触发，包括复位模式，暂停模式和事件模式，可以通过设置 TIMERx_SMCFG 寄存器中的 SMC [2:0]配置这些模式。这些模式的输入触发源可以通过设置 TIMERx_SMCFG 寄存器中的 TRGS [2:0]来选择。

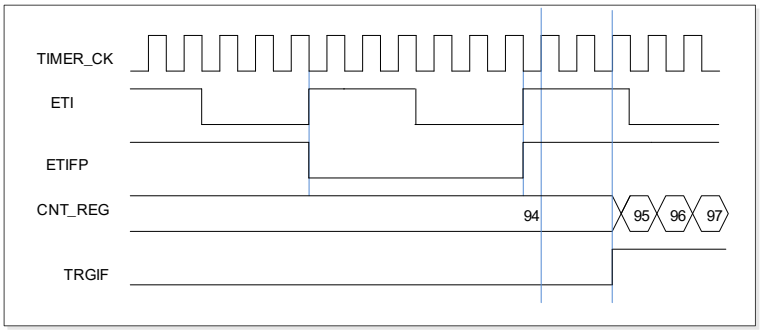
表 16-4. 从模式例子列表

| | 模式选择 | 触发源选择 | 极性选择 | 滤波和预分频 |
|---|---|---|---|---|
| 列举 | SMC[2:0]<br>3'b100 (复位模式)<br>3'b101 (暂停模式)<br>3'b110 (事件模式) | TRGS[2:0]<br>000: ITI0<br>001: ITI1<br>010: ITI2<br>011: ITI3<br>100: CI0F_ED<br>101: CI0FE0<br>110: CI1FE1 | 如果触发源是CI0FE0或者CI1FE1，配置CHxP和CHxNP来选择极性和反相<br>如果触发源是ETIF，配置ETP选择极性和反相 | 触发源ITIx，滤波和预分频不可用<br>触发源CIx，配置CHxCAPFLT设置滤波，分频不可用<br>触发源是ETIF，滤波和预 |

| | 模式选择 | 触发源选择 | 极性选择 | 滤波和预分频 |
|---|---|---|---|---|
| | | 111: ETIFP | | 分频不可用 |
| 例1 | 复位模式<br>当触发输入上升沿，计数器清零重启 | TRGIS[2:0]=3'b000<br>选择ITI0为触发源 | 触发源是ITI0，极性选择不可用 | 触发源是 ITI0，滤波和预分频不可用 |
| | **图 16-22. 复位模式下的控制电路**<br> | | | |
| 例2 | 暂停模式<br>当触发输入为低的时候，计数器暂停计数 | TRGIS[2:0]=3'b101<br>选择CI0FE0为触发源 | TI0S=0.(非异或)<br>[CH0NP==0, CH0P==0]<br>不反相.在上升沿捕获 | 在这个例子中滤波被旁路 |
| | **图 16-23. 暂停模式下的控制电路**<br> | | | |
| 例3 | 事件模式<br>触发输入的上升沿计数器开始计数 | TRGIS[2:0]=3'b111<br>选择ETIF为触发源. | ETP = 0 没有极性改变 | ETPSC = 1, 2分频.<br>ETFC = 0 , 无滤波 |

| | 模式选择 | 触发源选择 | 极性选择 | 滤波和预分频 |
|---|---|---|---|---|

图 16-24. 事件模式下的控制电路




## 单脉冲模式

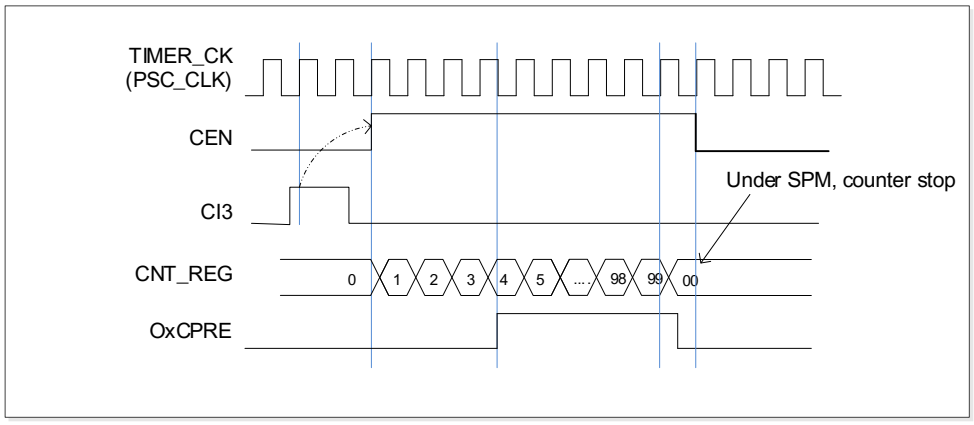
单脉冲模式与重复模式是相反的，设置 TIMERx_CTL0 寄存器的 SPM 位置 1，则使能单脉冲模式。当 SPM 置 1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置 CHxCOMCTL 配置 TIMERx 为 PWM 模式或者比较模式。

一旦设置定时器运行在单脉冲模式下，没有必要设置 TIMERx_CTL0 寄存器的定时器使能位 CEN=1 来使能计数器。触发信号沿或者软件写 CEN=1 都可以产生一个脉冲，此后 CEN 位一直保持为 1 直到更新事件发生或者 CEN 位被软件写 0。如果 CEN 位被软件清 0，计数器停止工作， 计数值被保持。

在单脉冲模式下，有效的外部触发边沿会将 CEN 位置 1，使能计数器。然而，执行计数值和 TIMERx_CHxCV 寄存器值的比较结果依然存在一些时钟延迟。为了最大限度减少延迟，用户可以将 TIMERx_CHCTL0/1 寄存器的 CHxCOMFEN 位置 1。单脉冲模式下，触发上升沿产生之后， OxCPRE 信号将被立即强制转换为与发生比较匹配时相同的电平，但是不用考虑比较结果。只有输出通道配置为 PWM1 或 PWM2 输出运行模式下时 CHxCOMFEN 位才可用，触发源来源于触发信号。

*图 16-25. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99* 展示了一个例子

图 16-25. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99

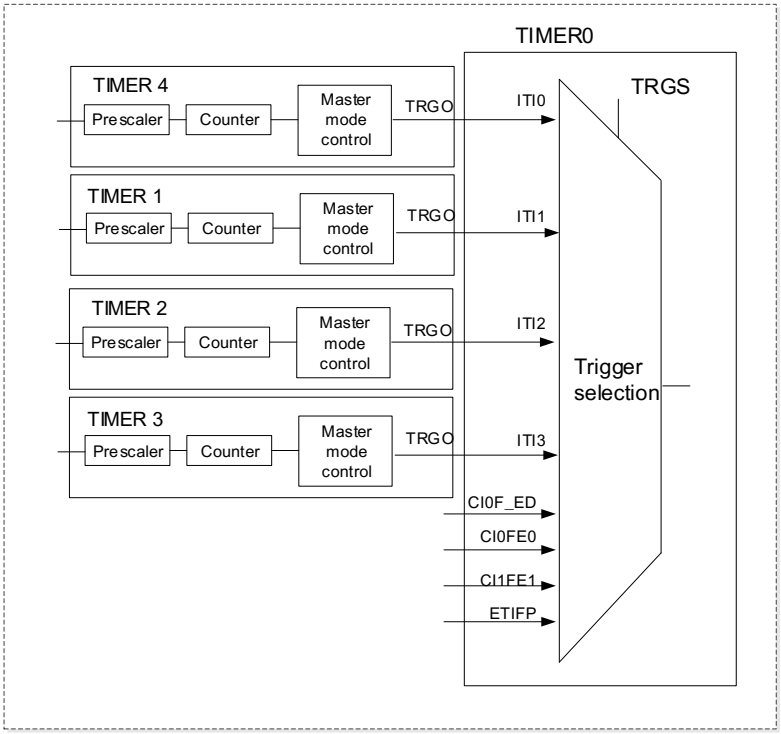


### 定时器互连

定时器之间相互连接可以定时器级联或者同步。可以通过配置一个定时器工作在主模式另一个定时器工作在从模式来实现。*图16-26. 定时器0主/从模式的例子*显示了一些主从模式触发选择的例子。

*图16-26. 定时器0主/从模式的例子*显示了当定时器0配置为从模式时的触发选择

**图 16-26. 定时器 0 主/从模式的例子**




其他定时器互连的例子：

■ 定时器2作为定时器0的预分频器

参考*图16-26. 定时器0主/从模式的例子*连接配置定时器2为定时器0的预分频器，步骤如下：

1. 配置定时器2为主模式，选择其更新事件(UPE)为触发输出(配置TIMER2_CTL1寄存器的MMC=3'b010)。定时器2在每次计数器溢出产生更新事件时，输出一个周期信号；
2. 配置定时器2周期(TIMER2_CAR寄存器)；
3. 选择定时器0输入触发源为定时器2 (配置TIMERx_SMCFG寄存器的TRGS=3'b010)；
4. 配置定时器0在外部时钟模式0(配置TIMERx_SMCFG寄存器的SMC=3'b111)；
5. 写1到CEN位启动定时器0 (TIMER0_CTL0寄存器)；
6. 写1到CEN位启动定时器2 (TIMER2_CTL0寄存器)。

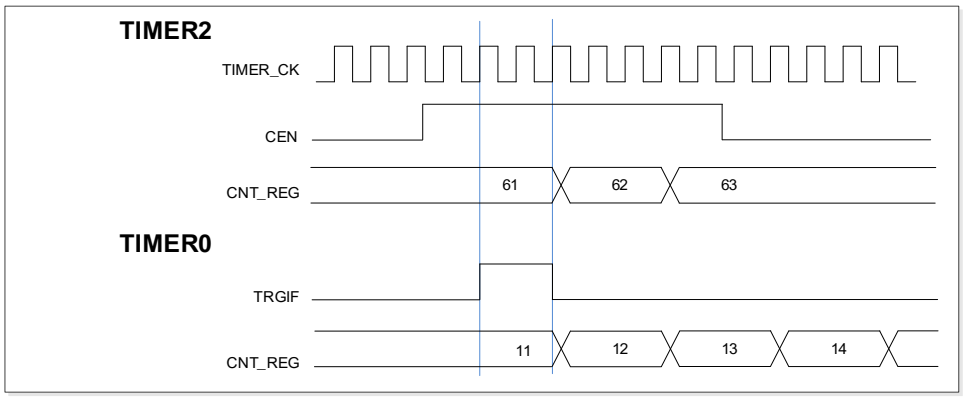
■ 用定时器2的使能/更新信号来启动定时器0

用定时器 2 的使能信号来启动定时器 0，见*图 16-27. 用定时器 2 的使能信号触发定时器 0*。在定时器 2 使能信号输出后，定时器 0 按照分频后的内部时钟从当前值开始计数。

当定时器 0 接收到触发信号，它的 CEN 位置 1，计数器计数直到禁能定时器 0。两个定时器的计数器频率都是 TIMER_CK 经过预分频器 3 分频后频率(fCNT_CLK = fTIMER_CK/3)。步骤如下：

1. 配置定时器2为主模式，发送它的使能信号作为触发输出(配置TIMER2_CTL1寄存器的MMC=3'b001)；
2. 配置定时器0选择输入触发来自定时器2 (配置TIMERx_SMCFG寄存器的TRGS=3'b010)；
3. 配置定时器0在事件模式 (配置TIMERx_SMCFG寄存器的SMC=3'b 110)；
4. 写1到CEN来开启定时器2 (TIMER2_CTL0寄存器)。

**图 16-27. 用定时器 2 的使能信号触发定时器 0**



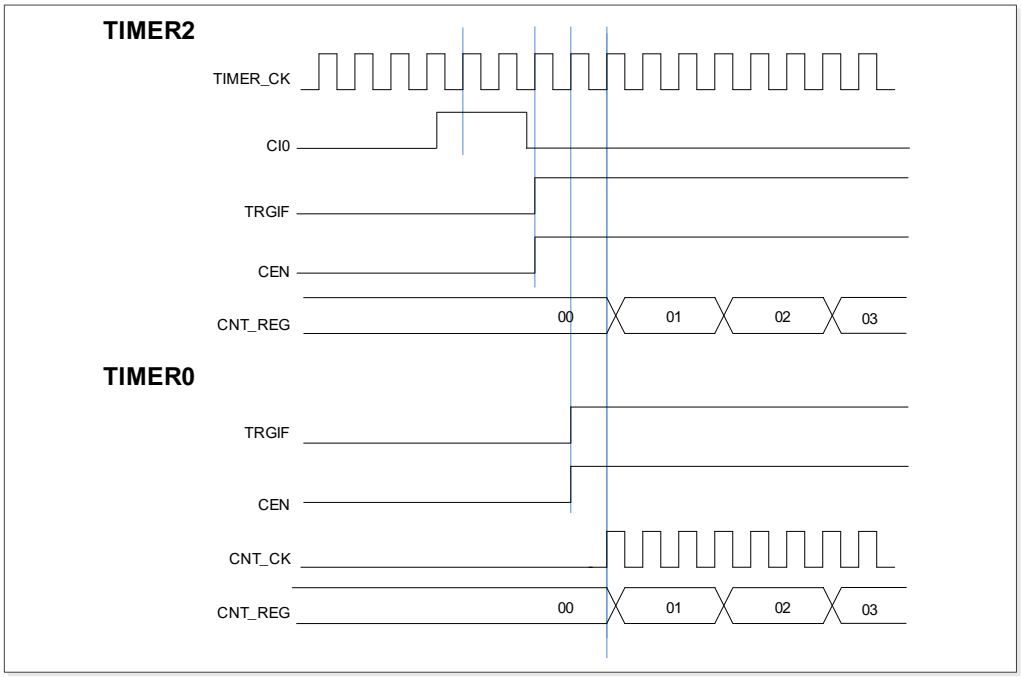

■ 使用一个外部触发来同步两个定时器

配置定时器2的使能信号触发定时器0的开启，配置定时器2的CI0输入信号上升沿来触发定时器2。为了确保两个定时器同步开启，定时器2必须配置在主/从模式。步骤如下：

1. 配置定时器2工作在从模式来获取来自CI0的触发输入(配置TIMER2_SMCFG寄存器的TRGS=3'b100)；
2. 配置定时器2工作在事件模式(配置TIMER2_SMCFG寄存器的SMC=3'b110)；
3. 写MSM=1(TIMER2_SMCFG寄存器)来配置定时器2工作在主/从模式；
4. 配置定时器0的触发输入来自定时器2 (配置TIMERx_SMCFG 寄存器的TRGS=3'b010)；
5. 配置定时器0工作在事件模式(配置TIMER0_SMCFG寄存器的SMC=3'b110)。

当定时器2的CI0信号产生上升沿时，两个定时器的计数器在内部时钟下开始同步计数，二者的TRGIF标志位都被置1。

图 16-28. 用定时器 2 的 CI0 输入来触发定时器 0 和定时器 2




## 定时器 DMA 模式

定时器 DMA 模式是指通过 DMA 模块配置定时器的寄存器。有两个跟定时器 DMA 模式相关的寄存器：TIMERx_DMACFG 和 TIMERx_DMATB。当然，必须要使能 DMA 请求，一些内部中断事件可以产生 DMA 请求。当中断事件发生，TIMERx 会给 DMA 发送请求。DMA 配置成 M2P 模式，PADDR 是 TIMERx_DMATB 寄存器地址，DMA 就会访问 TIMERx_DMATB 寄存器。实际上，TIMERx_DMATB 寄存器只是一个缓冲，定时器会将 TIMERx_DMATB 映射到一个内部寄存器，这个内部寄存器由 TIMERx_DMACFG 寄存器中的 DMATA 来指定。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值为 0，表示 1 次传输，定时器的发送 1 个 DMA 请求就可以完成。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值不为 1，例如其值为 3，表示 4 次传输，定时器就需要再多发 3 次 DMA 请求。在这 3 次请求下，DMA 对 TIMERx_DMATB 寄存器的访问会映射到访问定时器的 DMATA+0x4, DMATA+0x8, DMATA+0xc 寄存器。总之，发生一次 DMA 内部中断请求，定时器会连续发送(DMATC+1)次请求。

如果再来1次DMA请求事件，TIMERx将会重复上面的过程。

## 定时器调试模式

当Cortex®-M4内核停止，DBG_CTL0寄存器中的TIMERx_HOLD配置位被置1，定时器计数器停止。

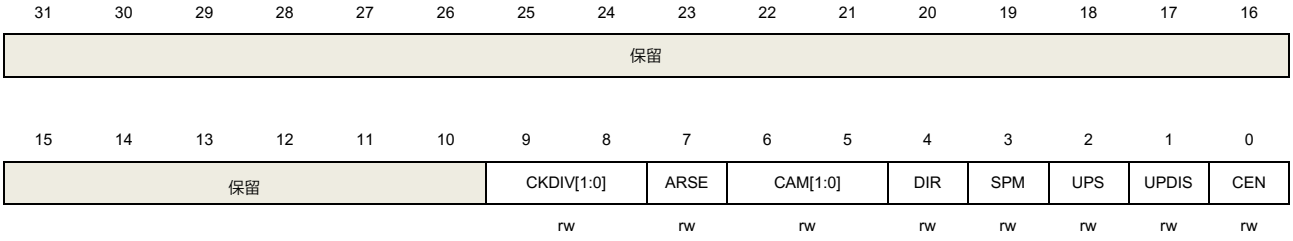
### 16.1.5. TIMERx 寄存器(x=0,7)

TIMER0基地址：0x4001 2C00
TIMER7基地址：0x4001 3400

#### 控制寄存器 0 (TIMERx_CTL0)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:10 | 保留 | 必须保持复位值 |
| 9:8 | CKDIV[1:0] | 时钟分频<br>通过软件配置CKDIV，规定定时器时钟(CK_TIMER) 与死区时间和数字滤波器采样时钟(DTS)之间的分频系数。<br>00：$f_{DTS}=f_{CK_TIMER}$<br>01：$f_{DTS}= f_{CK_TIMER} /2$<br>10：$f_{DTS}= f_{CK_TIMER} /4$<br>11：保留 |
| 7 | ARSE | 自动重载影子使能<br>0：禁能TIMERx_CAR寄存器的影子寄存器<br>1：使能TIMERx_CAR寄存器的影子寄存器 |
| 6:5 | CAM[1:0] | 计数器对齐模式选择<br>00：无中央对齐计数模式(边沿对齐模式)。 DIR位指定了计数方向<br>01：中央对齐向下计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式(TIMERx_CHCTL0寄存器中CHxMS=00)，只有在向下计数时，CHxF位置1<br>10：中央对齐向上计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式(TIMERx_CHCTL0寄存器中CHxMS=00)，只有在向上计数时，CHxF位置1<br>11：中央对齐上下计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式(TIMERx_CHCTL0寄存器中CHxMS=00)，在向上和向下计数时，CHxF位都会置1<br>当计数器使能以后，该位不能从0x00切换到非0x00 |
| 4 | DIR | 方向<br>0：向上计数 |

| | | |
|---|---|---|
| | | 1：向下计数<br>当计数器配置为中央对齐计数模式或编码器模式时，该位只读。 |
| 3 | SPM | 单脉冲模式<br>0：单脉冲模式禁能。更新事件发生后，计数器继续计数<br>1：单脉冲模式使能。在下一次更新事件发生时，计数器停止计数 |
| 2 | UPS | 更新请求源<br>软件配置该位，选择更新事件源.<br>0：以下事件均会产生更新中断或DMA请求：<br>UPG位被置1<br>计数器溢出/下溢<br>复位模式产生的更新<br>1：下列事件会产生更新中断或DMA请求：<br>计数器溢出/下溢 |
| 1 | UPDIS | 禁止更新.<br>该位用来使能或禁能更新事件的产生<br>0：更新事件使能. 更新事件发生时，相应的影子寄存器被装入预装载值，以下事件均会产生更新事件：<br>UPG位被置1<br>计数器溢出/下溢<br>复位模式产生的更新<br>1：更新事件禁能.<br>注意：当该位被置1时，UPG位被置1或者复位模式不会产生更新事件，但是计数器和预分频器被重新初始化 |
| 0 | CEN | 计数器使能<br>0：计数器禁能<br>1：计数器使能<br>在软件将CEN位置1后，外部时钟、暂停模式和编码器模式才能工作。 |

### 控制寄存器 1 (TIMERx_CTL1)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | ISO3 | ISO2N | ISO2 | ISO1N | ISO1 | ISO0N | ISO0 | TI0S | MMC[2:0] | | | DMAS | CCUC | 保留 | CCSE |
| | 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 | ISO3 | 通道3的空闲状态输出<br>参考ISO0位 |
| 13 | ISO2N | 通道2的互补通道空闲状态输出<br>参考ISO0N位 |
| 12 | ISO2 | 通道2的空闲状态输出<br>参考ISO0位 |
| 11 | ISO1N | 通道1的互补通道空闲状态输出<br>参考ISO0N位 |
| 10 | ISO1 | 通道1的空闲状态输出<br>参考ISO0位 |
| 9 | ISO0N | 通道0的互补通道空闲状态输出<br>0：当POEN位复位时，CH0_ON设置低电平<br>1：当POEN位复位时，CH0_ON设置高电平<br>此位只有在TIMERx_CCHP寄存器的PROT[1:0]位为00的时候可以被更改。 |
| 8 | ISO0 | 通道0的空闲状态输出<br>0：当POEN位复位时，CH0_O设置低电平<br>1：当POEN位复位时，CH0_O设置高电平<br>如果CH0_ON生效，一个死区时间后CH0_O输出改变。此位只有在TIMERx_CCHP寄存器的PROT[1:0]位为00的时候可以被更改。 |
| 7 | TI0S | 通道0触发输入选择<br>0：选择TIMERx_CH0引脚作为通道0的触发输入<br>1：选择TIMERx_CH0，TIMERx_CH1和TIMERx_CH2引脚异或的结果作为通道0的触发输入 |
| 6:4 | MMC[2:0] | 主模式控制<br>这些位控制TRGO信号的选择，TRGO信号由主定时器发给从定时器用于同步功能。<br>000：当产生一个定时器复位事件后，输出一个TRGO信号，定时器复位源为：<br>主定时器产生一个复位事件<br>TIMERx_SWEVG寄存器中UPG位置1<br>001：当产生一个定时器使能事件后，输出一个TRGO信号，定时器使能源为：<br>CEN位置1<br>在暂停模式下，触发输入置1<br>010：当产生一个定时器更新事件后，输出一个TRGO信号，更新事件源由UPDIS和UPS位决定<br>011：当通道0在发生一次捕获或一次比较成功时，主模式控制器产生一个TRGO脉冲<br>100：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O0CPRE<br>101：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O1CPRE<br>110：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O2CPRE<br>111：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O3CPRE |

| | | |
|---|---|---|
| 3 | DMAS | DMA请求源选择<br>0：当通道捕获/比较事件发生时，发送通道x的DMA请求 .<br>1：当更新事件发生，发送通道x的DMA请求 |
| 2 | CCUC | 换相控制影子寄存器更新控制<br>当换相控制影子寄存器(CHxEN，CHxNEN和CHxCOMCTL位)使能(CCSE=1)，这些影子寄存器更新控制如下：<br>0：CMTG位被置1时，更新影子寄存器<br>1：当CMTG位被置1或检测到TRIGI上升沿时，影子寄存器更新<br>当通道没有互补输出时，此位无效。 |
| 1 | 保留 | 必须保持复位值 |
| 0 | CCSE | 换相控制影子使能<br>0：影子寄存器(CHxEN，CHxNEN和CHxCOMCTL位)禁能<br>1：影子寄存器(CHxEN，CHxNEN和CHxCOMCTL位)使能<br>如果这些位已经被写入了，换相事件到来时这些位才被更新。<br>当通道没有互补输出时，此位无效。 |

### 从模式配置寄存器(TIMERx_SMCFG)

地址偏移：0x08
复位值： 0x0000 0000

该寄存器只能按字(32位)访问

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ETP | SMC1 | ETPSC[1:0] | | ETFC[3:0] | | | | MSM | TRGS[2:0] | | | 保留 | SMC[2:0] | | |
| 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 | ETP | 外部触发极性<br>该位指定 ETI 信号的极性<br>0：ETI 高电平或上升沿有效 .<br>1：ETI 低电平或下降沿有效 . |
| 14 | SMC1 | SMC 的一部分为了使能外部时钟模式 1<br>在外部时钟模式 1，计数器由 ETIF 信号上的任意有效边沿驱动<br>0：外部时钟模式 1 禁能<br>1：外部时钟模式 1 使能<br>当从模式配置为复位模式，暂停模式和事件模式时，定时器仍然可以工作在外部时钟模式 1。但是 TRGS 必须不能为 3'b111。 |

如果外部时钟模式 0 和外部时钟模式 1 同时被被配置，外部时钟的输入是 ETIF
注意：外部时钟模式 0 使能在寄存器的 SMC[2:0]位域。

13:12 ETPSC[1:0] 外部触发预分频

外部触发信号 ETIFP 的频率不能超过 TIMER_CK 频率的 1/4。当输入较快的外部时钟时，可以使用预分频降低 ETIFP 的频率。

00：预分频禁能

01：2 分频

10：4 分频

11：8 分频

11:8 ETFC[3:0] 外部触发滤波控制

外部触发信号可以通过数字滤波器进行滤波，该位域定义了数字滤波器的滤波能力。数字滤波器的基本原理是：以 $f_{SAMP}$ 频率连续采样外部触发信号，同时记录采样相同电平的次数。当该次数达到配置的滤波能力时，则认为是一个有效的电平信号。

| EXTFC[3:0] | 次数 | $f_{SAMP}$ |
|---|---|---|
| 4'b0000 | Filter disabled. | |
| 4'b0001 | 2 | $f_{CK_TIMER_}$ |
| 4'b0010 | 4 | |
| 4'b0011 | 8 | |
| 4'b0100 | 6 | $f_{DTS_CK}/2$ |
| 4'b0101 | 8 | |
| 4'b0110 | 6 | $f_{DTS_CK}/4$ |
| 4'b0111 | 8 | |
| 4'b1000 | 6 | $f_{DTS_CK}/8$ |
| 4'b1001 | 8 | |
| 4'b1010 | 5 | $f_{DTS_CK}/16$ |
| 4'b1011 | 6 | |
| 4'b1100 | 8 | |
| 4'b1101 | 5 | $f_{DTS_CK}/32$ |
| 4'b1110 | 6 | |
| 4'b1111 | 8 | |

7 MSM 主-从模式

该位被用来同步被选择的定时器同时开始计数。通过 TRIGI 和 TRGO，定时器被连接在一起，TRGO 用做启动事件。

0：主从模式禁能

1：主从模式使能

6:4 TRGS[2:0] 触发选择

该位域用来指定选择哪一个信号作为用来同步计数器的触发输入源

000: ITI0

001: ITI1

010: ITI2

011: ITI3

100: CI0F_ED

| | | |
|---|---|---|
| | | 101: CI0FE0<br>110: CI1FE1<br>111: ETIFP<br>从模式被使能后这些位不能改 |
| 3 | 保留 | 必须保持复位值 |
| 2:0 | SMC[2:0] | 从模式控制<br>000：关闭从模式. 如果 CEN=1，则预分频器直接由内部时钟驱动<br>001：编码器模式 0. 根据 CI1FE1 的电平，计数器在 CI0FE0 的边沿向上/下计数<br>010：编码器模式 1. 根据 CI0FE0 的电平，计数器在 CI1FE1 的边沿向上/下计数<br>011：编码器模式 2. 根据另一个信号的输入电平，计数器在 CI0FE0 和 CI1FE1 的边沿向上/ 下计数<br>100：复位模式. 选中的触发输入的上升沿重新初始化计数器，并且产生更新事件.<br>101：暂停模式. 当触发输入为高时，计数器的时钟开启。一旦触发输入变为低，则计数器时钟停止<br>110：事件模式.计数器在触发输入的上升沿启动。<br>111：外部时钟模式 0. 选中的触发输入的上升沿驱动计数器 |

### DMA 和中断使能寄存器 (TIMERx_DMAINTEN)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | TRGDEN | CMTDEN | CH3DEN | CH2DEN | CH1DEN | CH0DEN | UPDEN | BRKIE | TRGIE | CMTIE | CH3IE | CH2IE | CH1IE | CH0IE | UPIE |
| | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:15 | 保留 | 必须保持复位值 |
| 14 | TRGDEN | 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 | CMTDEN | 换相DMA更新请求使能<br>0：禁止换相DMA更新请求<br>1：使能换相DMA更新请求 |
| 12 | CH3DEN | 通道3比较/捕获DMA请求使能<br>0：禁止通道3比较/捕获DMA请求<br>1：使能通道3比较/捕获DMA请求 |

| | | |
|---|---|---|
| 11 | CH2DEN | 通道2比较/捕获DMA请求使能<br>0：禁止通道2比较/捕获DMA请求<br>1：使能通道2比较/捕获DMA请求 |
| 10 | CH1DEN | 通道1比较/捕获DMA请求使能<br>0：禁止通道1比较/捕获DMA请求<br>1：使能通道1比较/捕获DMA请求 |
| 9 | CH0DEN | 通道0比较/捕获DMA请求使能<br>0：禁止通道0比较/捕获DMA请求<br>1：使能通道0比较/捕获DMA请求 |
| 8 | UPDEN | 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 | BRKIE | 中止中断使能<br>0：禁止中止中断<br>1：使能中止中断 |
| 6 | TRGIE | 触发中断使能<br>0：禁止触发中断<br>1：使能触发中断 |
| 5 | CMTIE | 换相更新中断使能<br>0：禁止换相更新中断<br>1：使能换相更新中断 |
| 4 | CH3IE | 通道3比较/捕获中断使能<br>0：禁止通道3中断<br>1：使能通道3中断 |
| 3 | CH2IE | 通道2比较/捕获中断使能<br>0：禁止通道2中断<br>1：使能通道2中断 |
| 2 | CH1IE | 通道1比较/捕获中断使能<br>0：禁止通道1中断<br>1：使能通道1中断 |
| 1 | CH0IE | 通道0比较/捕获中断使能<br>0：禁止通道0中断<br>1：使能通道0中断 |
| 0 | UPIE | 更新中断使能<br>0：禁止更新中断<br>1：使能更新中断 |

### 中断标志寄存器（TIMERx_INTF)

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | CH3OF | CH2OF | CH1OF | CH0OF | 保留 | BRKIF | TRGIF | CMTIF | CH3IF | CH2IF | CH1IF | CH0IF | UPIF |
| | | | rc_w0 | rc_w0 | rc_w0 | rc_w0 | | rc_w0 | rc_w0 | rc_w0 | rc_w0 | rc_w0 | rc_w0 | rc_w0 | rc_w0 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:13 | 保留 | 必须保持复位值 |
| 12 | CH3OF | 通道3捕获溢出标志<br>参见CH0OF描述 |
| 11 | CH2OF | 通道2捕获溢出标志<br>参见CH0OF描述 |
| 10 | CH1OF | 通道1捕获溢出标志<br>参见CH0OF描述 |
| 9 | CH0OF | 通道0捕获溢出标志<br>当通道0被配置为输入模式时，在CH0IF标志位已经被置1后，捕获事件再次发生时，该标志位可以由硬件置1。该标志位由软件清0。<br>0：无捕获溢出中断发生<br>1：发生了捕获溢出中断 |
| 8 | 保留 | 必须保持复位值 |
| 7 | BRKIF | 中止中断标志位<br>当中止输入有效时，由硬件对该位置'1'。<br>当中止输入无效时，则该位可由软件清'0'。<br>0：无中止事件产生<br>1：中止输入上检测到有效电平 |
| 6 | TRGIF | 触发中断标志<br>当发生触发事件时，此标志会置 1，此位由软件清 0。当暂停模式使能时，触发输入的任意边沿都可以产生触发事件。否则，其它模式时，仅在触发输入端检测到有效边沿，产生触发事件。<br>0：无触发事件产生<br>1：触发中断产生 |
| 5 | CMTIF | 通道换相更新中断标志<br>当通道换相更新事件发生时，此标志位被硬件置1，此位由软件清0。 |

0：无通道换相更新中断发生
1：通道换相更新中断发生

| | | |
|---|---|---|
| 4 | CH3IF | 通道3比较/捕获中断标志<br>参见CH0IF描述 |
| 3 | CH2IF | 通道2比较/捕获中断标志<br>参见CH0IF描述 |
| 2 | CH1IF | 通道1比较/捕获中断标志<br>参见CH0IF描述 |
| 1 | CH0IF | 通道0比较/捕获中断标志<br>此标志由硬件置1，软件清0。<br>当通道0在输入模式下时，捕获事件发生时此标志位被置1；当通道0在输出模式下时，此标志位在一个比较事件发生时被置1。<br>0：无通道0中断发生<br>1：通道0中断发生 |
| 0 | UPIF | 更新中断标志<br>此位在更新事件发生时由硬件置1，软件清0。<br>0：无更新中断发生<br>1：发生更新中断 |

### 软件事件产生寄存器 (TIMERx_SWEVG)

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | BRKG | TRGG | CMTG | CH3G | CH2G | CH1G | CH0G | UPG |
| | | | | | | | | w | w | w | w | w | w | w | w |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 必须保持复位值 |
| 7 | BRKG | 产生中止事件<br>该位由软件置1，用于产生一个中止事件，由硬件自动清0。当此位被置1时，POEN位被清0且BRKIF位被置1，若开启对应的中断和DMA，则产生相应的中断和DMA传输。<br>0：不产生中止事件<br>1：产生中止事件 |

| | | |
|---|---|---|
| 6 | TRGG | 触发事件产生<br>此位由软件置1，由硬件自动清0。当此位被置1，TIMERx_INTF寄存器的TRGIF标志位被置1，若开启对应的中断和DMA，则产生相应的中断和DMA传输。<br>0：无触发事件产生<br>1：产生触发事件 |
| 5 | CMTG | 通道换相更新事件发生<br>此位由软件置1，由硬件自动清0。当此位被置1，根据CCSE位(TIMERx_CTL1寄存器中)的值，通道捕获/比较控制寄存器（CHxEN，CHxNEN 和CHxCOMCTL）的互补输出被更新。<br>0：不产生通道换相更新事件<br>1：产生通道换相更新事件 |
| 4 | CH3G | 通道3捕获或比较事件发生<br>参见CH0G描述 |
| 3 | CH2G | 通道2捕获或比较事件发生<br>参见CH0G描述 |
| 2 | CH1G | 通道1捕获或比较事件发生<br>参见CH0G描述 |
| 1 | CH0G | 通道0捕获或比较事件发生<br>该位由软件置1，用于在通道0产生一个捕获/比较事件，由硬件自动清0。当此位被置1，CH0IF标志位被置1，若开启对应的中断和DMA，则发出相应的中断和DMA请求。此外，如果通道0配置为输入模式，计数器的当前值被捕获到TIMERx_CH0CV寄存器，如果CH0IF标志位已经为1，则CH0OF标志位被置1。<br>0：不产生通道0捕获或比较事件<br>1：发生通道0捕获或比较事件 |
| 0 | UPG | 更新事件产生<br>此位由软件置1，被硬件自动清0。当此位被置1，如果选择了中央对齐或向上计数模式，计数器被清0。否则(向下计数模式)计数器将载入自动重载值，预分频计数器将同时被清除。<br>0：无更新事件产生<br>1：产生更新事件 |

### 通道控制寄存器 0 (TIMERx_CHCTL0)

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

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

| CH1COMCEN | CH1COMCTL[2:0] | CH1COMSEN | CH1COMFEN | CH1MS[1:0] | CH0COMCEN | CH0COMCTL[2:0] | CH0COMSEN | CH0COMFEN | CH0MS[1:0] |
|---|---|---|---|---|---|---|---|---|---|
| CH1CAPFLT[3:0] | | CH1CAPPSC[1:0] | | | CH0CAPFLT[3:0] | | CH0CAPPSC[1:0] | | |
| rw | | rw | | rw | rw | | rw | | rw |

**输出比较模式：**

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15 | CH1COMCEN | 通道1输出比较清0使能<br>参见CH0COMCEN描述 |
| 14:12 | CH1COMCTL[2:0] | 通道1输出比较模式<br>参见CH0COMCTL描述 |
| 11 | CH1COMSEN | 通道1输出比较影子寄存器使能<br>参见CH0COMSEN描述 |
| 10 | CH1COMFEN | 通道1输出比较快速使能<br>参见CH0COMFEN描述 |
| 9:8 | CH1MS[1:0] | 通道 1 模式选择<br>这些位定义了通道的方向和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH1EN 位被清 0)时这些位才可以写。<br>00：通道 1 配置为输出<br>01：通道 1 配置为输入，IS1 映射在 CI1FE1 上<br>10：通道 1 配置为输入，IS1 映射在 CI0FE1 上<br>11：通道 1 配置为输入，IS1 映射在 ITS 上<br>注意：当CH1MS[1:0]=11时，需要通过TRGS位(位于TIMERx_SMCFG寄存器)选择内部触发输入。 |
| 7 | CH0COMCEN | 通道 0 输出比较清 0 使能<br>当此位被置 1，当检测到 ETIFP 信号输入高电平时，O0CPRE 参考信号被清 0<br>0：禁止通道 0 输出比较清零<br>1：使能通道0输出比较清零 |
| 6:4 | CH0COMCTL[2:0] | 通道 0 输出比较模式<br>此位定义了输出准备信号 O0CPRE 的输出比较模式，而 O0CPRE 决定了 CH0_O、CH0_ON 的值。另外，O0CPRE 高电平有效，而 CH0_O、CH0_ON 通道的极性取决于 CH0P、CH0NP 位。<br>000：时基。输出比较寄存器 TIMERx_CH0CV 与计数器 TIMERx_CNT 间的比较对 O0CPRE 不起作用<br>001：匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为高。<br>010：匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为低。<br>011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 翻转。 |

100：强制为低。强制 O0CPRE 为低电平
101：强制为高。强制 O0CPRE 为高电平
110：PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。
111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。
如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，O0CPRE 电平才改变。
当TIMERx_CCHP寄存器的PROT [1:0]=11且CH0MS =00(比较模式)时此位不能被改变。

| | | |
|---|---|---|
| 3 | CH0COMSEN | 通道 0 输出比较影子寄存器使能<br>当此位被置 1，TIMERx_CH0CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。<br>0：禁止通道 0 输出/比较影子寄存器<br>1：使能通道 0 输出/比较影子寄存器<br>仅在单脉冲模式下(SPM =1)，可以在未确认影子寄存器的情况下使用 PWM 模式<br>当TIMERx_CCHP寄存器的PROT [1:0]=11且CH0MS =00时此位不能被改变。 |
| 2 | CH0COMFEN | 通道 0 输出比较快速使能<br>当该位为 1 时，如果通道配置为 PWM0 模式或者 PWM1 模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH0_O 被设置为比较电平而与比较结果无关。<br>0：禁止通道 0 输出比较快速.<br>1：使能通道0输出比较快速。 |
| 1:0 | CH0MS[1:0] | 通道 0 I/O 模式选择<br>这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH0EN 位被清 0)时这些位才可写。<br>00：通道 0 配置为输出<br>01：通道 0 配置为输入，IS0 映射在 CI0FE0 上<br>10：通道 0 配置为输入，IS0 映射在 CI1FE0 上<br>11：通道 0 配置为输入，IS0 映射在 ITS 上<br>注意：当CH0MS[1:0]=11时，需要通过TRGS位(位于TIMERx_SMCFG寄存器)选择内部触发输入 |

**输入捕获模式：**

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:12 | CH1CAPFLT[3:0] | 通道1输入捕获滤波控制<br>参见CH0CAPFLT描述 |
| 11:10 | CH1CAPPSC[1:0] | 通道1输入捕获预分频器 |

参见CH0CAPPSC描述

9:8 CH1MS[1:0] 通道1模式选择
与输出模式相同

7:4 CH0CAPFLT[3:0] 通道 0 输入捕获滤波控制
CI0 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。
数字滤波器的基本原理：根据 $f_{SAMP}$ 对 CI0 输入信号进行连续采样，并记录信号相同电平的次数。达到该位配置的滤波参数后，认为是有效电平。
滤波器参数配置如下：

| CH0CAPFLT [3:0] | 采样次数 | $f_{SAMP}$ |
|---|---|---|
| 4'b0000 | 无滤波器 | |
| 4'b0001 | 2 | $f_{CK_TIMER}$ |
| 4'b0010 | 4 | |
| 4'b0011 | 8 | |
| 4'b0100 | 6 | $f_{DTS}/2$ |
| 4'b0101 | 8 | |
| 4'b0110 | 6 | $f_{DTS}/4$ |
| 4'b0111 | 8 | |
| 4'b1000 | 6 | $f_{DTS}/8$ |
| 4'b1001 | 8 | |
| 4'b1010 | 5 | $f_{DTS}/16$ |
| 4'b1011 | 6 | |
| 4'b1100 | 8 | |
| 4'b1101 | 5 | $f_{DTS}/32$ |
| 4'b1110 | 6 | |
| 4'b1111 | 8 | |

3:2 CH0CAPPSC[1:0] 通道0输入捕获预分频器
这2位定义了通道0输入的预分频系数。当TIMERx_CHCTL2寄存器中的CH0EN=0时，则预分频器复位。
00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获。
01：每2个事件触发一次捕获。
10：每4个事件触发一次捕获。
11：每8个事件触发一次捕获。

1:0 CH0MS[1:0] 通道0模式选择
与输出比较模式相同

### 通道控制寄存器 1 (TIMERx_CHCTL1)

地址偏移：0x1C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 保留 |
|---|

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CH3COMCEN | CH3COMCTL[2:0] | | | CH3COMSEN | CH3COMFEN | CH3MS[1:0] | | CH2COMCEN | CH2COMCTL[2:0] | | | CH2COMSEN | CH2COMFEN | CH2MS[1:0] | |
| CH3CAPFLT[3:0] | | | | CH3CAPPSC[1:0] | | | | CH2CAPFLT[3:0] | | | | CH2CAPPSC[1:0] | | | |
| rw | | | | rw | | rw | | rw | | | | rw | | rw | |

**输出比较模式：**

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15 | CH3COMCEN | 通道3输出比较清0使能<br>参见CH0COMCEN描述 |
| 14:12 | CH3COMCTL[2:0] | 通道3输出比较模式<br>参见CH0COMCTL描述 |
| 11 | CH3COMSEN | 通道3输出比较影子寄存器使能<br>参见CH0COMSEN描述 |
| 10 | CH3COMFEN | 通道3输出比较快速使能<br>参见CH0COMFEN描述 |
| 9:8 | CH3MS[1:0] | 通道 3 模式选择<br>这些位定义了通道的方向和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH3EN 位被清 0)时这些位才可以写。<br>00：通道 3 配置为输出<br>01：通道 3 配置为输入，IS3 映射在 CI3FE3 上<br>10：通道 3 配置为输入，IS3 映射在 CI2FE3 上<br>11：通道 3 配置为输入，IS3 映射在 ITS 上<br>注意：当CH3MS[1:0]=11时，需要通过TRGS位(位于TIMERx_SMCFG寄存器)选择内部触发输入 |
| 7 | CH2COMCEN | 通道 2 输出比较清 0 使能<br>当此位被置 1，当检测到 ETIFP 输入高电平时，O2CPRE 参考信号被清 0<br>0：使能通道 2 输出比较清零<br>1：禁止通道2输出比较清零 |
| 6:4 | CH2COMCTL[2:0] | 通道 2 输出比较模式<br>此位定义了输出准备信号 O2CPRE 的输出比较模式，而 O2CPRE 决定了 CH2_O、CH2_ON 的值。另外，O2CPRE 高电平有效，而 CH2_O、CH2_ON 通道的极性取决于 CH2P、CH2NP 位。<br>000：时基。输出比较寄存器 TIMERx_CH2CV 与计数器 TIMERx_CNT 间的比较对 O2CPRE 不起作用<br>001：匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH2CV 相同时，强制 O2CPRE 为高。<br>010：匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH2CV 相同 |

| | | |
|---|---|---|
| | | 时，强制 O2CPRE 为低。<br>011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH2CV 相同时，强制 O2CPRE 翻转。<br>100：强制为低。强制 O2CPRE 为低电平<br>101：强制为高。强制 O2CPRE 为高电平<br>110：PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_CH2CV 时，O2CPRE 为高电平，否则为低电平。在向下计数时，一旦计数器的值大于 TIMERx_CH2CV 时，O2CPRE 为低电平，否则为高电平。<br>111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH2CV 时，O2CPRE 为低电平，否则为高电平。在向下计数时，一旦计数器的值大于 TIMERx_CH2CV 时，O2CPRE 为高电平，否则为低电平。<br>如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，O2CPRE 电平才改变。<br>当TIMERx_CCHP寄存器的PROT [1:0]=11且CH2MS =00(比较模式)时此位不能被改变。 |
| 3 | CH2COMSEN | 通道 0 输出比较影子寄存器使能<br>当此位被置 1，TIMERx_CH2CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。<br>0：禁止通道 2 输出/比较影子寄存器<br>1：使能通道 2 输出/比较影子寄存器<br>仅在单脉冲模式下(SPM =1)，可以在未确认影子寄存器情况下使用 PWM 模式<br>当TIMERx_CCHP寄存器的PROT [1:0]=11且CH2MS =00时此位不能被改变。 |
| 2 | CH2COMFEN | 通道 2 输出比较快速使能<br>当该位为 1 时，如果通道配置为 PWM0 模式或者 PWM1 模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH2_O 被设置为比较电平而与比较结果无关。<br>0：禁止通道 2 输出比较快速.<br>1：使能通道2输出比较快速。 |
| 1:0 | CH2MS[1:0] | 通道 2 I/O 模式选择<br>这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH2EN 位被清 0)时这些位才可写。<br>00：通道 2 配置为输出<br>01：通道 2 配置为输入，IS2 映射在 CI2FE2 上<br>10：通道 2 配置为输入，IS2 映射在 CI3FE2 上<br>11：通道 2 配置为输入，IS2 映射在 ITS 上.<br>注意：当CH2MS[1:0]=11时，需要通过TRGS位(位于TIMERx_SMCFG寄存器)选择内部触发输入 |

**输入捕获模式：**

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:12 | CH3CAPFLT[3:0] | 通道3输入捕获滤波控制 |

参见CH0CAPFLT描述

11:10 CH3CAPPSC[1:0] 通道3输入捕获预分频器
参见CH0CAPPSC描述

9:8 CH3MS[1:0] 通道3模式选择
与输出模式相同

7:4 CH2CAPFLT[3:0] 通道 2 输入捕获滤波控制
CI2 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。
数字滤波器的基本原理：根据 $f_{SAMP}$ 对 CI2 输入信号进行连续采样，并记录信号相同电平的次数。达到该位配置的滤波参数后，认为是有效电平。
滤波器参数配置如下：

| CH2CAPFLT [3:0] | 采样次数 | $f_{SAMP}$ |
|---|---|---|
| 4'b0000 | 无滤波器 | |
| 4'b0001 | 2 | $f_{CK_TIMER}$ |
| 4'b0010 | 4 | |
| 4'b0011 | 8 | |
| 4'b0100 | 6 | $f_{DTS}/2$ |
| 4'b0101 | 8 | |
| 4'b0110 | 6 | $f_{DTS}/4$ |
| 4'b0111 | 8 | |
| 4'b1000 | 6 | $f_{DTS}/8$ |
| 4'b1001 | 8 | |
| 4'b1010 | 5 | $f_{DTS}/16$ |
| 4'b1011 | 6 | |
| 4'b1100 | 8 | |
| 4'b1101 | 5 | $f_{DTS}/32$ |
| 4'b1110 | 6 | |
| 4'b1111 | 8 | |

3:2 CH2CAPPSC[1:0] 通道2输入捕获预分频器
这2位定义了通道2输入的预分频系数。当TIMERx_CHCTL2寄存器中的CH2EN =0时，则预分频器复位。
00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获。
01：每2个事件触发一次捕获。
10：每4个事件触发一次捕获。
11：每8个事件触发一次捕获。

1:0 CH2MS[1:0] 通道2模式选择
与输出比较模式相同

### 通道控制寄存器 2 (TIMERx_CHCTL2)

地址偏移：0x20
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | CH3P | CH3EN | CH2NP | CH2NEN | CH2P | CH2EN | CH1NP | CH1NEN | CH1P | CH1EN | CH0NP | CH0NEN | CH0P | CH0EN |
| | | rw | rw | rw | rw | rw | rw | 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 | CH3P | 通道3极性<br>参考CH0P描述 |
| 12 | CH3EN | 通道3使能<br>参考CH0EN描述 |
| 11 | CH2NP | 通道2互补输出极性<br>参考CH0NP描述 |
| 10 | CH2NEN | 通道2互补输出使能<br>参考CH0NEN描述 |
| 9 | CH2P | 通道2极性<br>参考CH0P描述 |
| 8 | CH2EN | 通道2使能<br>参考CH0EN描述 |
| 7 | CH1NP | 通道1互补输出极性<br>参考CH0NP描述 |
| 6 | CH1NEN | 通道1互补输出使能<br>参考CH0NEN描述 |
| 5 | CH1P | 通道1极性<br>参考CH0P描述 |
| 4 | CH1EN | 通道1使能<br>参考CH0EN描述 |
| 3 | CH0NP | 通道0互补输出极性<br>当通道0配置为输出模式，此位定义了互补输出信号的极性。<br>0：通道0互补输出高电平为有效电平<br>1：通道0互补输出低电平为有效电平<br>当通道0配置为输入模式时，此位和CH0P联合使用，作为输入信号CI0的极性选择控制信号。<br>当TIMERx_CCHP寄存器的PROT[1:0]=11或10时此位不能被更改。 |
| 2 | CH0NEN | 通道0互补输出使能 |

| | | |
|---|---|---|
| | | 当通道0配置为输出模式时，将此位置1使能通道0的互补输出。<br>0：禁止通道0互补输出<br>1：使能通道0互补输出 |
| 1 | CH0P | 通道0极性<br>当通道0配置为输出模式时，此位定义了输出信号极性。<br>0：通道0高电平为有效电平<br>1：通道0低电平为有效电平<br>当通道0配置为输入模式时，此位定义了CI0信号极性。<br>CH0P将选择CI0FE0或者CI1FE0的有效边沿或者捕获极性。<br>CH0P=0：把CIxFE0的上升沿作为捕获或者从模式下触发的有效信号，并且CIxFE0不会被翻转。<br>CH0P=1：把CIxFE0的下降沿作为捕获或者从模式下触发的有效信号，并且CIxFE0会被翻转。<br>当TIMERx_CCHP寄存器的PROT[1:0]=11或10 时此位不能被更改。 |
| 0 | CH0EN | 通道0捕获/比较使能<br>当通道0配置为输出模式时，将此位置1使能CH0_O信号有效。当通道0配置为输入模式时，将此位置1使能通道0上的捕获事件。<br>0：禁止通道0<br>1：使能通道0 |

### 计数器寄存器 (TIMERx_CNT)

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | CNT[15:0] | 这些位是当前的计数值。写操作能改变计数器值。 |

### 预分频寄存器 (TIMERx_PSC)

地址偏移：0x28
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | PSC[15:0] | 计数器时钟预分频值<br>计数器时钟等于TIMER_CK时钟除以(PSC+1)，每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。 |

### 计数器自动重载寄存器 (TIMERx_CAR)

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | CARL[15:0] | 计数器自动重载值<br>这些位定义了计数器的自动重载值。 |

### 重复计数寄存器 (TIMERx_CREP)

地址偏移：0x30
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 必须保持复位值 |
| 7:0 | CREP[7:0] | 重复计数器的值<br>这些位定义了更新事件的产生速率。重复计数器计数值减为0时产生更新事件。影子寄存器的更新速率也会受这些位影响(前提是影子寄存器被使能)。 |

### 通道 0 捕获/比较寄存器 (TIMERx_CH0CV)

地址偏移：0x34
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CH0VAL[15:0] | | | | | | | | | | | | | | | |
| rw | | | | | | | | | | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | CH0VAL[15:0] | 通道0的捕获或比较值<br>当通道0配置为输入模式时，这些位决定了上次捕获事件的计数器值，并且本寄存器为只读。<br>当通道0配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。 |

### 通道 1 捕获/比较寄存器 (TIMERx_CH1CV)

地址偏移：0x38
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CH1VAL[15:0] | | | | | | | | | | | | | | | |

rw

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | CH1VAL[15:0] | 通道1的捕获或比较值<br>当通道1配置为输入模式时，这些位决定了上次捕获事件的计数器值，并且本寄存器为只读。<br>当通道1配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。 |

### 通道 2 捕获/比较寄存器 (TIMERx_CH2CV)

地址偏移：0x3C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CH2VAL[15:0] | | | | | | | | | | | | | | | |

rw

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | CH2VAL[15:0] | 通道2的捕获或比较值<br>当通道2配置为输入模式时，这些位决定了上次捕获事件的计数器值，并且本寄存器为只读。<br>当通道2配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。 |

### 通道 3 捕获/比较寄存器 (TIMERx_CH3CV)

地址偏移：0x40
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CH3VAL[15:0] | | | | | | | | | | | | | | | |

rw

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | CH3VAL[15:0] | 通道3的捕获或比较值<br>当通道3配置为输入模式时，这些位决定了上次捕获事件的计数器值，并且本寄存器为只读。<br>当通道3配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。 |

### 互补通道保护寄存器 (TIMERx_CCHP)

地址偏移：0x44
复位值：0x0000

该寄存器可以按半字(16位)或字(32位)访问

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| POEN | OAEN | BRKP | BRKEN | ROS | IOS | PROT[1:0] | | DTCFG[7:0] | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | | rw | | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 15 | POEN | 所有的通道输出使能<br>该位通过以下方式置 1：<br>-写 1 置位<br>-如果 OAEN=1，则在下一次更新事件发生时置 1.<br>该位通过以下方式清 0：<br>-写 0 清 0<br>-有效的中止输入(异步)<br>如果一个通道配置为输出模式，如果设置了相应的使能位(TIMERx_CHCTL2 寄存器的 CHxEN, CHxNEN 位)，则开启 CHx_O 和 CHx_ON 输出。<br>0：禁止通道输出<br>1：使能通道输出<br>注意：仅当 CHxMS[1:0]=2'b00 时该位有效 |
| 14 | OAEN | 自动输出使能<br>0：POEN 位只能使用软件方式置 1。<br>1：如果中止输入无效，下一次更新事件发生时，POEN 位能被置 1<br>此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。 |
| 13 | BRKP | 中止极性<br>此位定义了中止输入信号 BKIN 的极性。<br>0：中止输入低电平有效<br>1：中止输入高电平有效 |
| 12 | BRKEN | 中止使能 |

此位置 1 使能中止事件和 CCS 时钟失败事件输入。
0：禁能中止输入
1：使能中止输入
此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。.

11 ROS

运行模式下"关闭状态"使能
当 POEN 位被置 1(运行模式)，此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出"关闭状态"。参见*表 16-2. 由参数控制的互补输出表*。
0：输出"关闭状态"禁能。当 CHxEN 或者 CHxNEN 位被清零，对应通道为输出"禁能状态"。
1：输出"关闭状态"使能。当 CHxEN 或者 CHxNEN 位被清零，对应通道为输出"关闭状态"。
此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。

10 IOS

空闲模式下"关闭状态"使能
当 POEN 位被清 0(空闲模式)，此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出"关闭状态"。参见*表 16-2. 由参数控制的互补输出表*。
0：输出"关闭状态"禁能。当 CHxEN 和 CHxNEN 位均被清零，对应通道为输出"禁能状态"。
1：输出"关闭状态"使能。不论 CHxEN 和 CHxNEN 位的值，对应通道为输出"关闭状态"。
此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。

9:8 PROT[1:0]

互补寄存器保护控制
这两位定义了寄存器的写保护特性。
00：禁能保护模式。无写保护.
01：PROT 模式 0。TIMERx_CTL1 寄存器中 ISOx/ISOxN 位，TIMERx_CCHP 寄存器中 BRKEN/BRKP/OAEN/DTCFG 位写保护
10：PROT 模式 1。除了 PROT 模式 0 下的寄存器写保护外，还有 TIMERx_CHCTL2 寄存器中 CHxP/CHxNP 位(如果相应通道配置为输出模式)，TIMERx_CCHP 寄存器中 ROS/IOS 位。
11：PROT 模式 2.。除了 PROT 模式 1 下的寄存器写保护外，还有 TIMERx_CHCTLR0/1 中 CHxCOMCTL/ CHxCOMSEN 位(如果相关通道配置为输出模式)写保护。
系统复位后这两位只能被写一次，一旦 TIMERx_CCHP 寄存器被写入，这两位被写保护

7:0 DTCFG[7:0]

死区时间控制
DTCFG 值和死区时间的关系如下：

| DTCFG[7:5] | The duration of dead-time |
|---|---|
| 3'b0xx | DTCFG[7:0] * $t_{DTS_CK}$ |
| 3'b10x | (64+ DTCFG[5:0]) * $t_{DTS_CK}$ *2 |
| 3'b110 | (32+ DTCFG[4:0]) * $t_{DTS_CK}$ *8 |
| 3'b111 | (32+ DTCFG[4:0]) * $t_{DTS_CK}$ *16 |

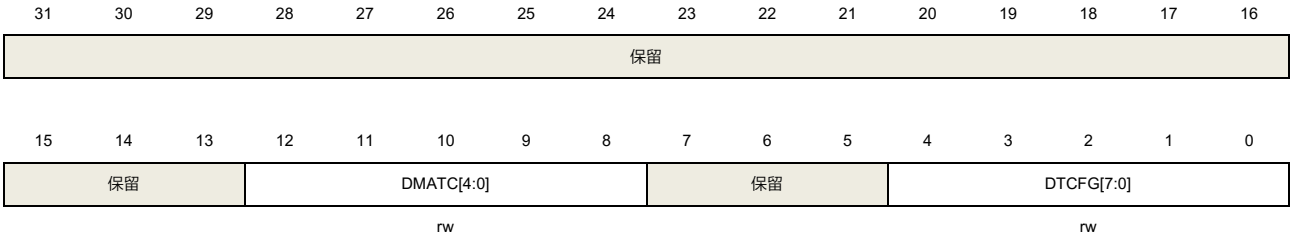
注意：
1. $t_{DTS_CK}$ 是 DTS_CK 的周期，由 TIMERx_CTL0 中的 CKDIC[1:0]定义。

2. 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]=00 时才可修改。

### DMA 配置寄存器 (TIMERx_DMACFG)

地址偏移：0x48
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

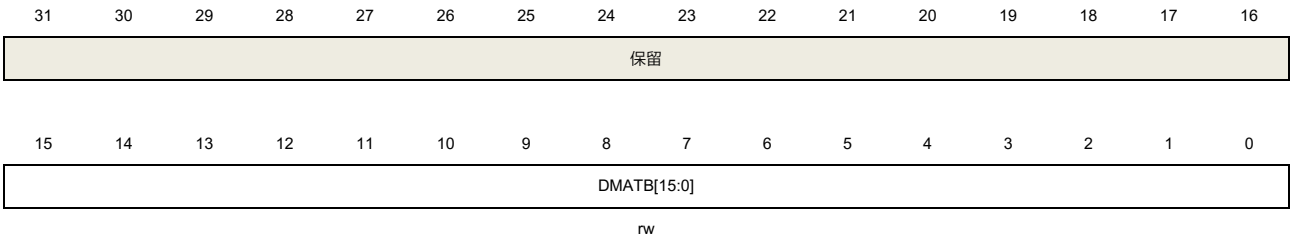



| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:13 | 保留 | 必须保持复位值 |
| 12:8 | DMATC[4:0] | DMA传输计数<br>该位域定义了DMA访问(读写)TIMERx_DMATB寄存器的数量n，n = (DMATC [4:0] +1). DMATC [4:0] 从 5'b0_0000 到 5'b1_0001. |
| 7:5 | 保留 | 必须保持复位值 |
| 4:0 | DMATA[4:0] | DMA传输起始地址<br>该位域定义了DMA访问TIMERx_DMATB寄存器的第一个地址。当第一次访问TIMERx_DMATB寄存器时，实际访问的就是该位域指定的地址。第二次访问TIMERx_DMATB时，将访问(起始地址+0x4)。 |

### DMA 发送缓冲区寄存器 (TIMERx_DMATB)

地址偏移：0x4C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



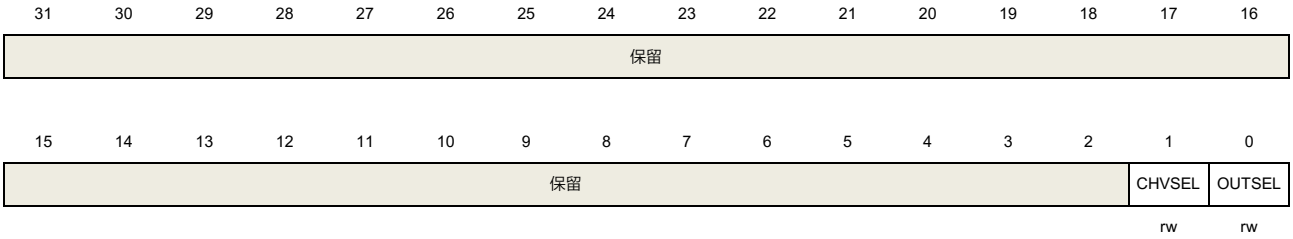

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |

| | | |
|---|---|---|
| 15:0 | DMATB[15:0] | DMA发送缓冲<br>对这个寄存器的读或写，从(起始地址)到(起始地址+传输次数*4)地址范围内的寄存器会被访问。传输次数由硬件计算，范围为0到DMATC。 |

### 配置寄存器 (TIMERx_CFG)

地址偏移：0xFC
复位值：0x0000 0000

该寄存器只能按字(32位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:2 | 保留 | 必须保持复位值 |
| 1 | CHVSEL | 写捕获比较寄存器选择位<br>此位由软件写1或清0。<br>1：当写入捕获比较寄存器的值与寄存器当前值相等时，写入操作无效。<br>0：无影响。 |
| 0 | OUTSEL | 输出值选择位<br>此位由软件写1或清0。<br>1：如果POEN位与IOS位均为0，则输出无效。<br>0：无影响。 |

# 16.2. 通用定时器 L0(TIMERx, x=1,2,3,4)

## 16.2.1. 简介

通用定时器L0(TIMER1/2/3/4)是4通道定时器，支持输入捕获，输出比较，产生PWM信号控制电机和电源管理。通用定时器L0计数器是16位无符号计数器。

通用定时器L0是可编程的，可以被用来计数，其外部事件可以驱动其他定时器。

定时器和定时器之间是相互独立，但是它们的计数器可以被同步在一起形成一个更大的定时器。

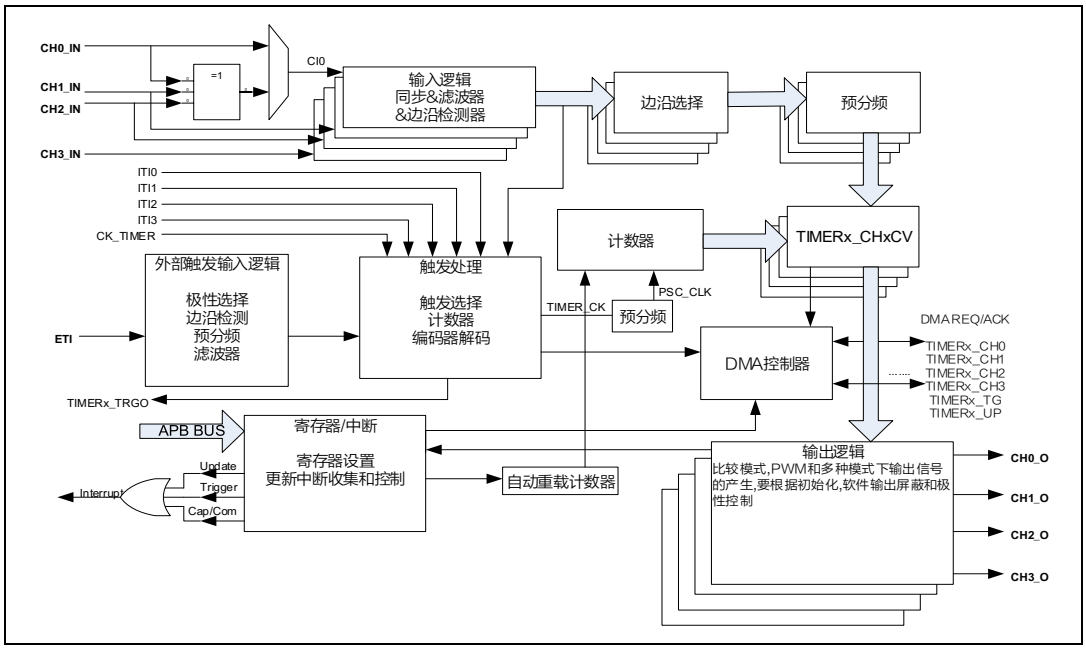
## 16.2.2. 主要特征

- 总通道数：4；
- 计数器宽度：16位；
- 时钟源可选：内部时钟，内部触发，外部输入，外部触发；
- 多种计数模式：向上计数，向下计数和中央计数；
- 正交编码器接口：被用来追踪运动和分辨旋转方向和位置；
- 霍尔传感器接口：用来做三相电机控制；
- 可编程的预分频器：16位，运行时可以被改变；
- 每个通道可配置：输入捕获模式，输出比较模式，可编程的PWM模式，单脉冲模式；
- 自动重装载功能；
- 中断输出和DMA请求：更新事件，触发事件，比较/捕获事件；
- 多个定时器的菊链使得一个定时器可以同时启动多个定时器；
- 定时器的同步允许被选择的定时器在同一个时钟周期开始计数；
- 定时器主-从管理。

## 16.2.3. 结构框图

*图16-29. 通用定时器L0结构框图*提供了通用定时器L0的内部细节。

图 16-29. 通用定时器 L0 结构框图




## 16.2.4. 功能描述

### 时钟源配置

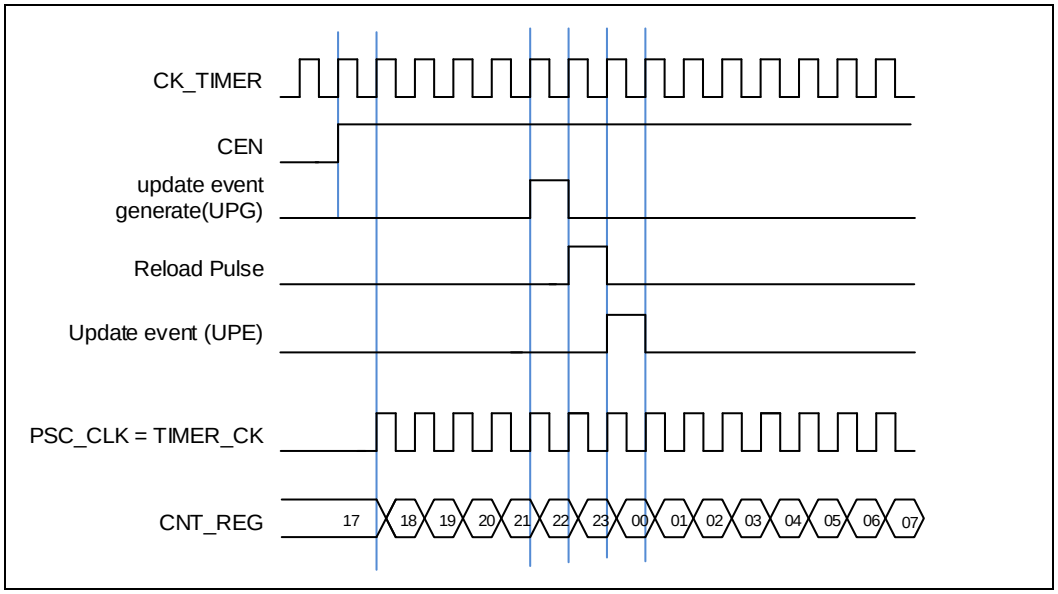
通用定时器 L0 可以由内部时钟源 CK_TIMER 或者由 SMC(TIMERx_SMCFG 寄存器位[2:0])控制的复用时钟源驱动。

■ SMC[2:0]==3'b000，定时器选择内部时钟源(连接到RCU模块的CK_TIMER)

如果 SMC[2:0]==3'b000，默认用来驱动计数器预分频器的是内部时钟源 CK_TIMER。当 CEN 置位，CK_TIMER 经过预分频器(预分频值由 TIMERx_PSC 寄存器确定)产生 PSC_CLK。

如果将 TIMERx_SMCFG 寄存器的 SMC[2:0]设置为 0x1、0x2、0x3 和 0x7，预分频器被其他时钟源(由 TIMERx_SMCFG 寄存器的 TRGS [2:0]区域选择)驱动，在下文说明。当 SMC 位被设置为 0x4、0x5 和 0x6，计数器预分频器时钟源由内部时钟 CK_TIMER 驱动。

**图 16-30. 内部时钟分频为 1 时，计数器的时序图**




■ SMC[2:0]==3'b111(外部时钟模式0)，定时器选择外部输入引脚作为时钟源

计数器预分频器可以在 TIMERx_CI0/ TIMERx_CI1 引脚的每个上升沿或下降沿计数。这种模式可以通过设置 SMC [2:0]为 0x7 同时设置 TRGS [2:0]为 0x4，0x5 或 0x6 来选择。CIx 是 TIMERx_CIx 通过数字滤波器采样后的信号。

计数器预分频器也可以在内部触发信号 ITI0/1/2/3 的上升沿计数。这种模式可以通过设置 SMC [2:0]为 0x7 同时设置 TRGS [2:0]为 0x0, 0x1, 0x2 或者 0x3。

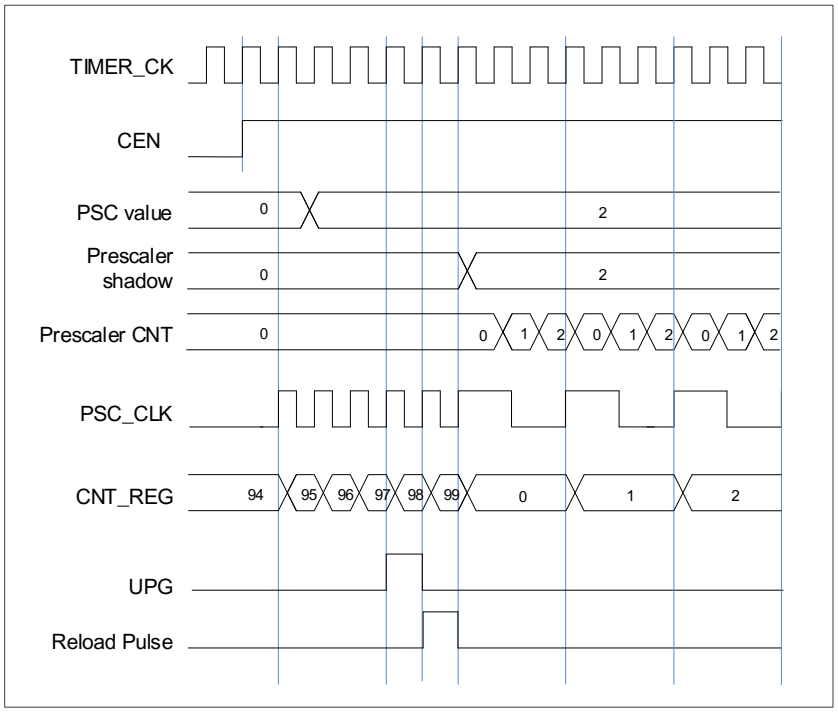
■ SMC==1'b1(外部时钟模式1)，定时器选择外部输入引脚ETI作为时钟源

计数器预分频器可以在外部引脚 ETI 的每个上升沿或下降沿计数。这种模式可以通过设置 TIMERx_SMCFG 寄存器中的 SMC1 位为 1 来选择。另一种选择 ETI 信号作为时钟源方式是，设置 SMC [2:0]为 0x7 同时设置 TRGS [2:0]为 0x7。注意 ETI 信号是通过数字滤波器采样 ETI 引脚得到的。。如果选择 ETIF 信号为时钟源，触发控制器包括边沿监测电路将在每个 ETI 信号上升沿产生一个时钟脉冲来为计数器预分频器提供时钟。

### 时钟预分频器

预分频器可以将定时器的时钟(TIMER_CK)频率按 1 到 65536 之间的任意值分频，分频后的时钟 PSC_CLK 驱动计数器计数。分频系数受预分频寄存器 TIMERx_PSC 控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 16-31. 当 PSC 数值从 0 变到 2 时，计数器的时序图




### 计数器向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从 0 开始向上连续计数到自动加载值(定义在 TIMERx_CAR 寄存器中)，一旦计数器计数到自动加载值，会重新从 0 开始向上计数并产生上溢事件。在向上计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 0。

当通过 TIMERx_SWEVG 寄存器的 UPG 位置 1 来设置更新事件时，计数值会被清 0，并产生更新事件。

如果 TIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(计数器自动重载寄存器，预分频寄存器)都将被更新。

下面的这些图给出了一些例子，当 TIMERx_CAR=0x99 时，计数器在不同预分频因子下的行为。

图 16-32. 向上计数时序图，PSC=0/2

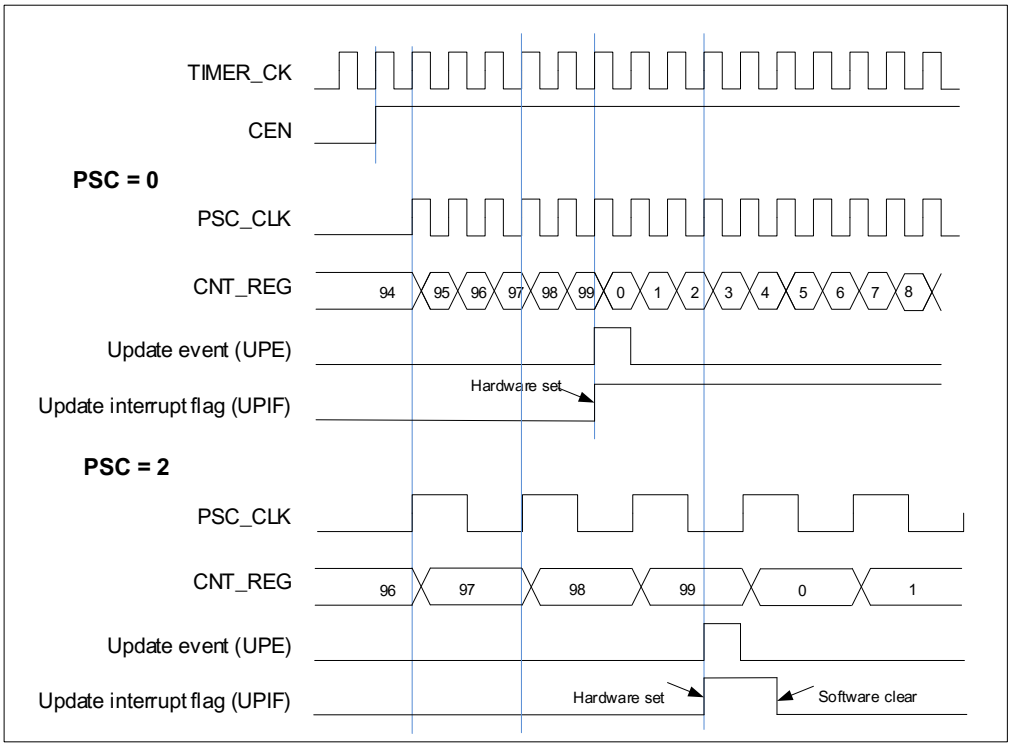
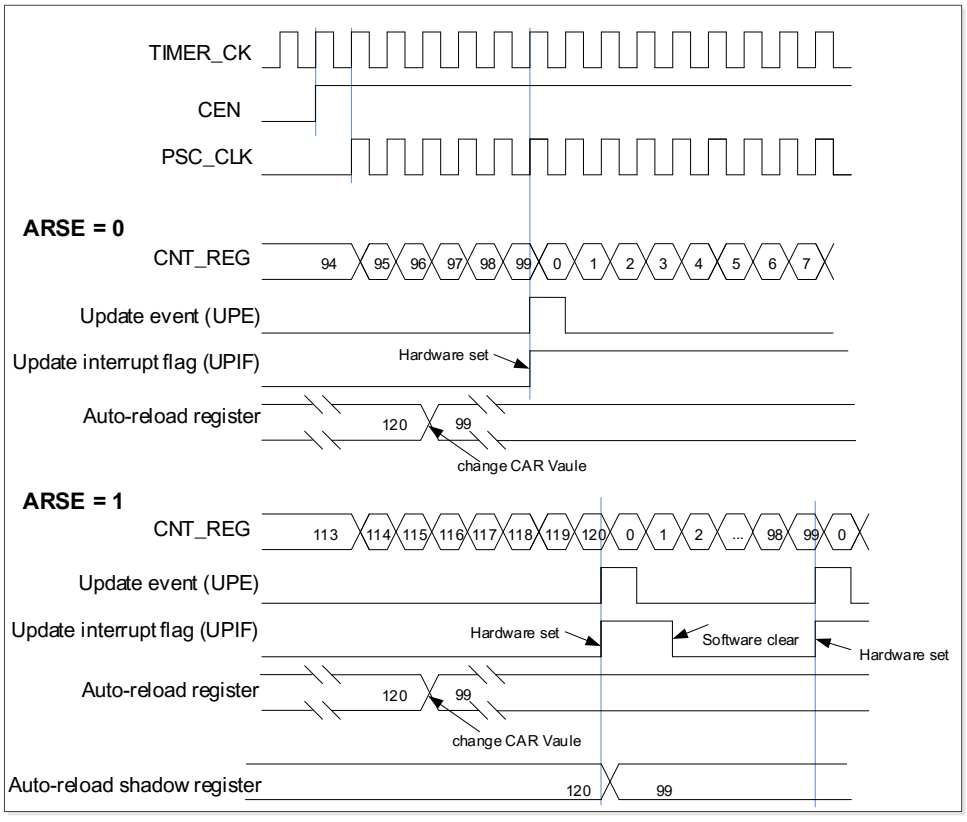



图 16-33. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



### 计数器向下计数模式

在这种模式，计数器的计数方向是向下计数。计数器从自动加载值(定义在 TIMERx_CAR 寄存器中)向下连续计数到 0。一旦计数器计数到 0，计数器会重新从自动加载值开始计数并产生下溢事件。在向下计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 1。

当通过 TIMERx_SWEVG 寄存器的 UPG 位置 1 来设置更新事件时，计数值会被初始化为自动加载值，并产生更新事件。

如果 TIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(计数器自动重载寄存器，预分频寄存器)都将被更新。

下面这些图给出了一些例子，当 TIMERx_CAR=0x99 时，计数器在不同时钟频率下的行为。

**图 16-34. 向下计数时序图，PSC=0/2**

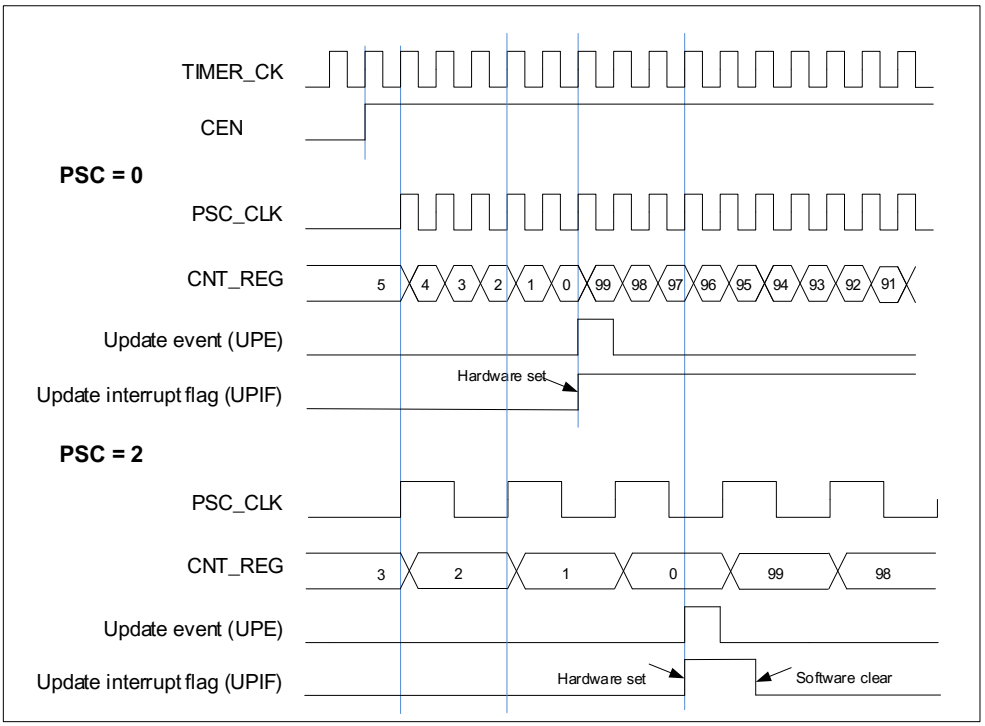
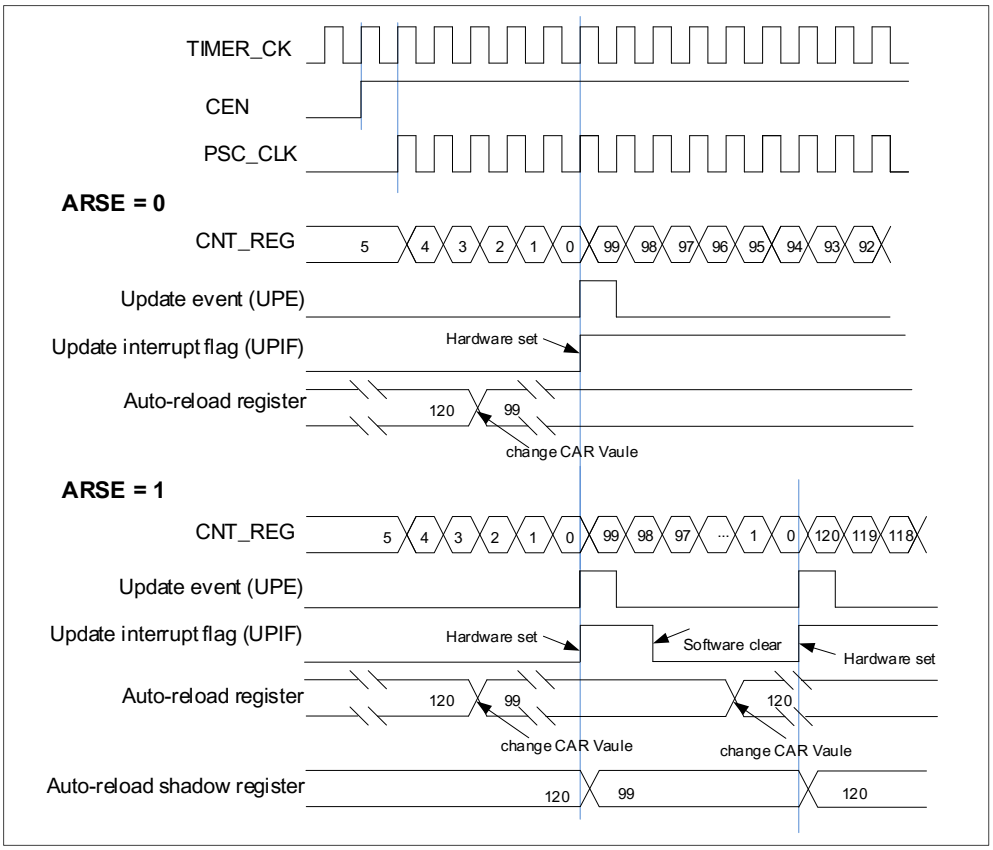


图 16-35. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值




## 计数器中央对齐模式

在中央对齐模式下，计数器交替的从 0 开始向上计数到自动加载值，然后再向下计数到 0。向上计数模式中，定时器模块在计数器计数到(自动加载值-1)产生一个上溢事件；向下计数模式中，定时器模块在计数器计数到 1 时产生一个下溢事件。在中央计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 只读，表明了的计数方向。

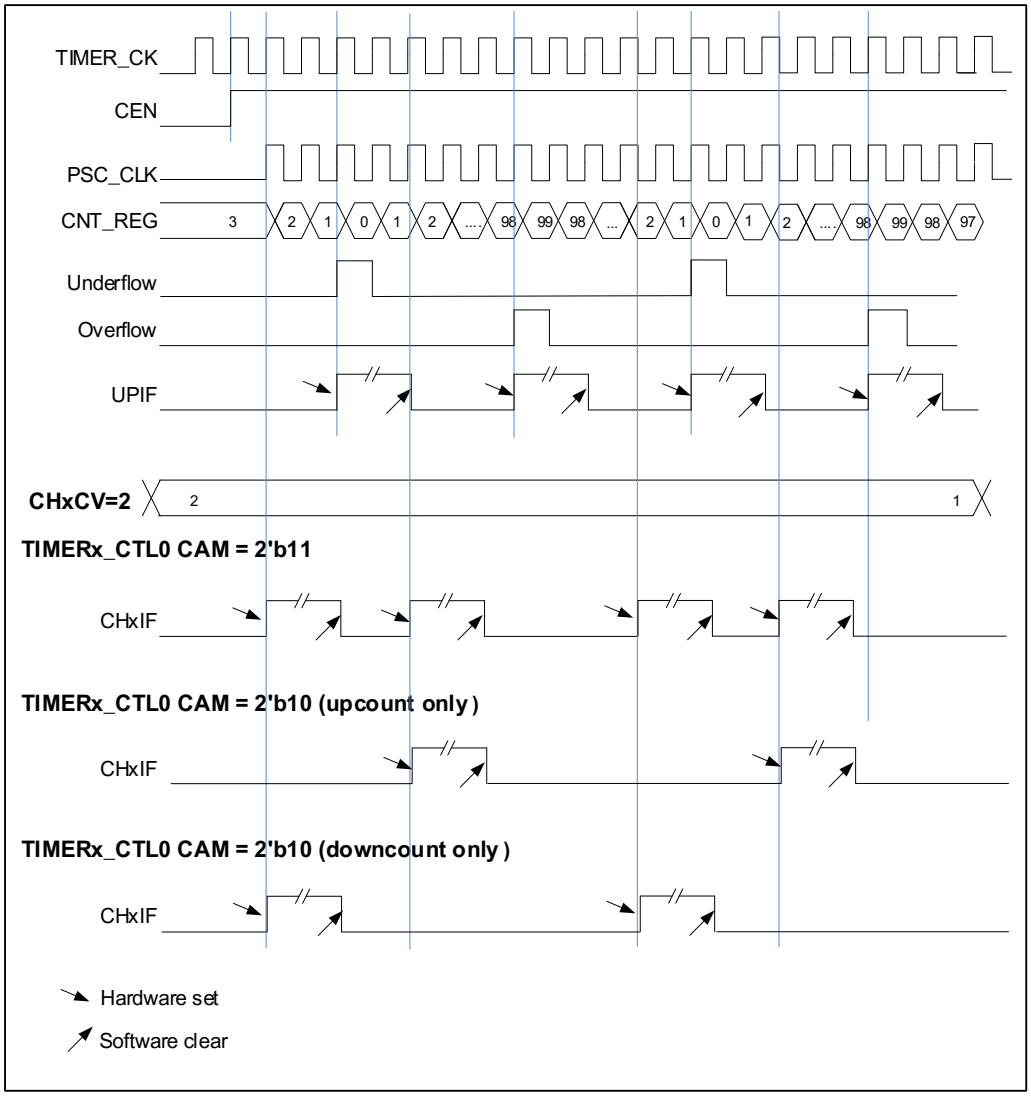
将 TIMERx_SWEVG 寄存器的 UPG 位置 1 可以初始化计数值为 0，并产生一个更新事件，而无需考虑计数器在中央模式下是向上计数还是向下计数。

上溢或者下溢时，TIMERx_INTF 寄存器中的 UPIF 位都会被置 1，然而 CHxIF 位置 1 与 TIMERx_CTL0 寄存器中 CAM 的值有关。具体细节参考*图 16-36. 中央计数模式计数器时序图*。如果 TIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(计数器自动重载寄存器，预分频寄存器)都将被更新。

下面这些图给出了一些例子，当 TIMERx_CAR=0x99，TIMERx_PSC=0x0 时，计数器的行为

图 16-36. 中央计数模式计数器时序图



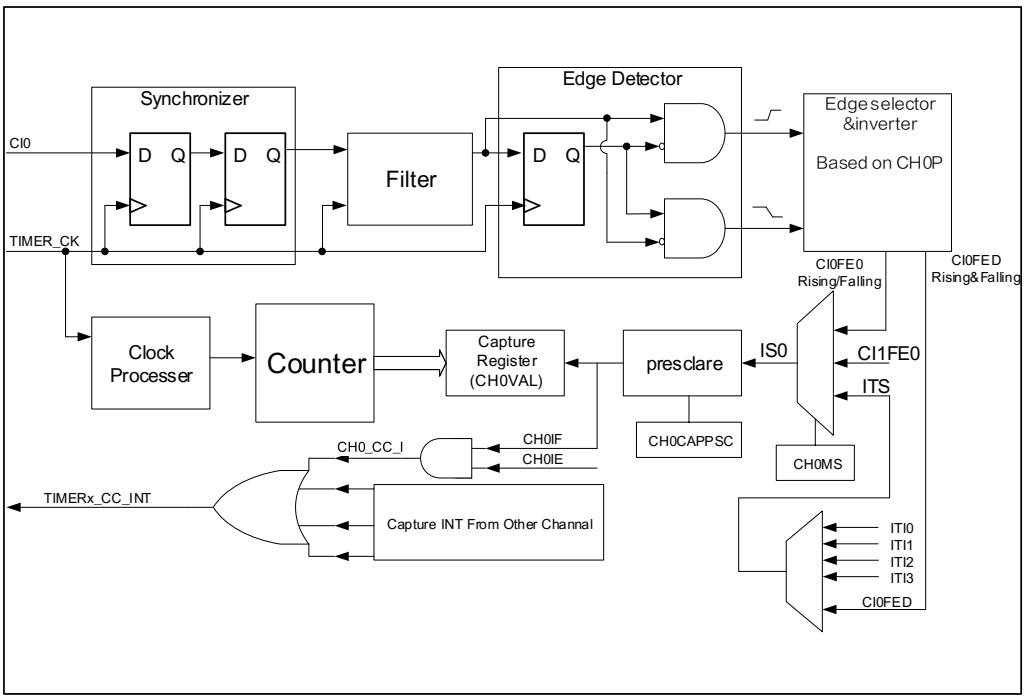

## 输入捕获和输出比较通道

通用定时器 L0 拥有四个独立的通道用于捕获输入或比较输出是否匹配。每个通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

■ 通道输入捕获功能

通道输入捕获功能允许通道测量一个波形时序，频率，周期，占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，TIMERx_CHxCV 寄存器会捕获计数器当前的值，同时 CHxIF 位被置 1，如果 CHxIE = 1 则产生通道中断。

**图 16-37. 通道输入捕获原理**




通道输入信号 CIx 有两种选择，一种是 TIMERx_CHx 信号，另一种是 TIMERx_CH0,TIMERx_CH1 和 TIMERx_CH2 异或之后的信号。通道输入信号 CIx 先被 TIMER_CK 信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置 CHxP 选择使用上升沿或者下降沿。配置 CHxMS.，可以选择其他通道的输入信号，内部触发信号。配置 IC 预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生，CHxVAL 存储计数器的值。

配置步骤如下：

**第一步：** 滤波器配置(TIMERx_CHCTL0寄存器中CHxCAPFLT)：
根据输入信号和请求信号的质量，配置相应的CHxCAPFLT。

**第二步：** 边沿选择(TIMERx_CHCTL2寄存器中CHxP)：
配置CHxP选择上升沿或者下降沿。

**第三步：** 捕获源选择(TIMERx_CHCTL0寄存器中CHxMS)：
一旦通过配置CHxMS选择输入捕获源，必须确保通道配置在输入模式(CHxMS!=0x0)，而且TIMERx_CHxCV寄存器不能再被写。

**第四步：** 中断使能(TIMERx_DMAINTEN寄存器中CHxIE 和 CHxDEN)：
使能相应中断，可以获得中断和DMA请求。

**第五步：** 捕获使能(TIMERx_CHCTL2寄存器中CHxEN)。

**结果：** 当期望的输入信号发生时，TIMERx_CHxCV被设置成当前计数器的值，CHxIF为置1。如果CHxIF位已经为1，则CHxOF位置1。根据TIMERx_DMAINTEN寄存器中CHxIE和CHxDEN的配置，相应的中断和DMA请求会被提出。

**直接产生：** 软件设置CHxG位，会直接产生中断和DMA请求。

通道输入捕获功能也可用来测量 TIMERx_CHx 引脚上信号的脉冲波宽度。例如，一个 PWM 波连接到 CI0。配置 TIMERx_CHCTL0 寄存器中 CH0MS 为 2'b01，选择通道 0 的捕获信号为 CI0 并设置上升沿捕获。配置 TIMERx_CHCTL0 寄存器中 CH1MS 为 2'b10，选择通道 1 捕获信号为 CI0 并设置下降沿捕获。计数器配置为复位模式，在通道 0 的上升沿复位。TIMERX_CH0CV 寄存器测量 PWM 的周期值，TIMERx_CH1CV 寄存器测量 PWM 占空比值。

■ 通道输出比较模式

在输出比较模式，TIMERx 可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的 CHxCV 寄存器与计数器的值匹配时，根据 CHxCOMCTL 的配置，这个通道的输出可以被置高电平，被置低电平或者反转。当计数器的值与 CHxCV 寄存器的值匹配时，CHxIF 位被置 1，如果 CHxIE = 1 则会产生中断，如果 CxCDE=1 则会产生 DMA 请求。

配置步骤如下：

**第一步：** 时钟配置：
配置定时器时钟源，预分频器等。

**第二步：** 比较模式配置：
设置CHxCOMSEN位来配置输出比较影子寄存器；
设置CHxCOMCTL位来配置输出模式(置高电平/置低电平/反转)；
设置CHxP位来选择有效电平的极性；
设置CHxEN使能输出。

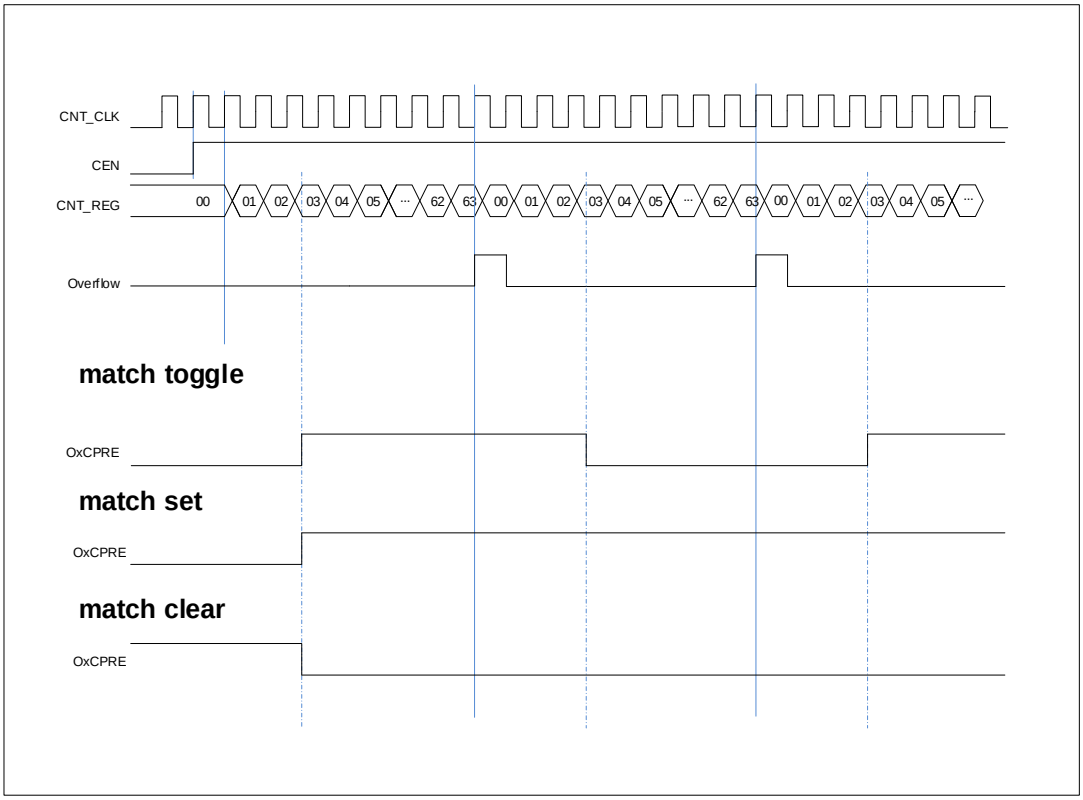
**第三步：** 通过CHxIE/CxCDE位配置中断/DMA请求使能。

**第四步：** 通过TIMERx_CAR寄存器和TIMERx_CHxCV寄存器配置输出比较时基：
CHxVAL可以在运行时根据你所期望的波形而改变。

**第五步：** 设置CEN位使能定时器。

*图16-38. 三种输出比较模式*显示了三种比较输出模式：反转/置高电平/置低电平，CAR=0x63，CHxVAL=0x3。

图 16-38. 三种输出比较模式




## 输出 PWM 功能

在 PWM 输出模式下(PWM 模式 0 是配置 CHxCOMCTL 为 3'b110，PWM 模式 1 是配置 CHxCOMCTL 为 3'b111)，通道根据 TIMERx_CAR 寄存器和 TIMERx_CHxCV 寄存器的值，输出 PWM 波形。

