GigaDevice Semiconductor Inc.

GD32A508xx
Arm® Cortex®-M33 32-bit MCU

用户手册

1.8 版本

(2026 年 01 月)

目录

目录 2
图索引 18
表索引 27
1. 系统及存储器架构 31
1.1. Arm® Cortex®-M33 处理器 31
1.2. 系统架构 32
1.3. 存储器映射 34
1.3.1. 片上 SRAM 存储器 38
1.3.2. 片上闪存 38
1.4. 引导配置 38
1.5. 设备电子签名 40
1.5.1. 存储容量信息 40
1.5.2. 设备唯一 ID(96 位/位域) 40
2. 闪存控制器（FMC） 42
2.1. 简介 42
2.2. 主要特征 42
2.3. 功能说明 42
2.3.1. 闪存结构 42
2.3.2. 读操作 43
2.3.3. FMC_CTL 寄存器解锁 44
2.3.4. 页擦除 45
2.3.5. 整片擦除 46
2.3.6. 主存储闪存块编程 47
2.3.7. OTP 编程 49
2.3.8. 选项字节擦除 49
2.3.9. 选项字节编程 50
2.3.10. 选项字节说明 50
2.3.11. 页擦除/编程保护 51
2.3.12. 安全保护 52
2.4. FMC 寄存器 53
2.4.1. 等待状态寄存器（FMC_WS） 53
2.4.2. 解锁寄存器（FMC_KEY） 54
2.4.3. 选项字节操作解锁寄存器（FMC_OBKEY） 54
2.4.4. 状态寄存器（FMC_STAT） 55
2.4.5. 控制寄存器（FMC_CTL） 55
2.4.6. 地址寄存器（FMC_ADDR） 57

2.4.7. 选项字节状态寄存器（FMC_OBSTAT）.....57
2.4.8. 擦除/编程保护寄存器（FMC_WP）.....58
2.4.9. 产品 ID 寄存器（FMC_PID）.....58
3. 备份寄存器（BKP）.....59
3.1. 简介.....59
3.2. 主要特点.....59
3.3. 功能描述.....59
3.3.1. RTC 时钟校准.....59
3.3.2. 侵入检测.....59
3.4. BKP 寄存器.....60
3.4.1. 备份数据寄存器（BKP_DATAx）（x= 0..41）.....60
3.4.2. RTC 信号输出控制寄存器（BKP_OCTL）.....60
3.4.3. 侵入引脚控制寄存器（BKP_TPCTL）.....61
3.4.4. 侵入控制状态寄存器（BKP_TPCS）.....62
4. 电源管理单元（PMU）.....64
4.1. 简介.....64
4.2. 主要特性.....64
4.3. 功能描述.....64
4.3.1. 备份域.....66
4.3.2. V_{DD} / V_{DDA} 电源域.....67
4.3.3. V_{CORE} 电源域.....69
4.3.4. 省电模式.....69
4.4. PMU 寄存器.....73
4.4.1. 控制寄存器 0（PMU_CTL0）.....73
4.4.2. 电源控制和状态寄存器 0（PMU_CS0）.....75
4.4.3. 控制寄存器 1（PMU_CTL1）.....77
4.4.4. 电源控制和状态寄存器 1（PMU_CS1）.....77
5. 复位和时钟单元（RCU）.....79
5.1. 复位控制单元（RCTL）.....79
5.1.1. 简介.....79
5.1.2. 功能描述.....79
5.2. 时钟控制单元（CCTL）.....80
5.2.1. 简介.....80
5.2.2. 主要特性.....82
5.2.3. 功能描述.....83
5.3. RCU 寄存器.....87
5.3.1. 控制寄存器（RCU_CTL）.....87
5.3.2. 时钟配置寄存器 0（RCU_CFG0）.....89
5.3.3. 时钟中断寄存器（RCU_INT）.....92

5.3.4. APB2 复位寄存器（RCU_APB2RST）.....95
5.3.5. APB1 复位寄存器（RCU_APB1RST）.....98
5.3.6. AHB 使能寄存器（RCU_AHBEN）.....101
5.3.7. APB2 使能寄存器（RCU_APB2EN）.....103
5.3.8. APB1 使能寄存器（RCU_APB1EN）.....105
5.3.9. 备份域控制寄存器（RCU_BDCTL）.....108
5.3.10. 复位源/时钟寄存器（RCU_RSTSCK）.....109
5.3.11. AHB 复位寄存器（RCU_AHBRST）.....111
5.3.12. 时钟配置寄存器 1（RCU_CFG1）.....112
5.3.13. 深度睡眠模式电压寄存器（RCU_DSV）.....115
5.3.14. 附加时钟控制寄存器（RCU_ADDCTL）.....115
5.3.15. 附加时钟配置寄存器（RCU_ADDCFG）.....117
5.3.16. 附加时钟中断寄存器（RCU_ADDINT）.....118
5.3.17. PLL 时钟扩频控制寄存器（RCU_PLLSSCTL）.....119
5.3.18. 配置寄存器 2（RCU_CFG2）.....120
5.3.19. APB1 附加复位寄存器（RCU_ADDAPB1RST）.....121
5.3.20. APB1 附加使能寄存器（RCU_ADDAPB1EN）.....121

6. 时钟校准控制器（CTC）.....123

6.1. 简介.....123

6.2. 主要特性.....123

6.3. 功能描述.....123

6.3.1. REF 同步脉冲发生器.....124
6.3.2. CTC 校准计数器.....124
6.3.3. 频率评估和自动校准过程.....125
6.3.4. 软件编程指南.....126

6.4. CTC 寄存器.....127

6.4.1. 控制寄存器 0（CTC_CTL0）.....127
6.4.2. 控制寄存器 1（CTC_CTL1）.....128
6.4.3. 状态寄存器（CTC_STAT）.....129
6.4.4. 中断清除寄存器（CTC_INTC）.....131

7. 中断/事件控制器（EXTI）.....133

7.1. 简介.....133

7.2. 主要特性.....133

7.3. 中断功能描述.....133

7.4. 外部中断及事件(EXTI)框图.....137

7.5. 外部中断及事件功能概述.....137

7.6. EXTI 寄存器.....139

7.6.1. 中断使能寄存器(EXTI_INTEN).....139
7.6.2. 事件使能寄存器(EXTI_EVEN).....139
7.6.3. 上升沿触发使能寄存器(EXTI_RTEN).....140

7.6.4. 下降沿触发使能寄存器(EXTI_FTEN) 140
7.6.5. 软件中断事件寄存器(EXTI_SWIEV) 140
7.6.6. 挂起寄存器(EXTI_PD) 141
8. 通用和备用输入/输出接口（GPIO 和 AFIO） 142
8.1. 简介 142
8.2. 主要特性 142
8.3. 功能描述 142
8.3.1. 外部中断/事件线 144
8.3.2. 备用功能(AF) 144
8.3.3. 输入配置 144
8.3.4. 输出配置 144
8.3.5. 模拟配置 145
8.3.6. 备用功能(AF)配置 145
8.3.7. IO 引脚功能选择 146
8.3.8. GPIO 锁定功能 146
8.3.9. GPIO I/O 补偿单元 146
8.4. I/O 重映射功能和调试配置 147
8.4.1. 介绍 147
8.4.2. 主要特性 147
8.4.3. JTAG/SWD 备用功能重映射 147
8.4.4. TIMER AF 重映射 148
8.4.5. USART AF 重映射 149
8.4.6. I2C0 备用功能重映射 150
8.4.7. SPI0/SPI2/I2S 备用功能重映射 150
8.4.8. CAN0/1 备用功能重映射 151
8.4.9. ENET 备用功能重映射 151
8.4.10. CTC 备用功能重映射 152
8.4.11. CLK 引脚 AF 重映射 152
8.5. GPIO 寄存器 153
8.5.1. 端口控制寄存器 0（GPIOx_CTL0，x=A..G） 153
8.5.2. 端口控制寄存器 1（GPIOx_CTL1, x=A..G） 155
8.5.3. 端口输入状态寄存器（GPIOx_ISTAT，x=A..G） 156
8.5.4. 端口输出控制寄存器（GPIOx_OCTL，x=A..G） 157
8.5.5. 端口位操作寄存器（GPIOx_BOP，x=A..G） 157
8.5.6. 位清除寄存器（GPIOx_BC，x=A..G） 158
8.5.7. 端口配置锁定寄存器（GPIOx_LOCK，x=A..G） 158
8.5.8. 端口位速度寄存器（GPIOx_SPD，x=A..G） 159
8.5.9. 事件控制寄存器（AFIO_EC） 160
8.5.10. AFIO 端口配置寄存器 0（AFIO_PCF0） 160
8.5.11. EXTI 源选择寄存器 0 寄存器（AFIO_EXTISS0） 164
8.5.12. EXTI 源选择寄存器 1 寄存器（AFIO_EXTISS1） 165
8.5.13. EXTI 源选择寄存器 2 寄存器（AFIO_EXTISS2） 166

8.5.14. EXTI 源选择寄存器 3 寄存器（AFIO_EXTISS3）.....167
8.5.15. AFIO 端口配置寄存器 1（AFIO_PCF1）.....169
8.5.16. IO 补偿控制寄存器（AFIO_CPSCTL）.....170
8.5.17. AFIO 端口配置寄存器 A（AFIO_PCFA）.....171
8.5.18. AFIO 端口配置寄存器 B（AFIO_PCFB）.....172
8.5.19. AFIO 端口配置寄存器 C（AFIO_PCFC）.....175
8.5.20. AFIO 端口配置寄存器 D（AFIO_PCFD）.....176
8.5.21. AFIO 端口配置寄存器 E（AFIO_PCFE）.....177
8.5.22. AFIO 端口配置寄存器 G（AFIO_PCFG）.....179
9. 循环冗余校验管理单元（CRC）.....181
9.1. 简介.....181
9.2. 主要特性.....181
9.3. 功能描述.....182
9.4. CRC 寄存器.....183
9.4.1. 数据寄存器（CRC_DATA）.....183
9.4.2. 独立数据寄存器（CRC_FDATA）.....183
9.4.3. 控制寄存器（CRC_CTL）.....184
9.4.4. 初值寄存器（CRC_IDATA）.....184
9.4.5. 多项式寄存器（CRC_POLY）.....185
10. 三角函数加速器（TMU）.....186
10.1. 简介.....186
10.2. 主要特性.....186
10.3. 功能描述.....186
10.3.1. TMU 结构图.....186
10.3.2. 数据格式.....187
10.3.3. 模式 0.....188
10.3.4. 模式 1.....188
10.3.5. 模式 2.....188
10.3.6. 模式 3.....189
10.3.7. 模式 4.....189
10.3.8. 模式 5.....190
10.3.9. 模式 6.....190
10.3.10. 模式 7.....192
10.3.11. 模式 8.....192
10.4. 软件流程.....192
10.5. TMU 寄存器.....194
10.5.1. 输入数据 0 寄存器 (TMU_IDATA0).....194
10.5.2. 输入数据 1 寄存器(TMU_IDATA1).....194
10.5.3. 控制寄存器(TMU_CTL).....195
10.5.4. 数据 0 寄存器(TMU_DATA0).....195

10.5.5. 数据 1 寄存器(TMU_DATA1) 196
10.5.6. 状态寄存器(TMU_STAT) 196
11. 直接存储器访问控制器（DMA） 198
11.1. 简介 198
11.2. 主要特性 198
11.3. 结构框图 198
11.4. 功能描述 199
11.4.1. DMA 操作 199
11.4.2. 外设握手 201
11.4.3. 仲裁 201
11.4.4. 地址生成 201
11.4.5. 循环模式 201
11.4.6. 存储器到存储器模式 202
11.4.7. 通道配置 202
11.4.8. 中断 202
11.4.9. DMA 请求映射 203
11.5. DMA 寄存器 207
11.5.1. 中断标志位寄存器(DMA_INTF) 207
11.5.2. 中断标志位清除寄存器(DMA_INTC) 207
11.5.3. 通道 x 控制寄存器(DMA_CHxCTL) 208
11.5.4. 通道 x 计数寄存器 (DMA_CHxCNT) 210
11.5.5. 通道 x 外设基地址寄存器 (DMA_CHxPADDR) 211
11.5.6. 通道 x 存储器基地址寄存器 (DMA_CHxMADDR) 211
12. 调试（DBG） 212
12.1. 简介 212
12.2. JTAG/SW 功能描述 212
12.2.1. 切换 JTAG/SW 接口 212
12.2.2. 引脚分配 212
12.2.3. JTAG 链状结构 213
12.2.4. 调试复位 213
12.2.5. JEDEC-106 ID code 213
12.3. 调试保持功能描述 213
12.3.1. 低功耗模式调试支持 213
12.3.2. TIMER, I2C, WWDGT, 和 FWDGT 外设调试支持 214
12.4. DBG 寄存器 215
12.4.1. ID 寄存器（DBG_ID） 215
12.4.2. 控制寄存器（DBG_CTL） 215
13. 模数转换器（ADC） 219
13.1. 简介 219

13.2. 主要特征 219
13.3. 引脚和内部信号 220
13.4. 功能说明 220
13.4.1. 前置校准功能 221
13.4.2. ADC 时钟 221
13.4.3. ADCON 使能 221
13.4.4. 单端和差分输入通道 221
13.4.5. 常规序列 222
13.4.6. 运行模式 222
13.4.7. 转换结果阈值监测功能 225
13.4.8. 数据存储模式 225
13.4.9. 采样时间配置 226
13.4.10. 外部触发配置 226
13.4.11. DMA 请求 226
13.4.12. ADC 内部通道 226
13.4.13. 可编程分辨率（DRES） 227
13.4.14. 片上硬件过采样 227
13.5. ADC 同步模式 229
13.5.1. 独立模式 230
13.5.2. 常规并行模式 230
13.5.3. 常规快速交叉模式 231
13.5.4. 常规慢速交叉模式 231
13.6. 中断 232
13.7. ADC 寄存器 233
13.7.1. 状态寄存器（ADC_STAT） 233
13.7.2. 控制寄存器 0（ADC_CTL0） 234
13.7.3. 控制寄存器 1（ADC_CTL1） 235
13.7.4. 采样时间寄存器 0（ADC_SAMPT0） 237
13.7.5. 采样时间寄存器 1（ADC_SAMPT1） 238
13.7.6. 看门狗 0 高阈值寄存器（ADC_WDHT0） 239
13.7.7. 看门狗 0 低阈值寄存器（ADC_WDLT0） 239
13.7.8. 常规序列寄存器 0（ADC_RSQ0） 240
13.7.9. 常规序列寄存器 1（ADC_RSQ1） 240
13.7.10. 常规序列寄存器 2（ADC_RSQ2） 241
13.7.11. 常规数据寄存器（ADC_RDATA） 242
13.7.12. 过采样控制寄存器（ADC_OVSAMPCTL） 242
13.7.13. 看门狗 1 通道选择寄存器（ADC_WD1SR） 243
13.7.14. 看门狗 2 通道选择寄存器（ADC_WD2SR） 244
13.7.15. 看门狗 1 阈值寄存器（ADC_WDT1） 245
13.7.16. 看门狗 2 阈值寄存器（ADC_WDT2） 245
13.7.17. 差分模式控制寄存器（ADC_DIFCTL） 246
14. 数模转换器（DAC） 247

14.1. 简介 247
14.2. 主要特征 247
14.3. 功能描述 248
14.3.1. DAC 使能 248
14.3.2. DAC 输出缓冲 248
14.3.3. DAC 数据配置 249
14.3.4. DAC 触发 249
14.3.5. DAC 转换 249
14.3.6. DAC 噪声波 249
14.3.7. DAC 输出电压 250
14.3.8. DMA 请求 251
14.3.9. DAC 并发转换 251
14.3.10. DAC 输出 FIFO 251
14.4. DAC 寄存器 252
14.4.1. DACx 控制寄存器（DAC_CTL0） 252
14.4.2. DACx 软件触发寄存器（DAC_SWT） 255
14.4.3. DACx_OUT0 12 位右对齐数据保持寄存器（DAC_OUT0_R12DH） 255
14.4.4. DACx_OUT0 12 位左对齐数据保持寄存器（DAC_OUT0_L12DH） 256
14.4.5. DACx_OUT0 8 位右对齐数据保持寄存器（DAC_OUT0_R8DH） 256
14.4.6. DACx_OUT1 12 位右对齐数据保持寄存器（DAC_OUT1_R12DH） 257
14.4.7. DACx_OUT1 12 位左对齐数据保持寄存器（DAC_OUT1_L12DH） 257
14.4.8. DACx_OUT1 8 位右对齐数据保持寄存器（DAC_OUT1_R8DH） 257
14.4.9. DACx 并发模式 12 位右对齐数据保持寄存器（DACC_R12DH） 258
14.4.10. DACx 并发模式 12 位左对齐数据保持寄存器（DACC_L12DH） 258
14.4.11. DACx 并发模式 8 位右对齐数据保持寄存器（DACC_R8DH） 259
14.4.12. DACx_OUT0 数据输出寄存器（DAC_OUT0_DO） 259
14.4.13. DACx_OUT1 数据输出寄存器（DAC_OUT1_DO） 260
14.4.14. DACx 状态寄存器 0（DAC_STAT0） 260
14.4.15. DACx 控制寄存器 1（DAC_CTL1） 261
14.4.16. DACx 状态寄存器 1（DAC_STAT1） 262
15. 比较器（CMP） 264
15.1. 简介 264
15.2. 主要特征 264
15.3. 功能描述 264
15.3.1. 比较器时钟 265
15.3.2. 比较器的 I / O 配置 265
15.3.3. 比较器寄存器写保护 266
15.3.4. 比较器输出消隐 266
15.4. CMP 寄存器 268
15.4.1. CMP1 控制状态寄存器（CMP1_CS） 268
15.4.2. CMP3 控制状态寄存器（CMP3_CS） 269

15.4.3. CMP5 控制状态寄存器（CMP5_CS） 271
16. 看门狗定时器（WDGT） 274
16.1. 独立看门狗定时器（FWDGT） 274
16.1.1. 简介 274
16.1.2. 主要特征 274
16.1.3. 功能说明 274
16.1.4. FWDGT 寄存器 277
16.2. 窗口看门狗定时器（WWDGT） 280
16.2.1. 简介 280
16.2.2. 主要特征 280
16.2.3. 功能说明 280
16.2.4. WWDGT 寄存器 283
17. 实时时钟（RTC） 285
17.1. 简介 285
17.2. 主要特性 285
17.3. 功能描述 285
17.3.1. RTC 复位 286
17.3.2. RTC 读取 286
17.3.3. RTC 配置 286
17.3.4. RTC 标志位 287
17.4. RTC 寄存器 288
17.4.1. RTC 中断使能寄存器(RTC_INTEN) 288
17.4.2. RTC 控制寄存器(RTC_CTL) 288
17.4.3. RTC 预分频寄存器高位(RTC_PSCH) 289
17.4.4. RTC 预分频寄存器低位(RTC_PSCL) 290
17.4.5. RTC 分频器高位(RTC_DIVH) 290
17.4.6. RTC 分频器低位(RTC_DIVL) 290
17.4.7. RTC 计数寄存器高位(RTC_CNTH) 291
17.4.8. RTC 计数寄存器低位(RTC_CNTL) 291
17.4.9. RTC 闹钟寄存器高位(RTC_ALRMH) 292
17.4.10. RTC 闹钟寄存器低位(RTC_ALRML) 292
18. 定时器（TIMER） 293
18.1. 高级定时器（TIMERx,x=0,7） 294
18.1.1. 简介 294
18.1.2. 主要特性 294
18.1.3. 结构框图 294
18.1.4. 功能描述 295
18.1.5. TIMERx 寄存器(x=0,7) 320
18.2. 通用定时器 L0（TIMERx, x=1,2,3,4） 345
18.2.1. 简介 345

18.2.2. 主要特性 345
18.2.3. 结构框图 345
18.2.4. 功能描述 346
18.2.5. TIMERx 寄存器（x=1,2,3,4） 361
18.3. 通用定时器 L1（TIMERx, x=8,11） 385
18.3.1. 简介 385
18.3.2. 主要特性 385
18.3.3. 结构框图 385
18.3.4. 功能描述 386
18.3.5. TIMERx 寄存器(x=8,11) 396
18.4. 通用定时器 L2（TIMERx, x=9,10,12,13） 408
18.4.1. 简介 408
18.4.2. 主要特性 408
18.4.3. 结构框图 408
18.4.4. 功能描述 408
18.4.5. TIMERx 寄存器(x=9,10,12,13) 416
18.5. 基本定时器（TIMERx, x=5,6） 426
18.5.1. 简介 426
18.5.2. 主要特性 426
18.5.3. 结构框图 426
18.5.4. 功能描述 426
18.5.5. TIMERx 寄存器(x=5,6) 430
19. 超高分辨率定时器（SHRTIMER） 435
19.1. 简介 435
19.2. 主要特征 435
19.3. 结构框图 435
19.4. 功能说明 436
19.4.1. Master_TIMER 单元 436
19.4.2. Slave_TIMERx(x=0..4)单元 443
19.4.3. DLL 校准 473
19.4.4. 突发模式 474
19.4.5. 同步输入/输出 478
19.4.6. 外部事件 479
19.4.7. 故障输入 481
19.4.8. ADC 触发 482
19.4.9. DAC 触发 483
19.4.10. 中断 484
19.4.11. DMA 请求 485
19.4.12. DMA 模式 485
19.4.13. Debug 模式 486
19.5. SHRTIMER 寄存器 488

19.5.1. Master_TIMER 寄存器 488
19.5.2. Slave_TIMERx 寄存器(x=0..4) 499
19.5.3. 通用寄存器 556
20. 通用同步异步收发器（USART） 598
20.1. 通用同步异步收发器（USARTx, x=0..4） 598
20.1.1. 简介 598
20.1.2. 主要特性 598
20.1.3. 功能描述 599
20.1.4. USART 寄存器 613
20.2. 通用同步异步收发器（USARTx, x=5） 626
20.2.1. 简介 626
20.2.2. 主要特性 626
20.2.3. 功能描述 627
20.2.4. USART 寄存器 641
21. 内部集成电路总线接口（I2C） 657
21.1. 内部集成电路总线接口（I2Cx, x=0, 1） 657
21.1.1. 简介 657
21.1.2. 主要特性 657
21.1.3. 功能描述 657
21.1.4. I2C 寄存器 672
21.2. 内部集成电路总线接口（I2Cx, x=2） 684
21.2.1. 简介 684
21.2.2. 主要特征 685
21.2.3. 功能说明 685
21.2.4. I2C 寄存器 708
22. 串行外设接口/片上音频接口（SPI/I2S） 721
22.1. 简介 721
22.2. 主要特性 721
22.2.1. SPI 主要特性 721
22.2.2. I2S 主要特性 721
22.3. SPI 功能说明 722
22.3.1. SPI 结构框图 722
22.3.2. SPI 信号线描述 722
22.3.3. SPI 时序和数据帧格式 723
22.3.4. NSS 功能 724
22.3.5. SPI 运行模式 725
22.3.6. DMA 功能 733
22.3.7. CRC 功能 733
22.3.8. SPI 中断 734
22.4. I2S 功能说明 735

22.4.1. I2S 结构框图 735
22.4.2. I2S 信号线描述 735
22.4.3. I2S 音频标准 736
22.4.4. I2S 时钟 743
22.4.5. 运行 744
22.4.6. DMA 功能 748
22.4.7. I2S 中断 748
22.5. SPI/I2S 寄存器 750
22.5.1. 控制寄存器 0（SPI_CTL0） 750
22.5.2. 控制寄存器 1（SPI_CTL1） 752
22.5.3. 状态寄存器（SPI_STAT） 753
22.5.4. 数据寄存器（SPI_DATA） 754
22.5.5. CRC 多项式寄存器（SPI_CRCPOLY） 755
22.5.6. 接收 CRC 寄存器（SPI_RCRC） 755
22.5.7. 发送 CRC 寄存器（SPI_TCRC） 756
22.5.8. I2S 控制寄存器（SPI_I2SCTL） 756
22.5.9. I2S 时钟预分频寄存器（SPI_I2SPSC） 758
22.5.10. SPI0 四线 SPI 控制寄存器（SPI_QCTL） 758
23. SPI/QPI（SQPI） 760
23.1. 简介 760
23.2. 主要特性 760
23.3. 功能描述 760
23.3.1. SQPI 模式定义 760
23.3.2. SQPI 控制器采样极性 761
23.3.3. SQPI 控制器特殊指令 761
23.3.4. SQPI 读 ID 命令 762
23.3.5. SQPI 控制器输出时钟配置 762
23.3.6. SQPI 控制器初始化 763
23.3.7. 读 ID 命令流程 763
23.3.8. 读写操作流程 763
23.3.9. SQPI 控制器模式时序 763
23.4. SQPI 寄存器 765
23.4.1. 初始化寄存器 (SQPI_INIT) 765
23.4.2. 读命令寄存器 (SQPI_RCMD) 766
23.4.3. 写命令寄存器 (SQPI_WCMD) 767
23.4.4. ID 低位寄存器 (SQPI_IDL) 768
23.4.5. ID 高位寄存器 (SQPI_IDH) 768
24. 外部存储器控制器（EXMC） 769
24.1. 简介 769
24.2. 主要特性 769

24.3. 功能描述 769
24.3.1. 结构框图 769
24.3.2. EXMC 访问基本规范 770
24.3.3. 外部设备地址映射 770
24.3.4. NOR/PSRAM 控制器 774
24.3.5. NAND Flash 或 PC Card 控制器 792
24.4. EXMC 寄存器 797
24.4.1. NOR/PSRAM 控制器寄存器 797
24.4.2. NAND Flash/PC Card 控制器寄存器 801
25. 控制器局域网络（CAN） 808
25.1. 简介 808
25.2. 主要特征 808
25.3. 功能说明 809
25.3.1. 工作模式 809
25.3.2. 通信模式 810
25.3.3. 数据发送 811
25.3.4. 数据接收 813
25.3.5. 过滤功能 814
25.3.6. 时间触发通信 817
25.3.7. 通信参数 817
25.3.8. CAN-FD 操作 819
25.3.9. 传输延迟补偿 820
25.3.10. 错误标志 820
25.3.11. 中断 821
25.4. CAN 寄存器 823
25.4.1. 控制寄存器（CAN_CTL） 823
25.4.2. 状态寄存器（CAN_STAT） 824
25.4.3. 发送状态寄存器（CAN_TSTAT） 826
25.4.4. 接收 FIFO0 寄存器（CAN_RFIFO0） 828
25.4.5. 接收 FIFO1 寄存器（CAN_RFIFO1） 829
25.4.6. 中断使能寄存器（CAN_INTEN） 829
25.4.7. 错误寄存器（CAN_ERR） 831
25.4.8. 位时序寄存器（CAN_BT） 832
25.4.9. FD 控制寄存器（CAN_FDCTL） 833
25.4.10. FD 状态寄存器（CAN_FDSTAT） 834
25.4.11. FD 传输延迟补偿寄存器（CAN_FDTDC） 834
25.4.12. 数据位时序寄存器（CAN_DBT） 835
25.4.13. 发送邮箱标识符寄存器（CAN_TMIx）（x = 0…2） 836
25.4.14. 发送邮箱属性寄存器（CAN_TMPx）（x = 0...2） 836
25.4.15. 发送邮箱 data0 寄存器（CAN_TMDATA0x）（x = 0...2） 837
25.4.16. 发送邮箱 data1 寄存器（CAN_TMDATA1x）（x = 0...2） 838
25.4.17. 接收 FIFO 邮箱标识符寄存器（CAN_RFIFOMIx）（x = 0,1） 838

25.4.18. 接收 FIFO 邮箱属性寄存器（CAN_RFIFOMPx）（x = 0,1） 839
25.4.19. 接收 FIFO 邮箱 data0 寄存器（CAN_RFIFOMDATA0x）（x = 0,1） 840
25.4.20. 接收 FIFO 邮箱 data1 寄存器（CAN_RFIFOMDATA1x）（x = 0,1） 840
25.4.21. 过滤器控制寄存器（CAN_FCTL） 841
25.4.22. 过滤器模式配置寄存器（CAN_FMCFG） 841
25.4.23. 过滤器位宽配置寄存器（CAN_FSCFG） 842
25.4.24. 过滤器关联 FIFO 寄存器（CAN_FAFIFO） 843
25.4.25. 过滤器激活寄存器（CAN_FW） 844
25.4.26. 过滤器（x）数据（y）寄存器（CAN_FxDATAy）（x = 0…27, y = 0,1） 845

26. 以太网（ENET） 846

26.1. 简介 846

26.2. 主要特性 846
26.2.1. 模块框图 847
26.2.2. MAC 802.3 以太网数据包描述 848
26.2.3. 以太网信号描述 848

26.3. 功能描述 850
26.3.1. 接口配置 850
26.3.2. MAC 功能简介 854
26.3.3. DMA 控制器描述 862
26.3.4. MAC 统计计数器：MSC 884
26.3.5. 唤醒管理：WUM 884
26.3.6. 精确时间协议：PTP 887
26.3.7. 典型的以太网配置流程示例 890
26.3.8. 以太网中断 891

26.4. ENET 寄存器 894
26.4.1. MAC 配置寄存器（ENET_MAC_CFG） 894
26.4.2. MAC 帧过滤器寄存器（ENET_MAC_FRMF） 896
26.4.3. MAC hash 列表高寄存器（ENET_MAC_HLH） 898
26.4.4. MAC hash 列表低寄存器（ENET_MAC_HLL） 898
26.4.5. MAC PHY 控制寄存器（ENET_MAC_PHY_CTL） 899
26.4.6. MAC PHY 数据寄存器（ENET_MAC_PHY_DATA） 899
26.4.7. MAC 流控寄存器（ENET_MAC_FCTL） 900
26.4.8. MAC VLAN 标签寄存器（ENET_MAC_VLT） 901
26.4.9. MAC 远程唤醒帧过滤器寄存器（ENET_MAC_RWFF） 902
26.4.10. MAC 唤醒管理寄存器（ENET_MAC_WUM） 903
26.4.11. MAC 调试寄存器（ENET_MAC_DBG） 904
26.4.12. MAC 中断状态寄存器（ENET_MAC_INTF） 906
26.4.13. MAC 中断屏蔽寄存器（ENET_MAC_INTMSK） 907
26.4.14. MAC 地址 0 高寄存器（ENET_MAC_ADDR0H） 907
26.4.15. MAC 地址 0 低寄存器（ENET_MAC_ADDR0L） 908
26.4.16. MAC 地址 1 高寄存器（ENET_MAC_ADDR1H） 908
26.4.17. MAC 地址 1 低寄存器（ENET_MAC_ADDR1L） 909

26.4.18. MAC 地址 2 高寄存器（ENET_ MAC_ADDR2H）.....909
26.4.19. MAC 地址 2 低寄存器（ENET_MAC_ADDR2L）.....910
26.4.20. MAC 地址 3 高寄存器（ENET_MAC_ADDR3H）.....910
26.4.21. MAC 地址 3 低寄存器（ENET_MAC_ADDR3L）.....911
26.4.22. MAC 流控阈值寄存器（ENET_MAC_FCTH）.....912
26.4.23. MSC 控制寄存（ENET_MSC_CTL）.....913
26.4.24. MSC 接收中断状态寄存器（ENET_MSC_RINTF）.....913
26.4.25. MSC 发送中断状态寄存器（ENET_MSC_TINTF）.....914
26.4.26. MSC 接收中断屏蔽寄存器（ENET_MSC_RINTMSK）.....915
26.4.27. MSC 发送中断屏蔽寄存器（ENET_MSC_TINTMSK）.....916
26.4.28. MSC 1 次冲突后发送”好”帧的计数器寄存器（ENET_MSC_SCCNT）.....916
26.4.29. MSC 1 次以上冲突后发送”好”帧的计数器寄存器（ENET_MSC_MSCCNT）.....917
26.4.30. MSC 发送”好”帧计数器寄存器（ENET_MSC_TGFCNT）.....917
26.4.31. MSC CRC 错误接收帧计数器寄存器（ENET_MSC_RFCECNT）.....918
26.4.32. MSC 对齐错误接收帧计数器寄存器（ENET_MSC_RFAECNT）.....918
26.4.33. MSC“好”单播帧接收帧计数器寄存器（ENET_MSC_RGUFCNT）.....918
26.4.34. PTP 时间戳控制寄存器（ENET_PTP_TSCTL）.....919
26.4.35. PTP 亚秒递增寄存器（ENET_PTP_SSINC）.....921
26.4.36. PTP 时间戳高寄存器（ENET_PTP_TSH）.....922
26.4.37. PTP 时间戳低寄存器（ENET_PTP_TSL）.....922
26.4.38. PTP 时间戳高更新寄存器（ENET_PTP_TSUH）.....923
26.4.39. PTP 时间戳低更新寄存器（ENET_PTP_TSUL）.....923
26.4.40. PTP 时间戳加数寄存器（ENET_PTP_TSADDEND）.....924
26.4.41. PTP 期望时间高寄存器（ENET_PTP_ETH）.....924
26.4.42. PTP 期望时间低寄存器（ENET_PTP_ETL）.....924
26.4.43. PTP 时间戳标志寄存器（ENET_PTP_TSF）.....925
26.4.44. PTP PPS 控制寄存器（ENET_PTP_PPSCTL）.....925
26.4.45. DMA 总线控制寄存器（ENET_DMA_BCTL）.....926
26.4.46. DMA 发送查询使能寄存器（ENET_DMA_TPEN）.....928
26.4.47. DMA 接收查询使能寄存器（ENET_DMA_RPEN）.....928
26.4.48. DMA 接收描述符列表地址寄存器（ENET_DMA_RDTADDR）.....929
26.4.49. DMA 发送描述符列表地址寄存器（ENET_DMA_TDTADDR）.....929
26.4.50. DMA 状态寄存器（ENET_DMA_STAT）.....930
26.4.51. DMA 控制寄存器（ENET_DMA_CTL）.....933
26.4.52. DMA 中断使能寄存器（ENET_DMA_INTEN）.....936
26.4.53. DMA 丢失帧和缓存溢出计数器寄存器（ENET_DMA_MFBOCNT）.....938
26.4.54. DMA 接收状态看门狗计数器寄存器（ENET_DMA_RSWDC）.....938
26.4.55. DMA 当前发送描述符地址寄存器（ENET_DMA_CTDADDR）.....939
26.4.56. DMA 当前接收描述符地址寄存器（ENET_DMA_CRDADDR）.....939
26.4.57. DMA 当前发送缓存地址寄存器（ENET_DMA_CTBADDR）.....940
26.4.58. DMA 当前接收缓存地址寄存器（ENET_DMA_CRBADDR）.....940

27. 通用串行总线高速接口（USBHS）.....941

27.1. 概述.....941

27.2. 主要特性 941
27.3. 结构框图 942
27.4. 信号线描述 942
27.5. 功能描述 942
27.5.1. USBHS PHY 选择、时钟及工作模式 942
27.5.2. USB 主机功能 946
27.5.3. USB 设备功能 948
27.5.4. OTG 功能概述 949
27.5.5. 数据 FIFO 950
27.5.6. DMA 功能 952
27.5.7. 操作手册 953
27.6. 中断 958
27.7. USBHS 寄存器 960
27.7.1. 全局控制与状态寄存器组 960
27.7.2. 主机控制和状态寄存器 988
27.7.3. 设备控制和状态寄存器 1000
27.7.4. 电源和时钟控制寄存器（USBHS_PWRCLKCTL） 1027
28. 附录 1029
28.1. 寄存器表中使用的缩写列表 1029
28.2. 术语表 1029
28.3. 可用外设 1029
29. 版本历史 1030

图索引

图 1-1. Cortex®-M33 处理器结构框图 31
图 1-2. GD32A508xx 系列器件的系统架构示意图 33
图 2-1. 页擦除操作流程 45
图 2-2. 整片擦除操作流程 47
图 2-3. 字编程操作流程 48
图 4-1. 电源域概览 65
图 4-2. V_{CORE} 电源域概览 66
图 4-3. 上电/掉电复位波形图 68
图 4-4. LVD 阈值波形图 68
图 5-1. 系统复位电路 79
图 5-2. 时钟树 81
图 5-3. HXTAL 时钟源 83
图 5-4. 旁路模式下 HXTAL 时钟源 83
图 6-1. CTC 简介 123
图 6-2. CTC 校准计数器 125
图 7-1. EXTI 框图 137
图 8-1. GPIO 端口位的基本结构 143
图 8-2. 输入配置的基本结构 144
图 8-3. 输出配置的基本结构 145
图 8-4. 模拟配置的基本结构 145
图 8-5. 备用功能配置的基本结构 146
图 9-1. CRC 管理单元框图 181
图 10-1. 三角函数加速器模块结构图 186
图 10-2. 基于 X 与 Y 的比值计算 R0 和 R1 191
图 10-3. TMU 软件流程 192
图 11-1. DMA 结构框图 198
图 11-2. 握手机制 201
图 11-3. DMA 中断逻辑图 202
图 11-4. DMA0 请求映射 203
图 11-5. DMA1 请求映射 205
图 13-1. ADC 模块框图（ADC0 和 ADC1） 220
图 13-2. 单次运行模式 222
图 13-3. 连续运行模式 223
图 13-4. 扫描运行模式，且连续运行模式禁能 223
图 13-5. 扫描运行模式，连续运行模式使能 224
图 13-6. 间断运行模式 224
图 13-7. 12 位数据存储模式 225
图 13-8. 6 位数据存储模式 225
图 13-9. 20 位到 16 位的结果截断 228
图 13-10. 右移 5 位和取整的数例 228
图 13-11. ADC 同步框图 229

图 13-12. 基于 10 个通道的常规并行模式 230
图 13-13. 常规序列上的快速交叉模式（两个 ADC 的 CTN=1） 231
图 13-14. 常规序列上的慢速交叉模式 232
图 14-1. DAC 结构框图 247
图 14-2. DAC LFSR 算法 250
图 14-3. DAC 三角噪声模式生成的波形 250
图 15-1.比较器框图 265
图 15-2. 比较器的输出消隐 267
图 16-1. 独立看门狗定时器框图 274
图 16-2. 窗口看门狗定时器框图 280
图 16-3. 窗口看门狗定时器时序图 281
图 17-1. RTC 框图 285
图 17-2. RTC 秒信号及闹钟信号的波形(RTC_PSC = 3,RTC_ALRM = 2) 287
图 17-3. RTC 秒信号及溢出信号的波形(RTC_PSC = 3) 287
图 18-1. 高级定时器结构框图 294
图 18-2. 内部时钟分频为 1 时，计数器的时序图 295
图 18-3. 当 PSC 数值从 0 变到 2 时，计数器的时序图 296
图 18-4. 向上计数时序图，PSC=0/2 297
图 18-5. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 298
图 18-6. 向下计数时序图，PSC=0/2 299
图 18-7. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值 299
图 18-8. 中央计数模式计数器时序图 300
图 18-9. 中央计数模式下计数器重复时序图 301
图 18-10. 在向上计数模式下计数器重复时序图 302
图 18-11. 在向下计数模式下计数器重复时序图 302
图 18-12. 输入捕获原理 303
图 18-13. 输出比较原理（带有互补输出的通道，x=0,1,2） 304
图 18-14. 输出比较原理 305
图 18-15. 三种输出比较模式 306
图 18-16. EAPWM 时序图 307
图 18-17. CAPWM 时序图 307
图 18-18. 带死区时间的互补输出 309
图 18-19. 通道响应中止输入（高电平有效）时，输出信号的行为 310
图 18-20. 在正交编码器模式 2 且 CI0FE0 极性不反相时计数器行为 312
图 18-21. 在正交译码器模式 2 且 CI0FE0 极性反相时计数器行为 312
图 18-22. 霍尔传感器用在 BLDC 电机控制中 313
图 18-23. 两个定时器之间的霍尔传感器时序图 313
图 18-24. 复位模式 315
图 18-25. 暂停模式下的控制电路 315
图 18-26. 事件模式 316
图 18-27. 单脉冲模式，TIMERx_CHxCV = 4，TIMERx_CAR=99 316
图 18-28. 定时器 0 主/从模式的例子 317
图 18-29. 用定时器 2 的 CI0 输入来触发定时器 0 和定时器 2 318
图 18-30. 通用定时器 L0 结构框图 345

图 18-31. 内部时钟分频为 1 时，计数器的时序图 346
图 18-32. 当 PSC 数值从 0 变到 2 时，计数器的时序图 347
图 18-33. 向上计数时序图，PSC=0/2 348
图 18-34. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 349
图 18-35. 向下计数时序图，PSC=0/2 350
图 18-36. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值 350
图 18-37. 中央计数模式计数器时序图 351
图 18-38. 通道输入捕获原理 352
图 18-39. 输出比较原理（x=0,1,2,3） 354
图 18-40. 三种输出比较模式 355
图 18-41. EAPWM 时序图 356
图 18-42. CAPWM 时序图 356
图 18-43. 复位模式 358
图 18-44. 暂停模式 358
图 18-45. 事件模式 359
图 18-46. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99 359
图 18-47. 通用定时器 L1 结构框图 385
图 18-48. 内部时钟分频为 1 时，计数器的时序图 386
图 18-49. 当 PSC 数值从 0 变到 2 时，计数器的时序图 387
图 18-50. 向上计数时序图，PSC=0/2 387
图 18-51. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 388
图 18-52. 通道输入捕获原理 389
图 18-53. 通道输出比较原理（x=0，1） 390
图 18-54. 三种输出比较模式 391
图 18-55. EAPWM 时序图 392
图 18-56. 复位模式下的控制电路 393
图 18-57. 暂停模式下的控制电路 393
图 18-58. 事件模式下的控制电路 394
图 18-59. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99 394
图 18-60. 通用定时器 L2 结构框图 408
图 18-61. 内部时钟分频为 1 时，计数器的时序图 409
图 18-62. 当 PSC 数值从 0 变到 2 时，计数器的时序图 409
图 18-63. 向上计数时序图，PSC=0/2 410
图 18-64. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 411
图 18-65. 通道输入捕获原理 412
图 18-66. 通道输出比较原理 413
图 18-67. 三种输出比较模式 414
图 18-68. 基本定时器结构框图 426
图 18-69. 内部时钟分频为 1 时，计数器的时序图 426
图 18-70. 当 PSC 数值从 0 变到 2 时，计数器的时序图 427
图 18-71. 向上计数时序图，PSC=0/2 428
图 18-72. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 428
图 19-1. SHRTIMER 结构框图 435
图 19-2. Master_TIMER 结构图 436

图 19-3. 分频为 32 时计数器时钟 438
图 19-4. 单脉冲模式下计数器的动作 439
图 19-5. 连续模式下计数器的动作 439
图 19-6. 连续模式下重复计数器的动作 440
图 19-7. 单脉冲模式下，CNTRSTM = 0 时重复计数器的动作 440
图 19-8. 单脉冲模式下，CNTRSTM = 1 时重复计数器的动作 440
图 19-9. 当预分频系数为 128 时，复位事件重新同步 441
图 19-10. Slave_TIMERx 结构框图 444
图 19-11. EXEV0 和 EXEV1 触发捕获 0 447
图 19-12. STxCAR=0x8, STxCMP1V=0x02 时，比较 1 寄存器的动作 448
图 19-13. 比较延迟模式框图 448
图 19-14. 比较 1 延迟模式 0 449
图 19-15. 比较 1 延迟模式 1 450
图 19-16.比较延迟模式（SHWEN=0） 451
图 19-17. 通道输出结构图 452
图 19-18. O0PRE 波形：CMP0 事件置位，CMP1 事件复位 453
图 19-19. 每个 $t_{SHRTMER_CK}$ 周期的仲裁机制过程 454
图 19-20. 仲裁机制示例 456
图 19-21. 脉冲宽度为 1 个 $t_{SHRTMER_CK}$ 周期 456
图 19-22.脉冲宽度为 2 个 $t_{SHRTMER_CK}$ 周期 457
图 19-23. 超高分辨率 OxPRE 波形 457
图 19-24. CNTCKDIV[3:0] = 4’b0110 时的 OxPRE 波形 458
图 19-25. 常规模式下的 C0OPRE 波形 458
图 19-26. 具有死区时间的 C0OPRE 和 C1OPRE 互补输出波形 459
图 19-27. 脉宽小于死区时间时的互补波形 460
图 19-28. 均衡模式的结构框图 461
图 19-29. 均衡模式下的 C0OPRE 和 C1OPRE 波形 462
图 19-30. 延迟空闲模式，ISO0 = 0 和 CHOP = 0 464
图 19-31. 延迟空闲模式，ISO0 = 1 和 CHOP = 0 464
图 19-32. 延迟空闲模式，ISO0 = 0 和 CHOP = 1 465
图 19-33. 延迟空闲模式，ISO0 = 1 和 CHOP = 1 465
图 19-34. 均衡空闲模式，ISO0 = 0 和 ISO1 = 0 466
图 19-35. CHyP=0 或 CHyP=1 时的 STxCHy_O 波形 468
图 19-36. 载波信号结构框图 469
图 19-37. 载波模式使能时的 SHRTIMER 输出 470
图 19-38. 消隐模式和窗口模式 472
图 19-39. 突发模式时序图 475
图 19-40. 突发模式的常规进入 476
图 19-41. 突发模式的延迟进入 476
图 19-42. 模拟突发模式示例 478
图 19-43. 外部事件 y（y=0..4）处理过程框图 479
图 19-44. 外部事件 y（y=5..9）处理过程框图 480
图 19-45. 故障输入结构图 481
图 19-46. ADC 触发源选择图 482

图 19-47. DAC 触发源选择图 483
图 19-48. DMA 模式运行流程图 486
图 20-1. USART 模块内部框图 599
图 20-2. USART 字符帧 （8 数据位和 1 停止位） 600
图 20-3. USART 发送步骤 602
图 20-4. 过采样方式接收一个数据位（OSB=0） 603
图 20-5. 采用 DMA 方式实现 USART 数据发送配置步骤 604
图 20-6. 采用 DMA 方式实现 USART 数据接收配置步骤 604
图 20-7. 两个 USART 之间的硬件流控制 605
图 20-8. 硬件流控制 606
图 20-9. 空闲状态下检测断开帧 607
图 20-10. 数据传输过程中检测断开帧 607
图 20-11. 同步模式下的 USART 示例 608
图 20-12. 8-bit 格式的 USART 同步通信波形（CLEN=1） 608
图 20-13. IrDA SIR ENDEC 模块 608
图 20-14. IrDA 数据调制 609
图 20-15. ISO7816-3 数据帧格式 610
图 20-16. USART 中断映射框图 612
图 20-17. USART 模块内部框图 627
图 20-18. USART 字符帧（8 数据位和 1 停止位） 628
图 20-19. USART 发送步骤 630
图 20-20. 过采样方式接收一个数据位（OSB=0） 631
图 20-21. 采用 DMA 方式实现 USART 数据发送配置步骤 632
图 20-22. 采用 DMA 方式实现 USART 数据接收配置步骤 632
图 20-23. 空闲状态下检测断开帧 634
图 20-24. 数据传输过程中检测断开帧 634
图 20-25. 同步模式下的 USART 示例 634
图 20-26. 8-bit 格式的 USART 同步通信波形（CLEN=1） 635
图 20-27. IrDA SIR ENDEC 模块 635
图 20-28. IrDA 数据调制 636
图 20-29. ISO7816-3 数据帧格式 636
图 20-30. USART 接收 FIFO 结构 638
图 20-31. USART 中断映射框图 640
图 21-1. I2C 模块框图 657
图 21-2. 数据有效性 658
图 21-3. 起始和停止信号 659
图 21-4. 时钟同步 659
图 21-5. SDA 线仲裁 660
图 21-6. 7 位地址的 I2C 通讯流程 660
图 21-7. 10 位地址的 I2C 通讯流程（主机发送） 660
图 21-8. 10 位地址的 I2C 通讯流程（主机接收） 660
图 21-9. 从机发送模式（10 位地址模式） 662
图 21-10. 从机接收模式（10 位地址模式） 663
图 21-11. 主机发送模式（10 位地址模式） 664

图 21-12. 主机接收使用方案 A 模式（10 位地址模式） 665
图 21-13. 主机接收使用方案 B 模式（10 位地址模式） 667
图 21-14. I2C 模块框图 685
图 21-15. 数据有效性 687
图 21-16. 开始和停止信号 687
图 21-17. 10 位地址的 I2C 通讯流程（主机发送） 688
图 21-18. 7 位地址的 I2C 通讯流程（主机发送） 688
图 21-19. 7 位地址的 I2C 通讯流程（主机接收） 688
图 21-20. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R=0） 688
图 21-21. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R=1） 688
图 21-22. 数据保持时间 689
图 21-23. 数据建立时间 689
图 21-24. 数据发送 691
图 21-25. 数据接收 692
图 21-26. I2C 从机初始化 694
图 21-27. I2C 从机发送编程模型（SS=0） 695
图 21-28. I2C 从机发送编程模型（SS=1） 695
图 21-29. I2C 从机接收编程模型 696
图 21-30. I2C 主机初始化 698
图 21-31. I2C 主机发送编程模型（N<=255） 698
图 21-32. I2C 主机发送编程模型（N>255） 699
图 21-33. I2C 主机接收编程模型（N<=255） 700
图 21-34. I2C 主机接收编程模型（N>255） 701
图 21-35. SMBus 主机发送器和从机接收器通信流程 705
图 21-36. SMBus 主机接收器和从机发送器通信流程 705
图 22-1. SPI 结构框图 722
图 22-2. SPI 常规模式下的时序图 723
图 22-3. SPI 四线模式下的 SPI 时序图（CKPL=1, CKPH=1, LF=0） 723
图 22-4. 典型的全双工模式连接 726
图 22-5. 典型的单工模式连接（主机：接收，从机：发送） 726
图 22-6. 典型的单工模式连接（主机：只发送，从机：接收） 727
图 22-7. 典型的双向线连接 727
图 22-8. 主机 TI 模式在不连续发送时的时序图 729
图 22-9. 主机 TI 模式在连续发送时的时序图 729
图 22-10. 从机 TI 模式时序图 729
图 22-11. NSS 脉冲模式时序图（主机连续发送） 730
图 22-12. SPI 四线模式写操作时序图 731
图 22-13. SPI 四线模式读操作时序图 731
图 22-14. I2S 结构框图 735
图 22-15. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=0, CKPL=0） 736
图 22-16. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=0, CKPL=1） 736
图 22-17. I2S 飞利浦标准时序图（DTLEN=10, CHLEN=1, CKPL=0） 736
图 22-18. I2S 飞利浦标准时序图（DTLEN=10, CHLEN=1, CKPL=1） 737
图 22-19. I2S 飞利浦标准时序图（DTLEN=01, CHLEN=1, CKPL=0） 737

图 22-20. I2S 飞利浦标准时序图（DTLEN=01, CHLEN=1, CKPL=1） 737
图 22-21. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=1, CKPL=0） 737
图 22-22. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=1, CKPL=1） 737
图 22-23. MSB 对齐标准时序图（DTLEN=00, CHLEN=0, CKPL=0） 738
图 22-24. MSB 对齐标准时序图（DTLEN=00, CHLEN=0, CKPL=1） 738
图 22-25. MSB 对齐标准时序图（DTLEN=10, CHLEN=1, CKPL=0） 738
图 22-26. MSB 对齐标准时序图（DTLEN=10, CHLEN=1, CKPL=1） 738
图 22-27. MSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=0） 738
图 22-28. MSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=1） 739
图 22-29. MSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=0） 739
图 22-30. MSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=1） 739
图 22-31. LSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=0） 739
图 22-32. LSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=1） 739
图 22-33. LSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=0） 740
图 22-34. LSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=1） 740
图 22-35. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=0） 740
图 22-36. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=1） 740
图 22-37. PCM 标准短帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=0） 741
图 22-38. PCM 标准短帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=1） 741
图 22-39. PCM 标准短帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=0） 741
图 22-40. PCM 标准短帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=1） 741
图 22-41. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=0） 741
图 22-42. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=1） 741
图 22-43. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=0） 742
图 22-44. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=1） 742
图 22-45. PCM 标准长帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=0） 742
图 22-46. PCM 标准长帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=1） 742
图 22-47. PCM 标准长帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=0） 742
图 22-48. PCM 标准长帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=1） 742
图 22-49. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=0） 743
图 22-50. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=1） 743
图 22-51. I2S 时钟生成结构框图 743
图 22-52. I2S 初始化流程 744
图 22-53. I2S 主机接收禁能流程 746
图 23-1. SQPI 采样极性 761
图 23-2. SQPI_SCMD 示例 761
图 23-3. SQPI_RDID 示例(SQPI_IDLEN=00) 762
图 23-4. SQPI_CLK 示例 763
图 23-5. SQPI SSS 模式时序(SPI) 763
图 23-6. SQPI SSQ 模式时序 764
图 23-7. SQPI SQQ 模式时序(SQPI) 764
图 23-8. SQPI QQQ 模式时序(QPI) 764
图 23-9. SQPI SSD 模式时序 764
图 23-10. SQPI SDD 模式时序 765

图 24-1. 系统架构 769
图 24-2. EXMC Bank 划分 770
图 24-3. Bank0 地址映射 771
图 24-4. NAND/PC Card 地址映射 772
图 24-5. Bank1 通用空间 773
图 24-6. 模式 1 读时序 777
图 24-7. 模式 1 写时序 777
图 24-8. 模式 A 读时序 778
图 24-9. 模式 A 写时序 778
图 24-10. 模式 2/B 读时序 780
图 24-11. 模式 2 写时序 780
图 24-12. 模式 B 写时序 780
图 24-13. 模式 C 读时序 782
图 24-14. 模式 C 写时序 782
图 24-15. 模式 D 读时序 783
图 24-16. 模式 D 写时序 784
图 24-17. 复用模式读时序 785
图 24-18. 复用模式写时序 785
图 24-19. 异步等待有效时的读时序 787
图 24-20. 异步等待有效时的写时序 787
图 24-21. 同步复用突发传输读时序 789
图 24-22. 同步复用突发传输写时序 790
图 24-23. PC Card 通用空间操作时序 793
图 24-24. NCE 敏感 NAND Flash 访问时序 794
图 25-1. CAN 模块结构框图 809
图 25-2. 发送寄存器 811
图 25-3. 发送邮箱状态转换 811
图 25-4. 接收寄存器 813
图 25-5. 32-bit 位宽过滤器 814
图 25-6. 16-bit 位宽过滤器 814
图 25-7. 32-bit 位宽掩码模式过滤器 815
图 25-8. 16-bit 位宽掩码模式过滤器 815
图 25-9. 32-bit 位宽列表模式过滤器 815
图 25-10. 16-bit 位宽列表模式过滤器 815
图 25-11. 位时序 818
图 25-12 传输延迟测量 820
图 26-1. 以太网模块框图 847
图 26-2. MAC/带标签的 MAC 帧格式 848
图 26-3. 站点管理接口信号 850
图 26-4. 媒体独立接口（MII）信号线 852
图 26-5. 精简媒体独立接口（RMII）信号线 853
图 26-6. 描述符的环结构和链结构 863
图 26-7. 常规发送描述符 867
图 26-8. 增强发送描述符 873

图 26-9. 常规接收描述符 876
图 26-10. 增强接收描述符 881
图 26-11. 唤醒帧过滤器寄存器 886
图 26-12. 系统时钟精细校准方法 887
图 26-13. MAC 控制器中断示意图 892
图 26-14. 以太网中断示意图 892
图 26-15. 远程唤醒帧过滤器寄存器 902
图 27-1. USBHS 结构框图 942
图 27-2. 在主机或设备模式下连接示意图 943
图 27-3. OTG 模式下使用内部嵌入式 PHY 连接示意图 944
图 27-4. 使用外部 ULPI PHY 的连接示意图 945
图 27-5. 主机端口状态转移图 946
图 27-6. 主机模式 FIFO 空间 950
图 27-7. 主机模式 FIFO 访问寄存器映射表 951
图 27-8. 设备模式 FIFO 空间 951
图 27-9. 设备模式 FIFO 访问寄存器映射表 952

表索引

表 1-1. AHB 互联矩阵的互联关系列表 32
表 1-2. GD32A508xx 系列器件的存储器映射表 35
表 1-3. 引导模式 39
表 2-1. GD32A508xx 闪存基地址和构成 42
表 2-2. WSCNT 与 AHB 时钟频率对应关系 43
表 2-3. 选项字节 50
表 2-4. OB_WP 位对应页保护 51
表 4-1. 节电模式总结 71
表 5-1. 时钟输出 0 的时钟源选择 85
表 5-2. 深度睡眠模式下 1.1V 域电压选择 86
表 7-1. Cortex®-M33 中的 NVIC 异常类型 133
表 7-2. 中断向量表 134
表 7-3. EXTI 触发源 138
表 8-1. GPIO 配置表 142
表 8-2. 调试接口信号 147
表 8-3. 调试端口和引脚可用性 147
表 8-4. TIMER 备用功能重映射 148
表 8-5. TMER4 备用功能重映射[1] 149
表 8-6. USART0/1 备用功能重映射 149
表 8-7. I2C0 备用功能重映射 150
表 8-8. SPI0/SPI2/I2S 备用功能重映射 150
表 8-9. CAN0/1 备用功能重映射 151
表 8-10. ENET 备用功能重映射 151
表 8-11. CTC 备用功能重映射 152
表 8-12. OSC32 引脚配置 152
表 8-13. OSC 引脚配置 152
表 10-1. 9 种不同的操作模式 186
表 10-2. IEEE-32 位单精度浮点格式 187
表 10-3. 模式 0 下单位值与弧度之间的转换 188
表 10-4. 模式 1 下弧度与单位值之间的转换 188
表 10-5. 模式 5 下输入的操作数与 R0 值的范围 190
表 10-6. 模式 6 下产生 UDRF 和 OVRF 的条件 191
表 10-7. 模式 7 下产生 UDRF 和 OVRF 的条件 192
表 11-1. DMA 传输操作 199
表 11-2. 中断事件 202
表 11-3. DMA0 各通道请求表 204
表 11-4. DMA1 各通道请求表 205
表 13-1. ADC 内部输入信号 220
表 13-2. ADC 输入引脚定义 220
表 13-3. ADC0 和 ADC1 的外部触发源 226
表 13-4. 不同分辨率对应的 t_{CONV} 时间 227

表 13-5. 不同 N 和 M 组合的最大输出值（灰色值表示截断） 228
表 13-6. ADC 同步模式表 229
表 14-1. DAC 引脚 248
表 14-2. DAC 触发与输出 248
表 14-3. DAC 外部触发 249
表 15-1. CMP 的输入和输出总结 266
表 16-1. 独立看门狗定时器在 40kHz（IRC40K）时的最小/最大超时周期 275
表 16-2.在 90MHz（f_{PCLK1}）时的最大/最小超时值 281
表 18-1. 定时器（TIMERx）分为五种类型 293
表 18-2. 由参数控制的互补输出表 308
表 18-3. 不同正交译码器模式下的计数方向 311
表 18-4. 从模式例子列表 314
表 18-5. 从模式列表和举例 357
表 18-6. 从模式列表和举例 392
表 19-1. 自动重载寄存器和比较 y（y = 0..3）寄存器的限值 437
表 19-2. $f_{SHRTIMER_CK}$ = 180MHz 的分辨率 438
表 19-3. Master_TIMER 影子寄存器和更新事件 442
表 19-4. 计数器和捕获 y(y=0,1)寄存器限值 445
表 19-5. Slave_TIMER 内部连接事件 454
表 19-6. 交叉开关和空闲控制同时运行 463
表 19-7. 进入和退出空闲状态的请求 463
表 19-8. 突发模式控制的空闲状态时的输出 467
表 19-9. 输出级状态编程（x=0..4, y=0,1） 468
表 19-10. Slave_TIMERx 影子寄存器和更新事件 471
表 19-11. STxUPINy（y=0..2）和芯片内部信号 471
表 19-12. 消隐模式下的滤波信号映射 472
表 19-13. 窗口模式的滤波信号映射 473
表 19-14. 突发模式的芯片内部信号 474
表 19-15. 外部事件映射 480
表 19-16. 故障通道映射 481
表 19-17. 中断映射 484
表 19-18. DMA 请求映射 485
表 20-1. USART 重要引脚描述 599
表 20-2. 停止位配置 600
表 20-3. USART 中断请求 611
表 20-4. USART 重要引脚描述 627
表 20-5. 停止位配置 628
表 20-6. USART 中断请求 639
表 21-1. I2C 总线术语说明（参考飞利浦 I2C 规范） 658
表 21-2. 事件状态标志位 671
表 21-3. 错误标志位 671
表 21-4. I2C 总线术语说明（参考飞利浦 I2C 规范） 685
表 21-5. 数据建立时间和数据保持时间 690
表 21-6. 可关闭通信模式 692

表 21-7. I2C 错误标志 706
表 21-8. I2C 中断事件 706
表 22-1. SPI 信号描述 722
表 22-2. SPI 四线信号描述 723
表 22-3. 从机模式 NSS 功能 724
表 22-4. 主机模式 NSS 功能 725
表 22-5. SPI 运行模式 725
表 22-6. SPI 中断请求 734
表 22-7. I2S 比特率计算公式 743
表 22-8. 音频采样频率计算公式 744
表 22-9. 各种运行模式下 I2S 接口信号的方向 744
表 22-10. I2S 中断 749
表 23-1. SQPI 控制器模式定义 760
表 24-1. NOR Flash 接口信号描述 774
表 24-2. PSRAM 非复用接口信号描述 774
表 24-3. EXMC 的 Bank0 支持的所有处理 775
表 24-4. NOR/PSRAM 控制时序参数 776
表 24-5. EXMC 时序模型 776
表 24-6. 模式 1 相关寄存器配置 777
表 24-7. 模式 A 相关寄存器配置 779
表 24-8. 模式 2/B 相关寄存器配置 781
表 24-9. 模式 C 相关寄存器配置 782
表 24-10. 模式 D 相关寄存器配置 784
表 24-11. 复用模式相关寄存器配置 786
表 24-12. 同步复用模式读时序配置 789
表 24-13. 同步复用模式写时序配置 791
表 24-14. 8 位/16 位 NAND 接口信号描述 792
表 24-15. 16 位 PC Card 接口信号描述 792
表 24-16. Bank1/2/3 支持的访问模式 792
表 24-17. NADN/PC Card 可编程参数 793
表 25-1. 32-bit 过滤序号 815
表 25-2. 过滤索引 816
表 25-3. CAN 事件/中断标志 822
表 26-1. 以太网信号（MII 默认） 848
表 26-2. 以太网信号（MII 重映射） 849
表 26-3. 以太网信号（RMII 默认） 849
表 26-4. 以太网信号（RMII 重映射） 850
表 26-5. 时钟范围 851
表 26-6. 接收接口信号编码 853
表 26-7. 目标地址过滤器结果列表 857
表 26-8. 源地址过滤器结果列表 858
表 26-9. 接收描述符 0 错误状态描述，仅适用于常规描述符（DFM=0） 879
表 26-10. 支持的 PTP 时间戳及其寄存器配置 921
表 27-1. USBHS 信号线描述 942

表 27-2. USBHS 支持速度列表 943
表 27-3. USBHS 全局中断 958
表 28-1. 寄存器功能位访问属性 1029
表 28-2. 术语 1029
表 29-1. 版本历史 1030

1. 系统及存储器架构

GD32A508xx 系列器件是基于 Arm® Cortex®-M33 处理器的 32 位通用微控制器。Cortex®-M33 处理器包含数据总线和系统总线这两条 AHB 总线，所有对 Cortex®-M33 处理器的存储访问，根据不同的目的和目标存储空间，都会在 AHB 总线上执行。存储器的组织采用了哈弗结构，预先定义的存储器映射和高达 4GB 的存储空间，充分保证了系统的灵活性和可扩展性。

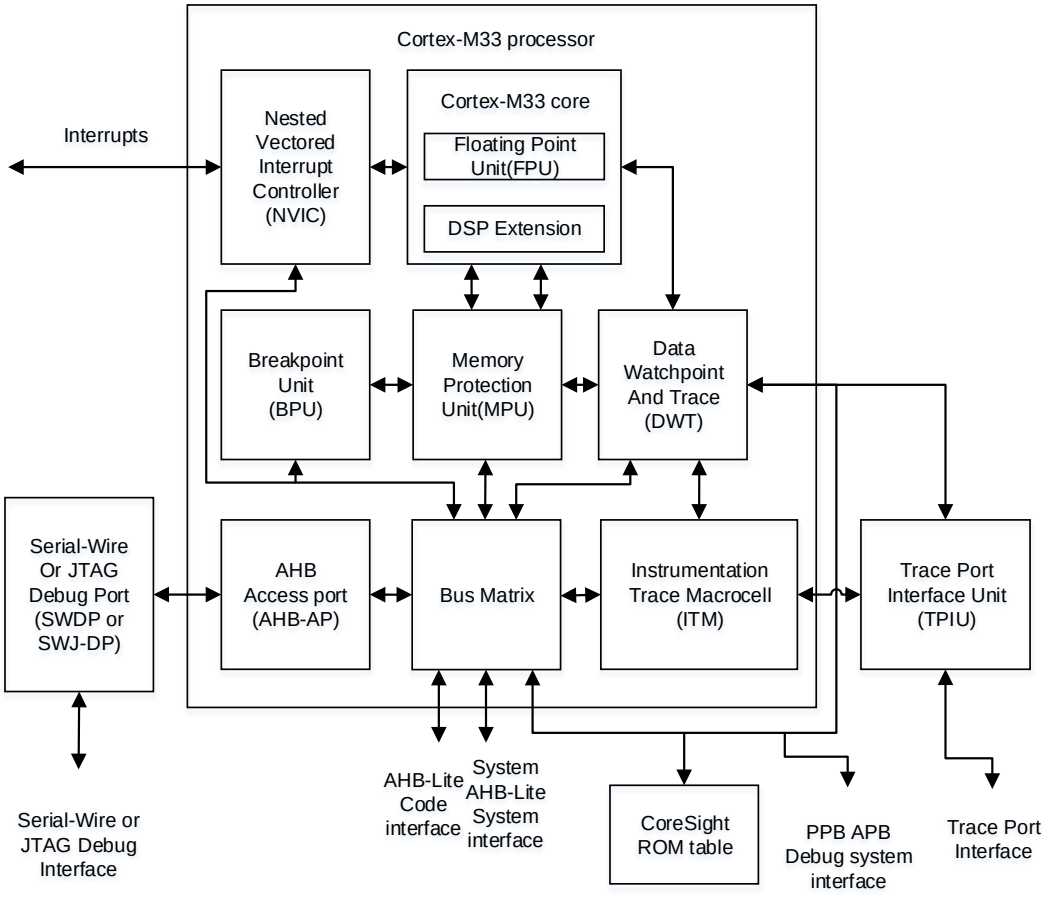
1.1. Arm® Cortex®-M33 处理器

Cortex®-M33 处理器是一个 32 位处理器，具有低中断延迟和低成本调试的特点。集成和先进的特性使 Cortex®-M33 处理器适合于需要高性能和低功耗微控制器的市场产品。Cortex®-M33 处理器基于 Armv8 架构，支持强大的可扩展指令集，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和 DSP。下面列出由 Cortex®-M33 提供的一些系统外围设：

- 与代码总线、系统总线和专用外围总线（PPB）相连的内部总线矩阵和调试访问；
- 嵌套矢量中断控制器（NVIC）；
- 断点单元（BPU）；
- 数据监视点和跟踪（DWT）；
- 测量跟踪宏单元（ITM）；
- 串行JTAG调试端口（SWJ-DP）；
- 跟踪端口接口单元（TPIU）；
- 存储器保护单元（MPU）；
- 浮点运算单元（FPU）；
- DSP拓展（DSP）。

*图1-1. Cortex®-M33处理器结构框图*显示了Cortex®-M33处理器结构框图。欲了解更多信息，请参阅Arm® Cortex®-M33技术参考手册。

图 1-1. Cortex®-M33 处理器结构框图



1.2. 系统架构

GD32A508xx 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

	CBUS	SBUS	DMA0	DMA1	ENET	OTGHS
FMC	1	1	1	1		1
SRAM	1	1	1	1	1	1
AHB1		1	1	1		
AHB2	1	1	1	1	1	1
APB1		1	1	1		
APB2		1	1	1		

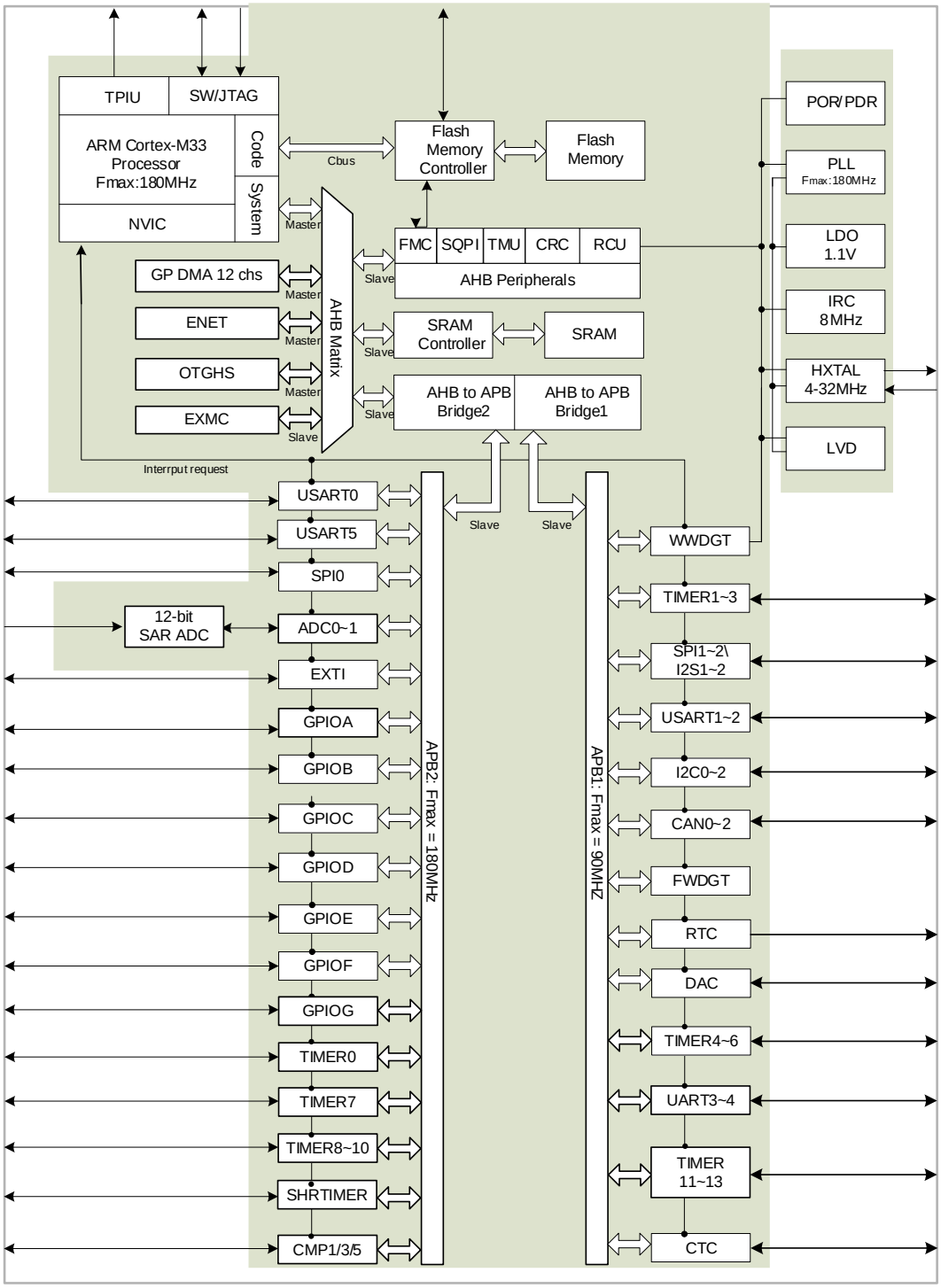
如上表所示，AHB 互联矩阵共连接一些主机，分别为：CBUS、SBUS、DMA0、DMA1、ENET

和 OTGHS。CBUS 是 Cortex®-M33 内核的指令总线，用于从代码区域中取指令和向量。SBUS 是 Cortex®-M33 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。ENET 是以太网。OT-GHS 是支持 OTG 协议的高速 USB 2.0 模块。

AHB 互联矩阵也连接了一些从机，分别为：FMC、SRAM、AHB1、AHB2、APB1 和 APB2。FMC 是闪存存储器控制器的总线。SRAM 是片上静态随机存取存储器。EXMC 是外部存储器控制器。AHB1 是连接除 EXMC 和 SQPI 之外所有 AHB 从机的 AHB 总线，AHB2 是连接 EXMC 和 SQPI 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 90MHz，APB2 可以全速运行（最高可到 180MHz）。

这些是使用多层 AHB 总线架构互连的，如下图所示。

图 1-2. GD32A508xx 系列器件的系统架构示意图



1.3. 存储器映射

Arm® Cortex®-M33 处理器采用哈佛架构，可以使用单独的总线来提取指令和加载/存储数据。程序存储器，数据存储器，寄存器和 I/O 端口组织在同一线性 4GB 地址空间内，这是 Cortex®-M33 的最大地址范围，因为总线地址宽度为 32 位。此外，Cortex®-M33 处理器提供了预定义的内存映射，以减少不同客户在相同应用时的软件复杂度。在内存映射表中，Arm® Cortex®-M33 系统外围设备使用的某些区域无法修改。但是，其他区域可供供应商使用。*表 1-2. GD32A508xx 系列器件的存储器映射表*显示了 GD32A508xx 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. GD32A508xx 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	0xC000 0000 - 0xDFFF FFFF	保留
		0xB000 0000 - 0xBFFF FFFF	SQPI_PSRAM(MEM)
		0xA000 1400 - 0xAFFF FFFF	保留
		0xA000 1000 - 0xA000 13FF	SQPI_PSRAM(REG)
		0xA000 0000 - 0xA000 0FFF	EXMC-SWREG
外部 RAM		0x9000 0000 - 0x9FFF FFFF	EXMC-PC CARD
		0x7000 0000 - 0x8FFF FFFF	EXMC-NAND
		0x6000 0000 - 0x6FFF FFFF	EXMC - NOR/PSRAM/SRAM
外设	AHB1	0x5000 0000 - 0x5003 FFFF	USBHS
		0x4008 0400 - 0x4FFF FFFF	保留
		0x4008 0000 - 0x4008 03FF	TMU
		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	ENET
		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

预先定义的地址空间	总线	地址范围	外设
		0x4002 0000 - 0x4002 03FF	DMA0
		0x4001 8400 - 0x4001 FFFF	保留
		0x4001 8000 - 0x4001 83FF	保留
	APB2	0x4001 7C00 - 0x4001 7FFF	CMP
		0x4001 7800 - 0x4001 7BFF	保留
		0x4001 7400 - 0x4001 77FF	SHRTIMER
		0x4001 7000 - 0x4001 73FF	USART5
		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	GPIOG
		0x4001 1C00 - 0x4001 1FFF	GPIOF
		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 CC00 - 0x4000 CFFF	CAN2
		0x4000 C800 - 0x4000 CBFF	CTC
		0x4000 C400 - 0x4000 C7FF	保留
		0x4000 C000 - 0x4000 C3FF	I2C2
		0x4000 8C00 - 0x4000 BFFF	保留
		0x4000 8800 - 0x4000 8BFF	CAN2SRAM
		0x4000 8400 - 0x4000 87FF	USBSRAM_B

预先定义的地址空间	总线	地址范围	外设
		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	CAN1
		0x4000 6400 - 0x4000 67FF	CAN0
		0x4000 6000 - 0x4000 63FF	Shared CAN SRAM 512 bytes
		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	I2S2_add
		0x4000 3C00 - 0x4000 3FFF	SPI2/I2S2
		0x4000 3800 - 0x4000 3BFF	SPI1/I2S1
		0x4000 3400 - 0x4000 37FF	I2S1_add
		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
		0x2000 0000 - 0x2001FFFF	SRAM
代码	AHB	0x1FFF F810 - 0x1FFF FFFF	保留
		0x1FFF F800 - 0x1FFF F80F	Option Bytes

预先定义的地址空间	总线	地址范围	外设
		0x1FFF F000 - 0x1FFF F7FF	Boot loader
		0x1FFF C010 - 0x1FFF EFFF	
		0x1FFF C000 - 0x1FFF C00F	
		0x1FFF B000 - 0x1FFF BFFF	
		0x1FFF 7800 - 0x1FFF AFFF	保留
		0x1FFF 7000 - 0x1FFF 77FF	OTP
		0x1FFF 0000 - 0x1FFF 6FFF	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
		0x0808 0000 - 0x082F FFFF	保留
		0x0800 0000 - 0x0807 FFFF	Main Flash
		0x0030 0000 - 0x07FF FFFF	保留
		0x0010 0000 - 0x002F FFFF	保留
		0x0008 0000 - 0x000F FFFF	保留
		0x0002 0000 - 0x0007 FFFF	Aliased to Main Flash or Boot loader
		0x0000 0000 - 0x0001 FFFF	

1.3.1. 片上 SRAM 存储器

GD32A508xx 系列微控制器含有高达 128KB 的片上 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。

1.3.2. 片上闪存

这些设备提供高密度的片上闪存，其结构如下:

- 最多512KB的主闪存；
- 最多18KB的信息块引导加载程序；
- 选项字节配置设备。

详情请参阅*闪存控制器（FMC）*一章。

1.4. 引导配置

GD32A508xx 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
系统存储器	0	1
片上SRAM	1	1

上电序列或系统复位后，Arm® Cortex®-M33 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 B000 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 对 FLASH 存储器进行重新编程。Bootloader 可以通过以下接口和外界交互：USART0 或 USART1 或 USB。

1.5. 设备电子签名

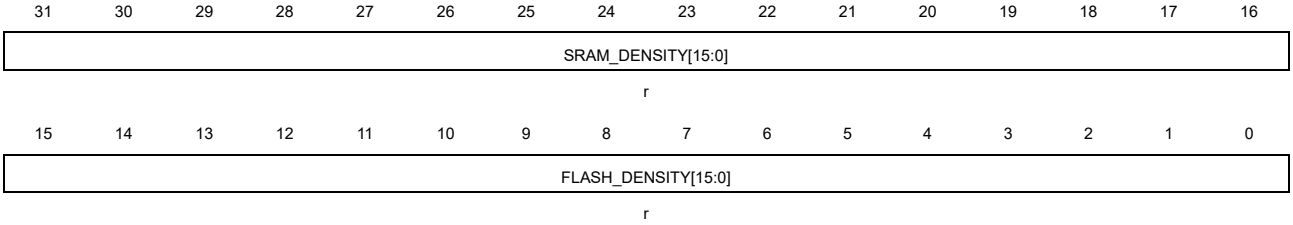
设备的电子签名中包含的存储容量信息和 96 位的唯一设备 ID。它被存储在片上闪存的信息模块中。96 位唯一设备 ID 对于每颗芯片而言都是唯一的。它可以用作序列号，或安全密钥的一部分，等等。

1.5.1. 存储容量信息

基地址：0x1FFF F7E0

该值是原厂设定的，不能由用户修改。

该寄存器只能按字（32 位）访问。



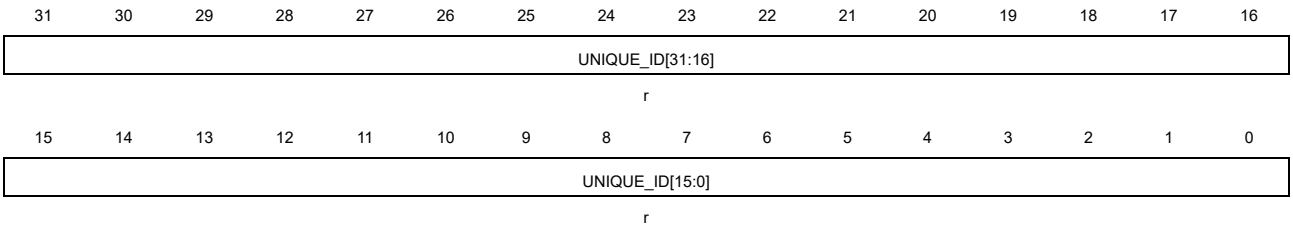
位/位域	名称	描述
31:16	SRAM_DENSITY [15:0]	SRAM存储器容量 该值表明芯片的片上SRAM存储器容量，以Kbytes为单位 例如：0x0080表示128Kbytes。
15:0	FLASH_DENSITY [15:0]	Flash存储器容量 该值表明芯片的片上Flash容量，以Kbytes为单位 例如：0x0200表示512Kbytes。

1.5.2. 设备唯一 ID(96 位/位域)

基地址：0x1FFF F7E8

该值是原厂设定的，不能由用户修改。

该寄存器只能按字（32 位）访问。



位/位域	名称	描述

31:0	UNIQUE_ID[31:0]	设备唯一ID

基地址：0x1FFF F7EC

该值是原厂设定的，不能由用户修改。

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
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

该值是原厂设定的，不能由用户修改。

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
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]	唯一设备 ID

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 512K 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512KB的片上闪存可用于存储指令或数据；
- 在闪存的前512K字节空间内，CPU执行指令需要0~4个等待时间；
- 闪存支持128位宽数据读取；
- 预取缓冲区以加速读操作；
- I-cache高速缓存区为2K字节，由64条缓存线组成，每条缓存线为2*128位；
- D-cache高速缓存区为256字节，由8条缓存线组成，每条缓存线为2*128位；
- 闪存页大小为8KB；
- 支持32位整字编程，页擦除和整片擦除操作；
- 2K字节OTP块（一次性编程），用于存储用户数据；
- 大小为16字节的选项字节可根据用户需求配置；
- 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 8KB。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. GD32A508xx 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 1FFF	8KB
	第1页	0x0800 2000 - 0x0800 3FFF	8KB
	第2页	0x0800 4000 - 0x0800 5FFF	8KB

闪存块	名称	地址范围	大小（字节）
	.	.	.
	第63页	0x0807 E000 - 0x0807 FFFF	8KB
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_77FF	2KB

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使用 CPU 的 AHB 总线。

增加等待状态：

根据 AHB 时钟频率，读闪存时需正确配置 FMC_WS 寄存器中的 WSCNT 位。WSCNT 位和 AHB 时钟频率的对应关系见*表 2-2. WSCNT 与 AHB 时钟频率对应关系*。

表 2-2. WSCNT 与 AHB 时钟频率对应关系

AHB时钟频率	WSCNT配置
<= 36MHz	0（0等待状态增加）
<= 72MHz	1（1等待状态增加）
<= 108MHz	2（2等待状态增加）
<= 144MHz	3（3等待状态增加）
<= 180MHz	4（4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 WSCNT 位和 AHB 时钟频率的对应关系表，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 WSCNT 位和 AHB 时钟频率的对应关系表，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。

由于添加了等待状态，读效率非常低（例如，180MHz 时需添加 4 个等待状态）。为了加速读

操作，需要用到一些功能。

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 128 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。预取缓存区仅在取指时执行。在顺序代码下，当 CPU 执行来自当前缓存区的数据时（128 位），按 32 位执行时需要至少 4 个时钟周期，按 16 位执行时需要至少 8 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。

I-cache 高速缓存区：

置位 FMC_WS 寄存器中 ICEN 位来使能 I-cache 高速缓存区。I-cache 高速缓存区仅在取指令时使用。I-cache 高速缓存区为 2K 字节，由 64 条缓存线组成，每条缓存线为 2*128 位。

当指令数据存在于 I-cache 高速缓存区时，CPU 从 I-cache 高速缓存区读取指令数据时无延迟。当指令数据不存在于 I-cache 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 I-cache 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。

D-cache 高速缓存区：

置位 FMC_WS 寄存器中 DCEN 位来使能 D-cache 高速缓存区。D-cache 高速缓存区仅在 CPU 取数据（不是通过 DMA）时使用，此时选项字节不可缓存。D-cache 高速缓存区为 256 字节，由 8 条缓存线组成，每条缓存线为 2*128 位。

当数据存在于 D-cache 高速缓存区时，CPU 从 D-cache 高速缓存区读取数据时无延迟。当数据不存在于 D-cache 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 D-cache 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。解锁序列是向 FMC_OBKEY 寄存器先后写入 0x45670123 和 0xCDEF89AB，然后硬件会将 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。

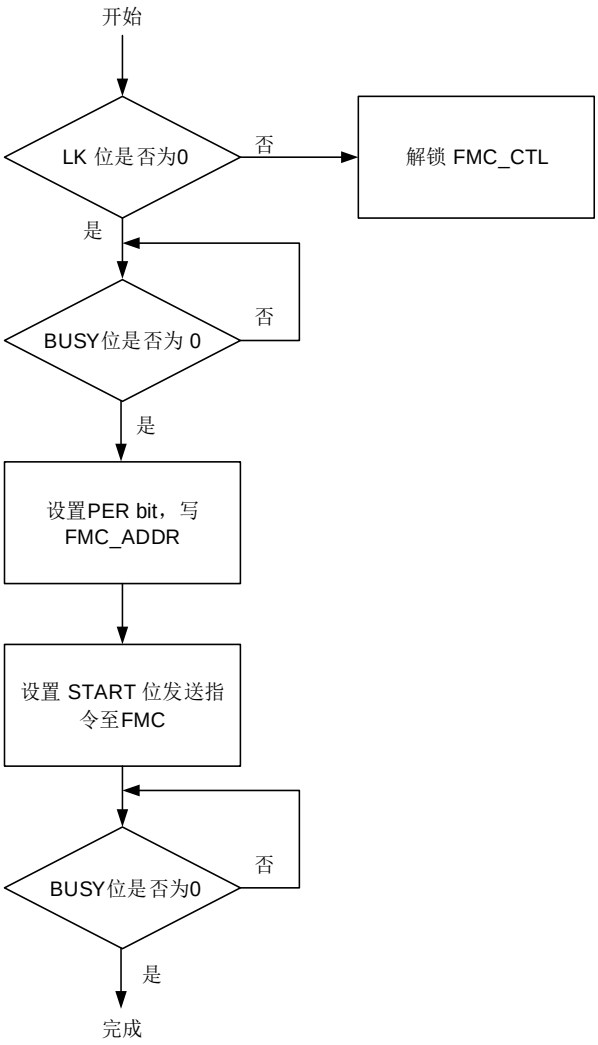
2.3.4. 页擦除

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，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：

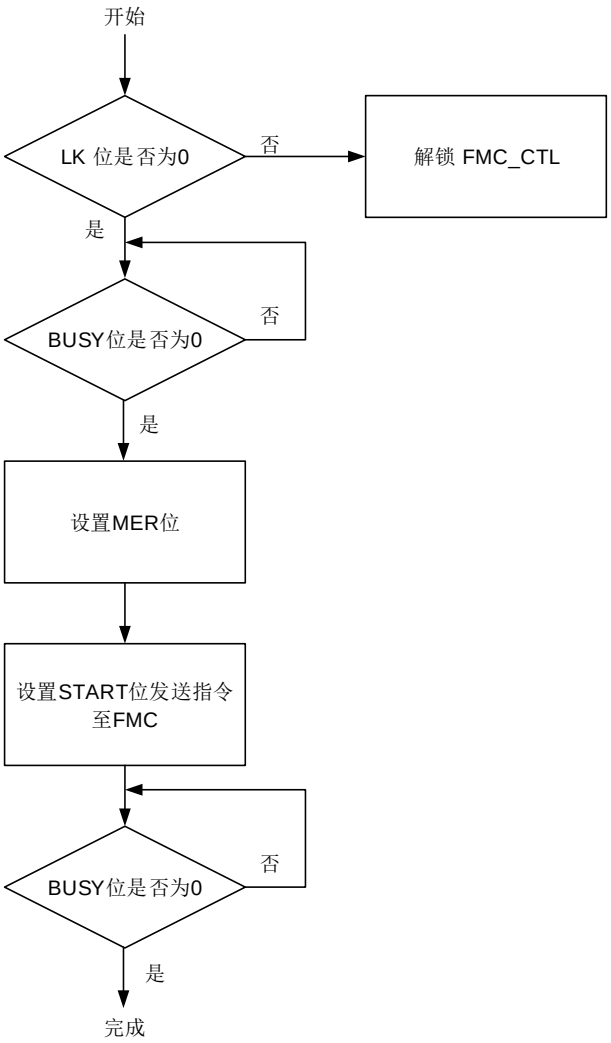
- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；
- 如果整片擦除闪存，置位FMC_CTL寄存器的MER位；
- 通过将FMC_CTL寄存器的START位置1来发送整片擦除命令到FMC；
- 等待擦除指令执行完毕，FMC_STAT寄存器的BUSY位清0；
- 如果需要，读并验证是否擦除成功。

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。

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；
- 置位FMC_CTL寄存器的PG位；
- 写一个32位整字/16位半字到目的绝对地址（0x08XX XXXX）；

按 32 位字编程，写 1 次即可将数据编程入闪存。待编程数据必须字对齐。

按 16 位字编程，写 2 次即可将数据编程入闪存。待编程数据必须字对齐。

为了缩短编程时间，建议采用按 32 位编程。

- 等待编程指令执行完毕，FMC_STAT寄存器的BUSY位清0；
- 如果需要，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，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。

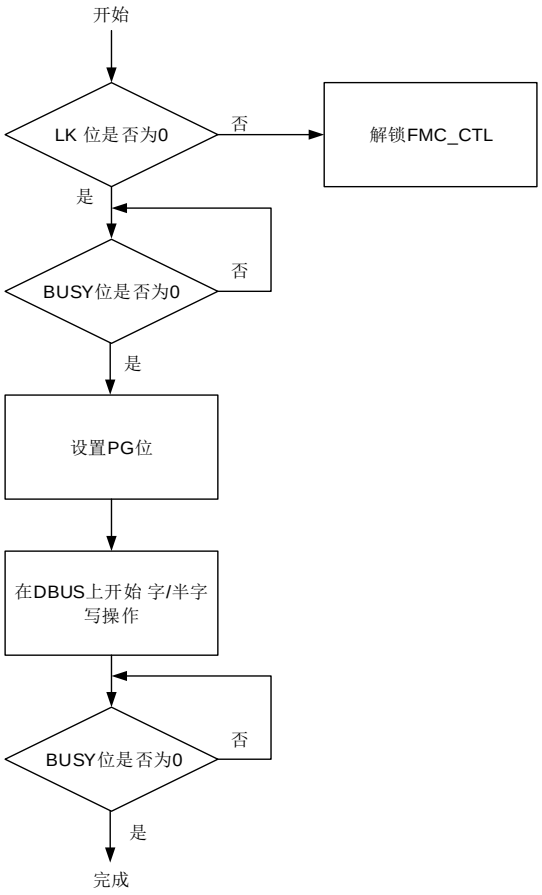
在下列情况下，FMC_STAT 寄存器中的 PGAERR 位会被置位。

- 按字节编程（非32位或16位）
- 混合编程。不允许混合32位和16位编程。
- 编程未对齐。

注意：如果编程数据未能写满 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，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；
- 写一个32位整字/16位半字到目的地址。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，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 0xA5：未保护状态 除0xA5和0xCC外的任何值：保护级别低 0xCC：保护级别高
0x1fff f801	SPC_N	SPC补字节
0x1fff f802	USER	[7:3]：保留 [2]：nRST_STDBY 0：设置待机模式时产生复位而不是进入待机模式 1：设置待机模式时进入待机模式而不产生复位 [1]：nRST_DPSLP 0：设置深度睡眠模式时产生复位而不进入深度睡眠模式

地址	名称	说明
		1：设置深度睡眠模式时进入深度睡眠模式而不产生复位 [0]：nWDG_HW 0：硬件使能独立看门狗功能 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 0：保护生效 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 WP[30:24]：每个bit可设置8KB闪存的保护状态。第0位设置前8KB闪存的保护状态，以此类推。这31位总计可设置前248KB的闪存保护状态。 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。

表 2-4. OB_WP 位对应页保护

OB_WP位	页保护
OB_WP[0]	页0
OB_WP[1]	页1
OB_WP[2]	页2
. . .	. . .
OB_WP[30]	页30
OB_WP[31]	页31 ~ 页63

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。

保护等级低：当设置 SPC 字节值为任何除 0xA5 或 0xCC 外的值，系统复位以后，低安全保护状态生效。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。在低安全保护状态下，主存储闪存块仅能被用户代码访问且前 8KB 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。

保护等级高：当设置 SPC 字节为 0xCC，激活高安全保护等级。当编程选择该保护等级时，调试模式，从 SRAM 中启动，或者从 boot loader 启动都被禁止。主存储闪存块可由用户代码的所有操作进行访问。SPC 字节禁止再次编程。所以，如果高保护等级被激活，将不能再降回到低保护等级或无保护等级。

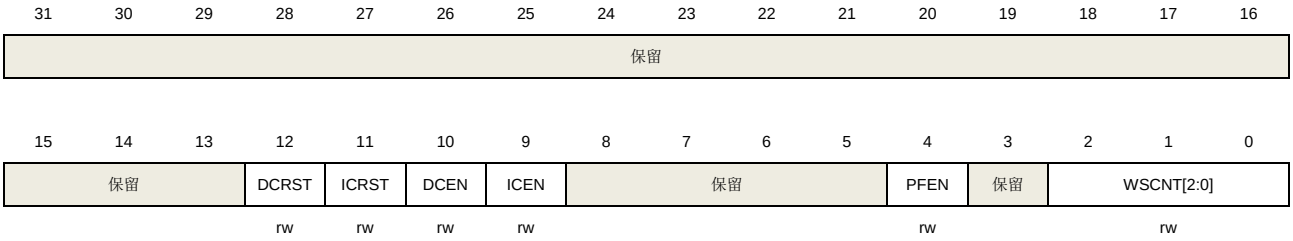
2.4. FMC 寄存器

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
31:13	保留	必须保持复位值
12	DCRST	复位 D-cache 高速缓存区。该位仅可在 DCEN 位置 0 时可写。 0：无效果 1：复位 D-cache 高速缓存区
11	ICRST	复位 I-cache 高速缓存区。该位仅可在 ICEN 位置 0 时可写。 0：无效果 1：复位 I-cache 高速缓存区
10	DCEN	D-cache 高速缓存区使能位 0：失能 D-cache 高速缓存区 1：使能 D-cache 高速缓存区
9	ICEN	I-cache 高速缓存区使能位 0：失能 I-cache 高速缓存区 1：使能 I-cache 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 0：失能预取功能 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 软件置 1 和清 0。 000：不增加等待状态

001：增加 1 个等待状态
010：增加 2 个等待状态
011：增加 3 个等待状态
100：增加 4 个等待状态
101 ~ 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 这些位仅能被软件写。 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 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 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 当写数据不对齐时，此位被硬件置 1。软件写 1 清 0。
2	PGERR	编程错误标志位 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 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 软件置 1 和清 0 0：无硬件中断产生 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 软件置 1 和清 0 0：无硬件中断产生. 1：使能出错中断
9	OBWEN	选项字节擦除/编程使能位 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 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 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 软件置 1 和清 0 0：无作用 1：选项字节擦除命令
4	OBPG	选项字节编程命令位 软件置 1 和清 0 0：无作用 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 软件置 1 和清 0 0：无作用 1：主存储块整片擦除命令
1	PER	主存储块页擦除命令位 软件置 1 和清 0 0：无作用 1：主存储块页擦除命令
0	PG	主存储块编程命令位 软件置 1 和清 0 0：无作用

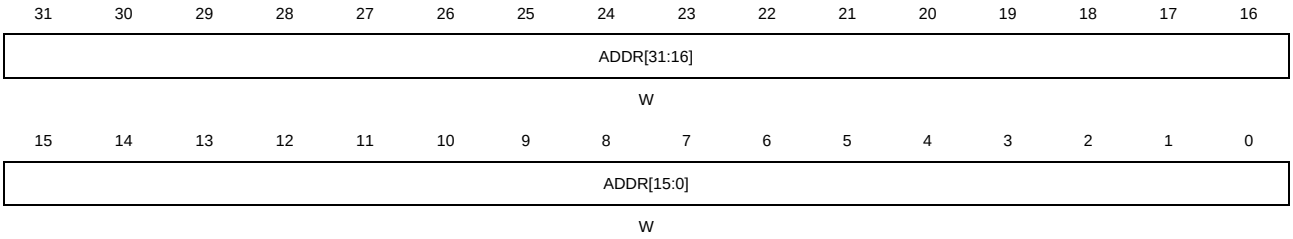
1：主存储块编程命令

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 该位通过软件设置。 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 0：未保护 1：已保护

0	OBERR	选项字节读错误位 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 该寄存器为只读 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. 备份寄存器（BKP）

3.1. 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。

3.2. 主要特点

- 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）可实现侵入检测中断或事件的控制

3.3. 功能描述

3.3.1. 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。通过置位 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。

3.3.2. 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。

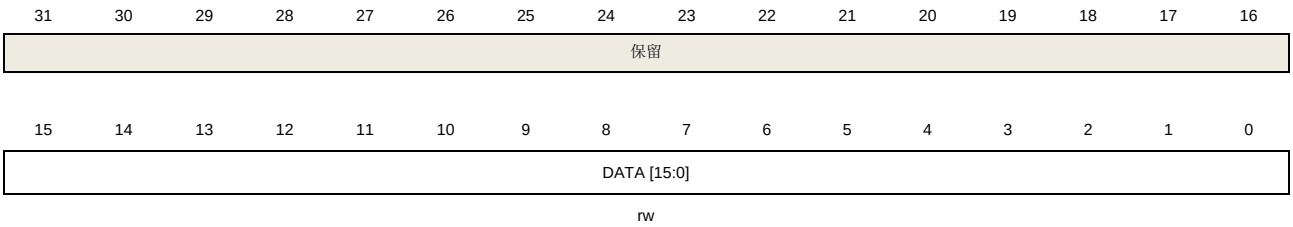
3.4. BKP 寄存器

BKP 基地址：0x4000 6C00

3.4.1. 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

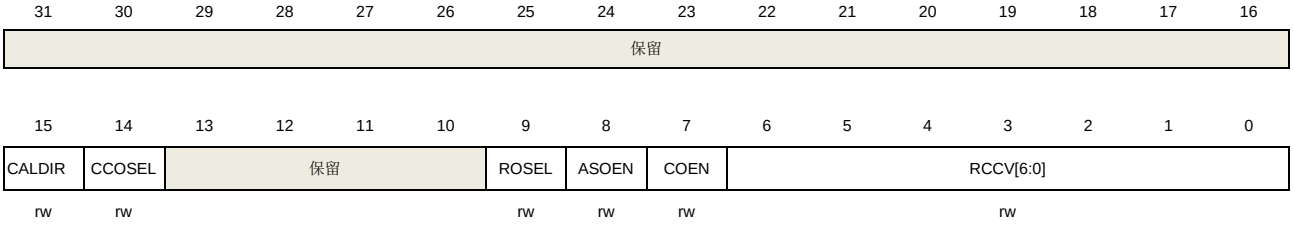


位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	DATA[15:0]	备份数据 这些位用来存储一般用户数据。即使从待机模式唤醒或系统复位或电源复位后，BKP_DATAx 寄存器的内容仍旧不会丢失。

3.4.2. 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
31:16	保留	必须保持复位值。
15	CALDIR	RTC 时钟校准方向 0：变慢 1：变快 该位只能被备份域复位清除。
14	CCOSEL	RTC 时钟输出选择

		0：RTC 时钟 64 分频 1：RTC 时钟 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 0：RTC 输出为闹钟脉冲 1：RTC 输出为秒脉冲 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 0：RTC 闹钟或秒信号输出禁止 1：RTC 闹钟或秒信号输出使能 使能后，TAMPER 引脚可作为 RTC 输出。 该位只能被备份域复位清除。
7	COEN	RTC 时钟校准输出使能 0：RTC 时钟校准输出禁止 1：RTC 时钟校准输出使能 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。 该位只能被上电复位（POR）清除。
6:0	RCCV[6:0]	RTC 时钟校准值 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或被加入。 该位只能被备份域复位清除。

3.4.3. 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

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														TPAL	TPEN
														rw	rw

位/位域	名称	描述
31:2	保留	必须保持复位值
1	TPAL	TAMPER 引脚有效电平 0：TAMPER 引脚高电平有效

1：TAMPER 引脚低电平有效

0	TPEN	TAMPER 引脚使能 0：TAMPER 引脚作为 GPIO 口使用 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。

3.4.4. 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

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						TIF	TEF	保留					TPIE	TIR	TER
						r	r						rw	w	w

位/位域	名称	描述
31:10	保留	必须保持复位值
9	TIF	侵入中断标志 0：没有侵入中断发生 1：有侵入中断发生 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 0：没有侵入事件发生 1：有侵入事件发生 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 0：侵入中断禁用 1：侵入中断使能 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 0：不影响 1：复位 TIF 位 该位一直读为 0。
0	TER	侵入事件复位 0：不影响 1：复位 TEF 位

该位一直读为 0。

4. 电源管理单元（PMU）

4.1. 简介

功耗设计是GD32A508xx系列产品比较注重的问题之一。电源管理单元提供了五种省电模式，包括睡眠模式，深度睡眠模式，深度睡眠模式 1，深度睡眠模式 2 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 4-1. 电源域概览*所示，GD32A508xx 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。

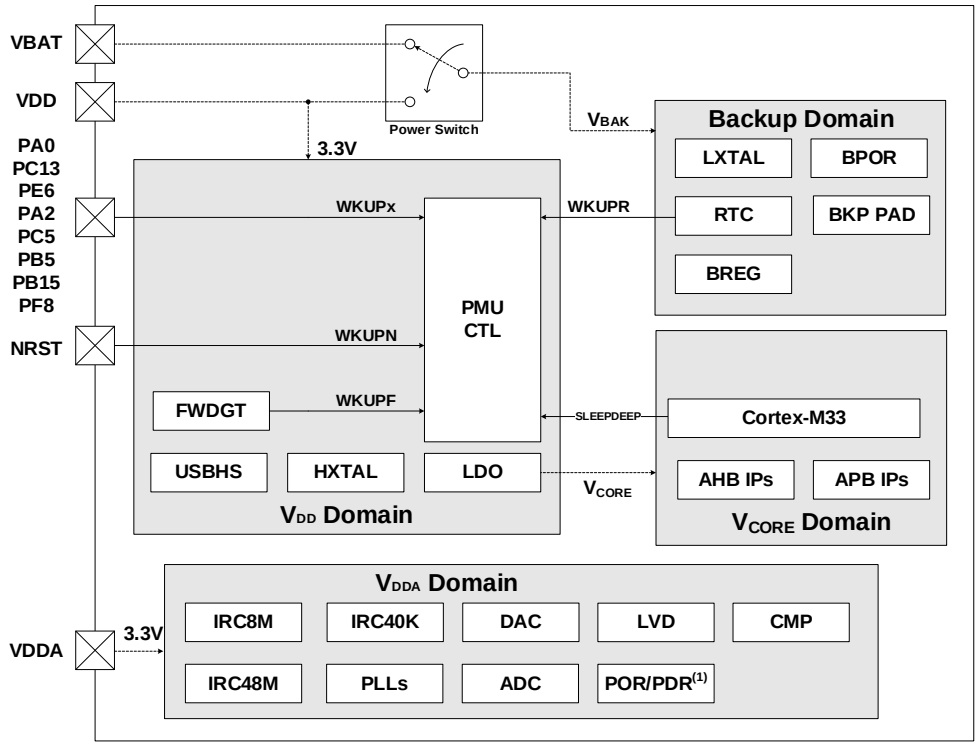
4.2. 主要特性

- 三个电源域：备份域、V_{DD} / V_{DDA}域和V_{CORE}电源域；
- 五种省电模式：睡眠模式、深度睡眠模式，深度睡眠模式1，深度睡眠模式2和待机模式；
- 内部电压调节器（LDO）为V_{CORE}电源域提供V_{CORE}电源；
- 提供低电压检测器，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；
- 低驱动模式用于在深入睡眠模式 / 深度睡眠模式1 / 深度睡眠模式2下超低功耗。高驱动模式用在高频模式中。

4.3. 功能描述

*图 4-1. 电源域概览*提供了 PMU 及相关电源域的内部结构框图。

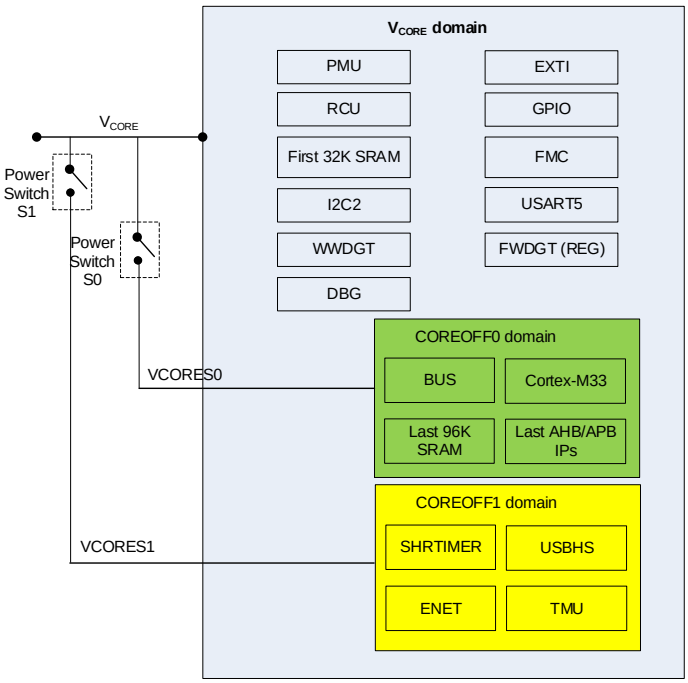
图 4-1. 电源域概览



注意：

（1）对于产品版本 C，POR / PDR 电路位于 V_{DD} 域，检测 VDD 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 4-2. V_{CORE} 电源域概览



如图*图 4-2. VCORE 电源域概览*所示，V_{CORE} 域包含三个部分：

1. COREOFF1 域包含：ENET，SHRTIMER，USBHS（USBHS 模块部分由 V_{DD} 域供电，部分由 COREOFF1 域供电），TMU 外设。
2. COREOFF0 域包含：
   – 剩余AHB/APB IPs：除了FMC，PMU，RCU，EXTI，GPIO，DBG，FWDGT（仅寄存器），WWDGT，USART5，I2C2，以及COREOFF1域之外的所有V_{CORE}域外设；
   – 剩余96K SRAM：SRAM（除前32K）；
   – Cortex®-M33 CPU；
   – 总线：包括一个AHB互联矩阵、两个AHB总线和两个APB总线。
3. 深度睡眠保持供电模块：FMC，PMU，RCU，EXTI，GPIO，DBG，FWDGT（仅寄存器），WWDGT，USART5，I2C2，以及前 32K SRAM。

电源切换器 S0 和 S1 用于省电模式的供电控制。

4.3.1. 备份域

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 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 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®-M33 需要通过 RTC 寄存器设置预期的闹钟时间并启用闹钟功能，通过 EXTI 线获取 RTC 闹钟事件。进入省电模式一定时间之后，当经过的时间与预设的闹钟时间匹配时，RTC 将唤醒设备。RTC 的配置和操作的细节将在*实时时钟（RTC）*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)。

4.3.2. 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）、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 POR / PDR（上电 / 掉电复位）、ADC / DAC（AD / DA 转换器）、CMP（比较器）、IRC8M（内部 8M RC 振荡器）、IRC48M（内部 48M RC 振荡器）、IRC40K（内部 40KHz RC 振荡器）、PLLs（锁相环）和 LVD（低电压检测器）等等。

V_{DD} 域

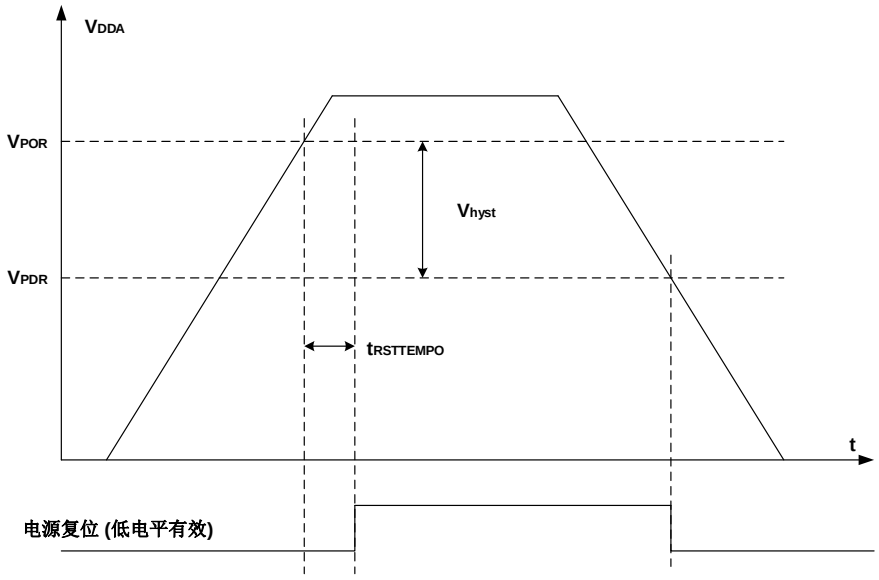
为 V_{CORE} 域供电的 LDO（电压调节器），其复位后保持使能。可以被配置为三种不同的工作状态：包括睡眠模式（全供电状态）、深度睡眠模式 / 深度睡眠模式 1 / 深度睡眠模式 2（全供电或低功耗状态）和待机模式（关闭状态）。

V_{DDA} 域

POR / PDR（上电 / 掉电复位）电路检测 V_{DDA} 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 4-3. 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。迟滞电压 V_{hyst} 值参考数据手册。

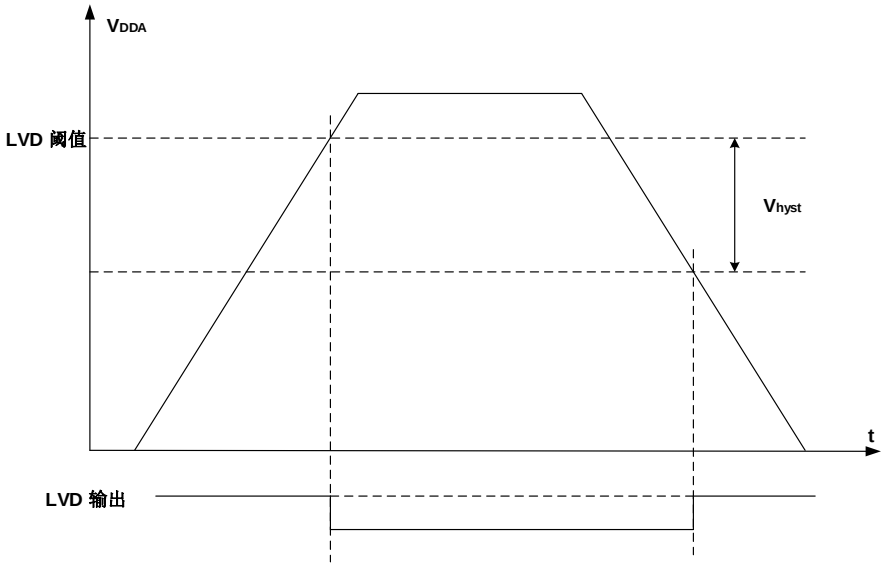
注意：对于产品版本 C，POR / PDR 电路位于 V_{DD} 域，检测 V_{DD} 并在电压低于特定阈值时产生电源复位信号复位除备份域之外的整个芯片。

图 4-3. 上电/掉电复位波形图



LVD 的功能是检测 V_{DDA} 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_CTL0)中的 LVDT[2:0]位进行配置。LVD 通过 LVDEN 置位使能，位于电源状态寄存器(PMU _CS0)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 4-4. 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_{hyst} 值参考数据手册。

图 4-4. 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 V_{DD} 与 V_{DDA} 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 13-2. ADC 输入引脚定义*和*表 14-1. DAC 引脚*，VREFN 须被连接至 VSSA 引脚，VREFP 引脚仅存在于不小于 100-pin 封装上，而在 64-pin 或更少引脚封装不存在，因其内部已经连接至 VDDA 和 VSSA。VREFN 仅存在于不小于 100-pin 封装上，在其他封装里，其内部连接至 VSSA。

4.3.3. V_{CORE} 电源域

主要功能包括 Cortex®-M33 内核逻辑、AHB / APB 外设、备份域和 V_{DD} / V_{DDA} 域的 APB 接口等。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。

高驱动模式

如果 V_{CORE} 电源域工作在高频状态下，且打开了多种功能，建议进入高驱动模式。使用高驱动模式有以下步骤：

- 选择系统时钟为IRC8M或HXTAL；
- 将PMU_CTL0寄存器的HDEN置1，使能高驱动模式；
- 等待PMU_CS0寄存器的HDRF被置位；
- 将PMU_CTL0寄存器的HDS置1，将LDO切换到高驱动模式；
- 等待PMU_CS0寄存器的HDSRF被置位。进入高驱动模式；
- 工作在高频状态。

在选择 IRC8M 或 HXTAL 作为系统时钟后，可以通过将 PMU_CTL0 寄存器的 HDEN 和 HDS 清 0 退出高驱动模式。当系统退出深度睡眠模式 / 深度睡眠模式 1 / 深度睡眠模式 2 时，将会自动退出高驱动模式。

4.3.4. 省电模式

系统复位或电源复位后，GD32A508xx 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。此外，五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，深度睡眠模式 1，深度睡眠模式 2 和待机模式。