根据计数模式，我们可以分为两种 PWM 波：EAPWM(边沿对齐 PWM)和 CAPWM(中央对齐 PWM)。

EAPWM 的周期由 TIMERx_CAR 寄存器值决定，占空比由 TIMERx_CHxCV 寄存器值决定。*图 16-39. EAPWM 时序图*显示了 EAPWM 的输出波形和中断。

CAPWM 的周期由(2*TIMERx_CAR 寄存器值)决定，占空比由(2*TIMERx_CHxCV 寄存器值)决定。*图 16-40. CAPWM 时序图*显示了 CAPWM 的输出波形和中断。

在 PWM0 模式下(CHxCOMCTL==3'b110)，如果 TIMERx_CHxCV 寄存器的值大于 TIMERx_CAR 寄存器的值，通道输出一直为有效电平。

在 PWM0 模式下(CHxCOMCTL==3'b110)，如果 TIMERx_CHxCV 寄存器的值等于 0，通道输出一直为无效电平。

图 16-39. EAPWM 时序图

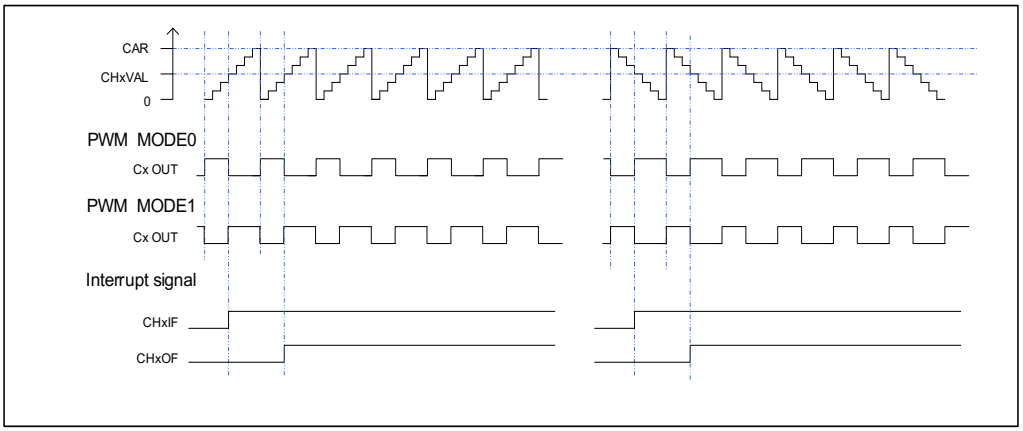
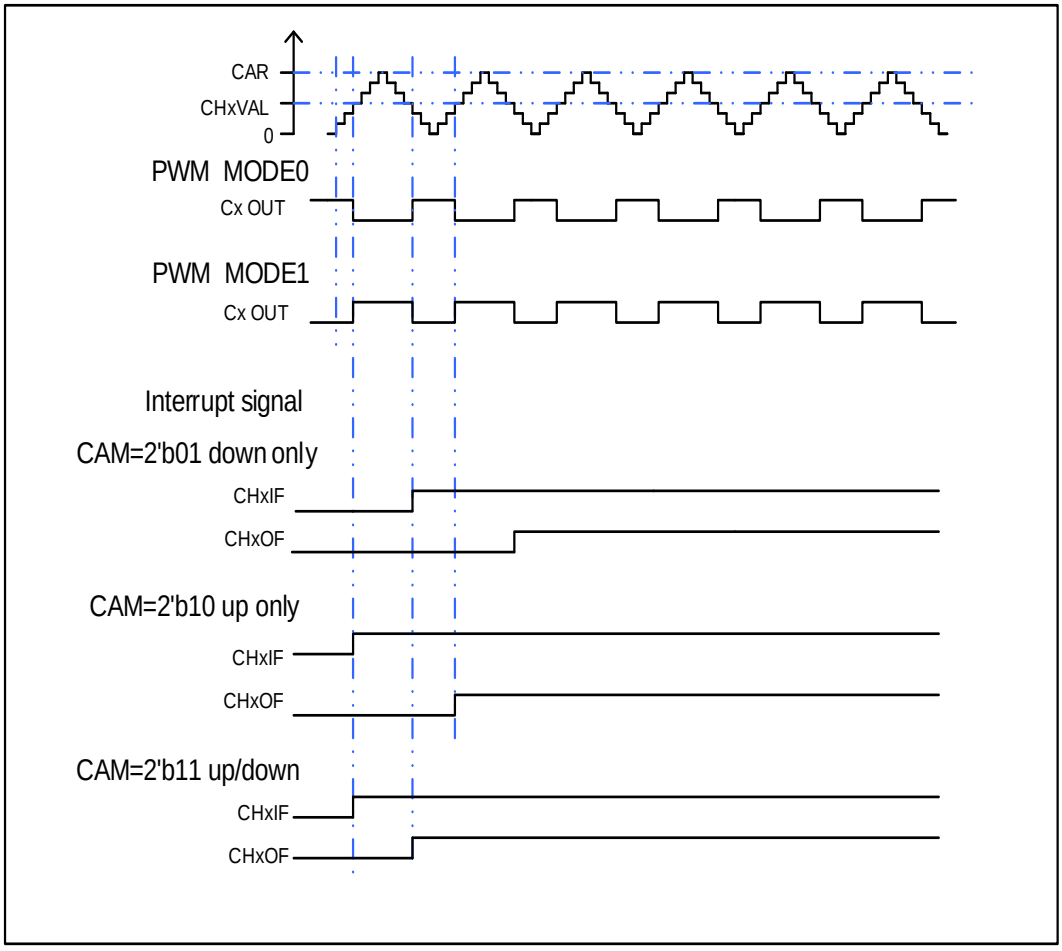



图 16-40. CAPWM 时序图




### 通道输出准备信号

当 TIMERx 用于输出匹配比较模式下，设置 CHxCOMCTL 位可以定义 OxCPRE 信号(通道 x 准备信号)类型。OxCPRE 信号有若干类型的输出功能，包括，设置 CHxCOMCTL=0x00 可以保持原始电平；设置 CHxCOMCTL=0x01 可以将 OxCPRE 信号设置为高电平；设置 CHxCOMCTL=0x02 可以将 OxCPRE 信号设置为低电平；设置 CHxCOMCTL=0x03，在计数

器值和 TIMERx_CHxCV 寄存器的值匹配时，可以翻转输出信号。

PWM 模式 0 和 PWM 模式 1 是 OxCPRE 的另一种输出类型，设置 CHxCOMCTL 位域位 0x06 或 0x07 可以配置 PWM 模式 0/PWM 模式 1。在这些模式中，根据计数器值和 TIMERx_CHxCV 寄存器值的关系以及计数方向，OxCPRE 信号改变其电平。具体细节描述，请参考相应的位。

设置 CHxCOMCTL =0x04 或 0x05 可以实现 OxCPRE 信号的强制输出功能。输出比较信号能够直接由软件强置为有效或无效状态，而不依赖于 TIMERx_CHxCV 的值和计数器值之间间的比较结果。

设置 CHxCOMCEN=1，当由外部 ETI 引脚信号产生的 ETIFE 信号为高电平时，OxCPRE 被强制为低电平。在下一次更新事件到来时，OxCPRE 信号才会回到有效电平状态。

### 正交译码器

参考*正交译码器*。

### 霍尔传感器接口功能

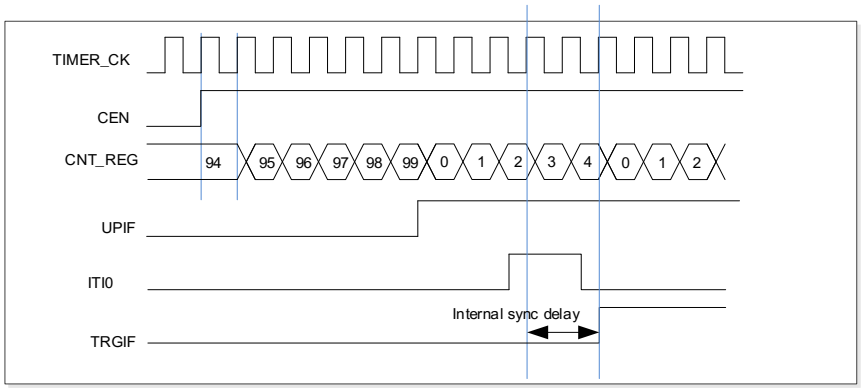
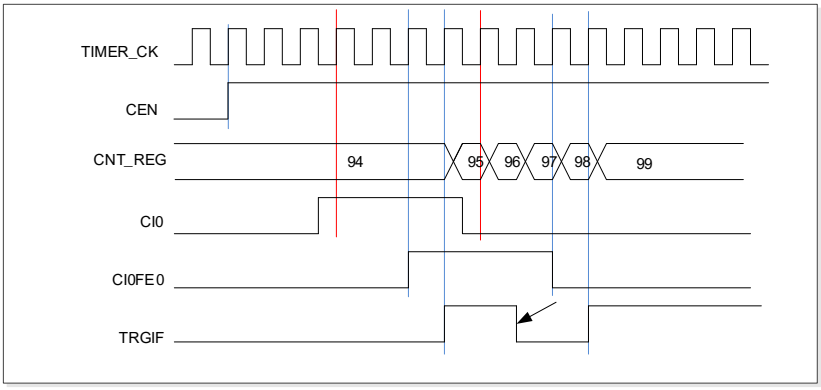
参考*霍尔传感器接口功能*。

### 主-从管理

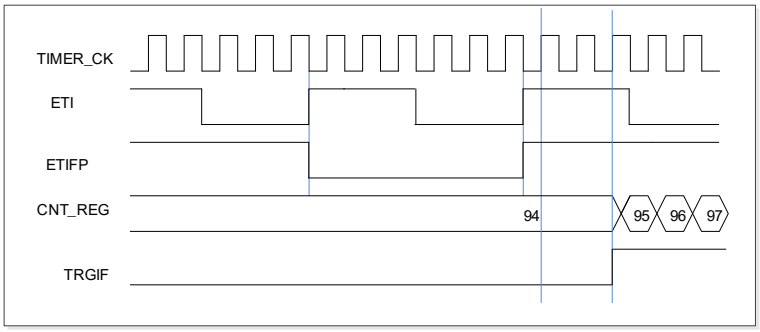
TIMERx 能在多种模式下同步外部触发，包括复位模式，暂停模式和事件模式，可以通过设置 TIMERx_SMCFG 寄存器中的 SMC[2:0]配置这些模式。这些模式的输入触发源可以通过设置 TIMERx_SMCFG 寄存器中的 TRGS[2:0]来选择。

**表 16-5. 从模式列表和举例**

| | 模式选择 | 触发源选择 | 极性选择 | 滤波和预分频 |
|---|---|---|---|---|
| 列举 | SMC[2:0]<br>3'b100 (复位模式)<br>3'b101 (暂停模式)<br>3'b110 (事件模式) | TRGS[2:0]<br>000: ITI0<br>001: ITI1<br>010: ITI2<br>011: ITI3<br>100: CI0F_ED<br>101: CI0FE0<br>110: CI1FE1<br>111: ETIFP | 如果触发源是CI0FE0或者CI1FE1，配置CHxP来选择极性和反相<br>如果触发源是ETIF，配置ETP选择极性和反相 | 触发源ITIx，滤波和预分频不可用<br>触发源CIx，配置CHxCAPFLT设置滤波，分频不可用<br>触发源是ETIF，滤波和预分频不可用 |
| 例1 | 复位模式<br>当触发输入上升沿，计数器清零重启 | TRGIS[2:0]=3'b000<br>选择ITI0为触发源 | 触发源是ITI0，极性选择不可用 | 触发源是 ITI0，滤波和预分频不可用 |

| | 模式选择 | 触发源选择 | 极性选择 | 滤波和预分频 |
|---|---|---|---|---|
| | 图 16-41. 复位模式下的控制电路<br> | | | |
| 例2 | 暂停模式<br>当触发输入为低的时候，计数器暂停计数 | TRGIS[2:0]=3'b101<br>选择CI0FE0为触发源 | TI0S=0.(非异或)<br>CH0P==0不反相.在上升沿捕获 | 在这个例子中滤波被旁路 |
| | 图 16-42. 暂停模式下的控制电路<br> | | | |
| 例3 | 事件模式<br>触发输入的上升沿计数器开始计数 | TRGIS[2:0]=3'b111<br>选择ETIF为触发源. | ETP = 0 没有极性改变 | ETPSC = 1, 2分频.<br>ETFC = 0，无滤波 |

| | 模式选择 | 触发源选择 | 极性选择 | 滤波和预分频 |
|---|---|---|---|---|
| | 图 16-43. 事件模式下的控制电路 | | | |




### 单脉冲模式

参考*单脉冲模式*。

### 定时器互连

参考*高级定时器(TIMERx,x=0,7)互连*

### 定时器 DMA 模式

定时器 DMA 模式是指通过 DMA 模块配置定时器的寄存器。有两个跟定时器 DMA 模式相关的寄存器：TIMERx_DMACFG 和 TIMERx_DMATB。当然，必须要使能 DMA 请求，一些内部中断事件可以产生 DMA 请求。当中断事件发生，TIMERx 会给 DMA 发送请求。DMA 配置成 M2P 模式，PADDR 是 TIMERx_DMATB 寄存器地址，DMA 就会访问 TIMERx_DMATB 寄存器。实际上，TIMERx_DMATB 寄存器只是一个缓冲，定时器会将 TIMERx_DMATB 映射到一个内部寄存器，这个内部寄存器由 TIMERx_DMACFG 寄存器中的 DMATA 来指定。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值为 0，表示 1 次传输，定时器的发送 1 个 DMA 请求就可以完成。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值不为 1，例如其值为 3，表示 4 次传输，定时器就需要再多发 3 次 DMA 请求。在这 3 次请求下，DMA 对 TIMERx_DMATB 寄存器的访问会映射到访问定时器的 DMATA+0x4, DMATA+0x8, DMATA+0xc 寄存器。总之，发生一次 DMA 内部中断请求，定时器会连续发送(DMATC+1)次请求。

如果再来 1 次 DMA 请求事件，TIMERx 将会重复上面的过程。

### 定时器调试模式

当Cortex®-M4内核停止，DBG_CTL0寄存器中的TIMERx_HOLD配置位被置1，定时器计数器停止。

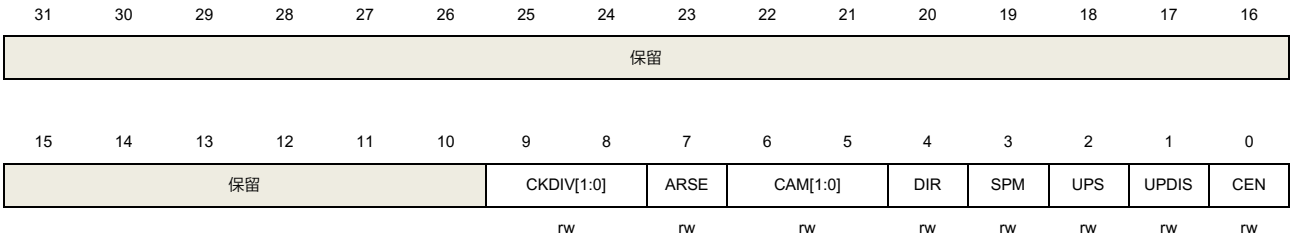
## 16.2.5. TIMERx 寄存器(x=1,2,3,4)

TIMER1基地址：0x4000 0000
TIMER2基地址：0x4000 0400
TIMER3基地址：0x4000 0800
TIMER4基地址：0x4000 0C00

### 控制寄存器 0 (TIMERx_CTL0)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:10 | 保留 | 必须保持复位值 |
| 9:8 | CKDIV[1:0] | 时钟分频<br>通过软件配置CKDIV，规定定时器时钟(CK_TIMER) 与死区时间和数字滤波器采样时钟(DTS)之间的分频系数。<br>00：$f_{DTS}=f_{CK_TIMER}$<br>01：$f_{DTS}= f_{CK_TIMER} /2$<br>10：$f_{DTS}= f_{CK_TIMER} /4$<br>11：保留 |
| 7 | ARSE | 自动重载影子使能<br>0：禁能TIMERx_CAR寄存器的影子寄存器<br>1：使能TIMERx_CAR寄存器的影子寄存器 |
| 6:5 | CAM[1:0] | 计数器对齐模式选择<br>00：无中央对齐计数模式(边沿对齐模式)。 DIR位指定了计数方向<br>01：中央对齐向下计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式(TIMERx_CHCTL0寄存器中CHxMS=00)，只有在向下计数时，CHxF位置1<br>10：中央对齐向上计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式(TIMERx_CHCTL0寄存器中CHxMS=00)，只有在向上计数时，CHxF位置1<br>11：中央对齐上下计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式(TIMERx_CHCTL0寄存器中CHxMS=00)，在向上和向下计数时，CHxF位都会置1<br>当计数器使能以后，该位不能从0x00切换到非0x00 |

| | | |
|---|---|---|
| 4 | DIR | 方向<br>0：向上计数<br>1：向下计数<br>当计数器配置为中央对齐计数模式或编码器模式时，该位只读。 |
| 3 | SPM | 单脉冲模式<br>0：单脉冲模式禁能。更新事件发生后，计数器继续计数<br>1：单脉冲模式使能。在下一次更新事件发生时，计数器停止计数 |
| 2 | UPS | 更新请求源<br>软件配置该位，选择更新事件源.<br>0：以下事件均会产生更新中断或DMA请求：<br>UPG位被置1<br>计数器溢出/下溢<br>复位模式产生的更新<br>1：下列事件会产生更新中断或DMA请求：<br>计数器溢出/下溢 |
| 1 | UPDIS | 禁止更新.<br>该位用来使能或禁能更新事件的产生<br>0：更新事件使能. 更新事件发生时，相应的影子寄存器被装入预装载值，以下事件均会产生更新事件：<br>UPG位被置1<br>计数器溢出/下溢<br>复位模式产生的更新<br>1：更新事件禁能.<br>注意：当该位被置1时，UPG位被置1或者复位模式不会产生更新事件，但是计数器和预分频器被重新初始化 |
| 0 | CEN | 计数器使能<br>0：计数器禁能<br>1：计数器使能<br>在软件将CEN位置1后，外部时钟、暂停模式和编码器模式才能工作。 |

### 控制寄存器 1 (TIMERx_CTL1)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | TI0S | MMC[2:0] | | | DMAS | 保留 | | |
| | | | | | | | | rw | rw | | | rw | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 必须保持复位值 |
| 7 | TI0S | 通道0触发输入选择<br>0：选择TIMERx_CH0引脚作为通道0的触发输入<br>1：选择TIMERx_CH0，TIMERx_CH1和TIMERx_CH2引脚异或的结果作为通道0的触发输入 |
| 6:4 | MMC[2:0] | 主模式控制<br>这些位控制TRGO信号的选择，TRGO信号由主定时器发给从定时器用于同步功能<br>000：当产生一个定时器复位事件后，输出一个TRGO信号，定时器复位源为：<br>主定时器产生一个复位事件<br>TIMERx_SWEVG寄存器中UPG位置1<br>001：当产生一个定时器使能事件后，输出一个TRGO信号，定时器使能源为：<br>CEN位置1<br>在暂停模式下，触发输入置1<br>010：当产生一个定时器更新事件后，输出一个TRGO信号，更新事件源由UPDIS和UPS位决定<br>011：当通道0在发生一次捕获或一次比较成功时，主模式控制器产生一个TRGO脉冲<br>100：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O0CPRE<br>101：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O1CPRE<br>110：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O2CPRE<br>111：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O3CPRE |
| 3 | DMAS | DMA请求源选择<br>0：当通道捕获/比较事件发生时，发送通道x的DMA请求<br>1：当更新事件发生，发送通道x的DMA请求 |
| 2:0 | 保留 | 必须保持复位值 |

### 从模式配置寄存器(TIMERx_SMCFG)

地址偏移：0x08
复位值： 0x0000 0000

该寄存器只能按字(32位)访问

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ETP | SMC1 | ETPSC[1:0] | | ETFC[3:0] | | | | MSM | TRGS[2:0] | | | 保留 | SMC[2:0] | | |
| 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 ETP 外部触发极性
该位指定 ETI 信号的极性
0：ETI 高电平或上升沿有效 .
1：ETI 低电平或下降沿有效 .

14 SMC1 SMC 的一部分为了使能外部时钟模式 1
在外部时钟模式 1，计数器由 ETIF 信号上的任意有效边沿驱动
0：外部时钟模式 1 禁能
1：外部时钟模式 1 使能
当从模式配置为复位模式，暂停模式和事件模式时，定时器仍然可以工作在外部时钟模式 1。但是 TRGS 必须不能为 3'b111。
如果外部时钟模式 0 和外部时钟模式 1 同时被被配置，外部时钟的输入是 ETIF
注意：外部时钟模式 0 使能在寄存器的 SMC[2:0]位域。

13:12 ETPSC[1:0] 外部触发预分频
外部触发信号 ETIFP 的频率不能超过 TIMER_CK 频率的 1/4。当输入较快的外部时钟时，可以使用预分频降低 ETIFP 的频率。
00：预分频禁能
01：2 分频
10：4 分频
11：8 分频

11:8 ETFC[3:0] 外部触发滤波控制
外部触发信号可以通过数字滤波器进行滤波，该位域定义了数字滤波器的滤波能力。数字滤波器的基本原理是：以 $f_{SAMP}$ 频率连续采样外部触发信号，同时记录采样相同电平的次数。当该次数达到配置的滤波能力时，则认为是一个有效的电平信号。

| EXTFC[3:0] | 次数 | $f_{SAMP}$ |
|---|---|---|
| 4'b0000 | Filter disabled. | |
| 4'b0001 | 2 | $f_{CK_TIMER}$ |
| 4'b0010 | 4 | |
| 4'b0011 | 8 | |
| 4'b0100 | 6 | $f_{DTS_CK}/2$ |
| 4'b0101 | 8 | |
| 4'b0110 | 6 | $f_{DTS_CK}/4$ |
| 4'b0111 | 8 | |
| 4'b1000 | 6 | $f_{DTS_CK}/8$ |
| 4'b1001 | 8 | |
| 4'b1010 | 5 | $f_{DTS_CK}/16$ |
| 4'b1011 | 6 | |
| 4'b1100 | 8 | |
| 4'b1101 | 5 | $f_{DTS_CK}/32$ |
| 4'b1110 | 6 | |
| 4'b1111 | 8 | |

7 MSM 主-从模式

| | | |
|---|---|---|
| | | 该位被用来同步被选择的定时器同时开始计数。通过 TRIGI 和 TRGO，定时器被连接在一起，TRGO 用做启动事件。<br>0：主从模式禁能<br>1：主从模式使能 |
| 6:4 | TRGS[2:0] | 触发选择<br>该位域用来指定选择哪一个信号作为用来同步计数器的触发输入源<br>000: ITI0<br>001: ITI1<br>010: ITI2<br>011: ITI3<br>100: CI0F_ED<br>101: CI0FE0<br>110: CI1FE1<br>111: ETIFP<br>从模式被使能后这些位不能改 |
| 3 | 保留 | 必须保持复位值 |
| 2:0 | SMC[2:0] | 从模式控制<br>000：关闭从模式. 如果 CEN=1，则预分频器直接由内部时钟驱动<br>001：编码器模式 0. 根据 CI1FE1 的电平，计数器在 CI0FE0 的边沿向上/下计数<br>010：编码器模式 1. 根据 CI0FE0 的电平，计数器在 CI1FE1 的边沿向上/下计数<br>011：编码器模式 2. 根据另一个信号的输入电平，计数器在 CI0FE0 和 CI1FE1 的边沿向上/ 下计数<br>100：复位模式. 选中的触发输入的上升沿重新初始化计数器，并且产生更新事件.<br>101：暂停模式. 当触发输入为高时，计数器的时钟开启。一旦触发输入变为低，则计数器时钟停止<br>110：事件模式.计数器在触发输入的上升沿启动。<br>111：外部时钟模式 0. 选中的触发输入的上升沿驱动计数器 |

### DMA 和中断使能寄存器 (TIMERx_DMAINTEN)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | TRGDEN | 保留 | CH3DEN | CH2DEN | CH1DEN | CH0DEN | UPDEN | 保留 | TRGIE | 保留 | CH3IE | CH2IE | CH1IE | CH0IE | UPIE |
| | rw | rw | rw | rw | rw | rw | rw | rw | rw | | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|

| | | |
|---|---|---|
| 31:15 | 保留 | 必须保持复位值 |
| 14 | TRGDEN | 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 | 必须保持复位值 |
| 12 | CH3DEN | 通道3比较/捕获 DMA请求使能<br>0：禁止通道3比较/捕获DMA请求<br>1：使能通道3比较/捕获DMA请求 |
| 11 | CH2DEN | 通道2比较/捕获 DMA请求使能<br>0：禁止通道2比较/捕获DMA请求<br>1：使能通道2比较/捕获DMA请求 |
| 10 | CH1DEN | 通道1比较/捕获 DMA请求使能<br>0：禁止通道1比较/捕获DMA请求<br>1：使能通道1比较/捕获DMA请求 |
| 9 | CH0DEN | 通道0比较/捕获 DMA请求使能<br>0：禁止通道0比较/捕获DMA请求<br>1：使能通道0比较/捕获DMA请求 |
| 8 | UPDEN | 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 | TRGIE | 触发中断使能<br>0：禁止触发中断<br>1：使能触发中断 |
| 5 | 保留 | 必须保持复位值 |
| 4 | CH3IE | 通道3比较/捕获中断使能<br>0：禁止通道3中断<br>1：使能通道3中断 |
| 3 | CH2IE | 通道2比较/捕获中断使能<br>0：禁止通道2中断<br>1：使能通道2中断 |
| 2 | CH1IE | 通道1比较/捕获中断使能<br>0：禁止通道1中断<br>1：使能通道1中断 |
| 1 | CH0IE | 通道0比较/捕获中断使能<br>0：禁止通道0中断<br>1：使能通道0中断 |

| | | |
|---|---|---|
| 0 | UPIE | 更新中断使能<br>0：禁止更新中断<br>1：使能更新中断 |

### 中断标志寄存器（TIMERx_INTF)

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | CH3OF | CH2OF | CH1OF | CH0OF | 保留 | | TRGIF | 保留 | CH3IF | CH2IF | CH1IF | CH0IF | UPIF |
| | | | rc_w0 | 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 |
|---|---|---|
| 31:13 | 保留 | 必须保持复位值 |
| 12 | CH3OF | 通道3捕获溢出标志<br>参见CH0OF描述 |
| 11 | CH2OF | 通道2捕获溢出标志<br>参见CH0OF描述 |
| 10 | CH1OF | 通道1捕获溢出标志<br>参见CH0OF描述 |
| 9 | CH0OF | 通道0捕获溢出标志<br>当通道0被配置为输入模式时，在CH0IF标志位已经被置1后，捕获事件再次发生时，该标志位可以由硬件置1。该标志位由软件清0<br>0：无捕获溢出中断发生<br>1：发生了捕获溢出中断 |
| 8:7 | 保留 | 必须保持复位值 |
| 6 | TRGIF | 触发中断标志<br>当发生触发事件时，此标志会置 1，此位由软件清 0。当暂停模式使能时，触发输入的任意边沿都可以产生触发事件。否则，其它模式时，仅在触发输入端检测到有效边沿，产生触发事件。<br>0：无触发事件产生<br>1：触发中断产生 |
| 5 | 保留 | 必须保持复位值 |
| 4 | CH3IF | 通道3比较/捕获中断标志 |

| | | |
|---|---|---|
| | | 参见CH0IF描述 |
| 3 | CH2IF | 通道2比较/捕获中断标志<br>参见CH0IF描述 |
| 2 | CH1IF | 通道1比较/捕获中断标志<br>参见CH0IF描述 |
| 1 | CH0IF | 通道0比较/捕获中断标志<br>此标志由硬件置1软件清0。当通道0在输入模式下时，捕获事件发生时此标志位被置1；当通道0在输出模式下时，此标志位在一个比较事件发生时被置1。<br>0：无通道0中断发生<br>1：通道0中断发生 |
| 0 | UPIF | 更新中断标志<br>此位在更新事件发生时由硬件置1，软件清0。<br>0：无更新中断发生<br>1：发生更新中断 |

### 软件事件产生寄存器 (TIMERx_SWEVG)

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | | TRGG | 保留 | CH3G | CH2G | CH1G | CH0G | UPG |
| | | | | | | | | | w | | w | w | w | w | w |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:7 | 保留 | 必须保持复位值 |
| 6 | TRGG | 触发事件产生<br>此位由软件置1，由硬件自动清0。 当此位被置1，TIMERx_INTF寄存器的TRGIF标志位被置1，若开启对应的中断和DMA，则产生相应的中断和DMA传输。<br>0：无触发事件产生<br>1：产生触发事件 |
| 5 | 保留 | 必须保持复位值 |
| 4 | CH3G | 通道3捕获或比较事件发生<br>参见CH0G描述 |
| 3 | CH2G | 通道2捕获或比较事件发生 |

| | | |
|---|---|---|
| | | 参见CH0G描述 |
| 2 | CH1G | 通道1捕获或比较事件发生<br>参见CH0G描述 |
| 1 | CH0G | 通道0捕获或比较事件发生<br>该位由软件置1，用于在通道0产生一个捕获/比较事件，由硬件自动清0。当此位被置1，CH0IF标志位被置1，若开启对应的中断和DMA，则发出相应的中断和DMA请求。此外，如果通道0配置为输入模式，计数器的当前值被捕获到TIMERx_CH0CV寄存器，如果CH0IF标志位已经为1，则CH0OF标志位被置1。<br>0：不产生通道0捕获或比较事件<br>1：发生通道0捕获或比较事件 |
| 0 | UPG | 更新事件产生<br>此位由软件置1，被硬件自动清0。当此位被置1，如果选择了中央对齐或向上计数模式，计数器被清0。否则(向下计数模式)计数器将载入自动重载值，预分频计数器将同时被清除。<br>0：无更新事件产生<br>1：产生更新事件 |

### 通道控制寄存器 0 (TIMERx_CHCTL0)

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CH1COMCEN | CH1COMCTL[2:0] | | | CH1COMSEN | CH1COMFEN | CH1MS[1:0] | | CH0COMCEN | CH0COMCTL[2:0] | | | CH0COMSEN | CH0COMFEN | CH0MS[1:0] | |
| CH1CAPFLT[3:0] | | | | CH1CAPPSC[1:0] | | | | CH0CAPFLT[3:0] | | | | CH0CAPPSC[1:0] | | | |
| rw | | | | rw | | rw | | rw | | | | rw | | rw | |

**输出比较模式：**

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15 | CH1COMCEN | 通道1输出比较清0使能<br>参见CH0COMCEN描述 |
| 14:12 | CH1COMCTL[2:0] | 通道1输出比较模式<br>参见CH0COMCTL描述 |
| 11 | CH1COMSEN | 通道1输出比较影子寄存器使能 |

| | | |
|---|---|---|
| | | 参见CH0COMSEN描述 |
| 10 | CH1COMFEN | 通道1输出比较快速使能<br>参见CH0COMFEN描述 |
| 9:8 | CH1MS[1:0] | 通道 1 模式选择<br>这些位定义了通道的方向和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH1EN 位被清 0)时这些位才可以写。<br>00：通道 1 配置为输出<br>01：通道 1 配置为输入，IS1 映射在 CI1FE1 上<br>10：通道 1 配置为输入，IS1 映射在 CI0FE1 上<br>11：通道 1 配置为输入，IS1 映射在 ITS 上<br>注意：当CH1MS[1:0]=11时，需要通过TRGS位(位于TIMERx_SMCFG寄存器)选择内部触发输入。 |
| 7 | CH0COMCEN | 通道 0 输出比较清 0 使能<br>当此位被置 1，当检测到 ETIFP 信号输入高电平时，O0CPRE 参考信号被清 0<br>0：禁止通道 0 输出比较清零<br>1：使能通道0输出比较清零 |
| 6:4 | CH0COMCTL[2:0] | 通道 0 输出比较模式<br>此位定义了输出准备信号 O0CPRE 的输出比较模式，而 O0CPRE 决定了 CH0_O、CH0_ON 的值。另外，O0CPRE 高电平有效，而 CH0_O、CH0_ON 通道的极性取决于 CH0P、CH0NP 位。<br>000：时基。输出比较寄存器 TIMERx_CH0CV 与计数器 TIMERx_CNT 间的比较对 O0CPRE 不起作用<br>001：匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为高。<br>010：匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为低。<br>011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 翻转。<br>100：强制为低。强制 O0CPRE 为低电平<br>101：强制为高。强制 O0CPRE 为高电平<br>110：PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。<br>111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。<br>如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，O0CPRE 电平才改变。 |
| 3 | CH0COMSEN | 通道 0 输出比较影子寄存器使能<br>当此位被置 1，TIMERx_CH0CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。<br>0：禁止通道 0 输出/比较影子寄存器 |

1：使能通道 0 输出/比较影子寄存器
仅在单脉冲模式下(SPM =1)，可以在未确认影子寄存器的情况下使用 PWM 模式

2 CH0COMFEN 通道 0 输出比较快速使能
当该位为 1 时，如果通道配置为 PWM0 模式或者 PWM1 模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH0_O 被设置为比较电平而与比较结果无关。
0：禁止通道 0 输出比较快速.
1：使能通道0输出比较快速。

1:0 CH0MS[1:0] 通道 0 I/O 模式选择
这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH0EN 位被清 0)时这些位才可写。
00：通道 0 配置为输出
01：通道 0 配置为输入，IS0 映射在 CI0FE0 上
10：通道 0 配置为输入，IS0 映射在 CI1FE0 上
11：通道 0 配置为输入，IS0 映射在 ITS 上
注意：当CH0MS[1:0]=11时，需要通过TRGS位(位于TIMERx_SMCFG寄存器)选择内部触发输入

**输入捕获模式：**

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:12 | CH1CAPFLT[3:0] | 通道1输入捕获滤波控制<br>参见CH0CAPFLT描述 |
| 11:10 | CH1CAPPSC[1:0] | 通道1输入捕获预分频器<br>参见CH0CAPPSC描述 |
| 9:8 | CH1MS[1:0] | 通道1模式选择<br>与输出模式相同 |
| 7:4 | CH0CAPFLT[3:0] | 通道 0 输入捕获滤波控制<br>CI0 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。<br>数字滤波器的基本原理：根据 $f_{SAMP}$ 对 CI0 输入信号进行连续采样，并记录信号相同电平的次数。达到该位配置的滤波参数后，认为是有效电平。<br>滤波器参数配置如下： |

| CH0CAPFLT [3:0] | 采样次数 | $f_{SAMP}$ |
|---|---|---|
| 4'b0000 | 无滤波器 | |
| 4'b0001 | 2 | $f_{CK_TIMER}$ |
| 4'b0010 | 4 | |
| 4'b0011 | 8 | |
| 4'b0100 | 6 | $f_{DTS}/2$ |
| 4'b0101 | 8 | |
| 4'b0110 | 6 | $f_{DTS}/4$ |

| | | |
|---|---|---|
| 4'b0111 | 8 | |
| 4'b1000 | 6 | $f_{DTS}/8$ |
| 4'b1001 | 8 | |
| 4'b1010 | 5 | $f_{DTS}/16$ |
| 4'b1011 | 6 | |
| 4'b1100 | 8 | |
| 4'b1101 | 5 | $f_{DTS}/32$ |
| 4'b1110 | 6 | |
| 4'b1111 | 8 | |

3:2 CH0CAPPSC[1:0] 通道0输入捕获预分频器

这2位定义了通道0输入的预分频系数。当TIMERx_CHCTL2寄存器中的CH0EN =0时，则预分频器复位。

00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获。

01：每2个事件触发一次捕获。

10：每4个事件触发一次捕获。

11：每8个事件触发一次捕获。

1:0 CH0MS[1:0] 通道0模式选择

与输出比较模式相同

### 通道控制寄存器 1 (TIMERx_CHCTL1)

地址偏移：0x1C

复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CH3COMCEN | CH3COMCTL[2:0] | | | CH3COMSEN | CH3COMFEN | CH3MS[1:0] | | CH2COMCEN | CH2COMCTL[2:0] | | | CH2COMSEN | CH2COMFEN | CH2MS[1:0] | |
| CH3CAPFLT[3:0] | | | | CH3CAPPSC[1:0] | | | | CH2CAPFLT[3:0] | | | | CH2CAPPSC[1:0] | | | |
| rw | | | | rw | | rw | | rw | | | | rw | | rw | |

**输出比较模式：**

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15 | CH3COMCEN | 通道3输出比较清0使能<br>参见CH0COMCEN描述 |
| 14:12 | CH3COMCTL[2:0] | 通道3输出比较模式<br>参见CH0COMCTL描述 |
| 11 | CH3COMSEN | 通道3输出比较影子寄存器使能 |

| | | |
|---|---|---|
| | | 参见CH0COMSEN描述 |
| 10 | CH3COMFEN | 通道3输出比较快速使能<br>参见CH0COMFEN描述 |
| 9:8 | CH3MS[1:0] | 通道 3 模式选择<br>这些位定义了通道的方向和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH3EN 位被清 0)时这些位才可以写。<br>00：通道 3 配置为输出<br>01：通道 3 配置为输入，IS3 映射在 CI3FE3 上<br>10：通道 3 配置为输入，IS3 映射在 CI2FE3 上<br>11：通道 3 配置为输入，IS3 映射在 ITS 上<br>注意：当CH3MS[1:0]=11时，需要通过TRGS位(位于TIMERx_SMCFG寄存器)选择内部触发输入 |
| 7 | CH2COMCEN | 通道 2 输出比较清 0 使能<br>当此位被置 1，当检测到 ETIFP 输入高电平时，O2CPRE 参考信号被清 0<br>0：使能通道 2 输出比较清零<br>1：禁止通道2输出比较清零 |
| 6:4 | CH2COMCTL[2:0] | 通道 2 输出比较模式<br>此位定义了输出准备信号 O2CPRE 的输出比较模式，而 O2CPRE 决定了 CH2_O、CH2_ON 的值。另外，O2CPRE 高电平有效，而 CH2_O、CH2_ON 通道的极性取决于 CH2P、CH2NP 位。<br>000：时基。输出比较寄存器 TIMERx_CH2CV 与计数器 TIMERx_CNT 间的比较对 O2CPRE 不起作用<br>001：匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH2CV 相同时，强制 O2CPRE 为高。<br>010：匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH2CV 相同时，强制 O2CPRE 为低。<br>011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH2CV 相同时，强制 O2CPRE 翻转。<br>100：强制为低。强制 O2CPRE 为低电平<br>101：强制为高。强制 O2CPRE 为高电平<br>110：PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_CH2CV 时，O2CPRE 为高电平，否则为低电平。在向下计数时，一旦计数器的值大于 TIMERx_CH2CV 时，O2CPRE 为低电平，否则为高电平。<br>111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH2CV 时，O2CPRE 为低电平，否则为高电平。在向下计数时，一旦计数器的值大于 TIMERx_CH2CV 时，O2CPRE 为高电平，否则为低电平。<br>如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，O2CPRE 电平才改变。 |
| 3 | CH2COMSEN | 通道 0 输出比较影子寄存器使能<br>当此位被置 1，TIMERx_CH2CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。<br>0：禁止通道 2 输出/比较影子寄存器 |

1：使能通道 2 输出/比较影子寄存器
仅在单脉冲模式下(SPM =1)，可以在未确认影子寄存器情况下使用 PWM 模式

2 CH2COMFEN 通道 2 输出比较快速使能
当该位为 1 时，如果通道配置为 PWM0 模式或者 PWM1 模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH2_O 被设置为比较电平而与比较结果无关。
0：禁止通道 2 输出比较快速.
1：使能通道2输出比较快速。

1:0 CH2MS[1:0] 通道 2 I/O 模式选择
这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH2EN 位被清 0)时这些位才可写。
00：通道 2 配置为输出
01：通道 2 配置为输入，IS2 映射在 CI2FE2 上
10：通道 2 配置为输入，IS2 映射在 CI3FE2 上
11：通道 2 配置为输入，IS2 映射在 ITS 上.
注意：当CH2MS[1:0]=11时，需要通过TRGS位(位于TIMERx_SMCFG寄存器)选择内部触发输入

**输入捕获模式：**

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:12 | CH3CAPFLT[3:0] | 通道3输入捕获滤波控制<br>参见CH0CAPFLT描述 |
| 11:10 | CH3CAPPSC[1:0] | 通道3输入捕获预分频器<br>参见CH0CAPPSC描述 |
| 9:8 | CH3MS[1:0] | 通道3模式选择<br>与输出模式相同 |
| 7:4 | CH2CAPFLT[3:0] | 通道 2 输入捕获滤波控制<br>CI2 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。<br>数字滤波器的基本原理：根据 $f_{SAMP}$ 对 CI2 输入信号进行连续采样，并记录信号相同电平的次数。达到该位配置的滤波参数后，认为是有效电平。<br>滤波器参数配置如下： |

| CH2CAPFLT [3:0] | 采样次数 | $f_{SAMP}$ |
|---|---|---|
| 4'b0000 | 无滤波器 | |
| 4'b0001 | 2 | $f_{CK_TIMER}$ |
| 4'b0010 | 4 | |
| 4'b0011 | 8 | |
| 4'b0100 | 6 | $f_{DTS}/2$ |
| 4'b0101 | 8 | |
| 4'b0110 | 6 | $f_{DTS}/4$ |

| | | |
|---|---|---|
| 4'b0111 | 8 | |
| 4'b1000 | 6 | $f_{DTS}/8$ |
| 4'b1001 | 8 | |
| 4'b1010 | 5 | $f_{DTS}/16$ |
| 4'b1011 | 6 | |
| 4'b1100 | 8 | |
| 4'b1101 | 5 | $f_{DTS}/32$ |
| 4'b1110 | 6 | |
| 4'b1111 | 8 | |

3:2 CH2CAPPSC[1:0] 通道2输入捕获预分频器

这2位定义了通道2输入的预分频系数。当TIMERx_CHCTL2寄存器中的CH2EN =0时，则预分频器复位。

00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获。

01：每2个事件触发一次捕获。

10：每4个事件触发一次捕获。

11：每8个事件触发一次捕获。

1:0 CH2MS[1:0] 通道2模式选择

与输出比较模式相同

### 通道控制寄存器 2 (TIMERx_CHCTL2)

地址偏移：0x20

复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | CH3P | CH3EN | 保留 | | CH2P | CH2EN | 保留 | | CH1P | CH1EN | 保留 | | CH0P | CH0EN |
| | | 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 | CH3P | 通道3极性<br>参考CH0P描述 |
| 12 | CH3EN | 通道3使能<br>参考CH0EN 描述 |
| 11:10 | 保留 | 必须保持复位值 |
| 9 | CH2P | 通道2极性 |

| | | |
|---|---|---|
| | | 参考CH0P 描述 |
| 8 | CH2EN | 通道2使能<br>参考CH0EN描述 |
| 7:6 | 保留 | 必须保持复位值 |
| 5 | CH1P | 通道1极性<br>参考CH0P 描述 |
| 4 | CH1EN | 通道1使能<br>参考CH0EN 描述 |
| 3:2 | 保留 | 必须保持复位值 |
| 1 | CH0P | 通道0极性<br>当通道0配置为输出模式时，此位定义了输出信号极性。<br>0：通道0高电平为有效电平<br>1：通道0低电平为有效电平<br>当通道0配置为输入模式时，此位定义了CI0信号极性<br>CH0P将选择CI0FE0或者CI1FE0的有效边沿或者捕获极性<br>CH0P=0：把CIxFE0的上升沿作为捕获或者从模式下触发的有效信号，并且CIxFE0不会被翻转。<br>CH0P=1：把CIxFE0的下降沿作为捕获或者从模式下触发的有效信号，并且CIxFE0会被翻转。 |
| 0 | CH0EN | 通道0捕获/比较使能<br>当通道0配置为输出模式时，将此位置1使能CH0_O信号有效。当通道0配置为输入模式时，将此位置1使能通道0上的捕获事件。<br>0：禁止通道0<br>1：使能通道0 |

### 计数器寄存器 (TIMERx_CNT)

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

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

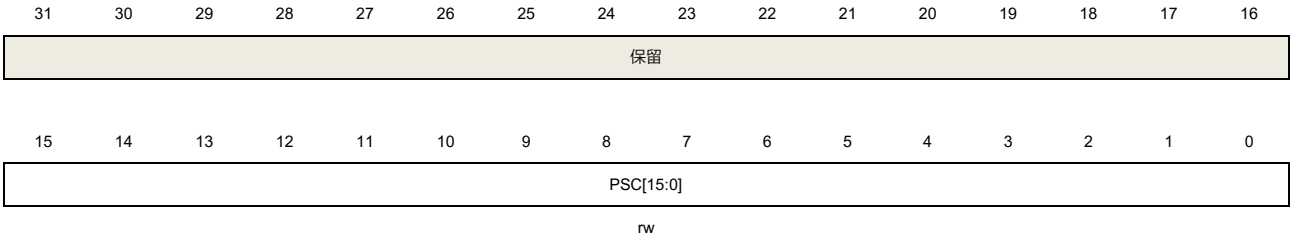
| 位/位域 | 名称 | 描述 |
|---|---|---|

| | | |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | CNT[15:0] | 这些位是当前的计数值。写操作能改变计数器值。 |

### 预分频寄存器 (TIMERx_PSC)

地址偏移：0x28
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

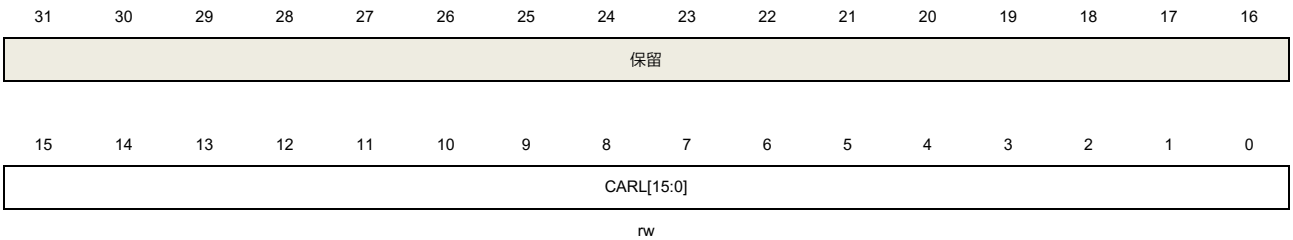



| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | PSC[15:0] | 计数器时钟预分频值<br>计数器时钟等于TIMER_CK时钟除以(PSC+1)，每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。 |

### 计数器自动重载寄存器 (TIMERx_CAR)

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | CARL[15:0] | 计数器自动重载值<br>这些位定义了计数器的自动重载值。 |

### 通道 0 捕获/比较值寄存器 (TIMERx_CH0CV)

地址偏移：0x34
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CH0VAL[15:0] | | | | | | | | | | | | | | | |
| rw | | | | | | | | | | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | CH0VAL[15:0] | 通道0的捕获或比较值<br>当通道0配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。<br>当通道0配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。 |

### 通道 1 捕获/比较值寄存器 (TIMERx_CH1CV)

地址偏移：0x38
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CH1VAL[15:0] | | | | | | | | | | | | | | | |
| rw | | | | | | | | | | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | CH1VAL[15:0] | 通道1的捕获或比较值<br>当通道1配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。<br>当通道1配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。 |

### 通道 2 捕获/比较值寄存器 (TIMERx_CH2CV)

地址偏移：0x3C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CH2VAL[15:0] | | | | | | | | | | | | | | | |
| rw | | | | | | | | | | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | CH2VAL[15:0] | 通道2的捕获或比较值<br>当通道2配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。<br>当通道2配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。 |

### 通道 3 捕获/比较值寄存器 (TIMERx_CH3CV)

地址偏移：0x40
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

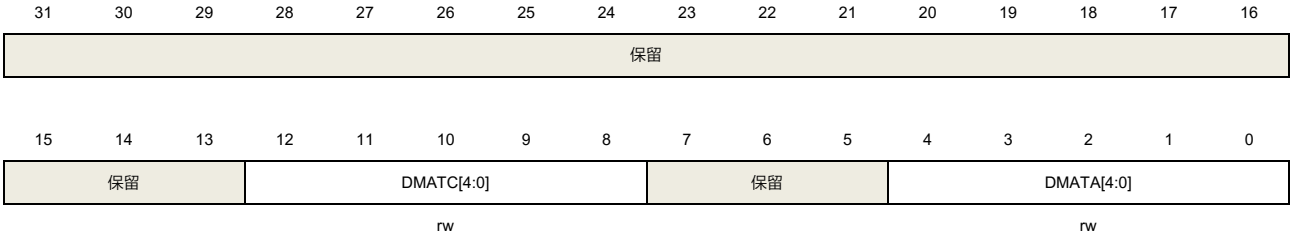
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CH3VAL[15:0] | | | | | | | | | | | | | | | |
| rw | | | | | | | | | | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | CH3VAL[15:0] | 通道3的捕获或比较值<br>当通道3配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。<br>当通道3配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。 |

### DMA 配置寄存器 (TIMERx_DMACFG)

地址偏移：0x48
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

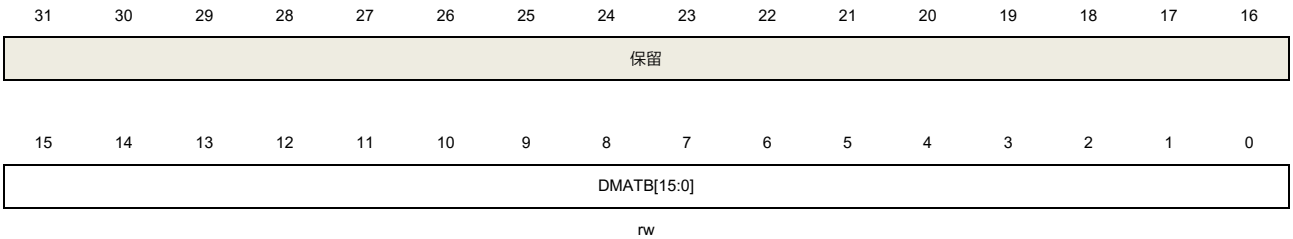



| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:13 | 保留 | 必须保持复位值 |
| 12:8 | DMATC[4:0] | DMA传输计数<br>该位域定义了DMA访问(读写)TIMERx_DMATB寄存器的数量n，n = (DMATC [4:0] +1). DMATC [4:0] 从 5'b0_0000 到 5'b1_0001 |
| 7:5 | 保留 | 必须保持复位值 |
| 4:0 | DMATA[4:0] | DMA传输起始地址<br>该位域定义了DMA访问TIMERx_DMATB寄存器的第一个地址。当通过TIMERx_DMATB第一次访问时，访问的就是该位域指定的地址。第二次访问TIMERx_DMATB时，将访问起始地址+0x4。 |

### DMA 发送缓冲区寄存器 (TIMERx_DMATB)

地址偏移：0x4C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



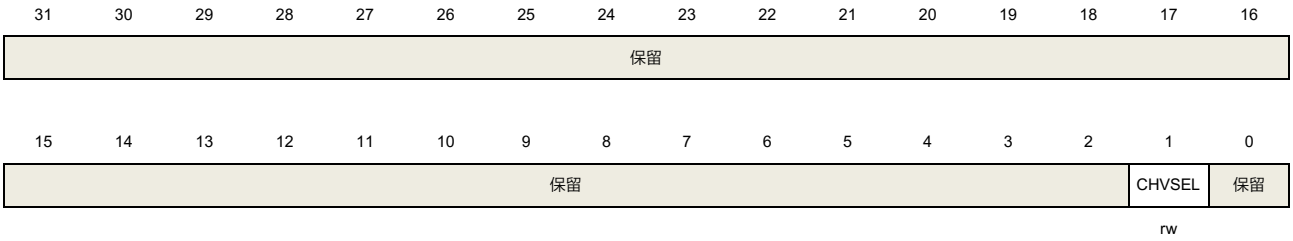

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | DMATB[15:0] | DMA发送缓冲<br>对这个寄存器的读或写，(起始地址+传输次数*4)地址范围内的寄存器会被访问 |

传输次数由硬件计算，范围为0到DMATC。

### 配置寄存器 (TIMERx_CFG)

地址偏移：0xFC
复位值：0x0000 0000

该寄存器只能按字(32位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:2 | 保留 | 必须保持复位值 |
| 1 | CHVSEL | 写捕获比较寄存器选择位<br>此位由软件写1或清0。<br>1：当写入捕获比较寄存器的值与寄存器当前值相等时，写入操作无效<br>0：无影响 |
| 0 | 保留 | 必须保持复位值 |

# 16.3. 通用定时器 L1(TIMERx, x=8,11)

## 16.3.1. 简介

通用定时器L1(TIMER8/11)是两通道定时器，支持输入捕获和输出比较，可以产生PWM信号控制电机和电源管理。通用定时器L1含有一个16位无符号计数器。

通用定时器L1是可编程的，可以用来计数，其外部事件可以驱动其他定时器

定时器和定时器之间是相互独立，但是它们的计数器可以被同步在一起形成一个更大的定时器。

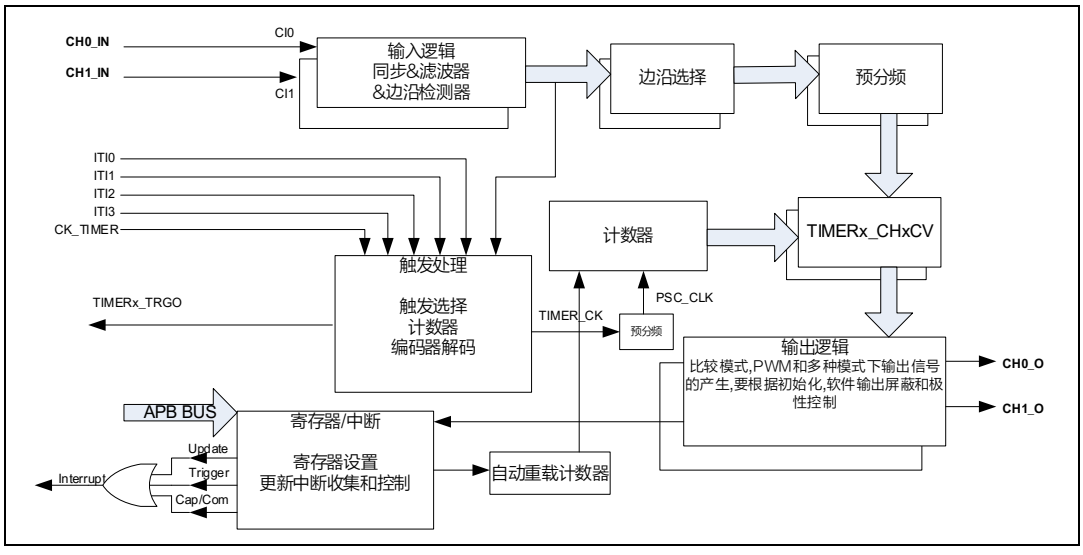
## 16.3.2. 主要特征

- 总通道数：2；
- 计数器宽度：16位；
- 时钟源可选：内部时钟，内部触发，外部输入；
- 计数模式：向上计数；
- 可编程的预分频器：16位，运行时可以被改变；
- 每个通道可配置：输入捕获模式，输出比较模式，可编程的PWM模式，单脉冲模式；
- 自动重装载功能；
- 中断输出：更新事件，触发事件，比较/捕获事件和中止事件；
- 多个定时器的菊链使得一个定时器可以同时启动多个定时器；
- 定时器的同步允许被选择的定时器在同一个时钟周期开始计数；
- 定时器主-从管理。

## 16.3.3. 结构框图

*图16-44. 通用定时器L1结构框图*提供了通用定时器L1的内部配置细节。

**图 16-44. 通用定时器 L1 结构框图**



## 16.3.4. 功能描述

### 时钟源配置

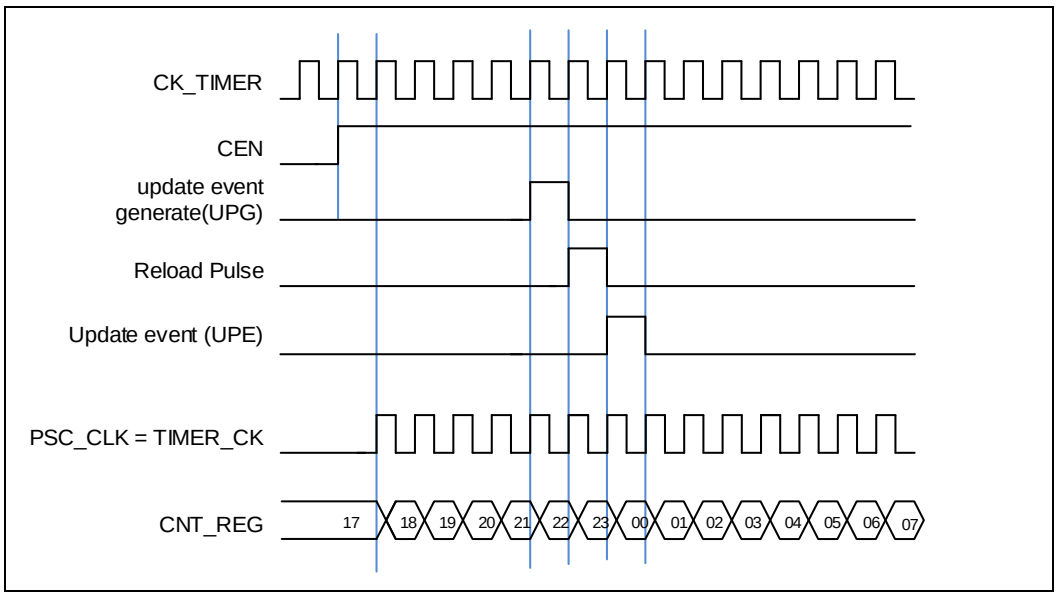
通用定时器L1可以由内部时钟源CK_TIMER或者由SMC(TIMERx_SMCFG寄存器位[2:0])控制的复用时钟源驱动。

■ SMC[2:0]==3'b000，定时器选择内部时钟源(连接到RCU模块的CK_TIMER)

如果 SMC[2:0]==3'b000，默认用来驱动计数器预分频器的是内部时钟源 CK_TIMER。当 CEN 置位，CK_TIMER 经过预分频器(预分频值由 TIMERx_PSC 寄存器确定)产生 PSC_CLK。

如果将 TIMERx_SMCFG 寄存器的 SMC[2:0]设置为 0x1、0x2、0x3 和 0x7，预分频器被其他时钟源(由 TIMERx_SMCFG 寄存器的 TRGS [2:0]区域选择)驱动，在下文说明。当 SMC 位被设置为 0x4、0x5 和 0x6，计数器预分频器时钟源由内部时钟 CK_TIMER 驱动。

**图 16-45. 内部时钟分频为 1 时，计数器的时序图**




■ SMC[2:0]==3'b111(外部时钟模式0)，定时器选择外部输入引脚作为时钟源

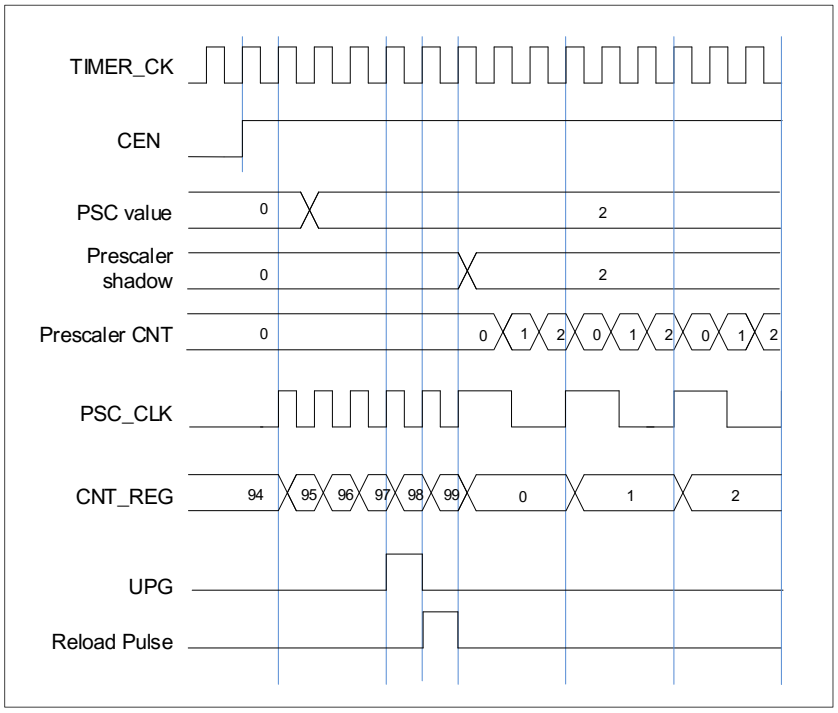
计数器预分频器可以在 TIMERx_CI0/ TIMERx_CI1 引脚的每个上升沿或下降沿计数。这种模式可以通过设置 SMC [2:0]为 0x7 同时设置 TRGS [2:0]为 0x4，0x5 或 0x6 来选择。CIx 是 TIMERx_CIx 通过数字滤波器采样后的信号。

计数器预分频器也可以在内部触发信号 ITI0/1/2/3 的上升沿计数。这种模式可以通过设置 SMC [2:0]为 0x7 同时设置 TRGS [2:0]为 0x0, 0x1, 0x2 或者 0x3。

### 时钟预分频器

预分频器可以将定时器的时钟(TIMER_CK)频率按 1 到 65536 之间的任意值分频，分频后的时钟 PSC_CLK 驱动计数器计数。分频系数受预分频寄存器 TIMERx_PSC 控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

**图 16-46. 当 PSC 数值从 0 变到 2 时，计数器的时序图**




## 计数器向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从 0 开始向上连续计数到自动加载值(定义在 TIMERx_CAR 寄存器中)，一旦计数器计数到自动加载值，会重新从 0 开始向上计数并产生上溢事件。在向上计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 0。

当通过 TIMERx_SWEVG 寄存器的 UPG 位置 1 来设置更新事件时，计数值会被清 0，并产生更新事件。

如果 TIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(计数器自动重载寄存器，预分频寄存器)都将被更新。

下面一些图给出了一些例子，当 TIMERx_CAR=0x99 时，计数器在不同预分频因子下的行为。

图 16-47. 向上计数时序图，PSC=0/2

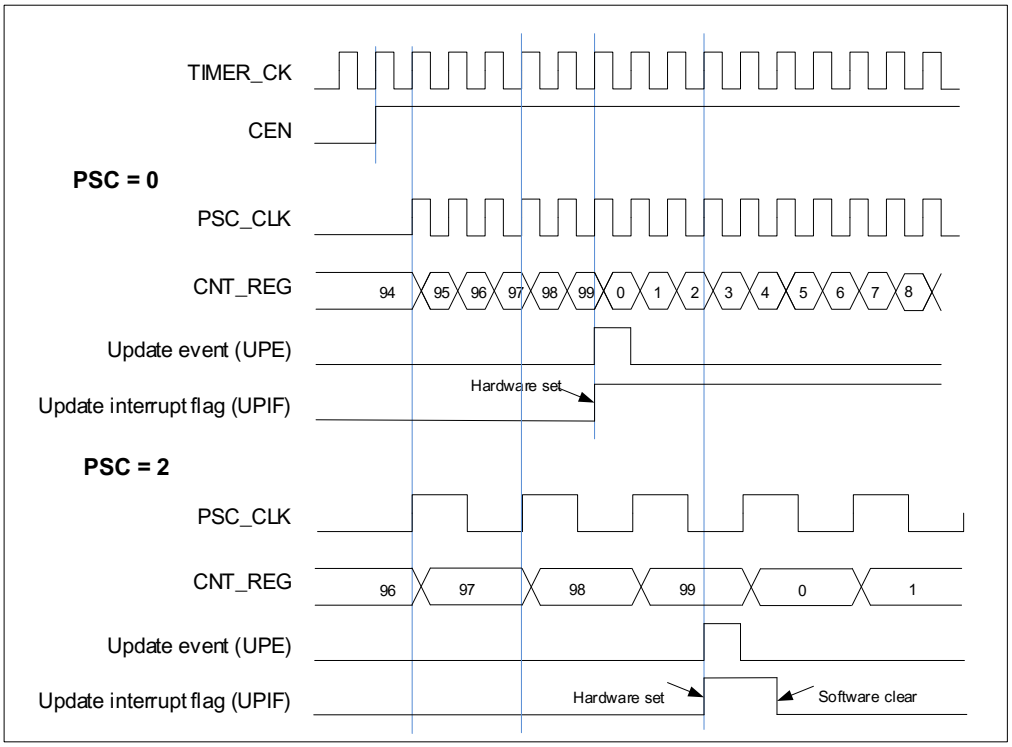
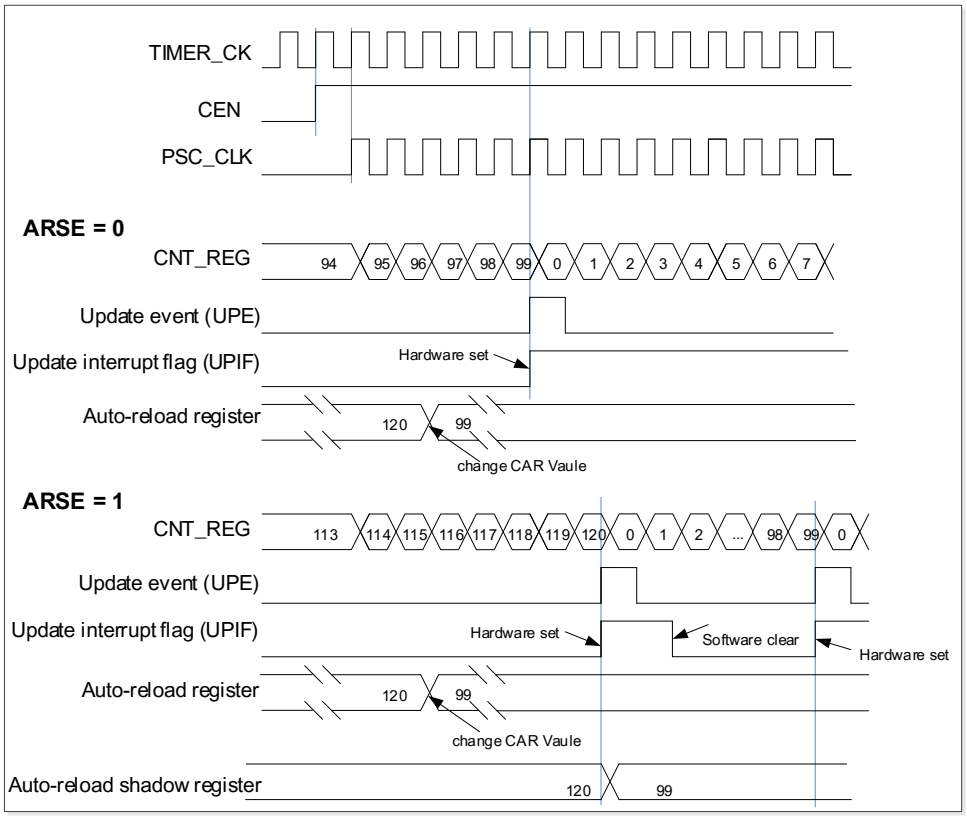



图 16-48. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



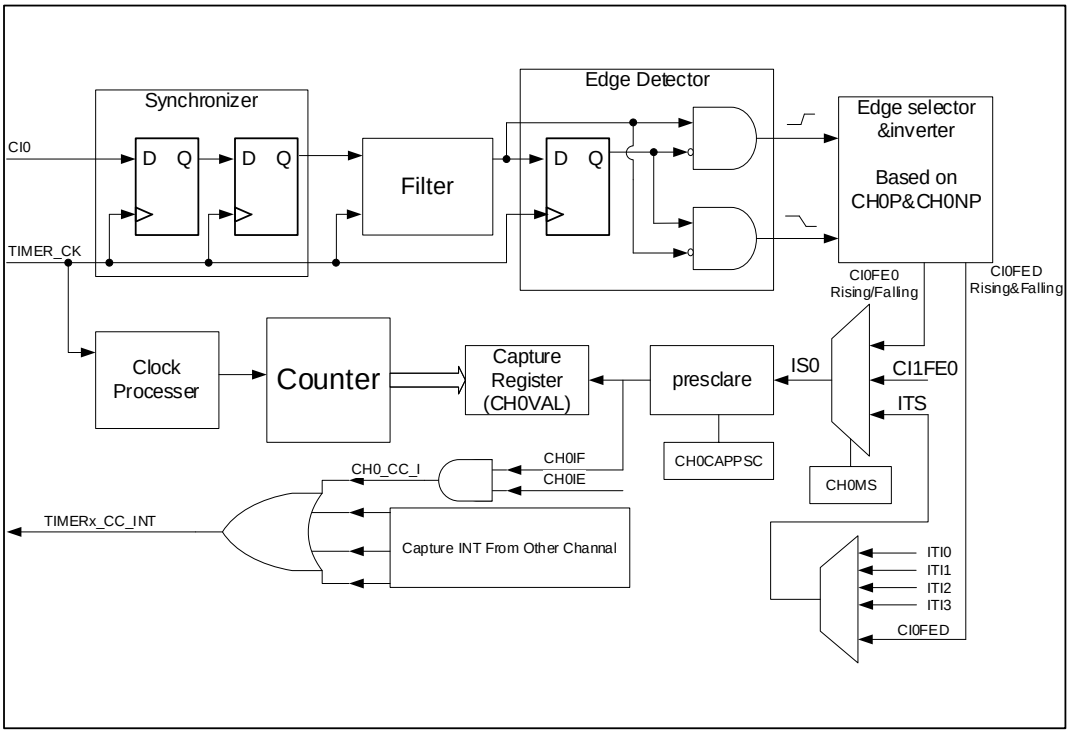
### 输入捕获和输出比较通道

通用定时器 L1 拥有两个独立的通道用于捕获输入或比较输出是否匹配。每个通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

■ 输入捕获功能

通道输入捕获功能允许通道测量一个波形时序，频率，周期，占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，TIMERx_CHxCV 寄存器会捕获计数器当前的值，同时 CHxIF 位被置 1，如果 CHxIE = 1 则产生通道中断。

**图 16-49. 输入捕获原理**




通道输入信号 CIx 先被 TIMER_CK 信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置 CHxP 选择使用上升沿或者下降沿。配置 CHxMS.，可以选择其他通道的输入信号，内部触发信号。配置 IC 预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生，TIMERx_CHxCV 存储计数器的值。

**第一步：** 滤波器配置(TIMERx_CHCTL0寄存器中CHxCAPFLT)：
根据输入信号和请求信号的质量，配置相应的CHxCAPFLT。

**第二步：** 边沿选择(TIMERx_CHCTL2寄存器中CHxP/CHxNP)：
配置CHxP/CHxNP选择上升沿或者下降沿。

**第三步：** 捕获源选择(TIMERx_CHCTL0寄存器中CHxMS)：
一旦通过配置CHxMS选择输入捕获源，必须确保通道配置在输入模式(CHxMS!=0x0)，而且TIMERx_CHxCV寄存器不能再被写。

**第四步：** 中断使能(TIMERx_DMAINTEN寄存器中CHxIE 和 CHxDEN)：

使能相应中断，可以获得中断和DMA请求。

**第五步：**捕获使能(TIMERx_CHCTL2寄存器中CHxEN)。

**结果：**当期望的输入信号发生时，TIMERx_CHxCV被设置成当前计数器的值，CHxIF为置1。如果CHxIF位已经为1，则CHxOF位置1。根据TIMERx_DMAINTEN寄存器中CHxIE和CHxDEN的配置，相应的中断和DMA请求会被提出。

**直接产生：**软件设置CHxG位，会直接产生中断和DMA请求。

通道输入捕获功能可用来测量 TIMERx_CHx 引脚上信号的脉冲波宽度。例如，一个 PWM 波连接到 CI0。配置 TIMERx_CHCTL0 寄存器中 CH0MS 为 2'b01，选择通道 0 的捕获信号为 CI0 并设置上升沿捕获。配置 TIMERx_CHCTL0 寄存器中 CH1MS 为 2'b10，选择通道 1 捕获信号为 CI0 并设置下降沿捕获。计数器配置为复位模式，在通道 0 的上升沿复位。TIMERX_CH0CV 寄存器测量 PWM 的周期值，TIMERx_CH1CV 寄存器测量 PWM 占空比值。

■ 通道输出比较功能

在通道输出比较功能，TIMERx 可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的 TIMERx_CHxCV 寄存器与计数器的值匹配时，根据 CHxCOMCTL 的配置，这个通道的输出可以被置高电平，被置低电平或者反转。当计数器的值与 TIMERx_CHxCV 寄存器的值匹配时，CHxIF 位被置 1，如果 CHxIE = 1 则会产生中断，如果 CxCDE=1 则会产生 DMA 请求。

配置步骤如下：

**第一步：**时钟配置：

配置定时器时钟源，预分频器等。

**第二步：**比较模式配置：

设置CHxCOMSEN位来配置输出比较影子寄存器；

设置CHxCOMCTL位来配置输出模式(置高电平/置低电平/反转)；

设置CHxP/CHxNP位来选择有效电平的极性；

设置CHxEN使能输出。

**第三步：**通过CHxIE/CxCDE位配置中断/DMA请求使能。

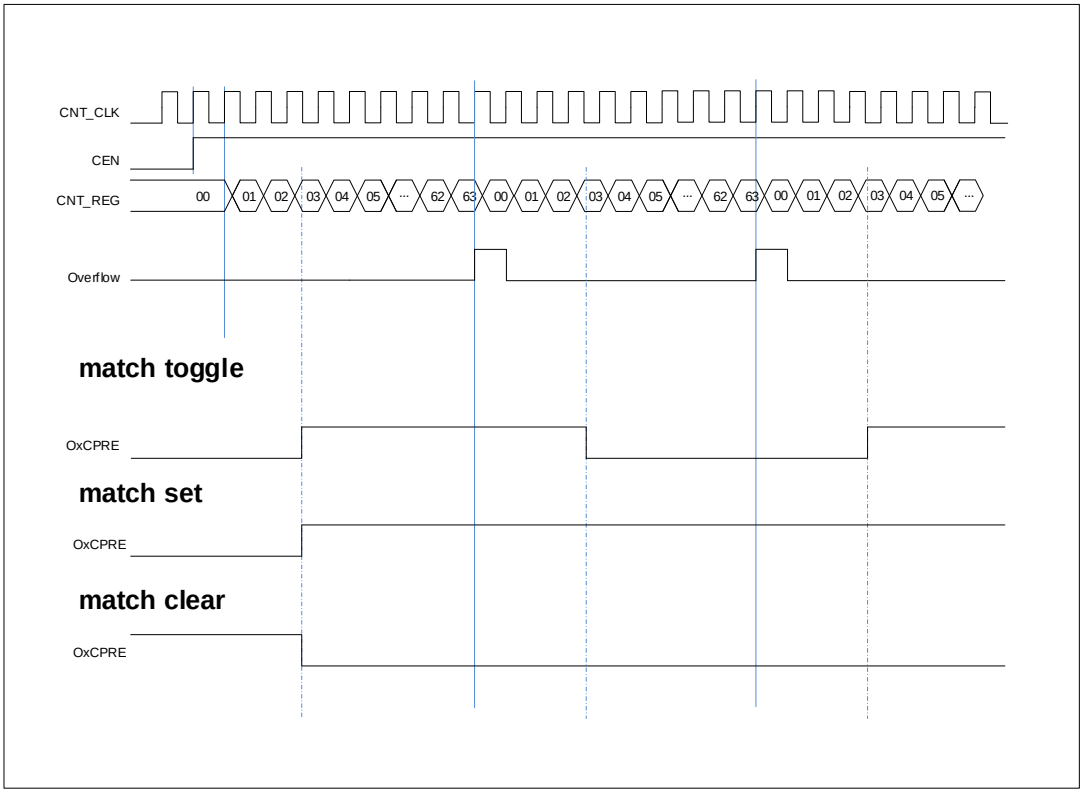
**第四步：**通过TIMERx_CAR寄存器和TIMERx_CHxCV寄存器配置输出比较时基：

TIMERx_CHxCV可以在运行时根据你所期望的波形而改变。

**第五步：**设置CEN位使能定时器。

*图16-50. 三种输出比较模式*显示了三种比较输出模式：反转/置高电平/置低电平，CAR=0x63，CHxVAL=0x3。

图 16-50. 三种输出比较模式




## 输出 PWM 功能

在 PWM 输出模式下(PWM 模式 0 是配置 CHxCOMCTL 为 3'b110， PWM 模式 1 是配置 CHxCOMCTL为3'b111)，通道根据TIMERx_CAR寄存器和TIMERx_CHxCV寄存器的值，输出PWM波形。

根据计数模式，我们可以分为两种PWM波：EAPWM(边沿对齐PWM)和CAPWM(中央对齐PWM)。

EAPWM的周期由TIMERx_CAR寄存器值决定，占空比由TIMERx_CHxCV寄存器值决定。*图 16-51. EAPWM时序图*显示了EAPWM的输出波形和中断。

CAPWM的周期由(2*TIMERx_CAR寄存器值)决定，占空比由(2*TIMERx_CHxCV寄存器值)决定。*图16-52. CAPWM时序图*显示了CAPWM的输出波形和中断。

在 PWM0 模式下 (CHxCOMCTL==3'b110)， 如果 TIMERx_CHxCV 寄存器的值大于TIMERx_CAR寄存器的值，通道输出一直为有效电平。

在PWM0模式下(CHxCOMCTL==3'b110)，如果TIMERx_CHxCV寄存器的值等于0，通道输出一直为无效电平。

图 16-51. EAPWM 时序图

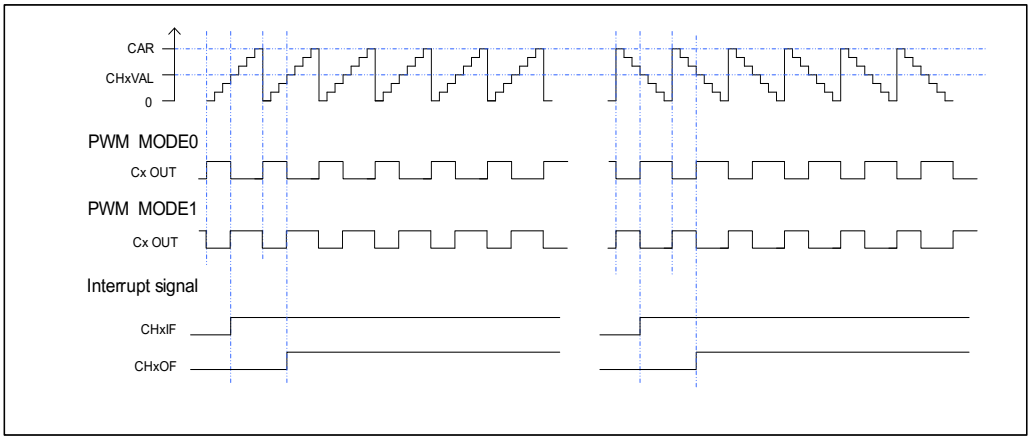
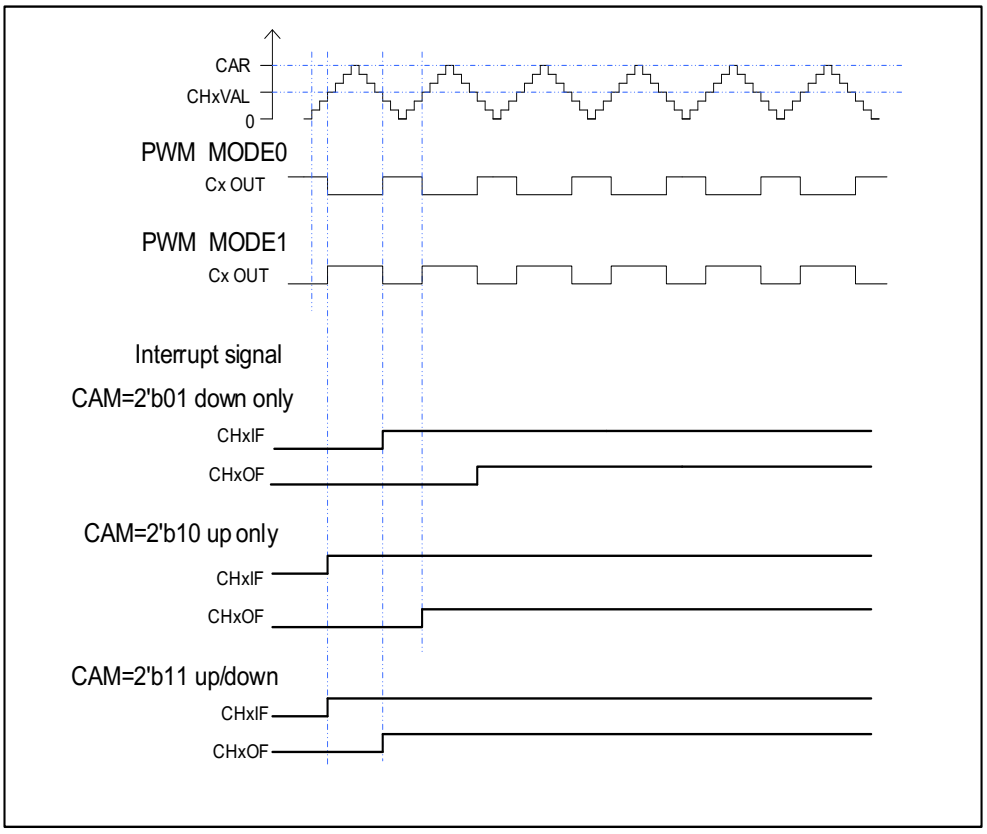



图 16-52. CAPWM 时序图




## 通道输出准备信号

当 TIMERx 用于输出匹配比较模式下，设置 CHxCOMCTL 位可以定义 OxCPRE 信号(通道 x 准备信号)类型。OxCPRE 信号有若干类型的输出功能，包括，设置 CHxCOMCTL=0x00 可以保持原始电平；设置 CHxCOMCTL=0x01 可以将 OxCPRE 信号设置为高电平；设置 CHxCOMCTL=0x02 可以将 OxCPRE 信号设置为低电平；设置 CHxCOMCTL=0x03，在计数器值和 TIMERx_CHxCV 寄存器的值匹配时，可以翻转输出信号。

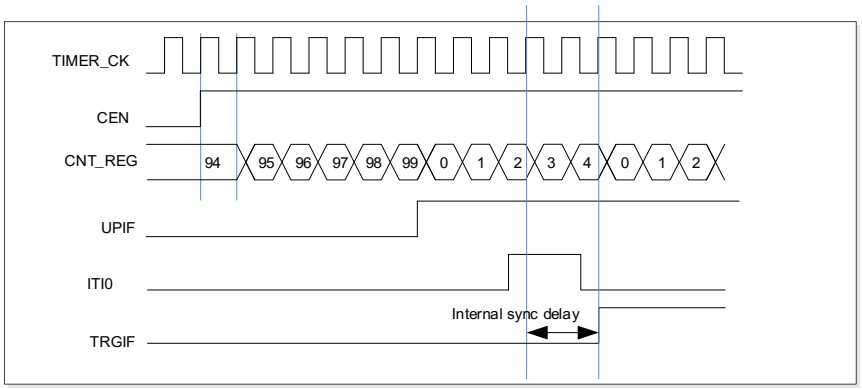
PWM 模式 0 和 PWM 模式 1 是 OxCPRE 的另一种输出类型，设置 CHxCOMCTL 位域位 0x06 或 0x07 可以配置 PWM 模式 0/PWM 模式 1。在这些模式中，根据计数器值和 TIMERx_CHxCV 寄存器值的关系以及计数方向，OxCPRE 信号改变其电平。具体细节描述，请参考相应的位。

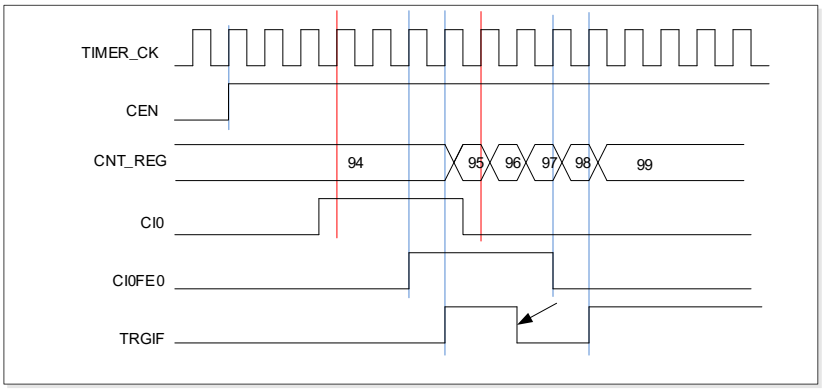
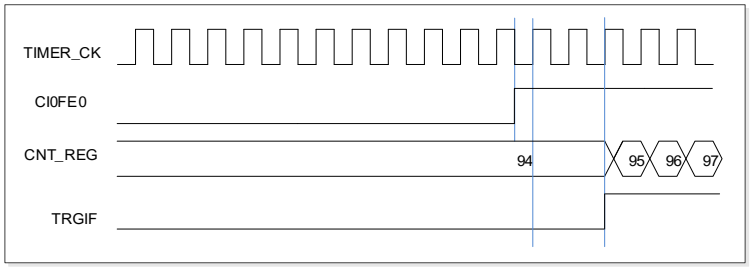
设置 CHxCOMCTL =0x04 或 0x05 可以实现 OxCPRE 信号的强制输出功能。输出比较信号能够直接由软件强置为有效或无效状态，而不依赖于 TIMERx_CHxCV 的值和计数器值之间间的比较结果。

### 主-从管理

TIMERx 能在多种模式下同步外部触发，包括复位模式，暂停模式和事件模式，可以通过设置 TIMERx_SMCFG 寄存器中的 SMC[2:0]配置这些模式。这些模式的输入触发源可以通过设置 TIMERx_SMCFG 寄存器中的 TRGS[2:0]来选择。

**表 16-6. 从机模式列表和举例**

| | 模式选择 | 触发源选择 | 极性选择 | 滤波和预分频 |
|---|---|---|---|---|
| 列举 | SMC[2:0]<br>3'b100 (复位模式)<br>3'b101 (暂停模式)<br>3'b110 (事件模式) | TRGS[2:0]<br>000: ITI0<br>001: ITI1<br>010: ITI2<br>011: ITI3<br>100: CI0F_ED<br>101: CI0FE0<br>110: CI1FE1<br>111: 保留 | 如果触发源是CI0FE0或者CI1FE1，配置CHxP和CHxNP来选择极性和反相 | 触发源ITIx，滤波和预分频不可用<br>触发源CIx，配置CHxCAPFLT设置滤波，分频不可用 |
| 例1 | 复位模式<br>当触发输入上升沿，计数器清零重启 | TRGIS[2:0]=3'b000<br>选择ITI0为触发源 | 触发源是ITI0，极性选择不可用 | 触发源是ITI0，滤波和预分频不可用 |
| | **图 16-53. 复位模式下的控制电路**<br> | | | |
| 例2 | 暂停模式<br>当触发输入为低的时候，计数器暂停计数 | TRGIS[2:0]=3'b101<br>选择CI0FE0为触发源 | TI0S=0.(非异或)<br>CH0P==0，不反相.<br>在上升沿捕获 | 在这个例子中滤波被旁路 |

| | 模式选择 | 触发源选择 | 极性选择 | 滤波和预分频 |
|---|---|---|---|---|
| | 图 16-54. 暂停模式下的控制电路   | | | |
| 例3 | 事件模式<br>触发输入的上升沿计数器开始计数 | TRGIS[2:0]=3'b101<br>选择CI0FE0为触发源 | TI0S=0.(非异或)<br>CH0P==0，不反相. | 在这个例子中滤波被旁路 |
| | 图 16-55. 事件模式下的控制电路   | | | |

## 单脉冲模式

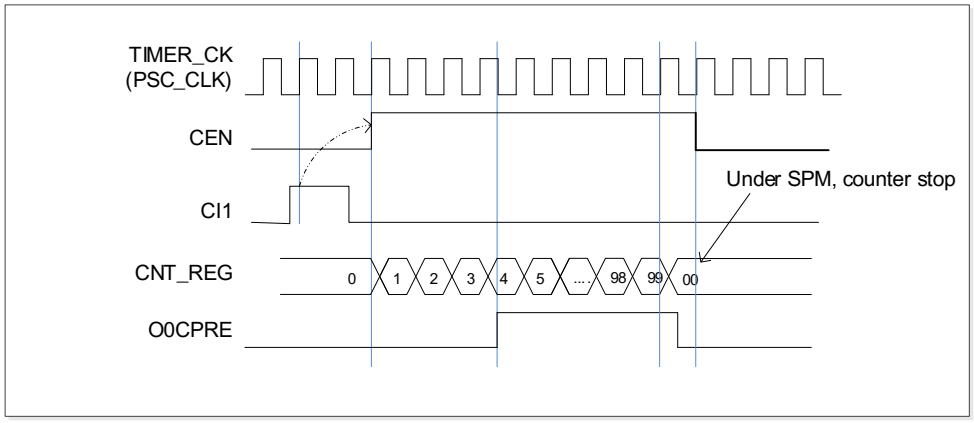
单脉冲模式与重复模式是相反的，设置 TIMERx_CTL0 寄存器的 SPM 位置 1，则使能单脉冲模式。当 SPM 置 1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置 CHxCOMCTL 配置 TIMERx 为 PWM 模式或者比较模式。

一旦设置定时器运行在单脉冲模式下，没有必要设置 TIMERx_CTL0 寄存器的定时器使能位 CEN=1 来使能计数器。触发信号沿或者软件写 CEN=1 都可以产生一个脉冲，此后 CEN 位一直保持为 1 直到更新事件发生或者 CEN 位被软件写 0。如果 CEN 位被软件清 0，计数器停止工作，计数值被保持。

在单脉冲模式下，有效的外部触发边沿会将 CEN 位置 1，使能计数器。然而，执行计数值和 TIMERx_CHxCV 寄存器值的比较结果依然存在一些时钟延迟。为了最大限度减少延迟，用户可以将 TIMERx_CHCTL0/1 寄存器的 CHxCOMFEN 位置 1。单脉冲模式下，触发上升沿产生之后， OxCPRE 信号将被立即强制转换为与发生比较匹配时相同的电平，但是不用考虑比较结果。只有输出通道配置为 PWM0 或 PWM1 输出运行模式下时 CHxCOMFEN 位才可用，触发源来源于触发信号。

*图 16-56. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99* 展示了一个例子。

**图 16-56. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99**




### 定时器互连

参考*高级定时器(TIMERx, x=0, 7)互连*

### 定时器调试模式

当Cortex®-M4内核停止，DBG_CTL0寄存器中的TIMERx_HOLD配置位被置1，定时器计数器停止

### 16.3.5. TIMERx 寄存器(x=8,11)

TIMER8基地址：0x4001 4C00
TIMER11基地址：0x4000 1800

#### 控制寄存器 0 (TIMERx_CTL0)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | CKDIV[1:0] | | ARSE | 保留 | | | SPM | UPS | UPDIS | CEN |
| | | | | | | rw | | rw | | | | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:10 | 保留 | 必须保持复位值 |
| 9:8 | CKDIV[1:0] | 时钟分频<br>通过软件配置CKDIV，规定定时器时钟(CK_TIMER) 与死区时间和数字滤波器采样时钟(DTS)之间的分频系数。<br>00：$f_{DTS}=f_{CK_TIMER}$<br>01：$f_{DTS}= f_{CK_TIMER} /2$<br>10：$f_{DTS}= f_{CK_TIMER} /4$<br>11：保留 |
| 7 | ARSE | 自动重载影子使能<br>0：禁能TIMERx_CAR寄存器的影子寄存器<br>1：使能TIMERx_CAR寄存器的影子寄存器 |
| 6:4 | 保留 | 必须保持复位值 |
| 3 | SPM | 单脉冲模式<br>0：单脉冲模式禁能，更新事件发生后，计数器继续计数。<br>1：单脉冲模式使能，在下一次更新事件发生时，计数器停止计数。 |
| 2 | UPS | 更新请求源<br>软件配置该位，选择更新事件源.<br>0：以下事件均会产生更新中断或DMA请求：<br>UPG位被置1<br>计数器溢出/下溢<br>复位模式产生的更新<br>1：下列事件会产生更新中断或DMA请求： |

| | | |
|---|---|---|
| | | 计数器溢出/下溢 |
| 1 | UPDIS | 禁止更新.<br>该位用来使能或禁能更新事件的产生<br>0：更新事件使能. 更新事件发生时，相应的影子寄存器被装入预装载值，以下事件均会产生更新事件：<br>UPG位被置1<br>计数器溢出/下溢<br>复位模式产生的更新<br>1：更新事件禁能.<br>注意：当该位被置1时，UPG位被置1或者复位模式不会产生更新事件，但是计数器和预分频器被重新初始化 |
| 0 | CEN | 计数器使能<br>0：计数器禁能<br>1：计数器使能<br>在软件将CEN位置1后，外部时钟、暂停模式和编码器模式才能工作。 |

### 从模式配置寄存器 (TIMERx_SMCFG)

地址偏移：0x08
复位值：0x0000

该寄存器可以按半字(16位)或字(32位)访问

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | MSM | TRGS[2:0] | | | Res. | SMC[2:0] | | |
| | | | | | | | | 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>该位被用来同步被选择的定时器同时开始计数。通过 TRIGI 和 TRGO，定时器被连接在一起，TRGO 用做启动事件。<br>0: 主从模式禁能<br>1: 主从模式使能 |
| 6:4 | TRGS[2:0] | 触发选择<br>该位域用来指定选择哪一个信号作为用来同步计数器的触发输入源<br>000: ITI0<br>001: ITI1<br>010: ITI2<br>011: ITI3<br>100: CI0F_ED<br>101: CI0FE0<br>110: CI1FE1 |

| | | |
|---|---|---|
| | | 111：保留<br>从模式被使能后这些位不能改 |
| 3 | 保留 | 必须保持复位值。 |
| 2:0 | SMC[2:0] | 从模式控制<br>000：关闭从模式. 如果 CEN=1，则预分频器直接由内部时钟驱动<br>001：保留<br>010：保留<br>011：保留<br>100：复位模式. 选中的触发输入的上升沿重新初始化计数器，并且产生更新事件<br>101：暂停模式. 当触发输入为高时，计数器的时钟开启。一旦触发输入变为低，则计数器时钟停止<br>110：事件模式.计数器在触发输入的上升沿启动。<br>111：外部时钟模式 0. 选中的触发输入的上升沿驱动计数器 |

### DMA 和中断使能寄存器 (TIMERx_DMAINTEN)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | | TRGIE | 保留 | | | CH1IE | CH0IE | UPIE |
| | | | | | | | | | rw | | | | rw | rw | rw |

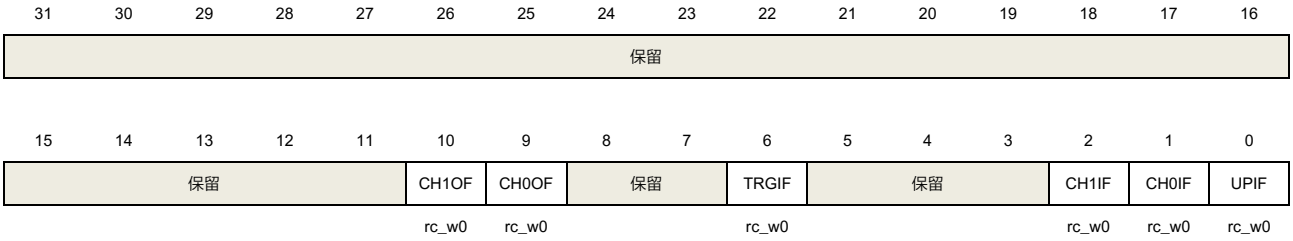
| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:7 | 保留 | 必须保持复位值 |
| 6 | TRGIE | 触发中断使能<br>0：禁止触发中断<br>1：使能触发中断 |
| 5:3 | 保留 | 必须保持复位值 |
| 2 | CH1IE | 通道1比较/捕获中断使能<br>0：禁止通道1中断<br>1：使能通道1中断 |
| 1 | CH0IE | 通道0比较/捕获中断使能<br>0：禁止通道0中断<br>1：使能通道0中断 |
| 0 | UPIE | 更新中断使能 |

0：禁止更新中断
1：使能更新中断

### 中断标志寄存器（TIMERx_INTF)

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



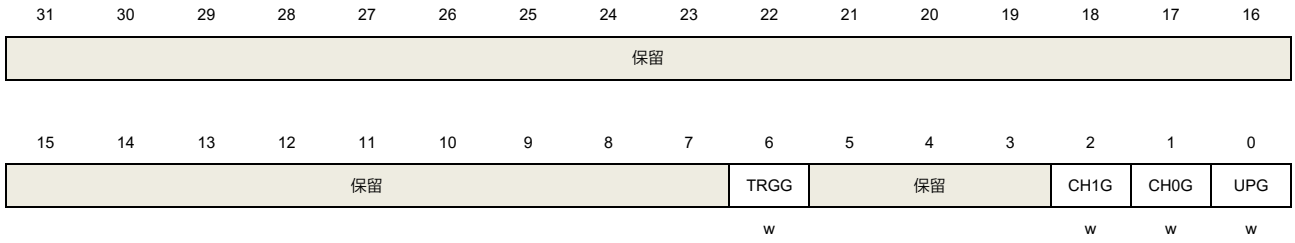

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:11 | 保留 | 必须保持复位值 |
| 10 | CH1OF | 通道1捕获溢出标志<br>参见CH0OF描述 |
| 9 | CH0OF | 通道0捕获溢出标志<br>当通道0被配置为输入模式时，在CH0IF标志位已经被置1后，捕获事件再次发生时，该标志位可以由硬件置1。该标志位由软件清0。<br>0：无捕获溢出中断发生<br>1：发生了捕获溢出中断 |
| 8:7 | 保留 | 必须保持复位值 |
| 6 | TRGIF | 触发中断标志<br>当发生触发事件时，此标志会置 1，此位由软件清 0。当暂停模式使能时，触发输入的任意边沿都可以产生触发事件。否则，其它模式时，仅在触发输入端检测到有效边沿，产生触发事件。<br>0：无触发事件产生<br>1：触发中断产生 |
| 5:3 | 保留 | 必须保持复位值 |
| 2 | CH1IF | 通道1比较/捕获中断标志<br>参见CH0IF描述 |
| 1 | CH0IF | 通道0比较/捕获中断标志<br>此标志由硬件置1软件清0。当通道0在输入模式下时，捕获事件发生时此标志位被置1；当通道0在输出模式下时，此标志位在一个比较事件发生时被置1。<br>0：无通道0中断发生 |

| | | |
|---|---|---|
| | | 1：通道0中断发生 |
| 0 | UPIF | 更新中断标志<br>此位在更新事件发生时由硬件置1，软件清0。<br>0：无更新中断发生<br>1：发生更新中断 |

### 软件事件产生寄存器 (TIMERx_SWEVG)

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字(32位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:7 | 保留 | 必须保持复位值 |
| 6 | TRGG | 触发事件产生<br>此位由软件置1，由硬件自动清0。当此位被置1，TIMERx_INTF寄存器的TRGIF标志位被置1，若开启对应的中断和DMA，则产生相应的中断和DMA传输。<br>0：无触发事件产生<br>1：产生触发事件 |
| 5:3 | 保留 | 必须保持复位值 |
| 2 | CH1G | 通道1捕获或比较事件发生<br>参见CH0G描述 |
| 1 | CH0G | 通道0捕获或比较事件发生<br>该位由软件置1，用于在通道0产生一个捕获/比较事件，由硬件自动清0。当此位被置1，CH0IF标志位被置1，若开启对应的中断和DMA，则发出相应的中断和DMA请求。此外，如果通道0配置为输入模式，计数器的当前值被捕获到TIMERx_CH0CV寄存器，如果CH0IF标志位已经为1，则CH0OF标志位被置1。<br>0：不产生通道0捕获或比较事件<br>1：发生通道0捕获或比较事件 |
| 0 | UPG | 更新事件产生<br>此位由软件置1，被硬件自动清0。当此位被置1，如果选择了中央对齐或向上计数模式，计数器被清0。否则(向下计数模式)计数器将载入自动重载值，预分频计数器将同时被清除。 |

0：无更新事件产生
1：产生更新事件

### 通道控制寄存器 0 (TIMERx_CHCTL0)

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | CH1COMCTL[2:0] | | | CH1COMSEN | CH1COMFEN | CH1MS[1:0] | | 保留 | CH0COMCTL[2:0] | | | CH0COMSEN | CH0COMFEN | CH0MS[1:0] | |
| CH1CAPFLT[3:0] | | | | CH1CAPPSC[1:0] | | | | CH0CAPFLT[3:0] | | | | CH0CAPPSC[1:0] | | | |
| rw | | | | rw | | rw | | rw | | | | rw | | rw | |

**输出比较模式：**

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:15 | 保留 | 必须保持复位值 |
| 14:12 | CH1COMCTL[2:0] | 通道1输出比较模式<br>参见CH0COMCTL描述 |
| 11 | CH1COMSEN | 通道1输出比较影子寄存器使能<br>参见CH0COMSEN描述 |
| 10 | CH1COMFEN | 通道1输出比较快速使能<br>参见CH0COMFEN描述 |
| 9:8 | CH1MS[1:0] | 通道1模式选择<br>这些位定义了通道的方向和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2寄存器的CH1EN位被清0)时这些位才可以写。<br>00：通道1配置为输出。<br>01：通道1配置为输入，IS1映射在CI1FE1上。<br>10：通道1配置为输入，IS1映射在CI0FE1上。<br>11：通道 1 配置为输入，IS1 映射在 ITS 上<br>注意：当CH1MS[1:0]=11时，需要通过TRGS位(位于TIMERx_SMCFG寄存器)选择内部触发输入。 |
| 7 | 保留 | 必须保持复位值 |
| 6:4 | CH0COMCTL[2:0] | 通道 0 输出比较模式<br>此位定义了输出准备信号 O0CPRE 的输出比较模式，而 O0CPRE 决定了 CH0_O、CH0_ON 的值。另外，O0CPRE 高电平有效，而 CH0_O、CH0_ON 通道的极性取 |

| | | |
|---|---|---|
| | | 决于 CH0P、CH0NP 位。<br>000：时基。输出比较寄存器 TIMERx_CH0CV 与计数器 TIMERx_CNT 间的比较对 O0CPRE 不起作用<br>001：匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为高。<br>010：匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为低。<br>011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 翻转。<br>100：强制为低。强制 O0CPRE 为低电平<br>101：强制为高。强制 O0CPRE 为高电平<br>110：PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。<br>111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。<br>如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，O0CPRE 电平才改变。 |
| 3 | CH0COMSEN | 通道0输出比较影子寄存器使能<br>当此位被置1，TIMERx_CH0CV寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。<br>0：禁止通道0输出/比较影子寄存器<br>1：使能通道0输出/比较影子寄存器<br>仅在单脉冲模式下(TIMERx_CTL0寄存器的SPM =1)，可以在未确认预装载寄存器情况下使用PWM 模式 |
| 2 | CH0COMFEN | 通道0输出比较快速使能<br>当该位为1时，如果通道配置为PWM模式0或者PWM模式1，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH0_O被设置为比较电平而与比较结果无关。<br>0：禁止通道0输出比较快速。<br>1：使能通道0输出比较快速。 |
| 1:0 | CH0MS[1:0] | 通道0 I/O模式选择<br>这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2寄存器的CH0EN位被清0)时这些位才可写。<br>00：通道0配置为输出。<br>01：通道0配置为输入，IS0映射在CI0FE0上。<br>10：通道0配置为输入，IS0映射在CI1FE0上。<br>11：通道 0 配置为输入，IS0 映射在 ITS 上<br>注意：当CH0MS[1:0]=11时，需要通过TRGS位(位于TIMERx_SMCFG寄存器)选择内部触发输入 |

**输入捕获模式：**

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:12 | CH1CAPFLT[3:0] | 通道1输入捕获滤波控制<br>参见CH0CAPFLT描述 |
| 11:10 | CH1CAPPSC[1:0] | 通道1输入捕获预分频器<br>参见CH0CAPPSC描述 |
| 9:8 | CH1MS[1:0] | 通道1模式选择<br>与输出模式相同 |
| 7:4 | CH0CAPFLT[3:0] | 通道 0 输入捕获滤波控制<br>CI0 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。<br>数字滤波器的基本原理：根据 $f_{SAMP}$ 对 CI0 输入信号进行连续采样，并记录信号相同电平的次数。达到该位配置的滤波参数后，认为是有效电平。<br>滤波器参数配置如下： |
| 3:2 | CH0CAPPSC[1:0] | 通道0输入捕获预分频器<br>这2位定义了通道0输入的预分频系数。当TIMERx_CHCTL2寄存器中的CH0EN =0时，则预分频器复位。<br>00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获。<br>01：每2个事件触发一次捕获。<br>10：每4个事件触发一次捕获。<br>11：每8个事件触发一次捕获。 |
| 1:0 | CH0MS[1:0] | 通道0模式选择<br>与输出比较模式相同 |

| CH0CAPFLT [3:0] | 采样次数 | $f_{SAMP}$ |
|---|---|---|
| 4'b0000 | 无滤波器 | |
| 4'b0001 | 2 | $f_{CK_TIMER}$ |
| 4'b0010 | 4 | |
| 4'b0011 | 8 | |
| 4'b0100 | 6 | $f_{DTS}/2$ |
| 4'b0101 | 8 | |
| 4'b0110 | 6 | $f_{DTS}/4$ |
| 4'b0111 | 8 | |
| 4'b1000 | 6 | $f_{DTS}/8$ |
| 4'b1001 | 8 | |
| 4'b1010 | 5 | $f_{DTS}/16$ |
| 4'b1011 | 6 | |
| 4'b1100 | 8 | |
| 4'b1101 | 5 | $f_{DTS}/32$ |
| 4'b1110 | 6 | |
| 4'b1111 | 8 | |

### 通道控制寄存器 2 (TIMERx_CHCTL2)

地址偏移：0x20
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | CH1NP | 保留 | CH1P | CH1EN | CH0NP | 保留 | CH0P | CH0EN |
| | | | | | | | | rw | | rw | rw | rw | | rw | rw |

| 位/位域 | 名称 | 名称 |
|---|---|---|
| 31:8 | 保留 | 必须保持复位值 |
| 7 | CH1NP | 通道1互补输出极性<br>参考CH0NP描述 |
| 6 | 保留 | 必须保持复位值 |
| 5 | CH1P | 通道1极性<br>参考CH0P描述 |
| 4 | CH1EN | 通道1使能<br>参考CH0EN描述 |
| 3 | CH0NP | 通道0互补输出极性<br>当通道0配置为输出模式，此位定义了互补输出信号的极性。<br>0：通道0互补输出高电平为有效电平<br>1：通道0互补输出低电平为有效电平<br>当通道0配置为输入模式时，此位和CH0P联合使用，作为输入信号CI0的极性选择控制信号。 |
| 2 | 保留 | 必须保持复位值 |
| 1 | CH0P | 通道0极性<br>当通道0配置为输出模式时，此位定义了输出信号极性。<br>0：通道0高电平为有效电平<br>1：通道0低电平为有效电平<br>当通道0配置为输入模式时，此位定义了CI0信号极性。<br>[CH0NP，CH0P]将选择CI0FE0或者CI1FE0的有效边沿或者捕获极性。<br>[CH0NP=0，CH0P=0]：把CIxFE0的上升沿作为捕获或者从模式下触发的有效信号，并且CIxFE0不会被翻转。<br>[CH0NP=0，CH0P=1]：把CIxFE0的下降沿作为捕获或者从模式下触发的有效信号，并且CIxFE0会被翻转。<br>[CH0NP=1，CH0P=0]：保留。 |

[CH0NP=1，CH0P=1]：把CIxFE0的上升沿和下降沿都作为捕获或者从模式下触发的有效信号，并且CIxFE0不会被翻转。

| | | |
|---|---|---|
| 0 | CH0EN | 通道0捕获/比较使能<br>当通道0配置为输出模式时，将此位置1使能CH0_O信号有效。当通道0配置为输入模式时，将此位置1使能通道0上的捕获事件。<br>0：禁止通道0<br>1：使能通道0 |

### 计数器寄存器 (TIMERx_CNT)

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | CNT[15:0] | 这些位是当前的计数值。写操作能改变计数器值。 |

### 预分频寄存器 (TIMERx_PSC)

地址偏移：0x28
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | PSC[15:0] | 计数器时钟预分频值 |

计数器时钟等于TIMER_CK时钟除以(PSC+1)，每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。

### 计数器自动重载寄存器 (TIMERx_CAR)

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | CARL[15:0] | 计数器自动重载值<br>这些位定义了计数器的自动重载值。 |

### 通道 0 捕获/比较值寄存器 (TIMERx_CH0CV)

地址偏移：0x34
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CH0VAL[15:0] | | | | | | | | | | | | | | | |
| rw | | | | | | | | | | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | CH0VAL[15:0] | 通道0的捕获或比较值<br>当通道0配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。<br>当通道0配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。 |

### 通道 1 捕获/比较值寄存器 (TIMERx_CH1CV)

地址偏移：0x38
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CH1VAL[15:0] | | | | | | | | | | | | | | | |
| rw | | | | | | | | | | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | CH1VAL[15:0] | 通道1的捕获或比较值<br>当通道1配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。<br>当通道1配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。 |

### 配置寄存器 (TIMERx_CFG)

地址偏移：0xFC
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:2 | 保留 | 必须保持复位值 |
| 1 | CHVSEL | 写捕获比较寄存器选择位<br>此位由软件写1或清0。<br>1：当写入捕获比较寄存器的值与寄存器当前值相等时，写入操作无效<br>0：无影响 |
| 0 | 保留 | 必须保持复位值 |

# 16.4. 通用定时器 L2(TIMERx, x=9,10,12,13)

## 16.4.1. 简介

通用定时器L2 (TIMERx，x=9，10，12，13)是单通道定时器，支持输入捕获和输出比较。可以产生PWM信号控制电机和电源管理。通用定时器L2含有一个16位无符号计数器。

通用定时器L2是可编程的，可以用来计数，其外部事件可以驱动其他定时器。

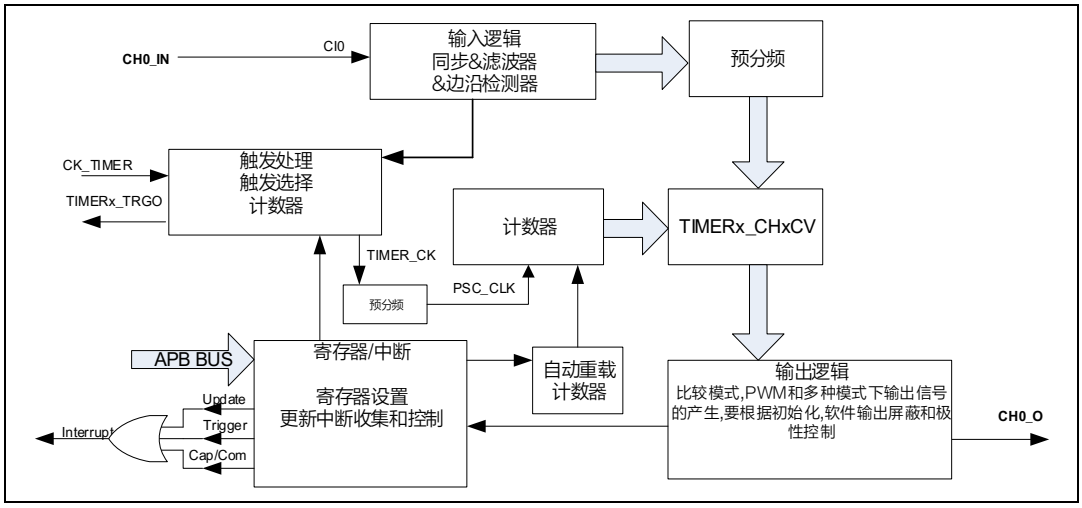
## 16.4.2. 主要特征

- 总通道数：1；
- 计数器宽度：16位；
- 时钟源：内部时钟；
- 计数模式：向上计数；
- 可编程的预分频器：16位，运行时可以被改变；
- 每个通道可配置：输入捕获模式，输出比较模式，可编程的PWM模式；
- 自动重装载功能；
- 中断输出：更新事件，比较/捕获事件。

## 16.4.3. 结构框图

*图16-57. 通用定时器L2结构框图*提供了通用定时器L2的内部配置细节。

**图 16-57. 通用定时器 L2 结构框图**




## 16.4.4. 功能描述

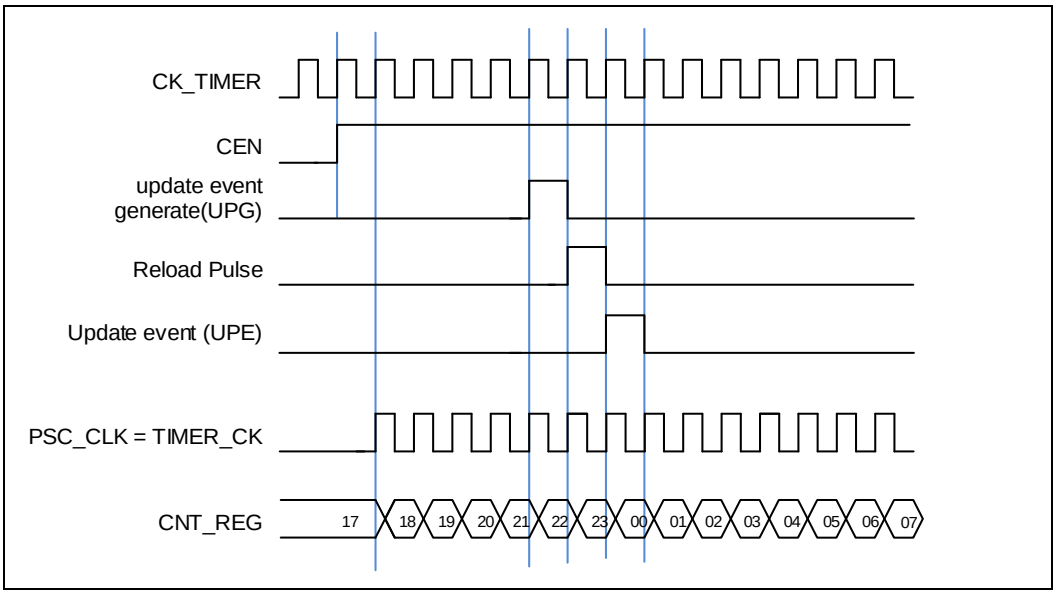
### 时钟源配置

通用定时器L2由内部时钟源CK_TIMER驱动

■ 定时器时钟TIMER_CK连接到RCU模块的CK_TIMER

通用定时器L2仅有一个时钟源CK_TIMER，用来驱动计数器预分频器。当CEN置位，CK_TIMER经过预分频器(预分频值由TIMERx_PSC寄存器确定)产生PSC_CLK。

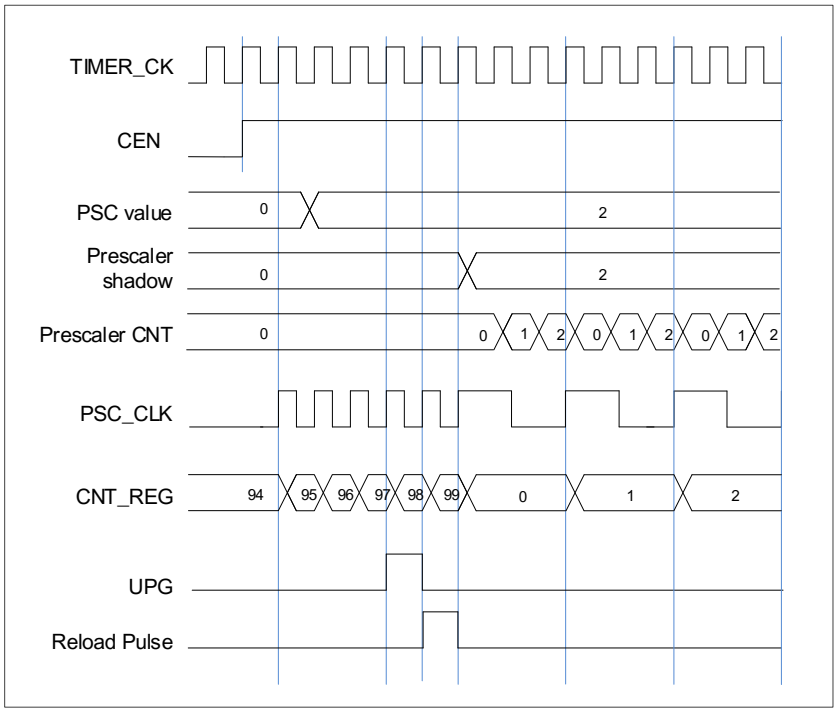
**图 16-58. 内部时钟分频为 1 时，计数器的时序图**




## 时钟预分频器

预分频器可以将定时器的时钟(TIMER_CK)频率按1到65536之间的任意值分频，分频后的时钟PSC_CLK驱动计数器计数。分频系数受预分频寄存器TIMERx_PSC控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 16-59. 当 PSC 数值从 0 变到 2 时，计数器的时序图




### 计数器向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从 0 开始向上连续计数到自动加载值(定义在 TIMERx_CAR 寄存器中)，一旦计数器计数到自动加载值，会重新从 0 开始向上计数并产生上溢事件。在向上计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 0。

当通过 TIMERx_SWEVG 寄存器的 UPG 位置 1 来设置更新事件时，计数值会被清 0，并产生更新事件。

如果 TIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(计数器自动重载寄存器，预分频寄存器)都将被更新。

*图 16-60. 向上计数时序图，PSC=0/2* 和 *图 16-61. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值*给出了一些例子，当 TIMERx_CAR=0x99 时，计数器在不同预分频因子下的行为。

图 16-60. 向上计数时序图，PSC=0/2

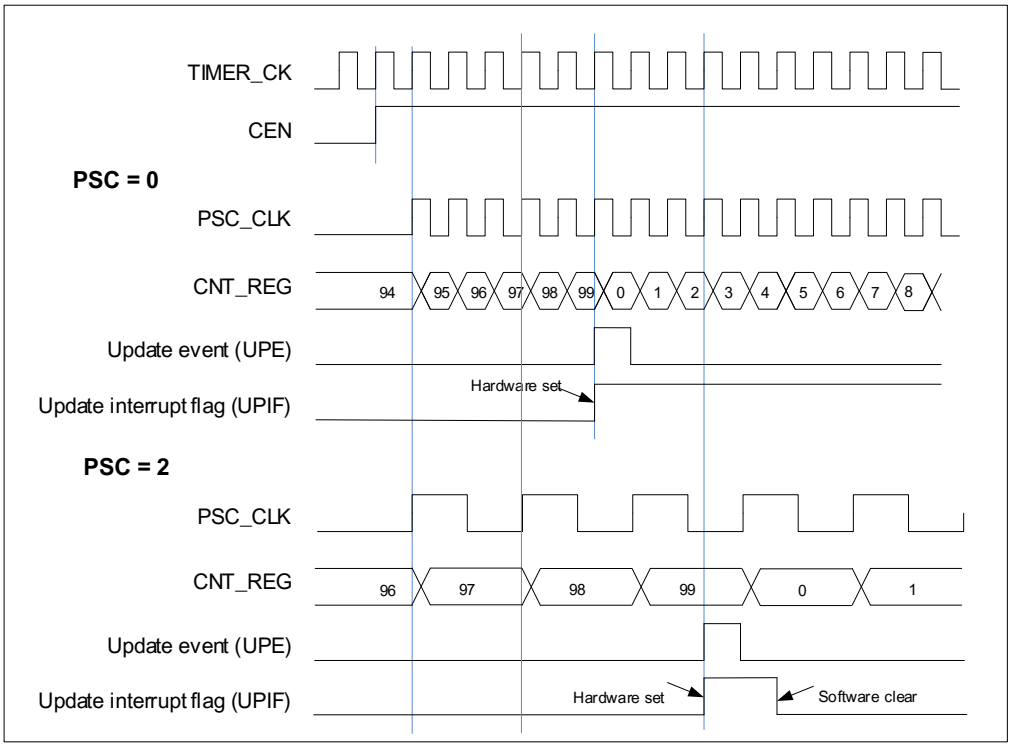
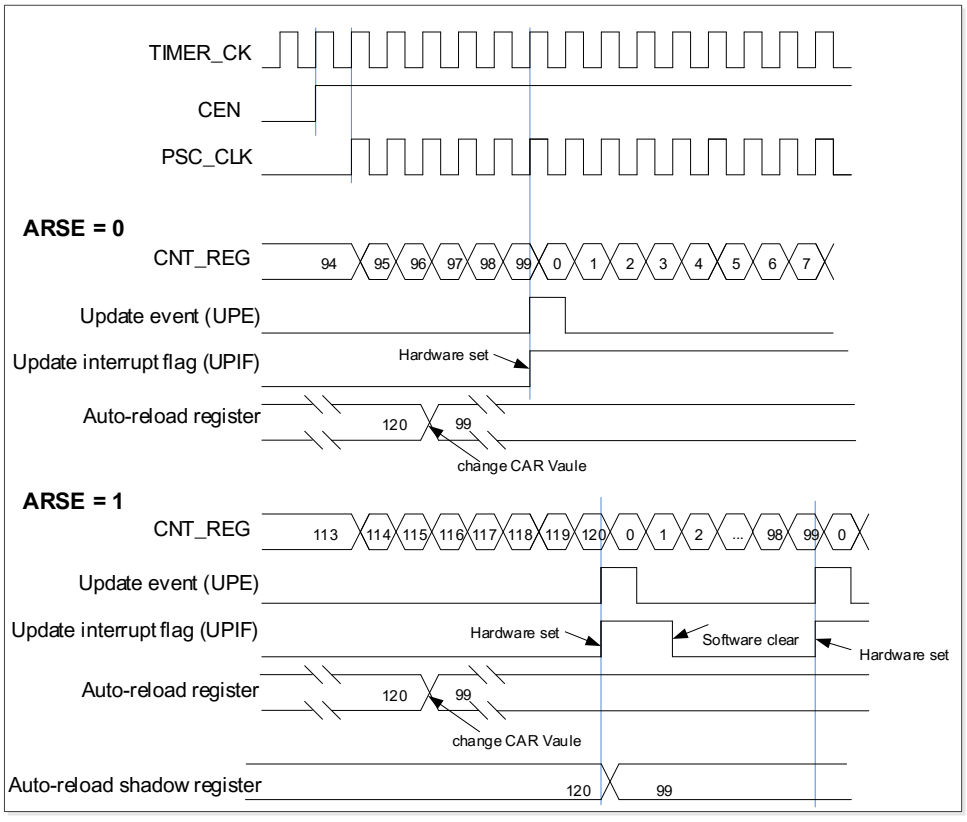



图 16-61. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



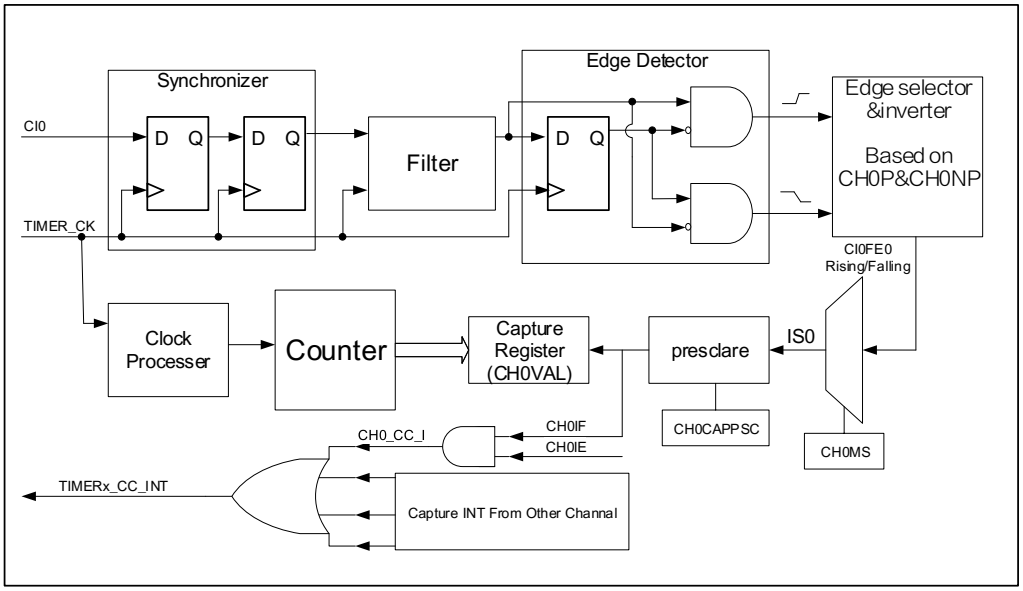
### 输入捕获和输出比较通道

通用定时器 L2 只有一个独立的通道用于捕获输入或比较输出是否匹配。该通道通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

■ 通道输入捕获功能

捕获模式允许通道测量一个波形时序，频率，周期，占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，TIMERx_CHxCV 寄存器会捕获计数器当前的值，同时 CHxIF 位被置 1，如果 CHxIE = 1 则产生通道中断。

**图 16-62. 通道输入捕获原理**




通道输入信号 CIx 先被 TIMER_CK 信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置 CHxP 选择使用上升沿或者下降沿。配置 CHxMS.，可以选择其他通道的输入信号，内部触发信号。配置 IC 预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生，TIMERx_CHxCV 存储计数器的值。

配置步骤如下：

**第一步：**滤波器配置(TIMERx_CHCTL0寄存器中CHxCAPFLT)：
根据输入信号和请求信号的质量，配置相应的CHxCAPFLT。
**第二步：**边沿选择(TIMERx_CHCTL2寄存器中CHxP/CHxNP)：
配置CHxP/CHxNP选择上升沿或者下降沿。
**第三步：**捕获源选择(TIMERx_CHCTL0寄存器中CHxMS)：
一旦通过配置CHxMS选择输入捕获源，必须确保通道配置在输入模式(CHxMS!=0x0)，而且TIMERx_CHxCV寄存器不能再被写。
**第四步：**中断使能(TIMERx_DMAINTEN寄存器中CHxIE)：
使能相应中断，可以获得中断。

**第五步：**捕获使能(TIMERx_CHCTL2寄存器中CHxEN)。

**结果：**当期望的输入信号发生时，TIMERx_CHxCV被设置成当前计数器的值，CHxIF为置1。如果CHxIF位已经为1，则CHxOF位置1。根据TIMERx_DMAINTEN寄存器中CHxIE的配置，相应的中断会被提出。

**直接产生：**软件设置CHxG位，会直接产生中断。

输入捕获模式也可用来测量 TIMERx_CHx 引脚上信号的脉冲波宽度。例如，一个 PWM 波连接到 CI0。配置 TIMERx_CHCTL0 寄存器中 CH0MS 为 2'b01，选择通道 0 的捕获信号为 CI0 并设置上升沿捕获。计数器配置为复位模式，在通道 0 的上升沿复位。TIMERX_CH0CV 寄存器测量 PWM 的周期值，TIMERx_CH1CV 寄存器测量 PWM 占空比值。

■ 通道输出比较功能

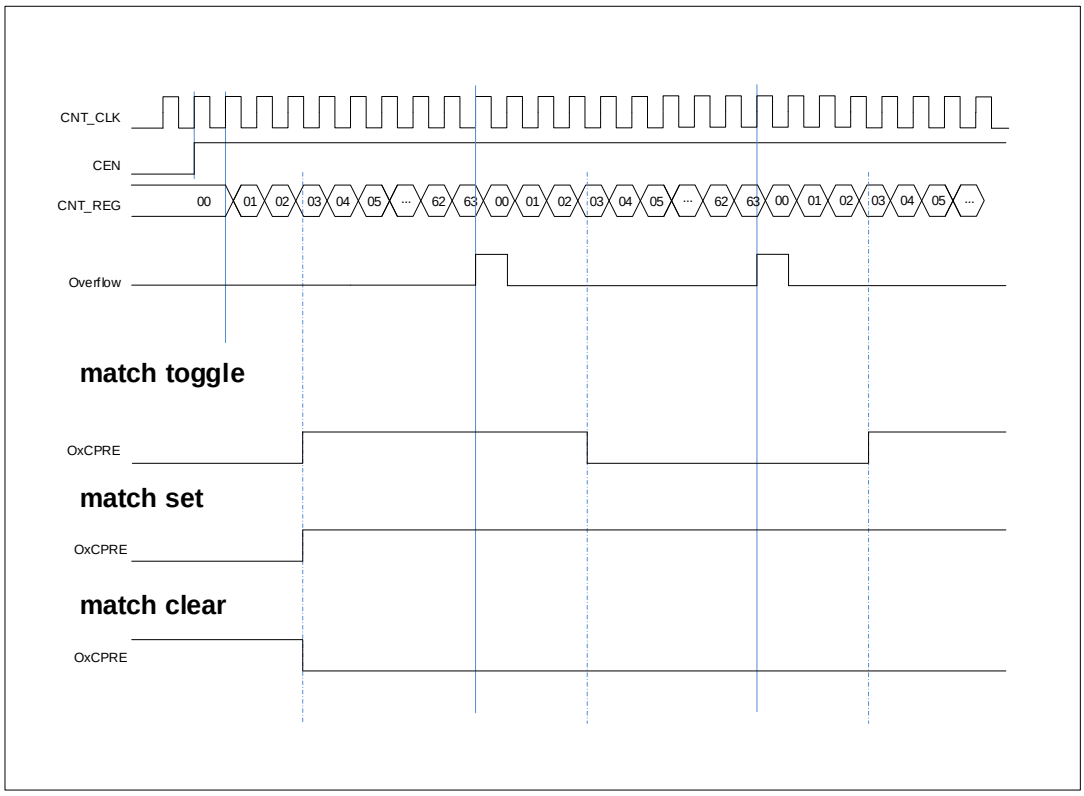
在通道输出比较功能，TIMERx 可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的 TIMERx_CHxCV 寄存器与计数器的值匹配时，根据 CHxCOMCTL 的配置，这个通道的输出可以被置高电平，被置低电平或者反转。当计数器的值与 TIMERx_CHxCV 寄存器的值匹配时，CHxIF 位被置 1，如果 CHxIE = 1 则会产生中断，如果 CxCDE=1 则会产生 DMA 请求。

配置步骤如下：

**第一步：**时钟配置：
配置定时器时钟源，预分频器等。
**第二步：**比较模式配置：
设置CHxCOMSEN位来配置输出比较影子寄存器；
设置CHxCOMCTL位来配置输出模式(置高电平/置低电平/反转)；
设置CHxP/CHxNP位来选择有效电平的极性；
设置CHxEN使能输出。
**第三步：**通过CHxIE位配置中断使能。
**第四步：**通过TIMERx_CAR寄存器和TIMERx_CHxCV寄存器配置输出比较时基：
TIMERx_CHxCV可以在运行时根据你所期望的波形而改变。
**第五步：**设置CEN位使能定时器。

*图 16-63. 三种输出比较模式*显示了三种比较输出模式：反转/置高电平/置低电平，CAR=0x63, CHxVAL=0x3。

图 16-63. 三种输出比较模式




## 通道输出参考信号

当 TIMERx 用于输出匹配比较模式下，设置 CHxCOMCTL 位可以定义 OxCPRE 信号(通道 x 准备信号)类型。OxCPRE 信号有若干类型的输出功能，包括，设置 CHxCOMCTL=0x00 可以保持原始电平；设置 CHxCOMCTL=0x01 可以将 OxCPRE 信号设置为高电平；设置 CHxCOMCTL=0x02 可以将 OxCPRE 信号设置为低电平；设置 CHxCOMCTL=0x03，在计数器值和 TIMERx_CHxCV 寄存器的值匹配时，可以翻转输出信号。

PWM 模式 0 和 PWM 模式 1 是 OxCPRE 的另一种输出类型，设置 CHxCOMCTL 位域位 0x06 或 0x07 可以配置 PWM 模式 0/PWM 模式 1。在这些模式中，根据计数器值和 TIMERx_CHxCV 寄存器值的关系以及计数方向，OxCPRE 信号改变其电平。具体细节描述，请参考相应的位。

设置 CHxCOMCTL =0x04 或 0x05 可以实现 OxCPRE 信号的强制输出功能。输出比较信号能够直接由软件强置为有效或无效状态，而不依赖于 TIMERx_CHxCV 的值和计数器值之间间的比较结果。

## 定时器互连

参考*高级定时器(TIMERx, x=0, 7)互连*

## 定时器调试模式

当Cortex®-M4内核停止，DBG_CTL0寄存器中的TIMERx_HOLD配置位被置1，定时器计数器停止。

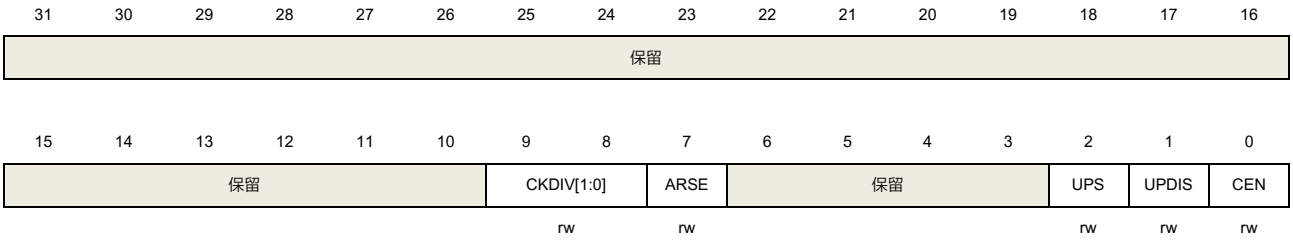
### 16.4.5. TIMERx 寄存器(x=9,10,12,13)

TIMER9基地址：0x4001 5000
TIMER10基地址：0x4001 5400
TIMER12基地址：0x4000 1C00
TIMER13基地址：0x4000 2000

#### 控制寄存器 0 (TIMERx_CTL0)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



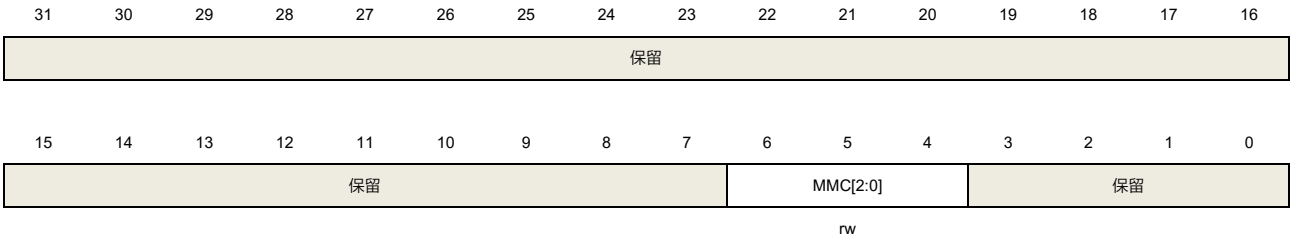

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:10 | 保留 | 必须保持复位值 |
| 9:8 | CKDIV[1:0] | 时钟分频<br>通过软件配置CKDIV，规定定时器时钟(CK_TIMER) 与死区时间和数字滤波器采样时钟(DTS)之间的分频系数。<br>00：$f_{DTS}=f_{CK_TIMER}$<br>01：$f_{DTS}= f_{CK_TIMER} /2$<br>10：$f_{DTS}= f_{CK_TIMER} /4$<br>11：保留 |
| 7 | ARSE | 自动重载影子使能<br>0：禁能TIMERx_CAR寄存器的影子寄存器<br>1：使能TIMERx_CAR寄存器的影子寄存器 |
| 6:3 | 保留 | 必须保持复位值 |
| 2 | UPS | 更新请求源<br>软件配置该位，选择更新事件源.<br>0：以下事件均会产生更新中断或DMA请求：<br>UPG位被置1<br>计数器溢出/下溢<br>复位模式产生的更新<br>1：下列事件会产生更新中断或DMA请求：<br>计数器溢出/下溢 |

| | | |
|---|---|---|
| 1 | UPDIS | 禁止更新.<br>该位用来使能或禁能更新事件的产生<br>0：更新事件使能. 更新事件发生时，相应的影子寄存器被装入预装载值，以下事件均会产生更新事件：<br>UPG位被置1<br>计数器溢出/下溢<br>复位模式产生的更新<br>1：更新事件禁能.<br>注意：当该位被置1时，UPG位被置1或者复位模式不会产生更新事件，但是计数器和预分频器被重新初始化 |
| 0 | CEN | 计数器使能<br>0：计数器禁能<br>1：计数器使能<br>在软件将CEN位置1后，外部时钟、暂停模式和编码器模式才能工作。 |

### 控制寄存器 1 (TIMERx_CTL1)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:7 | 保留 | 必须保持复位值 |
| 6:4 | MMC[2:0] | 主模式控制<br>这些位控制TRGO信号的选择，TRGO信号由主定时器发给从定时器用于同步功能<br>000：当产生一个定时器复位事件后，输出一个TRGO信号，定时器复位源为：<br>主定时器产生一个复位事件<br>TIMERx_SWEVG寄存器中UPG位置1<br>001：当产生一个定时器使能事件后，输出一个TRGO信号，定时器使能源为：<br>CEN位置1<br>在暂停模式下，触发输入置1<br>010：当产生一个定时器更新事件后，输出一个TRGO信号，更新事件源由UPDIS和UPS位决定<br>011：当通道0在发生一次捕获或一次比较成功时，主模式控制器产生一个TRGO脉冲<br>100：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O0CPRE |

101：保留。
110：保留。
111：保留。

| | | |
|---|---|---|
| 3:0 | 保留 | 必须保持复位值 |

### DMA 和中断使能寄存器（TIMERx_DMAINTEN)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | | | | | | | CH0IE | UPIE |
| | | | | | | | | | | | | | | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:2 | 保留 | 必须保持复位值 |
| 1 | CH0IE | 通道0比较/捕获中断使能<br>0：禁止通道0中断<br>1：使能通道0中断 |
| 0 | UPIE | 更新中断使能<br>0：禁止更新中断<br>1：使能更新中断 |

### 中断标志寄存器（TIMERx_INTF)

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

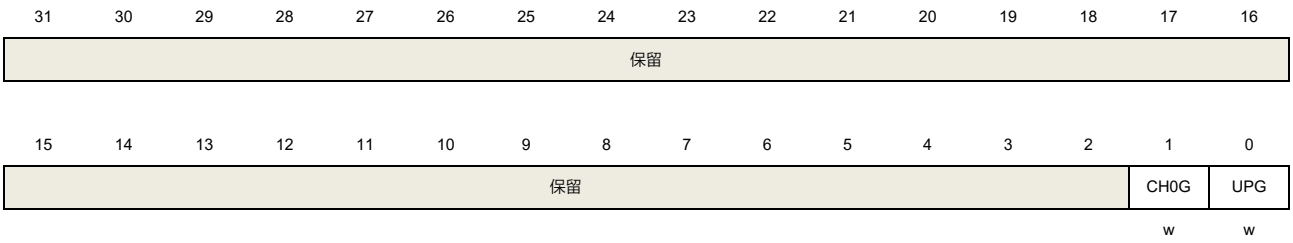
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | CH0OF | 保留 | | | | | | | CH0IF | UPIF |
| | | | | | | rc_w0 | | | | | | | | rc_w0 | rc_w0 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:10 | 保留 | 必须保持复位值 |
| 9 | CH0OF | 通道0捕获溢出标志<br>当通道0被配置为输入模式时，在CH0IF标志位已经被置1后，捕获事件再次发生时，该标志位可以由硬件置1。该标志位由软件清0。<br>0：无捕获溢出中断发生<br>1：发生了捕获溢出中断 |
| 8:2 | 保留 | 必须保持复位值 |
| 1 | CH0IF | 通道0比较/捕获中断标志<br>此标志由硬件置1软件清0。当通道0在输入模式下时，捕获事件发生时此标志位被置1；当通道0在输出模式下时，此标志位在一个比较事件发生时被置1。<br>0：无通道0中断发生<br>1：通道0中断发生 |
| 0 | UPIF | 更新中断标志<br>此位在更新事件发生时由硬件置1，软件清0。<br>0：无更新中断发生<br>1：发生更新中断 |

### 软件事件产生寄存器 (TIMERx_SWEVG)

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字(32位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:2 | 保留 | 必须保持复位值 |
| 1 | CH0G | 通道0捕获或比较事件发生<br>该位由软件置1，用于在通道0产生一个捕获/比较事件，由硬件自动清0。当此位被置1，CH0IF标志位被置1，若开启对应的中断和DMA，则发出相应的中断和DMA请求。此外，如果通道0配置为输入模式，计数器的当前值被捕获到TIMERx_CH0CV寄存器，如果CH0IF标志位已经为1，则CH0OF标志位被置1。<br>0：不产生通道0捕获或比较事件<br>1：发生通道0捕获或比较事件 |

| | | |
|---|---|---|
| 0 | UPG | 更新事件产生<br>此位由软件置1，被硬件自动清0。当此位被置1并且向上计数模式，计数器被清0，预分频计数器将同时被清除。<br>0：无更新事件产生<br>1：产生更新事件 |

### 通道控制寄存器 0 (TIMERx_CHCTL0)

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字(32位)访问。




**输出比较模式：**

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:7 | 保留 | 必须保持复位值 |
| 6:4 | CH0COMCTL[2:0] | 通道 0 输出比较模式<br>此位定义了输出准备信号 O0CPRE 的输出比较模式，而 O0CPRE 决定了 CH0_O、CH0_ON 的值。另外，O0CPRE 高电平有效，而 CH0_O、CH0_ON 通道的极性取决于 CH0P、CH0NP 位。<br>000：时基。输出比较寄存器 TIMERx_CH0CV 与计数器 TIMERx_CNT 间的比较对 O0CPRE 不起作用<br>001：匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为高。<br>010：匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为低。<br>011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 翻转。<br>100：强制为低。强制 O0CPRE 为低电平<br>101：强制为高。强制 O0CPRE 为高电平<br>110：PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。<br>111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时， |

O0CPRE 为高电平，否则为低电平。
如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，O0CPRE 电平才改变。

| | | |
|---|---|---|
| 3 | CH0COMSEN | 通道0输出比较影子寄存器使能<br>当此位被置1，TIMERx_CH0CV寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。<br>0：禁止通道0输出/比较影子寄存器<br>1：使能通道0输出/比较影子寄存器<br>仅在单脉冲模式下(TIMERx_CTL0寄存器的SPM =1)，可以在未确认预装载寄存器情况下使用PWM 模式。 |
| 2 | CH0COMFEN | 通道0输出比较快速使能<br>当该位为1时，如果通道配置为PWM模式0或者PWM模式1，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH0_O被设置为比较电平而与比较结果无关。<br>0：禁止通道0输出比较快速<br>1：使能通道0输出比较快速 |
| 1:0 | CH0MS[1:0] | 通道0 I/O模式选择<br>这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2寄存器的CH0EN位被清0)时这些位才可写。<br>00：通道0配置为输出<br>01：通道0配置为输入，IS0映射在CI0FE0上<br>Others：保留 |

**输入捕获模式：**

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 必须保持复位值 |
| 7:4 | CH0CAPFLT[3:0] | 通道 0 输入捕获滤波控制<br>CI0 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。<br>数字滤波器的基本原理：根据 $f_{SAMP}$ 对 CI0 输入信号进行连续采样，并记录信号相同电平的次数。达到该位配置的滤波参数后，认为是有效电平。<br>滤波器参数配置如下： |

| CH0CAPFLT [3:0] | 采样次数 | $f_{SAMP}$ |
|---|---|---|
| 4'b0000 | 无滤波器 | |
| 4'b0001 | 2 | $f_{CK_TIMER}$ |
| 4'b0010 | 4 | |
| 4'b0011 | 8 | |
| 4'b0100 | 6 | $f_{DTS}/2$ |
| 4'b0101 | 8 | |
| 4'b0110 | 6 | $f_{DTS}/4$ |
| 4'b0111 | 8 | |
| 4'b1000 | 6 | $f_{DTS}/8$ |
| 4'b1001 | 8 | |

| | | | |
|---|---|---|---|
| | | 4'b1010 | 5 | $f_{DTS}$/16 |
| | | 4'b1011 | 6 | |
| | | 4'b1100 | 8 | |
| | | 4'b1101 | 5 | $f_{DTS}$/32 |
| | | 4'b1110 | 6 | |
| | | 4'b1111 | 8 | |

| | | |
|---|---|---|
| 3:2 | CH0CAPPSC[1:0] | 通道0输入捕获预分频器<br>这2位定义了通道0输入的预分频系数。当TIMERx_CHCTL2寄存器中的CH0EN =0时，则预分频器复位。<br>00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获<br>01：每2个事件触发一次捕获<br>10：每4个事件触发一次捕获<br>11：每8个事件触发一次捕获 |
| 1:0 | CH0MS[1:0] | 通道0模式选择<br>与输出比较模式相同 |

### 通道控制寄存器 2 (TIMERx_CHCTL2)

地址偏移：0x20
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留. | | | | | | | | | | | | CH0NP | 保留 | CH0P | CH0EN |
| | | | | | | | | | | | | rw | | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:4 | 保留 | 必须保持复位值 |
| 3 | CH0NP | 通道0互补输出极性<br>当通道0配置为输出模式，此位定义了互补输出信号的极性。<br>0：通道0高电平有效<br>1：通道0低电平有效<br>当通道0配置为输入模式时，此位和CH0P联合使用，作为输入信号CI0的极性选择控制信号。 |
| 2 | 保留 | 必须保持复位值 |
| 1 | CH0P | 通道0极性<br>当通道0配置为输出模式时，此位定义了输出信号极性。<br>0：通道0高电平有效 |

| | | |
|---|---|---|
| | | 1：通道0低电平有效<br>当通道0配置为输入模式时，此位定义了CI0信号极性<br>[CH0NP, CH0P] 将选择CI0FE0或者CI1FE0的有效边沿或者捕获极性<br>[CH0NP=0, CH0P=0]：把CIxFE0的上升沿作为捕获或者从模式下触发的有效信号，并且CIxFE0不会被翻转。<br>[CH0NP=0, CH0P=1]：把CIxFE0的下降沿作为捕获或者从模式下触发的有效信号，并且CIxFE0会被翻转。<br>[CH0NP=1, CH0P=0]：保留。<br>[CH0NP=1, CH0P=1]：把CIxFE0的上升沿和下降沿都作为捕获或者从模式下触发的有效信号，并且CIxFE0不会被翻转。 |
| 0 | CH0EN | 通道0捕获/比较使能<br>当通道0配置为输出模式时，将此位置1使能CH0_O信号有效。当通道0配置为输入模式时，将此位置1使能通道0上的捕获事件。<br>0：禁止通道0<br>1：使能通道0 |

### 计数器寄存器 (TIMERx_CNT)

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | CNT[15:0] | 这些位是当前的计数值。写操作能改变计数器值。 |

### 预分频寄存器 (TIMERx_PSC)

地址偏移：0x28
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | PSC[15:0] | 计数器时钟预分频值<br>计数器时钟等于TIMER_CK时钟除以(PSC+1)，每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。 |

### 计数器自动重载寄存器 (TIMERx_CAR)

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | CARL[15:0] | 计数器自动重载值<br>这些位定义了计数器的自动重载值。 |

### 通道 0 捕获/比较值寄存器 (TIMERx_CH0CV)

地址偏移：0x34
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CH0VAL[15:0] | | | | | | | | | | | | | | | |
| rw | | | | | | | | | | | | | | | |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | CH0VAL[15:0] | 通道0的捕获或比较值<br>当通道0配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。<br>当通道0配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。 |

### 配置寄存器 (TIMERx_CFG)

地址偏移：0xFC
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

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

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:2 | 保留 | 必须保持复位值 |
| 1 | CHVSEL | 写捕获比较寄存器选择位<br>此位由软件写1或清0。<br>1：当写入捕获比较寄存器的值与寄存器当前值相等时，写入操作无效<br>0：无影响 |
| 0 | 保留 | 必须保持复位值 |

# 16.5. 基本定时器(TIMERx, x=5,6)

## 16.5.1. 简介

基本定时器(TIMER5/6)包含一个无符号16位计数器。基本定时器可以配置产生DMA请求，产生TRGO触发连接到DAC。

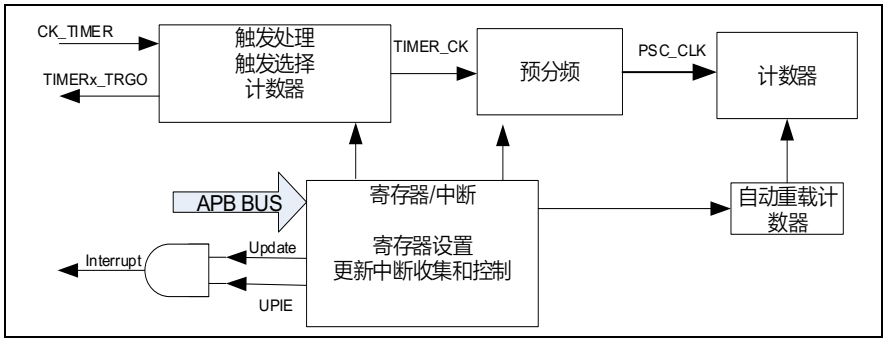
## 16.5.2. 主要特征

- 计数器宽度：16位；
- 定时器时钟源只有内部时钟；
- 计数模式：向上计数；
- 可编程的预分频器：16位，运行时可以被改变；
- 自动重装载功能；
- 中断输出和DMA请求：更新事件。

## 16.5.3. 结构框图

*图 16-64. 基本定时器结构框图*提供了基本定时器内部配置的细节。

**图 16-64. 基本定时器结构框图**




## 16.5.4. 功能描述

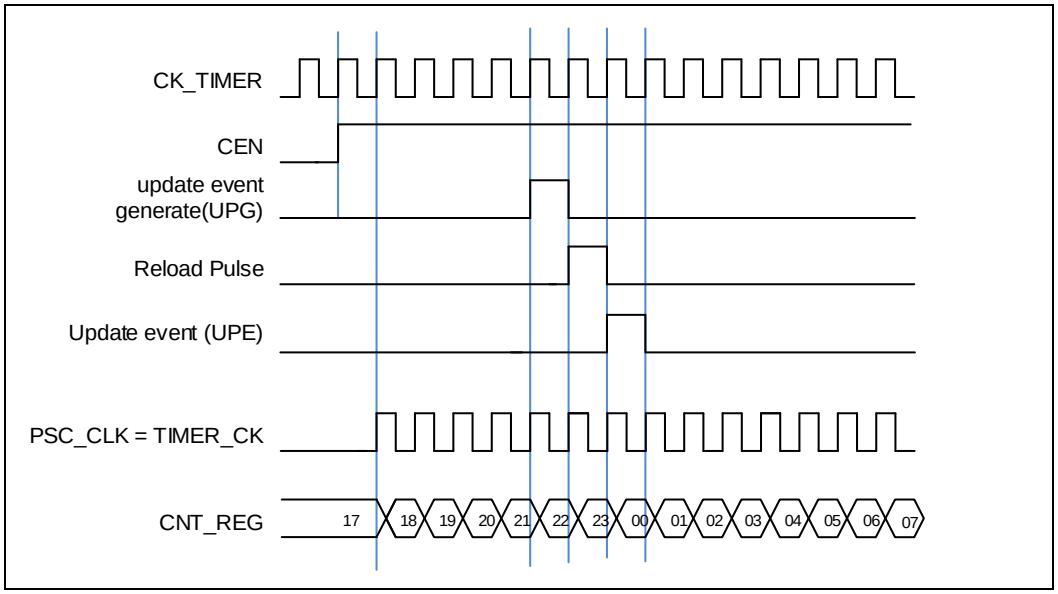
### 时钟源配置

基本定时器可以由内部时钟源CK_TIMER驱动 。

- 定时器时钟TIMER_CK连接到RCU模块的CK_TIMER

基本定时器仅有一个时钟源CK_TIMER，用来驱动计数器预分频器。当CEN置位，CK_TIMER经过预分频器(预分频值由TIMERx_PSC寄存器确定)产生PSC_CLK。

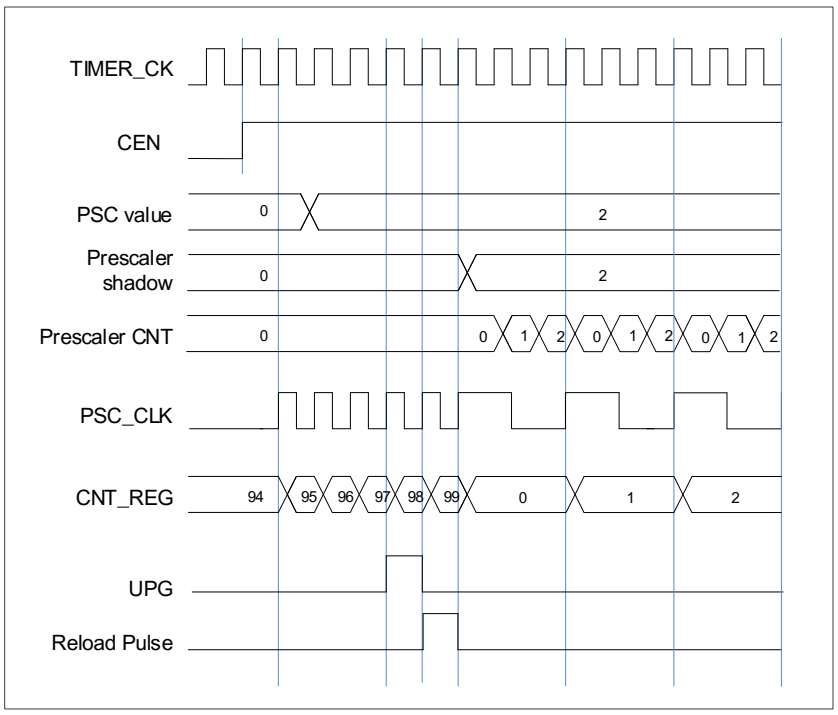
**图 16-65. 内部时钟分频为 1 时，计数器的时序图**




## 时钟预分频器

预分频器可以将定时器的时钟(TIMER_CK)频率按 1 到 65536 之间的任意值分频，分频后的时钟 PSC_CLK 驱动计数器计数。分频系数受预分频寄存器 TIMERx_PSC 控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

**图 16-66. 当 PSC 数值从 0 变到 2 时，计数器的时序图**



### 计数器向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从 0 开始向上连续计数到自动加载值(定义在 TIMERx_CAR 寄存器中)，一旦计数器计数到自动加载值，会重新从 0 开始向上计数并产生上溢事件。在向上计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 0。

当通过 TIMERx_SWEVG 寄存器的 UPG 位置 1 来设置更新事件时，计数值会被清 0，并产生更新事件。

如果 TIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(计数器自动重载寄存器，预分频寄存器)都将被更新。

下面这些图给出了一些例子，当 TIMERx_CAR=0x99 时，计数器在不同预分频因子下的行为。

**图 16-67. 向上计数时序图，PSC=0/2**

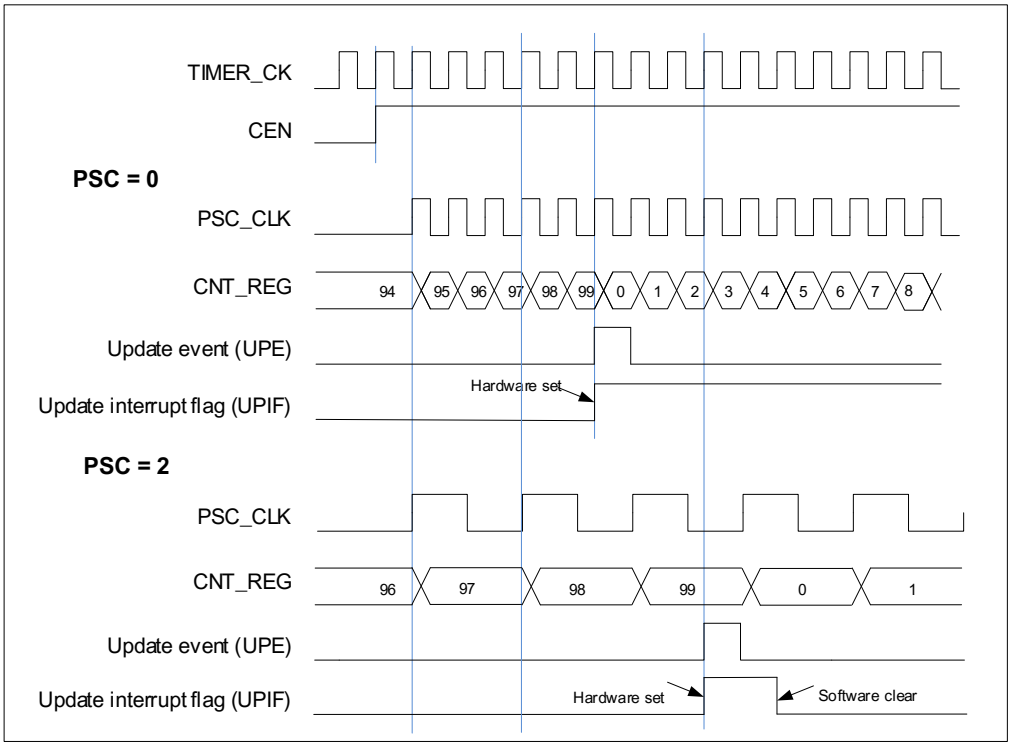
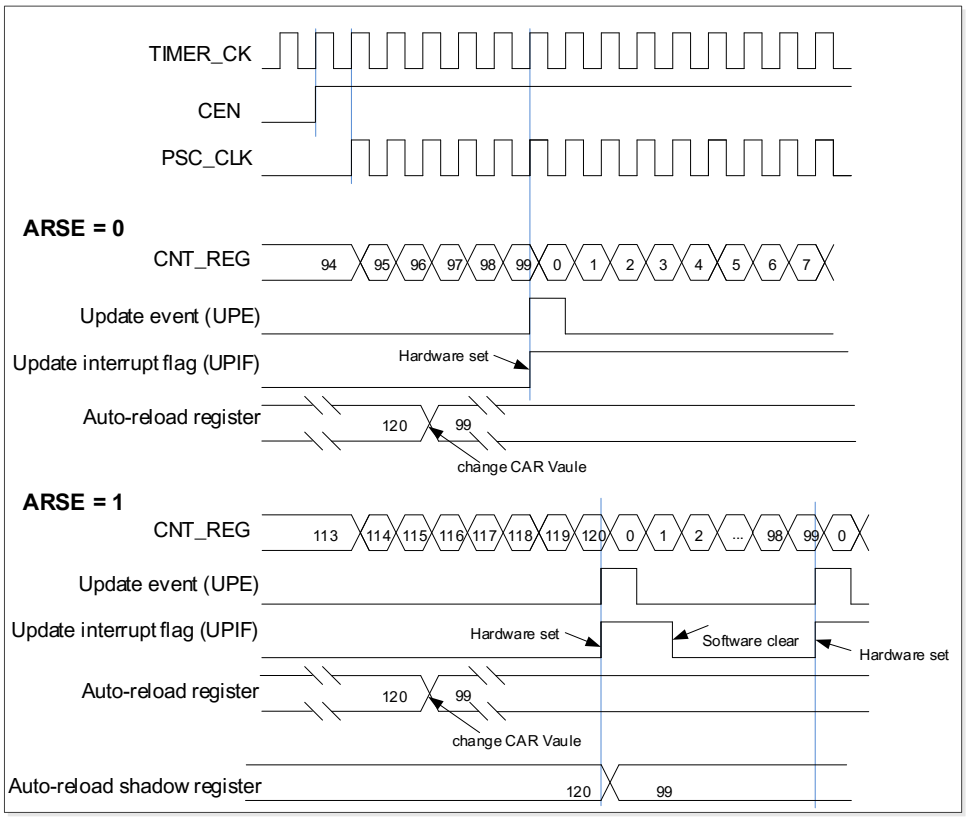


图 16-68. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值




## 单脉冲模式

单脉冲模式与重复模式是相反的，设置TIMERx_CTL0寄存器的SPM位置1，则使能单脉冲模式。当SPM置1，计数器在下次更新事件到来后清零并停止计数。

一旦设置定时器运行在单脉冲模式下，需要设置TIMERx_CTL0寄存器的定时器使能位CEN=1来使能计数器，此后CEN位一直保持为1直到更新事件发生或者CEN位被软件写0。如果CEN位被软件清0，计数器停止工作，计数值被保持。

## 定时器调试模式

当Cortex®-M4内核停止，DBG_CTL0寄存器中的TIMERx_HOLD配置位被置1，定时器计数器停止。

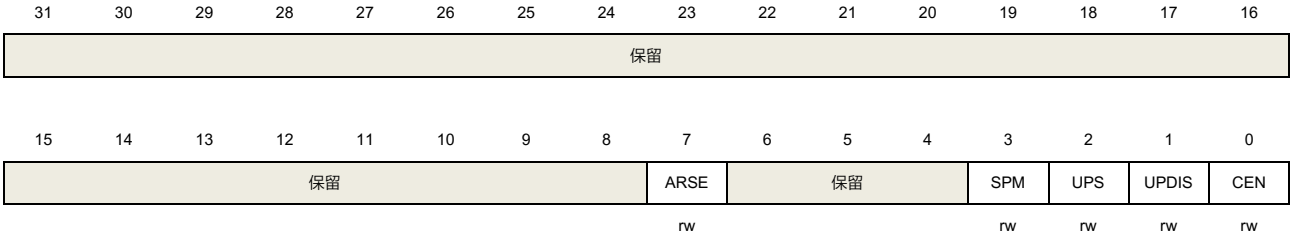
## 16.5.5. TIMERx 寄存器(x=5,6)

TIMER5基地址：0x4000 1000
TIMER6基地址：0x4000 1400

### 控制寄存器 0 (TIMERx_CTL0)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 必须保持复位值 |
| 7 | ARSE | 自动重载影子使能<br>0：禁能TIMERx_CAR寄存器的影子寄存器<br>1：使能TIMERx_CAR寄存器的影子寄存器 |
| 6:4 | 保留 | 必须保持复位值 |
| 3 | SPM | 单脉冲模式<br>0：单脉冲模式禁能，更新事件发生后，计数器继续计数<br>1：单脉冲模式使能，在下一次更新事件发生时，计数器停止计数 |
| 2 | UPS | 更新请求源<br>软件配置该位，选择更新事件源.<br>0：以下事件均会产生更新中断或DMA请求：<br>UPG位被置1<br>计数器溢出/下溢<br>复位模式产生的更新<br>1：下列事件会产生更新中断或DMA请求：<br>计数器溢出/下溢 |
| 1 | UPDIS | 禁止更新.<br>该位用来使能或禁能更新事件的产生<br>0：更新事件使能. 更新事件发生时，相应的影子寄存器被装入预装载值，以下事件均会产生更新事件：<br>UPG位被置1<br>计数器溢出/下溢 |

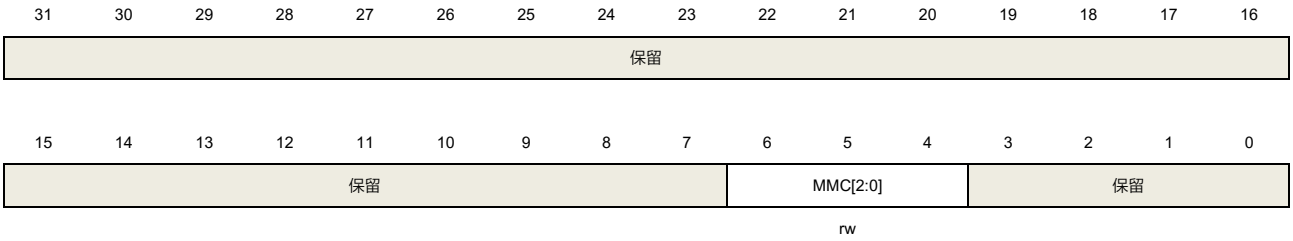
复位模式产生的更新
1：更新事件禁能.
注意：当该位被置1时，UPG位被置1或者复位模式不会产生更新事件，但是计数器和预分频器被重新初始化

| | | |
|---|---|---|
| 0 | CEN | 计数器使能<br>0：计数器禁能<br>1：计数器使能<br>在软件将CEN位置1后，外部时钟、暂停模式和编码器模式才能工作。 |

### 控制寄存器 1 (TIMERx_CTL1)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

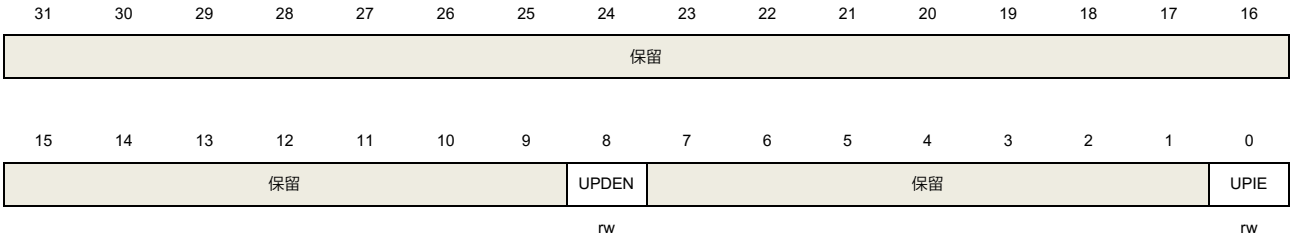



| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:7 | 保留 | 必须保持复位值 |
| 6:4 | MMC[2:0] | 这些位控制TRGO信号的选择，TRGO信号由主定时器发给从定时器用于同步功能<br>000：当产生一个定时器复位事件后，输出一个TRGO信号，定时器复位源为：<br>主定时器产生一个复位事件<br>TIMERx_SWEVG寄存器中UPG位置1<br>001：当产生一个定时器使能事件后，输出一个TRGO信号，定时器使能源为：<br>CEN位置1<br>在暂停模式下，触发输入置1<br>010：当产生一个定时器更新事件后，输出一个TRGO信号，更新事件源由UPDIS和UPS位决定 |
| 3:0 | 保留 | 必须保持复位值 |

### DMA 和中断使能寄存器 (TIMERx_DMAINTEN)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

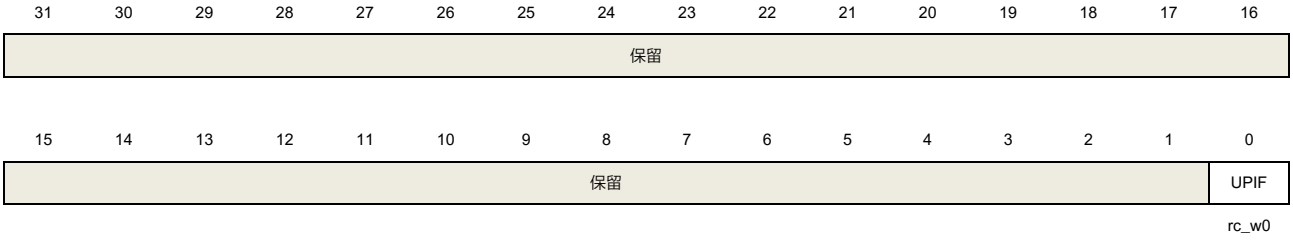



| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:9 | 保留 | 必须保持复位值 |
| 8 | UPDEN | 更新DMA请求使能<br>0：禁止更新DMA请求<br>1：使能更新DMA请求 |
| 7:1 | 保留 | 必须保持复位值 |
| 0 | UPIE | 更新中断使能<br>0：禁止更新中断<br>1：使能更新中断 |

### 中断标志寄存器（TIMERx_INTF)

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



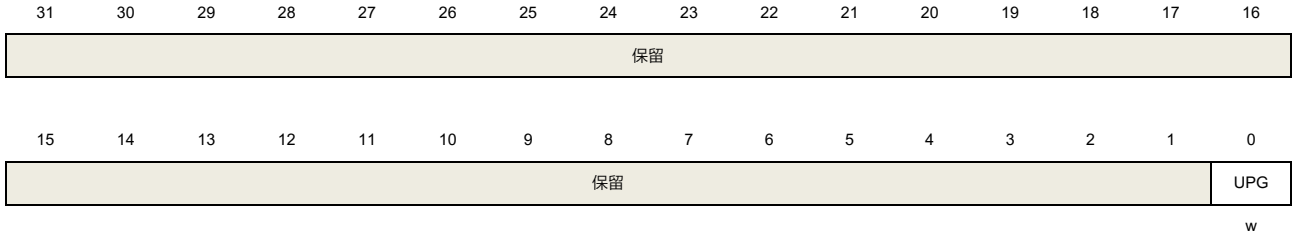

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:1 | 保留 | 必须保持复位值 |
| 0 | UPIF | 更新中断标志<br>此位在更新事件发生时由硬件置1，软件清0。<br>0：无更新中断发生<br>1：发生更新中断 |

### 软件事件产生寄存器（TIMERx_SWEVG)

地址偏移：0x14

复位值：0x0000 0000

该寄存器只能按字(32位)访问。

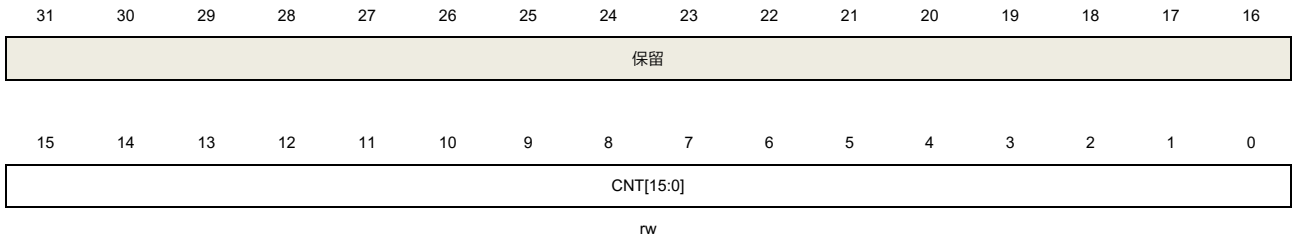



| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:1 | 保留 | 必须保持复位值 |
| 0 | UPG | 更新事件产生<br>此位由软件置1，被硬件自动清0。当此位被置1并且向上计数模式，计数器被清0，预分频计数器将同时被清除。<br>0：无更新事件产生<br>1：产生更新事件 |

### 计数器寄存器 (TIMERx_CNT)

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

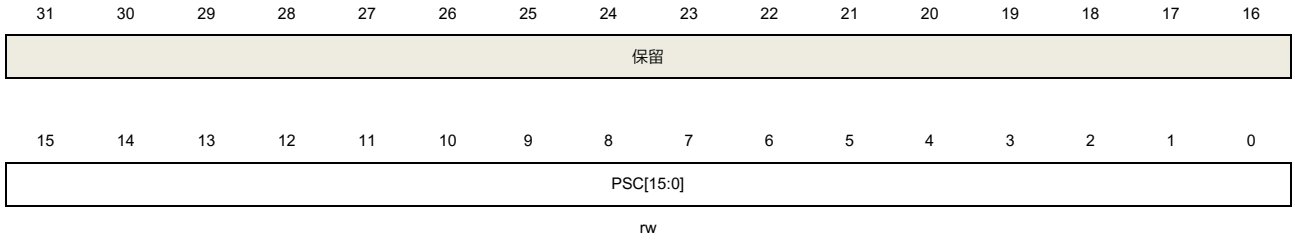



| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | CNT[15:0] | 这些位是当前的计数值。写操作能改变计数器值。 |

### 预分频寄存器 (TIMERx_PSC)

地址偏移：0x28
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

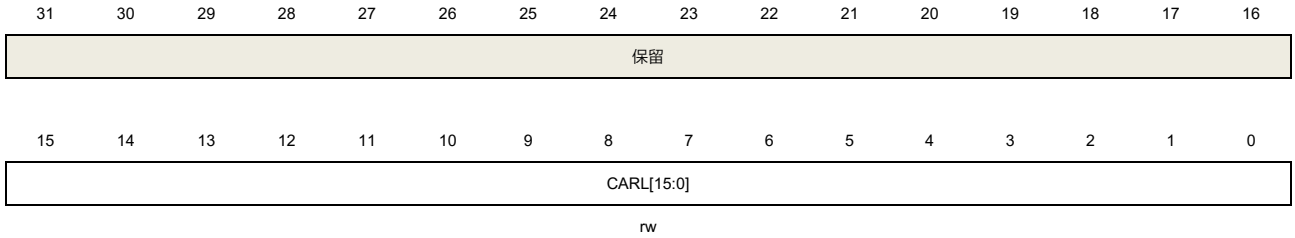



| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | PSC[15:0] | 计数器时钟预分频值<br>计数器时钟等于TIMER_CK时钟除以(PSC+1)，每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。 |

### 计数器自动重载寄存器 (TIMERx_CAR)

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值 |
| 15:0 | CARL[15:0] | 计数器自动重载值<br>这些位定义了计数器的自动重载值。 |

# 17. 通用同步异步收发器(USART)

## 17.1. 简介

通用同步异步收发器(USART)提供了一个灵活方便的串行数据交换接口，数据帧可以通过全双工或半双工，同步或异步的方式进行传输。USART提供了可编程的波特率发生器，能对UCLK(PCLK1或PCLK2)进行分频产生USART发送和接收所需的特定频率。

USART不仅支持标准的异步收发模式，还实现了一些其他类型的串行数据交换模式，如红外编码规范，SIR，智能卡协议，LIN，半双工以及同步模式。它还支持多处理器通信和Modem流控操作(CTS/RTS)。数据帧支持从LSB或者MSB开始传输。数据位的极性和TX/RX引脚都可以灵活配置。

USART支持DMA功能，以实现高速率的数据通信，除了UART4。

## 17.2. 主要特征

- NRZ标准格式；
- 全双工异步通信；
- 可编程的波特率产生器：
  - 由外设时钟分频产生，其中USART0由PCLK2分频得到，USART1/2和UART3/4由PCLK1分频得到；
  - 16倍过采样；
  - 当时钟频率为120M，过采样为16，最高速度可到7.5MBits/s；
- 完全可编程的串口特性：
  - 偶校验位，奇校验位，无校验位的生成/检测；
  - 数据位(8或9位)；
  - 产生0.5，1，1.5或者2个停止位；
- 发送器和接收器可分别使能；
- 支持硬件Modem流控操作(CTS/RTS)；
- DMA访问数据缓冲区；
- LIN断开帧的产生和检测；
- 支持红外数据协议(IrDA)。
- 同步传输模式以及为同步传输输出发送时钟；
- 支持兼容ISO7816-3的智能卡接口：
  - 字节模式(T=0)；
  - 块模式(T=1)；
  - 直接和反向转换；
- 多处理器通信：
  - 如果地址不匹配，则进入静默模式；
  - 通过线路空闲检测或者地址匹配检测从静默模式唤醒；
- 多种状态标志：
  - 传输检测标志：接收缓冲区不为空(RBNE)，发送缓冲区为空(TBE)，传输完成(TC)，

忙(BSY)；
- 错误检测标志：过载错误(ORERR)，噪声错误(NERR)，帧格式错误(FERR)，奇偶校验错误(PERR)；
- 硬件流控操作标志：CTS变化(CTSF)；
- LIN模式标志：LIN断开检测(LBDF)；
- 多处理器通信模式标志：IDLE帧检测(IDLEF)；
- 智能卡模式标志：块结束(EBF)和接收超时(RTF)；
- 若相应的中断使能，这些事件发生将会触发中断；

USART0/1/2完全实现上述功能，但是UART3/4只实现了上面所介绍的部分功能，下面这些功能在UART3/4中没有实现：

- 智能卡模式；
- 同步模式；
- 硬件流操作(CTS/RTS)；
- 设置数据极性。

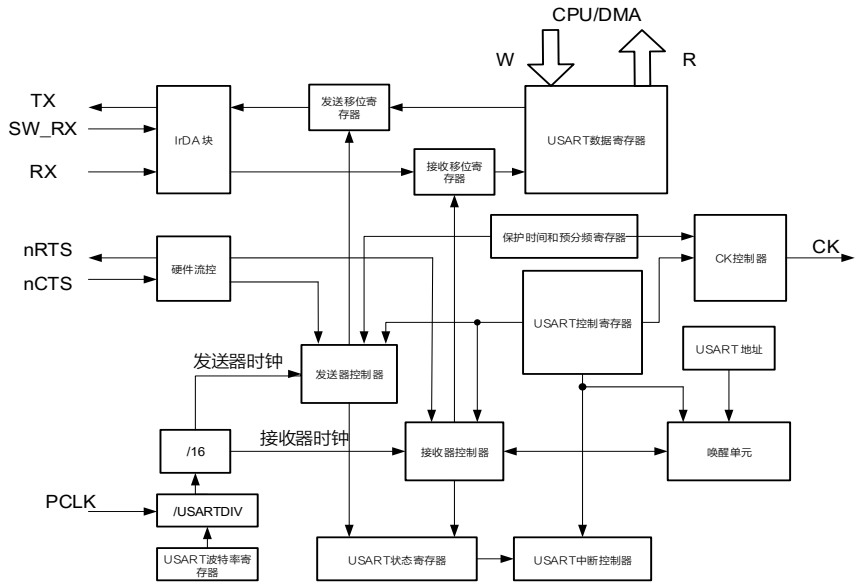
## 17.3. 功能说明

USART 接口通过*表 17-1. USART 重要引脚描述*中主要引脚从外部连接到其他设备。

**表 17-1. USART 重要引脚描述**

| 引脚 | 类型 | 描述 |
|---|---|---|
| RX | 输入 | 接收数据 |
| TX | 输出<br>I/O(单线模式/智能卡模式) | 发送数据。当 USART 使能后，若无数据发送，默认为高电平 |
| CK | 输出 | 用于同步通信的串行时钟信号 |
| nCTS | 输入 | 硬件流控模式发送使能信号 |
| nRTS | 输出 | 硬件流控模式发送请求信号 |

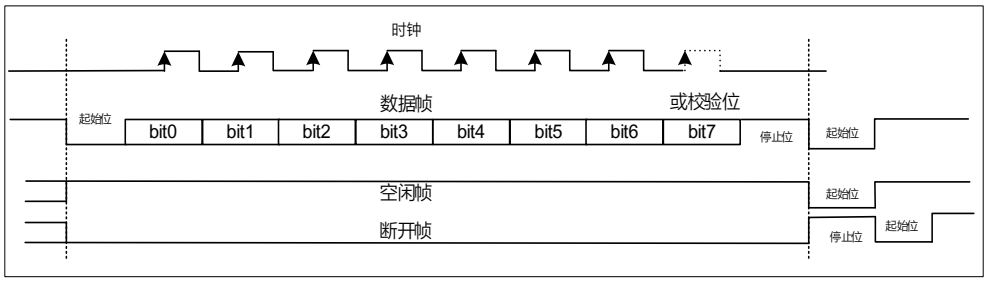
图 17-1. USART 模块内部框图




## 17.3.1. USART 帧格式

USART数据帧开始于起始位，结束于停止位。USART_CTL0寄存器中WL位可以设置数据长度。将USART_CTL0寄存器中PCEN置位，最后一个数据位可以用作校验位。若WL位为0，第七位为校验位。若WL位置1，第八位为校验位。USART_CTL0寄存器中PM位用于选择校验位的计算方法。

图 17-2. USART 字符帧(8 数据位和 1 停止位)




在发送和接收中，停止位可以由USART_CTL1寄存器中STB[1:0]位域配置。

表 17-2. 停止位配置

| STB[1:0] | 停止位长度(位) | 功能描述 |
|---|---|---|
| 00 | 1 | 默认值 |
| 01 | 0.5 | 智能卡模式接收 |
| 10 | 2 | 标准 USART，单线以及调制解调模式 |
| 11 | 1.5 | 智能卡模式发送和接收 |

在一个空闲帧中，所有位都为1。数据帧长度与正常USART数据帧长度相同。

紧随停止位后多个低电平为中断帧。USART数据帧的传输速度由UCLK时钟频率，波特率发生器的配置，以及过采样模式共同决定。

### 17.3.2. 波特率发生

波特率分频系数是一个16位的数字，包含12位整数部分和4位小数部分。波特率发生器使用这两部分组合所得的数值来确定波特率。由于具有小数部分的波特率分频系数，将使USART能够产生所有标准波特率。

如果过采样率是16，波特率分频系数（USARTDIV)与系统时钟具有如下关系：

$$\text{USARTDIV} = \frac{\text{UCLK}}{16 \times \text{Baud Rate}} \quad (17\text{-}1)$$

1. 由USART_BAUD寄存器的值得到USARTDIV：
   假设USART_BAUD=0x21D，则INTDIV=33(0x21)，FRADIV=13(0xD)。
   UASRTDIV=33+13/16=33.81。
2. 由USARTDIV得到USART_BAUD寄存器的值：
   假设要求UASRTDIV=30.37，INTDIV=30(0x1E)
   16*0.37=5.92，接近整数6，所以FRADIV=6(0x6)
   USART_BAUD=0x1E6。

**注意：**若取整后FRADIV=16(溢出)，则进位必须加到整数部分。

### 17.3.3. USART 发送器

如果USART_CTL0寄存器的发送使能位(TEN)被置位，当发送数据缓冲区不为空时，发送器将会通过TX引脚发送数据帧。TX引脚的极性可以通过USART_CTL3寄存器中TINV位来配置。时钟脉冲通过CK引脚输出。

TEN置位后发送器会发出一个空闲帧。TEN位在数据发送过程中是不可以被复位的。

系统上电后，TBE默认为高电平。在USART_STAT0寄存器中TBE置位时，数据可以在不覆盖前一个数据的情况下写入USART_DATA寄存器。当数据写入USART_DATA寄存器，TBE位将被清0。在数据由USART_DATA移入移位寄存器后，该位由硬件置1。如果数据在一个发送过程正在进行时被写入USART_DATA寄存器，它将首先被存入发送缓冲区，在当前发送过程完成时传输到发送移位寄存器中。如果数据在写入USART_DATA寄存器时，没有发送过程正在进行，TBE位将被清零然后迅速置位，原因是数据将立刻传输到发送移位寄存器。

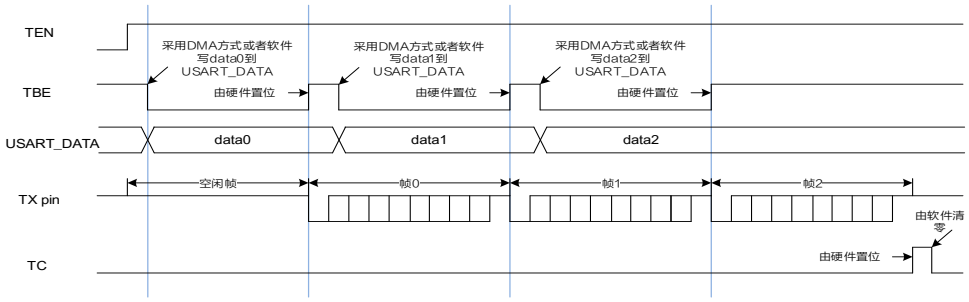
假如一帧数据已经发送出去，并且TBE位已经置位，那么USART_STAT0寄存器中TC位将被置1。如果USART_CTL0寄存器中的中断使能位(TCIE)为1，将会产生中断。

*图 17-3. USART 发送步骤*给出了 USART 发送步骤。软件操作按以下流程进行：

1. 在USART_CTL0寄存器中置位UEN位，使能USART；
2. 通过USART_CTL0寄存器的WL设置字长；
3. 在USART_CTL1寄存器中写STB[1:0]位来设置停止位的长度；
4. 如果选择了多级缓存通信方式，应该在USART_CTL2寄存器中使能DMA (DENT位)；
5. 在USART_BAUD寄存器中设置波特率；
6. 在USART_CTL0寄存器中设置TEN位；

7. 等待TBE置位；
8. 向USART_DATA寄存器写数据；
9. 若DMA未使能，每发送一个字节都需重复步骤7-8；
10. 等待TC=1，发送完成。

**图 17-3. USART 发送步骤**




在禁用USART或进入低功耗状态之前，必须等待TC置位。先读USART_STAT0然后再写USART_DATA可将TC位清0。在多级缓存通信方式(DENT=1)下，直接向TC写0，也能清TC。

### 17.3.4. USART 接收器

上电后，USART接收器使能按以下步骤进行：

1. 在USART_CTL0寄存器中置位UEN位，使能USART；
2. 写USART_CTL0寄存器的WL去设置字长；
3. 在USART_CTL1寄存器中写STB[1:0]位来设置停止位的长度；
4. 如果选择了多级缓存通信方式，应该在USART_CTL2寄存器中使能DMA(DENR位)；
5. 在USART_BAUD寄存器中设置波特率；
6. 在USART_CTL0中设置REN位。

接收器在使能后若检测到一个有效的起始脉冲便开始接收码流。在接收一个数据帧的过程中会检测噪声错误，奇偶校验错误，帧错误和过载错误。

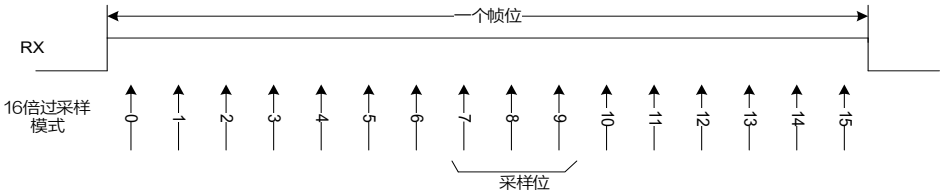
当接收到一个数据帧，USART_STAT0寄存器中的RBNE置位，如果设置了USART_CTL0寄存器中相应的中断使能位RBNEIE，将会产生中断。在USART_STAT0寄存器中可以观察接收状态标志。

软件可以通过读USART_DATA寄存器或者DMA方式获取接收到的数据。不管是直接读寄存器还是通过DMA，只要是对USART_DATA寄存器的一个读操作都可以清除RBNE位。

在接收过程中，需使能REN位，不然当前的数据帧将会丢失。

在默认情况下，接收器通过获取三个采样点的值来估计该位的值。如果在3个采样点中有2个或3个为0，该数据位被视为0，否则为1。如果3个采样点中有一个采样点的值与其他两个不同，不管是数据位，奇偶校验位或者停止位，都将产生噪声错误(NERR)。如果使能DMA，并置位USART_CTL2寄存器中ERRIE，将会产生中断。

**图 17-4. 过采样方式接收一个数据位**




通过置位USART_CTL0寄存器中的PCEN位使能奇偶校验功能，接收器在接收一个数据帧时计算预期奇偶校验值，并将其与接收到的奇偶校验位进行比较。如果不相等，USART_STAT0寄存器中PERR被置位。如果设置了USART_CTL0寄存器中的PERRIE位，将产生中断。

如果在停止位传输过程中RX引脚为0，将产生帧错误，USART_STAT0寄存器中FERR置位。如果使能DMA并置位USART_CTL2寄存器中ERRIE位，将产生中断。

当接收到一帧数据，而RBNE位还没有被清零，随后的数据帧将不会存储在数据接收缓冲区中。USART_STAT0寄存器中的溢出错误标志位ORERR将置位。如果使能DMA并置位USART_CTL2寄存器中ERRIE位或者置位RBNEIE，将产生中断。

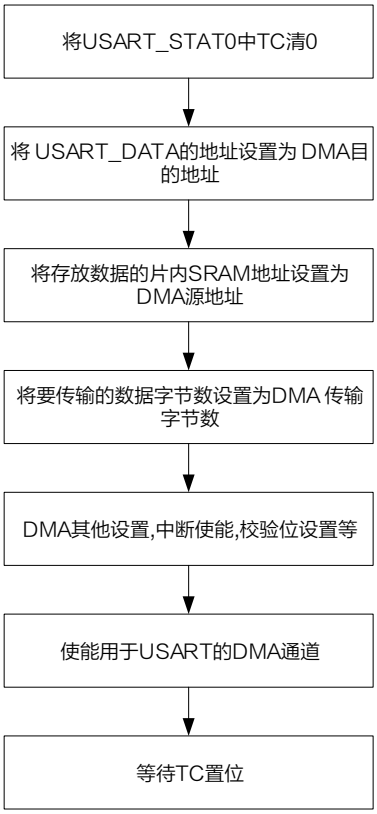
若接收过程中，产生了噪声错误(NERR)、校验错误(PERR)、帧错误(FERR)或溢出错误(ORERR)，则NERR、PERR、FERR或ORERR将和RBNE同时置位。如果没有使能DMA，RBNE中断发生时，软件需检查是否有噪声错误、校验错误、帧错误或溢出错误产生。

### 17.3.5. DMA 方式访问数据缓冲区

为减轻处理器的负担，可以采用DMA访问发送缓冲区或者接收缓冲区。置位USART_CTL2寄存器中DENT位可以使能DMA发送，置位USART_CTL2寄存器中DENR位可以使能DMA接收。

当DMA用于USART发送时，DMA将数据从片内SRAM传送到USART的数据缓冲区。配置步骤如*图17-5. 采用DMA方式实现USART数据发送配置步骤*所示。

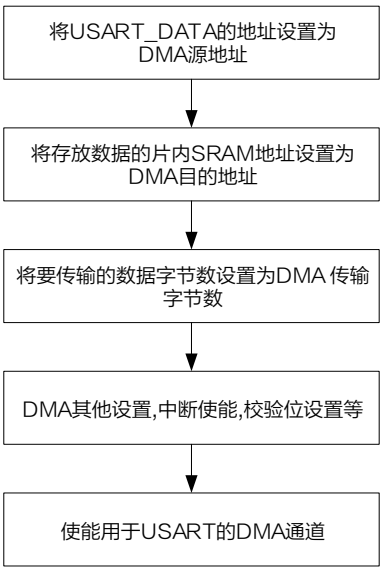
**图 17-5. 采用 DMA 方式实现 USART 数据发送配置步骤**




所有数据帧都传输完成后，USART_STAT0寄存器中TC位置1。如果USART_CTL0寄存器中TCIE置位，将产生中断。

当DMA用于USART接收时，DMA将数据从接收缓冲区传送到片内SRAM。配置步骤如*图17-6. 采用DMA方式实现USART数据接收配置步骤*所示。如果将USART_CTL2寄存器中ERRIE位置1，USART_STAT0寄存器中的错误标志位(FERR、ORERR和NERR)被置位时将产生中断。

**图 17-6. 采用 DMA 方式实现 USART 数据接收配置步骤**

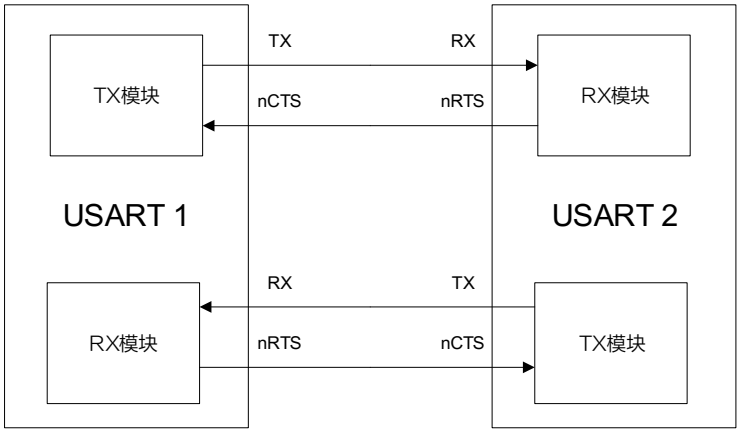


当USART接收到的数据数量达到了DMA传输数据数量，DMA模块将产生传输完成中断。

### 17.3.6. 硬件流控制

硬件流控制功能通过nCTS和nRTS引脚来实现。通过将USART_CTL2寄存器中RTSEN位置1来使能RTS流控，将USART_CTL2寄存器中CTSEN位置1来使能CTS流控。