睡眠模式

睡眠模式与 Cortex®-M33 的 SLEEPING 模式相对应。在睡眠模式下，仅关闭 Cortex®-M33 的时钟。如需进入睡眠模式，只要清除 Cortex®-M33 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®-M33 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®-M33 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®-M33 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_CTL0 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®-M33 系统控制寄存器的 SLEEPDEEP 位置 1，再清除 PMU_CTL0 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®-M33 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。

在深度睡眠模式下，通过配置 PMU_CTL0 寄存器的 LDEN，LDNP，LDLP，LDOLP 位可以进入低驱动模式。低驱动模式具有低驱动能力，低功耗模式工作能耗很低。

正常驱动/正常功耗：将 PMU_CTL0 寄存器的 LDEN 位配置为 00，深度睡眠模式就工作在正常驱动模式下。将 PMU_CTL0 寄存器的 LDOLP 清 0 可以退出低功耗模式。

正常驱动/低功耗：将 PMU_CTL0 寄存器的 LDEN 位配置为 00，深度睡眠模式就工作在正常驱动模式下。将 PMU_CTL0 寄存器的 LDOLP 置 1 可以进入低功耗模式。

低驱动/正常功耗：将 PMU_CTL0 寄存器的 LDEN 设置为 0b11，LDNP 置 1 可以进入深度睡眠模式的低驱动模式。将 PMU_CTL0 寄存器的 LDOLP 清 0 可以使 LDO 处于正常功耗模式。

低驱动/低功耗：将 PMU_CTL0 寄存器的 LDEN 设置为 0b11，LDLP 置 1 可以进入深度睡眠模式的低驱动模式。将 PMU_CTL0 寄存器的 LDOLP 置 1 可以使 LDO 处于低功耗模式。

非低驱动：将 PMU_CTL0 寄存器的 LDEN 设置为 00，深度睡眠模式将不会处在低驱动模式。

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）和和相关外设标志位必须被复位，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。

深度睡眠模式 1

深度睡眠模式 1 与 Cortex®-M33 的 SLEEPDEEP 模式相对应。在深度睡眠模式 1 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。COREOFF1 域停止供电。COREOFF1 域寄存器中的内容全部丢失。根据 PMU_CTL0 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 1 之前，先将 Cortex®-M33 系统控制寄存器的 SLEEPDEEP 位置 1，再清除 PMU_CTL0 寄存器的 STBMOD 位，再置位 PMU_CTL1 寄存器的 DPMOD1 位，然后执行 WFI 或 WFE 指令即可进入深度睡眠模式 1。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 1 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 1 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®-M33 技术手册）。刚退出深度睡眠模式 1 时，IRC8M 被选中作为系统时钟。从深度睡眠模式 1 唤醒需要额外的延时时间用于给 COREOFF1 域上电。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。

在深度睡眠模式 1 下，通过配置 PMU_CTL0 寄存器的 LDEN，LDNP，LDLP，LDOLP 位可以进入低驱动模式。低驱动模式具有低驱动能力，低功耗模式工作能耗很低。

深度睡眠模式 2

深度睡眠模式 2 与 Cortex®-M33 的 SLEEPDEEP 模式相对应。在深度睡眠模式 2 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。COREOFF0 / COREOFF1 域停止供电。SRAM（除前 32K）和 COREOFF0/COREOFF1 域寄存器中的内容全部丢失。根据 PMU_CTL0 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 2 之前，先将 Cortex®-M33 系统控制寄存器的 SLEEPDEEP 位置 1，再清除 PMU_CTL0 寄存器的 STBMOD 位，再置位 PMU_CTL1 寄存器的 DPMOD2 位，然后执行 WFI 或 WFE 指令即可进入深度睡眠模式 2。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 2 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 2 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®-M33 技术手册）。刚退出深度睡眠模式 2 时，IRC8M 被选中作为系统时钟。从深度睡眠模式 2 唤醒需要额外的延时时间用于给 COREOFF0 / COREOFF1 域上电。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。

在深度睡眠模式 2 下，通过配置 PMU_CTL0 寄存器的 LDEN，LDNP，LDLP，LDOLP 位可以进入低驱动模式。低驱动模式具有低驱动能力，低功耗模式工作能耗很低。

待机模式

待机模式是基于 Cortex®-M33 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®-M33 系统控制寄存器的 SLEEPDEEP 位置 1，再将 PMU_CTL0 寄存器的 STBMOD 位置 1，再清除 PMU_CS0 寄存器的 WUF 位，然后执行 WFI 或 WFE 指令，系统进入待机模式，PMU_CS0 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 报警，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®-M33 将从 0x00000000 地址开始执行指令代码。

表 4-1. 节电模式总结

模式	睡眠	深度睡眠	深度睡眠 1	深度睡眠 2	待机
描述	仅关闭 CPU 时钟	1、关闭 V_{CORE} 电源域的所有时钟 2、关闭 IRC8M、IRC48M、HXTAL 和 PLL	1、关闭 V_{CORE} 电源域的所有时钟 2、关闭 IRC8M、IRC48M、HXTAL 和 PLL	1、关闭 V_{CORE} 电源域的所有时钟 2、关闭 IRC8M、IRC48M、HXTAL 和 PLL	1、关闭 V_{CORE} 电源域的供电 2、关闭 IRC8M、IRC48M、HXTAL 和 PLL

模式	睡眠	深度睡眠	深度睡眠 1	深度睡眠 2	待机
			3、关闭 SHRTIMER、USBHS、TMU、ENET 的供电	3、关闭 CPU、SRAM（除前 32K）、COREOFF0 / COREOFF1 域外设的供电	
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，正常驱动或者低驱动模式）	开启（正常功耗模式或者低功耗模式，正常驱动或者低驱动模式）	开启（正常功耗模式或者低功耗模式，正常驱动或者低驱动模式）	关闭
配置	SLEEPDEEP = 0	SLEEPDEEP = 1 STBMOD = 0	SLEEPDEEP = 1 STBMOD = 0 DPMOD1 = 1	SLEEPDEEP = 1 STBMOD = 0 DPMOD2 = 1	SLEEPDEEP = 1 STBMOD = 1, WURST = 1
进入指令	WFI 或 WFE	WFI 或 WFE	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； 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；若通过 WFE 进入，来自 EXTI 的任何事件（或 SEVONPEND=1 时的中断）可唤醒	若通过 WFI 进入，来自 EXTI 的任何中断可唤醒；若通过 WFE 进入，来自 EXTI 的任何事件（或 SEVONPEND=1 时的中断）可唤醒	若通过 WFI 进入，来自 EXTI 的任何中断可唤醒；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 2、WKUP 引脚 3、FWDGT 复位 4、RTC
唤醒延迟	无	IRC8M 唤醒时间 如果 LDO 处于低功耗模式，需增加 LDO 唤醒时间	IRC8M 唤醒时间 COREOFF1 域上电时间 如果 LDO 处于低功耗模式，需增加 LDO 唤醒时间	IRC8M 唤醒时间 COREOFF0/COREOFF1 域上电时间 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 RESET 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。

4.4. PMU 寄存器

PMU 基地址：0x4000 7000

4.4.1. 控制寄存器 0（PMU_CTL0）

地址偏移：0x00
复位值：0x0000 C000（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												LDEN[1:0]		HDS	HDEN
												rw		rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				LDNP	LDLP	保留	BKPWEN	LVDT[2:0]			LVDEN	STBRST	WURST	STBMOD	LDOLP
				rw	rw		rw	rw			rw	rc_w1	rc_w1	rw	rw

位/位域	名称	描述
31:20	保留	必须保持复位值
19:18	LDEN[1:0]	深度睡眠模式 / 深度睡眠模式 1 / 深度睡眠模式 2 下，低驱动模式使能 00：在深度睡眠模式 / 深度睡眠模式 1 / 深度睡眠模式 2 下，禁用低驱动模式 01：保留 10：保留 11：在深度睡眠模式 / 深度睡眠模式 1 / 深度睡眠模式 2 下，使能低驱动模式
17	HDS	高驱动模式切换器 选择 IRC8M 或 HXTAL 作为系统时钟，当 HDRF 被置位时，由软件将该位置 1。该位被置位后，系统进入高驱动模式。可由软件清 0，也可在退出深度睡眠模式 / 深度睡眠模式 1 / 深度睡眠模式 2 或 HDEN 被清 0 时由硬件清 0。 0：没有高驱动模式切换器 1：有高驱动模式切换器
16	HDEN	高驱动模式使能 当系统时钟为 IRC8M 或 HXTAL 时，该位由软件置位。该位可由软件清零或当系统退出深度睡眠模式 / 深度睡眠模式 1 / 深度睡眠模式 2 由硬件清零。 0：禁用高驱动模式 1：使能高驱动模式
15:12	保留	必须保持复位值
11	LDNP	使用正常功耗 LDO 时，工作在低驱动模式 0：使用正常功耗 LDO 时，工作在正常驱动模式 1：使用正常功耗 LDO 且 LDEN 为 11 时，低驱动模式被使能
10	LDLP	使用低功耗 LDO 时，工作在低驱动模式

		0：使用低功耗 LDO 时，工作在正常驱动模式 1：使用低功耗 LDO 且 LDEN 为 11 时，低驱动模式被使能
9	保留	必须保持复位值
8	BKPWEN	备份域写使能 0：禁止对备份域寄存器的写访问 1：允许对备份域寄存器的写访问 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 000：2.1V 001：2.3V 010：2.4V 011：2.6V 100：2.7V 101：2.9V 110：3.0V 111：3.1V
4	LVDEN	低电压检测器使能 0：关闭低电压检测器 1：开启低电压检测器
3	STBRST	待机标志复位 0：无影响 1：复位待机标志 读该位，始终返回 0
2	WURST	唤醒标志复位 0：无影响 1：复位唤醒标志 读该位，始终返回 0
1	STBMOD	待机模式 0：当 Cortex®-M33 进入 SLEEPDEEP 模式时，系统进入深度睡眠模式 / 深度睡眠模式 1 / 深度睡眠模式 2 1：当 Cortex®-M33 进入 SLEEPDEEP 模式时，系统进入待机模式
0	LDOLP	LDO 低功耗模式 0：当系统进入深度睡眠模式 / 深度睡眠模式 1 / 深度睡眠模式 2 时，LDO 仍正常工作 1：当系统进入深度睡眠模式 / 深度睡眠模式 1 / 深度睡眠模式 2 时，LDO 进入低功耗模式 注意：在深度睡眠模式下，个别外设可能会开启 IRC8M 时钟来做一些工作。在这种情况下，如果 LDO 正处于低功耗模式，LDO 会自动从低功耗模式切换到正常工作模式，并保持正常工作模式，直到外设工作完毕。

4.4.2. 电源控制和状态寄存器 0（PMU_CS0）

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。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留												LDRF[1:0]		HDSRF	HDRF
												rc_w1		r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WUPEN7	保留	WUPEN5	WUPEN4	WUPEN3	WUPEN2	WUPEN1	WUPEN0	WUPEN6	保留				LVDF	STBF	WUF
rw		rw	rw	rw	rw	rw	rw	rw					r	r	r

位/位域	名称	描述
31:20	保留	必须保持复位值
19:18	LDRF[1:0]	低驱动模式就绪标志 在深度睡眠模式 / 深度睡眠模式 1 / 深度睡眠模式 2 下，且 LDO 处于低驱动模式，这些位由硬件设置。软件对这些位写 11 可以清 0。 00：深度睡眠模式 / 深度睡眠模式 1 / 深度睡眠模式 2 下，普通驱动模式 01：保留 10：保留 11：深度睡眠模式 / 深度睡眠模式 1 / 深度睡眠模式 2 下，低驱动模式
17	HDSRF	高驱动切换器就绪标志 0：高驱动切换器未就绪 1：高驱动切换器就绪
16	HDRF	高驱动准备就绪标志 0：高驱动未就绪 1：高驱动就绪
8	WUPEN7	WKUP 引脚 7（PF8）唤醒使能 0：关闭 WKUP 引脚 7 唤醒功能 1：开启 WKUP 引脚 7 唤醒功能 如果 WUPEN7 在进入省电模式之前置 1，WKUP 引脚 7 的上升沿会将系统从省电模式唤醒。由于 WKUP 引脚 7 为高电平有效，WKUP 引脚 7 内部被配置为输入下拉模式。当在输入已经为高的时候置位该控制位，将会触发一个唤醒事件。
14	保留	必须保持复位值
13	WUPEN5	WKUP 引脚 5（PB5）唤醒使能 0：关闭 WKUP 引脚 5 唤醒功能 1：开启 WKUP 引脚 5 唤醒功能 如果 WUPEN5 在进入省电模式之前置 1，WKUP 引脚 5 的上升沿会将系统从省电模式唤醒。由于 WKUP 引脚 5 为高电平有效，WKUP 引脚 5 内部被配置为输入下拉模式。当在输入已经为高的时候置位该控制位，将会触发一个唤醒事件。

12	WUPEN4	WKUP 引脚 4（PC5）唤醒使能 0：关闭 WKUP 引脚 4 唤醒功能 1：开启 WKUP 引脚 4 唤醒功能 如果 WUPEN4 在进入省电模式之前置 1，WKUP 引脚 4 的上升沿会将系统从省电模式唤醒。由于 WKUP 引脚 4 为高电平有效，WKUP 引脚 4 内部被配置为输入下拉模式。当在输入已经为高的时候置位该控制位，将会触发一个唤醒事件。
11	WUPEN3	WKUP 引脚 3（PA2）唤醒使能 0：关闭 WKUP 引脚 3 唤醒功能 1：开启 WKUP 引脚 3 唤醒功能 如果 WUPEN3 在进入省电模式之前置 1，WKUP 引脚 3 的上升沿会将系统从省电模式唤醒。由于 WKUP 引脚 3 为高电平有效，WKUP 引脚 3 内部被配置为输入下拉模式。当在输入已经为高的时候置位该控制位，将会触发一个唤醒事件。
10	WUPEN2	WKUP 引脚 2（PE6）唤醒使能 0：关闭 WKUP 引脚 2 唤醒功能 1：开启 WKUP 引脚 2 唤醒功能 如果 WUPEN2 在进入省电模式之前置 1，WKUP 引脚 2 的上升沿会将系统从省电模式唤醒。由于 WKUP 引脚 2 为高电平有效，WKUP 引脚 2 内部被配置为输入下拉模式。当在输入已经为高的时候置位该控制位，将会触发一个唤醒事件。
9	WUPEN1	WKUP 引脚 1（PC13）唤醒使能 0：关闭 WKUP 引脚 1 唤醒功能 1：开启 WKUP 引脚 1 唤醒功能 如果 WUPEN1 在进入省电模式之前置 1，WKUP 引脚 1 的上升沿会将系统从省电模式唤醒。由于 WKUP 引脚 1 为高电平有效，WKUP 引脚 1 内部被配置为输入下拉模式。当在输入已经为高的时候置位该控制位，将会触发一个唤醒事件。
8	WUPEN0	WKUP 引脚 0（PA0）唤醒使能 0：关闭 WKUP 引脚 0 唤醒功能 1：开启 WKUP 引脚 0 唤醒功能 如果 WUPEN0 在进入省电模式之前置 1，WKUP 引脚 0 的上升沿会将系统从省电模式唤醒。由于 WKUP 引脚 0 为高电平有效，WKUP 引脚 0 内部被配置为输入下拉模式。当在输入已经为高的时候置位该控制位，将会触发一个唤醒事件。
7	WUPEN6	WKUP 引脚 6（PB15）唤醒使能 0：关闭 WKUP 引脚 6 唤醒功能 1：开启 WKUP 引脚 6 唤醒功能 如果 WUPEN6 在进入省电模式之前置 1，WKUP 引脚 6 的上升沿会将系统从省电模式唤醒。由于 WKUP 引脚 6 为高电平有效，WKUP 引脚 6 内部被配置为输入下拉模式。当在输入已经为高的时候置位该控制位，将会触发一个唤醒事件。
6:3	保留	必须保持复位值
2	LVDF	低电压状态标志 0：低电压事件没出现（V_{DD} 高于设定的 LVD 阈值） 1：低电压事件出现（V_{DD} 等于或低于 LVD 阈值）

		注意：LVD 功能在待机模式被禁用。
1	STBF	待机标志 0：设备没进入过待机模式 1：设备曾进入过待机模式 该位只能由 POR/PDR 或通过置位 PMU_CTL0 寄存器的 STBRST 位来清零。
0	WUF	唤醒标志 0：没有收到唤醒事件 1：收到来自 WKUP 引脚或 RTC 闹钟事件 该位只能由 POR/PDR 或通过设置 PMU_CTL0 寄存器的 WURST 位来清零。

4.4.3. 控制寄存器 1（PMU_CTL1）

地址偏移：0x08

复位值：0x0000 0000（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
保留														DPMOD2	DPMOD1
														rw	rw

位/位域	名称	描述
31:2	保留	必须保持复位值
1	DPMOD2	深度睡眠模式 2 使能 0：无作用 1：当 SLEEPDEEP 位为 1，STBMOD 位为 0 时，进入深度睡眠模式 2
0	DPMOD1	深度睡眠模式 1 使能 0：无作用 1：当 SLEEPDEEP 位为 1，STBMOD 位为 0，且 DPMOD2 位为 0 时，进入深度睡眠模式 1

4.4.4. 电源控制和状态寄存器 1（PMU_CS1）

地址偏移：0x0C

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。

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														DPF2	DPF1
														rc_w0	rc_w0

位/位域	名称	描述
31:2	保留	必须保持复位值
1	DPF2	深度睡眠模式 2 状态标志位。当进入深度睡眠模式 2 时该位由硬件置位。软件写 0 清除该位。
0	DPF1	深度睡眠模式 1 状态标志位。当进入深度睡眠模式 1 时该位由硬件置位。软件写 0 清除该位。

5. 复位和时钟单元（RCU）

5.1. 复位控制单元（RCTL）

5.1.1. 简介

GD32A508xx 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，其复位除了备份域的所有系统。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. 功能描述

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 1.1V 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

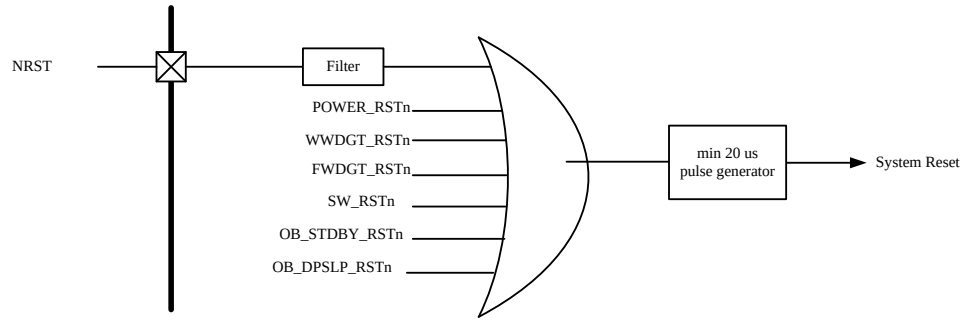
当发生以下任一事件时，产生一个系统复位：

- 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®-M33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（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 VDD 和 VBAT 两者都掉电的前提下，VDD 或 VBAT 上电）。

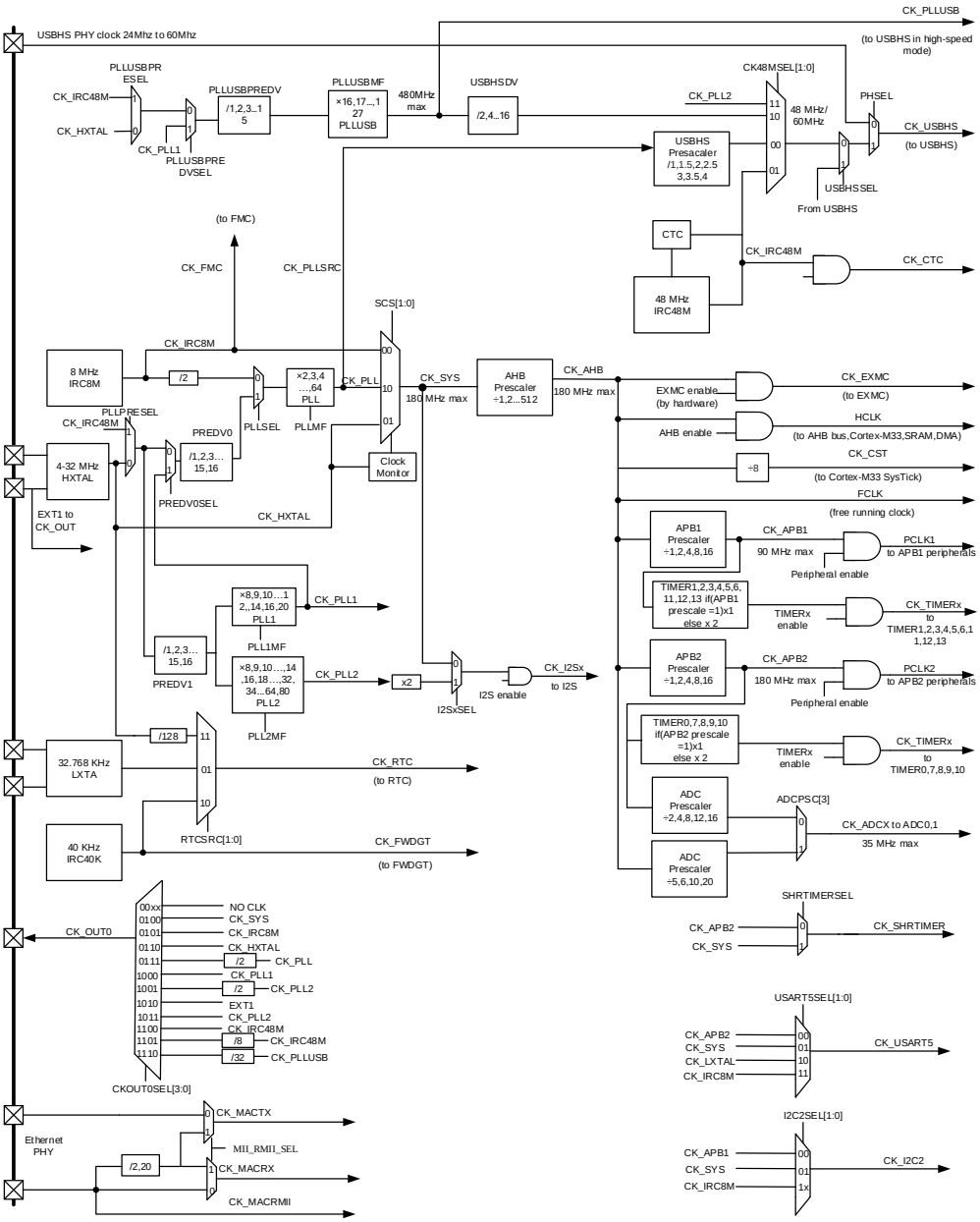
5.2. 时钟控制单元（CCTL）

5.2.1. 简介

时钟控制单元提供了一系列频率的时钟功能，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）、四个锁相环（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®-M33 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 180MHz。

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 180MHz、180MHz、90MHz。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 5、6、10、20 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。USART5 时钟由 IRC8M 或 LXTAL 或 CK_SYS 或 APB2 时钟提供，通过配置 RCU_CFG2 寄存器的 USART5SEL 位来选择。I2C2 的时钟由 IRC8M 或 CK_SYS 或 APB1 时钟提供，通过配置 RCU_CFG2 寄存器的 I2C2SEL 位来选择。

TIMER 时钟由 CK_APB1 和 CK_APB2 时钟分频获得，如果 APBx（x=0,1）的分频系数不为 1，则 TIMER 时钟为 CK_APBx（x = 0,1）的两倍。

PLLUSB 时钟由 HXTAL 或 IRC48M 提供。通过配置 RCC_ADDCFG 寄存器的 PLLUSBPRESEL 来选择。PLLUSB 时钟工作的最大频率为 480Mhz

USBHS 的时钟由外部 PHY 时钟或内部时钟提供。通过配置 USBHS_GUSBCS 寄存器的 EMBHY 位来选择。

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 AFIO_PCF0 寄存器的 ENET_PHY_SEL 位，以太网 TX/RX 时钟可以选择由外部引脚（ENET_TX_CLK/ENET_RX_CLK）输入时钟提供。

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 1.1V 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。

当 FMC 启动时，FMC 时钟被强制选择由 IRC8M 时钟作为时钟源。

SHRTIMER时钟由CK_APB2或CK_SYS提供。通过配置RCU_CFG1寄存器的SHRTIMERSEL位来选择。

如果用户不需要使用SHRTIMER高分辨率模式，可以保持RCU_CFG1寄存器中的SHRTIMERSEL位清零，在这种情况下，SHRTIMER_MTCTL0寄存器中的CNTCKDIV[2:0]值必须大于或等于5（预分频比大于或等于64）。

如果用户需要使用SHRTIMER高分辨率模式，必须在系统时钟源选择为PLL时，通过配置RCU_CFG1寄存器中的SHRTIMERSEL位为1，选择CK_SYS为时钟源，此时SHRTIMER_MTCTL0寄存器中的CNTCKDIV[2:0]值可以配置为任何可选值。

注意：在高分辨率配置中，必须配置 AHB 和 APB2 预分频器（RCU_CFG0 寄存器中的 AHBPSC[3:0]和 APB2PSC[2:0]位），将系统时钟 CK_SYS 与 APB2 时钟 PCLK2 之间的比率为 1，2 或 4。

5.2.2. 主要特性

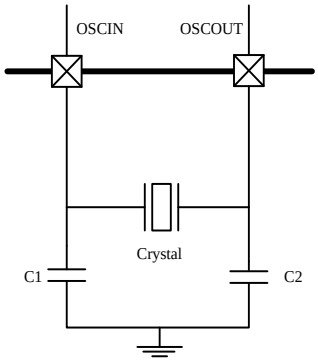
- 4到32MHz外部高速晶体振荡器（HXTAL）；
- 内部8MHz RC振荡器（IRC8M）；
- 内部48MHz RC振荡器；
- 32768 Hz外部低速晶体振荡器（LXTAL）；
- 内部40KHz RC振荡器（IRC40K）；
- PLL时钟源可选HXTAL、IRC8M或IRC48M；
- HXTAL时钟监视器。

5.2.3. 功能描述

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。

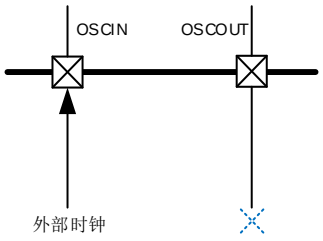
图 5-3. HXTAL 时钟源



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。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 PLLUSB/USBHS 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 PLLUSB/USBHS 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）

CL 系列的芯片，内部有四个锁相环，PLL，PLL1，PLL2 和 PLLUSB。

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。

PLLUSB 可以通过设置 RCU_ADDCTL 寄存器中的 PLLUSBEN 位被启动和关闭。RCU_ADDCTL 寄存器中的 PLLUSBSTB 位来指示 PLLUSB 时钟是否稳定。如果 RCU_ADDINT 寄存器中的相应中断使能位 PLLUSBSTBIE 被置‘1’，在 PLLUSB 稳定以后，将产生一个中断。

当进入 Deepsleep/Standby 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），四个 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®-M33 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。

时钟输出功能

时钟输出功能输出从 0.09375MHz 到 180MHz 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

时钟输出 0 的时钟源选择位域	时钟源
1010	EXT1
1011	CK_PLL2
1100	CK_IRC48M
1101	CK_IRC48M/8
1110	CK_PLLUSB/32

电压控制

深度睡眠模式电压寄存器（RCU_DSV）中的 DSLPVS[2:0]位域可以控制 1.1V 域在深度睡眠模式下的电压。

表 5-2. 深度睡眠模式下 1.1V 域电压选择

DSLPVS[2:0]	深度睡眠模式电压（V）
000	1.0
001	0.9
010	0.8
011	0.7

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 硬件置 1 来表示 PLL2 输出时钟是否稳定待用 0：PLL2 未稳定 1：PLL2 已稳定
28	PLL2EN	PLL2 使能 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 0：PLL2 被关闭 1：PLL2 被打开
27	PLL1STB	PLL1 时钟稳定标志位 硬件置 1 来表示 PLL1 输出时钟是否稳定待用 0：PLL1 未稳定 1：PLL1 已稳定
26	PLL1EN	PLL1 使能 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 0：PLL1 被关闭 1：PLL1 被打开
25	PLLSTB	PLL 时钟稳定标志位 硬件置 1 来表示 PLL 输出时钟是否稳定待用

		0：PLL 未稳定 1：PLL 已稳定
24	PLLEN	PLL 使能 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 0：PLL 被关闭 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 0：禁止高速 4 ~ 32 MHz 晶体振荡器（HXTAL）时钟监视器 1：使能高速 4 ~ 32 MHz 晶体振荡器（HXTAL）时钟监视器 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。 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 只有在 HXTALEN 位为 0 时 HXTALBPS 位才可写 0：禁止 HXTAL 旁路模式 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 硬件置‘1’来指示 HXTAL 振荡器时钟是否稳定待用 0：HXTAL 振荡器未稳定 1：HXTAL 振荡器已稳定
16	HXTALEN	高速晶体振荡器（HXTAL）使能 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 0：高速 4 ~ 32 MHz 晶体振荡器被关闭 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 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 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 硬件置‘1’来指示 IRC8M 振荡器时钟是否稳定待用 0：IRC8M 振荡器未稳定

		1：IRC8M 振荡器已稳定
0	IRC8MEN	内部 8MHz RC 振荡器使能 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。 0：内部 8MHz RC 振荡器被关闭 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
USBHSPSC[2]	PLLMF[5:4]		ADCPSC[2]	CKOUT0SEL[3:0]				USBHSPSC[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	USBHSPSC[2]	USBHSPSC 的第 2 位 参考寄存器 RCU_CFG0 的 22 到 23 位
30:29	PLLMF[5:4]	PLLMF 的第 5 位和第 4 位 参考寄存器 RCU_CFG0 的 18 到 21 位
28	ADCPSC[2]	ADCPSC 的第 2 位 参考寄存器 RCU_CFG0 的 14 到 15 位
27:24	CKOUT0SEL[3:0]	CKOUT0 时钟源选择 由软件置位或清零 00xx：无时钟输出 0100：选择系统时钟 CK_SYS 0101：选择内部 8M RC 振荡器时钟 0110：选择高速晶体振荡器时钟 （HXTAL） 0111：选择（CK_PLL / 2）时钟 1000：选择 CK_PLL1 时钟 1001：选择（CK_PLL2 / 2）时钟 1010：选择提供给 ENET 的 EXT1 时钟 1011：选择 CK_PLL2 时钟 1100：选择 CK_IRC48M 时钟 1101：选择（CK_IRC48M/8）时钟

1110：选择（CK_PLLUSB/32）时钟

23:22	USBHSPSC[1:0]	USBHS 的时钟分频系数 由软件置位或清零。USBHS 的时钟必须为 48MHz，当 USBHS 时钟使能的时候，这些位无法修改 000：CK_USBHS= CK_PLL / 1.5 001：CK_USBHS= CK_PLL 010：CK_USBHS= CK_PLL / 2.5 011：CK_USBHS= CK_PLL / 2 100：CK_USBHS= CK_PLL / 3 101：CK_USBHS= CK_PLL / 3.5 11x：CK_USBHS= CK_PLL / 4
21:18	PLLMF[3:0]	PLL 时钟倍频因子 与寄存器 RCU_CFG0 的 29,30 位共同构成倍频因子，由软件置位或清零 注意：PLL 输出时钟频率不能超过 180MHz 000000：（PLL 源时钟 x 2） 000001：（PLL 源时钟 x 3） 000010：（PLL 源时钟 x 4） 000011：（PLL 源时钟 x 5） 000100：（PLL 源时钟 x 6） 000101：（PLL 源时钟 x 7） 000110：（PLL 源时钟 x 8） 000111：（PLL 源时钟 x 9） 001000：（PLL 源时钟 x 10） 001001：（PLL 源时钟 x 11） 001010：（PLL 源时钟 x 12） 001011：（PLL 源时钟 x 13） 001100：（PLL 源时钟 x 14） 001101：（PLL 源时钟 x 6.5） 001110：（PLL 源时钟 x 16） 001111：（PLL 源时钟 x 16） 010000：（PLL 源时钟 x 17） 010001：（PLL 源时钟 x 18） 010010：（PLL 源时钟 x 19） 010011：（PLL 源时钟 x 20） 010100：（PLL 源时钟 x 21） 010101：（PLL 源时钟 x 22） 010110：（PLL 源时钟 x 23） 010111：（PLL 源时钟 x 24） 011000：（PLL 源时钟 x 25） 011001：（PLL 源时钟 x 26） 011010：（PLL 源时钟 x 27） 011011：（PLL 源时钟 x 28） 011100：（PLL 源时钟 x 29）

011101：（PLL 源时钟 x 30）
011110：（PLL 源时钟 x 31）
011111：（PLL 源时钟 x 32）
100000：（PLL 源时钟 x 33）
100001：（PLL 源时钟 x 34）
…
111110：（PLL 源时钟 x 63）
111111：（PLL 源时钟 x 64）

17	PREDV0_LSB	PREDV0 分频因子的最低位 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 由软件置位或复位，控制 PLL 时钟源 0：（IRC8M/2）被选择为 PLL 时钟的时钟源 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 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 0000：CK_ADC = CK_APB2 / 2 0001：CK_ADC = CK_APB2 / 4 0010：CK_ADC = CK_APB2 / 6 0011：CK_ADC = CK_APB2 / 8 0100：CK_ADC = CK_APB2 / 2 0101：CK_ADC = CK_APB2 / 12 0110：CK_ADC = CK_APB2 / 8 0111：CK_ADC = CK_APB2 / 16 1x00：CK_ADC = CK_AHB / 5 1x01：CK_ADC = CK_AHB / 6 1x10：CK_ADC = CK_AHB / 10 1x11：CK_ADC = CK_AHB / 20
13:11	APB2PSC[2:0]	APB2 预分频选择 由软件置位或清零，控制 APB2 时钟分频因子 0xx：CK_APB2 = CK_AHB 100：CK_APB2 = CK_AHB / 2 101：CK_APB2 = CK_AHB / 4 110：CK_APB2 = CK_AHB / 8 111：CK_APB2 = CK_AHB / 16
10:8	APB1PSC[2:0]	APB1 预分频选择 由软件置位或清零，控制 APB1 时钟分频因子. 0xx：CK_APB1 = CK_AHB

100：CK_APB1 = CK_AHB / 2
101：CK_APB1 = CK_AHB / 4
110：CK_APB1 = CK_AHB / 8
111：CK_APB1 = CK_AHB / 16

7:4	AHBPSC[3:0]	AHB 预分频选择 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 由硬件置位或清零，标识当前系统时钟的时钟源 00：选择 CK_IRC8M 时钟作为 CK_SYS 时钟源 01：选择 CK_HXTAL 时钟作为 CK_SYS 时钟源 10：选择 CK_PLL 时钟作为 CK_SYS 时钟源 11：保留
1:0	SCS[1:0]	系统时钟选择 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。 00：选择 CK_IRC8M 时钟作为 CK_SYS 时钟源 01：选择 CK_HXTAL 时钟作为 CK_SYS 时钟源 10：选择 CK_PLL 时钟作为 CK_SYS 时钟源 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	PLL1	PLL	HXTAL	IRC8M	LXTAL	IRC40K	CKMIF	PLL2	PLL1	PLL	HXTAL	IRC8M	LXTAL	IRC40K

	STBIE	STBIE	STBIE	STBIE	STBIE	STBIE	STBIE		STBIF	STBIF	STBIF	STBIF	STBIF	STBIF	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 软件写 1 复位 CKMIF 标志位. 0：不复位 CKMIF 标志位 1：复位 CKMIF 标志位
22	PLL2STBIC	PLL2 时钟稳定中断清零 软件写 1 复位 PLL2STBIF 标志位 0：不复位 PLL2STBIF 标志位 1：复位 PLL2STBIF 标志位
21	PLL1STBIC	PLL1 时钟稳定中断清零 软件写 1 复位 PLL1STBIF 标志位 0：不复位 PLL1STBIF 标志位 1：复位 PLL1STBIF 标志位
20	PLLSTBIC	PLL 时钟稳定中断清零 软件写 1 复位 PLLSTBIF 标志位 0：不复位 PLLSTBIF 标志位 1：复位 PLLSTBIF 标志位
19	HXTALSTBIC	HXTAL 时钟稳定中断清零 软件写 1 复位 HXTALSTBIF 标志位 0：不复位 HXTALSTBIF 标志位 1：复位 HXTALSTBIF 标志位
18	IRC8MSTBIC	IRC8M 时钟稳定中断清零 软件写 1 复位 IRC8MSTBIF 标志位 0：不复位 IRC8MSTBIF 标志位 1：复位 IRC8MSTBIF 标志位
17	LXTALSTBIC	LXTAL 时钟稳定中断清零 软件写 1 复位 LXTALSTBIF 标志位 0：不复位 LXTALSTBIF 标志位 1：复位 LXTALSTBIF 标志位
16	IRC40KSTBIC	IRC40K 时钟稳定中断清零 软件写 1 复位 IRC40KSTBIF 标志位 0：不复位 IRC40KSTBIF 标志位 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

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

0：无 PLL1 时钟稳定中断产生
1：产生 PLL1 时钟稳定中断

4	PLLSTBIF	PLL 时钟稳定中断标志位 当 PLL 时钟稳定且 PLLSTBIE 位被置 1 时由硬件置 1 软件置位 PLLSTBIC 位时清除该位 0：无 PLL 时钟稳定中断产生 1：产生 PLL 时钟稳定中断
3	HXTALSTBIF	HXTAL 时钟稳定中断标志位 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 软件置位 HXTALSTBIC 位时清除该位 0：无 HXTAL 时钟稳定中断产生 1：产生 HXTAL 时钟稳定中断
2	IRC8MSTBIF	IRC8M 时钟稳定中断标志位 当内部 8MHz RC 振荡器时钟稳定且 IRC8MSTBIE 位被置 1 时由硬件置 1 软件置位 IRC8MSTBIC 位时清除该位 0：无 IRC8M 时钟稳定中断产生 1：产生 IRC8M 时钟稳定中断
1	LXTALSTBIF	LXTAL 时钟稳定中断标志位 当低速晶体振荡器时钟稳定且 LXTALSTBIE 位被置 1 时由硬件置 1 软件置位 LXTALSTBIC 位时清除该位 0：无 LXTAL 时钟稳定中断产生 1：产生 LXTAL 时钟稳定中断
0	IRC40KSTBIF	IRC40K 时钟稳定中断标志位 当内部 40kHz RC 振荡器时钟稳定且 IRC40KSTBIE 位被置 1 时由硬件置 1 软件置位 IRC40KSTBIC 位时清除该位 0：无 IRC40K 时钟稳定中断产生 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
CMPRST	保留	SHRTIMERRST	USART5RST	保留						TIMER10RST	TIMER9RST	TIMER8RST	保留		
rw		rw	rw							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	PGRST	PFRST	PERST	PDRST	PCRST	PBRST	PARST	保留	AFRST
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

位/位域	名称	描述
31	CMPRST	CMP 复位 由软件置位或复位 0：无作用 1：复位 CMP
30	保留	必须保持复位值。
29	SHRTIMEREN	SHRTIMER 复位 由软件置位或复位 0：无作用 1：复位 SHRTIMER
28	USART5EN	USART5 复位 由软件置位或复位 0：无作用 1：复位 USART5
27:22	保留	必须保持复位值。
21	TIMER10RST	TIMER10 复位 由软件置位或复位 0：无作用 1：复位 TIMER10
20	TIMER9RST	TIMER9 复位 由软件置位或复位 0：无作用 1：复位 TIMER9
19	TIMER8RST	TIMER8 复位 由软件置位或复位 0：无作用 1：复位 TIMER8
18:15	保留	必须保持复位值。
14	USART0RST	USART0 复位 由软件置位或复位 0：无作用 1：复位 USART0
13	TIMER7RST	TIMER7 复位 由软件置位或复位 0：无作用 1：复位 TIMER7
12	SPI0RST	SPI0 复位 由软件置位或复位

		0：无作用 1：复位 SPI0
11	TIMER0RST	TIMER0 复位 由软件置位或复位 0：无作用 1：复位 TIMER0
10	ADC1RST	ADC1 复位 由软件置位或复位 0：无作用 1：复位 ADC1
9	ADC0RST	ADC0 复位 由软件置位或复位 0：无作用 1：复位 ADC0
8	PGRST	GPIO 端口 G 复位 由软件置位或复位 0：无作用 1：复位 GPIO 端口 G
7	PFRST	GPIO 端口 F 复位 由软件置位或复位 0：无作用 1：复位 GPIO 端口 F
6	PERST	GPIO 端口 E 复位 由软件置位或复位 0：无作用 1：复位 GPIO 端口 E
5	PDRST	GPIO 端口 D 复位 由软件置位或复位 0：无作用 1：复位 GPIO 端口 D
4	PCRST	GPIO 端口 C 复位 由软件置位或复位 0：无作用 1：复位 GPIO 端口 C
3	PBRST	GPIO 端口 B 复位 由软件置位或复位 0：无作用 1：复位 GPIO 端口 B
2	PARST	GPIO 端口 A 复位

由软件置位或复位
0：无作用
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 由软件置位或复位 0：无作用 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	CAN1RST	CAN0RST	I2C2RST	保留	I2C1RST	I2C0RST	UART4RST	UART3RST	USART2RST	USART1RST	保留
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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 由软件置位或复位 0：无作用 1：复位 DAC
28	PMURST	PMU 复位 由软件置位或复位 0：无作用 1：复位 PMU
27	BKPIRST	BKPI 复位 由软件置位或复位 0：无作用 1：复位 BKP
26	CAN1RST	CAN1 复位 由软件置位或复位 0：无作用

		1：复位 CAN1
25	CAN0RST	CAN0 复位 由软件置位或复位 0：无作用 1：复位 CAN0
24	I2C2RST	I2C2 复位 由软件置位或复位 0：无作用 1：复位 I2C2
23	保留	必须保持复位值
22	I2C1RST	I2C1 复位 由软件置位或复位 0：无作用 1：复位 I2C1
21	I2C0RST	I2C0 复位 由软件置位或复位 0：无作用 1：复位 I2C0
20	UART4RST	UART4 复位 由软件置位或复位 0：无作用 1：复位 UART4
19	UART3RST	UART3 复位 由软件置位或复位 0：无作用 1：复位 UART3
18	USART2RST	USART2 复位 由软件置位或复位 0：无作用 1：复位 USART2
17	USART1RST	USART1 复位 由软件置位或复位 0：无作用 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 0：无作用

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

		1：复位 TIMER3
1	TIMER2RST	TIMER2 复位 由软件置位或复位 0：无作用 1：复位 TIMER2
0	TIMER1RST	TIMER1 复位 由软件置位或复位 0：无作用 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

注意：Bit4 不能被清为 0。

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
SQPIEN	TMUEN	保留													ENETRXEN
rw	rw														rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENETTXEN	ENETEN	ULPIEN	USBHSEN	保留			EXMCEN	保留	CRCEN	保留			SRAMSPEN	DMA1EN	DMA0EN
rw	rw	rw	rw				rw		rw				rw	rw	rw

位/位域	名称	描述
31	SQPIEN	SQPI 时钟使能 由软件置位或复位 0：关闭 SQPI 时钟 1：开启 SQPI 时钟
30	TMUEN	TMU 时钟使能 由软件置位或复位 0：关闭 TMU 时钟 1：开启 TMU 时钟
29:17	保留	必须保持复位值。
16	ENETRXEN	以太网 RX 时钟使能 由软件置位或复位 0：关闭以太网 RX 时钟 1：开启以太网 RX 时钟
15	ENETTXEN	以太网 TX 时钟使能

		由软件置位或复位 0：关闭以太网 TX 时钟 1：开启以太网 TX 时钟
14	ENETEN	以太网时钟使能 由软件置位或复位 0：关闭以太网时钟 1：开启以太网时钟
13	ULPIEN	ULPI 时钟使能 由软件置位或复位 0：关闭 ULPI 时钟 1：开启 ULPI 时钟
12	USBHSEN	USBHS 时钟使能 由软件置位或复位 0：关闭 USBHS 时钟 1：开启 USBHS 时钟
11:9	保留	必须保持复位值。
8	EXMCEN	EXMC 时钟使能 由软件置位或复位 0：关闭 EXMC 时钟 1：开启 EXMC 时钟
7	保留	必须保持复位值。
6	CRCEN	CRC 时钟使能 由软件置位或复位 0：关闭 CRC 时钟 1：开启 CRC 时钟
5:3	保留	必须保持复位值。
2	SRAMSPEN	在睡眠模式下 SRAM 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 SRAM 时钟 1：在睡眠模式下开启 SRAM 时钟
1	DMA1EN	DMA1 时钟使能 由软件置位或复位 0：关闭 DMA1 时钟 1：开启 DMA1 时钟
0	DMA0EN	DMA0 时钟使能 由软件置位或复位 0：关闭 DMA0 时钟 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
CMPEN	保留	SHRTIMEREN	USART5EN	保留						TIMER10EN	TIMER9EN	TIMER8EN	保留		
rw		rw	rw							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	PGEN	PFEN	PEEN	PDEN	PCEN	PBEN	PAEN	保留	AFEN
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

位/位域	名称	描述
31	CMPEN	CMP 时钟使能 由软件置位或复位 0：关闭 CMP 时钟 1：开启 CMP 时钟
30	保留	必须保持复位值。
29	SHRTIMEREN	SHRTIMER 时钟使能 由软件置位或复位 0：关闭 SHRTIMER 时钟 1：开启 SHRTIMER 时钟
28	USART5EN	USART5 时钟使能 由软件置位或复位 0：关闭 USART5 时钟 1：开启 USART5 时钟
27:22	保留	必须保持复位值。
21	TIMER10EN	TIMER10 时钟使能 由软件置位或复位 0：关闭 TIMER10 时钟 1：开启 TIMER10 时钟
20	TIMER9EN	TIMER9 时钟使能 由软件置位或复位 0：关闭 TIMER9 时钟 1：开启 TIMER9 时钟
19	TIMER8EN	TIMER8 时钟使能 由软件置位或复位 0：关闭 TIMER8 时钟

		1：开启 TIMER8 时钟
18:15	保留	必须保持复位值。
14	USART0EN	USART0 时钟使能 由软件置位或复位 0：关闭 USART0 时钟 1：开启 USART0 时钟
13	TIMER7EN	TIMER7 时钟使能 由软件置位或复位 0：关闭 TIMER7 时钟 1：开启 TIMER7 时钟
12	SPI0EN	SPI0 时钟使能 由软件置位或复位 0：关闭 SPI0 时钟 1：开启 SPI0 时钟
11	TIMER0EN	TIMER0 时钟使能 由软件置位或复位 0：关闭 TIMER0 时钟 1：开启 TIMER0 时钟
10	ADC1EN	ADC1 时钟使能 由软件置位或复位 0：关闭 ADC1 时钟 1：开启 ADC1 时钟
9	ADC0EN	ADC0 时钟使能 由软件置位或复位 0：关闭 ADC0 时钟 1：开启 ADC0 时钟
8	PGEN	GPIO 端口 G 时钟使能 由软件置位或复位 0：关闭 GPIO 端口 G 时钟 1：开启 GPIO 端口 G 时钟
7	PFEN	GPIO 端口 F 时钟使能 由软件置位或复位 0：关闭 GPIO 端口 F 时钟 1：开启 GPIO 端口 F 时钟
6	PEEN	GPIO 端口 E 时钟使能 由软件置位或复位 0：关闭 GPIO 端口 E 时钟 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 由软件置位或复位 0：关闭 GPIO 端口 C 时钟 1：开启 GPIO 端口 C 时钟
3	PBEN	GPIO 端口 B 时钟使能 由软件置位或复位 0：关闭 GPIO 端口 B 时钟 1：开启 GPIO 端口 B 时钟
2	PAEN	GPIO 端口 A 时钟使能 由软件置位或复位 0：关闭 GPIO 端口 A 时钟 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 由软件置位或复位 0：关闭复用功能 IO 时钟 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	CAN1EN	CAN0EN	I2C2EN	保留	I2C1EN	I2C0EN	UART4EN	UART3EN	USART2EN	USART1EN	保留
		rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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 由软件置位或复位 0：关闭 DAC 时钟

		1：开启 DAC 时钟
28	PMUEN	PMU 时钟使能 由软件置位或复位 0：关闭 PMU 时钟 1：开启 PMU 时钟
27	BKPIEN	BKP 时钟使能 由软件置位或复位 0：关闭 BKP 时钟 1：开启 BKP 时钟
26	CAN1EN	CAN1 时钟使能 由软件置位或复位 0：关闭 CAN1 时钟 1：开启 CAN1 时钟
25	CAN0EN	CAN0 时钟使能 由软件置位或复位 0：关闭 CAN0 时钟 1：开启 CAN0 时钟
24	I2C2EN	I2C2 时钟使能 由软件置位或复位 0：关闭 I2C2 时钟 1：开启 I2C2 时钟
23	保留	必须保持复位值。
22	I2C1EN	I2C1 时钟使能 由软件置位或复位 0：关闭 I2C1 时钟 1：开启 I2C1 时钟
21	I2C0EN	I2C0 时钟使能 由软件置位或复位 0：关闭 I2C0 时钟 1：开启 I2C0 时钟
20	UART4EN	UART4 时钟使能 由软件置位或复位 0：关闭 UART4 时钟 1：开启 UART4 时钟
19	UART3EN	UART3 时钟使能 由软件置位或复位 0：关闭 UART3 时钟 1：开启 UART3 时钟
18	USART2EN	USART2 时钟使能

		由软件置位或复位 0：关闭 USART2 时钟 1：开启 USART2 时钟
17	USART1EN	USART1 时钟使能 由软件置位或复位 0：关闭 USART1 时钟 1：开启 USART1 时钟
16	保留	必须保持复位值。
15	SPI2EN	SPI2 时钟使能 由软件置位或复位 0：关闭 SPI2 时钟 1：开启 SPI2 时钟
14	SPI1EN	SPI1 时钟使能 由软件置位或复位 0：关闭 SPI1 时钟 1：开启 SPI1 时钟
13:12	保留	必须保持复位值。
11	WWDGTEN	WWDGT 时钟使能 由软件置位或复位 0：关闭 WWDGT 时钟 1：开启 WWDGT 时钟
10:9	保留	必须保持复位值。
8	TIMER13EN	TIMER13 时钟使能 由软件置位或复位 0：关闭 TIMER13 时钟 1：开启 TIMER13 时钟
7	TIMER12EN	TIMER12 时钟使能 由软件置位或复位 0：关闭 TIMER12 时钟 1：开启 TIMER12 时钟
6	TIMER11EN	TIMER11 时钟使能 由软件置位或复位 0：关闭 TIMER11 时钟 1：开启 TIMER11 时钟
5	TIMER6EN	TIMER6 时钟使能 由软件置位或复位 0：关闭 TIMER6 时钟 1：开启 TIMER6 时钟

4	TIMER5EN	TIMER5 时钟使能 由软件置位或复位 0：关闭 TIMER5 时钟 1：开启 TIMER5 时钟
3	TIMER4EN	TIMER4 时钟使能 由软件置位或复位 0：关闭 TIMER4 时钟 1：开启 TIMER4 时钟
2	TIMER3EN	TIMER3 时钟使能 由软件置位或复位 0：关闭 TIMER3 时钟 1：开启 TIMER3 时钟
1	TIMER2EN	TIMER2 时钟使能 由软件置位或复位 0：关闭 TIMER2 时钟 1：开启 TIMER2 时钟
0	TIMER1EN	TIMER1 时钟使能 由软件置位或复位 0：关闭 TIMER1 时钟 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。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCEN	保留					RTCSRC[1:0]		保留			LXTALDRI[1:0]		LXTALBPS	LXTALSTB	LXTALEN
rw						rw					rw		rw	r	rw

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

		由软件置位或复位 0：无作用 1：复位备份域
15	RTCEN	RTC 时钟使能 由软件置位或复位 0：关闭 RTC 时钟 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 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 01：选择 CK_LXTAL 时钟作为 RTC 的时钟源 10：选择 CK_IRC40K 时钟作为 RTC 的时钟源 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 由软件置位或复位。当备份域复位时将复位该值 00：弱驱动能力 01：中低驱动能力 10：中高驱动能力 11：强驱动能力（复位后的缺省值） 注意：LXTALDRI 位在旁路模式下无效
2	LXTALBPS	LXTAL 旁路模式使能 由软件置位或复位 0：禁止 LXTAL 旁路模式 1：使能 LXTAL 旁路模式
1	LXTALSTB	低速晶体振荡器稳定标志位 硬件置‘1’来指示 LXTAL 振荡器时钟是否稳定待用 0：LXTAL 未稳定 1：LXTAL 已稳定
0	LXTALEN	LXTAL 时钟使能 由软件置位或复位 0：关闭 LXTAL 时钟 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
LPRSTF	WWDGTRSTF	FWDGTRSTF	SWRSTF	PORRSTF	EPRSTF	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														IRC40KSTB	IRC40KEN
														r	rw

位/位域	名称	描述
31	LPRSTF	低功耗复位标志位 深度睡眠/待机复位发生时由硬件置位 向 RSTFC 位写 1 来清除该位 0：无低功耗管理复位发生 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 窗口看门狗定时器复位发生时由硬件置 1 向 RSTFC 位写 1 来清除该位 0：无窗口看门狗复位发生 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 独立看门狗复位发生时由硬件置 1 向 RSTFC 位写 1 来清除该位 0：无独立看门狗定时器复位发生 1：发生独立看门狗定时器复位
28	SWRSTF	软件复位标志位 软件复位发生时由硬件置 1 向 RSTFC 位写 1 来清除该位 0：无软件复位发生 1：发生软件复位
27	PORRSTF	电源复位标志位 电源复位发生时由硬件置 1 向 RSTFC 位写 1 来清除该位 0：无电源复位发生 1：发生电源复位
26	EPRSTF	外部引脚复位标志位 外部引脚复位发生时由硬件置 1 向 RSTFC 位写 1 来清除该位 0：无外部引脚复位发生

		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 由软件置 1 来清除所有复位标志位 0：无作用 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 该位由硬件置 1 指示 IRC40K 输出时钟是否稳定待用 0：IRC40K 时钟未稳定 1：IRC40K 已稳定
0	IRC40KEN	IRC40K 使能 由软件置位和复位 0：关闭 IRC40K 时钟 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
SQPIRST	TMURST	保留													
rw	rw														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	ENETRST	保留	USBHSRST	保留											
	rw		rw												

位/位域	名称	描述
31	SQPIRST	SQPI 复位 由软件置位或复位 0：无作用 1：复位 SQPI
30	TMURST	TMU 复位 由软件置位或复位 0：无作用 1：复位 TMU
29:15	保留	必须保持复位值。

14	ENETRST	ENET 复位 由软件置位或复位 0：无作用 1：复位 ENET
13	保留	必须保持复位值。
12	USBHSRST	USBHS 复位 由软件置位或复位 0：无作用 1：复位 USBHS
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
PLL2MF[4]	PLLREPSEL	ADCPSC[3]	PLL2MF[5]	保留								SHRTIMERSEL	I2S2SEL	I2S1SEL	PREDV0SEL
rw	rw	rw	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	PLL2MF[4]	PLL2MF 的第 4 位 参考寄存器 RCU_CFG1 的 12 到 15 位
30	PLLPRESEL	PLL 时钟源预选择 由软件置位或复位，控制 PLL 时钟源 0：HXTAL 被选择为 PLL 时钟的时钟源 1：CK_IRC48M 被选择为 PLL 时钟的时钟源
29	ADCPSC[3]	ADCPSC 的第 3 位 参考寄存器 RCU_CFG0 的 14 到 15 位
28	PLL2MF[5]	PLL2MF 的第 5 位 参考寄存器 RCU_CFG1 的 12 到 15 位
27:20	保留	必须保持复位值。
19	SHRTIMERSEL	SHRTIMER 时钟源选择 由软件置位或复位，控制 SHRTIMER 时钟源 0：APB2 时钟被选择为 SHRTIMER 时钟的时钟源

		1：系统时钟被选择为 SHRTIMER 时钟的时钟源
18	I2S2SEL	I2S2 时钟源选择 由软件置位或复位，控制 I2S2 时钟源 0：系统时钟被选择为 I2S2 时钟的时钟源 1：（CK_PLL2 x 2） 被选择为 I2S2 时钟的时钟源
17	I2S1SEL	I2S1 时钟源选择 由软件置位或复位，控制 I2S1 时钟源 0：系统时钟被选择为 I2S1 时钟的时钟源 1：（CK_PLL2 x 2） 被选择为 I2S1 时钟的时钟源
16	PREDV0SEL	PREDV0 时钟源选择 由软件置位或复位 0：HXTAL 或 IRC48M 被选择为 PREDV0 的时钟源 1：CK_PLL1 被选择为 PREDV0 的时钟源
15:12	PLL2MF[3:0]	PLL2 时钟倍频因子 与寄存器 RCU_CFG1 的 31 位和 28 位共同构成倍频因子，由软件置位或清零 000xx：保留 0010x：保留 0110：（PLL2 时钟源 x 8） 0111：（PLL2 时钟源 x 9） 1000：（PLL2 时钟源 x 10） 1001：（PLL2 时钟源 x 11） 1010：（PLL2 时钟源 x 12） 1011：（PLL2 时钟源 x 13） 1100：（PLL2 时钟源 x 14） 1101：保留 1110：（PLL2 时钟源 x 16） 1111：（PLL2 时钟源 x 20） 10000：（PLL2 时钟源 x 18） … 10110：（PLL2 时钟源 x 24） 10111：（PLL2 时钟源 x 25） 11000：（PLL2 时钟源 x 26） 11001：（PLL2 时钟源 x 27） 11010：（PLL2 时钟源 x 28） 11011：（PLL2 时钟源 x 29） 11100：（PLL2 时钟源 x 30） 11101：（PLL2 时钟源 x 31） 11110：（PLL2 时钟源 x 32） 11111：（PLL2 时钟源 x 40） 100000：（PLL2 时钟源 x 34） … 111110：（PLL2 时钟源 x 64）

111111：（PLL2 时钟源 x 80）

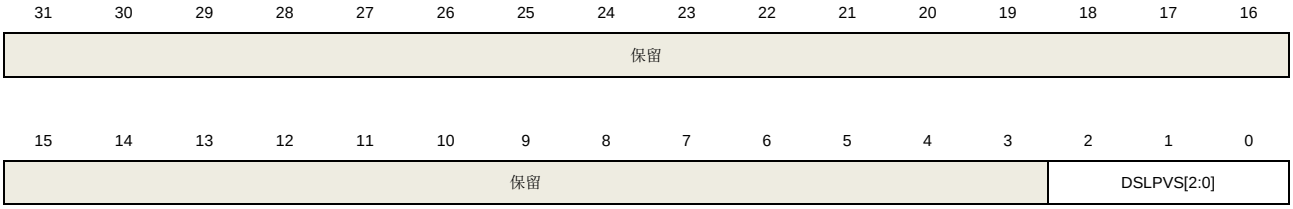
11:8	PLL1MF[3:0]	PLL1 时钟倍频因子 由软件置位或清零 00xx：保留 010x：保留 0110：（PLL1 源时钟 x 8） 0111：（PLL1 源时钟 x 9） 1000：（PLL1 源时钟 x 10） 1001：（PLL1 源时钟 x 11） 1010：（PLL1 源时钟 x 12） 1011：（PLL1 源时钟 x 13） 1100：（PLL1 源时钟 x 14） 1101：（PLL1 源时钟 x 15） 1110：（PLL1 源时钟 x 16） 1111：（PLL1 源时钟 x 20）
7:4	PREDV1[3:0]	PREDV1 分频因子 由软件置位或清零，PLL1 和 PLL2 未使能时，可以修改这些位 0000：PREDV1 输入源时钟未分频 0001：PREDV1 输入源时钟 2 分频 0010：PREDV1 输入源时钟 3 分频 0011：PREDV1 输入源时钟 4 分频 0100：PREDV1 输入源时钟 5 分频 0101：PREDV1 输入源时钟 6 分频 0110：PREDV1 输入源时钟 7 分频 0111：PREDV1 输入源时钟 8 分频 1000：PREDV1 输入源时钟 9 分频 1001：PREDV1 输入源时钟 10 分频 1010：PREDV1 输入源时钟 11 分频 1011：PREDV1 输入源时钟 12 分频 1100：PREDV1 输入源时钟 13 分频 1101：PREDV2 输入源时钟 14 分频 1110：PREDV2 输入源时钟 15 分频 1111：PREDV2 输入源时钟 16 分频
3:0	PREDV0[3:0]	PREDV0 分频因子 由软件置位或清零，PLL 未使能时，可以修改这些位 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 0000：PREDV0 输入源时钟未分频 0001：PREDV0 输入源时钟 2 分频 0010：PREDV0 输入源时钟 3 分频 0011：PREDV0 输入源时钟 4 分频 0100：PREDV0 输入源时钟 5 分频 0101：PREDV0 输入源时钟 6 分频

0110：PREDV0 输入源时钟 7 分频
0111：PREDV0 输入源时钟 8 分频
1000：PREDV0 输入源时钟 9 分频
1001：PREDV0 输入源时钟 10 分频
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。

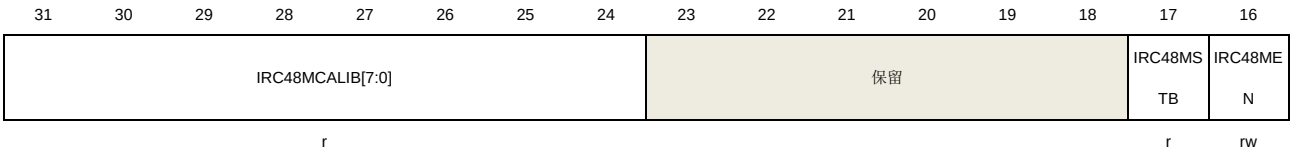


位/位域	名称	描述
31:3	保留	必须保持复位值。
2:0	DSLPVS[2:0]	深度睡眠模式电压选择 由软件置位和清零这些位 000：在深度睡眠模式下内核电压为 1.0V 001：在深度睡眠模式下内核电压为 0.9V 010：在深度睡眠模式下内核电压为 0.8V 011：在深度睡眠模式下内核电压为 0.7V 1xx：保留

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。



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLLUSBSTB	PLLUSBEN	保留							USBSWEN	USBHSDV[2:0]			USBHSSEL	CK48MSEL[1:0]	
r	rw								rw	rw			rw	rw	

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 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 硬件置‘1’来指示IRC48M振荡器时钟是否稳定待用 0：IRC48M未稳定 1：IRC48M已稳定
16	IRC48MEN	内部48MHz RC 振荡器使能 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 0：关闭IRC48M时钟 1：打开IRC48M时钟
15	PLLUSBSTB	PLLUSB时钟稳定标志位 硬件置‘1’来指示PLLUSB时钟是否稳定待用 0：PLLUSB时钟未稳定 1：PLLUSB时钟已稳定
14	PLLUSBEN	PLLUSB使能 由软件置位和复位。 0：关闭PLLUSB时钟 1：打开PLLUSB时钟
13:7	保留	必须保持复位值。
6	USBSWEN	USB时钟源选择使能 1：使用USBSW选择USB时钟 0：通过USBHS模块硬件选择USB时钟
5:3	USBHSDV[2:0]	USBHS时钟分频因子 由软件置位或清零。 000：USBHSDV输入源时钟2分频 001：USBHSDV输入源时钟4分频 … 111：USBHSDV输入源时钟16分频
2	USBHSSEL	USBHS时钟源选择 由软件置位和复位。 0：选择48M时钟作为USBHS时钟源 1：选择60M时钟作为USBHS时钟源

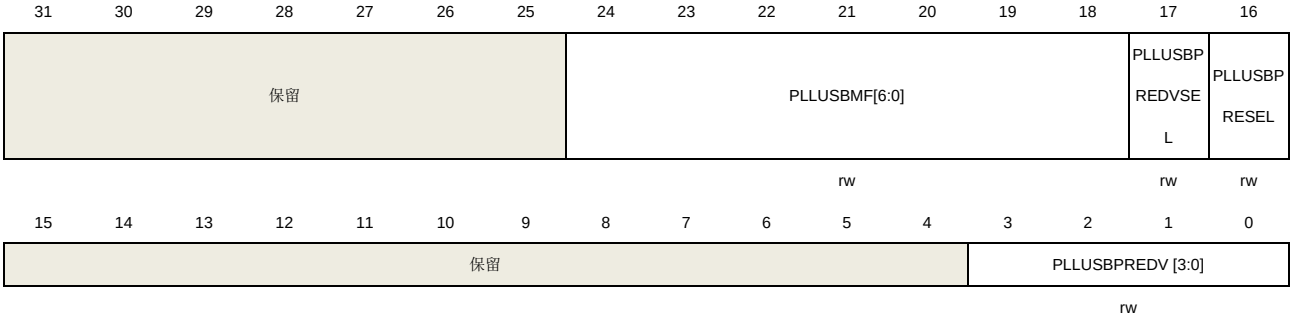
1:0	CK48MSEL	48MHz时钟源选择 由软件置位和复位。 00：选择CK_PLL / USBHSPSC时钟为48M时钟源 01：选择IRC48M时钟为48M时钟源 11：选择PLLUSB/USBHSDV为48M时钟源 11：选择PLL2位48M时钟源

5.3.15. 附加时钟配置寄存器（RCU_ADDCFG）

地址偏移： 0XC4

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
31:25	保留	必须保持复位值。
24:18	PLLUSBMF[6:0]	PLLUSB 时钟倍频因子 注意：PLLUSB 输出时钟频率不能超过 480MHz 0010000：CK_PLLUSB = CK_PLLUSBSRC x 16 0010001：CK_PLLUSB = CK_PLLUSBSRC x 17 0010010：CK_PLLUSB = CK_PLLUSBSRC x 18 0010011：CK_PLLUSB = CK_PLLUSBSRC x 19 … 1111111：CK_PLLUSB = CK_PLLUSBSRC x127
17	PLLUSBPREDVSEL	PLLUSBPREDV输入时钟源选择 由软件置位和复位。 0：PLLUSBSRC输出为PLLUSBPREDV输入时钟源 1：PLL1时钟为PLLUSBPREDV输入源时钟
16	PLLUSBPRESEL	PLLUSB时钟源预选择 由软件置位和复位选择PLLUSB时钟源。 0：选择CK_HXTAL时钟作为PLLUSB时钟源 1：选择CK_IRC48M时钟作为PLLUSB时钟源
15:4	保留	必须保持复位值。
3:0	PLLUSBPREDV[3:0]	PLLUSBPREDV的时钟分频系数

由软件置位和复位。

0000：保留

0001：PLLUSBPREDV 输入源时钟未分频

0010：PLLUSBPREDV 输入源时钟 2 分频

0011：PLLUSBPREDV 输入源时钟 3 分频

0100：PLLUSBPREDV 输入源时钟 4 分频

0101：PLLUSBPREDV 输入源时钟 5 分频

0110：PLLUSBPREDV 输入源时钟 6 分频

0111：PLLUSBPREDV 输入源时钟 7 分频

1000：PLLUSBPREDV 输入源时钟 8 分频

1001：PLLUSBPREDV 输入源时钟 9 分频

1010：PLLUSBPREDV 输入源时钟 10 分频

1011：PLLUSBPREDV 输入源时钟 11 分频

1100：PLLUSBPREDV 输入源时钟 12 分频

1101：PLLUSBPREDV 输入源时钟 13 分频

1110：PLLUSBPREDV 输入源时钟 14 分频

1111：PLLUSBPREDV 输入源时钟15分频

5.3.16. 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								PLLUSBSTBIC	IRC48MSTBIC	保留					
								w	w						

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

位/位域	名称	描述
31:24	保留	必须保持复位值。
23	PLLUSBSTBIC	PLLUSB 稳定中断清零 软件写 1 复位 PLLUSBSTBIF 标志位 0：不复位 PLLUSBSTBIF 标志位 1：复位 PLLUSBSTBIF 标志位
22	IRC48MSTBIC	内部 48 MHz RC 振荡器稳定中断清零 软件写 1 复位 IRC48MSTBIF 标志位 0：不复位 IRC48MSTBIF 标志位 1：复位 IRC48MSTBIF 标志位

21:16	保留	必须保持复位值。
15	PLLUSBSTBIE	PLLUSB 稳定中断使能 由软件置位和复位来使能/禁止 PLLUSB 时钟稳定中断 0：禁止 PLLUSB 时钟稳定中断 1：使能 PLLUSB 时钟稳定中断
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 0：禁止 IRC48M 时钟稳定中断 1：使能 IRC48M 时钟稳定中断
13:8	保留	必须保持复位值。
7	PLLUSBSTBIF	PLLUSB 时钟稳定中断标志位 当 PLLUSB 时钟稳定且 PLLUSB 位被置 1 时由硬件置 1 软件置位 PLLUSB 位时清除该位 0：无 PLLUSB 时钟稳定中断产生 1：产生 PLLUSB 时钟稳定中断
6	IRC48MSTBIF	IRC48M 时钟稳定中断标志位 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 软件置位 IRC48MSTBIC 位时清除该位 0：无 IRC48M 时钟稳定中断产生 1：产生 IRC48M 时钟稳定中断
5:0	保留	必须保持复位值。

5.3.17. PLL 时钟扩频控制寄存器（RCU_PLLSSCTL）

地址偏移：0xD0
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。

扩频调制仅适用于主 PLL 时钟

仅当 PLL 被禁止时，RCU_PLLSSCTL 寄存器才可写入

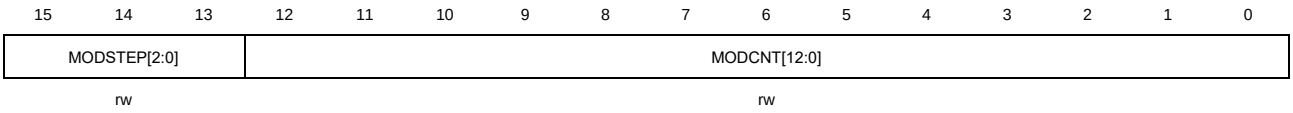
该寄存器用于配置 PLL 扩频时钟生成，需按照如下公式：

$$MODCNT = round（f_{PLLIN} / 4 / f_{mod}）$$

$$MODSTEP = round（mdamp * PLLN * 2^{15} /（MODCNT * 100））$$

f_{PLLIN} 表示 PLL 输入时钟频率，f_{mod} 表示扩频调制频率，mdamp 表示扩频调制振幅（按百分比表示），PLLN 表示 PLL 时钟频率倍频因子

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SSCGON	SS_TYPE	保留		MODSTEP[14:3]											
rw	rw			rw											

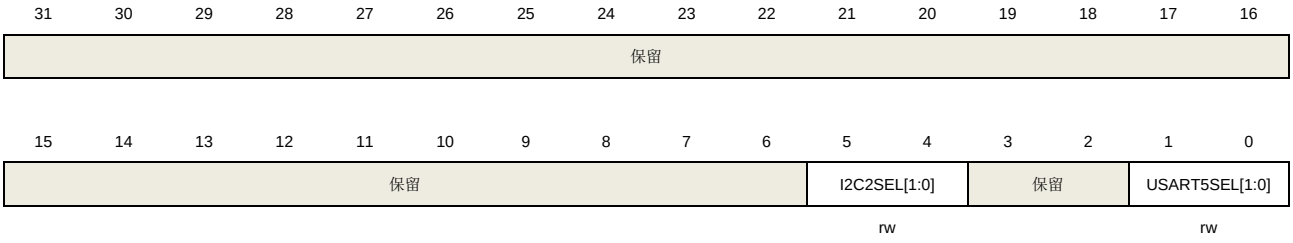


位/位域	名称	描述
31	SSCGON	PLL 扩频调制使能 0：禁止扩频调制 1：使能扩频调制
30	SS_TYPE	PLL 扩频调制类型选择 0：选择中心扩频 1：选择向下扩频
29:28	保留	必须保持复位值。
27:13	MODSTEP[14:0]	这些位配置 PLL 扩频调制曲线振幅和频率。必须满足如下条件： MODSTEP*MODCNT≤2^{15}-1
12:0	MODCNT[12:0]	这些位配置 PLL 扩频调制曲线振幅和频率。必须满足如下条件： MODSTEP*MODCNT≤2^{15}-1

5.3.18. 配置寄存器 2（RCU_CFG2）

地址偏移：0XD4
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
31:6	保留	必须保持复位值。
5:4	I2C2SEL[1:0]	I2C2时钟源选择 由软件置1或清0。 00：I2C2时钟源选择APB1时钟 01：I2C2时钟源选择系统时钟 1x：I2C2时钟源选择IRC8M
3:2	保留	必须保持复位值。
1:0	USART5SEL[1:0]	USART5时钟源选择 由软件置1或清0。 00：USART5时钟选择APB2时钟

01：USART5时钟选择系统时钟

10：USART5时钟选择LXTAL时钟

11：USART5时钟选择IRC8M时钟

5.3.19. 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
CAN2RST	保留			CTCRST	保留										
rw				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	CAN2RST	CAN2 复位 由软件置位或复位 0：无作用 1：复位 CAN2
30:28	保留	必须保持复位值。
27	CTCRST	CTC 复位 由软件置位或复位 0：无作用 1：复位 CTC
26:0	保留	必须保持复位值。

5.3.20. 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
CAN2EN	保留			CTCEN	保留										
rw				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	CAN2EN	CAN2 时钟使能 由软件置位或复位 0：关闭 CAN2 时钟 1：开启 CAN2 时钟
30:28	保留	必须保持复位值。
27	CTCEN	CTC 时钟使能 由软件置位或复位 0：关闭 CTC 时钟 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。

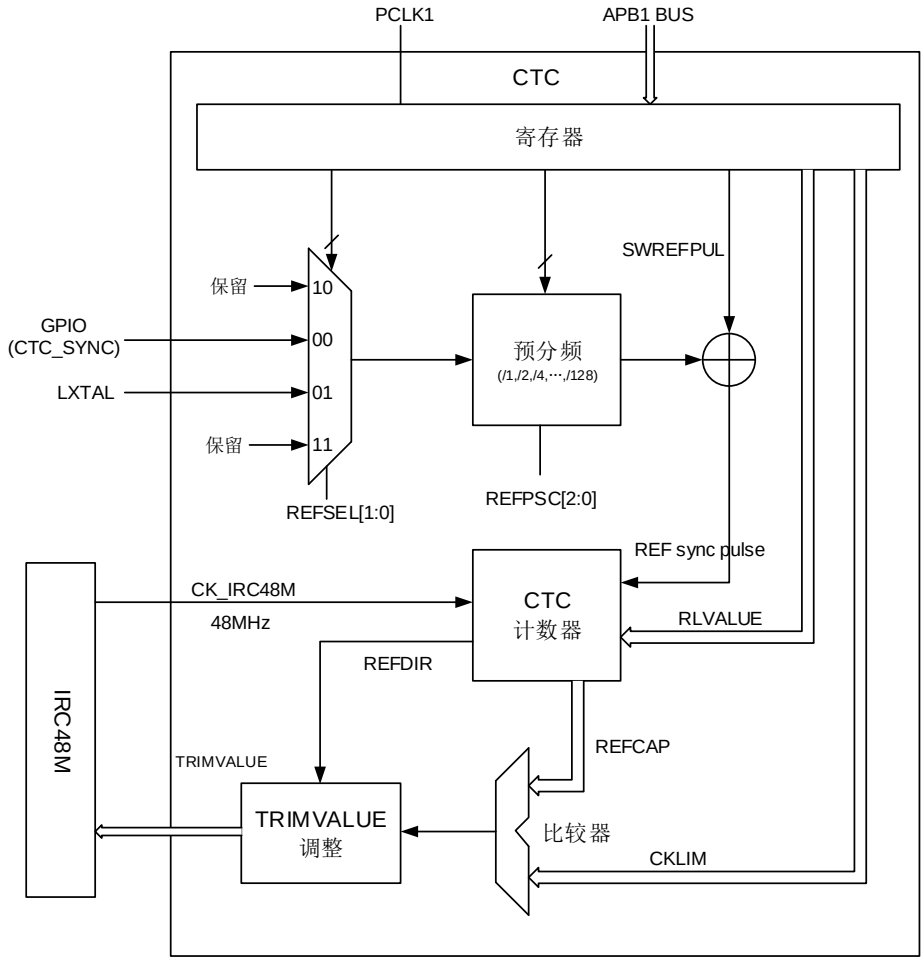
6.2. 主要特性

- 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 bits校准计数器；
- 用于频率评估和自动校准的8 bits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*图 6-1. CTC 简介*。

图 6-1. CTC 简介



6.3.1. REF 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，通过设置 CTC_CTL1 寄存器中的 REFPSC 位来产生一个合适的同步时钟频率信号。

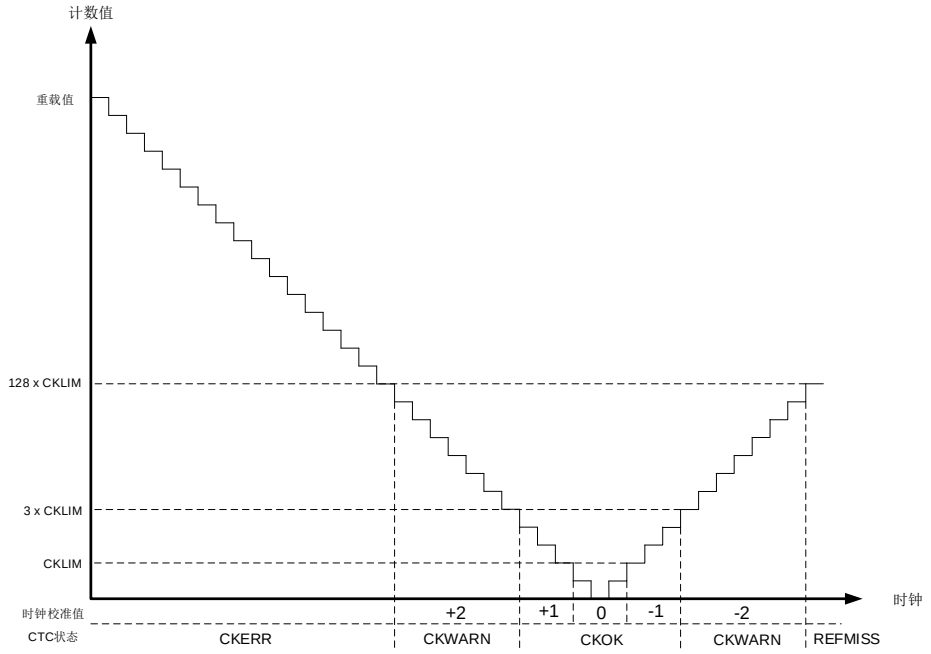
如果需要使用软件参考脉冲信号，则需要设置 CTC_CTL0 寄存器中的 SWREFPUL 位为 1。软件参考脉冲信号与外部参考脉冲信号最后进行逻辑或操作。

6.3.2. CTC 校准计数器

CTC 时钟校准计数器由 CK_IRC48M 提供时钟。在置位 CTC_CTL0 寄存器中的 CNTEN 位后，当检测到第一个 REF 同步脉冲信号，计数器开始从 RLVALUE 值（RLVALUE 在 CTC_CTL1 寄存器中定义）开始向下计数。每次检测到 REF 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 REF 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 REF 同步脉冲信号。一旦检测到 REF 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*图 6-2. CTC 校准计数器*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 REF 同步脉冲信号出现时，时钟频率评估功能开始执行。如果 REF 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 REF 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 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 REF 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 REF 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REF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REF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REF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REF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（REF 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 REF 同步脉冲信号的频率计算得到。理想状态是 REF 同步脉冲信号在 CTC 计数器计数到零时出现，所以 RLVALUE 的值为：

$$RLVALUE=(F_{clock} \div F_{REzF})-1 \quad (6\text{-}1)$$

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：

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

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

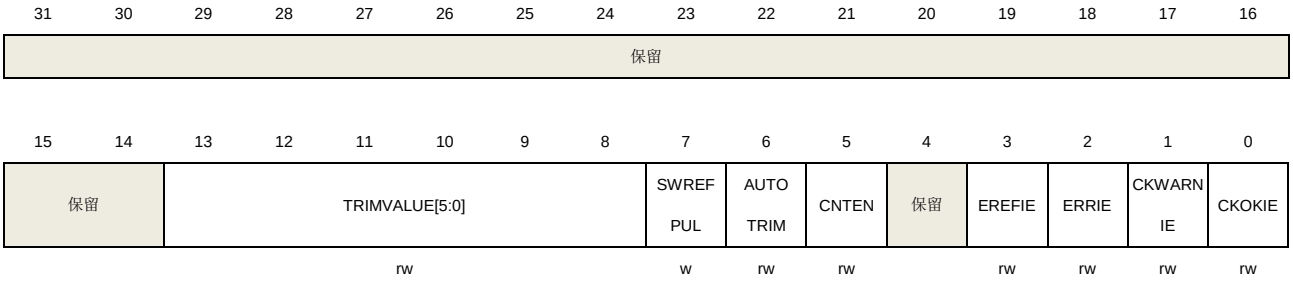
6.4. CTC 寄存器

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。



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 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。 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。 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	软件生成同步参考信号脉冲 该位由软件置位，并为 CTC 计数器提供一个同步参考脉冲信号。该位由硬件自动清除，读操作时返回 0。 0：没有影响 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 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。 0：禁止硬件自动校准模式 1：使能硬件自动校准模式
5	CNTEN	CTC 计数器使能 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。 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 0：禁止期望参考信号产生中断 1：使能期望参考信号产生中断
2	ERRIE	错误中断使能 0：禁止错误中断 1：使能错误中断
1	CKWARNIE	时钟校准警告中断使能 0：禁止时钟校准警告中断 1：使能时钟校准警告中断
0	CKOKIE	时钟校准完成中断使能 0：禁止时钟校准完成中断 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							
15	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 该位由软件置位或清除，用于选择参考信号源的同步极性 0：选择上升沿 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 该位由软件置位或清除，用于选择参考信号源 00：选择 GPIO（CTC_SYNC）输入信号 01：选择 LXTAL 时钟

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 该位由软件置位或清除 000：参考信号不分频 001：参考信号 2 分频 010：参考信号 4 分频 011：参考信号 8 分频 100：参考信号 16 分频 101：参考信号 32 分频 110：参考信号 64 分频 111：参考信号 128 分频
23:16	CKLIM[7:0]	时钟校准时基限值 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 该位由软件置位或清除，用于定义 CTC 计数器的重载值，当检测到一个同步参考脉冲时，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				TRIM ERR	REF MISS	CKERR	保留				EREFIF	ERRIF	CKWARN IF	CKOK IF
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 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 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。

		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 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。 0：无校准值错误发生 1：发生校准值错误
9	REFMISS	同步参考脉冲信号丢失 当同步参考脉冲信号丢失时，该位由硬件置位。当 CTC 校准计数器在增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。 0：无同步参考脉冲信号丢失 1：同步参考脉冲信号丢失
8	CKERR	时钟校准错误位 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。 0：无时钟校准错误发生 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 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。 0：无期望参考信号产生 1：期望参考信号产生
2	ERRIF	错误中断标志位 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。 0：无错误发生 1：发生错误
1	CKWARNIF	时钟校准警告中断标志位 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 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。 0：时钟校准未成功 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 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 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 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 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®-M33 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®-M33 技术参考手册》。

EXTI（中断/事件控制器）包括 22 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、下降沿触发和双边沿触发。EXTI 中的每一个边沿检测电路都可以独立配置和屏蔽。

7.2. 主要特性

- Cortex®-M33 系统异常；
- 多达 88 种可屏蔽的外设中断；
- 4 位中断优先级配置位，可配置 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 22 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. 中断功能描述

Arm Cortex®-M33 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。*表 7-1. Cortex®-M33 中的 NVIC 异常类型*和*表 7-2. 中断向量表*列出了 Cortex®-M33 中的 NVIC 异常类型。

表 7-1. Cortex®-M33 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

异常类型	向量编号	优先级（a）	向量地址	描述
存储器管理	4	可编程设置	0x0000_0010	存储器管理
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

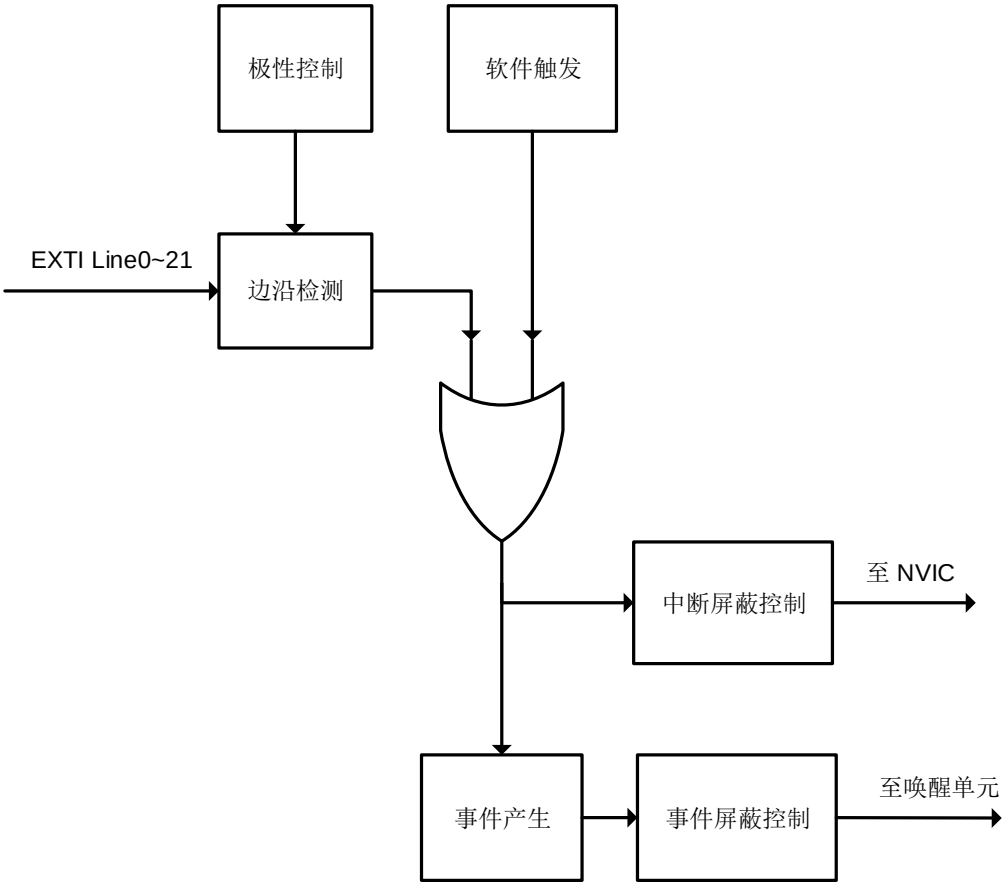
中断编号	向量编号	外设中断描述	向量地址
IRQ 0	16	窗口看门狗中断	0x0000_0040
IRQ 1	17	连接到 EXTI 线的 LVD 中断	0x0000_0044
IRQ 2	18	侵入检测中断	0x0000_0048
IRQ 3	19	RTC 全局中断	0x0000_004C
IRQ 4	20	FMC 全局中断	0x0000_0050
IRQ 5	21	RCU 和 CTC 中断	0x0000_0054
IRQ 6	22	EXTI 线 0 中断	0x0000_0058
IRQ 7	23	EXTI 线 1 中断	0x0000_005C
IRQ 8	24	EXTI 线 2 中断	0x0000_0060
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	35	CAN0 发送中断	0x0000_008C
IRQ 20	36	CAN0 接收 0 中断	0x0000_0090
IRQ 21	37	CAN0 接收 1 中断	0x0000_0094
IRQ 22	38	CAN0 EWMC 中断	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

中断编号	向量编号	外设中断描述	向量地址
IRQ 25	41	TIMER0 更新中断和 TIMER9 全局中断	0x0000_00A4
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 或 I2S1ADD 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 EXTI 线的 RTC 闹钟中断	0x0000_00E4
IRQ 42	58	连接 EXTI 线的 USBHS 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
IRQ50	66	TIMER4 全局中断	0x0000_0108
IRQ51	67	SPI2 或 I2S2ADD 全局中断	0x0000_010C
IRQ52	68	UART3 全局中断	0x0000_0110
IRQ53	69	UART4 全局中断	0x0000_0114
IRQ54	70	TIMER5 或 DAC 全局中断	0x0000_0118
IRQ55	71	TIMER6 全局中断	0x0000_011C
IRQ56	72	DMA1 通道 0 全局中断	0x0000_0120
IRQ57	73	DMA1 通道 1 全局中断	0x0000_0124
IRQ58	74	DMA1 通道 2 全局中断	0x0000_0128
IRQ59	75	DMA1 通道 3 全局中断	0x0000_012C
IRQ60	76	DMA1 通道 4 全局中断	0x0000_0130
IRQ61	77	以太网全局中断	0x0000_0134
IRQ62	78	连接到 EXTI 线的以太网唤醒中断	0x0000_0138
IRQ63	79	CAN1 发送中断	0x0000_013C

中断编号	向量编号	外设中断描述	向量地址
IRQ64	80	CAN1 接收 0 中断	0x0000_0140
IRQ65	81	CAN1 接收 1 中断	0x0000_0144
IRQ66	82	CAN1 EWMC 中断	0x0000_0148
IRQ67	83	USBHS 全局中断	0x0000_014C
IRQ68	84	保留	0x0000_0150
IRQ69	85	SHRTIMER_IRQ2 中断	0x0000_0154
IRQ70	86	SHRTIMER_IRQ3 中断	0x0000_0158
IRQ71	87	SHRTIMER_IRQ4 中断	0x0000_015C
IRQ72	88	SHRTIMER_IRQ5 中断	0x0000_0160
IRQ73	89	SHRTIMER_IRQ6 中断	0x0000_0164
IRQ74	90	USBHS 端点 1 输出中断	0x0000_0168
IRQ75	91	USBHS 端点 1 输入中断	0x0000_016C
IRQ76	92	SHRTIMER_IRQ0 中断	0x0000_0170
IRQ77	93	SHRTIMER_IRQ1 中断	0x0000_0174
IRQ78	94	CAN2 发送中断	0x0000_0178
IRQ79	95	CAN2 接收 0 中断	0x0000_017C
IRQ80	96	CAN2 接收 1 中断	0x0000_0180
IRQ81	97	CAN2 EWMC 中断	0x0000_0184
IRQ82	98	I2C2 事件中断	0x0000_0188
IRQ83	99	I2C 错误中断	0x0000_018C
IRQ84	100	USART5 全局中断	0x0000_0190
IRQ85	101	连接 EXTI 线的 I2C2 唤醒中断	0x0000_0194
IRQ86	102	连接 EXTI 线的 USART5 唤醒中断	0x0000_0198
IRQ87	103	TMU 中断	0x0000_019C

7.4. 外部中断及事件(EXTI)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 22 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 6 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）*。

除了中断，EXTI 还可以向处理器提供事件信号。The Cortex®-M33 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，例如一个特定的 I/O 管脚电平翻转或者 RTC 闹钟动作，EXTI 能唤醒处理器及整个系统。

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；

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 / PF0 / PG0
1	PA1 / PB1 / PC1 / PD1 / PE1 / PF1 / PG1
2	PA2 / PB2 / PC2 / PD2 / PE2 / PF2 / PG2
3	PA3 / PB3 / PC3 / PD3 / PE3 / PF3 / PG3
4	PA4 / PB4 / PC4 / PD4 / PE4 / PF4 / PG4
5	PA5 / PB5 / PC5 / PD5 / PE5 / PF5 / PG5
6	PA6 / PB6 / PC6 / PD6 / PE6 / PF6 / PG6
7	PA7 / PB7 / PC7 / PD7 / PE7 / PF7 / PG7
8	PA8 / PB8 / PC8 / PD8 / PE8 / PF8 / PG8
9	PA9 / PB9 / PC9 / PD9 / PE9 / PF9 / PG9
10	PA10 / PB10 / PC10 / PD10 / PE10 / PF10 / PG10
11	PA11 / PB11 / PC11 / PD11 / PE11 / PF11 / PG11
12	PA12 / PB12 / PC12 / PD12 / PE12 / PF12 / PG12
13	PA13 / PB13 / PC13 / PD13 / PE13 / PF13 / PG13
14	PA14 / PB14 / PC14 / PD14 / PE14 / PF14 / PG14
15	PA15 / PB15 / PC15 / PD15 / PE15 / PF15 / PG15
16	LVD
17	RTC 闹钟
18	USB 唤醒
19	以太网唤醒
20	I2C2 唤醒
21	USART5 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										INTEN21	INTEN20	INTEN19	INTEN18	INTEN17	INTEN16
										rw	rw	rw	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
31:22	保留	必须保持复位值。
21:0	INTENx	中断使能位 x（x = 0…21） 0：第 x 线中断被禁用 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										EVEN21	EVEN20	EVEN19	EVEN18	EVEN17	EVEN16
										rw	rw	rw	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
31:22	保留	必须保持复位值。
21:0	EVENx	事件使能位 x（x = 0…21） 0：第 x 线事件被禁用. 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										RTEN21	RTEN20	RTEN19	RTEN18	RTEN17	RTEN16
										rw	w	rw	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
31:22	保留	必须保持复位值。
21:0	RTENx	上升沿触发使能x（x = 0…21） 0：第x线上升沿触发无效 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										FTEN21	FTEN20	FTEN19	FTEN18	FTEN17	FTEN16
										rw	rw	rw	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
31:22	保留	必须保持复位值。
21:0	FTENx	下降沿触发使能 x（x = 0…21） 0：第 x 线下降沿触发无效 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										SWIEV21	SWIEV20	SWIEV19	SWIEV18	SWIEV17	SWIEV16
										rw	rw	rw	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
31:22	保留	必须保持复位值。
21:0	SWIEVx	中断/事件软件触发 x（x = 0…21） 0：禁用 EXTI 线 x 软件中断/事件请求 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：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
保留										PD21	PD20	PD19	PD18	PD17	PD16
										rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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
31:22	保留	必须保持复位值。
21:0	PDx	中断挂起状态 x（x = 0…21） 0：EXTI 线 x 没有被触发 1：EXTI 线 x 被触发 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 112 个通用 I/O 引脚(GPIO)，分别为 PA0 ~ PA15，PB0 ~ PB15，PC0 ~ PC15，PD0 ~ PD15，PE0 ~ PE15，PF0 ~ PF15 和 PG0 ~ PG15，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。外设 GPIO 引脚上的外部中断在中断/事件控制器（EXIT）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。

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、下拉或浮空。除模拟模式外，所有的 GPIO 引脚都具备大电流驱动能力。

8.2. 主要特性

- 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；
- 端口锁定配置。

8.3. 功能描述

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)和两个 32 位的数据寄存器(GPIOx_ISTAT 和 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，AFIO 推挽输出和 AFIO 开漏输出。详情请见*表 8-1. GPIO 配置表*。

表 8-1. GPIO 配置表

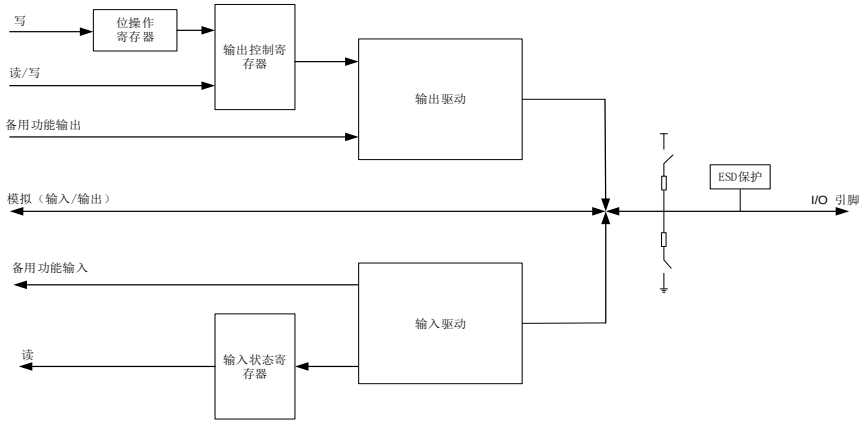
配置模式		CTL[1:0]	SPDy:MD[1:0]	OCTL
输入	模拟	00	x00	不使用
	浮空输入	01		不使用
	下拉输入	10		0
	上拉输入	10		1
配置模式		**CTL[1:0]**	**SPDy: MD[1:0]**	**OCTL**

普通输出 (GPIO)	推挽	00	x00：保留 x01：最大速度到 10MHz x10：最大速度到 2MHz 011：最大速度到 50MHz 111：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



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。

8.3.1. 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。

8.3.2. 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。

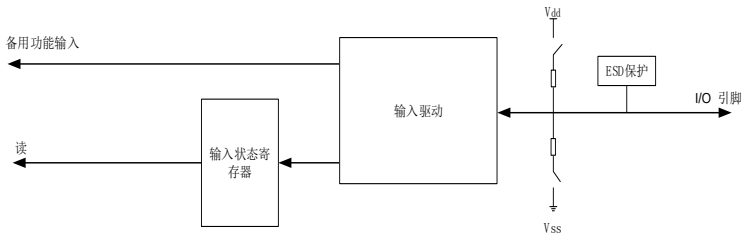
8.3.3. 输入配置

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



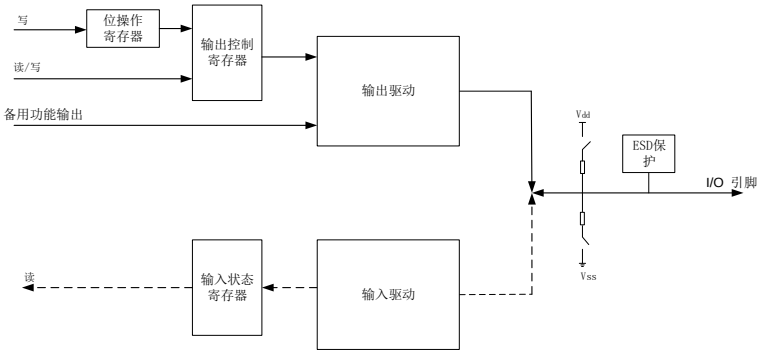
8.3.4. 输出配置

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



8.3.5. 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

8.3.6. 备用功能(AF)配置

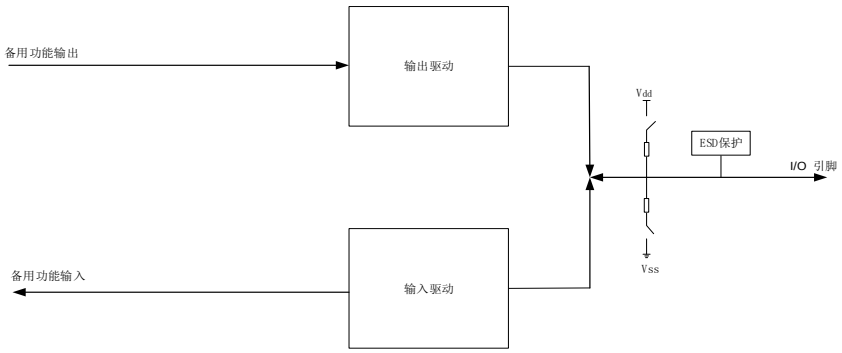
为了适应不同的器件封装，GPIO 端口支持软件配置将一些备用功能应用到其他引脚上。

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



8.3.7. IO 引脚功能选择

GPIO：

通过配置 GPIOx_CTL0 / GPIOx_CTL1 寄存器中的 MDy 位为 0b00，可以将每个 IO 引脚用于 GPIO 输入功能。通过配置 MDy 位为 0b01、0b10 或 0b11 并将 GPIOx_CTL0 / GPIOx_CTL1 寄存器中相应端口的 CTLy 位配置为 0b00（对于 GPIO 推挽输出）或 0b01（对于 GPIO 开漏输出）来设置输出功能。

备用功能：

通过配置 GPIOx_CTL0/GPIOx_CTL1 寄存器中的 MDy 位为 0b00，可以将每个 IO 引脚用于 AF 输入功能。通过配置 MDy 位为 0b01、0b10 或 0b11 并将 GPIOx_CTL0/GPIOx_CTL1 寄存器中相应端口的 CTLy 位配置为 0b10（对于 AF 推挽输出）或 0b11（对于 AF 漏极开路输出）来设置输出功能。

有些备用功能需要通过 AFIO_PCFA-AFIO_PCFE 和 AFIO_PCFG 寄存器进行配置。

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

默认情况下，I/O 补偿单元是不使用的，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。

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。

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. 主要特性

- APB从接口用于寄存器访问；
- 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 SW-DP 开启 （复位状态）	X	X	X	X	X
001	JTAG-DP 开启 SW-DP 开启 没有 NJTRST	X	X	X	X	√
010	JTAG-DP 关闭 SW-DP 开启	X	X	√	√ (1)	√
100	JTAG-DP 关闭 SW-DP 关闭	√	√	√	√	√

其他	禁用					

1. 只有在不使用异步跟踪时，I/O才能使用。
2. “√”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. TIMER AF 重映射

表 8-4. TIMER 备用功能重映射

备用功能	TIMERX_REMAP [1:0](X = 0，1，2)			
	TIMERX_REMAP(X = 8，9，10，12，13)		-	
	“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	PA6	-	PE15
TIMER0_CH0_ON	PB13	PA7	-	PE8
TIMER0_CH1_ON	PB14	PB0	-	PE10
TIMER0_CH2_ON	PB15	PB1	-	PE12
TIMER1_CH0/TIMER1_ETI(1)	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	-	-
TIMER3_CH1	PB7	PD13	-	-

备用功能	TIMERX_REMAP [1:0](X = 0，1，2)			
	TIMERX_REMAP(X = 8，9，10，12，13)		-	
	“0”/“00”（没有映射）	“1”/“01”（部分映射）	“10”（部分映射）	“11”（全映射）
TIMER3_CH2	PB8	PD14	-	-
TIMER3_CH3	PB9	PD15	-	-
TIMER8_CH0	PA2	PE5	-	-
TIMER8_CH1	PA3	PE6	-	-
TIMER9_CH0	PB8	PF6	-	-
TIMER10_CH0	PB9	PF7	-	-
TIMER12_CH0	PA6	PF8	-	-
TIMER13_CH0	PA7	PF9	-	-

1. TIMER1_CH0和TIMER1_ETI 共用一个引脚，但不能同时使用。
2. TIMER8/9/10/12/13 参考备用功能映射和调试I/O配置寄存器1（AFIO_PCF1）。

表 8-5. 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

8.4.5. USART AF 重映射

参考 AFIO 端口配置寄存器 0（AFIO_PCF0）。

表 8-6. USART0/1 备用功能重映射

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

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

8.4.6. I2C0 备用功能重映射

参考 AFIO 端口配置寄存器 0（AFIO_PCF0）。

表 8-7. 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

8.4.7. SPI0/SPI2/I2S 备用功能重映射

参考 AFIO 端口配置寄存器 0（AFIO_PCF0）。

表 8-8. SPI0/SPI2/I2S 备用功能重映射

寄存器	SPI0	SPI2/I2S
SPI0_REMAP = 0	PA4(SPI0_NSS) PA5(SPI0_SCK) PA6(SPI0_MISO) PA7(SPI0_MOSI)	-
SPI0_REMAP = 1	PA15(SPI0_NSS) PB3(SPI0_SCK) PB4(SPI0_MISO) PB5(SPI0_MOSI)	-

寄存器	SPI0	SPI2/I2S
SPI2_REMAP = 0	-	PA15(SPI2_NSS/ I2S2_WS) PB3(SPI2_SCK/ I2S2_CK) PB4(SPI2_MISO) PB5(SPI2_MOSI/I2S2_SD)
SPI2_REMAP = 1	-	PA4(SPI2_NSS/ I2S2_WS) PC10(SPI2_SCK/ I2S2_CK) PC11(SPI2_MISO) PC12(SPI2_MOSI/I2S2_SD)

8.4.8. CAN0/1 备用功能重映射

如下表所示，CAN0 的信号引脚可以映射到端口 A，端口 B 或端口 D。

表 8-9. CAN0/1 备用功能重映射

寄存器 (1)	CAN0	CAN1
CAN0_REMAP[1:0] = “00”	PA11(CAN0_RX) PA12(CAN0_TX)	-
CAN0_REMAPI[1:0] = “10”	PB8(CAN0_RX) PB9(CAN0_TX)	-
CAN0_REMAP[1:0] = “11”	PD0(CAN0_RX) PD1(CAN0_TX)	-
CAN1_REMAP = “0”	-	PB12(CAN1_RX) PB13(CAN1_TX)
CAN1_REMAP = “1”	-	PB5(CAN1_RX) PB6(CAN1_TX)

8.4.9. ENET 备用功能重映射

表 8-10. ENET 备用功能重映射

寄存器	ENET
ENET_REMAP = “0”	PA7(RX_DV-CRS_DV) PC4(RXD0) PC5(RXD1) PB0(RXD2) PB1(RXD3)
ENET_REMAP = “1”	PD8(RX_DV-CRS_DV) PD9(RXD0) PD10(RXD1) PD11(RXD2) PD12(RXD3)

8.4.10. CTC 备用功能重映射

参考 AFIO 端口配置寄存器 1(AFIO_PCF1)。

表 8-11. CTC 备用功能重映射

备用功能	CTC_REMAP [1:0] = “00”	CTC_REMAP [1:0] = “01”	CTC_REMAP [1:0] = “10” or “11”
CTC_SYNC	PA8	PD15	PF0

8.4.11. 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。HXTAL 的优先级比其他普通 IO 功能高。

注意：

- 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。
- 参考 *4.3.1* 电池备份域章节中的 IO 口用法。

表 8-12. 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 OSCIN/OSCOUT 可以用做普通的 I/O 端口 PD0/PD1。

表 8-13. OSC 引脚配置

备用功能	HXTAL= ON	HXTAL= OFF
PD0	OSCIN	PD0
PD1	OSCOUT	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

GPIOF 基地址：0x4001 1C00

GPIOG 基地址：0x4001 2000

AFIO 基地址：0x4001 0000

8.5.1. 端口控制寄存器 0（GPIOx_CTL0，x=A..G）

地址偏移：0x00

复位值：0x4444 4444(x= A, C..G) / 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]	Pin 7 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述。
29:28	MD7[1:0]	Pin 7 模式位 该位由软件置位和清除。 参考 MD0[1:0]的描述。
27:26	CTL6[1:0]	Pin 6 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述。
25:24	MD6[1:0]	Pin 6 模式位 该位由软件置位和清除。 参考 MD0[1:0]的描述。
23:22	CTL5[1:0]	Pin 5 配置位

		该位由软件置位和清除。 参考 CTL0[1:0]的描述。
21:20	MD5[1:0]	Pin 5 模式位 该位由软件置位和清除。 参考 MD0[1:0]的描述。
19:18	CTL4[1:0]	Pin 4 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述。
17:16	MD4[1:0]	Pin 4 模式位 该位由软件置位和清除。 参考 MD0[1:0]的描述。
15:14	CTL3[1:0]	Pin 3 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述。
13:12	MD3[1:0]	Pin 3 模式位 该位由软件置位和清除。 参考 MD0[1:0]的描述。
11:10	CTL2[1:0]	Pin 2 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述。
9:8	MD2[1:0]	Pin 2 模式位 该位由软件置位和清除。 参考 MD0[1:0]的描述。
7:6	CTL1[1:0]	Pin 1 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述。
5:4	MD1[1:0]	Pin 1 模式位 该位由软件置位和清除。 参考 MD0[1:0]的描述。
3:2	CTL0[1:0]	Pin 0 配置位 该位由软件置位和清除。 输入模式（MD[1:0] = 00） 00：模拟输入 01：浮空输入 10：上拉输入/下拉输入 11：保留 输出模式（MD[1:0] > 00） 00：GPIO 推挽输出 01：GPIO 开漏输出

10：AFIO 推挽输出
11：AFIO 开漏输出

1:0	MD0[1:0]	Pin 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..G）

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

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]	Pin 15 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述。
29:28	MD15[1:0]	Pin 15 模式位 该位由软件置位和清除。 参考 MD0[1:0]的描述。
27:26	CTL14[1:0]	Pin 14 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述。
25:24	MD14[1:0]	Pin 14 模式位 该位由软件置位和清除。 参考 MD0[1:0]的描述。
23:22	CTL13[1:0]	Pin13 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述。
21:20	MD13[1:0]	Pin 13 模式位 该位由软件置位和清除。

参考 MD0[1:0]的描述。

19:18	CTL12[1:0]	Pin 12 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述。
17:16	MD12[1:0]	Pin 12 模式位 该位由软件置位和清除。 参考 MD0[1:0]的描述。
15:14	CTL11[1:0]	Pin 11 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述。
13:12	MD11[1:0]	Pin 11 模式位 该位由软件置位和清除。 参考 MD0[1:0]的描述。
11:10	CTL10[1:0]	Pin 10 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述。
9:8	MD10[1:0]	Pin10 模式位 该位由软件置位和清除。 参考 MD0[1:0]的描述。
7:6	CTL9[1:0]	Pin 9 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述。
5:4	MD9[1:0]	Port 9 模式位 该位由软件置位和清除。 参考 MD0[1:0]的描述。
3:2	CTL8[1:0]	Pin8 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述。
1:0	MD8[1:0]	Pin 8 模式位 该位由软件置位和清除。 参考 MD0[1:0]的描述。

8.5.3. 端口输入状态寄存器（GPIOx_ISTAT，x=A..G）

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） 这些位由硬件置位和清除。 0：引脚输入信号为低电平 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..G）

地址偏移：0x0C

复位值: 0x0000 A000（GPIOA_OCTL）
0x0000 0010（GPIOB_OCTL）
0x0000 0000（GPIOx_OCTL, x=C..G）

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） 这些位由软件置位和清除。 0：引脚输出低电平 1：引脚输出高电平

8.5.5. 端口位操作寄存器（GPIOx_BOP，x=A..G）

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

8.5.6. 位清除寄存器（GPIOx_BC，x=A..G）

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） 这些位由软件置位和清除。 0：相应 OCTLy 位没有改变 1：清除相应的 OCTLy 位

8.5.7. 端口配置锁定寄存器（GPIOx_LOCK，x=A..G）

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 该位只能通过使用 Lock Key 写序列设置，始终可读。 0：GPIO_LOCK 寄存器和端口配置没有锁定 1：直到下一次 MCU 复位前，GPIO_LOCK 寄存器被锁定 LOCK Key 写序列： 写 1→写 0→写 1→读 0→读 1 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） 这些位由软件置位和清除。 0：相应的端口位配置没有锁定 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..G）

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

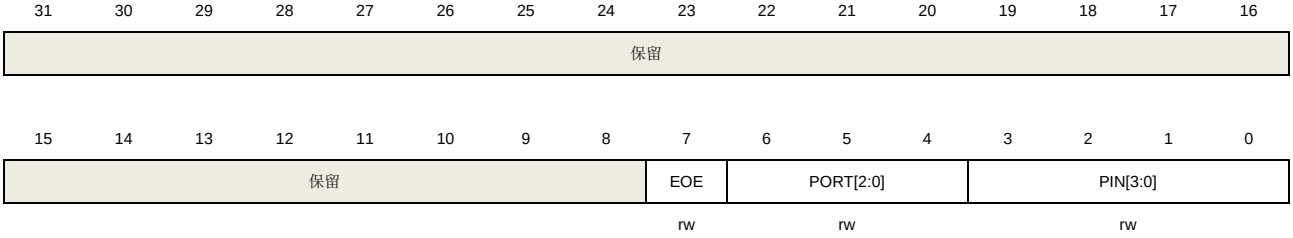
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	SPDy	当 MDx 值为 0b11 时，设置相应端口速度为高速（120MHz）。 如果端口输出速度大于 50MHz，该位置 1，同时设置 MDx 值为 0b11。这些位由软件置位和清除。 0：没有影响 1：最大输出速度大于 50MHz（同时，需要设置 MDx 值为 0b11） 注意：当端口输出速度大于 50MHz 时，需要使能 I/O 补偿单元。详见 AFIO_CPSCTL

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。

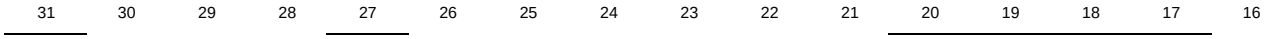


位/位域	名称	描述
31:8	保留	必须保持复位值。
7	EOE	事件输出使能 该位由软件置位和清除。当设置该位后，Cortex®的 EVENTOUT 输出将连接到由 PORT[2:0]和 PIN[3:0]位选择的 I/O 口。
6:4	PORT[2:0]	事件输出端口选择 这些位由软件置位和清除。选择用于输出 Cortex®的 EVENTOUT 信号的端口。 000：选择端口 A 001：选择端口 B 010：选择端口 C 011：选择端口 D 100：选择端口 E
3:0	PIN[3:0]	事件输出引脚选择 这些位由软件置位和清除。选择用于输出 Cortex®的 EVENTOUT 信号的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



保留	PTP_PPS_REMAP	TIMER1ITI1_REMAP	SPI2_REMAP	保留	SWJ_CFG[2:0]	ENET_PHY_SEL	CAN1_REMAP	ENET_REMAP	保留	TIMER4CH3_IREMAP
	rw	rw	rw		w	rw	rw	rw		rw

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

PD01_REMAP	CAN0_REMAP[1:0]	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	rw