**图 17-7. 两个 USART 之间的硬件流控制**



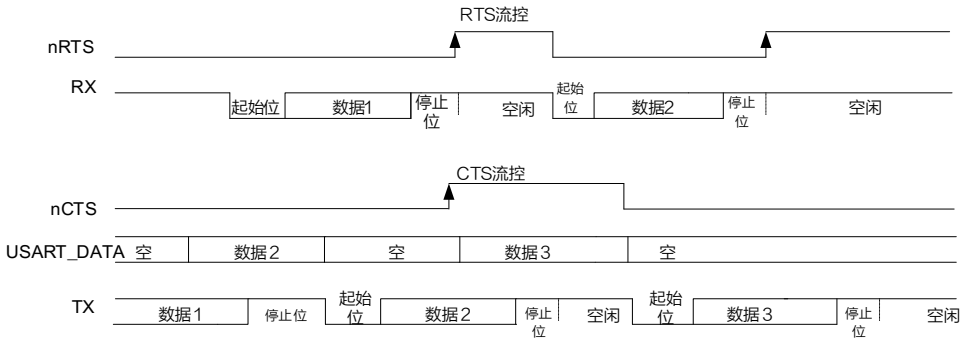

#### RTS 流控

USART接收器输出nRTS，它用于反映接收缓冲区状态。当一帧数据接收完成，nRTS变成高电平，这样是为了阻止发送器继续发送下一帧数据。当接收缓冲区满时，nRTS保持高电平，可以通过读USART_DATA寄存器来清零。

#### CTS 流控

USART发送器监视nCTS输入引脚来决定数据帧是否可以发送。如果USART_STAT0寄存器中TBE位是0且nCTS为低电平，发送器发送数据帧。在发送期间，若nCTS信号变为高电平，发送器将会在当前数据帧发送完成后停止发送。

**图 17-8. 硬件流控制**




如果CTS流控制被使能，在nCTS引脚信号发生变化时，USART_STAT0寄存器中CTSF位会置1。如果USART_CTL2寄存器中的CTSIE位被置位，将会产生中断。

### 17.3.7. 多处理器通信

在多处理器通信中，多个USART被连接成一个网络。对于一个设备来说，监视所有来自RX引脚的消息，是一种巨大的负担。为减轻设备负担，软件可以通过将USART_CTL0寄存器中RWU位置1使一个USART进入静默模式。

如果USART处于静默模式，所有的接收状态标志位将不会被置位。软件可以通过对RWU清零来唤醒USART。

此外，USART可以由硬件用以下两种方式中的一种来唤醒：空闲总线检测和地址匹配检测。

设备默认使用空闲总线检测方法唤醒USART。当在RX引脚检测到空闲帧时，硬件会将RWU清零，从而退出静默模式，但USART_STAT0寄存器中IDLEF位不会被置1。

当USART_CTL0寄存器中WM被置位，数据最高位会被认为是地址标志位。如果地址标志位为1，该字节被认为是地址字节。如果地址字节的低4位与USART_CTL1寄存器中的ADDR[3:0]相同，硬件会将RWU清零，并退出静默模式。接收到将USART唤醒的数据帧，RBNE将置位。状态标志可以从USART_STAT0寄存器中获取。如果地址字节的低4位与USART_CTL1寄存器中的ADDR[3:0]不相同，硬件会置位RWU并进入静默模式。在这种情况下，RBNE不会被置位。

如果采用地址匹配检测，默认情况下，接收器对地址字节不做奇偶校验。如果USART_CTL0寄存器中PCEN位被置位。地址字节最高位被视为校验位，其余位被视为地址位。

### 17.3.8. LIN 模式

将USART_CTL1寄存器的LMEN置位即可使能本地互联网络模式。

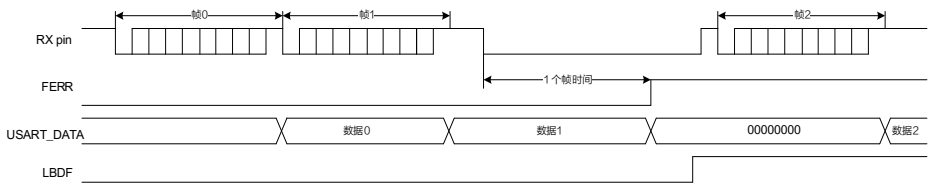
在LIN模式下，USART_CTL1寄存器中CKEN，WL，STB[1:0]以及USART_CTL2的SCEN，HDEN，IREN位都应该被清0。

在发送一个普通数据帧时，LIN发送过程与普通发送过程相同。数据位的长度只能是8。当USART_CTL0寄存器中SBKCMD置位时，USART在连续发送13个0，再发送完一个停止位。

断开检测功能完全独立于普通USART接收器。因此，断开检测可以是在空闲状态下，也可以在数据传输过程中。USART_CTL1寄存器中LBLEN位可以选择断开帧长度。如果在RX引脚检测到大于或等于与预期断开帧长度相等数量的0(LBLEN=0时，10个0；LBLEN=1时，11个0)，USART_STAT0寄存器中LBDF置位。如果USART_CTL1寄存器中LBDIE被置位，将产生中断。

如*图17-9. 空闲状态下检测断开帧*所示，如果断开帧发生在空闲状态下，USART接收器会接收到一个全0数据帧，同时FERR置位。

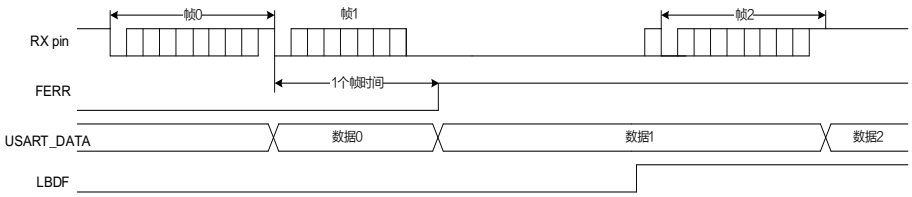
**图 17-9. 空闲状态下检测断开帧**




如*图17-10. 数据传输过程中检测断开帧*所示，如果断开帧发生在数据传输过程中，当前传输帧

发生错误，FERR置位。

**图 17-10. 数据传输过程中检测断开帧**




## 17.3.9. 同步通信模式

USART支持主机模式下的全双工同步串行通信，可以通过置位USART_CTL1的CKEN位来使能。在同步模式下，USART_CTL1的LMEN和USART_CTL2的SCEN，HDEN，IREN位应该被清0。CK引脚作为USART同步发送器的时钟输出，仅仅当TEN位被使能时，它才被激活。在起始位和停止位传送期间，不会从CK引脚输出时钟脉冲。USART_CTL1的CLEN位用来决定在最低位(地址索引位)发送期间是否有时钟信号输出。USART_CTL1的CPH位用来决定数据在第一个时钟沿被采样还是在第二个时钟沿被采样。USART_CTL1的CPL位用来决定在USART同步模式空闲状态下，时钟引脚的电平。

CK引脚输出波形由USART_CTL1寄存器中CPL，CPH，CLEN位决定。软件仅在USART禁用(UEN=0)时才可以改变它们的值。

如果USART_CTL0寄存器中REN置位，接收器的工作方式与普通模式下接收方式是不同的。接收器在时钟捕获沿采样数据，并无任何过采样。

**图 17-11. 同步模式下的 USART 示例**

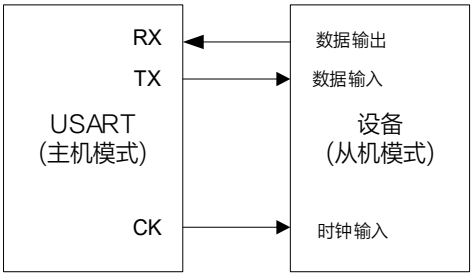
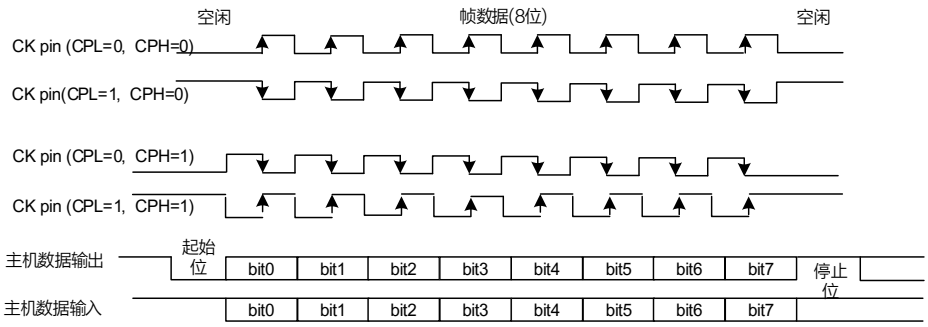



**图 17-12. 8-bit 格式的 USART 同步通信波形(CLEN=1)**

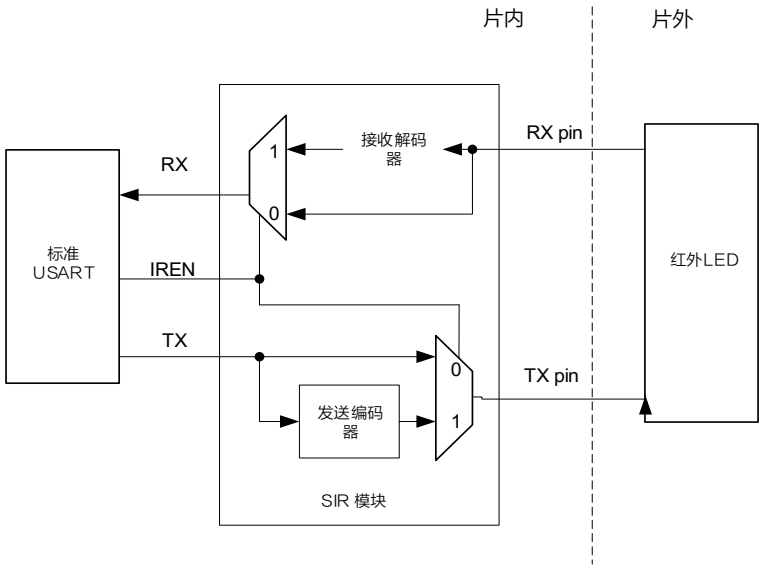


### 17.3.10. 串行红外(IrDA SIR)编解码功能模块

串行红外编解码功能通过置位USART_CTL2寄存器中IREN使能。在IrDA模式下，USART_CTL1寄存器的LMEN，STB[1:0]，CKEN位和USART_CTL2寄存器的HDEN，SCEN位将被清0。

在IrDA模式下，USART数据帧由SIR发送编码器进行调制，调制后的信号经由红外LED进行发送，经解调后将数据发送至USART接收器。对于编码器而言，波特率应小于115200。

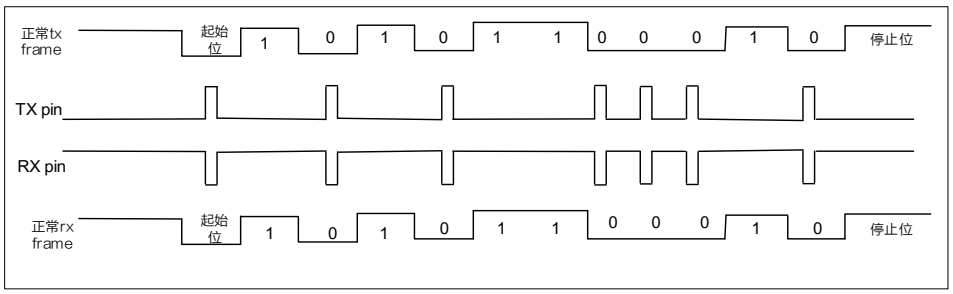
**图 17-13. IrDA SIR ENDEC 模块**




在IrDA模式下，TX引脚电平与RX引脚不同。TX引脚通常为低电平，RX引脚通常为高电平。IrDA引脚电平保持稳定代表逻辑'1'，红外光源脉冲(RTZ信号)代表逻辑'0'。其脉冲宽度通常占一个位时间的3/16。IrDA无法检测到宽度小于一个1个PSC时钟的脉冲。如果脉冲宽度大于1但是小于2倍PSC时钟，IrDA则无法可靠的检测到。

由于IrDA是一种半双工协议，因此在IrDA SIR ENDEC模块中，发送和接收不得同时进行。

**图 17-14. IrDA 数据调制**




将USART_CTL2寄存器中IRLP置位可以使SIR子模块工作在低功耗模式下。发送编码器由PCLK分频得到的低速时钟来驱动。分频系数在USART_GP寄存器中PSC[7:0]位配置，USART_BAUD寄存器需配置为16*PSC[7:0]。TX引脚脉冲宽度可以为低功耗波特率的3倍。接收器解码器工作模式与正常IrDA模式相同。

### 17.3.11. 半双工通信模式

通过设置USART_CTL2寄存器的HDEN位，可以使能半双工模式。

在半双工通信模式下，USART_CTL1寄存器的LMEN，CKEN位和USART_CTL2寄存器的SCEN，IREN位清零。

半双工模式下仅用单线通信，TX引脚和RX引脚从内部连接到一起，RX引脚不再使用。TX引脚应被配置为开漏模式，通信冲突由软件处理。

### 17.3.12. 智能卡(ISO7816-3)模式

智能卡模式是一种异步通信模式，支持ISO7816-3协议。支持字节模式(T=0)和块模式(T=1)。将USART_CTL2寄存器的SCEN位置1，即可使能智能卡模式。在智能卡模式下，USART_CTL1寄存器的LMEN位和USART_CTL2的HDEN，IREN位应该清0。

如果CKEN位被置位，USART通过CK引脚向智能卡提供一个由PCLK分频得到的时钟。分频系数可在USART_GP寄存器中PSC[4:0]配置。CK引脚只为智能卡提供时钟源。

智能卡模式是一种半双工通信协议模式。当与智能卡连接时，TX引脚需要被设置成开漏模式，外接上拉电阻，这个引脚将会与智能卡驱动同一条双向连线。智能卡模式下的帧格式为： 1起始位+9数据位(包括1奇偶校验位)+1.5停止位。其中0.5个停止位被配置为接收器的停止位。

**图 17-15. ISO7816-3 数据帧格式**




#### 字节模式(T=0)

相较于正常操作模式下的时序，从发送移位寄存器到TX引脚的传递时间延迟了半个波特率时钟，并且TC标志的置位将根据USART_GP寄存器的GUAT[7:0]设置延迟某一特定时间。在智能卡模式下，在最后一帧数据的停止位之后，内部保护时间计数器将开始计数，GUAT[7:0]的值配置为ISO7816-3协议的CGT减12。在保护时间寄存器向上计数这段时间TC将被强制拉低，当计数达到设定值时，TC被置位。

在USART发送期间，如果检测到有奇偶校验错误，TX引脚在停止位最后一个位时间内被拉低，智能卡发送一个NACK信号。根据协议，USART会自动重发SCRTNUM次。在重发数据帧前面会插入2位的帧间隔。最后一次重发字节后，TC会立即被置位。如果在最大重发次数后仍然收到NACK信号，USART将会停止发送，帧错误标志被置位。USART不会将NACK信号作为起始位。

在USART接收期间，如果在当前数据帧检测到校验错误，TX引脚在停止位的最后一个位时间

内会被拉低。智能卡会接收到NACK信号。然后在智能卡端会产生一个帧错误。如果接收到的字节是错误的，RBNE中断和接收DMA请求都不会被激活。根据协议，智能卡将要重新发送数据。如果在最大的重新发送次数后(这个次数的具体值在SCRTNUM位域)，接收到的字符仍然是错误的，USART停止发送NACK信号和标注这个错误为奇偶校验错误。将USART_CTL2寄存器中的NKEN置位可以使能NACK信号。

空闲帧和断开帧在智能卡模式下不适用。

### 块模式(T=1)

在T=1(块模式)下，USART_CTL2寄存器的NKEN位应该清零来关闭校验错误发送。

当要从智能卡读取数据时，软件必须将USART_RT寄存器设置成BWT (块等待)-11的值并将RBNEIE置位。这个超时时间体现在波特时间单元。如果这个时间到了，还没有从智能卡收到应答，USART_STAT1寄存器中RTF位被置位。如果设置了USART_CTL3寄存器中RTIE位，将会产生中断。如果在超时之前收到了第一个字节，则会引起RBNE中断。如果用DMA从智能卡读取数据，也只能在第一个字节接收好后再去使能DMA。

第一个字节接收到后，RT[23:0]的值设置成CWT(字节等待时间)-11来使能两个连续字节间最大帧间隔自动校验。如果在RT[23:0]周期内智能卡停止发送字节，USART_STAT1寄存器中RTF将被置位。

USART用一个块长度计数器统计收到的字节数，这个计数器在USART开始发送的时候自动清0(TBE=0)。这个块长度信息位于智能卡发出数据的第三个字节(序言部分)，这个值必须写入USART_RT寄存器BL[7:0]。块长度计数器从0开始计数到最大值BL[7:0]+4。在块计数器计数到最大值时，USART_STAT1寄存器中块结束状态标志位EBF置位。如果设置了USART_CTL3寄存器中的EBIE位，将会产生中断。如果块长度发生错误，RTF置位。

当使用DMA模式接收时，在块开始之前，这个寄存器必须被设定为最小值(0x0)。为了得到这个值，在收到第四个字节后，会引起一个中断。软件可以从接收缓冲区读取第三个字节作为块长度。

如果接收时不使用DMA方式，为避免产生EBF状态标志，BL[7:0]需首先配置为最大值0xFF。在收到第三个字节后，真正的块长度值可以重新写入到BL[7:0]。

### 直接和反向转换

智能卡协议定义了两种转换方式：直接转换和反向转换。

如果选择直接转换，从数据帧的最低位开始传输，TX引脚高电平代表逻辑'1'，偶校验。在这种情况下，USART_CTL3寄存器中MSBF位和DINV位都为0。

如果选择反向转换，从数据帧的最高位开始传输，TX引脚高电平代表逻辑'0'，偶校验。在这种情况下，USART_CTL3寄存器中MSBF位和DINV位都为1。

### 17.3.13. USART 中断

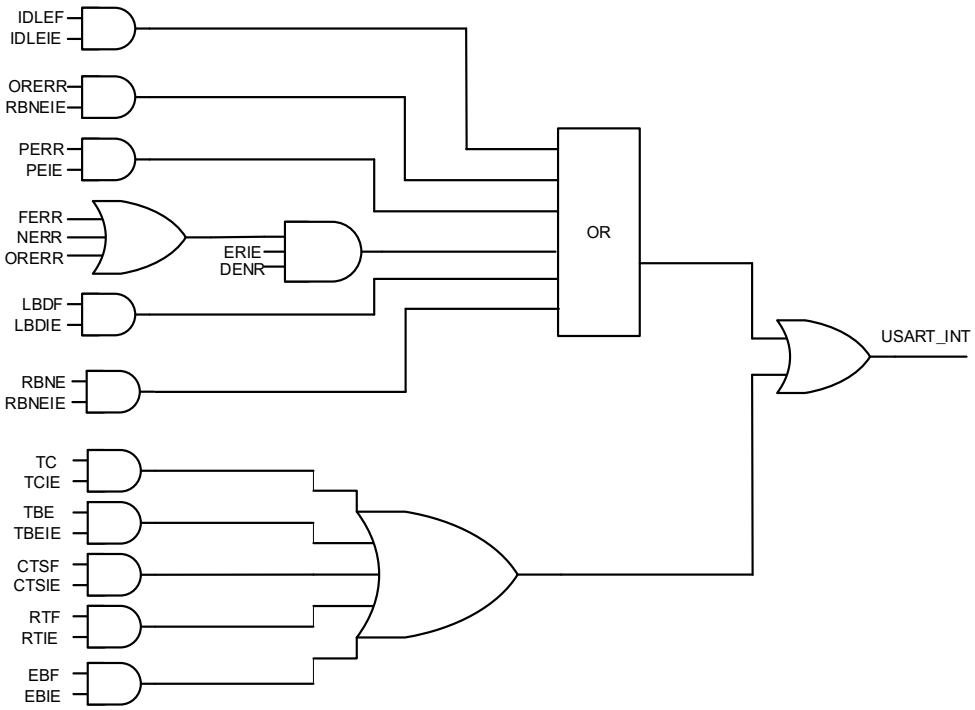
USART 中断事件和标志如*表 17-3. USART 中断请求*所示:

**表 17-3. USART 中断请求**

| 中断事件 | 事件标志 | 控制寄存器 | 使能控制位 |
|---|---|---|---|
| 发送数据寄存器空 | TBE | USART_CTL0 | TBEIE |
| CTS 标志 | CTSF | USART_CTL2 | CTSIE |
| 发送结束 | TC | USART_CTL0 | TCIE |
| 接收到的数据可以读取 | RBNE | USART_CTL0 | RBNEIE |
| 检测到过载错误 | ORERR | | |
| 检测到线路空闲 | IDLEF | USART_CTL0 | IDLEIE |
| 奇偶校验错误 | PERR | USART_CTL0 | PERRIE |
| LIN模式下，检测到断开标志 | LBDF | USART_CTL1 | LBDIE |
| 接收超时错误 | RTF | USART_CTL3 | RTIE |
| 发现块尾 | EBF | USART_CTL3 | EBIE |
| 接收错误(噪声错误、溢出错误、帧错误)当DMA接收使能时 | NERR或ORERR或FERR | USART_CTL2 | ERRIE |

在发送给中断控制器之前，所有的中断事件是逻辑或的关系。因此在任何时候 USART 只能向控制器产生一个中断请求。不过软件可以在一个中断服务程序里处理多个中断事件。

**图 17-16. USART 中断映射框图**



## 17.4. USART 寄存器

USART0 基地址：0x4001 3800

USART1 基地址：0x4000 4400

USART2 基地址：0x4000 4800

UART3 基地址：0x4000 4C00

UART4 基地址：0x4000 5000

### 17.4.1. 状态寄存器 0(USART_STAT0)

地址偏移：0x00
复位值：0x0000 00C0

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | CTSF | LBDF | TBE | TC | RBNE | IDLEF | ORERR | NERR | FERR | PERR |
| | | | | | | 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 | CTSF | CTS变化标志<br>如果设置了USART_CTL2寄存器中CTSEN位，当nCTS输入变化时，该位由硬件置位。如果设置了USART_CTL2寄存器中CTSIE位，将产生中断。<br>该位由软件清0。<br>0：nCTS状态线没有变化。<br>1：nCTS状态线发生变化。<br>该位对UART3/4无效。 |
| 8 | LBDF | LIN断开检测标志<br>寄存器USART_CTL1寄存器中LMEN置位，说明检测到LIN断开。如果USART_CTL1寄存器中LBDIE被置位时，将产生中断。<br>该位由软件清0。<br>0：没有检测到LIN断开字符。<br>1：检测到LIN断开字符。 |
| 7 | TBE | 发送数据缓冲区空<br>上电复位或待发送数据已发送至移位寄存器后，该位置1。USART_CTL0寄存器中TBEIE被置位将产生中断。<br>该位在软件将待发送数据写入USART_DATA时被清0。 |

0：发送数据缓冲区不为空。
1：发送数据缓冲区空。

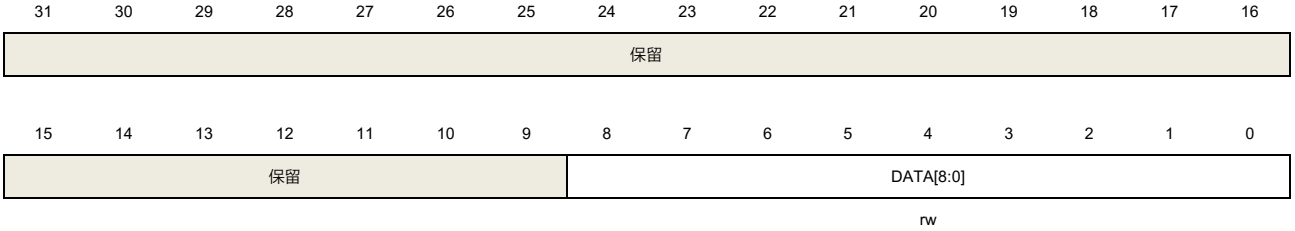
| | | |
|---|---|---|
| 6 | TC | 发送完成<br>上电复位后，该位被置1。如果TBE置位，在当前数据发送完成时该位置1。<br>USART_CTL0寄存器中TCIE被置位将产生中断。<br>该位由软件清0。<br>0：发送没有完成。<br>1：发送完成。 |
| 5 | RBNE | 读数据缓冲区非空<br>当读数据缓冲区接收到来自移位寄存器的数据时，该位置1。当寄存器USART_CTL0的RBNEIE位被置位，将会有中断产生。<br>软件可以通过对该位写0或读USART_DATA寄存器来将该位清0。<br>0：读数据缓冲区为空。<br>1：读数据缓冲区不为空。 |
| 4 | IDLEF | 空闲线检测标志<br>在一个帧时间内，在RX引脚检测到空闲状态，该位置1。当寄存器USART_CTL0的IDLEIE位被置位，将会有中断产生。<br>软件先读USART_STAT0，再读USART_DATA可清除该位。<br>0：未检测到空闲帧。<br>1：检测到空闲帧。 |
| 3 | ORERR | 溢出错误<br>在RBNE置位的情况下，如果USART_DATA寄存器接收到来自移位寄存器的数据，该位置1。当寄存器USART_CTL0的RBNEIE位被置位，将会有中断产生。在多处理器通信或者DMA模式下，当寄存器USART_CTL2的ERRIE位被置位，将会有中断产生。<br>软件先读USART_STAT0，再读USART_DATA可清除该位。<br>0：没有检测到溢出错误。<br>1：检测到溢出错误。 |
| 2 | NERR | 噪声错误标志<br>在接收数据时，如果在RX引脚检测到噪声，该位被置位。当寄存器USART_CTL2的ERRIE位被置位，将会有中断产生。<br>软件先读USART_STAT0，再读USART_DATA可清除该位。<br>0：没检测到噪声错误。<br>1：检测到噪声错误。 |
| 1 | FERR | 帧错误<br>接收数据期间，在停止位传输过程中，RX引脚检测到低电平，该位被置位。当寄存器USART_CTL2的ERRIE位被置位，将会有中断产生。<br>软件先读USART_STAT0，再读USART_DATA可清除该位。<br>0：未检测到帧错误。<br>1：检测到帧错误。 |
| 0 | PERR | 校验错误 |

当接收到的数据帧校验位与预期校验值不同时，该位置位。
软件先读USART_STAT0，再读USART_DATA可清除该位。
0：没检测到校验错误。
1：检测到校验错误。

### 17.4.2. 数据寄存器(USART_DATA)

地址偏移：0x04
复位值：未定义

该寄存器只能按字(32位)访问。



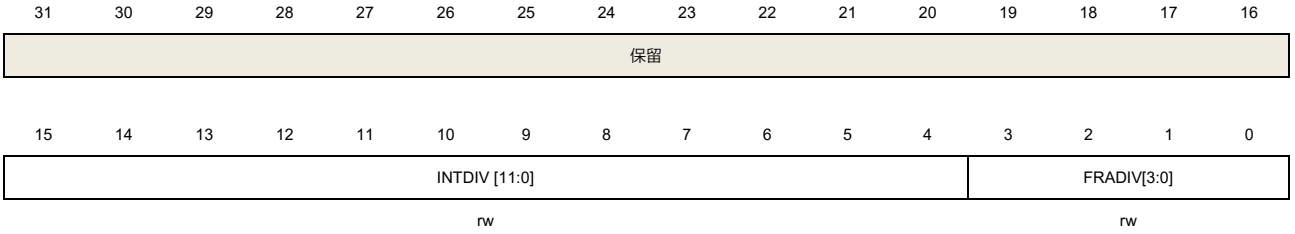

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:9 | 保留 | 必须保持复位值。 |
| 8:0 | DATA[8:0] | 发送或接收的数据值<br>软件可以通过写这些位来改变发送数据，或读这些位的值来获取接收数据。<br>如果使能了奇偶校验，当发送数据被写入寄存器，数据的最高位(第7位或第8位取决于USART_CTL0寄存器的WL位)将被校验位取代。 |

### 17.4.3. 波特率寄存器(USART_BAUD)

地址偏移：0x08
复位值：0x0000 0000

使能USART(UEN=1)时，不能写该寄存器。

该寄存器只能按字(32位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15:4 | INTDIV[11:0] | 波特率分频器的整数部分。 |

| | | |
|---|---|---|
| 3:0 | FRADIV [3:0] | 波特率分频器的小数部分。 |

### 17.4.4. 控制寄存器 0(USART_CTL0)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | UEN | WL | WM | PCEN | PM | PERRIE | TBEIE | TCIE | RBNEIE | IDLEIE | TEN | REN | RWU | SBKCMD |
| | | rw | rw | rw | rw | rw | rw | 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 | UEN | USART使能<br>0：USART禁用<br>1：USART使能 |
| 12 | WL | 字长<br>0：8数据位<br>1：9数据位 |
| 11 | WM | 从静默模式唤醒方法<br>0：空闲线<br>1：地址掩码 |
| 10 | PCEN | 校验控制使能<br>0：校验控制禁用<br>1：校验控制被使能 |
| 9 | PM | 校验模式<br>0：偶校验<br>1：奇校验 |
| 8 | PERRIE | 校验错误中断使能<br>如果该位置1，USART_STAT0寄存器中PERR被置位时产生中断。<br>0：校验错误中断禁用<br>1：校验错误中断使能 |
| 7 | TBEIE | 发送缓冲区空中断使能<br>如果该位置1，USART_STAT0寄存器中TBE被置位时产生中断。<br>0：发送缓冲区空中断禁止。<br>1：发送缓冲区空中断使能。 |

| | | |
|---|---|---|
| 6 | TCIE | 发送完成中断使能<br>如果该位置1，USART_STAT0寄存器中TC被置位时产生中断。<br>0：发送完成中断禁用。<br>1：发送完成中断使能。 |
| 5 | RBNEIE | 读数据缓冲区非空中断和过载错误中断使能<br>如果该位置1，USART_STAT0寄存器中RBNE或ORERR被置位时产生中断。<br>0：读数据缓冲区非空中断和过载错误中断禁用。<br>1：读数据缓冲区非空中断和过载错误中断使能。 |
| 4 | IDLEIE | IDLE线检测中断使能<br>如果该位置1，USART_STAT0寄存器中IDLEF被置位时产生中断。<br>0：IDLE线检测中断禁用。<br>1：IDLE线检测中断禁用使能。 |
| 3 | TEN | 发送器使能<br>0：发送器禁用<br>1：发送器使能 |
| 2 | REN | 接收器使能<br>0：接收器禁用<br>1：接收器使能 |
| 1 | RWU | 接收器从静默模式中唤醒<br>软件可以通过将该位置1使得USART进入静默模式，将该位清0唤醒USART。<br>空闲帧唤醒模式下(WM=0)，当检测到空闲帧时，该位由硬件清0。地址匹配模式下(WM=1)，当接收到一个地址匹配帧时，该位由硬件清0；或接收到一个地址非匹配帧时，由硬件置1。<br>0：接收器处于正常工作模式。<br>1：接收器处于静默模式。 |
| 0 | SBKCMD | 发送断开帧<br>软件通过将该位置1发送断开帧。<br>断开帧传输结束由硬件清0。<br>0：没有发送断开帧。<br>1：发送断开帧。 |

### 17.4.5. 控制寄存器 1(USART_CTL1)

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

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

| 保留 | LMEN | STB[1:0] | CKEN | CPL | CPH | CLEN | 保留 | LBDIE | LBLEN | 保留 | ADDR[3:0] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 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 | LMEN | LIN模式使能<br>0：LIN模式禁用<br>1：LIN模式使能 |
| 13:12 | STB[1:0] | STOP位长<br>00：1停止位<br>01：0.5停止位<br>10：2停止位<br>11：1.5停止位<br>对于UART3/4，只有1位停止位和2位停止位是有效的。 |
| 11 | CKEN | CK引脚使能<br>0：CK引脚禁用<br>1：CK引脚使能<br>该位对于UART3/4无效。 |
| 10 | CPL | 时钟极性<br>该位用来设定在同步模式下CK引脚的极性。<br>0：CK引脚不对外发送时保持为低电平。<br>1：CK引脚不对外发送时保持为高电平。<br>该位对于UART3/4无效。 |
| 9 | CPH | 时钟相位<br>该位用来设定在同步模式下CK引脚的相位。<br>0：在首个时钟边沿采样第一个数据。<br>1：在第二个时钟边沿采样第一个数据。<br>该位对于UART3/4无效。 |
| 8 | CLEN | CK信号长度<br>该位用来设定在同步模式下CK信号的长度。<br>0：8位数据帧中有7个CK脉冲，9位数据帧中有8个CK脉冲。<br>1：8位数据帧中有8个CK脉冲，9位数据帧中有9个CK脉冲。<br>该位对于UART3/4无效。 |
| 7 | 保留 | 必须保持复位值。 |
| 6 | LBDIE | LIN断开信号检测中断使能<br>如果该位置1，当USART_STAT0寄存器中LBDF被置位时将产生中断。<br>0：断开信号检测中断禁用。<br>1：断开信号检测中断使能。 |
| 5 | LBLEN | LIN断开帧长度<br>该位用来设定在断开帧长度。 |

| | | |
|---|---|---|
| | | 0：10位<br>1：11位 |
| 4 | 保留 | 必须保持复位值。 |
| 3:0 | ADDR[3:0] | USART地址<br>地址掩码唤醒模式下(WM=1)，如果接收到的数据帧低四位与ADDR[3:0]值不相等，USART就会进入静默模式；如果接收到的数据帧低四位与ADDR[3:0]值相等，USART就会被唤醒。 |

### 17.4.6. 控制寄存器 2(USART_CTL2)

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | CTSIE | CTSEN | RTSEN | DENT | DENR | SCEN | NKEN | HDEN | IRLP | IREN | ERRIE |
| | | | | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

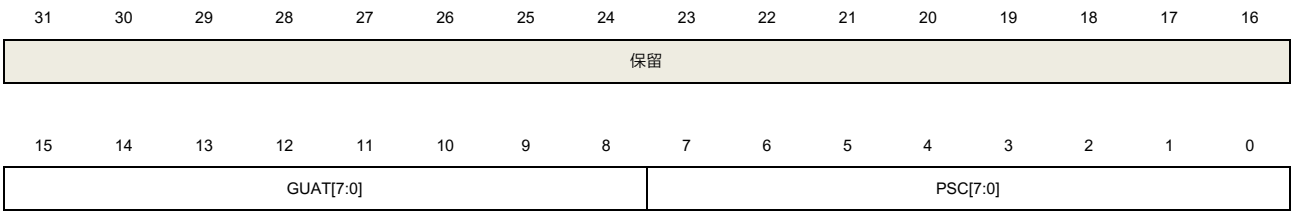
| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:11 | 保留 | 必须保持复位值。 |
| 10 | CTSIE | CTS中断使能<br>如果该位置1，当USART_STAT0寄存器中CTSF被置位时将产生中断。<br>0：CTS中断禁用。<br>1：CTS中断使能。<br>该位对于UART3/4无效。 |
| 9 | CTSEN | CTS使能<br>该位用于使能CTS硬件流控制功能。<br>0：CTS硬件流控制禁用。<br>1：CTS硬件流控制使能。<br>该位对于UART3/4无效。 |
| 8 | RTSEN | RTS使能<br>该位用于使能RTS硬件流控制功能。<br>0：RTS硬件流控制禁用。<br>1：RTS硬件流控制使能。<br>该位对于UART3/4无效。 |
| 7 | DENT | DMA发送使能<br>0：DMA发送模式禁用。<br>1：DMA发送模式使能。 |

| | | |
|---|---|---|
| 6 | DENR | DMA接收使能<br>0：DMA接收模式禁用。<br>1：DMA接收模式使能。 |
| 5 | SCEN | 智能卡模式使能<br>该位用于使能智能卡模式。<br>0：智能卡模式禁用。<br>1：智能卡模式使能。<br>该位对于UART3/4无效。 |
| 4 | NKEN | 在智能卡模式NACK使能<br>该位用于智能卡模式在奇偶校验错误发生时使能NACK发送。<br>0：当出现校验错误时不发送NACK。<br>1：当出现校验错误时发送NACK。<br>该位对于UART3/4无效。 |
| 3 | HDEN | 半双工使能<br>该位用于使能半双工模式。<br>0：半双工模式禁用。<br>1：半双工模式使能。 |
| 2 | IRLP | IrDA低功耗模式<br>该位用于为IrDA模式选择低功耗模式。<br>0：正常模式<br>1：低功耗模式 |
| 1 | IREN | IrDA模式使能<br>0：IrDA禁用<br>1：IrDA使能 |
| 0 | ERRIE | 错误中断使能<br>当DMA接收模式(DENR=1)使能时，如果该位被置1，USART_STAT0寄存器中FERR，ORERR，NERR被置位将产生中断。<br>0：错误中断禁用<br>1：错误中断使能。 |

### 17.4.7. 保护时间和预分频器寄存器(USART_GP)

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



rw rw

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15:8 | GUAT[7:0] | 智能卡模式下的保护时间值。<br>TC标志置位时间延时GUAT[7:0]个波特时钟周期。<br>该位对于UART3/4无效。 |
| 7:0 | PSC[7:0] | 使能USART IrDA低功耗模式，这些位用来设定将外设时钟(PCLK1/PCLK2)分频产生低功耗频率的分频系数。<br>00000000：保留 - 不要写入该值。<br>00000001：对源时钟1分频。<br>…<br>11111111：对源时钟255分频。<br>在IrDA正常模式下，PSC只能设置成00000001。<br>在智能卡模式下，PSC[4:0]用于设定外设时钟(APB1/APB2)生成智能卡时钟的分频系数。实际的分频系数为PSC[4:0]设定值的两倍。<br>00000：保留 - 不要写入该值。<br>00001：对源时钟2分频。<br>00010：对源时钟4分频。<br>…<br>11111：对源时钟62分频。<br>在智能卡模式下，PSC[7:5]保留。 |

## 17.4.8. 控制寄存器 3(USART_CTL3)

偏移地址：0x80
复位值：0x0000 0000

UART3/4未使用该寄存器

该寄存器只能按字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | MSBF | DINV | TINV | RINV | 保留 | | EBIE | RTIE | SCRTNUM[2:0] | | | RTEN |
| | | | | 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 | MSBF | 高位在前<br>该位用于设定数据在发送或接收时的顺序。<br>0：数据发送/接收，采用低位在前。 |

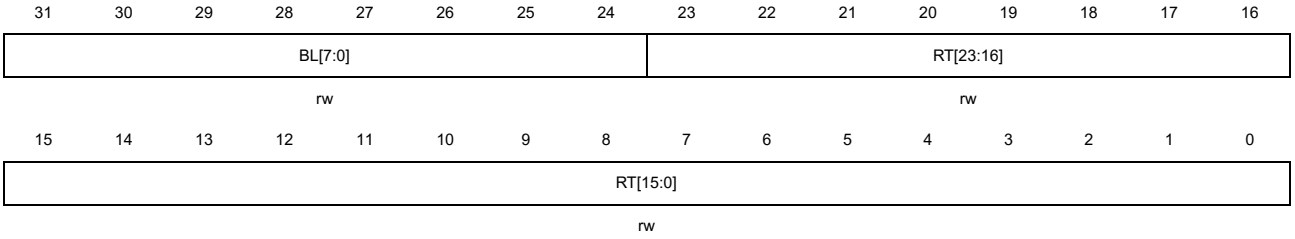
| | | |
|---|---|---|
| | | 1：数据发送/接收，采用高位在前。<br>USART被使能(UEN=1)时，这一位不能被改写。 |
| 10 | DINV | 数据位反转<br>该位用于设定在发送或接收时数据位的极性。<br>0：数据位信号值没有反转。<br>1：数据位信号值被反转。<br>USART被使能(UEN=1)时，这一位不能被改写。 |
| 9 | TINV | TX 引脚电平反转<br>该位用于设定TX引脚极性。<br>0：TX引脚信号值没有反转。<br>1：TX引脚信号值被反转。<br>USART被使能(UEN=1)时，这一位不能被改写。 |
| 8 | RINV | RX引脚电平反转<br>该位用于设定RX引脚极性。<br>0：RX引脚信号值没有反转。<br>1：RX引脚信号值被反转。<br>USART被使能(UEN=1)时，这一位不能被改写。 |
| 7:6 | 保留 | 必须保持复位值。 |
| 5 | EBIE | 块结束标志中断使能位<br>如果该位置1，USART_STAT1寄存器中EBF被置位时产生中断。<br>0：块中断使能。<br>1：块中断禁用。 |
| 4 | RTIE | 接收超时标志中断使能位<br>如果该位置1，USART_STAT1寄存器中RTF被置位时产生中断。<br>0：接收超时中断使能。<br>1：接收超时中断禁用。 |
| 3:1 | SCRTNUM[2:0] | 智能卡自动重试次数寄存器<br>在智能卡模式下，这些位用来设定在发送和接收时重试的次数。<br>在发送模式下，一帧数据可以重发SCRTNUM次。如果一帧数据发送失败SCRTNUM+1次，FERR被置位。<br>在接收模式下，USART接收一个数据帧可以执行SCRTNUM+1次。如果一个数据帧校验位不匹配事件产生SCRTNUM+1次，RBNE位和PERR位被置位。<br>当这些位被设置为0x0时，在发送模式下这些位将不会自动发送。 |
| 0 | RTEN | 接收器超时使能<br>该位用于使能USART接收超时。<br>0：接收器超时检测功能禁用。<br>1：接收器超时检测功能被使能。 |

### 17.4.9. 接收超时寄存器(USART_RT)

偏移地址：0x84
复位值：0x0000 0000

UART3/4未使用该寄存器。

该寄存器只能按字(32位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:24 | BL[7:0] | 块长度<br>这些位用于设定智能卡T=1的接收时，块的长度。它的值等于信息字节的长度+结束域的长度(1-LEC/2-CRC) - 1。<br>这个值可以在块接收开始去设置(采用从序言域中提取的LEN字段)，这个值在每一个接收时钟周期只能设置一次。在智能卡模式下，当TBE=0时，块的长度计数器被清0。在其他模式下，当REN=0 (禁用接收器)或者当USART_STAT1寄存器的EBF位被写0时，块的长度计数器被清0。 |
| 23:0 | RT[23:0] | 接收器超时阈值<br>该位域用于指定接收超时值，单位是波特时钟的时长。<br>标准模式下，如果在最后一个字节接收后，在RT规定的时长内，没有检测到新的起始位，USART_STAT1寄存器中RTF标志被置位。<br>在智能卡模式，这个值被用来实现CWT和BWT。在这种情况下，超时检测是从最后一个接收字节的起始位开始算的。<br>这些位可以在工作时改写。假如一个新数据到来的时间比RT规定的晚，RTF标志会被置位。对于每个接收字符，这个值只能改写一次。<br>**注意：**当16倍过采样时，RT[23:0]不能被配置为0xFFFFFF。 |

### 17.4.10. 状态寄存器 1(USART_STAT1)

偏移地址：0x88
复位值：0x0000 0000

UART3/4未使用该寄存器

该寄存器只能按字(32位)访问。



| 保留 | EBF | RTF | 保留 |
|---|---|---|---|
| | w0 | w0 | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:17 | 保留 | 必须保持复位值。 |
| 16 | BSY | 忙标志<br>USART接收一帧数据时被置位。<br>0：USART接收通道空闲<br>1：USART接收通道忙 |
| 15:13 | 保留 | 必须保持复位值。 |
| 12 | EBF | 块结束标志<br>该位在接收字节数(从块起始开始计数，包含序言)等于或者大于BLEN+4时被置位。USART_CTL3寄存器中EBIE被置位将产生中断。<br>软件可以通过写0清除该位。<br>0：块结束事件没有发生。<br>1：块结束事件发生。 |
| 11 | RTF | 接收超时标志<br>该位在RX引脚空闲时间已经超过RT值时被置位。USART_CTL3寄存器中RTIE被置位将产生中断。<br>软件可以通过写0清除该位。<br>0：接收器超时事件没有发生。<br>1：接收器超时事件发生。 |
| 10:0 | 保留 | 必须保持复位值。 |

## 17.4.11. 兼容性控制寄存器(USART_CHC)

偏移地址：0xC0
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | EPERR | 保留 | | | | | | | HCM |
| | | | | | | | w0c | | | | | | | | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:9 | 保留 | 必须保持复位值。 |
| 8 | EPERR | 校验错误超前检测标志。<br>该标志将在RBNE置位前，校验位被检测到时置位。<br>软件写0可以清除该位。 |

| | | |
|---|---|---|
| | | 0：没有检测到校验错误。<br>1：检测到校验错误。 |
| 7:1 | 保留 | 必须保持复位值。 |
| 0 | HCM | 硬件流控制兼容模式<br>0：nRTS信号与USART_STAT0寄存器中RBNE位相同。<br>1：nRTS信号在最后一个数据位被采样后被置位。 |

# 18. 内部集成电路总线接口(I2C)

## 18.1. 简介

I2C(内部集成电路总线)模块提供了符合工业标准的两线串行制接口，可用于 MCU 和外部 I2C 设备的通讯。I2C 总线使用两条串行线：串行数据线 SDA 和串行时钟线 SCL。

I2C 接口模块实现了 I2C 协议的标速模式，快速模式以及快速+ 模式，具备 CRC 计算和校验功能、支持 SMBus(系统管理总线)、PMBus(电源管理总线)和 SAM_V(验证安全控制模块)模式，此外还支持多主机 I2C 总线架构。I2C 接口模块也支持 DMA 模式，可有效减轻 CPU 的负担。

## 18.2. 主要特征

- 并行总线至 I2C 总线协议的转换及接口。
- 同一接口既可实现主机功能又可实现从机功能。
- 主从机之间的双向数据传输。
- 支持 7 位和 10 位的地址模式和广播寻址。
- 支持 I2C 多主机模式。
- 支持标速模式(最高 100 KHz)，快速模式(最高 400 KHz)和快速+ 模式(最高 1 MHz)。
- 从机模式下可配置的 SCL 主动拉低。
- 支持 DMA 模式。
- 兼容 SMBus 2.0 和 PMBus。
- 两个中断：字节成功传输中断和错误事件中断。
- 可选择的 PEC(报文错误校验)生成和校验。
- 支持 SAM_V 模式。

## 18.3. 功能说明

I2C 接口的内部结构如*图 17-17. I2C 模块框图*所示。

**图 17-17. I2C 模块框图**

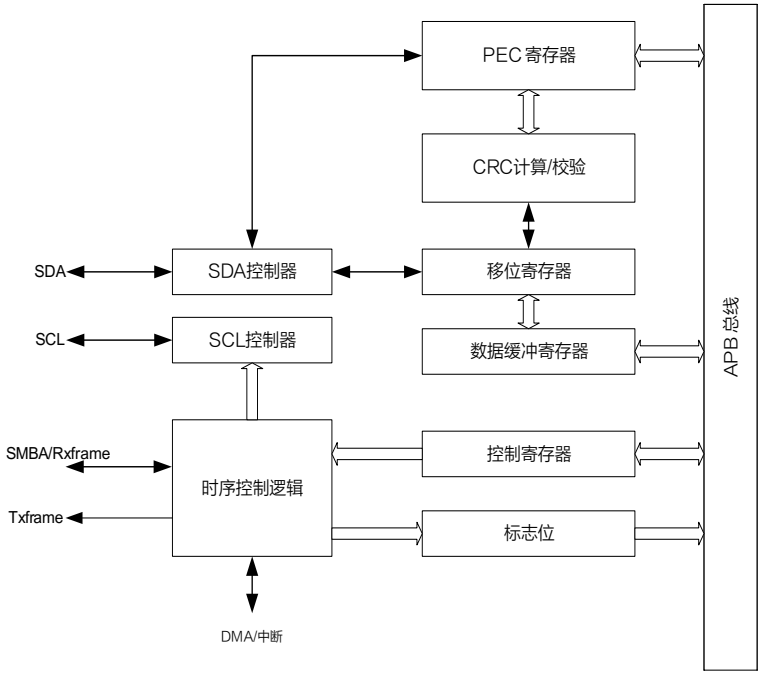



**表 17-4. I2C 总线术语说明(参考飞利浦 I2C 规范)**

| 术语 | 说明 |
|---|---|
| 发送器 | 发送数据到总线的设备 |
| 接收器 | 从总线接收数据的设备 |
| 主机 | 初始化数据传输，产生时钟信号和结束数据传输的设备 |
| 从机 | 由主机寻址的设备 |
| 多主 | 多个主机可以尝试在不破坏信息的前提下同时控制总线 |
| 同步 | 同步两个或更多设备之间的时钟信号的过程 |
| 仲裁 | 如果超过一个主机同时试图控制总线，只有一个主机被允许，且获胜主机的信息不被破坏 |

### 18.3.1. SDA 线和 SCL 线

I2C 模块有两条接口线：串行数据 SDA 线和串行时钟 SCL 线。连接到总线上的设备通过这两根线互相传递信息。

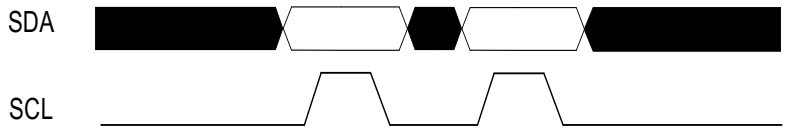
SDA 和 SCL 都是双向线，通过一个电流源或者上拉电阻接到电源正极。当总线空闲时，两条线都是高电平。连接到总线的设备输出极必须是开漏或者开集，以提供线与功能。I2C 总线上的数据在标准模式下可以达到 100 Kbit/s，在快速模式下可以达到 400 Kbit/s，当 I2C_FMPCFG 寄存器中 FMPEN 位被置位时，在快速+ 模式下可达 1 Mbit/s。由于 I2C 总线上可能会连接不

同工艺的设备(CMOS，NMOS，双极性器件)，逻辑'0'和逻辑'1'的电平并不是固定的，取决于$V_{DD}$的实际电平。

### 18.3.2. 数据有效性

时钟信号的高电平期间 SDA 线上的数据必须稳定。只有在时钟信号 SCL 变低的时候数据线 SDA 的电平状态才能跳变(如*图 17-18. 数据有效性*)。每个数据比特传输需要一个时钟脉冲。

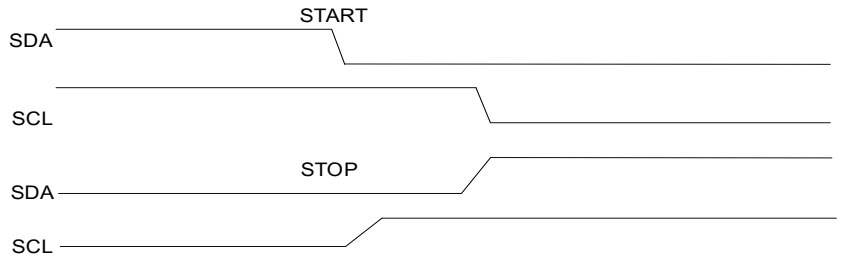
**图 17-18. 数据有效性**




### 18.3.3. 开始和停止信号

所有的数据传输起始于一个 START 结束于一个 STOP(参见*图 17-19. 起始和停止信号*)。START 信号定义为，在 SCL 为高时，SDA 线上出现一个从高到低的电平转换。STOP 信号定义为，在 SCL 为高时，SDA 线上出现一个从低到高的电平转换。

**图 17-19. 起始和停止信号**



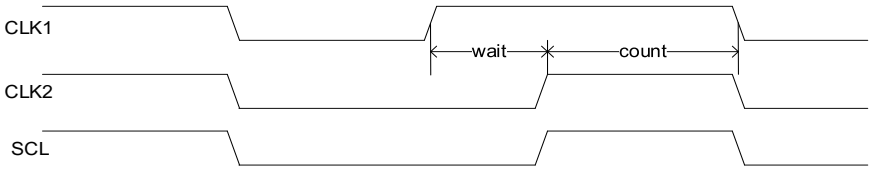

### 18.3.4. 时钟同步

两个主机可以同时在空闲总线上开始传送数据，因此必须通过一些机制来决定哪个主机获取总线的控制权并完成传输，这一般是通过时钟同步和仲裁来完成的。单主机系统下不需要时钟同步和仲裁机制。

时钟同步通过 SCL 线的线与来实现。这就是说 SCL 线的高到低切换会使器件开始计数它们的低电平周期，而且当主机的时钟变低电平时，它会使 SCL 线保持这种状态直到到达时钟的高电平(参见*图 17-20. 时钟同步*)。但是如果另一个时钟仍处于低电平周期，这个时钟的低到高切换不会改变 SCL 线的状态。因此 SCL 线被有最长低电平周期的器件保持低电平。此时低电平

周期短的器件会进入高电平的等待状态。

**图 17-20. 时钟同步**



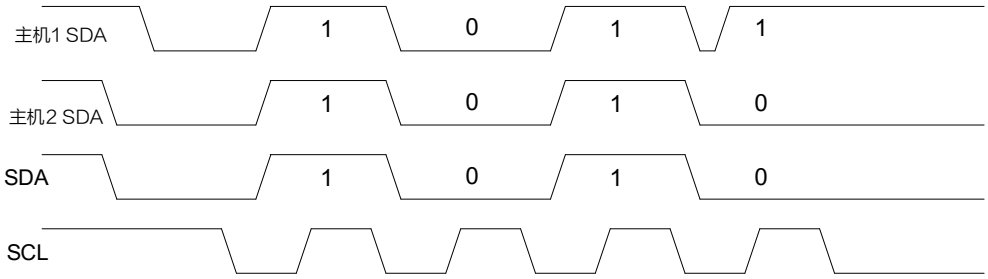

### 18.3.5. 仲裁

仲裁和同步一样，都是为了解决多主机情况下的总线控制冲突。仲裁的过程与从机无关。

只有在总线空闲的时候主机才可以启动传输。两个主机可能在START信号的最短保持时间内在总线上产生一个有效的START信号，这种情况下需要仲裁来决定由哪个主机来完成传输。

仲裁逐位进行，在每一位的仲裁期间，当SCL为高时，每个主机都检查SDA电平是否和自己发送的相同。仲裁的过程需要持续很多位。理论上讲，如果两个主机所传输的内容完全相同，那么它们能够成功传输而不出现错误。如果一个主机发送高电平但检测到SDA电平为低，则认为自己仲裁失败并关闭自己的SDA输出驱动，而另一个主机则继续完成自己的传输。

**图 17-21. SDA 线仲裁**




### 18.3.6. I2C 通讯流程

每个I2C设备(不管是微控制器，LCD驱动，存储器或者键盘接口)都通过唯一的地址进行识别，根据设备功能，他们既可以是发送器也可作为接收器。

I2C从机检测到I2C总线上的START信号之后，就开始从总线上接收地址，之后会把从总线接收到的地址和自身的地址(通过软件编程)进行比较，当两个地址相同时，I2C从机将发送一个确认应答(ACK)，并响应总线的后续命令：发送或接收所需数据。此外，如果软件开启了广播呼叫，则I2C从机始终对一个广播地址(0x00)发送确认应答。I2C模块始终支持7位和10位的地址。

I2C主机负责产生START信号和STOP信号来开始和结束一次传输，并且负责产生SCL时钟。

**图 17-22. 7 位地址的 I2C 通讯流程**

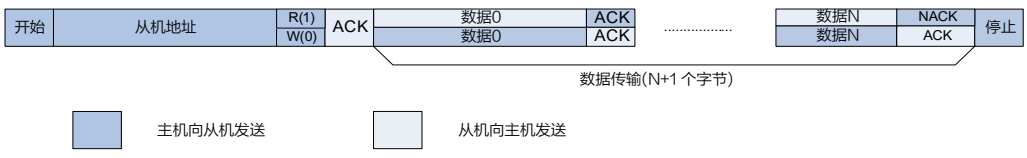


图 17-23. 10 位地址的 I2C 通讯流程(主机发送)

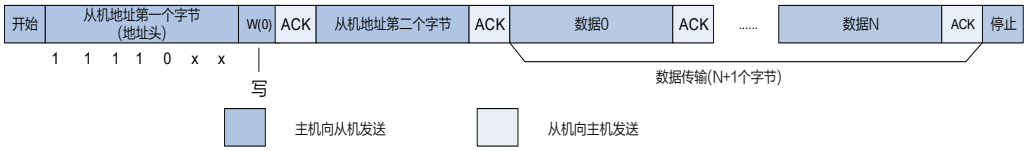
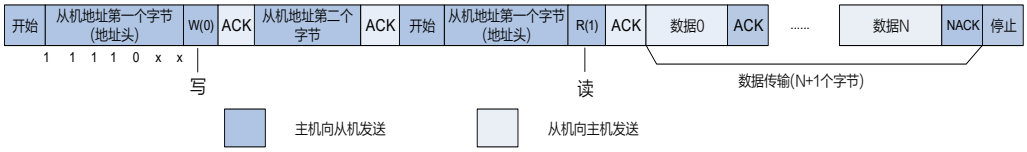



图 17-24. 10 位地址的 I2C 通讯流程(主机接收)




## 18.3.7. 软件编程模型

一个I2C设备例如LCD驱动器可能只是作为一个接收器，但是一个存储器既可以接收数据，也能发送数据。除了按照发送/接收方来区分，I2C设备也分为数据传输的主机和从机。主机是指负责初始化总线上数据的传输并产生时钟信号的设备，此时任何被寻址的设备都是从机。

不管I2C设备是主机还是从机，都可以发送或接收数据，因此，I2C设备有以下4种运行模式：

- 主机发送方
- 主机接收方
- 从机发送方
- 从机接收方

I2C模块支持以上四种模式。系统复位以后，I2C默认工作在从机模式下。通过软件配置使I2C在总线上发送一个START信号之后，I2C变为主机模式，软件配置在I2C总线上发送STOP信号后，I2C又变回从机模式。

### 从机发送模式下的软件流程

如*图17-25. 从机发送模式(10位地址模式)*所示，在从机模式下要发送数据，软件应该按照以下步骤来运行操作：

1. 首先，软件应该使能I2C外设时钟，以及配置I2C_CTL1中时钟相关寄存器来确保正确的I2C时序。使能和配置以后，I2C运行在默认的从机模式状态，等待I2C总线上的START信号和地址。
2. 当接收到一个START信号及随后的地址后，地址可以是7位格式也可以是10位格式，I2C硬件将I2C_STAT0寄存器的ADDSEND位置1，此位应该被软件查询或者中断监视，发现置位后，软件应该读I2C_STAT0寄存器然后读I2C_STAT1寄存器来清除ADDSEND位。如果地址是10位格式，I2C主机应该接着再产生一个START并发送一个地址头到I2C总线。从机在检测到START和紧接着的地址头之后会继续将ADDSEND位置1。软件可以通过读I2C_STAT0寄存器和接着读I2C_STAT1寄存器来第二次清除ADDSEND位。
3. 现在I2C进入数据发送状态，由于移位寄存器和数据寄存器I2C_DATA都是空的，硬件将

TBE位置1。软件此时可以写入第一个字节数据到I2C_DATA寄存器，但是TBE位并没有被清0，因为写入I2C_DATA寄存器的字节被立即移入内部移位寄存器。当移位寄存器非空的时候，I2C开始发送数据到I2C总线。

4. 第一个字节的发送期间，软件可以写第二个字节到I2C_DATA，此时TBE位被清0，因为I2C_DATA寄存器和移位寄存器都不是空。

5. 第一个字节的发送完成之后，TBE被再次置起，软件可以写第三个字节到I2C_DATA，同时TBE位被清0。在此之后，任何时候TBE被置1，只要依然有数据待被发送，软件都可以写入一个字节到I2C_DATA寄存器。

6. 倒数第二个字节发送期间，软件写最后一个数据到I2C_DATA寄存器来清除TBE标志位，之后就再不用关心TBE的状态。TBE位会在倒数第二个字节发送完成后置起，直到检测到STOP信号时被清0。

7. 根据I2C协议，I2C主机将不会对接收到的最后一个字节发送应答，所以在最后一个字节发送结束后，I2C从机的AERR(应答错误)会置起以通知软件发送结束。软件写0到AERR位可以清除此位。

**图 17-25. 从机发送模式(10 位地址模式)**

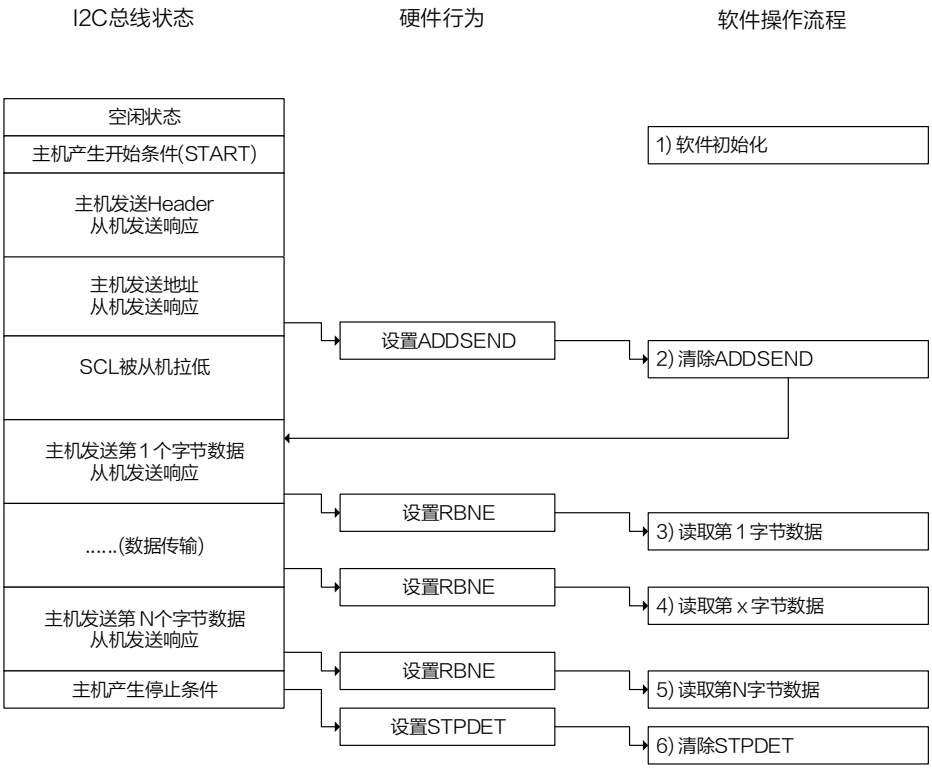


### 从机接收模式下的软件流程

如*图 17-26. 从机接收模式(10 位地址模式)*所示，在从机模式下接收数据时，软件应该遵循这些步骤来操作：

1. 首先，软件应该使能I2C外设时钟，以及配置I2C_CTL1中时钟相关寄存器来确保正确的I2C时序。使能和配置以后，I2C运行在默认的从机模式状态，等待START信号以及地址。
2. 在接收到START起始信号和匹配的7位或10地址之后，I2C硬件将I2C状态寄存器0的ADDSEND位置1，此位应该通过软件轮询或者中断来检测，发现置起后，软件通过先读I2C_STAT0寄存器然后读I2C_STAT1寄存器来清除ADDSEND位。当ADDSEND位被清0时，I2C就开始接收来自I2C总线的数据。
3. 当接收到第一个字节时，RBNE位被硬件置1，软件可以读取I2C_DATA寄存器的第一个字节，此时RBNE位也被清0。
4. 任何时候RBNE被置1，软件可以从I2C_DATA寄存器读取一个字节。
5. 接收到最后一个字节后，RBNE被置1，软件可以读取最后的字节。
6. 当I2C检测到I2C总线上一个STOP信号，STPDET位被置1，软件通过先读I2C_STAT0寄存器再写I2C_CTL0寄存器来清除STPDET位。

**图 17-26. 从机接收模式(10 位地址模式)**



### 主机发送模式下的软件流程

如*图17-27. 主机发送模式(10位地址模式)*所示，在主机模式下发送数据到I2C总线时，软件应该遵循这些步骤来运行I2C模块：

1. 首先，软件应该使能I2C外设时钟，以及配置I2C_CTL1中时钟相关寄存器来确保正确的I2C时序。使能和配置以后，I2C运行在默认的从机模式状态，等待START信号，随后等待I2C总线寻址。
2. 软件将START位置1，在I2C总线上产生一个START信号。
3. 发送一个START信号后，I2C硬件将I2C_STAT0的SBSEND位置1然后进入主机模式。现在软件应该读I2C_STAT0寄存器然后写一个7位地址位或10位地址的地址头到I2C_DATA寄存器来清除SBSEND位。当SBSEND位被清0时，I2C就开始发送地址或者地址头到I2C总线。如果发送的地址是10位地址的地址头，硬件在发送地址头的时候会将ADD10SEND位置1，软件应该通过读I2C_STAT0寄存器然后写10位低地址到I2C_DATA来清除ADD10SEND位。
4. 7位或10位的地址位发送出去之后，I2C硬件将ADDSEND位置1，软件通过读I2C_STAT0寄存器然后读I2C_STAT1寄存器清除ADDSEND位。
5. I2C进入数据发送状态，因为移位寄存器和数据寄存器I2C_DATA都是空的，所以硬件将TBE位置1。此时软件可以写第一个字节数据到I2C_DATA寄存器，但是TBE位此时不会被清零，因为写入I2C_DATA寄存器的字节会被立即移入内部移位寄存器。当移位寄存器非空时，I2C就开始发送数据到总线。
6. 在第一个字节的发送过程中，软件可以写第二个字节到I2C_DATA，此时TBE会被清零，因为I2C_DATA寄存器和移位寄存器都不为空。
7. 任意时刻TBE被置1，软件都可以向I2C_DATA寄存器写入一个字节，只要还有数据待发送。
8. 在倒数第二个字节发送过程中，软件写入最后一个字节数据到I2C_DATA来清除TBE标志位，此后就不用关心TBE位的状态。TBE位会在倒数第二个字节发送完成后被置起，直到发送STOP信号时被清零。
9. 最后一个字节发送结束后，I2C主机将BTC位置起，因为移位寄存器和I2C_DATA寄存器此时都为空。软件此时应该配置STOP来发送一个STOP信号，此后TBE和BTC状态位都将被清0。

**图 17-27. 主机发送模式(10 位地址模式)**




## 主机接收模式下的软件流程

在主机接收模式下，主机需要为最后一个字节接收产生NACK，然后发送STOP信号。因此，需要特别注意以确保最后接收到数据的正确性。下面提供了两种针对主机接收模式的软件编程方案：方案A和B。方案A需要保证软件能对I2C事件进行快速响应，方案B则不需要。

### 方案 A

1. 首先，软件应该使能I2C外设时钟，以及配置I2C_CTL1中时钟相关寄存器来确保正确的I2C时序。使能和配置以后，I2C运行在默认的从机模式状态，等待START信号，随后等待I2C总线寻址。
2. 软件将START位置1，从而在I2C总线上产生一个START信号。
3. 发送一个START信号后，I2C硬件将I2C_STAT0寄存器的SBSEND位置1然后进入主机模式。现在软件应该读I2C_STAT0寄存器然后写一个7位地址位或10位地址的地址头到I2C_DATA寄存器来清除SBSEND位。当SBSEND位被清0时，I2C就开始发送地址或者地址头到I2C总线。如果发送的地址是10位地址的地址头，硬件在发送地址头的时候会先将ADD10SEND位置1，软件应该通过读I2C_STAT0寄存器然后写10位低地址到I2C_DATA

来清除ADD10SEND位。

4. 7位或10位的地址位发送出去之后，I2C硬件将ADDSEND位置1，软件应该通过读I2C_STAT0寄存器然后读I2C_STAT1寄存器清除ADDSEND位。如果地址是10位格式，软件应该再次将START位置1来重新产生一个START。在START产生后，SBSEND位会被置1。软件应该通过先读I2C_STAT0然后写地址头到I2C_DATA来清除SBSEND位，然后地址头被发到I2C总线，ADDSEND再次被置1。软件应该再次通过先读I2C_STAT0然后读I2C_STAT1来清除ADDSEND位。
5. 当接收到第一个字节时，硬件会将RBNE位置1。此时软件可以从I2C_DATA寄存器读取第一个字节，之后RBNE位被清0。
6. 此后任何时候RBNE被置1，软件就可以从I2C_DATA寄存器读取一个字节。
7. 接收完倒数第二个字节(N-1)数据之后，软件应该立即将ACKEN位清0，并将STOP位置1，这一过程需要在最后一个字节接收完毕之前完成，以确保NACK发送给最后一个字节。
8. 最后一个字节接收完毕后，RBNE位被置1，软件可以读取最后一个字节。由于ACKEN已经在前一步骤中被清0，I2C不再为最后一个字节发送ACK，并在最后一个字节发送完毕后产生一个STOP信号。

以上步骤要求字节数目N>1，如果N=1，步骤7应该在步骤4之后就执行，且需要在字节接收完成之前完成。

**图 17-28. 主机接收使用方案 A 模式(10 位地址模式)**




**方案 B**

1. 首先，软件应该使能I2C外设时钟，配置I2C_CTL1中时钟相关寄存器来确保正确的I2C时序。初始化完成之后，I2C运行在默认的从机模式状态，等待START信号和地址。
2. 软件将START位置1，从而在I2C总线上产生一个START信号。
3. 发送一个START信号后，I2C硬件将I2C_STAT0寄存器的SBSEND位置1然后进入主机模式。现在软件应该读I2C_STAT0寄存器然后写一个7位地址位或10位地址的地址头到I2C_DATA寄存器来清除SBSEND位。当SBSEND位被清0时，I2C就开始发送地址或者地址头到I2C总线。如果发送的地址是10位地址的地址头，硬件在发送地址头之后会将ADD10SEND位置1，软件应该通过读I2C_STAT0寄存器然后写10位低地址到I2C_DATA来清除ADD10SEND位。

4. 7位或10位的地址位发送出去之后，I2C硬件将ADDSEND位置1，软件应该通过读I2C_STAT0寄存器然后读I2C_STAT1寄存器清除ADDSEND位。如果地址是10位格式，软件应该接着将START位再次置1来产生一个开始信号，START被发送出去以后SBSEND位被再次置1。软件应该通过先读I2C_STAT0然后写地址头到I2C_DATA来清除SBSEND位，然后地址头被发到I2C总线，ADDSEND再次被置1。软件应该再次通过先读I2C_STAT0然后读I2C_STAT1来清除ADDSEND位。

5. 当第一个字节被接收时，RBNE位会被硬件置1。此时软件可从I2C_DATA寄存器读取出第一个字节，同时RBNE位被清0。

6. 此后任何时刻，只要RBNE位被置1，软件就可以从I2C_DATA寄存器读取一个字节的数据，直到主机接收了N-3个字节。

如*图 17-29. 主机接收使用方案 B 模式(10 位地址模式)*所示，第 N-2 个字节还没被软件读出，之后第 N-1 个字节被接收，此时 BTC 和 RBNE 都被置位，总线就会被主机锁死以阻止最后一个字节的接收。然后软件应该清除 ACKEN 位。

7. 软件从I2C_DATA读出倒数第三个(N-2)字节数据，同时也将BTC位清0。此后第N-1个字节从移位寄存器被移到I2C_DATA，总线得到释放然后开始接收最后一个字节，由于ACKEN已经被清除，因此主机不会给最后一个字节数据发送ACK响应。

8. 最后一个字节接收完毕后，硬件再次把BTC位和RBNE置1，并拉低SCL，软件将STOP位置1，主机发出一个STOP信号。

9. 软件读取第N-1个字节，清除BTC。此后最后一个字节从移位寄存器被移动到I2C_DATA。

10. 软件读取最后一个字节，清除RBNE。

以上步骤需要字节数字N>2，N=1和N=2的情况近似：

**N=1**

在第 4 步，软件应该在清除 ADDSEND 位之前将 ACKEN 位清 0，在清除 ADDSEND 位之后将 STOP 位置 1。当 N=1 时步骤 5 是最后一步。

**N=2**

在第2步，软件应该在START置1之前将POAP置1。在第4步，软件应该在清除ADDSEND位之前将ACKEN位清0。在第5步，软件应该一直等到BTC位被置1然后将STOP位置1且读取I2C_DATA两次。

**图 17-29. 主机接收使用方案 B 模式(10 位地址模式)**




## 18.3.8. SCL 线控制

SCL 线拉低功能是为了避免在接收时发生上溢错误以及在发送时发生下溢错误。如在软件编程模型中所示，在发送模式，当 TBE 和 BTC 被置位，发送器保持 SCL 线为低电平直到下一个发送数据写入传输缓冲区寄存器。在接收模式，当 RBNE 和 BTC 被置位，发送器保持 SCL 线为低电平直到传输缓冲区寄存器里的数据被读出。

当工作在从模式的时候，可以通过置位 I2C_CTL0 寄存器的 SS 位禁止 SCL 线拉低功能。如果该位置位，软件要能足够快的处理 TBE，RBNE 和 BTC 状态，否则上溢或下溢的情况可能会发生。

### 18.3.9. DMA 模式下数据传输

按照前面的软件流程，每当 TBE 位或 RBNE 位被置 1 之后，软件都应该写或读一个字节，这样将导致 CPU 的负荷较重。I2C 的 DMA 功能可以在 TBE 或 RBNE 位置 1 时，自动进行一次写或读操作，从而减轻了 CPU 的负荷，具体 DMA 的配置请参看 DMA 相关章节。

DMA 请求通过 I2C_CTL1 寄存器的 DMAON 位使能。该位应该在清除 ADDSEND 状态位之后被置位。如果一个从机的 SCL 线延长功能被禁止，DMAON 位应该在 ADDSEND 事件前被置位。

参考 DMA 控制器的关于 DMA 的配置方法说明。DMA 必须在 I2C 传输开始之前配置和使能。当指定个数的字节已经传输完成，DMA 会发送一个传输结束(EOT)信号给 I2C 接口，并产生一个 DMA 传输完成中断。

当主机接收两个或两个以上字节时，需将 I2C_CTL1 寄存器的 DMALST 位置位。在接收到最后一个字节之后，I2C 主机发送 NACK。在 DMA 传输完成中断 ISR 中，通过置位 STOP 位，产生一个停止信号。

当主机仅接收一个字节时，清除ADDSEND状态前ACKEN位必须被清除。在清除ADDSEND状态后或在DMA传输完成中断ISR中，通过置位STOP位，产生一个停止信号。

### 18.3.10. 报文错误校验

I2C模块中有一个PEC(包错误检查)模块，它使用CRC-8计算器来执行I2C数据的报文校验，CRC多项式为x8 + x2 + x + 1，和SMBus协议兼容。将PECEN位置1就可以使能PEC功能。PEC将会计算所有通过I2C总线发送的数据(包括地址)。软件可以通过配置PECTRANS来控制I2C在最后一个字节发送完毕后发送PEC值，或者在接收完成后检查接收到的PEC值是否正确。在DMA模式下，如果PECEN位和PECTRANS位被置1，I2C将自动发送或者检查PEC值。

### 18.3.11. SMBus 支持

系统管理总线(System Management Bus，简写为SMBus或SMB)是一种结构简单的单端双线制总线，可实现轻量级的通信需求。一般来说，SMBus最常见于计算机主板，主要用于电源传输ON / OFF指令的通信。SMBus是I2C的一种衍生总线形式，主要用于计算机主板上的低带宽设备间通信，尤其是与电源相关的芯片，例如笔记本电脑的可充电电池子系统(参见Smart Battery Data)。

#### SMBus 协议

SMBus上每个报文交互都遵从SMBus协议中预定义的格式。SMBus是I2C规范中数据传输格式

的子集。只要I2C设备可通过SMBus协议之一进行访问，便视为兼容SMBus规范。不符合这些协议的I2C设备，将无法被SMBus和ACPI规范所定义的标准方法访问。

### 地址解析协议

SMBus是基于I2C硬件实现的，它使用了I2C的硬件寻址方式，但在I2C的基础上增加了二级软件处理，建立自己独特的系统。比较特别的是SMBus规范包含一个地址解析协议，可用于实现动态地址分配。动态识别硬件和软件使得总线设备能够支持热插拔，无需重启系统便能即插即用。总线中的设备将被自动识别并分配唯一地址。这个优点非常有利于实现即插即用的用户接口。在此协议中，系统中的host与设备之间有一个重要的区别，即host具有分配地址的功能。

### 超时特性

SMBus有一种超时特性：假如某个通信耗时太久，便会自动复位设备。这就解释了为什么最小时钟周期为10kHz--为了防止长时间锁死总线。I2C在本质上可以视为一个"直流"总线，也就是说当主机正在访问从机的时候，假如从机正在执行一些子程序无法及时响应，从机可以拉住主机的时钟。这样便可以提醒主机：从机正忙，但并不想放弃当前的通信。从机的当前任务结束之后，将可以继续I2C通信。I2C总线协议中并没有限制这个延时的上限，但在SMBus系统中，这个时间被限定为25ms。按照SMBus协议的假定，如果某个通信耗时太久，就意味着总线出了问题，此时所有设备都应当复位以消除这种问题。这样就并不允许从设备将时钟拉低太长时间。

### 报文错误校验

SMBus 2.0以及1.1都采用了报文错误校验(Packet Error Checking，缩写为PEC)。在这种模式中，每次通信最后都将传输PEC字节。该字节是按照CRC-8校验和的方式计算的，计算范围包括整个报文，包括地址以及读/写位。所采用的多项式为x8+x2+x+1(CRC-8-ATM HEC算法，初始化为0)。

### SMBus 警报

SMBus还有一个额外的中断信号，称为SMBALERT#。从机上发生事件后，可通过这个信号通知主机来访问从机。SMBus中还定义了较少见的"主机提醒协议"，基于I2C多主模式实现类似的提醒功能，但是可以传递更多数据。

### SMBus 通讯流程

SMBus的通讯流程和标准I2C的流程相似。为了使用SMBus模式，在程序中需要配置几个SMBus特定的寄存器，响应一些SMBus特定标志位，实现那些在SMBus手册中介绍的上层协议。

1. 在通信之前，需要将I2C_CTL0中SMBEN位置1，并且根据需求，配置SMBSEL和ARPEN的值。

2. 为了支持ARP协议(ARPEN=1)，在SMBus主机模式下(SMBSEL=1)，软件需要响应标志位HSTSMB(在SMBus从机模式下，响应DEFSMB标志位)，并实现ARP协议中的功能。

3. 为了支持SMBus警告模式，软件应该响应SMBALT标志位，并实现相应的功能。

### 18.3.12. SAM_V 支持

为了支持 SAM_V 标准，I2C 模块增加两个附加的引脚：txframe 和 rxframe。Txframe 是一个输出引脚，在主机模式下，当该引脚输出电平为高电平时，表示 I2C 是忙的。Rxframe 是一个输入引脚，应该与 SMBALERT 信号一起多路复用。

SAM_V 模式通过置位 I2C_SAMCS 寄存器的 SAMEN 位使能。txframe 和 rxframe 引脚的状态可以通过 I2C_SAMCS 寄存器的 RFR，RFF，TFR，TFF，RXF 和 TXF 标志反映。如果对应的中断使能位置位，将产生 I2C 中断。

### 18.3.13. 状态、错误和中断

I2C有一些状态、错误标志位，通过设置一些寄存器位，便可以从这些标志触发中断(详情参见*I2C寄存器*)。

**表 17-5. 事件状态标志位**

| 事件标志位名称 | 说明 |
|---|---|
| SBSEND | 主机发送 START 信号 |
| ADDSEND | 地址发送和接收 |
| ADD10SEND | 10 位地址模式中地址头发送 |
| STPDET | 监测到 STOP 信号 |
| BTC | 字节发送结束 |
| TBE | 发送时 I2C_DATA 为空 |
| RBNE | 接收时 I2C_DATA 非空 |
| RFR | SAM_V 模式时检测到 rxframe 上升沿 |
| RFF | SAM_V 模式时检测到 rxframe 下降沿 |
| TFR | SAM_V 模式时检测到 txframe 上升沿 |
| TFF | SAM_V 模式时检测到 txframe 下降沿 |

**表 17-6. 错误标志位**

| 错误名称 | 说明 |
|---|---|
| BERR | 总线错误 |
| LOSTARB | 仲裁丢失 |
| OUERR | 当禁用 SCL 拉低后，发生了上溢或下溢 |
| AERR | 没有接收到应答 |
| PECERR | CRC 值不相同 |
| SMBTO | SMBus 模式下总线超时 |
| SMBALT | SMBus 警报 |

## 18.4. I2C 寄存器

I2C0基地址：0x4000 5400
I2C1基地址：0x4000 5800

### 18.4.1. 控制寄存器 0(I2C_CTL0)

地址偏移：0x00
复位值：0x0000 0000

该寄存器可以按半字(16位)或字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SRESET | 保留 | SALT | PECTRANS | POAP | ACKEN | STOP | START | SS | GCEN | PECEN | ARPEN | SMBSEL | 保留 | SMBEN | I2CEN |
| 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 | SRESET | 软件复位 I2C，软件应该在 I2C 总线被释放时复位 I2C。<br>0：I2C 未复位<br>1：I2C 复位 |
| 14 | 保留 | 必须保持复位值。 |
| 13 | SALT | SMBus 警报<br>通过 SMBA 引脚发出警报。<br>软件置 1 和清 0，硬件清 0。<br>0：不通过 SMBA 发布警告<br>1：通过 SMBA 引脚发送警告 |
| 12 | PECTRANS | PEC 传输<br>软件置 1 和清 0，硬件在以下条件下清除此位：PEC 传输完成，或监测到 START / STOP 信号，或 I2CEN=0。<br>0：不传输 PEC 值<br>1：传输的 PEC 值 |
| 11 | POAP | ACK/PEC 的位置含义<br>软件置 1 和清 0，当 I2CEN=0 时，硬件清 0。<br>0：ACKEN 位决定对当前正在接收的字节是否发送 ACK，PECTRANS 位表明正在接收的字节是否为 PEC。<br>1：ACKEN 位决定是否对下一个字节发送 ACK，PECTRANS 位表明下一个即将被接收的字节是 PEC。 |

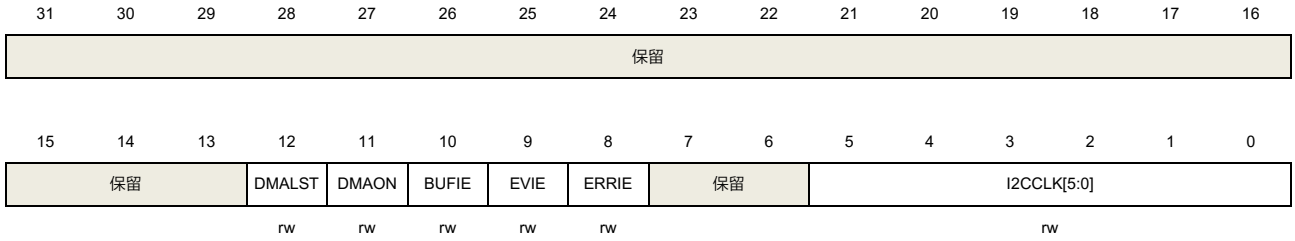
| | | |
|---|---|---|
| 10 | ACKEN | 是否发送 ACK<br>软件置 1 和清 0，当 I2CEN=0 时硬件清 0。<br>0：不发送 ACK<br>1：发送 ACK |
| 9 | STOP | I2C 总线上产生一个 STOP 信号<br>软件置 1 和清 0，SMBus 超时时，硬件置 1，监测到 STOP 信号时，硬件清 0。<br>0：不发送 STOP<br>1：发送 STOP |
| 8 | START | I2C 总线上产生一个 START 信号<br>软件置 1 和清 0，当监测到 START 信号或 I2CEN=0 时由硬件清 0。<br>0：不发送 START<br>1：发送 START |
| 7 | SS | 在从机模式下数据未就绪是否将 SCL 拉低<br>软件置 1 和清 0。<br>0：拉低 SCL<br>1：不拉低 SCL |
| 6 | GCEN | 是否响应对地址(0x00)的广播呼叫<br>0：从机不响应广播呼叫<br>1：从机将响应广播呼叫 |
| 5 | PECEN | PEC 计算使能<br>0：PEC 计算禁能<br>1：PEC 计算使能 |
| 4 | ARPEN | SMBus 下 ARP 协议使能<br>0：ARP 禁能<br>1：ARP 使能 |
| 3 | SMBSEL | SMBus 类型选择<br>0：从机<br>1：主机 |
| 2 | 保留 | 必须保持复位值 |
| 1 | SMBEN | SMBus/I2C 模式开关<br>0：I2C 模式<br>1：SMBus 模式 |
| 0 | I2CEN | I2C 外设使能<br>0：禁能 I2C<br>1：使能 I2C |

### 18.4.2. 控制寄存器 1(I2C_CTL1)

地址偏移：0x04

复位值：0x0000 0000

该寄存器可以按半字(16位)或字(32位)访问。

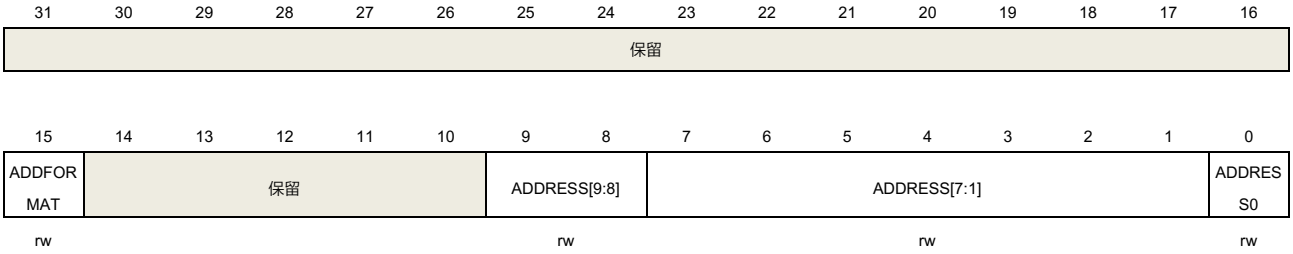



| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:13 | 保留 | 必须保持复位值。 |
| 12 | DMALST | DMA 最后传输配置<br>0：下一个 DMA EOT 不是最后传输<br>1：下一个 DMA EOT 是最后传输 |
| 11 | DMAON | DMA 模式开关<br>0：DMA 模式关<br>1：DMA 模式开 |
| 10 | BUFIE | 缓冲区中断使能<br>0：禁用缓冲区中断，如果 EVIE=1，当 TBE = 1 或 RBNE = 1 时不产生中断。<br>1：使能缓冲区中断，如果 EVIE=1，当 TBE = 1 或 RBNE = 1 时产生中断。 |
| 9 | EVIE | 事件中断使能<br>0：禁用事件中断<br>1: 使能事件中断，意味着当 SBSEND、ADDSEND、ADD10SEND、STPDET 或 BTC 标志位有效或当 BUFIE=1 时 TBE=1 或 RBNE=1 时产生中断。 |
| 8 | ERRIE | 错误中断使能<br>0：禁用错误中断<br>1: 使能错误中断，意味着当 BERR、LOSTARB、AERR、OUERR、PECERR、SMBTO 或 SMBALT 标志位生效时产生中断。 |
| 7:6 | 保留 | 必须保持复位值。 |
| 5:0 | I2CCLK[5:0] | I2C 外设时钟频率<br>I2CCLK[5:0]应该是输入 APB1 时钟频率，最低 2MHz。<br>000000 - 000001：无时钟<br>000010 - 111100：2MHz~60MHz<br>111101 - 111111：由于 APB1 时钟限制，无时钟<br>**注意：**<br>在标准模式下，APB1 时钟频率需大于或者等于 2MHz。在快速模式下，APB1 时钟频率需大于或者等于 8MHz。在快速 +模式下，APB1 时钟频率需大于或者等于 24MHz。 |

### 18.4.3. 从机地址寄存器 0(I2C_SADDR0)

地址偏移：0x08
复位值：0x0000 0000

该寄存器可以按半字(16位)或字(32位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15 | ADDFORMAT | I2C 从机地址格式<br>0：7 位地址<br>1：10 位地址 |
| 14:10 | 保留 | 必须保持复位值。 |
| 9:8 | ADDRESS[9:8] | 10 位地址的最高两位 |
| 7:1 | ADDRESS[7:1] | 7 位地址或者 10 位地址的第 7-1 位 |
| 0 | ADDRESS0 | 10 位地址的第 0 位 |

### 18.4.4. 从机地址寄存器 1(I2C_SADDR1)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器可以按半字(16位)或字(32位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 必须保持复位值。 |
| 7:1 | ADDRESS2[7:1] | 从机在双重地址模式下第二个 I2C 地址 |
| 0 | DUADEN | 双重地址模式使能 |

0：禁用双重地址模式
1：使能双重地址模式

### 18.4.5. 传输缓冲区寄存器(I2C_DATA)

地址偏移：0x10
复位值：0x0000 0000

该寄存器可以按半字(16位)或字(32位)访问。

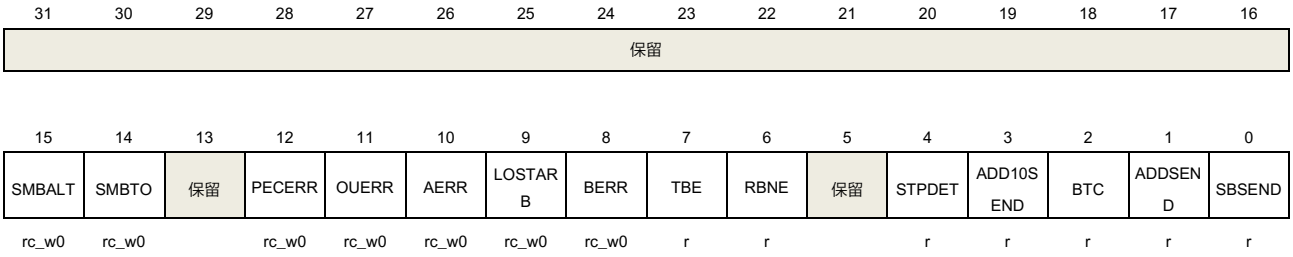



| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 必须保持复位值。 |
| 7:0 | TRB[7:0] | 数据发送接收缓冲区 |

### 18.4.6. 传输状态寄存器 0(I2C_STAT0)

地址偏移：0x14
复位值：0x0000 0000

该寄存器可以按半字(16位)或字(32位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15 | SMBALT | SMBus 警报状态<br>硬件置 1，软件写 0 清 0。<br>0：SMBA 引脚未被拉低(从机模式)或未监测到警报(主机模式)<br>1：SMBA 引脚被拉低且接收到警报地址(从机模式)或监测到警报(主机模式) |
| 14 | SMBTO | SMBus 模式下超时信号<br>硬件置 1，软件写 0 清 0。<br>0：无超时错误 |

| | | |
|---|---|---|
| | | 1：超时事件发生(SCL 被拉低达 25ms) |
| 13 | 保留 | 必须保持复位值。 |
| 12 | PECERR | 接收数据时 PEC 错误<br>硬件置 1，软件写 0 清 0。<br>0：接收到 PEC 且校验正确<br>1：接收到 PEC 但检验错误，此时不管 ACKEN 位的值，I2C 将发送 NACK |
| 11 | OUERR | 当禁用 SCL 拉低功能后，在从机模式下发生了上溢或下溢事件。在从机接收模式下，假如 I2C_DATA 中的最后一字节并未被读出，并且后续字节又接收完成，就会发生上溢错误。在从机发送模式下，假如当前字节已经发送完成，而 I2C_DATA 仍然为空，就会发生下溢错误。<br>硬件置 1，软件写 0 清 0。<br>0：无上溢或下溢错误发生<br>1：发生上溢或下溢错误 |
| 10 | AERR | 应答错误<br>硬件置 1，软件写 0 清 0。<br>0：未发生应答错误<br>1：发生了应答错误 |
| 9 | LOSTARB | 主机模式下仲裁丢失<br>硬件置 1，软件写 0 清 0。<br>0：无仲裁丢失<br>1：发生仲裁丢失，I2C 模块返回从机模式。 |
| 8 | BERR | 总线错误，表示 I2C 总线上发生了预料之外的 START 信号或 STOP 信号。<br>硬件置 1，软件写 0 清 0。<br>0：无总线错误<br>1：发生了总线错误 |
| 7 | TBE | 发送期间 I2C_DATA 为空<br>硬件从 I2C_DATA 寄存器移动一个字节到移位寄存器之后将此位置 1，软件写一个字节到 I2C_DATA 寄存器清除该位。如果移位寄存器和 I2C_DATA 寄存器都是空的，写 I2C_DATA 寄存器将不会清除 TBE 位(详见主机/从机发送模式下的软件操作流程)<br>0：I2C_DATA 非空<br>1：I2C_DATA 空，软件可以写 |
| 6 | RBNE | 接收期间 I2C_DATA 非空<br>硬件从移位寄存器移动一个字节到 I2C_DATA 寄存器之后将此位置 1，读 I2C_DATA 可以清除此位。如果 BTC 和 RBNE 都被置 1，读 I2C_DATA 将不会清除 RBNE，因为移位寄存器的字节将被立即移到 I2C_DATA。<br>0：I2C_DATA 为空<br>1：I2C_DATA 非空，软件可以读 |
| 5 | 保留 | 必须保持复位值。 |
| 4 | STPDET | 从机模式下监测到 STOP 信号 |

此位被硬件置 1，先读 I2C_STAT0 然后写 I2C_CTL0 可以清除此位。
0：从机模式下未监测到 STOP 信号
1：从机模式下监测到 STOP 信号

| | | |
|---|---|---|
| 3 | ADD10SEND | 主机模式下 10 位地址的地址头被发送<br>该位由硬件置 1，软件读 I2C_STAT0 和写 I2C_DATA 清除此位。<br>0：主机模式下未发送 10 位地址的地址头<br>1：主机模式下发送 10 位地址的地址头 |
| 2 | BTC | 字节发送结束<br>接收模式下，如果一个字节已经被移位寄存器接收但是此时 I2C_DATA 寄存器仍然是满的；或者发送模式下，一个字节已经被移位寄存器发送但是 I2C_DATA 寄存器仍然是空的，如果使能了 SCL 拉低功能，位硬件就会置起 BTC 标志。<br>此位由硬件置 1。<br>可由以下三种方式清除：<br>1、软件清除：读 I2C_STAT0，然后读或写 I2C_DATA 寄存器清除此位<br>2、硬件清除：发送一个 STOP 或 START 信号<br>3、寄存器 I2C_CTL0 中 I2CEN=0<br>0：未发生 BTC<br>1：发生了 BTC |
| 1 | ADDSEND | 主机模式下：成功发送了地址并收到 ACK<br>从机模式下：接收到的地址与自身的地址匹配<br>此位由硬件置 1，软件读 I2C_STAT0 寄存器和读 I2C_STAT1 清 0。<br>0：从机模式下，未收到地址或者收到的地址不匹配；主机模式下，无地址被发送或地址已发送但未收到从机回复的 ACK<br>1：从机模式下，接收到的地址与自身的地址匹配；主机模式下，地址已发送并收到 ACK |
| 0 | SBSEND | 主机模式下发送 START 信号<br>此位由硬件置 1，软件读 I2C_STAT0 和写 I2C_DATA 清 0。<br>0：未发送 START 信号<br>1：START 信号被发送 |

### 18.4.7. 传输状态寄存器 1(I2C_STAT1)

地址偏移：0x18
复位值：0x0000 0000

该寄存器可以按半字(16位)或字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PECV[7:0] | | | | | | | | DUMODF | HSTSMB | DEFSMB | RXGC | 保留 | TR | I2CBSY | MASTER |
| r | | | | | | | | r | r | r | r | | r | r | r |

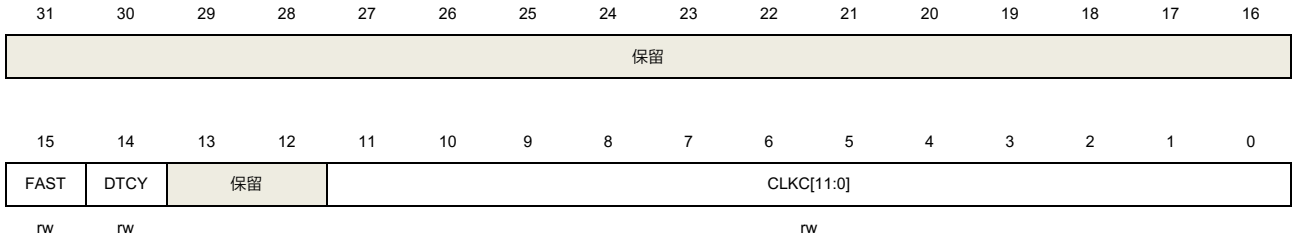
| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15:8 | PECV[7:0] | 当 PEC 使能后硬件计算出的 PEC 值。 |
| 7 | DUMODF | 从机模式下双标志位表明哪个地址和双地址模式匹配<br>STOP 或 START 信号产生后或 I2CEN=0 时此位由硬件清 0。<br>0：地址和 I2C_SADDR0 匹配<br>1：地址和 I2C_SADDR1 匹配 |
| 6 | HSTSMB | 从机模式下监测到 SMBus 主机地址头<br>STOP 或 START 信号产生后或 I2CEN=0 时此位由硬件清 0。<br>0：未监测到 SMBus 主机地址头<br>1：监测到 SMBus 主机地址头 |
| 5 | DEFSMB | SMBus 设备缺省地址<br>STOP 或 START 信号产生后或 I2CEN=0 时此位由硬件清 0。<br>0：SMBus 设备没有接收到缺省地址<br>1：SMBus 设备接收到缺省地址 |
| 4 | RXGC | 是否接收到广播地址(0x00)<br>STOP 或 START 信号产生后或 I2CEN=0 时此位由硬件清 0。<br>0：未接收到广播呼叫地址<br>1：接收到广播呼叫地址 |
| 3 | 保留 | 必须保持复位值。 |
| 2 | TR | 发送端或接收端<br>该位表明 I2C 作为发送端还是接收端。STOP 或 START 信号产生后或 I2CEN 或 LOSTARB=1 时此位由硬件清 0。<br>0：接收端<br>1：发送端 |
| 1 | I2CBSY | 忙标志<br>STOP 信号后硬件清 0。<br>0：无 I2C 通讯<br>1：I2C 正在通讯 |
| 0 | MASTER | 主机模式<br>表明 I2C 时钟在主机模式还是从机模式的标志位。<br>该位在 START 信号产生后由硬件置 1。<br>该位在 STOP 信号产生后或 I2CEN=0 或 LOSTARB=1 时此位由硬件清 0。<br>0：从机模式<br>1：主机模式 |

### 18.4.8. 时钟配置寄存器(I2C_CKCFG)

地址偏移：0x1C

复位值：0x0000 0000

该寄存器可以按半字(16位)或字(32位)访问。



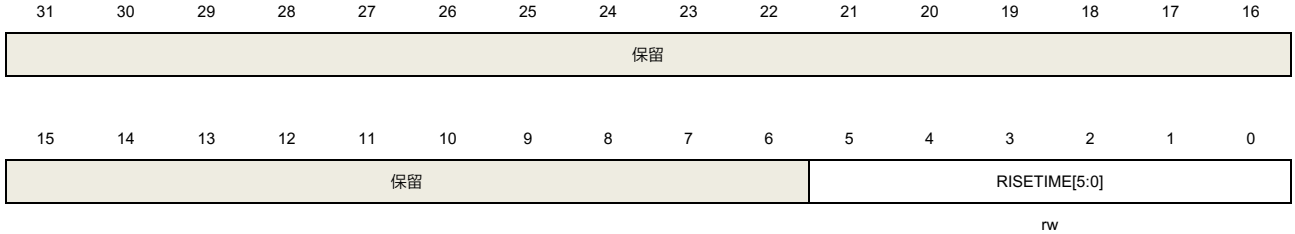

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15 | FAST | 主机模式下 I2C 速度选择<br>0：标准速度<br>1：快速 |
| 14 | DTCY | 快速模式或快速+ 模式下占空比<br>0：$T_{low}/T_{high}=2$<br>1：$T_{low}/T_{high}=16/9$ |
| 13:12 | 保留 | 必须保持复位值。 |
| 11:0 | CLKC[11:0] | 主机模式下 I2C 时钟控制<br>标准速度模式下：$T_{high}=T_{low}=CLKC*T_{PCLK1}$<br>如果 DTCY=0，快速模式或快速+ 模式下：<br>$T_{high}=CLKC*T_{PCLK1}$，$T_{low}=2*CLKC*T_{PCLK1}$<br>如果 DTCY=1，快速模式或快速+ 模式下：<br>$T_{high}=9*CLKC*T_{PCLK1}$，$T_{low}=16*CLKC*T_{PCLK1}$<br>**注意：**如果 DTCY=0，当 PCLK1 为 3 的整数倍时，波特率会比较准确。如果 DTCY=1，当 PCLK1 为 25 的整数倍时，波特率会比较准确。 |

## 18.4.9. 上升时间寄存器(I2C_RT)

地址偏移：0x20

复位值：0x0000 0002

该寄存器可以按半字(16位)或字(32位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|

| | | |
|---|---|---|
| 31:6 | 保留 | 必须保持复位值。 |
| 5:0 | RISETIME[5:0] | 主机模式下最大上升时间<br>RISETIME 值应该为 SCL 最大上升时间加 1 |

### 18.4.10. SAM 控制状态寄存器(I2C_SAMCS)

地址偏移：0x80
复位值：0x0000 0000

该寄存器可以按半字(16位)或字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RFR | RFF | TFR | TFF | 保留 | | RXF | TXF | RFRIE | RFFIE | TFRIE | TFFIE | 保留 | | STOEN | SAMEN |
| rc_w0 | rc_w0 | rc_w0 | rc_w0 | | | r | r | rw | rw | rw | rw | | | rw | rw |

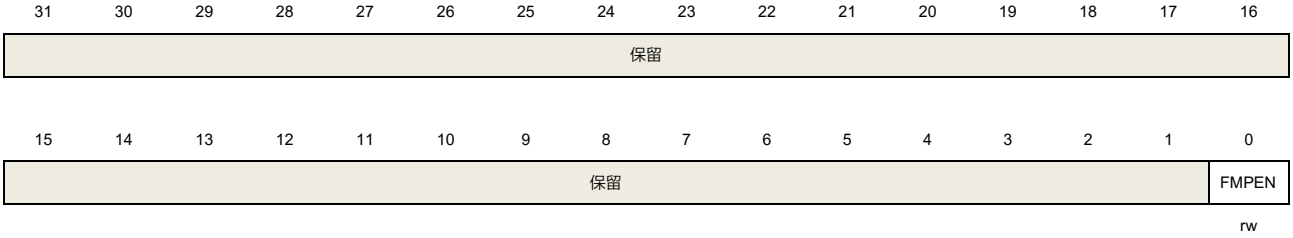
| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15 | RFR | 接收帧上升沿标志，由软件写 0 清 0 |
| 14 | RFF | 接收帧下降沿标志，由软件写 0 清 0 |
| 13 | TFR | 发送帧上升沿标志，由软件写 0 清 0 |
| 12 | TFF | 发送帧下降沿标志，由软件写 0 清 0 |
| 11:10 | 保留 | 必须保持复位值。 |
| 9 | RXF | 接收帧信号电平 |
| 8 | TXF | 发送帧信号电平 |
| 7 | RFRIE | 接收帧上升沿中断使能<br>0：接收帧上升沿中断失能<br>1：接收帧上升沿中断使能 |
| 6 | RFFIE | 接收帧下降沿中断使能<br>0：接收帧下降沿中断失能<br>1：接收帧下降沿中断使能 |
| 5 | TFRIE | 发送帧上升沿中断使能<br>0：发送帧上升沿中断失能<br>1：发送帧上升沿中断使能 |
| 4 | TFFIE | 发送帧下降沿中断使能<br>0：发送帧下降沿中断失能<br>1：发送帧下降沿中断使能 |

| | | |
|---|---|---|
| 3:2 | 保留 | 必须保持复位值。 |
| 1 | STOEN | SAM_V 接口超时检测使能<br>0：SAM_V 接口超时检测失能<br>1：SAM_V 接口超时检测使能 |
| 0 | SAMEN | SAM_V 接口使能<br>0：SAM_V 接口失能<br>1：SAM_V 接口使能 |

### 18.4.11. 快速+模式配置寄存器(I2C_FMPCFG)

地址偏移：0x90
复位值：0x0000 0000

该寄存器可以按半字(16位)或字(32位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:1 | 保留 | 必须保持复位值。 |
| 0 | FMPEN | 快速+ 模式使能<br>当该位被置 1 时，I2C 设备支持高达 1MHz 的通信速率。<br>0：快速+ 模式失能<br>1：快速+ 模式使能 |

# 19. 串行外设接口/片上音频接口(SPI/I2S)

## 19.1. 简介

SPI/I2S模块可以通过SPI协议或I2S音频协议与外部设备进行通信。

串行外设接口(Serial Peripheral Interface，缩写为SPI)提供了基于SPI协议的数据发送和接收功能，可以工作于主机或从机模式。SPI接口支持具有硬件CRC计算和校验的全双工和单工模式。SPI0还支持SPI四线主机模式。

片上音频接口(Inter-IC Sound，缩写为I2S)支持四种音频标准，分别是I2S飞利浦标准，MSB对齐标准，LSB对齐标准和PCM标准。它可以在四种模式下运行，包括主机发送模式，主机接收模式，从机发送模式和从机接收模式。

## 19.2. 主要特性

### 19.2.1. SPI 主要特性

- 具有全双工、半双工和单工模式的主从操作；
- 16位宽度，独立的发送和接收缓冲区；
- 8位或16位数据帧格式；
- 低位在前或高位在前的数据位顺序；
- 软件和硬件NSS管理；
- 硬件CRC计算、发送和校验；
- 发送和接收支持DMA模式；
- 支持SPI TI模式；
- 支持SPI NSS脉冲模式；
- 支持SPI四线功能的主机模式(只有SPI0)。

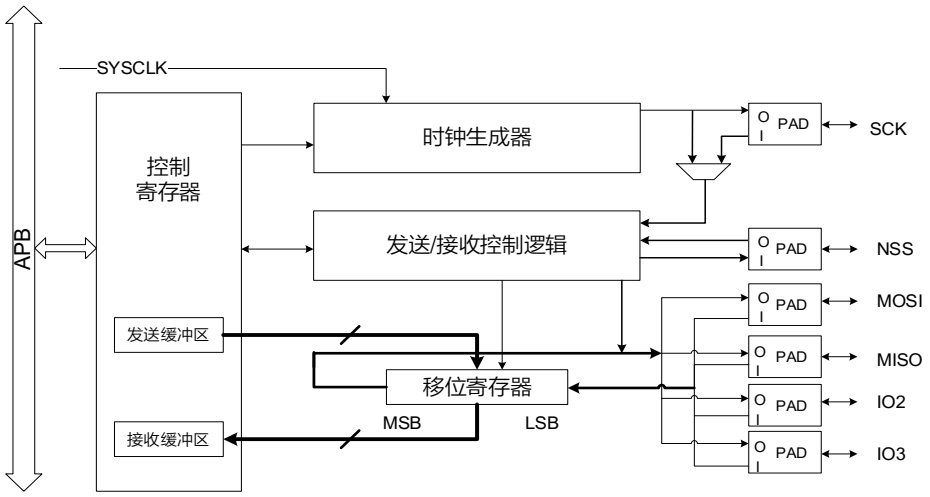
### 19.2.2. I2S 主要特性

- 具有发送和接收功能的主从操作；
- 支持四种I2S音频标准：飞利浦标准，MSB对齐标准，LSB对齐标准和PCM标准；
- 数据长度可以为16位，24位和32位；
- 通道长度为16位或32位；
- 16位缓冲区用于发送和接收；
- 通过I2S时钟分频器，可以得到8 kHz到192 kHz的音频采样频率；
- 可编程空闲状态时钟极性；
- 可以输出主时钟(MCK)；
- 发送和接收支持DMA功能。

# 19.3. SPI 功能说明

## 19.3.1. SPI 结构框图

**图 19-1. SPI 结构框图**




## 19.3.2. SPI 信号线描述

### 常规配置(非 SPI 四线模式)

**表 19-1. SPI 信号描述**

| 引脚名称 | 方向 | 描述 |
|---|---|---|
| SCK | I/O | 主机：SPI 时钟输出<br>从机：SPI 时钟输入 |
| MISO | I/O | 主机：数据接收线<br>从机：数据发送线<br>主机双向线模式：不使用<br>从机双向线模式：数据发送和接收线 |
| MOSI | I/O | 主机：数据发送线<br>从机：数据接收线<br>主机双向线模式：数据发送和接收线<br>从机双向线模式：不使用 |
| NSS | I/O | 软件 NSS 模式：不使用<br>主机硬件 NSS 模式：NSSDRV=1 时，为 NSS 输出，适用于单主机模式；NSSDRV=0 时，为 NSS 输入，适用于多主机模式。<br>从机硬件 NSS 模式：为 NSS 输入，作为从机的片选信号。 |

### SPI 四线配置

SPI默认配置为单线模式，当SPI_QCTL中的QMOD位置1时，配置为SPI四线模式(只适用于SPI0)。SPI四线模式只能工作在主机模式。

在SPI四线模式下，SPI通过以下6个引脚与外部设备连接：

**表 19-2. SPI 四线信号描述**

| 引脚名称 | 方向 | 描述 |
|---|---|---|
| SCK | O | SPI 时钟输出 |
| MOSI | I/O | 发送或接收数据 0 |
| MISO | I/O | 发送或接收数据 1 |
| IO2 | I/O | 发送或接收数据 2 |
| IO3 | I/O | 发送或接收数据 3 |
| NSS | O | NSS 输出 |

## 19.3.3. SPI 时序和数据帧格式

SPI_CTL0寄存器中的CKPL位和CKPH位决定了SPI时钟和数据信号的时序。CKPL位决定了空闲状态时SCK的电平，CKPH位决定了第一个或第二个时钟跳变沿为有效采样边沿。在TI模式下，这两位没有意义。

**图 19-2. 常规模式下的 SPI 时序图**

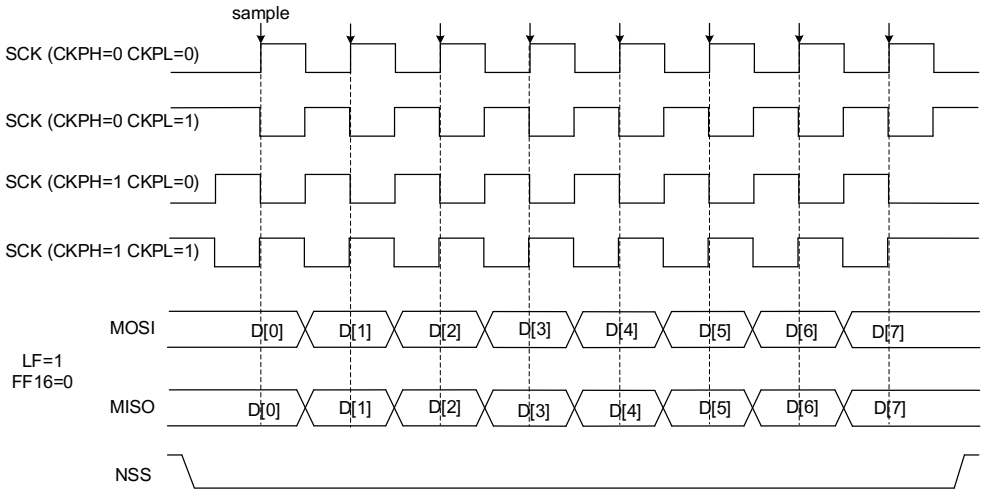
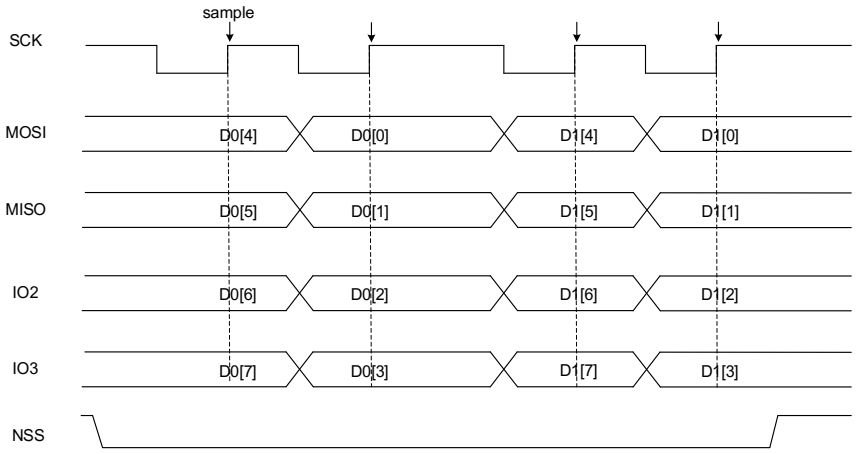


**图 19-3. SPI 四线模式下的 SPI 时序图(CKPL=1, CKPH=1, LF=0)**




在常规模式中，通过SPI_CTL0中的FF16位配置数据长度，当FF16=1时，数据长度为16位，否则为8位。在SPI四线模式下，数据帧长度固定为8位。

通过设置SPI_CTL0中的LF位可以配置数据顺序，当LF=1时，SPI先发送LSB位，当LF=0时，则先发送MSB位。在TI模式中，数据顺序固定为先发MSB位。

### 19.3.4. NSS 功能

#### 从机模式

当配置为从机模式(MSTMOD=0)时，在硬件NSS模式(SWNSSEN = 0)下，SPI从NSS引脚获取NSS电平，在软件NSS(SWNSSEN = 1)下，SPI根据SWNSS位得到NSS电平。只有当NSS为低电平时，发送或接收数据。在软件NSS模式下，不使用NSS引脚。

**表 19-3. 从机模式 NSS 功能**

| 模式 | 寄存器配置 | 描述 |
|---|---|---|
| 从机硬件 NSS 模式 | MSTMOD = 0<br>SWNSSEN = 0 | SPI 从机 NSS 电平从 NSS 引脚获取。 |
| 从机软件 NSS 模式 | MSTMOD = 0<br>SWNSSEN = 1 | SPI 从机 NSS 电平由 SWNSS 位决定。<br>SWNSS = 0：NSS 电平为低<br>SWNSS = 1：NSS 电平为高 |

#### 主机模式

在主机模式(MSTMOD=1)下，如果应用程序使用多主机连接方式，NSS可以配置为硬件输入模式(SWNSSEN=0，NSSDRV=0)或者软件模式(SWNSSEN=1)。一旦NSS引脚(在硬件NSS模式下)或SWNSS位(在软件NSS模式下)被拉低，SPI将自动进入从机模式，并且产生主机配置错误，CONFERR位置1。

如果应用程序希望使用NSS引脚控制SPI从设备，NSS应该配置为硬件输出模式(SWNSSEN=0，NSSDRV=1)。使能SPI之后，NSS变为低电平。

应用程序可以使用一个通用I/O口作为NSS引脚，以实现更加灵活的NSS应用。

**表 19-4. 主机模式 NSS 功能**

| 模式 | 寄存器配置 | 描述 |
|---|---|---|
| 主机硬件 NSS 输出模式 | MSTMOD = 1<br>SWNSSEN = 0<br>NSSDRV=1 | 适用于单主机模式，主机使用 NSS 引脚控制 SPI 从设备，此时 NSS 配置为硬件输出模式。使能 SPI 后 NSS 为低电平。 |
| 主机硬件 NSS 输入模式 | MSTMOD = 1<br>SWNSSEN = 0<br>NSSDRV=0 | 适用于多主机模式，此时 NSS 配置为硬件输入模式，一旦 NSS 引脚被拉低，SPI 将自动进入从机模式，并且产生主机配置错误，CONFERR 位置 1。 |
| 主机软件 NSS 模式 | MSTMOD = 1<br>SWNSSEN = 1<br>SWNSS = 0<br>NSSDRV：不要求 | 适用于多主机模式，一旦 SWNSS = 0，SPI 将自动进入从机模式，并且产生主机配置错误，CONFERR 位置 1。 |
| | MSTMOD = 1<br>SWNSSEN = 1<br>SWNSS = 1<br>NSSDRV：不要求 | 从机可以使用硬件或软件 NSS 模式 |

### 19.3.5. SPI 运行模式

**表 19-5. SPI 运行模式**

| 模式 | 描述 | 寄存器配置 | 数据引脚用法 |
|---|---|---|---|
| MFD | 全双工主机模式 | MSTMOD = 1<br>RO = 0<br>BDEN = 0<br>BDOEN：不要求 | MOSI：发送<br>MISO：接收 |
| MTU | 单向线连接主机发送模式 | MSTMOD = 1<br>RO = 0<br>BDEN = 0<br>BDOEN：不要求 | MOSI：发送<br>MISO：不使用 |
| MRU | 单向线连接主机接收模式 | MSTMOD = 1<br>RO = 1<br>BDEN = 0<br>BDOEN：不要求 | MOSI：不使用<br>MISO：接收 |
| MTB | 双向线连接主机发送模式 | MSTMOD = 1<br>RO = 0<br>BDEN = 1<br>BDOEN = 1 | MOSI：发送<br>MISO：不使用 |

| 模式 | 描述 | 寄存器配置 | 数据引脚用法 |
|---|---|---|---|
| MRB | 双向线连接主机接收模式 | MSTMOD = 1<br>RO = 0<br>BDEN = 1<br>BDOEN = 0 | MOSI：接收<br>MISO：不使用 |
| SFD | 全双工从机模式 | MSTMOD = 0<br>RO = 0<br>BDEN = 0<br>BDOEN：不要求 | MOSI：接收<br>MISO：发送 |
| STU | 单向线连接从机发送模式 | MSTMOD = 0<br>RO = 0<br>BDEN = 0<br>BDOEN：不要求 | MOSI：不使用<br>MISO：发送 |
| SRU | 单向线连接从机接收模式 | MSTMOD = 0<br>RO = 1<br>BDEN = 0<br>BDOEN：不要求 | MOSI：接收<br>MISO：不使用 |
| STB | 双向线连接从机发送模式 | MSTMOD = 0<br>RO = 0<br>BDEN = 1<br>BDOEN = 1 | MOSI：不使用<br>MISO：发送 |
| SRB | 双向线连接从机接收模式 | MSTMOD = 0<br>RO = 0<br>BDEN = 1<br>BDOEN = 0 | MOSI：不使用<br>MISO：接收 |

**图 19-4. 典型的全双工模式连接**

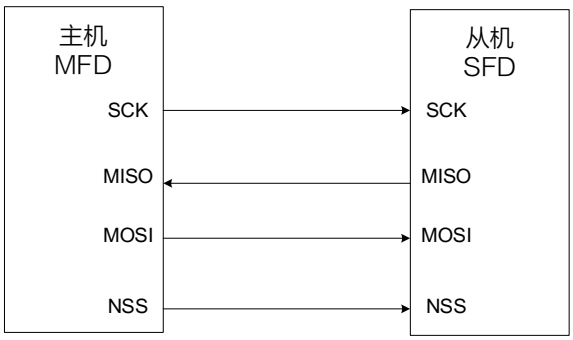


图 19-5. 典型的单工模式连接(主机：接收，从机：发送)

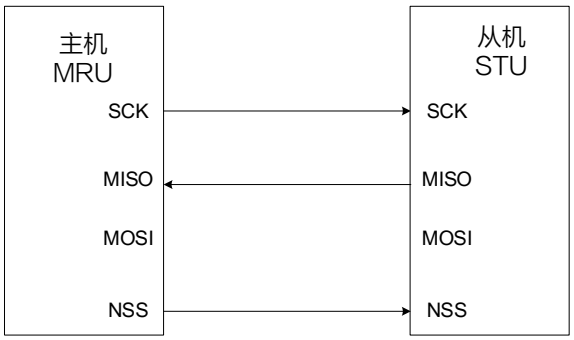



图 19-6. 典型的单工模式连接(主机：只发送，从机：接收)

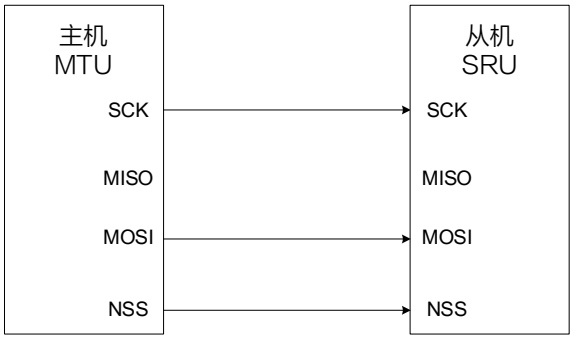
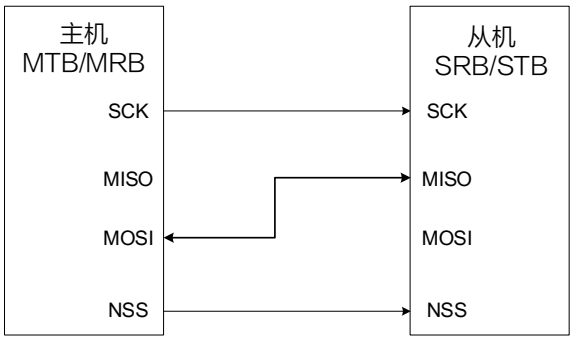



图 19-7. 典型的双向线连接




## SPI 初始化流程

在发送或接收数据之前，应用程序应遵循如下的SPI初始化流程：

1. 如果工作在主机模式或从机TI模式，配置SPI_CTL0中的PSC[2:0]位来生成预期波特率的SCK信号，或配置TI模式下的Td时间。否则，忽略此步骤。
2. 配置数据格式(SPI_CTL0中的FF16位)
3. 配置时钟时序(SPI_CTL0中的CKPL位和CKPH位)。
4. 配置帧格式(SPI_CTL0中的LF位)。
5. 按照上文***NSS功能***的描述，根据应用程序的需求，配置NSS模式(SPI_CTL0中的SWNSSEN位和NSSDRV位)。
6. 如果工作在TI模式，需要将SPI_CTL1中的TMOD位置1，否则，忽略此步骤。
7. 如果工作在 NSSP 模式，需要将 SPI_CTL1 中的 NSSP 位置 1，否则，忽略此步骤。

8. 根据*表19-5. SPI运行模式*，配置MSTMOD位、RO位、BDEN位和BDOEN位。
9. 如果工作在SPI四线模式，需要将SPI_QCTL中的QMOD位置1，如果不是，则忽略此步骤。
10. 使能SPI(将SPIEN位置1)。

**注意：**在通信过程中，不应更改CKPH、CKPL、MSTMOD、PSC[2:0]、LF位。

## SPI 基本发送和接收流程

### 发送流程

在完成初始化过程之后，SPI模块使能并保持在空闲状态。在主机模式下，当软件写一个数据到发送缓冲区时，发送过程开始。在从机模式下，当SCK引脚上的SCK信号开始翻转，且NSS引脚电平为低，发送过程开始。所以，在从机模式下，应用程序必须确保在数据发送开始前，数据已经写入发送缓冲区中。

当SPI开始发送一个数据帧时，首先将这个数据帧从数据缓冲区加载到移位寄存器中，然后开始发送加载的数据。在数据帧的第一位发送之后，TBE(发送缓冲区空)位置1。TBE标志位置1，说明发送缓冲区为空，此时如果需要发送更多数据，软件应该继续写SPI_DATA寄存器。

在主机模式下，若想要实现连续发送功能，那么在当前数据帧发送完成前，软件应该将下一个数据写入SPI_DATA寄存器中。

### 接收流程

在最后一个采样时钟边沿之后，接收到的数据将从移位寄存器存入到接收缓冲区，且RBNE(接收缓冲区非空)位置1。软件通过读SPI_DATA寄存器获得接收的数据，此操作会自动清除RBNE标志位。在MRU和MRB模式中，为了接收下一个数据帧，硬件需要连续发送时钟信号，而在全双工主机模式(MFD)中，仅当发送缓冲区非空时，硬件才接收下一个数据帧。

## SPI 不同模式下的操作流程(非 SPI 四线模式，TI 模式或 NSSP 模式)

在全双工模式下，无论是MFD模式或者SFD模式，应用程序都应该监视RBNE标志位和TBE标志位，并且遵循上文描述的操作流程。

发送模式(MTU，MTB，STU或STB)与全双工模式中的发送流程类似，不同的是需要忽略RBNE位和RXORERR位。

相比于发送模式的情况，主机接收模式(MRU或MRB)与全双工的接收流程大不相同。在MRU模式或MRB模式下，在SPI使能后，SPI产生连续的SCK信号，直到SPI停止。所以，软件应该忽略TBE标志位，并且在RBNE位置1后，读出接收缓冲区内的数据，否则，将会产生接收过载错误。

除了忽略TBE标志位，且只执行上述的接收流程之外，从机接收模式(SRU或SRB)与全双工模式类似。

### SPI TI 模式

SPI TI模式将NSS作为一种特殊的帧头标志信号，它的操作流程与上文描述的常规模式类似。上文描述的模式(MFD，MTU，MRU，MTB，MRB，SFD，STU，SRU，STB和SRB)都支持TI模式。但是，在TI模式中，SPI_CTL0中的CKPL位和CKPH位是没有意义的，SCK信号的采样边沿为下降沿。

**图 19-8. 主机 TI 模式在不连续发送时的时序图**

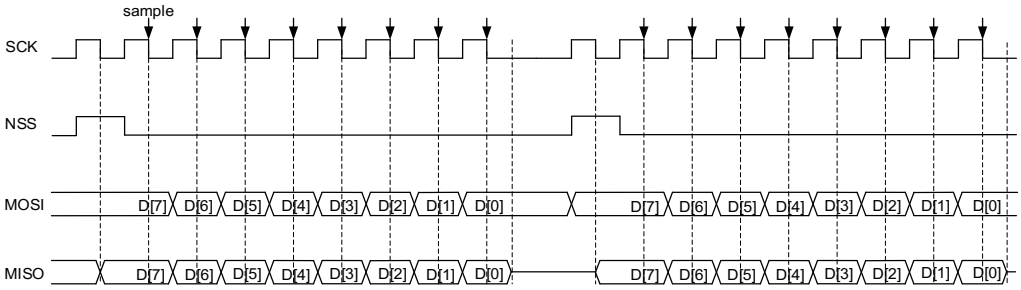
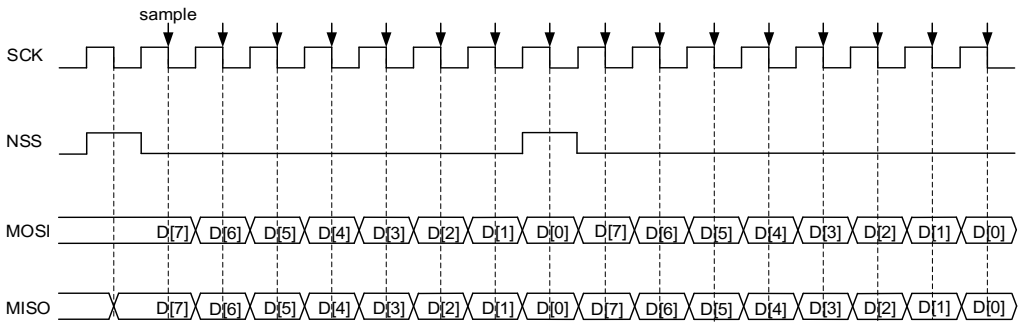
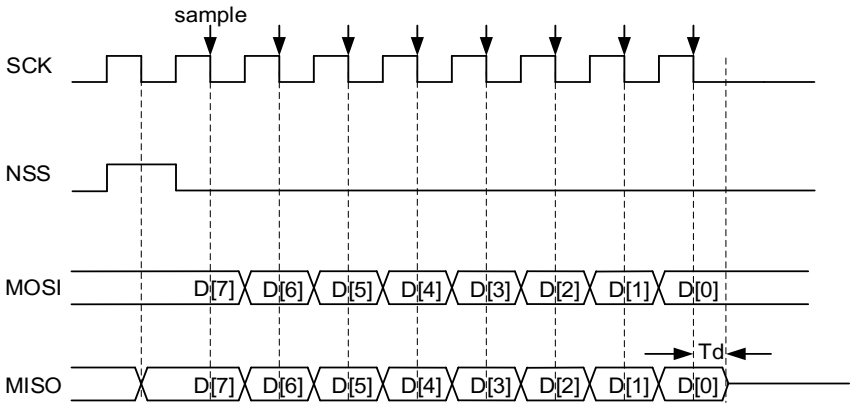



**图 19-9. 主机 TI 模式在连续发送时的时序图**




在主机TI模式下，SPI模块可实现连续传输或者不连续传输。如果主机写SPI_DATA的速度很快，那么就是连续传输，否则，为不连续传输。在不连续传输中，在每个字节传输前需要一个额外的时钟周期。在连续传输中，额外的时钟周期只存在于第一个字节之前，随后字节的起始时钟周期被前一个字节的最后一位的时钟周期覆盖。

**图 19-10. 从机 TI 模式时序图**



在从机TI模式中，在SCK信号的最后一个上升沿，从机开始发送最后一个字节的LSB位，在半位的时间之后，主机开始采集数据。为了确保主机采集到正确的数据，在释放该引脚之前，从机需要在SCK信号的下降沿之后继续驱动该位一段时间，这段时间称为$T_d$，$T_d$通过SPI_CTL0寄存器中的PSC[2:0]位来设置。

$$T_d=\frac{T_{bit}}{2}+5*T_{pclk} \quad (19\text{-}1)$$

例如，如果PSC[2:0] = 010，那么$T_d$数值为$9*T_{pclk}$。

在从机模式下，从机需要监视NSS信号，如果检测到错误的NSS信号，将会置位FERR标志位。例如，NSS信号在一个字节的中间位发生翻转。

### NSS 脉冲模式操作流程

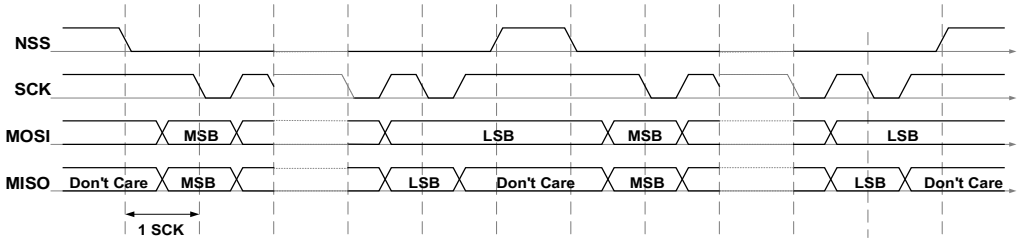
配置SPI_CTL1寄存器中的NSSP位使能该功能，为了确保使用该功能实现，需满足以下几个条件：配置设备为主机模式，使用普通SPI协议的数据帧格式，同时在第一个时钟跳变沿采样数据。

总之：MSTMOD = 1，NSSP = 1，CKPH = 0。

当使用NSS脉冲模式时，根据内部数据发送缓冲区的状态，NSS脉冲会在两个连续的数据帧之间产生，且持续时间至少为1个SCK时钟周期。如果数据发送缓冲区保持为空，可能会持续多个SCK时钟周期。NSS脉冲功能专为单一的主从应用设计，支持从机锁存数据。

下图描述了NSS脉冲模式在主机连续发送时的时序图。

**图 19-11. NSS 脉冲模式时序图(主机连续发送)**




### SPI 四线模式操作流程

SPI四线模式用于控制四线SPI flash外设。

要配置成SPI四线模式，首先要确认TBE位置1，且TRANS位清零，然后将SPI_QCTL寄存器中的QMOD位置1。在SPI四线模式，SPI_CTL0寄存器中BDEN位、BDOEN位、CRCEN位、CRCNT位、FF16位、RO位和LF位保持清零，且MSTMOD位置1，以保证SPI工作于主机模式。SPIEN位、PSC位、CKPL位和CKPH位根据需要进行配置。

SPI四线模式有两种运行模式：四线写模式和四线读模式，通过SPI_QCTL寄存器中的QRD位进行配置。

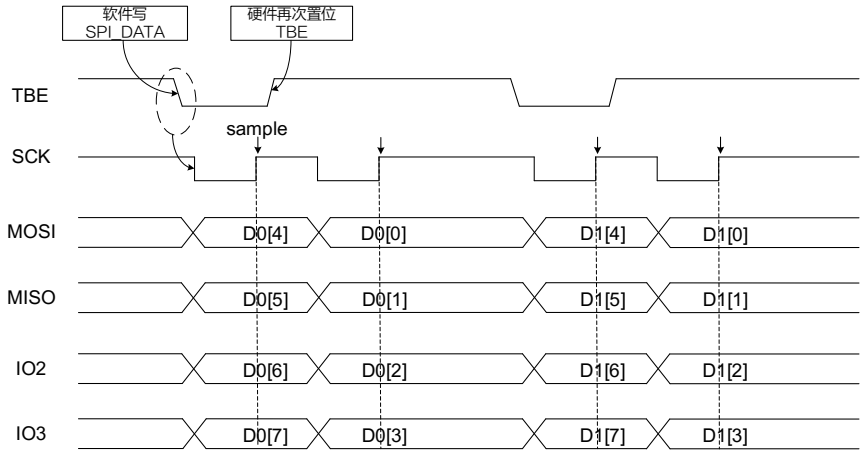
### 四线写模式

当SPI_QCTL寄存器中的QMOD位置1且QRD位清零时，SPI工作在四线写模式。在四线写模式中，MOSI、MISO、IO2和IO3都用作输出引脚，在SCK产生时钟信号后，一旦数据写入SPI_DATA寄存器(TBE位清零)且SPIEN位置1时，将会通过这四个引脚发送写入的数据。SPI开始数据传输之后，每发送一个数据帧都要检测TBE标志位，若不能满足条件则停止传输。

四线模式下发送操作流程：

1. 根据应用需求，配置SPI_CTL0和SPI_CTL1中的时钟预分频、时钟极性、相位等参数；
2. 将SPI_QCTL中的QMOD位置1，然后将SPI_CTL0中的SPIEN位置1来使能SPI功能；
3. 向SPI_DATA寄存器中写入一个字节的数据，TBE标志位将会清零；
4. 等待硬件将TBE位重新置位，然后写入下一个字节数据。

**图 19-12. SPI 四线模式四线写操作时序图**



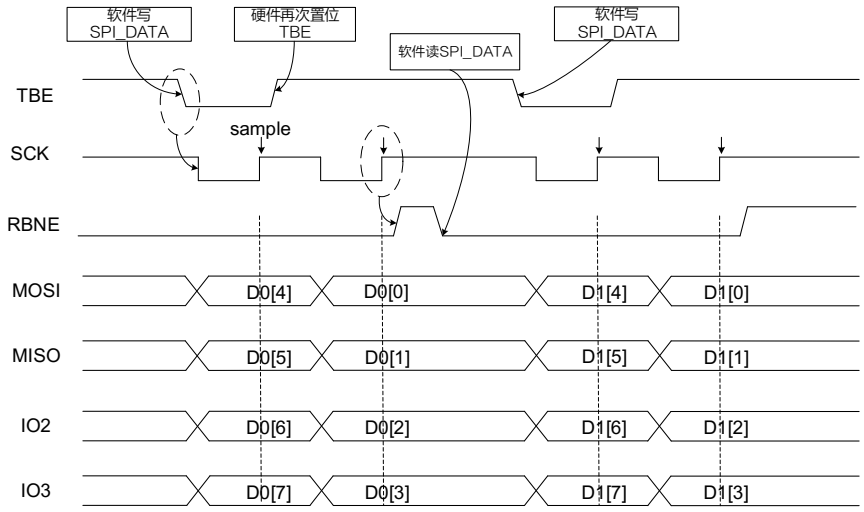

### 四线读模式

当SPI_QCTL寄存器中的QMOD位都置1时，SPI工作在四线读模式。在四线读模式中，MOSI、MISO、IO2和IO3都用作输入引脚，一旦数据写入SPI_DATA寄存器(TBE位清零)且SPIEN位置1时，在SCK信号线产生时钟信号。写数据到SPI_DATA寄存器只是为了产生SCK时钟信号，所以可以写入任何数据。SPI开始数据传输之后，每发送一个数据帧都要检测SPIEN位和TBE位，若条件不满足则停止传输。所以软件需要一直向SPI_DATA写空闲数据，以产生SCK时钟信号。

四线模式下接收操作流程：

1. 根据应用需求，配置SPI_CTL0和SPI_CTL1中时钟预分频、时钟极性、相位等参数；
2. 将SPI_QCTL中的QMOD位和QRD位置1，然后将SPI_CTL0中的SPIEN位置1来使能SPI功能；
3. 写任意数据(例如0xFF)到SPI_DATA寄存器；
4. 等待RBNE位置1，然后读SPI_DATA寄存器来获取接收的数据；
5. 写任意数据(例如0xFF)到SPI_DATA寄存器，以接收下一个字节数据。

**图 19-13. SPI 四线模式四线读操作时序图**




## SPI 停止流程

不同运行模式下采用不同的流程来停止SPI功能。

### MFD SFD

等待最后一个RBNE位并接收最后一个数据，等待TBE=1和TRANS=0，最后，通过清零SPIEN位关闭SPI。

### MTU MTB STU STB

将最后一个数据写入SPI_DATA寄存器，等待TBE位置1，等待TRANS位清零，通过清零SPIEN位关闭SPI。

### MRU MRB

等待倒数第二个RBNE位置1，从SPI_DATA寄存器读数据，等待一个SCK时钟周期，然后通过清零SPIEN位关闭SPI。等待最后一个RBNE位置1，并从SPI_DATA读数据。

### SRU SRB

应用程序可以在任何时候关闭SPI功能，然后等待TRANS=0以确保当前通信过程结束。

### TI模式

TI模式的停止流程与上面描述过程相同。

### NSS脉冲模式

NSS脉冲模式的停止流程与上面描述过程相同。

### SPI四线模式

在禁用SPI四线模式和关闭SPI功能之前，软件应该先检查：TBE位置1，TRANS位清零，

SPI_QCTL中的QMOD位和SPI_CTL0中的SPIEN位清零。

### 19.3.6. DMA 功能

DMA功能在传输过程中将应用程序从数据读写过程中释放出来，从而提高了系统效率。

通过置位SPI_CTL1寄存器中的DMATEN位和DMAREN位，使能SPI模式的DMA功能。为了使用DMA功能，软件首先应当正确配置DMA模块，然后通过初始化流程配置SPI模块，最后使能SPI。

SPI使能后，如果DMATEN位置1，每当TBE=1时，SPI将会发出一个DMA请求，然后DMA应答该请求，并自动写数据到SPI_DATA寄存器。如果DMAREN位置1，每当RBNE=1时，发出一个DMA请求，然后DMA应答该请求，并自动从SPI_DATA寄存器读取数据。

### 19.3.7. CRC 功能

SPI模块包含两个CRC计算单元：分别用于发送数据和接收数据。CRC计算单元使用SPI_CRCPOLY寄存器中定义的多项式。

通过配置SPI_CTL0中的CRCEN位使能CRC功能。对于数据线上每个发送和接收的数据，CRC单元逐位计算CRC值，计算得到的CRC值可以从SPI_TCRC寄存器和SPI_RCRC寄存器中读取。

为了传输计算得到的CRC值，应用程序需要在最后一个数据写入发送缓冲区之后，设置SPI_CTL0中的CRCNT位。在全双工模式(MFD或SFD)，当SPI发送一个CRC值并且准备校验接收到的CRC值时，会将最新接收到的数据当作CRC值。在接收模式(MRB，MRU，SRU 和SRB)下，在倒数第二个数据帧被接收后，软件将CRCNT位置1。在CRC校验失败时，CRCERR错误标志位将会置1。

如果使能了DMA功能，软件不需要设置CRCNT位，硬件将会自动处理CRC传输和校验。

**注意：**当SPI处于从机模式且CRC功能使能时，无论SPI是否使能，CRC计算器都对输入SCK时钟敏感。只有当时钟稳定时，软件才能启用CRC，以避免错误的CRC计算。当SPI作为从机工作时，在数据阶段和CRC阶段之间，内部NSS信号需要保持低电平。

当配置SPI为从模式并且使用CRC功能时，即使NSS引脚为高时仍然会执行CRC的计算(当NSS信号为高时，只要SCK引脚上有时钟脉冲，则CRC计算会继续执行)。当主设备交替地与多个从设备进行通信时，将会出现这种情况,此时建议在NSS信号为低时重启CRC功能。当从设备未选中(NSS信号为高)转换到被选中为一个新的从设备(NSS信号为低)的时候，为了保持主从设备端下次CRC计算结果的同步，应该清除主从两端的CRC数值。建议按照下述步骤清除CRC数值：

1. 关闭SPI模块(SPIEN=0)；
2. 清除CRCEN位(CRCEN=0)；
3. 设置CRCEN位(CRCEN=1)；
4. 使能SPI模块(SPIEN=1)。

### 19.3.8. SPI 中断

#### 状态标志位

■ **发送缓冲区空标志位(TBE)**

当发送缓冲区为空时，TBE置位。软件可以通过写SPI_DATA寄存器将下一个待发送数据写入发送缓冲区。

■ **接收缓冲区非空标志位(RBNE)**

当接收缓冲区非空时，RBNE置位，表示此时接收到一个数据，并已存入到接收缓冲区中，软件可以通过读SPI_DATA寄存器来读取此数据。

■ **SPI通信进行中标志位(TRANS)**

TRANS位是用来指示当前传输是否正在进行或结束的状态标志位，它由内部硬件置位和清除，无法通过软件控制。该标志位不会产生任何中断。

#### 错误标志

■ **配置错误标志(CONFERR)**

在主机模式中，CONFERR位是一个错误标志位。在硬件NSS模式中，如果NSSDRV没有使能，当NSS被拉低时，CONFERR位被置1。在软件NSS模式中，当SWNSS位为0时，CONFERR位置1。当CONFERR位置1时，SPIEN位和MSTMOD位由硬件清除，SPI关闭，设备强制进入从机模式。

在CONFERR位清零之前，SPIEN位和MSTMOD位保持写保护，从机的CONFERR位不能置1。在多主机配置中，设备可以在CONFERR位置1时进入从机模式，这意味着发生了系统控制的多主冲突。

■ **接收过载错误(RXORERR)**

在RBNE位为1时，如果再有数据被接收，RXORERR位将会置1。这说明，上一帧数据还未被读出而新的数据已经接收了。接收缓冲区的内容不会被新接收的数据覆盖，所以新接收的数据丢失。

■ **帧错误(FERR)**

在TI从机模式下，从机也要监视NSS信号，如果检测到错误的NSS信号，将会置位FERR标志位。例如，NSS信号在一个字节的中间位发生翻转。

■ **CRC错误(CRCERR)**

当CRCEN位置1时，SPI_RCRC寄存器中接收到的数据的CRC计算值将会和紧随着最后一帧数据接收到的CRC值进行比较，当两者不同时，CRCERR位将会置1。

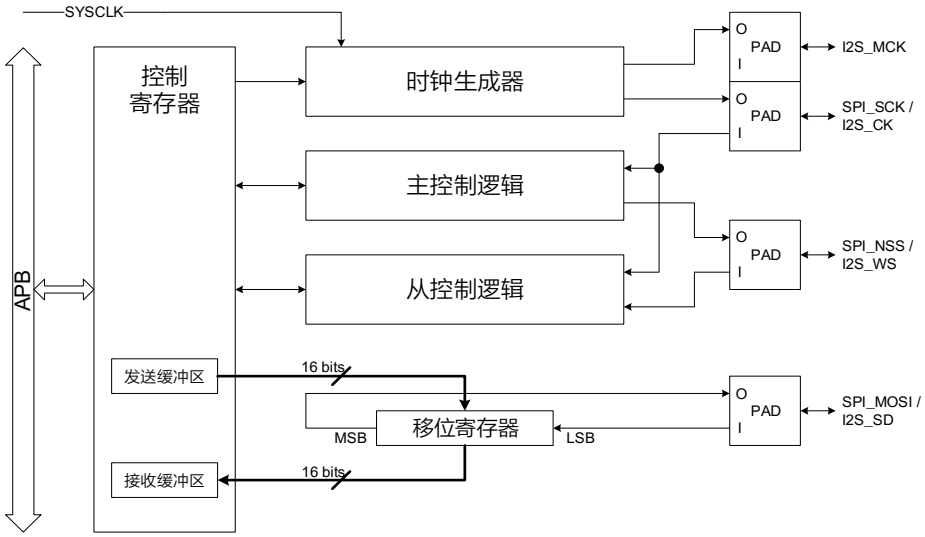
表 19-6. SPI 中断请求

| 中断事件 | 描述 | 清除方式 | 中断使能位 |
|---|---|---|---|
| TBE | 发送缓冲区空 | 写SPI_DATA寄存器 | TBEIE |
| RBNE | 接收缓冲区非空 | 读SPI_DATA寄存器 | RBNEIE |
| CONFERR | 配置错误 | 读或写 SPI_STAT 寄存器，然后写 SPI_CTL0 寄存器 | ERRIE |
| RXORERR | 接收过载错误 | 读SPI_DATA寄存器，然后读 SPI_STAT寄存器 | |
| CRCERR | CRC错误 | 写0到CRCERR位 | |
| FERR | TI模式帧错误 | 写0到FERR位 | |

## 19.4. I2S 功能说明

### 19.4.1. I2S 结构框图

图 19-14. I2S 结构框图




I2S功能有5个子模块，分别是控制寄存器、时钟生成器、主机控制逻辑、从机控制逻辑和移位寄存器。所有的用户可配置寄存器都在控制寄存器模块实现，其中包括发送缓冲区和接收缓冲区。时钟生成器用来在主机模式下生成I2S通信时钟。主机控制逻辑用来在主机模式下生成I2S_WS信号并控制通信。从机控制逻辑根据接收到的I2S_CK和I2S_WS信号来控制从机模式的通信。移位寄存器控制I2S_SD上的串行数据发送和接收。

### 19.4.2. I2S 信号线描述

I2S接口有4个引脚，分别是I2S_CK、I2S_WS、I2S_SD和I2S_MCK。I2S_CK是串行时钟信号，与SPI_SCK共享引脚。I2S_WS是数据帧控制信号，与SPI_NSS共享引脚。I2S_SD是串行数据信号，与SPI_MOSI共享引脚。I2S_MCK是主时钟信号，它提供了一个256倍于Fs的时钟频率，其中Fs是音频采样率。

### 19.4.3. I2S 音频标准

I2S音频标准是通过设置SPI_I2SCTL寄存器中的I2SSTD位来选择的，可以选择四种音频标准：I2S飞利浦标准，MSB对齐标准，LSB对齐标准和PCM标准。除PCM之外的所有标准都是两个通道(左通道和右通道)的音频数据分时复用I2S接口的，并通过I2S_WS信号来区分当前数据属于哪个通道。对于PCM标准，I2S_WS信号表示帧同步信息。

数据长度和通道长度可以通过SPI_I2SCTL寄存器中的DTLEN位和CHLEN位来设置。由于通道长度必须大于或等于数据长度，所以有四种数据包类型可供选择。它们分别是：16位数据打包成16位数据帧格式，16位数据打包成32位数据帧格式，24位数据打包成32位数据帧格式，32位数据打包成32位数据帧格式。用于发送和接收的数据缓冲区都是16位宽度。所以，要完成数据长度为24位或32位的数据帧传输，SPI_DATA寄存器需要被访问2次；而要完成数据长度为16位的数据帧传输，SPI_DATA寄存器只需被访问1次。如需将16位数据打包成32位数据帧，硬件会自动插入16位0将16位数据扩展为32位格式。

对于所有标准和数据包类型来说，数据的最高有效位总是最先被发送的。对于所有基于两通道分时复用的标准来说，总是先发送左通道，然后是右通道。

#### I2S 飞利浦标准

对于I2S飞利浦标准，I2S_WS和I2S_SD在I2S_CK的下降沿变化，I2S_WS在数据的前一个时钟开始有效。各种配置情况的时序图如下所示。

**图 19-15. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=0)**

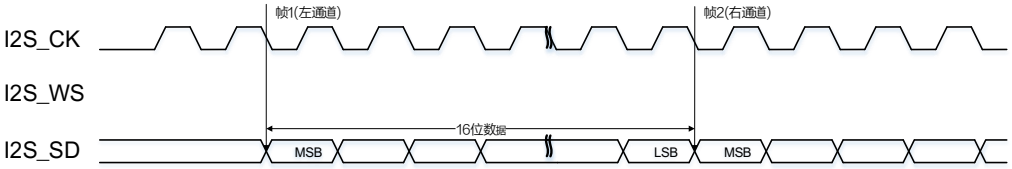
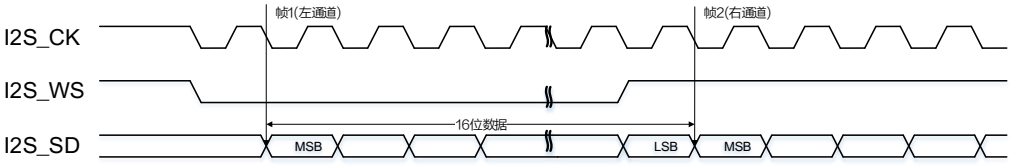



**图 19-16. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=1)**




当 16 位数据打包成 16 位数据帧时，每完成一帧数据的传输只需要访问 SPI_DATA 寄存器一次。

**图 19-17. I2S 飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=0)**

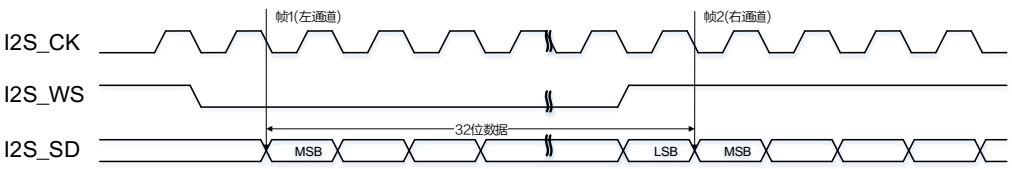
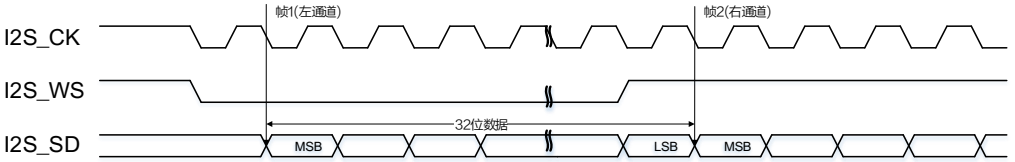


**图 19-18. I2S 飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=1)**




当32位数据打包成32位数据帧的帧格式时，每完成1帧数据的传输需要访问SPI_DATA寄存器2次。在发送模式下，如果要发送一个32位数据，第一个写入SPI_DATA寄存器的数据应该是高16位数据，第二个数据应该是低16位数据。在接收模式下，如果要接收一个32位数据，第一个从SPI_DATA寄存器读到的数据应该是高16位数据，第二个数据应该是低16位数据。

**图 19-19. I2S 飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=0)**

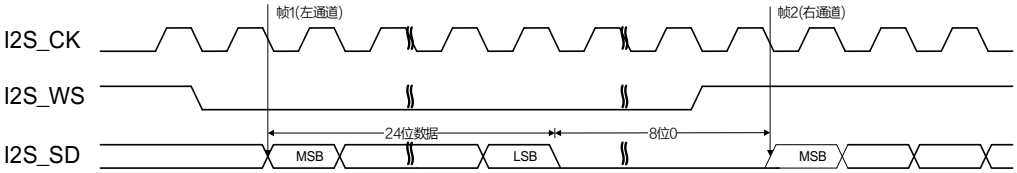
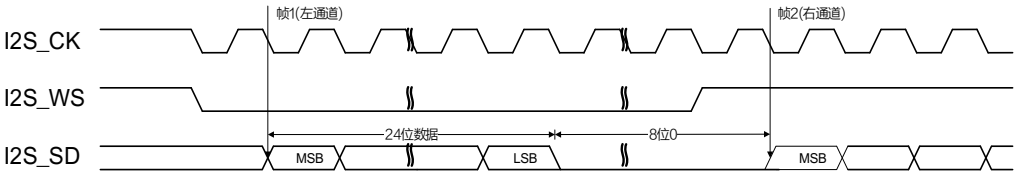



**图 19-20. I2S 飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=1)**




当24位数据打包成32位数据帧的帧格式时，每完成1帧数据的传输需要访问SPI_DATA寄存器2次。在发送模式下，如果要发送一个24位数据D[23:0]，第一个写入SPI_DATA寄存器的数据应该是高16位数据D[23:8]，第二个数据应该是一个16位数据，该16位数据的高8位是D[7:0]，低8位数据可以是任意值。在接收模式下，如果要接收一个24位数据D[23:0]，第一个从SPI_DATA寄存器读到的数据应该是高16位数据D[23:8]，第二个数据应该是一个16位数据，该16位数据的高8位是D[7:0]，低8位数据全是0。

**图 19-21. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=0)**

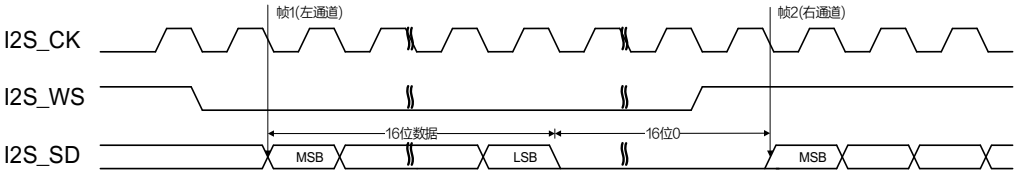
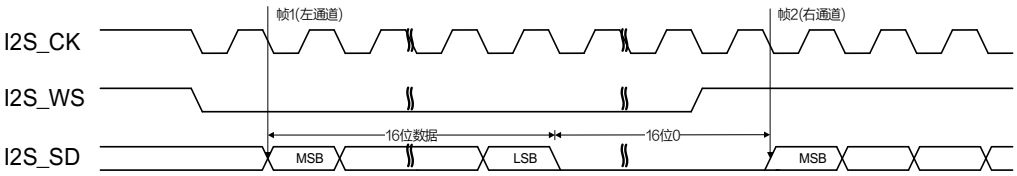



**图 19-22. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=1)**




当16位数据打包成32位数据帧时，每完成一帧数据的传输只需要访问SPI_DATA寄存器一次。

为了将该16位数据扩展成32位数据，剩下的16位被硬件强制填充为0x0000。

### MSB 对齐标准

对于MSB对齐标准，I2S_WS和I2S_SD在I2S_CK的下降沿变化。SPI_DATA 寄存器的处理方式与I2S飞利浦标准完全相同。各个配置情况的时序图如下所示。

**图 19-23. MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=0)**

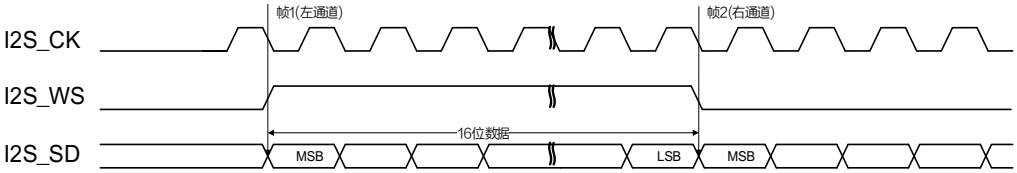



**图 19-24. MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=1)**

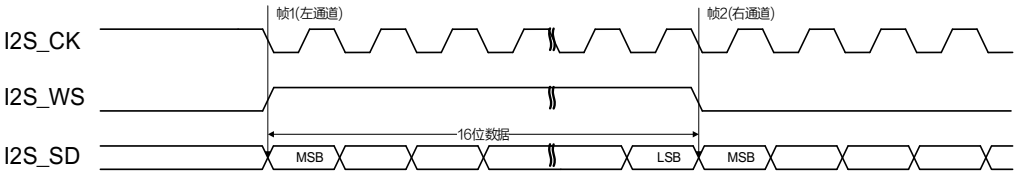



**图 19-25. MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=0)**

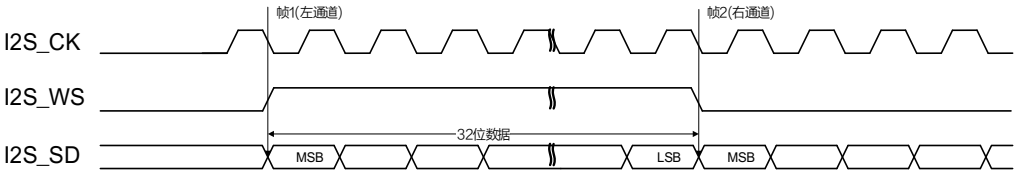



**图 19-26. MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=1)**

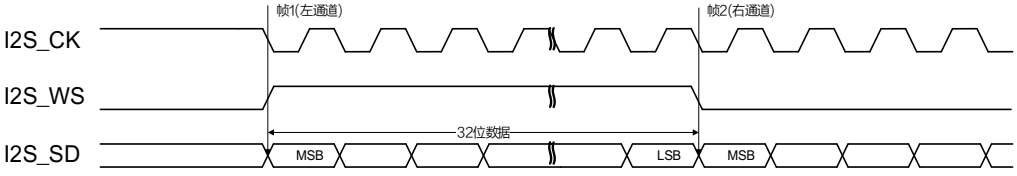



**图 19-27. MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0)**

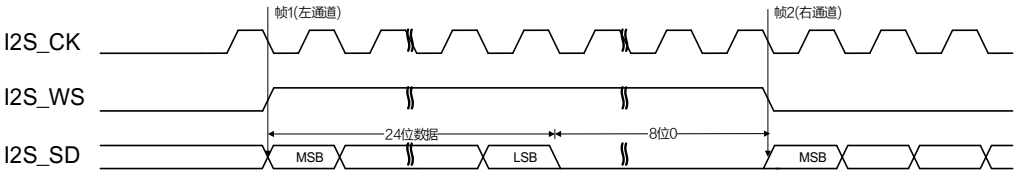


**图 19-28. MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1)**

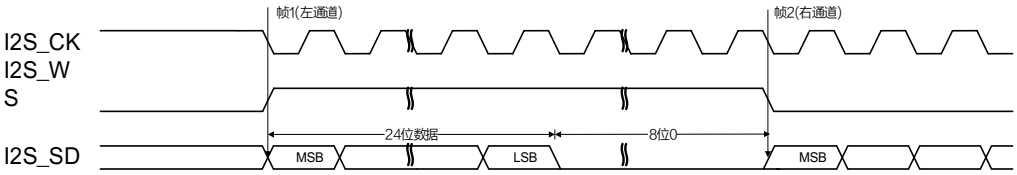



**图 19-29. MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0)**

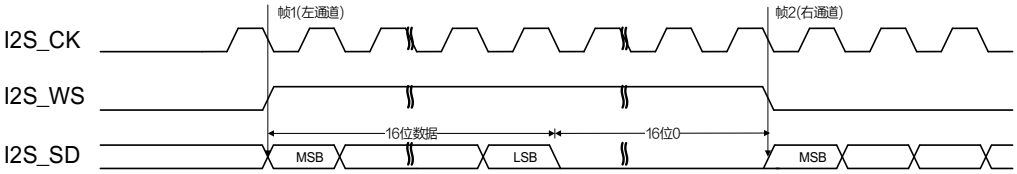
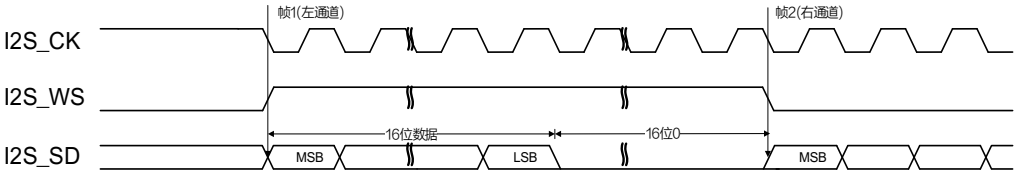



**图 19-30. MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1)**




## LSB 对齐标准

对于LSB对齐标准，I2S_WS和I2S_SD在I2S_CK的下降沿变化。在通道长度与数据长度相同的情况下，LSB对齐标准和MSB对齐标准是完全相同的。对于通道长度大于数据长度的情况，LSB对齐标准的有效数据与最低位对齐，而MSB对齐标准的有效数据与最高位对齐。通道长度大于数据长度的各种配置情况时序图如下所示。

**图 19-31. LSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0)**

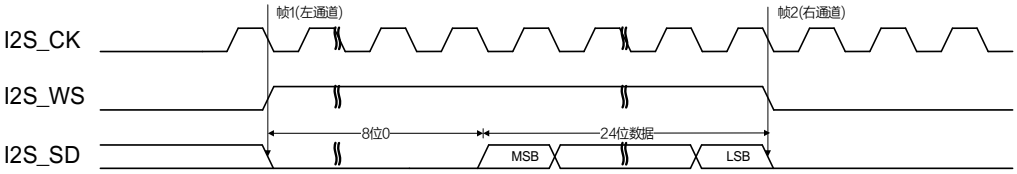
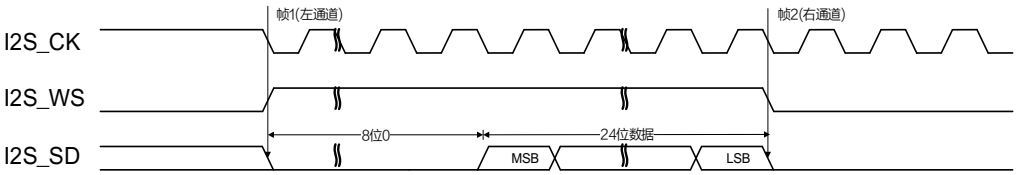



**图 19-32. LSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1)**




当24位数据打包成32位数据帧的帧格式时，每完成1帧数据的传输需要访问SPI_DATA寄存器2次。在发送模式下，如果要发送一个24位数据D[23:0]，第一个写入SPI_DATA寄存器的数据应该是一个16位数据，该16位数据的高8位可以是任意值，低8位是D[23:16]，第二个数据应该是低16位数据D[15:0]。在接收模式下，如果要接收一个24位数据D[23:0]，第一个从SPI_DATA

寄存器读到的数据应该是一个16位数据，该16位数据的高8位是0，低8位是D[23:16]，第二个数据应该是低16位数据D[15:0]。

**图 19-33. LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0)**

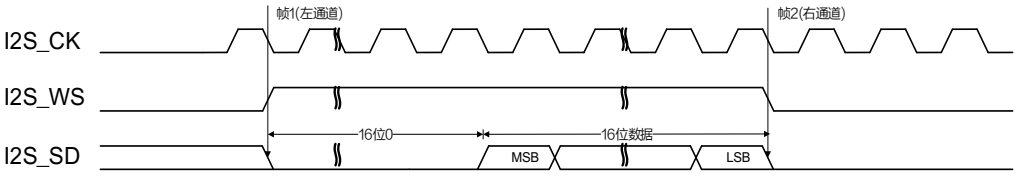
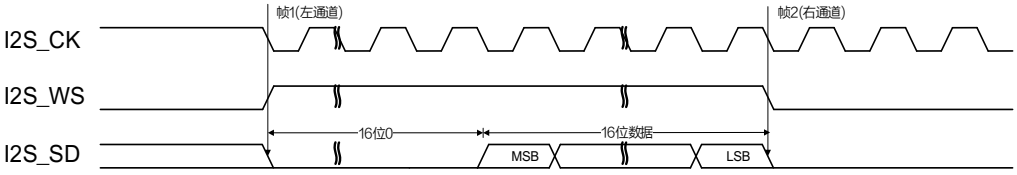



**图 19-34. LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1)**




当16位数据打包成32位数据帧时，每完成一帧数据的传输只需要访问SPI_DATA寄存器一次。为了将该16位数据扩展成32位数据，剩下的16位被硬件强制填充为0x0000。

## PCM 标准

对于PCM标准，I2S_WS和I2S_SD在I2S_CK的上升沿变化，I2S_WS信号表示帧同步信息。可以通过SPI_I2SCTL寄存器的PCMSMOD位来选择短帧同步模式和长帧同步模式。SPI_DATA寄存器的处理方式与I2S飞利浦标准完全相同。短帧同步模式的各种配置情况时序图如下所示。

**图 19-35. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=0)**

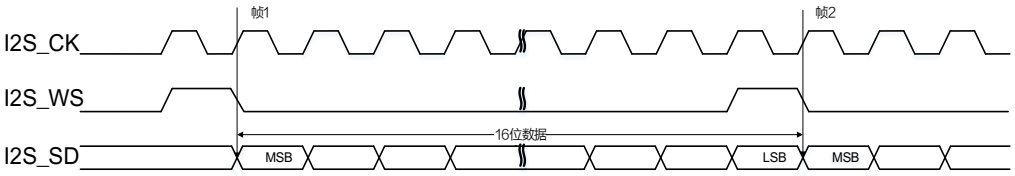



**图 19-36. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=1)**

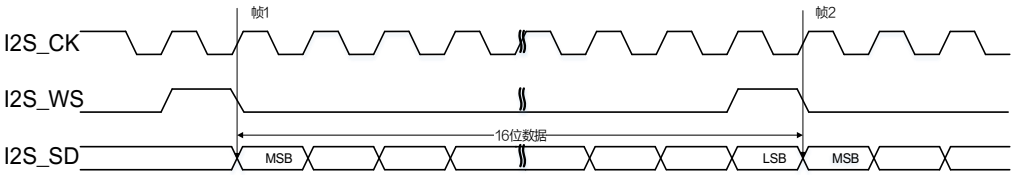



**图 19-37. PCM 标准短帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=0)**

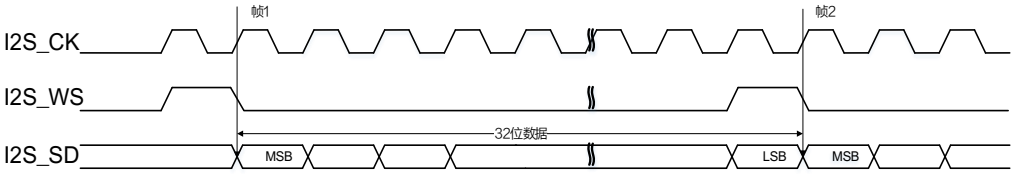


图 19-38. PCM 标准短帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=1)

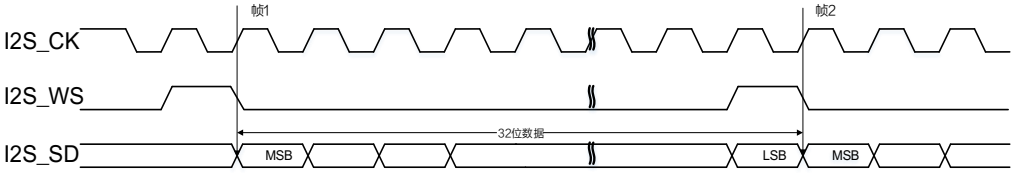



图 19-39. PCM 标准短帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=0)

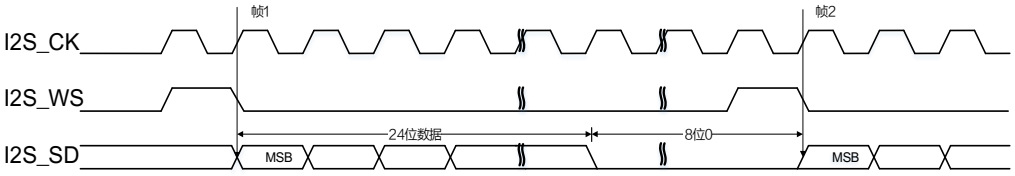



图 19-40. PCM 标准短帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=1)

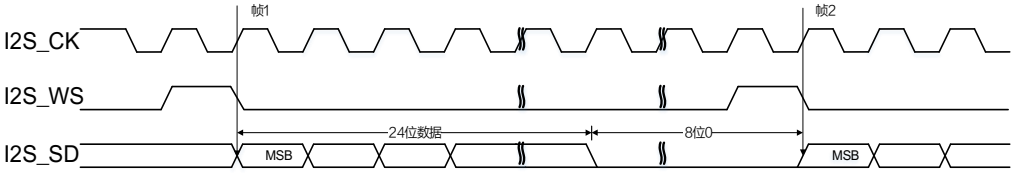



图 19-41. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=0)

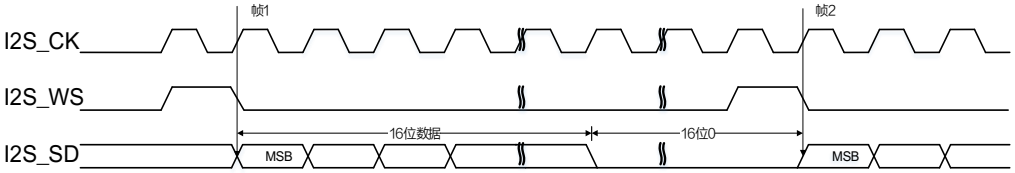
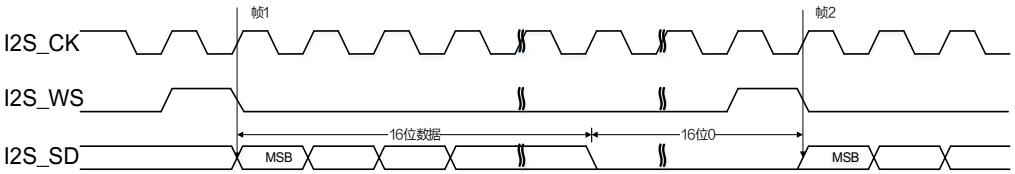



图 19-42. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=1)




长帧同步模式的各种配置情况时序图如下所示。

图 19-43. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=0)

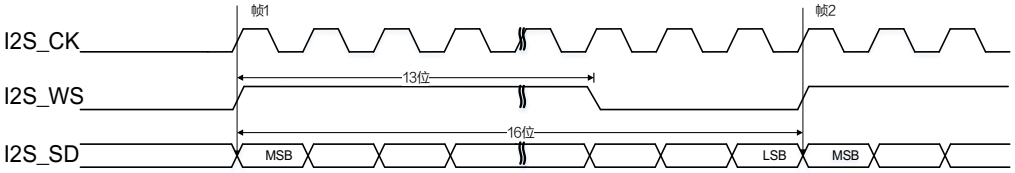


图 19-44. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=1)

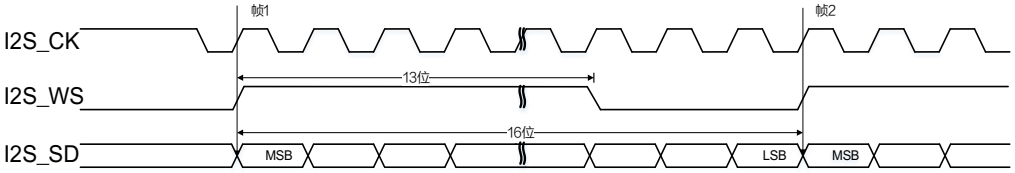



图 19-45. PCM 标准长帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=0)

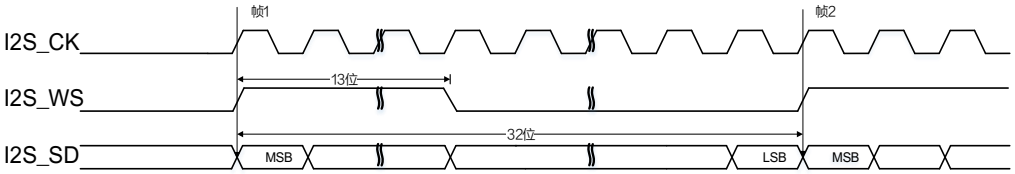



图 19-46. PCM 标准长帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=1)

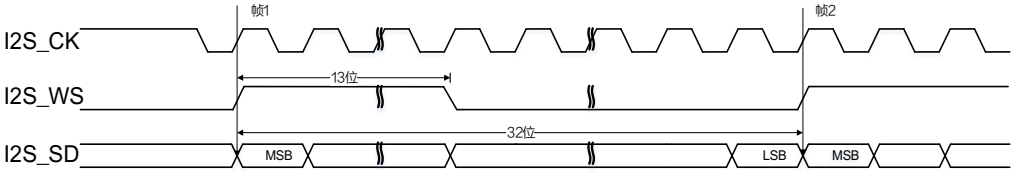



图 19-47. PCM 标准长帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=0)

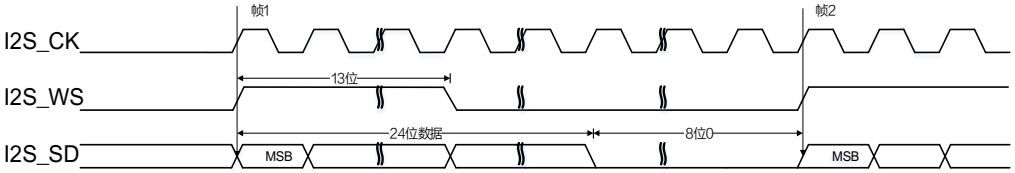



图 19-48. PCM 标准长帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=1)

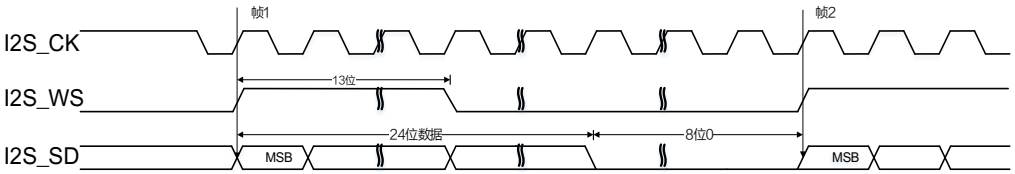



图 19-49. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=0)

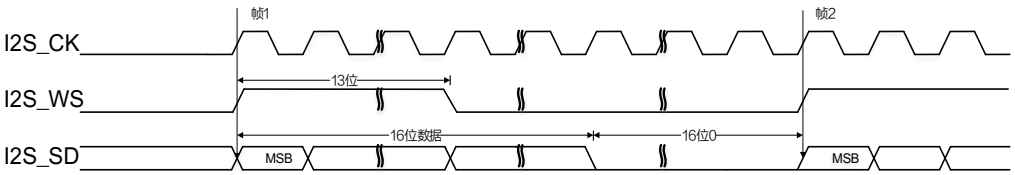
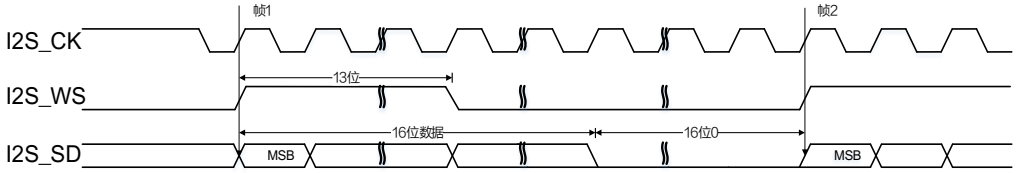
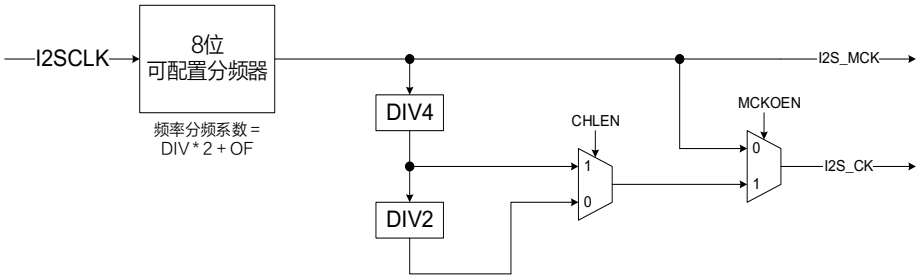


**图 19-50. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=1)**




### 19.4.4. I2S 时钟

**图 19-51. I2S 时钟生成结构框图**




I2S时钟生成器框图如*图 19-51. I2S时钟生成结构框图*所示。I2S接口时钟是通过SPI_I2SPSC寄存器的DIV位，OF位和MCKOEN位以及SPI_I2SCTL寄存器的CHLEN位来配置的。时钟源是系统时钟(CK_SYS)。I2S比特率可以通过*表19-7. I2S比特率计算公式*所示的公式计算。

**表 19-7. I2S 比特率计算公式**

| MCKOEN | CHLEN | 公式 |
|---|---|---|
| 0 | 0 | I2SCLK / (DIV * 2 + OF) |
| 0 | 1 | I2SCLK / (DIV * 2 + OF) |
| 1 | 0 | I2SCLK / (8 * (DIV * 2 + OF)) |
| 1 | 1 | I2SCLK / (4 * (DIV * 2 + OF)) |

音频采样率(Fs)和I2S比特率的关系由如下公式定义：

Fs = I2S比特率 / (通道长度 * 通道数)

所以，为了得到期望的音频采样率，时钟生成器需要按*表19-8. 音频采样频率计算公式*所列的公式进行配置。

表 19-8. 音频采样频率计算公式

| MCKOEN | CHLEN | 公式 |
|---|---|---|
| 0 | 0 | I2SCLK / (32 * (DIV * 2 + OF)) |
| 0 | 1 | I2SCLK / (64 * (DIV * 2 + OF)) |
| 1 | 0 | I2SCLK / (256 * (DIV * 2 + OF)) |
| 1 | 1 | I2SCLK / (256 * (DIV * 2 + OF)) |

**注意：** I2S串行时钟的配置值需设置为低于PCLK时钟的1/6倍以下(不包含1/6)。

## 19.4.5. 运行

### 运行模式

运行模式是通过SPI_I2SCTL寄存器的I2SOPMOD位来选择的。共有四种运行模式可供选择：主机发送模式，主机接收模式，从机发送模式和从机接收模式。各种运行模式下I2S接口信号的方向如*表19-9. 各种运行模式下I2S接口信号的方向*所示。

表 19-9. 各种运行模式下 I2S 接口信号的方向

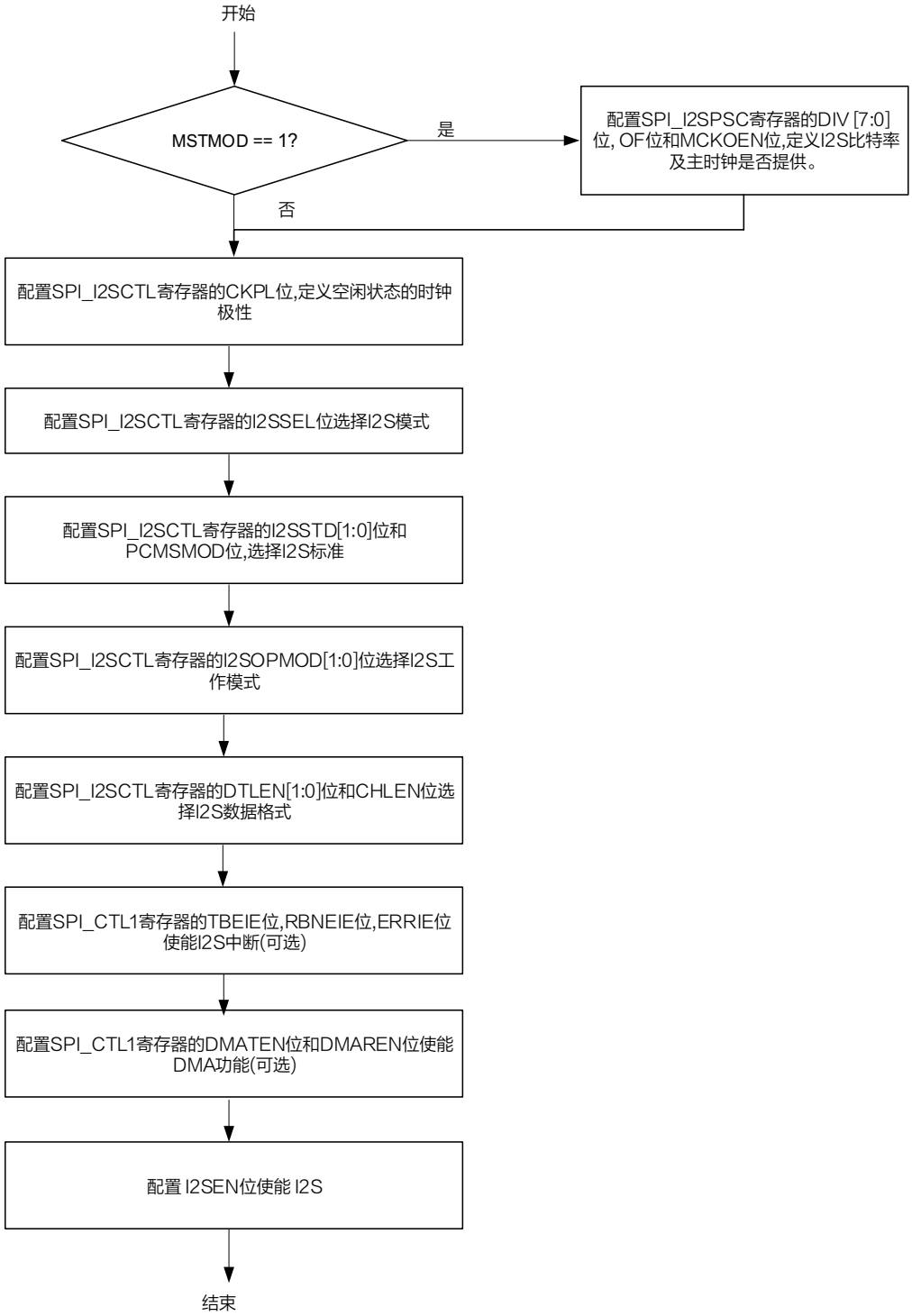
| 运行模式 | I2S_MCK | I2S_CK | I2S_WS | I2S_SD |
|---|---|---|---|---|
| 主机发送 | 输出或 NU(1) | 输出 | 输出 | 输出 |
| 主机接收 | 输出或 NU(1) | 输出 | 输出 | 输入 |
| 从机发送 | 输出或 NU(1) | 输入 | 输入 | 输出 |
| 从机接收 | 输出或 NU(1) | 输入 | 输入 | 输入 |

1. NU表示该引脚没有被I2S使用，可以用于其他功能。

### I2S 初始化流程

I2S初始化过程如*图19-52. I2S初始化流程*所示。

**图 19-52. I2S 初始化流程**




## I2S 主机发送流程

TBE标志位被用来控制发送流程。如前文所述，TBE标志位表示发送缓冲区空，此时，如果SPI_CTL1寄存器的TBEIE位为1，将产生中断。首先，发送缓冲区为空(TBE为1)，且移位寄存

器中没有发送序列。当16位数据被写入SPI_DATA寄存器时(TBE变为0)，数据立即从发送缓冲区装载到移位寄存器中(TBE变为1)。此时，发送序列开始。

数据是并行地装载到16位移位寄存器中的，然后串行地从I2S_SD引脚发出(高位先发)。下一个数据应该在TBE为1时写入SPI_DATA寄存器。数据写入SPI_DATA寄存器之后，TBE变为0。当前发送序列结束时，发送缓冲区的数据会自动装载到移位寄存器中，然后TBE标志变回1。为了保证连续的音频数据发送，下一个将要发送的数据必须在当前发送序列结束之前写入SPI_DATA寄存器。

对于除PCM标准外的所有标准，I2SCH标志用来区别当前传输数据所属的通道。I2SCH标志在每次TBE标志由0变1的时候更新。刚开始I2SCH标志为0，表示左通道的数据应该被写入SPI_DATA寄存器。

为了关闭I2S，I2SEN位必须在TBE标志为1且TRANS标志为0之后清零。

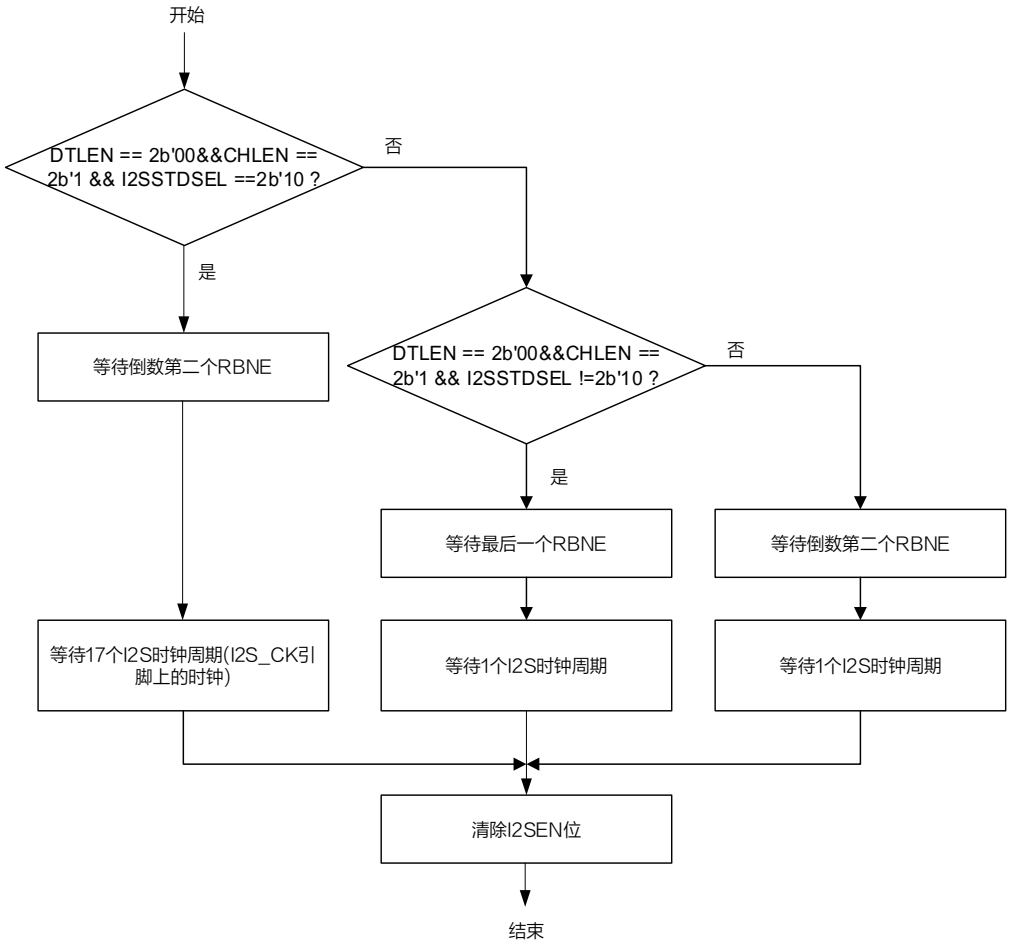
### I2S 主机接收流程

RBNE标志被用来控制接收序列。如前文所述，RBNE标志表示接收缓冲区非空，如果SPI_CTL1寄存器的RBNEIE位为1，将产生中断。当SPI_I2SCTL寄存器的I2SEN位被置1时，接收流程立即开始。首先，接收缓冲区为空(RBNE为0)。当一个接收流程结束时，接收到的数据将从移位寄存器装载到接收缓冲区(RBNE变为1)。当RBNE为1时，用户应该将数据从SPI_DATA寄存器中读走。读操作完成后，RBNE变为0。必须在下一次接收结束之前读走SPI_DATA寄存器中的数据，否则将发生接收过载错误。此时RXORERR标志位会被置1，如果SPI_CTL1寄存器的ERRIE位为1，将会产生中断。这种情况下，必须先关闭I2S再打开I2S，然后再恢复通讯。

对于除PCM之外的所有标准来说，I2SCH标志用来区分当前传输数据所属的通道。I2SCH标志在每次RBNE标志由0变1时更新。

为了关闭I2S，不同的音频标准，数据长度和通道长度采用不同的操作步骤。每种情况的操作如*图19-53. I2S主机接收禁能流程*所示。

图 19-53. I2S 主机接收禁能流程




## I2S 从机发送流程

从机发送流程和主机发送流程相似，不同之处如下：

在从机模式下，从机需要在外部主机开始通讯之前使能。当外部主机开始发送时钟信号且I2S_WS信号请求传输数据时，发送流程开始。数据需要在外部主机发起通讯之前写入SPI_DATA寄存器。为了确保音频数据的连续传输，必须在当前发送序列结束之前将下一个待发送的数据写入SPI_DATA寄存器，否则会产生发送欠载错误。此时TXURERR标志会置1，如果SPI_CTL1寄存器的ERRIE位为1，将会产生中断。这种情况下，必须先关闭I2S再打开I2S来恢复通讯。从机模式下，I2SCH标志是根据外部主机发送的I2S_WS信号而变化的。

为关闭I2S，必须在TBE标志变为1且TRANS标志变为0之后，才能清除I2SEN位。

## I2S 从机接收流程

从机接收流程与主机接收流程类似。不同之处如下。

在从机模式下，从机需要在外部主机开始通讯之前使能。当外部主机开始发送时钟信号且

I2S_WS信号指示数据开始时，接收流程开始。从机模式下，I2SCH标志是根据外部主机发送的I2S_WS信号而变化的。

为了关闭I2S，必须在收到最后一个RBNE之后立即清除I2SEN位。

### 19.4.6. DMA 功能

DMA功能与SPI模式完全一样，唯一不同的地方就是I2S模式不支持CRC功能。

### 19.4.7. I2S 中断

#### 状态标志位

SPI_STAT寄存器中有4个可用的标志位，分别是TBE、RBNE、TRANS和I2SCH，用户通过这些标志位可以全面监视I2S总线的状态。

- 发生缓冲区空标志(TBE)：
  当发送缓冲区为空时，TBE置位。软件可以通过写SPI_DATA寄存器将下一个数据写入发送缓冲区。
- 接收缓冲区非空标志(RBNE)：
  接收缓冲区非空时，RBNE置位，表示此时接收到一个数据，并已存入接收缓冲区中，软件可以通过读SPI_DATA寄存器来读取此数据。
- I2S通信进行中标志(TRANS)：
  TRANS是用来指示当前传输是否正在进行或结束的状态标志，它由内部硬件置位和清除，无法进行软件操作。该标志位不会产生如何中断。
- I2S通道标志(I2SCH)：
  I2SCH用来表明当前传输数据的通道信息，对PCM音频标准来说没有意义。在发送模式下，I2SCH标志在每次TBE由0变1时更新，在接收模式下，I2SCH标志在每次RBNE由0变1时更新。该标志位不会产生任何中断。

#### 错误标志

有三个错误标志：

- 发送欠载错误标志(TXURERR)：
  在从发送模式下，有效的SCK信号开始发送，当发送缓冲区为空时，发送欠载错误标志TXURERR置位。
- 接收过载错误标志(RXORERR)：
  当接收缓冲区已满且又接收到一个新的数据时，接收过载错误标志RXORERR置位。当接收过载发生时，接收缓冲区中的数据没有更新，新接收的数据丢失。
- 帧格式错误(FERR)：
  在从I2S模式下，I2S模块监视I2S_WS信号，如果I2S_WS信号在一个错误的位置发生翻转，将会置位FERR帧错误标志位。

*表 19-10. I2S 中断*总结了 I2S 中断事件和相应的使能位。

**表 19-10. I2S 中断**

| 中断标志 | 描述 | 清除方式 | 中断使能位 |
|---|---|---|---|
| TBE | 发送缓冲区空 | 写 SPI_DATA 寄存器 | TBEIE |
| RBNE | 接收缓冲区非空 | 读 SPI_DATA 寄存器 | RBNEIE |
| TXURERR | 发送欠载错误 | 读 SPI_STAT 寄存器 | ERRIE |
| RXORERR | 接收过载错误 | 读 SPI_DATA 寄存器，然后再读 SPI_STAT 寄存器 | |
| FERR | I2S 帧错误 | 读 SPI_STAT 寄存器 | |

## 19.5. SPI/I2S 寄存器

SPI0基地址：0x4001 3000
SPI1/I2S1基地址：0x4000 3800
SPI2/I2S2基地址：0x4000 3C00

### 19.5.1. 控制寄存器 0 (SPI_CTL0)

地址偏移：0x00
复位值：0x0000 0000

该寄存器可以按半字(16位)或字(32位)访问。
该寄存器在I2S模式下没有意义。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BDEN | BDOEN | CRCEN | CRCNT | FF16 | RO | SWNSSEN | SWNSS | LF | SPIEN | PSC[2:0] | | | MSTMOD | CKPL | CKPH |
| 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 | BDEN | 双向数据模式使能<br>0：2线单向传输模式<br>1：1线双向传输模式。数据在主机的MOSI引脚和从机的MISO引脚之间传输。 |
| 14 | BDOEN | 双向传输输出使能<br>当BDEN置位时，该位决定了数据的传输方向。<br>0：工作在只接收模式<br>1：工作在只发送模式 |
| 13 | CRCEN | CRC计算使能<br>0：CRC计算禁止<br>1：CRC计算使能 |
| 12 | CRCNT | 下一次传输CRC<br>0：下一次传输值为数据<br>1：下一次传输值为CRC值(TCRC)<br>当数据传输由DMA管理时，CRC值由硬件传输，该位应该被清零。<br>在全双工和只发送模式下，当最后一个数据写入SPI_DATA寄存器后应将该位置1。<br>在只接收模式下，在接收完倒数第二个数据后应将该位置1。 |
| 11 | FF16 | 数据帧格式<br>0：8位数据帧格式 |