位/位域	名称	描述
31	保留	必须保持复位值。
30	PTP_PPS_REMAP	以太网 PTP PPS 重映射 该位由软件置位和清除，使用以太网 MAC_PPS 输出到 PB5 引脚。 0：PPT_PPS 没有输出到 PB5 引脚 1：PPT_PPS 输出到 PB5 引脚
29	TIMER1ITI1_REMAP	TIMER1 内部触发 1 重映射 该位由软件置位和清除，用于控制 TIMER1_ITI1 内部重映射。 0：TIMER1_ITI1 内部连接到以太网 PTP 输出，用于校准。 1：TIMER1_ITI1 内部连接到 USB OTG SOF（起始帧）输出，用于校准。
28	SPI2_REMAP	SPI2/I2S2 重映射 该位由软件置位和清除。 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） 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 这些位只写（读这些位，将返回未定义值）。 000：JTAG-DP 使能和 SW-DP 使能：复位状态 001：JTAG-DP 使能和 SW-DP 使能：没有 NJTRST 010：JTAG-DP 禁用和 SW-DP 使能 100：JTAG-DP 禁用和 SW-DP 禁用 其他组合：无作用
23	ENET_PHY_SEL	以太网 MII 或 RMII PHY 选择 该位由软件置位和清除，它配置以太网内部 MAC 使用外部 MII 或 RMII PHY。 0：配置以太网 MAC 使用外部 MII PHY 1：配置以太网 MAC 使用外部 RMII PHY
22	CAN1_REMAP	CAN1 I/O 重映射 该位由软件置位和清除，控制着 CAN1_TX 和 CAN1_RX 引脚。 0：关闭重映射功能（CAN1_RX/PB12，CAN_TX/PB13）

		1：开启重映射功能（CAN1_RX/PB5，CAN_TX/PB6）
21	ENET_REMAP	以太网 MAC I/O 重映射 该位由软件置位和清除，控制着以太网 MAC 连接到 PHY。 0：关闭重映射功能（RX_DV-CRS_DV/PA7，RXD0/PC4，RXD1/PC5，RXD2/PB0，RXD3/PB1） 1：开启重映射功能（RX_DV-CRS_DV/PD8，RXD0/PD9，RXD1/PD10，RXD2/PD11，RXD3/PD12）
20:17	保留	必须保持复位值。
16	TIMER4CH3_IREMAP	TIMER4 通道 3 内部重映射 该位由软件置位和清除。 0：连接 TIMER4_CH3 与 PA3 1：连接 IRC40K 内部振荡器与 TIMER4_CH3，用于对 IRC40K 校准
15	PD01_REMAP	OSC_IN/OSC_OUT 重映射到 Port D0/Port D1 该位由软件置位和清除。 0：关闭重映射功能 1：OSC_IN 重映射到 PD0，OSC_OUT 重映射到 PD1
14:13	CAN0_REMAP[1:0]	CAN0 备用功能接口重映射 这些位由软件置位和清除。 00：关闭重映射功能（CAN0_RX/PA11，CAN0_TX/PA12） 01：没有使用 10：开启重映射部分功能（CAN0_RX/PB8，CAN0_TX/PB9） 11：完全开启重映射功能（CAN0_RX/PD0，CAN0_TX/PD1）
12	TIMER3_REMAP	TIMER3 重映射 该位由软件置位和清除。 0：关闭重映射功能（TIMER3_CH0/PB6，TIMER3_CH1/PB7，TIMER3_CH2/PB8，TIMER3_CH3/PB9） 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 这些位由软件置位和清除。 00：关闭重映射功能（TIMER2_CH0/PA6，TIMER2_CH1/PA7，TIMER2_CH2/PB0，TIMER2_CH3/PB1） 01：没有使用 10：开启重映射部分功能（TIMER2_CH0/PB4，TIMER2_CH1/PB5，TIMER2_CH2/PB0，TIMER2_CH3/PB1） 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 这些位由软件置位和清除。 00：关闭重映射功能（TIMER1_CH0-TIMER1_ETI/PA0，TIMER1_CH1/PA1，

		TIMER1_CH2/PA2，TIMER1_CH3/PA3） 01：开启重映射部分功能 0（TIMER1_CH0-TIMER1_ETI/PA15，TIMER1_CH1/PB3，TIMER1_CH2/PA2，TIMER1_CH3/PA3） 10：开启重映射部分功能 1（TIMER1_CH0-TIMER1_ETI/PA0，TIMER1_CH1/PA1，TIMER1_CH2/PB10，TIMER1_CH3/PB11） 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 这些位由软件置位和清除。 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） 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） 10：没有使用 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 这些位由软件置位和清除。 00：关闭重映射功能（USART2_TX/PB10，USART2_RX /PB11，USART2_CK/PB12，USART2_CTS/PB13，USART2_RTS/PB14） 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） 10：没有使用 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 该位由软件置位和清除。 0：关闭重映射功能（USART1_CTS/PA0，USART1_RTS/PA1，USART1_TX/PA2，USART1_RX /PA3，USART1_CK/PA4） 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 该位由软件置位和清除。 0：关闭重映射功能（USART0_TX/PA9，USART0_RX /PA10） 1：开启重映射功能（USART0_TX/PB6，USART0_RX /PB7）
1	I2C0_REMAP	I2C0 重映射 该位由软件置位和清除。

0：关闭重映射功能（I2C0_SCL/PB6，I2C0_SDA /PB7）
1：开启重映射功能（I2C0_SCL/PB8，I2C0_SDA /PB9）

0	SPI0_REMAP	SPI0 重映射 该位由软件置位和清除。 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） 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 0000：PA3 引脚 0001：PB3 引脚 0010：PC3 引脚 0011：PD3 引脚 0100：PE3 引脚 0101：PF3 引脚 0110：PG3 引脚 其他配置保留。
11:8	EXTI2_SS[3:0]	EXTI 2 源选择 0000：PA2 引脚 0001：PB2 引脚 0010：PC2 引脚 0011：PD2 引脚 0100：PE2 引脚 0101：PF2 引脚 0110：PG2 引脚 其他配置保留。

7:4	EXTI1_SS[3:0]	EXTI 1 源选择 0000：PA1 引脚 0001：PB1 引脚 0010：PC1 引脚 0011：PD1 引脚 0100：PE1 引脚 0101：PF1 引脚 0110：PG1 引脚 其他配置保留。
3:0	EXTI0_SS[3:0]	EXTI 0 源选择 0000：PA0 引脚 0001：PB0 引脚 0010：PC0 引脚 0011：PD0 引脚 0100：PE0 引脚 0101：PF0 引脚 0110：PG0 引脚 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 0000：PA7 引脚 0001：PB7 引脚 0010：PC7 引脚 0011：PD7 引脚 0100：PE7 引脚 0101：PF7 引脚 0110：PG7 引脚

其他配置保留。

11:8	EXTI6_SS[3:0]	EXTI 6 源选择 0000：PA6 引脚 0001：PB6 引脚 0010：PC6 引脚 0011：PD6 引脚 0100：PE6 引脚 0101：PF6 引脚 0110：PG6 引脚 其他配置保留。
7:4	EXTI5_SS[3:0]	EXTI 5 源选择 0000：PA5 引脚 0001：PB5 引脚 0010：PC5 引脚 0011：PD5 引脚 0100：PE5 引脚 0101：PF5 引脚 0110：PG5 引脚 其他配置保留。
3:0	EXTI4_SS[3:0]	EXTI 4 源选择 0000：PA4 引脚 0001：PB4 引脚 0010：PC4 引脚 0011：PD4 引脚 0100：PE4 引脚 0101：PF4 引脚 0110：PG4 引脚 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。

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
EXTI11_SS [3:0]				EXTI10_SS [3:0]				EXTI9_SS [3:0]				EXTI8_SS [3:0]			
rw				rw				rw				rw			

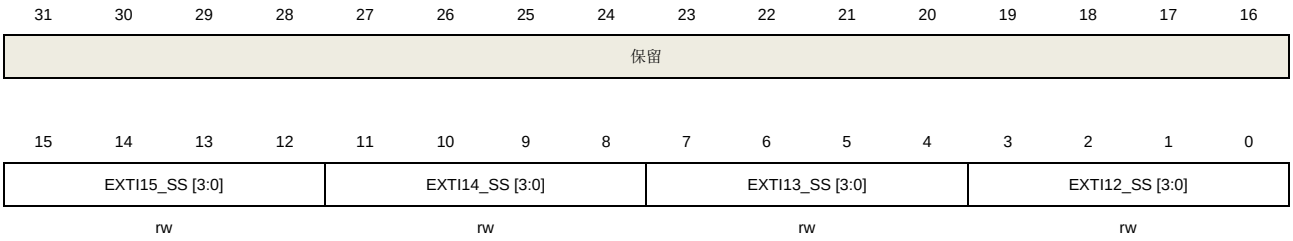
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI11_SS[3:0]	EXTI 11 源选择 0000：PA11 引脚 0001：PB11 引脚 0010：PC11 引脚 0011：PD11 引脚 0100：PE11 引脚 0101：PF11 引脚 0110：PG11 引脚 其他配置保留。
11:8	EXTI10_SS[3:0]	EXTI 10 源选择 0000：PA10 引脚 0001：PB10 引脚 0010：PC10 引脚 0011：PD10 引脚 0100：PE10 引脚 0101：PF10 引脚 0110：PG10 引脚 其他配置保留。
7:4	EXTI9_SS[3:0]	EXTI 9 源选择 0000：PA9 引脚 0001：PB9 引脚 0010：PC9 引脚 0011：PD9 引脚 0100：PE9 引脚 0101：PF9 引脚 0110：PG9 引脚 其他配置保留。
3:0	EXTI8_SS[3:0]	EXTI 8 源选择 0000：PA8 引脚 0001：PB8 引脚 0010：PC8 引脚 0011：PD8 引脚 0100：PE8 引脚 0101：PF8 引脚 0110：PG8 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 0000：PA15 引脚 0001：PB15 引脚 0010：PC15 引脚 0011：PD15 引脚 0100：PE15 引脚 0101：PF15 引脚 0110：PG15 引脚 其他配置保留。
11:8	EXTI14_SS[3:0]	EXTI 14 源选择 0000：PA14 引脚 0001：PB14 引脚 0010：PC14 引脚 0011：PD14 引脚 0100：PE14 引脚 0101：PF14 引脚 0110：PG14 引脚 其他配置保留。
7:4	EXTI13_SS[3:0]	EXTI 13 源选择 0000：PA13 引脚 0001：PB13 引脚 0010：PC13 引脚 0011：PD13 引脚 0100：PE13 引脚 0101：PF13 引脚 0110：PG13 引脚 其他配置保留。
3:0	EXTI12_SS[3:0]	EXTI 12 源选择 0000：PA12 引脚 0001：PB12 引脚

0010：PC12 引脚
0011：PD12 引脚
0100：PE12 引脚
0101：PF12 引脚
0110：PG12 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。

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			CTC_REMAP[1:0]		EXMC_NADV	TIMER13_REMAP	TIMER12_REMAP	TIMER10_REMAP	TIMER9_REMAP	TIMER8_REMAP	保留				
			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。
12:11	CTC_REMAP[1:0]	CTC 重映射 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。 00：关闭重映射功能（PA8） 01：开启重映射功能 0（PD15） 10/11：开启重映射功能 1（PF0）
10	EXMC_NADV	EXMC_NADV 连接/不连接 该位由软件置位和清除，控制着可选的 EXMC_NADV 信号 0：NADV 信号连接到输出（默认值） 1：NADV 信号没有连接，I/O 引脚可以用于其他外设。
9	TIMER13_REMAP	TIMER13 重映射 该位由软件置位和清除，控制着将 TIMER13_CH0 备用功能重映射到 GPIO 端口。 0：关闭重映射功能（PA7） 1：开启重映射功能（PF9）
8	TIMER12_REMAP	TIMER12 重映射 该位由软件置位和清除，控制着将 TIMER12_CH0 备用功能重映射到 GPIO 端口。 0：关闭重映射功能（PA6） 1：开启重映射功能（PF8）
7	TIMER10_REMAP	TIMER10 重映射

		该位由软件置位和清除，控制着将 TIMER10_CH0 备用功能重映射到 GPIO 端口。 0：关闭重映射功能（PB9） 1：开启重映射功能（PF7）
6	TIMER9_REMAP	TIMER9 重映射 该位由软件置位和清除，控制着将 TIMER9_CH0 备用功能重映射到 GPIO 端口。 0：关闭重映射功能（PB8） 1：开启重映射功能（PF6）
5	TIMER8_REMAP	TIMER8 重映射 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。 0：关闭重映射功能（TIMER8_CH0 连接到 PA2 和 TIMER8_CH1 连接到 PA3） 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。

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							CPS_RDY	保留							CPS_EN
							r								rw

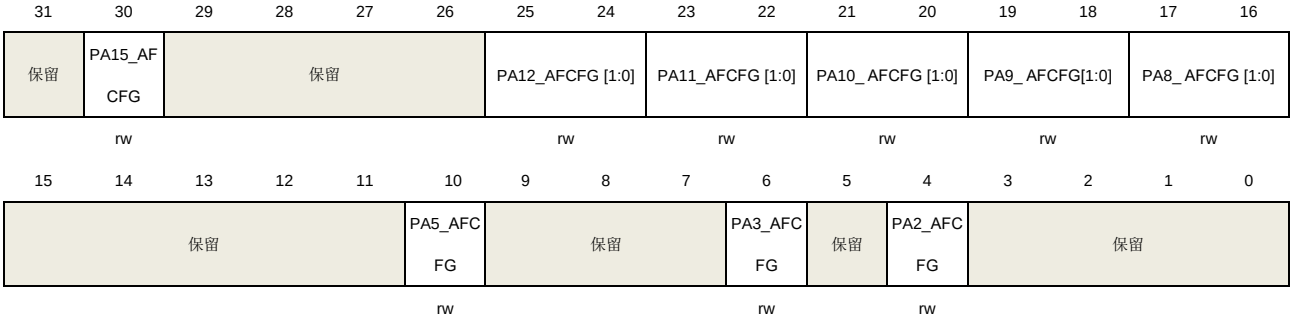
位/位域	名称	描述
31:9	保留	必须保持复位值。
8	CPS_RDY	I/O 补偿单元是否准备好，该位只读。 0：I/O 补偿单元没有准备好 1：I/O 补偿单元准备好
7:1	保留	必须保持复位值。
0	CPS_EN	I/O 补偿单元使能 当端口输出速度大于 50MHz 时，需要使能 I/O 补偿单元。 0：I/O 补偿单元掉电 1：I/O 补偿单元使能

8.5.17. AFIO 端口配置寄存器 A（AFIO_PCFA）

地址偏移：0x3C

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	PA15_AFCFG	PA15 AF 功能配置位 该位由软件置位和清除。 0：不配置 PA15 备用功能到 SHRTIMER 1：配置 PA15 备用功能到 SHRTIMER
29:26	保留	必须保持复位值。
25:24	PA12_AFCFG[1:0]	PA12 AF 功能配置位 这些位由软件置位和清除。 00：不配置 PA12 备用功能到 SHRTIMER/ CMP1/USART5 01：配置 PA12 备用功能到 CMP1 10：配置 PA12 备用功能到 USART5 11：配置 PA12 备用功能到 SHRTIMER
23:22	PA11_AFCFG[1:0]	PA11 AF 功能配置位 这些位由软件置位和清除。 00：不配置 PA11 备用功能到 SHRTIMER/USART5 01：配置 PA11 备用功能到 USART5 10/11：配置 PA11 备用功能到 SHRTIMER
21:20	PA10_AFCFG[1:0]	PA10 AF 功能配置位 这些位由软件置位和清除。 00：不配置 PA10 备用功能到 SHRTIMER/CAN2/CMP5 01：配置 PA10 备用功能到 CAN2 10：配置 PA10 备用功能到 CMP5 11：配置 PA10 备用功能到 SHRTIMER
19:18	PA9_AFCFG[1:0]	PA9 AF 功能配置位 这些位由软件置位和清除。

		00：不配置 PA9 备用功能到 SHRTIMER/CAN2/I2C2 01：配置 PA9 备用功能到 CAN2 10：配置 PA9 备用功能到 I2C2 11：配置 PA9 备用功能到 SHRTIMER
17:16	PA8_AFCFG[1:0]	PA8 AF 功能配置位 这些位由软件置位和清除。 00：不配置 PA8 备用功能到 SHRTIMER/I2C2 01：配置 PA8 备用功能到 I2C2 10/11：配置 PA8 备用功能到 SHRTIMER
15:11	保留	必须保持复位值。
10	PA5_AFCFG	PA5 AF 功能配置位 该位由软件置位和清除。 0：不配置 PA5 备用功能到 USBHS 1：配置 PA5 备用功能到 USBHS
9:7	保留	必须保持复位值。
6	PA3_AFCFG	PA3 AF 功能配置位 该位由软件置位和清除。 0：不配置 PA3 备用功能到 USBHS 1：配置 PA3 备用功能到 USBHS
5	保留	必须保持复位值。
4	PA2_AFCFG	PA2 AF 功能配置位 该位由软件置位和清除。 0：不配置 PA2 备用功能到 CMP1 1：配置 PA2 备用功能到 CMP1
3:0	保留	必须保持复位值。

8.5.18. AFIO 端口配置寄存器 B（AFIO_PCFB）

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	PB15_AFCFG	PB14_AFCFG[1:0]		PB13_AFCFG[1:0]		PB12_AFCFG[1:0]		PB11_AFCFG[1:0]		PB10_AFCFG[1:0]		PB9_AFCFG[1:0]		PB8_AFCFG[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
保留	PB7_AFCFG	保留	PB6_AFCFG	PB5_AFCFG[1:0]		PB4_AFCFG[1:0]		保留	PB3_AFCFG	PB2_AFCFG[1:0]		PB1_AFCFG[1:0]		保留	PB0_AFCFG
	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	PB15_AFCFG	PB15 AF 功能配置位 该位由软件置位和清除。 0：不配置 PB15 备用功能到 SHRTIMER 1：配置 PB15 备用功能到 SHRTIMER
29:28	PB14_AFCFG[1:0]	PB14 AF 功能配置位 这些位由软件置位和清除。 00：不配置 PB14 备用功能到 SHRTIMER/I2S1 01：配置 PB14 备用功能到 I2S1 10/11：配置 PB14 备用功能到 SHRTIMER
27:26	PB13_AFCFG[1:0]	PB13 AF 功能配置位 这些位由软件置位和清除。 00：不配置 PB13 备用功能到 SHRTIMER/USBHS 01/11：配置 PB13 备用功能到 SHRTIMER 10：配置 PB13 备用功能到 USBHS
25:24	PB12_AFCFG[1:0]	PB12 AF 功能配置位 这些位由软件置位和清除。 00：不配置 PB12 备用功能到 SHRTIMER/USBHS 01/11：配置 B12 备用功能到 SHRTIMER 10：配置 PB12 备用功能到 USBHS
23:22	PB11_AFCFG[1:0]	PB11 AF 功能配置位 这些位由软件置位和清除。 00：不配置 PB11 备用功能到 SHRTIMER/USBHS/CAN2 01：配置 PB11 备用功能到 CAN2 10：配置 PB11 备用功能到 USBHS 11：配置 PB11 备用功能到 SHRTIMER
21:20	PB10_AFCFG[1:0]	PB10 AF 功能配置位 这些位由软件置位和清除。 00：不配置 PB10 备用功能到 SHRTIMER/USBHS/CAN2 01：配置 PB10 备用功能到 CAN2 10：配置 PB10 备用功能到 USBHS 11：配置 PB10 备用功能到 SHRTIMER
19:18	PB9_AFCFG[1:0]	PB9 AF 功能配置位 这些位由软件置位和清除。 00：不配置 PB9 备用功能到 SHRTIMER/CMP1 01：配置 PB9 备用功能到 CMP1 10/11：配置 PB9 备用功能到 SHRTIMER
17:16	PB8_AFCFG[1:0]	PB8 AF 功能配置位

		这些位由软件置位和清除。 00：不配置 PB8 备用功能到 SHRTIMER/I2C2 01：配置 PB8 备用功能到 I2C2 10/11：配置 B8 备用功能到 SHRTIMER
15	保留	必须保持复位值。
14	PB7_AFCFG	PB7 AF 功能配置位 该位由软件置位和清除。 0：不配置 PB7 备用功能到 SHRTIMER 1：配置 PB7 备用功能到 SHRTIMER
13	保留	必须保持复位值。
12	PB6_AFCFG	PB6 AF 功能配置位 该位由软件置位和清除。 0：不配置 PB6 备用功能到 SHRTIMER 1：配置 PB6 备用功能到 SHRTIMER
11:10	PB5_AFCFG[1:0]	PB5 AF 功能配置位 这些位由软件置位和清除。 00：不配置 PB5 备用功能到 SHRTIMER/USBHS/I2C2 01：配置 PB5 备用功能到 I2C2 10：配置 PB5 备用功能到 USBHS 11：配置 PB5 备用功能到 SHRTIMER
9:8	PB4_AFCFG[1:0]	PB4 AF 功能配置位 这些位由软件置位和清除。 00：不配置 PB4 备用功能到 SHRTIMER/I2C2/I2S2 01：配置 PB4 备用功能到 I2S2 10：配置 PB4 备用功能到 I2C2 11：配置 PB4 备用功能到 SHRTIMER
7	保留	必须保持复位值。
6	PB3_AFCFG	PB3 AF 功能配置位 该位由软件置位和清除。 0：不配置 PB3 备用功能到 SHRTIMER 1：配置 PB3 备用功能到 SHRTIMER
5:4	PB2_AFCFG[1:0]	PB2 AF 功能配置位 这些位由软件置位和清除。 00：不配置 PB2 备用功能到 SHRTIMER/USBHS 10：配置 PB2 备用功能到 USBHS 01/11：配置 PB2 备用功能到 SHRTIMER
3:2	PB1_AFCFG[1:0]	PB1 AF 功能配置位 这些位由软件置位和清除。 00：不配置 PB1 备用功能到 SHRTIMER/USBHS/CMP3

		01：配置 PB1 备用功能到 CMP3 10：配置 PB1 备用功能到 USBHS 11：配置 PB1 备用功能到 SHRTIMER
1	保留	必须保持复位值。
0	PB0_AFCFG	PB0 AF 功能配置位 该位由软件置位和清除。 0：不配置 PB0 备用功能到 USBHS 1：配置 PB0 备用功能到 USBHS

8.5.19. AFIO 端口配置寄存器 C（AFIO_PCFC）

地址偏移：0x44
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							PC12_AFCFG	PC11_AFCFG[1:0]		保留	PC10_AFCFG	PC9_AFCFG[1:0]		PC8_AFCFG[1:0]	
							rw	rw			rw	rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PC7_AFCFG[1:0]		PC6_AFCFG[1:0]		保留					PC3_AFCFG	PC2_AFCFG[1:0]		保留			PC0_AFCFG
rw		rw							rw	rw					rw

位/位域	名称	描述
31:25	保留	必须保持复位值。
24	PC12_AFCFG	PC12 AF 功能配置位 该位由软件置位和清除。 0：不配置 PC12 备用功能到 SHRTIMER 1：配置 PC12 备用功能到 SHRTIMER
23:22	PC11_AFCFG[1:0]	PC11 AF 功能配置位 这些位由软件置位和清除。 00：不配置 PC11 备用功能到 SHRTIMER/I2S2 01/11：配置 PC11 备用功能到 SHRTIMER 10：配置 PC11 备用功能到 I2S2
21	保留	必须保持复位值。
20	PC10_AFCFG	PC10 AF 功能配置位 该位由软件置位和清除。 0：不配置 PC10 备用功能到 I2C2 1：配置 PC10 备用功能到 I2C2

19:18	PC9_AFCFG[1:0]	PC9 AF 功能配置位 这些位由软件置位和清除。 00：不配置 PC9 备用功能到 SHRTIMER/I2C2 01/11：配置 PC9 备用功能到 SHRTIMER 10：配置 PC9 备用功能到 I2C2
17:16	PC8_AFCFG[1:0]	PC8 AF 功能配置位 这些位由软件置位和清除。 00：不配置 PC8 备用功能到 SHRTIMER/ USART5 10：配置 PC8 备用功能到 USART5 01/11：配置 PC8 备用功能到 SHRTIMER
15:14	PC7_AFCFG[1:0]	PC7 AF 功能配置位 这些位由软件置位和清除。 00：不配置 PC7 备用功能到 SHRTIMER/ USART5 01/11：配置 PC7 备用功能到 SHRTIMER 10：配置 PC7 备用功能到 USART5
13:12	PC6_AFCFG[1:0]	PC6 AF 功能配置位 这些位由软件置位和清除。 00：不配置 PC6 备用功能到 SHRTIMER/CMP5/USART5 01：配置 PC6 备用功能到 CMP5 10：配置 PC6 备用功能到 USART5 11：配置 PC6 备用功能到 SHRTIMER
11:7	保留	必须保持复位值。
6	PC3_AFCFG	PC3 AF 功能配置位 该位由软件置位和清除。 0：不配置 PC3 备用功能到 USBHS 1：配置 PC3 备用功能到 USBHS
5:4	PC2_AFCFG[1:0]	PC2 AF 功能配置位 这些位由软件置位和清除。 00：不配置 PC2 备用功能到 USBHS/I2S1 01/11：配置 PC2 备用功能到 I2S1 10：配置 PC2 备用功能到 USBHS
3:1	保留	必须保持复位值。
0	PC0_AFCFG	PC0 AF 功能配置位 该位由软件置位和清除。 0：不配置 PC0 备用功能到 USBHS 1：配置 PC0 备用功能到 USBHS

8.5.20. AFIO 端口配置寄存器 D（AFIO_PCFD）

地址偏移：0x48

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

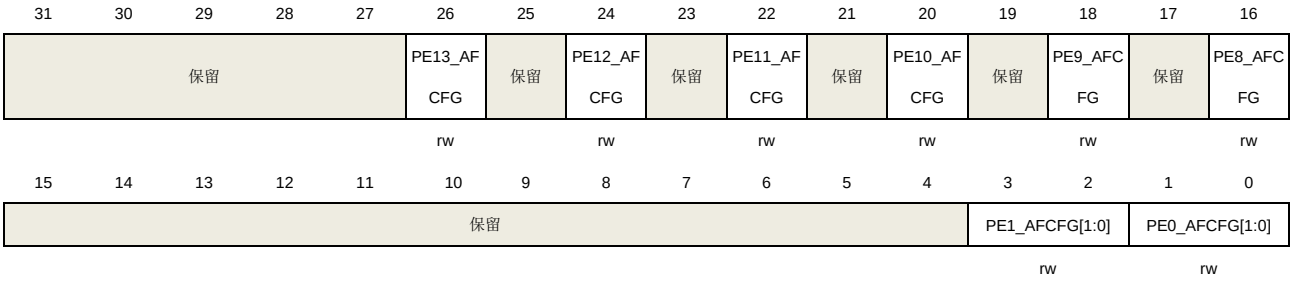
位/位域	名称	描述
31:11	保留	必须保持复位值。
10	PD5_AFCFG	PD5 AF 功能配置位 该位由软件置位和清除。 0：不配置 PD5 备用功能到 SHRTIMER 1：配置 PD5 备用功能到 SHRTIMER
9	保留	必须保持复位值。
8	PD4_AFCFG	PD4 AF 功能配置位 该位由软件置位和清除。 0：不配置 PD4 备用功能到 SHRTIMER 1：配置 PD4 备用功能到 SHRTIMER
7:0	保留	必须保持复位值。

8.5.21. AFIO 端口配置寄存器 E（AFIO_PCFE）

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
31:27	保留	必须保持复位值。
26	PE13_AFCFG	PE13 AF 功能配置位

		该位由软件置位和清除。 0：不配置 PE13 备用功能到 CMP1 1：配置 PE13 备用功能到 CMP1
25	保留	必须保持复位值。
24	PE12_AFCFG	PE12 AF 功能配置位 该位由软件置位和清除。 0：不配置 PE12 备用功能到 CMP3 1：配置 PE12 备用功能到 CMP3
23	保留	必须保持复位值。
22	PE11_AFCFG	PE11 AF 功能配置位 该位由软件置位和清除。 0：不配置 PE11 备用功能到 CMP5 1：配置 PE11 备用功能到 CMP5
21	保留	必须保持复位值。
20	PE10_AFCFG	PE10 AF 功能配置位 该位由软件置位和清除。 0：不配置 PE10 备用功能到 CMP5 1：配置 PE10 备用功能到 CMP5
19	保留	必须保持复位值。
18	PE9_AFCFG	PE9 AF 功能配置位 该位由软件置位和清除。 0：不配置 PE9 备用功能到 CMP3 1：配置 PE9 备用功能到 CMP3
17	保留	必须保持复位值。
16	PE8_AFCFG	PE8 AF 功能配置位 该位由软件置位和清除。 0：不配置 PE8 备用功能到 CMP1 1：配置 PE8 备用功能到 CMP1
15:4	保留	必须保持复位值。
3:2	PE1_AFCFG[1:0]	PE1 AF 功能配置位 这些位由软件置位和清除。 00：不配置 PE1 备用功能到 SHRTIMER/CAN2 01：配置 PE1 备用功能到 CAN2 10/11：配置 PE1 备用功能到 SHRTIMER
1:0	PE0_AFCFG[1:0]	PE0 AF 功能配置位 这些位由软件置位和清除。 00：不配置 PE0 备用功能到 SHRTIMER/CAN2 01：配置 PE0 备用功能到 CAN2

10/11：配置 PE0 备用功能到 SHRTIMER

8.5.22. AFIO 端口配置寄存器 G（AFIO_PCFG）

地址偏移：0x54

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			PG14_AFCFG	保留	PG13_AFCFG	保留	PG12_AFCFG	保留	PG11_AFCFG	保留	PG10_AFCFG	保留	PG9_AFCFG	保留	
			rw		rw		rw		rw		rw		rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PG7_AFCFG[1:0]		保留	PG6_AFCFG	保留											
rw			rw												

位/位域	名称	描述
31:29	保留	必须保持复位值。
28	PG14_AFCFG	PG14 AF 功能使能配置位 该位由软件置位和清除。 0：不配置 PG14 备用功能到 USART5 1：配置 PG14 备用功能到 USART5
27	保留	必须保持复位值。
26	PG13_AFCFG	PG13 AF 功能使能配置位 该位由软件置位和清除。 0：不配置 PG13 备用功能到 SHRTIMER 1：配置 PG13 备用功能到 SHRTIMER
25	保留	必须保持复位值。
24	PG12_AFCFG	PG12 AF 功能使能配置位 该位由软件置位和清除。 0：不配置 PG12 备用功能到 SHRTIMER 1：配置 PG12 备用功能到 SHRTIMER
23	保留	必须保持复位值。
22	PG11_AFCFG	PG11 AF 功能使能配置位 该位由软件置位和清除。 0：不配置 PG11 备用功能到 SHRTIMER 1：配置 PG11 备用功能到 SHRTIMER
21	保留	必须保持复位值。

20	PG10_AFCFG	PG10 AF 功能使能配置位 该位由软件置位和清除。 0：不配置 PG10 备用功能到 SHRTIMER 1：配置 PG10 备用功能到 SHRTIMER
19	保留	必须保持复位值。
18	PG9_AFCFG	PG9 AF 功能使能配置位 该位由软件置位和清除。 0：不配置 PG9 备用功能到 USART5 1：配置 PG9 备用功能到 USART5
17:16	保留	必须保持复位值。
15:14	PG7_AFCFG[1:0]	PG7 AF 功能使能配置位 这些位由软件置位和清除。 00：不配置 PG7 备用功能到 SHRTIMER/ USART5 01：配置 PG7 备用功能到 USART5 10/11：配置 PG7 备用功能到 SHRTIMER
13	保留	必须保持复位值。
12	PG6_AFCFG	PG6 AF 功能使能配置位 该位由软件置位和清除。 0：不配置 PG6 备用功能到 SHRTIMER 1：配置 PG6 备用功能到 SHRTIMER
11:0	保留	必须保持复位值。

9. 循环冗余校验管理单元（CRC）

9.1. 简介

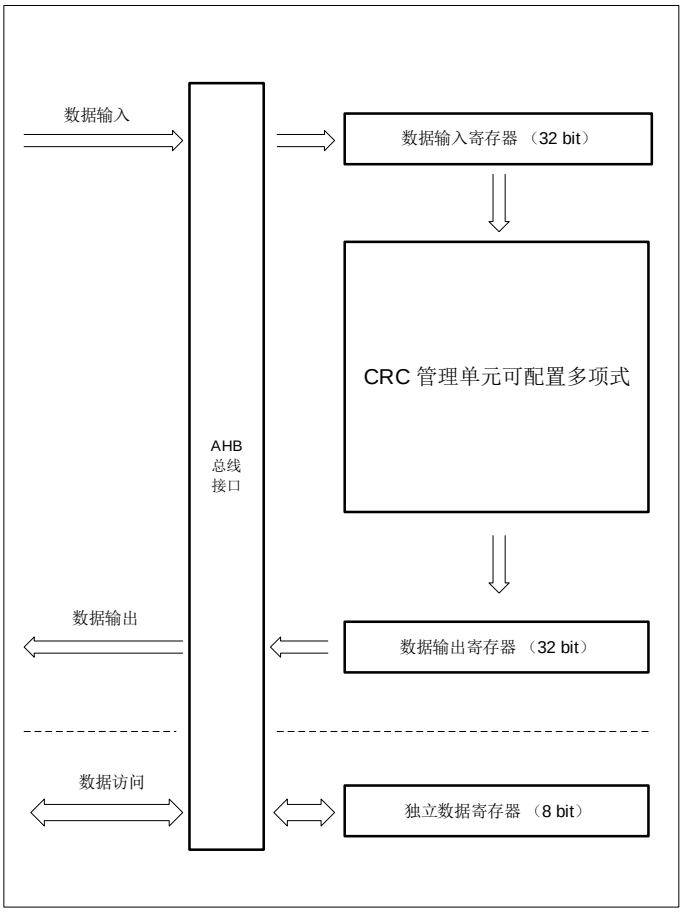
循环冗余校验码是一种用在数字网络和存储设备上的差错校验码，可以校验原始数据的偶然误差。

CRC 管理单元能用用户配置的多项式来计算 7/8/16/32 位的 CRC 校验码。

9.2. 主要特性

- 支持7/8/16/32位数据输入；
- 对于7（8）/16/32位的输入数据长度，计算周期分别为1/2/4个AHB时钟周期；
- 用户可以配置多项式及多项式长度；
- CRC复位后，用户可以配置计算初值；
- 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。

图 9-1. CRC 管理单元框图



9.3. 功能描述

- CRC管理单元可以用来计算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，新输入的原始数据将会基于前一次CRC_DATA寄存器中的结果进行计算。
  对于32/16/8（7）位的数据长度，CRC的计算分别要花费4/2/1个AHB的时钟周期。在此期间，因为32位输入缓存的原因，AHB总线将不会被挂起。
- 此模块提供了一个8位的独立寄存器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。
- 逆序功能可以交换输入输出数据的位序。
  输入数据可选择三种逆序形式。
  以原始数据0x3456CDEF为例：
  1）按字节逆序：
  32位数据被分成四组，组内完成颠倒。逆序后的数据为：0x2C6AB3F7
  2）按半字逆序：
  32位数据被分成两组，组内完成颠倒。逆序后的数据为：0x6A2CF7B3
  3）按字逆序：
  32位数据被分成一组，组内完成颠倒，逆序后的数据为：0xF7B36A2C
  对于输出数据来说，逆序形式为按字逆序。
  例如：当REV_O=1，计算结果0x3344CCDD将被逆序成0xBB3322CC。
- 用户可配置的初始计算数据。
  当RST位置位或对CRC_IDATA寄存器进行写操作时，CRC_DATA寄存器将自动初始化为CRC_IDATA寄存器中的值。
- 用户配置多项式。
  通过配置PS[1:0]，用户可以选择有效多项式和输出数据位宽。如果多项式少于32位，那么输入和输出数据的高位无效。当PS[1:0]或多项式改变后，需要复位CRC。

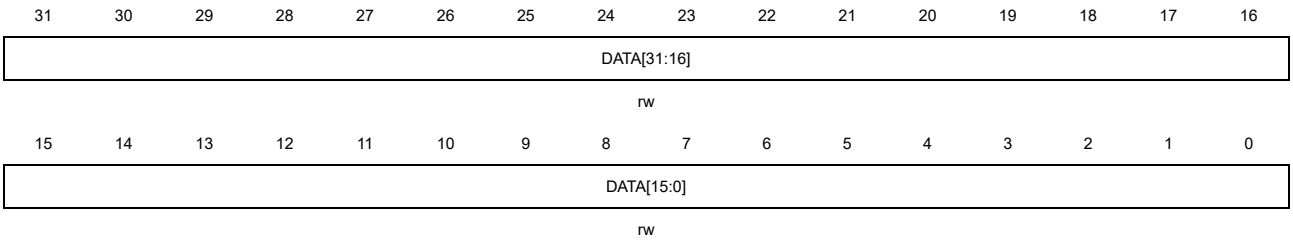
9.4. CRC 寄存器

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。

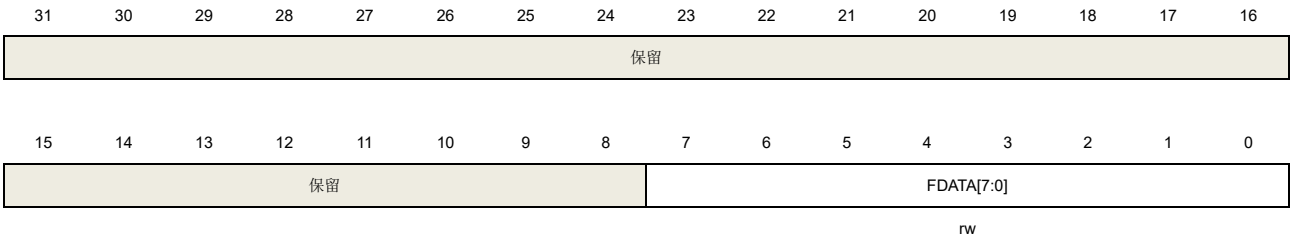


位/位域	名称	描述
31:0	DATA[31:0]	CRC 计算结果位 软件可读可写。 该寄存器用于接收待计算的新数据，直接将其写入即可。刚写入的数据不能被读出来因为读取该寄存器得到的是上次 CRC 计算的结果。

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。

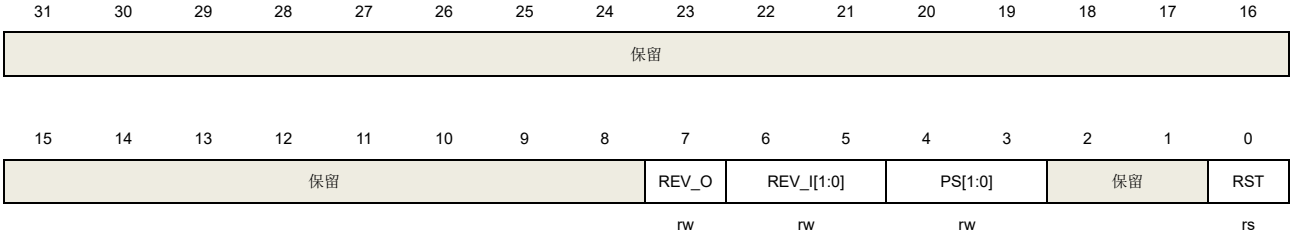


位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	FDATA[7:0]	独立数据寄存器位 软件可读可写。 这些位与 CRC 计算无关。该字节能被任何其他外设用于其他任何目的。该字节不受 CRC_CTL 寄存器的影响。

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。

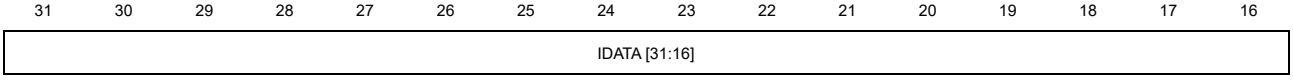


位/位域	名称	描述
31:8	保留	必须保持复位值
7	REV_O	按位顺序翻转输出数据功能 0：输出数据不翻转 1：输出数据按位顺序翻转
6:5	REV_I[1:0]	翻转输入数据功能 0：输入数据不翻转 1：输入数据按字节翻转 2：输入数据按半字翻转 3：输入数据按字翻转
4:3	PS[1:0]	多项式长度 0：32 位 1：16（POLY[15:0]用于计数）位 2：8（POLY[7:0]用于计数）位 3：7（POLY[6:0]用于计数）位
2:1	保留	必须保持复位值
0	RST	软件可读写 该位用来复位 CRC_DATA 寄存器。 置位时，CRC_DATA 寄存器的值将自动初始化为 CRC_IDATA 寄存器中的值，然后自动清零。该位对 CRC_FDATA 寄存器没有影响。

9.4.4. 初值寄存器（CRC_IDATA）

地址偏移：0x10
复位值：0xFFFF FFFF

该寄存器只能按字（32 位）访问



rw

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

rw

位/位域	名称	描述
31:0	IDATA[31:0]	配置 CRC 初值 CRC_CTL 寄存器的 RST 位置位后，CRC_DATA 寄存器的值将被更新为此寄存器的值。

9.4.5. 多项式寄存器（CRC_POLY）

地址偏移：0x14
复位值：0x04C1 1DB7

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
POLY [31:16]															

rw

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

rw

位/位域	名称	描述
31:0	POLY[31:0]	配置多项式值 配合 PS[1:0]使用。

10. 三角函数加速器（TMU）

10.1. 简介

三角函数加速器（TMU）是一个完全可配置的单元，可执行常见的三角运算和算术运算操作。TMU 总共有 9 种运算操作，其操作数据必须符合 IEEE-32 位单精度浮点格式。

10.2. 主要特性

- 输入数据和计算结果支持32位单精度浮点格式；
- 9种不同的操作模式；

表 10-1. 9 种不同的操作模式

模式	操作
0	R0 =x * 2π
1	R0 = x/2π
2	R0 = $\sqrt{x}$
3	R0 = sin (x)
4	R0 = cos (x)
5	R0 = arctan (x)
6	R0 = Ratio of X & Y, R1 = Quadrant value (0.0, ±0.25, ±0.5)
7	R0 = x/y
8	R0 = $\sqrt{x^2+y^2}$

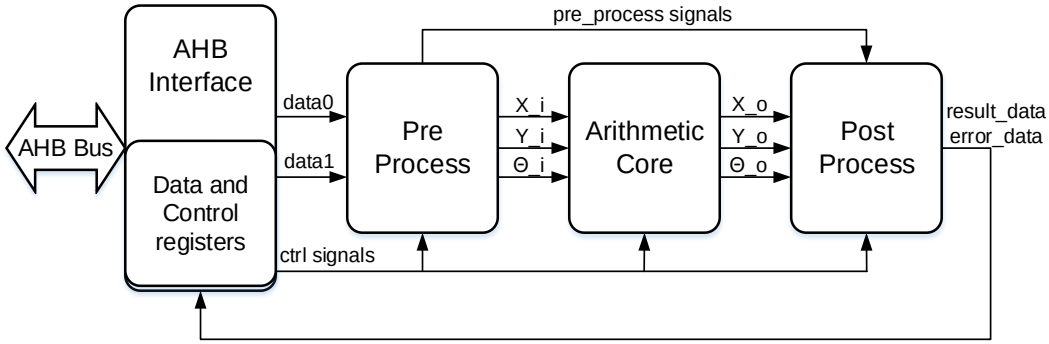
- 对于模式0和模式1，完成计算操作需要4个时钟周期，对于其他模式，则需要7个时钟周期；
- 可读取状态寄存器中的上溢和下溢错误标志；
- 具有可选择使能的计算操作完成中断；

10.3. 功能描述

10.3.1. TMU 结构图

TMU 模块结构图如*图 10-1. 三角函数加速器模块结构图*所示。

图 10-1. 三角函数加速器模块结构图



10.3.2. 数据格式

TMU 模块的操作数和计算结果的格式如表。数据格式必须满足 IEEE-32 位单精度浮点格式。

表 10-2. IEEE-32 位单精度浮点格式

S [31]	E [30:23]	M [22:0]	数值 (V)
0	0	0	零 (V = 0)
1	0	0	负零 (V = -0)
0 +ve 1 -ve	0	non zero	非标准 ($V=(-1)^s*2^{(-126)}*$ (0.M))
0 +ve 1 -ve	1 to 254	0 to 0x7FFFFF	正常范围($V=(-1)^{□}*2^{(E-127)}*$ (1.M))
0	254	0x7FFFFF	正最大值(V = +Max)
1	254	0x7FFFFF	负最大值(V = -Max)
0	max=255	0	正无穷(V = +Infinity)
1	max=255	0	负无穷(V = -Infinity)
x	max=255	non zero	非数 (V = NaN)

TMU 的几种 IEEE 浮点数值格式的处理如下：

负零：如果操作的结果是零，TMU 操作只产生一个正零（S=0，E=0，M=0），永远不会产生负零。TMU 将所以的负零操作视为零。

非标准数：TMU 将输入的非标准化操作数（E=0，M！=0）视为零（E=0，M=0）。TMU 操作从不生成非标准化值。

下溢：当计算结果的值太小，无法用给定的浮点格式表示时，会产生下溢。在这种情况下，返回零值。如果 TMU 操作生成下溢条件，则锁存下溢标志（UDRF）设置为 1。UDRF 标志将保持锁定状态，直到下一个新操作启动。

上溢：当计算结果的值太大，无法用给定的浮点格式表示时，会发生上溢。在这种情况下，返回正无穷大或负无穷大。如果 TMU 操作生成溢出条件，则锁存溢出标志（OVRF）设置为 1。OVRF 标志将保持锁定状态，直到下一个新操作开始。

舍入：IEEE 标准支持多种舍入格式。舍入对于 TMU 操作没有意义（舍入是实现中固有的）。因此，TMU 操作会忽略舍入模式。

无穷和非数（NaN）：对于所有的操作，TMU 都将输入一个 NaN 操作数（E=max，M！=0）

视为无穷大（E=max，M=0）。TMU 操作永远不会生成 NaN 值，而是生成无穷大。

10.3.3. 模式 0

模式 0 的运算为 R0 = x * 2 π 。x 为输入操作数据，R0 为计算结果。

此操作用于将单位值转换为弧度。在控制应用程序中将单位值表示标准化弧度。使用模式 0 将单位值转换为弧度时，则输入数据应满足[-1,1]的范围：

表 10-3. 模式 0 下单位值与弧度之间的转换

Per-unit	Radians
1.0	2π
0.0	0
-1.0	-2π

模式 0 下只有上溢（OVRF）标志位，且下溢标志位（UDRF）始终为零。上溢标志的产生条件如下：

如果浮点数格式表示的 R0 值太大（E>255）时，R0 等于正无穷或负无穷，OVRF=1。

10.3.4. 模式 1

模式 1 的运算为 R0 = x/2 π ，x 为输入操作数据，R0 为计算结果。

此操作用于将弧度转换为单位值。单位值用于表示标准化弧度的控件中。如果模式 1 用于将弧度转换为单位值，则输入数据应满足[-2 π , 2 π]的范围。

表 10-4. 模式 1 下弧度与单位值之间的转换

Per-unit	Radians
1.0	2π
0.0	0
-1.0	-2π

模式 1 下只有下溢（UDRF）标志位，且上溢标志位（OVRF）始终为零。上溢标志的产生条件如下：

如果浮点数格式表示的 R0 值太小（E< 0）时，R0=0.0，OVRF=1。

10.3.5. 模式 2

模式 2 的运算为 R0 =$\sqrt{x}$。x 为输入操作数据，R0 为计算结果。

模式 2 下只有上溢（OVRF）标志位，且下溢标志位（UDRF）始终为零。上溢标志的产生条件如下：

/* 检测输入数是否为负数 */

If(x < 0.0 or x == -Infinity) {

/* 返回 0 */

```
S = 0;
E = 0;
M = 0;
/*上溢标志位置 1 */
OVRF = 1;
}
If( x == +Infinity ) {
/*返回无穷*/
S = 0;
E = 255;
M = 0;
/*上溢标志位置 1 */
OVRF = 1;
}
```

10.3.6. 模式 3

模式 3 的操作等效以下操作：

1. 令 PerUnit 取输入操作数 x 的小数部分，即 PerUnit = fraction(x)。
2. R0 = sin(PerUnit * 2π)。

在应用时，弧度通常被标准化为-1.0 到 1.0 的范围，而 PerUnit * 2π 值的范围是(-2π, 2π)。

模式 3 下只使用输入操作数 x 的小数部分，因为正弦函数的周期为 2π，x 的整数部分对结果没有影响。

此模式既没有下溢标志（UDRF）也没有上溢标志（OVRF）。如果计算结果太小，则返回 0。

10.3.7. 模式 4

模式 4 的操作等效以下操作：

1. 令 PerUnit 取输入操作数 x 的小数部分，即 PerUnit = fraction(x)。
2. R0 = cos(PerUnit * 2π)。

在应用时，弧度通常被标准化为-1.0 到 1.0 的范围，而 PerUnit * 2π 值的范围是(-2π, 2π)。

模式 4 下只使用输入操作数 x 的小数部分，因为正弦函数的周期为 2π，x 的整数部分对结果没有影响。

此模式既没有 UDRF 也没有 OVRF。如果结果太小，则返回 0。

10.3.8. 模式 5

模式 5 用于计算给定值的反正切，并以单位值的形式返回结果：R0 = PerUnit = arctan(x)/2π。

该操作将输入值 x 的输入范围限制在[-1,1]之间。

超出此范围的值返回 0.125，如下表所示：

表 10-5. 模式 5 下输入的操作数与 R0 值的范围

x	Per Unit	Radians	R0 Value	OVRF
>1.0	0.125	pi/4	0.125	1
1.0	0.125	pi/4	0.125	0
0	0	0	0	0
-1.0	-0.125	-pi/4	-0.125	0
<-1.0	-0.125	-pi/4	-0.125	1

模式 5 下只有上溢（OVRF）标志位，且下溢标志位（UDRF）始终为零。当输入的操作数 x 超出[-1,1]范围时，上溢标志位置位。

10.3.9. 模式 6

模式 6 操作与 arctan（x）一起用于计算整圆的 arctan(x)。整圆 arctan(x) = R1 + arctan(R0)。R0 和 R1 运算结果如下：

X = x value.

Y = y value.

R0 = Ratio of X & Y.

R1 = Quadrant value (0.0, ±0.25, ±0.5).

此模式的运算规则如下：

If ((fabs(Y) == 0.0) & (fabs(X) == 0.0)) {

R1(Quadrant) = 0.0;

R0(Ratio) = 0.0;

}else if (fabs(Y) < = fabs(X)) {

R0(Ratio) = Y / X;

If(X >= 0.0)

R1(Quadrant) = 0.0;

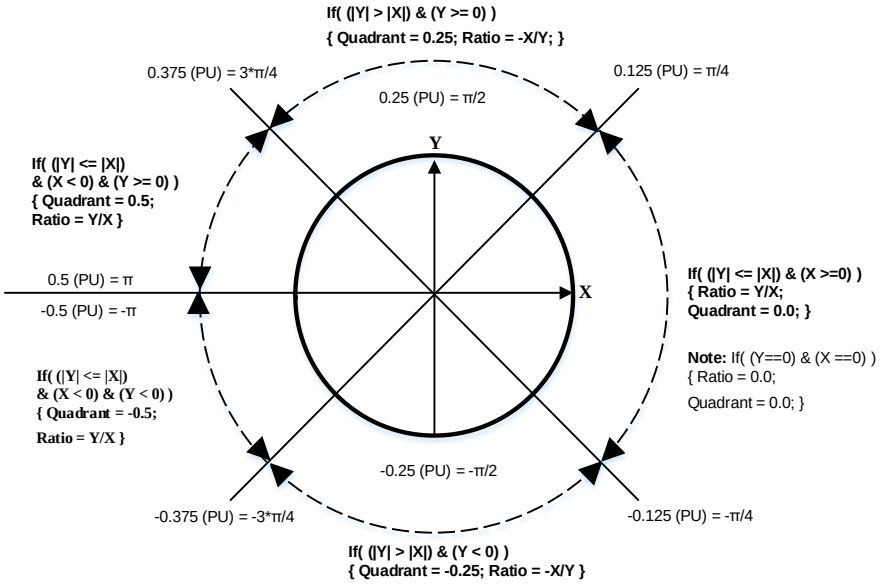
else {

If(Y >= 0.0)

```
R1( Quadrant ) = 0.5;
else
R1( Quadrant ) = -0.5;
}
}else {
R0( Ratio ) = - X / Y;
if( Y >= 0.0 )
R1( Quadrant ) = 0.25;
else
R1( Quadrant ) = -0.25;
}
```

图 10-2. 基于 X 与 Y 的比值计算 R0 和 R1 展示了如何根据 X 与 Y 比值计算 R0 和 R1 的值。

图 10-2. 基于 X 与 Y 的比值计算 R0 和 R1



模式 6 下具有下溢标志（UDRF）和上溢标志（OVRF）。产生 UDRF 和 OVRF 的条件如下表：

表 10-6. 模式 6 下产生 UDRF 和 OVRF 的条件

Division(Ratio of X & Y)	R0	OVRF	UDRF
0/0	0	1	0
0/Inf	0	0	1
Inf/Inf	inf	0	1
Normal/Inf	0	0	1

10.3.10. 模式 7

模式 7 的运算为 R0 = x/y。x 和 y 为输入操作数据，R0 为计算结果。

此模式下具有下溢标志（UDRF）和上溢标志（OVRF）。产生 UDRF 和 OVRF 的条件如下表：

表 10-7. 模式 7 下产生 UDRF 和 OVRF 的条件

Division(X/Y)	R0	OVRF	UDRF
0/0	0	1	0
0/Inf	0	0	1
Inf/Normal	inf	1	0
Inf/0	inf	1	0
Inf/Inf	inf	0	1
Normal/0	inf	1	0
Normal/Inf	0	0	1

10.3.11. 模式 8

模式 8 的运算为 R0 = $\sqrt{x^2+y^2}$。x 和 y 为输入操作数据，R0 为计算结果。

此模式只有上溢（OVRF）标志位，且下溢标志位（UDRF）始终为零。上溢标志的产生条件如下：

如果浮点数格式表示的 R0 值太大（E>255）时，R0 等于正无穷或负无穷，OVRF=1。

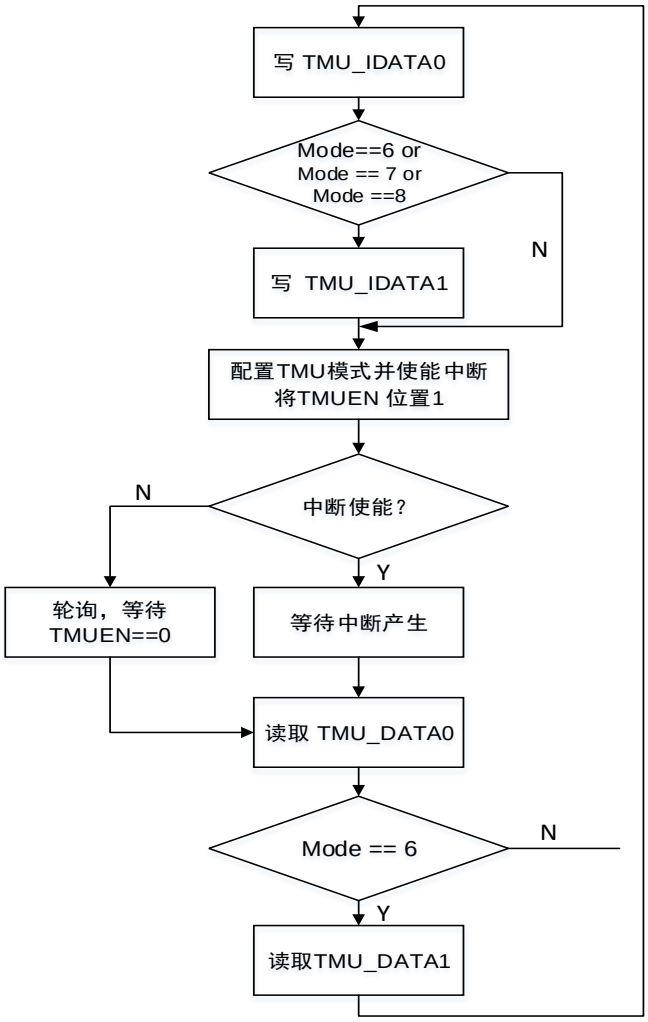
10.4. 软件流程

建议使用以下步骤读取 TMU 结果：

1. 写入 TMU IDATA0 寄存器，如果模式为 6、7 或 8，需要再写入 TMU IDATA1 寄存器。
2. 配置操作模式和中断启用位，将 TMUEN 位置 1，开始计算。
3. 如果中断使能为 1，则在计算完成后立即产生中断，否则软件应轮询 TMUEN 位并等待 TMUEN 位为 0。
4. 读取 TMU 数据 0 寄存器，如果模式为 6，也应读取 TMU 数据 1 寄存器。
5. 读取 UDRF 和 OVRF 位，确保没有错误发生。

TMU 程序指南如*图 10-3. TMU 软件流程*所示。

图 10-3. TMU 软件流程


写 TMU_IDATA0
Mode==6 or
Mode == 7 or
Mode ==8
N
写 TMU_IDATA1
配置TMU模式并使能中断
将TMUEN 位置1
中断使能?
N
Y
轮询，等待
TMUEN==0
等待中断产生
读取 TMU_DATA0
Mode == 6
N
Y
读取TMU_DATA1

10.5. TMU 寄存器

TMU 基地址：0x4008 0000

10.5.1. 输入数据 0 寄存器 (TMU_IDATA0)

地址偏移：0x00
复位值：0x3F80 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
IDATA0[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDATA0[15:0]															
rw															

位/位域	名称	描述
31:0	IDATA0[31:0]	输入的数值 模式 0~5：只使用 TMU_IDATA0 模式 6：TMU_IDATA0 用于存放 X 的值 模式 7：TMU_IDATA0 用于存放被除数 模式 8：TMU_IDATA0 用于存放 X 或 Y 的值 TMU_IDATA0 必须符合 IEEE-32 位单精度浮点格式。

10.5.2. 输入数据 1 寄存器(TMU_IDATA1)

地址偏移：0x04
复位值：0x3F80 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
IDATA1[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDATA1[15:0]															
rw															

位/位域	名称	描述
31:0	IDATA1[31:0]	输入的数值 模式 0~5：不使用 TMU_IDATA1 模式 6：TMU_IDATA1 用于存放 Y 的值 模式 7：TMU_IDATA1 用于存放除数的值

模式 8：IDATA1 用于存放 X 或 Y 的值

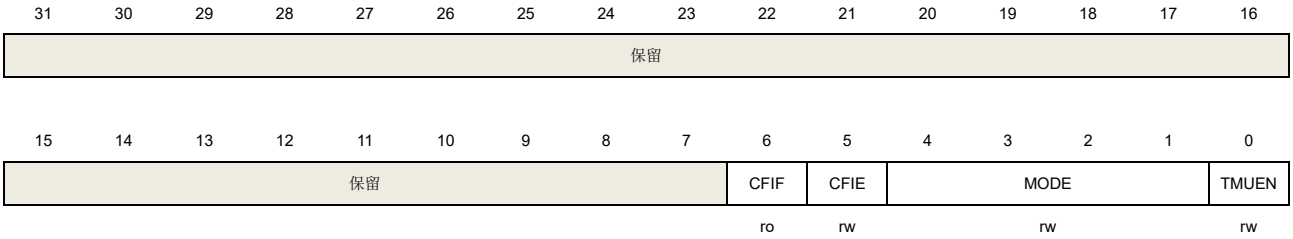
TMU_IDATA1 必须符合 IEEE-32 位单精度浮点格式。

10.5.3. 控制寄存器(TMU_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
31:7	保留	必须保持复位值
6	CFIF	计算完成标志位 当 CFIE 使能时，一旦计算完成，此位被置 1 并产生中断信号。读取 TMU_DATA0、TMU_DATA1 或 TMU_STAT 寄存器，此位被清零。
5	CFIE	计算完成中断使能位 1：使能计算完成中断 0：禁止计算完成中断
4:1	MODE[3:0]	配置 TMU 操作模式： 0000：R0 = x*2π 0001：R0 = x/2π 0010：R0 = $\sqrt{x}$ 0011：R0 = sin(x) 0100：R0 = cos(x) 0101：R0 = arctan(x) 0110：R0 = Ratio of X & Y, R1 = Quadrant value 0111：R0 = x/y 1000：R0 = $\sqrt{x^2+y^2}$ 1001~1111：保留
0	TMUEN	TMU 使能位 该位置 1，开始 TMU 模块计算功能，当完成计算时，该位被硬件清零。

10.5.4. 数据 0 寄存器(TMU_DATA0)

地址偏移：0x0C

复位值：0x3400 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
DATA0[31:16]															
ro															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA0[15:0]															
ro															

位/位域	名称	描述
31:0	DATA0[31:0]	计算结果 模式 0~5,7,8：只使用 TMU_DATA0 模式 6：TMU_DATA0 存放 X 和 Y 的比值结果 TMU_DATA0 必须符合 IEEE-32 位单精度浮点格式。

10.5.5. 数据 1 寄存器(TMU_DATA1)

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
DATA1[31:16]															
ro															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA1[15:0]															
ro															

位/位域	名称	描述
31:0	DATA1[31:0]	计算结果 模式 0~5,7,8：不使用 TMU_DATA1 模式 6：TMU_DATA1= Quadrant value (0.0, ±0.25, ±0.5) TMU_DATA1 必须符合 IEEE-32 位单精度浮点格式。

10.5.6. 状态寄存器(TMU_STAT)

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	UDRF	OVRF
	ro	ro

位/位域	名称	描述
31:2	保留	必须保持复位值
1	UDRF	下溢标志位 0：没有下溢 1：产生下溢 该位由硬件清零或置位，当启动下一次 TMU 计算时，该位被硬件清零。
0	OVRF	上溢标志位 0：没有上溢 1：产生上溢 该位由硬件清零或置位，当启动下一次 TMU 计算时，该位被硬件清零。

11. 直接存储器访问控制器（DMA）

11.1. 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，从而使 CPU 可以专注在处理其他系统功能上。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。

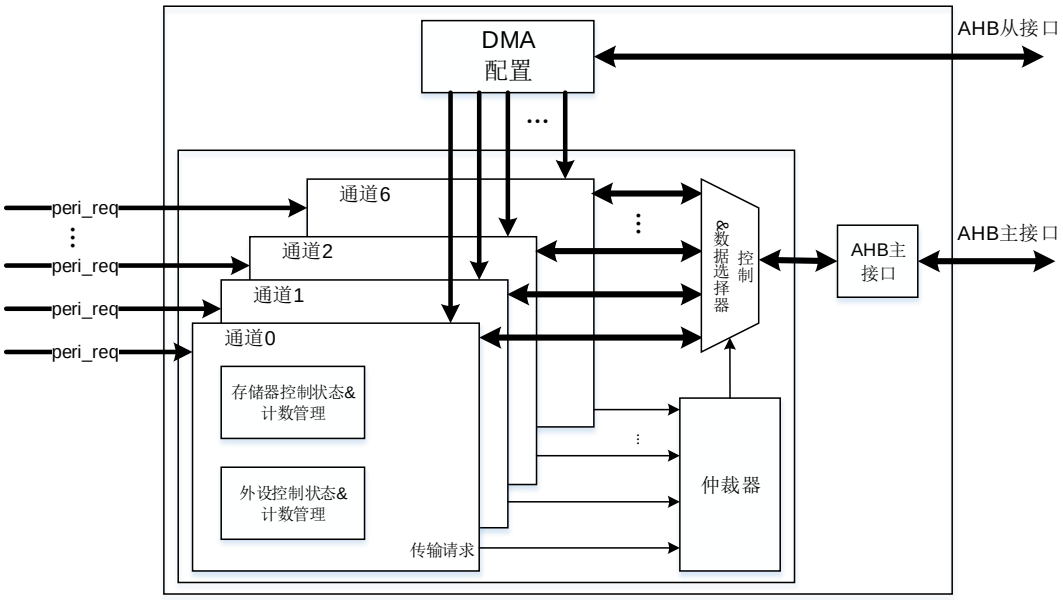
DMA 控制器和 Cortex®-M33 内核共享系统总线。当 DMA 和 CPU 访问同样的地址空间时，DMA 访问可能会阻挡 CPU 访问系统总线几个总线周期。总线矩阵中实现了循环仲裁算法来分配 DMA 与 CPU 的访问权，它可以确保 CPU 得到至少一半的系统总线带宽。

11.2. 主要特性

- 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。

11.3. 结构框图

图 11-1. DMA 结构框图



由*图 11-1. 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
- 仲裁器进行 DMA 请求的优先级管理
- 数据处理和计数

11.4. 功能描述

11.4.1. 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 11-1. DMA 传输操作*。

表 11-1. 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 2: Read B7B6B5B4[31:0] @0x4 3: Read BBBAB9B8[31:0] @0x8 4: Read BFBEBDBC[31:0] @0xC	1: Write B3B2B1B0[31:0] @0x0 2: Write B7B6B5B4[31:0] @0x4 3: Write BBBAB9B8[31:0] @0x8 4: Write BFBEBDBC[31:0] @0xC
32 bits	16 bits	1: Read B3B2B1B0[31:0] @0x0 2: Read B7B6B5B4[31:0] @0x4 3: Read BBBAB9B8[31:0] @0x8	1: Write B1B0[15:0] @0x0 2: Write B5B4[15:0] @0x2 3: Write B9B8[15:0] @0x4

传输宽度		传输操作	
源	目标	源	目标
		4: Read BFBEBDBC[31:0] @0xC	4: Write BDBC[5:0] @0x6
32 bits	8 bits	1: Read B3B2B1B0[31:0] @0x0 2: Read B7B6B5B4[31:0] @0x4 3: Read BBBAB9B8[31:0] @0x8 4: Read BFBEBDBC[31:0] @0xC	1: Write B0[7:0] @0x0 2: Write B4[7:0] @0x1 3: Write B8[7:0] @0x2 4: Write BC[7:0] @0x3
16 bits	32 bits	1: Read B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 3: Read B5B4[15:0] @0x4 4: Read B7B6[15:0] @0x6	1: Write 0000B1B0[31:0] @0x0 2: Write 0000B3B2[31:0] @0x4 3: Write 0000B5B4[31:0] @0x8 4: Write 0000B7B6[31:0] @0xC
16 bits	16 bits	1: Read B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 3: Read B5B4[15:0] @0x4 4: Read B7B6[15:0] @0x6	1: Write B1B0[15:0] @0x0 2: Write B3B2[15:0] @0x2 3: Write B5B4[15:0] @0x4 4: Write B7B6[15:0] @0x6
16 bits	8 bits	1: Read B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 3: Read B5B4[15:0] @0x4 4: Read B7B6[15:0] @0x6	1: Write B0[7:0] @0x0 2: Write B2[7:0] @0x1 3: Write B4[7:0] @0x2 4: Write B6[7:0] @0x3
8 bits	32 bits	1: Read B0[7:0] @0x0 2: Read B1[7:0] @0x1 3: Read B2[7:0] @0x2 4: Read B3[7:0] @0x3	1: Write 000000B0[31:0] @0x0 2: Write 000000B1[31:0] @0x4 3: Write 000000B2[31:0] @0x8 4: Write 000000B3[31:0] @0xC
8 bits	16 bits	1: Read B0[7:0] @0x0 2: Read B1[7:0] @0x1 3: Read B2[7:0] @0x2 4: Read B3[7:0] @0x3	1, Write 00B0[15:0] @0x0 2, Write 00B1[15:0] @0x2 3, Write 00B2[15:0] @0x4 4, Write 00B3[15:0] @0x6
8 bits	8 bits	1: Read B0[7:0] @0x0 2: Read B1[7:0] @0x1 3: Read B2[7:0] @0x2 4: Read B3[7:0] @0x3	1, Write B0[7:0] @0x0 2, Write B1[7:0] @0x1 3, Write B2[7:0] @0x2 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，对任意相关寄存器进行了操作，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进行操作前便使能 DMA 通道，则不会触发任何 DMA 传输。

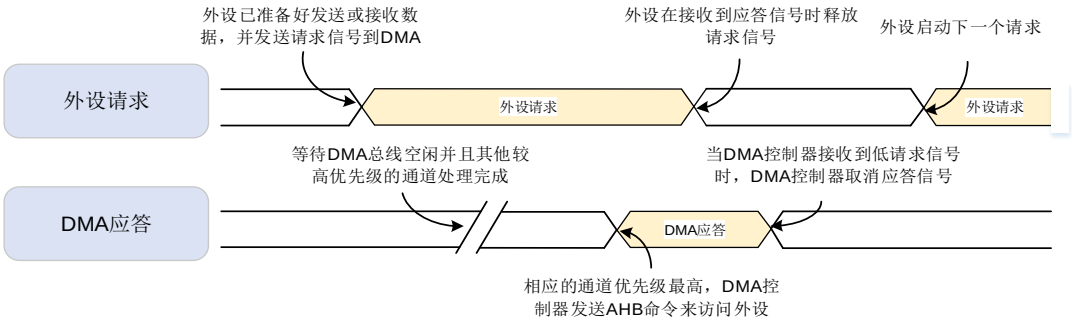
11.4.2. 外设握手

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11-2. 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 11-2. 握手机制



11.4.3. 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。

11.4.4. 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。

11.4.5. 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。

11.4.6. 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。

11.4.7. 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，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 开始新的传输；
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。

11.4.8. 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。

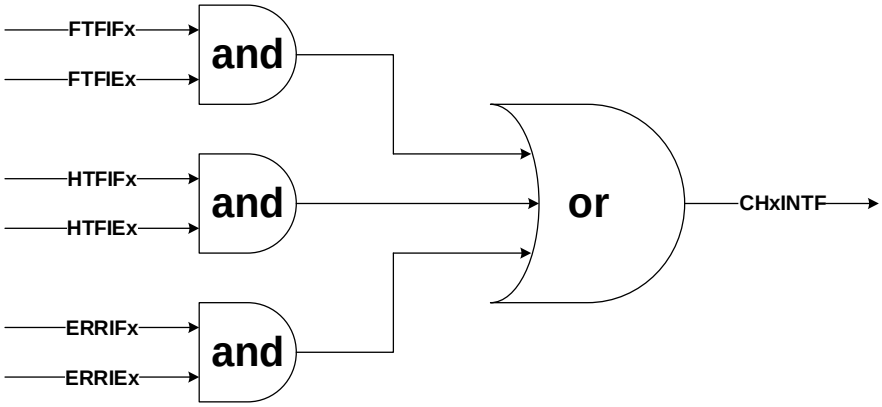
每一个中断事件在 DMA_INTF 寄存器中有专用的标志位，在 DMA_INTC 寄存器中有专用的清除位，在 DMA_CHxCTL 寄存器中有专用的使能位。*表 11-2. 中断事件*描述了其对应关系。

表 11-2. 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	FTFIF	FTFIFC	FTFIE
半传输完成	HTFIF	HTFIFC	HTFIE
传输错误	ERRIF	ERRIFC	ERRIE

DMA中断逻辑如*表11-3. DMA0各通道请求表*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 11-3. 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）

11.4.9. 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 11-4. DMA0 请求映射***和***图 11-5. 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 11-3. 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 11-4. DMA1 各通道请求表***各通道请求表列举了 DMA1 的每个通道所支持的外设请求。

图 11-4. DMA0 请求映射

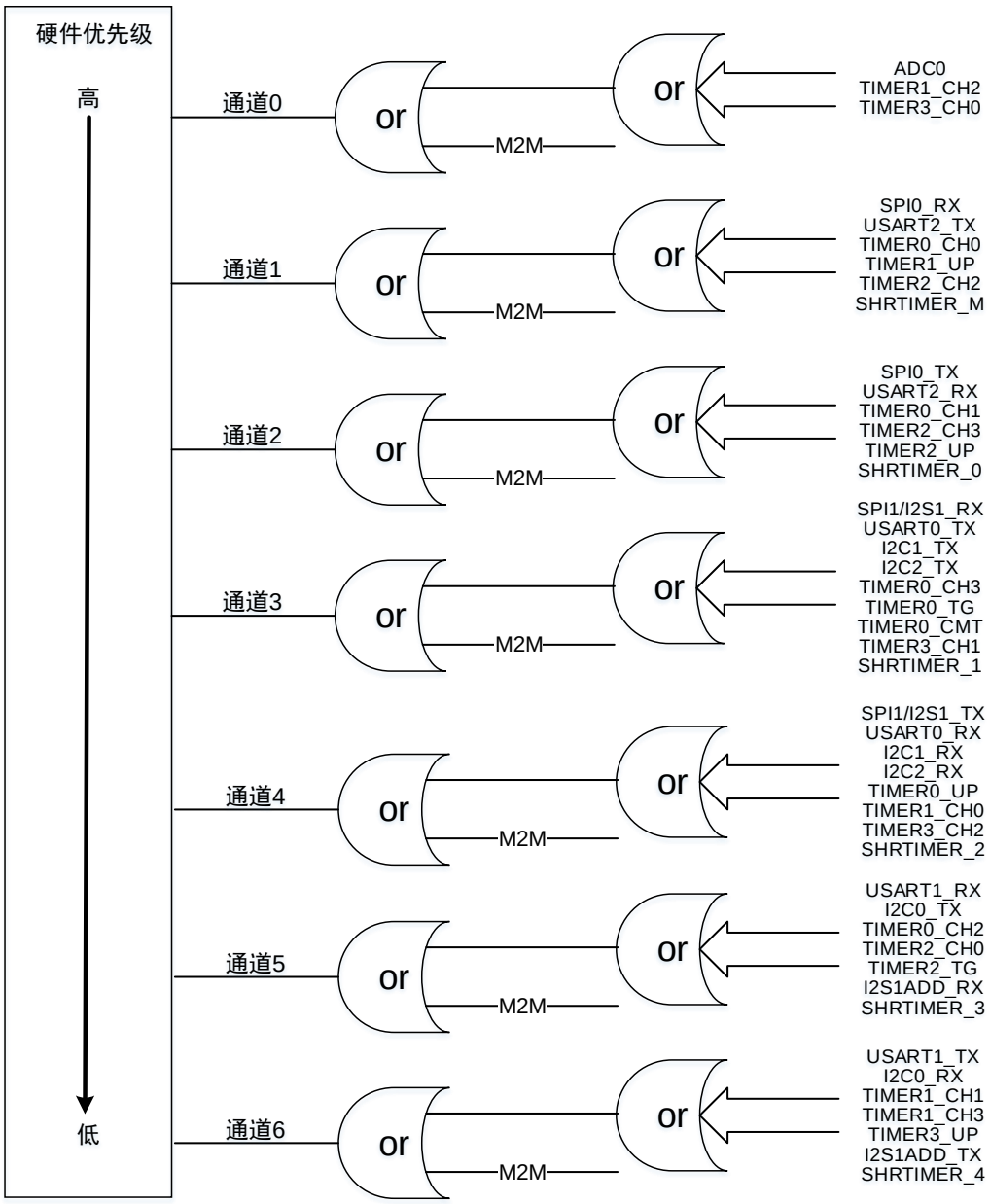


表 11-3. DMA0 各通道请求表

Peripheral	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6
TIMER0	•	TIMER0_CH0	TIMER0_CH1	TIMER0_CH3 TIMER0_TG TIMER0_CMT	TIMER0_UP	TIMER0_CH2	•
TIMER1	TIMER1_CH2	TIMER1_UP	•	•	TIMER1_CH0	•	TIMER1_CH1 TIMER1_CH3
TIMER2	•	TIMER2_CH2	TIMER2_CH3 TIMER2_UP	•	•	TIMER2_CH0 TIMER2_TG	•
TIMER3	TIMER3_CH0	•	•	TIMER3_CH1	TIMER3_CH2	•	TIMER3_UP
ADC0	ADC0	•	•	•	•	•	•

Peripheral	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6
SPI/I2S	•	SPI0_RX	SPI0_TX	SPI1/I2S1_RX	SPI1/I2S1_TX	I2S1ADD_RX	I2S1ADD_TX
USART	•	USART2_TX	USART2_RX	USART0_TX	USART0_RX	USART1_RX	USART1_TX
I2C	•	•	•	I2C1_TX I2C2_TX	I2C1_RX I2C2_RX	I2C0_TX	I2C0_RX
SHRTIM	•	SHRTIMER_M	SHRTIMER_0	SHRTIMER_1	SHRTIMER_2	SHRTIMER_3	SHRTIMER_4

图 11-5. DMA1 请求映射

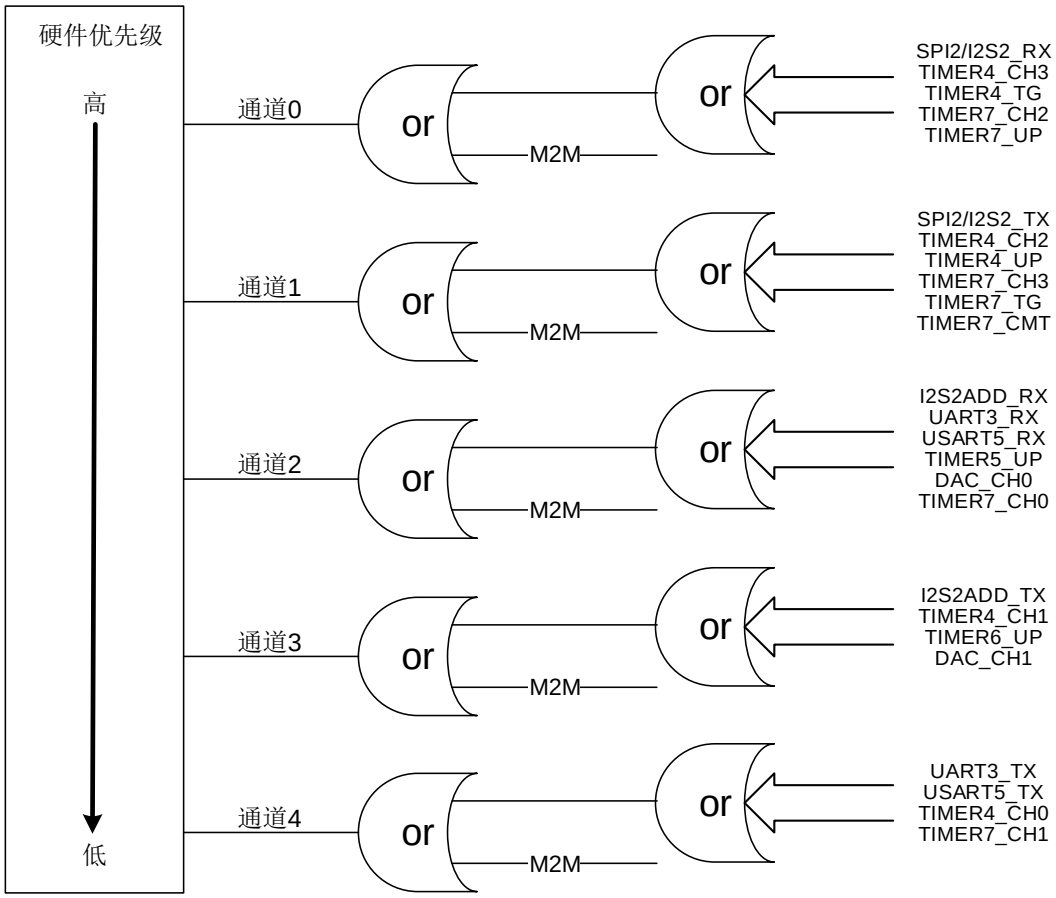


表 11-4. DMA1 各通道请求表

Peripheral	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4
TIMER4	TIMER4_CH3 TIMER4_TG	TIMER4_CH2 TIMER4_UP	•	TIMER4_CH1	TIMER4_CH0
TIMER5	•	•	TIMER5_UP	•	•
TIMER6	•	•	•	TIMER6_UP	•
TIMER7	TIMER7_CH2 TIMER7_UP	TIMER7_CH3 TIMER7_TG TIMER7_CMT	TIMER7_CH0	•	TIMER7_CH1
DAC	•	•	DAC_CH0	DAC_CH1	•
SPI/I2S	SPI2/I2S2_RX	SPI2/I2S2_TX	I2S2ADD_RX	I2S2ADD_TX	•
USART	•	•	UART3_RX	•	UART3_TX

Peripheral	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4
			USART5_RX		USART5_TX

11.5. 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。

11.5.1. 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/ 15/11/7/3	ERRIFx	通道 x 错误标志位(x=0…6) 硬件置位，软件写 DMA_INTC 相应位为 1 清零 0：通道 x 未发生传输错误 1：通道 x 发生传输错误
26/22/18/ 14/10/6/2	HTFIFx	通道 x 半传输完成标志位(x=0…6) 硬件置位，软件写 DMA_INTC 相应位为 1 清零 0：通道 x 半传输未完成 1：通道 x 半传输完成
25/21/17/ 13/9/5/1	FTFIFx	通道 x 传输完成标志位(x=0…6) 硬件置位，软件写 DMA_INTC 相应位为 1 清零 0：通道 x 传输未完成 1：通道 x 传输完成
24/20/16/ 12/8/4/0	GIFx	通道 x 全局中断标志位(x=0…6) 硬件置位，软件写 DMA_INTC 相应位为 1 清零 0：通道 x ERRIF, HTFIF 或 FTFIF 标志位未置位 1：通道 x 至少发生 ERRIF, HTFIF 或 FTFIF 之一置位

11.5.2. 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/ 15/11/7/3	ERRIFCx	清除通道 x(x=0…6)的错误标志位 0：无影响 1：清零 DMA_INTF 寄存器的 ERRIFx 位
26/22/18/ 14/10/6/2	HTFIFCx	清除通道 x(x=0…6)的半传输完成标志位 0：无影响 1：清零 DMA_INTF 寄存器的 HTFIFx 位
25/21/17/ 13/9/5/1	FTFIFCx	清除通道 x(x=0…6)的传输完成标志位 0：无影响 1：清零 DMA_INTF 寄存器的 FTFIFx 位
24/20/16/ 12/8/4/0	GIFCx	清除通道 x(x=0…6)的全局中断标志位 0：无影响 1：清零 DMA_INTF 寄存器的 GIFx，ERRIFx，HTFIFx 和 FTFIFx 位

11.5.3. 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 软件置位和清零 0：禁止存储器到存储器模式

		1：使能存储器到存储器模式 CHEN 位为 1 时，该位不能被配置
13:12	PRIO[1:0]	软件优先级 软件置位和清零 00：低 01：中 10：高 11：极高 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 软件置位和清零 00：8-bit 01：16-bit 10：32-bit 11：保留 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 软件置位和清零 00：8-bit 01：16-bit 10：32-bit 11：保留 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 软件置位和清零 0：固定地址模式 1：增量地址模式 CHEN 位为 1 时，该位不能被配置
6	PNAGA	外设的地址生成算法 软件置位和清零 0：固定地址模式 1：增量地址模式 CHEN 位为 1 时，该位不能被配置
5	CMEN	循环模式使能 软件置位和清零 0：禁止循环模式 1：使能循环模式 CHEN 位为 1 时，该位不能被配置
4	DIR	传输方向 软件置位和清零 0：从外设读出并写入存储器

		1：从存储器读出并写入外设 CHEN 位为 1 时，该位不能被配置
3	ERRIE	通道错误中断使能位 软件置位和清零 0：禁止通道错误中断 1：使能通道错误中断
2	HTFIE	通道半传输完成中断使能位 软件置位和清零 0：禁止通道半传输完成中断 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 软件置位和清零 0：禁止通道传输完成中断 1：使能通道传输完成中断
0	CHEN	通道使能 软件置位和清零 0：禁止该通道 1：使能该通道

11.5.4. 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 该寄存器表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。

11.5.5. 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 CHEN 位为 1 时，该位域不能被配置 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。 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。

11.5.6. 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 CHEN 位为 1 时，该位域不能被配置 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。

12. 调试（DBG）

12.1. 简介

GD32A508xx 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®-M33 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®-M33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在低功耗模式下调试一些外设。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、WWDGT、FWDGT、I2C 和 CAN。

12.2. JTAG/SW 功能描述

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。

12.2.1. 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。

12.2.2. 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 ***JTAG/SWD 备用功能重映射***。

12.2.3. JTAG 链状结构

Cortex®-M33 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®-M33 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 Cortex®-M33 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。

12.2.4. 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®-M33 的绝大部分组件，除了 NVIC，调试逻辑（FPB、DWT 和 TM）。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。

12.2.5. JEDEC-106 ID code

Cortex®-M33 集成了 JEDEC-106 ID 代码。位于 ROM 表中，映射地址为 0xE00FF000_0xE00FFFFF。

12.3. 调试保持功能描述

12.3.1. 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。

12.3.2. 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：

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；

对于 CAN 外设，接收寄存器停止计数并进行调试。

12.4. DBG 寄存器

DEBUG 基地址：0xE004 4000

12.4.1. ID 寄存器（DBG_ID）

地址偏移：0x00

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 这些位由软件读取，这些位是不变的常数。

12.4.2. 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
SHRTIMER_HOLD.	TIMER10_HOLD	TIMER9_HOLD	TIMER8_HOLD	TIMER13_HOLD	TIMER12_HOLD	TIMER11_HOLD	保留.		I2C2_HOLD	CAN1_HOLD	TIMER6_HOLD	TIMER5_HOLD	TIMER4_HOLD	TIMER7_HOLD	I2C1_HOLD
rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C0_HOLD	CAN0_HOLD	TIMER3_HOLD	TIMER2_HOLD	TIMER1_HOLD	TIMER0_HOLD	WWDGT_HOLD	FWDGT_HOLD	保留.		TRACE_IOEN	保留.	CAN2_HOLD	STB_HOLD	DSLP_HOLD	SLP_HOLD
rw	rw	rw	rw	rw	rw	rw	rw			rw		rw	rw	rw	rw

位/位域	名称	描述
31	SHRTIMER_HOLD	SHRTIMER 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持 SHRTIMER 定时器计数器不变，用于调试。
30	TIMER10_HOLD	TIMER10 保持寄存器 该位由软件置位和复位。 0: 无影响

		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 该位由软件置位和复位。 0: 无影响 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 该位由软件置位和复位。 0: 无影响 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 该位由软件置位和复位。 0: 无影响 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 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持定时器 12 计数器不变，用于调试。
25	TIMER11_HOLD	TIMER11 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持定时器 11 计数器不变，用于调试。
24:23	保留	必须保持复位值
22	I2C2_HOLD	I2C2 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持 I2C2 的 SMBUS 状态不变，用于调试。
21	CAN1_HOLD	CAN1 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时 CAN1 接收寄存器停止接收数据。
20	TIMER6_HOLD	TIMER6 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持定时器 6 计数器不变，用于调试。
19	TIMER5_HOLD	TIMER5 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持定时器 5 计数器不变，用于调试。
18	TIMER4_HOLD	TIMER4 保持寄存器

该位由软件置位和复位。
0: 无影响
1: 当内核停止时保持定时器 4 计数器不变，用于调试。

17 TIMER7_HOLD TIMER7 保持寄存器
该位由软件置位和复位。
0: 无影响
1: 当内核停止时保持定时器 7 计数器不变，用于调试。

16 I2C1_HOLD I2C1 保持寄存器
该位由软件置位和复位。
0: 无影响
1: 当内核停止时保持 I2C1 的 SMBUS 状态不变，用于调试。

15 I2C0_HOLD I2C0 保持寄存器
该位由软件置位和复位。
0: 无影响
1: 当内核停止时保持 I2C0 的 SMBUS 状态不变，用于调试。

14 CAN0_HOLD CAN0 保持寄存器
该位由软件置位和复位。
0: 无影响
1: 当内核停止时 CAN0 接收寄存器停止接收数据。

13 TIMER3_HOLD TIMER3 保持寄存器
该位由软件置位和复位。
0: 无影响
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 TIMER2 保持寄存器
该位由软件置位和复位。
0: 无影响
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 TIMER1 保持寄存器
该位由软件置位和复位。
0: 无影响
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 TIMER0 保持寄存器
该位由软件置位和复位。
0: 无影响
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
该位由软件置位和复位。
0: 无影响
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 该位由软件置位和复位。 0: 无影响 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 该位由软件置位和复位。 0: 跟踪引脚分配禁用 1: 跟踪引脚分配使能
4	保留	必须保持复位值
3	CAN2_HOLD	CAN2 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时 CAN2 接收寄存器停止接收数据。 注意：仅在 CL 系列支持。
2	STB_HOLD	待机模式保持寄存器 该位由软件置位和复位。 0: 无影响 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 该位由软件置位和复位。 0: 无影响 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 该位由软件置位和复位。 0: 无影响 1: 在睡眠模式下, AHB 时钟继续运行。

13. 模数转换器（ADC）

13.1. 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（LSB）或最高有效位对齐（MSB）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。

13.2. 主要特征

- 高性能：
  - ADC 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（VSENSE）；
  - 1 个内部参考电压输入通道（VREFINT）。
- 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，每次触发转换一次所选择的输入通道；
  - 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。
- 模块供电要求：一般供电电压为3.3V
  - 1.62V 到 2.4V，ADC 最大时钟频率可达 14MHz；
  - 2.4V 到 3.6V，ADC 最大时钟频率可达 35MHz。
- 通道输入范围：$V_{REF-} \leqslant V_{IN} \leqslant V_{REF+}$。

13.3. 引脚和内部信号

*图13-1. ADC模块框图（ADC0和ADC1）*给出了ADC模块框图。*表13-2. ADC输入引脚定义*给出了ADC引脚说明。

表 13-1. ADC 内部输入信号

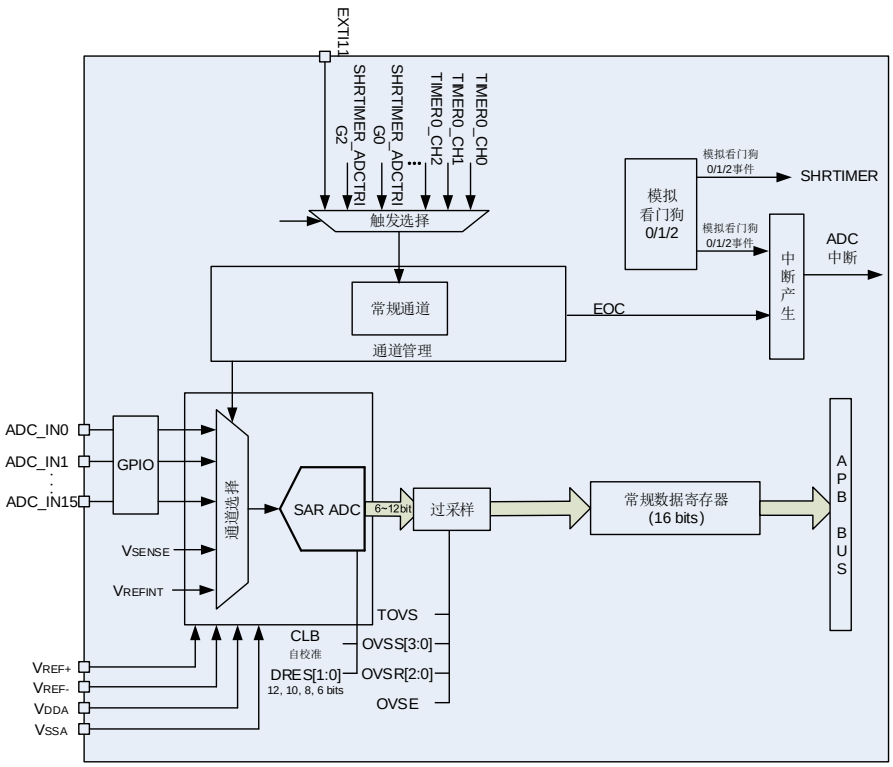
内部信号名称	说明
V_{SENSE}	内部温度传感器电压输出
V_{REFINT}	内部参考电压输出

表 13-2. ADC 输入引脚定义

名称	说明
V_{DDA}	模拟电源输入等于V_{DD}， 1.62V ≤ V_{DDA}≤ 2.4V（ADC最大时钟频率为14MHz） 2.4V ≤ V_{DDA}≤ 3.6V（ADC最大时钟频率为35MHz）
V_{SSA}	模拟地，等于V_{SS}
V_{REF+}	ADC正参考电压， 1.62V ≤ V_{DDA}≤ 2.4V（ADC最大时钟频率为14MHz） 2.4V ≤ V_{DDA}≤ 3.6V（ADC最大时钟频率为35MHz）
V_{REF-}	ADC负参考电压，V_{REF-} = V_{SSA}
ADCx_IN[15:0]	多达16路外部通道

13.4. 功能说明

图 13-1. ADC 模块框图（ADC0 和 ADC1）



13.4.1. 前置校准功能

在前置校准期间，ADC 计算一个校准因子，这个因子是应用于 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，必须等到校准完成。在开始 A/D 转换前应执行校准操作。通过设置 CLB=1 启动校准，在校准期间 CLB 位会一直保持 1。一旦校准完成，CLB 位由硬件清 0。通过设置 ADC_CTL1 寄存器中 CALNUM[2:0]位域来配置校准过程的重复次数，并对结果进行平均以得到更精确的校准结果。

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. 设置校准次数CALNUM（该步骤是可选的）；
4. 设置RSTCLB（该步骤是可选的）；
5. 设置CLB=1；
6. 等待直到CLB=0。

13.4.2. ADC 时钟

CK_ADC 时钟是由 RCU 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。

13.4.3. 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。

13.4.4. 单端和差分输入通道

通过配置 ADC_DIFCTL 寄存器中的 DIFCTL[14:0]位域，可以配置 ADC 通道为单端输入模式或差分输入模式。只有在 ADC 禁能（ADCON = 0）的情况下才能进行该配置。

单端输入模式下，通道 n 要转换的模拟电压是外部电压 V_{INn}（正输入）和 V_{REF-}（负输入）之间的差。差分输入模式下，通道 n 要转换的模拟电压是外部电压 V_{INn}（正输入）和 $V_{IN(n+1)}$（负输入）之间的差。此时，通道（n+1）不能用于单端模式和差分模式，且不能配置转换功能。

通道 15、16 和 17 被强制为单端配置（相应的 DIFCTL[n]位始终为零），因为它们已连接到内部通道。

当通道 n 用于差分输入模式时，两个通道的输入电压应为差分信号（共模电压为 $V_{REF+}/2$），电压输入范围仍为（V_{REF-}~V_{REF+}）。

以最低有效位对齐，12 位分辨率为例，

1) 当V_{INn}为V_{REF+}，$V_{IN(n+1)}$为V_{REF-}时，通道n的转换结果为0x0FFF；
2) 当V_{INn}为V_{REF-}，$V_{IN(n+1)}$为V_{REF+}时，通道n的转换结果为0x0000；
3) 当V_{INn}为V_{REF+}/2，$V_{IN(n+1)}$为V_{REF+}/2时，通道n的转换结果为0x07FF。

D_{out}是 ADC 通道 n 的转换结果，则通道 n 转换的差分电压为：

$$V_{INn}-V_{IN(n+1)} = V_{REF+}*(2*D_{out}/4095-1) \quad (13\text{-}1)$$

13.4.5. 常规序列

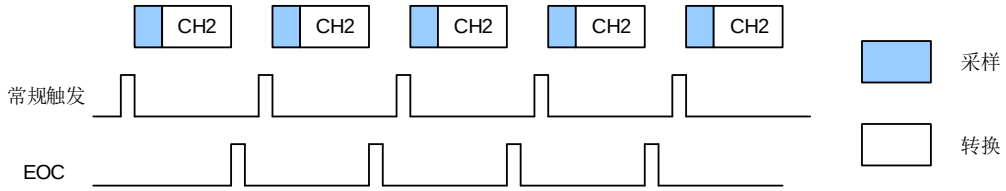
通道管理电路可以将采样通道组织成一个序列：常规序列。常规序列支持最多 16 个通道，每个通道称为常规通道。ADC_RSQ0 寄存器的 RL[3:0]位规定了整个常规序列转换序列的长度。ADC_RSQ0~ADC_RSQ2 寄存器规定了常规序列的通道选择。

13.4.6. 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 13-2. 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。

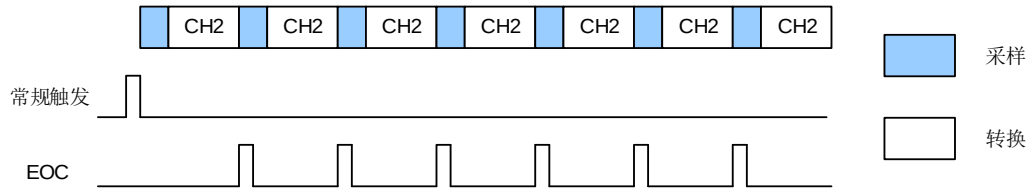
注意：当EOC置1后，需延迟一个CK_ADC再读取ADC转换结果。

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 13-3. 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。

可以使用 DMA 来传输转换数据，不需循环查询 EOC 标志位：

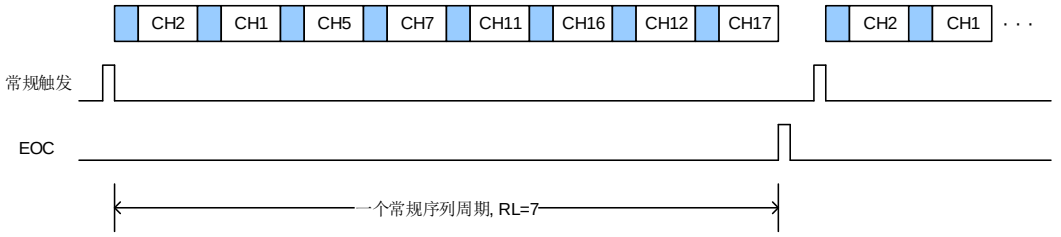
1. 设置ADC_CTL1寄存器的CTN位为1；
2. 根据模拟通道编号配置RSQ0；
3. 配置ADC_SAMPTx寄存器；
4. 如果有需要，配置ADC_CTL1寄存器的ETERC和ETSRC位；
5. 准备DMA模块，用于传输来自ADC_RDATA的数据；
6. 设置SWRCST位，或者给常规序列产生一个外部触发。

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。当常规序列通道工作在扫描运行模式下时，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，则在常规序列通道转换完之后，转换会自动重新开始。

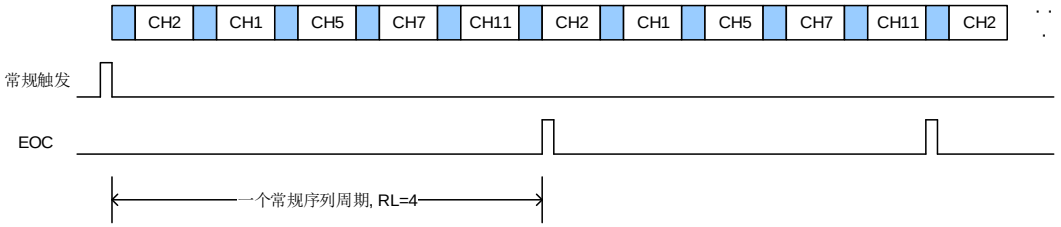
图 13-4. 扫描运行模式，且连续运行模式禁能



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 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。

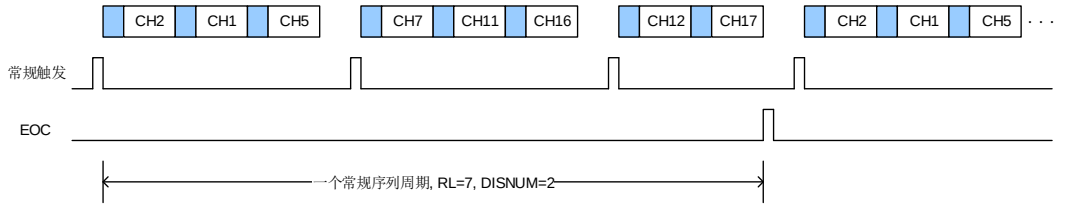
图 13-5. 扫描运行模式，连续运行模式使能



间断运行模式

对于常规序列，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 13-6. 间断运行模式



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 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。

13.4.7. 转换结果阈值监测功能

模拟看门狗 0

配置 ADC_CTL0 寄存器的 RWD0EN 位为 1，可使能常规序列的模拟看门狗 0 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 WDE0 位将置 1。若 WDE0IE 位置 1，将产生中断。ADC_WDHT0 和 ADC_WDLT0 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 RWD0EN，WD0SC 和 WD0CHSEL[4:0]位可以用来选择模拟看门狗 0 监控单一通道或者多通道。

模拟看门狗 1/2

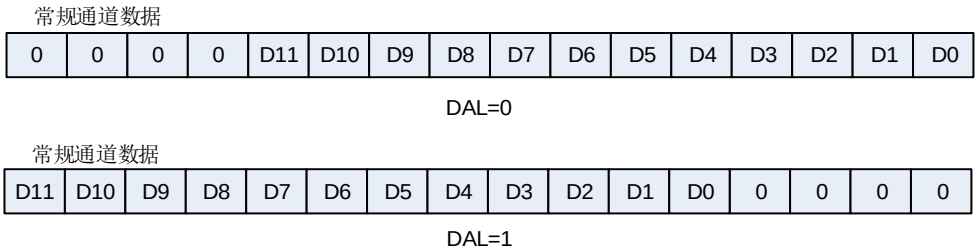
模拟看门狗 1/2 更加的灵活，可以进行单个或多个通道的看门狗功能配置。

通过配置 ADC_WD1SR 寄存器中的 AWD1CS[17:0]位域中的相应位，可以使能相应通道的模拟看门狗 1 功能，同理，可以配置看门狗 2 功能。模拟看门狗 1/2 的高/低阈值可在 ADC_WDT1 寄存器和 ADC_WDT2 寄存器中进行配置。

13.4.8. 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 13-7. 12 位数据存储模式



6 位分辨率的数据对齐不同于 12 位/10 位/8 位分辨率数据对齐，如*图 13-8. 6 位数据存储模式*所示。

图 13-8. 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

常规通道数据

0	0	0	0	0	0	0	0	D5	D4	D3	D2	D1	D0	0	0

DAL=1

13.4.9. 采样时间配置

ADC 使用若干个 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。

13.4.10. 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[3:0]位域控制。

表 13-3. ADC0 和 ADC1 的外部触发源

ETSRC[3:0]	触发源	触发类型
0000	TIMER0_CH0	硬件触发
0001	TIMER0_CH1	
0010	TIMER0_CH2	
0011	TIMER1_CH1	
0100	TIMER2_TRGO	
0101	TIMER3_CH3	
0110	EXTI11	
0111	SWRCST	软件触发
1000	SHRTIMER_ADCTRIG0	硬件触发
1001	SHRTIMER_ADCTRIG2	
1010~1111	保留	-

13.4.11. DMA 请求

DMA 请求，可以通过设置 ADC_CTL1 寄存器的 DMA 位来使能，用来传输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。

13.4.12. 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_CH16）和 V_{REFINT} 通道（ADC0_CH17）。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（具体数值请参考相关型号 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会发生线性变化，由于芯片生产过程的多样化，温度变化曲线的偏差在芯片间会有不同（最多相差 45°C）。内部温度传感器更适用于检测温度的变化，而不是用于测量绝对温度。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_CH17 输入通道。

使用温度传感器：

1. 配置温度传感器通道（ADC_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，或者由外部触发触发 ADC 转换；

4. 从 ADC 数据寄存器中读取并计算温度传感器数据 $V_{temperature}$，并由下面公式计算出实际温度：

$$温度(°C) = \{(V25 - V_{temperature}) / Avg_Slope\} + 25$$

V25：内部温度传感器在 25°C 下的电压，典型值请参考数据手册。

Avg_Slope：温度与内部温度传感器电压曲线的均值斜率，典型值请参考数据手册。

13.4.13. 可编程分辨率（DRES）

对 ADC_OVSAMPCTL 寄存器中的 DRES[1:0]位进行配置可以设置 ADC 分辨率。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 13-4. 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 13-4. 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

13.4.14. 片上硬件过采样

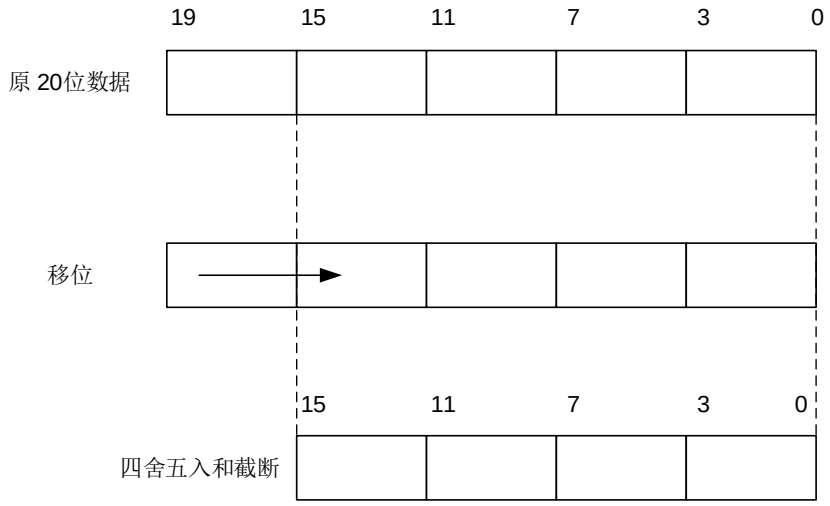
片上硬件过采样单元执行数据预处理以减轻 CPU 负担。它能够处理多个转换，并将多个转换的结果取平均，得出一个 16 位宽的数据。其结果根据如下公式计算得出，其中 N 和 M 的值可以被调整，过采样单元可以通过设置 ADC_OVSAMPCTL 寄存器的 OVSE 位来使能，它是以降低数据输出率为代价，换取较高的数据分辨率。$D_{out}(n)$是指 ADC 输出的第 n 个数字信号：

$$Result=\frac{1}{M}*\sum_{n=0}^{N-1}D_{out}(n) \quad (13\text{-}2)$$

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 13-9. 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 13-10. 右移 5 位和取整的数例*所示为将原始 20 位的累积数值处理成 16 位结果值。

图 13-10. 右移 5 位和取整的数例

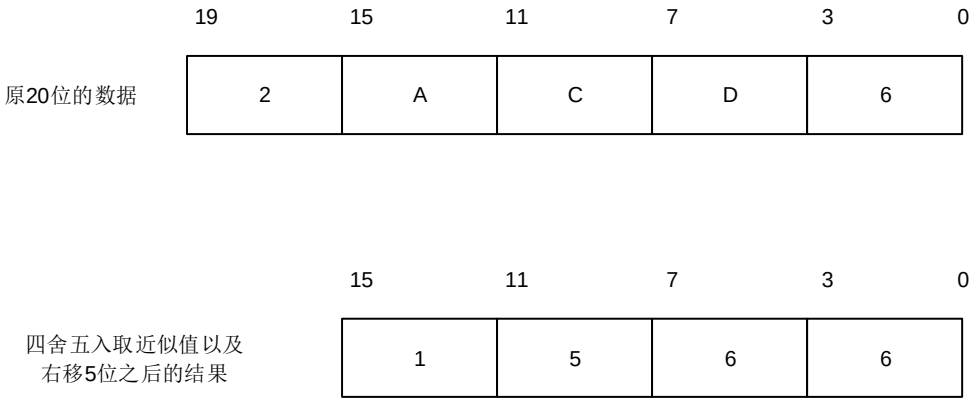


*表 13-5. 不同 N 和 M 组合的最大输出值（灰色值表示截断）*给出了 N 和 M 各种组合的数据格式，初始转换值为 0xFFF。

表 13-5. 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	0x0800	0x0400	0x0200	0x0100	0x0080	0x0040	0x0020
4x	0x3FFC	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080	0x0040
8x	0x7FF8	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080
16x	0xFFF0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100
32x	0x1FFE0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200
64x	0x3FFC0	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400
128x	0x7FF80	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800
256x	0xFFF00	0xFF00	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF

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：

$$N \times t_{ADC} = N \times (t_{SMPL} + t_{CONV}) \quad (13\text{-}3)$$

13.5. 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，ADC0 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 13-6. ADC 同步模式表*所示。

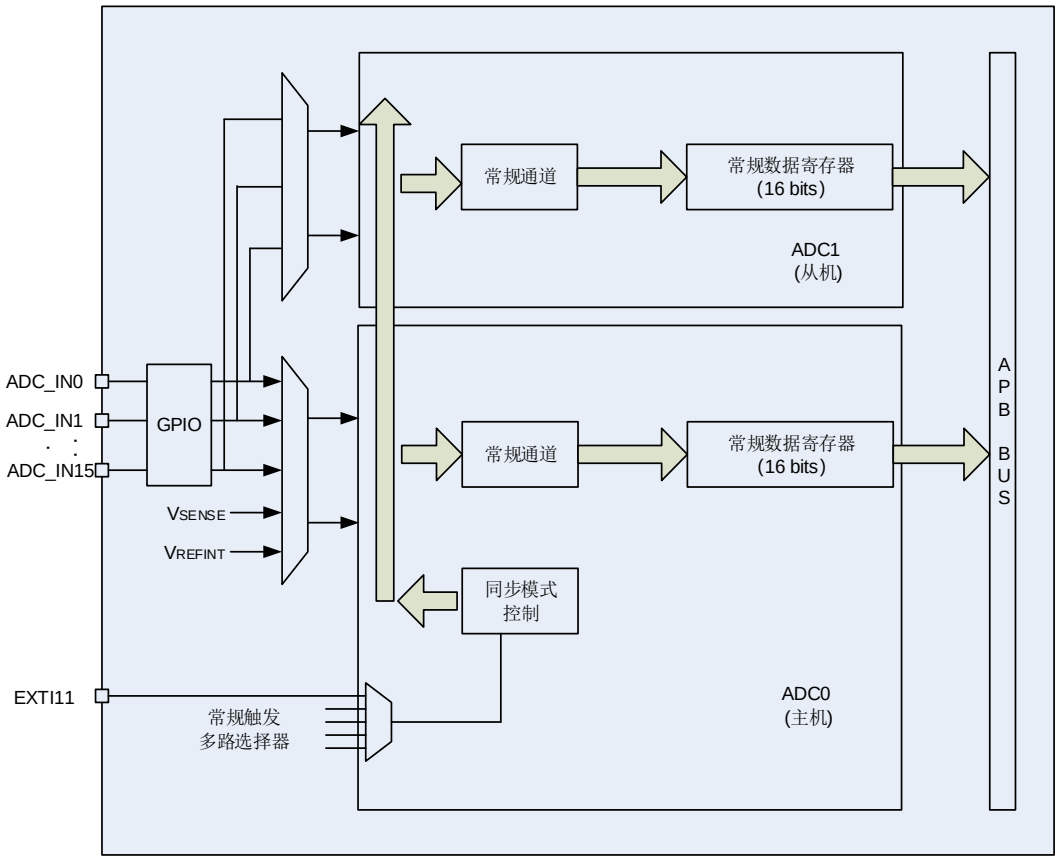
表 13-6. ADC 同步模式表

SYNCM[2: 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 13-11. ADC 同步框图*所示。

图 13-11. ADC 同步框图



13.5.1. 独立模式

在这种模式下，每个 ADC 都独立工作，互不干扰。

13.5.2. 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（由 ADC_CTL1 寄存器的 ETSRC[2:0]决定），ADC1 常规序列通道配置为软件触发模式。

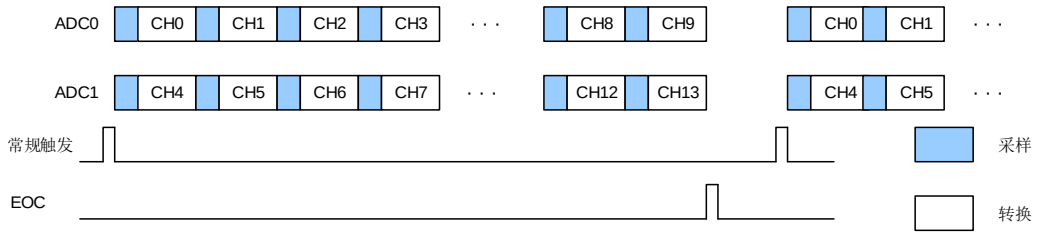
在 ADC0 或 ADC1 的转换事件结束时，即 ADC0 或 ADC1 的常规序列转换完毕，会产生一个 EOC 中断（如果某个 ADC 中断使能）。常规并行模式请参考*图 13-12. 基于 10 个通道的常规并行模式*。

32 位 ADC_RDATA 寄存器（[15: 0]位域用于保存 ADC0 常规组采样数据，[31: 16]位域用于保存 ADC1 常规组采样数据），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 13-12. 基于 10 个通道的常规并行模式



13.5.3. 常规快速交叉模式

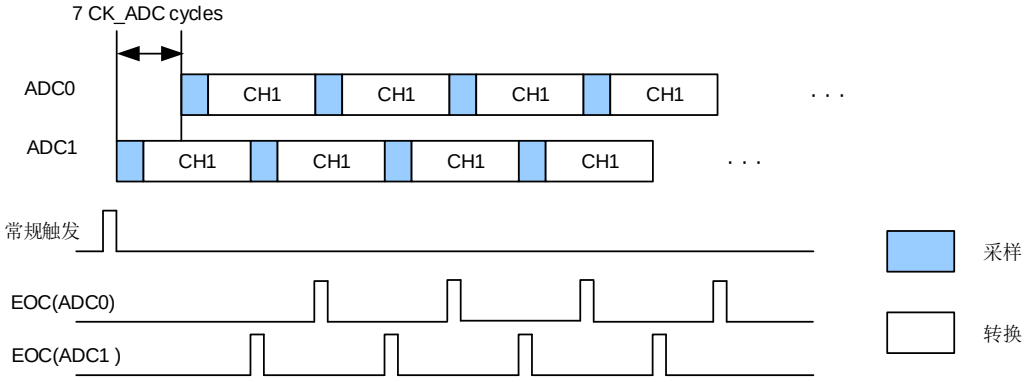
快速交叉模式适用于两个 ADC 的常规序列采样同一个通道，，外部触发来源于 ADC0 常规序列（由 ADC_CTL1 寄存器的 ETSRC[2:0]决定）。当触发产生时，ADC1 立刻启动，而 ADC0 在 7 个 ADC 时钟周期后启动。

如果 ADC0 和 ADC1 的 CTN 位被置位，所选的规常规序列在两个 ADC 中被不停的转换，如 *图 13-13. 常规序列上的快速交叉模式（两个 ADC 的 CTN=1）*。

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 模块常规通道的采样时间都应小于 7 个 ADC 时钟周期。

图 13-13. 常规序列上的快速交叉模式（两个 ADC 的 CTN=1）



13.5.4. 常规慢速交叉模式

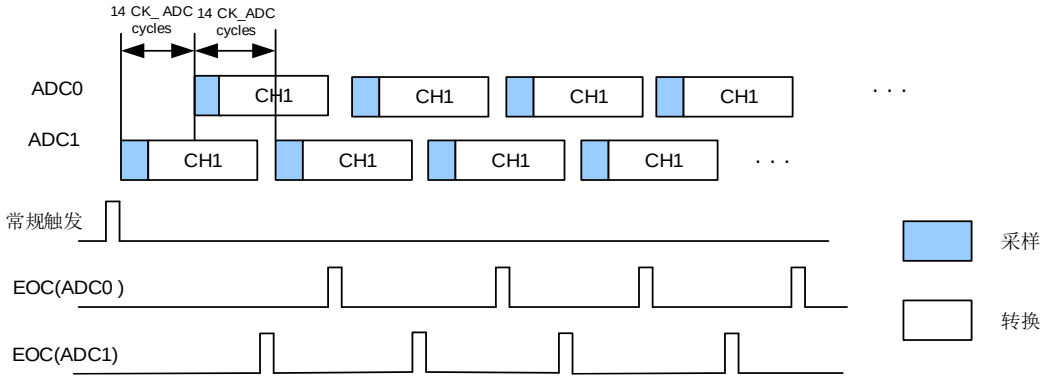
此模式应用于两个 ADC 的常规序列（通常一个常规通道），外部触发来源于 ADC0 常规序列（由 ADC_CTL1 寄存器的 ETSRC[2:0]决定）。当触发产生时，ADC1 立刻启动，而 ADC0 在 14 个 ADC 时钟周期后启动，在 ADC0 启动后的 14 个时钟周期，ADC1 再次启动。

在这种模式下，不能使用连续运行模式，因为在这种模式下所选的常规通道在两个 ADC 中被不停的转换，如*图 13-14. 常规序列上的慢速交叉模式*。

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）。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 13-14. 常规序列上的慢速交叉模式



13.6. 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。

单独的中断使能位可使得使用更灵活。

ADC0 和 ADC1 都映射到同一个中断向量 IRQ[18]。

13.7. ADC 寄存器

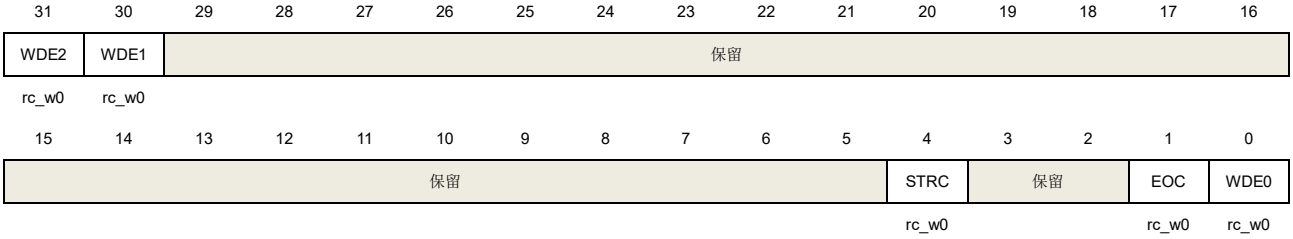
ADC0 基地址：0x4001 2400

ADC1 基地址：0x4001 2800

13.7.1. 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	描述
31	WDE2	模拟看门狗2事件标志 0：模拟看门狗2事件没有发生 1：模拟看门狗2事件发生 转换电压超过ADC_WDLT2寄存器设定的阈值时由硬件置1，软件写0清除。
30	WDE1	模拟看门狗1事件标志 0：模拟看门狗1事件没有发生 1：模拟看门狗1事件发生 转换电压超过ADC_WDLT1寄存器设定的阈值时由硬件置1，软件写0清除。
29:5	保留	必须保持复位值。
4	STRC	常规序列转换开始标志 0：转换没有开始 1：转换开始 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 0：转换没有结束 1：转换结束 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	WDE0	模拟看门狗0事件标志 0：模拟看门狗0事件没有发生 1：模拟看门狗0事件发生

转换电压超过ADC_WDLT0和ADC_WDHT0寄存器设定的阈值时由硬件置1，软件写0清除。

13.7.2. 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
WDE2IE	WDE1IE	保留						RWD0EN	保留			SYNCM[3:0]			
rw	rw							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	WD0SC	SM	保留	WDE0IE	EOCIE	WD0CHSEL[4:0]				
rw				rw		rw	rw		rw	rw		rw			

位/位域	名称	描述
31	WDE2IE	WDE2中断使能 0：中断禁止 1：中断使能
30	WDE1IE	WDE1中断使能 0：中断禁止 1：中断使能
29:24	保留	必须保持复位值
23	RWD0EN	常规序列模拟看门狗0使能 0：模拟看门狗0禁止 1：模拟看门狗0使能
22:20	保留	必须保持复位值。
19:16	SYNCM[3:0]	同步模式选择 这些位用于运行模式选择。 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 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 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 0：常规序列间断运行模式禁止

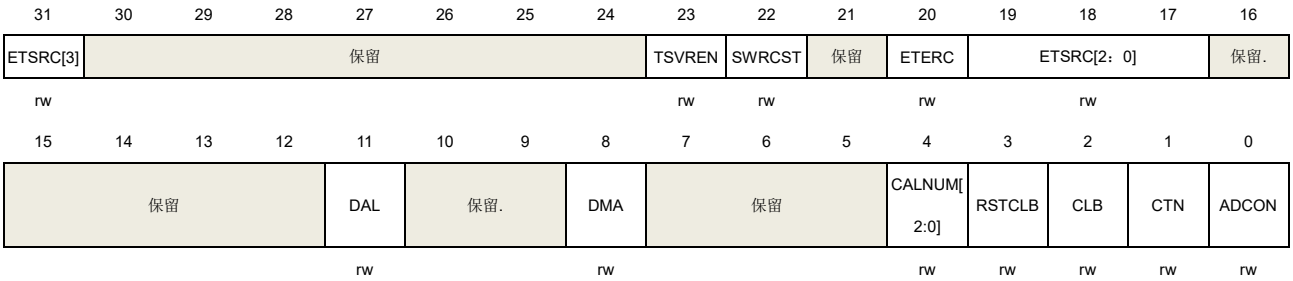
		1：常规序列间断运行模式使能
10	保留	必须保持复位值。
9	WD0SC	扫描运行模式下，模拟看门狗0在通道的配置 0：模拟看门狗0在所有通道有效 1：模拟看门狗0在单通道有效
8	SM	扫描模式 0：扫描运行模式禁止 1：扫描运行模式使能
7	保留	必须保持复位值。
6	WDE0IE	WDE0中断使能 0：中断禁止 1：中断使能
5	EOCIE	EOC中断使能 0：EOC中断禁止 1：EOC中断使能
4:0	WD0CHSEL[4:0]	模拟看门狗0通道选择 00000：ADC通道0 00001：ADC通道1 00010：ADC通道2 00011：ADC通道3 00100：ADC通道4 00101：ADC通道5 00110：ADC通道6 00111：ADC通道7 01000：ADC通道8 01001：ADC通道9 01010：ADC通道10 01011：ADC通道11 01100：ADC通道12 01101：ADC通道13 01110：ADC通道14 01111：ADC通道15 10000：ADC通道16 10001：ADC通道17 其他值保留。 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}。ADC1的模拟输入通道16和通道17内部都连接到V_{SSA}。

13.7.3. 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。



位/位域	名称	描述
31	ETSRC[3]	常规序列外部触发选择，具体请参考ETSRC[2:0]的描述。
30:24	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 1：ADC0的通道16和17使能
22	SWRCST	常规序列转换开始 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 0：外部触发禁止 1：外部触发使能
19:17	ETSRC[2:0]	常规序列外部触发选择 ETSRC[2:0]与ETSRC[3]共同确定常规序列外部触发源。 对于ADC0与ADC1： 0000：TIMER0_CH0 0001：TIMER0_CH1 0010：TIMER0_CH2 0011：TIMER1_CH1 0100：TIMER2_TRGO 0101：TIMER3_CH3 0110：中断线11 0111：软件触发 1000：SHRTIMER_ADCTRIG0 1001：SHRTIMER_ADCTRIG2 其他值保留。
16:12	保留	必须保持复位值。
11	DAL	数据对齐 0：最低有效位（LSB）对齐

		1：最高有效位（MSB）对齐
10:9	保留	必须保持复位值。
8	DMA	DMA请求使能 0：DMA请求禁止 1：DMA请求使能
6:4	CALNUM[2:0]	校准次数 这些位用于配置ADC的校准次数。 000：1次 001：2次 010：4次 011：8次 100：16次 101：32次 其他值保留。
7	保留	必须保持复位值
3	RSTCLB	校准复位 软件置位，在校准寄存器初始化后该位硬件清零。 0：校准寄存器初始化结束. 1：校准寄存器初始化开始
2	CLB	ADC校准 0：校准结束 1：校准开始
1	CTN	连续模式 0：禁止连续运行模式 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。该位置1后，不改变寄存器的其他位仅对该位写1，将开启转换。 0：禁能ADC并掉电 1：使能ADC

13.7.4. 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	描述
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 000：通道采样时间为1.5周期 001：通道采样时间为7.5周期 010：通道采样时间为13.5周期 011：通道采样时间为28.5周期 100：通道采样时间为41.5周期 101：通道采样时间为55.5周期 110：通道采样时间为71.5周期 111：通道采样时间为239.5周期

13.7.5. 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	

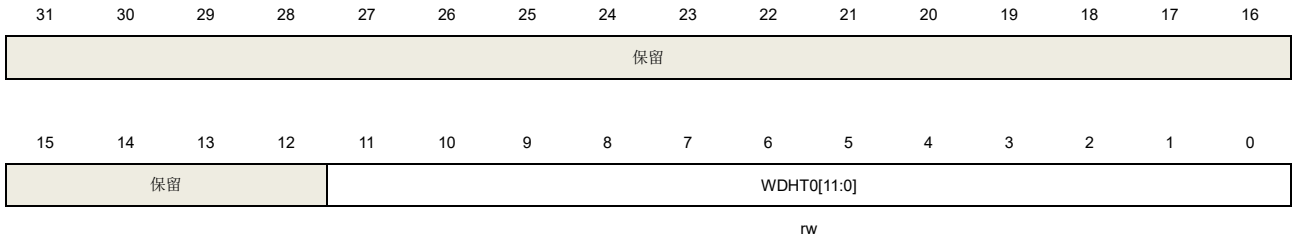
位/位域	名称	描述
31:30	保留	必须保持复位值
29:27	SPT9[2:0]	参考SPT0[2:0]的描述
26:24	SPT8[2:0]	参考SPT0[2:0]的描述

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	SPT1[2:0]	参考SPT0[2:0]的描述
2:0	SPT0[2:0]	通道采样时间 000：通道采样时间为1.5周期 001：通道采样时间为7.5周期 010：通道采样时间为13.5周期 011：通道采样时间为28.5周期 100：通道采样时间为41.5周期 101：通道采样时间为55.5周期 110：通道采样时间为71.5周期 111：通道采样时间为239.5周期

13.7.6. 看门狗 0 高阈值寄存器（ADC_WDHT0）

地址偏移：0x24
复位值：0x0000 0FFF

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	WDHT0[11:0]	模拟看门狗0高侧阈值 这些位定义了模拟看门狗0的高侧阈值。

13.7.7. 看门狗 0 低阈值寄存器（ADC_WDLT0）

地址偏移：0x28
复位值：0x0000 0000

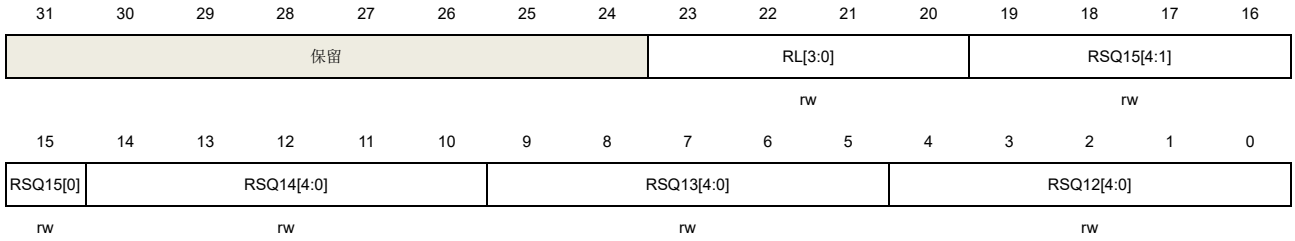
该寄存器只能按字（32 位）访问。

位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	WDLT0[11:0]	模拟看门狗0低侧阈值 这些位定义了模拟看门狗0的低侧阈值。

13.7.8. 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	描述
31:24	保留	必须保持复位值
23:20	RL[3:0]	常规序列长度 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

13.7.9. 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。

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	描述
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

13.7.10. 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			

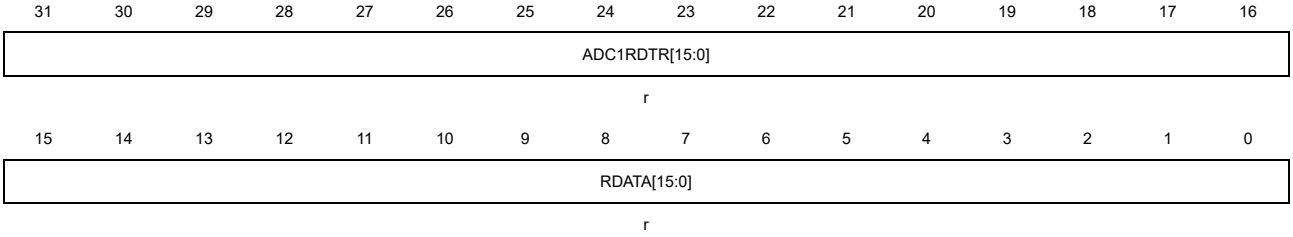
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSQ3[0]	RSQ2[4:0]					RSQ1[4:0]					RSQ0[4:0]				
rw	rw					rw					rw				

位/位域	名称	描述
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。

13.7.11. 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。

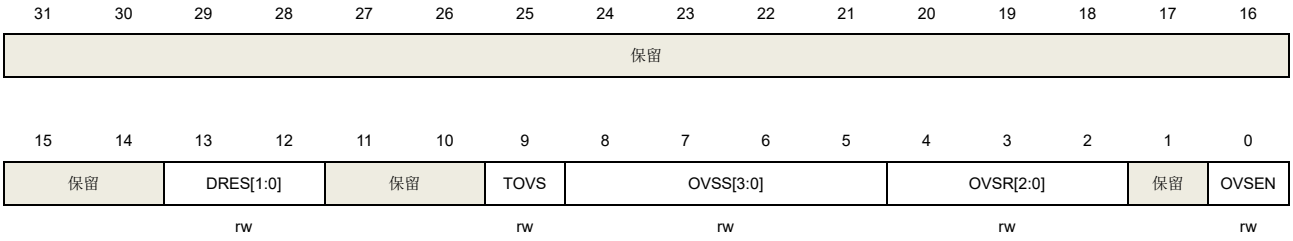


位/位域	名称	描述
31:16	ADC1RDTR[15:0]	ADC1常规通道数据 ADC0：在同步模式下，这些位包含着ADC1的常规通道数据 这些位只在ADC0中使用。
15:0	RDATA[15:0]	常规通道数据 这些位包含了常规通道的转换结果，只读。

13.7.12. 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。



位/位域	名称	描述
31:14	保留	必须保持复位值。
13:12	DRES[1:0]	ADC分辨率 00：12位 01：10位 10：8位 11：6位
11:10	保留	必须保持复位值
9	TOVS	触发过滤采样 该位通过软件设置和清除。

0：在一个触发之后，对一个通道连续进行过采样转换
1：在一个触发之后，对一个通道只进行一次过采样转换
注意：当ADCON= 0时，才允许软件写该位（确定没有转换正在进行）。

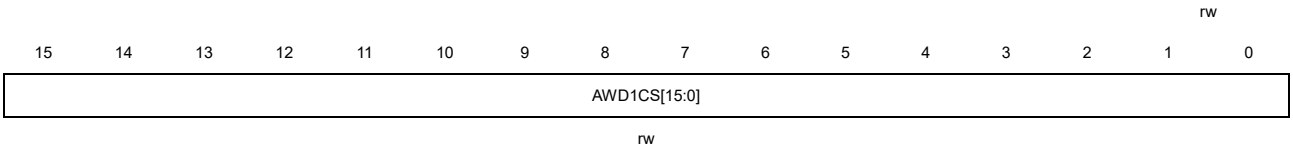
8:5	OVSS[3:0]	过滤采样移位 该位通过软件设置和清除。 0000：不移位 0001：移1位 0010：移2位 0011：移3位 0100：移4位 0101：移5位 0110：移6位 0111：移7位 1000：移8位 其他保留 注意：当ADCON= 0时，才允许软件写该位（确定没有转换正在进行）。
4:2	OVSR[2:0]	过采样率 这些位定义了过采样率的大小。 000：2x 001：4x 010：8x 011：16x 100：32x 101：64x 110：128x 111：256x 注意：当ADCON= 0时，才允许软件写该位（确定没有转换正在进行）。
1	保留	必须保持复位值
0	OVSEN	过滤采样使能 该位通过软件和设置和清除。 0：过滤采样失能 1：过滤采样使能 注意：当ADCON= 0时，才允许软件写该位（确定没有转换正在进行）。

13.7.13. 看门狗 1 通道选择寄存器（ADC_WD1SR）

地址偏移：0xA0
复位值：0x00000000

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														AWD1CS[17:16]	



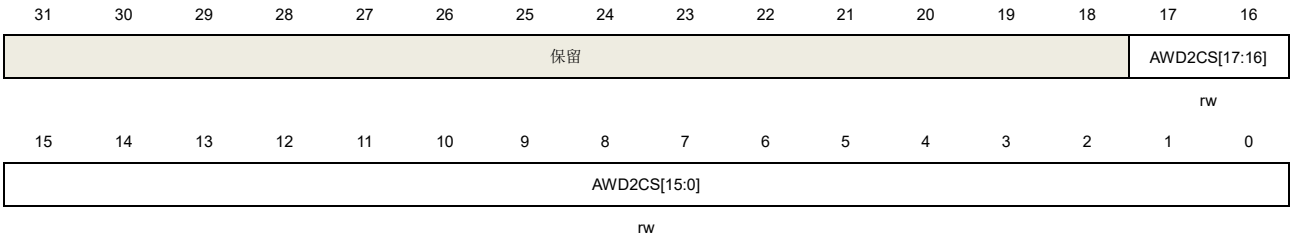
位/位域	名称	描述
31:18	保留	必须保持复位值
17:0	AWD1CS[17:0]	模拟看门狗1通道选择 这些位由软件置位和复位，它们使能并选择要由模拟看门狗1保护的输入通道。 AWD1CS[n] = 0：ADC模拟输入通道n不由模拟看门狗1保护。 AWD1CS[n] = 1：ADC模拟输入通道n由模拟看门狗1保护。 当AWD1CH[17:0] = 000..0，模拟看门狗1禁能。 注意： 1）通过AWD1CS位域配置的模拟看门狗1功能的通道，必须是ADC_RSQn寄存器和ADC_ISQ寄存器中配置的通道； 2）只有在ADC禁能（ADCON=0）时，才能软件写这些位。

13.7.14. 看门狗 2 通道选择寄存器（ADC_WD2SR）

地址偏移：0xA4

复位值：0x00000000

该寄存器只能按字（32 位）访问。



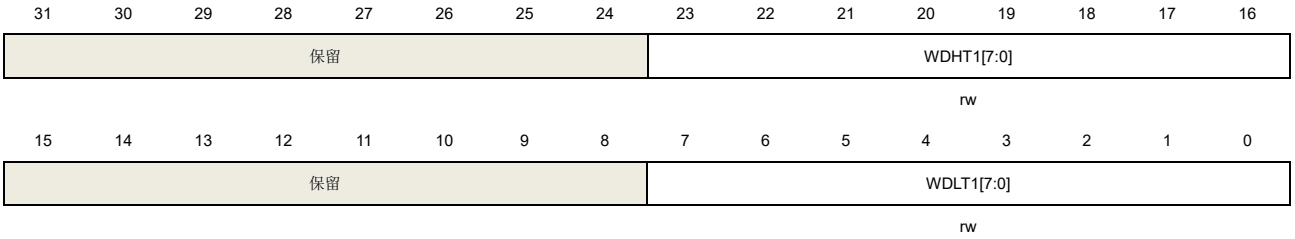
位/位域	名称	描述
31:18	保留	必须保持复位值
17:0	AWD2CS[17:0]	模拟看门狗2通道选择 这些位由软件置位和复位，它们使能并选择要由模拟看门狗2保护的输入通道。 AWD2CS[n] = 0：ADC模拟输入通道n不由模拟看门狗2保护。 AWD2CS[n] = 1：ADC模拟输入通道n由模拟看门狗2保护。 当AWD2CH[17:0] = 000..0，模拟看门狗2禁能。 注意： 1) 通过AWD2CS位域配置模拟看门狗2功能的通道，必须是ADC_RSQn寄存器和ADC_ISQ寄存器中配置的通道； 2) 只有在ADC禁能（ADCON=0）时，才能软件写这些位。

13.7.15. 看门狗 1 阈值寄存器（ADC_WDT1）

地址偏移：0xA8

复位值：0x00FF 0000

该寄存器只能按字（32 位）访问。



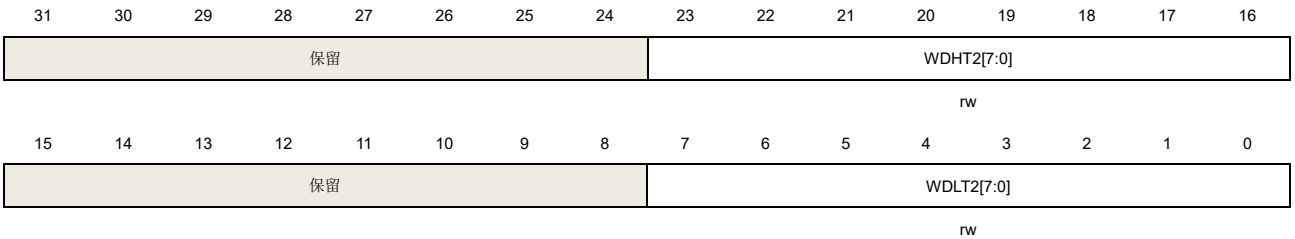
位/位域	名称	描述
31:24	保留	必须保持复位值
23:16	WDHT1[7:0]	模拟看门狗1高侧阈值 这些位定义了模拟看门狗1的高侧阈值。 注意：只有在ADC禁能（ADCON=0）时，才能软件写这些位。
15:8	保留	必须保持复位值
7:0	WDLT1[7:0]	模拟看门狗1低侧阈值 这些位定义了模拟看门狗1的低侧阈值。 注意：只有在ADC禁能（ADCON=0）时，才能软件写这些位。

13.7.16. 看门狗 2 阈值寄存器（ADC_WDT2）

地址偏移：0xAC

复位值：0x00FF 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:24	保留	必须保持复位值
23:16	WDHT2[7:0]	模拟看门狗2高阈值 这些位定义了模拟看门狗2的高阈值。 注意：只有在ADC禁能（ADCON=0）时，才能软件写这些位。
15:8	保留	必须保持复位值

7:0	WDLT2[7:0]	模拟看门狗2低侧阈值 这些位定义了模拟看门狗2的低侧阈值。 注意：只有在ADC禁能（ADCON=0）时，才能软件写这些位。

13.7.17. 差分模式控制寄存器（ADC_DIFCTL）

地址偏移：0xB0

复位值：0x00000000

该寄存器只能按字（32 位）访问。

位/位域	名称	描述
31:18	保留	必须保持复位值
17:15	DIFCTL[17:15]	差分模式通道17..15。 这些位只读，通道17..15被强制为单端输入模式（连接到单端I/O端口或内部通道）。
14:0	DIFCTL[14:0]	差分模式通道14..0。 这些位用于配置通道用于单端输入模式还是差分模式。 DIFCTL[n] = 0：ADC模拟输入通道n配置为单端模式。 DIFCTL[n] = 1：ADC模拟输入通道n配置为差分模式。 注意：只有在ADC禁能（ADCON=0）时，才能软件写这些位。

14. 数模转换器（DAC）

14.1. 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。

14.2. 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 功能与欠载检测；
- 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}；
- 输出 FIFO；
- 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 14-1. DAC 结构框图*为 DAC 的结构框图，*表 14-1. DAC 引脚*给出了引脚描述。

图 14-1. DAC 结构框图

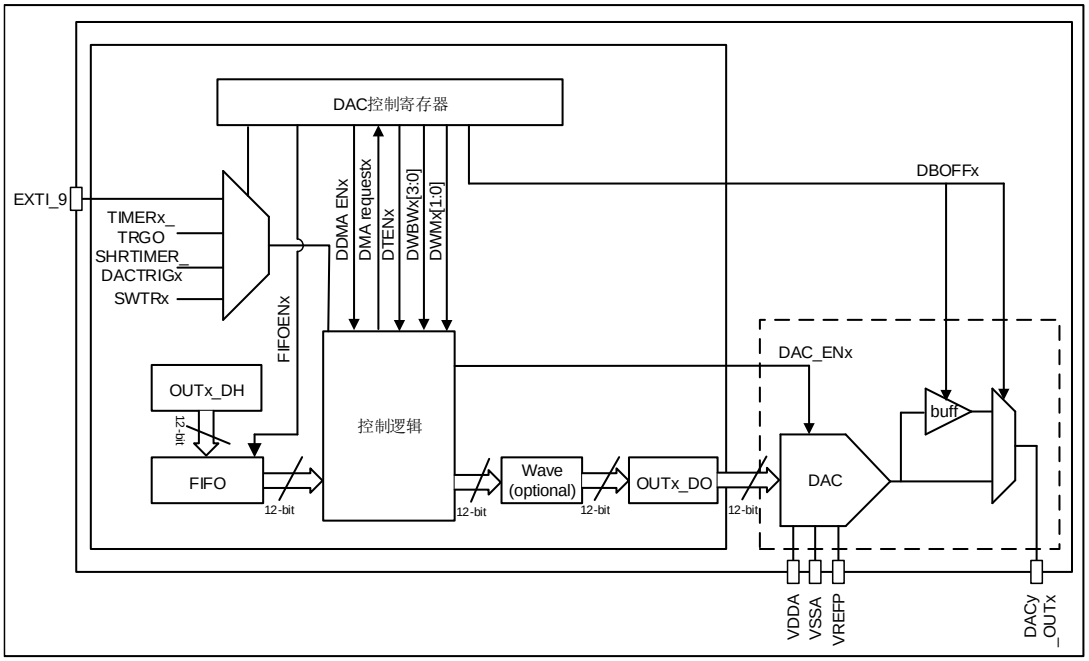


表 14-1. DAC 引脚

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 14-2. 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 TIMER2_TRGO TIMER3_TRGO TIMER4_TRGO TIMER5_TRGO TIMER6_TRGO	
SHRTIMER 触发信号	SHRTIMER_DACTRIG0 SHRTIMER_DACTRIG1 SHRTIMER_DACTRIG2	

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

14.3. 功能描述

14.3.1. 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。

14.3.2. 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。

14.3.3. 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。

14.3.4. 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 14-3. DAC 外部触发*所示。

表 14-3. DAC 外部触发

DTSELx[3:0]	Trigger Source	Trigger Type
4b'0000	TIMER5_TRGO	硬件触发
4b'0001	TIMER2_TRGO	
4b'0010	TIMER6_TRGO	
4b'0011	TIMER4_TRGO	
4b'0100	TIMER1_TRGO	
4b'0101	TIMER3_TRGO	
4b'0110	EXTI_9	
4b'0111	SWTR	软件触发
4b'1000	SHRTIMER_DACTRIG0	硬件触发
4b'1001	SHRTIMER_DACTRIG1	
4b'1010	SHRTIMER_DACTRIG2	
4b'1011~1111	保留	保留

TIMERx_TRGO 信号是由定时器生成的，SHRTIMER_DACTRIGx 信号是由高精度定时器 SHRTIMER 提供，而软件触发是通过设置 DAC_SWT 寄存器的 SWTRx 位生成的。

14.3.5. 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。

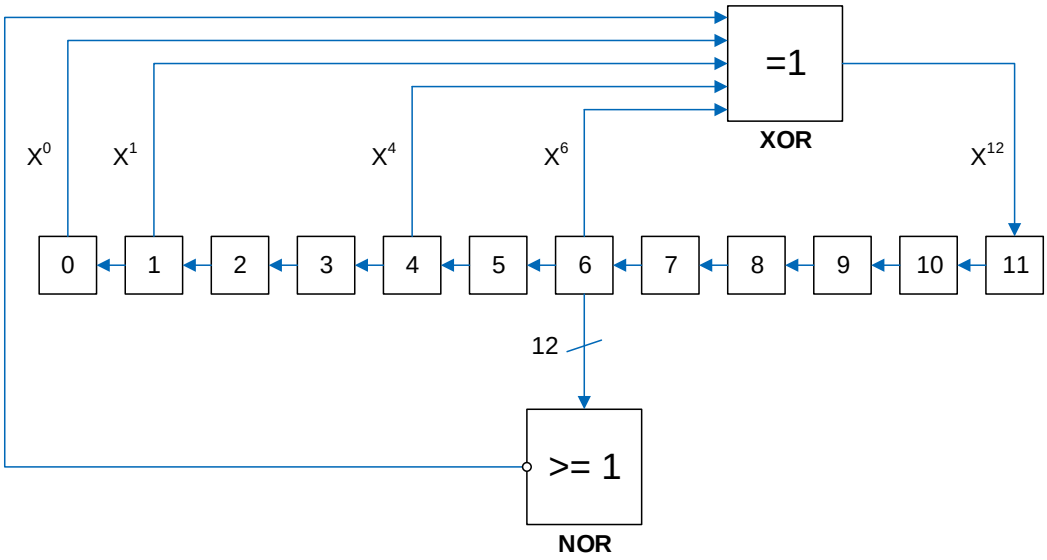
14.3.6. DAC 噪声波

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。

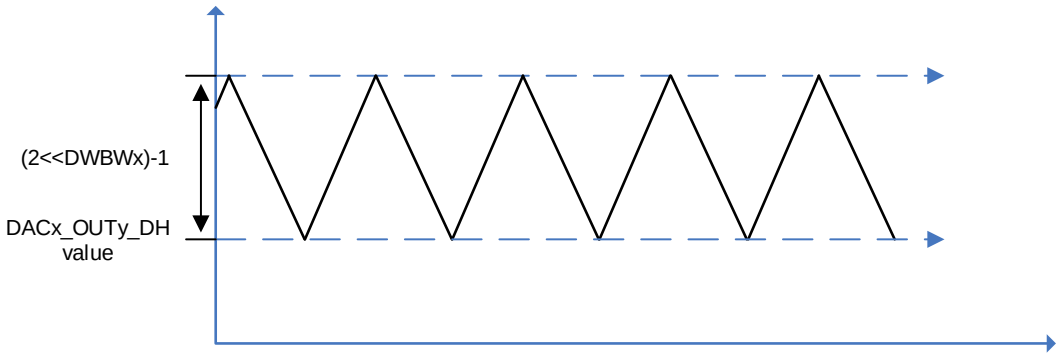
LFSR 噪声模式：在 DAC 控制逻辑中有一个线性反馈移位寄存器（LFSR）。在此模式下，LFSR 的值与 OUTx_DH 值相加后，被写入到 DAC 数据输出寄存器（DAC_OUTx_DO）。当配置的 DAC 噪声波位宽小于 12 时，LFSR 的值等于 LFSR 寄存器最低的 DWBWx 位，高位被屏蔽。

图 14-2. 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 14-3. DAC 三角噪声模式生成的波形



14.3.7. DAC 输出电压

DAC 引脚上的模拟输出电压取决于下面的等式：

$$V_{DAC_OUT}=V_{REFP}*OUTx_DO/4096 \quad (14\text{-}1)$$

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}。

14.3.8. 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。

如果在前一个请求响应之前第二个外部触发到达，则不响应新到的触发请求，并且发生欠载错误事件。DAC_STAT0 寄存器中的 DDUDRx 位置 1，如果 DAC_CTL0 寄存器中的 DDUDRIEx 位置 1，则会产生中断。

14.3.9. 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。

14.3.10. DAC 输出 FIFO

在数据保持寄存器和输出寄存器之间有一个 4 位深度的数据 FIFO。通过将 DAC_CTL1 寄存器中的 FIFOENx 位置 1，可以使能输出数据 FIFO。

当 FIFOENx 位置 1 时，需要软件设置 DDMAENx=0（DAC_OUTx DMA 模式禁能）和 DTENx=1（DAC_OUTx 外部触发使能）。

当数据 FIFO 中数据满了时，FIFO 为满载状态，FIFOFx 标志位会置 1；当数据 FIFO 中数据为空时，FIFO 为空载状态，FIFOEx 标志位会置 1。

当数据 FIFO 中数据已满，但没有触发到来时，过载标志位 FIFOOVRx 置 1；当数据 FIFO 为空，但有触发到来时，欠载标志位 FIFOUDRx 置 1。如果设置了相应的中断使能位，则在发生过载或欠载时将产生中断。

通过读取 DAC_STAT1 寄存器内容，可以获取数据 FIFO 的状态（满载、空载、过载和欠载）。

14.4. DAC 寄存器

DAC0 基地址：0x4000 7400

14.4.1. 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	DTSEL1[3]	DDUDRIE1	DDMAEN1	DWBW1[3:0]				DWM1[1:0]		DTSEL1[2:0]			DTEN1	DBOFF1	DEN1
		rw	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	DTSEL0[3]	DDUDRIE0	DDMAEN0	DWBW0[3:0]				DWM0[1:0]		DTSEL0[2:0]			DTEN0	DBOFF0	DEN0
		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	DTSEL1[3]	DACx_OUT1 触发源选择的位[3]，参考 DTSEL1[2:0]。
29	DDUDRIE1	DACx_OUT1 DMA 欠载中断使能 0: DACx_OUT1 DMA 欠载中断禁能 1: DACx_OUT1 DMA 欠载中断使能
28	DDMAEN1	DACx_OUT1 DMA 使能 0：DACx_OUT1 DMA 模式禁能 1：DACx_OUT1 DMA模式使能
27:24	DWBW1[3:0]	DACx_OUT1 噪声波位宽 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。 0000：波形信号的位宽为 1 0001：波形信号的位宽为 2 0010：波形信号的位宽为 3 0011：波形信号的位宽为 4 0100：波形信号的位宽为 5 0101：波形信号的位宽为 6 0110：波形信号的位宽为 7 0111：波形信号的位宽为 8 1000：波形信号的位宽为 9 1001：波形信号的位宽为 10

		1010：波形信号的位宽为 11 ≥1011：波形信号的位宽为12
23:22	DWM1[1:0]	DACx_OUT1 噪声波模式 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。 00：波形生成禁能 01：LFSR 噪声模式 1x：三角噪声模式
21:19	DTSEL1[2:0]	DACx_OUT1 触发选择 这些位与 DTSEL1[3]位合并，在外部触发使能(DTEN1=1)时，选择触发 DAC_OUT1 的外部事件。 0000：TIMER5 TRGO 0001：TIMER2 TRGO 0010：TIMER6 TRGO 0011：TIMER4 TRGO 0100：TIMER1 TRGO 0101：TIMER3 TRGO 0110：外部中断线9 0111：软件触发 1000：SHRTIMER_DACTRIG0 1001：SHRTIMER_DACTRIG1 1010：SHRTIMER_DACTRIG2 1011~1111：保留
18	DTEN1	DACx_OUT1 触发使能 0：DACx_OUT1 触发禁能 1：DACx_OUT1触发使能
17	DBOFF1	DACx_OUT1输出缓冲区关闭 0：DACx_OUT1输出缓冲区打开，以降低输出阻抗，提高驱动能力 1：DACx_OUT1输出缓冲区关闭
16	DEN1	DACx_OUT1 使能 0：DACx_OUT1 禁能 1：DACx_OUT1使能
15	保留	必须保持复位值
14	DTSEL0[3]	DACx_OUT0 触发源选择的位[3]，参考 DTSEL0[2:0]。
13	DDUDRIE0	DACx_OUT0 DMA 欠载中断使能 0: DACx_OUT0 DMA 欠载中断禁能 1: DACx_OUT0 DMA 欠载中断使能
12	DDMAEN0	DACx_OUT0 DMA 使能 0：DACx_OUT0 DMA 模式禁能

		1：DACx_OUT0 DMA模式使能
11:8	DWBW0[3:0]	DACx_OUT0 噪声波位宽 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。 0000：波形信号的位宽为 1 0001：波形信号的位宽为 2 0010：波形信号的位宽为 3 0011：波形信号的位宽为 4 0100：波形信号的位宽为 5 0101：波形信号的位宽为 6 0110：波形信号的位宽为 7 0111：波形信号的位宽为 8 1000：波形信号的位宽为 9 1001：波形信号的位宽为 10 1010：波形信号的位宽为 11 ≥1011：波形信号的位宽为12
7:6	DWM0[1:0]	DACx_OUT0 噪声波模式 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。 00：波形生成禁能 01：LFSR 噪声模式 1x：三角噪声模式
5:3	DTSEL0[2:0]	DACx_OUT0 触发选择 这些位与 DTSEL0[3]位合并，在外部触发使能(DTEN0=1)时，选择触发 DAC_OUT0 的外部事件。 0000：TIMER5 TRGO 0001：TIMER2 TRGO 0010：TIMER6 TRGO 0011：TIMER4 TRGO 0100：TIMER1 TRGO 0101：TIMER3 TRGO 0110：外部中断线9 0111：软件触发 1000：SHRTIMER_DACTRIG0 1001：SHRTIMER_DACTRIG1 1010：SHRTIMER_DACTRIG2 1011~1111：保留
2	DTEN0	DACx_OUT0 触发使能 0：DACx_OUT0 触发禁能 1：DACx_OUT0触发使能
1	DBOFF0	DACx_OUT0输出缓冲区关闭

		0：DACx_OUT0输出缓冲区打开，以降低输出阻抗，提高驱动能力 1：DACx_OUT0输出缓冲区关闭
0	DEN0	DACx_OUT0 使能 0：DACx_OUT0 禁能 1：DACx_OUT0使能

14.4.2. 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。 0：软件触发禁能 1：软件触发使能
0	SWTR0	DACx_OUT0 软件触发，由硬件清除。 0：软件触发禁能 1：软件触发使能

14.4.3. 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											

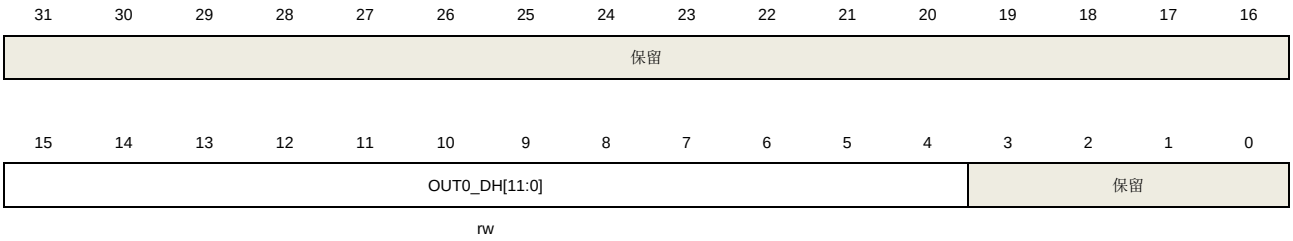
位/位域	名称	描述

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 这些位指定了将由 DACx_OUT0 转换的数据。

14.4.4. 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。

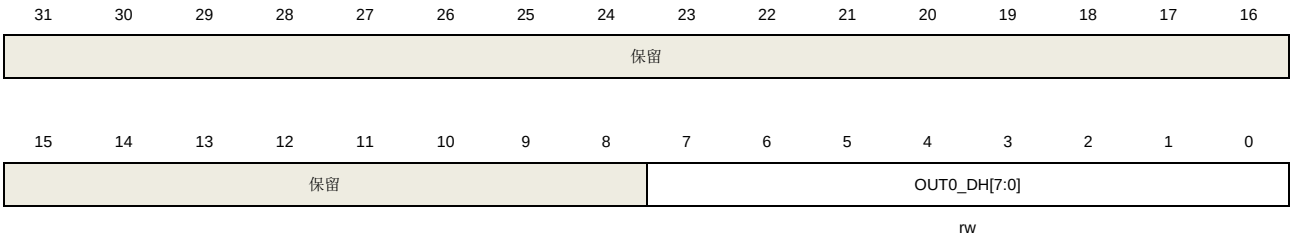


位/位域	名称	描述
31:16	保留	必须保持复位值
15:4	OUT0_DH[11:0]	DACx_OUT0 12 位左对齐数据 这些位指定了将由 DACx_OUT0 转换的数据。
3:0	保留	必须保持复位值

14.4.5. 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。



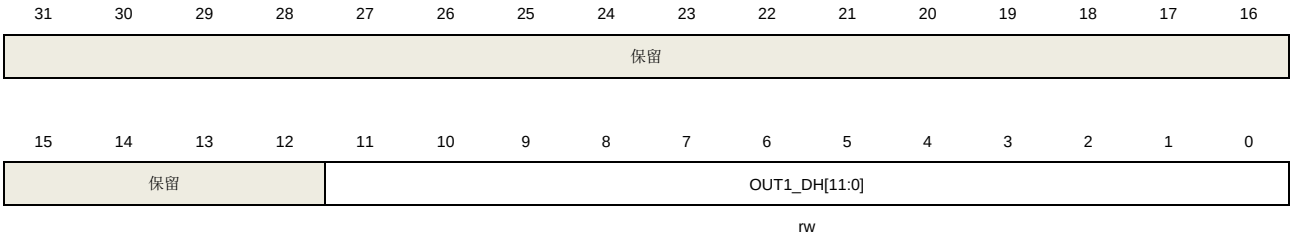
位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	OUT0_DH[7:0]	DACx_OUT0 8 位右对齐数据 这些位指定了将由 DACx_OUT0 转换的数据的最高 8 位有效位。

14.4.6. 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。



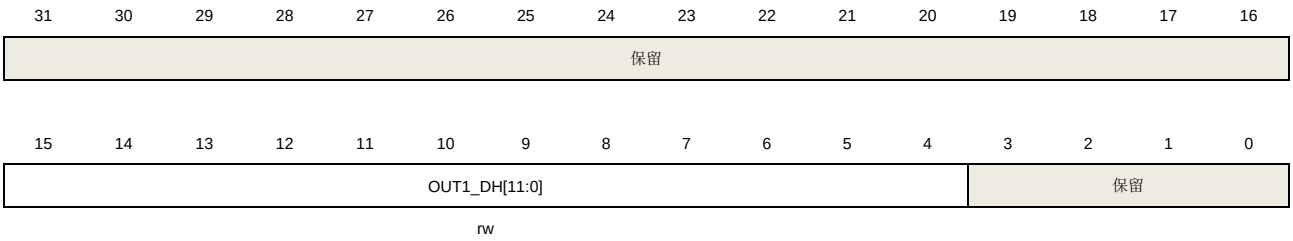
位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	OUT1_DH[11:0]	DACx_OUT1 12 位右对齐数据 这些位指定了将由 DACx_OUT1 转换的数据。

14.4.7. 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。



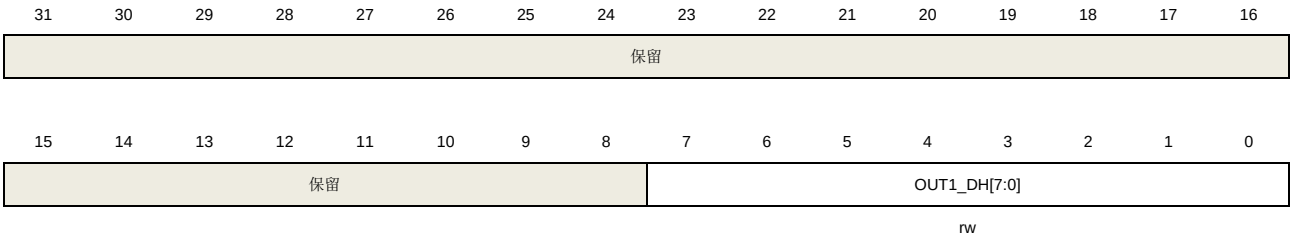
位/位域	名称	描述
31:16	保留	必须保持复位值
15:4	OUT1_DH[11:0]	DACx_OUT1 12 位左对齐数据 这些位指定了将由 DACx_OUT1 转换的数据。
3:0	保留	必须保持复位值

14.4.8. 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。



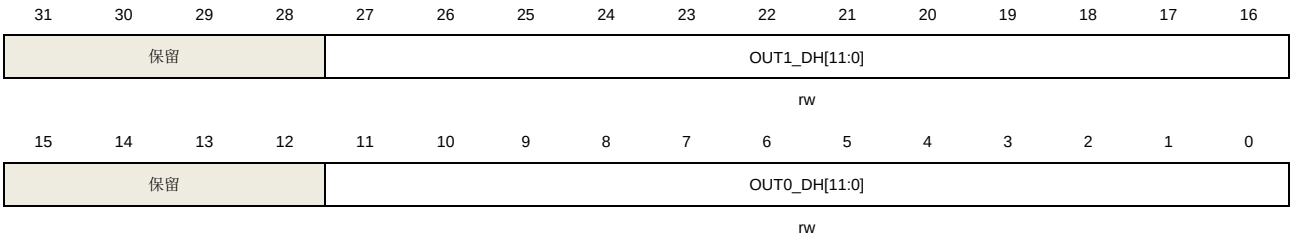
位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	OUT1_DH[7:0]	DACx_OUT1 8 位右对齐数据 这些位指定了将由 DACx_OUT1 转换的数据的 8 位最高有效位。

14.4.9. 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。



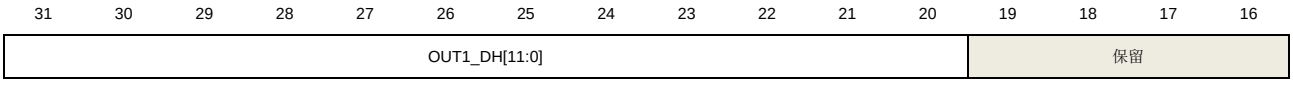
位/位域	名称	描述
31:28	保留	必须保持复位值
27:16	OUT1_DH[11:0]	DACx_OUT1 12 位右对齐数据 这些位指定了将由 DACx_OUT1 转换的数据。
15:12	保留	必须保持复位值
11:0	OUT0_DH[11:0]	DACx_OUT0 12 位右对齐数据 这些位指定了将由 DACx_OUT0 转换的数据。

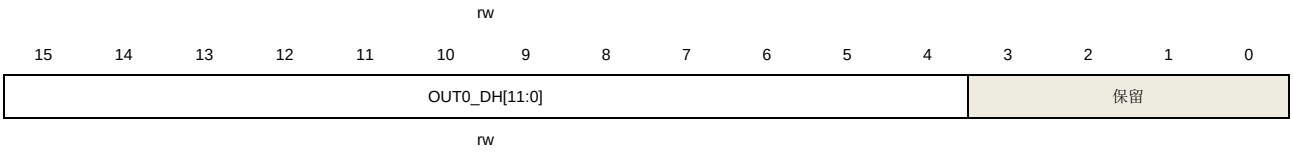
14.4.10. DACx 并发模式 12 位左对齐数据保持寄存器（DACC_L12DH）

地址偏移：0x24

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



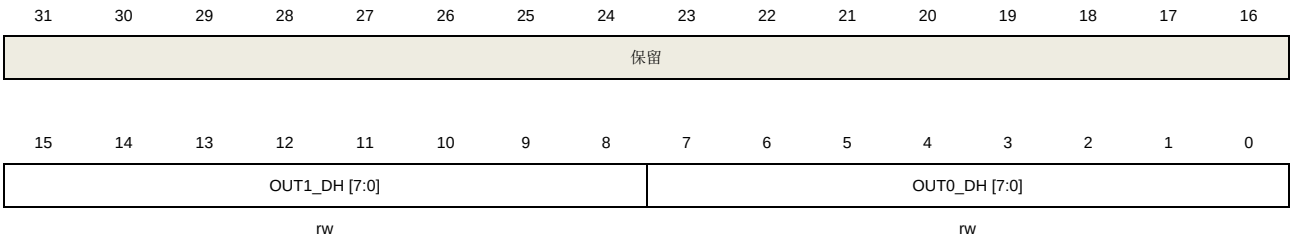


位/位域	名称	描述
31:20	OUT1_DH[11:0]	DACx_OUT1 12 位左对齐数据 这些位指定了将由 DACx_OUT1 转换的数据。
19:16	保留	必须保持复位值
15:4	OUT0_DH[11:0]	DACx_OUT0 12 位左对齐数据 这些位指定了将由 DACx_OUT0 转换的数据。
3:0	保留	必须保持复位值

14.4.11. 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。

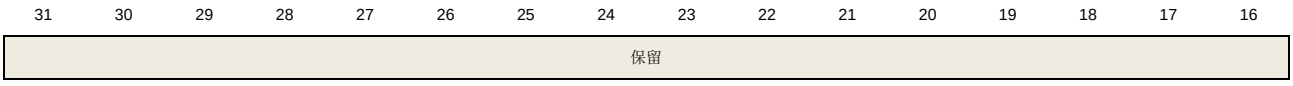


位/位域	名称	描述
31:16	保留	必须保持复位值
15:8	OUT1_DH[7:0]	DACx_OUT1 8 位右对齐数据 这些位指定了将由 DACx_OUT1 转换的数据的 8 位最高有效位。
7:0	OUT0_DH[7:0]	DACx_OUT0 8 位右对齐数据 这些位指定了将由 DACx_OUT0 转换的数据的 8 位最高有效位。

14.4.12. 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。

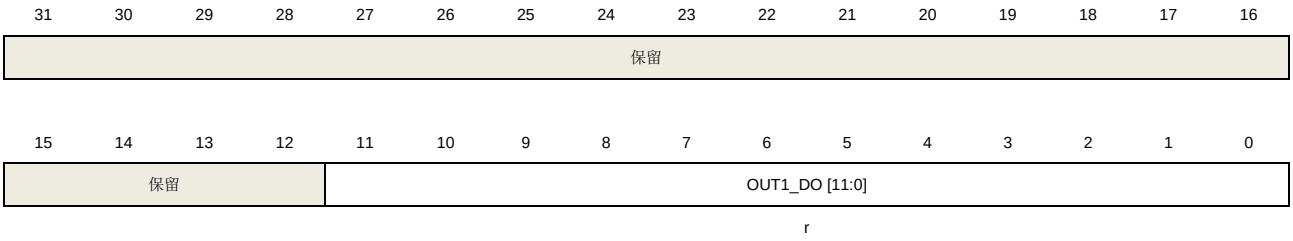


位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	OUT0_DO [11:0]	DACx_OUT0 数据输出。 这些位为只读类型，存储由 DACx_OUT0 转换的数据。

14.4.13. 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。

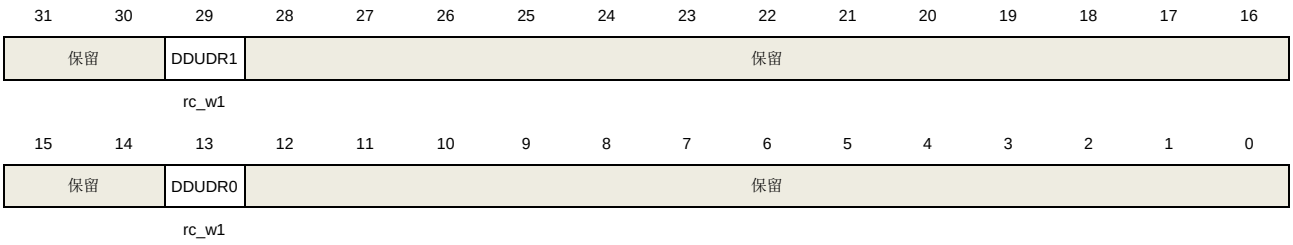


位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	OUT1_DO [11:0]	DACx_OUT1 数据输出。 这些位为只读类型，存储由 DACx_OUT1 转换的数据。

14.4.14. DACx 状态寄存器 0（DAC_STAT0）

地址偏移：0x34
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述

31:30	保留	必须保持复位值。
29	DDUDR1	DACx_OUT1 DMA 欠载标志位，硬件置位，软件写 1 清零。 0：没有欠载发生 1：发生欠载（DAC 触发产生速度快于 DMA 传输速度）
28:14	保留	必须保持复位值。
13	DDUDR0	DACx_OUT0 DMA 欠载标志位，硬件置位，软件写 1 清零。 0：没有欠载发生 1：发生欠载（DAC 触发产生速度快于 DMA 传输速度）
12:0	保留	必须保持复位值。

14.4.15. DACx 控制寄存器 1（DAC_CTL1）

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													FIFO UDRIE1	FIFO OVRIE1	FIFO EN1
													rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													FIFO UDRIE0	FIFO OVRIE0	FIFO EN0
													rw	rw	rw

位/位域	名称	描述
31:19	保留	必须保持复位值
18	FIFOUDRIE1	DACx_OUT1 FIFO 欠载中断使能 0：DACx_OUT1 FIFO 欠载中断禁能 1：DACx_OUT1 FIFO 欠载中断使能
17	FIFOOVRIE1	DACx_OUT1 FIFO 过载中断使能 0：DACx_OUT1 FIFO 过载中断禁能 1：DACx_OUT1 FIFO 过载中断使能
16	FIFOEN1	DACx_OUT1 数据 FIFO 使能。当 FIFOEN1 位置 1 时，需要软件设置 DDMAEN1=0 和 DTEN1=1。 0：DACx_OUT1 数据 FIFO 禁能 1：DACx_OUT1 数据 FIFO 使能
15:3	保留	必须保持复位值
2	FIFOUDRIE0	DACx_OUT0 FIFO 欠载中断使能

0：DACx_OUT0 FIFO 欠载中断禁能
1：DACx_OUT0 FIFO 欠载中断使能

1	FIFOOVRIE0	DACx_OUT0 FIFO 过载中断使能 0：DACx_OUT0 FIFO 过载中断禁能 1：DACx_OUT0 FIFO 过载中断使能
0	FIFOEN0	DACx_OUT0 数据 FIFO 使能。当 FIFOEN0 位置 1 时，需要软件设置 DDMAEN0=0 和 DTEN0=1。 0：DACx_OUT0 数据 FIFO 禁能 1：DACx_OUT0 数据 FIFO 使能

14.4.16. DACx 状态寄存器 1（DAC_STAT1）

地址偏移：0x84
复位值：0x0002 0002

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									FIFONUM1			FIFO UDR1	FIFO OVR1	FIFOE1	FIFOF1
										r		rc_w1	rc_w1	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									FIFONUM0			FIFO UDR0	FIFO OVR0	FIFOE0	FIFOF0
										r		rc_w1	rc_w1	r	r

位/位域	名称	描述
31:23	保留	必须保持复位值
22:20	FIFONUM1	DACx_OUT1 FIFO 的数据个数 000：数据个数为 0 001：数据个数为 1 010：数据个数为 2 011：数据个数为 3 100：数据个数为 4 101~111：保留
19	FIFOUDR1	DACx_OUT1 FIFO 欠载标志位 0：DACx_OUT1 FIFO 没有欠载 1：DACx_OUT1 FIFO 欠载
18	FIFOOVR1	DACx_OUT1 FIFO 过载标志位 0：DACx_OUT1 FIFO 没有过载 1：DACx_OUT1 FIFO 过载
17	FIFOE1	DACx_OUT1 FIFO 空载

		0：DACx_OUT1 FIFO 没有空载 1：DACx_OUT1 FIFO 空载
16	FIFOF1	DACx_OUT1 FIFO 满载标志位 0：DACx_OUT1 FIFO 没有满载 1：DACx_OUT1 FIFO 满载
15:7	保留	必须保持复位值
6:4	FIFONUM0	DACx_OUT0 FIFO 的数据个数 000：数据个数为 0 001：数据个数为 1 010：数据个数为 2 011：数据个数为 3 100：数据个数为 4 101~111：保留
3	FIFOUDR0	DACx_OUT0 FIFO 欠载标志位 0：DACx_OUT0 FIFO 没有欠载 1：DACx_OUT0 FIFO 欠载
2	FIFOOVR0	DACx_OUT0 FIFO 过载标志位 0：DACx_OUT0 FIFO 没有过载 1：DACx_OUT0 FIFO 过载
1	FIFOE0	DACx_OUT0 FIFO 空载标志位 0：DACx_OUT0 FIFO 没有空载 1：DACx_OUT0 FIFO 空载
0	FIFOF0	DACx_OUT0 FIFO 满载 0：DACx_OUT0 FIFO 没有满载 1：DACx_OUT0 FIFO 满载

15. 比较器（CMP）

15.1. 简介

通用比较器可独立工作，其输出端可用于 I / O 口，也可和定时器结合使用。

比较器在一定的条件下，可将模拟信号作为 TIMER 的触发源。

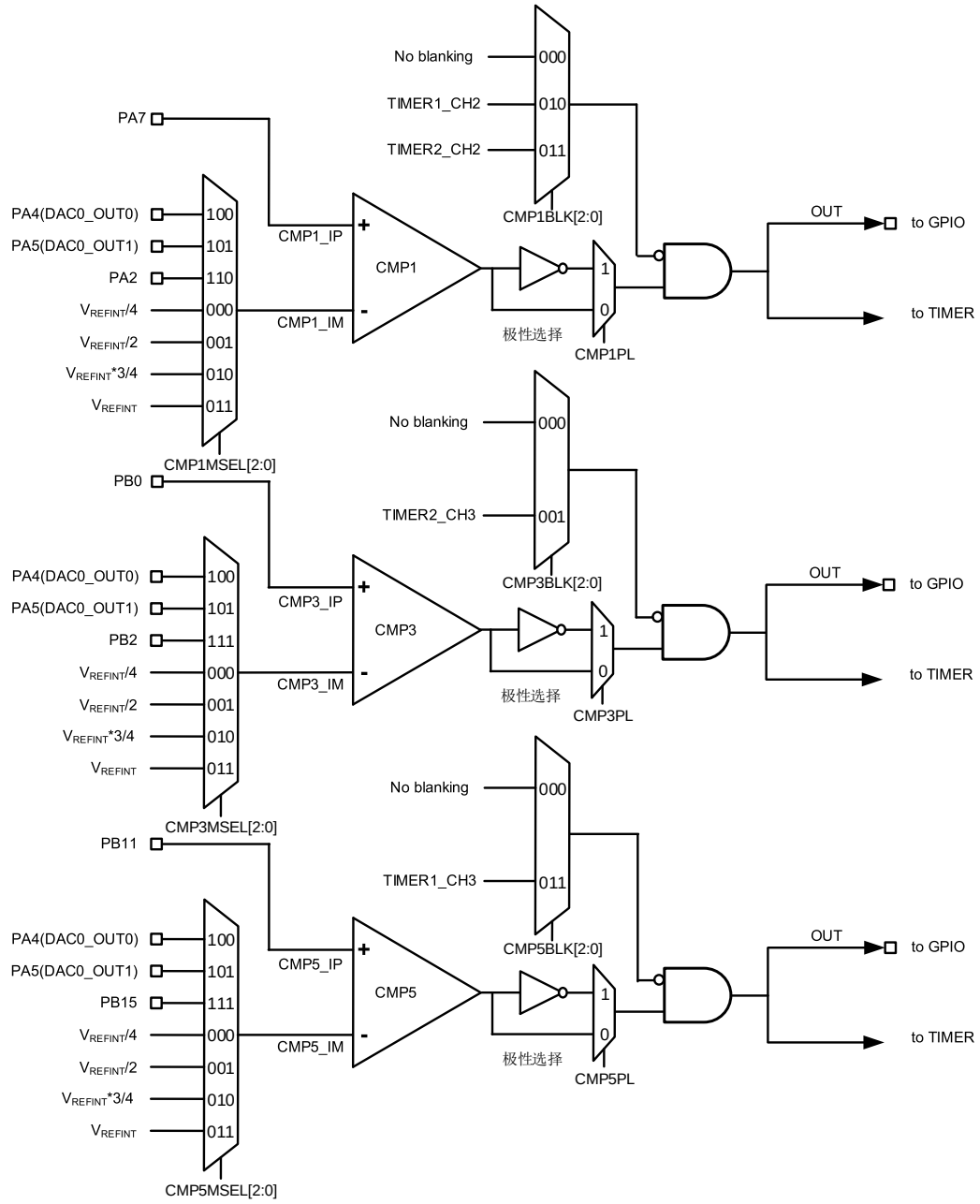
15.2. 主要特征

- 轨对轨比较器；
- 每个比较器可配置以下模拟信号作为输入源；
  - DAC 输出；
  - 多路复用 I / O 引脚；
  - 0.25、0.5、0.75、1 倍的内部参考电压；
- 比较器输出消隐；
- 输出到 I / O 口；
- 作为触发源输出到定时器。

15.3. 功能描述

比较器的框图展示如下：

图 15-1.比较器框图



注意：V_{REFINT} 是 1.2V.

15.3.1. 比较器时钟

比较器与 APB 总线连接，时钟与 PCLK 同步。

15.3.2. 比较器的 I / O 配置

在被选为比较器输入端之前，相关引脚必须配置为模拟模式。

比较器的输出可同时实现内部和外部输出。

参考 Datasheet 的引脚定义，比较器输出可以通过 GPIO 的备用功能连接到对应的 I / O 口。

比较器输出内部连接到定时器，他们的连接关系如下：

- CMP 输出连接到定时器输入通道。
- CMP 输出连接到定时器中止功能。

*表 15-1. CMP 的输入和输出总结*详细描述了 CMP 的输入和输出。

表 15-1. CMP 的输入和输出总结

	CMP1	CMP3	CMP5
CMP 同相输入连接到 I / O	PA7	PB0	PB11
CMP 反相输入连接到 I / O	PA2 PA4 PA5	PA4 PA5 PB2	PA4 PA5 PB15
CMP 反相输入连接到内部信号	V_{REFINT} / 4 V_{REFINT} / 2 V_{REFINT} * 3 / 4 V_{REFINT} DAC0_OUT0 DAC0_OUT1	V_{REFINT} / 4 V_{REFINT} / 2 V_{REFINT} * 3 / 4 V_{REFINT} DAC0_OUT0 DAC0_OUT1	V_{REFINT} / 4 V_{REFINT} / 2 V_{REFINT} * 3 / 4 V_{REFINT} DAC0_OUT0 DAC0_OUT1
CMP 输出连接到 I / O	PA2 PA12 PB9 PE8 PE13	PB1 PE9 PE12	PA10 PC6 PE10 PE11
CMP 输出连接到内部信号	TIMER0_CH0 TIMER1_CH3 TIMER2_CH0 SHRTIMER_EXEV0 SHRTIMER_EXEV5	TIMER2_CH2 SHRTIMER_EXEV1 SHRTIMER_EXEV6	TIMER1_CH1 SHRTIMER_EXEV2 SHRTIMER_EXEV7
CMP 输出（电机控制保护）	TIMER0 BRKIN		

注意：CMP1/3/5 的输出直接连接到 SHRTIMER 外设。

15.3.3. 比较器寄存器写保护

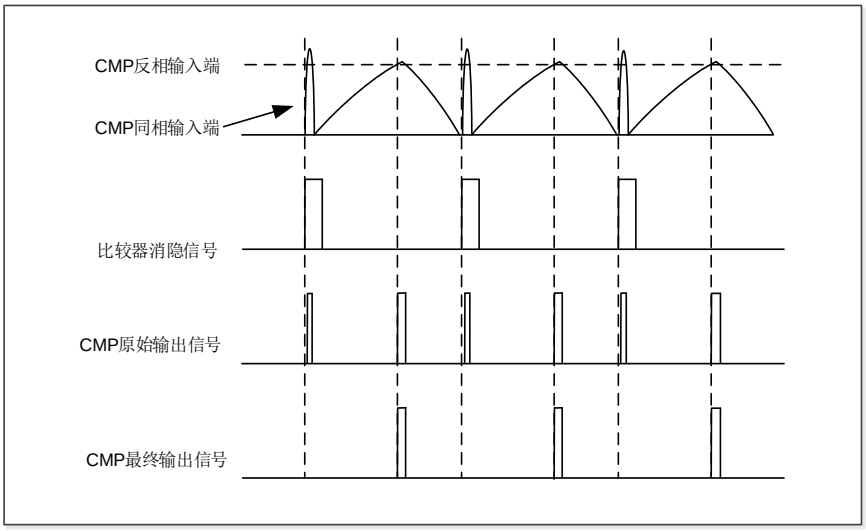
比较器的控制状态寄存器（CMPx_CS）可通过设置 CMPxLK 位为 1 来进行写保护，CMPx_CS 寄存器，包含 CMPxLK 位，就会变为只读位，只有在 MCU 复位时才可以复位。

15.3.4. 比较器输出消隐

比较器输出消隐功能可以避免比较器输入信号中的短脉冲对输出信号的干扰。如果 CMPx_CS 寄存器中的 CMPxBLK[2:0]位域设置为有效值，则比较器最终输出的信号由所选消隐信号的互补信号和比较器的原始输出进行“与”运算获得。

*图 15-2. 比较器的输出消隐*显示了比较器的输出消隐功能。

图 15-2. 比较器的输出消隐



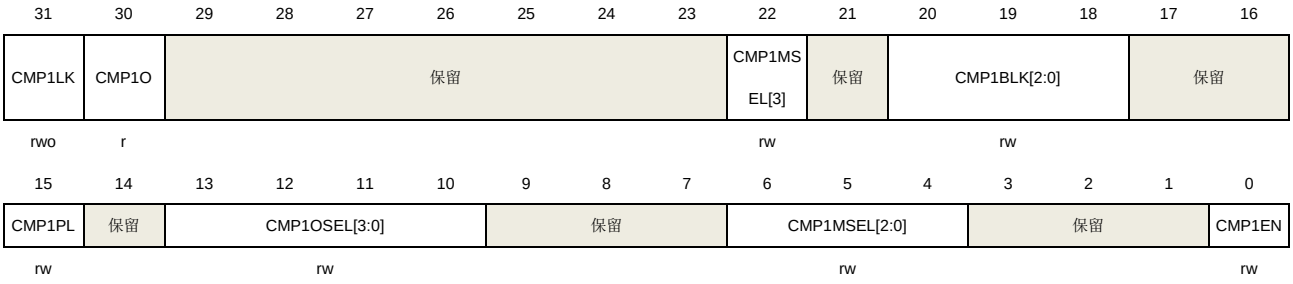
15.4. CMP 寄存器

CMP 基地址：0x4001 7C00

15.4.1. CMP1 控制状态寄存器（CMP1_CS）

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	CMP1LK	CMP1 写保护 该位可将 CMP1 的各控制位设为只读，该位可写一次，通过系统复位清除，可通过软件置位。 0：CMP1_CS 是可读可写位 1：CMP1_CS 是只读位
30	CMP1O	CMP1 输出 该位反映 CMP1 输出状态，是只读位。 0：同相输入端低于反相输入端，输出为低电平。 1：同相输入端高于反相输入端，输出为高电平。
29:23	保留	必须保持复位值。
22	CMP1MSEL[3]	CMP1MSEL 位域的位 3 见 CMP1_CS 的 6:4 位。
21	保留	必须保持复位值。
20:18	CMP1BLK[2:0]	CMP1 输出消隐源 该位域用于选择哪个定时器输出控制 CMP1 的输出消隐。 000：无消隐 001：保留 010：选择 TIMER1_CH2 输出比较信号为消隐源 011：选择 TIMER2_CH2 输出比较信号为消隐源 100~111：保留
17:16	保留	必须保持复位值。

15	CMP1PL	CMP1 输出极性 该位用于控制 CMP1 输出极性。 0：输出是正相的 1：输出是反相的
14	保留	必须保持复位值。
13:10	CMP1OSEL[2:0]	CMP1 的输出选择 该位域用于控制 CMP1 输出选择。 0000：无选择 0001：定时器 0 中止输入 0010~0110：保留 0111：定时器 0 通道 0 输入捕获 1000：定时器 1 通道 3 输入捕获 1001：保留 1010：定时器 2 通道 0 输入捕获 1011~1111：保留 **注意**：使用定时器捕获比较器的输出信号时，建议先使能 CMP，再配置定时器通道
9:7	保留	必须保持复位值。
6:4	CMP1MSEL[2:0]	CMP1_IM 输入选择 该位域结合位 22，用于选择 CMP1 的输入端 CMP1_IM 的输入源 0000：V_{REFINT} / 4 0001：V_{REFINT} / 2 0010：V_{REFINT} * 3 / 4 0011：V_{REFINT} 0100：PA4（DAC0_OUT0） 0101：PA5（DAC0_OUT1） 0110：PA2 0111~1111：保留
3:1	保留	必须保持复位值。
0	CMP1EN	CMP1 使能 0：CMP1 禁能 1：CMP1 使能

15.4.2. CMP3 控制状态寄存器（CMP3_CS）

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
CMP3LK	CMP3O	保留							CMP3MSEL[3]	保留	CMP3BLK[2:0]			保留	

rwo	r								rw			rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP3PL	保留	CMP3OSEL[3:0]				保留			CMP3MSEL[2:0]			保留			CMP3EN
rw		rw							rw						rw

位/位域	名称	描述
31	CMP3LK	CMP3 写保护 该位可将 CMP3 的各控制位设为只读，该位可写一次，通过系统复位清除，可通过软件置位。 0：CMP3_CS 是可读可写位 1：CMP3_CS 是只读位
30	CMP3O	CMP3 输出 该位反映 CMP3 输出状态，是只读位。 0：同相输入端低于反相输入端，输出为低电平。 1：同相输入端高于反相输入端，输出为高电平。
29:23	保留	必须保持复位值。
22	CMP3MSEL[3]	CMP3MSEL 位域的位 3 见 CMP3_CS 的 6:4 位。
21	保留	必须保持复位值。
20:18	CMP3BLK[2:0]	CMP3 输出消隐源 该位域用于选择哪个定时器输出控制 CMP3 的输出消隐。 000：无消隐 001：选择 TIMER2_CH3 输出比较信号为消隐源 010~111：保留
17:16	保留	必须保持复位值。
15	CMP3PL	CMP3 输出极性 该位用于控制 CMP3 输出极性。 0：输出是正相的 1：输出是反相的
14	保留	必须保持复位值。
13:10	CMP3OSEL[2:0]	CMP3 的输出选择 该位域用于控制 CMP3 输出选择。 0000：无选择 0001：定时器 0 中止输入 0010~0101：保留 0110：定时器 2 通道 2 输入捕获 0111~1111：保留 **注意**：使用定时器捕获比较器的输出信号时，建议先使能 CMP，再配置定时器通道
9:7	保留	必须保持复位值。

6:4	CMP3MSEL[2:0]	CMP3_IM 输入选择 该位域结合位 22，用于选择 CMP3 的输入端 CMP3_IM 的输入源。 0000：V_{REFINT} / 4 0001：V_{REFINT} / 2 0010：V_{REFINT} * 3 / 4 0011：V_{REFINT} 0100：PA4（DAC0_OUT0） 0101：PA5（DAC0_OUT1） 0110：保留 0111：PB2 1000~1111：保留
3:1	保留	必须保持复位值。
0	CMP3EN	CMP3 使能 0：CMP3 禁能 1：CMP3 使能

15.4.3. CMP5 控制状态寄存器（CMP5_CS）

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
CMP5LK	CMP5O	保留							CMP5MSEL[3]	保留	CMP5BLK[2:0]			保留	
rwo	r								rw		rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP5PL	保留	CMP5OSEL[3:0]				保留			CMP5MSEL[2:0]			保留			CMP5EN
rw		rw							rw						rw

位/位域	名称	描述
31	CMP5LK	CMP5 写保护 该位可将 CMP5 的各控制位设为只读，该位可写一次，通过系统复位清除，可通过软件置位。 0：CMP5_CS 是可读可写位 1：CMP5_CS 是只读位
30	CMP5O	CMP5 输出 该位反映 CMP5 输出状态，是只读位。 0：同相输入端低于反相输入端，输出为低电平。 1：同相输入端高于反相输入端，输出为高电平。
29:23	保留	必须保持复位值。
22	CMP5MSEL[3]	CMP5MSEL 位域的位 3

		见 CMP5_CS 的 6:4 位
21	保留	必须保持复位值。
20:18	CMP5BLK[2:0]	CMP5 输出消隐源 该位域用于选择哪个定时器输出 CMP5 的输出消隐。 000：无消隐 001：保留 010：保留 011：选择 TIMER1_CH3 输出比较信号为消隐源 100~111：保留
17:16	保留	必须保持复位值。
15	CMP5PL	CMP5 输出极性 该位用于控制 CMP5 输出极性。 0：输出是正相的 1：输出是反相的
14	保留	必须保持复位值。
13:10	CMP5OSEL[2:0]	CMP5 的输出选择 该位域用于控制 CMP5 输出选择。 0000：无选择 0001：定时器 0 中止输入 0010~0101：保留 0110：定时器 1 通道 1 输入捕获 0111~1111：保留 **注意**：使用定时器捕获比较器的输出信号时，建议先使能 CMP，再配置定时器通道
9:7	保留	必须保持复位值。
6:4	CMP5MSEL[2:0]	CMP5_IM 输入选择 该位域结合位 22，用于选择 CMP5 的输入端 CMP5_IM 的输入源。 0000：V_{REFINT} / 4 0001：V_{REFINT} / 2 0010：V_{REFINT} * 3 / 4 0011：V_{REFINT} 0100：PA4（DAC0_OUT0） 0101：PA5（DAC0_OUT1） 0110：保留 0111：PB15 1000~1111：保留
3:1	保留	必须保持复位值。
0	CMP5EN	CMP5 使能 0：CMP5 禁能 1：CMP5 使能

16. 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。

16.1. 独立看门狗定时器（FWDGT）

16.1.1. 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。

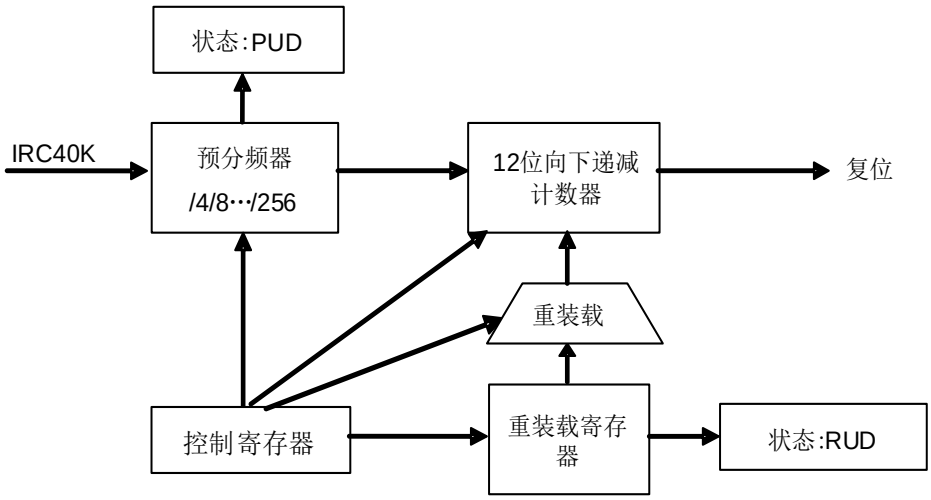
16.1.2. 主要特征

- 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

16.1.3. 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 16-1. 独立看门狗定时器框图*为独立看门狗定时器的功能模块。

图 16-1. 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 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 寄存器的相应状态位被置 1。

如果 DBG 控制寄存器 0（DBG_CTL0）中的 FWDGT_HOLD 位被清 0，即使 Cortex®-M33 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 16-1. 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。

16.1.4. FWDGT 寄存器

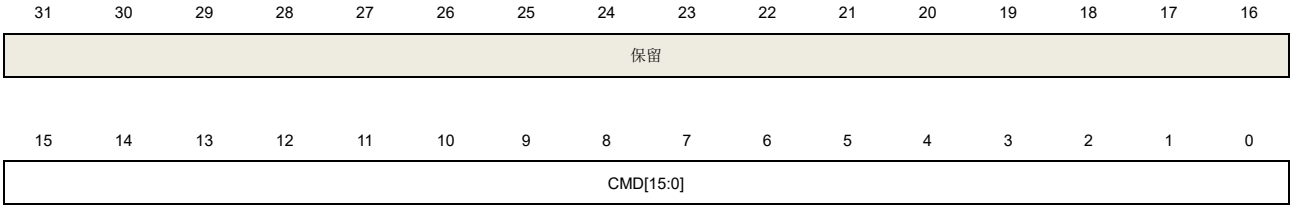
FWDGT 基地址：0x4000 3000

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。



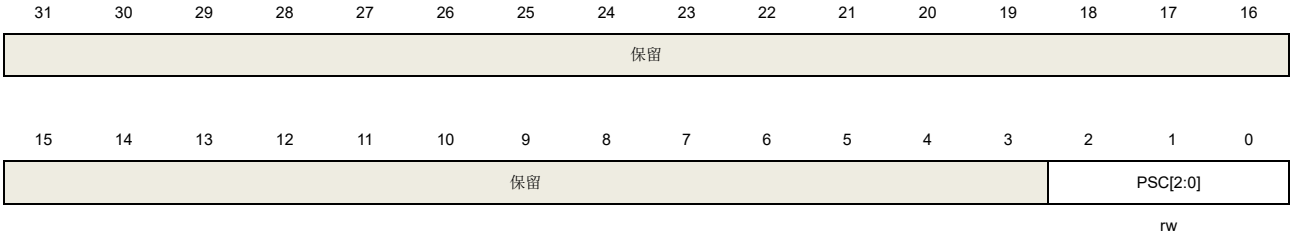
位/位域	名称	说明
31:16	保留	必须保持复位值。
15:0	CMD[15:0]	只可写，写入不同的值来产生不同的功能 0x5555：关闭FWDGT_PSC和FWDGT_RLD的写保护 0xCCCC：开启独立看门狗定时器定时计数器。计数减到0时产生复位 0xAAAA：重装载计数器

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	说明
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。 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（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。

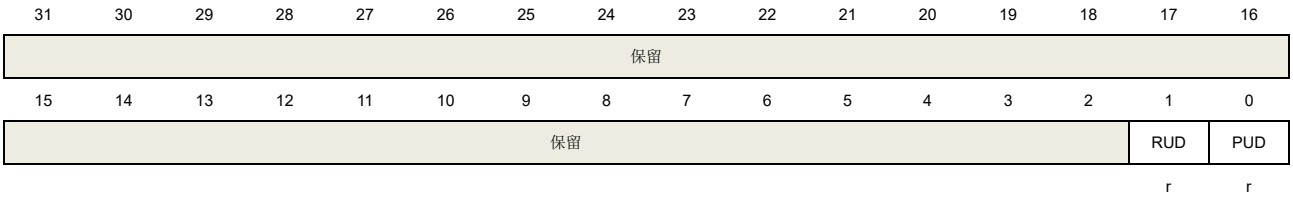
位/位域	名称	说明
31:12	保留	必须保持复位值。
11:0	RLD[11:0]	独立看门狗定时器定时计数器重装载值，向FWDGT_CTL寄存器写入0xAAAA的时候，这个值会被更新到看门狗定时器计数器中。 这些位有写保护功能。在写这些位之前需向FWDGT_CTL寄存器中写0x5555。在改写这个寄存器的过程中，FWDGT_STAT寄存器的RUD位被置1，从此寄存器中读取的任何值都是无效的。 如果应用需要使用几个重装载值，改变重装载值之前必须等到RUD位被清0。更新了重装载寄存器的值后，在代码持续执行之前不必等待RUD值被清零（进入省电模式前需等待RUD值清零）。

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。



位/位域	名称	说明
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 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 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。

16.2. 窗口看门狗定时器（WWDGT）

16.2.1. 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。

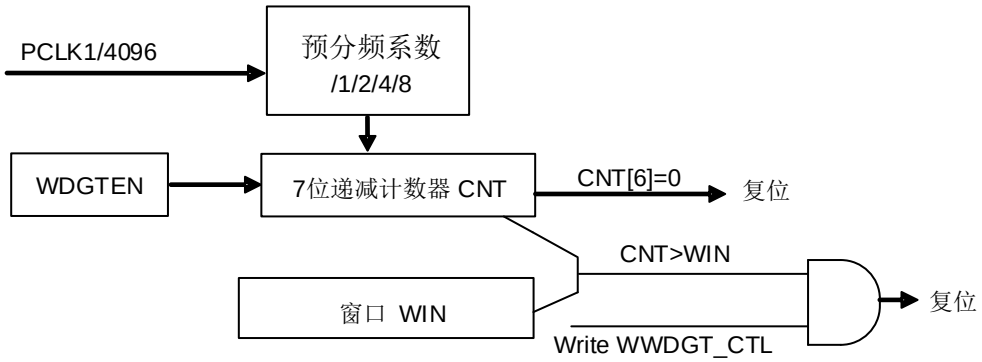
16.2.2. 主要特征

- 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。

16.2.3. 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）或是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 16-2. 窗口看门狗定时器框图



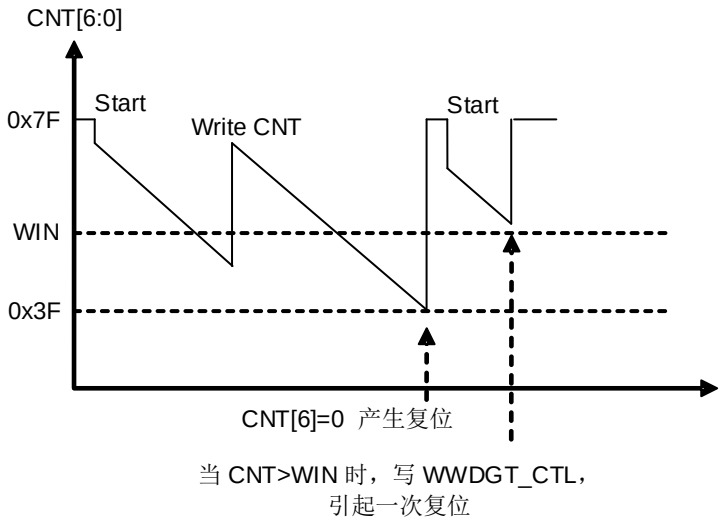
上电复位之后窗口看门狗定时器总是关闭的。软件可以向 WWDGT_CTL 的 WDGTEN 写 1 开启窗口看门狗定时器。窗口看门狗定时器打开后，计数器始终递减计数，计数器配置的值应该大于 0x3F，也就是说 CNT[6]位应该被置 1。CNT[5:0]决定了两次重装载之间的最大间隔时间。计数器的递减速度取决于 APB1 时钟和预分频器（WWDGT_CFG 寄存器的 PSC[1:0]位）。

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 16-3. 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：

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

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 16-2.在 90MHz（fPCLK1）时的最大/最小超时值*。

表 16-2.在 90MHz（f_{PCLK1}）时的最大/最小超时值

预分频系数	PSC[1:0]	最小超时 CNT[6:0] =0x40	最大超时 CNT[6:0]=0x7F
1 / 1	00	45.51μs	2.91ms
1 / 2	01	91.02μs	5.83ms
1 / 4	10	182.04μs	11.65ms
1 / 8	11	364.08μs	23.30ms

如果 MCU 调试模块中的 WWDGT_HOLD 位被清 0，即使 Cortex®-M33 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。

16.2.4. WWDGT 寄存器

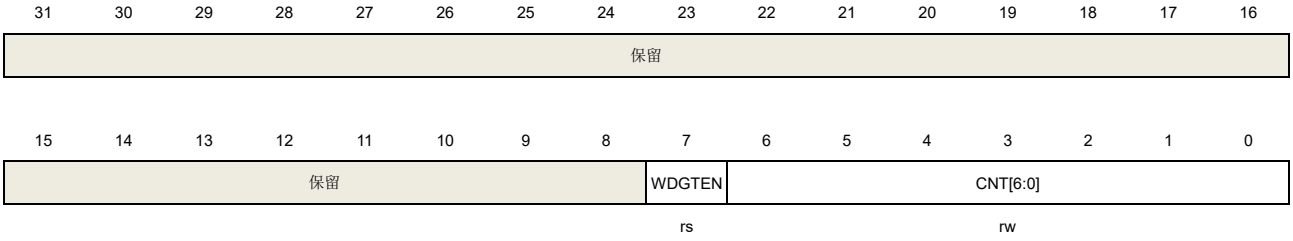
WWDGT 基地址：0x4000 2C00

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。



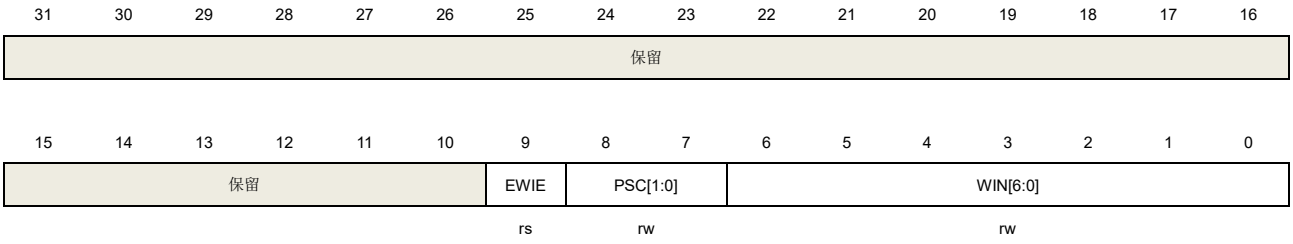
位/位域	名称	说明
31:8	保留	必须保持复位值。
7	WDGTEN	开启窗口看门狗定时器，硬件复位的时候清0，写0无效。 0：关闭窗口看门狗定时器 1：开启窗口看门狗定时器
6:0	CNT[6:0]	看门狗定时器计数器的值。当计数值从0x40降到0x3F时，产生看门狗定时器复位。 当计数器值高于窗口值的时候，写计数器可以产生看门狗定时器系统复位。

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	说明
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。该位由硬件复位清0，或通过RCU模块的WWDGTRST位进行软件复位。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 00：PCLK1 / 4096 / 1

01：PCLK1 / 4096 / 2

10：PCLK1 / 4096 / 4

11：PCLK1 / 4096 / 8

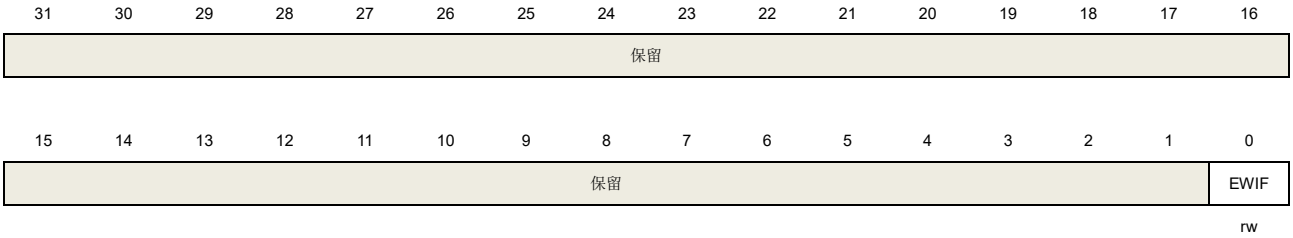
6:0	WIN[6:0]	窗口值，当看门狗定时器计数器的值大于窗口值时，写看门狗定时器计数器（WWDGT_CTL的CNT位）会产生系统复位。

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	说明
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，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bit可以通过写0清零，写1无效。

17. 实时时钟（RTC）

17.1. 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 VDD 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。

17.2. 主要特性

- 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

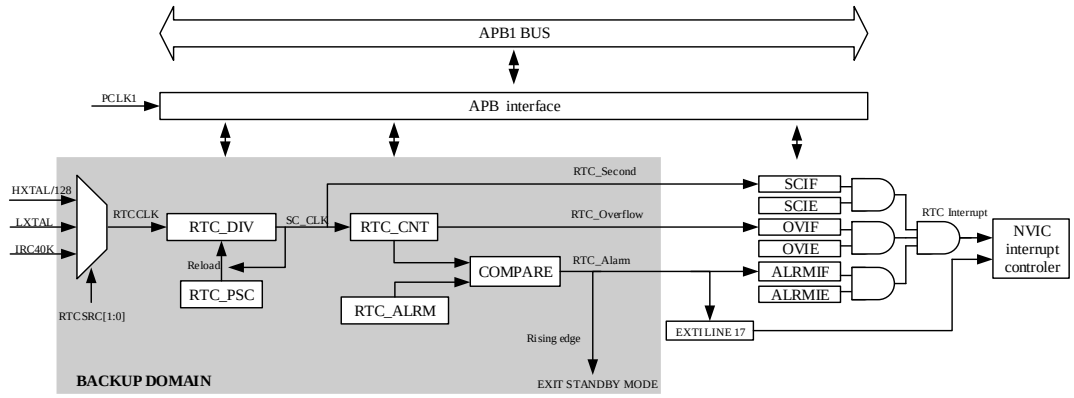
17.3. 功能描述

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位当前系统时间。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 17-1. RTC 框图



17.3.1. 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。

17.3.2. 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。

17.3.3. 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. 通过将 RTC_CTL 寄存器中的 CMF 置位来进入配置模式；
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。

17.3.4. RTC 标志位

RTC 秒中断标志（SCIF）在 RTC 计数器更新之前的每个 RTCCLK 周期置位。

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任一序列进行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 17-2. RTC 秒信号及闹钟信号的波形(RTC_PSC = 3,RTC_ALRM = 2)

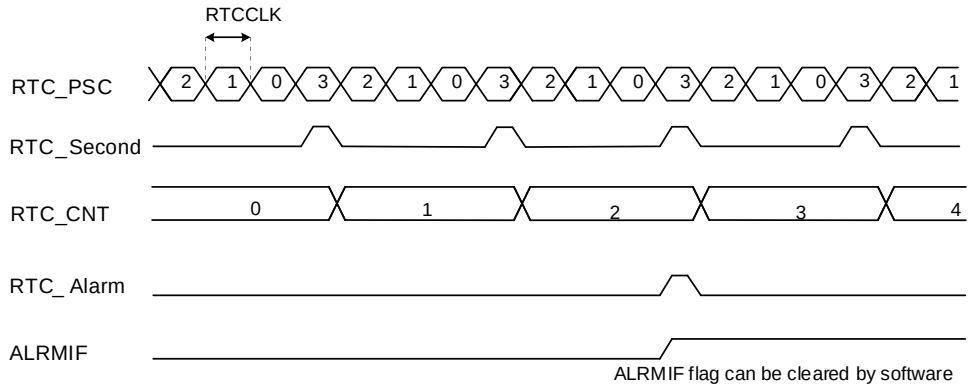
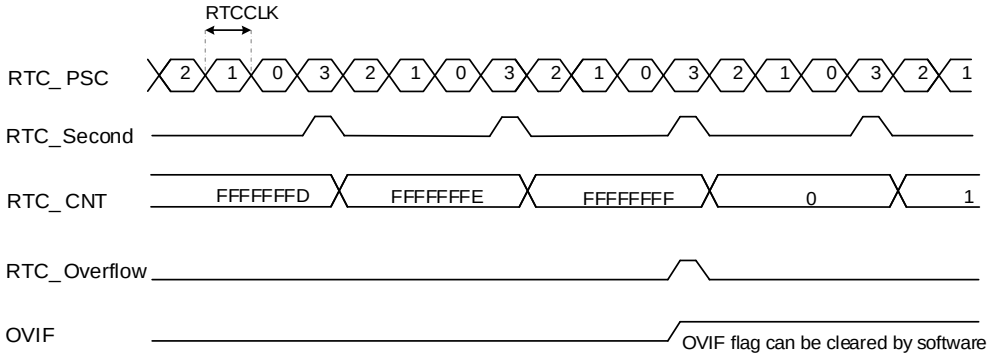


图 17-3. RTC 秒信号及溢出信号的波形(RTC_PSC = 3)



17.4. RTC 寄存器

RTC 基地址：0x4000 2800

17.4.1. 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

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													OVIE	ALRMIE	SCIE
													rw	rw	rw

位/位域	名称	描述
31:3	保留	必须保持复位值
2	OVIE	溢出中断使能 0：禁用溢出中断 1：使能溢出中断
1	ALRMIE	闹钟中断使能 0：禁用闹钟中断 1：使能闹钟中断
0	SCIE	秒中断使能 0：禁用秒中断 1：使能秒中断

17.4.2. 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

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

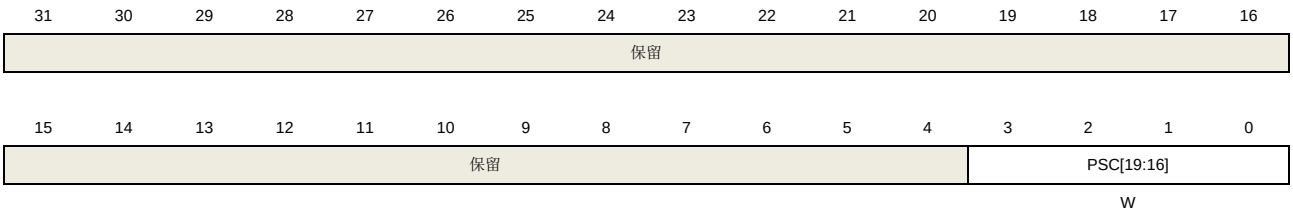
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留										LWOFF	CMF	RSYNF	OVIF	ALRMIF	SCIF
										r	rw	rc_w0	rc_w0	rc_w0	rc_w0

位/位域	名称	描述
31:6	保留	必须保持复位值
5	LWOFF	上次对 RTC 寄存器写操作标志 0：上次对 RTC 寄存器写操作没有完成 1：上次对 RTC 寄存器写操作已经完成
4	CMF	配置模式标志 0：退出配置模式 1：进入配置模式
3	RSYNF	寄存器同步标志 0：寄存器没有与 APB1 时钟同步 1：寄存器已经与 APB1 时钟同步
2	OVIF	溢出中断标志 0：没有检测到溢出事件 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 0：没有检测到闹钟事件 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 0：没有检测到秒事件 1：检测到秒事件。当 RTC_INTEN 寄存器的 SCIE 位被置 1，中断发生。 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。

17.4.3. 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

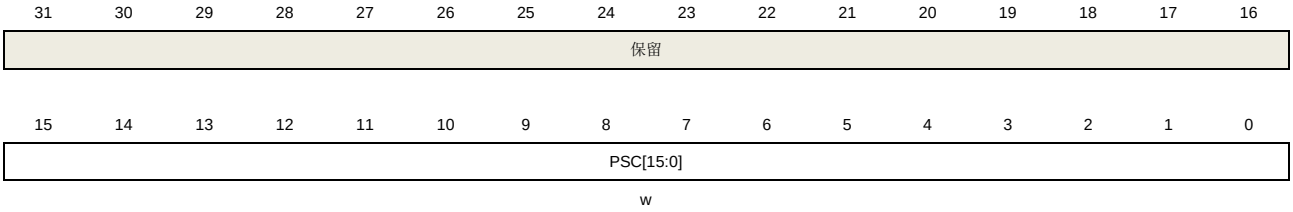


位/位域	名称	描述
31:4	保留	必须保持复位值
3:0	PSC[19:16]	RTC 预分频器高位值

17.4.4. 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

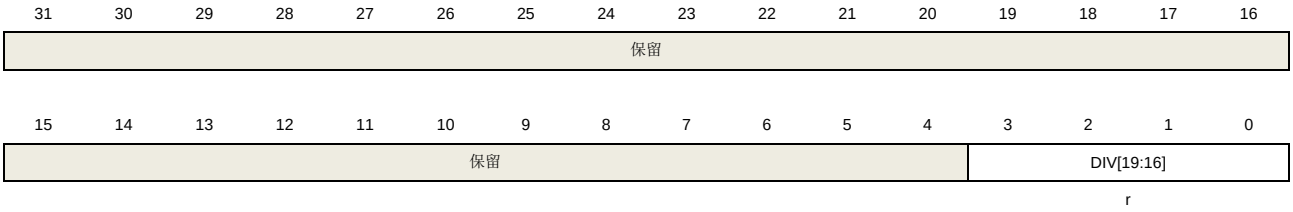


位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	PSC[15:0]	RTC 预分频器低位值 SC_CLK 的频率是 RTCCLK 的频率除以(PSC[19:0]+1)

17.4.5. 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
31:4	保留	必须保持复位值
3:0	DIV[19:16]	RTC 分频器高位

17.4.6. 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

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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	DIV[15:0]	RTC 分频器低位 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

17.4.7. 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

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[31:16]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CNT[31:16]	RTC 计数寄存器高位

17.4.8. 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

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															

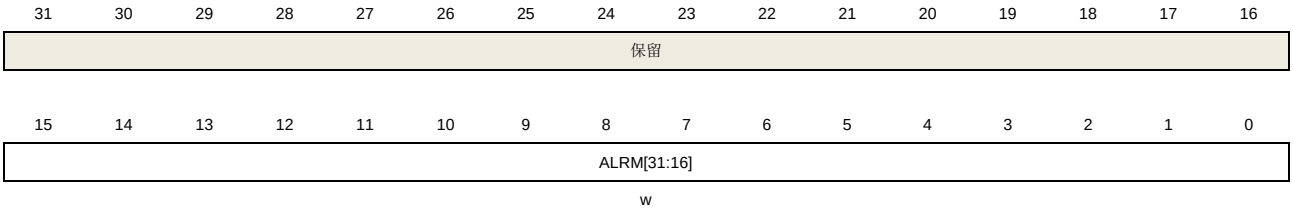
位/位域	名称	描述
31:16	保留	必须保持复位值

15:0	CNT[15:0]	RTC 计数寄存器低位

17.4.9. 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

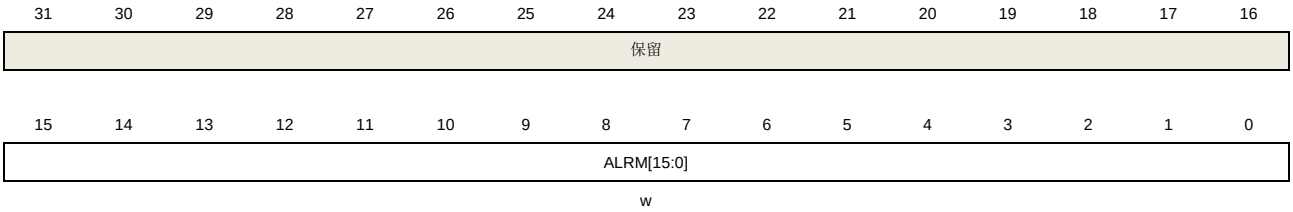


位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	ALRM[31:16]	RTC 闹钟值高位

17.4.10. 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



位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	ALRM[15:0]	RTC 闹钟值低位

18. 定时器（TIMER）

表 18-1. 定时器（TIMERx）分为五种类型

定时器	定时器 0/7	定时器 1~4	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	32 位（定时器 1） 16 位（定时器 2~4）	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:** 参考注释(5) **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。但是定时器 5 和定时 6 中没有 DMA 配置寄存器。

(5) 在互联型产品中，TIMER1 的 ITI1 由 ***AFIO 端口配置寄存器 0（AFIO_PCF0）*** 中的 TIMER1ITI1_REMAP 位来决定内部连接的信号源；
在非互联型产品中，TIMER1 的 ITI1 内部连接到 TIMER7_TRGO；

18.1. 高级定时器（TIMERx,x=0,7）

18.1.1. 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，可以被用来计数，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。

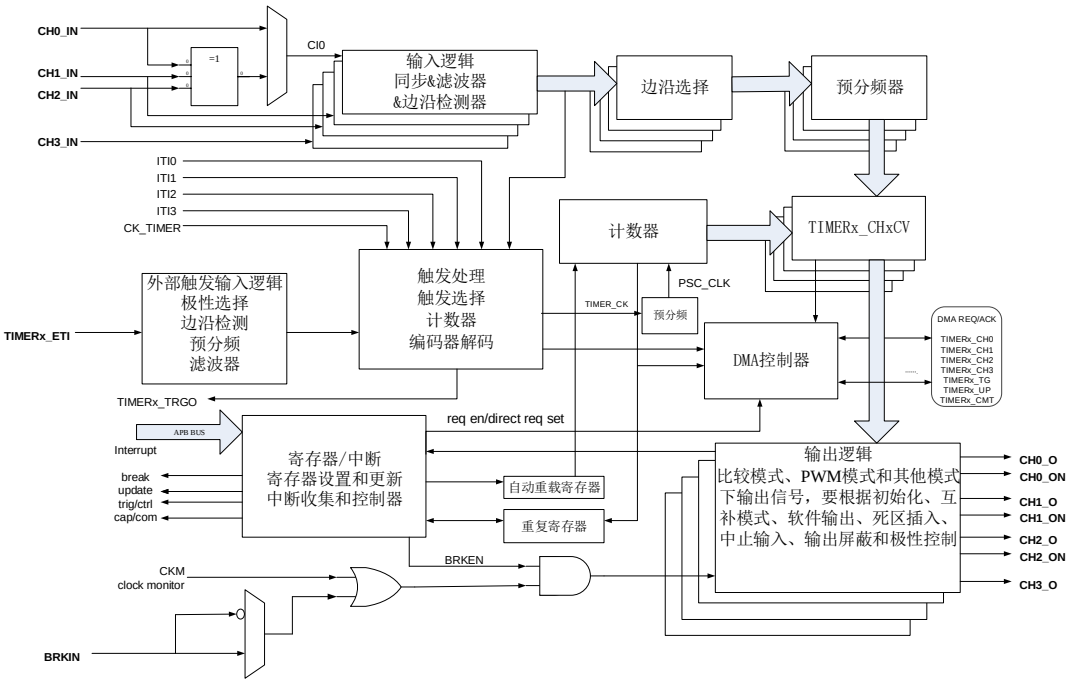
18.1.2. 主要特性

- 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；
- 正交译码器接口：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；
- 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。

18.1.3. 结构框图

*图 18-1. 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 18-1. 高级定时器结构框图



18.1.4. 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。

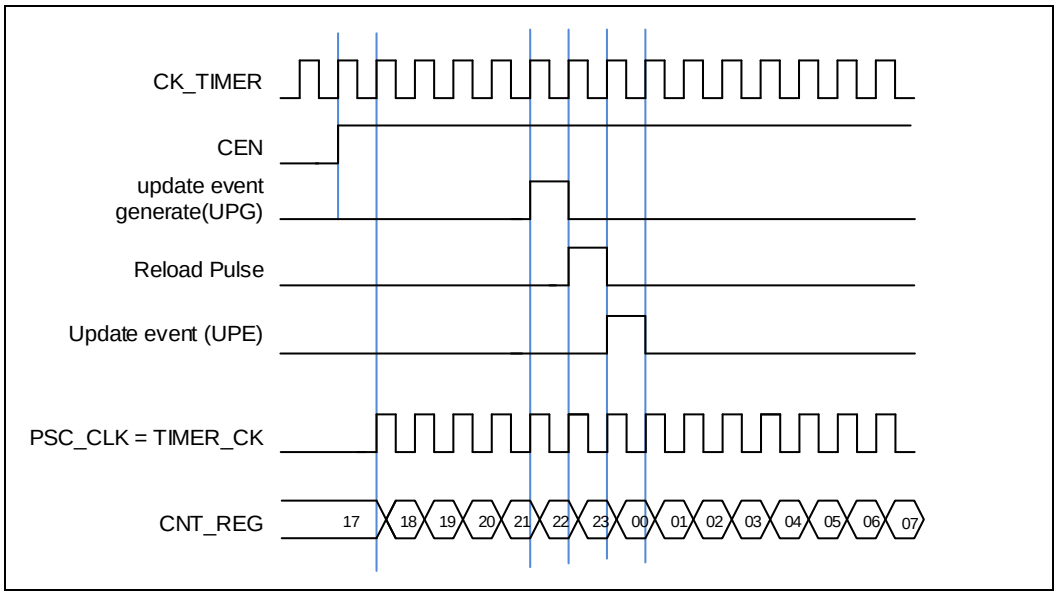
- SMC[2:0]==3’b000，定时器选择内部时钟源（连接到RCU模块的CK_TIMER）

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 18-2. 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。

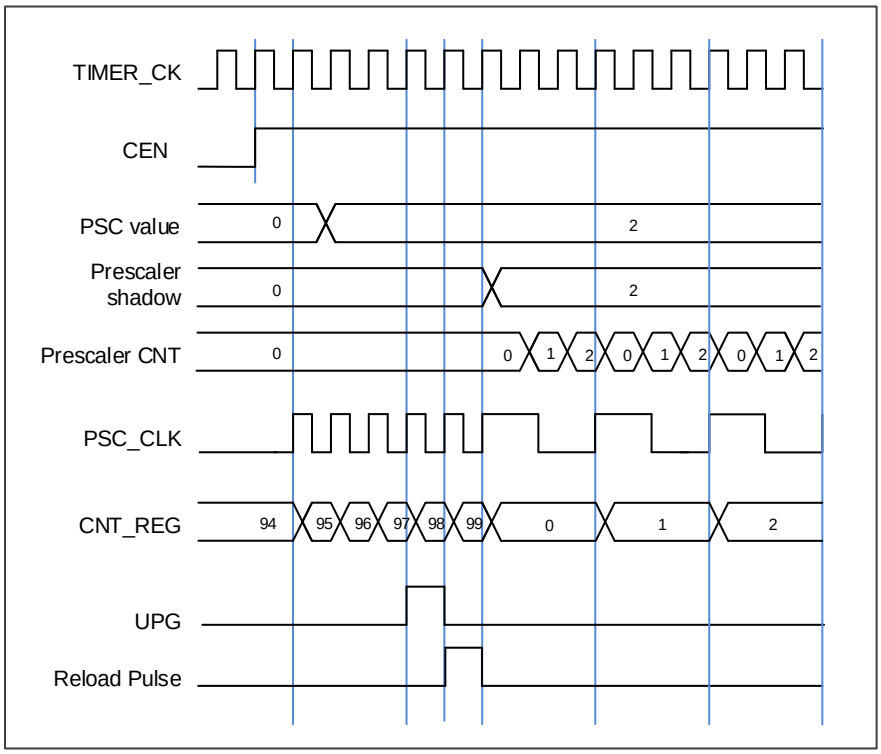
■ SMC1==1’b1(外部时钟模式1)，定时器选择外部输入引脚ETI作为时钟源

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 18-3. 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。而且，在(TIMERx_CREP+1)次上溢后会产生更新事件。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。

图 18-4. 向上计数时序图，PSC=0/2 和 *图 18-5. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值* 给出一些例子，当 TIMERx_CAR=0x99 时，计数器在不同预分频因子下的行为。

图 18-4. 向上计数时序图，PSC=0/2

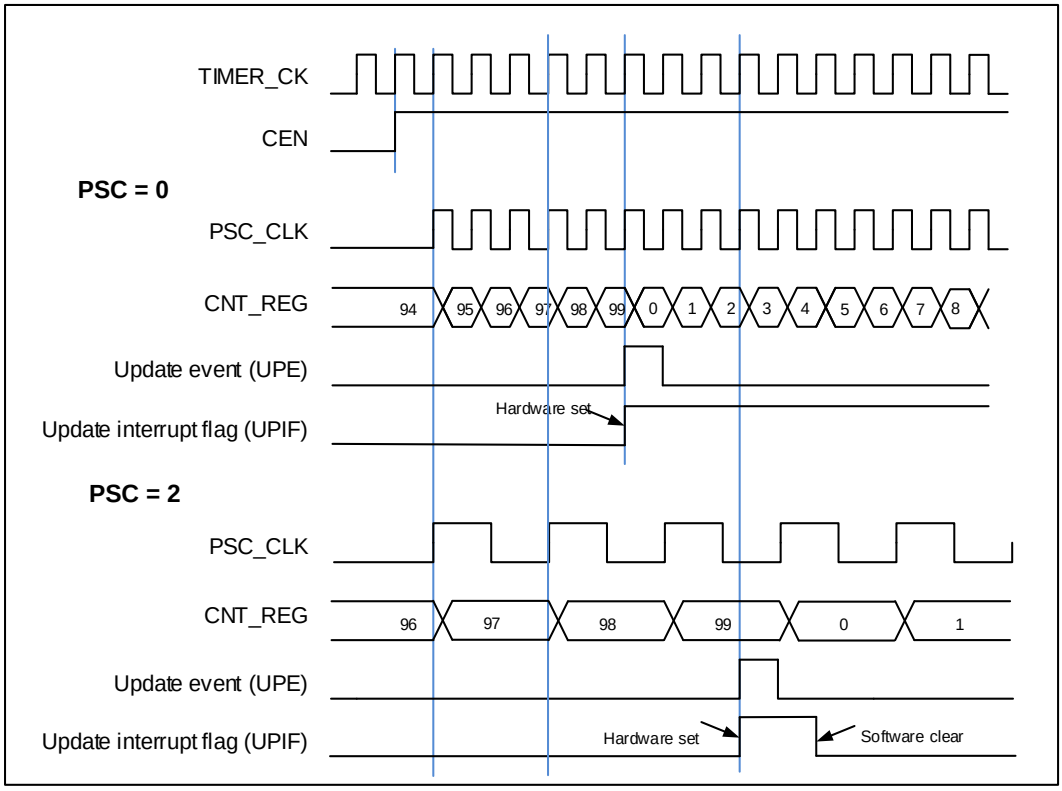
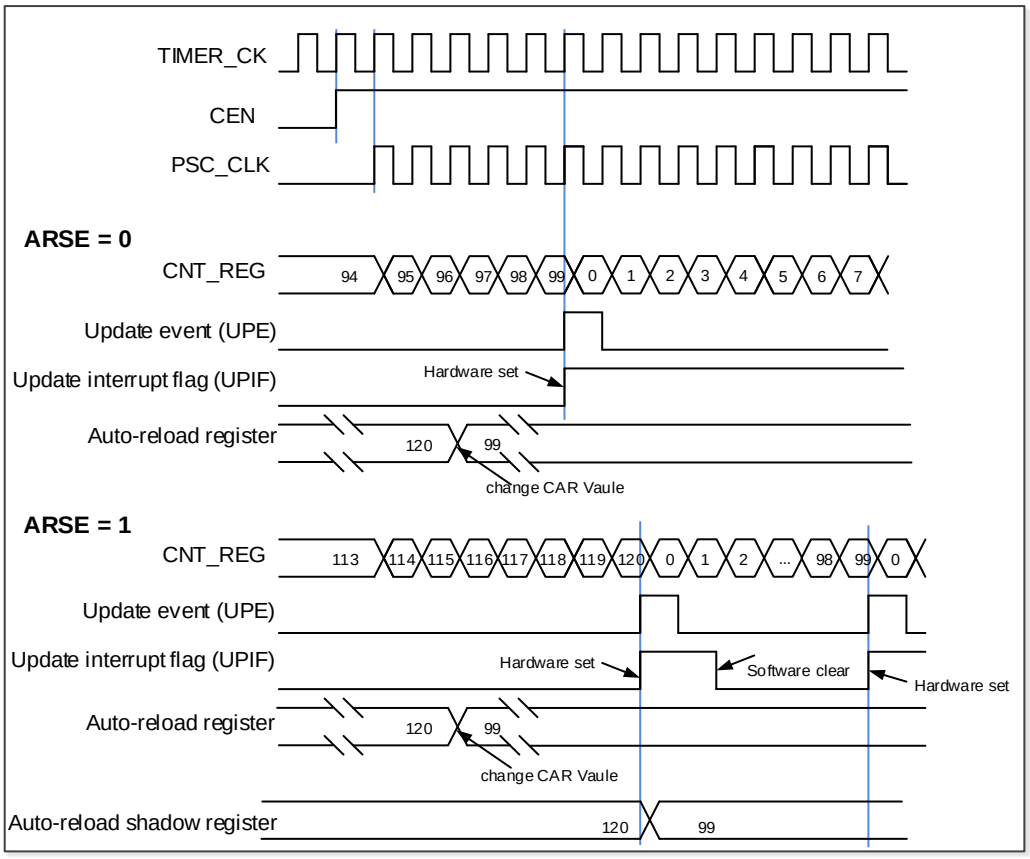


图 18-5. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



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。另外，在(TIMERx_CREP+1)次下溢后产生更新事件。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。

*图18-6. 向下计数时序图,PSC=0/2*和*图18-7. 向下计数时序图,在运行时改变TIMERx_CAR寄存器值*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 18-6. 向下计数时序图，PSC=0/2