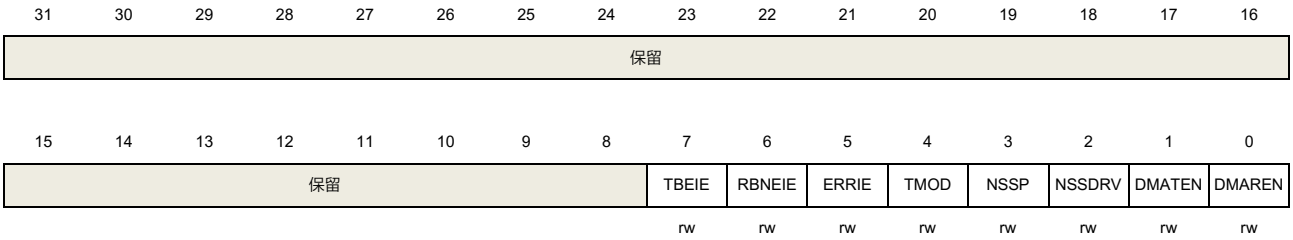
| | | |
|---|---|---|
| | | 1：16位数据帧格式 |
| 10 | RO | 只接收模式<br>当BDEN清零时，该位决定了数据的传输方向。<br>0：全双工模式<br>1：只接收模式 |
| 9 | SWNSSEN | NSS软件模式使能<br>0：NSS硬件模式，NSS电平取决于NSS引脚<br>1：NSS软件模式，NSS电平取决于SWNSS位<br>该位在SPI TI模式下没有意义。 |
| 8 | SWNSS | NSS软件模式下NSS引脚选择<br>0：NSS引脚拉低<br>1：NSS引脚拉高<br>只有在SWNSSEN置位时，该位有效。<br>该位在SPI TI模式下没有意义。 |
| 7 | LF | 最低有效位先发模式<br>0：先发送最高有效位<br>1：先发送最低有效位<br>该位在SPI TI模式下没有意义。 |
| 6 | SPIEN | SPI使能<br>0：SPI设备禁止<br>1：SPI设备使能 |
| 5:3 | PSC[2:0] | 主时钟预分频选择<br>000：PCLK/2<br>001：PCLK/4<br>010：PCLK/8<br>011：PCLK/16<br>100：PCLK/32<br>101：PCLK/64<br>110：PCLK/128<br>111：PCLK/256<br>当使用SPI0时，PCLK=PCLK2，当使用SPI1和SPI2时，PCLK=PCLK1。 |
| 2 | MSTMOD | 主从模式使能<br>0：从机模式<br>1：主机模式 |
| 1 | CKPL | 时钟极性选择<br>0：SPI为空闲状态时，CLK引脚拉低<br>1：SPI为空闲状态时，CLK引脚拉高 |
| 0 | CKPH | 时钟相位选择<br>0：在第一个时钟跳变沿采集第一个数据 |

1：在第二个时钟跳变沿时钟跳变沿采集第一个数据

### 19.5.2. 控制寄存器 1 (SPI_CTL1)

地址偏移：0x04
复位值：0x0000 0000

该寄存器可以按半字(16位)或字(32位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 必须保持复位值。 |
| 7 | TBEIE | 发送缓冲区空中断使能<br>0：TBE中断禁止<br>1：TBE中断使能。当TBE置位时，产生中断。 |
| 6 | RBNEIE | 接收缓冲区非空中断使能<br>0：RBNE中断禁止.<br>1：RBNE中断使能。当RBNE置位时，产生中断。 |
| 5 | ERRIE | 错误中断使能<br>0：错误中断禁止<br>1：错误中断使能。当CRCERR位，CONFERR位，RXORERR位或者TXURERR位置1时，产生中断。 |
| 4 | TMOD | SPI TI模式使能<br>0：SPI TI模式禁止<br>1：SPI TI模式使能 |
| 3 | NSSP | SPI NSS脉冲模式使能<br>0：SPI NSS脉冲模式禁止<br>1：SPI NSS脉冲模式使能 |
| 2 | NSSDRV | NSS输出使能<br>0：NSS输出禁止<br>1：NSS输出使能。<br>当SPI使能时，如果NSS引脚配置为输出模式，NSS引脚在主模式时被拉低。如果NSS引脚配置为输入模式，NSS引脚在主模式时被拉高，此时该位无效。 |
| 1 | DMATEN | 发送缓冲区DMA使能<br>0：发送缓冲区DMA禁止 |

| | | |
|---|---|---|
| | | 1：发送缓冲区DMA使能。当SPI_STAT中的TBE置位时，将会在相应的DMA通道上产生一个DMA请求。 |
| 0 | DMAREN | 接收缓冲区DMA使能<br>0：接收缓冲区DMA禁止<br>1：接收缓冲区DMA使能。当SPI_STAT中的RBNE置位时，将会在相应的DMA通道上产生一个DMA请求。 |

### 19.5.3. 状态寄存器 (SPI_STAT)

地址偏移：0x08
复位值：0x0000 0002

该寄存器可以按半字(16位)或字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | FERR | TRANS | RXORERR | CONFERR | CRCERR | TXURERR | I2SCH | TBE | RBNE |
| | | | | | | | rc_w0 | r | r | r | rc_w0 | r | r | r | r |

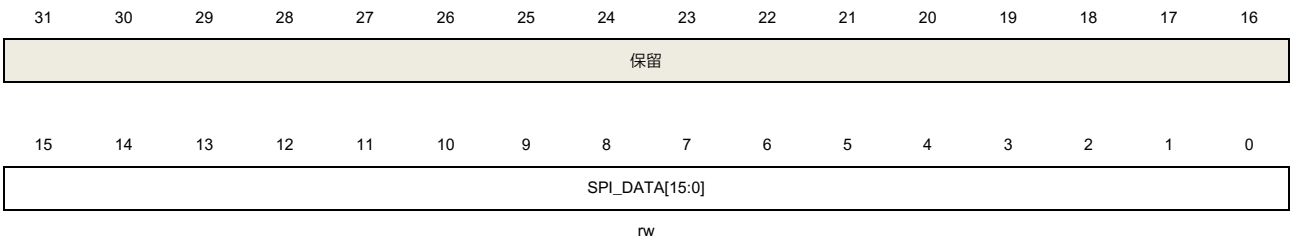
| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:9 | 保留 | 必须保持复位值。 |
| 8 | FERR | 帧错误<br>SPI TI模式：<br>0：没有TI模式帧错误发生<br>1：TI模式帧错误发生<br>I2S模式：<br>0：没有I2S帧错误发生<br>1：I2S帧错误发生<br>该位由硬件置位，可以通过写0清除。 |
| 7 | TRANS | 通信进行中标志<br>0：SPI或I2S空闲<br>1：SPI或I2S当前正在发送且/或接收数据<br>该位由硬件置位和清除。 |
| 6 | RXORERR | 接收过载错误标志<br>0：没有接收过载错误发生<br>1：接收过载错误发生<br>该位由硬件置位，软件序列清零。软件序列为：先读SPI_DATA寄存器，然后读SPI_STAT寄存器。 |
| 5 | CONFERR | SPI配置错误<br>0：无配置错误发生 |

| | | |
|---|---|---|
| | | 1：配置错误发生(主机模式下，在硬件NSS模式时NSS引脚被拉低，或者软件NSS模式时SWNSS位为0，都会产生CONFERR错误)<br>该位由硬件置位，软件序列清零。软件序列为：先读或写SPI_STAT寄存器，然后写SPI_CTL0寄存器。<br>I2S模式下不使用该位。 |
| 4 | CRCERR | SPI CRC错误标志<br>0：SPI_RCRC值等于最后接收到的CRC值<br>1：SPI_RCRC值不等于最后接收到的CRC值该位由硬件置位，可以通过写0清除。<br>I2S模式下不使用该位。 |
| 3 | TXURERR | 发送欠载错误标志<br>0：无发送欠载错误发生<br>1：发送欠载错误发生<br>该位由硬件置位，通过写SPI_STAT寄存器清除。<br>SPI模式下不使用该位。 |
| 2 | I2SCH | I2S通道标志<br>0：下一个将要发送或接收的数据属于左通道<br>1：下一个要发送或接收的数据属于右通道<br>该位由硬件置位和清除。<br>SPI模式下该位无用，I2S PCM模式下该位没有意义。 |
| 1 | TBE | 发送缓冲区空<br>0：发送缓冲区非空<br>1：发送缓冲区空 |
| 0 | RBNE | 接收缓冲区非空<br>0：接收缓冲区空<br>1：接收缓冲区非空 |

### 19.5.4. 数据寄存器（SPI_DATA)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器可以按半字(16位)或字(32位)访问。




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |

| | | |
|---|---|---|
| 15:0 | SPI_DATA[15:0] | 数据传输寄存器值<br>硬件有两个缓冲区：发送缓冲区和接收缓冲区。向SPI_DATA写数据将会把数据存入发送缓冲区，从SPI_DATA读数据，将从接收缓冲区获得数据。<br>当数据帧格式为8位时，SPI_DATA[15:8]强制为0，SPI_DATA[7:0]用来发送和接收数据，发送和接收缓冲区都是8位。如果数据帧格式为16位，SPI_DATA[15:0]用于发送和接收数据，发送和接收缓冲区也是16位。 |

### 19.5.5. CRC 多项式寄存器 (SPI_CRCPOLY)

地址偏移：0x10
复位值：0x0000 0007

该寄存器可以按半字(16位)或字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CRCPOLY[15:0] | | | | | | | | | | | | | | | |
| 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多项式寄存器值<br>该值包含了CRC多项式，用于CRC计算，默认值为0007h。 |

### 19.5.6. 接收 CRC 寄存器 (SPI_RCRC)

地址偏移：0x14
复位值：0x0000 0000

该寄存器可以按半字(16位)或字(32位)访问。

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15:0 | RCRC[15:0] | 接收CRC寄存器值<br>当SPI_CTL0中的CRCEN置位时，硬件计算接收数据的CRC值，并保存到RCRC寄存 |

器中。如果是8位数据帧格式，CRC计算基于CRC8标准进行，保存数据到RCRC[7:0]。如果是16位数据帧格式，CRC计算基于CRC16标准进行，保存数据到RCRC[15:0]。硬件在接收到每个数据位后都会计算CRC值，当TRANS置位时，读该寄存器将返回一个中间值。

当SPI_CTL0寄存器中的CRCEN位或RCU复位寄存器中的SPIxRST位置位时，该寄存器复位。

### 19.5.7. 发送 CRC 寄存器 (SPI_TCRC)

地址偏移：0x18

复位值：0x0000 0000

该寄存器可以按半字(16位)或字(32位)访问。

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15:0 | TCRC[15:0] | 发送CRC寄存器值<br>当SPI_CTL0中的CRCEN置位时，硬件计算发送数据的CRC值，并保存到TCRC寄存器中。如果是8位数据帧格式，CRC计算基于CRC8标准进行，保存数据到TCRC[7:0]。如果是16位数据帧格式，CRC计算基于CRC16标准进行，保存数据到TCRC[15:0]。硬件在发送出每个数据位后都会计算CRC值，当TRANS置位时，读该寄存器将返回一个中间值。不同的数据帧格式(SPI_CTL0中的LF位决定)将会得到不同的CRC值。<br>当SPI_CTL0寄存器中的CRCEN位或RCU复位寄存器中的SPIxRST位置位时，该寄存器复位。 |

### 19.5.8. I2S 控制寄存器 (SPI_I2SCTL)

地址偏移：0x1C

复位值：0x0000 0000

该寄存器可以按半字(16位)或字(32位)访问。

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | I2SSEL | I2SEN | I2SOPMOD[1:0] | | PCMSMOD | 保留 | I2SSTD[1:0] | | CKPL | DTLEN[1:0] | | CHLEN |
| | | | | rw | rw | rw | | rw | | rw | | rw | rw | | rw |

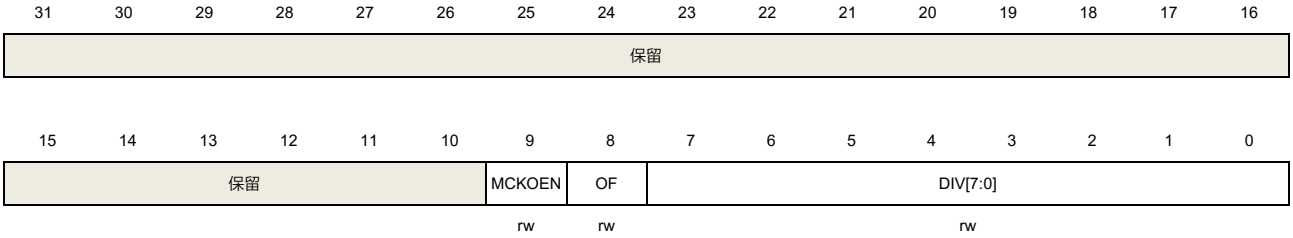
| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:12 | 保留 | 必须保持复位值。 |
| 11 | I2SSEL | I2S模式选择<br>0：SPI模式<br>1：I2S模式<br>当SPI或I2S关闭时配置该位。 |
| 10 | I2SEN | I2S使能<br>0：I2S禁止<br>1：I2S使能<br>SPI模式不使用该位。 |
| 9:8 | I2SOPMOD[1:0] | I2S运行模式<br>00：从机发送模式<br>01：从机接收模式<br>10：主机发送模式<br>11：主机接收模式<br>当I2S关闭时配置该位。SPI模式不使用该位。 |
| 7 | PCMSMOD | PCM帧同步模式<br>0：短帧同步<br>1：长帧同步<br>只有在PCM标准下，该位才有意义。<br>当I2S关闭时配置该位。SPI模式不使用该位。 |
| 6 | 保留 | 必须保持复位值。 |
| 5:4 | I2SSTD[1:0] | I2S标准选择<br>00：I2S飞利浦标准<br>01：MSB对齐标准<br>10：LSB对齐标准<br>11：PCM标准<br>当I2S关闭时配置该位。SPI模式不使用该位。 |
| 3 | CKPL | 空闲状态时钟极性<br>0：I2S_CK空闲状态为低电平<br>1：I2S_CK空闲状态为高电平<br>当I2S关闭时配置该位。SPI模式不使用该位。 |
| 2:1 | DTLEN[1:0] | 数据长度<br>00：16位<br>01：24位<br>10：32位<br>11：保留<br>当I2S关闭时配置该位。SPI模式不使用该位。 |
| 0 | CHLEN | 通道长度 |

0：16位
1：32位
通道长度必须大于或等于数据长度。
当I2S关闭时配置该位。SPI模式不使用该位。

### 19.5.9. I2S 时钟预分频寄存器 (SPI_I2SPSC)

地址偏移：0x20
复位值：0x0000 0002

该寄存器可以按半字(16位)或字(32位)访问。

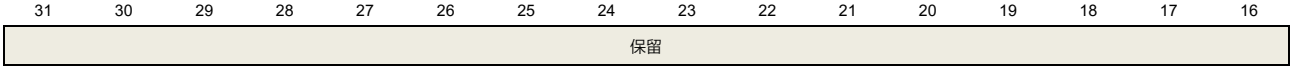



| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:10 | 保留 | 必须保持复位值。 |
| 9 | MCKOEN | I2S_MCK输出使能<br>0：I2S_MCK输出禁止<br>1：I2S_MCK输出使能<br>当I2S关闭时配置该位。<br>SPI模式不使用该位。 |
| 8 | OF | 预分频器的奇系数<br>0：实际分频系数为DIV * 2<br>1：实际分频系数为DIV * 2 + 1<br>当I2S关闭时配置该位。SPI模式下不使用该位。 |
| 7:0 | DIV[7:0] | 预分频器的分频系数<br>实际分频系数是DIV * 2 + OF。<br>DIV不能为0。<br>当I2S关闭时配置该位。SPI模式下不使用该位。 |

### 19.5.10. SPI0 四线 SPI 控制寄存器 (SPI_QCTL)

地址偏移：0x80
复位值：0x0000 0000

该寄存器可以按半字(16位)或字(32位)访问。



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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:2 | 保留 | 必须保持复位值。 |
| 1 | QRD | 四线SPI模式读选择<br>0：SPI四线模式写操作<br>1：SPI四线模式读操作<br>该位仅能在SPI未通信时配置(TRANS位清零)。<br>该位仅适用于SPI0。 |
| 0 | QMOD | 四线SPI模式使能<br>0：SPI工作在非四线模式<br>1：SPI工作在四线模式<br>该位仅能在SPI未通信时配置(TRANS位清零)。<br>该位仅适用于SPI0。 |

# 20. 外部存储器控制器(EXMC)

## 20.1. 简介

外部存储器控制器EXMC，用来访问各种片外存储器。通过配置寄存器，EXMC可以把AMBA协议转换为专用的片外存储器通信协议，包括SRAM，PSRAM，ROM和NOR Flash。用户还可以调整配置寄存器中的时间参数来提高通信效率。

## 20.2. 主要特征

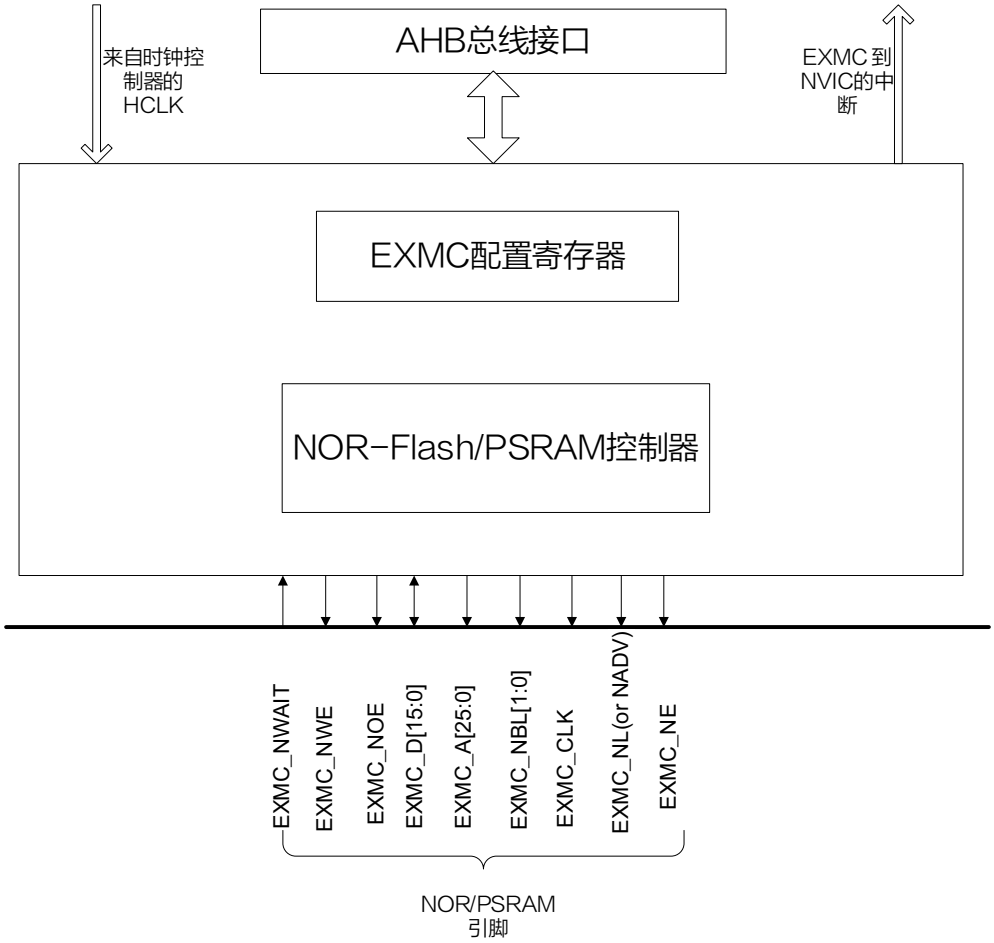
- 支持片外存储器类型：
  - SRAM
  - PSRAM
  - ROM
  - NOR Flash
- AMBA协议与各种片外存储器协议转换；
- 时序参数可编程可以满足用户特定需求；
- 对于部分存储器类型支持独立的读写时序；
- 支持8位，或16位总线带宽；
- NOR Flash和PSRAM支持地址总线和数据总线的复用；
- 提供写使能和字节选择信号；
- 当AMBA总线宽度与外部存储器数据宽度不同时，会自动分割操作。

## 20.3. 功能说明

### 20.3.1. 结构框图

EXMC由4个模块组成：AHB总线接口，EXMC配置寄存器，NOR/PSRAM控制器和外部设备接口。AHB时钟(HCLK)是参考时钟。

**图 20-1. EXMC 结构框图**

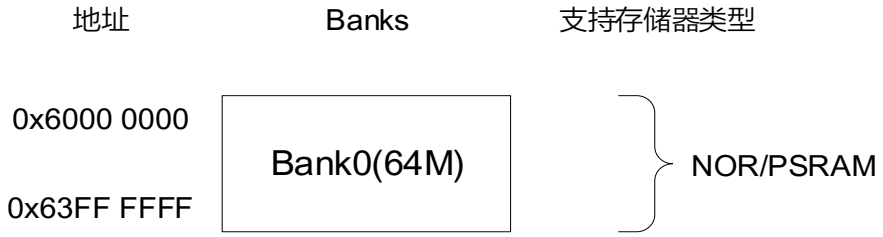



## 20.3.2. EXMC 访问基本规范

EXMC是AHB总线至外部设备协议的转换接口。32位的AHB读写操作可以转化为几个连续的8位或16位读写操作。在数据传输的过程中，AHB数据宽度和存储器数据宽度可能不相同。为了保证数据传输的一致性，EXMC读写访问需要遵从以下规范。

- AHB访问宽度等于存储器宽度，则没有问题。
- AHB访问宽度大于存储器宽度，则自动将AHB访问分割成几个连续的存储器数据宽度的传输。
- AHB访问宽度小于存储器宽度，如果外部存储设备具有字节选择功能，如SRAM、ROM、PSRAM，则可通过它的字节通道EXMC_NBL[1:0]来访问对应的字节。否则禁止写操作，只允许读操作。

**图 20-2. EXMC Bank 划分**




EXMC访问区域为Bank0，占64M字节，用于访问NOR、PSRAM设备。

由于HADDR[25:0]是字节地址，而外部存储器访问有可能不是按字节访问的，所以会出现地址不一致的情况，但EXMC能实现对HADDR的调整以适应外部存储器的数据宽度。具体规则如下：

- 如果外部存储器的数据宽度是8位按字节对齐，HADDR[25:0]与EXMC_A[25:0]相连，然后用EXMC_A[25:0]去连接外部存储器的地址线；
- 如果外部存储器的数据宽度是16位按半字对齐，就需要将HADDR的字节地址转化为半字地址之后再连接外存储器，所以需要将HADDR[25:1]与EXMC_A[24:0]相连。然后用EXMC_A[24:0]去连接外部存储器的地址线。

### 20.3.3. NOR/PSRAM 控制器

控制器控制Bank0，它可以支持NOR Flash、PSRAM、SRAM、ROM和蜂窝RAM外部存储器。

**注意：**

在异步模式下，所有控制器输出信号在内部AHB总线时钟(HCLK)的上升沿改变。

在同步模式下，所有控制器输出数据在外部存储器时钟(EXMC_CLK)的下降沿改变。

#### NOR/PSRAM 接口描述

**表 20-1. NOR Flash 接口信号描述**

| EXMC 引脚 | 传输方向 | 模式 | 功能描述 |
|---|---|---|---|
| EXMC_CLK | 输出 | 同步 | 同步时钟信号 |
| 非复用 EXMC_A[25:0] | 输出 | 异步/同步 | 地址总线信号 |
| 复用 EXMC_A[25:16] | | | |
| EXMC_D[15:0] | 输入/输出 | 异步/同步 (复用) | 地址/数据总线 |
| | 输入/输出 | 异步/同步 (非复用) | 数据总线 |
| EXMC_NE | 输出 | 异步/同步 | 片选 |
| EXMC_NOE | 输出 | 异步/同步 | 输出使能(读使能) |
| EXMC_NWE | 输出 | 异步/同步 | 写使能 |
| EXMC_NWAIT | 输入 | 异步/同步 | 等待输入信号 |

| EXMC 引脚 | 传输方向 | 模式 | 功能描述 |
|---|---|---|---|
| EXMC_NL(NADV) | 输出 | 异步/同步 | 地址有效 |

**表 20-2. PSRAM 非复用接口信号描述**

| EXMC 引脚 | 传输方向 | 模式 | 功能描述 |
|---|---|---|---|
| EXMC_CLK | 输出 | 同步 | 同步时钟信号 |
| EXMC_A[25:0] | 输出 | 异步/同步 | 地址总线 |
| EXMC_D[15:0] | 输入/输出 | 异步/同步 | 数据总线 |
| EXMC_NE | 输出 | 异步/同步 | 片选 |
| EXMC_NOE | 输出 | 异步/同步 | 输出使能(读使能) |
| EXMC_NWE | 输出 | 异步/同步 | 写使能 |
| EXMC_NWAIT | 输入 | 异步/同步 | 等待输入信号 |
| EXMC_NL(NADV) | 输出 | 异步/同步 | 锁存使能(地址有效使能，NADV) |
| EXMC_NBL[1] | 输出 | 异步/同步 | 高字节使能 |
| EXMC_NBL[0] | 输出 | 异步/同步 | 低字节使能 |

## 支持的存储器访问模式

下表列出了当NOR，PSRAM和SRAM存储器数据总线为16位时，所支持的设备类型、访问模式和传输的示例。

**表 20-3. EXMC 的 Bank0 支持的所有传输**

| 存储器类型 | 访问模式 | 读/写 | AHB 传输宽度 | 存储器传输宽度 | 注释 |
|---|---|---|---|---|---|
| NOR Flash | 异步 | R | 8 | 16 | |
| | 异步 | R | 16 | 16 | |
| | 异步 | W | 16 | 16 | |
| | 异步 | R | 32 | 16 | 分成 2 次 EXMC 访问 |
| | 异步 | W | 32 | 16 | 分成 2 次 EXMC 访问 |
| | 同步 | R | 16 | 16 | |
| | 同步 | R | 32 | 16 | |
| PSRAM | 异步 | R | 8 | 16 | |
| | 异步 | W | 8 | 16 | 使用字节信号 EXMC_NBL[1:0] |
| | 异步 | R | 16 | 16 | |
| | 异步 | W | 16 | 16 | |
| | 异步 | R | 32 | 16 | 分成 2 次 EXMC 访问 |
| | 异步 | W | 32 | 16 | 分成 2 次 EXMC 访问 |
| | 同步 | R | 16 | 16 | |
| | 同步 | R | 32 | 16 | |
| | 同步 | W | 8 | 16 | 使用字节信号 EXMC_NBL[1:0] |
| | 同步 | W | 16 | 16 | |

| 存储器类型 | 访问模式 | 读/写 | AHB 传输宽度 | 存储器传输宽度 | 注释 |
|---|---|---|---|---|---|
| | 同步 | W | 32 | 16 | |
| SRAM 和 ROM | 异步 | R | 8 | 8 | |
| | 异步 | R | 8 | 16 | |
| | 异步 | R | 16 | 8 | 分成 2 次 EXMC 访问 |
| | 异步 | R | 16 | 16 | |
| | 异步 | R | 32 | 8 | 分成 4 次 EXMC 访问 |
| | 异步 | R | 32 | 16 | 分成 2 次 EXMC 访问 |
| | 异步 | W | 8 | 8 | |
| | 异步 | W | 8 | 16 | 使用字节信号 EXMC_NBL[1:0] |
| | 异步 | W | 16 | 8 | |
| | 异步 | W | 16 | 16 | |
| | 异步 | W | 32 | 8 | |
| | 异步 | W | 32 | 16 | |

## NOR Flash/PSRAM 控制时序

EXMC为SRAM、ROM、PSRAM、NOR Flash等外部静态存储器提供可编程的时序参数以及多种时序模型以满足不同的需求。

**表 20-4. NOR/PSRAM 控制时序参数**

| 参数 | 功能 | 访问模式 | 单位 | 最小值 | 最大值 |
|---|---|---|---|---|---|
| CKDIV | 同步时钟分频比 | 同步 | HCLK | 2 | 16 |
| DLAT | 数据延迟 | 异步 | EXMC_CLK | 2 | 17 |
| BUSLAT | 总线延迟 | 异步/同步读 | HCLK | 1 | 16 |
| DSET | 数据建立时间 | 异步 | HCLK | 2 | 256 |
| AHLD | 地址保持时间 | 异步(复用) | HCLK | 2 | 16 |
| ASET | 地址建立时间 | 异步 | HCLK | 1 | 16 |

**表 20-5. EXMC 时序模型**

| 时序模型 | | 扩展模式 | 模式描述 | 写时序参数 | 读时序参数 |
|---|---|---|---|---|---|
| 异步 | 模式 1 | 0 | SRAM/PSRAM/CRAM | DSET<br>ASET | DSET<br>ASET |
| | 模式 2 | 0 | NOR Flash | DSET<br>ASET | DSET<br>ASET |
| | 模式 A | 1 | SRAM/PSRAM/CRAM 在数据阶段时 EXMC_NOE 翻转 | WDSET<br>WASET | DSET<br>ASET |
| | 模式 B | 1 | NOR Flash | WDSET<br>WASET | DSET<br>ASET |
| | 模式 C | 1 | NOR Flash 在数据阶段时 EXMC_NOE 翻转 | WDSET<br>WASET | DSET<br>ASET |
| | 模式 D | 1 | 有地址保持功能 | WDSET | DSET |

| 时序模型 | | 扩展模式 | 模式描述 | 写时序参数 | 读时序参数 |
|---|---|---|---|---|---|
| | | | | WAHLD<br>WASET | AHLD<br>ASET |
| | 模式 AM | 0 | NOR Flash 数据/地址复用 | DSET<br>AHLD<br>ASET<br>BUSLAT | DSET<br>AHLD<br>ASET<br>BUSLAT |
| 同步 | 模式 E | 0 | NOR/PSRAM/CRAM 同步读，<br>PSRAM/CRAM 同步写 | DLAT<br>CKDIV | DLAT<br>CKDIV |
| | 模式 SM | 0 | NOR Flash 数据/地址复用 | DLAT<br>CKDIV | DLAT<br>CKDIV |

如*表20-5. EXMC时序模型*所示，EXMC模块NOR Flash/PSRAM控制器可以提供多种时序模型。用户可以通过修改*表20-4. NOR/PSRAM控制时序参数*中列出的参数来使之适合不同类型外部存储器的时序以及满足用户的要求。当将寄存器EXMC_SNCTL位EXMODEN置1使能扩展模式后，可以通过寄存器EXMC_SNTCFG和EXMC_SNWTCFG将读写配置成独立的时序。

### 异步访问时序

模式1 - SRAM/CRAM

**图 20-3. 模式 1 读访问**

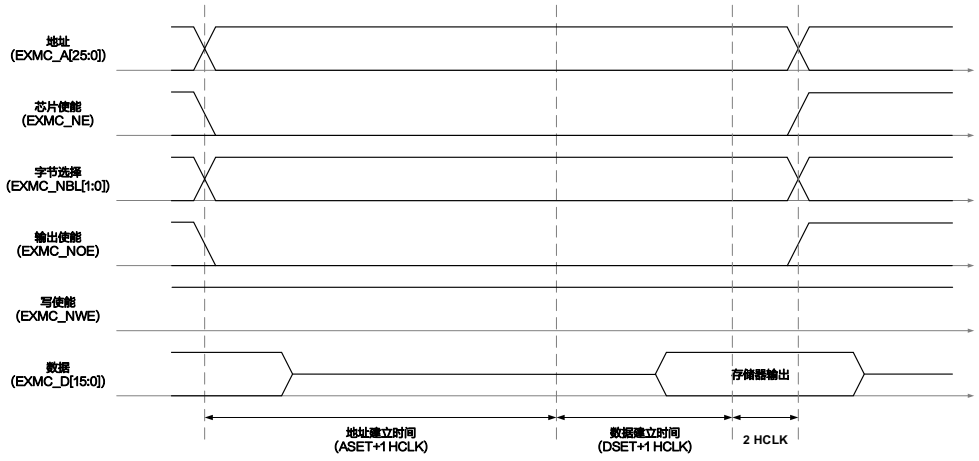


图 20-4. 模式 1 写访问

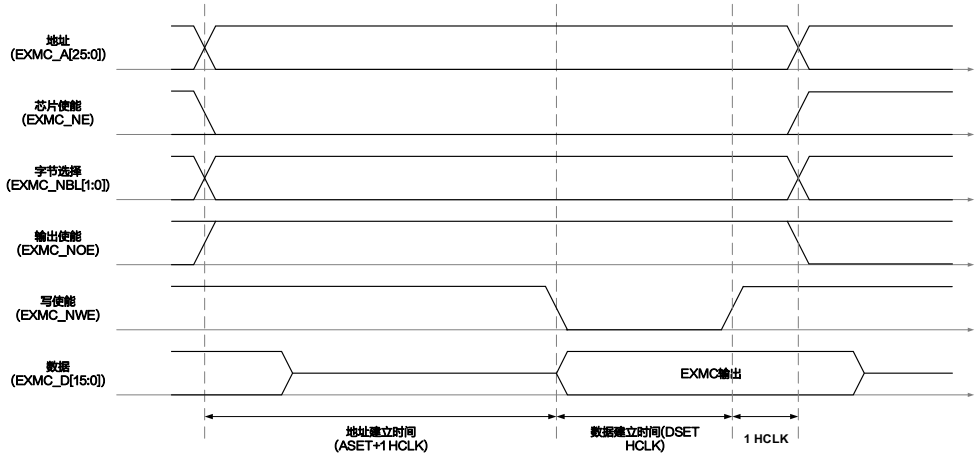



表 20-6. 模式 1 相关寄存器配置

| 位/位域 | 位名 | 参考设定值 |
|---|---|---|
| **EXMC_SNCTL** | | |
| 31-20 | 保留 | 0x000 |
| 19 | SYNCWR | 0x0 |
| 18-16 | CPS | 0x0 |
| 15 | ASYNCWAIT | 取决于存储器 |
| 14 | EXMODEN | 0x0 |
| 13 | NRWTEN | 0x0 |
| 12 | WREN | 取决于用户 |
| 11 | NRWTCFG | 无影响 |
| 10 | WRAPEN | 0x0 |
| 9 | NRWTPOL | 仅当位 15 为 1 时有效 |
| 8 | SBRSTEN | 0x0 |
| 7 | 保留 | 0x1 |
| 6 | NREN | 无影响 |
| 5-4 | NRW | 取决于存储器 |
| 3-2 | NRTP | 取决于存储器，除了 0x2(NOR Flash) |
| 1 | NRMUX | 0x0 |
| 0 | NRBKEN | 0x1 |
| **EXMC_SNTCFG** | | |
| 31-30 | 保留 | 0x0 |
| 29-28 | ASYNCMOD | 无影响 |
| 27-24 | DLAT | 无影响 |
| 23-20 | CKDIV | 无影响 |
| 19-16 | BUSLAT | 从 EXMC_NE 上升沿到下降沿的时间 |
| 15-8 | DSET | 取决于存储器与用户(写操作为 DSET+1 HCLK 时钟周期，读操作为 DSET+3 HCLK 时钟周期) |
| 7-4 | AHLD | 无影响 |
| 3-0 | ASET | 取决于存储器与用户 |

模式A - SRAM/PSRAM(CRAM) OE翻转

**图 20-5. 模式 A 读访问**

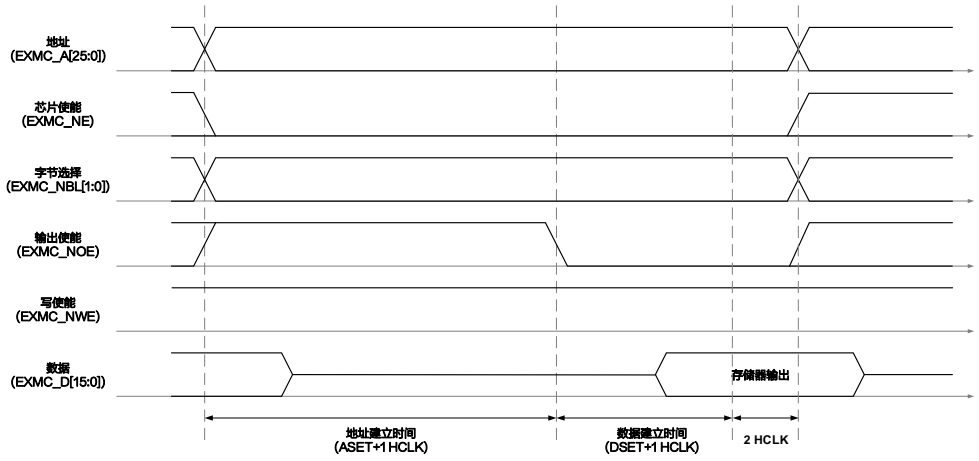
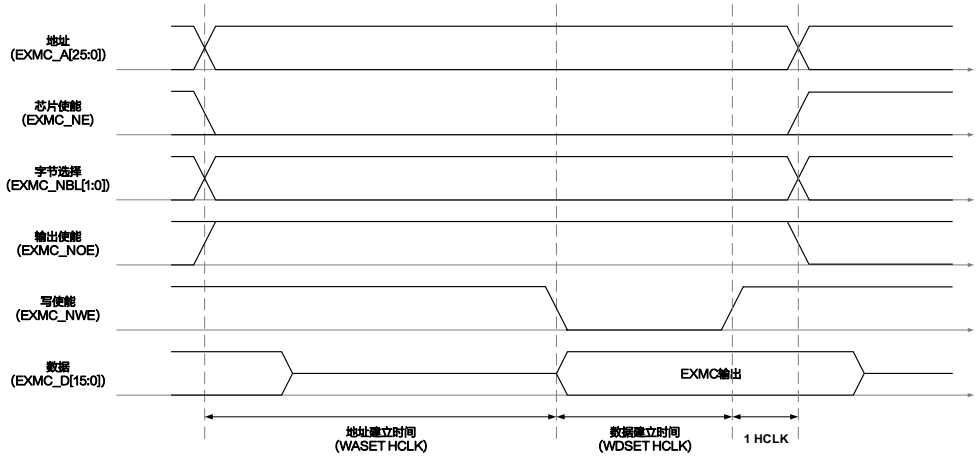



**图 20-6. 模式 A 写访问**




模式A和模式1写时序的区别在于，当读和写访问具有相同的时序配置时，模式A的写时序是独立于读时序的。

**表 20-7. 模式 A 相关寄存器配置**

| 位/位域 | 位名 | 参考设定值 |
|---|---|---|
| **EXMC_SNCTL** | | |
| 31-20 | 保留 | 0x000 |
| 19 | SYNCWR | 0x0 |
| 18-16 | CPS | 0x0 |
| 15 | ASYNCWTEN | 取决于存储器 |
| 14 | EXMODEN | 0x1 |
| 13 | NRWTEN | 0x0 |
| 12 | WREN | 取决于用户 |
| 11 | NRWTCFG | 无影响 |
| 10 | WRAPEN | 0x0 |
| 9 | NRWTPOL | 仅当位 15 为 1 时有效 |

| 位/位域 | 位名 | 参考设定值 |
|---|---|---|
| 8 | SBRSTEN | 0x0 |
| 7 | 保留 | 0x1 |
| 6 | NREN | 无影响 |
| 5-4 | NRW | 取决于存储器 |
| 3-2 | NRTP | 取决于存储器，除了 0x2(NOR Flash) |
| 1 | NRMUX | 0x0 |
| 0 | NRBKEN | 0x1 |
| **EXMC_SNTCFG(读)** | | |
| 31-30 | 保留 | 0x0 |
| 29-28 | ASYNCMOD | 0x0 |
| 27-24 | DLAT | 无影响 |
| 23-20 | CKDIV | 无影响 |
| 19-16 | BUSLAT | 从 EXMC_NE 上升沿到下降沿的时间 |
| 15-8 | DSET | 取决于存储器与用户(读操作为 DSET+3 HCLK 时钟周期) |
| 7-4 | AHLD | 无影响 |
| 3-0 | ASET | 取决于存储器与用户 |
| **EXMC_SNWTCFG(写)** | | |
| 31-30 | 保留 | 0x0 |
| 29-28 | WASYNCMOD | 0x0 |
| 27-20 | 保留 | 0xFF |
| 19-16 | WBUSLAT | 从 EXMC_NE 上升沿到下降沿的时间 |
| 15-8 | WDSET | 取决于存储器与用户(写操作为 WDSET+1 HCLK 时钟周期) |
| 7-4 | WAHLD | 0x0 |
| 3-0 | WASET | 取决于存储器与用户 |

模式2/B - NOR Flash

**图 20-7. 模式 2/B 读访问**

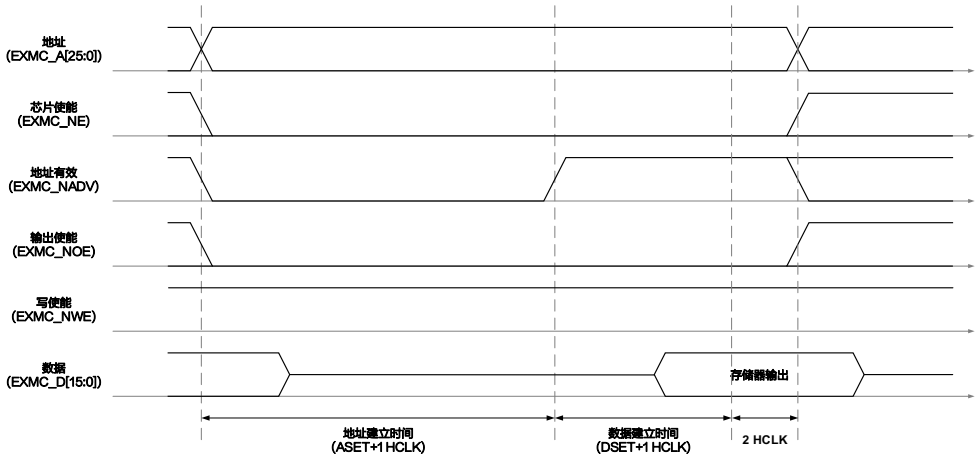


图 20-8. 模式 2 写访问

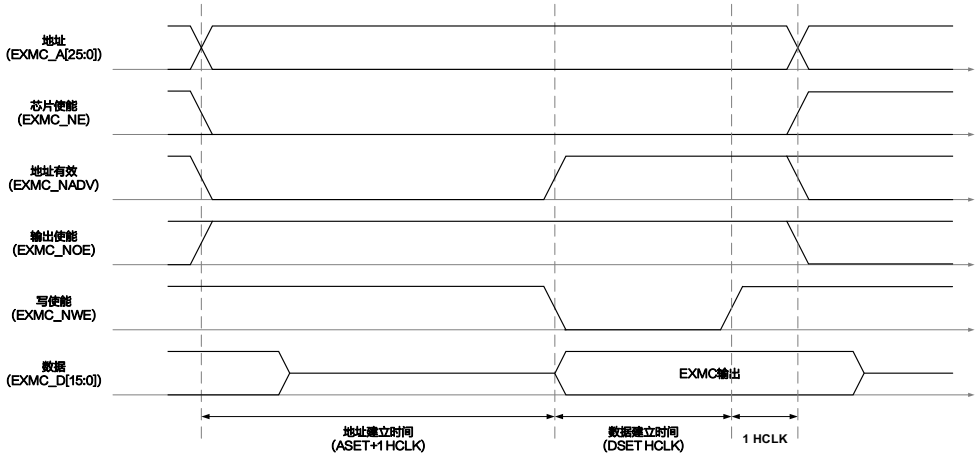



图 20-9. 模式 B 写访问

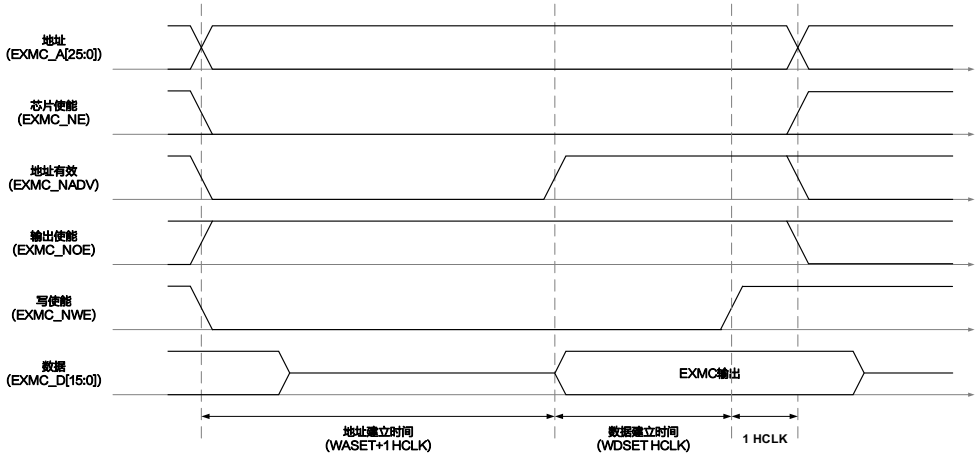



表 20-8. 模式 2/B 相关寄存器配置

| 位/位域 | 位名 | 参考设定值 |
|---|---|---|
| EXMC_SNCTL(模式 2，模式 B) | | |
| 31-20 | 保留 | 0x000 |
| 19 | SYNCWR | 0x0 |
| 18-16 | CPS | 0x0 |
| 15 | ASYNCWTEN | 取决于存储器 |
| 14 | EXMODEN | 模式 2：0x0，模式 B：0x1 |
| 13 | NRWTEN | 0x0 |
| 12 | WREN | 取决于用户 |
| 11 | NRWTCFG | 无影响 |
| 10 | WRAPEN | 0x0 |
| 9 | NRWTPOL | 仅当位 15 为 1 时有效 |
| 8 | SBRSTEN | 0x0 |
| 7 | 保留 | 0x1 |
| 6 | NREN | 0x1 |
| 5-4 | NRW | 取决于存储器 |

| 位/位域 | 位名 | 参考设定值 |
|---|---|---|
| 3-2 | NRTP | NOR Flash：0x2 |
| 1 | NRMUX | 0x0 |
| 0 | NRBKEN | 0x1 |
| **EXMC_SNTCFG(模式 2 读/写操作，模式 B 读操作)** | | |
| 31-30 | 保留 | 0x0 |
| 29-28 | ASYNCMOD | 模式 B：0x1 |
| 27-24 | DLAT | 无影响 |
| 23-20 | CKDIV | 无影响 |
| 19-16 | BUSLAT | 从 EXMC_NE 上升沿到下降沿的时间 |
| 15-8 | DSET | 取决于存储器与用户(读操作为 DSET+3 HCLK 时钟周期) |
| 7-4 | AHLD | 0x0 |
| 3-0 | ASET | 取决于存储器与用户 |
| **EXMC_SNWTCFG(模式 B 写操作)** | | |
| 31-30 | 保留 | 0x0 |
| 29-28 | WASYNCMOD | 模式 B：0x1 |
| 27-20 | 保留 | 0xFF |
| 19-16 | WBUSLAT | 从 EXMC_NE 上升沿到下降沿的时间 |
| 15-8 | WDSET | 取决于存储器与用户(写操作为 WDSET+1 HCLK 时钟周期) |
| 7-4 | WAHLD | 0x0 |
| 3-0 | WASET | 取决于存储器与用户 |

模式C - NOR Flash OE翻转

**图 20-10. 模式 C 读访问**

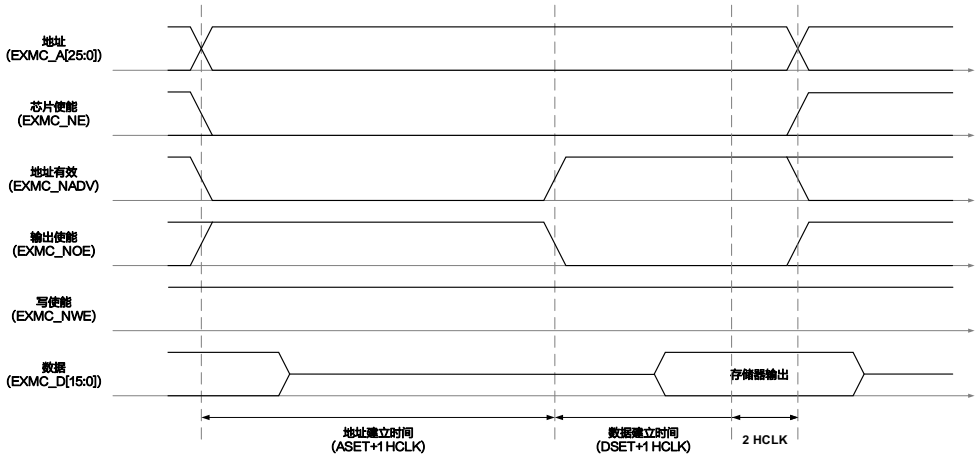
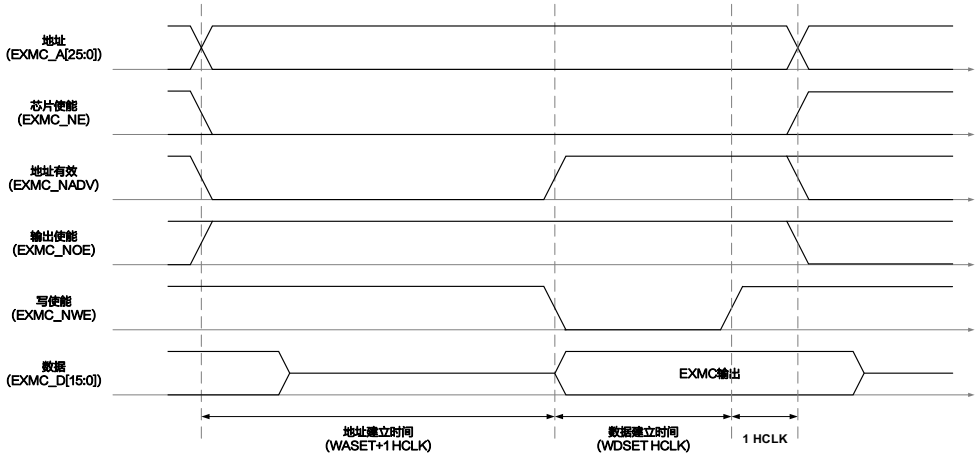


图 20-11. 模式 C 写访问




模式C和模式1写时序的区别在于，当读和写访问具有相同的时序配置时，模式C的写时序是独立于读时序的。

表 20-9. 模式 C 相关寄存器配置

| 位/位域 | 位名 | 参考设定值 |
|---|---|---|
| **EXMC_SNCTL** | | |
| 31-20 | 保留 | 0x000 |
| 19 | SYNCWR | 0x0 |
| 18-16 | CPS | 0x0 |
| 15 | ASYNCWTEN | 取决于存储器 |
| 14 | EXMODEN | 0x1 |
| 13 | NRWTEN | 0x0 |
| 12 | WREN | 取决于用户 |
| 11 | NRWTCFG | 无影响 |
| 10 | WRAPEN | 0x0 |
| 9 | NRWTPOL | 仅当位 15 为 1 时有效 |
| 8 | SBRSTEN | 0x0 |
| 7 | 保留 | 0x1 |
| 6 | NREN | 0x1 |
| 5-4 | NRW | 取决于存储器 |
| 3-2 | NRTP | NOR Flash：0x2 |
| 1 | NRMUX | 0x0 |
| 0 | NRBKEN | 0x1 |
| **EXMC_SNTCFG** | | |
| 31-30 | 保留 | 0x0 |
| 29-28 | ASYNCMOD | 模式 C：0x2 |
| 27-24 | DLAT | 0x0 |
| 23-20 | CKDIV | 0x0 |
| 19-16 | BUSLAT | 从 EXMC_NE 上升沿到下降沿的时间 |
| 15-8 | DSET | 取决于存储器与用户(读操作为 DSET+3 HCLK 时钟周 |

| 位/位域 | 位名 | 参考设定值 |
|---|---|---|
| | | 期) |
| 7-4 | AHLD | 0x0 |
| 3-0 | ASET | 取决于存储器与用户 |
| **EXMC_SNWTCFG** | | |
| 31-30 | 保留 | 0x0 |
| 29-28 | WASYNCMOD | 模式 C：0x2 |
| 27-20 | 保留 | 0xFF |
| 19-16 | WBUSLAT | 从 EXMC_NE 上升沿到下降沿的时间 |
| 15-8 | WDSET | 取决于存储器与用户(写操作为 WDSET+1 HCLK 时钟周期) |
| 7-4 | WAHLD | 0x0 |
| 3-0 | WASET | 取决于存储器与用户 |

模式D -带地址扩展的异步操作

**图 20-12. 模式 D 读访问**

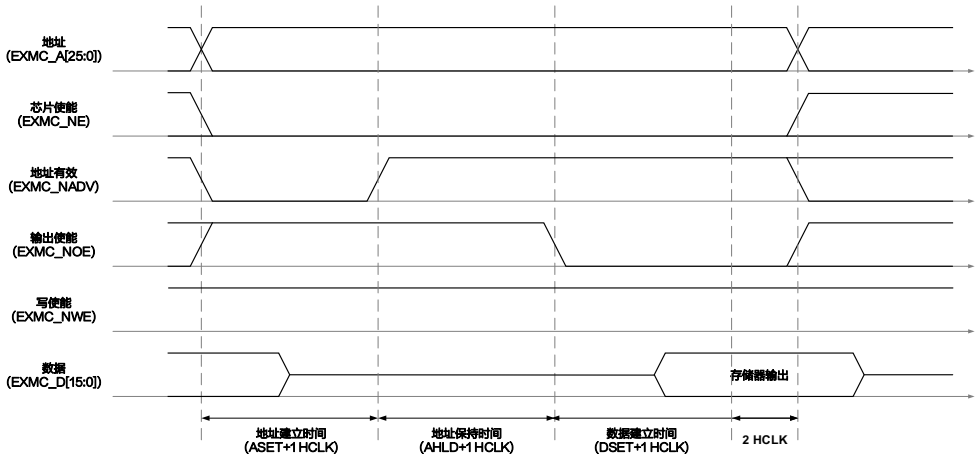



**图 20-13. 模式 D 写访问**

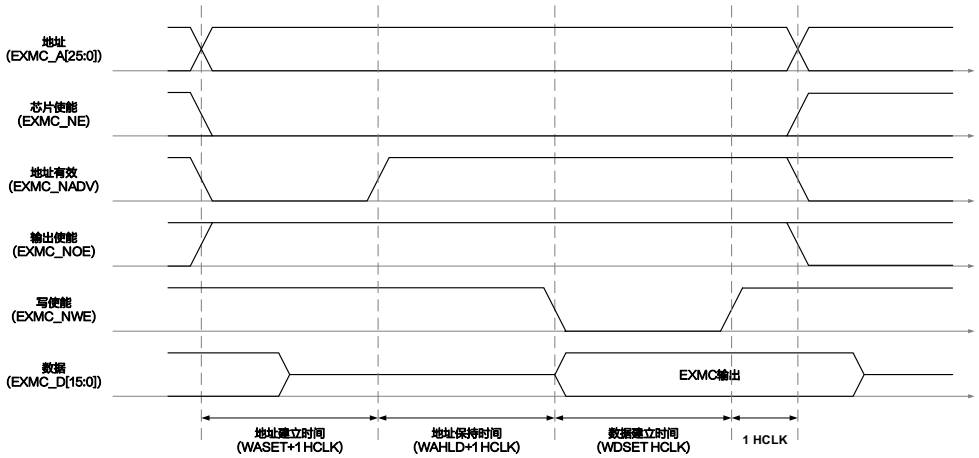


表 20-10. 模式 D 相关寄存器配置

| 位/位域 | 位名 | 参考设定值 |
|---|---|---|
| **EXMC_SNCTL** | | |
| 31-20 | 保留 | 0x000 |
| 19 | SYNCWR | 0x0 |
| 18-16 | CPS | 0x0 |
| 15 | ASYNCWTEN | 取决于存储器 |
| 14 | EXMODEN | 0x1 |
| 13 | NRWTEN | 0x0 |
| 12 | WREN | 取决于用户 |
| 11 | NRWTCFG | 无影响 |
| 10 | WRAPEN | 0x0 |
| 9 | NRWTPOL | 仅当位 15 为 1 时有效 |
| 8 | SBRSTEN | 0x0 |
| 7 | 保留 | 0x1 |
| 6 | NREN | 取决于存储器 |
| 5-4 | NRW | 取决于存储器 |
| 3-2 | NRTP | 取决于存储器 |
| 1 | NRMUX | 0x0 |
| 0 | NRBKEN | 0x1 |
| **EXMC_SNTCFG** | | |
| 31-30 | 保留 | 0x0 |
| 29-28 | ASYNCMOD | 模式 D：0x3 |
| 27-24 | DLAT | 无关 |
| 23-20 | CKDIV | 无影响 |
| 19-16 | BUSLAT | 从 EXMC_NE 上升沿到下降沿的时间 |
| 15-8 | DSET | 取决于存储器与用户(读操作为 DSET+3 HCLK 时钟周期) |
| 7-4 | AHLD | 取决于存储器与用户 |
| 3-0 | ASET | 取决于存储器与用户 |
| **EXMC_SNWTCFG** | | |
| 31-30 | 保留 | 0x0 |
| 29-28 | WASYNCMOD | 模式 D：0x3 |
| 27-20 | 保留 | 0xFF |
| 19-16 | WBUSLAT | 从 EXMC_NE 上升沿到下降沿的时间 |
| 15-8 | WDSET | 取决于存储器与用户(写操作为 WDSET+1 HCLK 时钟周期) |
| 7-4 | WAHLD | 取决于存储器与用户 |
| 3-0 | WASET | 取决于存储器与用户 |

模式AM - NOR Flash地址/数据总线复用

图 20-14. 复用模式读访问

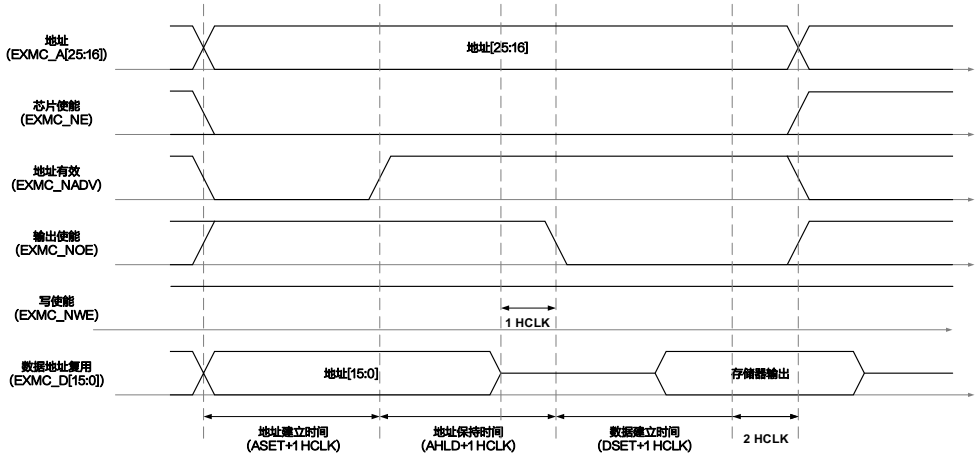



图 20-15. 复用模式写访问

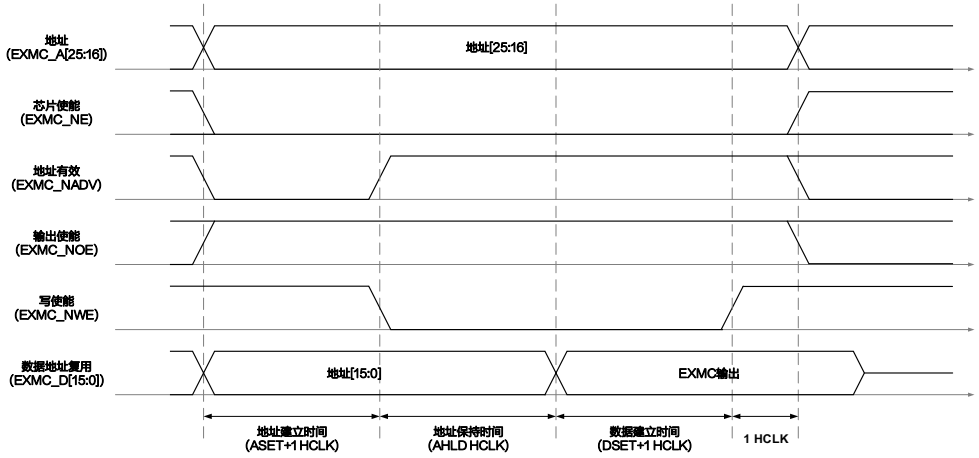



表 20-11. 复用模式相关寄存器配置

| 位/位域 | 位名 | 参考设定值 |
|---|---|---|
| EXMC_SNCTL | | |
| 31-20 | 保留 | 0x000 |
| 19 | SYNCWR | 0x0 |
| 18-16 | CPS | 0x0 |
| 15 | ASYNCWTEN | 取决于存储器 |
| 14 | EXMODEN | 0x0 |
| 13 | NRWTEN | 0x0 |
| 12 | WREN | 取决于存储器 |
| 11 | NRWTCFG | 无影响 |
| 10 | WRAPEN | 0x0 |
| 9 | NRWTPOL | 仅当位 15 为 1 时有效 |
| 8 | SBRSTEN | 0x0 |
| 7 | 保留 | 0x1 |
| 6 | NREN | 0x1 |
| 5-4 | NRW | 取决于存储器 |

| 位/位域 | 位名 | 参考设定值 |
|---|---|---|
| 3-2 | NRTP | 0x2：NOR Flash |
| 1 | NRMUX | 0x1 |
| 0 | NRBKEN | 0x1 |
| **EXMC_SNTCFG** | | |
| 31-30 | 保留 | 0x0 |
| 29-28 | ASYNCMOD | 0x0 |
| 27-24 | DLAT | 无影响 |
| 23-20 | CKDIV | 无影响 |
| 19-16 | BUSLAT | 从 EXMC_NE 上升沿到下降沿的时间 |
| 15-8 | DSET | 取决于存储器与用户(写操作为 DSET+2 HCLK 时钟周期，读操作为 DSET+3 HCLK 时钟周期) |
| 7-4 | AHLD | 取决于存储器与用户 |
| 3-0 | ASET | 取决于存储器与用户 |

**异步通信的等待时序**

等待功能由寄存器EXMC_SNCTL位ASYNCWAIT控制。在访问外部存储器期间，若使能异步等待功能(ASYNCWAIT=1)，数据建立时间将会根据EXMC_NWAIT的有效信号而自动延长。延长时间的计算如下：

若存储器等待信号与EXMC_NOE/ EXMC_NWE信号对齐：

$$T_{DATA_SETUP} >= maxT_{WAIT_ASSERTION}+4HCLK \quad (20\text{-}1)$$

若存储器等待信号与EXMC_NE信号对齐：

如果

$$maxT_{WAIT_ASSERTION} >= T_{ADDRES_PHASE}+ T_{HOLD_PHASE} \quad (20\text{-}2)$$

则

$$T_{DATA_SETUP}>=(maxT_{WAIT_ASSERTION}-T_{ADDRES_PHASE}-T_{HOLD_PHASE})+4HCLK \quad (20\text{-}3)$$

否则

$$T_{DATA_SETUP} >= 4HCLK \quad (20\text{-}4)$$

图 20-16. 异步等待有效时的读时序

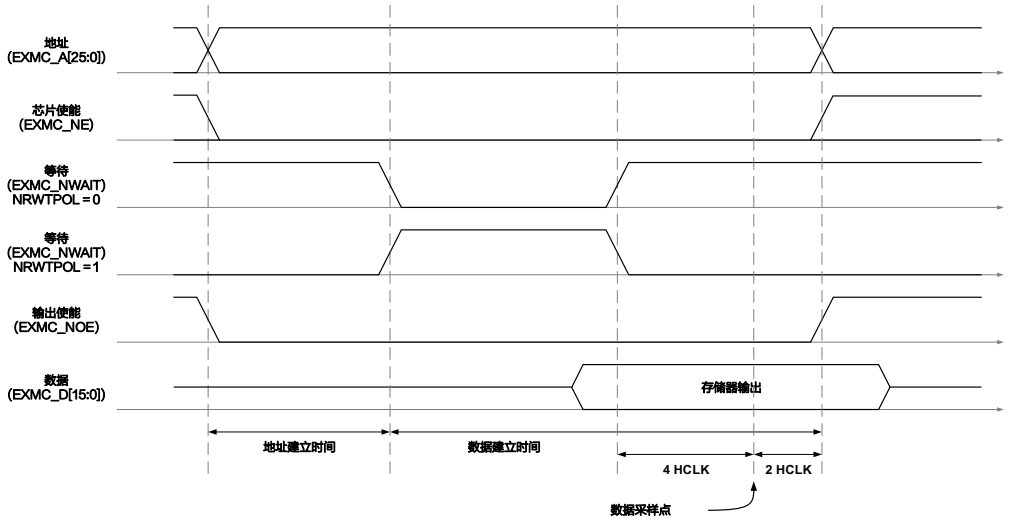
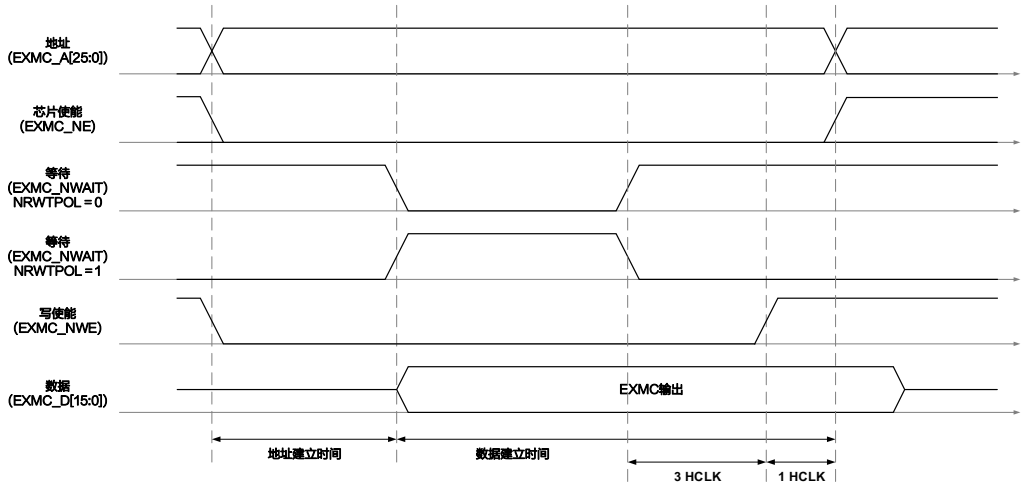



图 20-17. 异步等待有效时的写时序




同步访问时序

存储器时钟(EXMC_CLK)与系统时钟(HCLK)关系如下：

$$EXMC_CLK=\frac{HCLK}{CKDIV+1} \tag{20-5}$$

其中CKDIV是同步时钟分频比，通过配置寄存器EXMC_SNTCFG中的CKDIV位来设置不同的值。

1. 数据延迟与NOR Flash延迟

数据延迟DLAT是指在采样数据之前需要等待的EXMC_CLK周期数。它和NOR闪存延迟的关系如下所述。

NOR闪存延迟不包含EXMC_NADV，二者之间的关系为：

NOR 闪存延迟=DLAT+2 (20-6)

NOR闪存延迟包含EXMC_NADV，二者之间的关系为：

NOR 闪存延迟=DLAT+3 (20-7)

2. 数据等待

用户需要保证EXMC_NWAIT信号与外部设备一致。该信号通过寄存器EXMC_SNCTL来设置，由位NRWTEN来使能，位NRWTCFG决定EXMC_NWAIT信号是等待状态同时有效，或者比等待状态提前一个时钟周期有效，位NRWTPOL设置EXMC_NWAIT信号极性。

在NOR Flash的同步突发模式中，当寄存器EXMC_SNCTL的位NRWTEN置为1，则在数据延迟之后会检测EXMC_NWAIT信号。如果检测到EXMC_NWAIT有效，就会插入等待时钟，直到EXMC_NWAIT变为无效。

■ EXMC_NWAIT有效极性：
NRWTPOL= 1，EXMC_NWAIT高电平有效
NRWTPOL= 0，EXMC_NWAIT低电平有效
■ 在同步突发模式中，EXMC_NWAIT信号有两种配置：
NRWTCFG = 1，EXMC_NWAIT信号有效时，当前时钟周期数据无效
NRWTCFG = 0，EXMC_NWAIT信号有效时，下一个时钟周期数据无效，这是复位后的默认配置。

在EXMC_NWAIT信号有效的等待周期内，EXMC会持续的给存储器发送时钟信号，保持片选和输出使能有效，并且忽视总线上的无效数据。

3. CRAM页边界突发传输的自动分组

CRAM1.5中禁止突发传输跨越页边界，EXMC遇到边界会进行传输的自动分组。为了保证正确的突发分组操作，用户需要在寄存器EXMC_SNCTL位CPS中需要设定CRAM的页大小。

4. 模式SM - 单次突发传输

对于同步突发传输，如果AHB需要的数据为16位，则EXMC会执行一次长度为1的成组传输；如果AHB需要的数据为32位，则EXMC会把这次传输分成2次16位的传输，即执行一次长度为2的突发传输。

对于其他的配置，请参考*表20-3. EXMC的Bank0支持的所有传输*.

同步复用突发读时序 - NOR，PSRAM(CRAM)

图 20-18. 同步复用突发传输读时序

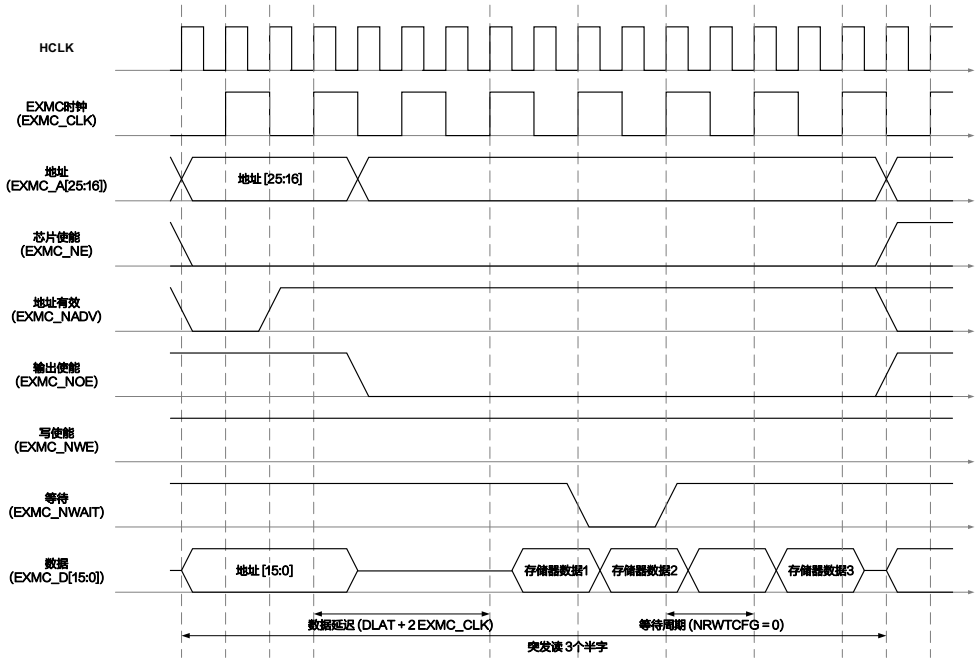



表 20-12. 同步复用模式读时序配置

| 位/位域 | 位名 | 参考设定值 |
|---|---|---|
| **EXMC_SNCTL** | | |
| 31-20 | 保留 | 0x000 |
| 19 | SYNCWR | 无影响 |
| 18-16 | CPS | 0x0 |
| 15 | ASYNCWTEN | 0x0 |
| 14 | EXMODEN | 0x0 |
| 13 | NRWTEN | 取决于存储器 |
| 12 | WREN | 无影响 |
| 11 | NRWTCFG | 取决于存储器 |
| 10 | WRAPEN | 0x0 |
| 9 | NRWTPOL | 取决于存储器 |
| 8 | SBRSTEN | 0x1，突发读使能 |
| 7 | 保留 | 0x1 |
| 6 | NREN | 取决于存储器 |
| 5-4 | NRW | 0x1 |
| 3-2 | NRTP | 取决于存储器，0x1/0x2 |
| 1 | NRMUX | 0x1，取决于存储器与用户 |
| 0 | NRBKEN | 0x1 |
| **EXMC_SNTCFG(读)** | | |
| 31-30 | 保留 | 0x0 |
| 29-28 | ASYNCMOD | 0x0 |
| 27-24 | DLAT | 数据延迟 |
| 23-20 | CKDIV | 上图设置：0x1，EXMC_CLK=2HCLK |

| 位/位域 | 位名 | 参考设定值 |
|---|---|---|
| 19-16 | BUSLAT | 从 EXMC_NE 上升沿到下降沿的时间 |
| 15-8 | DSET | 无影响 |
| 7-4 | AHLD | 无影响 |
| 3-0 | ASET | 无影响 |

模式SM - 同步复用突发写时序- PSRAM(CRAM)

**图 20-19. 同步复用突发传输写时序**

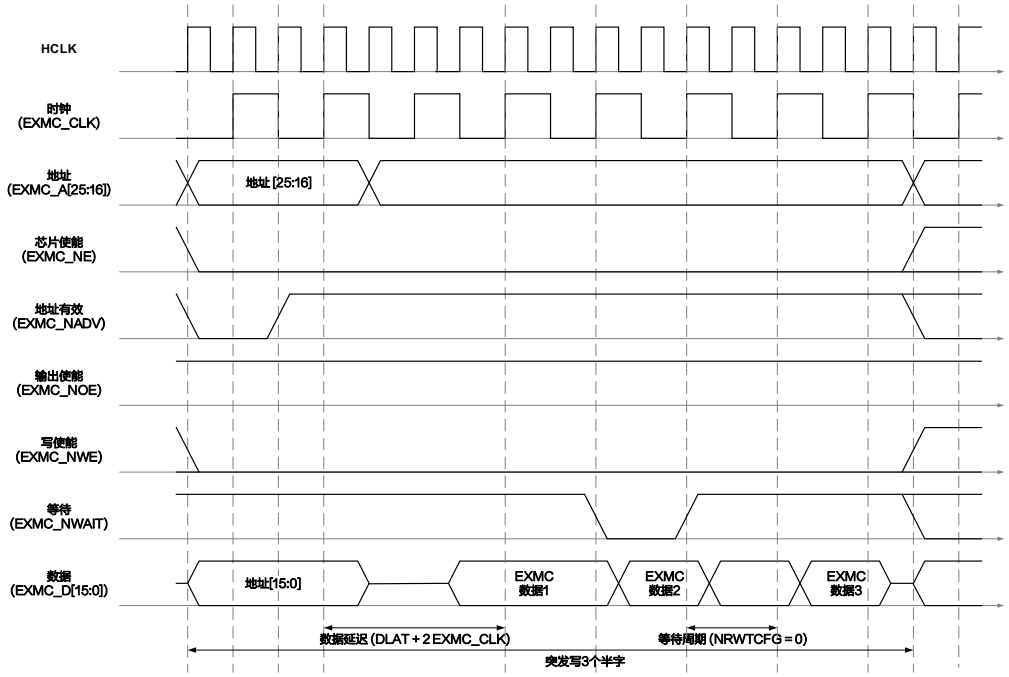



**表 20-13. 同步复用模式写时序配置**

| 位/位域 | 位名 | 参考设定值 |
|---|---|---|
| **EXMC_SNCTL** | | |
| 31-20 | 保留 | 0x000 |
| 19 | SYNCWR | 0x1，同步写使能 |
| 18-16 | CPS | 0x0 |
| 15 | AYSNCWAIT | 0x0 |
| 14 | EXMODEN | 0x0 |
| 13 | NRWTEN | 取决于存储器 |
| 12 | WREN | 0x1 |
| 11 | NRWTCFG | 0x0(这里必须为 0) |
| 10 | WRAPEN | 0x0 |
| 9 | NTWTPOL | 取决于存储器 |
| 8 | SBRSTEN | 无影响 |
| 7 | 保留 | 0x1 |
| 6 | NREN | 取决于存储器 |
| 5-4 | NRW | 0x1 |
| 3-2 | NRTP | 0x1 |

| 位/位域 | 位名 | 参考设定值 |
|---|---|---|
| 1 | NRMUX | 0x1，取决于用户 |
| 0 | NRBKEN | 0x1 |
| **EXMC_SNTCFG(写)** | | |
| 31-30 | 保留 | 0x0 |
| 29-28 | ASYNCMOD | 0x0 |
| 27-24 | DLAT | 数据延迟 |
| 23-20 | CKDIV | 上图设置：0x1，EXMC_CLK=2HCLK |
| 19-16 | BUSLAT | 从 EXMC_NE 上升沿到下降沿的时间 |
| 15-8 | DSET | 无影响 |
| 7-4 | AHLD | 无影响 |
| 3-0 | ASET | 无影响 |

## 20.4. EXMC 寄存器

EXMC基地址：0xA000 0000

### 20.4.1. SRAM/NOR Flash 控制寄存器 (EXMC_SNCTL)

偏移地址：0x00
复位值：0x0000 30DB

该寄存器只能按字(32位)访问

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | | | | | | SYNCWR | CPS[2:0] | | |
| | | | | | | | | | | | | rw | | rw | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ASYNCWAIT | EXMODEN | NRWTEN | WREN | NRWTCFG | WRAPEN | NRWTPOL | SBRSTEN | 保留 | NREN | NRW[1:0] | | NRTP[1:0] | | NRMUX | NRBKEN |
| rw | rw | rw | rw | rw | rw | rw | rw | | rw | rw | | rw | | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:20 | 保留 | 必须保持复位值。 |
| 19 | SYNCWR | 选择写操作模式<br>0：异步写操作<br>1：同步写操作 |
| 18:16 | CPS[2:0] | CRAM页大小<br>000：页边界自动突发分割<br>001：128字节<br>010：256字节<br>011：512字节<br>100：1024字节<br>其他：保留 |
| 15 | ASYNCWAIT | 异步等待功能使能位<br>0：禁用异步等待功能<br>1：使能异步等待功能 |
| 14 | EXMODEN | 扩展模式使能<br>0：禁用扩展模式<br>1：使能扩展模式 |
| 13 | NRWTEN | NWAIT信号使能<br>对于存储器的突发模式访问，该位使能/禁用在NWAIT信号中插入等待状态功能。<br>0：禁用NWAIT信号<br>1：使能NWAIT信号 |

| | | |
|---|---|---|
| 12 | WREN | 写操作使能<br>0：禁止EXMC对外部存储器的写操作，否则产生一个AHB错误<br>1：允许EXMC对外部存储器的写操作(复位缺省值) |
| 11 | NRWTCFG | NWAIT信号配置，只在同步模式有效<br>0：NWAIT信号在等待状态前的一个数据周期有效<br>1：NWAIT信号在等待状态期间有效 |
| 10 | WRAPEN | 非对齐突发模式使能<br>0：禁止非对齐突发操作<br>1：允许非对齐突发操作 |
| 9 | NRWTPOL | NWAIT信号极性<br>0：NWAIT低电平有效<br>1：NWAIT高电平有效 |
| 8 | SBRSTEN | 同步突发模式使能<br>0：禁止同步突发模式<br>1：使能同步突发模式 |
| 7 | 保留 | 必须保持复位值。 |
| 6 | NREN | NOR闪存访问使能<br>0：禁止NOR Flash访问<br>1：允许NOR Flash访问 |
| 5:4 | NRW[1:0] | 存储器数据宽度<br>00：8位<br>01：16位(复位缺省值)<br>10/11：保留 |
| 3:2 | NRTP[1:0] | 存储器类型<br>00：SRAM<br>01：PSRAM(CRAM)<br>10：NOR Flash(复位之后的默认值)<br>11：保留 |
| 1 | NRMUX | 数据线/地址线复用<br>0：禁用地址/数据复用功能<br>1：允许地址/数据复用功能 |
| 0 | NRBKEN | 存储块使能<br>0：禁用对应的存储器块<br>1：使能对应的存储器块 |

### 20.4.2. SRAM/NOR Flash 时序配置寄存器（EXMC_SNTCFG)

偏移地址：0x04

复位值：0x0FFF FFFF

该寄存器只能按字(32位)访问

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | ASYNCMOD[1:0] | | DLAT[3:0] | | | | CKDIV[3:0] | | | | BUSLAT[3:0] | | | |
| | | rw | | rw | | | | rw | | | | rw | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DSET[7:0] | | | | | | | | AHLD[3:0] | | | | ASET[3:0] | | | |
| rw | | | | | | | | rw | | | | rw | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:30 | 保留 | 必须保持复位值。 |
| 29:28 | ASYNCMOD[1:0] | 异步访问模式<br>该位域仅在当EXMC_SNCTL寄存器的EXMODEN位为1时有效。<br>00：模式A<br>01：模式B<br>10：模式C<br>11：模式D |
| 27:24 | DLAT[3:0] | NOR Flash数据延时，仅在同步模式有效<br>0x0：第一组突发传输时，数据延迟时间为2个EXMC_CLK时钟周期<br>0x1：第一组突发传输时，数据延迟时间为3个EXMC_CLK时钟周期<br>……<br>0xF：第一组突发传输时，数据延迟时间为17个EXMC_CLK时钟周期 |
| 23:20 | CKDIV[3:0] | 同步模式时钟分频比，仅在同步模式有效<br>0x0：保留<br>0x1：EXMC_CLK周期=2个HCLK周期<br>……<br>0xF：EXMC_CLK周期=16个HCLK周期 |
| 19:16 | BUSLAT[3:0] | 总线延迟时间<br>在复用读模式中使用，避免总线冲突，是总线恢复到高阻态的最小时间。<br>0x0：总线延迟=1个HCLK周期<br>0x1：总线延迟=2个HCLK周期<br>……<br>0xF：总线延迟=16个HCLK周期 |
| 15:8 | DSET[7:0] | 异步数据建立时间<br>该位域仅在异步模式有效。<br>0x00：保留<br>0x01：数据建立时间=2个HCLK周期<br>……<br>0xFF：数据建立时间=256个HCLK周期 |
| 7:4 | AHLD[3:0] | 异步地址保持时间 |

| | | |
|---|---|---|
| | | 该位域设置地址保持时间，仅在模式D与复用模式有效。<br>0x0：保留<br>0x1：地址保持时间=2个HCLK<br>……<br>0xF：地址保持时间=16个HCLK |
| 3:0 | ASET[3:0] | 异步地址建立时间<br>该位域设置地址建立时间。<br>**注意**：该位域仅在SRAM，ROM，NOR Flash的异步模式有效。<br>0x0：地址建立时间= 1个HCLK<br>……<br>0xF：地址建立时间= 16个HCLK |

### 20.4.3. SRAM/NOR Flash 写时序寄存器 (EXMC_SNWTCFG)

偏移地址：0x104
复位值：0x0FFF FFFF

该寄存器仅在扩展模式使能(寄存器EXMC_SNCTL位EXMODEN置1)后有效。

该寄存器只能按字(32位)访问

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | WASYNCMOD[1:0] | | 保留 | | | | | | | | WBUSLAT[3:0] | | | |
| | | rw | | | | | | | | | | | rw | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WDSET[7:0] | | | | | | | | WAHLD[3:0] | | | | WASET[3:0] | | | |
| rw | | | | | | | | rw | | | | rw | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:30 | 保留 | 必须保持复位值。 |
| 29:28 | WASYNCMOD[1:0] | 异步访问模式<br>该位只有在扩展模式中使用。<br>00：模式A<br>01：模式B<br>10：模式C<br>11：模式D |
| 27:20 | 保留 | 必须保持复位值。 |
| 19:16 | WBUSLAT[3:0] | 总线延迟时间<br>在每次写传输结束的时候增加总线延时时间来满足连续传输之间的最小时间。<br>0x0：总线延迟=1个HCLK周期<br>0x1：总线延迟=2个HCLK周期<br>…… |

| | | |
|---|---|---|
| | | 0xF：总线延迟=16个HCLK周期 |
| 15:8 | WDSET[7:0] | 异步数据建立时间<br>该位域仅在异步模式有效。<br>0x00：保留<br>0x01：数据建立时间=2个HCLK周期<br>……<br>0xFF：数据建立时间=256个HCLK周期 |
| 7:4 | WAHLD[3:0] | 异步地址保持时间<br>该位域设置地址保持时间，仅在模式D与复用模式有效。<br>0x0：保留<br>0x1：地址保持时间=2个HCLK<br>……<br>0xF：地址保持时间=16个HCLK |
| 3:0 | WASET[3:0] | 异步地址建立时间<br>该位域设置地址建立时间<br>**注意**：该位域仅在SRAM，ROM，NOR Flash的异步模式有效。<br>0x0：地址建立时间= 1个HCLK<br>0x1：地址建立时间= 2个HCLK<br>……<br>0xF：地址建立时间= 16个HCLK |

# 21. 通用串行总线全速接口(USBFS)

USBFS适用于GD32E11x系列芯片。

## 21.1. 概述

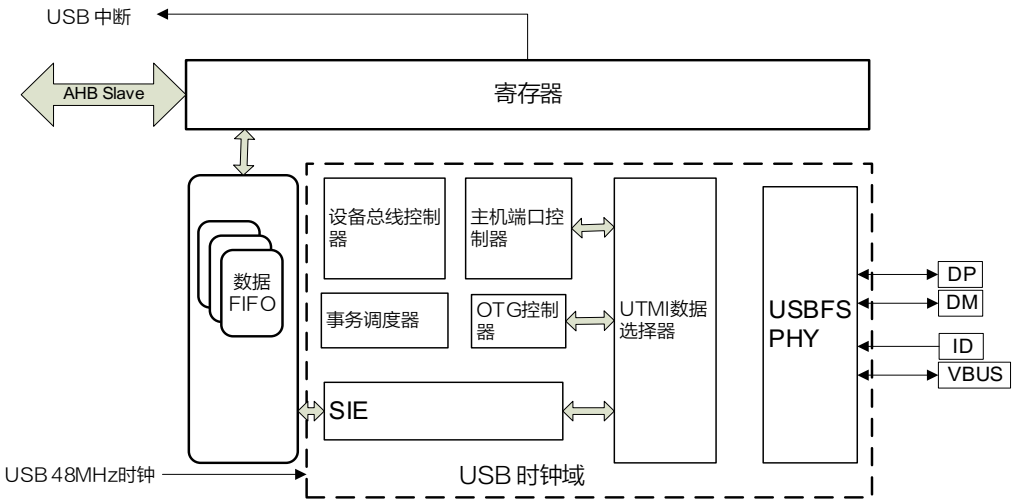
USB全速(USBFS)控制器为便携式设备提供了一套USB互联解决方案。USBFS不仅支持主机模式和设备模式，也支持遵循HNP(主机协商协议)和SRP(会话请求协议)的OTG模式。USBFS包含了一个内部的全速USB PHY，并且不再需要外部PHY芯片。USBFS可以支持USB 2.0协议所定义的所有四种传输方式(控制传输、批量传输、中断传输和同步传输)。

## 21.2. 主要特征

- 支持USB 2.0全速(12Mb/s)/低速(1.5Mb/s)主机模式；
- 支持USB 2.0全速(12Mb/s)设备模式；
- 支持遵循HNP(主机协商协议)和SRP(会话请求协议)的OTG协议；
- 支持所有的4种传输方式：控制传输、批量传输、中断传输和同步传输；
- 在主机模式下，包含USB事务调度器，用于有效地处理USB事务请求；
- 包含一个1.25KB的FIFO RAM；
- 在主机模式下，支持8个通道；
- 在主机模式下，包含2个发送FIFO(周期性发送FIFO和非周期性发送FIFO)和1个接收FIFO(由所有的通道共享)；
- 在设备模式下，包含4个发送FIFO(每个IN端点一个发送FIFO)和1个接收FIFO(由所有的OUT端点共享)；
- 在设备模式下，支持4个OUT端点和4个IN端点；
- 在设备模式下，支持远程唤醒功能；
- 包含一个支持USB协议的全速USB PHY；
- 在主机模式下，SOF的时间间隔可动态调节；
- 可将SOF脉冲输出到PAD；
- 可检测ID引脚电平和VBUS电压；
- 在主机模式或者OTG A设备模式下，需要外部部件为连接的USB设备提供电源。

## 21.3. 结构框图

图 21-1. USBFS 结构框图




## 21.4. 信号线描述

表 21-1. USBFS 信号线描述

| I/O 端口 | 类型 | 描述 |
|---|---|---|
| VBUS | 输入/输出 | 总线电源端口 |
| DM | 输入/输出 | 差分信号 D-端口 |
| DP | 输入/输出 | 差分信号 D+端口 |
| ID | 输入 | USB 识别：微连接器识别接口 |

## 21.5. 功能说明

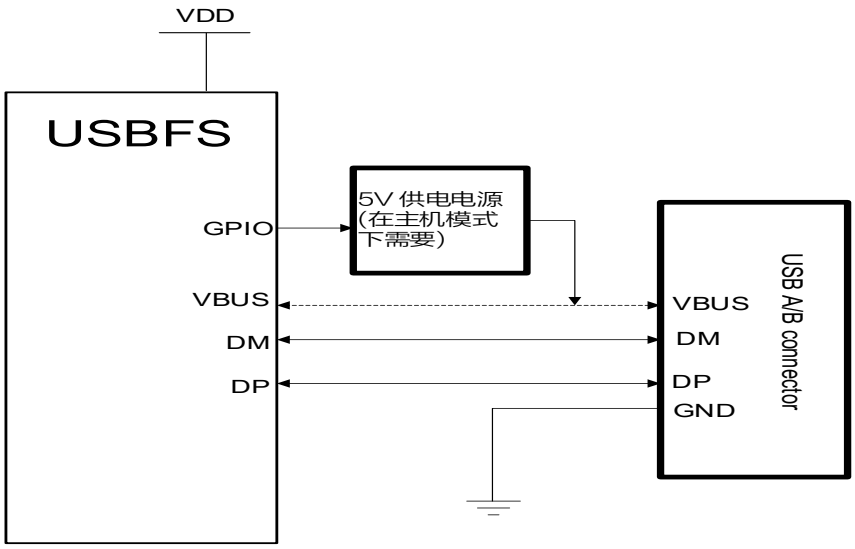
### 21.5.1. USBFS 时钟及工作模式

USBFS可以作为一个主机、一个设备或者一个DRD(双角色设备),并且包含一个内部全速PHY。USBFS可支持的最大速率为全速。

内部PHY支持全速和低速的主机模式、全速的设备模式以及具备HNP和SRP的OTG模式。USBFS所使用的USB时钟需要配置为48MHz。该48MHz USB时钟从系统内部时钟产生，并且其时钟源和分频器需要在RCU模块中配置。

上拉或下拉电阻已经集成在内部全速PHY的内部，并且USBFS可根据当前模式(主机、设备或OTG模式)和连接状态进行自动控制。一个利用内部全速PHY的典型连接示意图如*图21-2. 在主机或设备模式下连接示意图*所示。

**图 21-2. 在主机或设备模式下连接示意图**

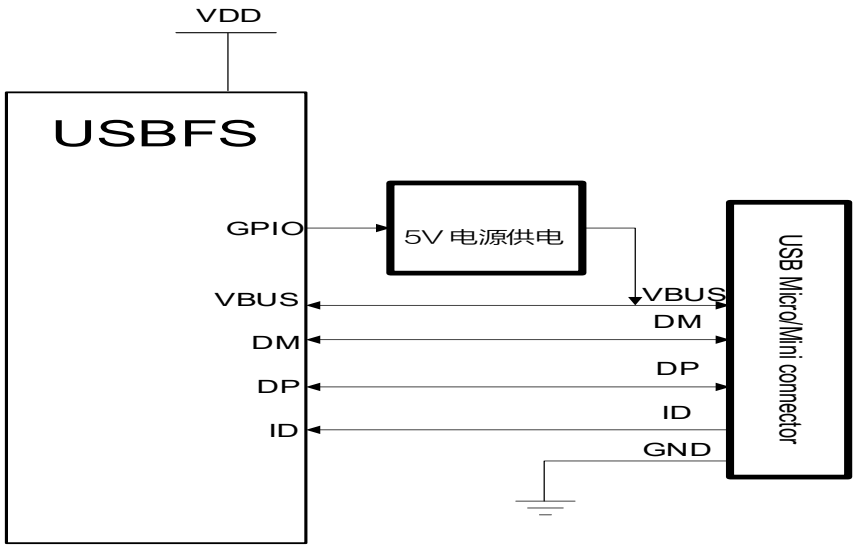



当USBFS工作在主机模式下时(FHM控制位置位、FDM控制位清除)，VBUS为USB协议所定义的5V电源检测引脚。内部PHY不能提供5V VBUS电源，仅在VBUS信号线上具有电压比较器和充电、放电电路。所以，如果应用需要提供VBUS电源，那么则需要一个外部的供电电源IC。在主机模式下，USBFS和USB连接器之间的VUBS连接可以被忽略，这是由于USBFS并不检测VBUS引脚的电平状态，并假定5V供电电源一直存在。

当USBFS工作在设备模式下时(FHM控制位清除、FDM控制位置位)，VBUS检测电路由USBFS_GCCFG寄存器中的VBUSIG控制位所配置。因此，如果设备不需要检测VBUS引脚电压，可以配置VBUSIG控制位，并可释放VBUS引脚作为其他用途。否则，VBUS引脚的连接不能够被忽略，并且USBFS需要不断的检测VBUS电平状态，一旦VBUS电压降至所需有效值以下，需要立即关闭DP信号线上的上拉电阻，从而产生一个断开状态。

OTG 模式连接示意图如*图 21-3. OTG 模式下连接示意图*所示。当 USBFS 工作在 OTG 模式下时，USBFS_GUSBCS 寄存器内的 FHM、FDM 控制位和 USBFS_GCCFG 寄存器的 VBUSIG 位都应该被清除。在这种模式下，USBFS 需要以下四个引脚：DM、DP、VBUS 和 ID，并且需要使用若干个电压比较器检测这些引脚的电压。 USBFS 也包含 VBUS 充电和放电电路，用以完成 OTG 协议中所描述的 SRP 请求。OTG A 设备或 B 设备由 ID 引脚的电平状态所决定。在实现 HNP 协议的过程中，USBFS 控制上拉和下拉电阻。

图 21-3. OTG 模式下连接示意图

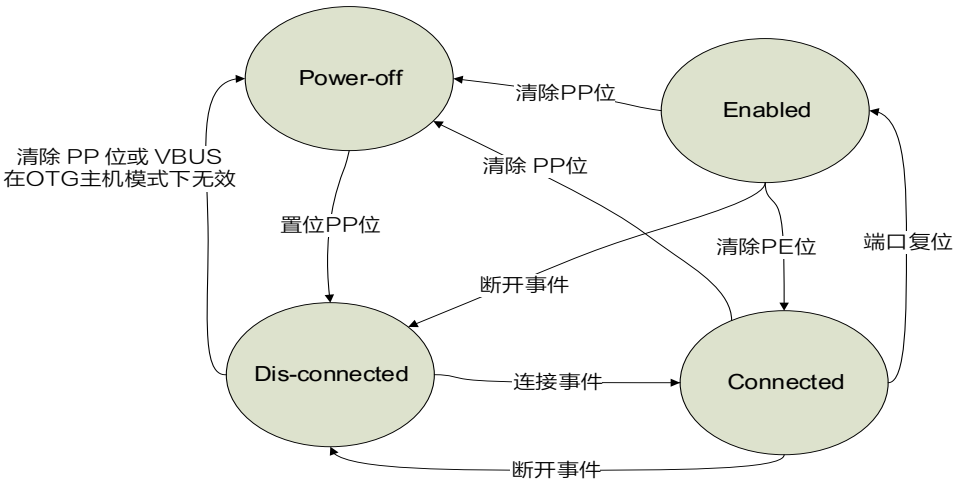



## 21.5.2. USB 主机功能

### USB主机端口状态

主机应用可以通过USBFS_HPCS寄存器控制USB端口状态。系统初始化之后，USB端口保持掉电状态。通过软件置位PP控制位后，内部USB PHY将被上电，并且USB端口变为断开状态。检测到连接后，USB端口变为连接状态。在USB总线上产生一个复位后，USB端口将变为使能状态。