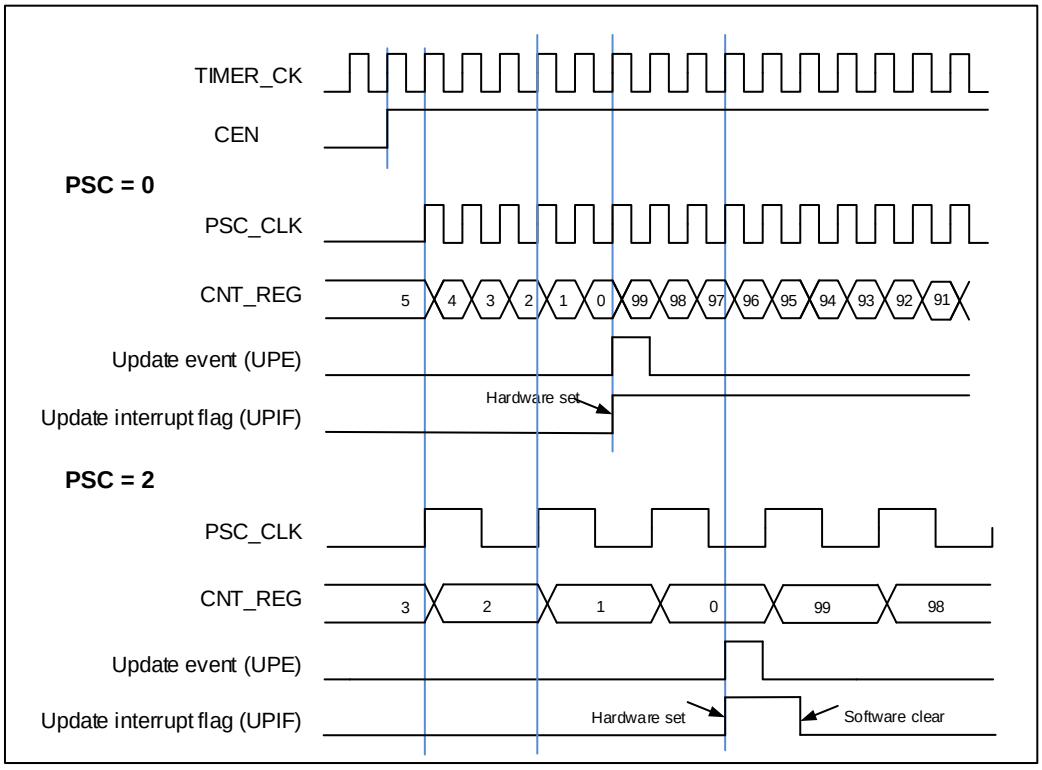
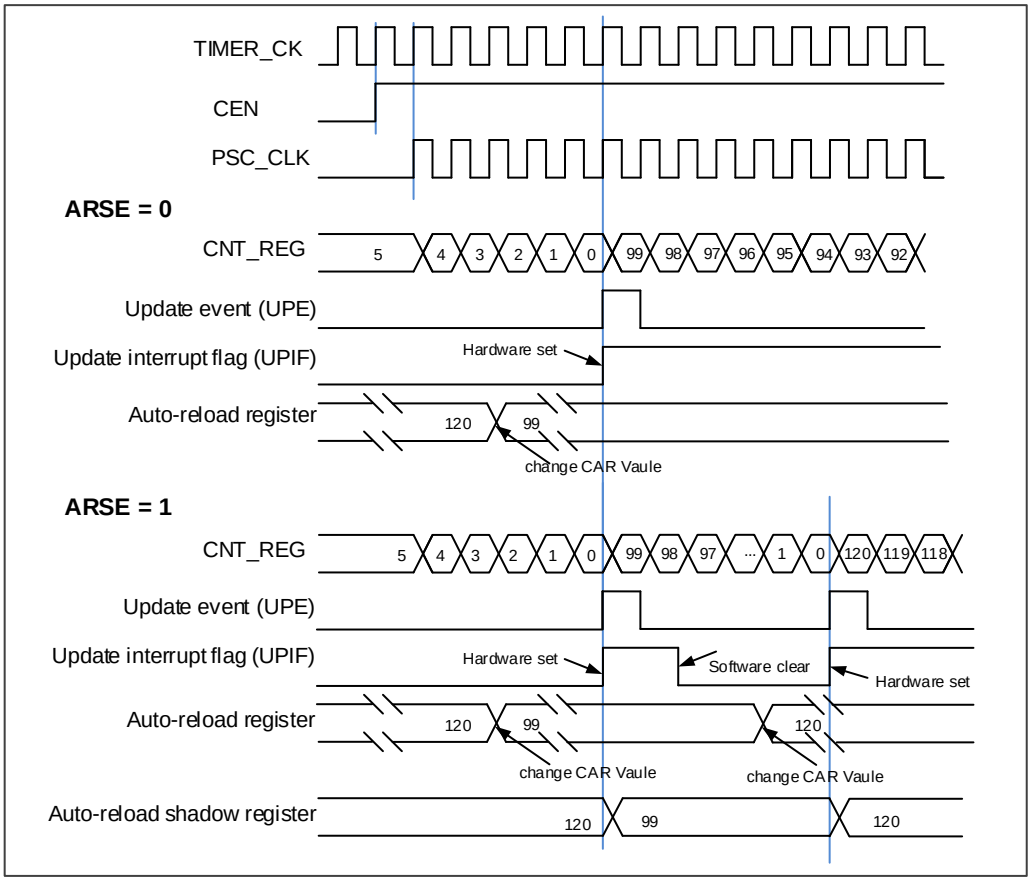


图 18-7. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值



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。

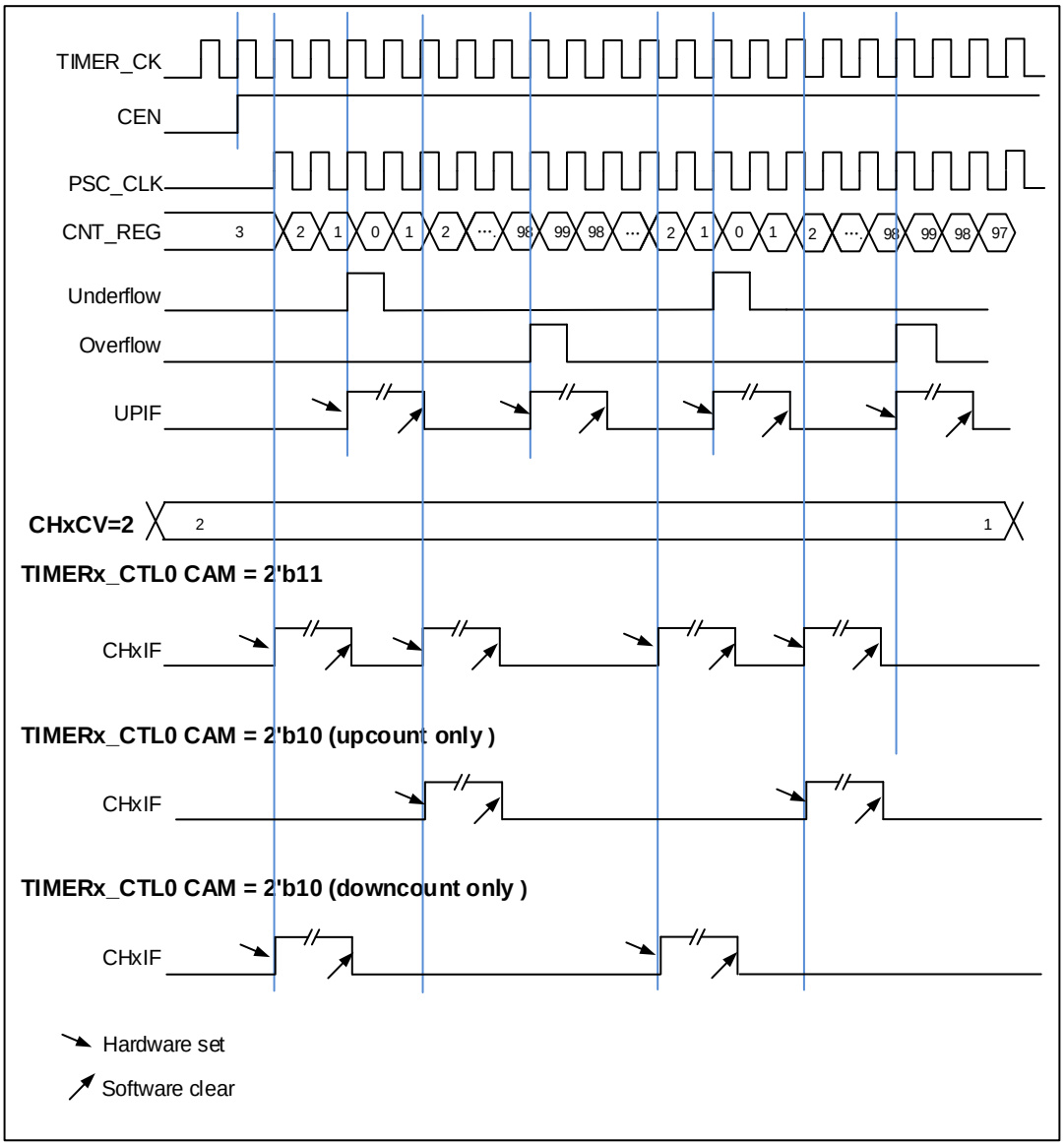
上溢或者下溢时，TIMERx_INTF 寄存器中的 UPIF 位都会被置 1，然而 CHxIF 位置 1 与 TIMERx_CTL0 寄存器中 CAM 的值有关。具体细节参考*图18-8. 中央计数模式计数器时序图*。

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 18-8. 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 18-8. 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 18-9. 中央计数模式下计数器重复时序图

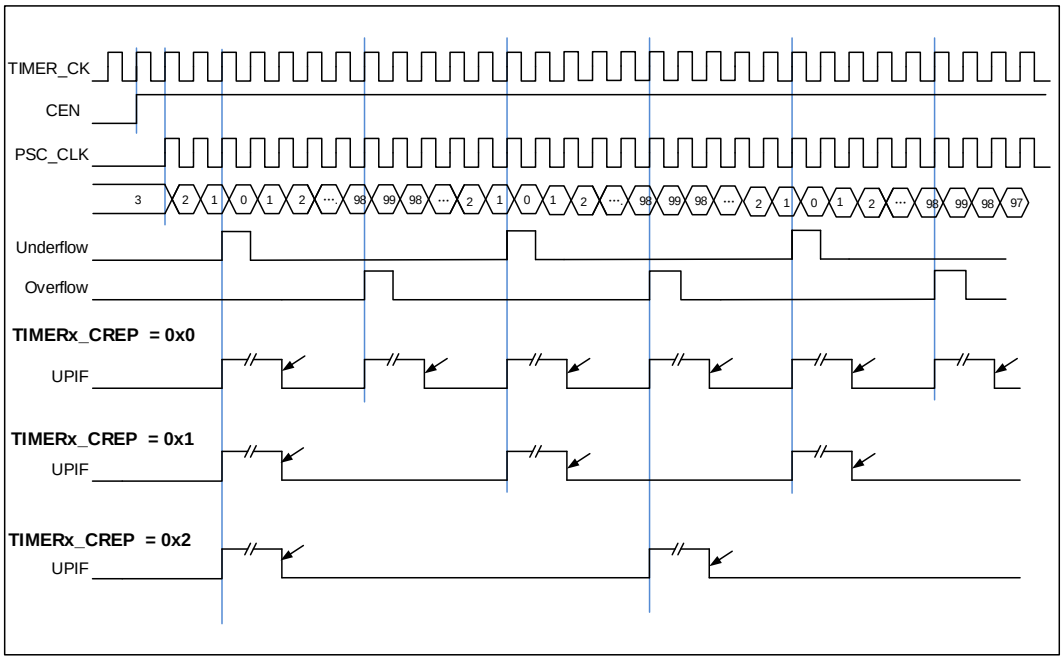


图 18-10. 在向上计数模式下计数器重复时序图

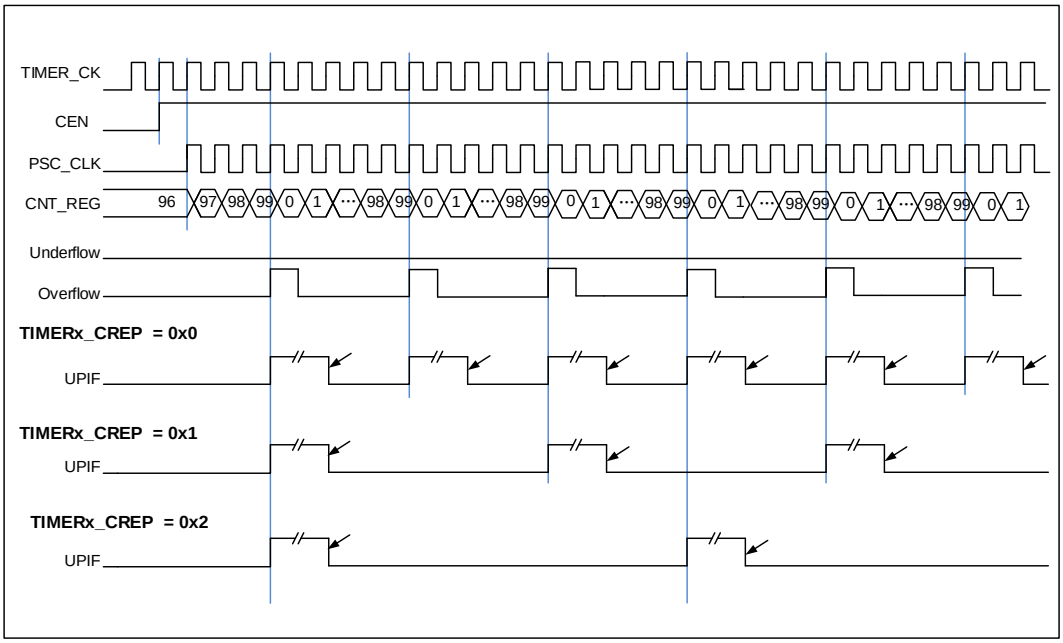
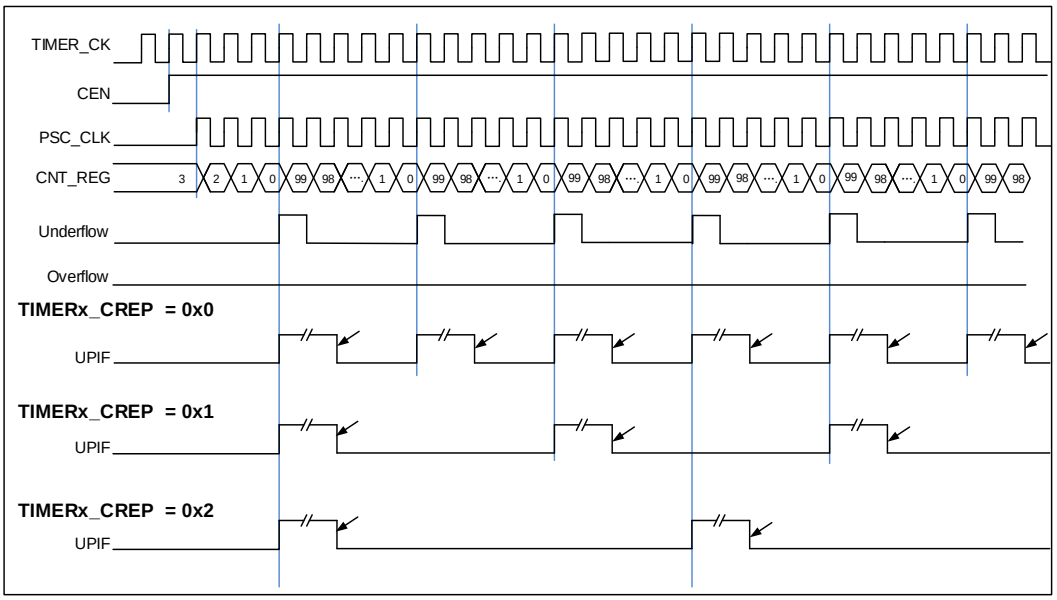


图 18-11. 在向下计数模式下计数器重复时序图



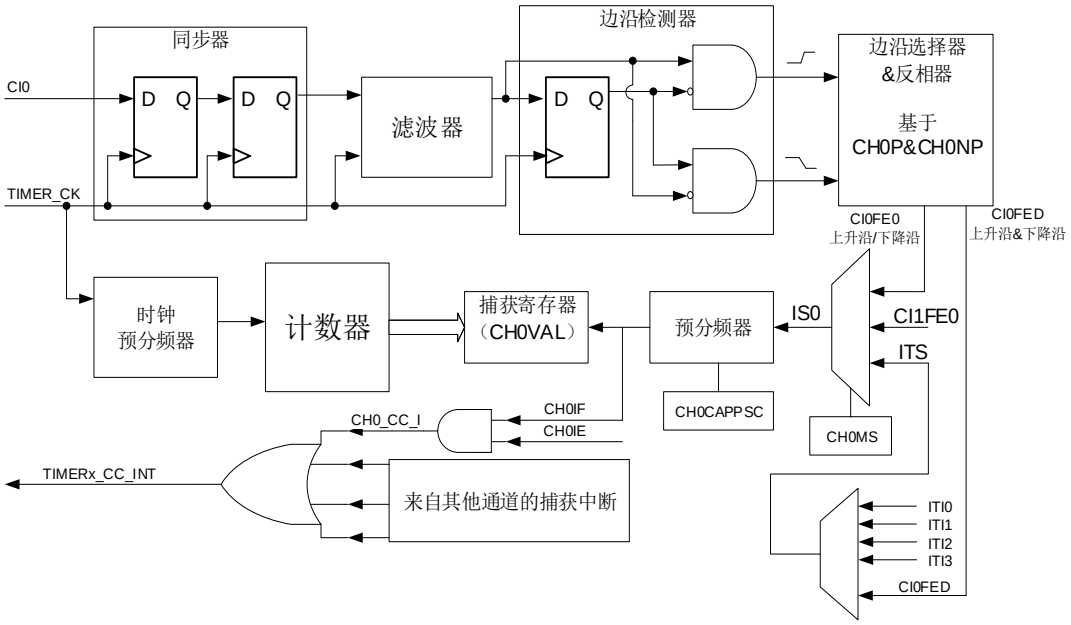
输入捕获和输出比较通道

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。

■ 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 18-12. 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**

图 18-13. 输出比较原理（带有互补输出的通道，x=0,1,2）

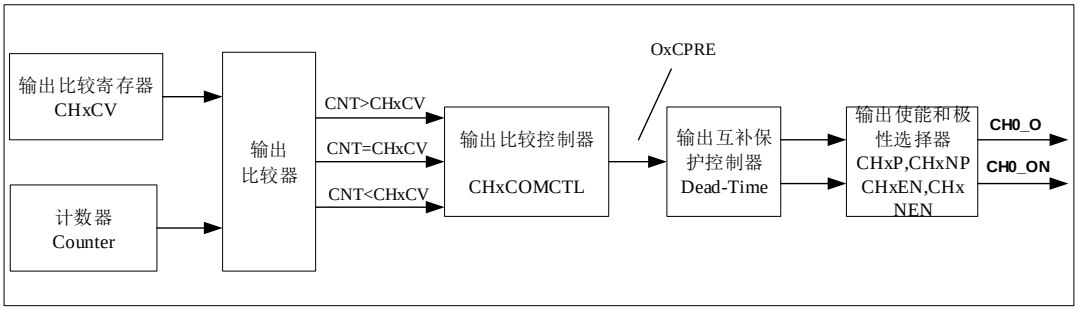


图 18-14. 输出比较原理

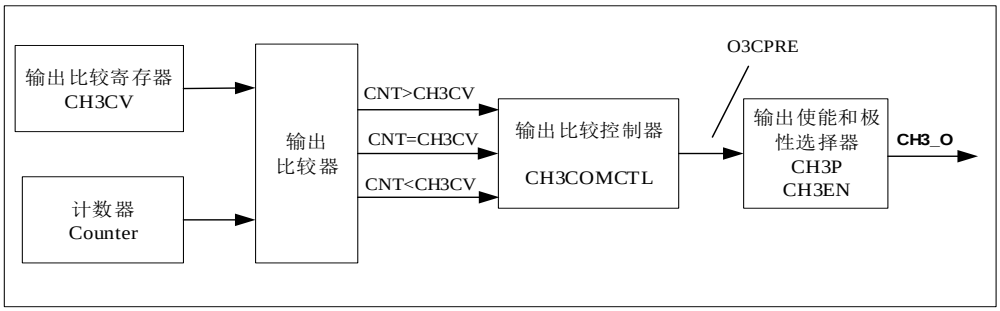


*图 18-13. 输出比较原理（带有互补输出的通道，x=0,1,2）*和*图 18-14. 输出比较原理*分别给出了输出比较的逻辑电路。通道输出信号 CHx_O/CHx_ON 与 OxCPRE 信号（详情请见*通道输准备信号*）的关系描述如下：OxCPRE 信号高电平有效，CHx_O/CHx_ON 的输出情况与 OxCPRE 信号，CHxP/CHxNP 位和 CHxEN/CHxNEN 位有关（具体情况请见 TIMERx_CHCTL2 寄存器中的描述）。例如：

1）当设置 CHxP=0（CHx_O 高电平有效，与 OxCPRE 输出极性相同）、CHxEN=1（CHx_O 输出使能）时：

若 OxCPRE 输出有效（高）电平，则 CHx_O 输出有效（高）电平；

若 OxCPRE 输出无效（低）电平，则 CHx_O 输出无效（低）电平。

2）当设置 CHxNP=1（CHx_ON 低电平有效，与 OxCPRE 输出极性相反）、CHxNEN=1（CHx_ON 输出使能）时：

若 OxCPRE 输出有效（高）电平，则 CHx_ON 输出有效（低）电平；

若 OxCPRE 输出无效（低）电平，则 CHx_ON 输出无效（高）电平。

当 CH0_O 和 CH0_ON 同时输出时，CH0_O 和 CH0_ON 的具体输出情况还与 TIMERx_CCHP 寄存器中的相关位（ROS、IOS、POE 和 DTCFG 等位）有关。详情请见*通道输出互补 PWM*。

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 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。

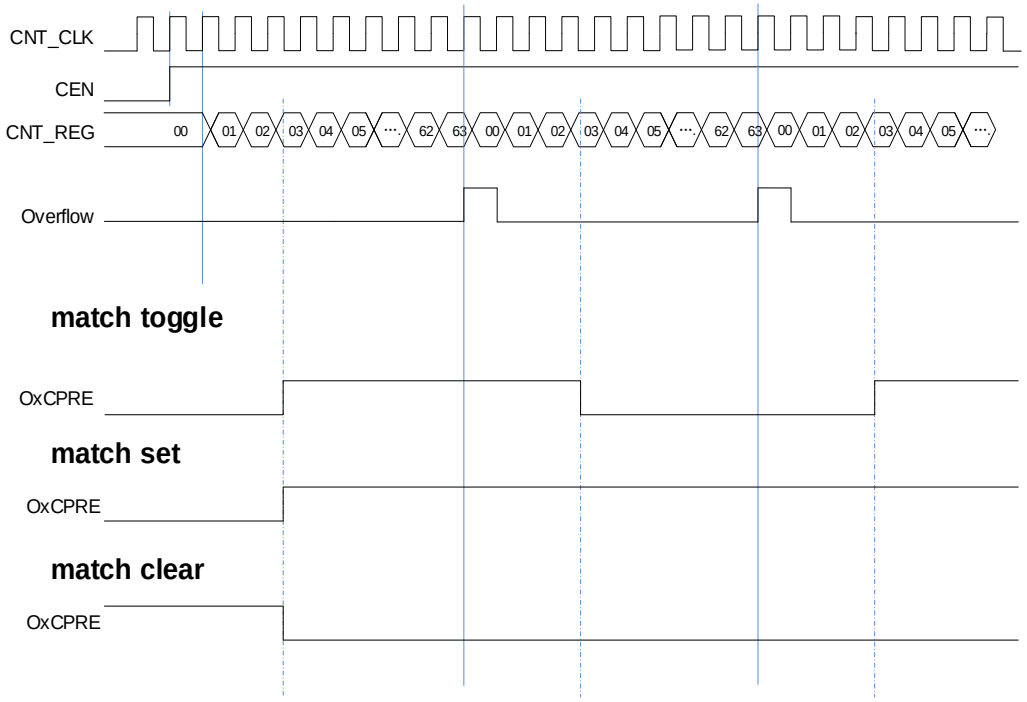
第四步：通过 TIMERx_CAR 寄存器和 TIMERx_CHxCV 寄存器配置输出比较时基：

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18-15. 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 18-15. 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 18-16. 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 18-17. 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 18-16. EAPWM 时序图

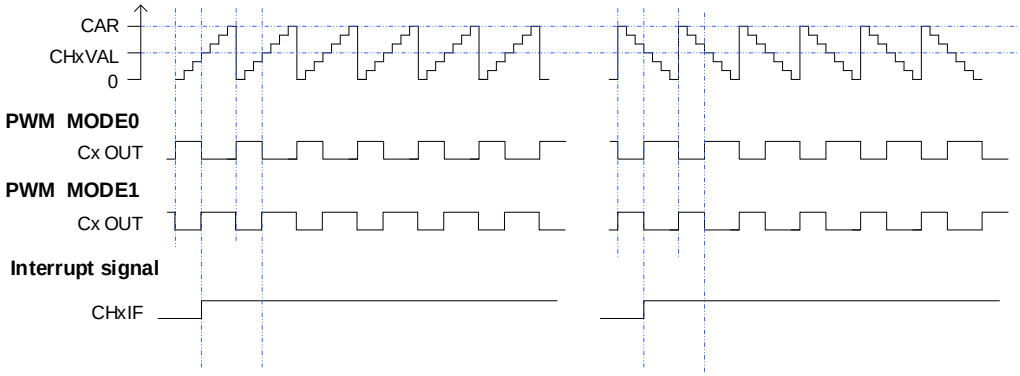
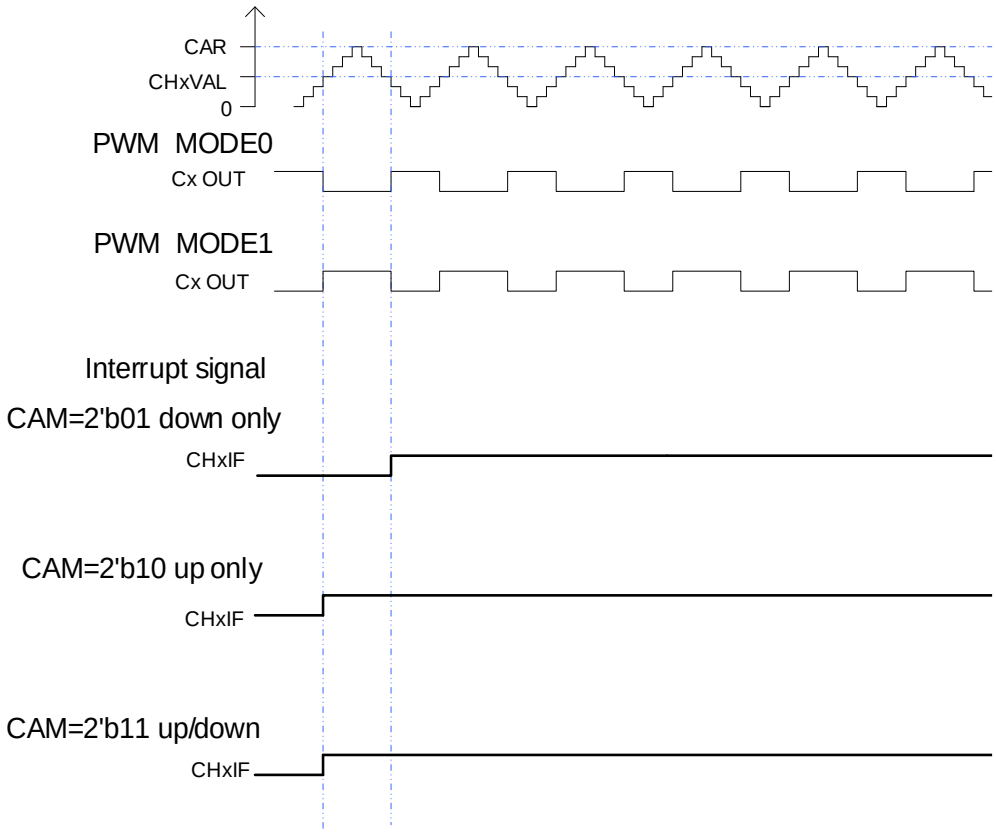


图 18-17. CAPWM 时序图



通道输准备信号

根据*图 18-13. 输出比较原理（带有互补输出的通道，x=0,1,2）*所示，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 18-2. 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="3">0</td><td>0</td><td>0</td><td colspan="2">CHx_O / CHx_ON = LOW
CHx_O / CHx_ON 输出禁能(1)</td></tr>
<tr><td rowspan="3">1</td><td>1</td><td colspan="2" rowspan="3">CHx_O/CHx_ON输出关闭状态(2)：
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：CHx_O = ISOx，CHx_ON = ISOxN (3)</td></tr>
<tr><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：
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：CHx_O = ISOx，CHx_ON = ISOxN</td></tr>
<tr><td rowspan="4">1</td><td rowspan="4">0</td><td rowspan="4">0/1</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O/CHx_ON = LOW
CHx_O/CHx_ON输出禁能</td></tr>
<tr><td>1</td><td>CHx_O = LOW
CHx_O输出禁能</td><td>CHx_ON=OxCPRE ⊕ (4)CHxNP
CHx_ON输出使能</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td>CHx_O=OxCPRE ⊕ CHxP
CHx_O输出使能</td><td>CHx_ON = LOW
CHx_ON输出禁能</td></tr>
<tr><td>1</td><td>CHx_O=OxCPRE ⊕ CHxP</td><td>CHx_ON=(!OxCPRE)(5) ⊕</td></tr>
</table>

互补参数					输出状态	
POEN	ROS	IOS	CHxEN	CHxNEN	CHx_O	CHx_ON
					CHx_O输出使能	CHxNP CHx_ON输出使能
	1		0	0	CHx_O = CHxP CHx_O输出关闭状态	CHx_ON = CHxNP CHx_ON输出关闭状态
				1	CHx_O = CHxP CHx_O输出关闭状态	CHx_O=OxCPRE⊕CHxNP CHx_ON输出使能
			1	0	CHx_O=OxCPRE⊕CHxP CHx_O输出使能	CHx_ON = CHxNP CHx_ON输出关闭状态
				1	CHx_O=OxCPRE⊕CHxP CHx_O输出使能	CHx_ON=（!OxCPRE）⊕ CHxNP 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。

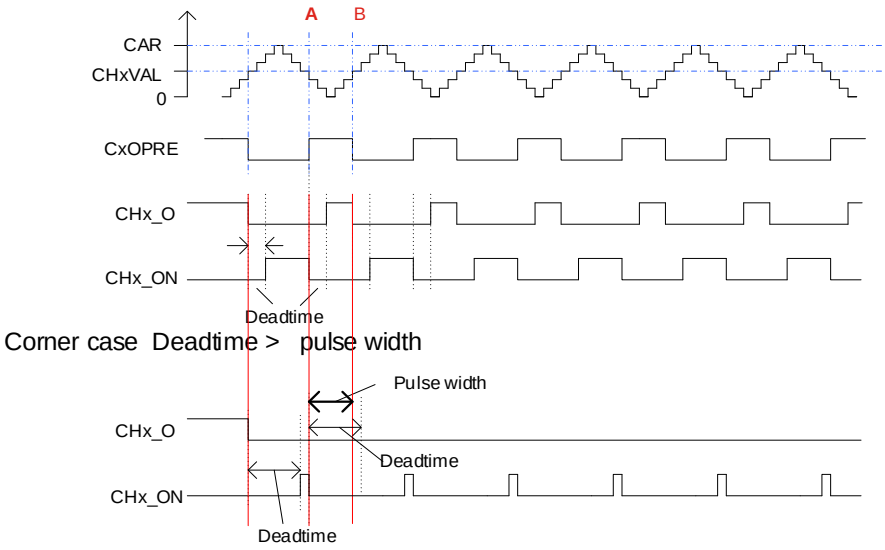
死区时间的插入，确保了通道互补的两路信号不会同时有效。

在 PWM0 模式，当通道 x 匹配发生时（TIMERx 计数器=TIMERx_CHxCV），OxCPRE 反转。在*图 18-18. 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。（如*图 18-18. 带死区时间的互补输出*所示）

图 18-18. 带死区时间的互补输出



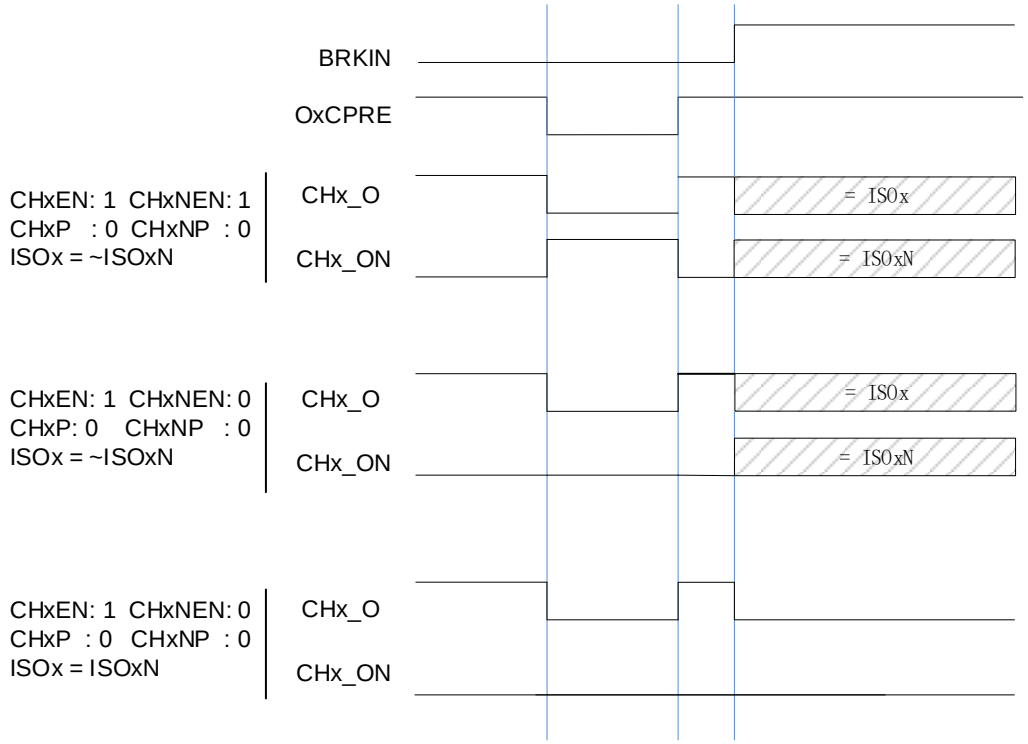
中止模式

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 18-19. 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，或者可以同时有 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 18-3. 不同正交译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 18-3. 不同正交译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
正交译码器模式0 SMC[2:0]=3'b001	CI1FE1=1	向下	向上	-	-
	CI1FE1=0	向上	向下	-	-
正交译码器模式1 SMC [2:0]=3'b010	CI0FE0=1	-	-	向上	向下
	CI0FE0=0	-	-	向下	向上
正交译码器模式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
	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 18-20. 在正交编码器模式 2 且 CI0FE0 极性不反相时计数器行为

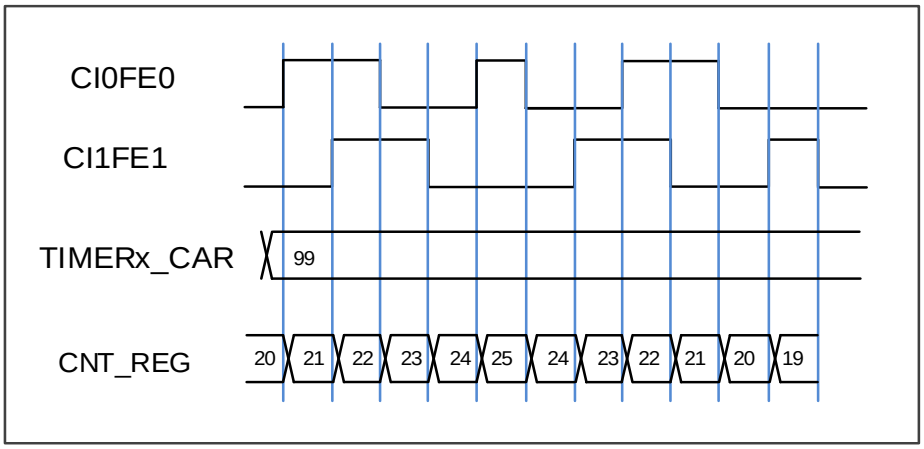
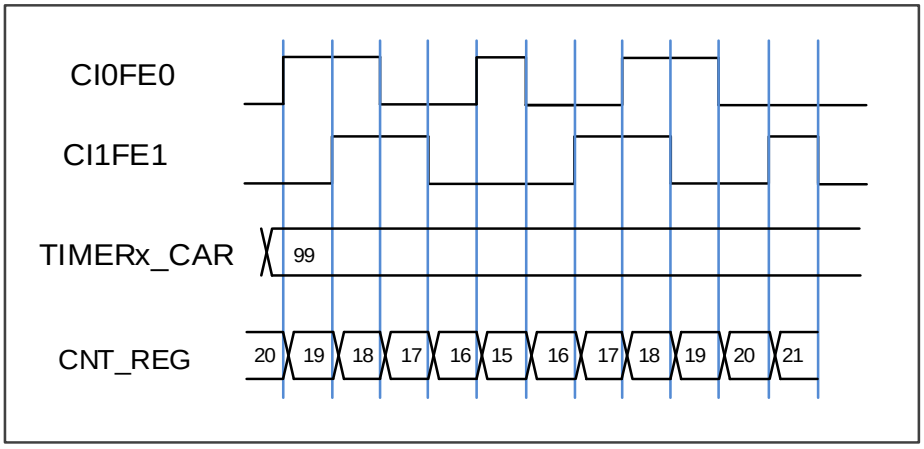


图 18-21. 在正交译码器模式 2 且 CI0FE0 极性反相时计数器行为



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18-22. 霍尔传感器用在BLDC电机控制中*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 18-22. 霍尔传感器用在 BLDC 电机控制中

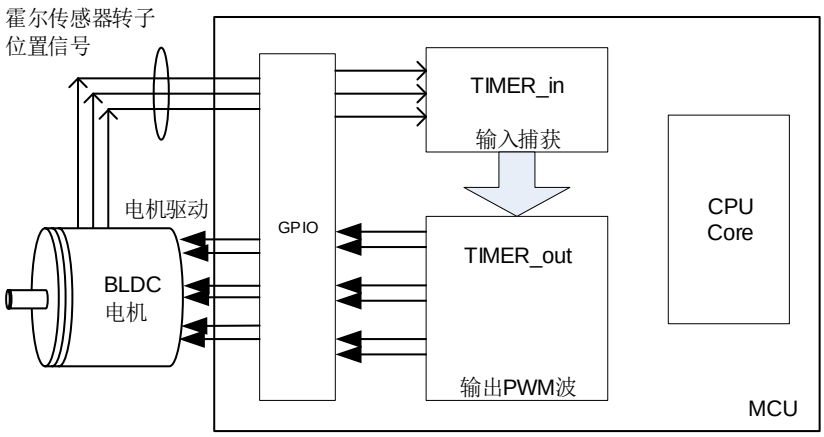
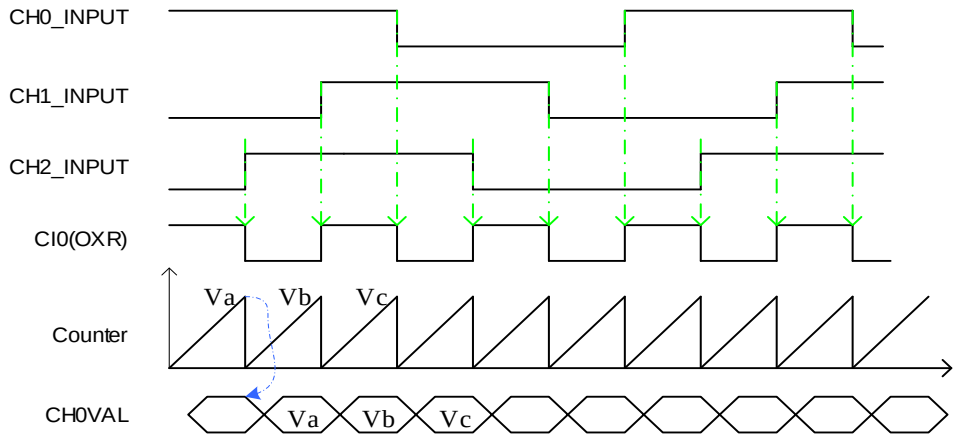
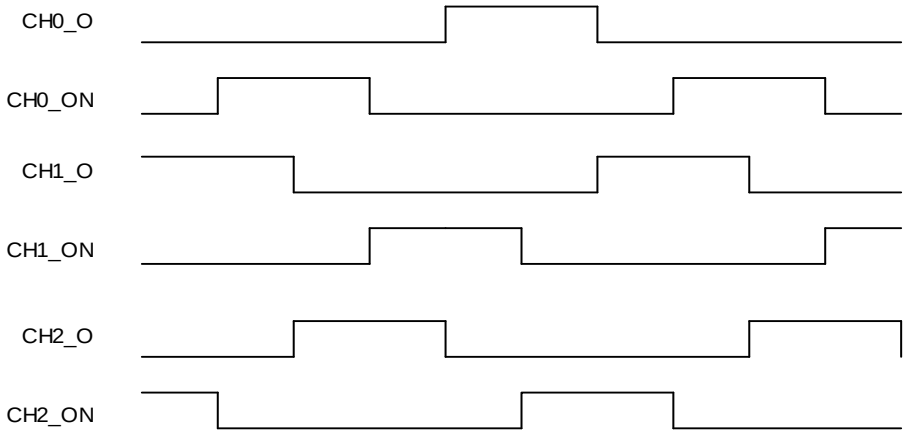


图 18-23. 两个定时器之间的霍尔传感器时序图

高级/通用 L0定时器 TIMER_in 工作在输入捕获模式



高级定时器 TIMER_out 工作在输出比较模式(带有死区的PWM)

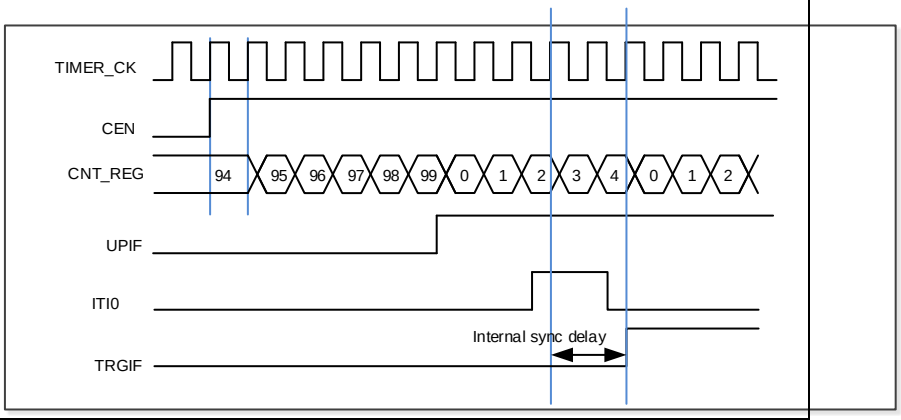
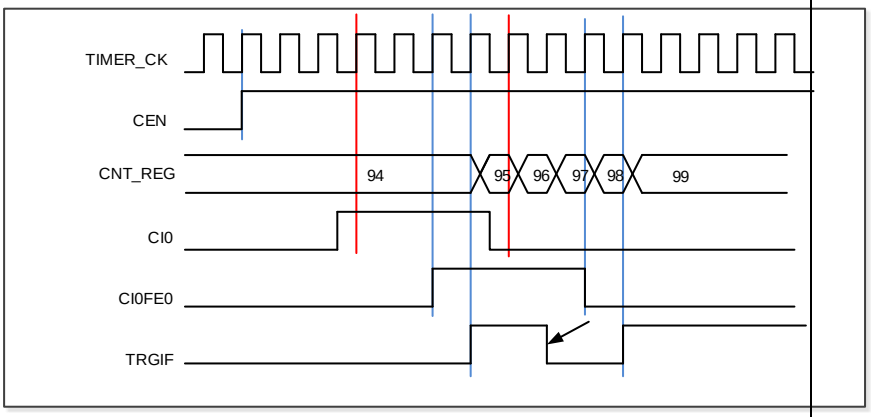


主-从管理

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。

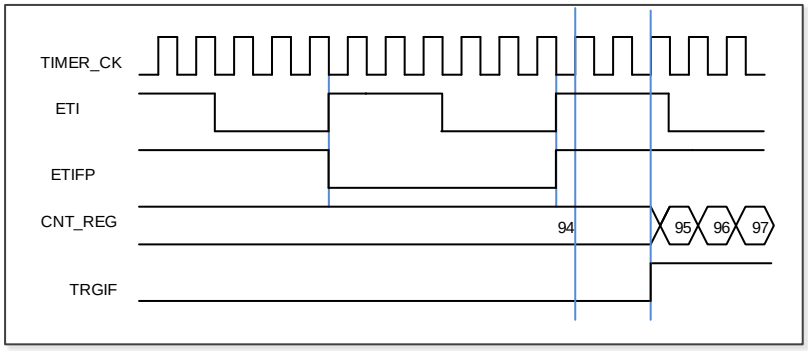
表 18-4. 从模式例子列表

	模式选择	触发源选择	极性选择	滤波和预分频
列举	SMC[2:0] 3'b100 (复位模式) 3'b101 (暂停模式) 3'b110 (事件模式)	TRGS[2:0] 000: ITI0 001: ITI1 010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0 110: CI1FE1 111: ETIFP	如果触发源是CI0FE0或者CI1FE1，配置CHxP和CHxNP来选择极性和反相 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 触发源CIx，配置CHxCAPFLT设置滤波，分频不可用 触发源是ETIF，滤波和预分频不可用

<table>
<tr><th></th><th>模式选择</th><th>触发源选择</th><th>极性选择</th><th>滤波和预分频</th></tr>
<tr><td>例1</td><td>复位模式
当触发输入上升沿，计数器清零重启</td><td>TRGIS[2:0]=3’b000
选择ITI0为触发源</td><td>触发源是ITI0，极性选择不可用</td><td>触发源是 ITI0，滤波和预分频不可用</td></tr>
<tr><td colspan="4">图 18-24. 复位模式
 
</td></tr>
<tr><td>例2</td><td>暂停模式
当触发输入为低的时候，计数器暂停计数</td><td>TRGIS[2:0]=3’b101
选择CI0FE0为触发源</td><td>TI0S=0.（非异或）[CH0NP==0, CH0P==0]不反相.在上升沿捕获</td><td>在这个例子中滤波被旁路</td></tr>
<tr><td colspan="4">图 18-25. 暂停模式下的控制电路
 
</td></tr>
<tr><td>例3</td><td>事件模式
触发输入的上升沿计数器开始计数</td><td>TRGIS[2:0]=3’b111
选择ETIF为触发源.</td><td>ETP = 0 没有极性改变</td><td>ETPSC = 1, 2分频.
ETFC = 0，无滤波</td></tr>
</table>

	模式选择	触发源选择	极性选择	滤波和预分频

图 18-26. 事件模式



单脉冲模式

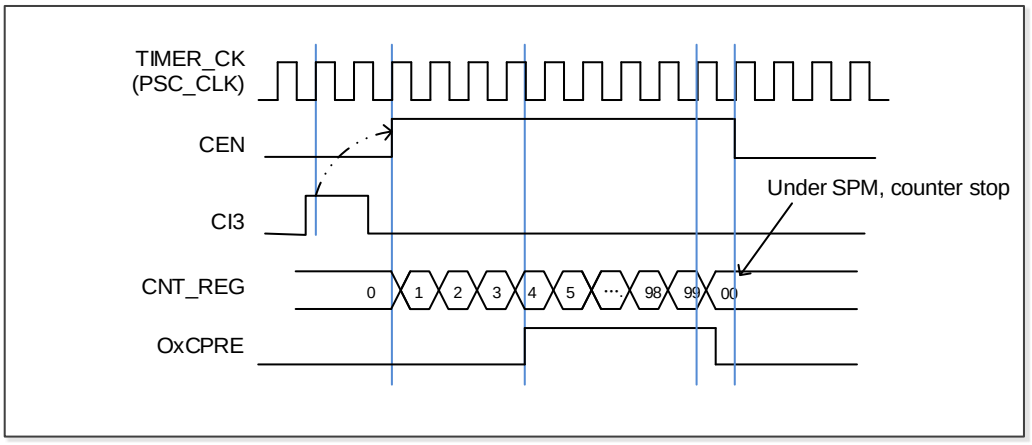
单脉冲模式与重复模式是相反的，设置 TIMERx_CTL0 寄存器的 SPM 位置 1，则使能单脉冲模式。当 SPM 置 1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置 CHxCOMCTL 配置 TIMERx 为 PWM 模式或者比较模式。

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18-27. 单脉冲模式，TIMERx_CHxCV = 4，TIMERx_CAR=99 展示了一个例子。

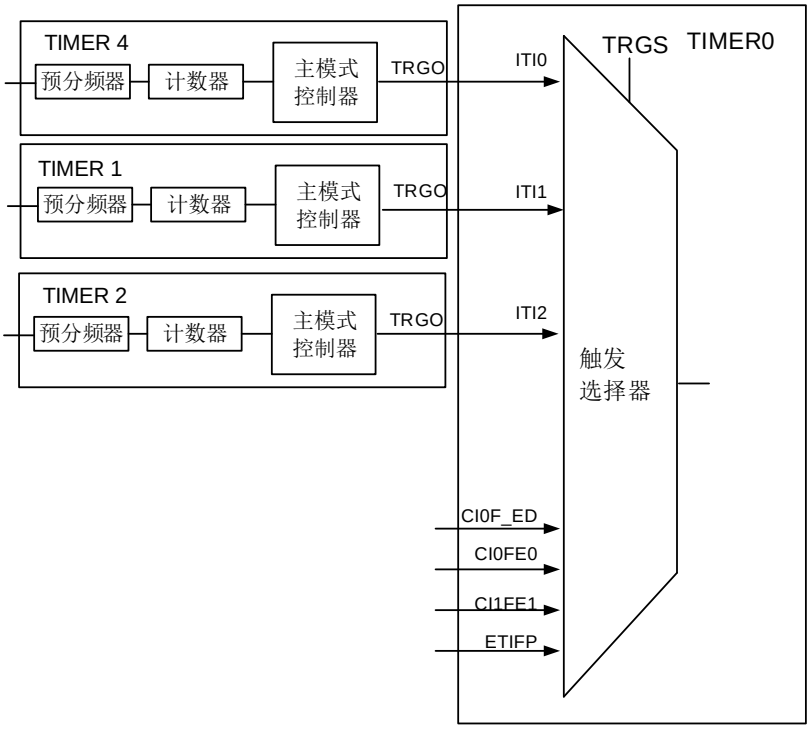
图 18-27. 单脉冲模式，TIMERx_CHxCV = 4，TIMERx_CAR=99



定时器互连

定时器之间可配置为内部级联，一个定时器配置为主模式输出 TRGO 信号，另一个定时器配置为从模式，TRGO 信号包括复位事件、使能事件、更新事件、捕获比较脉冲事件、比较事件。从定时器接收到 ITIx 信号，并执行对应的操作，包括内部时钟模式、正交编码模式、复位模式、暂停模式、事件模式、外部时钟模式。

图 18-28. 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 18-28. 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)。

- 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)；

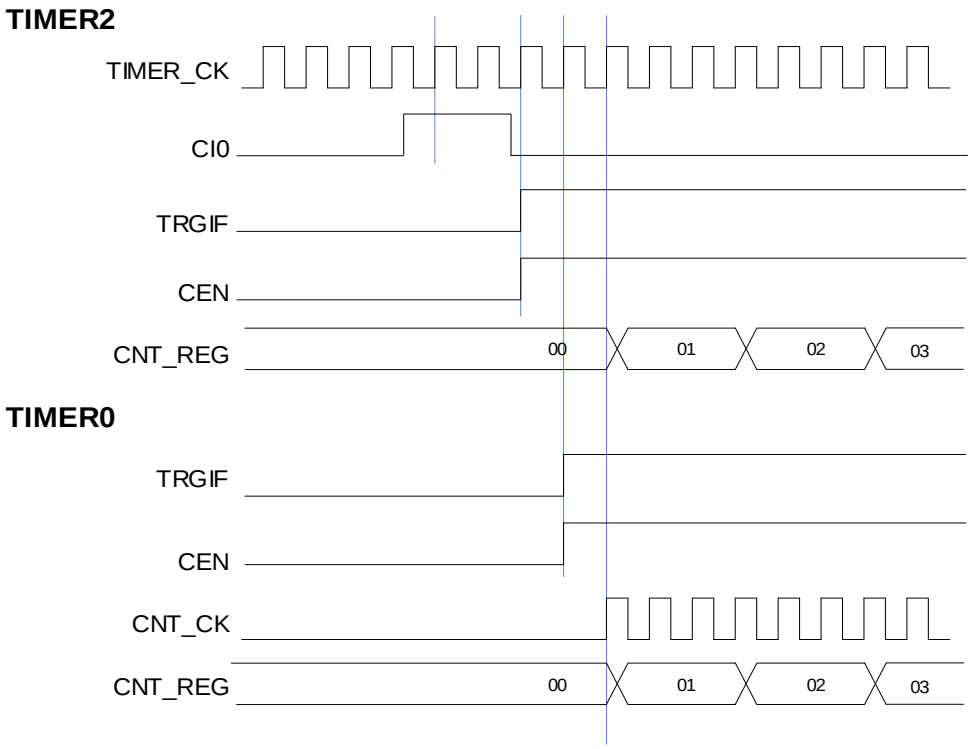
3. 写 MSM=1(TIMER2_SMCFG 寄存器)来配置定时器 2 工作在主/从模式；

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 18-29. 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 and 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®-M33 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。

18.1.5. 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

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	CAM[1:0]		DIR	SPM	UPS	UPDIS	CEN
						rw		rw	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 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。 00：$f_{DTS}=f_{CK_TIMER}$ 01：$f_{DTS}= f_{CK_TIMER}$ /2 10：$f_{DTS}= f_{CK_TIMER}$ /4 11：保留
7	ARSE	自动重载影子使能 0：禁能 TIMERx_CAR 寄存器的影子寄存器 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 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 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 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 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 当计数器使能以后，该位不能从 0x00 切换到非 0x00
4	DIR	方向 0：向上计数

1：向下计数

当计数器配置为中央对齐计数模式或正交译码器模式时，该位只读。

3	SPM	单脉冲模式 0：单脉冲模式禁能。更新事件发生后，计数器继续计数 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 软件配置该位，选择更新事件源. 0：以下事件均会产生更新中断或DMA请求： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 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. 该位用来使能或禁能更新事件的产生 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： UPG位被置1 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 0：计数器禁能 1：计数器使能 在软件将 CEN 位置 1 后，外部时钟、暂停模式和正交译码器模式才能工作。

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 参考 ISO0 位
13	ISO2N	通道 2 的互补通道空闲状态输出 参考 ISO0N 位
12	ISO2	通道 2 的空闲状态输出 参考 ISO0 位
11	ISO1N	通道 1 的互补通道空闲状态输出 参考 ISO0N 位
10	ISO1	通道 1 的空闲状态输出 参考 ISO0 位
9	ISO0N	通道 0 的互补通道空闲状态输出 0：当 POEN 复位，CH0_ON 设置低电平. 1：当 POEN 复位，CH0_ON 设置高电平 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 0：当 POEN 复位，CH0_O 设置低电平 1：当 POEN 复位，CH0_O 设置高电平 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 0：选择 TIMERx_CH0 引脚作为通道 0 的触发输入 1：选择 TIMERx_CH0, CH1 and 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 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 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： 主定时器产生一个复位事件 TIMERx_SWEVG寄存器中UPG位置1 001：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： CEN位置1 在暂停模式下，触发输入置1 010：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 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 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：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 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 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

0：当通道捕获/比较事件发生时，发送通道 x 的 DMA 请求 .
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 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： 0：CMTG 位被置 1 时更新影子寄存器 1：当 CMTG 位被置 1 或检测到 TRIGI 上升沿时，影子寄存器更新 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 0：影子寄存器 CHxEN, CHxNEN 和 CHxCOMCTL 位禁能. 1：影子寄存器 CHxEN, CHxNEN 和 CHxCOMCTL 位使能. 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 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 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。

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

		110: CI1FE1 111: ETIFP 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 000：关闭从模式. 如果 CEN=1，则预分频器直接由内部时钟驱动 001：正交译码器模式 0. 根据 CI1FE1 的电平，计数器在 CI0FE0 的边沿向上/下计数 010：正交译码器模式 1. 根据 CI0FE0 的电平，计数器在 CI1FE1 的边沿向上/下计数 011：正交译码器模式 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 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. 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 110：事件模式.计数器在触发输入的上升沿启动。 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 0：禁止触发 DMA 请求 1：使能触发 DMA 请求
13	CMTDEN	换相 DMA 更新请求使能 0：禁止换相 DMA 更新请求 1：使能换相 DMA 更新请求
12	CH3DEN	通道 3 比较/捕获 DMA 请求使能 0：禁止通道 3 比较/捕获 DMA 请求

		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 0：禁止通道 2 比较/捕获 DMA 请求 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 0：禁止通道 1 比较/捕获 DMA 请求 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 0：禁止通道 0 比较/捕获 DMA 请求 1：使能通道 0 比较/捕获 DMA 请求
8	UPDEN	更新 DMA 请求使能 0：禁止更新 DMA 请求 1：使能更新 DMA 请求
7	BRKIE	中止中断使能 0：禁止中止中断 1：使能中止中断
6	TRGIE	触发中断使能 0：禁止触发中断 1：使能触发中断
5	CMTIE	换相更新中断使能 0：禁止换相更新中断 1：使能换相更新中断
4	CH3IE	通道 3 比较/捕获中断使能 0：禁止通道 3 中断 1：使能通道 3 中断
3	CH2IE	通道 2 比较/捕获中断使能 0：禁止通道 2 中断 1：使能通道 2 中断
2	CH1IE	通道 1 比较/捕获中断使能 0：禁止通道 1 中断 1：使能通道 1 中断
1	CH0IE	通道 0 比较/捕获中断使能 0：禁止通道 0 中断 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

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 参见 CH0OF 描述
11	CH2OF	通道 2 捕获溢出标志 参见 CH0OF 描述
10	CH1OF	通道 1 捕获溢出标志 参见 CH0OF 描述
9	CH0OF	通道 0 捕获溢出标志 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. 0：无捕获溢出中断发生 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 当中止输入有效时，由硬件对该位置‘1’。 当中止输入无效时，则该位可由软件清‘0’。 0：无中止事件产生 1：中止输入上检测到有效电平
6	TRGIF	触发中断标志 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。 0：无触发事件产生 1：触发中断产生
5	CMTIF	通道换相更新中断标志 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 参见 CH0IF 描述
3	CH2IF	通道 2 比较/捕获中断标志 参见 CH0IF 描述
2	CH1IF	通道 1 比较/捕获中断标志 参见 CH0IF 描述
1	CH0IF	通道 0 比较/捕获中断标志 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。 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 此位在任何更新事件发生时由硬件置 1，软件清 0。 0：无更新中断发生 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 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。 0：不产生中止事件 1：产生中止事件
6	TRGG	触发事件产生 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。
0：无触发事件产生
1：产生触发事件

5	CMTG	通道换相更新事件发生 此位由软件置 1，由硬件自动清 0. 当此位被置 1，通道捕获/比较控制寄存器 (CHxEN, CHxNEN 和 CHxCOMCTL) 的互补输出被更新。 0：不产生通道控制更新事件 1：产生通道控制更新事件
4	CH3G	通道 3 捕获或比较事件发生 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 参见 CH0G 描述
1	CH0G	通道 0 捕获或比较事件发生 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 TIMERx_CH0CV 寄存器捕获，如果 CH0IF 标志位已经为 1，则 CH0OF 标志位被置 1。 0：不产生通道 0 捕获或比较事件 1：发生通道 0 捕获或比较事件
0	UPG	更新事件产生 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
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 参见 CH0COMCEN 描述
14:12	CH1COMCTL[2:0]	通道 1 输出比较模式 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 参见 CH0COMFEN 描述
9:8	CH1MS[1:0]	通道 1 模式选择 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。 00：通道 1 配置为输出 01：通道 1 配置为输入，IS1 映射在 CI1FE1 上 10：通道 1 配置为输入，IS1 映射在 CI0FE1 上 11：通道 1 配置为输入，IS1 映射在 ITS 上 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 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 0：禁止通道 0 输出比较清零 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 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。 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 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。 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。 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 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。 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 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 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 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 参见 CH0COMCEN 描述
14:12	CH3COMCTL[2:0]	通道 3 输出比较模式 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 参见 CH0COMFEN 描述
9:8	CH3MS[1:0]	通道 3 模式选择 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。 00：通道 3 配置为输出 01：通道 3 配置为输入，IS3 映射在 CI3FE3 上 10：通道 3 配置为输入，IS3 映射在 CI2FE3 上 11：通道 3 配置为输入，IS3 映射在 ITS 上 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 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 0：使能通道 2 输出比较清零 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 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。 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 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。 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。 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。 100：强制为低。强制 O2CPRE 为低电平 101：强制为高。强制 O2CPRE 为高电平 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。

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。
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。
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 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。 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 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 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 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

Bits	Fields	Descriptions
31:14	保留	必须保持复位值
13	CH3P	通道 3 极性 参考 CH0P 描述
12	CH3EN	通道 3 使能 参考 CH0EN 描述
11	CH2NP	通道 2 互补输出极性 参考 CH0NP 描述
10	CH2NEN	通道 2 互补输出使能 参考 CH0NEN 描述
9	CH2P	通道 2 极性 参考 CH0P 描述
8	CH2EN	通道 2 使能 参考 CH0EN 描述
7	CH1NP	通道 1 互补输出极性 参考 CH0NP 描述
6	CH1NEN	通道 1 互补输出使能 参考 CH0NEN 描述
5	CH1P	通道 1 极性 参考 CH0P 描述
4	CH1EN	通道 1 使能 参考 CH0EN 描述
3	CH0NP	通道 0 互补输出极性 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。 0：通道0互补输出高电平为有效电平 1：通道0互补输出低电平为有效电平 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。 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。 0：禁止通道 0 互补输出 1：使能通道 0 互补输出
1	CH0P	通道 0 极性 当通道 0 配置为输出模式时，此位定义了输出信号极性。 0：通道0高电平为有效电平 1：通道0低电平为有效电平 当通道 0 配置为输入模式时，此位定义了 CI0 信号极性

		[CH0NP, CH0P] 将选择 CI0FE0 或者 CI1FE0 的有效边沿或者捕获极性 [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。 [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。 [CH0NP==1, CH0P==0]：保留。 [CH0NP==1, CH0P==1]：保留。 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 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。 0：禁止通道 0 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															

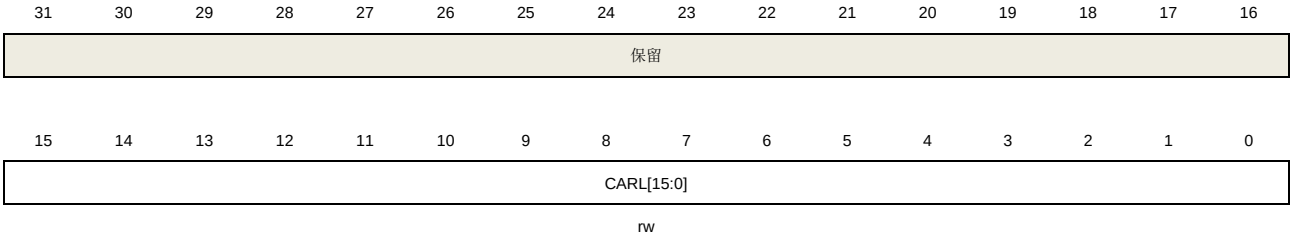
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 TIMER_CK 时钟除以(PSC+1)，每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。

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



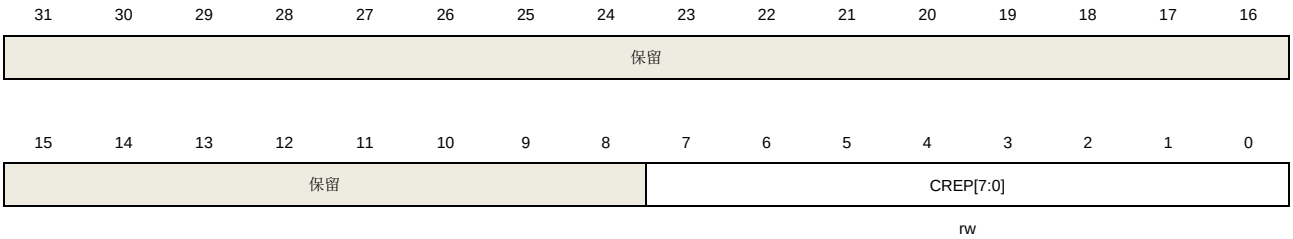
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

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



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 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 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。 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															

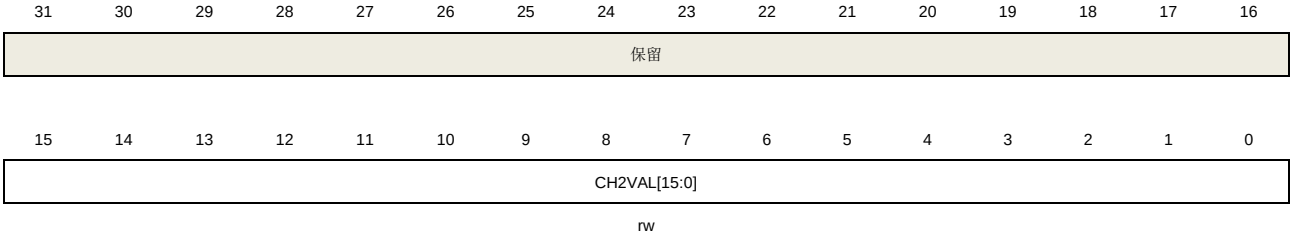
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH1VAL[15:0]	通道 1 的捕获或比较值 当通道 1 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 1 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

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



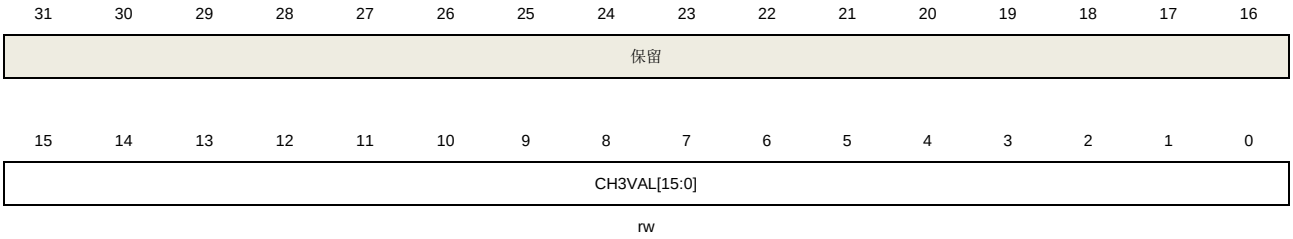
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH2VAL[15:0]	通道 2 的捕获或比较值 当通道 2 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 2 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

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



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 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。 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 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
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
31:16	保留	必须保持复位值
15	POEN	所有的通道输出使能 该位通过以下方式置 1： –写 1 置位 –如果 OAEN=1，则在下一次更新事件发生时置 1. 该位通过以下方式清 0： –写 0 清 0 –有效的中止输入（异步） 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。 0：禁止通道输出 1：使能通道输出 注意：仅当 CHxMS[1:0]=2’b00 时该位有效
14	OAEN	自动输出使能 0：POEN 位只能使用软件方式置 1 1：如果中止输入无效，下一次更新事件发生时，POEN 位将会置 1 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 此位定义了中止输入信号 BRKIN 的极性。 0：中止输入低电平有效 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 CKM 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”。

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”。

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 位写保护。

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：

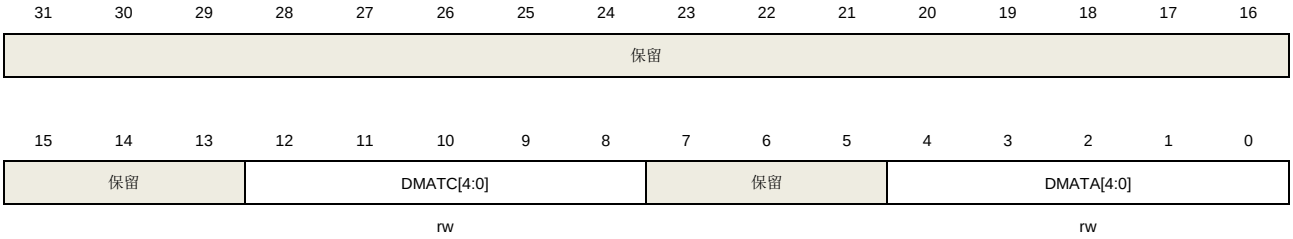
1. t_{DTS_CK} 是 DTS_CK 的周期，由 TIMERx_CTL0 中的 CKDIC[1:0]定义。
2. 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]=00 时才可修改。

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



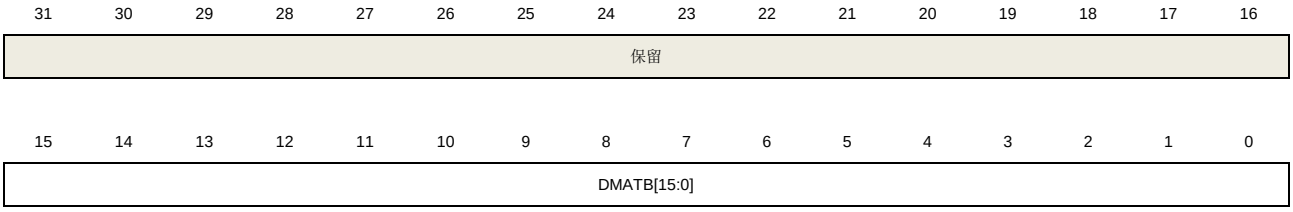
位/位域	名称	描述
31:13	保留	必须保持复位值.
12:8	DMATC [4:0]	DMA 传输计数 该位域定义了 DMA 访问（读写）TIMERx_DMATB 寄存器的数量 n，n = (DMATC [4:0] +1). DMATC [4:0] 从 5'b0_0000 到 5'b1_0001.
7:5	保留	必须保持复位值
4:0	DMATA [4:0]	DMA 传输起始地址 该位域定义了 DMA 访问 TIMERx_DMAVB 寄存器的第一个地址。当通过 TIMERx_DMA 第一次访问时，访问的就是该位域指定的地址。第二次访问 TIMERx_DMAVB 时，将访问起始地址+0x4。

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



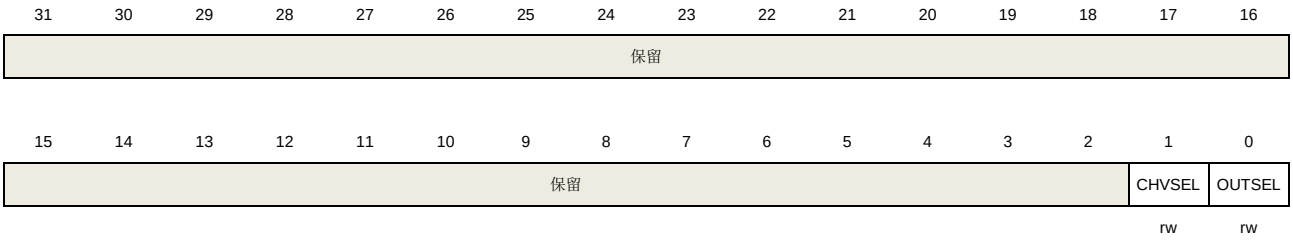
位/位域	名称	描述
31:16	保留	必须保持复位值.
15:0	DMATB [15:0]	DMA 发送缓冲 对这个寄存器的读或写，（起始地址+传输次数*4）地址范围内的寄存器会被访问传输次数由硬件计算，范围为 0 到 DMATC。

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 此位由软件写 1 或清 0。 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 0：无影响
0	OUTSEL	输出值选择位 此位由软件写 1 或清 0。 1：如果 POEN 位与 IOS 位均为 0，则输出无效 0：无影响

18.2. 通用定时器 L0（TIMERx, x=1,2,3,4）

18.2.1. 简介

通用定时器 L0（定时器 1，2，3，4）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（定时器 2~4）或 32 位（定时器 1）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。

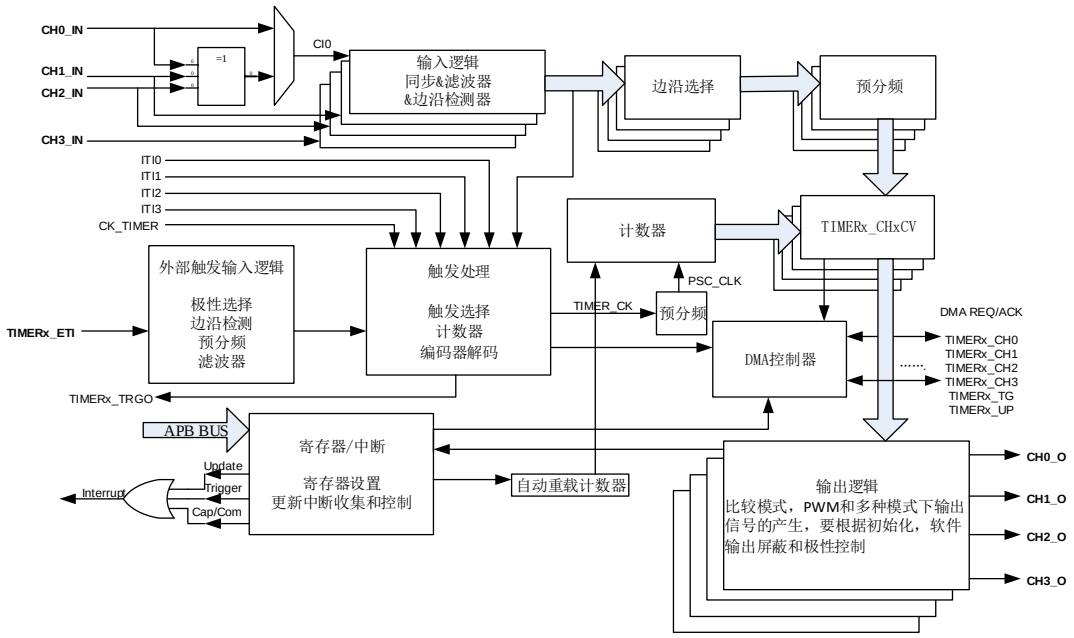
18.2.2. 主要特性

- 总通道数：4；
- 计数器宽度：16位（定时器2~4），32位（定时器1）；
- 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；
- 正交译码器接口：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。

18.2.3. 结构框图

图 18-30. 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 18-30. 通用定时器 L0 结构框图



18.2.4. 功能描述

时钟源配置

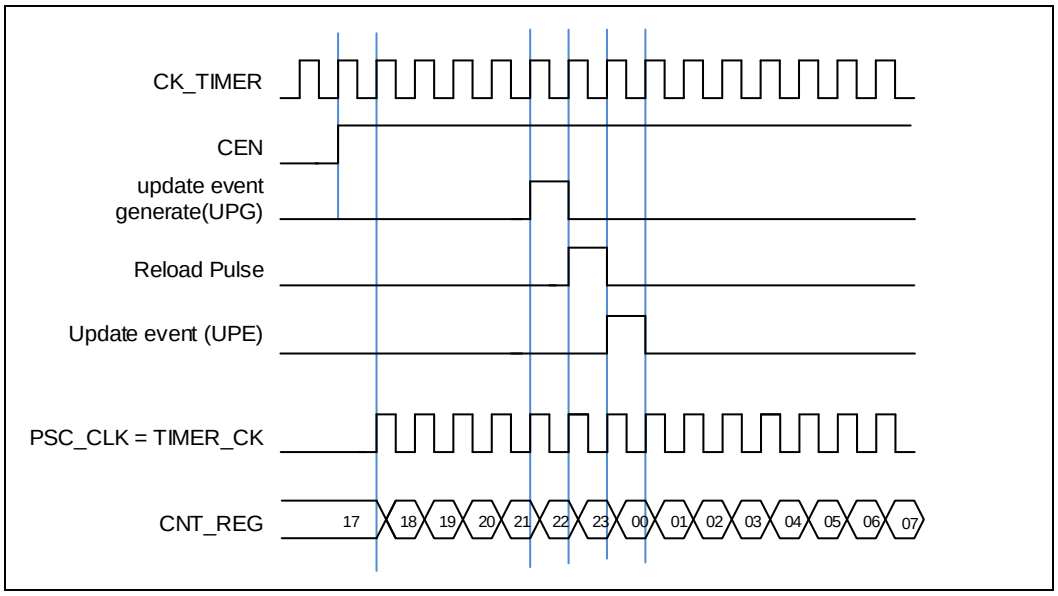
通用定时器 L0 可以由内部时钟源 CK_TIMER 或者由 SMC（TIMERx_SMCFG 寄存器位[2:0]）控制的复用时钟源驱动。

- 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 18-31. 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。

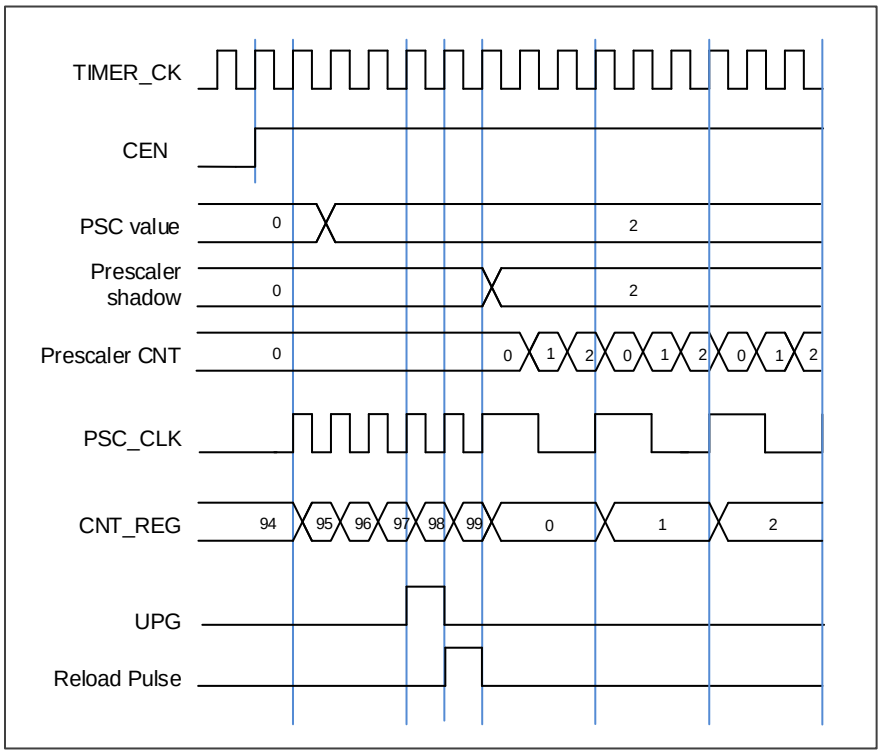
■ SMC==1’b1(外部时钟模式1)，定时器选择外部输入引脚ETI作为时钟源

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 18-32. 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 18-33. 向上计数时序图，PSC=0/2 和 *图 18-34. 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 18-33. 向上计数时序图，PSC=0/2

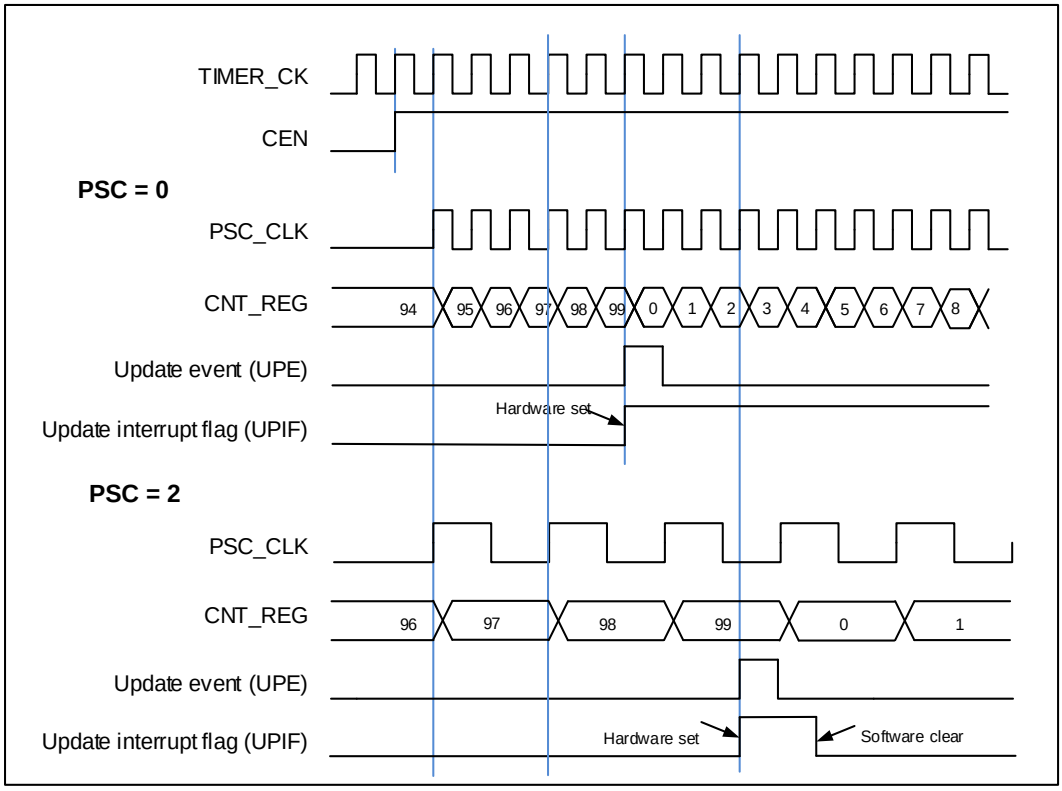
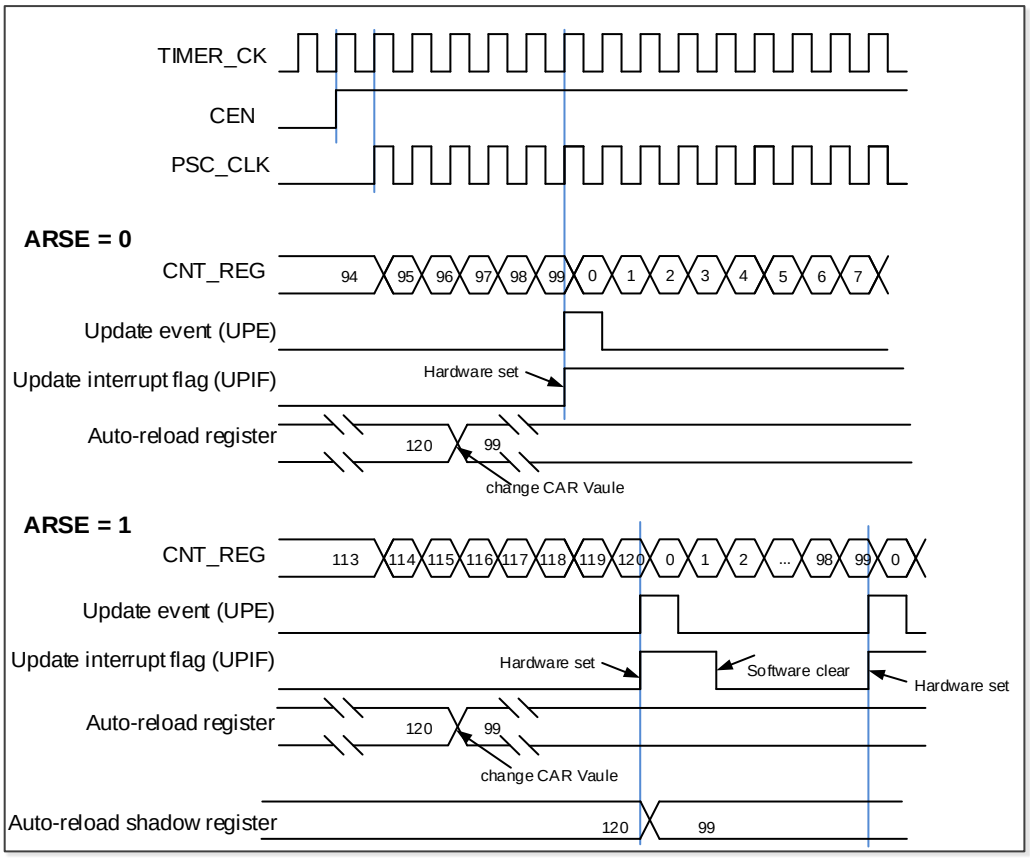


图 18-34. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



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。

图 18-35. 向下计数时序图，PSC=0/2 和 *图 18-36. 向下计数时序图，在运行时改变 TIMERx CAR 寄存器值*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 18-35. 向下计数时序图，PSC=0/2

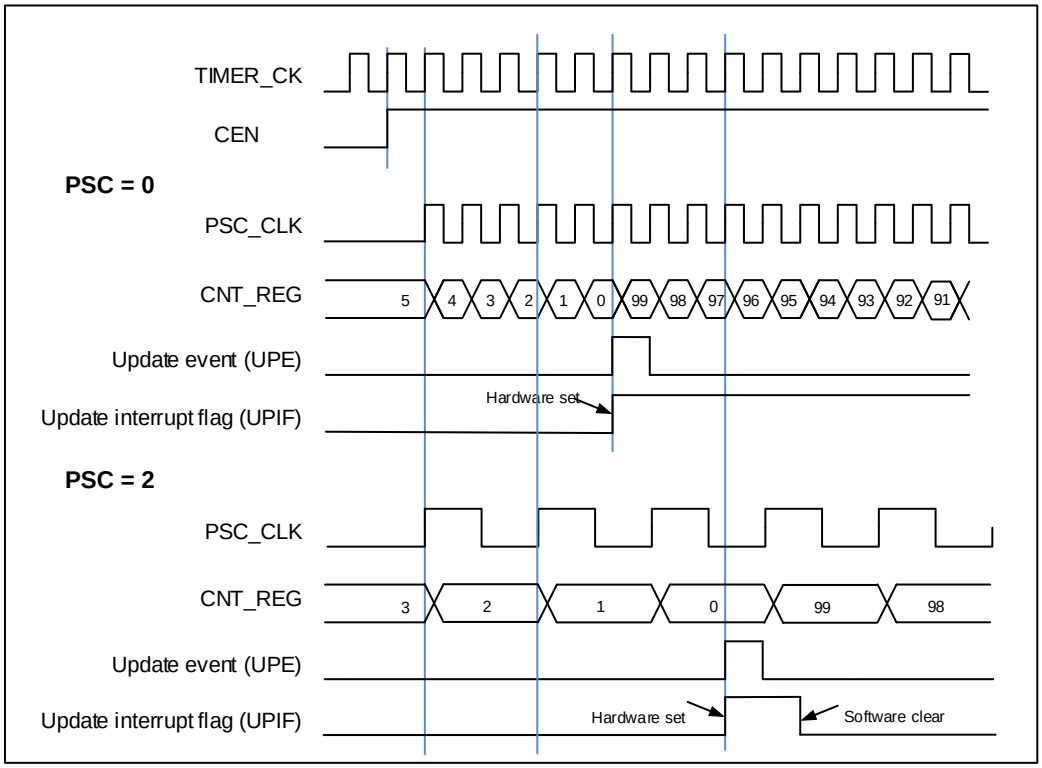
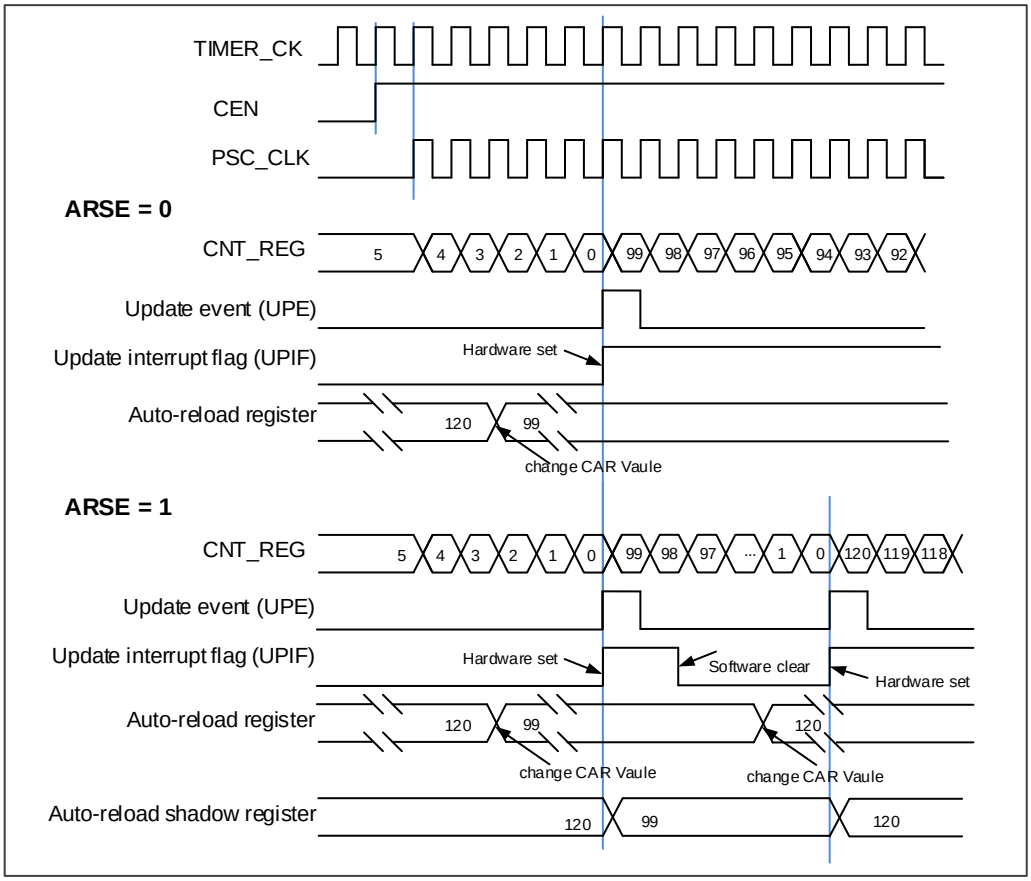


图 18-36. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值



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。

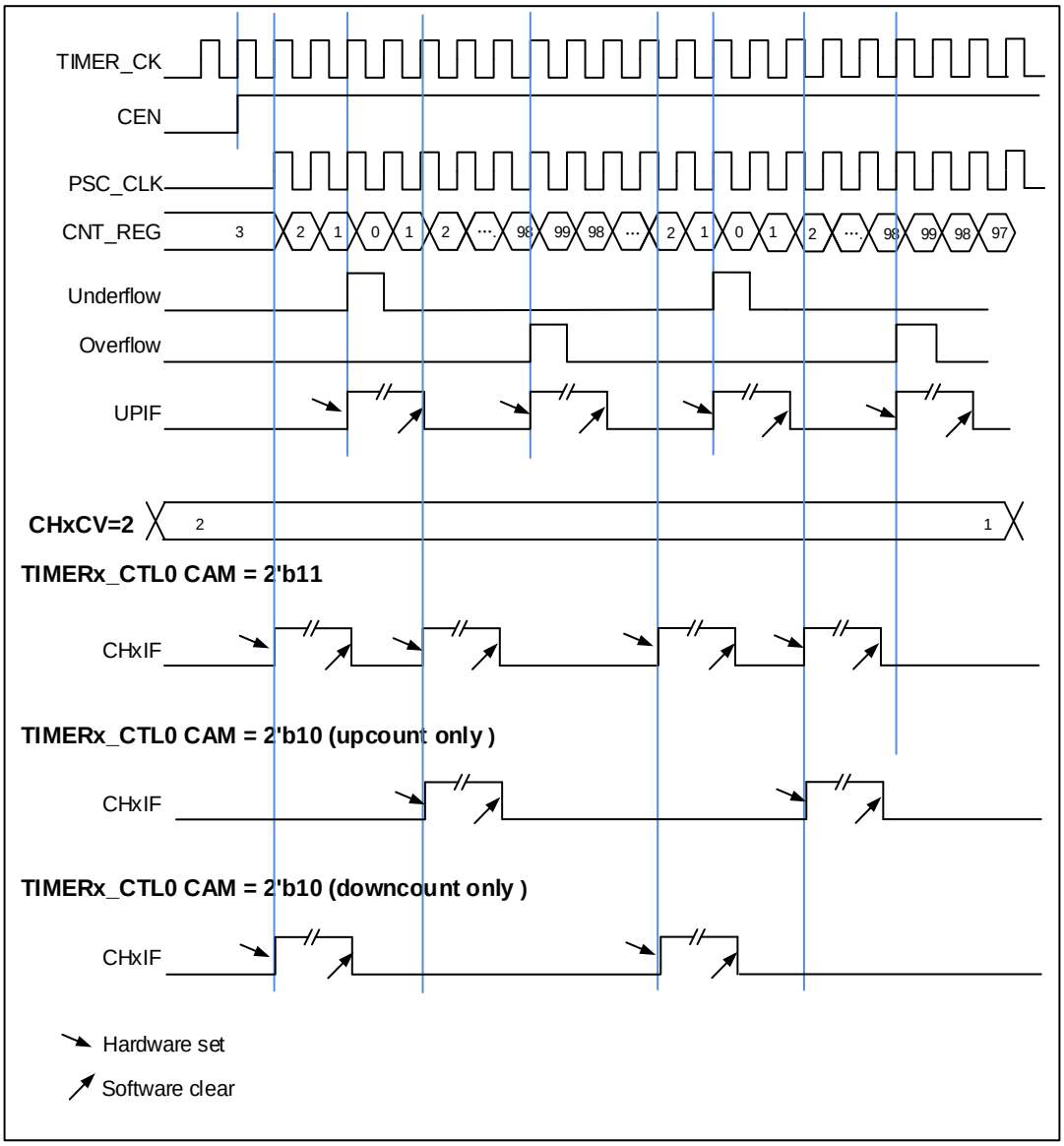
将 TIMERx_SWEVG 寄存器的 UPG 位置 1 可以初始化计数值为 0，并产生一个更新事件，而无需考虑计数器在中央模式下是向上计数还是向下计数。

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 18-37. 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。

*图 18-37. 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 18-37. 中央计数模式计数器时序图



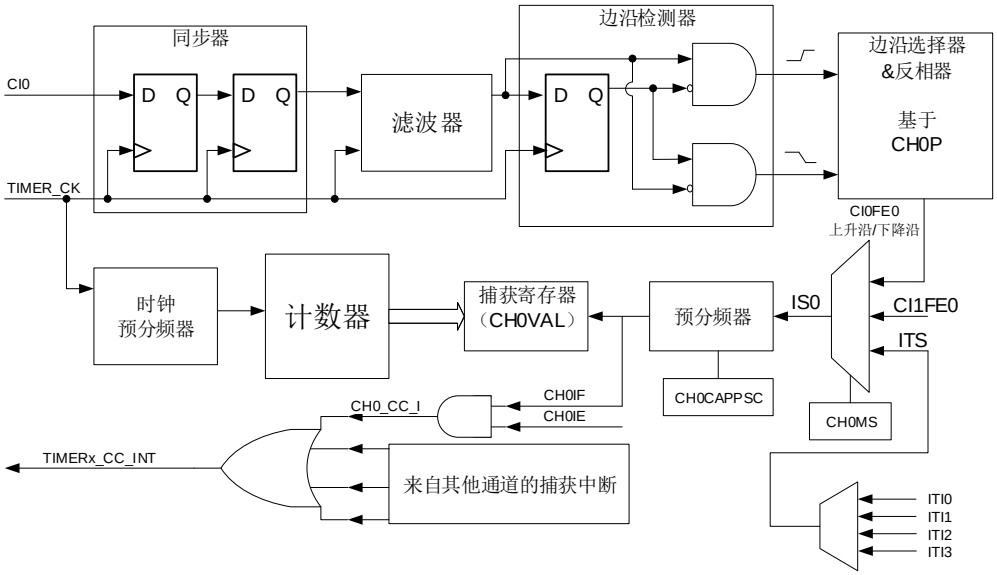
输入捕获和输出比较通道

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。

- 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 18-38. 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。

- 通道输出比较功能

图 18-39. 输出比较原理（x=0,1,2,3）

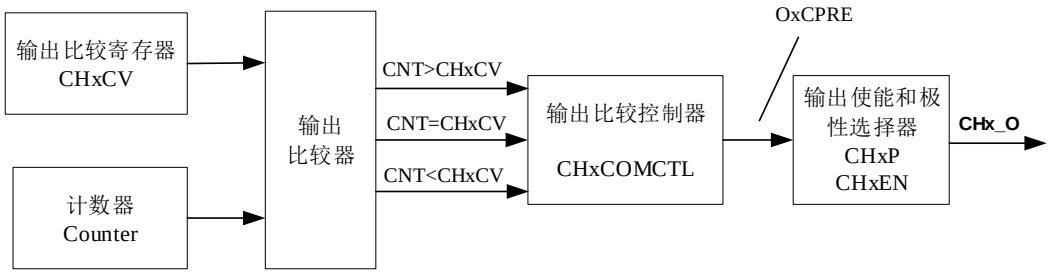


*图 18-39. 输出比较原理（x=0,1,2,3）*给出了输出比较的原理电路。通道输出信号 CHx_O 与 OxCPRE 信号（详情请见*通道输出准备信号*）的关系描述如下：OxCPRE 信号高电平有效，CHx_O 的输出情况与 OxCPRE 信号，CHxP 位和 CHxEN 位有关（具体情况请见 TIMERx_CHCTL2 寄存器中的描述）。例如，当设置 CHxP=0（CHx_O 高电平有效，与 OxCPRE 输出极性相同）、CHxEN=1（CHx_O 输出使能）时：

若 OxCPRE 输出有效（高）电平，则 CHx_O 输出有效（高）电平；

若 OxCPRE 输出无效（低）电平，则 CHx_O 输出无效（低）电平。

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。

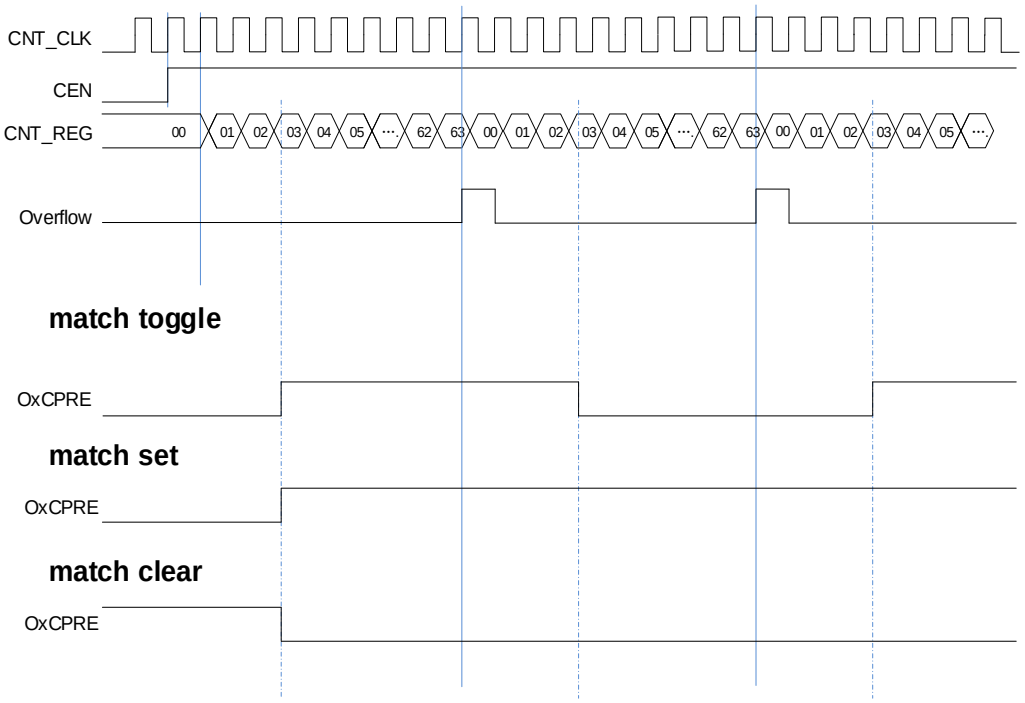
第四步：通过 TIMERx_CAR 寄存器和 TIMERx_CHxCV 寄存器配置输出比较时基：

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 18-40. 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 18-40. 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 18-41. 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 18-42. 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 18-41. EAPWM 时序图

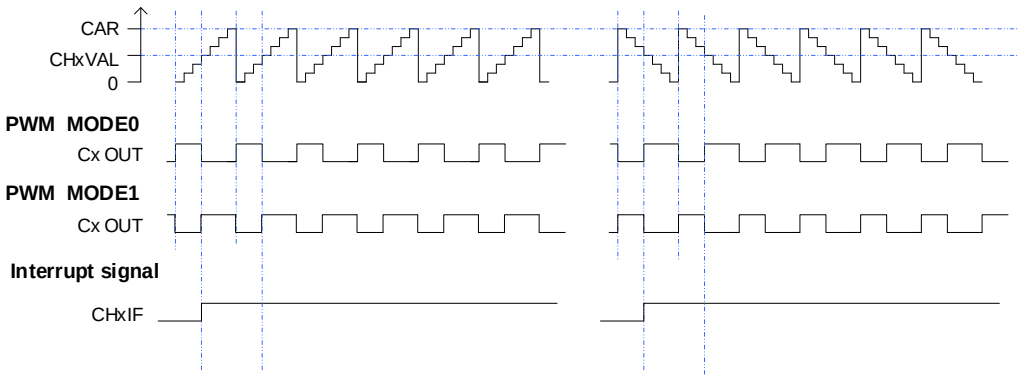
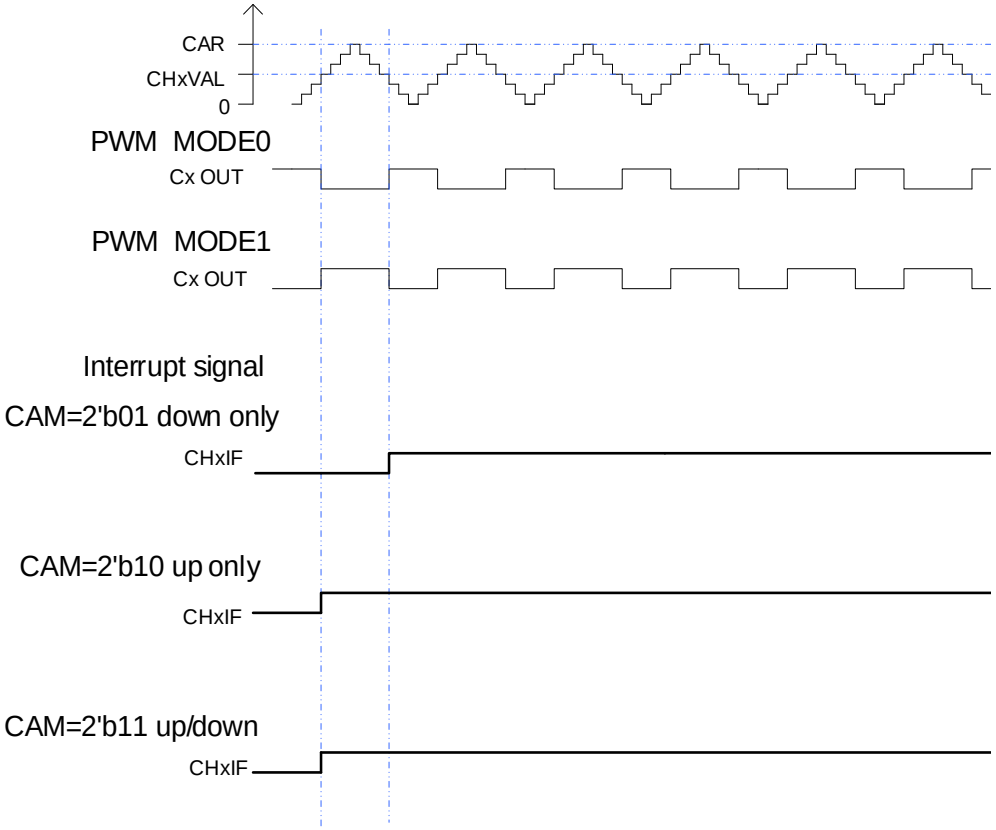


图 18-42. CAPWM 时序图



通道输出准备信号

根据*图 18-39. 输出比较原理（x=0,1,2,3）*所示，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

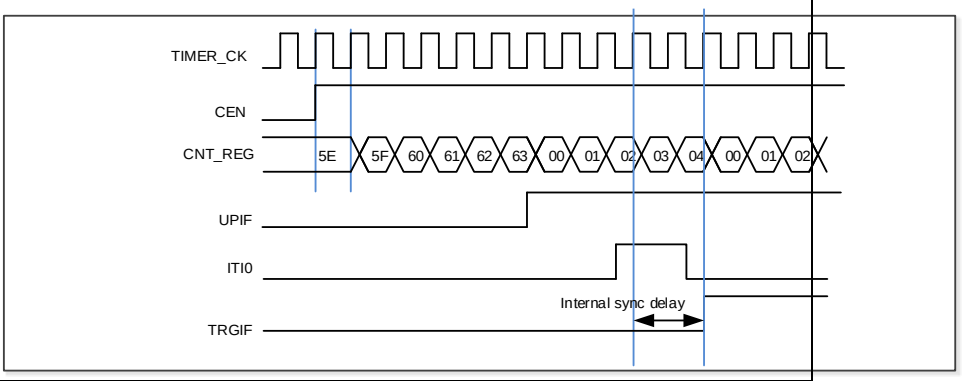
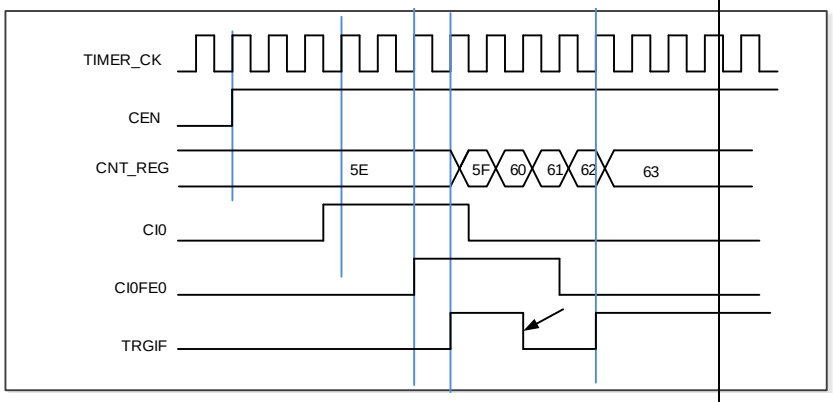
参考*霍尔传感器接口功能*。

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。

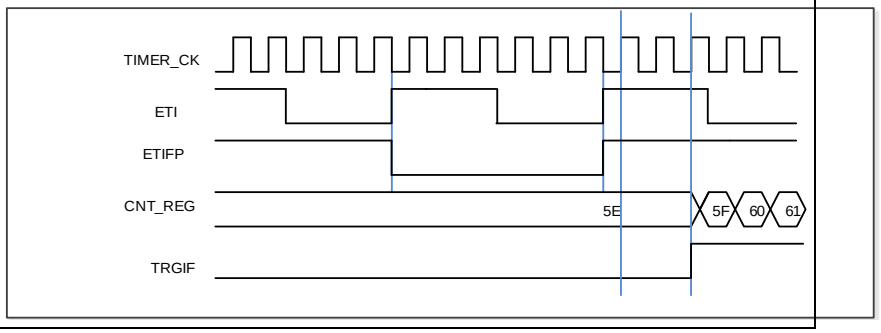
表 18-5. 从模式列表和举例

	模式选择	触发源选择	极性选择	滤波和预分频
列举	SMC[2:0] 3'b100 (复位模式) 3'b101 (暂停模式) 3'b110 (事件模式)	TRGS[2:0] 000: ITI0 001: ITI1 010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0 110: CI1FE1 111: ETIFP	如果触发源是CI0FE0或者CI1FE1，配置CHxP来选择极性和反相 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 触发源CIx，配置CHxCAPFLT设置滤波，分频不可用 触发源是ETIF，滤波和预分频不可用
例1	复位模式 当触发输入上升沿，计数器清零重启	TRGIS[2:0]=3'b000 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 18-43. 复位模式 			
例2	暂停模式 当触发输入为低的时候，计数器暂停计数	TRGIS[2:0]=3'b101 选择CI0FE0为触发源	TI0S=0.（非异或） CH0P==0不反相.在上升沿捕获	在这个例子中滤波被旁路
	图 18-44. 暂停模式 			
例3	事件模式 触发输入的上升沿计数器开始计数	TRGIS[2:0]=3'b111 选择ETIF为触发源.	ETP = 0 没有极性改变	ETPSC = 1, 2分频. 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 18-45. 事件模式



单脉冲模式

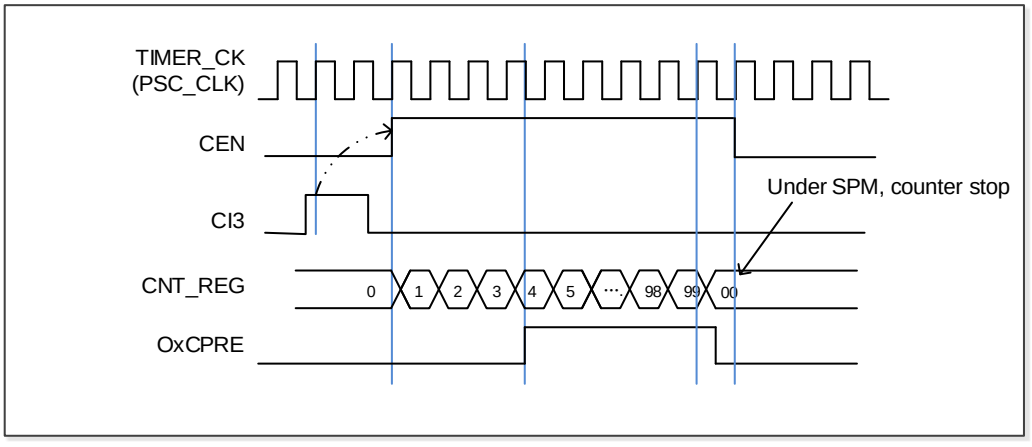
单脉冲模式与重复模式是相反的，设置 TIMERx_CTL0 寄存器的 SPM 位置 1，则使能单脉冲模式。当 SPM 置 1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置 CHxCOMCTL 配置 TIMERx 为 PWM 模式或者比较模式。

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18-46. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99*展示了一个例子。

图 18-46. 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 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®-M33 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。

18.2.5. 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

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	CAM[1:0]		DIR	SPM	UPS	UPDIS	CEN
						rw		rw	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 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。 00：$f_{DTS}=f_{CK_TIMER}$ 01：$f_{DTS}= f_{CK_TIMER} /2$ 10：$f_{DTS}= f_{CK_TIMER} /4$ 11：保留
7	ARSE	自动重载影子使能 0：禁能 TIMERx_CAR 寄存器的影子寄存器 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 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 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 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 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

当计数器使能以后，该位不能从 0x00 切换到非 0x00

4	DIR	方向 0：向上计数 1：向下计数 当计数器配置为中央对齐计数模式或正交译码器模式时，该位只读。
3	SPM	单脉冲模式 0：单脉冲模式禁能。更新事件发生后，计数器继续计数 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 软件配置该位，选择更新事件源. 0：以下事件均会产生更新中断或DMA请求： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 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. 该位用来使能或禁能更新事件的产生 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： UPG位被置1 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 0：计数器禁能 1：计数器使能 在软件将 CEN 位置 1 后，外部时钟、暂停模式和正交译码器模式才能工作。

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 0：选择 TIMERx_CH0 引脚作为通道 0 的触发输入 1：选择 TIMERx_CH0, CH1 和 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 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 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： 主定时器产生一个复位事件 TIMERx_SWEVG寄存器中UPG位置1 001：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： CEN位置1 在暂停模式下，触发输入置1 010：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 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 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：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 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 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 0：当通道捕获/比较事件发生时，发送通道 x 的 DMA 请求 . 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 ETIFP 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。 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 110: CI1FE1 111: ETIFP 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 000：关闭从模式. 如果 CEN=1，则预分频器直接由内部时钟驱动 001：正交译码器模式 0. 根据 CI1FE1 的电平，计数器在 CI0FE0 的边沿向上/下计数 010：正交译码器模式 1. 根据 CI0FE0 的电平，计数器在 CI1FE1 的边沿向上/下计数 011：正交译码器模式 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 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. 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 110：事件模式.计数器在触发输入的上升沿启动。 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