图 21-4. 主机端口状态转移图




### 连接、复位和速度识别

作为USB主机，在检测到一个连接事件后，USBFS会为应用触发一个连接标志；同样，若检测到一个断开事件后，将会触发一个断开标志。

PRST控制位用于实现USB复位序列。应用可以置位该控制位以启动一个USB复位序列，或者

清除该控制位以结束USB复位序列。仅当端口在连接或使能状态时，该控制位有效。

USBFS在对设备连接和复位时执行速度检测，并且速度检测的结果会反馈在USBFS_HPCS寄存器的PS位域中。USBFS以DM或DP的电平状态确定设备速度，如USB协议所描述，全速设备上拉DP信号线，而低速设备上拉DM信号线。

**挂起和复位**

USBFS支持挂起和复位状态，当USBFS端口在使能状态时，向USBFS_HPCS寄存器的PSP控制位写1，USBFS会进入到挂起状态。在挂起状态下，USBFS停止在USB总线上发送SOF，并且这样会让所连接的USB设备在3ms后进入挂起状态。应用程序能够置位USBFS_HPCS寄存器中的PREM控制位以启动一个恢复序列，从而唤醒挂起的设备，当清除该控制位时，则可以停止恢复序列。如果主机在挂起状态下检测到一个远程唤醒信号，将会置位USBFS_GINTF寄存器的WKUPIF标志位，并且触发USBFS唤醒中断。

**SOF产生器**

在主机模式下，USBFS向USB总线发送SOF令牌包。如USB 2.0协议所描述，全速连接下，每毫秒产生一次SOF令牌包(由主机控制器或者HUB事务转换器产生)。

每当USBFS进入到使能状态后，它将会按照USB2.0所定义的周期发送SOF令牌包。然而，应用程序可以通过写USBFS_HFT寄存器中的FRI位来调整一帧的间隔。FRI位定义了在一帧中的USB时钟周期个数，并且应用程序应该基于USBFS所使用的USB时钟频率计算该值。FRT位显示当前帧剩余的时钟周期个数，并且在挂起状态时，该值将停止改变。

USBFS能够在每个SOF令牌包中产生一个脉冲信号，并且将其输出至一个引脚。该脉冲信号长度为12个HCLK周期。如果应用程序希望使用该功能，需要置位USBFS_GCCFG寄存器的SOFOEN控制位，并且配置相应的引脚寄存器为GPIO功能。

**USB通道和事务**

USBFS在主机模式下包含8个独立的通道。每个通道能够与一个USB设备端点通信。通道的传输类型、方向、数据包长和其他信息都在通道相应的寄存器中配置，例如USBFS_HCHxCTL和USBFS_HCHxLEN寄存器。

USBFS支持所有的四种传输类型：控制、批量、中断和同步。USB 2.0协议将这些传输类型划分为两类：非周期性传输(控制和批量)和周期性传输(中断和同步)。基于此，为了有效地进行事务调度，USBFS包含两种请求队列：周期性请求队列和非周期性请求队列。在上述请求队列中的请求条目可能代表一个USB事务请求或者一个通道操作请求。

如果应用程序想要在USB总线上启动一个OUT事务，需要通过AHB寄存器接口向数据FIFO中写入数据包。USBFS硬件会在整包数据写完后，自动产生一个事务请求并进入请求队列。

请求队列中的请求条目通过事务控制模块按顺序处理。USBFS通常首先尝试处理周期性请求队列，然后处理非周期性请求队列。

帧起始后，USBFS首先开始处理周期性队列，直到队列为空抑或当前周期性请求队列所需时间不够，然后处理非周期性队列。这种做法保证了一帧中周期性传输的带宽。每次USBFS从请求队列中读取并取出一个请求条目。如果取出的是通道禁用请求，这将直接禁用通道并准备处理下个条目。

如果当前请求是一个事务请求并且USB总线时间能够处理这个请求，USBFS会使用SIE在USB总线上产生该事务。

在当前帧内，当前请求所需的总线时间不足时，如果当前请求为周期性请求，USBFS停止处理该周期性请求队列，并启动处理非周期性请求。如果当前请求为非周期性请求，USBFS会停止处理任何队列，并等待直到当前帧结束。

### 21.5.3. USB 设备功能

**USB设备连接**

在设备模式下，USBFS在初始化后保持掉电状态。利用VBUS引脚上的5V电源连接USB主机后或者置位USBFS_GCCFG寄存器中VBUSIG控制位，USBFS将进入供电状态。USBFS首先打开DP信号线上的上拉电阻，之后主机将会检测到一个连接事件。

**复位和速度识别**

USB主机在检测到设备连接之后，总是会启动一个USB复位序列，并且在设备模式下，检测到USB总线复位事件后，USBFS会为软件触发一个复位中断。

在复位序列后，USBFS将会触发USBFS_GINTF寄存器中的ENUMF中断，并且利用USBFS_DSTAT寄存器内的ES标志位指示当前枚举设备速度，该位总是为11(全速)。

如USB 2.0协议所描述，USBFS在外设模式下不支持低速。

**挂起和唤醒**

USB总线保持IDLE状态并且数据线3ms无变化，USB设备将会进入挂起状态。当USB设备在挂起状态时，软件能够关闭大部分的时钟以节省电能。USB主机可以通过在USB总线上产生恢复信号，来唤醒挂起的设备。USBFS检测到恢复信号后，将置位USBFS_GINTF寄存器的WKUPIF标志位并且触发USBFS唤醒中断。

在挂起设备模式，USBFS也能够远程唤醒USB总线。软件可以通过置位USBFS_DCTL寄存器的RWKUP控制位来发送一个远程唤醒信号，并且如果USB主机支持远程唤醒，主机会在USB总线上启动发送一个恢复信号。

**软件断开**

USBFS支持软件断开。设备进入到供电状态后，USBFS会打开DP信号线的上拉电阻，并且这样主机会检测到设备连接。然后，软件可以通过置位USBFS_DCTL寄存器中SD控制位进行强制断开。在SD控制位被置位后，USBFS将会直接关闭上拉电阻。这样，USB主机将会在USB总线上检测到设备断开。

**SOF跟踪**

当USBFS在USB总线上接收到一个SOF令牌包时，将触发一个SOF中断，并且开始利用本地USB时钟计算总线时间。当前帧的帧号将会反应在USBFS_DSTAT寄存器的FNRSOF位域中。当USB总线时间达到EOF1或EOF2点(帧结束，在USB 2.0协议中描述)，USBFS会触发USBFS_GINTF寄存器中的EOPFIF中断。软件能够使用这些标志位和寄存器以获得当前总线时间和位置信息。

### 21.5.4. OTG 功能概述

USBFS支持OTG协议1.3中所描述的OTG功能，OTG功能包括SRP和HNP。

**A设备和B设备**

当标准A或微型A插头插入相应的插座时，具有OTG能力的USB设备为A设备。A设备向VBUS供电，并且在会话开始时默认为主机。当标准B、微型B、迷你B插头插入相应的插座或采用一端为标准A插头的不可分离电缆时，具有OTG能力的USB设备为B设备。B设备在会话开始时默认为外设。USBFS使用ID引脚电平状态决定A设备或B设备。ID引脚状态反馈在USBFS_GOTGCS寄存器的IDPS状态位。为了了解A设备和B设备之间传输的详细状态，请参考OTG1.3协议。

**HNP**

主机协商协议(HNP)允许主机功能在两个直接连接的OTG设备之间转换，并且用户不需要为了设备之间通信控制的改变而切换电缆线的连接。典型地，HNP协议是由B设备上的用户或应用启动，HNP只能通过设备上的微型AB插座执行。

一旦OTG设备具有一个微型AB插座，该OTG设备可通过插入的插头类型决定默认为主机或设备(微型A插头插入为主机，微型B插头插入为设备)。通过使用主机协商协议(HNP)，一个默认为外设的OTG设备可以请求成为主机。主机角色切换的过程在下段中描述。此协议使用户不需要为了更改连接设备的角色而切换电缆线的连接。

当USBFS工作在OTG A主机模式时，并且其想放弃主机角色，可以首先置位USBFS_HPCS寄存器的PSP控制位来使USB总线进入挂起状态，然后B设备在3ms后进入挂起状态。如果B设备想要变为主机，软件需要置位USBFS_GOTGCS寄存器的HNPREQ控制位，然后USBFS会开始在总线上执行HNP协议，最后，HNP的结果会反馈在USBFS_GOTGCS寄存器的HNPS状态位。另外，软件总能从USBFS_GINTF寄存器的COPM状态位获取当前设备角色(主机或外设)。

**SRP**

会话请求协议(SRP)允许B设备请求A设备打开VBUS并启动一个会话。该协议允许A设备(或许是电池供电)当总线无活动时通过关闭VBUS以节省电能，并为B设备启动总线活动提供了一种方法。如OTG协议中所描述，OTG设备必须和几个阈值比较VBUS电压，并且将比较结果反馈在USBFS_GOTGCS寄存器的ASV和BSV状态位中。

当USBFS工作在B设备OTG模式时，软件可以通过置位USBFS_GOTGCS寄存器的SRPREQ控制位来启动一个SRP请求，并且如果SRP请求成功，USBFS会在USBFS_GOTGCS寄存器中产生一个成功标志位SRPS。

当USBFS工作在OTG A设备模式且从B设备检测到一个SRP请求时，USBFS将会置位USBFS_GINTF寄存器中的SESIF标志位。软件获取该标志位后，需要准备为VBUS引脚打开5V供电电源。

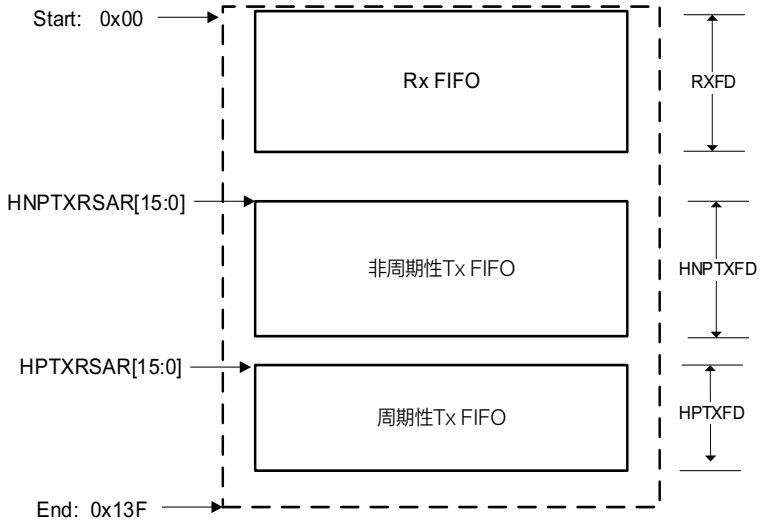
### 21.5.5. 数据 FIFO

USBFS中采用1.25K字节数据FIFO存储包数据，数据FIFO是通过USBFS的内部SRAM实现的。

### 主机模式

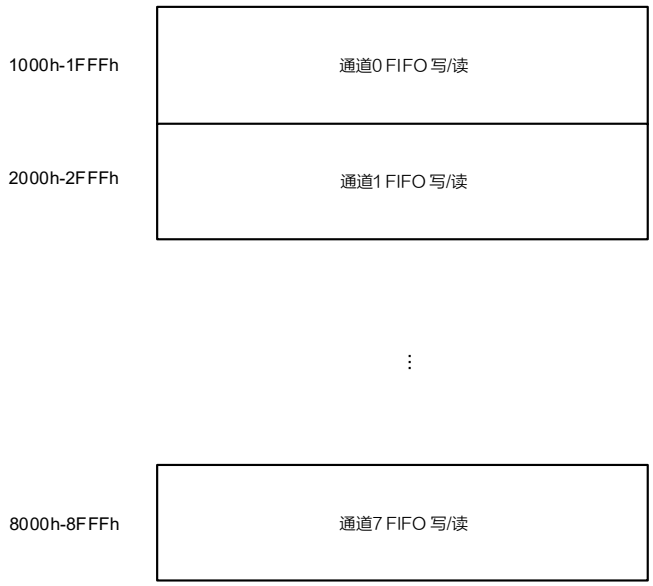
主机模式下，数据 FIFO 空间分为三个部分，分别是：用于接收数据包的 Rx FIFO、用于非周期性发送数据包的非周期性 Tx FIFO 和用于周期性发送数据包的周期性 Tx FIFO。所有的 IN 通道通过共享 Rx FIFO 接收数据。所有的周期性 OUT 通道通过共享周期性 Tx FIFO 来发送数据，所有的非周期性 OUT 通道通过共享非周期性 Tx FIFO 来发送数据。通过寄存器 USBFS_GRFLEN、USBFS_HNPTFLEN 和 USBFS_HPTFLEN，软件可以配置以上数据 FIFO 的大小和起始偏移地址。*图 21-5. 主机模式 FIFO 空间*所描述的是 SRAM 中各 FIFO 的结构，图中的数值是按照 32 位为单位写的。

**图 21-5. 主机模式 FIFO 空间**




USBFS 为程序提供了专有寄存器空间来读写数据 FIFO。*图 21-6. 主机模式 FIFO 访问寄存器映射表*所描述的是数据 FIFO 所访问的寄存器存储空间，图中的数值是以字节为单位寻址。尽管所有的非周期通道共享相同的 FIFO 以及所有的周期通道共享相同的 FIFO，每个通道都拥有它们的 FIFO 访问寄存器空间。对 USBFS 而言，获知当前压入数据包的通道号是非常重要的，通过寄存器 USBFS_GRXTATR/USBFS_GRSTATP 来访问数据包所从属的 Rx FIFO。

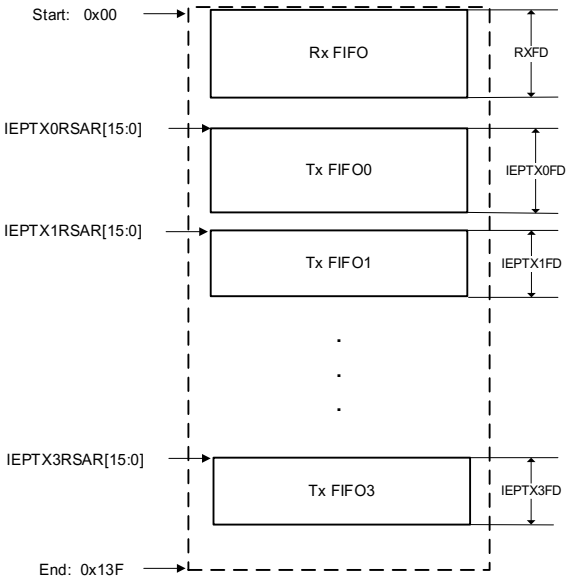
**图 21-6. 主机模式 FIFO 访问寄存器映射表**




### 设备模式

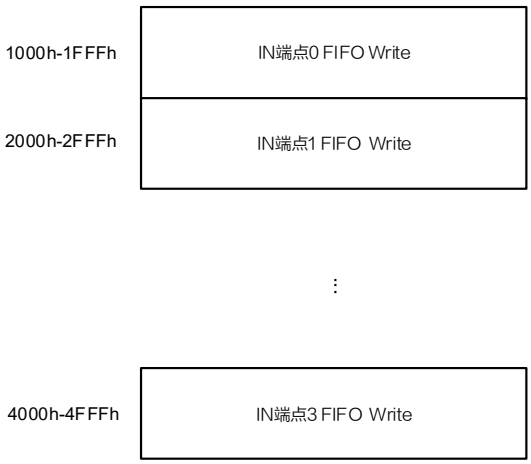
在设备模式下，数据 FIFO 分为多个部分，其中包含 1 个 Rx FIFO 和 4 个 Tx FIFO，每个 Tx FIFO 对应着一个 IN 端点，所有的 OUT 端点通过共享 Rx FIFO 接收数据包。通过寄存器 USBFS_GRFLEN 和 USBFS_DIEPxTFLEN (x=0…3)，程序可配置数据 FIFO 的大小和起始偏移地址。*图 21-7. 设备模式 FIFO 空间*所描述的是 SRAM 中各 FIFO 的结构，图中的数值是以按照 32 位写的。

**图 21-7. 设备模式 FIFO 空间**




USBFS 为程序提供了专有寄存器空间来读写数据 FIFO。*图 21-8. 设备模式 FIFO 访问寄存器映射表*所描述的是数据 FIFO 所访问的寄存器存储空间，图中的数值是以字节为单位寻址。每个端点都拥有它们的 FIFO 访问寄存器空间。通过寄存器 USBFS_GRXTATR/USBFS_GRSTATP 来访问 Rx FIFO。

**图 21-8. 设备模式 FIFO 访问寄存器映射表**




## 21.5.6. 操作手册

该部分描述的是USBFS的操作手册。

### 主机模式

**全局寄存器初始化顺序：**

1、 根据应用的需求，如Tx FIFO的空阈值等，设置寄存器USBFS_GAHBCS，此时，GINTEN位需要保持清零状态；
2、 根据应用的需求，如操作模式(主机、设备或OTG)、某些OTG参数和USB协议，设置寄存器USBFS_GUSBCS；
3、 根据应用的需求，设置寄存器USBFS_GCCFG；
4、 根据应用的需求，设置寄存器USBFS_GRFLEN、USBFS_HNPTFLEN_DIEP0TFLEN、USBFS_HPTFLEN，配置数据FIFO；
5、 通过设置寄存器USBFS_GINTEN使能模式错误和主机端口中断，置位USBFS_GAHBCS寄存器的GINTEN位使能全局中断；
6、 设置寄存器USBFS_HPCS，置位PP位；
7、 等待设备连接，当设备连接后，触发寄存器USBFS_HPCS的PCD位，然后置位PRST位，执行一次端口复位，等待至少10毫秒后，清除PRST位；
8、 等待USBFS_HPCS寄存器的PEDC中断，然后读取PE位以确认端口被成功地使能，读取PS位以获取连接的设备速度，之后，如果软件需要改变SOF间隔，设置USBFS_HFT寄存器。

**通道初始化和使能顺序：**

1、 根据期望的传输类型、方向、包大小等信息，设置寄存器USBFS_HCHxCTL，在设置期间，要保证位CEN和CDIS保持清除；
2、 设置寄存器USBFS_HCHxINTEN，设置期望的中断使能位；
3、 设置寄存器USBFS_HCHxLEN，PCNT表示一次传输中的包数，TLEN表示一次传输中发送或接收的包数据的总字节数；

4、 对于OUT通道，如果PCNT为1，单包的大小等于TLEN。如果PCNT大于1，前PCNT-1个包被认定为最大包长度的包，其大小是由寄存器USBFS_HCHxCTL的位MPL所定义。最后一包的大小可通过PCNT、TLEN和MPL计算得到。如果程序想要发出一个零长度的包，应该设定TLEN为0，PCNT位1；
5、 对于IN通道，因为在IN事务结束之前，程序不知道实际接收的数据大小，程序可将TLEN设定为Rx FIFO所支持的最大值；
6、 置位寄存器USBFS_HCHxCTL中的CEN位以使能通道。

**通道除能顺序：**

程序可以通过同时置位CEN和CDIS除能通道。在寄存器操作后，USBFS将在请求队列中产生一个通道除能请求条目。当这个请求条目到达请求队列的顶部时，USBFS立即进行处理。

对于OUT通道而言，特定的通道将被立即除能。然后，会产生CH标志，USBFS将清除CEN和CDIS位。

对于IN通道而言，USBFS将通道除能状态条目压入Rx FIFO，然后，程序应该处理Rx FIFO非空事件：读和取出该状态条目，然后会产生CH标志，USBFS将清除CEN和CDIS位。

**IN传输操作顺序：**

1、 初始化USBFS全局寄存器；
2、 初始化相应的通道；
3、 使能相应的通道；
4、 通过软件使能IN通道后，USBFS在相应请求队列中生成一个Rx请求条目；
5、 当Rx请求条目到达请求队列的顶部时，USBFS开始执行该请求条目。对于由请求条目所指示的事务而言，如果总线时间足够，USBFS在USB总线上开始IN事务；
6、 当IN事务结束时(收到ACK握手包)，USBFS将接收到的数据包压入Rx FIFO，ACK标志位被触发，否则，状态标志(NAK)会指示事务结果；
7、 如果步骤5所描述的IN事务完成后，步骤2的PCNT的数值比1大，程序将会返回步骤3，继续接收剩下的数据包。如果步骤5中描述的IN事务没有成功完成，程序将会返回步骤3来再次发送该数据包；
8、 在所有的传输中的所有事务都被成功接收后，USBFS将TF状态条目压入Rx FIFO的最后的数据包的顶部，这样，软件在读取所有接收的数据包后，再读取TF状态条目。USBFS生成TF标志来指示传输成功结束；
9、 除能通道，当通道处于空闲状态，即可为其他传输做准备。

**OUT传输操作顺序：**

1、 初始化USBFS全局寄存器；
2、 初始化及使能相应通道；
3、 将数据包写入通道的Tx FIFO(周期性Tx FIFO或非周期性Tx FIFO)。在所有的数据包都被写入FIFO后，USBFS在相应的请求队列中产生一个Tx请求条目，并且将USBFS_HCHxTLEN中的TLEN值减少，减少的数值等于已写的包大小；
4、 当请求条目到达请求队列的顶部时，USBFS开始执行该请求条目。如果请求条目对应的事务的总线时间足够，USBFS在USB总线上开展OUT事务；
5、 当由请求条目所指示的OUT事务结束时，寄存器USBFS_HCHnTLEN的位PCNT减1。如果该事务完成(收到ACK握手包)，ACK标志位被触发，否则，状态标志(NAK)会指示事务结果；

6、 如果步骤5所描述的OUT事务完成后且步骤2的PCNT的数值比1大，程序将会返回步骤3，继续发送剩下的数据包。如果步骤5中描述的OUT事务没有成功完成，程序将会返回步骤3来再次发送该包；
7、 在所有的传输中的所有事务都被成功送达后，USBFS生成TF标志来指示传输成功结束；
8、 除能通道，当通道处于空闲状态，即可为其他传输做准备。

### 设备模式

**全局寄存器初始化顺序：**

1、 根据应用的需求，如Tx FIFO的空阈值等，设置寄存器USBFS_GAHBCS，此时，GINTEN位需要保持清零状态；
2、 根据应用的需求，如操作模式(主机、设备或OTG)、某些OTG参数、USB协议，设置寄存器USBFS_GUSBCS；
3、 根据应用的需求，设置寄存器USBFS_GCCFG；
4、 根据应用的需求，设置寄存器USBFS_GRFLEN、USBFS_HNPTFLEN_DIEP0TFLEN、USBFS_HPTFLEN，配置数据FIFO；
5、 通过设置寄存器USBFS_GINTEN使能模式错误、挂起、SOF、枚举完成和USB复位中断，置位USBFS_GAHBCS寄存器的GINTEN位使能全局中断；
6、 根据应用的需求，如设备的地址等，设置寄存器USBFS_DCFG；
7、 在设备连接上主机上后，主机在USB总线上执行端口复位，触发寄存器USBFS_GINTF的RST中断；
8、 等待寄存器USBFS_GINTF的ENUMF中断。

**端点初始化和使能顺序：**

1、 根据预期的传输类型、包大小等信息，设置寄存器 USBFS_DIEPnCTL 或 USBFS_DOEPxCTL；
2、 设定寄存器 USBFS_DIEPINTEN 或 USBFS_DOEPINTEN，置位相应中断使能位；
3、 设定寄存器 USBFS_DIEPxLEN 或 USBFS_DOEPxLEN，PCNT 表示一次传输中的包数，TLEN 表示一次传输中发送或接收的数据包的总字节数；
4、 对于 IN 端点，如果 PCNT 等于 1，单数据包的大小等于 TLEN。如果 PCNT 大于 1，前 PCNT-1 个包被认定为最大包长度的包，其大小是由寄存器 USBFS_DIEPxCTL 的位 MPL 所定义。最后一包的大小可通过 PCNT、TLEN 和 MPL 计算得到。如果程序想要发出一个零长度的包，应该设定 TLEN 为 0，PCNT 位 1；
5、 对于 OUT 端点，因为在 IN 事务结束之前，程序不知道实际接收的数据大小，程序可将 TLEN 设定为 Rx FIFO 所支持的最大值；
6、 置位 USBFS_DIEPxCTL 或 USBFS_DOEPxCTL 寄存器 EPEN 位使能端点。

**端点除能顺序**

当USBFS_DIEPnCTL或USBFS_DOEPnCTL寄存器的EPEN位被清除时，程序可以在任何时候除能端点

**IN传输操作顺序：**

1、 初始化USBFS全局寄存器；
2、 初始化和使能IN端点；
3、 将数据包写入端点的Tx FIFO，每当数据包写入FIFO，USBFS减少USBFS_DIEPxLEN寄存器的TLEN域的数值，其减少的数值等于已写的数据包大小；
4、 当IN令牌接收后，USBFS发送数据包，在USB总线上的事务完成后，USBFS_DIEPxLEN寄存器的PCNT值减1。如果事务成功完成(接收到ACK握手包)，ACK标志被触发，或者，其他状态标志表示事务的结果；
5、 在一次传输的所有数据包都被成功发送，USBFS生成一个TF标志位以表明传输成功结束，除能相应IN端点。

**OUT传输操作顺序：**

1、 初始化USBFS全局寄存器；
2、 初始化和使能端点；
3、 当OUT令牌接收后，USBFS接收数据包或基于Rx FIFO状态和寄存器配置回复NAK握手包。如果事务成功完成(USBFS接收并保存数据到Rx FIFO，发送ACK握手包)，USBFS_DOEPxLEN寄存器的PCNT值减1。如果事务成功完成(接收到ACK握手包)，ACK标志被触发，或者，其他状态标志表示事务的结果；
4、 在一次传输的所有数据包都被成功接收，USBFS将TF状态条目压入Rx FIFO的最后的数据包的顶部，这样，软件在读取所有接收的数据包后，再读取TF状态条目。USBFS生成TF标志来指示传输成功结束。USBFS生成一个TF标志位以表明传输成功结束，除能相应OUT端点。

## 21.6. 中断

OTG 有两种中断：全局中断、唤醒中断。

全局中断是软件需要处理的主要中断，全局中断的标志位可在 USBFS_GINTF 寄存器读取，列举在*表 22-2. USBFS 全局中断*中。

**表 21-2. USBFS 全局中断**

| 中断标志 | 描述 | 运行模式 |
|---|---|---|
| SESIF | 会话中断 | 主机或设备模式 |
| DISCIF | 断开连接中断标志 | 主机模式 |
| IDPSC | ID 引脚状态变化 | 主机或设备模式 |
| PTXFEIF | 周期性 Tx FIFO 空中断标志 | 主机模式 |
| HCIF | 主机通道中断标志 | 主机模式 |
| HPIF | 主机端口中断 | 主机模式 |
| ISOONCIF/PXNCIF | 周期性传输未完成中断标志 / | 主机或设备模式 |

| 中断标志 | 描述 | 运行模式 |
| --- | --- | --- |
| | 同步OUT传输未完成中断标志 | |
| ISOINCIF | 同步 IN 传输未完成中断标志 | 设备模式 |
| OEPIF | OUT 端点中断标志 | 设备模式 |
| IEPIF | IN 端点中断标志 | 设备模式 |
| EOPFIF | 周期性帧尾中断标志 | 设备模式 |
| ISOOPDIF | 同步 OUT 丢包中断标志 | 设备模式 |
| ENUMF | 枚举完成 | 设备模式 |
| RST | USB 复位 | 设备模式 |
| SP | USB 挂起 | 设备模式 |
| ESP | 早挂起 | 设备模式 |
| GONAK | 全局 OUT NAK 有效 | 设备模式 |
| GNPINAK | 全局非周期 IN NAK 有效 | 设备模式 |
| NPTXFEIF | 非周期 Tx FIFO 空中断标志 | 主机模式 |
| RXFNEIF | Rx FIFO 非空中断标志 | 主机或设备模式 |
| SOF | 帧首 | 主机或设备模式 |
| OTGIF | OTG 中断标志 | 主机或设备模式 |
| MFIF | 模式错误中断标志 | 主机或设备模式 |

唤醒中断可以在 USBFS 处于挂起状态时触发，即使 USBFS 的时钟停止。寄存器 USBFS_GINTF 的位 WKUPIF 是唤醒源。

# 21.7. USBFS 寄存器

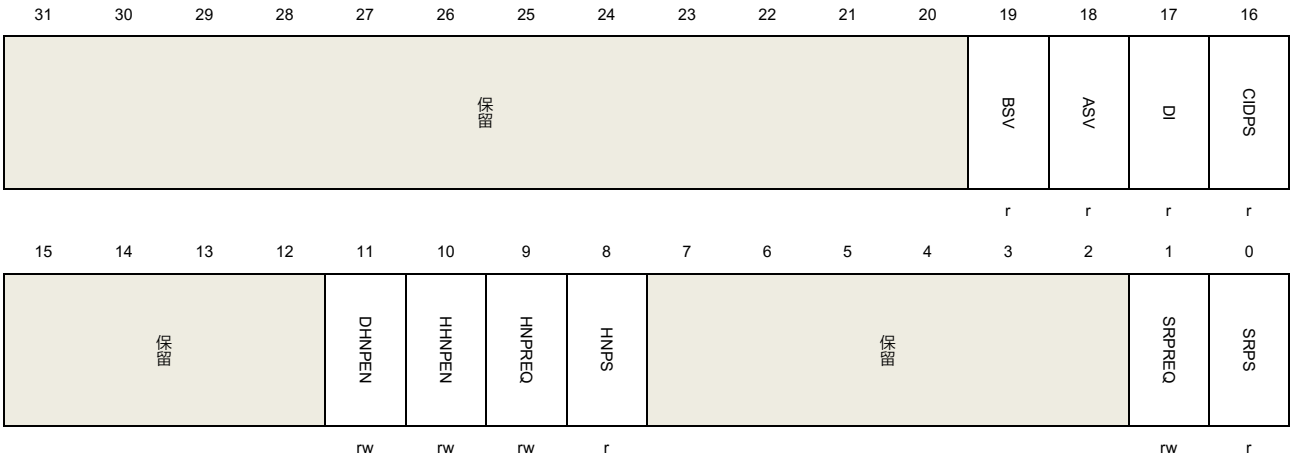
USBFS基地址：0x5000 0000

## 21.7.1. 全局控制与状态寄存器组

### 全局 OTG 控制和状态寄存器 (USBFS_GOTGCS)

地址偏移：0x0000
复位值：0x0000 0800

该寄存器只能按字(32位)访问




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:20 | 保留 | 必须保持复位值。 |
| 19 | BSV | B会话有效(在OTG协议中描述)<br>0：OTG B设备$V_{BUS}$电压水平低于$V_{BSESSVLD}$<br>1：OTG B设备$V_{BUS}$电压水平不低于$V_{BSESSVLD}$<br>**注意：**仅在OTB B设备模式下可访问 |
| 18 | ASV | A会话有效<br>A主机模式收发器状态<br>0：OTG A设备$V_{BUS}$电压水平低于$V_{ASESSVLD}$<br>1：OTG A设备$V_{BUS}$电压水平不低于$V_{ASESSVLD}$<br>在会话的开始，A设备默认是主机。<br>**注意：**仅在OTG A设备模式下可访问 |
| 17 | DI | 去抖动间隔<br>检测到连接的去抖动间隔。<br>0：当USB总线上发生插入和连接时，表示长去抖动间隔<br>1：当HNP协议中使用一个软连接时，指示短去抖动间隔 |

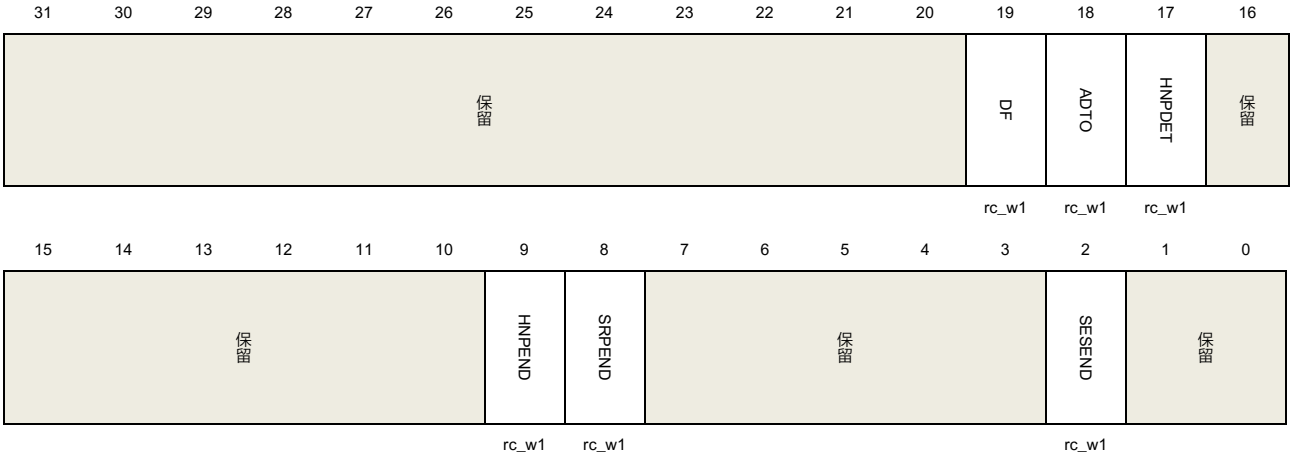
| | | |
|---|---|---|
| | | **注意：**仅在主机模式下可访问 |
| 16 | CIDPS | ID引脚状态<br>连接器ID引脚的电压水平<br>0：USBFS工作在A设备模式<br>1：USBFS工作在B设备模式<br>**注意：**在设备和主机模式下均可访问 |
| 15:12 | 保留 | 必须保持复位值。 |
| 11 | DHNPEN | 设备HNP使能<br>使能B设备HNP功能。如果该控制位清除，当应用置位USBFS_GOTGCS寄存器中的HNPREQ控制位c时，USBFS并不启动HNP协议。<br>0：HNP功能不使能<br>1：HNP功能使能<br>**注意：**仅在设备模式下访问 |
| 10 | HHNPEN | 主机HNP使能<br>使能A设备HNP功能。如果该控制位清除，USBFS不能够响应B设备的HNP请求。<br>0：HNP功能不使能<br>1：HNP功能使能<br>**注意：**仅在主机模式下访问 |
| 9 | HNPREQ | HNP请求<br>软件通过置位该控制位在USB总线上启动一个HNP。当USBFS_GOTGINTF寄存器中HNPEND控制位置位时，软件可以通过向该控制位写0或者清除USBFS_GOTGINTF寄存器中的HNPEND控制位来清除该控制位。<br>0：不发送HNP请求<br>1：发送HNP请求<br>**注意：**仅在设备模式下访问 |
| 8 | HNPS | HNP成功标志位<br>当HNP成功时，该标志位由内核置位。当HNPREQ置位时，该控制位被清除。<br>0：HNP失败<br>1：HNP成功<br>**注意：**仅在设备模式下访问 |
| 7:2 | 保留 | 必须保持复位值。 |
| 1 | SRPREQ | SRP请求<br>软件通过置位该控制位在USB总线上启动一个SRP会话请求。当USBFS_GOTGINTF寄存器中的SRPEND控制位置位时，软件可以通过向该控制位写0或者清除USBFS_GOTGINTF寄存器中的SRPEND控制位来清除该控制位。<br>0：没有会话请求<br>1：会话请求<br>**注意：**仅在设备模式下访问 |
| 0 | SRPS | SRP会话请求成功<br>当SRP会话请求成功时，该标志位由内核置位。当SRPREQ控制位被置位时，该标志 |

位被清除。
0：SRP会话请求失败
1：SRP会话请求成功
**注意：**仅在设备模式下访问

### 全局 OTG 中断状态寄存器 (USBFS_GOTGINTF)

地址偏移：0x0004
复位值：0x0000 0000

该寄存器只能按字(32位)访问



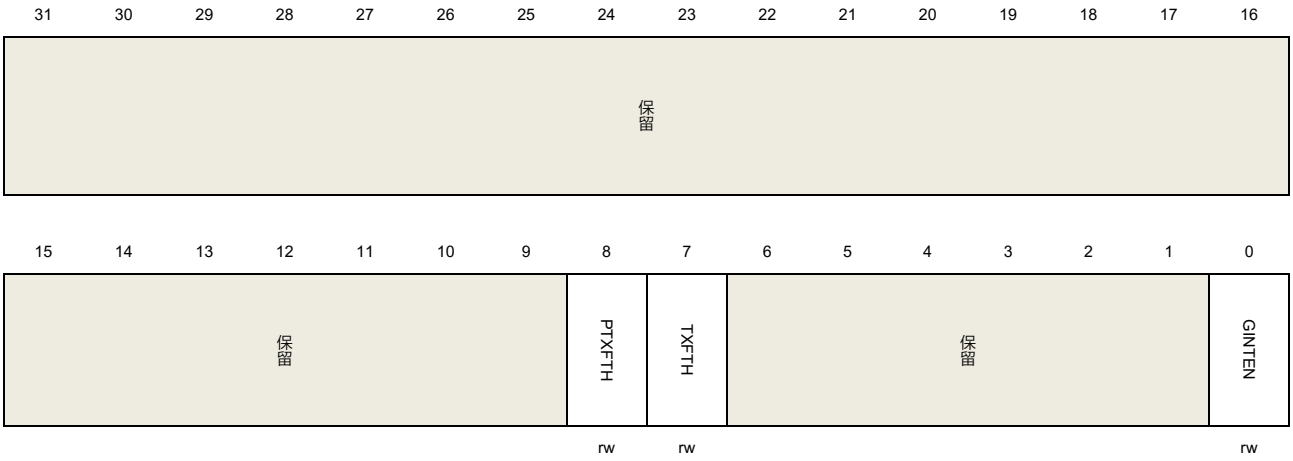

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:20 | 保留 | 必须保持复位值。 |
| 19 | DF | 去抖动完成<br>当设备连接去抖动完成时，USBFS置位该控制位<br>**注意：**仅在主机模式下可访问 |
| 18 | ADTO | A设备超时<br>当A设备等待B设备连接发生超时，USBFS置位该控制位<br>**注意：**在设备和主机模式下，均可访问 |
| 17 | HNPDET | 检测到主机协商请求<br>当A设备检测到一个HNP请求时，USBFS置位该标志位<br>**注意：**在设备和主机模式下，均可访问 |
| 16:10 | 保留 | 必须保持复位值。 |
| 9 | HNPEND | HNP结束<br>当一个HNP结束时，内核置位该标志位。软件应该读取USBFS_GOTGCS寄存器中HNPS标志位，以获取HNP结果。<br>**注意：**在设备和主机模式下，均可访问。 |
| 8 | SRPEND | SRPEND |

| | | |
|---|---|---|
| | | 当一个SRP结束时，内核置位该标志位。软件应该读取USBFS_GOTGCS寄存器中SRPS标志位，以获取SRP结果。<br>**注意：**在设备和主机模式下，均可访问。 |
| 7:3 | 保留 | 必须保持复位值。 |
| 2 | SESEND | 会话结束<br>当$V_{BUS}$电压低于Vb_ses_vld时，内核置位该标志位。 |
| 1:0 | 保留 | 必须保留复位值 |

### 全局 AHB 控制和状态寄存器 (USBFS_GAHBCS)

地址偏移：0x0008
复位值：0x0000 0000

该寄存器只能按字(32位)访问



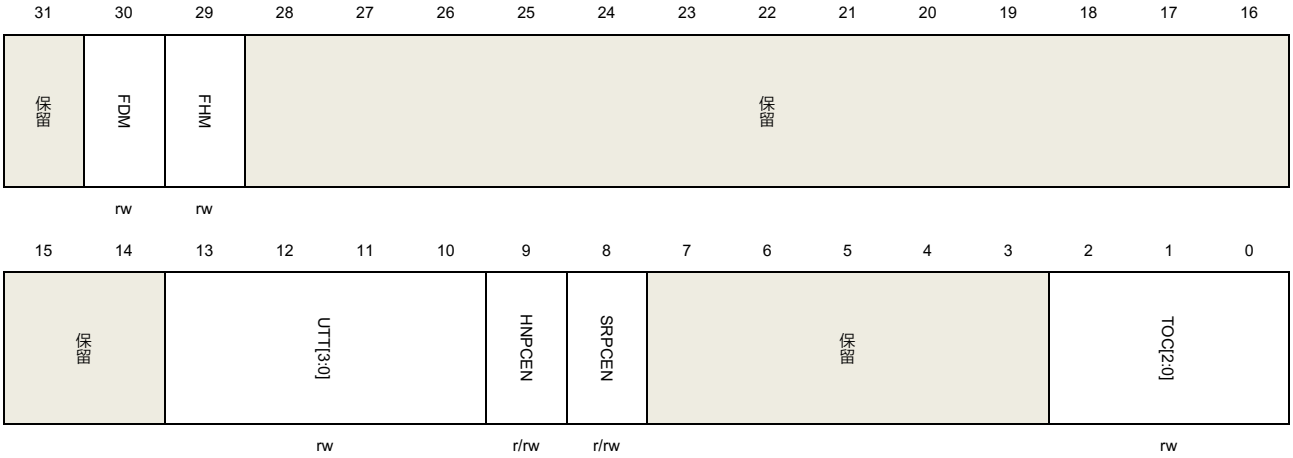

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:9 | 保留 | 必须保持复位值。 |
| 8 | PTXFTH | 周期性Tx FIFO阈值<br>0：当周期性发送FIFO半空时，将触发PTXFEIF标志位<br>1：当周期性发送FIFO全空时，将触发PTXFEIF标志位<br>**注意：**只在主机模式下访问 |
| 7 | TXFTH | Tx FIFO 阈值<br>**设备模式：**<br>0：当IN端点发送FIFO半空时，将触发TXFEIF标志位<br>1：当IN端点发送FIFO全空时，将触发TXFEIF标志位<br>**主机模式：**<br>0：当非周期性发送FIFO半空时，将触发NPTXFEIF标志位<br>1：当非周期性发送FIFO全空时，将触发NPTXFEIF标志位 |

| | | |
|---|---|---|
| 6:1 | 保留 | 必须保持复位值。 |
| 0 | GINTEN | 全局中断使能<br>0：全局中断不使能<br>1：全局中断使能<br>**注意：**在主机和设备模式下，均可访问 |

### 全局 USB 控制和状态寄存器 (USBFS_GUSBCS)

地址偏移：0x000C
复位值：0x0000 0A80

该寄存器只能按字(32位)访问




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | 保留 | 必须保持复位值。 |
| 30 | FDM | 强制设备模式<br>通过置位该控制位，可强制USB内核为设备模式，并且忽略USBFS ID引脚的输入状态<br>0：正常模式<br>1：设备模式<br>设置该控制位后，应用必须等待至少25ms，让修改产生作用。<br>**注意：**在设备和主机模式下，均可访问。 |
| 29 | FHM | 强制主机模式<br>通过置位该控制位，可强制USB内核为主机模式，并且忽略USBFS ID引脚的输入状态<br>0：正常模式<br>1：主机模式<br>设置该控制位后，应用必须等待至少25ms，让修改产生作用。<br>**注意：**在设备和主机模式下，均可访问。 |

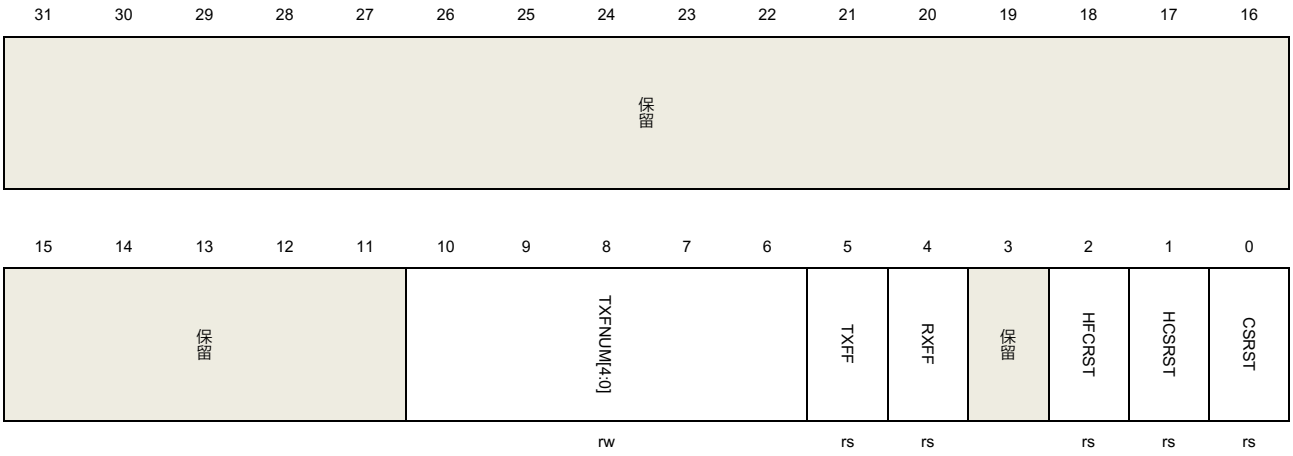
| | | |
|---|---|---|
| 28:14 | 保留 | 必须保持复位值。 |
| 13:10 | UTT[3:0] | USB运转时间<br>以物理时钟数来设定运转时间<br>**注意：**仅在设备模式下访问 |
| 9 | HNPCEN | HNP能力使能<br>控制HNP能力是否使能<br>0：HNP能力禁用<br>1：HNP能力使能<br>**注意：**在设备和主机模式下，均可访问。 |
| 8 | SRPCEN | SRP能力使能<br>控制SRP能力是否使能<br>0：SRP能力禁用<br>1：SRP能力使能<br>**注意：**在设备和主机模式下，均可访问。 |
| 7:3 | 保留 | 必须保持复位值。 |
| 2:0 | TOC[2:0] | 超时校准<br>当等待一个包时，USBFS需要使用USB2.0协议中需要的超时数值。应用可以使用TOC[2:0]增加该数值(以PHY时钟为单位)。PHY时钟频率为48MHz。 |

### 全局复位控制寄存器 (USBFS_GRSTCTL)

地址偏移：0x0010
复位值：0x8000 0000

应用通过该寄存器来复位内核的不同硬件特性。

该寄存器只能按字(32位)访问



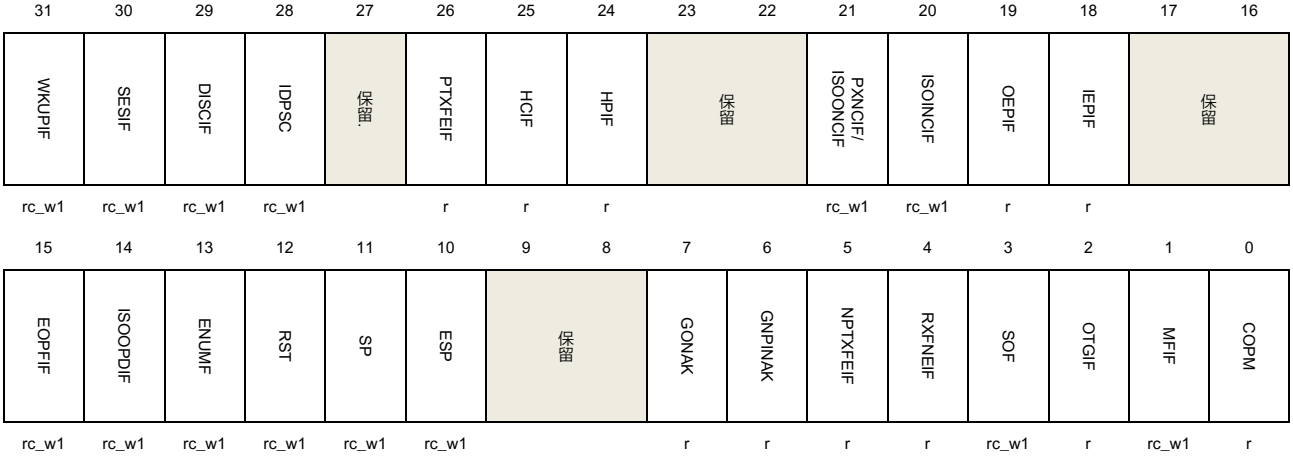

| 位/位域 | 名称 | 描述 |
|---|---|---|

| | | |
|---|---|---|
| 31:11 | 保留 | 必须保持复位值。 |
| 10:6 | TXFNUM[4:0] | Tx FIFO序号<br>当本寄存器中TXFF控制位置位时，该标志位决定那个Tx FIFO会被刷新<br>**主机模式：**<br>00000：仅非周期性Tx FIFO被刷新<br>00001：仅周期性Tx FIFO被刷新<br>1xxxx：周期性和非周期性Tx FIFO均被刷新<br>其他：没有数据被刷新<br>**设备模式：**<br>00000：仅Tx FIFO0被刷新<br>00001：仅Tx FIFO1被刷新<br>…<br>00011：仅Tx FIFO3被刷新<br>1xxxx：所有的Tx FIFO均被刷新<br>其他：没有数据被冲刷 |
| 5 | TXFF | Tx FIFO刷新控制位<br>应用通过置位该控制位来刷新Tx FIFO数据，并且TXFNUM[4:0]决定刷新的FIFO数目。当刷新完成后，硬件自动清除该控制位。置位该控制位后，应用程序应该等待该控制位清除，并且，在此之前USBFS不应有其他操作。<br>**注意：**在设备和主机模式下，均可访问 |
| 4 | RXFF | Rx FIFO刷新控制位<br>应用通过置位该控制位来刷新Rx FIFO数据。当刷新完成后，硬件自动清除该控制位。置位该控制位后，应用程序应该等待该控制位清除，并且，在此之前USBFS不应有其他操作。<br>**注意：**在设备和主机模式下，均可访问 |
| 3 | 保留 | 必须保持复位值。 |
| 2 | HFCRST | 主机帧计数器复位<br>应用通过置位该控制位来复位USBFS内的帧计数器。该控制位置位后，接下来SOF的帧计数器将变为0。当复位操作完成后，硬件自动清除该控制位。置位该控制位后，应用应该等待该控制位清除，并且，在此之前USBFS不应有其他操作。<br>**注意：**仅在主机模式下访问 |
| 1 | HCSRST | HCLK软件复位<br>应用通过置位该控制位来复位AHB时钟域电路<br>在复位操作完成后，硬件自动清除该控制位。置位该控制位后，应用程序应该等待该控制位清除，并且，在此之前USBFS不应有其他操作。<br>**注意：**在设备和主机模式下，均可访问 |
| 0 | CSRST | USB内核软件复位<br>复位AHB和USB时钟域电路，以及大多数的寄存器。 |

### 全局中断标志寄存器（USBFS_GINTF)

地址偏移：0x0014
复位值：0x0400 0021

该寄存器只能按字(32位)访问




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | WKUPIF | 唤醒中断标志位<br>当在USB总线上检测到一个恢复信号(在设备模式下)或者一个远程唤醒信号(在主机模式下)，硬件将置位该中断标志位。<br>**注意：**在设备和主机模式下，均可访问 |
| 30 | SESIF | 会话中断标志位<br>当在A设备模式下检测到一个SRP会话请求或在B设备模式下B设备的VBUS变为可用时，硬件将置位该中断标志位<br>**注意：**在设备和主机模式下，均可访问 |
| 29 | DISCIF | 断开中断标志位<br>当设备断开后，将触发该标志位。<br>**注意：**仅在主机模式下访问 |
| 28 | IDPSC | ID引脚状态改变中断标志位<br>当ID引脚状态改变时，内核将置位该标志位<br>**注意：**在设备和主机模式下，均可访问 |
| 27 | 保留 | 必须保持复位值。 |
| 26 | PTXFEIF | 周期性Tx FIFO空中断标志位<br>当周期性发送FIFO半空或全空时，将触发该标志位。空阈值由USBFS_GAHBCS寄存器中周期性Tx FIFO空等级控制位(PTXFTH)决定。<br>**注意：**仅在主机模式下访问 |
| 25 | HCIF | 主机通道中断标志位<br>当在主机模式下其中一个通道挂起一个中断时，USBFS将置位该标志位。软件应该 |

| | | |
|---|---|---|
| | | 首先读取USBFS_HACHINT寄存器以获取通道号，然后读取相应的USBFS_HCHxINTF寄存器以获取产生中断的通道标志位。当产生通道中断的独立通道标志位被清除后，该中断标志位将自动清除。<br>**注意：**仅在主机模式下访问 |
| 24 | HPIF | 主机端口中断标志位<br>当USBFS在主机模式下检测到端口状态改变时，USB内核将置位该标志位。软件应该读取USBFS_HPCSR寄存器以获取该中断源。当产生端口中断的标志被清除后，该中断标志位将自动清除。<br>**注意：**仅在主机模式下访问 |
| 23:22 | 保留 | 必须保持复位值。 |
| 21 | PXNCIF | 周期性传输未完成中断标志位<br>在当前帧内，当帧结束时，周期性传输未完成，USBFS将置位该标志位(主机模式)。 |
| | ISOONCIF | 同步OUT传输未完成中断标志位<br>在周期性帧结束时(由USBFS_DCFG寄存器的EOPFT控制位定义)，如果仍有同步OUT端点未完成传输，USBFS将置位该标志位(设备模式)。 |
| 20 | ISOINCIF | 同步IN传输未完成中断标志位<br>在周期性帧结束时(由USBFS_DCFG寄存器的EOPFT控制位定义)，如果仍有同步IN端点未完成传输，USBFS将置位该标志位(设备模式)。<br>**注意：**仅在设备模式下访问 |
| 19 | OEPIF | OUT端点中断标志位<br>当在设备模式下，其中一个OUT端点挂起一个中断时，USBFS将置位该中断标志位。软件应该首先读取USBFS_DAEPINT寄存器以获取设备号，然后读取相应的USBFS_ DOEPxINTF寄存器以获取产生中断的端点标志位。当产生中断的相应端点标志位被清除后，该中断标志位被自动清除。<br>**注意：**仅在设备模式下访问 |
| 18 | IEPIF | IN端点中断标志位<br>当在设备模式下，其中一个IN端点挂起一个中断时，USBFS将置位该标志位。软件应该首先读取USBFS_DAEPINT寄存器以获取设备号，然后读取相应的USBFS_DIEPxINTF寄存器以获取产生中断的端点标志位。当相应产生中断的端点标志位被清除后，该中断标志位被自动清除。 |
| 17:16 | 保留 | 必须保持复位值。 |
| 15 | EOPFIF | 周期性帧结束中断标志位<br>当一帧内USB总线时间已经达到USBFS_DCFG寄存器中EOPFT控制位所定义的数值时，USBFS将置位该中断标志位。<br>**注意：**仅在设备模式下访问 |
| 14 | ISOOPDIF | 同步OUT包丢失中断标志位<br>如果USBFS接收到一个同步OUT包，但是Rx FIFO没有足够的空间来接收该OUT包，USBFS将置位该标志位。<br>**注意：**仅在设备模式下访问 |

| | | |
|---|---|---|
| 13 | ENUMF | 枚举完成中断标志位<br>在速度枚举完成后，USBFS将置位该中断标志位。软件能够读取USBFS_DSTAT寄存器以获取当前设备速度。<br>**注意：**仅在设备模式下访问 |
| 12 | RST | USB复位中断标志位<br>当USBFS在USB总线上检测到一个USB复位信号后，USBFS将置位该中断标志位。<br>**注意：**仅在设备模式下访问 |
| 11 | SP | USB挂起中断标志位<br>当USBFS检测到USB总线空闲3ms并且进入挂起状态，USBFS将置位该中断标志位。<br>**注意：**仅在设备模式下访问 |
| 10 | ESP | 早期挂起中断标志位<br>当USBFS检测到USB总线空闲3ms时，USBFS将置位该中断标志位。 |
| 9:8 | 保留 | 必须保持复位值。 |
| 7 | GONAK | 全局OUT NAK有效标志位<br>软件能够向USBFS_DCTL寄存器的SGONAK控制位写1，并且USBFS将会在SGONAK写入有效后，置位GONAK标志位。<br>**注意：**仅在设备模式下可访问 |
| 6 | GNPINAK | 全局非周期性IN NAK有效标志位<br>软件能够向USBFS_DCTL寄存器中的SGINAK控制位写1，并且USBFS将会在SGINAK写入有效后，置位GNPINAK标志位<br>**注意：**仅在设备模式下可访问 |
| 5 | NPTXFEIF | 非周期性Tx FIFO空中断标志位<br>当非周期性Tx FIFO为半空或全空时，将置位该中断标志位。该阈值由USBFS_GAHBCS寄存器中的非周期Tx FIFO空等级控制位(TXFTH)决定。<br>**注意：**仅在主机模式下访问 |
| 4 | RXFNEIF | Rx FIFO非空中断标志位<br>当至少有一个包或状态条目在Rx FIFO中时，USBFS将置位该标志位。<br>**注意：**在主机和设备模式下，均可访问。 |
| 3 | SOF | 帧起始中断标志位<br>主机模式：<br>当准备在USB总线上发送一个SOF或保持有效信号，USBFS将置位该中断标志位。软件可以通过写1清除该中断标志位。<br>设备模式：<br>当USBFS接收到一个SOF令牌包后，USBFS置位该标志位。应用可以读取设备状态寄存器以获取当前帧号。软件可以通过写1清除该中断标志位。<br>**注意：**在设备和主机模式下，均可访问。 |
| 2 | OTGIF | OTG中断标志位<br>当USBFS_GOTGINTF寄存器中标志位产生一个中断时，USBFS置位该中断标志位。 |

| | | |
|---|---|---|
| | | 软件应该读取USBFS_GOTGINTF寄存器以获取产生该中断的信号源，当USBFS_GOTGINTF寄存器中产生该中断的标志位被清除后，该中断标志位也被自动清除。<br>**注意：**在设备和主机模式下，均可访问 |
| 1 | MFIF | 模式错误中断标志位<br>如果软件在设备模式下操作仅主机可访问的寄存器或者在主机模式下操作仅设备可访问的寄存器，USBFS将置位该中断标志位。这些错误操作不会产生作用。<br>**注意：**在主机和设备模式下，均可访问。 |
| 0 | COPM | 当前操作模式<br>0：设备模式<br>1：主机模式<br>**注意：**在主机和设备模式下，均可访问。 |

### 全局中断使能寄存器 (USBFS_GINTEN)

地址偏移：0x0018
复位值：0x0000 0000

这个寄存器同全局中断标志寄存器(USBFS_GINTF)一起工作来中断应用程序。当中断使能位被禁止后，相应的中断就不会产生。然而，相应的全局中断标志位依然会被置位。

该寄存器只能按字(32位)访问

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WKUPIE | SESIE | DISCIE | IDPSCIE | 保留 | PTXFEIE | HCIE | HPIE | 保留 | | PXNCIE/ISOONCIE | ISOINCIE | OEPIE | IEPIE | 保留 | |
| rw | rw | rw | rw | | rw | rw | r | | | rw | rw | rw | rw | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EOPFIE | ISOOPDIE | ENUMFIE | RSTIE | SPIE | ESPIE | 保留 | | GONAKIE | GNPINAKIE | NPTXFEIE | RXFNEIE | SOFIE | OTGIE | MFIE | 保留 |
| rw | rw | rw | rw | rw | rw | | | rw | rw | rw | rw | rw | rw | rw | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | WKUPIE | 唤醒中断使能<br>0：禁用唤醒中断<br>1：使能唤醒中断<br>**注意：**在主机和设备模式下，均可访问 |
| 30 | SESIE | 会话中断使能<br>0：禁用会话中断<br>1：使能会话中断 |

**注意：**在主机和设备模式下，均可访问

| | | |
|---|---|---|
| 29 | DISCIE | 断开中断使能<br>0：禁用断开中断<br>1：使能断开中断<br>**注意：**仅在设备模式下使用 |
| 28 | IDPSCIE | ID引脚状态改变中断使能<br>0：禁用连接器ID引脚状态中断<br>1：使能连接器ID引脚状态中断<br>**注意：**在主机和设备模式下，均可访问 |
| 27 | 保留 | 必须保持复位值。 |
| 26 | PTXFEIE | 周期性Tx FIFO空中断使能<br>0：禁用周期性Tx FIFO空中断<br>1：使能周期性Tx FIFO空中断<br>**注意：**仅在主机模式下访问 |
| 25 | HCIE | 主机通道中断使能<br>0：禁用主机通道中断<br>1：使能主机通道中断<br>**注意：**仅在主机模式下访问 |
| 24 | HPIE | 主机端口中断使能<br>0：禁止主机端口中断<br>1：使能主机端口中断<br>**注意：**仅在主机模式下访问 |
| 23:22 | 保留 | 必须保持复位值。 |
| 21 | PXNCIE | 周期性传输未完成中断使能<br>0：禁止周期性未完成传输中断<br>1：使能周期性未完成传输中断<br>**注意：**仅在主机模式下访问 |
| | ISOONCIE | 同步OUT传输未完成中断使能<br>0：禁止同步OUT传输未完成中断<br>1：使能同步OUT传输未完成中断<br>**注意：**仅在设备模式下访问 |
| 20 | ISOINCIE | 同步IN传输未完成中断使能<br>0：禁止同步IN传输未完成中断<br>1：使能同步IN传输未完成中断<br>**注意：**仅在设备模式下访问 |
| 19 | OEPIE | OUT端点中断使能<br>0：禁止OUT端点中断<br>1：使能OUT端点中断 |

| | | |
|---|---|---|
| | | **注意：**仅在设备模式下访问 |
| 18 | IEPIE | IN端点中断使能<br>0：禁止IN端点中断<br>1：使能IN端点中断<br>**注意：**仅在设备模式下访问 |
| 17:16 | 保留 | 必须保持复位值。 |
| 15 | EOPFIE | 周期性帧结束中断使能<br>0：禁止周期性帧结束中断<br>1：使能周期性帧结束中断<br>**注意：**仅在设备模式下访问 |
| 14 | ISOOPDIE | 同步OUT包丢失中断使能<br>0：禁止同步OUT包丢失中断<br>1：使能同步OUT包丢失中断<br>**注意：**仅在设备模式下访问 |
| 13 | ENUMFIE | 枚举完成中断使能<br>0：禁止枚举完成中断<br>1：使能枚举完成中断<br>**注意：**仅在设备模式下访问 |
| 12 | RSTIE | USB复位中断使能<br>0：禁止USB复位中断<br>1：使能USB复位中断<br>**注意：**仅在设备模式下访问 |
| 11 | SPIE | USB挂起中断使能<br>0：禁止USB挂起中断<br>1：使能USB挂起中断<br>**注意：**仅在设备模式下访问 |
| 10 | ESPIE | 早期挂起中断使能<br>0：禁止早期挂起中断<br>1：使能早期挂起中断<br>**注意：**仅在设备模式下访问 |
| 9:8 | 保留 | 必须保持复位值。 |
| 7 | GONAKIE | 全局OUT NAK有效中断使能<br>0：禁止全局OUT NAK有效中断<br>1：使能全局OUT NAK有效中断<br>**注意：**仅在设备模式下访问 |
| 6 | GNPINAKIE | 全局非周期性IN NAK有效中断使能<br>0：禁止全局非周期性IN NAK有效中断<br>1：使能全局非周期性IN NAK有效中断 |

**注意：**仅在设备模式下访问

| | | |
|---|---|---|
| 5 | NPTXFEIE | 非周期性发送FIFO空中断使能<br>0：禁止非周期性发送FIFO空中断<br>1：使能非周期性发送FIFO空中断<br>**注意：**仅在主机模式下访问 |
| 4 | RXFNEIE | 接收FIFO非空中断使能<br>0：禁止接收FIFO非空中断<br>1：使能接收FIFO非空中断<br>**注意：**在设备模式与主机模式下，均可访问 |
| 3 | SOFIE | 帧首中断使能<br>0：禁止帧首中断<br>1：使能帧首中断<br>**注意：**在设备模式下与主机模式下，均可访问 |
| 2 | OTGIE | OTG中断使能<br>0：禁止OTG中断<br>1：使能OTG中断<br>**注意：**在设备模式下与主机模式下，均可访问 |
| 1 | MFIE | 模式错误中断使能<br>0：禁止模式错误中断<br>1：使能模式错误中断<br>**注意：**在设备模式下与主机模式下，均可访问 |
| 0 | 保留 | 必须保持复位值。 |

### 全局接收状态读取/弹出寄存器(USBFS_GRSTATR/USBFS_GRSTATP)

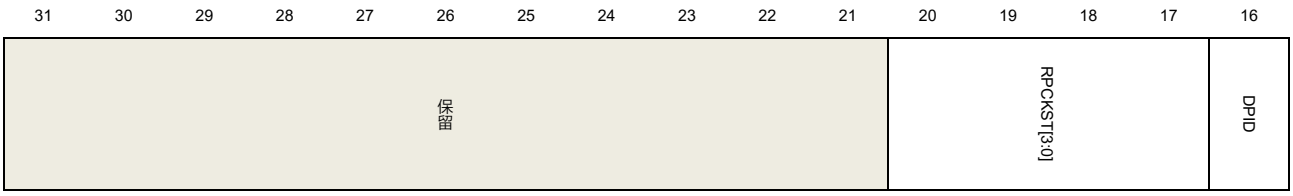
读地址偏移：0x001C
弹出地址偏移：0x0020
复位值：0x0000 0000

对接收状态读寄存器的读操作，将返回接收FIFO中顶部的条目。对接收状态弹出寄存器的读操作，将弹出Rx FIFO的顶部条目。

在主机模式和设备模式下，Rx FIFO中的条目具有不同的含义。当全局中断标志寄存器(USBFS_GINTF)中的接收FIFO非空中断标志位(RXFNEIF)置位后，软件应该读取该寄存器。

该寄存器只能按字(32位)访问

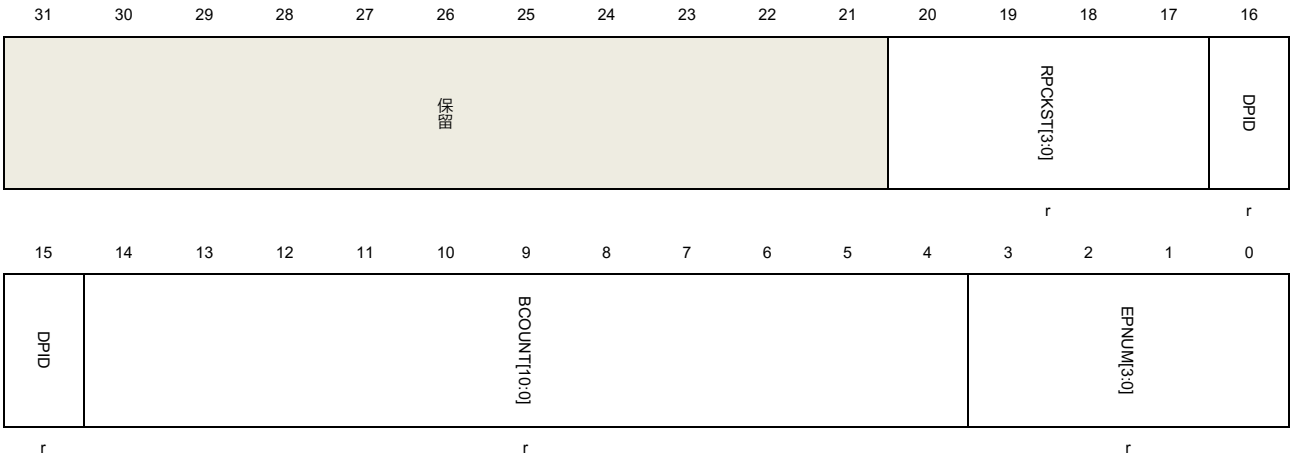
**主机模式：**






| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:21 | 保留 | 必须保持复位值。 |
| 20:17 | RPCKST[3:0] | 接收包状态<br>0010：接收到IN数据包<br>0011：IN传输完成(如果取出，触发一个中断)<br>0101：数据翻转错误(如果取出，触发一个中断)<br>0111：通道中止(如果取出，触发一个中断)<br>其他：保留 |
| 16:15 | DPID[1:0] | 数据PID<br>接收包的数据PID<br>00：DATA0<br>10：DATA1<br>其他：保留 |
| 14:4 | BCOUNT[10:0] | 字节数<br>接收IN数据包字节数。 |
| 3:0 | CNUM[3:0] | 通道数<br>当前接收包所属通道编号。 |

**设备模式：**



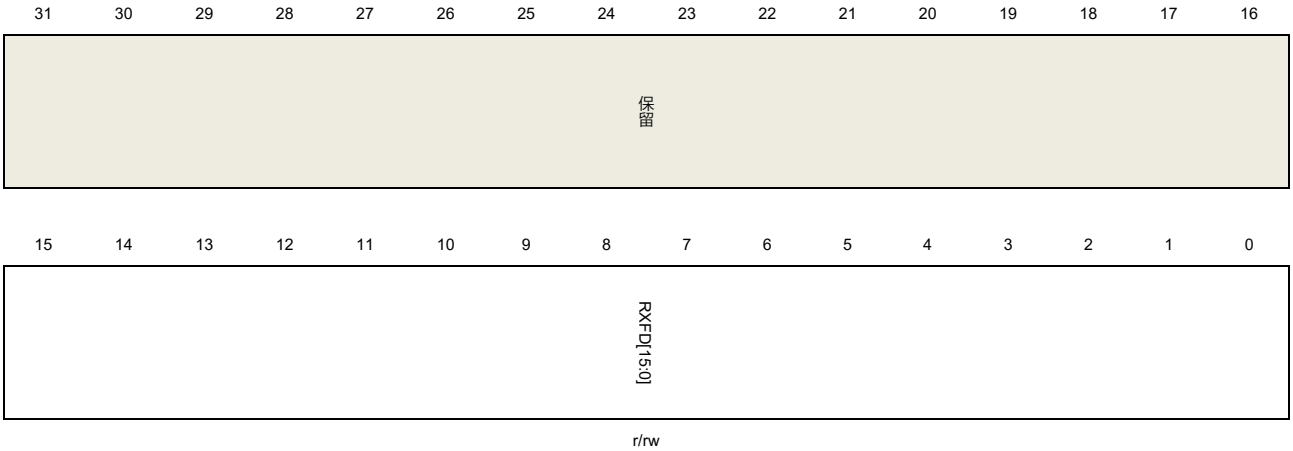

| 位/位域 | 名称 | 描述 |
|---|---|---|

| | | |
|---|---|---|
| 31:21 | 保留 | 必须保持复位值。 |
| 20:17 | RPCKST[3:0] | 接收包状态<br>0001：全局OUT NAK(产生一个中断)<br>0010：接收到OUT数据包<br>0011：OUT传输完成(产生一个中断)<br>0100：SETUP传输完成(产生一个中断)<br>0110：接收到SETUP数据包<br>其他：保留 |
| 16:15 | DPID[1:0] | 数据PID<br>接收到OUT数据包的数据PID<br>00：DATA0<br>10：DATA1<br>其他：保留 |
| 14:4 | BCOUNT[10:0] | 字节数<br>接收数据包的字节数 |
| 3:0 | EPNUM[3:0] | 端点号<br>当前接收包所属端点编号 |

### 全局接收 FIFO 长度寄存器 (USBFS_GRFLEN)

地址偏移：0x0024
复位值：0x0000 0200

该寄存器只能按字(32位)访问




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15:0 | RXFD[15:0] | Rx FIFO深度<br>以32位字计数 |

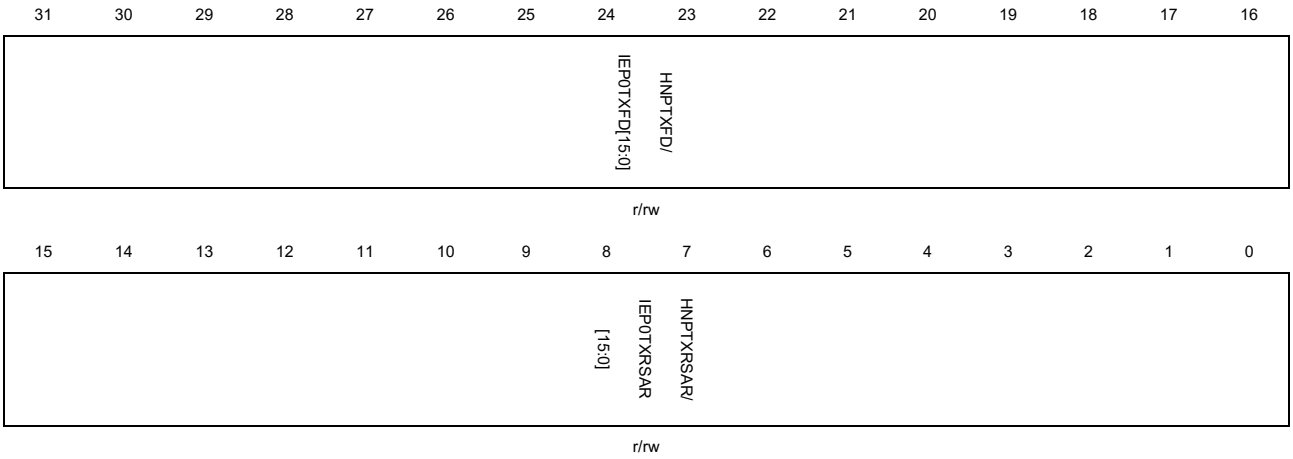
1<=RXFD<=1024

### 主机非周期性发送 FIFO 长度寄存器/设备 IN 端点 0 发送 FIFO 长度寄存器 (USBFS_HNPTFLEN_DIEP0TFLEN)

地址偏移：0x0028

复位值：0x0200 0200

该寄存器只能按字(32位)访问




**主机模式：**

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | HNPTXFD[15:0] | 主机非周期性Tx FIFO深度<br>以32位字计数<br>1<=HNPTXFD<=1024 |
| 15:0 | HNPTXRSAR[15:0] | 主机非周期性Tx FIFO RAM起始地址<br>非周期性发送FIFO RAM的起始地址以32位字计数 |

**设备模式：**

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | IEP0TXFD[15:0] | 输入端点0 Tx FIFO深度<br>以32位字计数<br>16<=IEP0TXFD<=140 |
| 15:0 | IEP0TXRSAR[15:0] | 输入端点0 TX FIFO RAM起始地址<br>端点0发送FIFO RAM的起始地址以32位字计数 |

### 主机非周期性发送 FIFO/队列状态寄存器 (USBFS_HNPTFQSTAT)

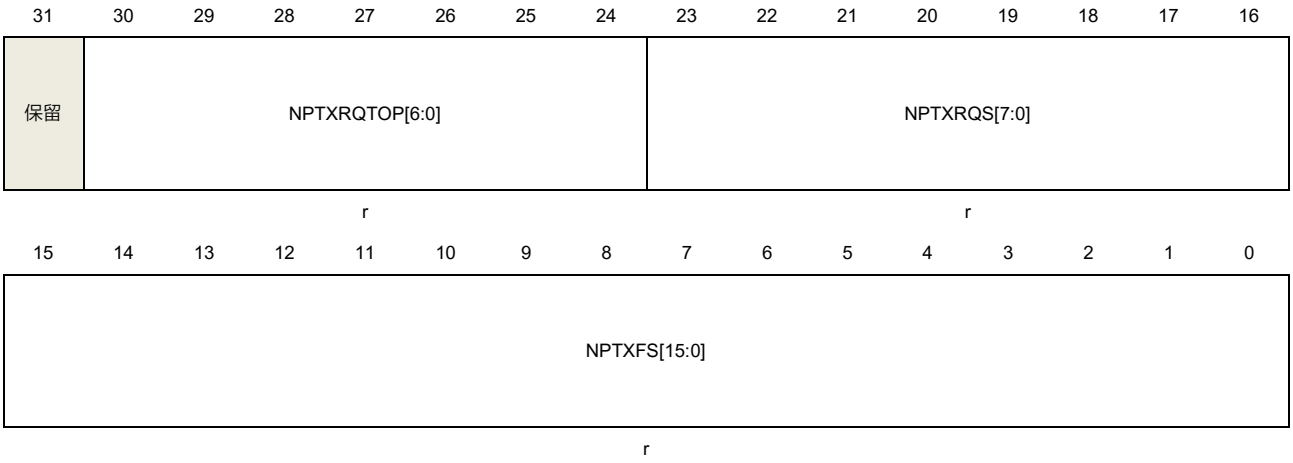
地址偏移：0x002C

复位值：0x0008 0200

该寄存器反映了非周期性Tx FIFO和请求队列的当前状态。请求队列包括主机模式下的IN、OUT或其他请求条目。

**注意：**在设备模式下，该寄存器不可用。

该寄存器只能按字(32位)访问




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | 保留 | 必须保持复位值。 |
| 30:24 | NPTXRQTOP[6:0] | 非周期性发送请求队列的顶部条目<br>在非周期性传输请求队列中的条目。<br>位30:27：通道号<br>位26:25：<br>- 00：IN/OUT令牌<br>- 01：0长度OUT包<br>- 11：通道中止请求<br>位24：结束标志位，表明所选通道的最后一个条目 |
| 23:16 | NPTXRQS[7:0] | 非周期性发送请求队列空间<br>非周期性请求队列的剩余空间<br>0：请求队列满<br>1：1个条目<br>2：2个条目<br>…<br>n：n个条目 (0<=n<=8)<br>其他：保留 |
| 15:0 | NPTXFS[15:0] | 非周期性Tx FIFO空间<br>非周期性发送FIFO剩余空间<br>以32位字计数<br>0：非周期性Tx FIFO满<br>1：1个字<br>2：2个字 |

…

n：n个字(0<=n<=NPTXFD)

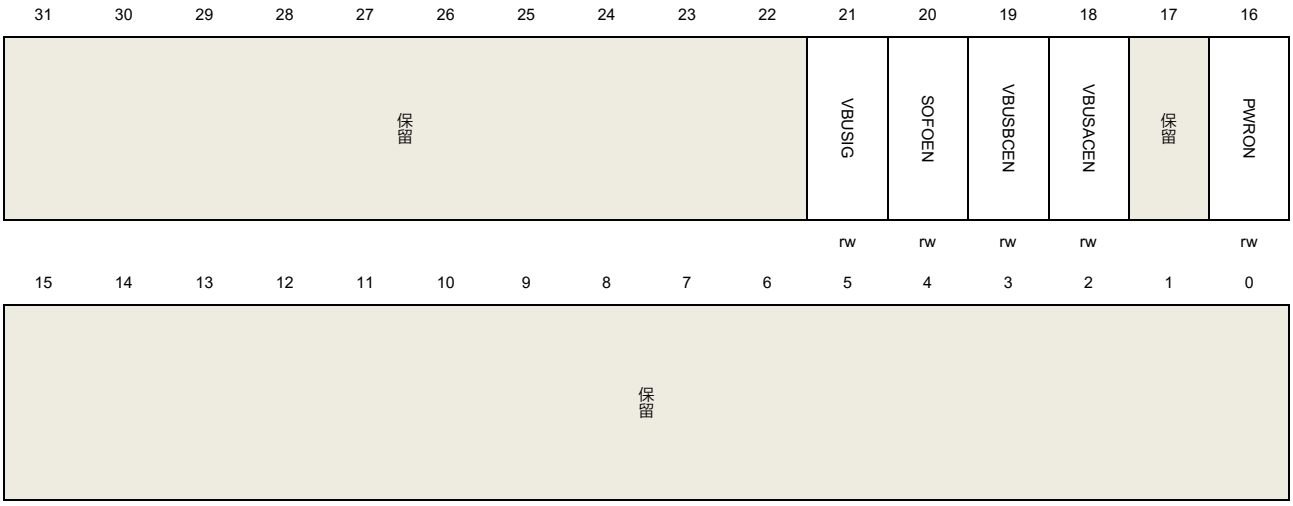
其他：保留

### 全局内核配置寄存器 (USBFS_GCCFG)

地址偏移：0x0038

复位值：0x0000 0000

该寄存器只能按字(32位)访问




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:22 | 保留 | 必须保持复位值。 |
| 21 | VBUSIG | VBUS忽略<br>当该控制位被置位，USBFS并不监测$V_{BUS}$引脚电压，并且认为在主机和设备模式下，$V_{BUS}$电压一直有效，然后可释放$V_{BUS}$引脚作为其他用途。<br>0：$V_{BUS}$不被忽略<br>1：$V_{BUS}$被忽略，并认为$V_{BUS}$电压一直有效 |
| 20 | SOFOEN | SOF输出使能<br>0：SOF脉冲输出禁止<br>1：SOF脉冲输出使能 |
| 19 | VBUSBCEN | $V_{BUS}$ B设备比较器使能<br>0：$V_{BUS}$ B设备比较器禁止<br>1：$V_{BUS}$ B设备比较器使能 |
| 18 | VBUSACEN | $V_{BUS}$ A设备比较器使能<br>0：$V_{BUS}$ A设备比较器禁止<br>1：$V_{BUS}$ A设备比较器使能 |

| | | |
|---|---|---|
| 17 | 保留 | 必须保持复位值。 |
| 16 | PWRON | 上电<br>该控制位为内部嵌入式全速PHY的电源开关<br>0：嵌入式全速PHY掉电<br>1：嵌入式全速PHY上电 |
| 15:0 | 保留 | 必须保持复位值。 |

### 内核 ID 寄存器 (USBFS_CID)

地址偏移：0x003C
复位值：0x0000 1000

该寄存器包含产品ID

该寄存器只能按字(32位)访问

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

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

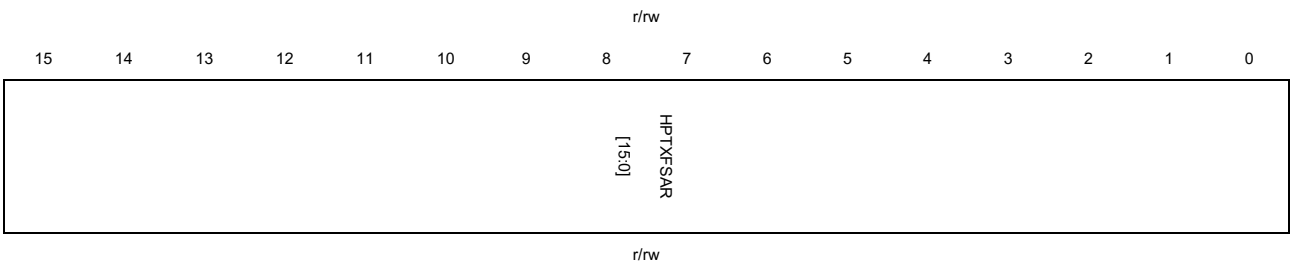
| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | CID[31:0] | 内核ID<br>软件能够写入或读取该域值，并利用该域值为应用产生一个唯一ID。 |

### 主机周期性发送 FIFO 长度寄存器 (USBFS_HPTFLEN)

地址偏移：0x0100
复位值：0x0200 0600

该寄存器只能按字(32位)访问

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



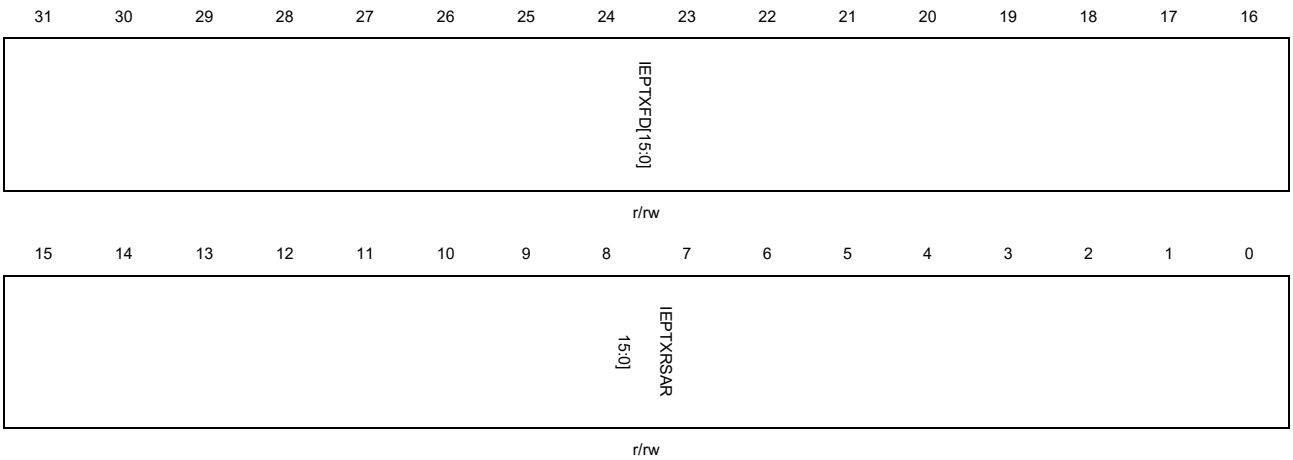

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | HPTXFD[15:0] | 主机周期性Tx FIFO深度<br>以32位字计数<br>1<=HPTXFD<=1024 |
| 15:0 | HPTXFSAR[15:0] | 主机周期性Tx FIFO RAM起始地址<br>主机周期性发送FIFO RAM起始地址以32位字计数 |

### 设备 IN 端点发送 FIFO 长度寄存器 (USBFS_DIEPxTFLEN) (x = 1..3，其中 x 为 FIFO 编号)

地址偏移：0x0104 + (FIFO_number - 1) × 0x04

复位值：0x0200 0400

该寄存器只能按字(32位)访问




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | IEPTXFD[15:0] | IN端点Tx FIFO深度<br>以32位字计数<br>1<=HPTXFD<=1024 |
| 15:0 | IEPTXRSAR[15:0] | IN端点Tx FIFO RAM起始地址 |

以32位字为单位的IN端点发送FIFOx起始地址

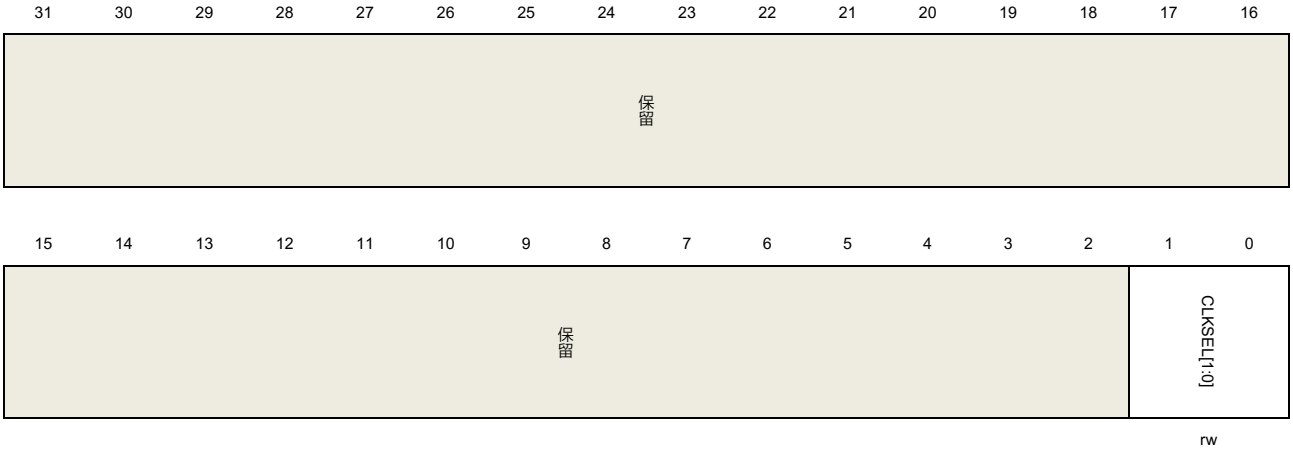
## 21.7.2. 主机控制和状态寄存器

### 主机控制寄存器 (USBFS_HCTL)

地址偏移：0x0400
复位值：0x0000 0000

在主机模式下，上电后，该寄存器配置USB内核。主机初始化后，无需修改。

该寄存器只能按字(32位)访问



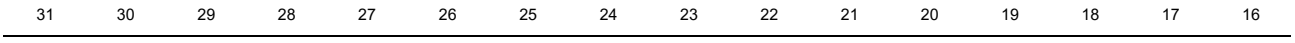

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:2 | 保留 | 必须保持复位值。 |
| 1:0 | CLKSEL[1:0] | USB时钟选择<br>01：48MHz时钟<br>其他：保留 |

### 主机帧间隔寄存器 (USBFS_HFT)

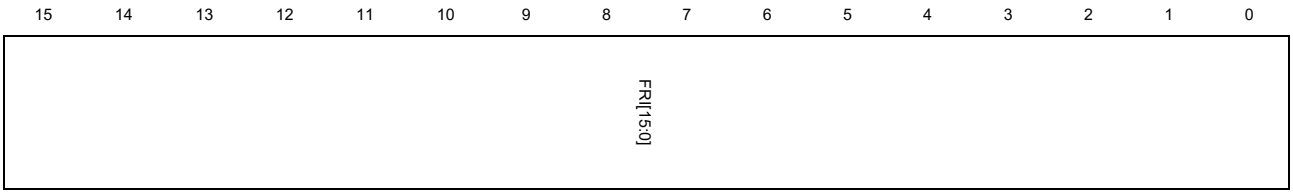
地址偏移：0x0404
复位值：0x0000 BB80

当USBFS控制器正在枚举USB设备时，该寄存器设置帧间隔。

该寄存器只能按字(32位)访问





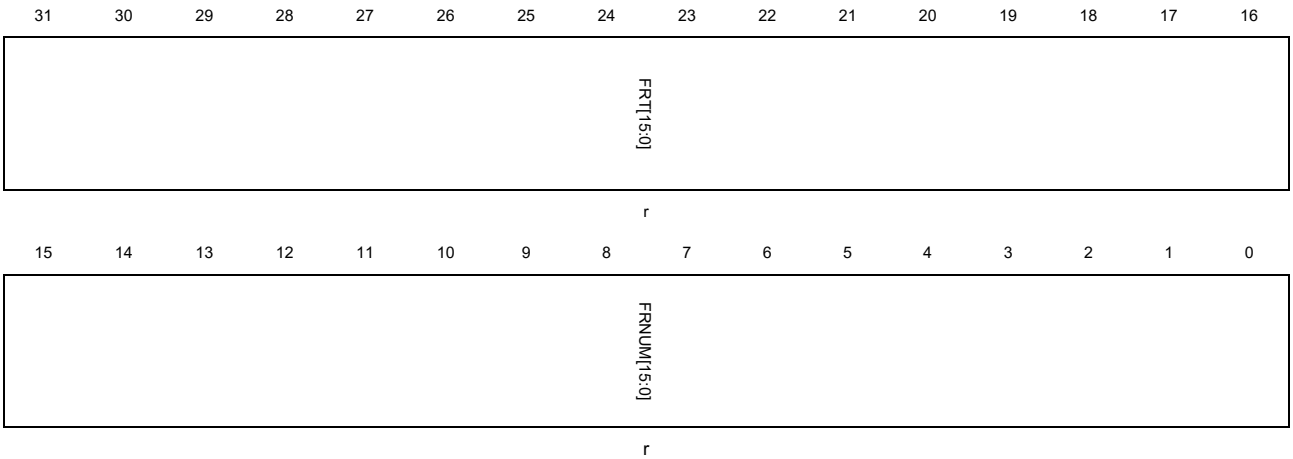




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15:0 | FRI[15:0] | 帧间隔<br>该值描述了以PHY时钟为单位的帧周期。每次端口复位操作后，端口被使能，USBFS根据当前速度采用一个固有值，并且软件可以向该位域写值以改变该固有值。该值需要采用以下描述的频率来进行计算：<br>全速：48MHz<br>低速：6MHz |

### 主机帧信息保持寄存器 (USBFS_HFINFR)

地址偏移：0x0408
复位值：0xBB80 0000

该寄存器只能按字(32位)访问




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | FRT[15:0] | 帧剩余时间<br>该位域以PHY时钟为单位反映了当前帧剩余时间。 |

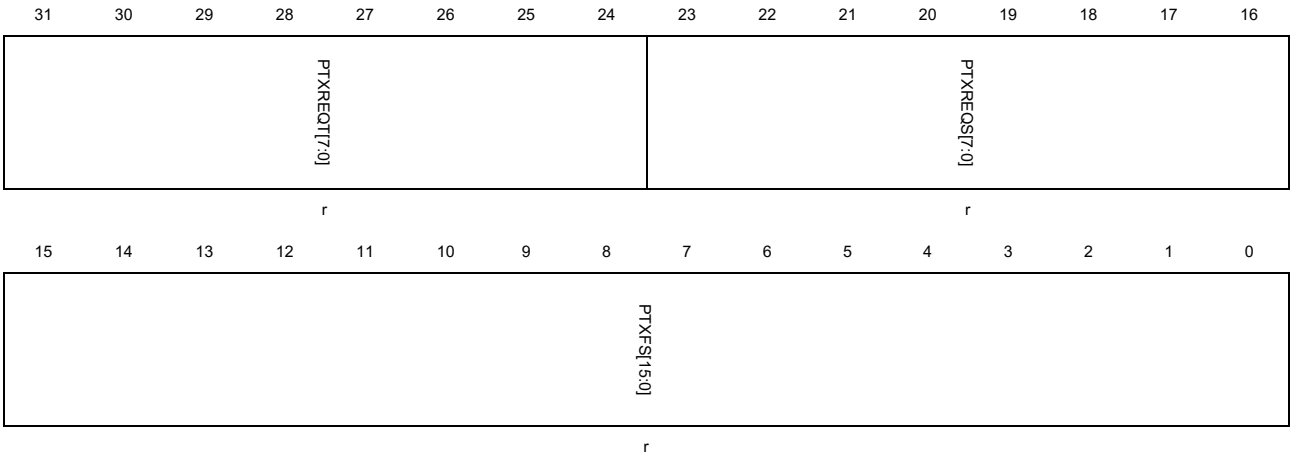
| | | |
|---|---|---|
| 15:0 | FRNUM[15:0] | 帧号<br>该位域反映了当前帧的帧号，当其增加到0x3FFF后，其值变为0。 |

### 主机周期性发送 FIFO/队列状态寄存器 (USBFS_HPTFQSTAT)

地址偏移：0x0410
复位值：0x0008 0200

该寄存器反映了主机周期性Tx FIFO和请求队列的当前状态。请求队列包括在主机模式下的IN、OUT或其他请求条目。

该寄存器只能按字(32位)访问




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:24 | PTXREQT[7:0] | 周期性发送请求队列的顶部条目<br>在周期性发送请求队列中的条目<br>位30:27：通道号<br>位26:25：<br>- 00：IN/OUT令牌<br>- 01：0长度OUT包<br>- 11：通道中止请求<br>位24：中止标志，指示所选通道的最后一个条目 |
| 23:16 | PTXREQS[7:0] | 周期性发送请求队列空间<br>周期性发送请求队列剩余空间<br>0：请求队列满<br>1：1个条目<br>2：2个条目<br>…<br>n：n个条目 (0<=n<=8)<br>其他：保留 |
| 15:0 | PTXFS[15:0] | 周期性发送FIFO空间 |

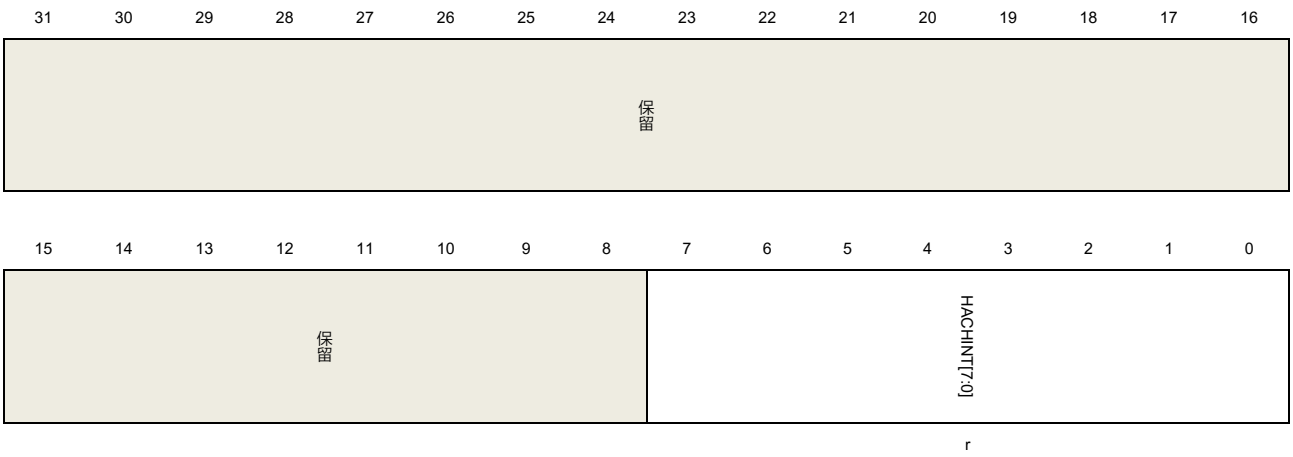
周期性发送FIFO剩余空间
以32位字计数
0：周期性发送FIFO满
1：1个字
2：2个字
…
n：n个字(0<=n<=PTXFD))
其他：保留