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 0：禁止触发 DMA 请求 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 0：禁止通道 3 比较/捕获 DMA 请求 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 0：禁止通道 2 比较/捕获 DMA 请求 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 0：禁止通道 1 比较/捕获 DMA 请求 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 0：禁止通道 0 比较/捕获 DMA 请求 1：使能通道 0 比较/捕获 DMA 请求
8	UPDEN	更新 DMA 请求使能 0：禁止更新 DMA 请求 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 0：禁止触发中断 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 0：禁止通道 3 中断 1：使能通道 3 中断
3	CH2IE	通道 2 比较/捕获中断使能 0：禁止通道 2 中断 1：使能通道 2 中断
2	CH1IE	通道 1 比较/捕获中断使能 0：禁止通道 1 中断 1：使能通道 1 中断

1	CH0IE	通道 0 比较/捕获中断使能 0：禁止通道 0 中断 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

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

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 参见 CH0OF 描述
11	CH2OF	通道 2 捕获溢出标志 参见 CH0OF 描述
10	CH1OF	通道 1 捕获溢出标志 参见 CH0OF 描述
9	CH0OF	通道 1 捕获溢出标志 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. 0：无捕获溢出中断发生 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 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。 0：无触发事件产生 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 参见 CH0IF 描述
2	CH1IF	通道 1 比较/捕获中断标志 参见 CH0IF 描述
1	CH0IF	通道 0 比较/捕获中断标志 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。 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 此位在任何更新事件发生时由硬件置 1，软件清 0。 0：无更新中断发生 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 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。 0：无触发事件产生 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 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 参见 CH0G 描述
1	CH0G	通道 0 捕获或比较事件发生 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 TIMERx_CH0CV 寄存器捕获，如果 CH0IF 标志位已经为 1，则 CH0OF 标志位被置 1。 0：不产生通道 0 捕获或比较事件 1：发生通道 0 捕获或比较事件
0	UPG	更新事件产生 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
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 参见 CH0COMCEN 描述
14:12	CH1COMCTL[2:0]	通道 1 输出比较模式 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

参见 CH0COMFEN 描述

9:8	CH1MS[1:0]	通道 1 模式选择 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。 00：通道 1 配置为输出 01：通道 1 配置为输入，IS1 映射在 CI1FE1 上 10：通道 1 配置为输入，IS1 映射在 CI0FE1 上 11：通道 1 配置为输入，IS1 映射在 ITS 上 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 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 0：禁止通道 0 输出比较清零 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 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。 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 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。 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。 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 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。 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
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 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 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

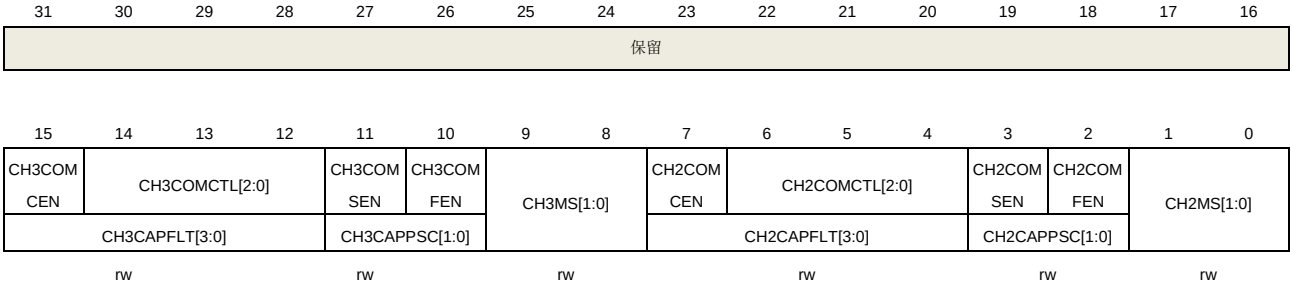
与输出比较模式相同

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



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 参见 CH0COMCEN 描述
14:12	CH3COMCTL[2:0]	通道 3 输出比较模式 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 参见 CH0COMFEN 描述
9:8	CH3MS[1:0]	通道 3 模式选择 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。 00：通道 3 配置为输出 01：通道 3 配置为输入，IS3 映射在 CI3FE3 上 10：通道 3 配置为输入，IS3 映射在 CI2FE3 上 11：通道 3 配置为输入，IS3 映射在 ITS 上 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 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 0：使能通道 2 输出比较清零 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 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。 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 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。 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。 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。 100：强制为低。强制 O2CPRE 为低电平 101：强制为高。强制 O2CPRE 为高电平 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。 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。 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。 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 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。

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
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 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 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
CH3NP	保留	CH3P	CH3EN	CH2NP	保留	CH2P	CH2EN	CH1NP	保留	CH1P	CH1EN	CH0NP	保留	CH0P	CH0EN
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	CH3NP	通道 3 互补输出极性 参考 CH0NP 描述
14	保留	必须保持复位值
13	CH3P	通道 3 极性 参考 CH0P 描述
12	CH3EN	通道 3 使能

		参考 CH0EN 描述
11	CH2NP	通道 2 互补输出极性 参考 CH0NP 描述
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 参考 CH0EN 描述
7	CH1NP	通道 1 互补输出极性 参考 CH0NP 描述
6	保留	必须保持复位值
5	CH1P	通道 1 极性 参考 CH0P 描述
4	CH1EN	通道 1 使能 参考 CH0EN 描述
3	CH0NP	通道 0 互补输出极性 当通道 0 配置为输出模式，该位必须保持复位值 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 当通道 0 配置为输出模式时，此位定义了输出信号极性。 0：通道0高电平为有效电平 1：通道0低电平为有效电平 当通道 0 配置为输入模式时，此位定义了 CI0 信号极性 [CH0NP, CH0P] 将选择 CI0FE0 或者 CI1FE0 的有效边沿或者捕获极性 [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。 [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。 [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 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。 0：禁止通道 0 1：使能通道 0

计数器寄存器 (TIMERx_CNT)(x=1)

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
CNT[31:16]															
rw															

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
31:0	CNT[31:0]	这些位是当前的计数值。写操作能改变计数器值。

计数器寄存器 (TIMERx_CNT)(x=2,3,4)

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)(x=1)

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
CARL[31:16]															
rw															

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
31:0	CARL[31:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

计数器自动重载寄存器 (TIMERx_CAR)(x=2,3,4)

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 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)(x=1)

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
CH0VAL[31:16]															
rw															

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
31:0	CH0VAL[31:0]	通道 0 的捕获或比较值 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。 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 0 捕获/比较值寄存器 (TIMERx_CH0CV)(x=2,3,4)

地址偏移：0x34

该寄存器只能按字（32 位）访问

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 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。 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)(x=1)

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
CH1VAL[31:16]															
rw															
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
31:0	CH1VAL[31:0]	通道 1 的捕获或比较值 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。 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 1 捕获/比较值寄存器 (TIMERx_CH1CV)(x=2,3,4)

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 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。 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)(x=1)

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
CH2VAL[31:16]															
rw															

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
31:0	CH2VAL[31:0]	通道 2 的捕获或比较值 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。 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 2 捕获/比较值寄存器 (TIMERx_CH2CV)(x=2,3,4)

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 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。 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)(x=1)

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
CH3VAL[31:16]															
rw															

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
31:0	CH3VAL[31:0]	通道 3 的捕获或比较值 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。 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。

通道 3 捕获/比较值寄存器 (TIMERx_CH3CV)(x=2,3,4)

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 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。 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

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			DMATC[4:0]					保留			DMATA[4:0]				
			rw								rw				

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 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 该位域定义了 DMA 访问 TIMERx_DMATB 寄存器的第一个地址。当通过 TIMERx_DMA 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)(x=1)

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
DMATB[31:16]															

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

rw

位/位域	名称	描述
31:0	DMATB [31:0]	DMA 发送缓冲 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。

DMA 发送缓冲区寄存器 (TIMERx_DMATB)(x=2,3,4)

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
保留															

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

rw

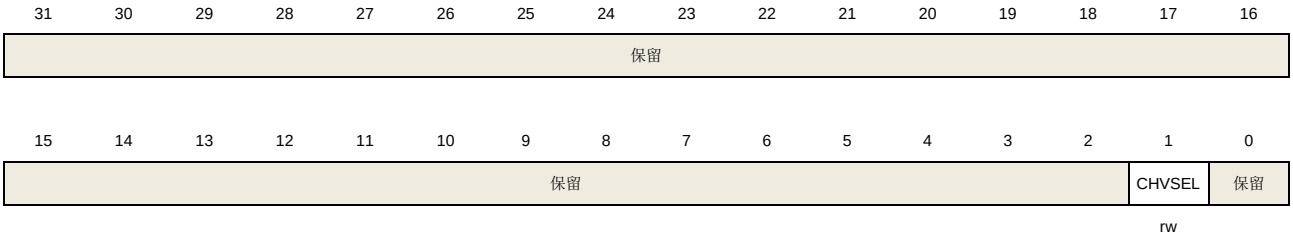
位/位域	名称	描述
31:16	保留	必须保持复位值.
15:0	DMATB [15:0]	DMA 发送缓冲 对这个寄存器的读或写，（起始地址+传输次数*4）地址范围内的寄存器会被访问 传输次数由硬件计算，范围为 0 到 DMATC。

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 此位由软件写 1 或清 0。 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 0：无影响
0	保留	必须保持复位值

18.3. 通用定时器 L1（TIMERx, x=8,11）

18.3.1. 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。

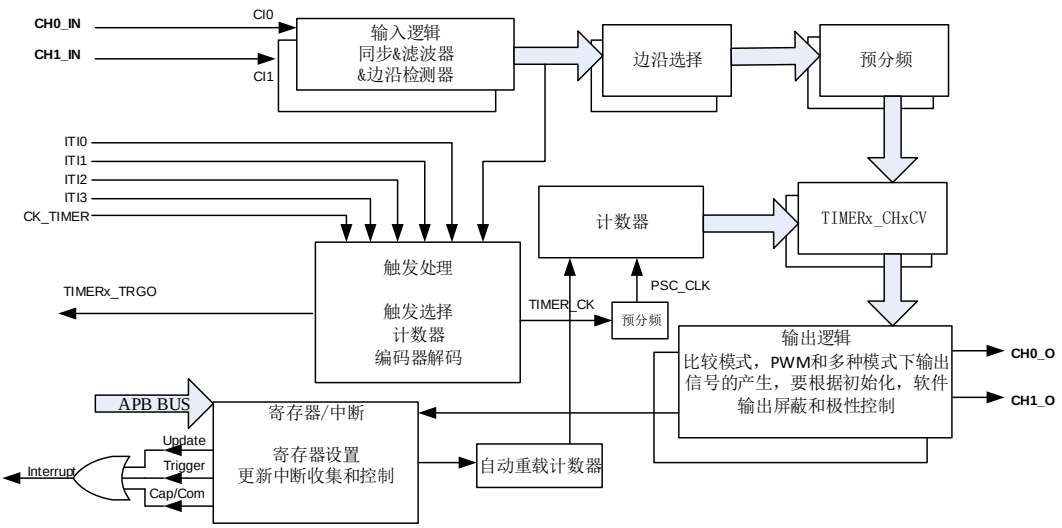
18.3.2. 主要特性

- 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。

18.3.3. 结构框图

*图 18-47. 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 18-47. 通用定时器 L1 结构框图



18.3.4. 功能描述

时钟源配置

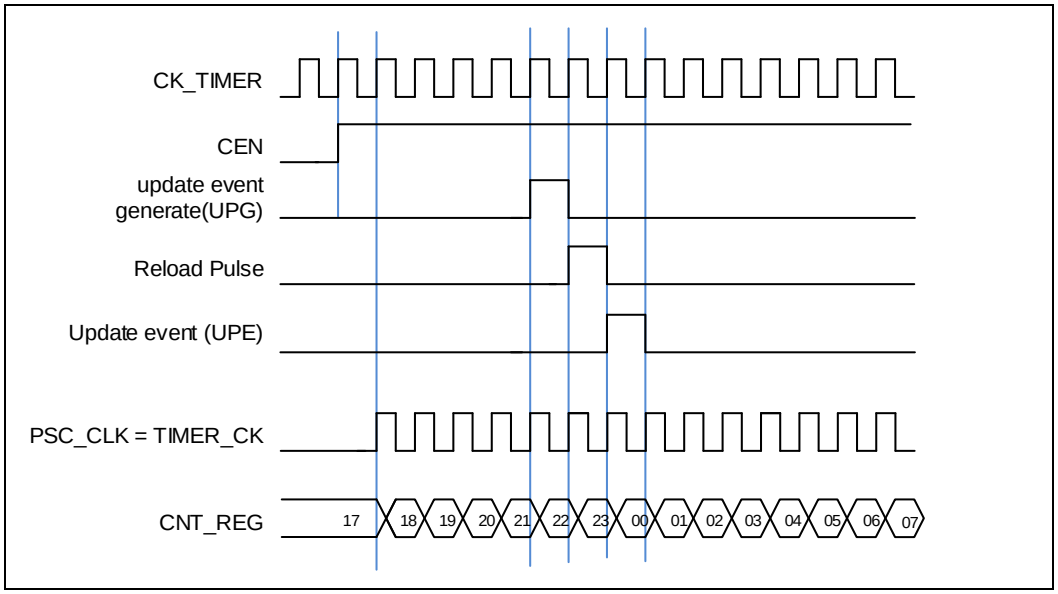
通用定时器 L1 可以由内部时钟源 CK_TIMER 或者由 SMC（TIMERx_SMCFG 寄存器位[2:0]）控制的复用时钟源驱动。

- 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 18-48. 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。

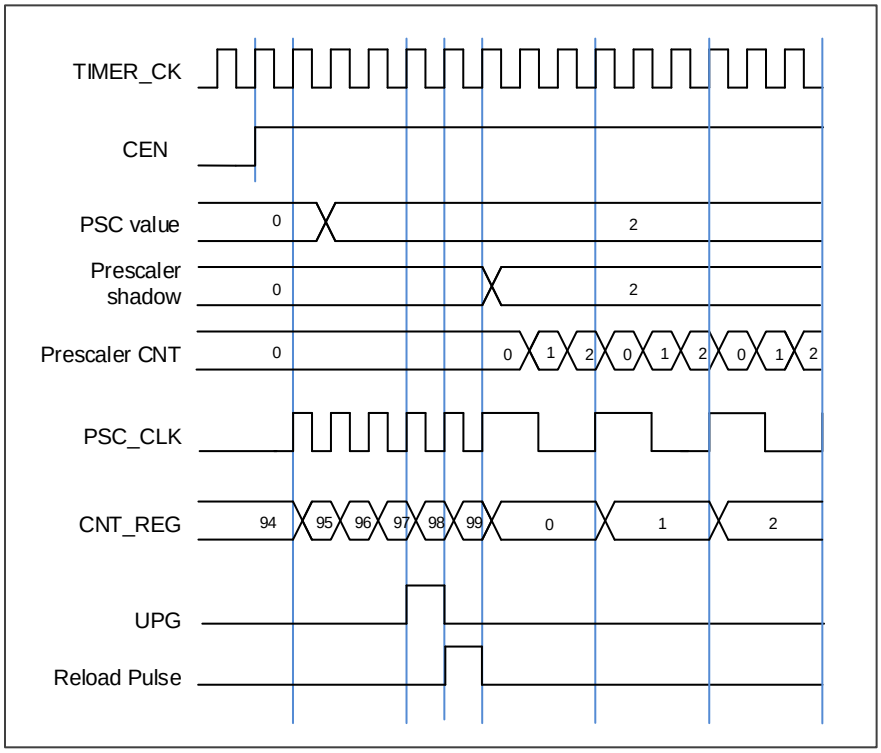
计数器预分频器也可以在内部触发信号 ITI0/1/2/3 的上升沿计数。这种模式可以通过设置 SMC [2:0]为 0x7 同时设置 TRGS [2:0]为 0x0, 0x1, 0x2 或者 0x3。

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 18-49. 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 18-50. 向上计数时序图，PSC=0/2 和 *图 18-51. 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 18-50. 向上计数时序图，PSC=0/2

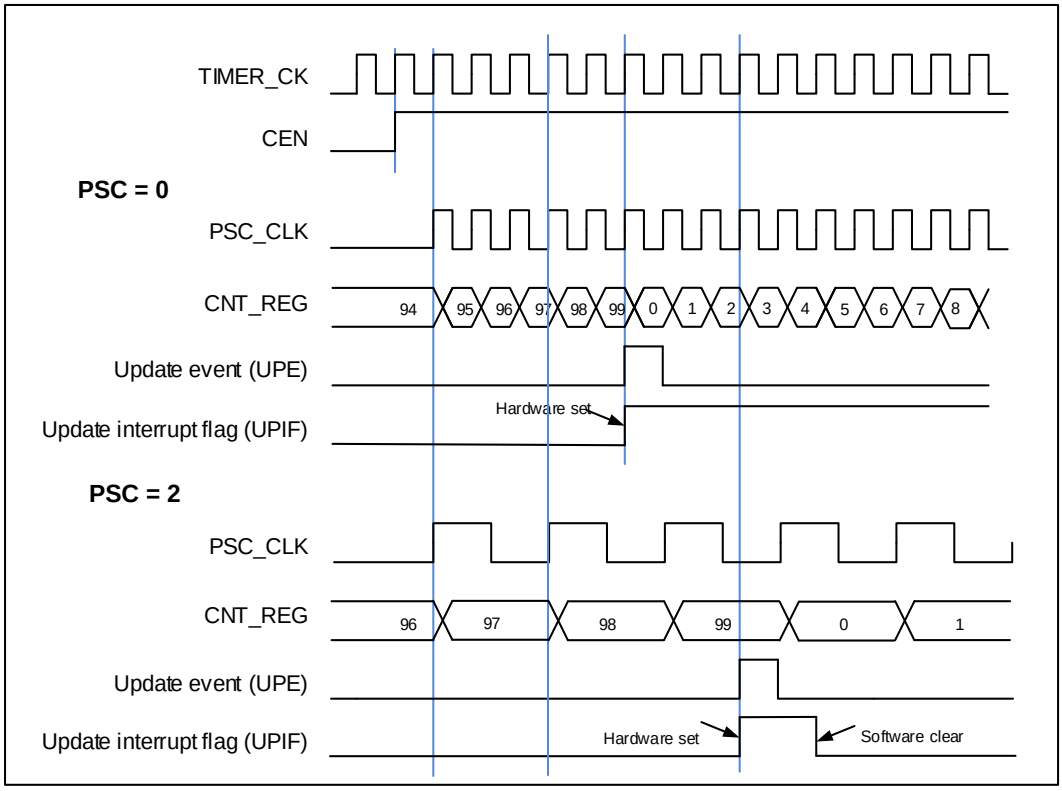
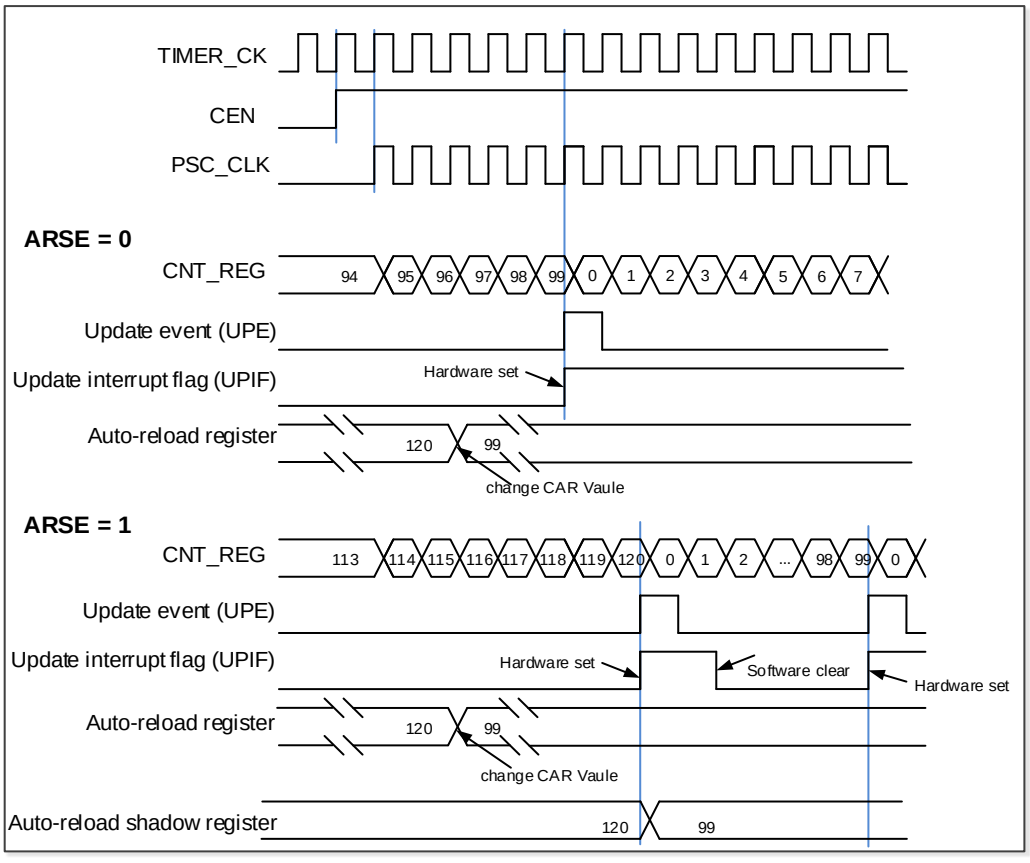


图 18-51. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



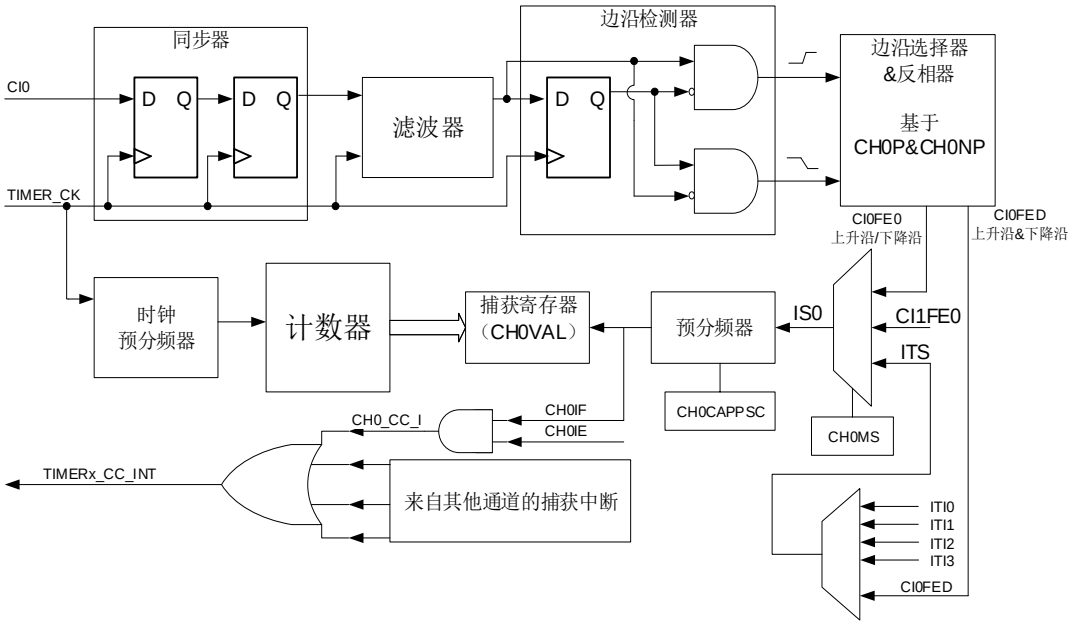
捕获/比较通道

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。

■ 输入捕获模式

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 18-52. 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。

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。

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。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**

图 18-53. 通道输出比较原理（x=0，1）

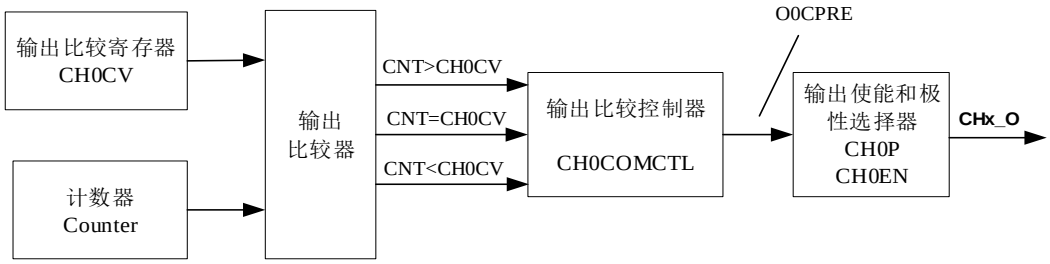


*图 18-53. 通道输出比较原理（x=0，1）*给出了输出比较的逻辑电路。通道输出信号 CHx_O 与 OxCPRE 信号（详情请见*通道输出准备信号*）的关系描述：OxCPRE 信号高电平有效，CHx_O 的输出情况与 OxCPRE 信号，CHxP 位和 CHxE 位有关（具体情况请见 TIMERx_CHCTL2 寄存器中的描述）。例如，当设置 CHxP=0（CHx_O 高电平有效，与 OxCPRE 输出极性相同）、CHxEN=1（CHx_O 输出使能）时：

若 OxCPRE 输出有效（高）电平，则 CHx_O 输出有效（高）电平；

若 OxCPRE 输出无效（低）电平，则 CHx_O 输出无效（低）电平。

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 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。

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。

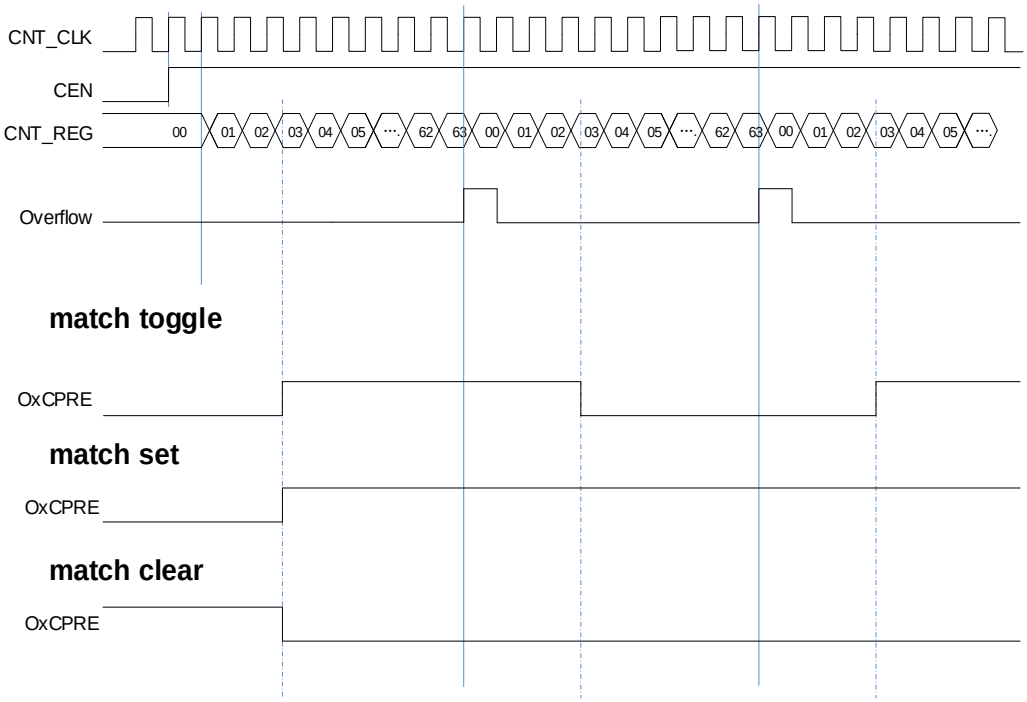
第四步：通过 TIMERx_CAR 寄存器和 TIMERx_CHxCV 寄存器配置输出比较时基：

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 18-54. 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 18-54. 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。

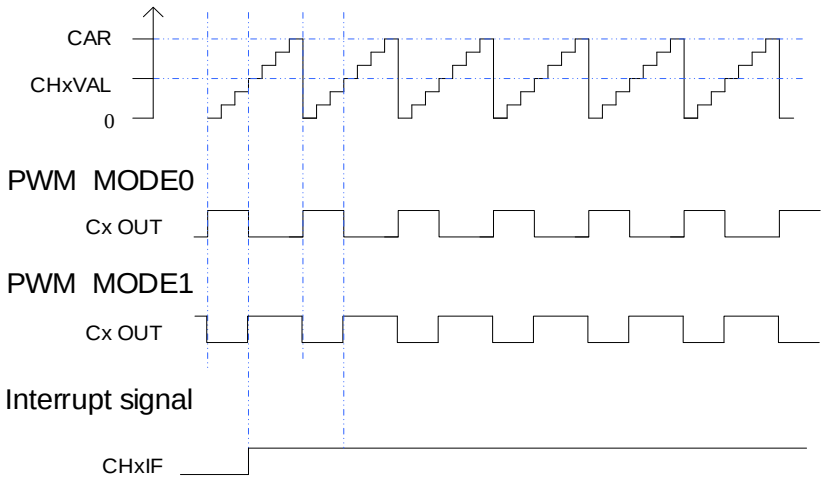
EAPWM 的周期由 TIMERx_CAR 寄存器值决定，占空比由 TIMERx_CHxCV 寄存器值决定。*图 18-55. EAPWM 时序图*显示了 EAPWM 的输出波形和中断。

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 18-55. EAPWM 时序图



通道输出准备信号

根据*图 18-53. 通道输出比较原理（x=0，1）*所示，当 TIMERx 用于输出匹配比较模式下，在通道输出信号之前会产生一个中间信号 OxCPRE 信号(通道 x 输出准备信号)。设置 CHxCOMCTL 位可以定义 OxCPRE 信号类型。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 输出准备信号)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 位域为 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。

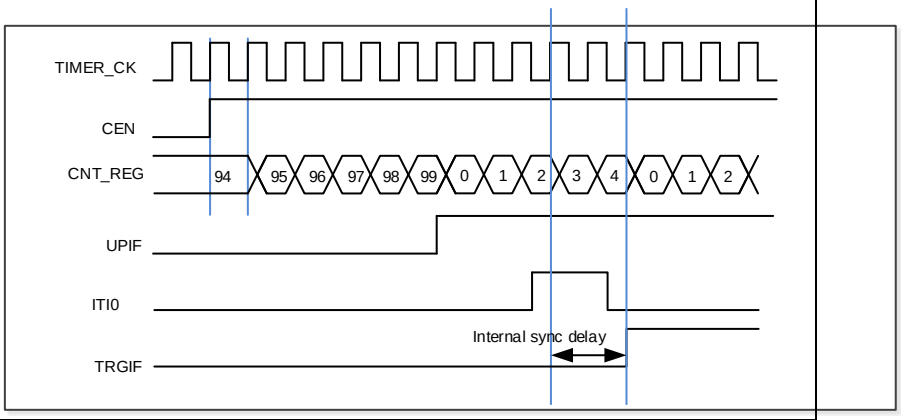
主-从管理

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 18-6. 从模式列表和举例

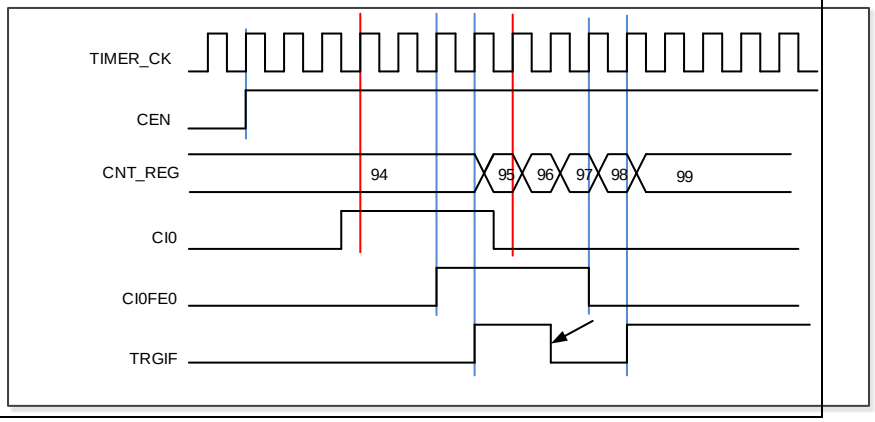
	模式选择	触发源选择	极性选择	滤波和预分频
列举	SMC[2:0] 3'b100 (复位模式) 3'b101 (暂停模式) 3'b110 (事件模式)	TRGS[2:0] 000: ITI0 001: ITI1 010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0 110: CI1FE1 111: 保留	如果触发源是CI0FE0或者CI1FE1，配置CHxP和CHxNP来选择极性和反相	触发源ITIx，滤波和预分频不可用 触发源CIx，配置CHxCAPFLT设置滤波，分频不可用
例1	复位模式 当触发输入上升沿，计数器清零重启	TRGIS[2:0]=3'b000 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 18-56. 复位模式下的控制电路



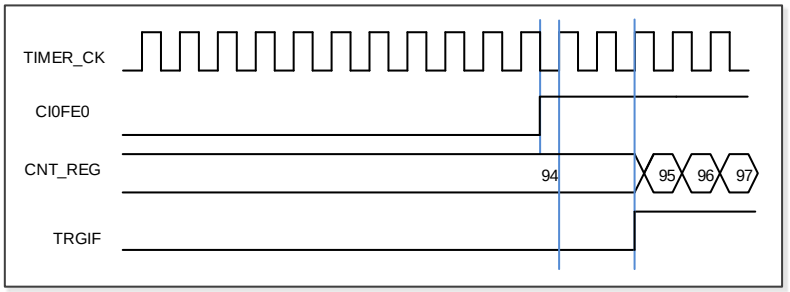
例2	暂停模式 当触发输入为低的时候，计数器暂停计数	TRGIS[2:0]=3'b101 选择CI0FE0为触发源	CH0P==0，不反相.在上升沿捕获	在这个例子中滤波被旁路

图 18-57. 暂停模式下的控制电路



	模式选择	触发源选择	极性选择	滤波和预分频
例3	事件模式 触发输入的上升沿计数器开始计数	TRGIS[2:0]=3'b101 选择CI0FE0为触发源	CH0P==0，不反相.在上升沿捕获	在这个例子中滤波被旁路

图 18-58. 事件模式下的控制电路



单脉冲模式

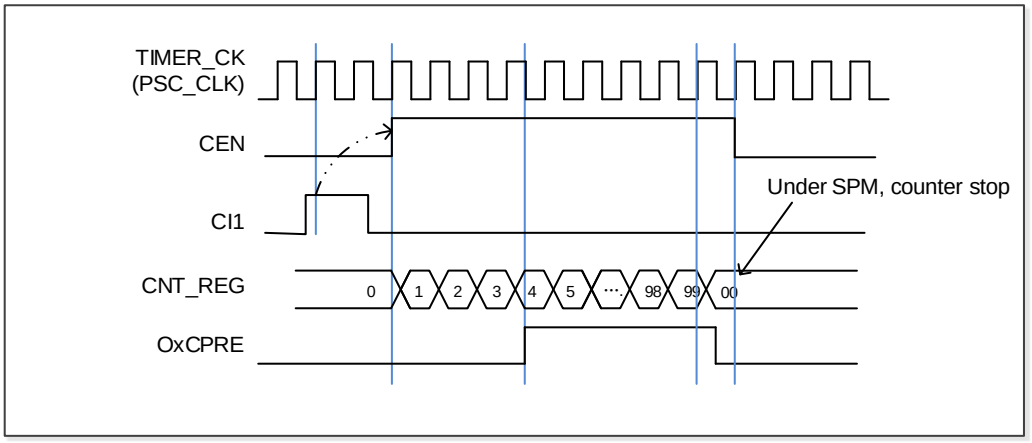
单脉冲模式与重复模式是相反的，设置 TIMERx_CTL0 寄存器的 SPM 位置 1，则使能单脉冲模式。当 SPM 置 1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置 CHxCOMCTL 配置 TIMERx 为 PWM 模式或者比较模式。

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18-59. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99*展示了一个例子。

图 18-59. 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®-M33 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

18.3.5. TIMERx 寄存器(x=8,11)

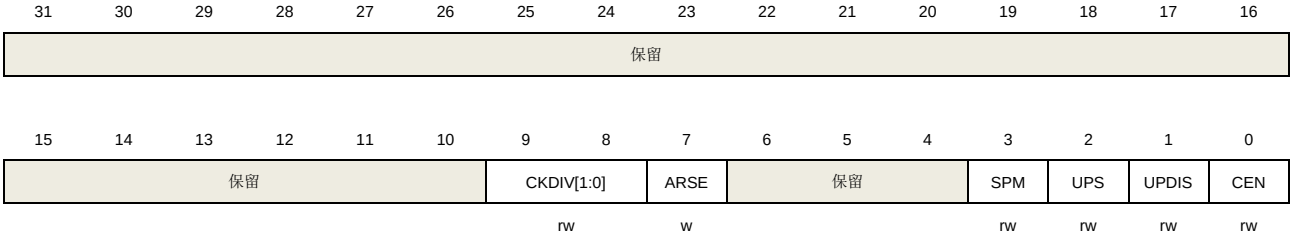
TIMER8 基地址：0x4001 4C00

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



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 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。 00：$f_{DTS}=f_{CK_TIMER}$ 01：$f_{DTS}= f_{CK_TIMER}$ /2 10：$f_{DTS}= f_{CK_TIMER}$ /4 11：保留
7	ARSE	自动重载影子使能 0: 禁能 TIMERx_CAR 寄存器的影子寄存器 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 0：单脉冲模式禁能。更新事件发生后，计数器继续计数 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 软件配置该位，选择更新事件源. 0：以下事件均会产生更新中断或DMA请求： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 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. 该位用来使能或禁能更新事件的产生 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： UPG位被置1 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 0: 计数器禁能 1: 计数器使能 在软件将 CEN 位置 1 后，外部时钟、暂停模式和正交译码器模式才能工作。

从模式配置寄存器 (TIMERx_SMCFG)

地址偏移：0x08

复位值：0x0000 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								MSM	TRGS[2:0]			保留	SMC[2:0]		
								rw	rw			rw	rw		

位/位域	名称	描述
31:8	保留	必须保持复位值
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 110: CI1FE1 111: 保留 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 000: 关闭从模式. 如果 CEN=1，则预分频器直接由内部时钟驱动 001: 保留 010: 保留 011: 保留 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. 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 110: 事件模式.计数器在触发输入的上升沿启动。 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 0：禁止触发中断 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 0：禁止通道 1 中断 1：使能通道 1 中断
1	CH0IE	通道 0 比较/捕获中断使能 0：禁止通道 0 中断

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

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					CH1OF	CH0OF	保留		TRGIF	保留			CH1IF	CH0IF	UPIF
					rc_w0	rc_w0			rc_w0				rc_w0	rc_w0	rc_w0

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 参见 CH0OF 描述
9	CH0OF	通道 1 捕获溢出标志 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. 0: 无捕获溢出中断发生 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 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。 0：无触发事件产生 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 参见 CH0IF 描述
1	CH0IF	通道 0 比较/捕获中断标志 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。

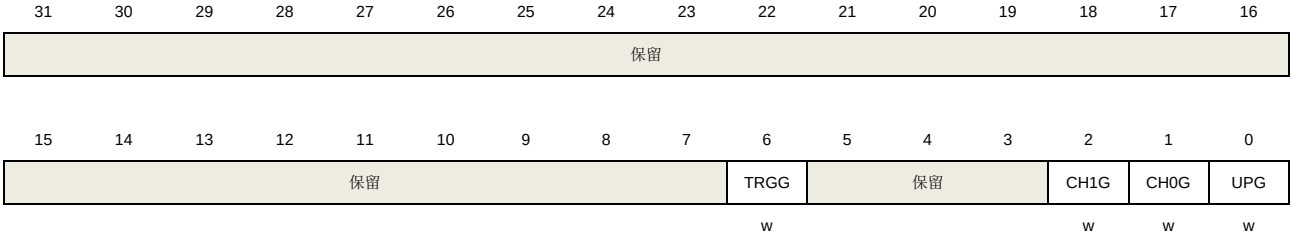
0：无通道 0 中断发生
1：通道 0 中断发生

0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置 1，软件清 0。 0：无更新中断发生 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 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。 0：无触发事件产生 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 参见 CH0G 描述
1	CH0G	通道 0 捕获或比较事件发生 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 TIMERx_CH0CV 寄存器捕获，如果 CH0IF 标志位已经为 1，则 CH0OF 标志位被置 1。 0：不产生通道 0 捕获或比较事件 1：发生通道 0 捕获或比较事件
0	UPG	更新事件产生 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 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 参见 CH0COMFEN 描述
9:8	CH1MS[1:0]	通道 1 模式选择 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。 00：通道 1 配置为输出 01：通道 1 配置为输入，IS1 映射在 CI1FE1 上 10：通道 1 配置为输入，IS1 映射在 CI0FE1 上 11：通道 1 配置为输入，IS1 映射在 ITS 上 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 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。 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

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。
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。
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 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。 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 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 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
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

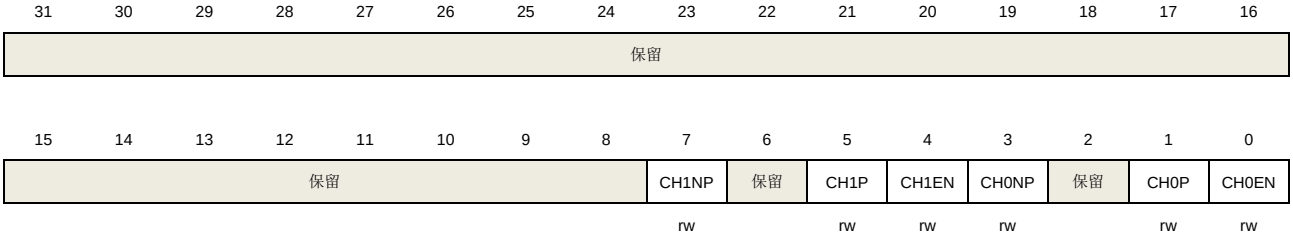
1:0 CH0MS[1:0] 通道 0 模式选择
与输出比较模式相同

通道控制寄存器 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



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 参考 CH0NP 描述
6	保留	必须保持复位值
5	CH1P	通道 1 极性 参考 CH0P 描述
4	CH1EN	通道 1 使能 参考 CH0EN 描述
3	CH0NP	通道 0 互补输出极性 当通道 0 配置为输出模式，该位必须保持复位值 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 当通道 0 配置为输出模式时，此位定义了输出信号极性。 0：通道0高电平为有效电平 1：通道0低电平为有效电平 当通道 0 配置为输入模式时，此位定义了 CI0 信号极性 [CH0NP, CH0P] 将选择 CI0FE0 或者 CI1FE0 的有效边沿或者捕获极性 [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。 [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。 [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 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。 0：禁止通道 0 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]															

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															

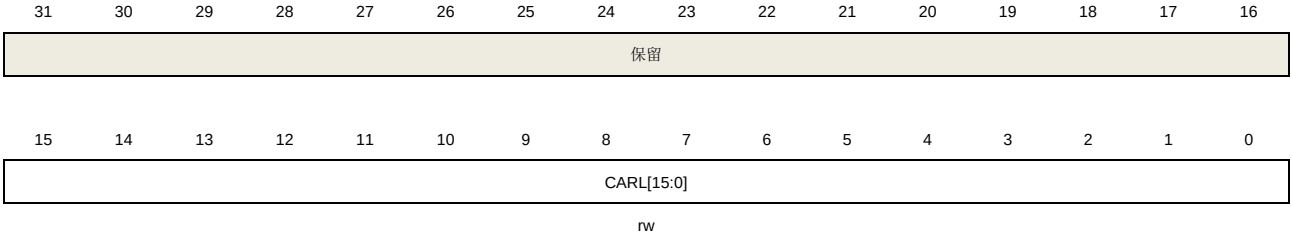
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 TIMER_CK 时钟除以(PSC+1)，每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。

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



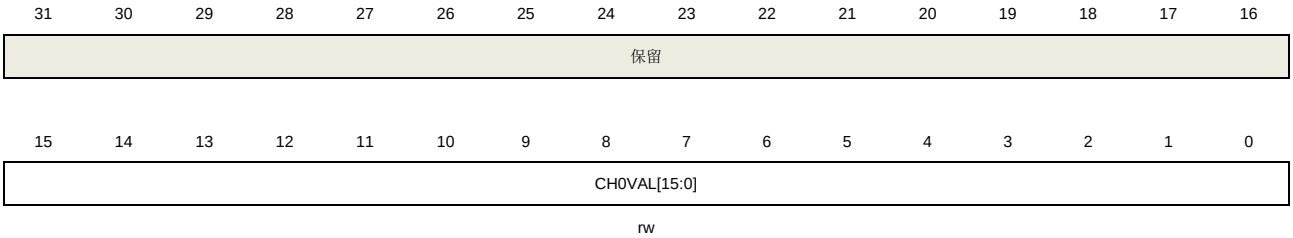
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

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



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 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。 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 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。 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 此位由软件写 1 或清 0。 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 0：无影响
0	保留	必须保持复位值

18.4. 通用定时器 L2（TIMERx, x=9,10,12,13）

18.4.1. 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，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，可以被用来计数，其外部事件可以驱动其他定时器

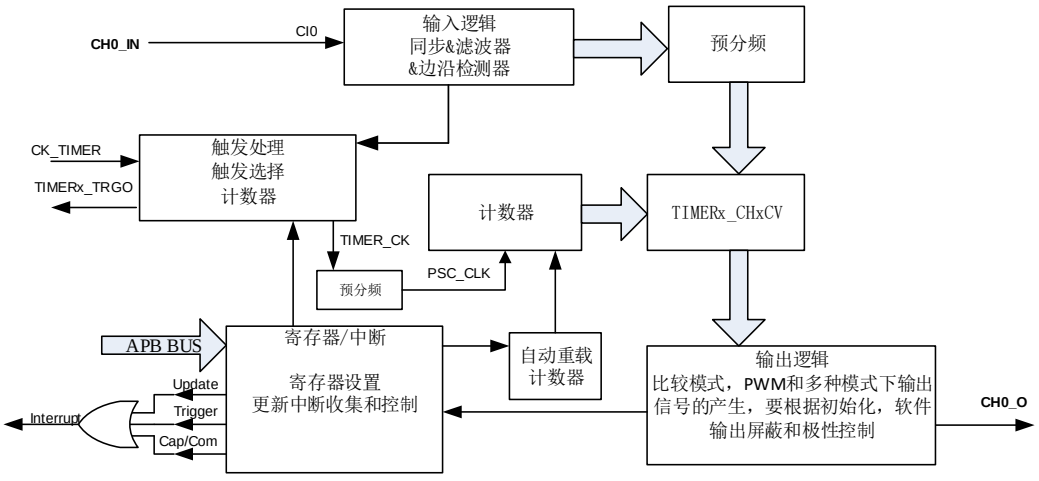
18.4.2. 主要特性

- 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，向下计数和中央计数
- 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，比较/捕获事件

18.4.3. 结构框图

*图 18-60. 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 18-60. 通用定时器 L2 结构框图



18.4.4. 功能描述

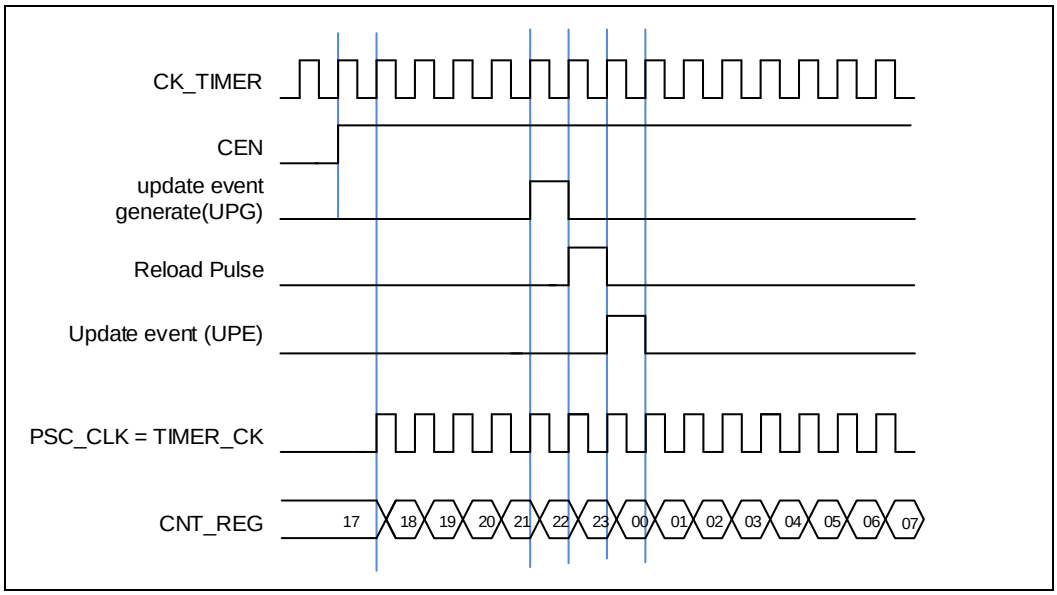
时钟源配置

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。

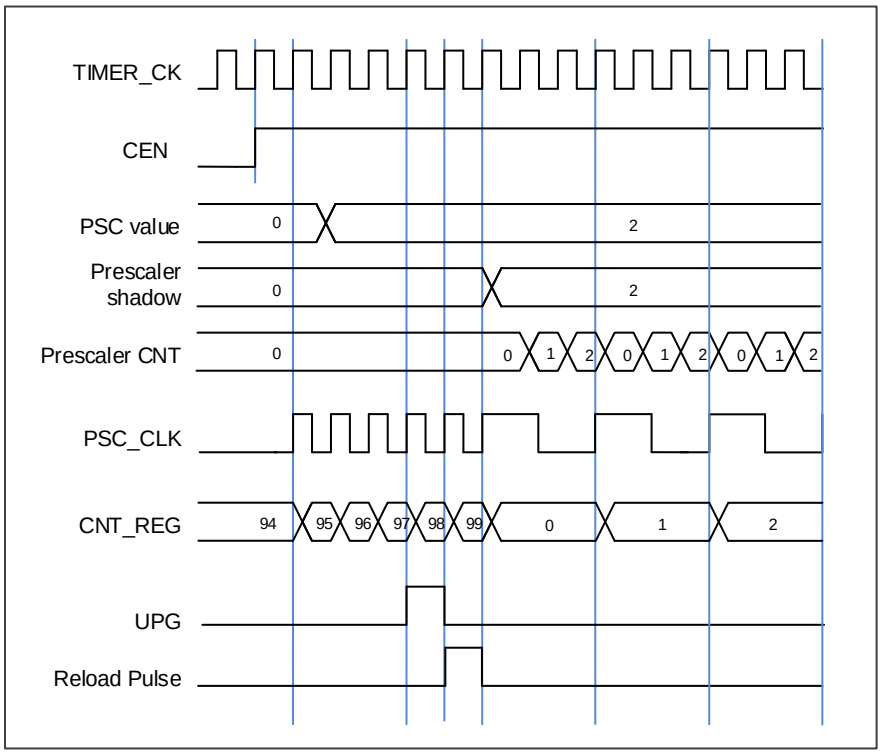
图 18-61. 内部时钟分频为 1 时，计数器的时序图



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 18-62. 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 18-63. 向上计数时序图，PSC=0/2 和 *图 18-64. 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 18-63. 向上计数时序图，PSC=0/2

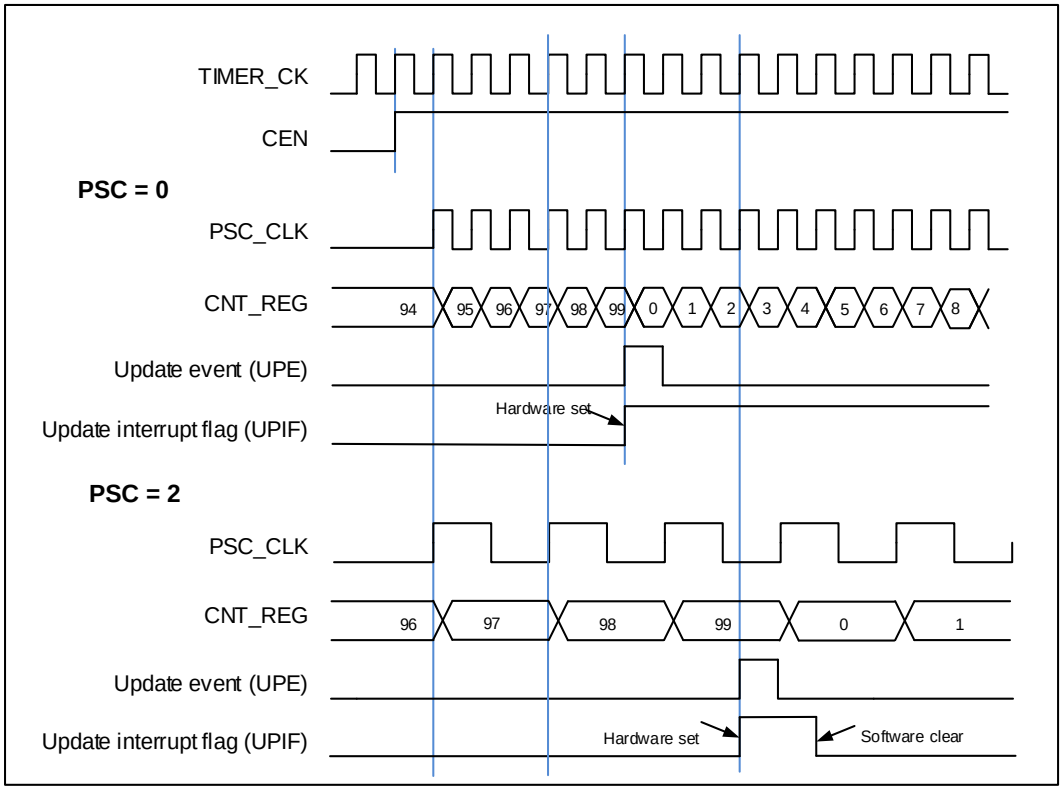
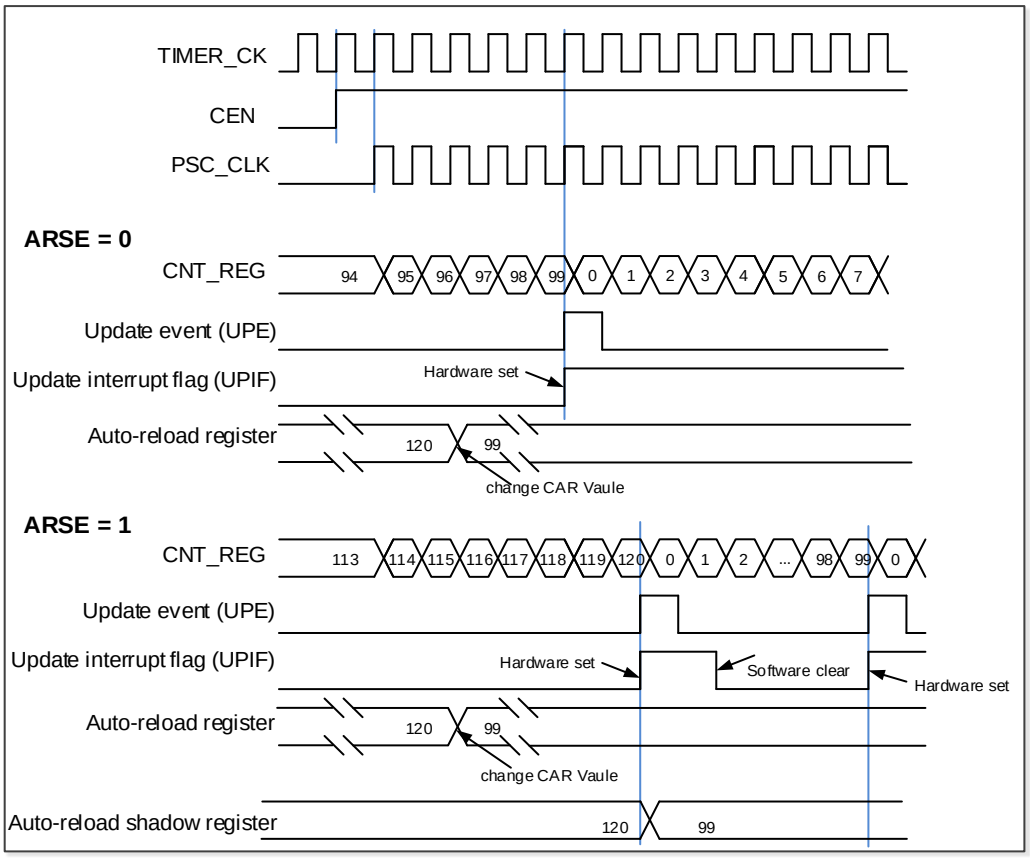


图 18-64. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



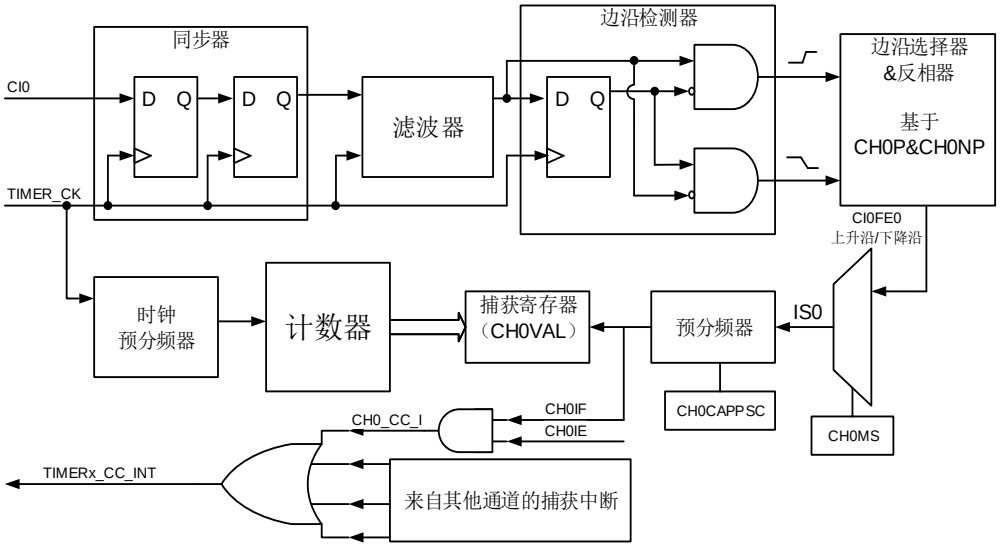
输入捕获和输出比较通道

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

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 18-65. 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。

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**

图 18-66. 通道输出比较原理

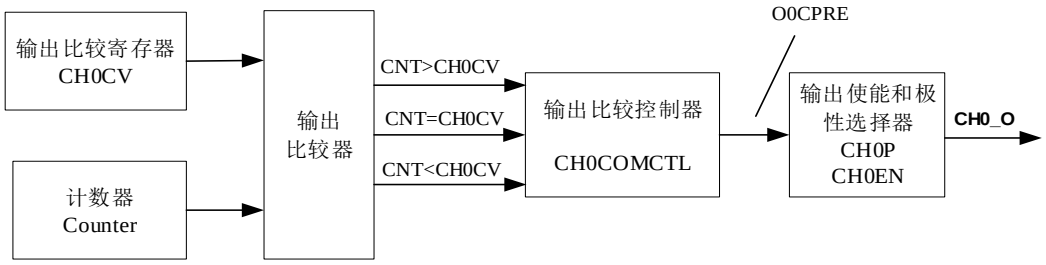


*图 18-66. 通道输出比较原理*给出了输出比较的逻辑电路。通道输出信号 CHx_O 与 OxCPRE 信号（详情请见*通道输出准备信号*）的关系描述：OxCPRE 信号高电平有效，CHx_O 的输出情况与 OxCPRE 信号，CHxP 位和 CHxEN 位有关（具体情况请见 TIMERx_CHCTL2 寄存器中的描述）。例如，当设置 CHxP=0（CHx_O 高电平有效，与 OxCPRE 输出极性相同）、CHxEN=1（CHx_O 输出使能）时：

若 OxCPRE 输出有效（高）电平，则 CHx_O 输出有效（高）电平；

若 OxCPRE 输出无效（低）电平，则 CHx_O 输出无效（低）电平。

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。

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。

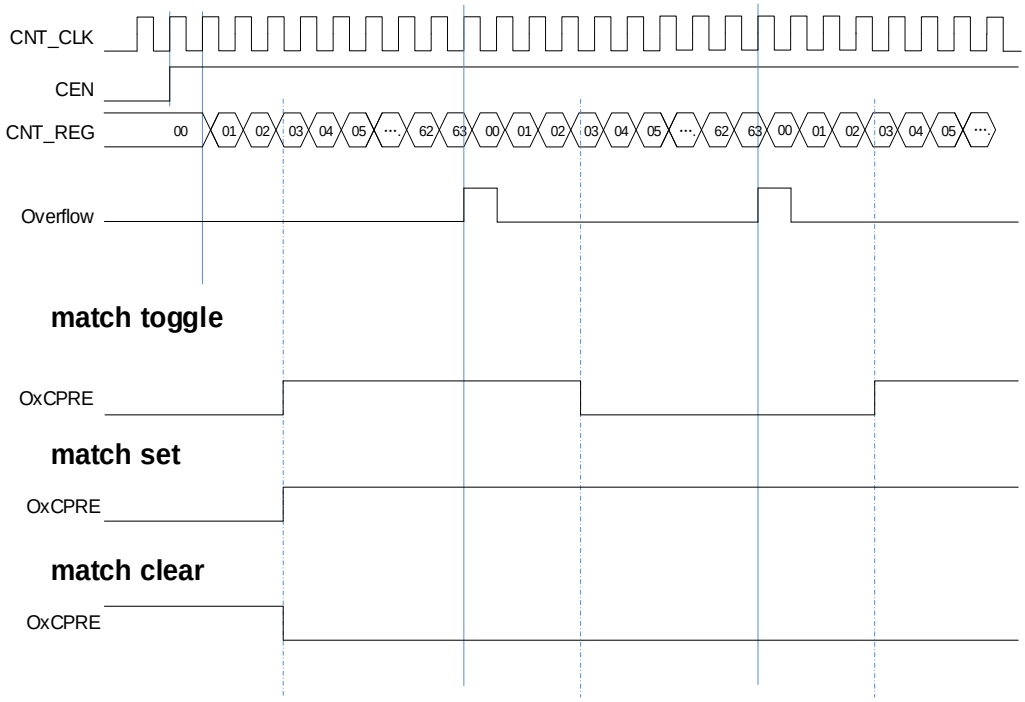
第四步：通过 TIMERx_CAR 寄存器和 TIMERx_CHxCV 寄存器配置输出比较时基：

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 18-67. 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 18-67. 三种输出比较模式



通道输出准备信号

根据*图 18-66. 通道输出比较原理*所示，当 TIMERx 用于输出匹配比较模式下，在通道输出信号之前会产生一个中间信号 OxCPRE 信号(通道 x 输出准备信号)。设置 CHxCOMCTL 位可以定义 OxCPRE 信号类型。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 输出准备信号)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 位域为 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。

定时器调试模式

当 Cortex®-M33 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。

18.4.5. TIMERx 寄存器(x=9,10,12,13)

TIMER9 基地址：0x4001 5000

TIMER10 基地址：0x4001 5400

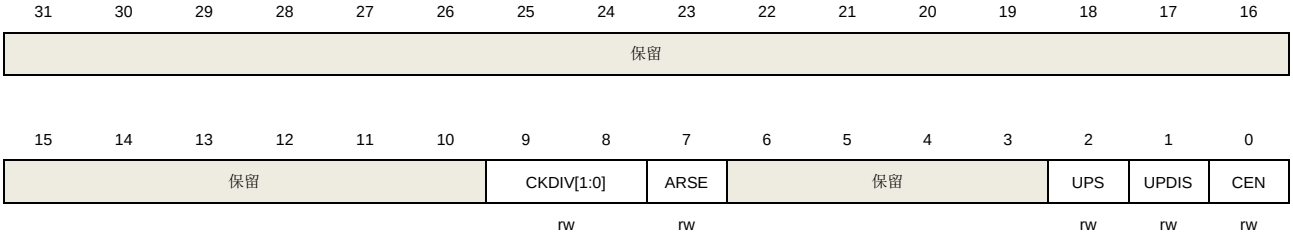
TIMER12 基地址：0x4000 1C00

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 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。 00：$f_{DTS}=f_{CK_TIMER}$ 01：$f_{DTS}= f_{CK_TIMER} /2$ 10：$f_{DTS}= f_{CK_TIMER} /4$ 11：保留
7	ARSE	自动重载影子使能 0：禁能 TIMERx_CAR 寄存器的影子寄存器 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 软件配置该位，选择更新事件源. 0：以下事件均会产生更新中断或DMA请求： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 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. 该位用来使能或禁能更新事件的产生 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： UPG位被置1 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 0：计数器禁能 1：计数器使能 在软件将 CEN 位置 1 后，外部时钟、暂停模式和正交译码器模式才能工作。

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									MMC[2:0]			保留			
									rw						

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 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 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： 主定时器产生一个复位事件 TIMERx_SWEVG寄存器中UPG位置1 001：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： CEN位置1 在暂停模式下，触发输入置1 010：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 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 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

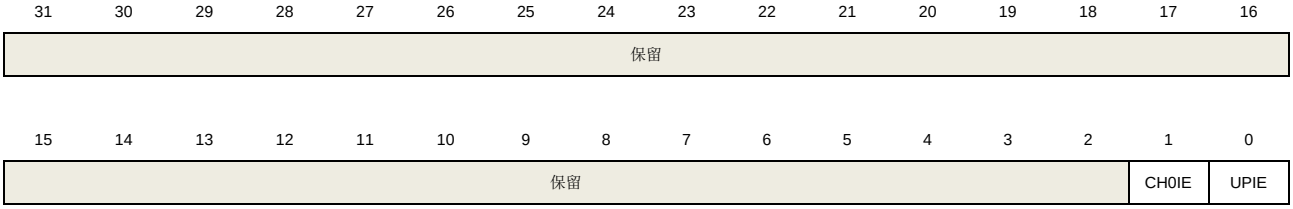
101：保留
110：保留
111：保留

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

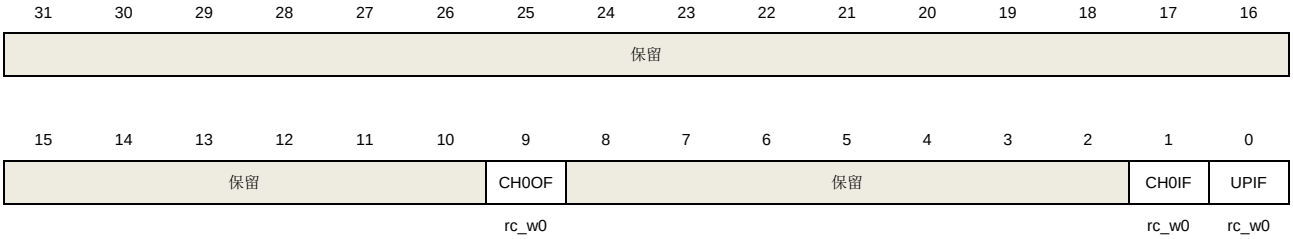


位/位域	名称	描述
31:2	保留	必须保持复位值
1	CH0IE	通道 0 比较/捕获中断使能 0：禁止通道 0 中断 1：使能通道 0 中断
0	UPIE	更新中断使能 0：禁止更新中断 1：使能更新中断

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
31:10	保留	必须保持复位值

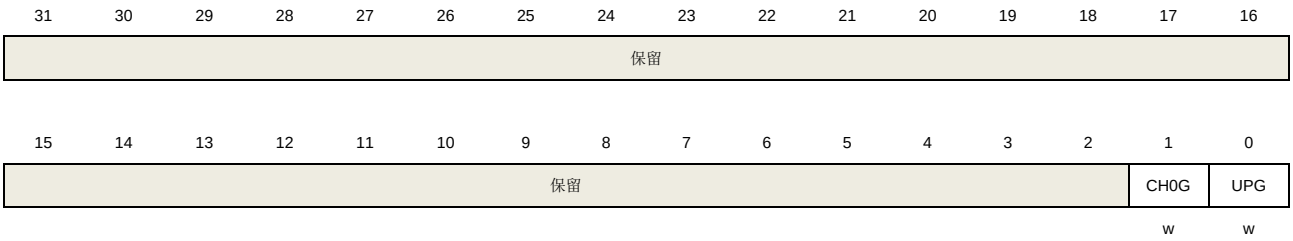
9	CH0OF	通道 0 捕获溢出标志 当通道 0 被配置为输入模式时，在 CH0IF 标志位已经被置 1 后，捕获事件再次发生时，该标志位可以由硬件置 1。该标志位由软件清 0. 0：无捕获溢出中断发生 1：发生了捕获溢出中断
8:2	保留	必须保持复位值
1	CH0IF	通道 0 比较/捕获中断标志 此标志由硬件置 1 软件清 0。当通道 0 在输入模式下时，捕获事件发生时此标志位被置 1；当通道 0 在输出模式下时，此标志位在一个比较事件发生时被置 1。 0：无通道 0 中断发生 1：通道 0 中断发生
0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置 1，软件清 0。 0：无更新中断发生 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 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 TIMERx_CH0CV 寄存器捕获，如果 CH0IF 标志位已经为 1，则 CH0OF 标志位被置 1。 0：不产生通道 0 捕获或比较事件 1：发生通道 0 捕获或比较事件
0	UPG	更新事件产生 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。 0：无更新事件产生

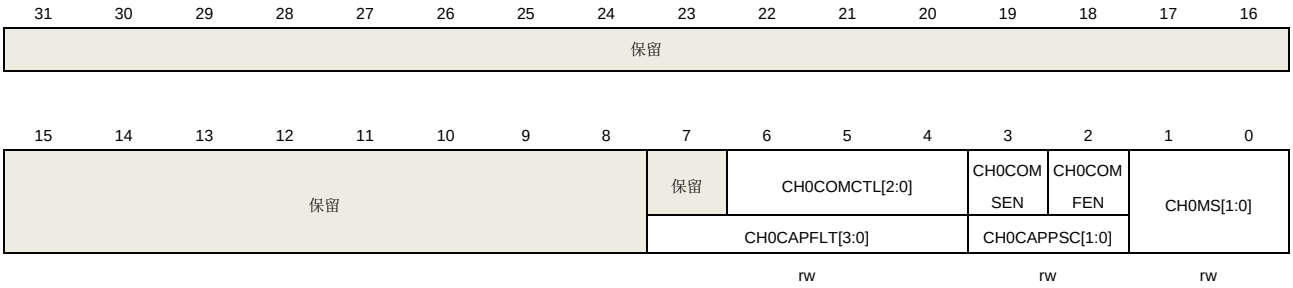
1：产生更新事件

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



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 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。 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 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。 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。 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

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。
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
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 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：保留 11：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 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 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 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。 0：通道0互补输出高电平为有效电平 1：通道0互补输出低电平为有效电平 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。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。
2	保留	必须保持复位值.
1	CH0P	通道 0 极性 当通道 0 配置为输出模式时，此位定义了输出信号极性。 0：通道0高电平为有效电平 1：通道0低电平为有效电平 当通道 0 配置为输入模式时，此位定义了 CI0 信号极性 [CH0NP, CH0P] 将选择 CI0FE0 或者 CI1FE0 的有效边沿或者捕获极性

		[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。 [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。 [CH0NP==1, CH0P==0]：保留。 [CH0NP==1, CH0P==1]：保留。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改保留。
0	CH0EN	通道 0 捕获/比较使能 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。 0：禁止通道 0 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															

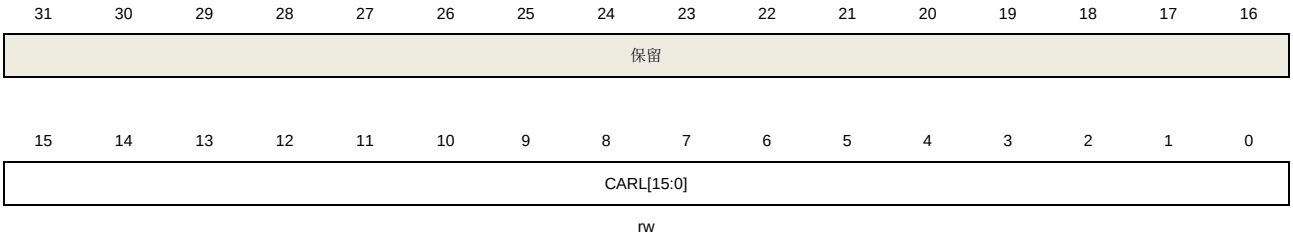
位/位域	名称	描述
31:16	保留	必须保持复位值.
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 TIMER_CK 时钟除以(PSC+1)，每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。

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



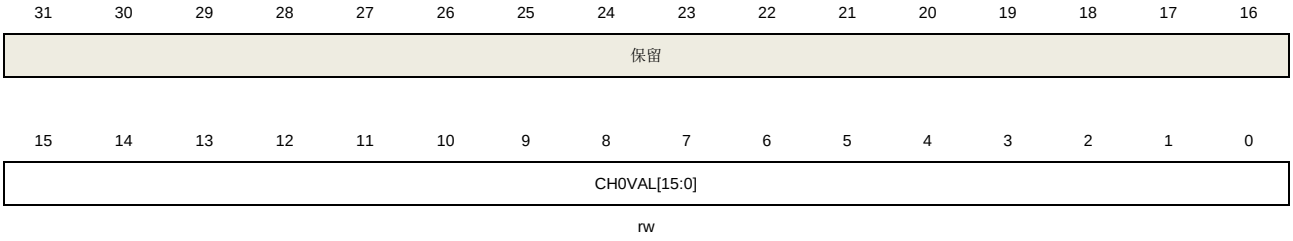
位/位域	名称	描述
31:16	保留	必须保持复位值.
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

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



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 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。 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

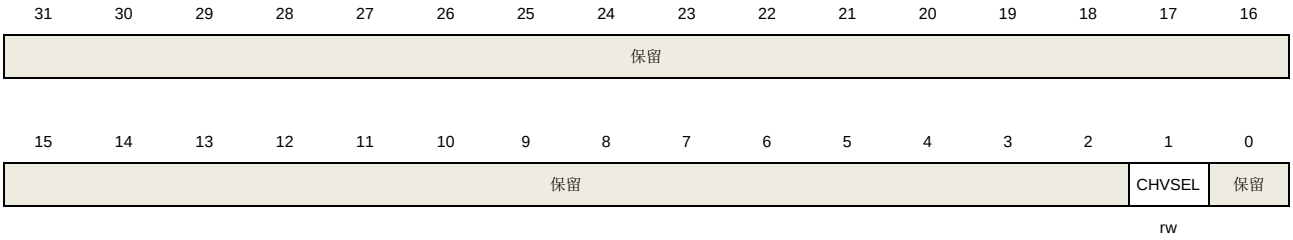
存器后，影子寄存器值随每次更新事件更新。

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



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 此位由软件写 1 或清 0。 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 0：无影响
0	保留	必须保持复位值

18.5. 基本定时器（TIMERx, x=5,6）

18.5.1. 简介

基本定时器(Timer5, 6)包含一个无符号 16 位计数器。可以被用作通用定时器和为 DAC (数字到模拟转换器)提供时钟。基本定时器可以配置产生 DMA 请求，TRGO 触发连接到 DAC。

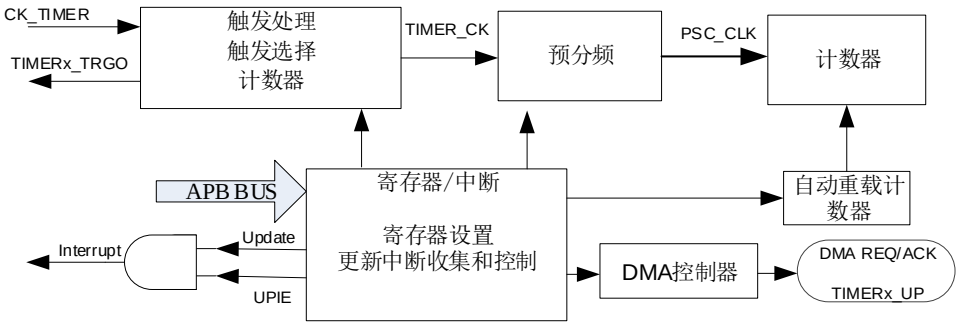
18.5.2. 主要特性

- 计数器宽度：16位
- 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

18.5.3. 结构框图

*图 18-68. 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 18-68. 基本定时器结构框图



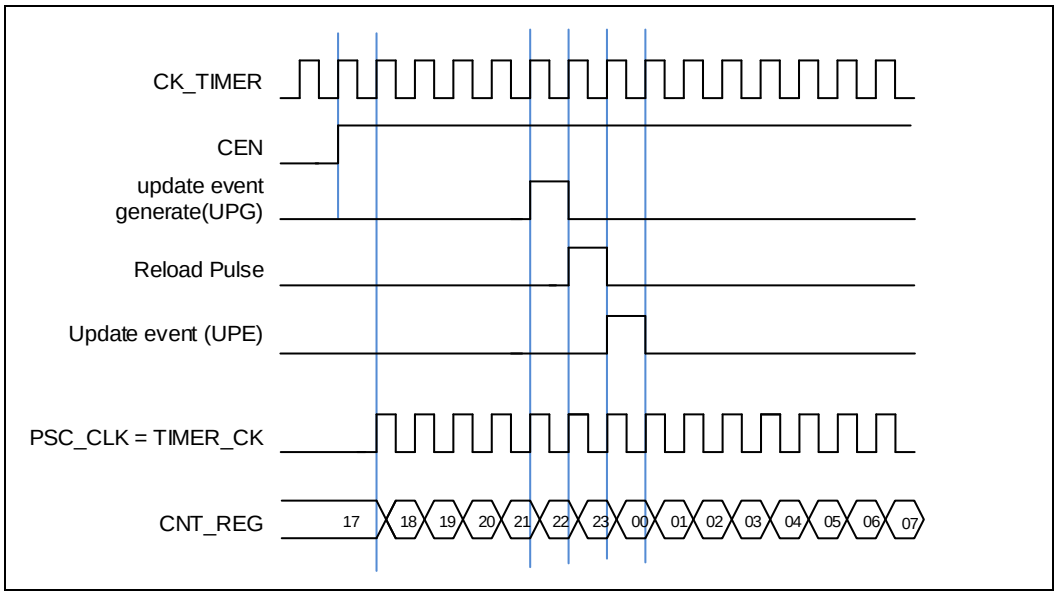
18.5.4. 功能描述

时钟源配置

基本定时器可以由内部时钟源 CK_TIMER 驱动 。

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。

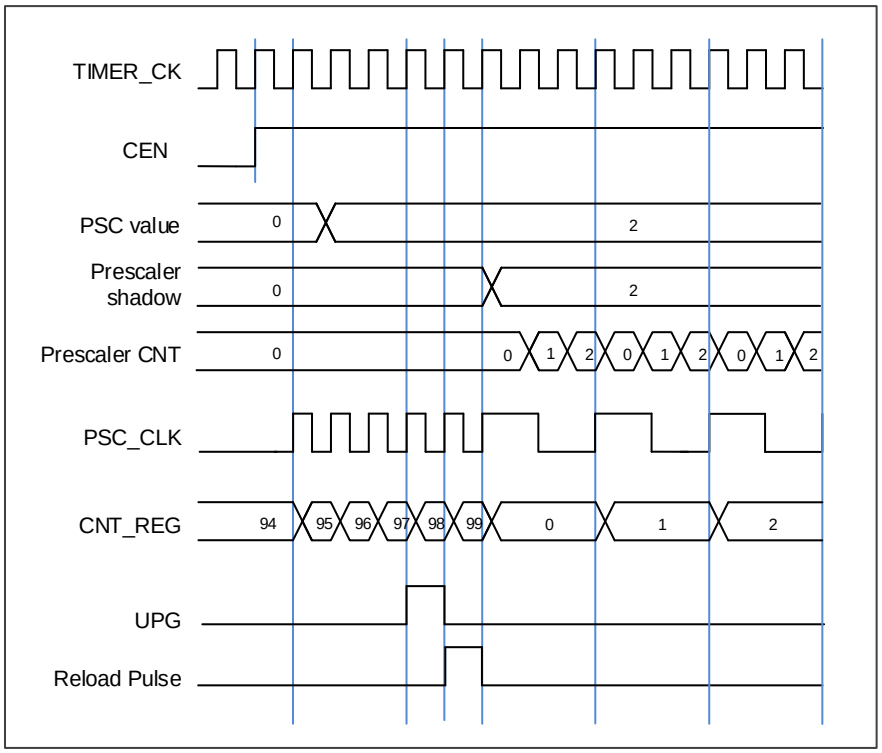
图 18-69. 内部时钟分频为 1 时，计数器的时序图



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 18-70. 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 18-71. 向上计数时序图，PSC=0/2

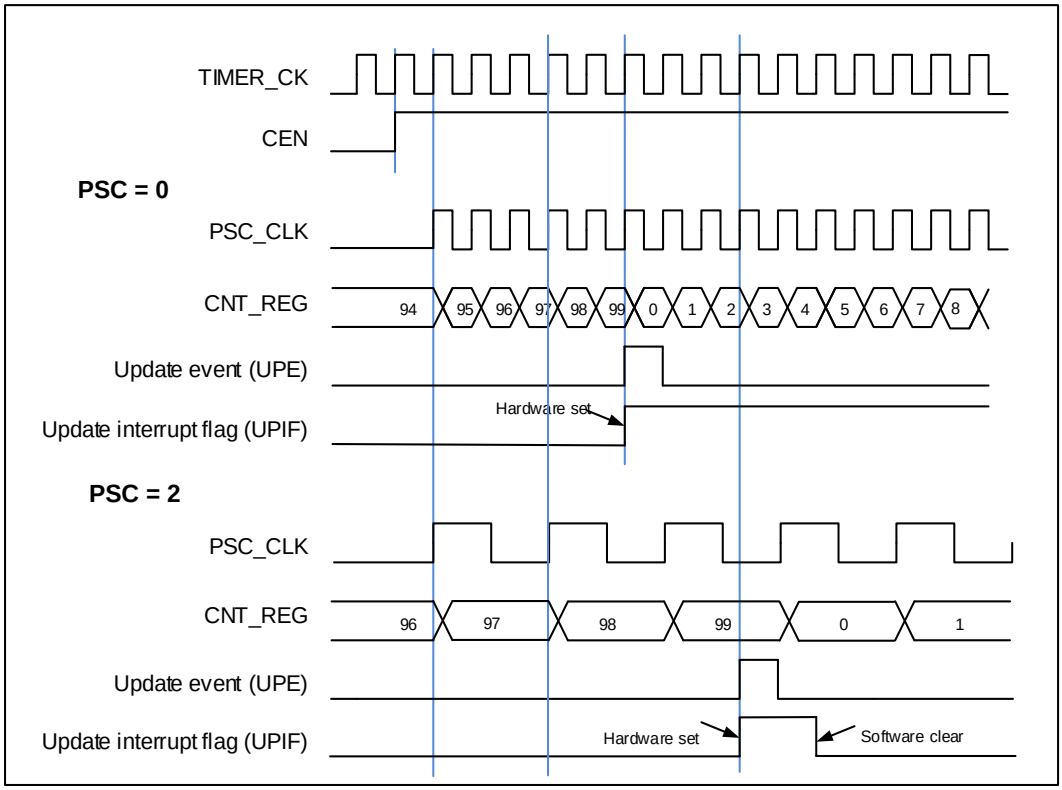
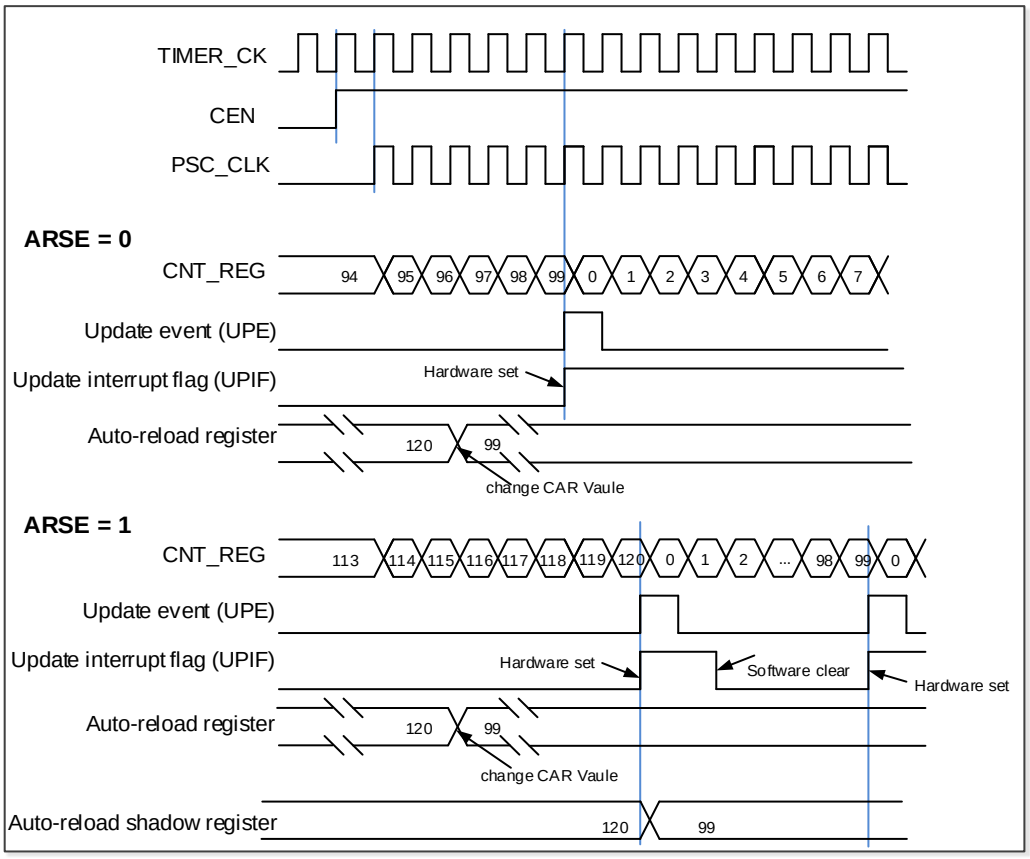


图 18-72. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



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®-M33 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。

18.5.5. TIMERx 寄存器(x=5,6)

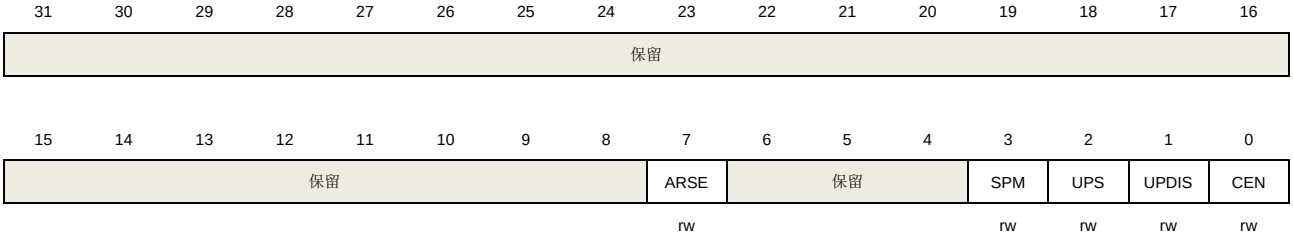
TIMER5 基地址：0x4000 1000

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 0: 禁能 TIMERx_CAR 寄存器的影子寄存器 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 0：单脉冲模式禁能。更新事件发生后，计数器继续计数 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 软件配置该位，选择更新事件源. 0：以下事件均会产生更新中断或DMA请求： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 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. 该位用来使能或禁能更新事件的产生 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： UPG位被置1 计数器溢出/下溢

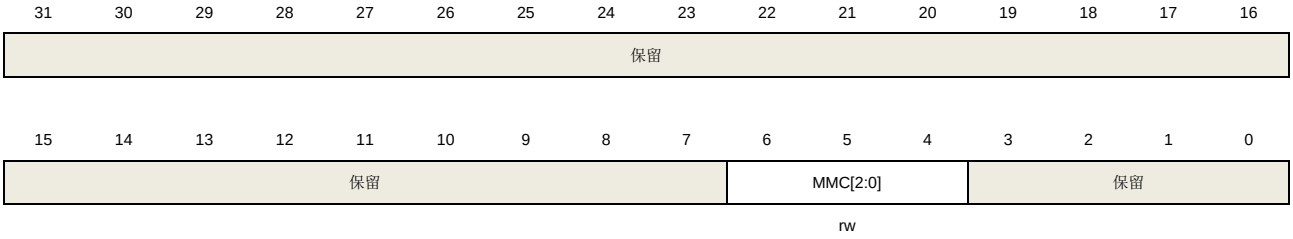
复位模式产生的更新
1：更新事件禁能.
注意：当该位被置 1 时，UPG 位被置 1 或者复位模式不会产生更新事件，但是计数器和预分频器被重新初始化

0	CEN	计数器使能 0: 计数器禁能 1: 计数器使能 在软件将 CEN 位置 1 后，外部时钟、暂停模式和正交译码器模式才能工作。

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

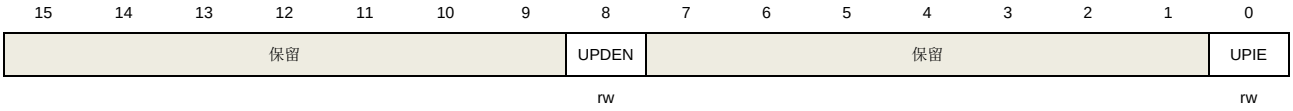


位/位域	名称	描述
31:7	保留	必须保持复位值
6:4	MMC[2:0]	这些位控制 TRGO 信号的选择，TRGO 信号由主定时器发给从定时器用于同步功能 000：当产生一个定时器复位事件后，输出一个TRGO信号，定时器复位源为： 主定时器产生一个复位事件 TIMERx_SWEVG寄存器中UPG位置1 001：当产生一个定时器使能事件后，输出一个TRGO信号，定时器使能源为： CEN位置1 在暂停模式下，触发输入置1 010：当产生一个定时器更新事件后，输出一个TRGO信号，更新事件源由UPDIS和UPS位决定
3:0	保留	必须保持复位值

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



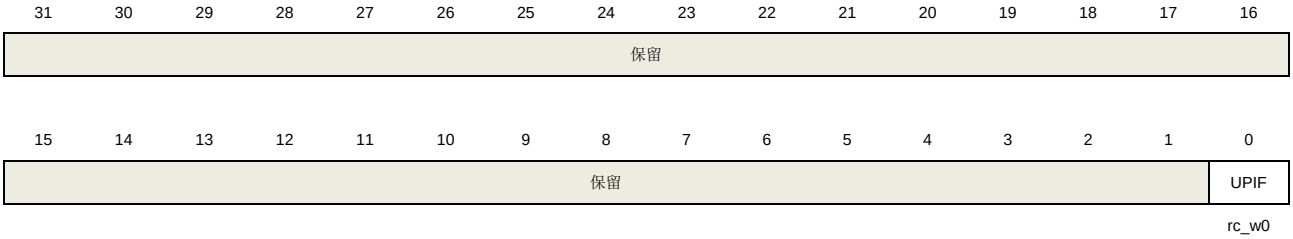
位/位域	名称	描述
31:9	保留	必须保持复位值
8	UPDEN	更新 DMA 请求使能 0：禁止更新 DMA 请求 1：使能更新 DMA 请求
7:1	保留	必须保持复位值
0	UPIE	更新中断使能 0：禁止更新中断 1：使能更新中断

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



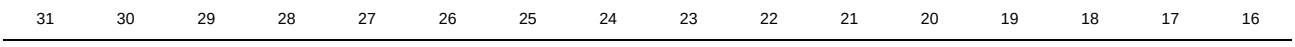
位/位域	名称	描述
31:1	保留	必须保持复位值
0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置 1，软件清 0。 0：无更新中断发生 1：发生更新中断

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



保留

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

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 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。 0：无更新事件产生 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

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															

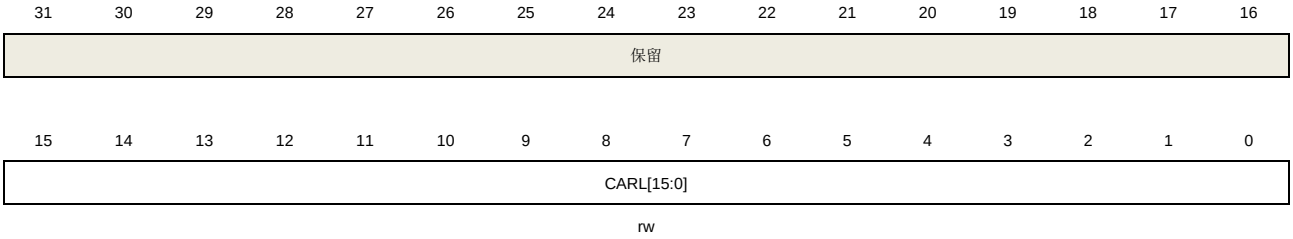
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 TIMER_CK 时钟除以(PSC+1)，每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。

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



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 这些位定义了计数器的自动重载值。

19. 超高分辨率定时器（SHRTIMER）

19.1. 简介

SHRTIMER 具有超高分辨率计数时钟，可用于高精度定时。它可以产生 10 个超高分辨率的数字信号来灵活地控制电动机或用于电源管理应用。这 10 个数字信号可以独立输出，也可以耦合成 5 对互补信号输出。

SHRTIMER 具有灵活的捕获功能，可用于捕获输入信号的时序。它具有多个连接到 ADC 和 DAC 的内部信号，可用于控制和监视。

为了安全起见，SHRTIMER 可处理各种故障输入。

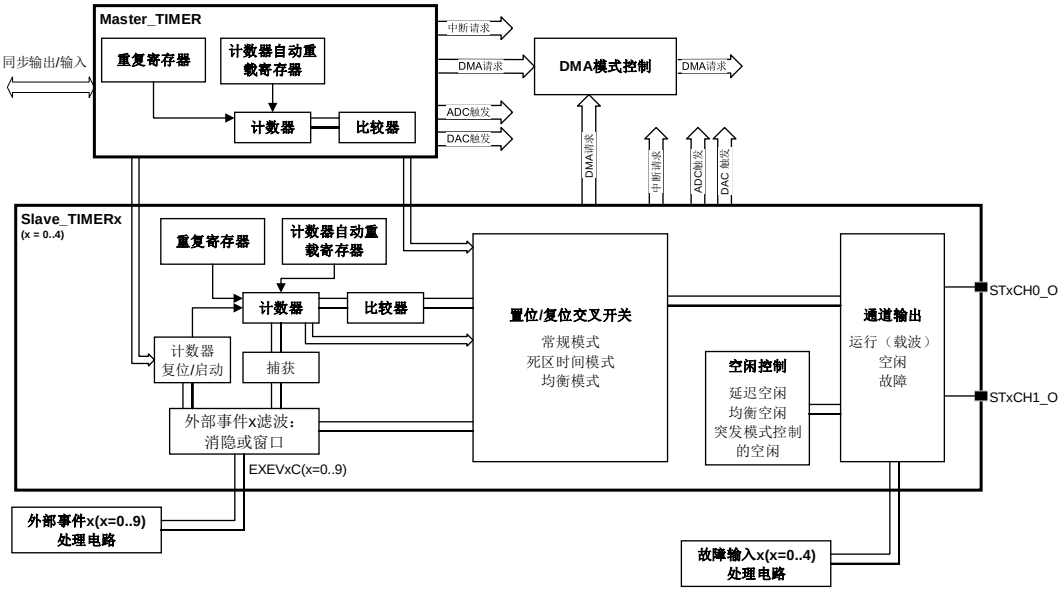
19.2. 主要特征

- 超高分辨率定时单元：Master_TIMER，Slave_TIMERx (x=0..4)；
- 10个数字信号输出：它们可由任意一个定时单元控制，可独立输出也可耦合成5对互补输出；
- 同步输出：作为主机同步外部资源；
- 同步输入：作为从机与外部资源同步；
- 多个内部信号连接到ADC和DAC；
- 多种故障输入保护机制：故障输入通道和系统故障；
- 突发模式控制器应用于轻载操作；
- 7个中断向量：Master_TIMER中断，Slave_TIMERx（x = 0..4）中断和故障中断；
- 6个DMA请求：Master_TIMER请求和Slave_TIMERx（x = 0..4）请求。DMA模式可以更新多个寄存器；
- DMA模式用于多个寄存器的更新。

19.3. 结构框图

*图 19-1. SHRTIMER 结构框图*给出了 SHRTIMER 的内部细节配置。

图 19-1. SHRTIMER 结构框图



19.4. 功能说明

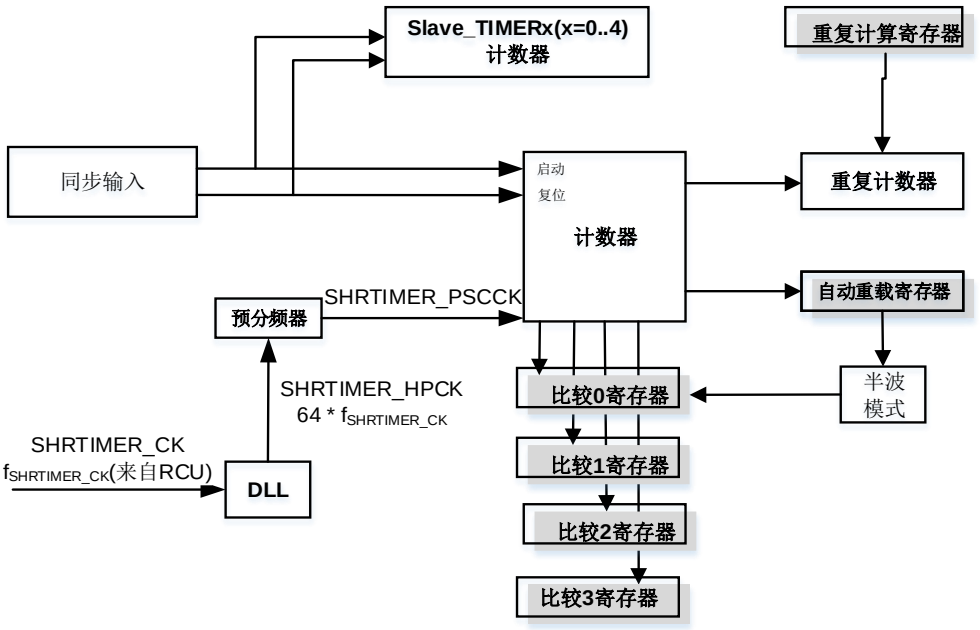
19.4.1. Master_TIMER 单元

Master_TIMER 单元由以下模块组成：

- 16位计数器
- 自动重载寄存器：确定计数周期
- 重复计数器
- 比较寄存器y(y=0..3)

*图 19-2. Master_TIMER 结构图*给出了 Master_TIMER 的内部细节配置。

图 19-2. Master_TIMER 结构图



自动重载寄存器和比较 y（y = 0..3）寄存器具有以下限值：

当 CNTCKDIV[3:0] < 4'b0101 或 CNTCKDIV[3:0] = 4'b1000

- 最小值必须大于或等于3个$t_{SHRTIMER_CK}$周期对应的计数值；
- 最大值必须小于或等于（0xFFFF – 1个$t_{SHRTIMER_CK}$周期对应的计数值）。

注意：每个 $t_{SHRTIMER_CK}$ 周期对应的计数值 = $f_{SHRTIMER_PSSCK}$ / $f_{SHRTIMER_CK}$。

当 CNTCKDIV[3:0] >= 4'b0101(不包括 CNTCKDIV[3:0] = 4'b1000)

- 最小值必须大于或等于0x0003；
- 最大值必须小于或等于0xFFFE。

具体请见*表 19-1. 自动重载寄存器和比较 y（y = 0..3）寄存器的限值*。

表 19-1. 自动重载寄存器和比较 y（y = 0..3）寄存器的限值

CNTCKDIV[3:0]	最小值	最大值
4'b1000	0x00C0	0xFFBF
4'b0000	0x0060	0xFFDF
4'b0001	0x0030	0xFFEF
4'b0010	0x0018	0xFFF7
4'b0011	0x000C	0xFFFB
4'b0100	0x0006	0xFFFD
4'b0101	0x0003	0xFFFE
4'b0110	0x0003	0xFFFE
4'b0111	0x0003	0xFFFE

计数器时钟

Master_TIMER 的时钟源是由 RCU 模块提供的 SHRTIMER_CK，DLL 用于产生超高分辨率时

钟 SHRTIMER_HPCK（$f_{SHRTIMER_HPCK}$ = 64 * $f_{SHRTIMER_CK}$），更多信息请参考 ***DLL 校准***。

当 SHRTIMER_MTACTL 寄存器中的 CNTCKDIV[3]位为 0 时，预分频器（PSC）将超高分辨率时钟（SHRTIMER_HPCK）除以分频因子 $2^{CNTCKDIV[2:0]+1}$，得到计数器时钟（SHRTIMER_PSCCK）。该分频因子由 SHRTIMER_MTCTL0 寄存器中的 CNTCKDIV[2:0]位域控制。它们之间的频率关系可以表示如下：

$$f_{SHRTIMER_PSCCK} = f_{SHRTIMER_HPCK} / 2^{CNTCKDIV[2:0]+1} \quad (19\text{-}1)$$

当 SHRTIMER_MTACTL 寄存器中的 CNTCKDIV[3]位为 1 时，CNTCKDIV[2:0]位只能配置为 3’b000， SHRTIMER_PSCCK 和 SHRTIMER_HPCK 之间的频率关系可以表示为：

$$f_{SHRTIMER_PSCCK} = f_{SHRTIMER_HPCK} \quad (19\text{-}2)$$

注意：一旦 Master_TIMER 使能了，就不能修改时钟分频 CNTCKDIV[3:0]的值。

*图 19-3. 分频为 32 时计数器时钟*显示了将寄存器 SHRTIMER_MTCAR 设置为 0x0104，SHRTIMER_MTCTL0 寄存器中的位域 CNTCKDIV[3: 0]设置为 4'b0100 时计数器的动作。

图 19-3. 分频为 32 时计数器时钟

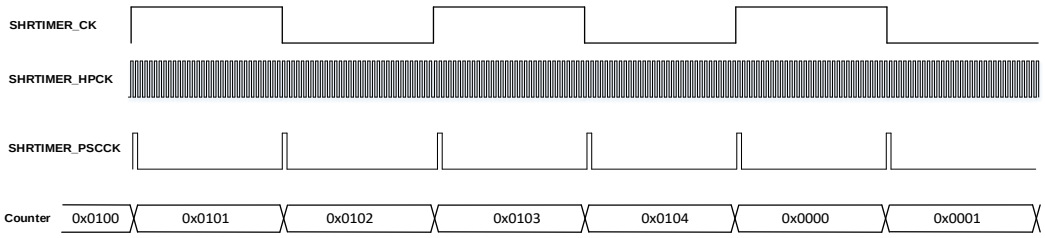


*表 19-2. $f_{SHRTIMER_CK}$ = 180MHz 的分辨率*列出了 $f_{SHRTIMER_CK}$ 为 180MHz 时的不同分辨率。

表 19-2. $f_{SHRTIMER_CK}$ = 180MHz 的分辨率

CNTCKDIV[3:0]	$f_{SHRTIMER_PSCCK}$	分辨率
4'b0000	180*32MHz=5.76GHz	173.6ps
4'b0001	180*16MHz=2.880GHz	347.2ps
4'b0010	180*8MHz=1.440GHz	694.4ps
4'b0011	180*4MHz=720MHz	1.4ns
4'b0100	180*2MHz=360MHz	2.8ns
4'b0101	180*1MHz=180MHz	5.6ns
4'b0110	180/2MHz=90MHz	11.1ns
4'b0111	180/4MHz=45MHz	22.2ns
4'b1000	180*64MHz=11.520G	86.8ps

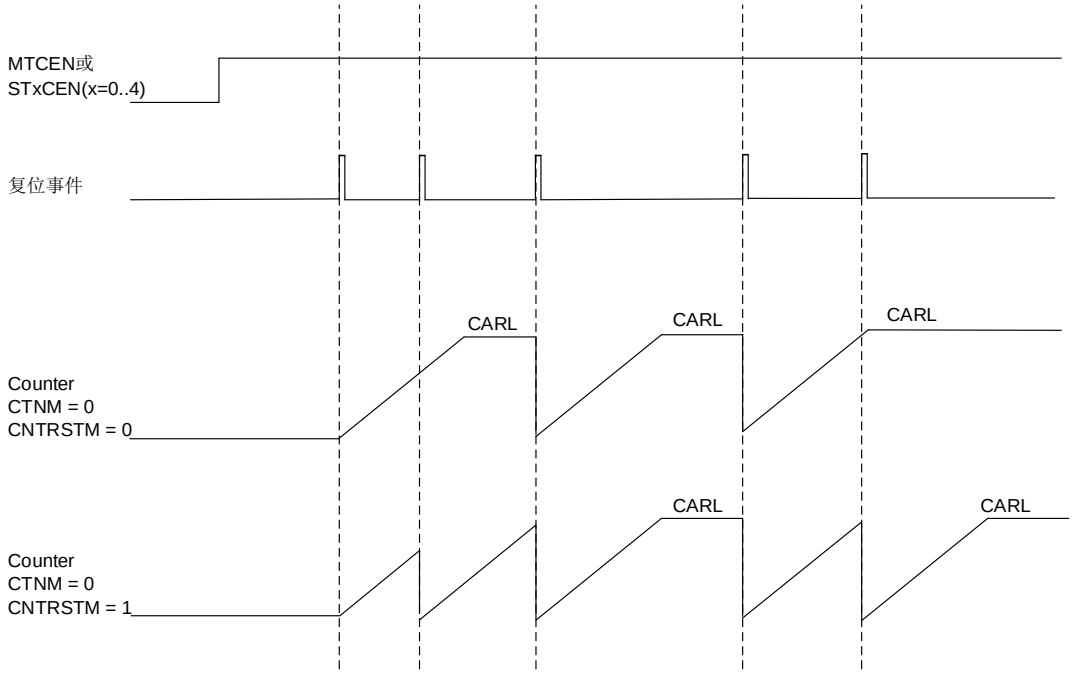
向上计数模式

计数器从 0 连续递增到计数器重载值，该值在 SHRTIMER_MTCAR 寄存器中定义。计数器有两种工作模式：单脉冲模式（SHRTIMER_MTCTL0 寄存器中的 CTNM = 0）或连续模式（CTNM = 1）。

在单脉冲模式下，将 SHRTIMER_MTCTL0 寄存器中的 MTCEN 位置 1 后，第一个复位事件将启动计数器。当计数到计数器重载值时，计数器停止并生成周期事件。然后，其他的复位事

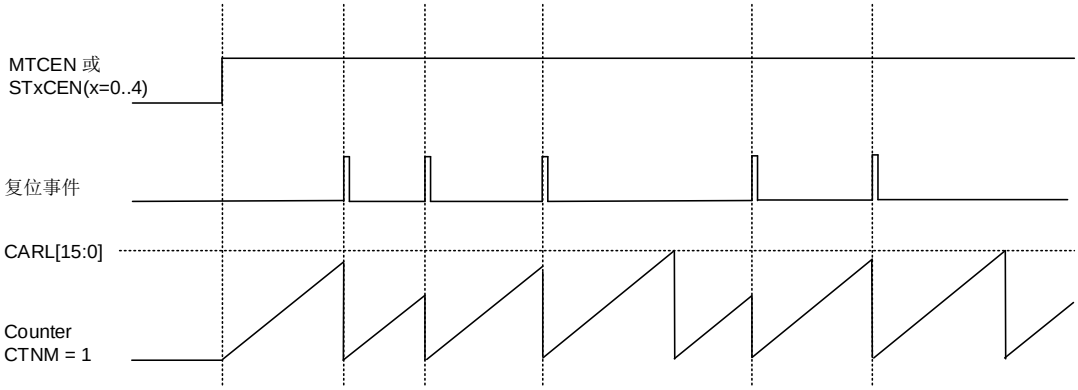
件将复位并重新启动计数器。在计数过程中，如果 SHRTIMER_MTCTL0 寄存器中的 CNTRSTM = 1，则复位事件将复位并重新启动计数器，否则将被忽略。*图 19-4. 单脉冲模式下计数器的动作*显示了单脉冲模式下的计数器运行情况。

图 19-4. 单脉冲模式下计数器的动作



在连续模式下，一旦 SHRTIMER_MTCTL0 寄存器中的 MTCEN 位置 1，计数器将立即启动。当计数到计数器重载值时，计数器从 0 重新启动，并产生翻转事件。与单脉冲模式不同，随时生成的复位事件将复位并重启计数器。*图 19-5. 连续模式下计数器的动作*显示了连续模式下的计数器运行情况。

图 19-5. 连续模式下计数器的动作



重复计数器

SHRTIMER_MTCTL0 寄存器中的 MTCEN 位置 1 时，重复计数器将加载 SHRTIMER_MTCREP 寄存器的值。当由于复位事件或连续模式下的翻转事件清零计数器时，重复计数器值递减。当重复计数器值达到零时，复位事件或连续模式下的翻转事件将产生一个

重复事件并重新加载 SHRTIMER_MTCREP 寄存器的值。

重复事件会将 SHRTIMER_MTINTF 寄存器中的 REPIF 位置 1，如果使能了相应中断或 DMA 请求（SHRTIMER_MTDMAINTEN 寄存器中的 REPIE = 1 或 REPDEN = 1），则会产生重复中断和 DMA 请求。可以通过向 SHRTIMER_MTINTFC 寄存器中的 REPIFC 位写 1 来清除重复中断标志。

*图 19-6. 连续模式下重复计数器的动作*显示了在连续模式下重复计数器的运行情况。

图 19-6. 连续模式下重复计数器的动作

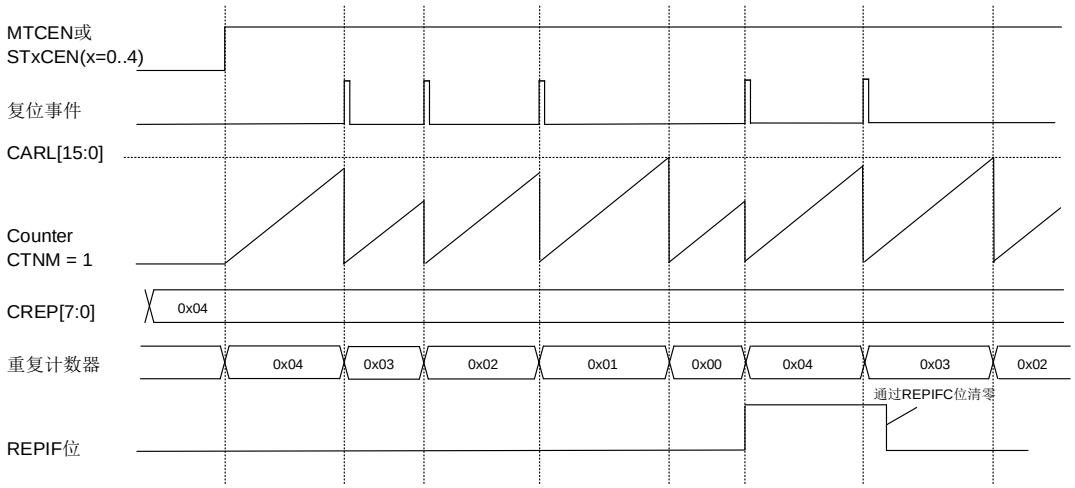


*图 19-7. 单脉冲模式下，CNTRSTM = 0 时重复计数器的动作*显示了单脉冲模式下，CNTRSTM = 0 时重复计数器的运行情况。

图 19-7. 单脉冲模式下，CNTRSTM = 0 时重复计数器的动作

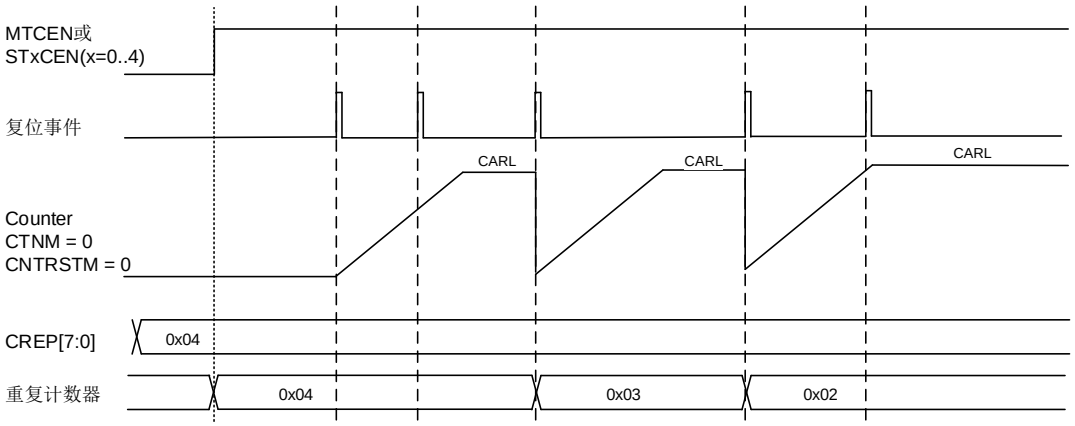
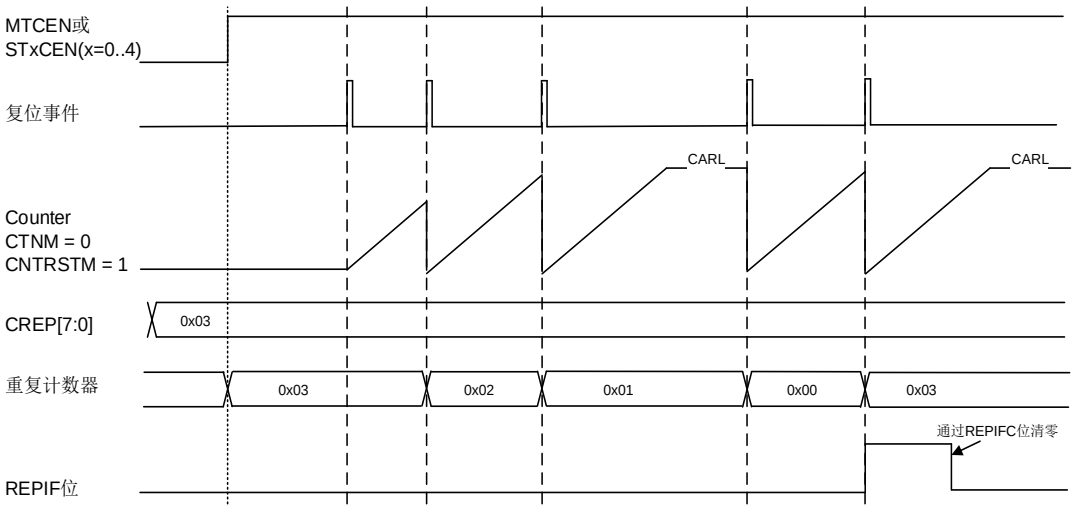


*图 19-8. 单脉冲模式下，CNTRSTM = 1 时重复计数器的动作*显示了单脉冲模式下，CNTRSTM = 1 时重复计数器的运行情况。

图 19-8. 单脉冲模式下，CNTRSTM = 1 时重复计数器的动作



计数器复位

一旦计数器（MTCEN = 1）使能了，就可以通过软件或同步输入将计数器复位为 0。

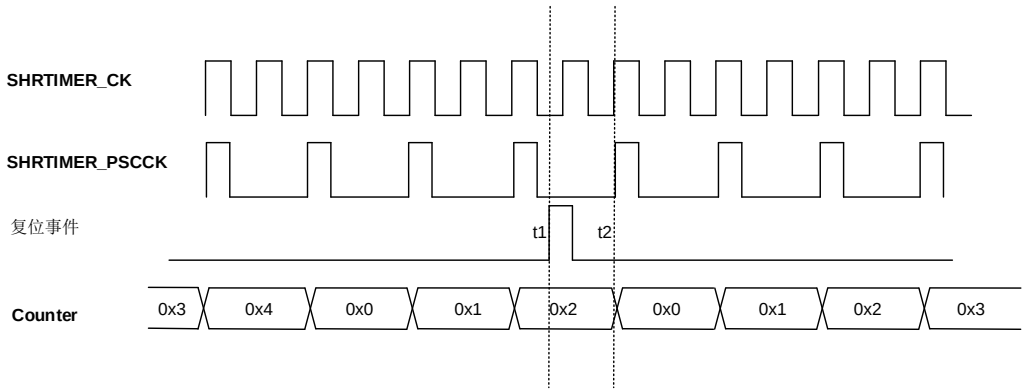
将 MTSRST 位置 1（由硬件自动清除）将使计数器复位。

当 SHRTIMER_MTCTL0 寄存器中的 SYNIRST 位置 1 时，同步输入可以复位计数器。详细信息请参考*同步输入*。

当计数器时钟 SHRTIMER_PSCCK 的预分频系数大于 64（CNTCKDIV [3]=1’b0，且 CNTCKDIV [2:0] > 3’b101）时，计数器复位事件将延迟到 SHRTIMER_PSCCK 的下一个上升沿。

*图 19-9. 当预分频系数为 128 时，复位事件重新同步*显示了连续模式下，CNTCKDIV[3:0] = 4’b0110，SHRTIMER_MTCAR = 0x4 时的运行情况。

图 19-9. 当预分频系数为 128 时，复位事件重新同步



比较

Master_TIMER 具有四个比较寄存器：SHRTIMER_MTCMPxV（x = 0..3）。当计数器值与比较寄存器值匹配时，将生成一个对应的比较事件。

比较事件会将相应的比较中断标志位置 1（SHRTIMER_MTINTF 寄存器中的 CMPxIF 位，x =

0..3），如果比较中断或 DMA 请求使能（SHRTIMER_MTDMAINTEN 寄存器中的 CMPxIE = 1 或 CMPxDEN = 1，x = 0..3），则会产生一个比较中断或 DMA 请求。通过写 1 到 SHRTIMER_MTINTFC 中的 CMPxIF 位（x = 0..3）可以清除比较中断标志。

半波模式

当 SHRTIMER_MTCTL0 中的 HALFM 位置 1 时，半波模式使能。此模式将比较 0 有效寄存器的值强制为计数器重载值的一半，但 SHRTIMER_MTCMP0V 寄存器的值不会更新为（SHRTIMER_MTCAR / 2）的值。半波模式主要用于生成固定占空比为 50%的方波。

当 SHRTIMER_MTCTL0 寄存器中的 SHWEN 位置 1 时，将使能影子寄存器，比较 0 有效寄存器的值在更新事件时刷新。反之，比较 0 有效寄存器在新值写入 SHRTIMER_MTCAR 后立即刷新。

同步输入启动/复位计数器

当 SHRTIMER_MTCTL0 寄存器中的 SYNIRST 位置 1 时，同步输入可以产生计数器复位事件；当 SHRTIMER_MTCTL0 寄存器中的 SYNISTRT 位置 1 时，同步输入可以启动计数器。更多信息请参考*同步输入*。

同步输入请求会将 SHRTIMER_MTINTF 寄存器中的 SYNIIF 位置 1，如果使能了中断或 DMA 请求（SHRTIMER_MTDMAINTEN 寄存器中的 SYNIIE = 1 或 SYNIDEN = 1），会产生相应的中断或 DMA 请求。可以通过写 1 到 SHRTIMER_MTINTFC 寄存器中的 SYNIIFC 位清除同步输入中断标志。

更新事件和影子寄存器

Master_TIMER 中的某些寄存器具有影子寄存器。MCU 复位后，影子寄存器被禁用。如果将 SHRTIMER_MTCTL0 寄存器中的 SHWEN 位清 0，则将禁用影子寄存器。写入这些寄存器的值将转移到活动寄存器中并生效。

SHRTIMER_MTCTL0 寄存器中的 SHWEN 位置 1，将使能影子寄存器并预加载这些寄存器。写入这些寄存器的值将被传送到影子寄存器，且不会立即生效。当发生更新事件时，影子寄存器内容将转移到活动寄存器中并立即生效。

注意：当 SHWEN=1 时，才会产生更新事件。

*表 19-3. Master_TIMER 影子寄存器和更新事件*列出了具有影子寄存器的寄存器和相应的更新事件。

表 19-3. Master_TIMER 影子寄存器和更新事件

具有影子寄存器的寄存器	影子寄存器使能位	更新事件
SHRTIMER_MTDMAINTEN	SHRTIMER_MTCTL0 寄存器中的 SHWEN 位	软件（MTSUP位）
SHRTIMER_MTCAR		重复事件（UPREP = 1）
SHRTIMER_MTCREP		DMA模式结束事件（UPSEL[1:0] = 2'b01）
SHRTIMER_MTCMP0V		DMA模式结束事件之后的翻转事件（UPSEL[1:0] = 2'b10）
SHRTIMER_MTCMP1V		

具有影子寄存器的寄存器	影子寄存器使能位	更新事件
SHRTIMER_MTCMP2V		
SHRTIMER_MTCMP3V		

Master_TIMER 有 4 个更新选项：

1. 软件生成更新事件。写 1 到 SHRTIMER_CTL1 寄存器的 MTSUP 位可以产生更新事件。此时，无论 SHRTIMER_MTCTL0 寄存器中的 UPSEL[1:0]位如何配置，所有挂起的硬件更新事件都将被忽略；
2. 重复事件生成更新事件。如果 SHRTIMER_MTCTL0 寄存器中的 UPREP 位置 1，由翻转事件或复位事件引起的重复事件会生成更新事件。SHRTIMER_MTCTL0 寄存器中的 UPSEL[1:0] = 2'b10，则重复事件不生成更新事件；
3. 当 DMA 模式下的 DMA 传输完成时，生成更新事件。如果 SHRTIMER_MTCTL0 寄存器中的 UPSEL[1:0] = 2'b01，则在 DMA 模式下的 DMA 传输完成时，硬件会自动生成更新事件。也可以通过软件或重复事件来生成更新事件。
4. 当 DMA 模式下的 DMA 传输完成后，计数器的翻转会产生更新事件。如果 SHRTIMER_MTCTL0 寄存器中的 UPSEL [1:0] = 2'b10，则在 DMA 模式下的 DMA 传输完成后，计数器发生翻转事件时，硬件会自动生成更新事件。也可以通过软件生成更新事件。

更新事件会将 SHRTIMER_MTINTF 寄存器中的 UPIF 位置 1，如果使能了相应的中断和 DMA 功能（SHRTIMER_MTDMAINTEN 寄存器中的 UPIE = 1 或 UPDEN = 1），则会产生中断或 DMA 请求。可以通过将 SHRTIMER_MTINTFC 中的 UPIFC 位写 1，来清除更新事件中断标志。

DAC 触发

当 Master_TIMER 的更新事件发生时，如果 SHRTIMER_MTCTL0 寄存器中的 DACTRGS[1:0] != 2'b00，则在 SHRTIMER_DACTRIGOx（x = 0..3）上生成 DAC 触发请求。如果 SHRTIMER_MTCTL0 寄存器中的 DACTRGS [1:0] = 2'b00，则不会生成 DAC 触发请求。SHRTIMER_DACTRIGOx（x = 0..3）是从 Master_TIMER 连接到 DAC 模块的内部信号。有关更多信息，请参考 ***DAC 触发***。

19.4.2. Slave_TIMERx(x=0..4)单元

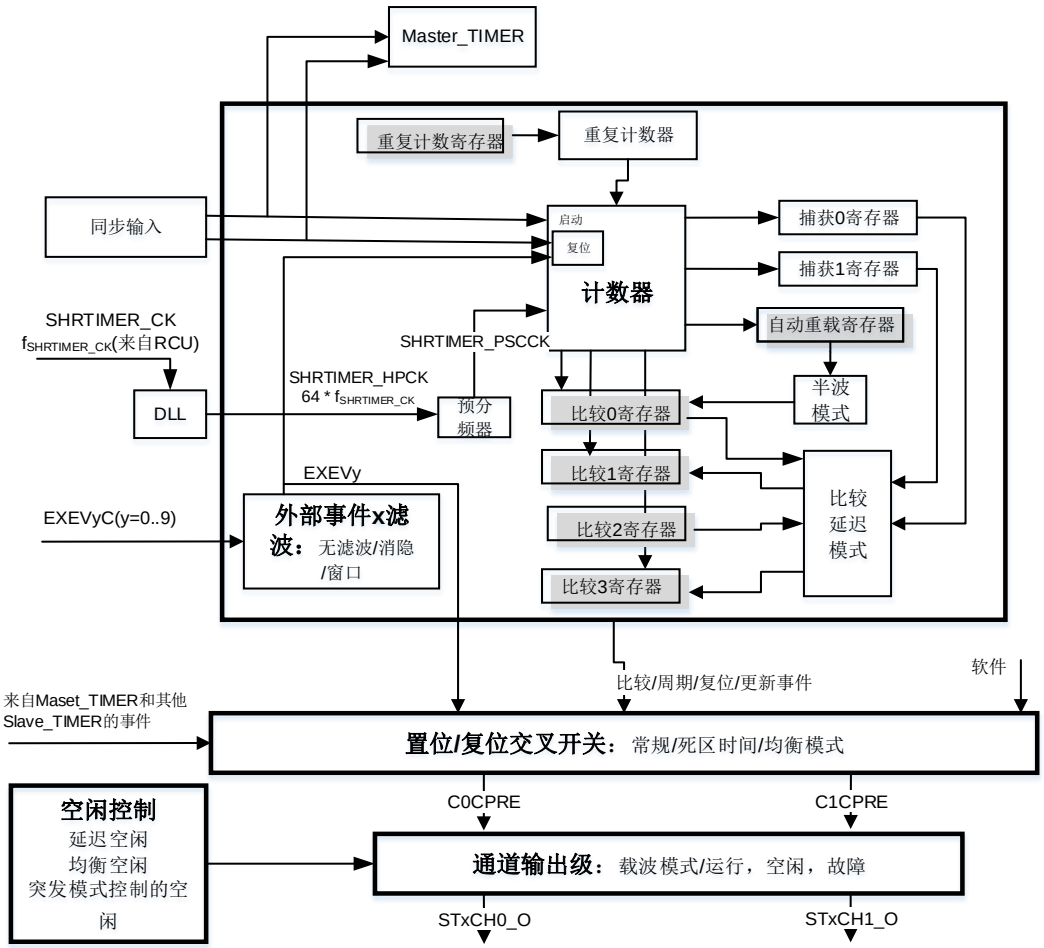
SHRTIMER 具有 5 个相同结构的从定时器：Slave_TIMERx（x = 0..4）。每个从定时器都由以下组件构成：

- 16位计数器
- 自动重载寄存器：计数周期值
- 重复计数器
- 比较寄存器y(y=0..3)
- 捕获寄存器y(y=0,1)
- 置位/复位交叉开关
- 空闲控制级

- 通道输出级

*图 19-10. Slave_TIMERx 结构框图*显示了 Slave_TIMERx 的结构框图。

图 19-10. Slave_TIMERx 结构框图



自动重载寄存器和比较 y（y = 0..3）寄存器具有以下限值：

当 CNTCKDIV[3:0] < 4'b0101 或 CNTCKDIV[3:0] = 4'b1000

- 最小值必须大于或等于3个$t_{SHRTIMER_CK}$周期对应的计数值；
- 最大值必须小于或等于（0xFFFF – 1个$t_{SHRTIMER_CK}$周期对应的计数值）。

注意：每个 $t_{SHRTIMER_CK}$ 周期对应的计数值 = $f_{SHRTIMER_PSSCK}$ / $f_{SHRTIMER_CK}$。

当 CNTCKDIV[3:0] >= 4'b0101(不包括 CNTCKDIV[3:0] = 4'b1000)

- 最小值必须大于或等于0x0003；
- 最大值必须小于或等于0xFFFE。

具体请见*表 19-1. 自动重载寄存器和比较 y（y = 0..3）寄存器的限值*。

计数器和捕获 y（y = 0,1）寄存器还具有以下限制：对于计数器时钟分频低于 64（CNTCKDIV[3：0] <5），最低有效位忽略。它们不能进行写操作和读操作时值为零。详见*表 19-4. 计数器和捕获 y(y=0,1) 寄存器限值*。

表 19-4. 计数器和捕获 y(y=0,1)寄存器限值

CNTCKDIV[3:0]	无效位
4'b1000	位 5~位 0
4'b0000	位 4~位 0
4'b0001	位 3~位 0
4'b0010	位 2~位 0
4'b0011	位 1~位 0
4'b0100	位 0
4'b0101	x
4'b0110	x
4'b0111	x

注意：“x”表示所有位都有效。

计数器时钟

Slave_TIMERx 的时钟源是来自 RCU 模块的 SHRTIMER_CK，DLL 用于产生超高分辨率时钟 SHRTIMER_HPCK（$f_{SHRTIMER_HPCK} = 64 * f_{SHRTIMER_CK}$），更多信息请参考 ***DLL 校准***。

当 SHRTIMER_STxACTL 寄存器中的 CNTCKDIV[3]位为 0 时，预分频器（PSC）将超高分辨率时钟（SHRTIMER_HPCK）除以分频因子 $2^{CNTCKDIV[2:0]+1}$，得到计数器时钟（SHRTIMER_PSCCK）。该分频因子由 SHRTIMER_STxCTL0 寄存器中的 CNTCKDIV[2:0]位域控制。它们之间的频率关系可以表示如下：

$$f_{SHRTIMER_PSCCK} = f_{SHRTIMER_HPCK} / 2^{CNTCKDIV[2:0]+1} \quad (19\text{-}3)$$

当 SHRTIMER_STxACTL 寄存器中的 CNTCKDIV[3]位为 1 时，只能将 CNTCKDIV[2:0]位配置为3’b000，并且SHRTIMER_PSCCK和SHRTIMER_HPCK之间的频率关系可以表示为：

$$f_{SHRTIMER_PSCCK} = f_{SHRTIMER_HPCK} \quad (19\text{-}4)$$

注意：一旦 Slave_TIMERx 使能了，就不能修改时钟分频 CNTCKDIV[3:0]的值。其中，CNTCKDIV[3] 位在 SHRTIMER_STxACTL 寄存器中，CNTCKDIV[2:0] 在 SHRTIMER_STxCTL0 寄存器中。

参考***图 19-3. 分频为 32 时计数器时钟***和***表 19-2. fSHRTIMER CK = 180MHz 的分辨率***可得更多细节。

向上计数模式

计数器从 0 连续递增到计数器重载值，该值在 SHRTIMER_STxCAR 寄存器中定义。计数器有两种工作模式：单脉冲模式（SHRTIMER_STxCTL0 寄存器中的 CTNM = 0）和连续模式（SHRTIMER_STxCTL0 寄存器中的 CTNM = 1）。

在单脉冲模式下，将 SHRTIMER_MTCTL0 寄存器中的 STxCEN 位置 1 后，第一个复位事件将启动计数器。当计数到计数器重载值时，计数器停止并生成周期事件。然后，其他的复位事件将复位并重新启动计数器。在计数过程中，如果 SHRTIMER_STxCTL0 寄存器中的 CNTRSTM = 1，则复位事件将复位并重新启动计数器，否则将被忽略。***图 19-4. 单脉冲模式下计数器的动作***显示了单脉冲模式下的计数器运行情况。

在连续模式下，一旦 SHRTIMER_MTCTL0 寄存器中的 STxCEN 位置 1，计数器将立即启动。当计数到计数器重载值时，计数器从 0 重新启动，并产生翻转事件。与单脉冲模式不同，随时生成的复位事件将复位并重启计数器。*图 19-5. 连续模式下计数器的动作*显示了连续模式下的计数器运行情况。

重复计数器

SHRTIMER_MTCTL0 寄存器中的 STxCEN 位置 1 时，重复计数器将加载 SHRTIMER_STxCREP 寄存器的值。当由于复位事件或连续模式下的翻转事件清零计数器时，重复计数器值递减。当重复计数器值达到零时，复位事件或连续模式下的翻转事件将产生一个重复事件并重新加载 SHRTIMER_STxCREP 寄存器的值。

重复事件会将 SHRTIMER_STxINTF 寄存器中的 REPIF 位置 1，如果使能了相应中断或 DMA 请求（SHRTIMER_STxDMAINTEN 寄存器中的 REPIE = 1 或 REPDEN = 1），则会产生重复中断和 DMA 请求。可以通过向 SHRTIMER_STxINTFC 寄存器中的 REPIFC 位写 1，清除重复中断标志。

连续模式下重复计数器的运行情况如*图 19-6. 连续模式下重复计数器的动作*所示。

单脉冲模式下，CNTRSTM = 0 时重复计数器的运行情况如*图 19-7. 单脉冲模式下，CNTRSTM = 0 时重复计数器的动作*所示。

单脉冲模式下，CNTRSTM = 1 时重复计数器的运行情况如*图 19-8. 单脉冲模式下，CNTRSTM = 1 时重复计数器的动作*所示。

计数器复位

计数器可以通过以下三种信号源复位：

1.软件。软件写 1 到 SHRTIMER_CTL1 寄存器的 STxSRST 位。

2.同步输入启动/复位计数器。

3.SHRTIMER_STxCNTRST 寄存器中配置的事件。

所有这些源都是逻辑或，它们可以同时有效。如果在同一 $t_{SHRTIMER_CK}$ 周期中发生多个复位事件，则仅最后一个有效。

注意：如果外部事件配置为电平有效，则只能在 SHRTIMER_STxCNTRST 寄存器中启用一个外部事件。

写 1 到 STxSRST 位（由硬件自动清除）使计数器复位。Master_TIMER 和 Slave_TIMERx（x = 0..4）的这些控制位都在 SHRTIMER_CTL1 寄存器中，可以同时复位多个计数器。

当 SHRTIMER_STxCTL0 寄存器中的 SYNIRST 位置 1 时，同步输入可以复位计数器。请参考*同步输入*。

可以在 SHRTIMER_STxCNTRST 寄存器中同时配置 30 个事件来复位计数器，这些事件可以分为四类：

- Slave_TIMERx：更新事件，比较1事件和比较3事件；
- 其他Slave_TIMERy（例如x = 1，则y = 0，2..4）：比较0事件，比较1事件和比较3事件；

- Master_TIMER：比较0事件，比较1事件，比较2事件，比较3事件和复位事件；
- 外部事件y（y = 0..9）：EXEVy为Slave_TIMERx中的外部事件的滤波信号。

当计数器时钟 SHRTIMER_PSCCK 的预分频系数大于 64（CNTCKDIV [3]=1’b0，且 CNTCKDIV [2:0] > 3’b101）时，计数器复位事件将延迟到 SHRTIMER_PSCCK 的下一个上升沿。具体请见*图 19-9. 当预分频系数为 128 时，复位事件重新同步*。

计数器复位事件会将 SHRTIMER_STxINTF 寄存器中的 RSTIF 位置 1，如果使能了计数器复位中断或 DMA 请求（SHRTIMER_STxDMAINTEN 寄存器中的 RSTIE = 1 或 RSTDEN = 1），则会产生复位中断或 DMA 请求。可通过向 SHRTIMER_STxINTFC 中的 RSTIFC 位写 1，清除复位中断标志。

捕获

捕获功能不仅使 Slave_TIMERx 实现了脉冲宽度，频率，周期，占空比的测量，而且还可以在延迟模式下（参见*比较延迟模式*）更新比较 1 寄存器和比较 3 寄存器的值。

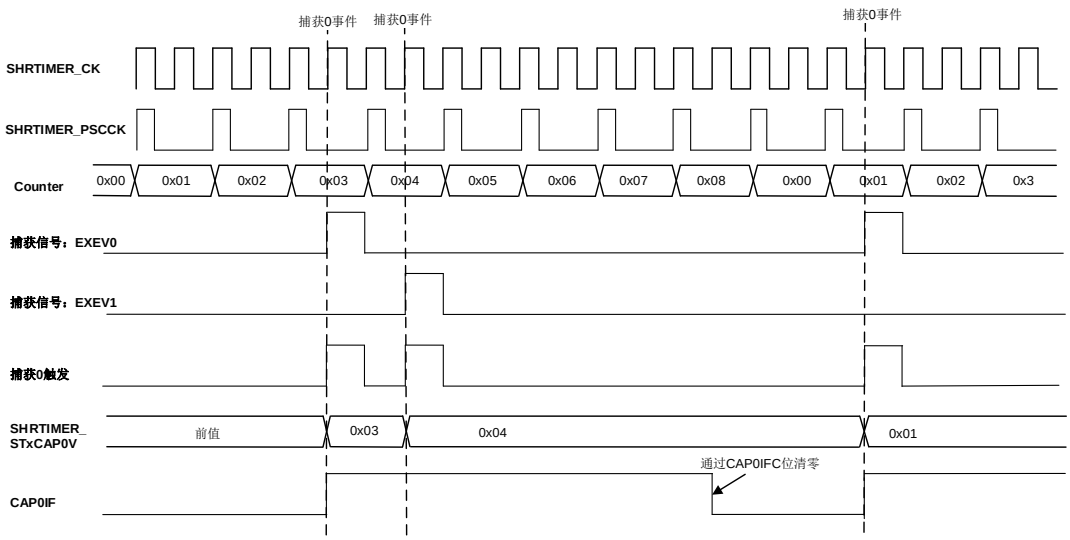
当选定的触发信号发生时，计数器的当前值被捕获到 SHRTIMER_STxCAPyV（y = 0,1）寄存器中。同时，SHRTIMER_STxINTF 寄存器中的 CAPyIF（y = 0,1）位置 1，如果 SHRTIMER_STxDMAINTEN 寄存器中的 CAPyIE（y = 0,1）= 1 或 CAPyDEN（y = 0,1）= 1，则生成相应的捕获中断和 DMA 请求。可以通过写 1 到 SHRTIMER_STxINTFC 寄存器中的 CAPyIFC 位来清除捕获中断标志位 CAPyIF。

捕获 0 触发事件在 SHRTIMER_STxCAP0TRG 寄存器中定义，捕获 1 触发事件在 SHRTIMER_STxCAP1TRG 寄存器中定义。当选择了多个触发事件时，所有的触发事件是逻辑“或”运算的。

注意：如果将外部事件配置为具有电平有效，则只能在 SHRTIMER_STxCAP0TRG 和 SHRTIMER_STxCAP1TRG 寄存器中使能一个外部事件。

捕获溢出是无法防止的，即使先前的捕获值未读取或捕获标志未清除，新的捕获仍将被触发，并且新的捕获值将覆盖先前的值。请参考*图 19-11. EXEV0 和 EXEV1 触发捕获 0*。

图 19-11. EXEV0 和 EXEV1 触发捕获 0

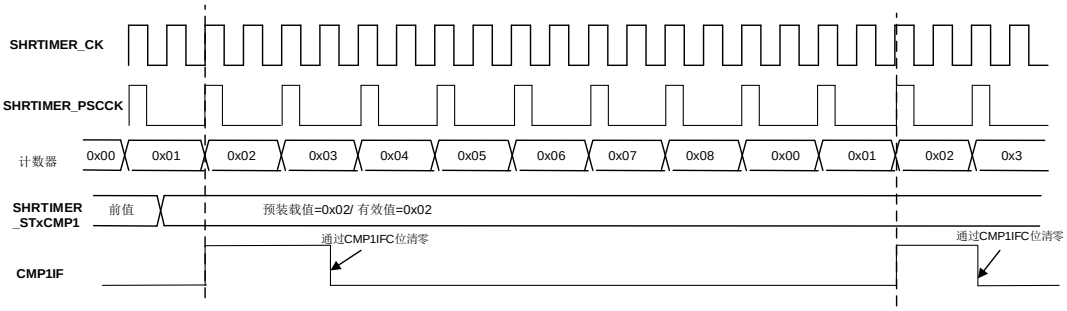


比较

Slave_TIMERx 单元有四个比较寄存器：SHRTIMER_STxCMPyV（y = 0..3）。当计数器值与比较寄存器值匹配时，将产生一个比较事件。具体请见*图 19-12. STxCAR=0x8, STxCMP1V=0x02 时，比较 1 寄存器的动作*。

比较事件会将相应的比较中断标志位置 1（SHRTIMER_STxINTF 中的 CMPyIF 位，其中 y = 0..3），如果使能了比较中断或 DMA 请求（SHRTIMER_STxDMAINTEN 寄存器中的 CMPyIE = 1 或 CMPyDEN = 1，其中 y = 0..3），则会生成比较中断或 DMA 请求。通过写 1 到 SHRTIMER_STxINTFC 中的 CMPyIF 位可以清除比较中断标志。

图 19-12. STxCAR=0x8, STxCMP1V=0x02 时，比较 1 寄存器的动作



半波模式

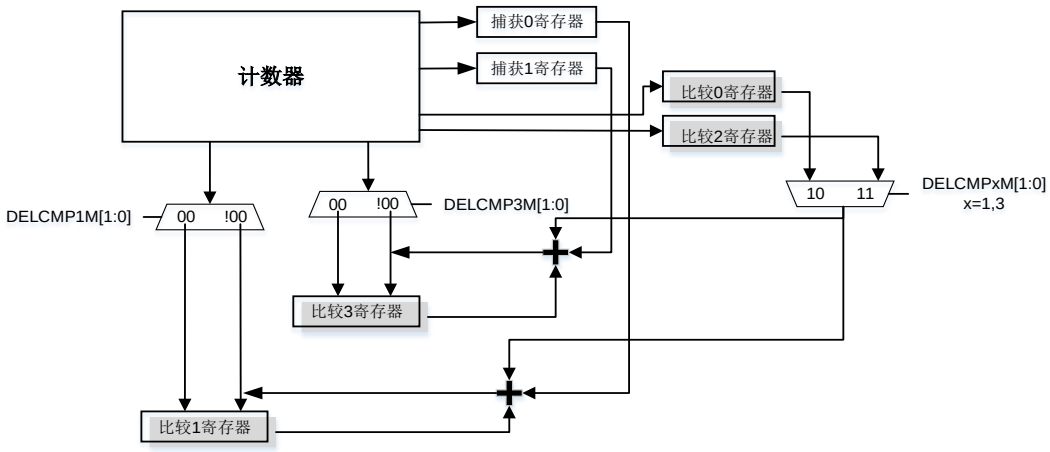
当 SHRTIMER_MTCTL0 中的 HALFM 位置 1 时，半波模式使能。此模式将比较 0 有效寄存器的值强制为计数器重载值的一半，但 SHRTIMER_MTCMP0V 寄存器的值不会更新为（SHRTIMER_MTCAR / 2）的值。半波模式主要用于生成固定占空比为 50%的方波。

当 SHRTIMER_MTCTL0 寄存器中的 SHWEN 位置 1 时，将使能影子寄存器，比较 0 有效寄存器的值在更新事件时刷新。反之，比较 0 有效寄存器在新值写入 SHRTIMER_MTCAR 后立即刷新。

比较延迟模式

此模式仅用于比较 y（y = 1,3）寄存器，并由 SHRTIMER_STxCTL0 寄存器中的 DELCMPyM[1:0] 位域控制。比较寄存器与计数器比较的实际值是重新计算值，该值在捕获 0/1 触发或比较 0/ 2 事件之后重新计算得到，具体如*图 19-13. 比较延迟模式框图*。此模式允许通过硬件将生成的波形与捕获触发同步。

图 19-13. 比较延迟模式框图



在延迟模式下，比较 y（y = 1，3）事件从相应的捕获/比较事件发生到周期事件期间有效。当计数器达到周期值时，将禁用比较 y（y = 1，3）事件，直到出现新的捕获/比较事件。

当没有捕获触发或比较事件发生时，不生成比较 y 事件。捕获触发事件发生后，将比较 y 有效寄存器中的值与对应的 SHRTIMER_STxCAP0V 或 SHRTIMER_STxCAP1V 寄存器值相加，然后将其与计数器进行比较。比较 1 寄存器与捕获 0 寄存器和比较 0 寄存器/比较 2 寄存器关联，而比较 3 寄存器与捕获 1 寄存器和比较 0 寄存器/比较 2 寄存器关联。

注意：重新计算的值被传输到一个内部寄存器，且无法读取。

SHRTIMER_STxCTL0 寄存器中的 DELCMP1M[1:0]位域（比较 1 事件）和 DELCMP3M[1:0]位域（比较 3 事件）可用于配置延迟模式。下面以 DELCMP1M[1:0]为例：

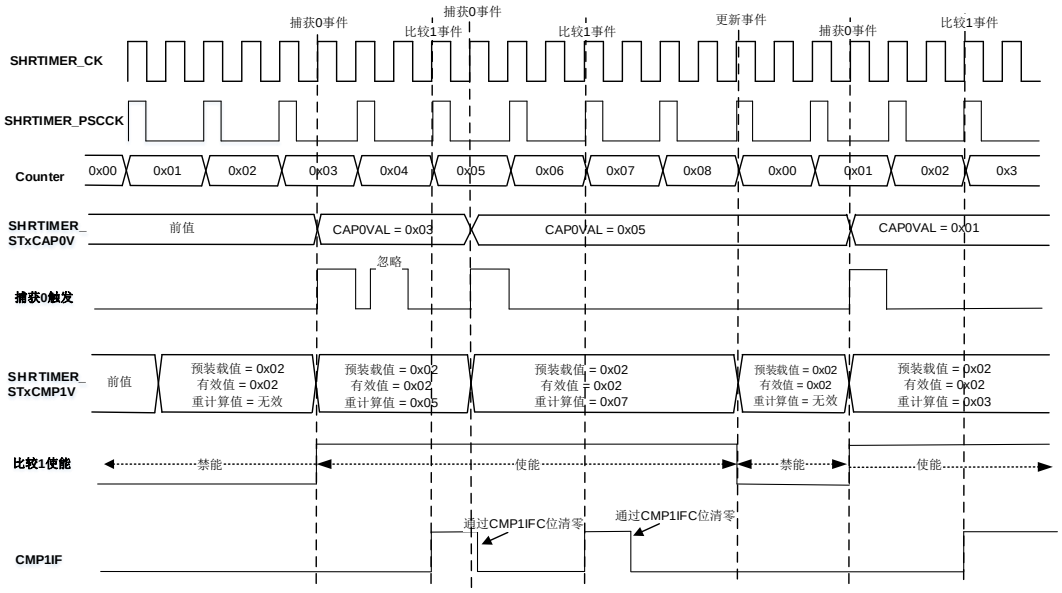
- 2’b00，比较1延迟模式禁能

比较 1 延迟模式禁用。一旦计数器值等于比较 1 寄存器的值，就会发生比较匹配。参见*图 19-12. STxCAR=0x8, STxCMP1V=0x02 时，比较 1 寄存器的动作*。

- 2’b01，比较1延迟模式0

捕获 0 事件发生后，将重新计算比较 1 寄存器的值（比较 1 有效寄存器值+捕获 0 寄存器值）。一旦计数器值等于重新计算后的比较 1 寄存器值，就会发生比较 1 事件。参见*图 19-14. 比较 1 延迟模式 0*。

图 19-14. 比较 1 延迟模式 0



■ 2’b10，比较1延迟模式1

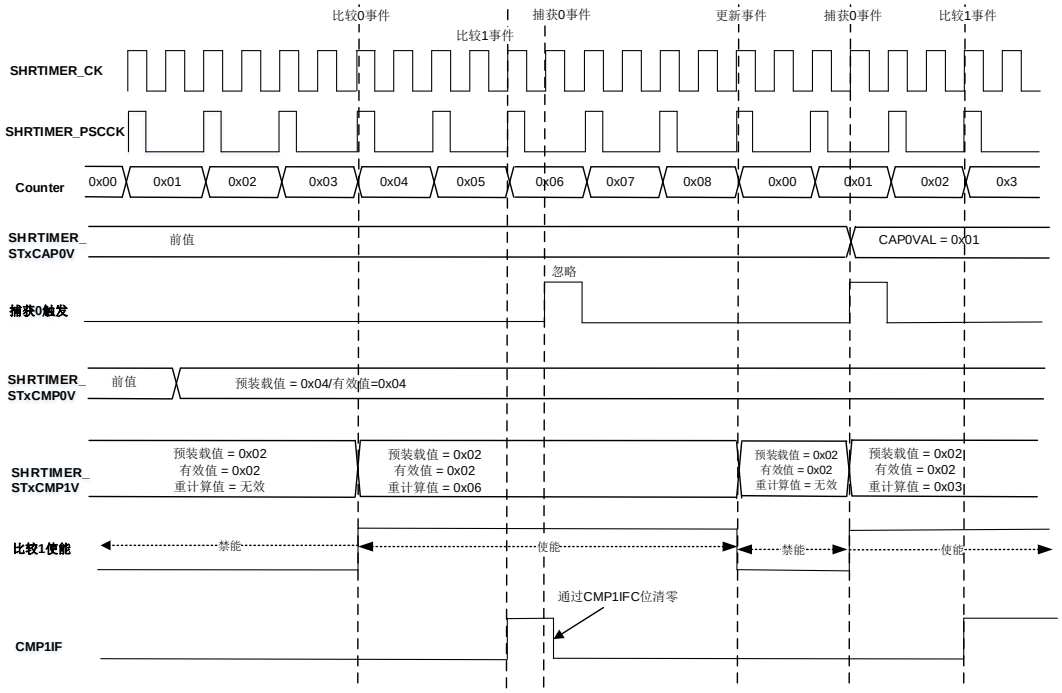
在捕获 0 事件或比较 0 事件之后，将重新计算比较 1 寄存器的值。

发生捕获 0 事件时，比较 1 寄存器的重新计算值 = 比较 1 有效寄存器值 + 捕获 0 事件的捕获值。

发生比较 0 事件时，比较 1 寄存器的重新计算值 = 比较 1 有效寄存器值 + 比较 0 有效寄存器值。

一旦计数器值等于重新计算的比较 1 寄存器值，就会发生比较 1 事件。如果捕获 0 事件先发生，则比较 0 事件将被忽略。同样，如果先发生比较 0 事件，则将忽略捕获 0 事件。详情请见 *图 19-15. 比较 1 延迟模式 1*。

图 19-15. 比较 1 延迟模式 1



■ 2’b11，比较1延迟模式2

该模式与比较 1 延迟模式 1 相同。在捕获 0 事件或比较 2 事件之后，重新计算比较 1 寄存器的值。

发生捕获 0 事件时，比较 1 寄存器的重新计算值 = 比较 1 有效寄存器值 + 捕获 0 事件的捕获值。

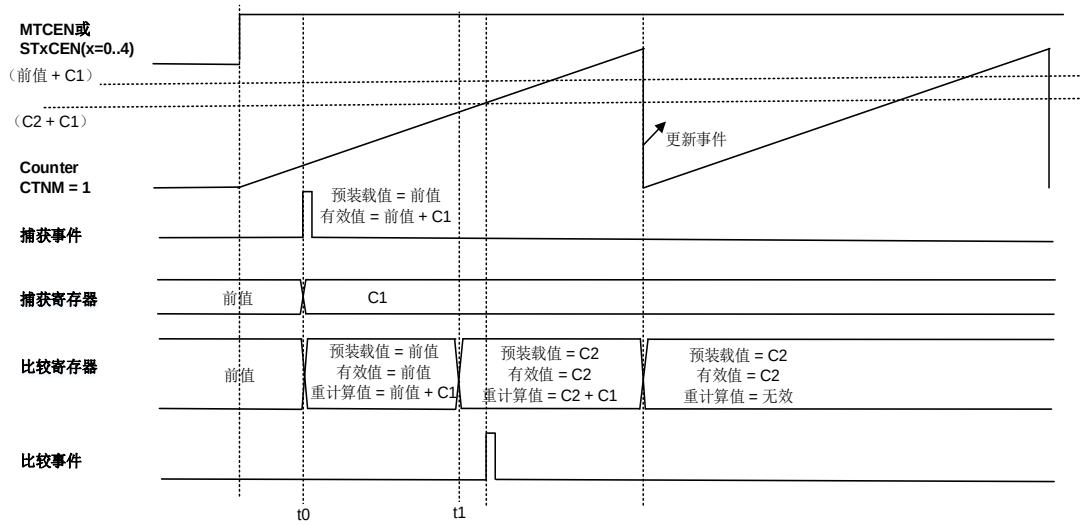
发生比较 2 事件时，比较 1 寄存器的重新计算值 = 比较 1 有效寄存器值 + 比较 2 有效寄存器值。

一旦计数器值等于重新计算的比较 1 寄存器值，就会发生比较 1 事件。如果捕获 0 事件先发生，则比较 2 事件将被忽略。同样，如果先发生比较 2 事件，则将忽略捕获 0 事件。详见*图 19-15. 比较 1 延迟模式 1*。

影子寄存器（SHWEN = 0）禁能时，即使在发生捕获事件后修改 SHRTIMER_STxCMP0V 位或 SHRTIMER_STxCMP2V 位的值，新的比较值也会立即被带入有效寄存器。*图 19-16.比较延迟模式（SHWEN=0）*显示了一个示例：

在 t0 处发生捕获事件，C1 值被捕获到寄存器中，重新计算的值 = 比较有效寄存器的值 + C1。在 t1 处将新的比较值（C2）写入比较寄存器，则重新计算的值 = C2 + C1。

图 19-16.比较延迟模式（SHWEN=0）



使用延迟模式（DELCMP3M [1:0] = 01，10,11），可以防止捕获溢出发生。在同一个计数周期（由 SHRTIMER_STxCAR 确定）内，仅考虑第一个捕获事件。新的捕获事件在以下三种情况下有效：

- 当比较寄存器的重新计算值与计数器值匹配时；
- 发生了周期事件；
- 计数器复位。

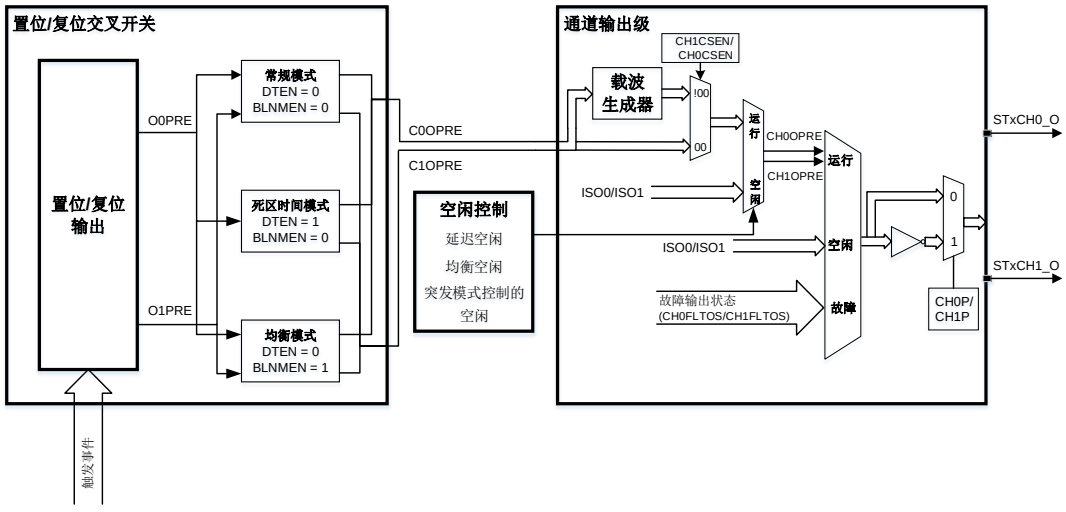
置位/复位交叉开关

通道输出波形功能由三个部分实现：

- 置位/复位交叉开关
- 空闲控制
- 通道输出级

*图 19-17. 通道输出结构图*显示了这三个部分的结构。

图 19-17. 通道输出结构图



交叉开关模块有三种输出模式：常规模式，死区时间模式和均衡模式，输出时只能选择其中一种模式。

输出准备信号

Slave_TIMERx 有一个置位/复位输出模块。该模块可以生成两个输出准备信号：O0PRE 和 O1PRE。其中，O0PRE 由 SHRTIMER_STxCH0SET 和 SHRTIMER_STxCH0RST 寄存器控制。O1PRE 由 SHRTIMER_STxCH1SET 和 SHRTIMER_STxCH1RST 寄存器控制。OyPRE（y = 0,1）的高电平为有效电平，低电平为无效电平。

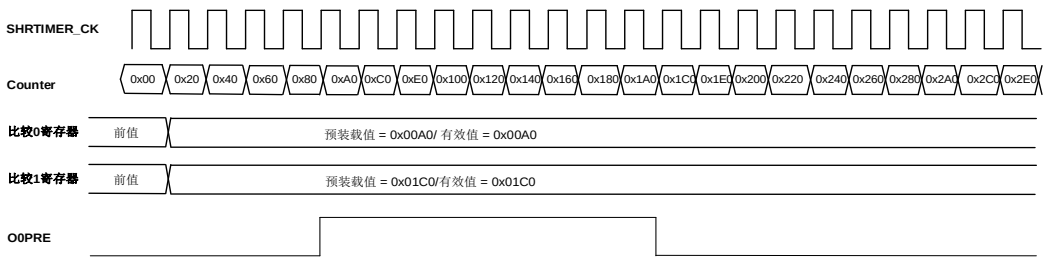
当 SHRTIMER_STxCHySET 寄存器中配置的事件发生时，此模块将产生置位请求，并使 OyPRE 输出高电平。当 SHRTIMER_STxCHyRST 寄存器中配置的事件发生时，此模块会产生复位请求，并使 OyPRE 输出低电平。如果在 SHRTIMER_STxCHySET 和 SHRTIMER_STxCHyRST 寄存器中配置了相同的事件，则此模块会生成输出翻转请求，并在配置的事件发生时使 OyPRE 输出翻转。

注意：如果 SHRTIMER_STxCTL0 和 SHRTIMER_STxACTL 寄存器中的 CNTCKDIV[3:0]位域等于 4’b0110 或 4’b0111，则不得同时设置 SHRTIMER_STxCH0SET 和 SHRTIMER_STxCH0RST 寄存器中的同一事件。

*图19-18. O0PRE波形：CMP0事件置位，CMP1事件复位*显示了以下配置时的O0PRE输出波形：

- SHRTIMER_STxCTL0寄存器中的CNTCKDIV [2:0] = 3’b000；
- SHRTIMER_STxCH0SET= 0x0000 0008：比较0事件产生置位请求，O0PRE将输出高电平；
- SHRTIMER_STxCH0RST= 0x0000 0010：比较1事件产生复位请求，O0PRE将输出低电平；
- SHRTIMER_STxCMP0V = 0x00A0；
- SHRTIMER_STxCMP1V = 0x01C0。

图 19-18. O0PRE 波形：CMP0 事件置位，CMP1 事件复位



OyPRE（y = 0,1）最多可以选择 32 个事件：

- Slave_TIMERx：更新事件，复位事件，周期事件和比较y（y = 0..3）事件；
- Master_TIMER：周期事件，比较y（y = 0..3）事件；
- Slave_TIMERx互连事件：其他Slave_TIMERy有9个互连事件（例如x = 1，然后y = 0，2..4）,参见*表19-5. Slave_TIMER内部连接事件*;
- 外部事件y（y = 0..9）：EXEVy为Slave_TIMERx中的外部事件的滤波信号；
- 软件事件。

无论 STxCEN 位是否为 1，软件事件始终有效。但是，只有 STxCEN 为 1 时，才会考虑其他事件。

表 19-5. Slave_TIMER 内部连接事件

内部连接	来自 ST0				来自 ST1				来自 ST2				来自 ST3				来自 ST4			
	CMP0	CMP1	CMP2	CMP3	CMP0	CMP1	CMP2	CMP3	CMP0	CMP1	CMP2	CMP3	CMP0	CMP1	CMP2	CMP3	CMP0	CMP1	CMP2	CMP3
到 ST0	×	×	×	×	0	1	×	2	×	3	×	×	5	6	×	×	×	×	7	8
到 ST1	0	×	×	×	×	×	×	×	×	×	3	4	×	×	5	6	7	×	×	×
到 ST2	×	×	1	×	×	2	3	×	×	×	×	×	×	4	×	5	×	6	7	8
到 ST3	0	×	×	1	×	2	×	3	4	×	5	6	×	×	×	×	7	×	×	8
到 ST4	×	×	0	1	×	×	2	3	4	5	×	×	6	7	×	8	×	×	×	×

注意：（1）表中的数字表示 Slave_TIMERx 互连事件。

（2）“x”代表无效。

可以同时选择多个事件源（进行逻辑或运算），并且当它们在同一 t_{HPTMER_CK} 周期内发生时，将执行仲裁。

仲裁机制

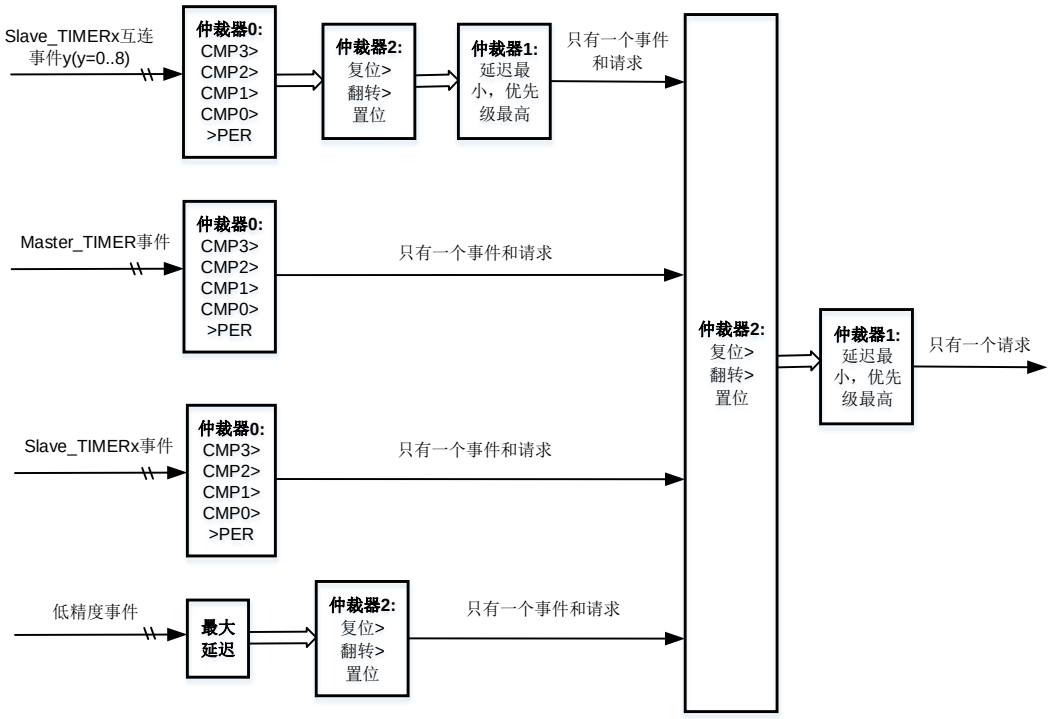
当 SHRTIMER_STxCH1SET 和 SHRTIMER_STxCH1RST 寄存器中配置的多个事件发生在同一个 t_{HPTMER_CK} 周期内时，将执行仲裁过程，且只有一个事件有效，可以更改 OyPRE（y = 0,1）的输出。

这 32 个事件可以分为四种类型：

- Slave_TIMERx：比较y（y = 0..3）事件，周期事件。
- Master_TIMER：比较y（y = 0..3）事件，周期事件。
- Slave_TIMERx互连事件：互连事件y（y = 0..8）
- 低精度事件：Slave_TIMERx的更新事件和复位事件，外部事件y（y = 0..9），软件事件。

具体的仲裁过程*图 19-19. 每个 tSHRTMER_CK 周期的仲裁机制过程*所示。

图 19-19. 每个 $t_{SHRTMER_CK}$ 周期的仲裁机制过程



三个仲裁器的功能如下：

■ 仲裁器 0 的优先级顺序（从最高优先级到最低优先级）：

比较 3 事件 > 比较 2 事件 > 比较 1 事件 > 比较 0 事件 > 周期事件。

■ 仲裁器 1 根据事件在 t_{HPTMER_CK} 期间的延迟来仲裁优先级：

延迟越小，优先级越高。

■ 仲裁器 2 根据事件对 OyPRE（y = 0,1）输出的影响来仲裁优先级：

复位请求 > 输出翻转请求 > 置位请求。

以 Slave_TIMER0 中的 O0PRE 输出为例，配置如下：

■ SHRTIMER_STxCH0SET = 0x0060 5898，选定产生置位请求的事件是：

Master_TIMER：比较 2 事件，周期事件；

Slave_TIMER0：比较 1 事件，比较 0 事件；

Slave_TIMER0 的互连事件：互连事件 0（Slave_TIMER1 比较 0 事件），互连事件 2（Slave_TIMER1 比较 3 事件）；

低精度事件：外部事件 0（EXEV0），外部事件 1（EXEV1）。

■ SHRTIMER_STxCH0RST = 0x0198 0344，选定产生置位请求的事件是：

Master_TIMER：比较 0 事件，比较 1 事件；

Slave_TIMER0：比较 3 事件，周期事件；

Slave_TIMER0 的互连事件：互连事件 7（Slave_TIMER4 比较 2 事件），互连事件 8

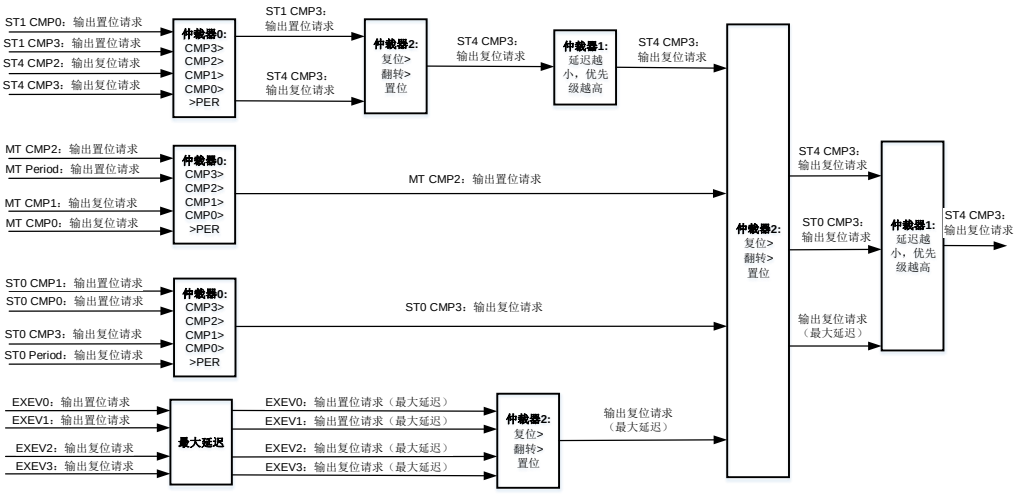
（Slave_TIMER4 比较 3 事件）；

低精度事件：外部事件 2（EXEV2），外部事件 3（EXEV3）。

- 延迟：Slave_TIMER4 比较 3 < Slave_TIMER0 比较 3

上述选定的事件如果在同一个 $t_{SHRTMER_CK}$ 周期内发生，则仲裁过程和结果如*图 19-20. 仲裁机制示例*所示。最终，Slave_TIMER4 比较 3 事件产生的复位请求在该 $t_{SHRTMER_CK}$ 期间有效，并且 O0PRE 将被设置为低电平。

图 19-20. 仲裁机制示例



输出准备信号：窄脉冲管理

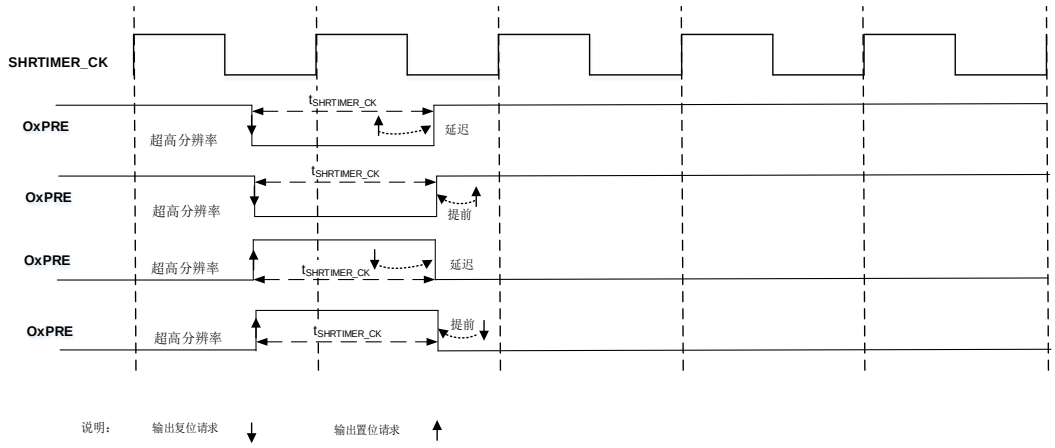
当几个输出置位和/或复位请求在 3 个连续的 $t_{SHRTMER_CK}$ 周期内发生时，OyPRE（y = 0,1）输出信号是一个窄脉冲。窄脉冲的输出管理由 SHRTIMER_STxCTL0 寄存器中的 CNTCKDIV[3:0] 位域配置，有以下两种情形：

- 情形0：CNTCKDIV[3:0] < 4’b0101或CNTCKDIV[3:0] = 4’b1000
- 情形1：CNTCKDIV[3:0] >= 4’b0101，CNTCKDIV[3] = 0

情形 0：CNTCKDIV[3:0] < 4’b0101 或 CNTCKDIV[3:0] = 4’b1000

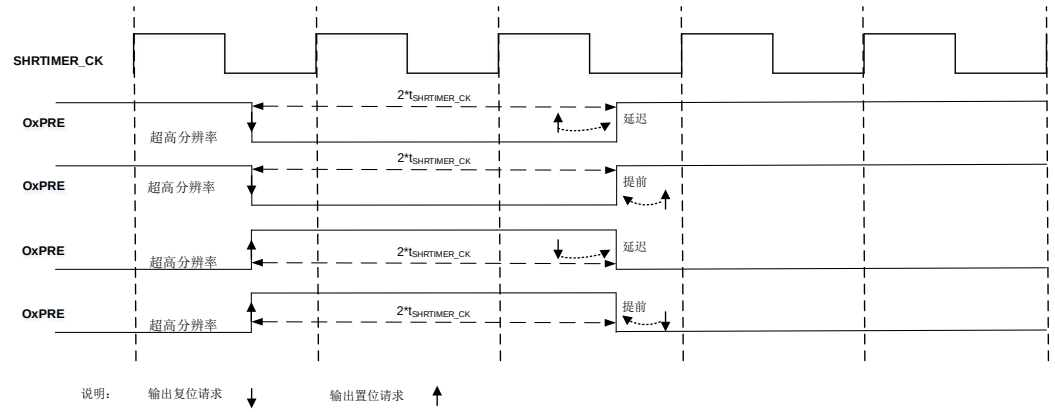
如果输出置位和复位请求在两个连续的 $t_{SHRTMER_CK}$ 周期内产生，则会生成脉宽为 1 个 t_{HPTMER_CK} 周期的脉冲。具体如*图 19-21. 脉冲宽度为 1 个 tSHRTMER CK 周期*所示。

图 19-21. 脉冲宽度为 1 个 $t_{SHRTMER_CK}$ 周期



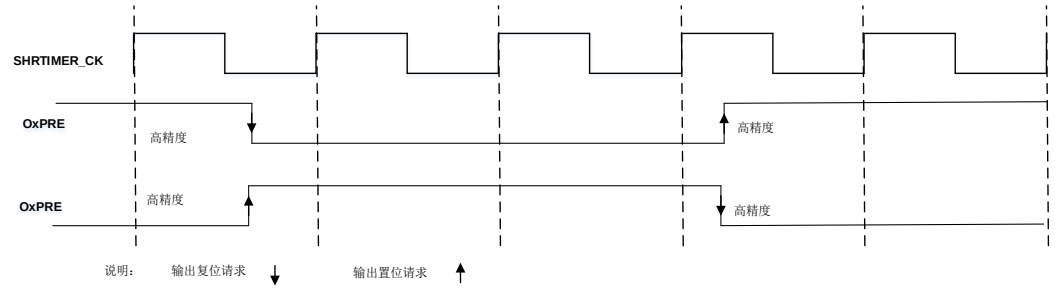
如果输出置位和复位请求的时间间隔包括一个完整的 $t_{SHRTMER_CK}$ 周期，则会生成脉宽为 2 个 $t_{SHRTMER_CK}$ 周期的脉冲。具体如*图 19-22.脉冲宽度为 2 个 tSHRTMER_CK 周期*所示。

图 19-22.脉冲宽度为 2 个 $t_{SHRTMER_CK}$ 周期



如果输出置位和复位请求的时间间隔大于两个完整的 t_{HPTMER_CK} 周期，则需要使用超高分辨率时钟。具体如*图 19-23. 超高分辨率 OxPRE 波形*所示。

图 19-23. 超高分辨率 OxPRE 波形

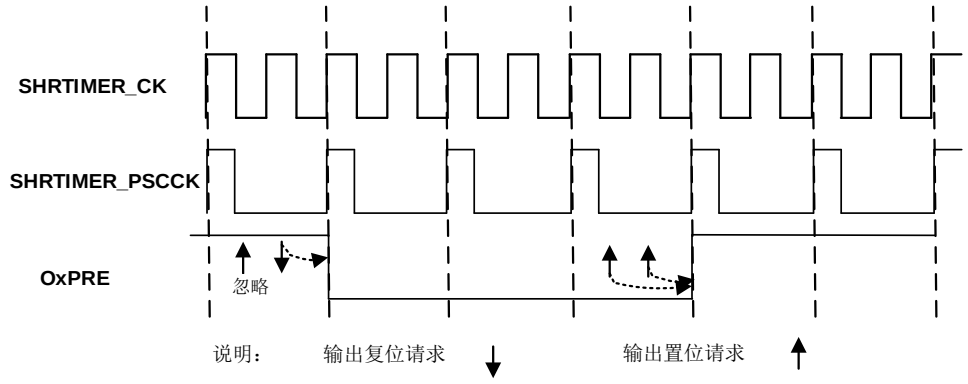


情形 1: CNTCKDIV[3:0] >= 4’b0101，CNTCKDIV[3] = 0

这种情况下，即使在每个 $t_{SHRTIMER_CK}$ 周期内执行仲裁，发生在 1 个 SHRTIMER_PSCCK 周期内的输出置位或复位请求，都会延迟到 SHRTIMER_PSCCK 时钟的下一个有效边沿。

当来自不同事件源的置位请求和复位请求在 1 个 $t_{SHRTIMER_CK}$ 周期中同时发生时，复位请求具有最高优先级，详见*图 19-24. CNTCKDIV[3:0] = 4’b0110 时的 OxPRE 波形*。

图 19-24. CNTCKDIV[3:0] = 4’b0110 时的 OxPRE 波形



常规模式

当 SHRTIMER_STxCHOCTL 寄存器中的 DTEN = 0，SHRTIMER_STxCTL0 寄存器中的 BLNMEN = 0 时，置位/复位交叉开关以常规模式运行。

该模式中，C0OPRE 和 C1OPRE 是独立的。C0OPRE（C1OPRE）直接连接到 O0PRE（O1PRE）。

当 Slave_TIMERx（x = 0..4）运行在 RUN 或 IDLE 状态下，C0OPRE 从无效状态变为有效状态时，SHRTIMER_STxINTF 寄存器中的 CH0OAIF 位将被设置为 1。如果使能了中断或 DMA 请求（SHRTIMER_STxDMAINTEN 寄存器中的 CH0OAIE = 1 或 CH0OADEN = 1），则产生相应的中断或 DMA 请求。通过将 1 写到 SHRTIMER_STxINTFC 中的 CH0OAIFC 位可以清除 CH0OAIF 中断标志。

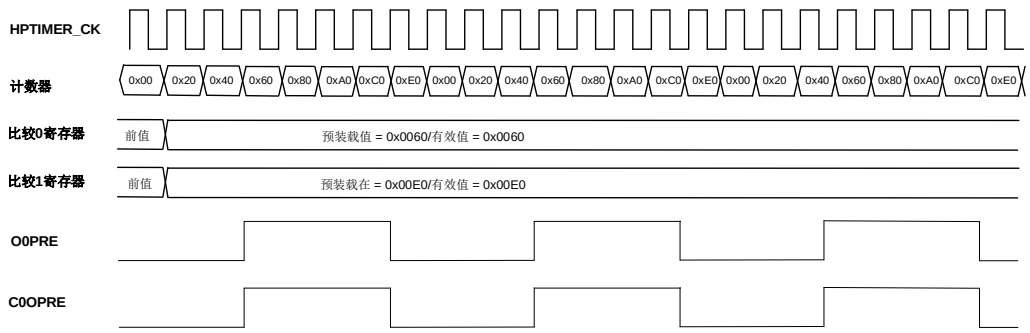
当 Slave_TIMERx（x = 0..4）运行在 RUN 或 IDLE 状态下，C0OPRE 从有效状态变为无效状态时，SHRTIMER_STxINTF 寄存器中的 CH0ONAIF 位将置 1，如果使能了中断或 DMA 请求（SHRTIMER_STxDMAINTEN 寄存器中的 CH0ONAIE = 1 或 CH0ONADEN = 1），则产生相应的中断或 DMA 请求。通过将 1 写到 SHRTIMER_STxINTFC 中的 CH0ONAIFC 位可以清除 CH0ONAIF 中断标志。

通道 1 与通道 0 输出情况相同。

*图 19-25. 常规模式下的 C0OPRE 波形*显示了以下配置时的 C0OPRE 波形：

- SHRTIMER_STxCTL0寄存器中的CNTCKDIV[3:0] = 4’b0000；
- SHRTIMER_STxCH0SET = 0x0000 0008：比较0事件产生置位请求，O0PRE输出高电平；
- SHRTIMER_STxCH0RST = 0x0000 0010：比较1事件产生复位请求，O0PRE输出低电平；
- SHRTIMER_STxCMP0V = 0x0060；
- SHRTIMER_STxCMP1V = 0x00E0。

图 19-25. 常规模式下的 C0OPRE 波形



死区时间模式

当 SHRTIMER_STxCHOCTL 寄存器中的 DTEN = 1，SHRTIMER_STxCTL0 寄存器中的 BLNMEN = 0 时，置位/复位交叉开关在死区模式下运行。

死区模式中，只对 O0PRE 进行编程，以驱动 C0OPRE 和 C1OPRE 的输出。C0OPRE 和 C1OPRE 是一对互补信号，在有效状态转换之间插入可编程的死区时间。

死区时间值是由 SHRTIMER_STxDTCTL 寄存器中的 DTFCFG [15:0]位域和 DTRCFG [15:0]位域确定的。DTFCFG [15:0]位域定义在 O0PRE 下降沿之后的死区时间，而 DTRCFG [15:0]位域定义在 O0PRE 上升沿之后的死区时间。

注意：DTFCFG [8:0]和 DTRCFG [8:0]位域在 SHRTIMER_STxDTCTL 寄存器中，DTFCFG [15:9]和 DTRCFG [15:9]位域在 SHRTIMER_STxACTL 寄存器中

死区时间值可以由 SHRTIMER_STxDTCTL 寄存器中的 DTRS 位和 DTFS 位配置为正或负。当需要某些波重叠时，可以定义负的死区时间值。

死区时间由 SHRTIMER_STxDTCTL 寄存器中的 DTGCKDIV [3:0]位域定义的时钟确定。

注意：DTGCKDIV[2:0]位域在 SHRTIMER_STxDTCTL 寄存器中，而 DTGCKDIV[3]位在 SHRTIMER_STxACTL 寄存器中。

当 Slave_TIMERx（x = 0..4）运行在 RUN 或 IDLE 状态下，C0OPRE 从无效状态变为有效状态时，SHRTIMER_STxINTF 寄存器中的 CH0OAIF 位将被设置为 1。如果使能了中断或 DMA 请求（SHRTIMER_STxDMAINTEN 寄存器中的 CH0OAIE = 1 或 CH0OADEN = 1），则产生相应的中断或 DMA 请求。通过将 1 写到 SHRTIMER_STxINTFC 中的 CH0OAIFC 位可以清除 CH0OAIF 中断标志。

当 Slave_TIMERx（x = 0..4）运行在 RUN 或 IDLE 状态下，C0OPRE 从有效状态变为无效状态时，SHRTIMER_STxINTF 寄存器中的 CH0ONAIF 位将置 1，如果使能了中断或 DMA 请求（SHRTIMER_STxDMAINTEN 寄存器中的 CH0ONAIE = 1 或 CH0ONADEN = 1），则产生相应的中断或 DMA 请求。通过将 1 写到 SHRTIMER_STxINTFC 中的 CH0ONAIFC 位可以清除 CH0ONAIF 中断标志。

通道 1 与通道 0 输出情况相同。

*图 19-26. 具有死区时间的 C0OPRE 和 C1OPRE 互补输出波形*显示了 O0PRE 脉冲宽度大于死区时间时的 C0OPRE 和 C1OPRE 波形。

图 19-26. 具有死区时间的 C0OPRE 和 C1OPRE 互补输出波形

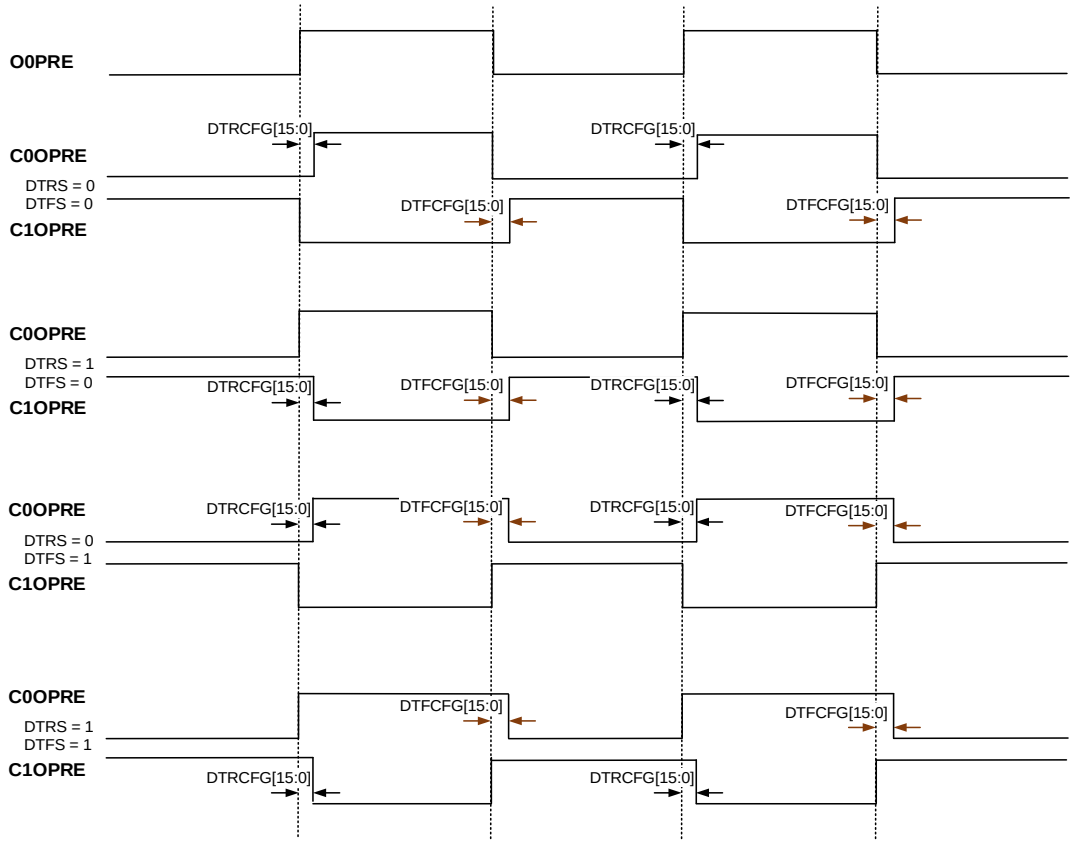
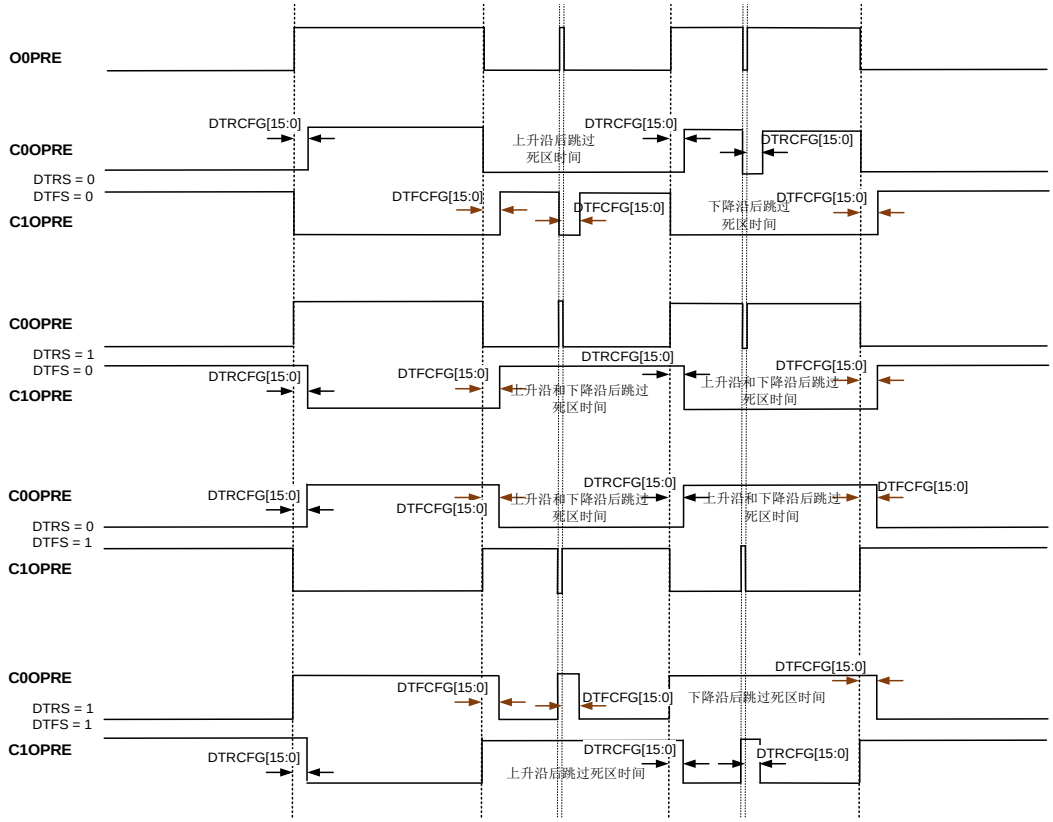


*图 19-27. 脉宽小于死区时间时的互补波形*显示了 O0PRE 脉冲宽度小于死区时间时的 C0OPRE 和 C1OPRE 互补波形。

图 19-27. 脉宽小于死区时间时的互补波形

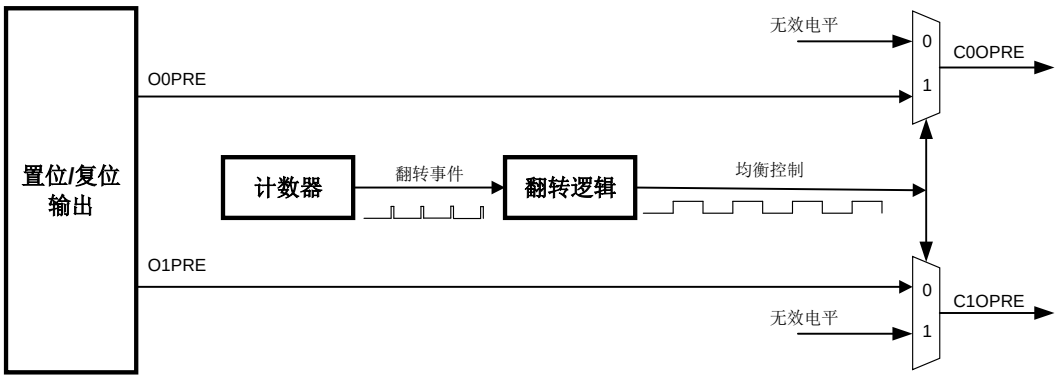


均衡模式

当 SHRTIMER_STxCHOCTL 寄存器中的 DTEN = 0，SHRTIMER_STxCTL0 寄存器中的 BLNMEN = 1 时，置位/复位交叉开关在均衡模式下运行。当计数器在连续模式下运行时才能使用均衡模式，并且一旦使能了计数器就不得使其复位。

*图 19-28. 均衡模式的结构框图*显示了均衡模式的信号控制过程。

图 19-28. 均衡模式的结构框图



一旦接收到翻转事件，翻转逻辑模块的输出就会翻转。当翻转逻辑模块的输出为 1（高电平）时，C0OPRE 连接到 O0PRE，C1OPRE 为无效电平（低电平）。当翻转逻辑模块的输出为 0（低电平）时，C1OPRE 连接到 O1PRE，C0OPRE 为无效电平（低电平）。

建议配置 SHRTIMER_STxCH0SET = SHRTIMER_STxCH1SET，SHRTIMER_STxCH0RST = SHRTIMER_STxCH1RST，实现相同波形的均衡操作。在进行其他应用时，也可以对两个输

出进行不同的配置。

均衡模式禁能时，SHRTIMER_STxINTF 寄存器中的 CBLNF 位将复位，该位用于指示当前哪个通道正在输出信号（O0PRE 或 O1PRE）。

当 Slave_TIMERx（x = 0..4）运行在 RUN 或 IDLE 状态下，C0OPRE 从无效状态变为有效状态时，SHRTIMER_STxINTF 寄存器中的 CH0OAIF 位将被设置为 1。如果使能了中断或 DMA 请求（SHRTIMER_STxDMAINTEN 寄存器中的 CH0OAIE = 1 或 CH0OADEN = 1），则产生相应的中断或 DMA 请求。通过将 1 写到 SHRTIMER_STxINTFC 中的 CH0OAIFC 位可以清除 CH0OAIF 中断标志。

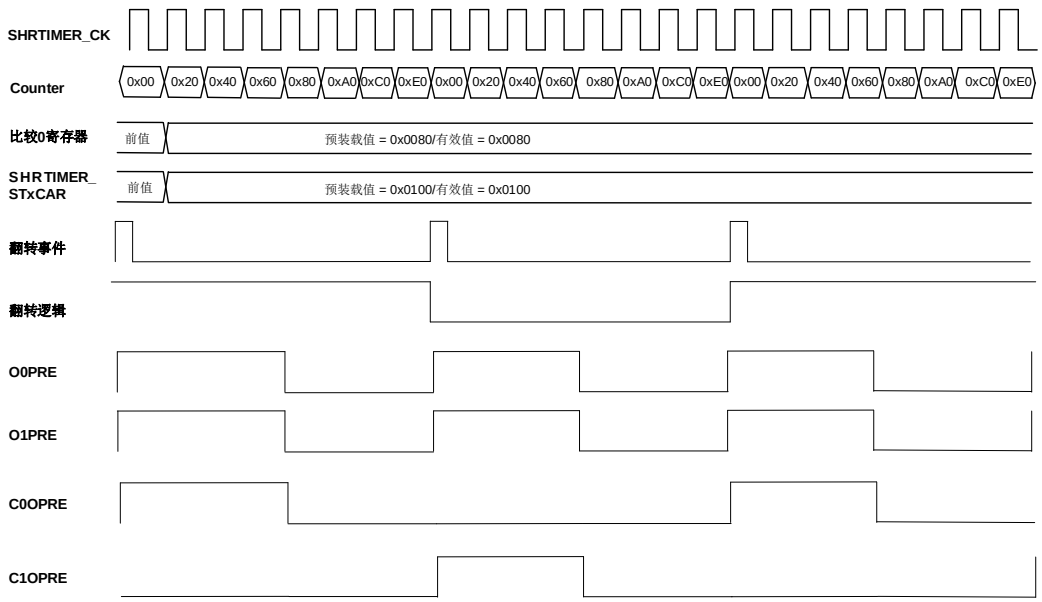
当 Slave_TIMERx（x = 0..4）运行在 RUN 或 IDLE 状态下，C0OPRE 从有效状态变为无效状态时，SHRTIMER_STxINTF 寄存器中的 CH0ONAIF 位将置 1，如果使能了中断或 DMA 请求（SHRTIMER_STxDMAINTEN 寄存器中的 CH0ONAIE = 1 或 CH0ONADEN = 1），则产生相应的中断或 DMA 请求。通过将 1 写到 SHRTIMER_STxINTFC 中的 CH0ONAIFC 位可以清除 CH0ONAIF 中断标志。

通道 1 与通道 0 输出情况相同。

*图 19-29. 均衡模式下的 C0OPRE 和 C1OPRE 波形*显示了以下配置时的 C0OPRE 和 C1OPRE 波形：

- SHRTIMER_STxCH0SET = SHRTIMER_STxCH1SET = 0x0000 0004：周期事件产生置位请求，O0PRE和O1PRE输出高电平；
- SHRTIMER_STxCH0RST = SHRTIMER_STxCH1RST = 0x0000 0008：比较0事件产生复位请求，O0PRE和O1PRE输出低电平。

图 19-29. 均衡模式下的 C0OPRE 和 C1OPRE 波形



空闲控制

空闲控制级有三种控制空闲状态的方式：

- 延迟空闲模式
- 均衡空闲模式
- 突发模式控制的空闲模式

延迟空闲和均衡空闲模式不能同时使用。均衡空闲仅在均衡模式下可用。延迟空闲或均衡空闲可以与突发模式控制的空闲模式同时使用，但突发模式的优先级最低。当置位/复位开关在不同模式下操作时，可以使用不同的空闲控制模式，详见*表 19-6. 交叉开关和空闲控制同时运行*。

表 19-6. 交叉开关和空闲控制同时运行

置位/复位交叉开关运行模式	IDLE 控制级运行模式
常规模式	延迟空闲，突发模式控制的空闲
死区时间模式	延迟空闲，突发模式控制的空闲
均衡模式	延迟空闲，均衡控制和突发模式控制空闲

SHRTIMER_STxINTF 寄存器中的 CHyF(y=0,1)位，指示了 CHyOPRE 的输出电平。

延迟空闲

在延迟空闲模式，所选外部事件（对于 Slave_TIMER0/ 1/ 2 为 EXEV5/ 6，对于 Slave_TIMER3/ 4 为 EXEV7/ 8）之后的置位请求或复位请求会导致 CHyOPRE（y = 0，1）输出进入空闲状态。具体情况与 SHRTIMER_STxCHOCTL 寄存器中的 ISOy/ CHyP（y = 0,1）有关，详见*表 19-7. 进入和退出空闲状态的请求*。ISOy 位用于定义空闲状态时 CHyOPRE 的输出电平。空闲模式会永久保持，计数器将继续运行，直到重新使能输出才退出该模式。将 STxCH0EN 和 STxCH1EN 位重新置 1 后，通过置位请求或复位请求可以重新使能延迟空闲模式。

表 19-7. 进入和退出空闲状态的请求

ISOy/CHyP (y=0,1)值	进入空闲状态的请求	退出空闲状态的请求
ISOy = 0 CHyP = 0	复位请求	置位请求和复位请求
ISOy = 1 CHyP = 0	置位请求	置位请求和复位请求
ISOy = 0 CHyP = 1	置位请求	置位请求和复位请求
ISOy = 1 CHyP = 1	复位请求	置位请求和复位请求

延迟空闲模式可以应用于单个输出（CHyOPRE）或两个输出（CH0OPRE 和 CH1OPRE）情况（由 SHRTIMER_STxCHOCTL 寄存器中的 DLYISCH[2:0]位域定义）：

- DLYISCH[2:0] = 3’b000：延迟空闲模式应用于CH0OPRE；
- DLYISCH[2:0] = 3’b001：延迟空闲模式应用于CH1OPRE；
- DLYISCH[2:0] = 3’b010：延迟空闲模式应用于CH0OPRE和CH1OPRE。

一旦选定的外部事件（EXEV5/ 6 或 EXEV7/ 8）到达，SHRTIMER_STxINTF 寄存器中的 DLYIIF 位置 1，如果使能了相应的中断和 DMA（SHRTIMER_STxDMAINTEN 寄存器中的 DLYIIE = 1 或 DLYIDEN = 1），则产生中断或 DMA 请求。通过写 1 到 SHRTIMER_STxINTFC 寄存器中的 DLYIIFC 位可以清除中断标志。

当选定的外部事件（EXEV5 / 6 或 EXEV7/ 8）触发延迟空闲模式时，SHRTIMER_STxINTF 寄

存器中的 CHyDLYF（y = 0,1）位可以指示 CHyOPRE 信号的状态。

下面四张图显示了延迟空闲模式中的 CH0OPRE 波形：

- C0OPRE 运行在常规模式：SHRTIMER_STxCHOCTL 寄存器中的 DTEN = 0，SHRTIMER_STxCTL0寄存器中的BLNMEN = 0；
- 比较0事件产生置位请求；
- 比较1事件产生复位请求。

图 19-30. 延迟空闲模式，ISO0 = 0 和 CHOP = 0

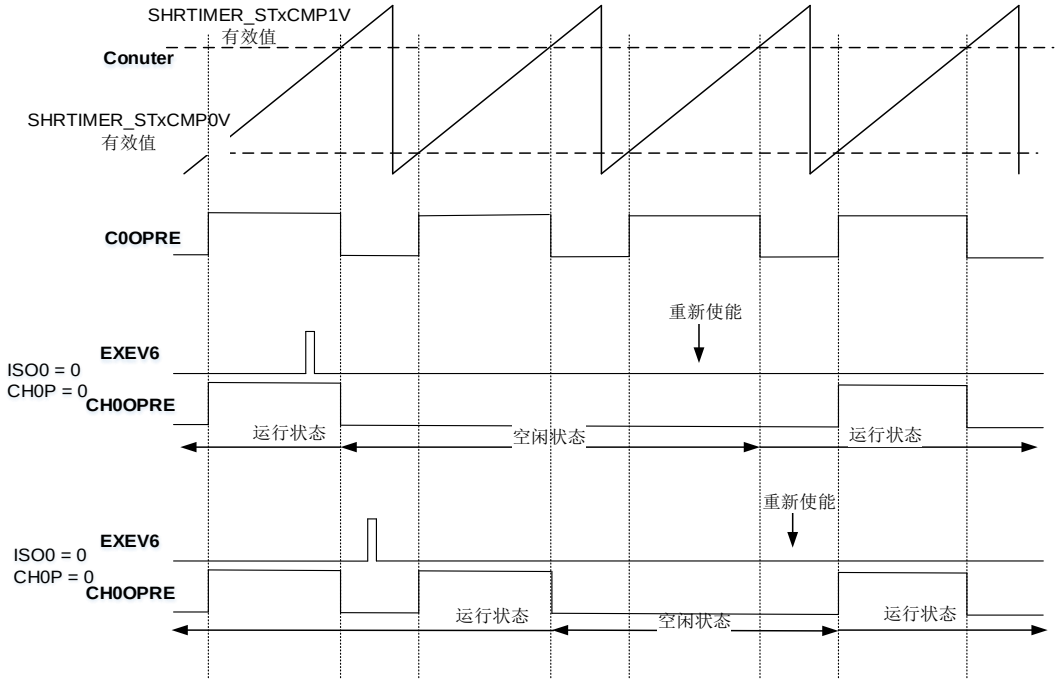


图 19-31. 延迟空闲模式，ISO0 = 1 和 CHOP = 0

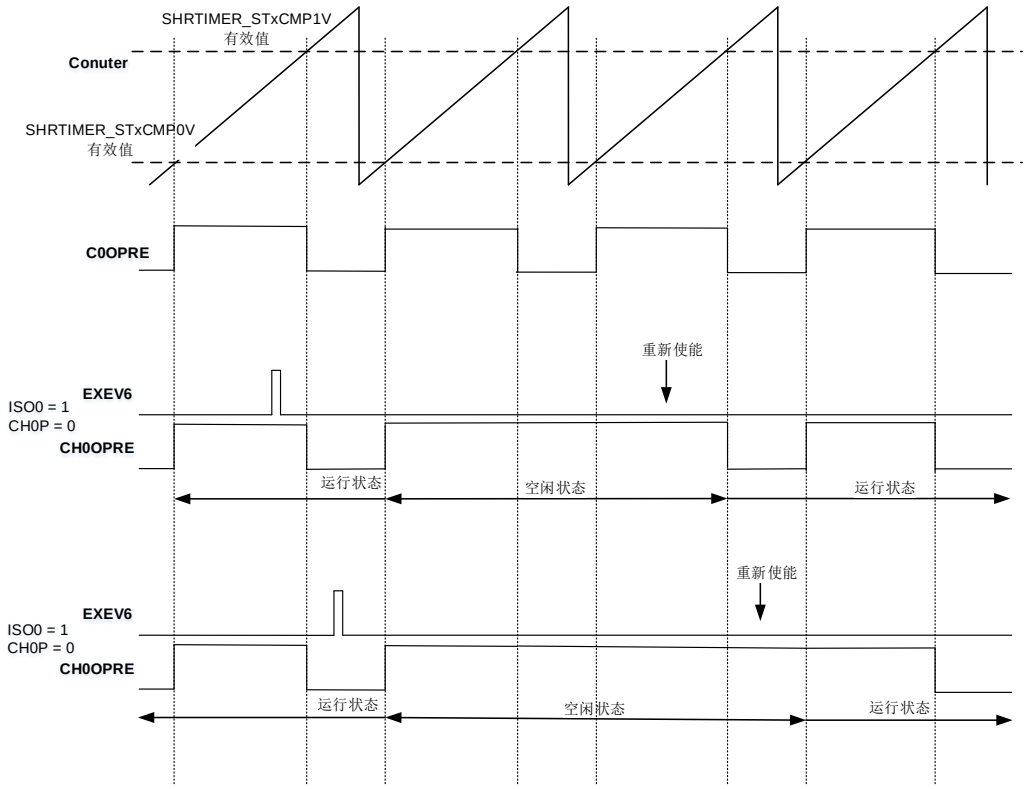


图 19-32. 延迟空闲模式，ISO0 = 0 和 CHOP = 1

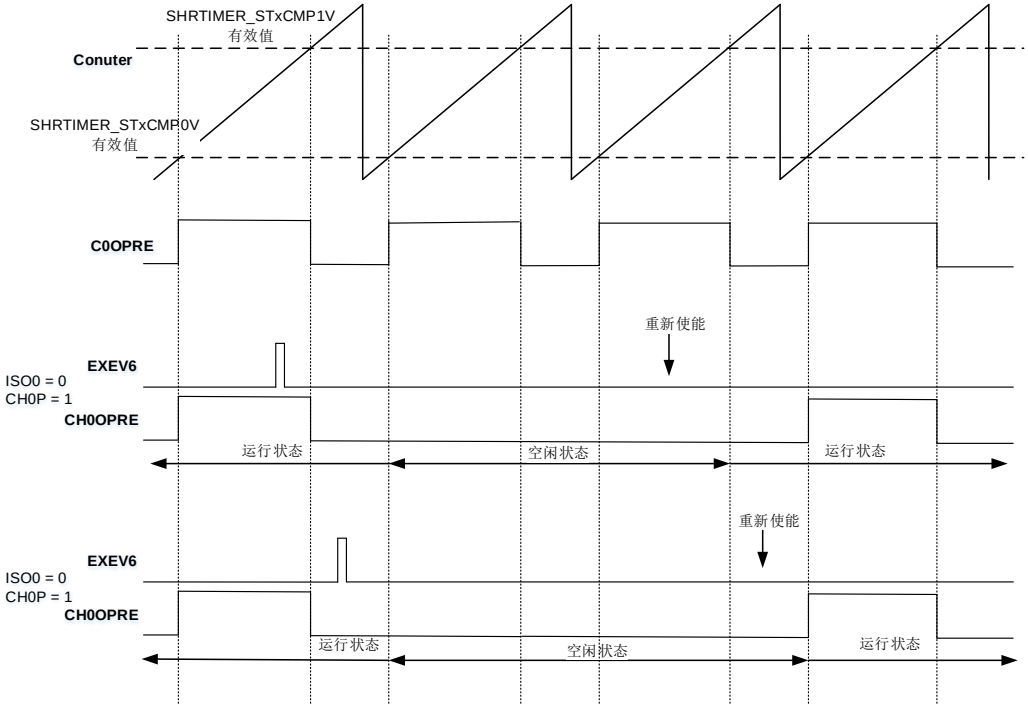
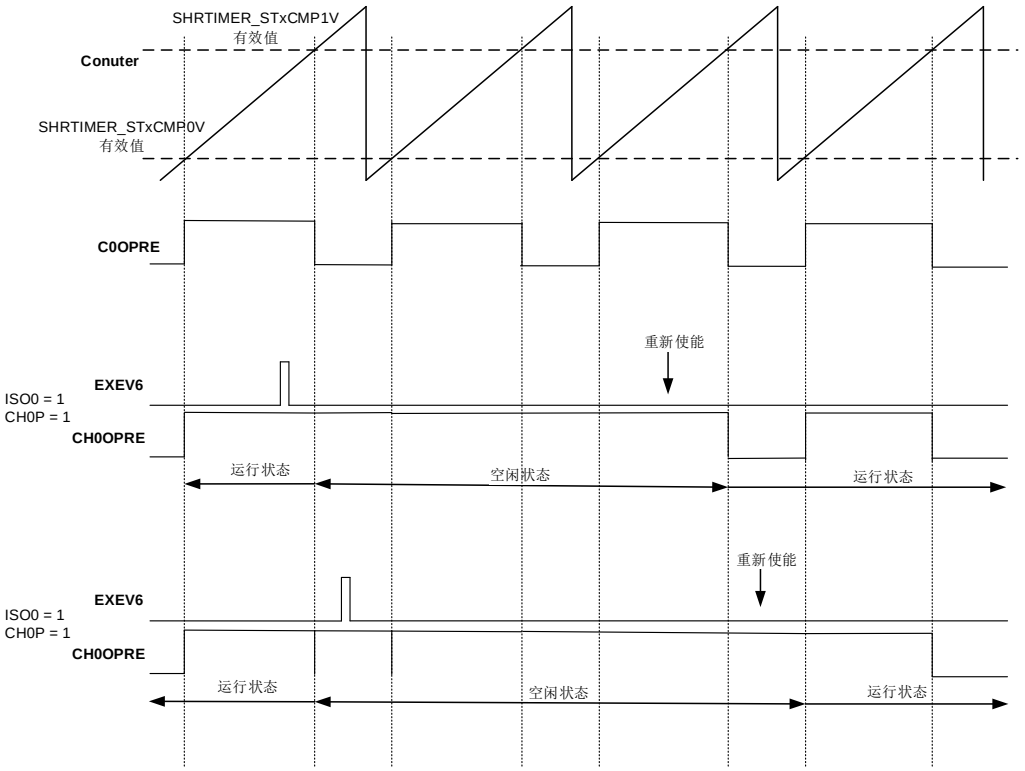


图 19-33. 延迟空闲模式，ISO0 = 1 和 CHOP = 1



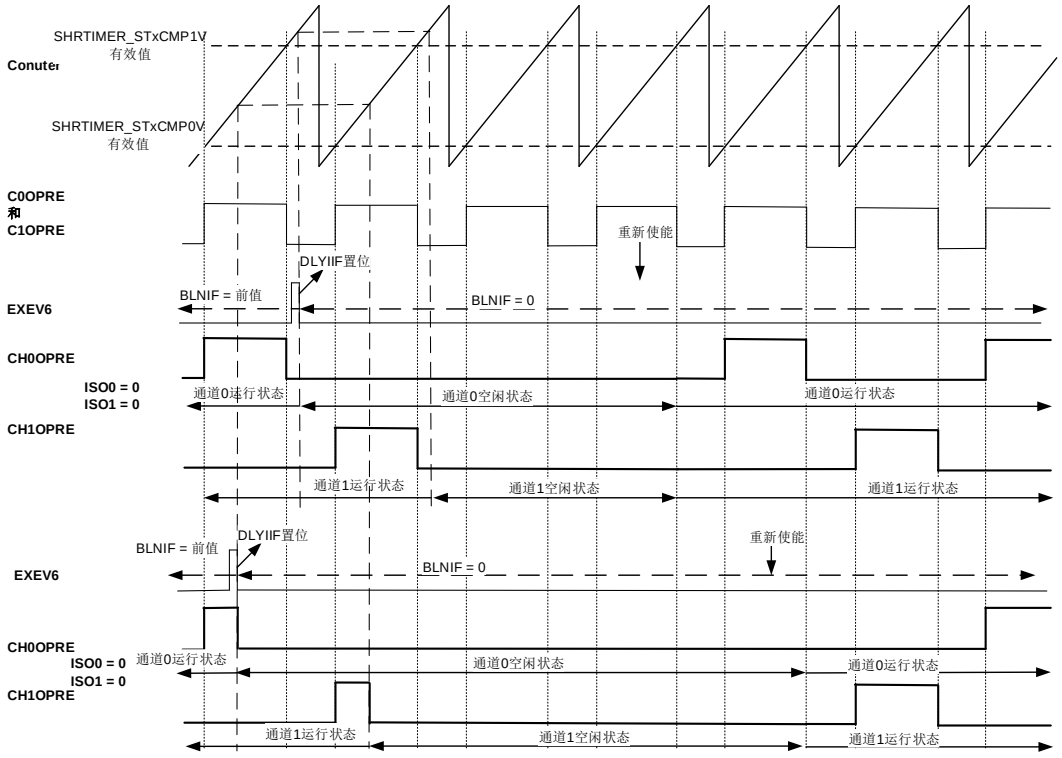
均衡空闲

均衡空闲模式仅在均衡模式下可用。通过将 SHRTIMER_STxCHOCTL 寄存器的 DLYISCH[2:0] 位域设为 3’bx11，使能均衡空闲模式。Slave_TIMER0/1/2 的外部事件 5/6（EXEV5/6）和 Slave_TIMER3/4 的外部事件 7/8（EXEV7/8）可使用均衡空闲模式。

外部事件发生时，CHyOPRE（y = 0,1）进入空闲状态，并输出由 SHRTIMER_STxCHOCTL 寄存器中的 ISOy 位定义的电平，且 SHRTIMER_STxINTF 寄存器中的 DLYIIF 位置 1。该外部事件触发捕获，将计数器值捕获到比较 3 有效寄存器中（该值用户不可访问）。均衡模式会再维持一个周期，使互补输出 CHzOPRE（z = 0，1 且 z≠y）可以重复 CHyOPRE 上的短脉冲：*图 19-34. 均衡空闲模式，ISO0 = 0 和 ISO1 = 0* 显示了均衡空闲模式下 Slave_TIMER0 的 CH0OPRE/ CH1OPRE 波形，配置如下：

- C0OPRE 处于均衡模式：SHRTIMER_STxCHOCTL 寄存器中的 DTEN = 0，SHRTIMER_STxCTL0寄存器中的BLNMEN = 1；
- 比较0事件产生置位请求；
- 比较1事件产生复位请求；
- 在外部事件6发生时，通道0和通道1输出为均衡空闲模式：Slave_TIMER0的DLYISCH [2：0] = 111。

图 19-34. 均衡空闲模式，ISO0 = 0 和 ISO1 = 0



SHRTIMER_STxINTF 寄存器中的 BLNIF 位指示了发生均衡空闲模式时哪个通道正在输出信号。如*图 19-34. 均衡空闲模式，ISO0 = 0 和 ISO1 = 0*，外部事件 6（EXEV6）到来时，通道 0 输出信号，通道 1 输出无效，且 BLNIF 位复位为零。

在计数器继续运行时，IDLE 模式将永久保持，直到重新使能输出才退出均衡空闲模式。将 STxCH0EN 和 STxCH1EN 位重新置 1 后，通过置位请求或复位请求可以重新使能均衡空闲模式。

在以下情况下，均衡空闲模式可以与突发模式一起使用：

- BMSTx位必须复位（保持计数器时钟SHRTIMER_PSCCK，且计数器正常运行）；
- 当输出配置为突发模式控制的空闲状态时，不会触发均衡空闲模式。

突发模式控制的空闲模式

在突发模式中，空闲状态由突发模式控制器控制。具体请参考*突发模式*。

均衡空闲和延迟空闲的优先级高于突发模式：一旦触发均衡空闲和延迟空闲模式，任何突发模式的退出请求都将被丢弃。相反，如果在均衡空闲或延迟空闲退出时，突发模式仍有效，则突发模式将正常恢复。

突发模式控制器可以对任意两个输出 CHyOPRE（y = 0,1）进行控制。见*表 19-8. 突发模式控制的空闲状态时的输出*，SHRTIMER_STxCHOCTL 寄存器中的 ISOy 位和 BMCHyIEN（y = 0,1）位可以对突发模式控制的空闲模式期间的每个输出的状态进行配置。

表 19-8. 突发模式控制的空闲状态时的输出

ISOy	BMCHyIEN	CHyOPRE (y=0,1)
x	0	无影响：输出不受突发模式控制器影响

ISOy	BMCHyIEN	CHyOPRE (y=0,1)
0	1	在突发模式控制的空闲模式中输出复位电平
1	1	在突发模式控制的空闲模式中输出置位电平

通道输出级

每个 Slave_TIMERx 可以控制一对输出（STxCH0_O 和 STxCH1_O）。输出级由三种工作状态：

- 运行状态：STxCHy_O（y = 0,1）输出CHyOPRE（y = 0,1）的电平。
- 空闲状态：STxCHy_O（y = 0,1）输出由SHRTIMER_STxCHOCTL寄存器中ISOy位定义的电平。
- 故障状态：STxCHy_O（y = 0,1）可以永久有效，无效或Hi-Z（由SHRTIMER_STxCHOCTL寄存器的CHyFLTOS位定义）。详见*故障输入*。

SHRTIMER_CHOUTEN 寄存器中的 STxCHyEN 位和 SHRTIMER_CHOUTDISF 寄存器中的 STxCHyDISF 位可以指示输出的状态，如*表 19-9. 输出级状态编程（x=0..4, y=0,1）*所述。

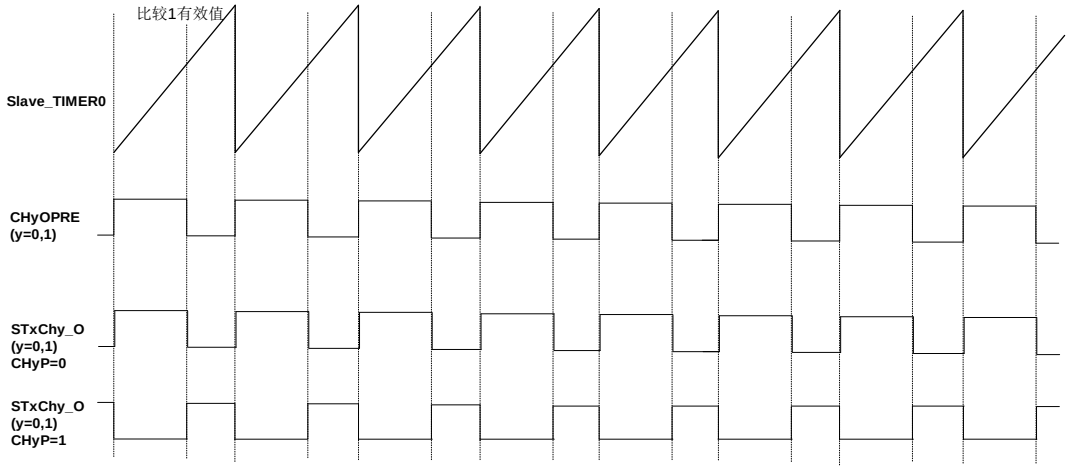
表 19-9. 输出级状态编程（x=0..4, y=0,1）

STxCHyEN	STxCHyDISF	输出级状态
1	x	运行状态
0	0	空闲状态
0	1	故障状态

将 SHRTIMER_CHOUTDIS 寄存器中的 STxCHyDIS 位置 1，输出禁能，并使输出进入空闲状态。三种工作状态的优先级顺序为：空闲状态 > 故障状态 > 运行状态。

SHRTIMER_STxCHOCTL 寄存器中的 CHyP 位可以设置输出极性。当 CHyP = 0 时，输出极性为高电平有效。当 CHyP = 1 时，输出极性为低电平有效。详见*图 19-35. CHyP=0 或 CHyP=1 时的 STxCHy_O 波形*。

图 19-35. CHyP=0 或 CHyP=1 时的 STxCHy_O 波形



使用 SHRTIMER_STxCHOCTL 寄存器中的 CHyFLTOS [1:0]位域可以配置故障状态下的输出电平，如下所示：

- 2’b00：输出永远不会进入故障状态，并保持运行或空闲状态；
- 2’b01：故障状态时，输出有效电平；
- 2’b10：故障状态时，输出无效电平；
- 2’b11：故障状态时，输出为三态。

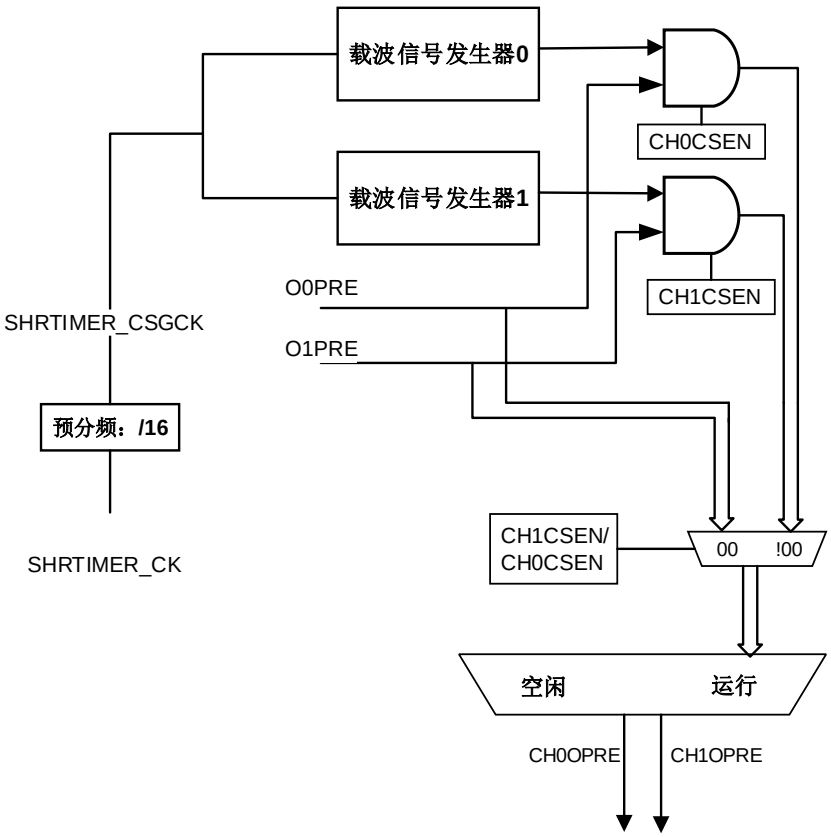
使用 SHRTIMER_STxCHOCTL 寄存器中的 ISOy 位配置处于空闲状态的输出电平，如下：

- 2’b0：空闲状态时，输出无效电平；
- 2’b1：空闲状态时，输出有效电平。

载波信号模式

可以在 OyPRE(y=0,1)信号顶部添加一个高频载波信号，如*图 19-36. 载波信号结构框图*所示。

图 19-36. 载波信号结构框图



载波信号模式中，可以在载波信号开始之前定义一个特定的脉冲宽度。载波信号的频率和占空比是可配置的。详见*图 19-37. 载波模式使能时的 SHRTIMER 输出*。

将 SHRTIMER_STxCHOCTL 寄存器中的 CH0CSEN 和 CH1CSEN 位置 1，可以分别在通道 0 和 1 上使能载波信号模式。

第一个脉冲的脉冲宽度由 SHRTIMER_STxCSCTL 寄存器中的 CSFSTPW [3:0]位域配置，公式如下：

$$t_{CSFSTPW}=(CSFSTPW[3:0]+1)* t_{SHRTIMER_CSGCK}，其中t_{SHRTIMER_CSGCK}=16 *t_{SHRTIMER_CK}$$

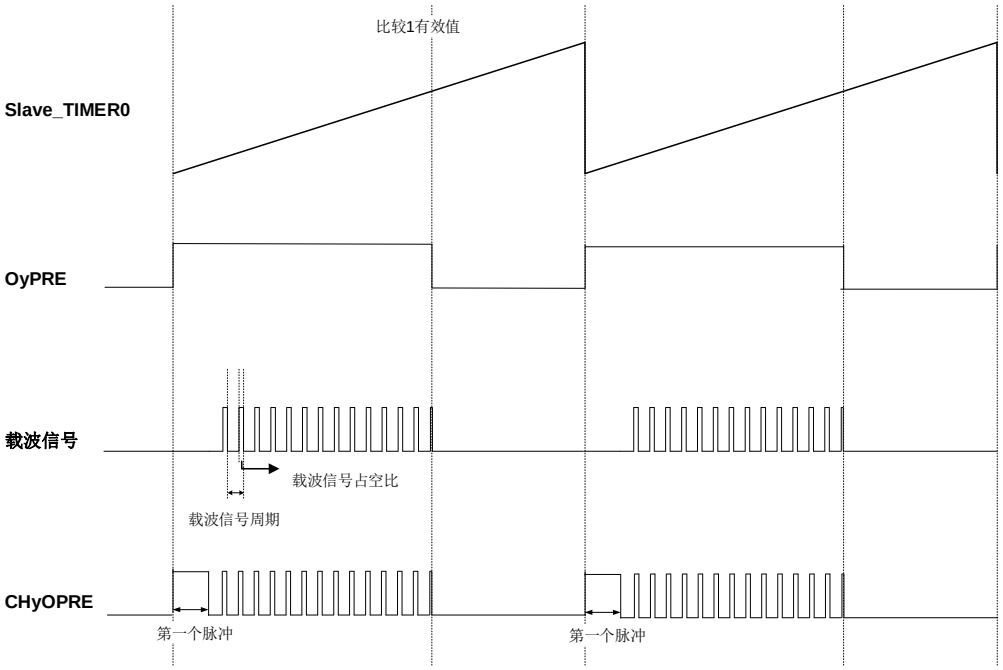
载波信号的频率由 SHRTIMER_STxCSCTL 寄存器中的 CSPRD [3:0]位域配置，公式如下：

t_{CSPRD}=(CSPRD[3:0]+1)* $t_{SHRTIMER_CSGCK}$，其中$t_{SHRTIMER_CSGCK}$=16 *$t_{SHRTIMER_CK}$

载波信号的占空比由 SHRTIMER_STxCSCTL 寄存器中的 CSDTY[2:0]位域配置，步长为 12.5%。

载波信号模式中，载波信号发生器的输出与 OyPRE 逻辑与运算后输出。OyPRE（y = 0,1）输出无效时，载波信号会立即停止，即使当前的载波周期未完成。具体请参考*图 19-37. 载波模式使能时的 SHRTIMER 输出*。

图 19-37. 载波模式使能时的 SHRTIMER 输出



同步输入启动/复位计数器

当 SHRTIMER_MTCTL0 寄存器中的 SYNIRST 位置 1 时，同步输入可以产生计数器复位事件；当 SHRTIMER_MTCTL0 寄存器中的 SYNISTRT 位置 1 时，同步输入可以启动计数器。详见*同步输入*。

同步输入请求会将 SHRTIMER_MTINTF 寄存器中的 SYNIIF 位置 1，如果使能了中断或 DMA 请求（SHRTIMER_MTDMAINTEN 寄存器中的 SYNIIE = 1 或 SYNIDEN = 1），会产生相应的中断或 DMA 请求。可以通过写 1 到 SHRTIMER_MTINTFC 寄存器中的 SYNIIFC 位，来清除同步输入中断标志。

更新事件和影子寄存器

Slave_TIMERx 中的某些寄存器具有影子寄存器。MCU 复位后，影子寄存器被禁能。如果将 SHRTIMER_STxCTL0 寄存器中的 SHWEN 位清 0，则禁能影子寄存器。写入这些寄存器的值将立即转移到有效寄存器中并生效。

如果 SHRTIMER_STxCTL0 寄存器中的 SHWEN 位置 1，则使能影子寄存器，*表 19-10. Slave_TIMERx 影子寄存器和更新事件*中列出的寄存器被预加载。写入这些寄存器的值将被传

送到影子寄存器，且不会立即生效。当发生更新事件时，影子寄存器内容将转移到有效寄存器中并立即生效。

注意：当 SHWEN=1 时，才会产生更新事件。

*表 19-10. Slave_TIMERx 影子寄存器和更新事件*列出了具有影子寄存器的寄存器和相应的更新事件。

表 19-10. Slave_TIMERx 影子寄存器和更新事件

具有影子寄存器的寄存器	影子寄存器使能位	更新事件
SHRTIMER_STxDMAINTEN	SHRTIMER_STxCTL0寄存器中的SHWEN位	软件（STxSUP 位）
SHRTIMER_STxCAR		重复事件（UPREP = 1）
SHRTIMER_STxCREP		计数器复位或翻转事件（UPRST = 1）
SHRTIMER_STxCMP0V		来自其他定时器的更新事件（Slave_TIMERx 是 UPBSTX，Master_TIMER 是 UPBMT）
SHRTIMER_STxCMP0CP		DMA 模式结束事件（UPSEL[3:0] = 4'b0001）
SHRTIMER_STxCMP1V		
SHRTIMER_STxCMP2V		
SHRTIMER_STxCMP3V		DMA 模式结束事件之后的更新事件（UPSEL[3:0] = 4'b0010）
SHRTIMER_STxDTCTL		
SHRTIMER_STxCH0SET		
SHRTIMER_STxCH0RST		STxUPINy(y=0..2)的上升沿产生更新事件
SHRTIMER_STxCH1SET		
SHRTIMER_STxCH1RST		STxUPINy(y=0..2)的上升沿之后产生更新事件
SHRTIMER_STxCNTRST		
SHRTIMER_STxACTL 寄存器中的 DTFCFG[15:9]和 DTRCFG[15:9]		

更新使能输入 STxUPINy（y = 0..2）是来自通用定时器的芯片内部信号，上升沿有效。具体请见*表 19-11. STxUPINy（y=0..2）和芯片内部信号*。