### 主机所有通道中断寄存器 (USBFS_HACHINT)

地址偏移：0x0414
复位值：0x0000 0000

当触发一个通道中断时，USBFS在该寄存器中置位相应的位，并且软件可以读取该寄存器以获取产生中断的通道。

该寄存器只能按字(32位)访问



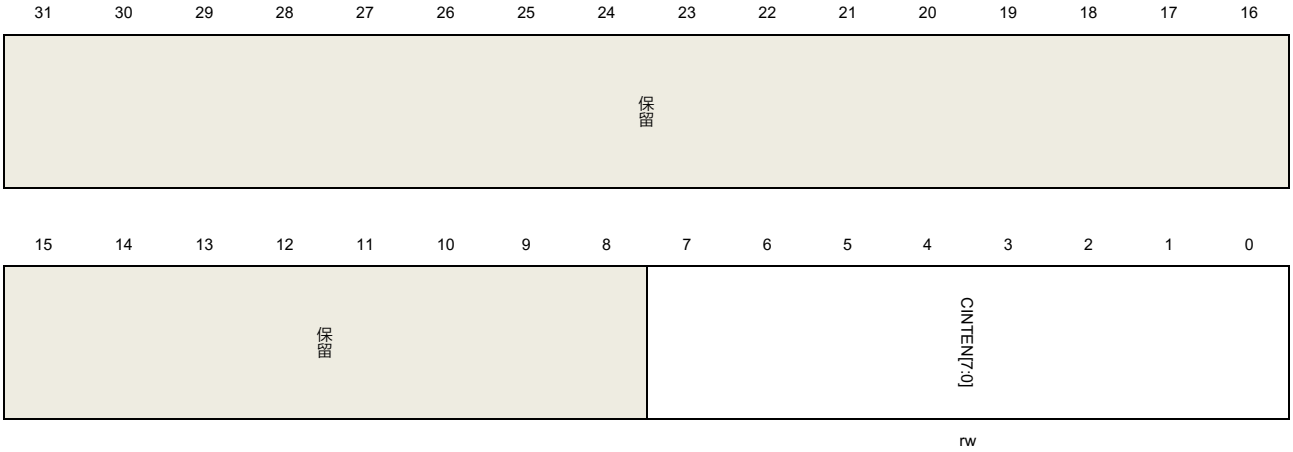

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 必须保持复位值。 |
| 7:0 | HACHINT[7:0] | 主机所有通道中断<br>每一位表示一个通道：位0代表通道0，位7表示通道7。 |

### 主机所有通道中断使能寄存器 (USBFS_HACHINTEN)

地址偏移：0x0418
复位值：0x0000 0000

软件可以使用该寄存器使能或禁用一个通道的中断。只有当该寄存器中相应通道的中断使能控制位被置位，USBFS_GINTF寄存器中的通道中断标志位HCIF标志位才可产生。

该寄存器只能按字(32位)访问



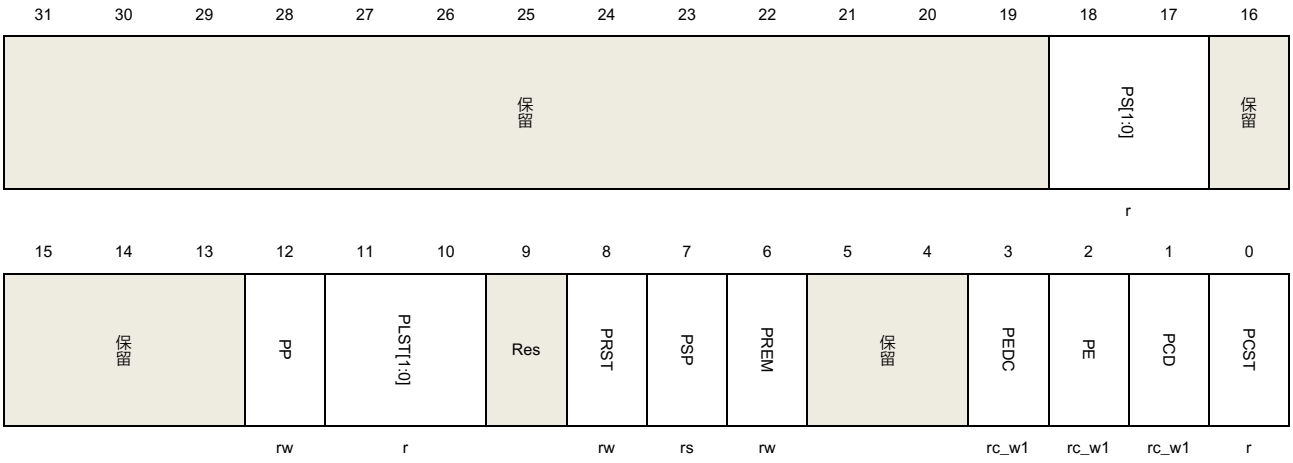

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 必须保持复位值。 |
| 7:0 | CINTEN[7:0] | 通道中断使能<br>0：禁用通道n中断<br>1：使能通道n中断<br>每一位表示一个通道：位0代表通道0，位7代表通道7 |

### 主机端口控制和状态寄存器 (USBFS_HPCS)

地址偏移：0x0440
复位值：0x0000 0000

该寄存器控制端口行为，并且也包含一些反映端口状态的标志位。如果本寄存器中的PRST、PEDC和PCD标志位被USBFS置位的话，USBFS_GINTF寄存器中的HPIF标志位会被置位。

该寄存器只能按字(32位)访问



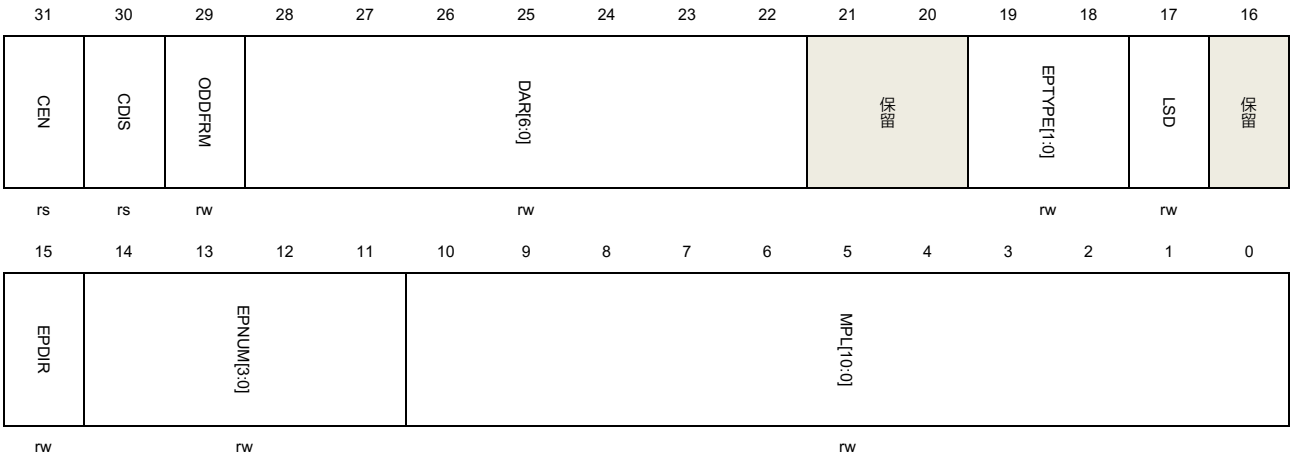
| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:19 | 保留 | 必须保持复位值。 |
| 18:17 | PS[1:0] | 端口速度<br>反映连接到该端口的设备的枚举速度。<br>01：全速<br>10：低速<br>其他：保留 |
| 16:13 | 保留 | 必须保持复位值。 |
| 12 | PP | 端口供电<br>在端口被使用后，该控制位应该被置位。由于USBFS不具有电源供应能力，它只能使用该控制位以获取该端口是否在供电状态。软件应该在设置该控制位之前，保证在$V_{BUS}$引脚上具有电源供应。<br>0：端口掉电<br>1：端口供电 |
| 11:10 | PLST[1:0] | 端口线状态<br>反映USB数据线当前状态。<br>位10：DP线状态<br>位11：DM线状态 |
| 9 | 保留 | 必须保持复位值。 |
| 8 | PRST | 端口复位<br>应用通过设置该控制位以在USB端口上启动一个复位信号。当应用希望停止复位信号时，应用应该清除该控制位。<br>0：端口不在复位状态<br>1：端口处于复位状态 |
| 7 | PSP | 端口挂起<br>应用设置该控制位，从而使端口进入挂起状态。当该控制位被置位后，端口停止发送SOF令牌包。该控制位只能够通过以下操作清除。<br>– 置位该寄存器中的PRST控制位<br>– 置位该寄存器中的PREM控制位<br>– 检测到一个远程唤醒信号<br>– 检测到一个设备断开<br>0：端口不在挂起状态<br>1：端口处于挂起状态 |
| 6 | PREM | 端口恢复<br>应用通过置位该控制位以在USB端口上启动一个恢复信号。当应用希望停止恢复信号时，应用可以清除该控制位。<br>0：无恢复驱动<br>1：恢复驱动 |
| 5:4 | 保留 | 必须保持复位值。 |

| | | |
|---|---|---|
| 3 | PEDC | 端口使能/禁止变化<br>当该寄存器中的位2状态变化时，USB内核置位该标志位。 |
| 2 | PE | 端口使能<br>当USB复位信号完成后，USBFS自动置位该位，并且该位不可由软件置位。<br>该位可通过以下事件清除：<br>– 一个断开状态<br>– 软件清除该位<br>0：端口禁止<br>1：端口使能 |
| 1 | PCD | 端口连接检测<br>当检测到设备连接时，USBFS置位该标志位。可通过向该位写1清除该标志位。 |
| 0 | PCST | 端口连接状态<br>0：设备没有连接到该端口<br>1：设备连接到该端口 |

### 主机通道 x 控制寄存器 (USBFS_HCHxCTL) (x = 0…7，其中 x 为通道号)

地址偏移：0x0500 +(通道号 × 0x20)
复位值：0x0000 0000

该寄存器只能按字(32位)访问




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | CEN | 通道使能<br>由应用设置，并且由USBFS清除<br>0：通道禁止<br>1：通道使能<br>软件应该遵循操作指南来禁用或者使能一个通道。 |
| 30 | CDIS | 通道禁止<br>软件可以置位该控制位，来从处理事务中禁用该通道。软件应该遵循操作指南来禁用 |

| | | |
|---|---|---|
| | | 或者使能一个通道。 |
| 29 | ODDFRM | 奇偶帧控制<br>对于周期性传输(中断或同步传输)，该位控制通道事务是在奇数帧还是偶数帧中处理。<br>0：偶数帧<br>1：奇数帧 |
| 28:22 | DAR[6:0] | 设备地址<br>与该通道通信的USB设备地址。 |
| 21:20 | 保留 | 必须保持复位值。 |
| 19:18 | EPTYPE[1:0] | 端点类型<br>与该通道通信的端点的传输类型<br>00：控制<br>01：同步<br>10：批量<br>11：中断 |
| 17 | LSD | 低速设备<br>与该通道通信的设备是一个低速设备。 |
| 16 | 保留 | 必须保持复位值。 |
| 15 | EPDIR | 端点方向<br>与该通道通信的端点的传输方向<br>0：OUT<br>1：IN |
| 14:11 | EPNUM[3:0] | 端点号<br>与该通道通信的端点号 |
| 10:0 | MPL[10:0] | 最大包长<br>目标端点的最大包长 |

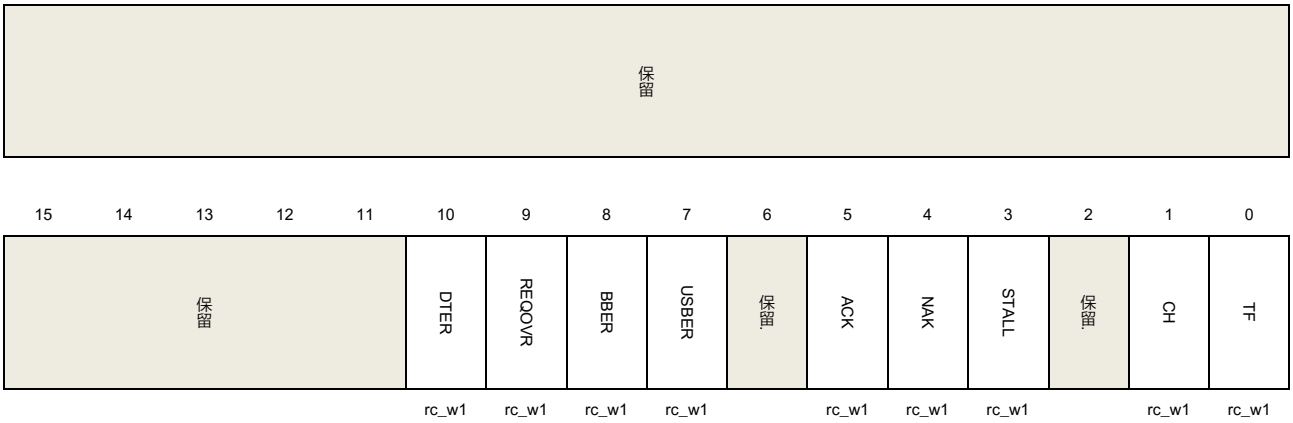
### 主机通道 x 中断标志寄存器 (USBFS_HCHxINTF) (x = 0…7，其中 x = 通道号)

地址偏移：0x0508 +(通道号 × 0x20)
复位值：0x0000 0000

该寄存器包含一个通道的状态和事件，当一个通道中断发生时，软件需要为相应通道读取该寄存器以获取产生中断的中断源。该寄存器中的标志位均由硬件置位，并且写1清除。

该寄存器只能按字(32位)访问

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




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:11 | 保留 | 必须保持复位值。 |
| 10 | DTER | 数据切换错误<br>IN事务获取一个数据包，但是该包的PID和USBFS_HCHxLEN寄存器中的DPID[1:0]控制位不匹配。 |
| 9 | REQOVR | 请求队列上溢<br>当请求队列已满且出现新的传输时，请求队列上溢。 |
| 8 | BBER | 串扰错误<br>USB总线上发生一个串扰事件。产生串扰事件的典型原因是端点发送了一个数据包，但是数据包长度超过了端点的最大包长。 |
| 7 | USBER | USB总线错误<br>当在接收一个数据包的过程中，发生以下事件时，将置位USB总线错误标志位：<br>– 接收包有一个错误的CRC域<br>– 在USB总线上检测到填充错误<br>– 当等待一个响应包时，超时 |
| 6 | 保留 | 必须保持复位值。 |
| 5 | ACK | ACK<br>接收到或者发送了一个ACK响应包 |
| 4 | NAK | NAK<br>接收到一个NAK响应包 |
| 3 | STALL | STALL<br>接收到一个STALL响应包 |
| 2 | 保留 | 必须保持复位值。 |
| 1 | CH | 通道中止<br>通道被当前请求所禁用，在当前请求处理的过程中，并不响应其他请求处理。 |
| 0 | TF | 发送完成<br>该通道所有的事务成功完成并且无错误发生。 |

对于IN通道，在USBFS_HCHxLEN寄存器的PCNT位减到0后，该标志位被置位。
对于OUT通道，当软件从Rx FIFO中读取和取出一个TF状态条目时，该标志位被置位。

### 主机通道 x 中断使能寄存器 (USBFS_HCHxINTEN) (x = 0…7，其中 x = 通道号)

地址偏移：0x050C +(通道号 × 0x20)
复位值：0x0000 0000

该寄存器包含USBFS_HCHxINTF寄存器内中断标志位的中断使能位。如果该寄存器的某位被软件置位，USBFS_HCHxINTF寄存器内的相应位能够触发一个通道中断。该寄存器内的位可由软件置位和清除。

该寄存器只能按字(32位)访问

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | | | | | DTERIE | REQOVRIE | BBERIE | USBERIE | 保留 | ACKIE | NAKIE | STALLIE | 保留 | CHIE | TFIE |
| | | | | | rw | rw | rw | rw | | rw | rw | rw | | rw | rw |

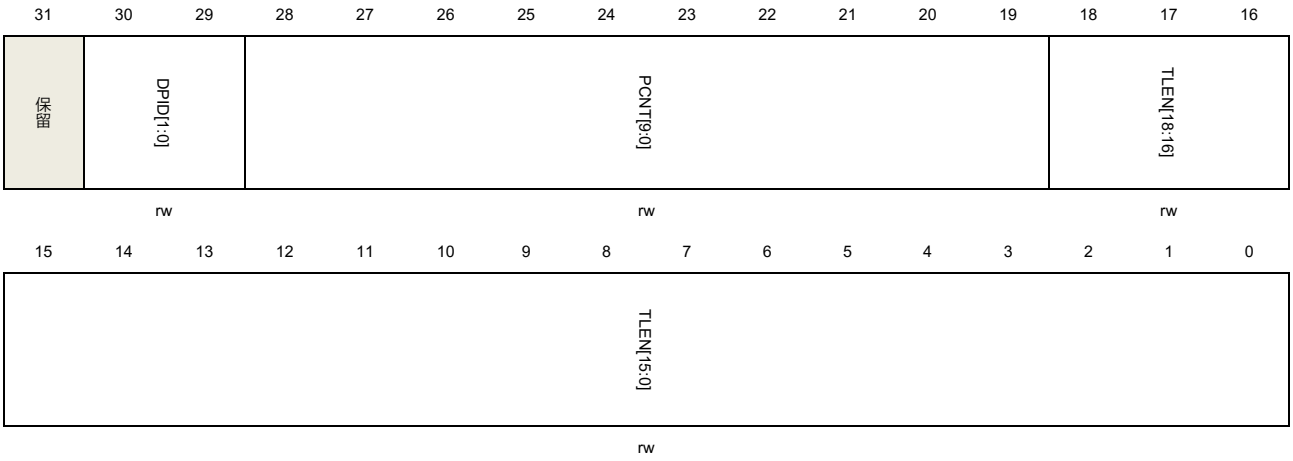
| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:11 | 保留 | 必须保持复位值。 |
| 10 | DTERIE | 数据切换错误中断使能<br>0：禁用数据切换错误中断<br>1：使能数据切换错误中断 |
| 9 | REQOVRIE | 请求队列上溢中断使能<br>0：禁用请求队列上溢中断<br>1：使能请求队列上溢中断 |
| 8 | BBERIE | 串扰错误中断使能<br>0：禁用串扰错误中断<br>1：使能串扰错误中断 |
| 7 | USBERIE | USB总线错误中断使能<br>0：禁用USB总线错误中断<br>1：使能USB总线错误中断 |

| | | |
|---|---|---|
| 6 | 保留 | 必须保持复位值。 |
| 5 | ACKIE | ACK中断使能<br>0：禁用ACK中断<br>1：使能ACK中断 |
| 4 | NAKIE | NAK中断使能<br>0：禁用NAK中断<br>1：使能NAK中断 |
| 3 | STALLIE | STALL中断使能<br>0：禁用STALL中断<br>1：使能STALL中断 |
| 2 | 保留 | 必须保持复位值。 |
| 1 | CHIE | 通道中止中断使能<br>0：禁用通道中止中断<br>1：使能通道中止中断 |
| 0 | TFIE | 传输完成中断使能<br>0：禁用传输完成中断<br>1：使能传输完成中断 |

### 主机通道 x 长度寄存器 (USBFS_HCHxLEN) (x = 0…7，其中 x = 通道号)

地址偏移：0x0510 +(通道号 × 0x20)
复位值：0x0000 0000

该寄存器只能按字(32位)访问




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | 保留 | 必须保持复位值。 |
| 30:29 | DPID[1:0] | 数据PID |

| | | |
|---|---|---|
| | | 软件应该在传输起始之前写该段位域。对于OUT传输，该位域包含第一个传输包的数据PID。对于IN传输，该位域包含第一个接收包的数据PID，并且如果数据PID不匹配的话，将会触发DTER标志位。在传输开始之后，USBFS遵循USB协议自动改变和切换该位域。<br>00：DATA0<br>10：DATA1<br>11：SETUP(仅对于控制传输)<br>01：保留 |
| 28:19 | PCNT[9:0] | 包计数<br>在一个传输中希望发送(OUT)或接收(IN)的数据包个数。<br>软件应该在通道使能之前写该位域。在传输启动之后，该位域在正确传输每个数据包后，自动减少。 |
| 18:0 | TLEN[18:0] | 传输长度<br>一次传输的总数据字节数。<br>对于OUT传输，该位域为OUT传输中期望发送的所有数据包总数据字节数。软件应该在通道使能之前写该位域。当软件正确向通道的数据Tx FIFO中写入一个包时，该位域以包中字节大小进行递减。<br>对于IN传输，每次软件从Rx FIFO中读取一个包后，该位域也以包中字节大小进行减少。 |

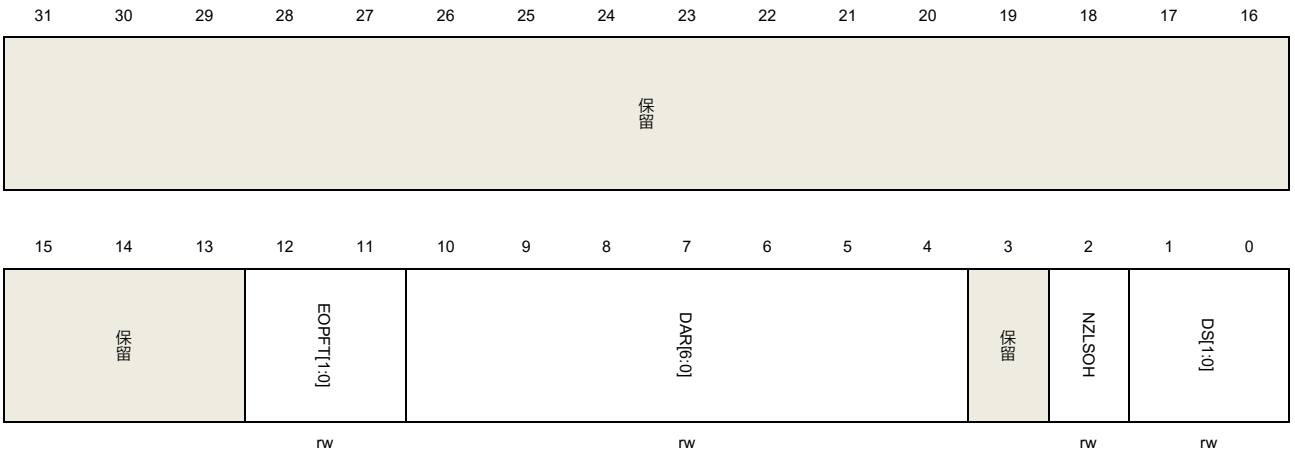
## 21.7.3. 设备控制和状态寄存器

### 设备配置寄存器 (USBFS_DCFG)

地址偏移：0x0800
复位值：0x0000 0000

在上电、枚举或执行某些控制命令后，该寄存器配置内核为设备模式。在设备初始化后，不可以改变该寄存器值。

该寄存器采用字(32位)访问

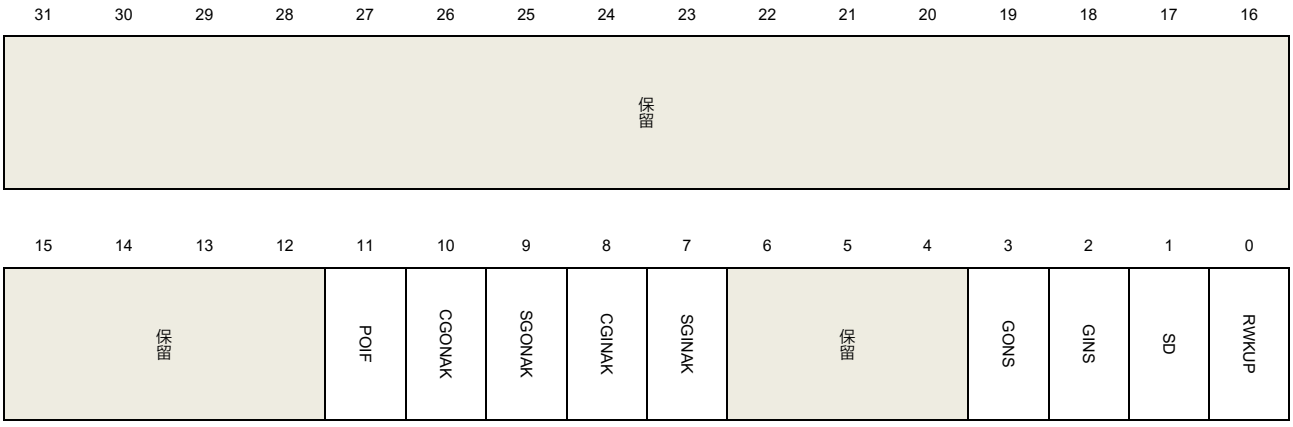


| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:13 | 保留 | 必须保持复位值。 |
| 12:11 | EOPFT[1:0] | 周期性帧尾时间<br>该域定义周期性帧时间的帧尾标志触发的时间点<br>00：80%的帧时间<br>01：85%的帧时间<br>10：90%的帧时间<br>11：95%的帧时间 |
| 10:4 | DAR[6:0] | 设备地址<br>该位定义USB设备地址，USBFS采用该位匹配接收的设备令牌地址域，在接收到来自主机的设置地址的命令后，软件设置该域。 |
| 3 | 保留 | 必须保持复位值。 |
| 2 | NZLSOH | 非零长度OUT状态阶段握手<br>在OUT状态阶段，当USB设备接收到一个非零长度数据包时，该域控制USBFS是接收该包，还是用STALL握手信号拒绝该包。<br>0：把该包视为正常包，根据设备OUT端点控制寄存器的NAKS和STALL位，回复握手相应握手包<br>1：发送STALL握手，不保存接收到的OUT数据包 |
| 1:0 | DS[1:0] | 设备速度<br>该域控制设备连入主机后的设备速度<br>11：全速<br>其他：保留 |

### 设备控制寄存器 (USBFS_DCTL)

地址偏移：0x0804
复位值：0x0000 0000

该寄存器采用字(32 位)访问



rw w w w w r r rw rw

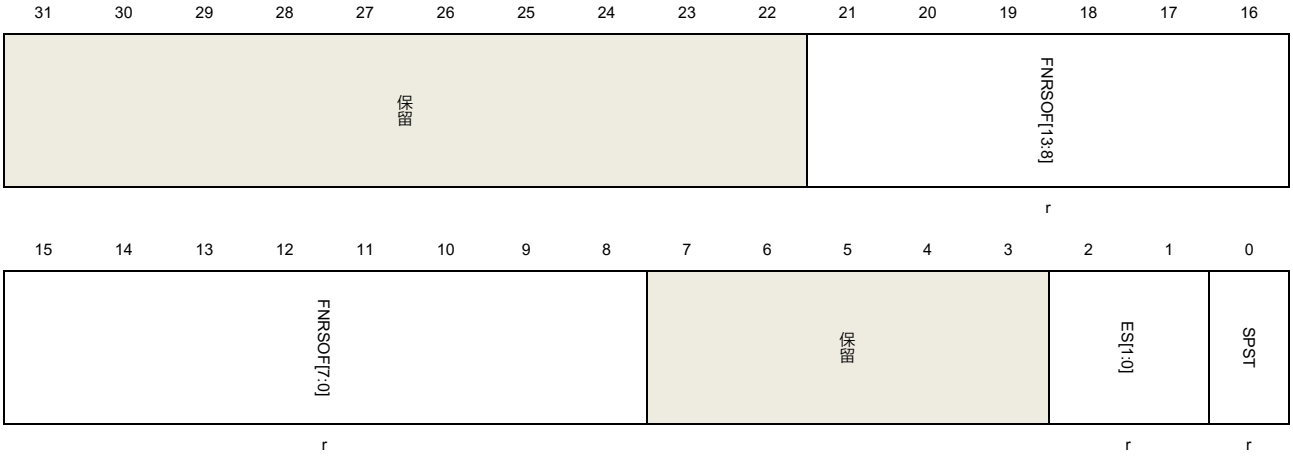
| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:12 | 保留 | 必须保持复位值。 |
| 11 | POIF | 上电初始化完成<br>软件通过设置该位，通知USBFS寄存器从掉电模式下唤醒后已完成初始化。 |
| 10 | CGONAK | 清零全局OUT NAK<br>软件设置该位从而清零该寄存器的GONS位 |
| 9 | SGONAK | 设置全局OUT NAK<br>软件设置该位从而实现该寄存器的位GONS置位。<br>当GONS位为零，设置该位一段时间后会引起USBFS_GINTF寄存器的GONAK标志触发，软件应该在再次写该位前清除GONAK标志。 |
| 8 | CGINAK | 清零全局IN NAK<br>软件设置该位从而清零该寄存器的GINS位 |
| 7 | SGINAK | 设置全局IN NAK<br>软件设置该位从而实现该寄存器的位GINS置位<br>当GINS位为零，设置该位一段时间会引起USBFS_GINTF寄存器的GINAK标志触发，软件应该在再次写该位前清除GINAK标志。 |
| 6:4 | 保留 | 必须保持复位值。 |
| 3 | GONS | 全局OUT NAK状态<br>0：USBFS回复OUT事务的握手信号以及是否保存OUT数据包由Rx FIFO状态、端点的NAK、STALL位确定。<br>1：USBFS回复OUT事务NAK握手信号，不保存接收的OUT数据包。 |
| 2 | GINS | 全局IN NAK状态<br>0：USBFS回复IN事务的握手信号由Tx FIFO状态、端点的NAK、STALL位确定。<br>1：USBFS通常回复IN事务NAK握手信号 |
| 1 | SD | 软断开<br>软件可实现USB总线上的软断开，在置1该位后，关掉DP线上的上拉电阻，从而引起主机检测设备的断开。<br>0：没有软断开生成<br>1：生成软断开 |
| 0 | RWKUP | 远程唤醒<br>在挂起状态，软件可通过该位来生成一个远程唤醒信号来通知主机恢复USB总线<br>0：没有远程唤醒信号生成<br>1：生成远程唤醒信号 |

### 设备状态寄存器 (USBFS_DSTAT)

地址偏移：0x0808
复位值：0x0000 0000

该寄存器包含设备模式下的 USBFS 的状态和信息。

该寄存器采用字(32 位)访问




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:22 | 保留 | 必须保持复位值。 |
| 21:8 | FNRSOF[13:0] | 所接收的SOF帧编号<br>USBFS会在接收到一个SOF令牌后更新该域。 |
| 7:3 | 保留 | 必须保持复位值。 |
| 2:1 | ES[1:0] | 枚举速度<br>该域指示所枚举的设备速度，在寄存器USBFS_GINTF的ENUMF标志触发后，软件可以读取该域。<br>11：全速<br>其他：保留 |
| 0 | SPST | 挂起状态<br>该位指示设备是否处于挂起状态。<br>0：设备不在挂起状态<br>1：设备在挂起状态 |

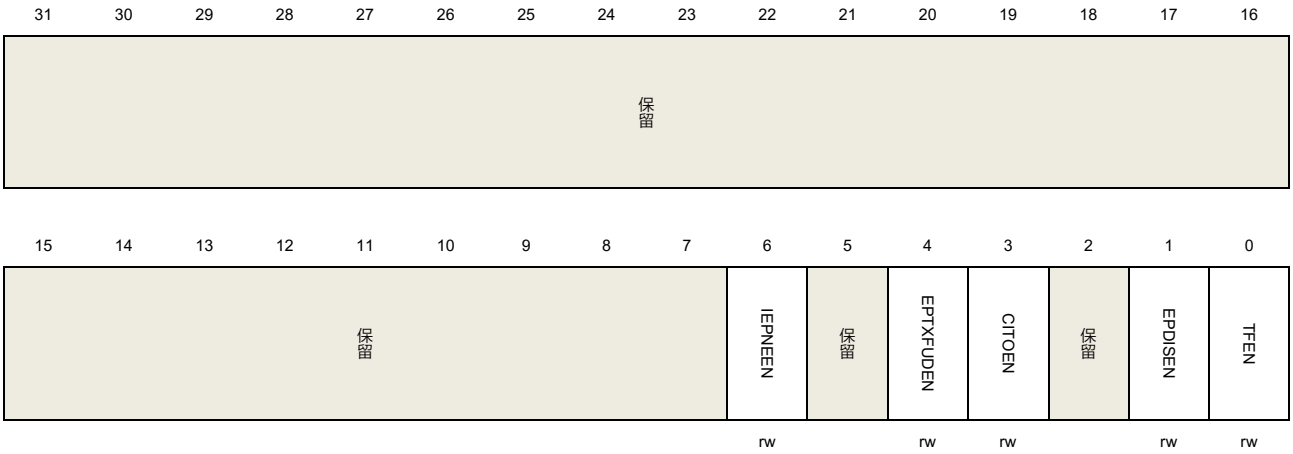
### 设备 IN 端点通用中断使能寄存器 (USBFS_DIEPINTEN)

地址偏移：0x0810
复位值：0x0000 0000

该寄存器包含寄存器 USBFS_DIEPxINTF 的中断使能位，如果软件置 1 某位，其在寄存器 USBFS_DIEPxINTF 中对应的位可以触发寄存器 USBFS_DAEPINT 中的一个端点中断。该位

可以通过软件置位和清零。

该寄存器采用字(32 位)访问



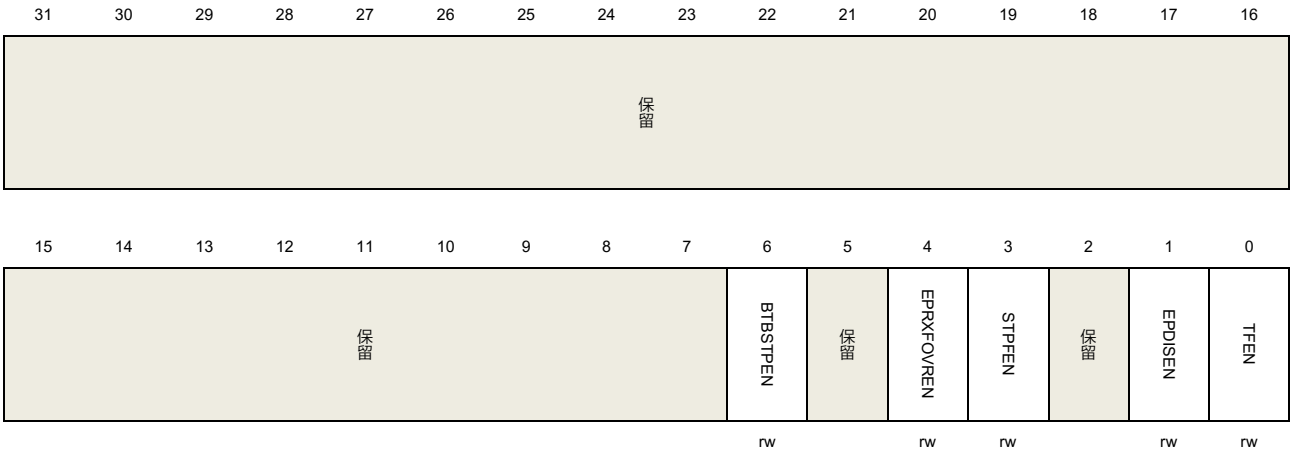

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:7 | 保留 | 必须保持复位值。 |
| 6 | IEPNEEN | IN端点NAK有效中断使能位<br>0：除能中断<br>1：使能中断 |
| 5 | 保留 | 必须保持复位值。 |
| 4 | EPTXFUDEN | 端点Tx FIFO下溢中断使能位<br>0：除能中断<br>1：使能中断 |
| 3 | CITOEN | 控制IN事务超时中断使能位<br>0：除能中断<br>1：使能中断 |
| 2 | 保留 | 必须保持复位值。 |
| 1 | EPDISEN | 端点除能中断使能位<br>0：除能中断<br>1：使能中断 |
| 0 | TFEN | 传输完成中断使能位<br>0：除能中断<br>1：使能中断 |

### 设备 OUT 端点通用中断使能寄存器 (USBFS_DOEPINTEN)

地址偏移：0x0814
复位值：0x0000 0000

该寄存器包含寄存器 USBFS_DOEPxINTF 中的标志的中断使能位，如果软件置 1 某位，其在寄存器 USBFS_DOEPxINTF 中对应的位可以触发寄存器 USBFS_DAEPINT 中的一个端点中断。该位可以通过软件置位和清零。

该寄存器采用字(32 位)访问



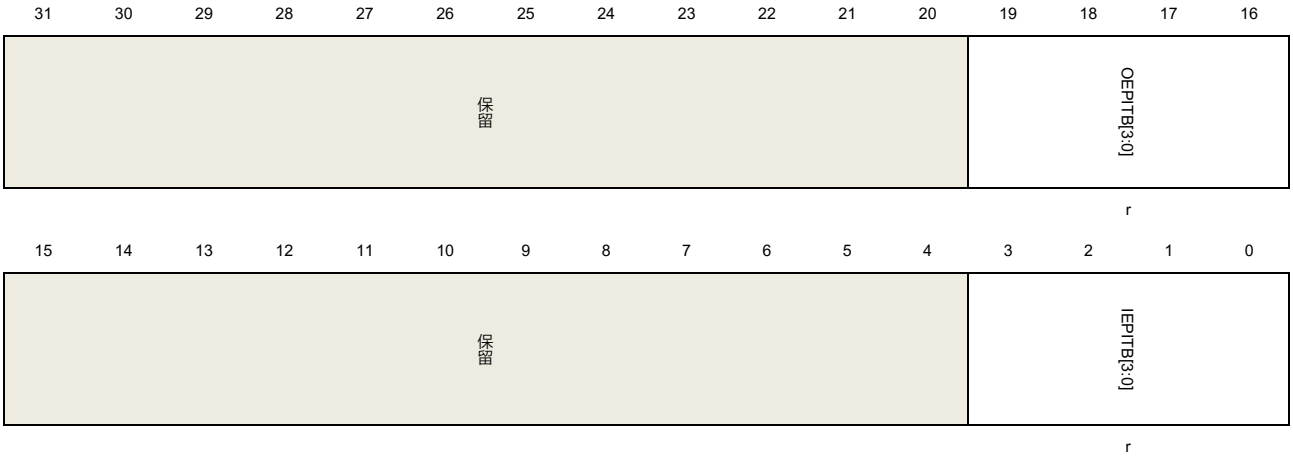

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:6 | 保留 | 必须保持复位值。 |
| 6 | BTBSTPEN | 连续SETUP包中断使能位(仅适用于控制OUT端点)<br>0：除能中断<br>1：使能中断 |
| 5 | 保留 | 必须保持复位值。 |
| 4 | EPRXFOVREN | 端点Rx FIFO上溢中断使能位<br>0：除能中断<br>1：使能中断 |
| 3 | STPFEN | SETUP阶段完成中断使能位(仅适用于控制OUT端点)<br>0：除能中断<br>1：使能中断 |
| 2 | 保留 | 必须保持复位值。 |
| 1 | EPDISEN | 端点除能中断使能位<br>0：除能中断<br>1：使能中断 |
| 0 | TFEN | 传输完成中断使能位<br>0：除能中断<br>1：使能中断 |

### 设备端点中断寄存器 (USBFS_DAEPINT)

地址偏移：0x0818
复位值：0x0000 0000

当一个端点的中断被触发，USBFS 置 1 该寄存器的相应位，软件可通过该寄存器获取在本次中断中的端点号。

该寄存器采用字(32 位)访问




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:20 | 保留 | 必须保持复位值。 |
| 19:16 | OEPITB[3:0] | 设备OUT端点中断位<br>每个位代表一个OUT端点：位16代表OUT端点0，位19代表OUT端点3。 |
| 15:4 | 保留 | 必须保持复位值。 |
| 3:0 | IEPITB[3:0] | 设备IN端点中断位<br>每个位代表一个IN端点：位0代表IN端点0，位3代表IN端点3 |

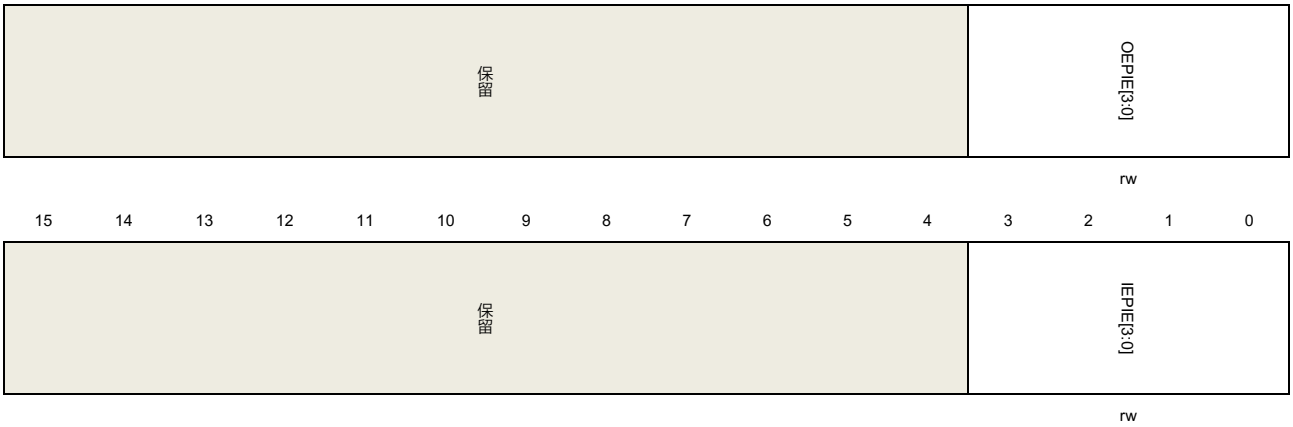
### 设备端点中断使能寄存器 (USBFS_DAEPINTEN)

地址偏移：0x081C
复位值：0x0000 0000

软件可通过该寄存器使能或除能端点的中断，只有当端点在该寄存器中相应位被置 1 才能触发寄存器 USBFS_GINTF 的端点中断标志 OEPIF 或 IEPIF。

该寄存器采用字(32 位)访问



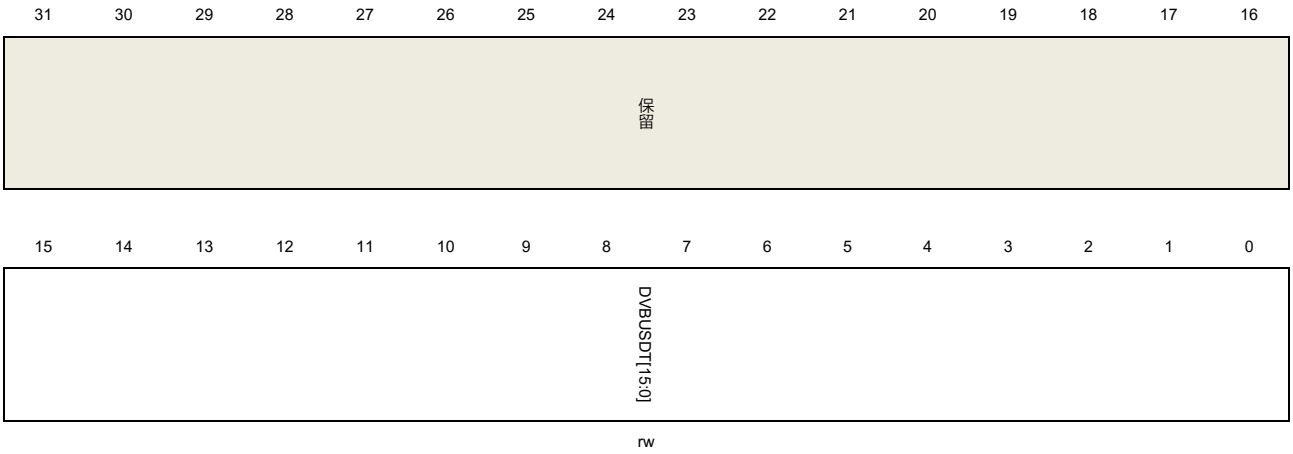



| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:20 | 保留 | 必须保持复位值。 |
| 19:16 | OEPIE[3:0] | OUT端点中断使能位<br>0：除能OUT端点n中断<br>1：使能OUT端点n中断<br>每个位代表一个OUT端点：位16对应OUT端点0，位19对应OUT端点3。 |
| 15:4 | 保留 | 必须保持复位值。 |
| 3:0 | IEPIE[3:0] | IN端点中断使能位<br>0：除能IN端点n中断<br>1：使能IN端点n中断<br>每个位代表一个IN端点：位0对应IN端点0，位3对应IN端点3。 |

### 设备 VBUS 放电时间寄存器 (USBFS_DVBUSDT)

地址偏移：0x0828
复位值：0x0000 17D7

该寄存器采用字(32 位)访问

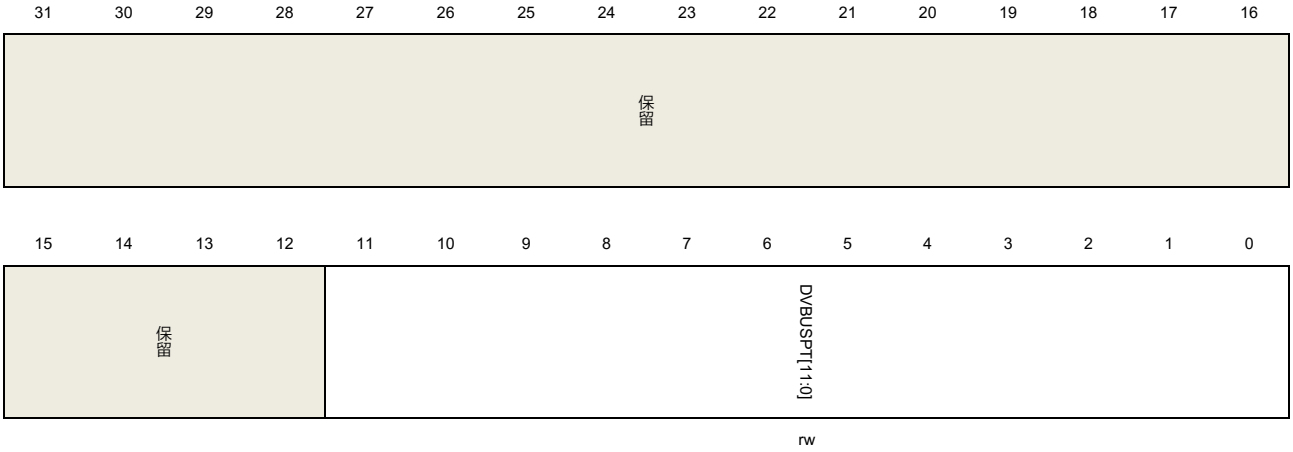


| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15:0 | DVBUSDT[15:0] | 设备$V_{BUS}$放电时间<br>在SRP协议中，在$V_{BUS}$脉冲产生后，有一个放电过程，该域定义了$V_{BUS}$的放电时间，真正的放电时间是1024*DVBUSDT[15:0] *$T_{USBCLOCK}$，$T_{USBCLOCK}$是USB时钟周期时间。 |

### 设备 VBUS 脉冲时间寄存器 (USBFS_DVBUSPT)

地址偏移：0x082C
复位值：0x0000 05B8

该寄存器采用字(32 位)访问




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:12 | 保留 | 必须保持复位值。 |
| 11:0 | DVBUSPT[11:0] | 设备$V_{BUS}$脉冲时间<br>该域定义$V_{BUS}$的脉冲时间，真正的充电时间是1024*DVBUSPT[15:0] *$T_{USBCLOCK}$，$T_{USBCLOCK}$是USB时钟周期时间 |

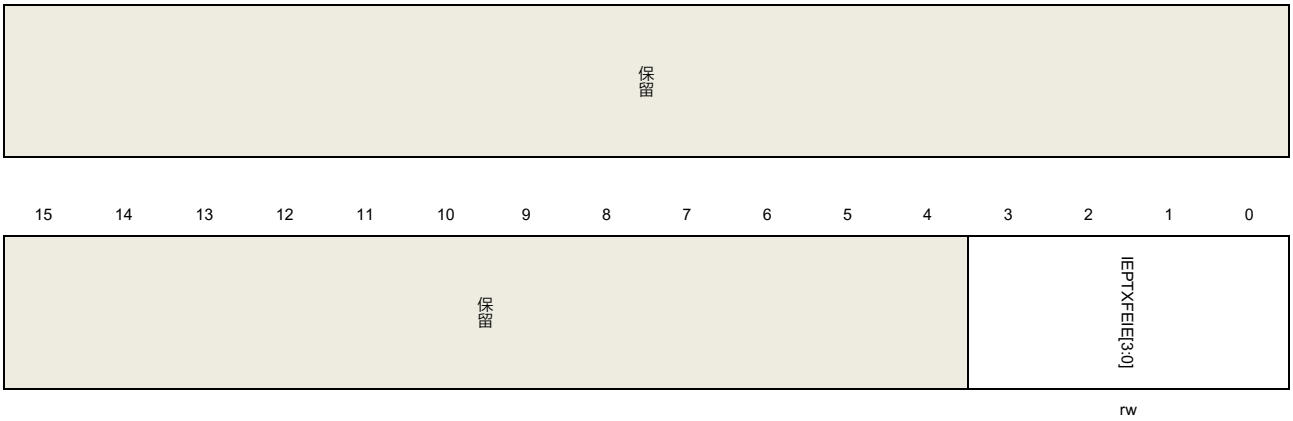
### 设备 IN 端点 FIFO 空中断使能寄存器 (USBFS_DIEPFEINTEN)

地址偏移：0x0834
复位值：0x0000 0000

该寄存器包含 IN 端点 Tx FIFO 空中断的使能位。

寄存器采用字(32 位)访问






| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:4 | 保留 | 必须保持复位值。 |
| 3:0 | IEPTXFEIE[3:0] | IN端点Tx FIFO空中断的使能位<br>该域控制着USBFS_DIEPxINTF寄存器的TXFE位能否置位寄存器USBFS_DAEPINT中的端点中断位。<br>位0对应IN端点0，位3对应IN端点3<br>0：除能FIFO空中断<br>1：使能FIFO空中断 |

### 设备 IN 端点 0 控制寄存器 (USBFS_DIEP0CTL)

地址偏移：0x0900
复位值：0x0000 8000

该寄存器采用字(32 位)访问



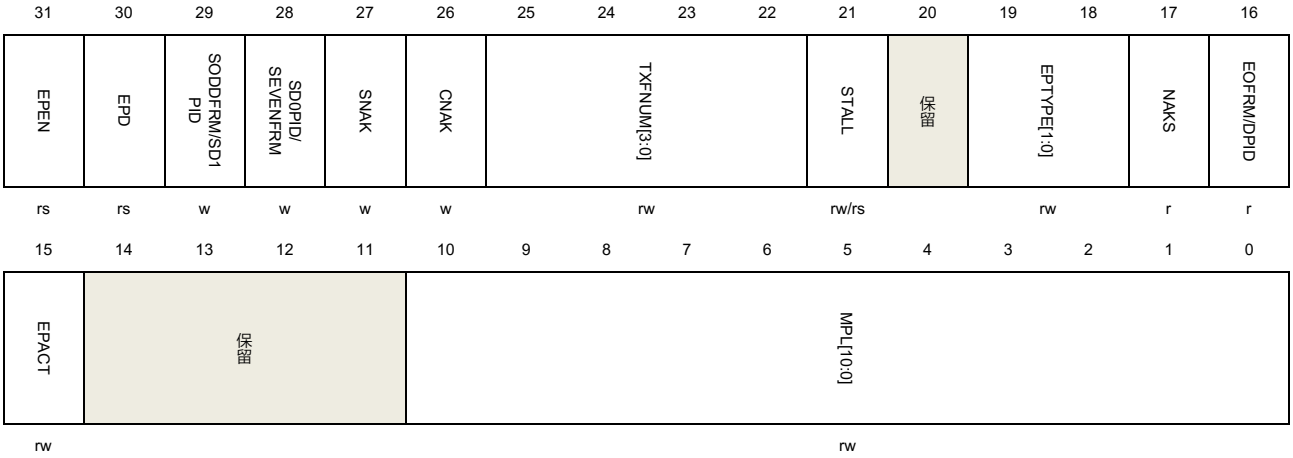

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | EPEN | 端点使能<br>软件置位、USBFS清零 |

| | | |
|---|---|---|
| | | 0：端点除能<br>1：端点使能<br>软件应该按照操作指南使能或除能端点。 |
| 30 | EPD | 端点除能<br>软件可通过置位该位从而除能端点，软件应该按照操作指南使能或除能端点。 |
| 29:28 | 保留 | 必须保持复位值。 |
| 27 | SNAK | 置位NAK<br>软件置位该位来设置该寄存器的NAKS位 |
| 26 | CNAK | 清零NAK<br>软件置位该位来清零该寄存器的NAKS位 |
| 25:22 | TXFNUM[3:0] | Tx FIFO编号<br>定义IN端点0的Tx FIFO编号 |
| 21 | STALL | STALL握手<br>当接收IN令牌时，软件可以通过置1该位发送STALL握手包，对于相应的OUT端点0，在接收SETUP令牌后，USBFS清除此位。该位比该寄存器的NAKS位和寄存器USBFS_DCTL的GINS位优先级要高，如果STALL和NAKS位都被置位，STALL位生效。 |
| 20 | 保留 | 必须保持复位值。 |
| 19:18 | EPTYPE[1:0] | 端点类型<br>该域固定为'00'，控制端点。 |
| 17 | NAKS | NAK状态<br>当该寄存器的STALL位和寄存器USBFS_DCTL的位GINS被清零，该位控制USBFS的NAK状态。<br>0：根据端点Tx FIFO的状态，USBFS发送数据或握手包<br>1：USBFS总为IN令牌发送NAK握手包<br>该位是只读位，可以通过该寄存器的位CNAK和位SNAK控制该位 |
| 16 | 保留 | 必须保持复位值。 |
| 15 | EPACT | 端点激活<br>对于端点0来说，该域固定为'1' |
| 14:2 | 保留 | 必须保持复位值。 |
| 1:0 | MPL[1:0] | 最大包长<br>此域定义了控制数据包的最大包长，如USB 2.0协议所描述，对控制传输而言，有四种包长度：<br>00：64字节<br>01：32字节<br>10：16字节<br>11：8字节 |

### 设备 IN 端点 x 控制寄存器 (USBFS_DIEPxCTL) (x = 1..3, x 是端点编号)

地址偏移：0x0900 + (x * 0x20)
复位值：0x0000 0000

该寄存器采用字(32 位)访问



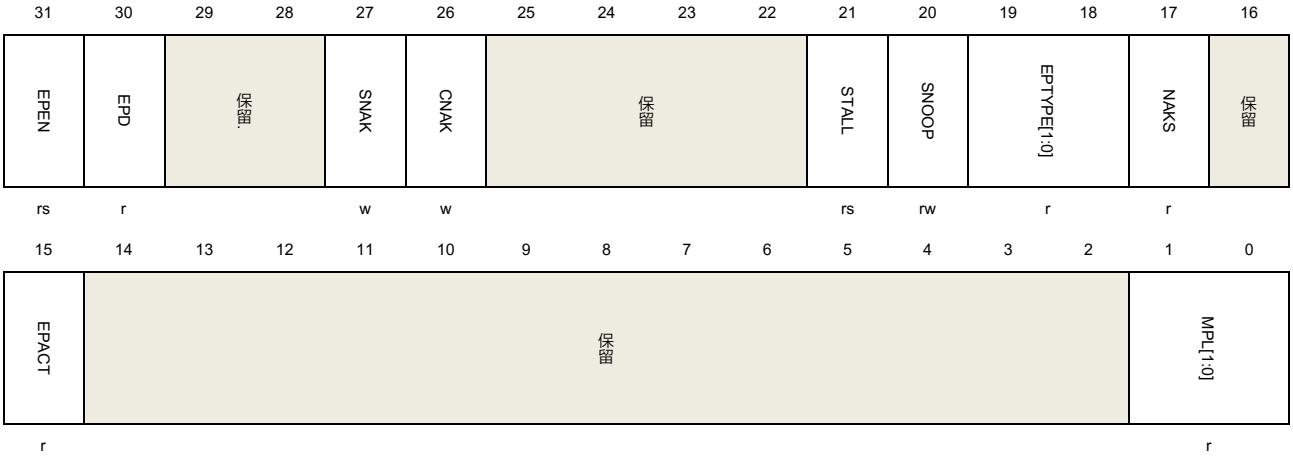

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | EPEN | 端点使能<br>软件置位，USBFS清零<br>0：端点除能<br>1：端点使能<br>软件应该按照操作指南使能或除能端点 |
| 30 | EPD | 端点除能<br>软件可通过置位该位从而除能端点，软件应该按照操作指南使能或除能端点。 |
| 29 | SODDFRM | 设置奇数帧(适用于同步IN端点)<br>软件通过置1该位置1该寄存器的EOFRM位。 |
| | SD1PID | 设置DATA1 PID(适用于中断和大容量IN端点)<br>软件可通过置1该位置1该寄存器的DPID位。 |
| 28 | SEVENFRM | 设置偶数帧(适用于同步IN端点)<br>软件通过置1该位清零该寄存器的EOFRM位。 |
| | SD0PID | 设置DATA1(适用于中断和大容量IN端点)<br>软件可通过置1该位清零该寄存器的DPID位。 |
| 27 | SNAK | 设置NAK<br>软件置1该位置1该寄存器的NAKS位。 |
| 26 | CNAK | 清零NAK<br>软件置1该位清零该寄存器的NAKS位。 |
| 25:22 | TXFNUM[3:0] | Tx FIFO编号 |

| | | |
|---|---|---|
| | | 该位定义了IN端点的Tx FIFO编号。 |
| 21 | STALL | STALL握手<br>当接收IN令牌时，软件可以通过置1该位发送STALL握手包。该位比该寄存器的NAKS位和寄存器USBFS_DCTL的GINS位优先级要高，如果STALL和NAKS位都被置位，STALL位生效。<br>对于控制IN端点：<br>当对应的OUT端点接收到SETUP令牌时，只有USBFS可以清零此位，软件不可清除此位。<br>对于中断或大容量IN端点：<br>只有软件可以清零此位。 |
| 20 | 保留 | 必须保持复位值。 |
| 19:18 | EPTYPE[1:0] | 端点类型<br>该域定义端点的传输类型：<br>00：控制<br>01：同步<br>10：大容量<br>11：中断 |
| 17 | NAKS | NAK状态<br>当该寄存器的STALL位和寄存器USBFS_DCTL的位GINS被清零，该位控制USBFS的NAK状态：<br>0：根据端点Tx FIFO的状态，USBFS发送数据或握手包<br>1：USBFS总为IN令牌发送NAK握手包<br>该位是只读位，可以通过该寄存器的位CNAK和位SNAK控制该位。 |
| 16 | EOFRM | 奇偶帧(适用于同步IN端点)<br>对于同步传输，软件通过使用该位控制USBFS仅使用奇数帧或偶数帧为IN事务发送数据包，如果当前帧号的奇偶性不匹配该位，USBFS回复一个零长度的包：<br>0：只在偶数帧发送数据<br>1：只在奇数帧发送数据 |
| | DPID | 端点数据PID(适用于中断或大容量IN端点)<br>在端点或大容量传输中，有数据PID翻转机制，在传输开始之前，软件通过设定SD0PID来设置此位，按照USB协议中描述的数据PID翻转机制，USBFS在传输过程中保持该位。<br>0：数据包的PID是DATA0<br>1：数据包的PID是DATA1 |
| 15 | EPACT | 端点激活<br>该位控制端点是否激活，当端点没有激活，忽略任何令牌，不做任何回复。 |
| 14:11 | 保留 | 必须保持复位值。 |
| 10:0 | MPL[10:0] | 该域定义最大包长度以字节为单位 |

### 设备 OUT 端点 0 控制寄存器 (USBFS_DOEP0CTL)

地址偏移: 0x0B00
复位值: 0x0000 8000

该寄存器采用字(32 位)访问




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | EPEN | 端点使能<br>软件置位，USBFS清零<br>0：端点除能<br>1：端点使能<br>软件应该按照操作指南使能或除能端点。 |
| 30 | EPD | 端点除能<br>对于OUT端点0，该位固定为0 |
| 29:28 | 保留 | 必须保持复位值。 |
| 27 | SNAK | 设置NAK<br>软件置1该位置1该寄存器的NAKS位 |
| 26 | CNAK | 清零NAK<br>软件置1该位清零该寄存器的NAKS位 |
| 25:22 | 保留 | 必须保持复位值。 |
| 21 | STALL | STALL握手<br>在OUT事务中，软件可以通过置1该位发送STALL握手包，对于OUT端点0，在接收SETUP令牌后，USBFS清除此位。该位比该寄存器的NAKS位和寄存器USBFS_DCTL的GINS位优先级要高，即如果STALL和NAKS位都被置位，STALL位生效。 |
| 20 | SNOOP | 调查模式<br>该位控制OUT端点的调查模式，在调查模式中，USBFS不再检查接收数据包的CRC值 |

| | | |
|---|---|---|
| | | 0：调查模式除能<br>1：调查模式使能 |
| 19:18 | EPTYPE[1:0] | 端点类型<br>对于控制端点，该位固定为"00" |
| 17 | NAKS | NAK状态<br>当该寄存器的STALL位和寄存器USBFS_DCTL的位GINS被清零，该位控制USBFS的NAK状态：<br>0：根据端点Rx FIFO的状态，USBFS发送数据或握手包<br>1：USBFS为OUT事务发NAK握手包<br>该位是只读位，通过该寄存器的CNAK和SNAK位控制该位。 |
| 16 | 保留 | 必须保持复位值。 |
| 15 | EPACT | 端点激活<br>对于端点0，该域固定为1。 |
| 14:2 | 保留 | 必须保持复位值。 |
| 1:0 | MPL[1:0] | 最大包长<br>该位是只读位，其数值来自于寄存器USBFS_DIEP0CTL的位MPL：<br>00：64字节<br>01：32字节<br>10：16字节<br>11：8字节 |

### 设备 OUT 端点 x 控制寄存器（USBFS_DOEPxCTL） (x= 1..3, x 是端点编号)

地址偏移：0x0B00 + (端点编号 × 0x20)
复位值：0x0000 0000

软件用该寄存器控制 OUT 端点 0 以外的每个逻辑 OUT 端点。

该寄存器采用字(32 位)访问

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EPEN | EPD | SODDFRM/SD1PID | SEVENFRM/SD0PID | SNAK | CNAK | 保留 | | | | STALL | SNOOP | EPTYPE[1:0] | | NAKS | EOFRM/DPID |
| rs | rs | w | w | w | w | | | | | rw/rs | rw | rw | | r | r |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EPACT | 保留 | | | | MPL[10:0] | | | | | | | | | | |
| rw | | | | | rw | | | | | | | | | | |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 31 | EPEN | 端点使能<br>软件置位，USBFS清零<br>0：端点除能<br>1：端点使能<br>软件应该按照操作指南使能或除能端点。 |
| 30 | EPD | 端点除能<br>软件通过置1该位除能端点，软件应该按照操作指南使能或除能端点。 |
| 29 | SODDFRM | 设置奇数帧(适用于同步OUT端点)<br>该位只针对同步OUT端点有效。<br>软件置1该位来置位该寄存器的EOFRM位。 |
| | SD1PID | 设置DATA1 PID(适用于中断和大容量OUT端点)<br>软件置1该位来置位该寄存器的DPID位。 |
| 28 | SEVENFRM | 设置偶数帧(适用于同步OUT端点)<br>软件置1该位来清零该寄存器的EOFRM位。 |
| | SD0PID | 设置DATA0 PID(适用于中断和大容量OUT端点)<br>软件置1该位来清零该寄存器的DPID位。 |
| 27 | SNAK | 设置NAK<br>软件置1该位从而置1该寄存器的NAKS位 |
| 26 | CNAK | 清零NAK<br>软件置1该位从而清零该寄存器的NAKS位 |
| 25:22 | 保留 | 必须保持复位值。 |
| 21 | STALL | STALL握手<br>在OUT事务中，软件可以通过置1该位发送STALL握手包。该位比该寄存器的NAKS位和寄存器USBFS_DCTL的GINS位优先级要高，如果STALL和NAKS位都被置位，STALL位生效。<br>对于控制OUT端点：<br>当OUT端点接收SETUP令牌时，只有USBFS可以清零该位，软件不可清零此位。<br>对于中断或大容量OUT端点：<br>只有软件可以清零该位。 |
| 20 | SNOOP | 调查模式<br>该位控制OUT端点的调查模式，在调查模式中，USBFS不再检查接收数据包的CRC值<br>0：调查模式除能<br>1：调查模式使能 |
| 19:18 | EPTYPE[1:0] | 端点类型<br>该域定义端点的传输类型<br>00：控制 |

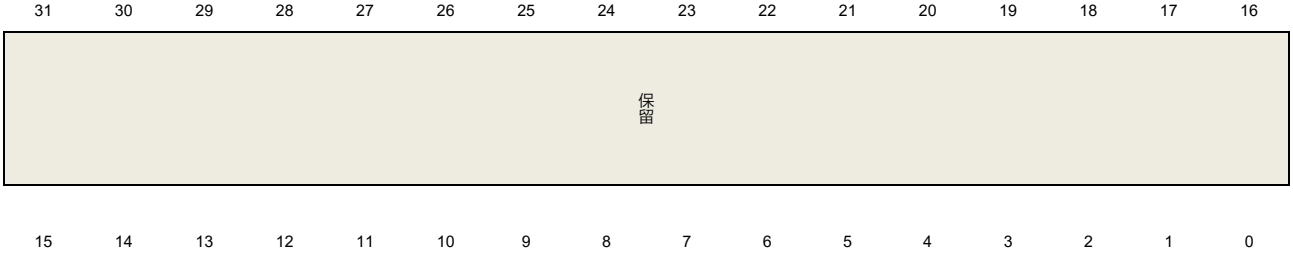
| | | |
|---|---|---|
| | | 01：同步<br>10：大容量<br>11：中断 |
| 17 | NAKS | NAK状态<br>当该寄存器的STALL位和寄存器USBFS_DCTL的位GONS被清零，该位控制USBFS的NAK状态：<br>0：根据端点的Rx FIFO的状态，发送握手包<br>1：USBFS为OUT事务发送NAK握手<br>该位是只读位，通过该寄存器的CNAK和SNAK位控制该位 |
| 16 | EOFRM | 奇偶帧(适用于同步OUT端点)<br>对于同步传输，软件通过使用该位控制USBFS只在奇数帧或偶数帧发送数据包给OUT事务，如果当前帧号的奇偶性不匹配该位，USBFS丢弃数据包。<br>0：只在偶数帧发送数据<br>1：只在奇数帧发送数据 |
| | DPID | 端点数据PID(适用于中断或大容量端点)<br>在端点或大容量传输中，有数据PID翻转机制，在传输开始之前，软件通过设定SD0PID来设置此位，按照USB协议中描述的数据PID翻转机制，USBFS在传输过程中保持该位。<br>0：数据包PID是DATA0<br>1：数据包PID是DATA1 |
| 15 | EPACT | 端点激活<br>此位控制端点是否激活，当端点没有激活，忽略任何令牌，不作任何回复。 |
| 14:11 | 保留 | 必须保持复位值。 |
| 10:0 | MPL[10:0] | 该位定义最大包长 |

### 设备 IN 端点 x 中断标志寄存器 (USBFS_DIEPxINTF) (x = 0…3, x 是端点编号)

地址偏移：0x0908 + (端点编号 × 0x20)
复位值：0x0000 0080

该寄存器包含 IN 端点的状态和事件，当一个 IN 端点的中断发生时，读取该端点的中断标志寄存器，从而获知中断源。该寄存器的标志位通常由硬件置位，除了只读的 TXFE 位，其他位写 1 清零。

该寄存器采用字(32 位)访问



| 保留 | TXFE | IEPNE | 保留 | EPTXFUD | CITO | 保留 | EPDIS | TF |
|---|---|---|---|---|---|---|---|---|
| | r | rc_w1 | | rc_w1 | rc_w1 | | rc_w1 | rc_w1 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 必须保持复位值。 |
| 7 | TXFE | 发送FIFO空<br>IN端点的Tx FIFO达到寄存器USBFS_GAHBCS的位TXFTH所定义的空阈值。 |
| 6 | IEPNE | IN端点NAK有效<br>寄存器USBFS_DIEPxCTL的位SNAK的设置生效，该位可以通过写1或置位CNAK位来清零。 |
| 5 | 保留 | 必须保持复位值。 |
| 4 | EPTXFUD | 端点Tx FIFO下溢<br>如果当IN令牌被接收后，Tx FIFO没有数据包可发送，该标志被触发。 |
| 3 | CITO | 控制IN事务超时中断<br>在控制IN事务中，如果设备等待的握手包超时，该标志位被触发。 |
| 2 | 保留 | 必须保持复位值。 |
| 1 | EPDIS | 端点除能<br>端点除能时，该标志位被触发。 |
| 0 | TF | 传输完成<br>当该端点的所有IN事务完成，该标志位被触发。 |

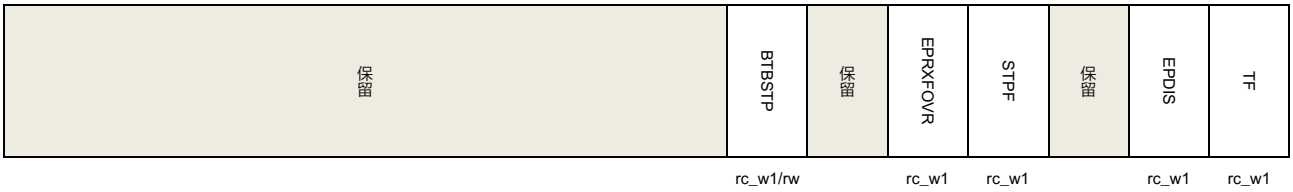
### 设备 OUT 端点 x 中断标志寄存器 (USBFS_DOEPxINTF) (x = 0..3, x 是端点编号)

地址偏移：0x0B08 + (端点编号 × 0x20)
复位值：0x0000 0000

该寄存器包含 OUT 端点的状态和事件，当一个 OUT 端点中断发生时，读取该端点的中断标志寄存器，从而获知中断源。该寄存器的标志位通常硬件置位，各位写 1 清零。

该寄存器采用字(32 位)访问

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

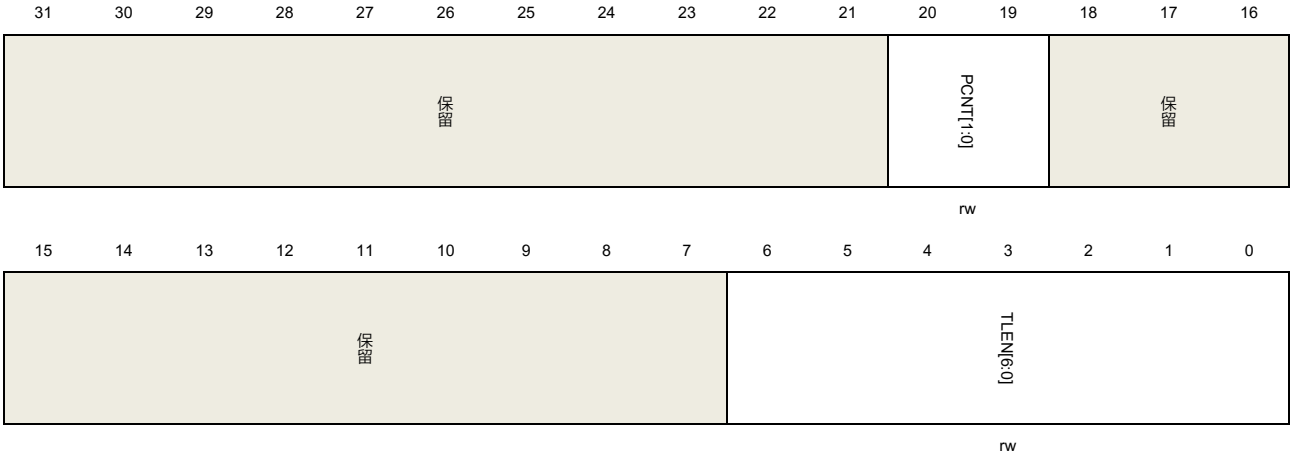



| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:7 | 保留 | 必须保持复位值。 |
| 6 | BTBSTP | 连续SETUP包(适用于控制OUT端点)<br>当一个控制OUT端点接收超过连续3个SETUP包时，该标志被触发。 |
| 5 | 保留 | 必须保持复位值。 |
| 4 | EPRXFOVR | 端点Rx FIFO上溢<br>当OUT令牌被接收时，如果OUT端点的Rx FIFO没有足够的空间存放数据包，该位被触发。在这种情况下，USBFS不能接收OUT数据包，发送一个NAK握手包。 |
| 3 | STPF | SETUP阶段完成(适用于控制OUT端点)<br>当一个SETUP阶段完成，也就是USBFS在一个setup令牌后接收了一个IN或OUT令牌，该位被置位。 |
| 2 | 保留 | 必须保持复位值。 |
| 1 | EPDIS | 端点除能<br>端点除能时，该标志位被触发。 |
| 0 | TF | 传输完成<br>当该端点的所有OUT事务完成，该标志位被触发。 |

### 设备 IN 端点 0 传输长度寄存器 (USBFS_DIEP0LEN)

地址偏移：0x0910
复位值：0x0000 0000

该寄存器采用字(32 位)访问

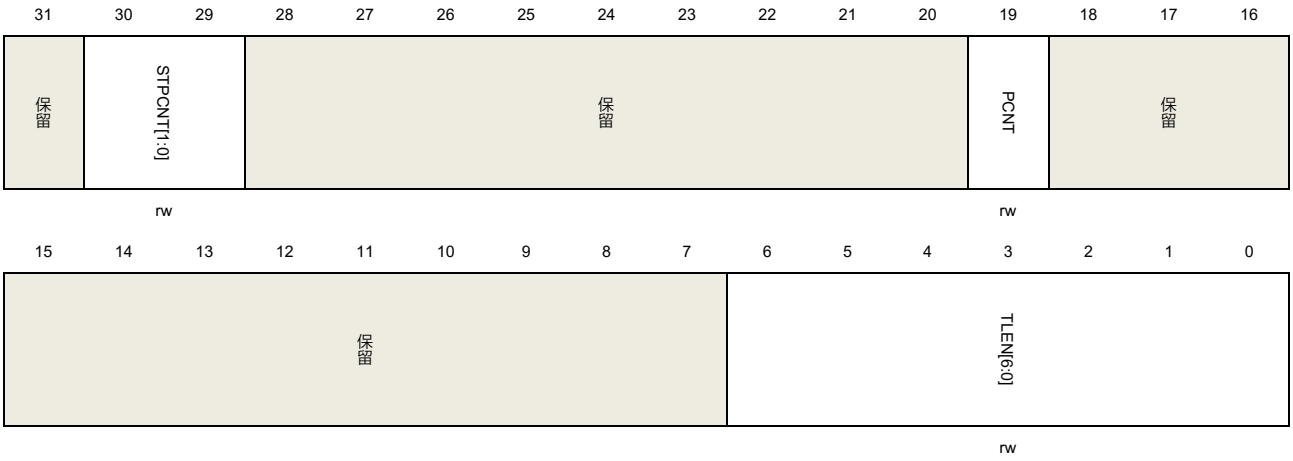


| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:21 | 保留 | 必须保持复位值。 |
| 20:19 | PCNT[1:0] | 包数<br>传输中被发送的数据包数量。<br>在端点使能之前，软件设置该位，在传输开始后，该域在每次数据包成功发送后递减。 |
| 18:7 | 保留 | 必须保持复位值。 |
| 6:0 | TLEN[6:0] | 传输长度<br>一次传输的数据总字节数。<br>该域是IN传输中需要发送的数据包的总字节数，在端点使能之前，软件设置该位，在成功地将数据包写入端点的Tx FIFO中，该域按照数据包大小递减。 |

### 设备 OUT 端点 0 传输长度寄存器 (USBFS_DOEP0LEN)

地址偏移：0x0B10
复位值：0x0000 0000

该寄存器采用字(32 位)访问



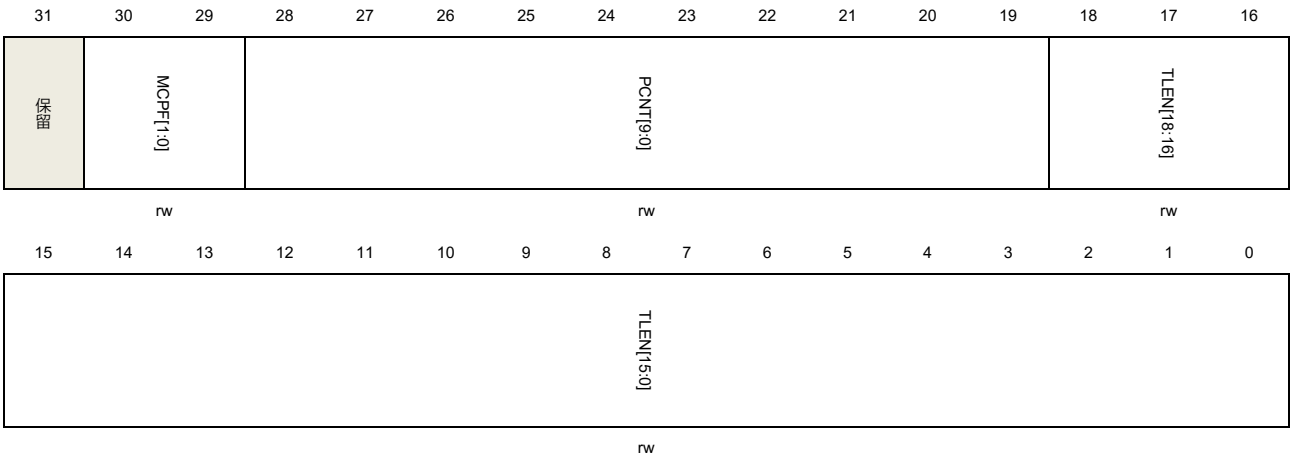

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 31 | 保留 | 必须保持复位值。 |
| 30:29 | STPCNT[1:0] | SETUP包数目<br>该域定义端点可以接受的最大连续SETUP包数量。<br>在SETUP传输之前，设置该域，每当连续SETUP包接收到时，该域值减1，当该域达到0时，寄存器USBFS_DOEP0INTF的BTBSTP标志被触发。<br>00：0个包<br>01：1个包<br>10：2个包<br>11：3个包 |

| | | |
|---|---|---|
| 28:20 | 保留 | 必须保持复位值。 |
| 19 | PCNT | 包数目<br>一次传输中应该接收到包数量。<br>在使能端点前，软件设置该位，在传输开始后，每当数据包接收到后，该域数值递减。 |
| 18:7 | 保留 | 必须保持复位值。 |
| 6:0 | TLEN[6:0] | 传输长度<br>传输中数据总字数。<br>该域是OUT传输中需要接收的数据包的总字节数，在端点使能之前，软件设置该位，在软件成功地将数据包从Rx FIFO中读取后，该域按照数据包大小递减。 |

### 设备 IN 端点 x 传输长度寄存器 (USBFS_DIEPxLEN) (x = 1...3, x 是端点编号)

地址偏移：0x910 + (端点编号 × 0x20)
复位值：0x0000 0000

该寄存器采用字(32 位)访问



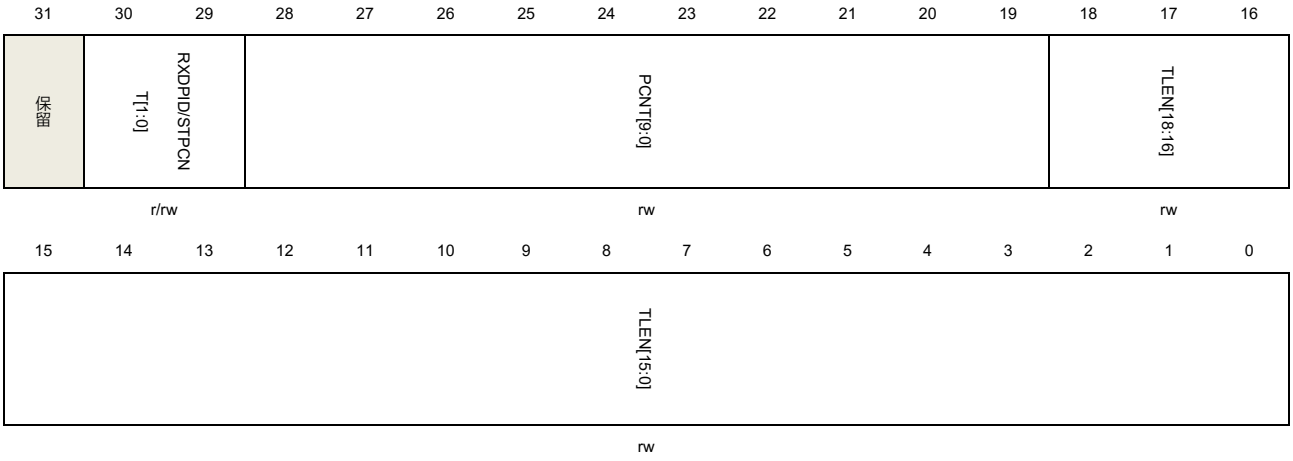

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | 保留 | 必须保持复位值。 |
| 30:29 | MCPF[1:0] | 每帧多包数目<br>该域是USB周期性IN端点每帧必须发的包数目。用于计算同步IN端点的数据PID。<br>01：1个包<br>10：2个包<br>11：3个包 |
| 28:19 | PCNT[9:0] | 包数量<br>传输中被发送的数据包数量。<br>在端点使能之前，软件设置该位，在传输开始后，该域在每次数据包成功发送后递减。 |

| | | |
|---|---|---|
| 18:0 | TLEN[18:0] | 传输长度<br>传输的数据总字节数。<br>该域是IN传输中需要发送的数据包的总字节数，在端点使能之前，软件设置该位，在成功地将数据包写入端点的Tx FIFO后，该域按照数据包大小递减。 |

### 设备 OUT 端点 x 传输长度寄存器 (USBFS_DOEPxLEN) (x = 1…3, x 是端点编号)

地址偏移：0x0B10 + (端点编号 × 0x20)
复位值：0x0000 0000

该寄存器采用字(32 位)访问




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | 保留 | 必须保持复位值。 |
| 30:29 | RXDPID[1:0] | 接收数据PID(适用于同步OUT端点)<br>该域保存该端点该数据包所接受的最后一个数据包的PID<br>00：DATA0<br>10：DATA1<br>其他：保留 |
| | STPCNT[1:0] | SETUP包数(适用于控制OUT端点)<br>该位定义该端点可以接受连续SETUP最大包数<br>在SETUP传输之前，设置该域，每当连续SETUP包接收到时，该域值减1，当该域达到0时，寄存器USBFS_DOEP0INTF的BTBSTP标志被触发。<br>00：0个包<br>01：1个包<br>10：2个包<br>11：3个包 |
| 28:19 | PCNT[9:0] | 包数目<br>传输中应该接收到包数量。<br>在端点使能前，软件设置该位，在传输开始后，每当数据包接收到后，该域数值递减。 |

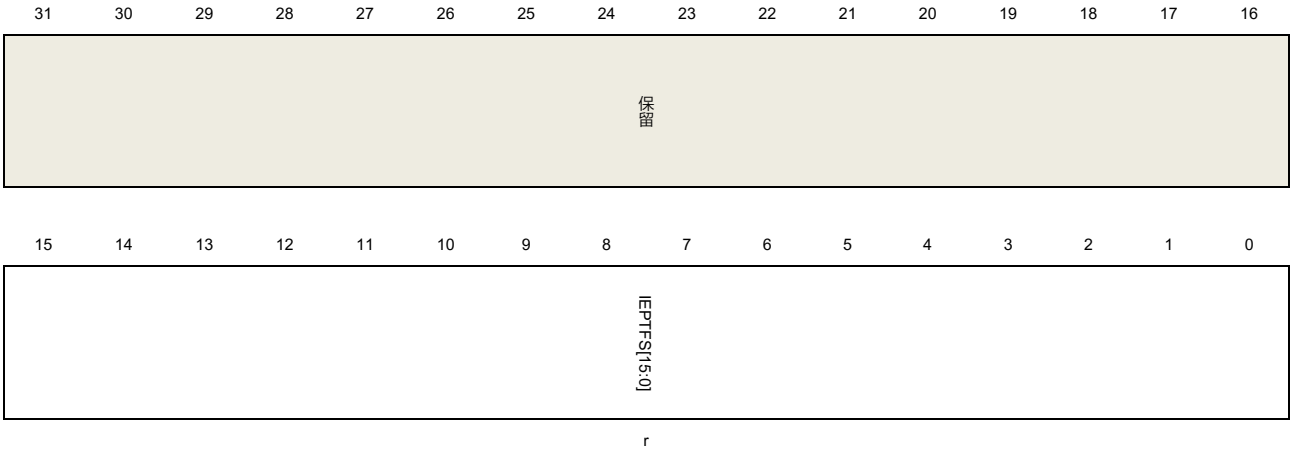
| | | |
|---|---|---|
| 18:0 | TLEN[18:0] | 传输长度<br>传输中数据总字数。<br>该域是IN传输中需要接收的包数据的总字节数，在端点使能之前，软件设置该位，在软件成功地将数据包从Rx FIFO中读取后，该域按照数据包大小递减。 |

**设备 IN 端点 x 发送 FIFO 状态寄存器 (USBFS_DIEPxTFSTAT) (x = 0…3, x 是端点编号)**

地址偏移：0x0918 + (端点编号 × 0x20)
复位值：0x0000 0200

该寄存器包含每个端点的 Tx FIFO 的信息。

该寄存器采用字(32 位)访问

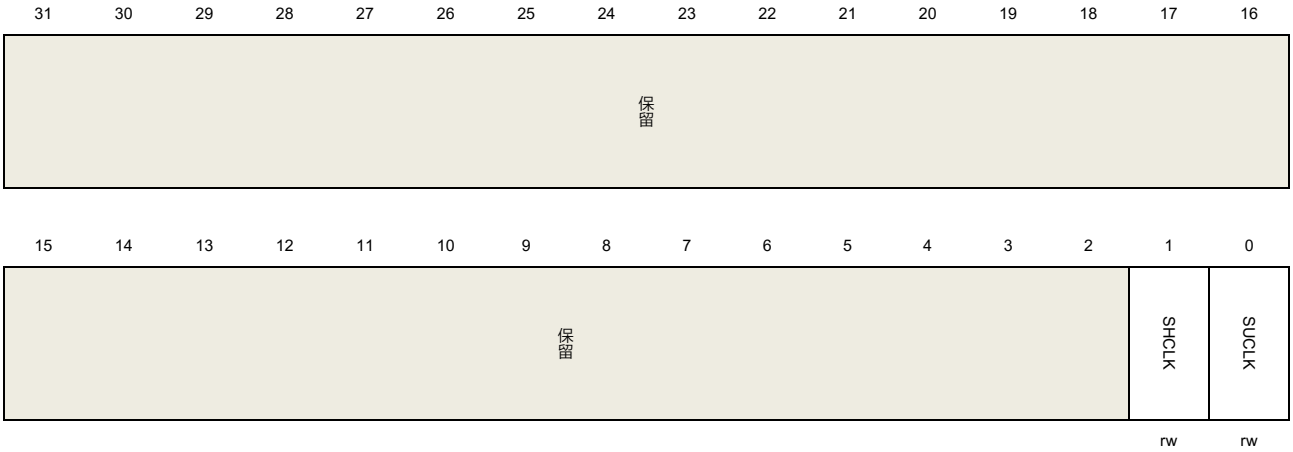



| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 必须保持复位值。 |
| 15:0 | IEPTFS[15:0] | IN端点的Tx FIFO可用空间<br>IN端点的Tx FIFO可用空间用32位字为单位<br>0：Tx FIFO满<br>1：1字可用<br>…<br>n：n字可用 |

## 21.7.4. 电源和时钟控制寄存器 (USBFS_PWRCLKCTL)

地址偏移：0x0E00
复位值：0x0000 0000

该寄存器采用字(32 位)访问




| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:2 | 保留 | 必须保持复位值。 |
| 1 | SHCLK | 停止HCLK<br>停止HCLK，节省电量。<br>0：HCLK未停止<br>1：HCLK停止 |
| 0 | SUCLK | 停止USB时钟<br>停止USB时钟，节省电量。<br>0：USB时钟未停止<br>1：USB时钟停止 |

# 22. 附录

## 22.1. 寄存器表中使用的缩写列表

**表 22-1. 寄存器功能位访问属性**

| 功能位访问属性 | 描述 |
|---|---|
| 读/写(rw) | 软件可以对这个位进行读写。 |
| 只读(r) | 软件只能对这个位进行读。 |
| 只写(w) | 软件只能对这个位进行写。读取该位将返回复位值。 |
| 读/写 1 清零(rc_w1) | 软件可以读该位，对该位写入 1 可以清除这个位。写入 0 对位值没有影响。 |
| 读/写 0 清零(rc_w0) | 软件可以读该位，对该位写入 0 可以清除这个位。写入 1 对位值没有影响。 |
| 翻转(t) | 软件可以通过写 1 来翻转该位。写入 0 对位值没有效果。 |
| 只读/写 1 触发(rt_w1) | 软件可以读该位，写入 1 触发事件，但对位值没有影响。 |

## 22.2. 术语表

**表 22-2. 术语**

| 术语 | 描述 |
|---|---|
| 字 | 32 位长度数据 |
| 半字 | 16 位长度数据. |
| 字节 | 8 位长度数据 |
| IAP(应用内编程) | IAP 是在用户程序运行时对微控制器的闪存重新编程的能力。 |
| ICP(在线编程) | ICP 是当设备安装在用户应用板上时，一个使用 JTAG 协议，SWD 协议或引导加载程序的微控制器的闪存编程能力。 |
| 选项字节 | 存储在闪存中的产品配置位 |
| AHB | 高级高性能总线 |
| APB | 高级外设总线 |
| RAZ | 读为 0 |
| WI | 写忽略 |
| RAZ/WI | 读为 0/写忽略 |

## 22.3. 可用外设

对于各个 MCU 系列的外设及其数量，请参考相应型号的数据手册。

# 23. 版本历史

表 23-1. 版本历史

| 版本号. | 说明 | 日期 |
|---|---|---|
| 1.0 | 初稿发布 | 2022 年 7 月 11 日 |
| 1.1 | 1.更新 GPIO 章节*表 8-5. TIMER 备用功能重映射*中 TIMER2_CH0 / TIMER2_CH1 映射引脚。<br>2.删除*通用定时器 L2(TIMERx, x=9,10,12,13)*章节中涉及 CI1FE0 和 CH1MS 的描述。<br>3.更新 *VDDA 域*中关于 VDD 和 VDDA 的描述：当 VDD 和 VDDA 不是同一个电源提供时，在上电和运行过程中 VDD 与 VDDA 差值不超过 0.3V。<br>4.更新*数模转换器(DAC)*章节。 | 2024 年 1 月 13 日 |
| 1.2 | 1. 更新*运行模式*章节中的软件流程。<br>2. 修改 *VDDA 域*章节中的 LVD 监测电压为 VDDA。<br>3. 更新*表 12-2. ADC 输入引脚定义*。<br>4. 更新*图 3-1.电源域概览*，并增加相关的备注说明。<br>5.更新 *$V_{DD}$ / $V_{DDA}$ 电源域*章节中关于 VDD、VDDA、POR、PDR 的描述。 | 2025 年 2 月 20 日 |
| 1.3 | 1. 修改*半双工通信模式*章节中 USART 半双工通信模式下 Tx 引脚配置描述。<br>2. 修改*端口控制寄存器 0 (GPIOx_CTL0, x=A..E)*和*端口控制寄存器 1 (GPIOx_CTL1, x=A..E)*的复位值。<br>3. 更新 *SMBus 支持*章节中 SMBUS 最大超时值为 25ms。<br>4. 新增 *I2S 时钟*章节 I2S 串行时钟配置约束。 | 2025 年 8 月 5 日 |
| 1.4 | 1. 增加 *CRC 功能*章节中 SPI 模块 CRC 功能使用注意事项。<br>2. 修改*电源管理单元(PMU)*章节电源域描述，1.2v 电源域修改位 Vcore 电源域。<br>3. 删除 *SPI0 四线 SPI 控制寄存器 (SPI_QCTL)*中的 IO23_DRV 功能位和相关描述。<br>4. 修改 *USART 接收器*章节噪声错误触发条件，起始位不会触发噪声错误。<br>5. 修改*智能卡(ISO7816-3)模式*章节描述：字节模式(T=0)时，在重发数据帧前插入帧间隔的位数由 2.5 位修改为 2 位。<br>6. 修改 *JTAG 链状结构*章节 BSD JTAG TAP 声明为 MCU JTAG TAP。<br>7. 增加*功能说明*章节中看门狗连续重装载最小间隔时间。<br>8. 增加 FMC 模块*主要特征*章节描述，闪存支持 64 位宽数据读取。<br>9. 增加 *Receiver timeout register (USART_RT)*中 USART_RT 位域描述注意事项。 | 2026 年 1 月 20 日 |

| | | |
|---|---|---|
| | 10. 增加***串行红外(IrDA SIR)编解码功能模块***章节描述：SIR 模块工作在低功耗模式下，配置 PSC[7:0]的同时需要配置波特率寄存器的值为 16*PSC[7:0]。 | |
| 1.5 | 1. 增加引导配置小节中 Bootloader 引脚使用注意事项。 | 2026 年 7 月 30 日 |

# Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the "Product"), is owned by the Company according to the laws of the People's Republic of China and other applicable laws. The Company reserves all rights under such laws and no Intellectual Property Rights are transferred (either wholly or partially) or licensed by the Company (either expressly or impliedly) herein. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

To the maximum extent permitted by applicable law, the Company makes no representations or warranties of any kind, express or implied, with regard to the merchantability and the fitness for a particular purpose of the Product, nor does the Company assume any liability arising out of the application or use of any Product. Any information provided in this document is provided only for reference purposes. It is the sole responsibility of the user of this document to determine whether the Product is suitable and fit for its applications and products planned, and properly design, program, and test the functionality and safety of its applications and products planned using the Product. The Product is designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only, and the Product is not designed or intended for use in (i) safety critical applications such as weapons systems, nuclear facilities, atomic energy controller, combustion controller, aeronautic or aerospace applications, traffic signal instruments, pollution control or hazardous substance management; (ii) life-support systems, other medical equipment or systems (including life support equipment and surgical implants); (iii) automotive applications or environments, including but not limited to applications for active and passive safety of automobiles (regardless of front market or aftermarket), for example, EPS, braking, ADAS (camera/fusion), EMS, TCU, BMS, BSG, TPMS, Airbag, Suspension, DMS, ICMS, Domain, ESC, DCDC, e-clutch, advanced-lighting, etc.. Automobile herein means a vehicle propelled by a self-contained motor, engine or the like, such as, without limitation, cars, trucks, motorcycles, electric cars, and other transportation devices; and/or (iv) other uses where the failure of the device or the Product can reasonably be expected to result in personal injury, death, or severe property or environmental damage (collectively "Unintended Uses"). Customers shall take any and all actions to ensure the Product meets the applicable laws and regulations. The Company is not liable for, in whole or in part, and customers shall hereby release the Company as well as its suppliers and/or distributors from, any claim, damage, or other liability arising from or related to all Unintended Uses of the Product. Customers shall indemnify and hold the Company, and its officers, employees, subsidiaries, affiliates as well as its suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Product.

Information in this document is provided solely in connection with the Product. The Company reserves the right to make changes, corrections, modifications or improvements to this document and the Product described herein at any time without notice. The Company shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2026 GigaDevice Semiconductor Inc. - All rights reserved