表 19-11. STxUPINy（y=0..2）和芯片内部信号

更新使能输入 STxUPINy（y = 0..2）	芯片内部信号
STxUPIN0	保留
STxUPIN1	保留
STxUPIN2	TIMER5_TRGO

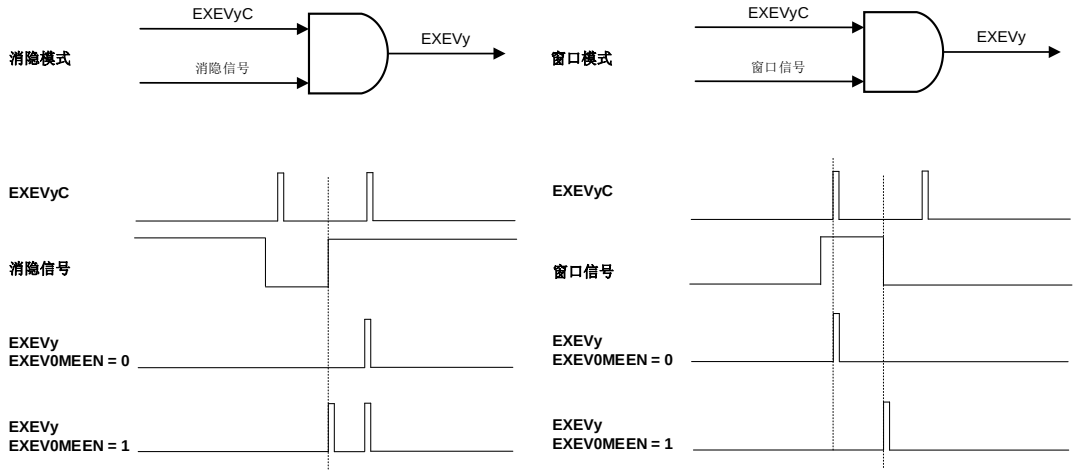
外部事件滤波

外部事件 EXEVyC（y = 0..9）在指定时间内被滤波，有两种滤波模式：

- 消隐模式：在指定时间内发生的外部事件被忽略；
- 窗口模式：在指定时间内发生的外部事件有效。

具体请见*图 19-38. 消隐模式和窗口模式*。

图 19-38. 消隐模式和窗口模式



参考*外部事件*章节可得更多关于外部事件 EXEVyC(y=0..9)的信息。

消隐模式

消隐模式中，在指定时间内发生的外部事件 EXEVyC（y = 0..9）被忽略，其他时间发生的外部事件有效。在指定的时间内，消隐信号为低电平。该模式由 EXEVyFM [3:0]位域配置，范围从 4’b0001 到 4’b1100。

消隐信号源有两种类型：

- Slave_TIMERx本身：指定的时间是指从计数器复位到比较事件发生持续的时间。（EXEVyFM [3:0] = 4’b0001~4’b0100，用于设置比较0~比较3事件）；
- 来自其他Slave_TIMER单元（EXEVyFM [3:0] = 4’b0101~4’b1100）的STBLKSRCz（z = 0..7）：指定的时间是指从选定的Slave_TIMER计数器复位到比较事件发生持续的时间。也可以是选定Slave_TIMER中的CH1OPRE信号（在这种情况下，只要CH1OPRE为低电平，事件将被忽略）具体请见*表19-12. 消隐模式下的滤波信号映射*。

表 19-12. 消隐模式下的滤波信号映射

到 / 来自	Slave_TIMER0	Slave_TIMER1	Slave_TIMER2	Slave_TIMER3	Slave_TIMER4
STBLKSRC0	Slave_TIMER1 比较 0	Slave_TIMER0 比较 0	Slave_TIMER0 比较 1	Slave_TIMER0 比较 0	Slave_TIMER0 比较 1
STBLKSRC1	Slave_TIMER1 比较 3	Slave_TIMER0 比较 3	Slave_TIMER1 比较 0	Slave_TIMER1 比较 1	Slave_TIMER1 比较 0
STBLKSRC2	Slave_TIMER1 CH1OPRE	Slave_TIMER0 CH1OPRE	Slave_TIMER1 比较 3	Slave_TIMER2 比较 0	Slave_TIMER2 比较 0
STBLKSRC3	Slave_TIMER2 比较 0	Slave_TIMER2 比较 0	Slave_TIMER1 CH1OPRE	Slave_TIMER2 比较 1	Slave_TIMER2 比较 3
STBLKSRC4	Slave_TIMER2 比较 3	Slave_TIMER2 比较 1	Slave_TIMER3 比较 0	Slave_TIMER2 CH1OPRE	Slave_TIMER2 CH1OPRE
STBLKSRC5	Slave_TIMER2	Slave_TIMER2	Slave_TIMER3	Slave_TIMER4	Slave_TIMER3

到 / 来自	Slave_TIMER0	Slave_TIMER1	Slave_TIMER2	Slave_TIMER3	Slave_TIMER4
	CH1OPRE	CH1OPRE	比较 3	比较 0	比较 0
STBLKSRC6	Slave_TIMER3 比较 0	Slave_TIMER3 比较 1	Slave_TIMER3 CH1OPRE	Slave_TIMER4 比较 3	Slave_TIMER3 比较 3
STBLKSRC7	Slave_TIMER4 比较 1	Slave_TIMER4 比较 0	Slave_TIMER3 比较 3	Slave_TIMER4 CH1OPRE	Slave_TIMER3 CH1OPRE

当 EXEVyMEEN 位置 1 时，将使能外部事件的存储功能，外部事件不会立即生效。一旦指定的时间完成，该外部事件将被存储并生成。

窗口模式

窗口模式中，在指定时间内发生的外部事件 EXEVyC（y = 0..9）有效，其他时间发生的则被忽略。在指定的时间内，窗口信号为高电平。此模式由 EXEVyFM[3:0]位域配置，范围从 4’b1101 到 4’b1111。

如果在指定时间内未发生外部事件 EXEVyC(y=0..9)，则在指定时间结束时将产生超时事件。

窗口信号源有两种类型：

- Slave_TIMERx本身：指定的时间是指从计数器复位到比较事件发生持续的时间。（EXEVyFM [3:0] = 4’b1101和4’b1110，用于分别设置比较1和比较2事件）；
- 来自其他Slave_TIMER单元（EXEVyFM [3:0] = 4’b0101~4’b1100）的STWDSRC：指定的时间是指从选定的Slave_TIMER计数器复位到比较事件发生持续的时间。具体请见*表19-13. 窗口模式的滤波信号映射*。

表 19-13. 窗口模式的滤波信号映射

到 / 来自	Slave_TIMER0	Slave_TIMER1	Slave_TIMER2	Slave_TIMER3	Slave_TIMER4
STWDSRC	Slave_TIMER1 比较 1	Slave_TIMER0 比较 1	Slave_TIMER3 比较 1	Slave_TIMER2 比较 1	Slave_TIMER3 比较 1

当 EXEVyMEEN 位置 1 时，将使能外部事件的存储功能，外部事件不会立即生效。一旦指定的时间完成，该外部事件将被存储并生成。

DAC 触发

当 Slave_TIMERx 的更新事件发生时，如果 SHRTIMER_STxCTL0 寄存器中的 DACTRGS [1:0]！= 2’b00，则在 SHRTIMER_DACTRIGOx（x = 0..3）上生成 DAC 触发请求。如果 DACTRGS [1:0] = 2’b00，则 Slave_TIMERx 不会生成 DAC 触发请求。SHRTIMER_DACTRIGOx（x = 0..3）是从 Slave_TIMERx 连接到 DAC 模块的内部信号。具体请参考 *DAC 触发*章节。

19.4.3. DLL 校准

DLL 可以产生并校准超高分辨率时钟 SHRTIMER_HPCK（$f_{SHRTIMER_HPCK} = 64 * f_{SHRTIMER_CK}$）。DLL 模块可以一次或定期校准超高分辨率时钟 SHRTIMER_HPCK。

当 SHRTIMER_DLLCCTL 寄存器中的 CLBPEREN 位置 1 时，使能定期的 DLL 校准，CLBPER [1:0]位域配置校准周期。DLL 将在整个 SHRTIMER 运行期间定期校准时钟。

当 SHRTIMER_DLLCCTL 寄存器中的 CLBPEREN 位清 0 时，将 CLBSTRT 置 1，DLL 只校准一次超高分辨率时钟 SHRTIMER_HPCK。

19.4.4. 突发模式

突发模式控制器允许通过硬件使 CHyOPRE（y = 0,1）交替输出空闲和运行状态。该模式由 SHRTIMER_BMCTL 寄存器中的 BMEN 位使能，通常在轻载情况中使用。

突发模式控制器包括：

- 1个计数器（BM-counter）；
- 1个比较寄存器：SHRTIMER_BMCMPV，用于定义空闲状态的持续时间；
- 1个周期寄存器：SHRTIMER_BMCAR，用于定义空闲和运行状态持续时间的总和。

BM-counter 的计数模式

BM-counter 可以运行在连续模式或单脉冲模式下。

当 BMCTN = 1 时，BM-counter 运行在连续模式下。BM-counter 从 0 连续计数到计数器重载值（SHRTIMER_BMCAR）。当计数到计数器重载值时，计数器将从 0 重新启动。突发模式过程一直持续到 SHRTIMER_BMCTL 中的 BMOPTF 位被复位。

当 BMCTN = 0 时，BM-counter 运行在单脉冲模式下。BM-counter 从 0 连续计数到计数器重载值（SHRTIMER_BMCAR）。当计数到计数器重载值时，BM-counter 停止计数。

当计数到计数器重载值（SHRTIMER_BMCAR）时，SHRTIMER_INTF 寄存器中的 BMPERIF 位置 1，如果 BMPERIE = 1（在 SHRTIMER_INTEN 寄存器中），则突发模式控制器产生突发模式周期中断请求。可以写 1 到 SHRTIMER_INTC 中的 BMPERIFC 位来清除 BMPERIF 位。

突发模式的时序

BM-counter 由几个时钟源提供时钟，可以通过 SHRTIMER_BMCTL 寄存器中的 BMCLKS[3:0] 位选择。当选定的时钟源信号的上升沿到达时，BM-counter 计数值加 1。

当 BMCLKS[3:0] = 4’b1010 时，BM-counter 的时钟源是 $f_{SHRTIMER_CK}$ 分频后得到的，分频系数由 SHRTIMER_BMCTL 寄存器中的 BMPSC [3:0]位域定义。

当 BMCLKS[3:0] = 4’b0110~4’b1001 时，BM-counter 的时钟源是芯片内部信号：BMCLKy（y = 0..3），具体请见*表 19-14. 突发模式的芯片内部信号*。

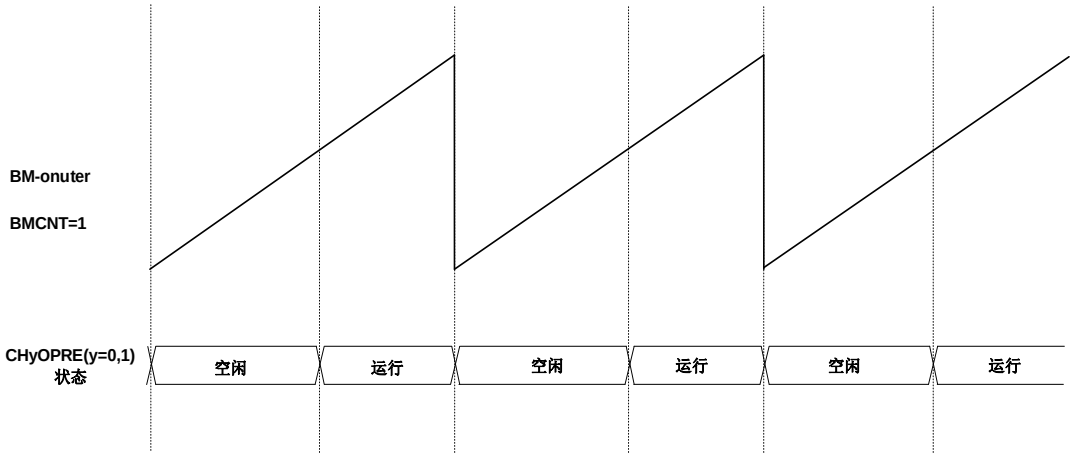
表 19-14. 突发模式的芯片内部信号

BMCLKy(y=0..3)	芯片内部信号
BMCLK0	保留
BMCLK1	保留
BMCLK2	TIMER6_TRGO
BMCLK3	保留

空闲状态的持续时间由 SHRTIMER_BMCMPV 寄存器定义，并且 SHRTIMER_BMCAR 寄存

器定义了突发模式的周期，该周期值是空闲状态和运行状态持续时间之和，具体请见*图 19-39. 突发模式时序图*。

图 19-39. 突发模式时序图



当 BMSE 置位时，SHRTIMER_BMCMPV 和 SHRTIMER_BMCAR 寄存器是预装载，在下列情形下会从预装载传输到有效寄存器：

- 当使能突发模式时（BMEN=1）；
- 当突发模式周期结束时。

注意：当写 SHRTIMER_BMCAR 后会暂时禁能更新，直到写 SHRTIMER_BMCMPV 寄存器后才恢复更新。

突发模式进入

SHRTIMER_BMSTRG 寄存器中定义了 32 个可触发突发模式的事件。这些触发事件可以同时选择，然后再进行逻辑或运算。而在 BM-counter 的计数过程中，这些触发事件被忽略。这些触发事件分为七种：

1. Master_TIMER 事件：重复事件，复位/翻转事件，比较 0/1/2/3 事件；

2. Slave_TIMERx 事件：重复事件，复位/翻转事件，比较 0 和比较 1 事件；

3.外部事件：EXEV6 和 EXEV7；

4. EXEV6 事件之后的 Slave_TIMER0 周期事件；

5. EXEV7 事件之后的 Slave_TIMER3 周期事件；

6.芯片内部信号：TIMER6_TRGO；

7.软件：写 1 到 SHRTIMER_BMSTRG 寄存器的 SWTRG 位。

触发事件发生时，有两种进入突发模式的方式：常规进入和延迟进入。

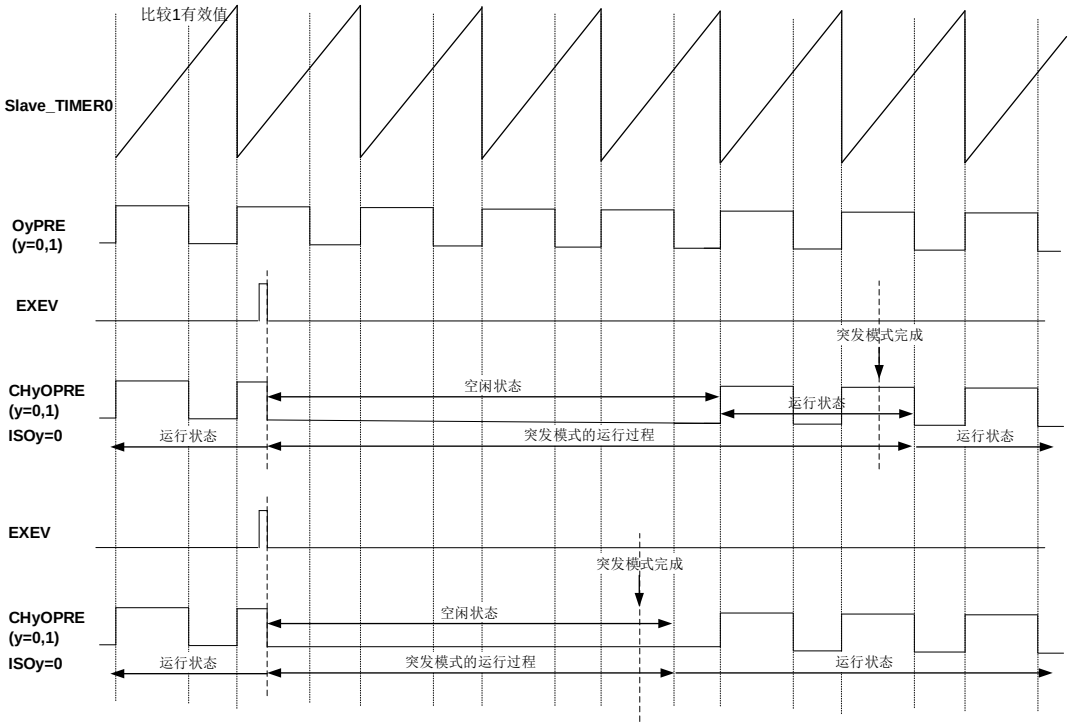
常规进入

当 SHRTIMER_STxCHOCTL 寄存器中的 BMCHyDTI（y = 0，1）位为 0 时，突发模式是常规进入模式。选定事件发生后的第一个 BM-counter 计数时钟到来时，输出将进入突发模式，并输出空闲电平（根据 ISO0 位和 ISO1 位设置）。

*图 19-40. 突发模式的常规进入*显示了在以下配置时，Slave_TIMER0 中的 CHyOPRE 波形：

- CyOPRE 处于常规模式：SHRTIMER_ST0CHOCTL 寄存器中的 DTEN = 0，SHRTIMER_ST0CTL0寄存器中的BLNMEN = 0；
- 周期事件产生置位请求；
- 比较1事件产生复位请求；
- BM-counter的时钟源是Slave_TIMER0的翻转事件：BMCLKS [3:0] = 4’b0001。

图 19-40. 突发模式的常规进入



延迟进入

当 SHRTIMER_STxCHOCTL 寄存器中的 BMCHyDTI（y = 0,1）位为 1 时，突发模式的进入被延迟。在进入突发模式之前，CHyOPRE 被强制插入死区时间。

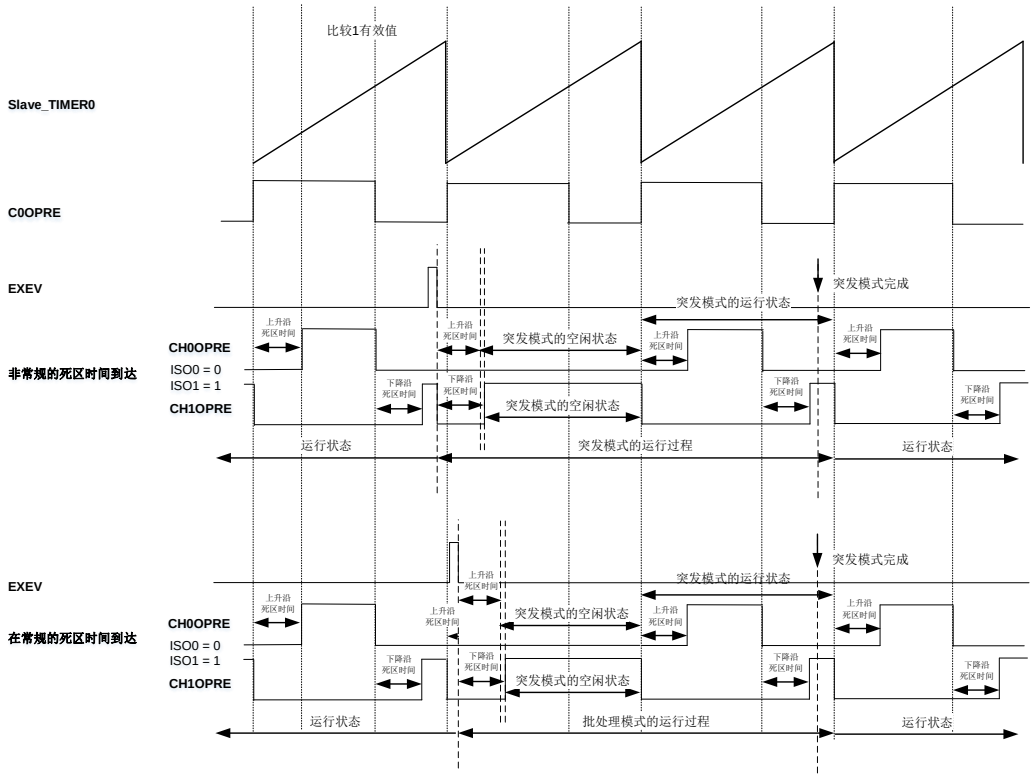
每个 CHyOPRE 都有自己的死区插入值：

- BMCH0DTI = 1时，DTRCFG[15:0]用于配置CH0OPRE的死区时间；
- BMCH1DTI = 1时，DTFCFG[15:0]用于配置CH1OPRE的死区时间。

延迟进入模式适用于以下情况：CHyOPRE(y=0,1)之一具有有效的空闲电平（ISOy = 1），且死区时间为正（DTRS /DTFS 设置为 0）。

在常规死区时间内，突发模式被触发，当前死区过程中止，将重新开始新的死区插入过程。详见*图 19-41. 突发模式的延迟进入*。

图 19-41. 突发模式的延迟进入



突发模式退出

在连续模式下，突发模式由软件强制退出。BMOPTF 或 BMEN 位重写为 0 后，发生输出置位/复位请求时，会退出突发模式。详情请见*图 19-40. 突发模式的常规进入*和*图 19-41. 突发模式的延迟进入*。

在单脉冲模式下，一旦经过空闲周期，就退出突发模式。

突发模式的时钟

可以在突发模式工作（运行，空闲）期间停止并复位 Master_TIMER 和 Slave_TIMERx（x = 0..4）单元的计数器，可通过 SHRTIMER_BMCTL 寄存器中的 BMMT 位和 BMSTx（x = 0..4）位进行配置：

- BMMT或BMSTx（x = 0..4）= 0：保持Master_TIMER或Slave_TIMERx（x = 0..4）的计数器时钟（SHRTIMER_PSCCK），且计数器正常运行；
- BMMT或BMSTx（x = 0..4）= 1：Master_TIMER或Slave_TIMERx（x = 0..4）计数器时钟（SHRTIMER_PSCCK）停止，并复位计数器。

使用 SHRTIMER_STxCMP0CP 寄存器模拟突发模式

可以使用 SHRTIMER_STxCMP0CP 寄存器来生成类似于突发模式控制的波形，配置如下：

- 比较0事件用于产生复位请求；
- 周期事件用于产生置位请求；
- 使用DMA（重复事件）连续将两个32位数据写入SHRTIMER_STxCMP0CP寄存器，如下所示：

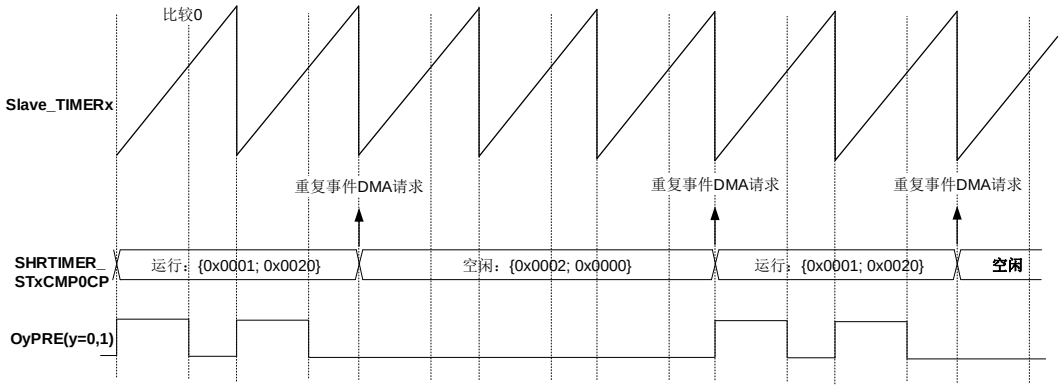
SHRTIMER_STxCMP0CP = {CREP [7:0] =运行周期数-1；CMP0VAL [15:0] =占空比}

SHRTIMER_STxCMP0CP = {CREP [7:0] =空闲周期数-1；CMP0VAL [15:0] = 0}

例如，要生成每 5 个周期中有 2 个周期输出置位的 PWM 波，可进行如下配置：

- 运行：SHRTIMER_STxCMP0CP = {0x0001; 0x0020}；
- 空闲：SHRTIMER_STxCMP0CP = {0x0002; 0x0000}。

图 19-42. 模拟突发模式示例



19.4.5. 同步输入/输出

同步电路在 Master_TIMER 内部：

- 同步输出：SHRTIMER可以作为主机产生同步信号；
- 同步输入：SHRTIMER也可以作为从机等待触发同步。

同步输出

可以将 SHRTIMER 配置为主机，同步外部资源；

可以配置 SHRTIMER_MTCTL0 寄存器中的 SYNOSRC[1:0]位域，选择发送到同步输出上的源。有以下四个源：

- 2’b00：Master_TIMER启动事件。在以下三种情况下可以将生成的启动事件用作同步输出，当MTCEN位置1时，当计数器在单脉冲模式下达到周期值后重新启动时，还有当CTNM或CNTRSTM位置1时，在计数期间发生的复位事件；
- 2’b01：Master_TIMER比较0事件；
- 2’b10：Slave_TIMER0复位和启动事件。它与Master_TIMER启动事件类似，除以下情况外：连续模式下的计数器翻转，在CNTRSTM = 0时的单脉冲模式下放弃的复位请求；
- 2’b11：Slave_TIMER0比较0事件。

SHRTIMER_MTCTL0 寄存器中的 SYNOPLS[1:0]位域确定同步输出信号的极性：

- 2’b00：脉冲产生禁能。同步输出引脚SHRTIMER_SCOUT上没有脉冲；
- 2’b01：保留；
- 2’b10：在同步输出引脚SHRTIMER_SCOUT上生成正脉冲。正脉冲的长度为16 $t_{SHRTIMER_CK}$个周期；

- 2’b11：在同步输出引脚SHRTIMER_SCOUT上生成负脉冲。负脉冲的长度为16 $t_{SHRTIMER_CK}$个周期。

同步输入

SHRTIMER 可以作为从机等待触发同步。可以通过 SHRTIMER_MTCTL0 寄存器中的SYNISRC [1:0]位域选择同步输入源。有四个输入触发源可选：

- 2’b00：同步输入禁能；
- 2’b01：保留；
- 2’b10：芯片内部信号。高级定时器TIMER0的TIMER0_TRGO信号；
- 2’b11：芯片外部引脚。芯片外部引脚（SHRTIMER_SCIN）上的正脉冲（上升沿有效）。

Master_TIMER 由 SHRTIMER_MTCTL0 寄存器中的 SYNISTRT 位和 SYNIRST 位配置。Slave_TIMERx 由 SHRTIMER_STxCTL0 寄存器中的 SYNISTRT 位和 SYNIRST 位配置。

当 SYNISTRT 置 1 时，必须先使能定时器（将 STxCEN 位或 MTCEN 位置 1），则同步输入信号将启动计数器。在连续模式下，即使 STxCEN 位或 MTCEN 位被置 1，计数器也不会启动，只有在同步输入信号到达后才会启动。

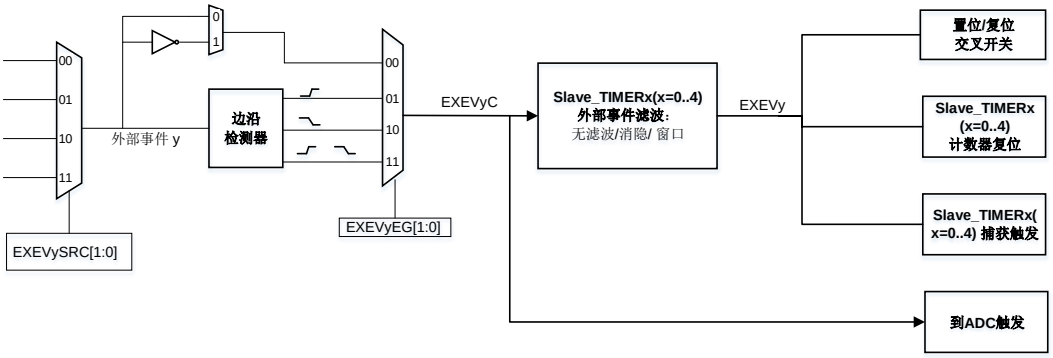
当 SYNIRST 置 1 时，同步输入信号将复位计数器，并像其他任何重置事件一样递减重复计数器。

19.4.6. 外部事件

10 个外部事件可以同时用于 5 个 Slave_TIMER 中的任意 1 个。通过 SHRTIMER_EXEVCFG0 寄存器配置外部事件 y（y = 0..4），通过 SHRTIMER_EXEVCFG0 和 SHRTIMER_EXEVDFCTL 寄存器配置外部事件 y（y = 5..9）。

处理外部事件 y（y = 0..4）的过程如*图 19-43. 外部事件 y（y=0..4）处理过程框图*所示。

图 19-43. 外部事件 y（y=0..4）处理过程框图

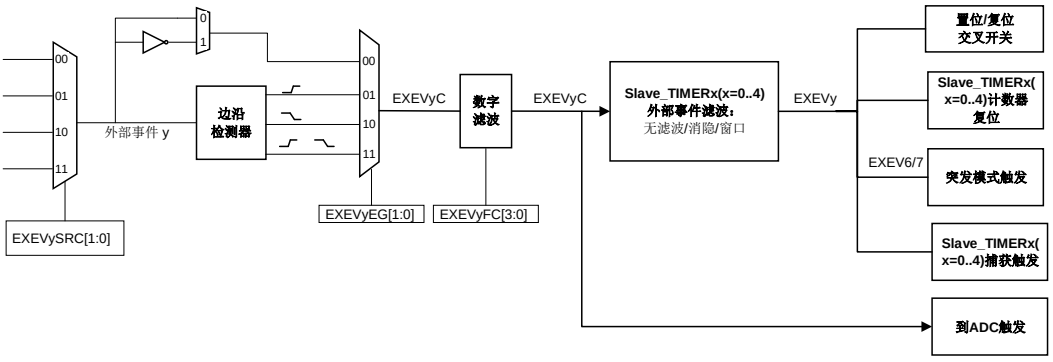


外部事件 y（y = 0..4）的配置如下：

- 4个源：通过EXEVySRC[1:0]位域进行配置；
- 有效沿选择：通过EXEVyEG[1:0]位域进行配置。可以是电平有效的或边沿有效（上升沿，下降沿或两者兼有）；
- 极性选择：在电平有效（EXEVyEG [1:0] = 2’b00）时，由EXEV0P位进行配置。

处理外部事件 y（y = 5..9）的过程如*图 19-44. 外部事件 y（y=5..9）处理过程框图*所示。

图 19-44. 外部事件 y（y=5..9）处理过程框图



外部事件 y（y=5..9）的配置如下：

- 4个源：通过EXEVySRC[1:0]位域进行配置；
- 有效沿选择：通过EXEVyEG[1:0]位域进行配置。可以是电平有效的或边沿有效（上升沿，下降沿或两者兼有）；
- 极性选择：在电平有效（EXEVyEG [1:0] = 2’b00）时，由EXEV0P位进行配置；
- 数字滤波配置：配置SHRTIMER_EXEVDFCTL寄存器中的EXEVyFC[3:0]位域。

数字滤波器的采样时钟 $f_{SHRTIMER_EXEVFCK}$ 由 SHRTIMER_EXEVDFCTL 寄存器中的 EXEVFDIV[2:0]位域定义。

这些外部事件源 EXEVySRCz（y = 0..9，z = 0..4）可以来自比较器、数字输入引脚、ADC 的模拟看门狗和 TIMER_TRGO。具体请参考*表 19-15. 外部事件映射*。

表 19-15. 外部事件映射

外部事件	EXEVySRC0	EXEVySRC1	EXEVySRC2	EXEVySRC3
外部事件 0	PC12	比较器 1	TIMER0_TRGO	ADC0_AWD0
外部事件 1	PC11	比较器 3	TIMER1_TRGO	ADC0_AWD1
外部事件 2	PB7/PD5	比较器 5	TIMER2_TRGO	ADC0_AWD2
外部事件 3	PB6/PG11	x	x	ADC1_AWD0
外部事件 4	PB9/PG12	x	x	ADC1_AWD1
外部事件 5	PB5	比较器 1	TIMER5_TRGO	ADC1_AWD2
外部事件 6	PB4	比较器 3	TIMER6_TRGO	x
外部事件 7	PB8	比较器 5	x	x
外部事件 8	PB3	x	x	x
外部事件 9	PC6/PG13	x	x	x

注意：“×”表示不可用。

可以直接使用外部事件 y（y = 0..9），也可以对其进行滤波处理（以在指定时间内限制其操作）。具体参考*外部事件滤波*。

19.4.7. 故障输入

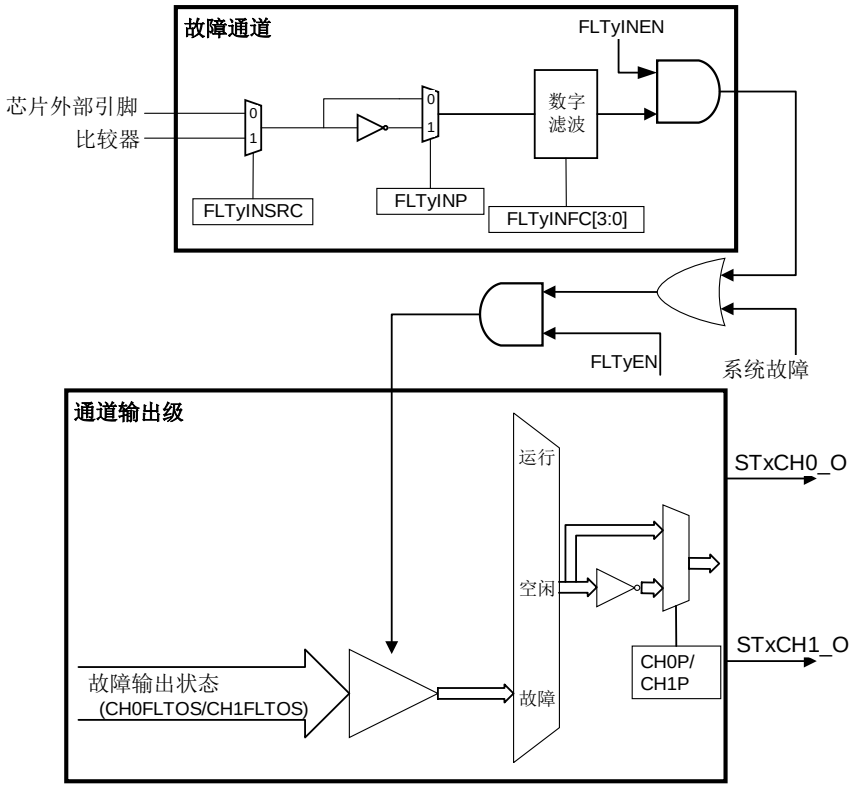
SHRTIMER 具有故障保护机制，可用于每个 Slave_TIMERx。具体请参考*图 19-45. 故障输入结构图*。

发生故障事件时，输出（STxCHy_O，x = 0..4，y = 0,1）为预置的电平，该电平将一直保持到软件重新使能输出（写 1 到 STxCHyEN 位）时为止。

预置电平由 SHRTIMER_STxCHOCTL 寄存器中的 CHyFLTOS[1:0]位域配置，保护机制可以处理两种类型的故障源：

- 故障通道：来自数字输入引脚或比较器的故障事件；
- 系统故障：来自MCU内部的信号，例如SRAM奇偶校验器。

图 19-45. 故障输入结构图



通过 SHRTIMER_STxFLTCTL 寄存器中的 FLTyEN 位使能故障输入。对 SHRTIMER_STxFLTCTL 寄存器中的 FLTENPROT 位的一次写入可以保护 FLTyEN 位。当 FLTENPROT 位置 1 时，FLTyEN 位写保护（只读）。

故障通道

可以通过 SHRTIMER_FLTINCFG0 和 SHRTIMER_FLTINCFG1 寄存器配置所有的故障通道。

FLTyINSRC（y = 0..4）位用于选择故障通道源，即可以是数字输入引脚，也可以是比较器输出。具体请见*表 19-16. 故障通道映射*。

表 19-16. 故障通道映射

故障通道	FLTyINSRC = 0（输入引脚）	FLTyINSRC = 1（比较器）
故障通道 0	PA12	比较器 1
故障通道 1	PA15	比较器 3
故障通道 2	PB10 / PD4	比较器 5
故障通道 3	PB11	x
故障通道 4	PC7 / PG10	x

注意：“×”表示不可用。

可以通过 SHRTIMER_FLTINCFG0 和 SHRTIMER_FLTINCFG1 寄存器中的 FLTyINP 位来配置故障信号的极性。如果 FLTyINP = 0，信号低电平有效；如果 FLTyINP = 1，则高电平有效。

可通过 SHRTIMER_FLTINCFG0 和 SHRTIMER_FLTINCFG1 寄存器中的 FLTyINFC[3:0]位域，对设置极性后的数字信号滤波器进行配置。数字滤波器采样时钟 $f_{SHRTIMER_FLTFCK}$ 由 SHRTIMER_FLTINCFG1 寄存器中的 FLTFDIV[2:0]位域定义。

可通过 SHRTIMER_FLTINCFG0 和 SHRTIMER_FLTINCFG1 寄存器中的 FLTyINEN 位来使能故障通道 y（y = 0..4），所有通道可同时使能。

对 SHRTIMER_FLTINCFG0 和 SHRTIMER_FLTINCFG1 寄存器中 FLTyINPROT 位的一次写入，可保护 FLT0INEN 位，FLT0INP 位，FLT0INSRC 位和 FLT0INFC [3:0]位域。当 FLTyINPROT 位置 1 时，这些位写保护（只读）。

系统故障

系统故障来自芯片内部的信号：

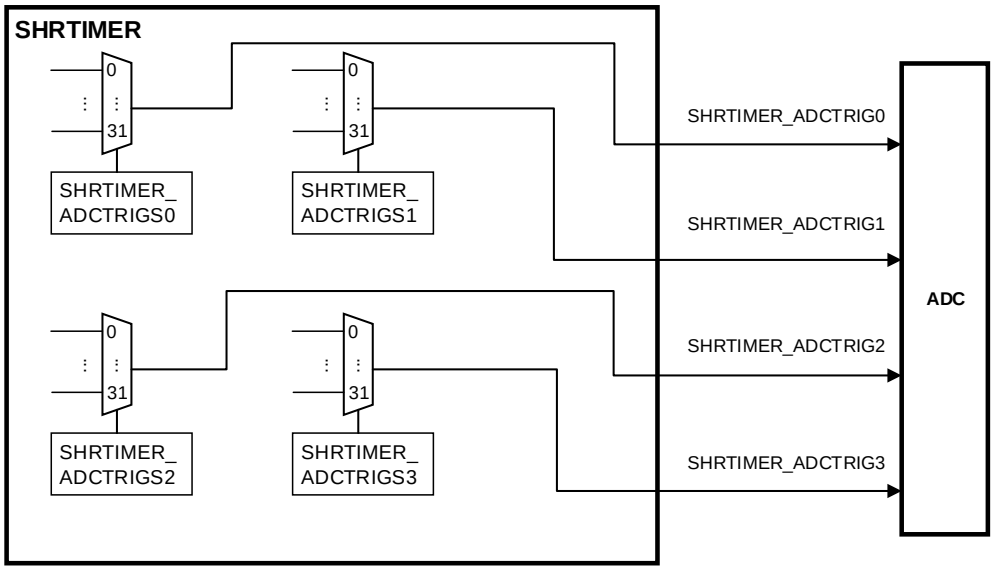
- 时钟监视生成的HXTAL故障事件；
- Cortex®-M33锁定信号；
- 低压检测器（LVD）的输出。

当 SHRTIMER_STxFLTCTL 寄存器中的 FLTyEN 位置 1 时，系统故障才有效。系统故障可以覆盖故障通道输入（逻辑或）。

19.4.8. ADC 触发

Master_TIMER 和 Slave_TIMERx 可以触发 ADC，4 个独立的触发（SHRTIMER_ADCTRIGy，y = 0..3）可用于使能 ADC 常规序列。具体请参考*图 19-46. ADC 触发源选择图*。

图 19-46. ADC 触发源选择图



每个触发输出最多可以连接（逻辑或运算）32 个事件。它们在 SHRTIMER_ADCTRIGSy（y = 0..3）寄存器中定义。

SHRTIMER_ADCTRIGSy（y = 0..3）寄存器使能预加载，并可以使用与其相关的定时器进行同步更新。更新源由 SHRTIMER_CTL0 寄存器中的 ADTGyUSRC[2:0]位域定义。例如，ADTGyUSRC [2:0] = 3’b001，Slaver_TIMER0 是更新源：

- 如果SHRTIMER_STxCTL0寄存器中的SHWEN = 1，SHRTIMER_ADCTRIGSy（y = 0..3）寄存器被预加载，可以与Slaver_TIMER0同步更新；
- 如果SHRTIMER_STxCTL0寄存器中的SHWEN = 0，SHRTIMER_ADCTRIGSy（y = 0..3）寄存器不会被预加载，写访问将使触发源立即更新。

19.4.9. DAC 触发

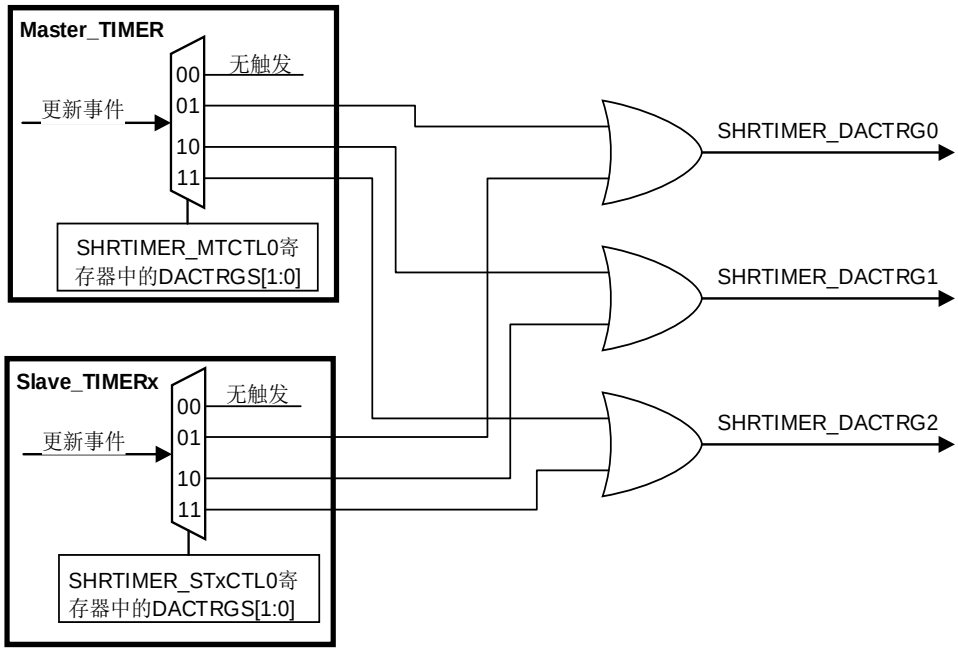
SHRTIMER 允许使用定时器更新同步更新片上 DAC。Master_TIMER 和 Slave_TIMERx 的更新事件可以在 SHRTIMER_DACTRIGy（y = 0..2）上生成 DAC 更新触发。

SHRTIMER_MTCTL0 和 SHRTIMER_STxCTL0 寄存器中的 DACTRGS [1:0]位域配置如下：

- 00：没有DAC触发事件发生；
- 01：在SHRTIMER_DACTRIG0上生成DAC触发事件；
- 10：在SHRTIMER_DACTRIG1上生成DAC触发事件；
- 11：在SHRTIMER_DACTRIG2上生成DAC触发事件。

在多个计时器中使能 DACTRGS [1:0]位域时，SHRTIMER_DACTRIGy（y = 0..2）将由所有定时器的更新事件或组成。具体请参考*图 19-47. DAC 触发源选择图*。

图 19-47. DAC 触发源选择图



19.4.10. 中断

大多数事件可以生成中断请求，所有的中断请求可分组到 7 个中断向量(SHRTIMER_IRQy，y=0..6)。详见*表 19-17. 中断映射*。

表 19-17. 中断映射

中断号	事件	Control bit
Master_TIMER：SHRTIMER_IRQ0	更新事件	SHRTIMER_MTDMAINTEN 中的 UPIE 位
	同步输入事件	SHRTIMER_MTDMAINTEN 中的 SYNIIE 位
	重复事件	SHRTIMER_MTDMAINTEN 中的 REPIE 位
	比较 0 事件	SHRTIMER_MTDMAINTEN 中的 CMP0IE 位
	比较 1 事件	SHRTIMER_MTDMAINTEN 中的 CMP1IE 位
	比较 2 事件	SHRTIMER_MTDMAINTEN 中的 CMP2IE 位
	比较 3 事件	SHRTIMER_MTDMAINTEN 中的 CMP3IE 位
Slave_TIMER0：SHRTIMER_IRQ1	延迟空闲模式进入	SHRTIMER_STxDMAINTEN 中的 DLYIIE 位
	计数器复位事件	SHRTIMER_STxDMAINTEN 中的 RSTIE 位
Slave_TIMER1：SHRTIMER_IRQ2	C1OPRE 从有效到无效	SHRTIMER_STxDMAINTEN 中的 CH1ONAIE 位
	C1OPRE 从无效到有效	SHRTIMER_STxDMAINTEN 中的 CH1OAIE 位
Slave_TIMER2：SHRTIMER_IRQ3	C0OPRE 从有效到无效	SHRTIMER_STxDMAINTEN 中的 CH0ONAIE 位
	C0OPRE 从无效到有效	SHRTIMER_STxDMAINTEN 中的 CH0OAIE 位
Slave_TIMER3：SHRTIMER_IRQ4	捕获 1 事件	SHRTIMER_STxDMAINTEN 中的 CAP1IE 位
	捕获 0 事件	SHRTIMER_STxDMAINTEN 中的 CAP0IE 位
	更新事件	SHRTIMER_STxDMAINTEN 中的 UPIE 位
	重复事件	SHRTIMER_STxDMAINTEN 中的 REPIE 位
Slave_TIMER4：	比较 3 事件	SHRTIMER_STxDMAINTEN 中的 CMP3IE 位

中断号	事件	Control bit
SHRTIMER_IRQ5	比较 2 事件	SHRTIMER_STxDMAINTEN 中的 CMP2IE 位
	比较 1 事件	SHRTIMER_STxDMAINTEN 中的 CMP1IE 位
	比较 0 事件	SHRTIMER_STxDMAINTEN 中的 CMP0IE 位
SHRTIMER_IRQ0	突发模式周期事件	SHRTIMER_INTEN 中的 BMPERIE 位
	DLL 校准完成	SHRTIMER_INTEN 中的 DLLCALIE 位
SHRTIMER_IRQ6	系统故障	SHRTIMER_INTEN 中的 SYSFLTIE 位

19.4.11. DMA 请求

大多数事件可以生成 DMA 请求，每个定时器对应一个 DMA 通道，详见*表 19-18. DMA 请求映射*。

表 19-18. DMA 请求映射

DMA channel	Event	Control bit
Master_TIMER：DMA0_Channel1	更新事件	SHRTIMER_MTDMAINTEN 中的 UPDEN 位
	同步输入事件	SHRTIMER_MTDMAINTEN 中的 SYNIDEN 位
	重复事件	SHRTIMER_MTDMAINTEN 中的 REPDEN 位
	比较 0 事件	SHRTIMER_MTDMAINTEN 中的 CMP0DEN 位
	比较 1 事件	SHRTIMER_MTDMAINTEN 中的 CMP1DEN 位
	比较 2 事件	SHRTIMER_MTDMAINTEN 中的 CMP2DEN 位
	比较 3 事件	SHRTIMER_MTDMAINTEN 中的 CMP3DEN 位
Slave_TIMER0：DMA0_Channel2	延迟空闲模式进入	SHRTIMER_STxDMAINTEN 中的 DLYIDEN 位
	计数器复位事件	SHRTIMER_STxDMAINTEN 中的 RSTDEN 位
	C1OPRE 从有效到无效	SHRTIMER_STxDMAINTEN 中的 CH1ONADEN 位
Slave_TIMER1：DMA0_Channel3	C1OPRE 从无效到有效	SHRTIMER_STxDMAINTEN 中的 CH1OADEN 位
	C0OPRE 从有效到无效	SHRTIMER_STxDMAINTEN 中的 CH0ONADEN 位
Slave_TIMER2：DMA0_Channel4	C0OPRE 从无效到有效	SHRTIMER_STxDMAINTEN 中的 CH0OADEN 位
	捕获 1 事件	SHRTIMER_STxDMAINTEN 中的 CAP1DEN 位
	捕获 0 事件	SHRTIMER_STxDMAINTEN 中的 CAP0DEN 位
Slave_TIMER3：DMA0_Channel5	更新事件	SHRTIMER_STxDMAINTEN 中的 UPDEN 位
	重复事件	SHRTIMER_STxDMAINTEN 中的 REPDEN 位
	比较 3 事件	SHRTIMER_STxDMAINTEN 中的 CMP3DEN 位
Slave_TIMER4：DMA0_Channel6	比较 2 事件	SHRTIMER_STxDMAINTEN 中的 CMP2DEN 位
	比较 1 事件	SHRTIMER_STxDMAINTEN 中的 CMP1DEN 位
	比较 0 事件	SHRTIMER_STxDMAINTEN 中的 CMP0DEN 位

注意：必须先禁能 DMA 控制器，然后再禁能 DMA 请求。

19.4.12. DMA 模式

定时器的 DMA 模式是通过 DMA 模块，实现单个 DMA 请求配置 SHRTIMER 的多个寄存器的

功能。相关的寄存器（总共七个寄存器）如下：

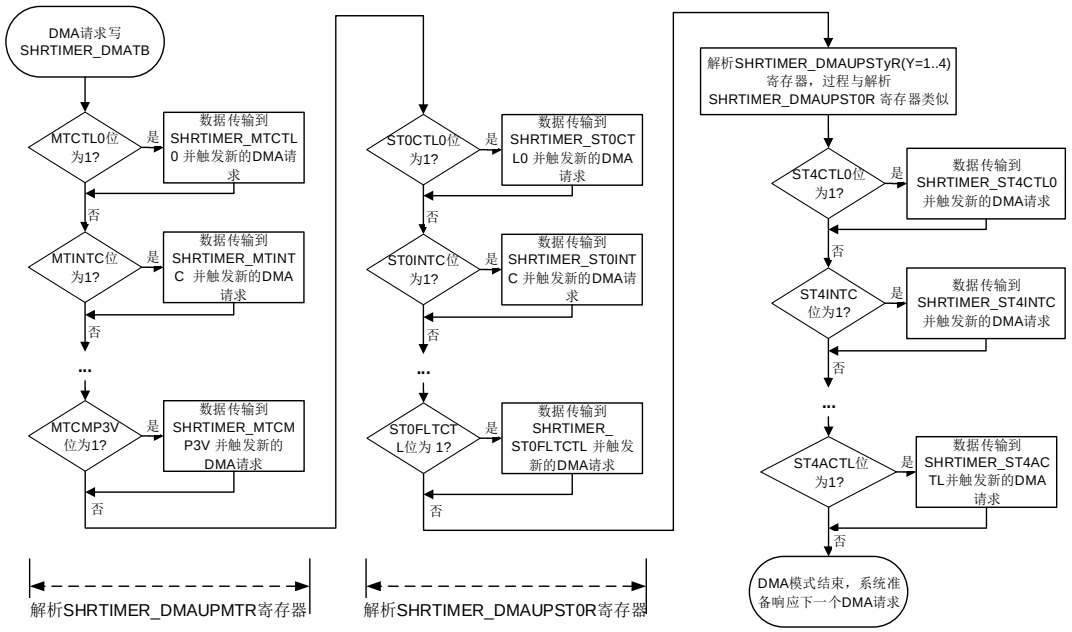
- SHRTIMER_DMAUPMTR：定义更新Master_TIMER中的哪些寄存器。Master_TIMER的大多数控制和数据寄存器都与一个选择位关联。如果该选择位置位，则写访问将重定向到关联的寄存器；
- SHRTIMER_DMAUPSTxR：定义更新Slave_TIMERx中的哪些寄存器。大多数Slave_TIMERx控制和数据寄存器都与一个选择位相关联。如果该选择位被置位，则写访问将重定向到关联的寄存器。
- SHRTIMER_DMATB：DMA传输缓冲区寄存器。只需要将指向SHRTIMER_DMATB寄存器的DMA模块作为目标，并禁能外设增量模式的外设配置。所有对该寄存器的写访问都将通过重定向机制在内部重新传输到最终目标寄存器。

DMA 模式功能仅适用于一个 DMA 通道，6 个通道中的任何一个都进行 DMA 传输。

DMA 模式是永久使能的（没有使能位）。通过对 SHRTIMER_DMATB 寄存器的首次写访问来启动 DMA 操作。

发生 DMA 请求时，SHRTIMER 会生成多个 32 位 DMA 请求并解析要更新的寄存器（在 SHRTIMER_DMAUPMTR 和 SHRTIMER_DMAUPSTxR 寄存器中定义）。如果选择位置 1，则写访问将重定向到关联的寄存器。如果选择位为 0，则跳过相关寄存器更新，并继续进行寄存器解析，直到检测到新的位置 1，触发新的 DMA 请求。6 个寄存器（SHRTIMER_DMAUPMTR 和 SHRTIMER_DMAUPSTxR 寄存器）全部解析后，DMA 模式完成，系统已准备好等待下一个 DMA 触发。若再有 DMA 请求事件发生，则 SHRTIMER 将重复上述过程。详见*图 19-48. DMA 模式运行流程图*。

图 19-48. DMA 模式运行流程图



19.4.13. Debug 模式

当 Cortex®-M33 内核暂停时，DBG_CTL0 寄存器中的 SHRTIMER_HOLD 位决定了计数器是

否停止运行。

SHRTIMER_HOLD = 0

若 SHRTIMER_HOLD = 0，则 SHRTIMER 继续正常运行。

SHRTIMER_HOLD = 1

若 SHRTIMER_HOLD = 1，则将停止 Master_TIMER 和所有 Slave_TIMERx 中的计数器。

如果 CHyFLTOS[1:0] = 2’b01、2’b10、2’b11，则输出进入 FAULT 状态。可以通过将 SHRTIMER_CHOUTEN 寄存器中的 STxCHyEN 位置 1，清零 SHRTIMER_HOLD 位来再次使能输出。如果 CHyFLTOS [1:0] = 2’b00，则输出保持其当前状态。退出调试模式时，输出将返回其原始状态。

所有计数器的复位/启动和捕获触发功能都禁能。除 ADC 触发外，所有外部事件的触发都被禁能。更新事件将被丢弃。突发模式电路被冻结：触发都被忽略，突发模式计数器停止。

DLL 校准正常运行。在运行模式下驱动正常输出的单元不受调试影响，例如死区时间单元，载波信号和置位/复位交叉开关等。

19.5. SHRTIMER 寄存器

SHRTIMER 基地址：0x4001 7400

寄存器进行分段寻址：

SHRTIMER Master_TIMER 寄存器基地址：0x4001 7400

SHRTIMER Slave_TIMER0 寄存器基地址：0x4001 7480

SHRTIMER Slave_TIMER1 寄存器基地址：0x4001 7500

SHRTIMER Slave_TIMER2 寄存器基地址：0x4001 7580

SHRTIMER Slave_TIMER3 寄存器基地址：0x4001 7600

SHRTIMER Slave_TIMER4 寄存器基地址：0x4001 7680

SHRTIMER 通用寄存器基地址：0x4001 7780

19.5.1. Master_TIMER 寄存器

SHRTIMER Master_TIMER 寄存器基地址：0x4001 7400

Master_TIMER 控制寄存器 0 (SHRTIMER_MTCTL0)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UPSEL[1:0]		UPREP	保留	SHWEN	DACTRGS[1:0]		保留			ST4CEN	ST3CEN	ST2CEN	ST1CEN	ST0CEN	MTCEN
rw		rw		rw	rw					rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNOSRC[1:0]		SYNOPLS[1:0]		SYNISTRT	SYNIRST	SYNISRC[1:0]		保留		HALFM	CNTRSTM	CTNM	CNTCKDIV[2:0]		
rw		rw		rw	rw	rw				rw	rw	rw	rw		

位/位域	名称	描述
31:30	UPSEL[1:0]	更新事件选择 该位域用于配置更新事件与 DMA 模式的关系。 00：更新事件产生与 DMA 模式无关。 01：在 DMA 模式下完成 DMA 传输时生成更新事件。 10：在 DMA 模式下 DMA 传输完成后，计数器翻转产生更新事件。仅适用于连续模式。 11：保留。
29	UPREP	重复事件生成更新事件

		该位用于使能重复事件生成更新事件。 0：重复事件生成更新事件禁能 1：重复事件生成更新事件使能 注意：仅当 UPSEL [1:0] = 2'b00 或 2'b01 时才能设置 UPREP。
28	保留	必须保持复位值
27	SHWEN	影子寄存器使能 0：影子寄存器禁能 1：影子寄存器使能
26:25	DACTRGS[1:0]	DAC 触发源 发生更新事件时，定时器生成 DAC 触发事件。该位域用于配置哪个触发源生成 DAC 触发事件。 00：不生成 DAC 触发事件 01：在 SHRTIMER_DACTRIG0 上生成 DAC 触发事件 10：在 SHRTIMER_DACTRIG1 上生成 DAC 触发事件 11：在 SHRTIMER_DACTRIG2 上生成 DAC 触发事件
24:22	保留	必须保持复位值
21	ST4CEN	Slave_TIMER4 计数器使能 0：Slave_TIMER4 计数器禁能 1：Slave_TIMER4 计数器使能 注意：不得在小于 8 个 $t_{SHRTIMER_CK}$ 时钟周期内修改该位。
20	ST3CEN	Slave_TIMER3 计数器使能 0：Slave_TIMER3 计数器禁能 1：Slave_TIMER3 计数器使能 注意：不得在小于 8 个 $t_{SHRTIMER_CK}$ 时钟周期内修改该位。
19	ST2CEN	Slave_TIMER2 计数器使能 0：Slave_TIMER2 计数器禁能 1：Slave_TIMER2 计数器使能 注意：不得在小于 8 个 $t_{SHRTIMER_CK}$ 时钟周期内修改该位。
18	ST1CEN	Slave_TIMER1 计数器使能 0：Slave_TIMER1 计数器禁能 1：Slave_TIMER1 计数器使能 注意：不得在小于 8 个 $t_{SHRTIMER_CK}$ 时钟周期内修改该位。
17	ST0CEN	Slave_TIMER0 计数器使能 0：Slave_TIMER0 计数器禁能 1：Slave_TIMER0 计数器使能 注意：不得在小于 8 个 $t_{SHRTIMER_CK}$ 时钟周期内修改该位。
16	MTCEN	Master_TIMER 计数器使能 0：Master_TIMER 计数器禁能 1：Master_TIMER 计数器使能

		注意：不得在小于 8 个 $t_{SHRTIMER_CK}$ 时钟周期内修改该位。
15:14	SYNOSRC[1:0]	同步输出源 该位域用于配置发送到同步输出 SHRTIMER_SCOUT 上的事件。 00：Master_TIMER 启动事件。 01：Master_TIMER 比较 0 事件 10：Slave_TIMER0 复位和启动事件 11：Slave_TIMER0 比较 0 事件
13:12	SYNOPLS[1:0]	同步输出脉冲 该位域用于配置同步输出 SHRTIMER_SCOUT 上的脉冲。 00：脉冲生成禁能。SHRTIMER_SCOUT 上无脉冲。 01：保留。 10：在 SHRTIMER_SCOUT 上产生正脉冲。它的长度是 16 个 $t_{SHRTIMER_CK}$ 个周期。 11：在 SHRTIMER_SCOUT 上产生负脉冲。它的长度是 16 个 $t_{SHRTIMER_CK}$ 个周期。
11	SYNISTRT	同步输入启动计数器 该位用于配置同步输入启动计数器。 0：同步输入不能启动计数器。 1：同步输入可以启动计数器。
10	SYNIRST	同步输入复位计数器 该位用于配置同步输入复位计数器。 0：同步输入不能复位计数器。 1：同步输入可以复位计数器。
9:8	SYNISRC[1:0]	同步输入源 该位域用于配置同步输入源。 00：同步输入禁能。 01：保留。 10：内部信号：高级定时器 TIMER0 中的 TIMER0_TRGO。 11：外部信号：SHRTIMER_SCIN 引脚上的正脉冲。 注意：相应的计时器使能后，将无法修改此位字段
7:6	保留	必须保持复位值
5	HALFM	半波模式 该位置 1 时，SHRTIMER_MTCMP0V 有效寄存器始终是计数器自动重载值（SHRTIMER_MTCAR）的一半。 0：半波模式禁能。 1：半波模式使能。
4	CNTRSTM	计数器复位模式 该位用于定义单脉冲模式下计数器的行为。 0：计数器只能计数到周期值后才能复位。 1：可以随时复位计数器（运行或停止）。
3	CTNM	连续模式。 0：单脉冲模式。当计数器达到 SHRTIMER_MTCAR 值时，它将由硬件停止。

1：连续模式。计数器在达到 SHRTIMER_MTCAR 值时，翻转到 0 并连续计数。

2:0	CNTCKDIV[2:0]	计数器时钟分频 该位域可以由软件配置，确定超高分辨率时钟（SHRTIMER_HPCK）和计数器时钟（SHRTIMER_PSCCK）的分频比。 当 SHRTIMER_MTACTL 中的 CNTCKDIV [3]为 0 时，$f_{SHRTIMER_PSCCK} = f_{SHRTIMER_HPCK} / 2^{CNTCKDIV[2:0]+1}$。 当 SHRTIMER_MTACTL 中的 CNTCKDIV [3]位为 1，且 CNTCKDIV[2:0]配置为 3'b000 时：$f_{SHRTIMER_PSSCK} = f_{SHRTIMER_HPCK}$ 0000：$f_{SHRTIMER_PSSCK} = f_{SHRTIMER_HPCK} / 2$ 0001：$f_{SHRTIMER_PSSCK} = f_{SHRTIMER_HPCK} / 4$ 0010：$f_{SHRTIMER_PSSCK} = f_{SHRTIMER_HPCK} / 8$ 0011：$f_{SHRTIMER_PSSCK} = f_{SHRTIMER_HPCK} / 16$ 0100：$f_{SHRTIMER_PSSCK} = f_{SHRTIMER_HPCK} / 32$ 0101：$f_{SHRTIMER_PSSCK} = f_{SHRTIMER_HPCK} / 64$ 0110：$f_{SHRTIMER_PSSCK} = f_{SHRTIMER_HPCK} / 128$ 0111：$f_{SHRTIMER_PSSCK} = f_{SHRTIMER_HPCK} / 256$ 1000：$f_{SHRTIMER_PSSCK} = f_{SHRTIMER_HPCK}$ 其他值保留。 注意：一旦使能定时器，就不能修改 CNTCKDIV [3:0]位域。

Master_TIMER 中断标志寄存器(SHRTIMER_MTINTF)

地址偏移：0x04

复位值：0x0000 0000

该寄存器只能进行字（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									UPIF	SYNIIF	REPIF	CMP3IF	CMP2IF	CMP1IF	CMP0IF
									r	r	r	r	r	r	r

位/位域	名称	描述
31:7	保留	必须保持复位值
6	UPIF	更新中断标志 发生更新事件时，此标志由硬件置位。 0：更新中断未发生 1：更新中断发生
5	SYNIIF	同步输入中断标志 同步输入到来时，此标志由硬件置位。 0：同步输入未发生

1：同步输入发生

4	REPIF	重复中断标志 发生重复事件时，此标志由硬件置位。 0：重复中断未发生 1：重复中断发生
3	CMP3IF	比较 3 中断标志 当发生比较 3 事件时，此标志由硬件置位。 0：比较 3 中断未发生 1：比较 3 中断发生
2	CMP2IF	比较 2 中断标志 当发生比较 2 事件时，此标志由硬件置位。 0：比较 2 中断未发生 1：比较 2 中断发生
1	CMP1IF	比较 1 中断标志 当发生比较 3 事件时，此标志由硬件置位。 0：比较 1 中断未发生 1：比较 1 中断发生
0	CMP0IF	比较 0 中断标志 当发生比较 0 事件时，此标志由硬件置位。 0：比较 0 中断未发生 1：比较 0 中断发生

Master_TIMER 中断标志清除寄存器(SHRTIMER_MTINTC)

地址偏移：0x08

复位值：0x0000 0000

该寄存器只能进行字（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									UPIFC	SYNIIFC	REPIFC	CMP3IFC	CMP2IFC	CMP1IFC	CMP0IFC
									w	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	UPIFC	更新中断标志清除 0：没有影响 1：清除更新中断标志
5	SYNIIFC	同步输入中断标志清除

		0：没有影响 1：清除同步输入中断标志
4	REPIFC	重复中断标志清除 0：没有影响 1：清除重复中断标志
3	CMP3IFC	比较 3 中断标志清除 0：没有影响 1：清除比较 3 中断标志
2	CMP2IFC	比较 2 中断标志清除 0：没有影响 1：清除比较 2 中断标志
1	CMP1IFC	比较 1 中断标志清除 0：没有影响 1：清除比较 1 中断标志
0	CMP0IFC	比较 0 中断标志清除 0：没有影响 1：清除比较 0 中断标志

Master_TIMER DMA 和中断使能寄存器 (SHRTIMER_MTDMAINTEN)

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留									UPDEN	SYNIDEN	REPDEN	CMP3DEN	CMP2DEN	CMP1DEN	CMP0DEN
									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									UPIE	SYNIIE	REPIE	CMP3IE	CMP2IE	CMP1IE	CMP0IE
									rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:23	保留	必须保持复位值
22	UPDEN	更新 DMA 请求使能 0：禁能 1：使能
21	SYNIDEN	同步输入 DMA 请求使能 0：禁能 1：使能

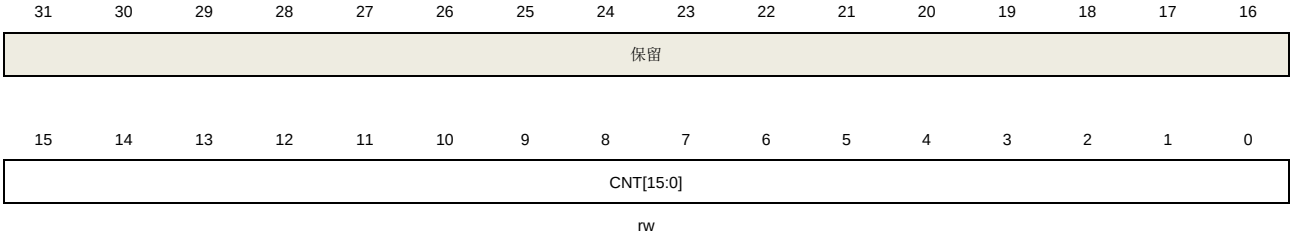
20	REPDEN	重复 DMA 请求使能 0：禁能 1：使能
19	CMP3DEN	比较 3 DMA 请求使能 0：禁能 1：使能
18	CMP2DEN	比较 2 DMA 请求使能 0：禁能 1：使能
17	CMP1DEN	比较 1 DMA 请求使能 0：禁能 1：使能
16	CMP0DEN	比较 0 DMA 请求使能 0：禁能 1：使能
15:7	保留	必须保持复位值
6	UPIE	更新中断使能 0：禁能 1：使能
5	SYNIIE	同步输入中断使能 0：禁能 1：使能
4	REPIE	重复中断使能 0：禁能 1：使能
3	CMP3IE	比较 3 中断使能 0：禁能 1：使能
2	CMP2IE	比较 2 中断使能 0：禁能 1：使能
1	CMP1IE	比较 1 中断使能 0：禁能 1：使能
0	CMP0IE	比较 0 中断使能 0：禁能 1：使能

Master_TIMER 计数器寄存器(SHRTIMER_MTCNT)

地址偏移：0x10

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。



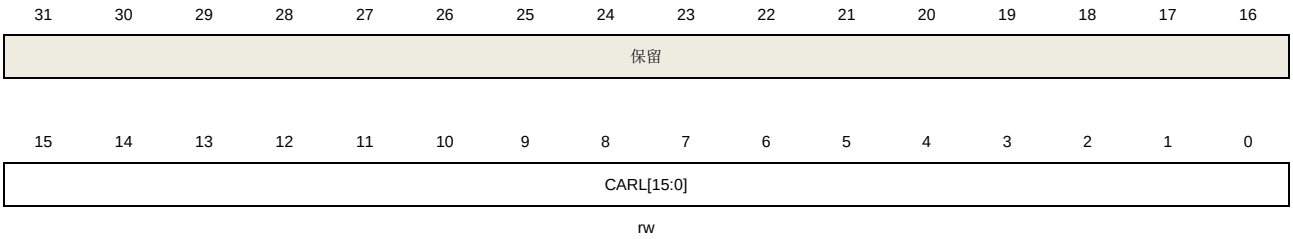
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CNT[15:0]	当前计数器值。对该位域进行写操作可以更改计数器的值。 仅当 Master_TIMER 停止（SHRTIMER_MTCTL0 寄存器中的 MTCEN = 0）时，对其进行写操作才能更改计数器的值。 注意： （1）计数器时钟分频系数小于 64（CNTCKDIV [3:0] < 5）时，计数器的最低有效位无效，它们不能被写入，读出值为 0。 （2）如果写入该位域的值高于 HRTIM_MPER 寄存器值，则定时器的行为是不可预测的。

Master_TIMER 计数器自动重载寄存器(SHRTIMER_MTCAR)

地址偏移：0x14

复位值：0x0000 FFDF

该寄存器只能进行字（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 该位域定义了计数器的自动重载值。该寄存器具有影子寄存器。如果影子寄存器被禁能（SHWEN = 0），它将保存有效寄存器的内容；否则，它将保存影子寄存器的内容。

注意：

（1）最小值必须大于或等于（3*$t_{SHRTIMER_CK}$）。例如：当 CNTCKDIV [3:0] = 4’b0000 时，CARL [15:0] > = 0x60。

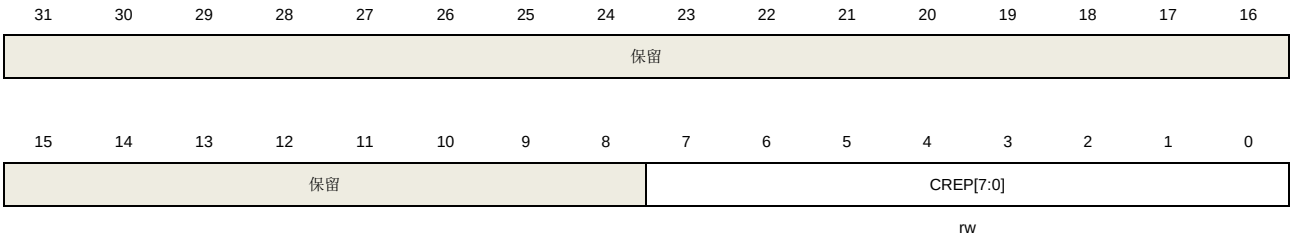
（2）最大值必须小于或等于（0xFFFF － 1*$t_{SHRTIMER_CK}$）。例如：当 CNTCKDIV [3:0] = 4’b0000 时，CARL [15:0] <= 0xFFDF。

Master_TIMER 重复计数寄存器(SHRTIMER_MTCREP)

地址偏移：0x18

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。



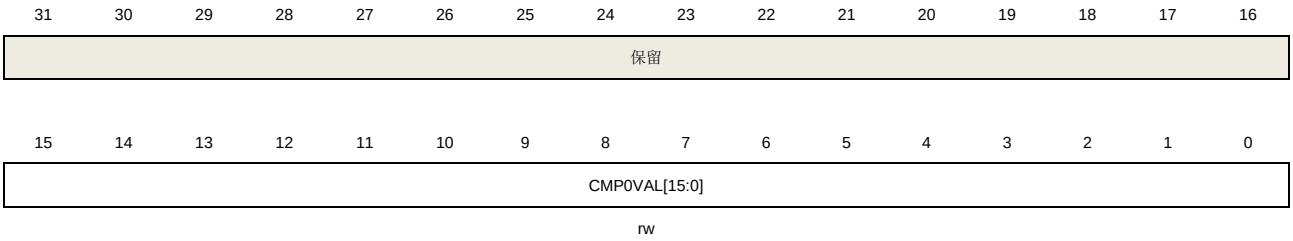
位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	CREP[7:0]	重复计数器值 该位域用于定义重复事件的发生率。当重复计数器递减计数到零时，连续模式下即将发生的翻转事件或复位事件将产生一个重复事件。 该寄存器具有影子寄存器。如果影子寄存器被禁能（SHWEN = 0），它将保存有效寄存器的内容；否则，它将保存影子寄存器的内容。

Master_TIMER 比较 0 寄存器(SHRTIMER_MTCMP0V)

地址偏移：0x1C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。



位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CMP0VAL[15:0]	比较 0 值

该位域用于配置与计数器进行比较的值。
该寄存器具有影子寄存器。如果影子寄存器被禁能（SHWEN = 0），它将保存有效寄存器的内容；否则，它将保存影子寄存器的内容。
注意：
（1）最小值必须大于或等于 3 个 $t_{SHRTIMER_CK}$。例如：当 CNTCKDIV [3:0] = 4'b0000 时，CARL [15:0] > = 0x60。
（2）最大值必须小于或等于（0xFFFF －1*$t_{SHRTIMER_CK}$）。例如：当 CNTCKDIV [3:0] = 4'b0000 时，CARL [15:0] <= 0xFFDF。

Master_TIMER 比较 1 寄存器(SHRTIMER_MTCMP1V)

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能进行字（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
CMP1VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CMP1VAL[15:0]	比较 1 值 该位域用于配置与计数器进行比较的值。 该寄存器具有影子寄存器。如果影子寄存器被禁能（SHWEN = 0），它将保存有效寄存器的内容；否则，它将保存影子寄存器的内容。 注意： （1）最小值必须大于或等于 3 个 $t_{SHRTIMER_CK}$。例如：当 CNTCKDIV [3:0] = 4'b0000 时，CARL [15:0] > = 0x60。 （2）最大值必须小于或等于（0xFFFF －1*$t_{SHRTIMER_CK}$）。例如：当 CNTCKDIV [3:0] = 4'b0000 时，CARL [15:0] <= 0xFFDF。

Master_TIMER 比较 2 寄存器(SHRTIMER_MTCMP2V)

地址偏移：0x28
复位值：0x0000 0000

该寄存器只能进行字（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

CMP2VAL[15:0]

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CMP2VAL[15:0]	比较 2 值 该位域用于配置与计数器进行比较的值。 该寄存器具有影子寄存器。如果影子寄存器被禁能（SHWEN = 0），它将保存有效寄存器的内容；否则，它将保存影子寄存器的内容。 注意： （1）最小值必须大于或等于 3 个 $t_{SHRTIMER_CK}$。例如：当 CNTCKDIV [3:0] = 4'b0000 时，CARL [15:0] > = 0x60。 （2）最大值必须小于或等于（0xFFFF －1*$t_{SHRTIMER_CK}$）。例如：当 CNTCKDIV [3:0] = 4'b0000 时，CARL [15:0] <= 0xFFDF。

Master_TIMER 比较 3 寄存器(SHRTIMER_MTCMP3V)

地址偏移：0x2C

复位值：0x0000 0000

该寄存器只能进行字（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
CMP3VAL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CMP3VAL[15:0]	比较 3 值 该位域用于配置与计数器进行比较的值。 该寄存器具有影子寄存器。如果影子寄存器被禁能（SHWEN = 0），它将保存有效寄存器的内容；否则，它将保存影子寄存器的内容。 注意： （1）最小值必须大于或等于 3 个 $t_{SHRTIMER_CK}$。例如：当 CNTCKDIV [3:0] = 4'b0000 时，CARL [15:0] > = 0x60。 （2）最大值必须小于或等于（0xFFFF －1*$t_{SHRTIMER_CK}$）。例如：当 CNTCKDIV [3:0] = 4'b0000 时，CARL [15:0] <= 0xFFDF。

Master_TIMER 附加控制寄存器(SHRTIMER_MTACTL)

地址偏移：0x7C

复位值：0x0000 0000

该寄存器只能进行字（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												CNTCKDIV[3]	保留		
												rw			

位/位域	名称	描述
31:4	保留	必须保持复位值
3	CNTCKDIV[3]	计数器时钟分频 该位位域可以由软件配置，确定超高分辨率时钟（SHRTIMER_HPCK）和计数器时钟（SHRTIMER_PSCCK）的分频比。 当 SHRTIMER_MTACTL 中的 CNTCKDIV [3]为 0 时，$f_{SHRTIMER_PSCCK} = f_{SHRTIMER_HPCK} / 2^{CNTCKDIV[2:0]+1}$。 当 SHRTIMER_MTACTL 中的 CNTCKDIV [3]位为 1 并且 CNTCKDIV [2：0]只能配置为 3’b000 时：$f_{SHRTIMER_PSSCK} = f_{SHRTIMER_HPCK}$ 注意：一旦使能定时器，就不能修改 CNTCKDIV [3:0]位域。
2:0	保留	必须保持复位值

19.5.2. Slave_TIMERx 寄存器(x=0..4)

SHRTIMER Slave_TIMER0 寄存器基地址：0x4001 7480

SHRTIMER Slave_TIMER1 寄存器基地址：0x4001 7500

SHRTIMER Slave_TIMER2 寄存器基地址：0x4001 7580

SHRTIMER Slave_TIMER3 寄存器基地址：0x4001 7600

SHRTIMER Slave_TIMER4 寄存器基地址：0x4001 7680

Slave_TIMERx 控制寄存器 0 (SHRTIMER_STxCTL0)

地址偏移：0x00

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UPSEL[3:0]				SHWEN	DACTRGS[1:0]		UPBMT	UPBST4	UPBST3	UPBST2	UPBST1	UPBST0	UPRST	UPREP	保留
rw				rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	

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

DELCMP3M[1:0]	DELCMP1M[1:0]	SYNISTRT	SYNIRST	保留	BLNMEN	HALFM	CNTRSTM	CTNM	CNTCKDIV[2:0]
rw	rw	rw	rw		rw	rw	rw	rw	rw

位/位域	名称	描述
31:28	UPSEL[3:0]	更新事件选择 该位域用于配置更新事件与 DMA 模式的关系。 0000：更新事件的生成独立于 DMA 模式。 0001：在 DMA 模式下完成 DMA 传输时生成更新事件。 0010：在 DMA 模式下完成 DMA 传输后的更新事件，生成更新事件。 0011：在 STxUPIN0 的上升沿生成更新事件。 0100：在 STxUPIN1 的上升沿生成更新事件。 0101：在 STxUPIN2 的上升沿生成更新事件。 0110：在 STxUPIN0 的上升沿到来之后的更新事件，生成更新事件。 0111：在 STxUPIN1 的上升沿到来之后的更新事件，生成更新事件。 1000：在 STxUPIN2 的上升沿到来之后的更新事件，生成更新事件。 其他值保留。 注意： （1）在写入新值之前，必须先复位该位域。 （2）当 UPSEL [3:0] = 4'b0001，4'b0011，4'b0100 和 4'b0101 时，可以有多个并发更新源。例如，通过 Master_TIMER（UPBMT = 1）和 DMA 模式进行更新。
27	SHWEN	影子寄存器使能 0：影子寄存器禁能 1：影子寄存器使能
26:25	DACTRGS[1:0]	DAC 触发源 发生更新事件时，定时器生成 DAC 触发事件。该位域用于配置哪个触发源生成 DAC 触发事件。 00：不生成 DAC 触发事件 01：在 SHRTIMER_DACTRIG0 上生成 DAC 触发事件 10：在 SHRTIMER_DACTRIG1 上生成 DAC 触发事件 11：在 SHRTIMER_DACTRIG2 上生成 DAC 触发事件
24	UPBMT	通过 Master_TIMER 更新事件进行更新 该位置 1 时，Slave_TIMERx（x = 0..4）更新事件与 Master_TIMER 更新事件同步，且它们的有效寄存器由 Master_TIMER 更新事件进行更新。 0：有效寄存器不由 Master_TIMER 更新。 1：有效寄存器由 Master_TIMER 更新。
23	UPBST4	通过 Slave_TIMER4 更新事件进行更新 该位置 1 时，Slave_TIMERx（x = 0..3）更新事件与 Slave_TIMER4 更新事件同步，且它们的有效寄存器由 Slave_TIMER4 更新事件进行更新。 0：有效寄存器不由 Slave_TIMER4 更新。 1：有效寄存器由 Slave_TIMER4 更新。

		注意：Slave_TIMER4 的寄存器中不存在此位。
22	UPBST3	通过 Slave_TIMER3 更新事件进行更新 该位置 1 时，Slave_TIMERx（x = 0,1,2,4）更新事件与 Slave_TIMER3 更新事件同步，且它们的有效寄存器由 Slave_TIMER3 更新事件进行更新。 0：有效寄存器不由 Slave_TIMER3 更新。 1：有效寄存器由 Slave_TIMER3 更新。 注意：Slave_TIMER3 的寄存器中不存在此位。
21	UPBST2	通过 Slave_TIMER2 更新事件进行更新 该位置 1 时，Slave_TIMERx（x = 0,1,3,4）更新事件与 Slave_TIMER2 更新事件同步，且它们的有效寄存器由 Slave_TIMER2 更新事件进行更新。 0：有效寄存器不由 Slave_TIMER2 更新。 1：有效寄存器由 Slave_TIMER2 更新。 注意：Slave_TIMER2 的寄存器中不存在此位。
20	UPBST1	通过 Slave_TIMER1 更新事件进行更新 该位置 1 时，Slave_TIMERx（x = 0,2,3,4）更新事件与 Slave_TIMER1 更新事件同步，且它们的有效寄存器由 Slave_TIMER1 更新事件进行更新。 0：有效寄存器不由 Slave_TIMER1 更新。 1：有效寄存器由 Slave_TIMER1 更新。 注意：Slave_TIMER1 的寄存器中不存在此位。
19	UPBST0	通过 Slave_TIMER0 更新事件进行更新 该位置 1 时，Slave_TIMERx（x = 0,2,3,4）更新事件与 Slave_TIMER0 更新事件同步，且它们的有效寄存器由 Slave_TIMER0 更新事件进行更新。 0：有效寄存器不由 Slave_TIMER0 更新。 1：有效寄存器由 Slave_TIMER0 更新。 注意：Slave_TIMER0 的寄存器中不存在此位。
18	UPRST	更新事件由复位事件生成 该位用于使能计数器复位事件或翻转事件生成更新事件。 0：复位事件或翻转事件生成更新事件禁能 1：复位事件或翻转事件生成更新事件使能
17	UPREP	更新事件由重复事件生成 该位用于使能重复事件生成更新事件。 0：重复事件生成更新事件禁能 1：重复事件生成更新事件使能
16	保留	必须保持复位值
15:14	DELCMP3M[1:0]	比较 3 延迟模式 00：比较 3 延迟模式禁能。只要计数器值等于比较 3 有效寄存器值，就会发生比较匹配。 01：比较 3 延迟模式 0。捕获 1 事件后，比较 3 的重新计算值为：（比较 3 有效寄存器值+捕获 1 值）。一旦计数器等于重新计算的值，就会发生比较匹配。 10：比较 3 延迟模式 1。在捕获 1 事件或比较 0 事件之后，比较 3 的重新计算值是：

		（比较 3 有效寄存器值+捕获 1 值，或者比较 3 有效寄存器值+比较 0 值）。一旦计数器等于重新计算的值，就会发生比较匹配。 11：比较 3 延迟模式 2。在捕获 1 事件或比较 2 事件之后，比较 3 的重新计算值为：（比较 3 有效寄存器值+捕获 1 的值，或比较 3 有效寄存器值+比较 2 的值）。一旦计数器等于重新计算的值，就会发生比较匹配。 注意：一旦使能计数器（SHRTIMER_MTCTL0 寄存器中的 STxCEN = 1），就不得修改此位域。
13:12	DELCMP1M[1:0]	比较 1 延迟模式 00：比较 1 延迟模式禁能。只要计数器值等于比较 1 有效寄存器值，就会发生比较匹配。 01：比较 1 延迟模式 0。捕获 0 事件后，比较 1 的重新计算值为：（比较 1 有效寄存器值+捕获 0 值）。一旦计数器等于重新计算的值，就会发生比较匹配。 10：比较 1 延迟模式 1。在捕获 0 事件或比较 0 事件之后，比较 1 的重新计算值是：（比较 1 有效寄存器值+捕获 0 值，或者比较 1 有效寄存器值+比较 0 值）。一旦计数器等于重新计算的值，就会发生比较匹配。 11：比较 1 延迟模式 2。在捕获 0 事件或比较 2 事件之后，比较 1 的重新计算值为：（比较 1 有效寄存器值+捕获 0 的值，或比较 1 有效寄存器值+比较 2 的值）。一旦计数器等于重新计算的值，就会发生比较匹配。 注意：一旦使能计数器（SHRTIMER_MTCTL0 寄存器中的 STxCEN = 1），就不得修改此位域。
11	SYNISTRT	同步输入启动计数器 该位用于配置同步输入启动计数器。 0：同步输入不能启动计数器。 1：同步输入可以启动计数器。
10	SYNIRST	同步输入复位计数器 该位用于配置同步输入复位计数器。 0：同步输入不能复位计数器。 1：同步输入可以复位计数器。
9:7	保留	必须保持复位值
6	BLNMEN	均衡模式使能 0：均衡模式禁能 1：均衡模式使能 注意：一旦使能计数器（SHRTIMER_MTCTL0 寄存器中的 STxCEN = 1），就不得修改此位域。
5	HALFM	半波模式 该位置 1 时，SHRTIMER_STxCMP0V 有效寄存器始终是计数器自动重载值（SHRTIMER_STxCAR）的一半。 0：半波模式禁能。 1：半波模式使能。
4	CNTRSTM	计数器复位模式 该位用于定义单脉冲模式下定时器计数器的行为。

		0：计数器只能计数到周期值后才能复位 1：可以随时复位计数器（运行或停止）。
3	CTNM	连续模式。 0：单脉冲模式。当计数器达到 SHRTIMER_STxCAR 值时，它将由硬件停止。 1：连续模式。计数器在达到SHRTIMER_STxCAR值时，翻转到0并连续计数。
2:0	CNTCKDIV[2:0]	计数器时钟分频 该位域可以由软件配置，确定超高分辨率时钟（SHRTIMER_HPCK）和计数器时钟（SHRTIMER_PSCCK）的分频比。 当 SHRTIMER_MTACTL 中的 CNTCKDIV [3]为 0 时，$f_{SHRTIMER_PSCCK} = f_{SHRTIMER_HPCK} / 2^{CNTCKDIV[2:0]+1}$。 当 SHRTIMER_MTACTL 中的 CNTCKDIV [3]位为 1，且 CNTCKDIV[2:0]配置为 3'b000 时：$f_{SHRTIMER_PSSCK} = f_{SHRTIMER_HPCK}$ 0000：$f_{SHRTIMER_PSSCK} = f_{SHRTIMER_HPCK} / 2$ 0001：$f_{SHRTIMER_PSSCK} = f_{SHRTIMER_HPCK} / 4$ 0010：$f_{SHRTIMER_PSSCK} = f_{SHRTIMER_HPCK} / 8$ 0011：$f_{SHRTIMER_PSSCK} = f_{SHRTIMER_HPCK} / 16$ 0100：$f_{SHRTIMER_PSSCK} = f_{SHRTIMER_HPCK} / 32$ 0101：$f_{SHRTIMER_PSSCK} = f_{SHRTIMER_HPCK} / 64$ 0110：$f_{SHRTIMER_PSSCK} = f_{SHRTIMER_HPCK} / 128$ 0111：$f_{SHRTIMER_PSSCK} = f_{SHRTIMER_HPCK} / 256$ 1000：$f_{SHRTIMER_PSSCK} = f_{SHRTIMER_HPCK}$ 其他值保留。 注意：一旦使能定时器（SHRTIMER_MTCTL0 寄存器中的 STxCEN =1），就不能修改 CNTCKDIV [3:0]位域。

Slave_TIMERx 中断标志寄存器 (SHRTIMER_STxINTF)

地址偏移：0x04

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留										CH1F	CH0F	保留		BLNIF	CBLNF
										r	r			r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	DLYIIF	RSTIF	CH1ONAIF	CH1OAIF	CH0ONAIF	CH0OAIF	CAP1IF	CAP0IF	UPIF	保留	REPIF	CMP3IF	CMP2IF	CMP1IF	CMP0IF
	r	r	r	r	r	r	r	r	r		r	r	r	r	r

位/位域	名称	描述
31:22	保留	必须保持复位值
21	CH1F	通道 1 输出标志 该位用于指示通道 1 的输出电平状态。

		0：通道 1 输出无效电平。 1：通道 1 输出有效电平。
20	CH0F	通道 0 输出标志 该位用于指示通道 0 的输出电平状态。 0：通道 0 输出无效电平。 1：通道 0 输出有效电平。
19:18	保留	必须保持复位值
17	BLNIF	均衡空闲标志 该位用于指示在进入均衡空闲状态时，哪个通道正在输出信号。 0：当进入均衡空闲模式时，通道 0 输出 CH0OPRE 信号，而通道 1 输出无效电平。 1：当进入均衡空闲模式时，通道 1 输出 CH1OPRE 信号，而通道 0 输出无效电平。
16	CBLNF	当前的均衡状态标志 该位仅在均衡模式下有效。该位用于指示当前正在输出信号的通道。 0：通道 0 输出 CH0OPRE 信号，通道 1 输出无效电平。 1：通道 1 输出 CH1OPRE 信号，通道 0 输出无效电平。
15	保留	必须保持复位值
14	DLYIIF	延迟空闲模式进入中断标志 进入延迟空闲或均衡空闲模式时，此标志由硬件置位。 0：延迟空闲模式进入中断未发生 1：延迟空闲模式进入中断发生
13	RSTIF	计数器复位中断标志 计数器复位或翻转事件发生时，此标志由硬件置位。 0：计数器复位或翻转事件中断未发生 1：计数器复位或翻转事件中断发生
12	CH1ONAIF	通道 1 输出无效中断标志 请参阅 CH0ONAIF 说明。
11	CH1OAIF	通道 1 输出有效中断标志 请参阅 CH0OAIF 说明。
10	CH0ONAIF	通道 0 输出无效中断标志 当通道 0 输出无效（C0OPRE 从有效变为无效）发生时，该标志由硬件置位。 0：通道 0 输出无效中断未发生 1：通道 0 输出无效中断发生
9	CH0OAIF	通道 0 输出有效中断标志 当通道 0 输出有效（C0OPRE 从无效变为有效）发生时，该标志由硬件置位。 0：通道 0 输出有效中断未发生 1：通道 0 输出有效中断发生
8	CAP1IF	捕获 1 中断标志 当捕获 1 事件发生，该标志由硬件置位。

		0：捕获 1 中断未发生 1：捕获 1 中断发生
7	CAP0IF	捕获 0 中断标志 当捕获 0 事件发生，该标志由硬件置位。 0：捕获 0 中断未发生 1：捕获 0 中断发生
6	UPIF	更新中断标志 当更新事件发生，该标志由硬件置位。 0：更新中断未发生 1：更新中断发生
5	保留	必须保持复位值
4	REPIF	重复中断标志 当重复事件发生时，此标志由硬件置位。 0：重复中断未发生 1：重复中断发生
3	CMP3IF	比较 3 中断标志 当比较 3 事件发生，该标志由硬件置位。 0：比较 3 中断未发生 1：比较 3 中断发生
2	CMP2IF	比较 2 中断标志 当比较 2 事件发生，该标志由硬件置位。 0：比较 2 中断未发生 1：比较 2 中断发生
1	CMP1IF	比较 1 中断标志 当比较 1 事件发生，该标志由硬件置位。 0：比较 1 中断未发生 1：比较 1 中断发生
0	CMP0IF	比较 0 中断标志 当比较 2 事件发生，该标志由硬件置位。 0：比较 0 中断未发生 1：比较 0 中断发生

Slave_TIMERx 中断标志清除寄存器 (SHRTIMER_STxINTC)

地址偏移：0x08

复位值：0x0000 0000

该寄存器只能进行字（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	DLYIIFC	RSTIFC	CH1ONAIFC	CH1OAIFC	CH0ONAIFC	CH0OAIFC	CAP1IFC	CAP0IFC	UPIFC	保留	REPIFC	CMP3IFC	CMP2IFC	CMP1IFC	CMP0IFC
	w	w	w	w	w	w	w	w	w		w	w	w	w	w

位/位域	名称	描述
31:15	保留	必须保持复位值
14	DLYIIFC	延迟空闲模式进入中断标志清除 0：无效 1：清除延迟空闲模式进入中断标志（SHRTIMER_STxINTF 寄存器中的 DLYIIF 位）
13	RSTIFC	计数器复位中断标志清除 0：无效 1：清除计数器复位中断标志（SHRTIMER_STxINTF 寄存器中的 RSTIF 位）
12	CH1ONAIFC	通道 1 输出无效中断标志清除 清除 SHRTIMER_STxINTF 寄存器中的 CH1ONAIF 位 请参考 CH0ONAIFC 的描述
11	CH1OAIFC	通道 1 输出有效中断标志清除 清除 SHRTIMER_STxINTF 寄存器中的 CH1OAIF 位 请参考 CH0OAIFC 的描述
10	CH0ONAIFC	通道 0 输出无效中断标志清除 0：无效 1：清除通道 0 输出无效中断标志（SHRTIMER_STxINTF 寄存器中的 CH0ONAIF 位）
9	CH0OAIFC	通道 0 输出有效中断标志清除 0：无效 1：清除通道 0 输出有效中断标志（SHRTIMER_STxINTF 寄存器中的 CH0OAIF 位）
8	CAP1IFC	比较 1 捕获中断标志清除 0：无效 1：清除比较 1 捕获中断标志（SHRTIMER_STxINTF 寄存器中的 CAP1IF 位）
7	CAP0IFC	比较 0 捕获中断标志清除 0：无效 1：清除比较 0 捕获中断标志（SHRTIMER_STxINTF 寄存器中的 CAP0IF 位）
6	UPIFC	更新中断标志清除 0：无效 1：清除更新中断标志（SHRTIMER_STxINTF 寄存器中的 UPIF 位）
5	保留	必须保持复位值
4	REPIFC	重复中断标志清除 0：无效

		1：清除重复中断标志（SHRTIMER_STxINTF 寄存器中的 REPIF 位）
3	CMP3IFC	比较 3 中断标志清除 0：无效 1：清除比较 3 中断标志（SHRTIMER_STxINTF 寄存器中的 CMP3IF 位）
2	CMP2IFC	比较 2 中断标志清除 0：无效 1：清除比较 2 中断标志（SHRTIMER_STxINTF 寄存器中的 CMP2IF 位）
1	CMP1IFC	比较 1 中断标志清除 0：无效 1：清除比较 1 中断标志（SHRTIMER_STxINTF 寄存器中的 CMP1IF 位）
0	CMP0IFC	比较 0 中断标志清除 0：无效 1：清除比较 0 中断标志（SHRTIMER_STxINTF 寄存器中的 CMP0IF 位）

Slave_TIMERx DMA 和中断使能寄存器 (SHRTIMER_STxDMAINTEN)

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	DLYIDEN	RSTDEN	CH1ONADEN	CH1OADEN	CH0ONADEN	CH0OADEN	CAP1DEN	CAP0DEN	UPDEN	保留	REPDEN	CMP3DEN	CMP2DEN	CMP1DEN	CMP0DEN
	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	DLYIIE	RSTIE	CH1ONAIE	CH1OAIE	CH0ONAIE	CH0OAIE	CAP1IE	CAP0IE	UPIE	保留	REPIE	CMP3IE	CMP2IE	CMP1IE	CMP0IE
	rw	rw	rw	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	DLYIDEN	延迟空闲模式进入 DMA 请求使能 0：禁能 1：使能
29	RSTDEN	计数器复位 DMA 请求使能 0：禁能 1：使能
28	CH1ONADEN	通道 1 输出无效 DMA 请求使能 请参考 CH0ONADEN 位描述。
27	CH1OADEN	通道 1 输出有效 DMA 请求使能

		请参考 CH0OADEN 位描述。
26	CH0ONADEN	通道 0 输出无效 DMA 请求使能 0：禁能 1：使能
25	CH0OADEN	通道 0 输出有效 DMA 请求使能 0：禁能 1：使能
24	CAP1DEN	捕获 1 DMA 请求使能 0：禁能 1：使能
23	CAP0DEN	捕获 0 DMA 请求使能 0：禁能 1：使能
22	UPDEN	更新 DMA 请求使能 0：禁能 1：使能
21	保留	必须保持复位值
20	REPDEN	重复 DMA 请求使能 0：禁能 1：使能
19	CMP3DEN	比较 3 DMA 请求使能 0：禁能 1：使能
18	CMP2DEN	比较 2 DMA 请求使能 0：禁能 1：使能
17	CMP1DEN	比较 1 DMA 请求使能 0：禁能 1：使能
16	CMP0DEN	比较 0 DMA 请求使能 0：禁能 1：使能
15	保留	必须保持复位值
14	DLYIIE	延迟空闲模式进入中断使能 0：禁能 1：使能
13	RSTIE	计数器复位中断使能

		0：禁能 1：使能
12	CH1ONAIE	通道 1 输出无效中断使能 请参考 CH0ONAIE 位描述。
11	CH1OAIE	通道 1 输出有效中断使能 请参考 CH0OAIE 位描述。
10	CH0ONAIE	通道 0 输出无效中断使能 0：禁能 1：使能
9	CH0OAIE	通道 0 输出有效中断使能 0：禁能 1：使能
8	CAP1IE	捕获 1 中断使能 0：禁能 1：使能
7	CAP0IE	捕获 0 中断使能 0：禁能 1：使能
6	UPIE	更新中断使能 0：禁能 1：使能
5	保留	必须保持复位值
4	REPIE	重复中断使能 0：禁能 1：使能
3	CMP3IE	比较 3 中断使能 0：禁能 1：使能
2	CMP2IE	比较 2 中断使能 0：禁能 1：使能
1	CMP1IE	比较 1 中断使能 0：禁能 1：使能
0	CMP0IE	比较 0 中断使能 0：禁能 1：使能

Slave_TIMERx 计数器寄存器 (SHRTIMER_STxCNT)

地址偏移：0x1C

复位值：0x0000 0000

该寄存器只能进行字（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]	该位域用于配置当前计数器值。对该位域进行写操作可以更改计数器的值。 仅当 Slave_TIMERx 停止（SHRTIMER_STxCTL0 寄存器中的 STxCEN = 0）时，才能对其进行写操作，更改计数器的值。 注意： （1）计数器时钟分频系数小于 64（CNTCKDIV [3:0] < 5）时，计数器的最低有效位无效，它们不能被写入，读出值为 0。 （2）如果写入该位域的值高于 HRTIM_MPER 寄存器值，则定时器的行为是不可预测的。

Slave_TIMERx 计数器自动重载寄存器 (SHRTIMER_STxCAR)

地址偏移：0x14

复位值：0x0000 FFDF

该寄存器只能进行字（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 该位域定义计数器的自动重载值。该寄存器具有影子寄存器。如果影子寄存器被禁能（SHWEN = 0），它将保存有效寄存器的内容；否则，它将保存影子寄存器的内容。 注意：

（1）最小值必须大于或等于（3*$t_{SHRTIMER_CK}$）。例如：当 CNTCKDIV [3:0] = 4’b0000 时，CARL [15:0]> = 0x60。

（2）最大值必须小于或等于（0xFFFF －1*$t_{SHRTIMER_CK}$）。例如：当 CNTCKDIV [3:0] = 4’b0000 时，CARL [15:0] <= 0xFFDF。

Slave_TIMERx 重复计数寄存器 (SHRTIMER_STxCREP)

地址偏移：0x18

复位值：0x0000 0000

该寄存器只能进行字（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]	重复计数器值 该位域用于定义重复事件的发生率。当重复计数器递减计数到零时，连续模式下即将发生的翻转事件或复位事件将产生一个重复事件。 该寄存器具有影子寄存器。如果影子寄存器被禁能（SHWEN = 0），它将保存有效寄存器的内容；否则，它将保存影子寄存器的内容。

Slave_TIMERx 比较 0 寄存器 (SHRTIMER_STxCMP0V)

地址偏移：0x1C

复位值：0x0000 0000

该寄存器只能进行字（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
CMP0VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CMP0VAL[15:0]	比较 0 值

该位域包含要与计数器进行比较的值。
该寄存器具有影子寄存器。如果影子寄存器被禁能（SHWEN = 0），它将保存有效寄存器的内容；否则，它将保存影子寄存器的内容。
注意：
（1）最小值必须大于或等于 3 个 $t_{SHRTIMER_CK}$。例如：当 CNTCKDIV [3:0] = 4'b0000 时，CARL [15:0] > = 0x60。
（2）最大值必须小于或等于（0xFFFF －1*$t_{SHRTIMER_CK}$）。例如：当 CNTCKDIV [3:0] = 4'b0000 时，CARL [15:0] <= 0xFFDF。

Slave_TIMERx 比较 0 复合寄存器 (SHRTIMER_STxCMP0CP)

地址偏移：0x20
复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP0VAL[15:0]															
rw															

位/位域	名称	描述
31:24	保留	必须保持复位值
23:16	CREP[7:0]	计数器重复值 该位域是 SHRTIMER_STxCREP 寄存器中 CREP[7:0]的别名。
15:0	CMP0VAL[15:0]	比较 0 值 该位域是 SHRTIMER_STxCMP0V 寄存器中 CMP0VAL[15:0]的别名。

Slave_TIMERx 比较 1 寄存器 (SHRTIMER_STxCMP1V)

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能进行字（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
CMP1VAL[15:0]															
rw															

位/位域	名称	描述

31:16	保留	必须保持复位值
15:0	CMP1VAL[15:0]	比较 1 值 该位域包含要与计数器进行比较的值。 该寄存器具有影子寄存器。如果影子寄存器被禁能（SHWEN = 0），它将保存有效寄存器的内容；否则，它将保存影子寄存器的内容。 延迟模式中，有效寄存器的值会重新进行计算。 注意： （1）最小值必须大于或等于 3 个 $t_{SHRTIMER_CK}$。例如：当 CNTCKDIV [3:0] = 4'b0000 时，CARL [15:0] > = 0x60。 （2）最大值必须小于或等于（0xFFFF －1*$t_{SHRTIMER_CK}$）。例如：当 CNTCKDIV [3:0] = 4'b0000 时，CARL [15:0] <= 0xFFDF。

Slave_TIMERx 比较 2 寄存器 (SHRTIMER_STxCMP2V)

地址偏移：0x28

复位值：0x0000 0000

该寄存器只能进行字（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
CMP2VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CMP2VAL[15:0]	比较 2 值 该位域包含要与计数器进行比较的值。 该寄存器具有影子寄存器。如果影子寄存器被禁能（SHWEN = 0），它将保存有效寄存器的内容；否则，它将保存影子寄存器的内容。 注意： （1）最小值必须大于或等于 3 个 $t_{SHRTIMER_CK}$。例如：当 CNTCKDIV [3:0] = 4'b0000 时，CARL [15:0] > = 0x60。 （2）最大值必须小于或等于（0xFFFF －1*$t_{SHRTIMER_CK}$）。例如：当 CNTCKDIV [3:0] = 4'b0000 时，CARL [15:0] <= 0xFFDF。

Slave_TIMERx 比较 3 寄存器 (SHRTIMER_STxCMP3V)

地址偏移：0x2C

复位值：0x0000 0000

该寄存器只能进行字（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
CMP3VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CMP3VAL[15:0]	比较 3 值 该位域包含要与计数器进行比较的值。 该寄存器具有影子寄存器。如果影子寄存器被禁能（SHWEN = 0），它将保存有效寄存器的内容；否则，它将保存影子寄存器的内容。 延迟模式中，有效寄存器的值会重新进行计算。 注意： （1）最小值必须大于或等于 3 个 $t_{SHRTIMER_CK}$。例如：当 CNTCKDIV [3:0] = 4’b0000 时，CARL [15:0] > = 0x60。 （2）最大值必须小于或等于（0xFFFF －1*$t_{SHRTIMER_CK}$）。例如：当 CNTCKDIV [3:0] = 4’b0000 时，CARL [15:0] <= 0xFFDF。

Slave_TIMERx 捕获 0 寄存器 (SHRTIMER_STxCAP0V)

地址偏移：0x30

复位值：0x0000 0000

该寄存器只能进行字（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
CAP0VAL[15:0]															
r															

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CAP0VAL[15:0]	捕获 0 值 该位域保持上一个捕获事件发生时的计数器值，且该位域只读。 注意：计数器时钟分频系数小于 64（CNTCKDIV [3:0] <5）时，计数器的最低有效位无效，它们不能被写入，读出值为 0。

Slave_TIMERx 捕获 1 寄存器 (SHRTIMER_STxCAP1V)

地址偏移：0x34

复位值：0x0000 0000

该寄存器只能进行字（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
CAP1VAL[15:0]															
r															

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CAP1VAL[15:0]	捕获 1 值 该位域保持上一个捕获事件发生时的计数器值，且该位域只读。 注意：计数器时钟分频系数小于 64（CNTCKDIV [3:0] <5）时，计数器的最低有效位无效，它们不能被写入，读出值为 0。

Slave_TIMERx 死区控制寄存器 (SHRTIMER_STxDTCTL)

地址偏移：0x38

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DTFSVPROT	DTFSPROT	保留				DTFS	DTFCFG[8:0]								
rwo	rwo					rw	rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTRSVPROT	DTRSPROT	DTGCKDIV[3:0]				DTRS	DTRCFG[8:0]								
rwo	rwo	rw				rw	rw								

位/位域	名称	描述
31	DTFSVPROT	死区下降沿（值和符号）保护 该位域用于死区下降沿（值和符号）的写保护。 0：保护禁能。DTFS 位和 DTFCFG [15:0]位域是可写的。 1：保护使能。DTFS 位和 DTFCFG [15:0]位域是只读的。 注意： （1）DTFCFG [15:9]位域在 SHRTIMER_STxACTL 寄存器中。 （2）该位不进行预装载。
30	DTFSPROT	死区下降沿（符号）保护 该位域用于死区下降沿（符号）的写保护。 0：保护禁能。SHRTIMER_STxDTCTL 寄存器中的 DTFS 位是可写的。 1：保护使能。SHRTIMER_STxDTCTL 寄存器中的 DTFS 位是只读的。 注意：

（1）该位不进行预装载。

29:26	保留	必须保持复位值
25	DTFS	死区下降沿值的符号 0：死区下降沿值的符号为正。 1：死区下降沿值的符号为负。 注意：当 SHRTIMER_STxDTCTL 寄存器中的 DTFSPROT 位或 DTFSVPROT 位置 1 时，无法修改此位。
24:16	DTFCFG[8:0]	死区下降沿值 该位域用于配置跟随输出准备信号（OyPRE，y = 0,1）下降沿之后的死区时间值。 DTF 值 = DTFCFG [15:0] x $t_{SHRTIMER_DTGCK}$，其中，$t_{SHRTIMER_DTGCK}$ = 1/$f_{SHRTIMER_DTGCK}$。 写入该位域可以更改 DTFCFG [15:0]位域的低 9 位。 注意： （1）DTFCFG [15:9] 位域在 SHRTIMER_STxACTL 寄存器中。 （2）当 SHRTIMER_STxDTCTL 寄存器中的 DTFSVPROT 位置 1 时，无法修改此位域。
15	DTRSVPROT	死区上升沿（值和符号）保护 该位域用于死区上升沿（值和符号）的写保护。 0：保护禁能。DTRS 位和 DTRCFG [15:0]位域是可写的。 1：保护使能。DTRS 位和 DTRCFG [15:0]位域是只读的。 注意： （1）DTRCFG [15:9]位域在 SHRTIMER_STxACTL 寄存器中。 （2）该位不进行预装载。
14	DTRSPROT	死区上升沿（符号）保护 该位域用于死区上升沿（符号）的写保护。 0：保护禁能。SHRTIMER_STxDTCTL 寄存器中的 DTRS 位是可写的。 1：保护使能。SHRTIMER_STxDTCTL 寄存器中的 DTRS 位是只读的。 注意： （1）该位不进行预装载。
13:10	DTGCKDIV[2:0]	死区时间发生器时钟分频 该位域可以通过软件配置，用于确定 SHRTIMER 时钟（SHRTIMER_CK）和死区时间发生器时钟（SHRTIMER_DTGCK）的分频系数。 DTGCKDIV[3] = 0 时，$f_{SHRTIMER_DTGCK}$ = (8*$f_{SHRTIMER_CK}$)/$2^{DTGCKDIV[2:0]}$。 DTGCKDIV[3] = 1 时，$f_{SHRTIMER_DTGCK}$ = $2^{(DTGCKDIV[2:0]+4)}$*$f_{SHRTIMER_CK}$。 0000：$f_{SHRTIMER_DTGCK}$ = 8*$f_{SHRTIMER_CK}$ 0001：$f_{SHRTIMER_DTGCK}$ = (8*$f_{SHRTIMER_CK}$)/2 0010：$f_{SHRTIMER_DTGCK}$ = (8*$f_{SHRTIMER_CK}$)/4 0011：$f_{SHRTIMER_DTGCK}$ = (8*$f_{SHRTIMER_CK}$)/8 0100：$f_{SHRTIMER_DTGCK}$ = (8*$f_{SHRTIMER_CK}$)/16 0101：$f_{SHRTIMER_DTGCK}$ = (8*$f_{SHRTIMER_CK}$)/32 0110：$f_{SHRTIMER_DTGCK}$ = (8*$f_{SHRTIMER_CK}$)/64

		0111：$f_{SHRTIMER_DTGCK}$ = (8*$f_{SHRTIMER_CK}$)/128 1000：$f_{SHRTIMER_DTGCK}$ = 16* $f_{SHRTIMER_CK}$ 1001：$f_{SHRTIMER_DTGCK}$ = 32* $f_{SHRTIMER_CK}$ 1010：$f_{SHRTIMER_DTGCK}$ = 64* $f_{SHRTIMER_CK}$ 其他位保留。 注意：如果任何一个保护位（DTFSPROT，DTFSVPROT，DTRSPROT 和 DTRSVPROT）置位，则不能修改此位域。
9	DTRS	死区上升沿值的符号 0：死区上升沿值的符号为正。 1：死区上升沿值的符号为负。 注意：当 SHRTIMER_STxDTCTL 寄存器中的 DTRSPROT 位或 DTRSVPROT 位置 1 时，无法修改此位。
8:0	DTRCFG[8:0]	死区上升沿值 该位域用于配置跟随输出准备信号（OyPRE，y = 0,1）上升沿之后的死区时间值。DTR 值 = DTRCFG[15:0] x $t_{SHRTIMER_DTGCK}$，其中，$t_{SHRTIMER_DTGCK}$ = 1/ $f_{SHRTIMER_DTGCK}$。 写入该位域可以更改 DTRCFG[15:0]位域的低 9 位。 注意： （1）DTRCFG [15:9]位域在 SHRTIMER_STxACTL 寄存器中。 （2）当 SHRTIMER_STxDTCTL 寄存器中的 DTRSVPROT 位置 1 时，无法修改此位域。

Slave_TIMERx 通道 0 置位请求寄存器 (SHRTIMER_STxCH0SET)

地址偏移：0x3C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH0SUP	CH0SEXEV9	CH0SEXEV8	CH0SEXEV7	CH0SEXEV6	CH0SEXEV5	CH0SEXEV4	CH0SEXEV3	CH0SEXEV2	CH0SEXEV1	CH0SEXEV0	CH0SSTEV8	CH0SSTEV7	CH0SSTEV6	CH0SSTEV5	CH0SSTEV4
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0SSTEV3	CH0SSTEV2	CH0SSTEV1	CH0SSTEV0	CH0SMTCMP3	CH0SMTCMP2	CH0SMTCMP1	CH0SMTCMP0	CH0SMTPER	CH0SCMP3	CH0SCMP2	CH0SCMP1	CH0SCMP0	CH0SPER	CH0SRST	CH0SSEV
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	CH0SUP	更新事件生成通道 0 置位请求 该位置 1 时，更新事件可以产生置位请求。 0：更新事件不生成置位请求。 1：更新事件生成置位请求。
30	CH0SEXEV9	外部事件 9 生成通道 0 置位请求

请参考 CH0SEXEV0 说明。

29 CH0SEXEV8 外部事件 8 生成通道 0 置位请求
请参考 CH0SEXEV0 说明。

28 CH0SEXEV7 外部事件 7 生成通道 0 置位请求
请参考 CH0SEXEV0 说明。

27 CH0SEXEV6 外部事件 6 生成通道 0 置位请求
请参考 CH0SEXEV0 说明。

26 CH0SEXEV5 外部事件 5 生成通道 0 置位请求
请参考 CH0SEXEV0 说明。

25 CH0SEXEV4 外部事件 4 生成通道 0 置位请求
请参考 CH0SEXEV0 说明。

24 CH0SEXEV3 外部事件 3 生成通道 0 置位请求
请参考 CH0SEXEV0 说明。

23 CH0SEXEV2 外部事件 2 生成通道 0 置位请求
请参考 CH0SEXEV0 说明。

22 CH0SEXEV1 外部事件 1 生成通道 0 置位请求
请参考 CH0SEXEV0 说明。

21 CH0SEXEV0 外部事件 0 生成通道 0 置位请求
该位置 1 时，外部事件 0 可以产生置位请求。
0：该事件不生成置位请求。
1：该事件生成置位请求。

20 CH0SSTEV8 Slave_TIMERx 互连事件 8 生成通道 0 置位请求
请参考 CH0SSTEV0 说明。

19 CH0SSTEV7 Slave_TIMERx 互连事件 7 生成通道 0 置位请求
请参考 CH0SSTEV0 说明。

18 CH0SSTEV6 Slave_TIMERx 互连事件 6 生成通道 0 置位请求
请参考 CH0SSTEV0 说明。

17 CH0SSTEV5 Slave_TIMERx 互连事件 5 生成通道 0 置位请求
请参考 CH0SSTEV0 说明。

16 CH0SSTEV4 Slave_TIMERx 互连事件 4 生成通道 0 置位请求
请参考 CH0SSTEV0 说明。

15 CH0SSTEV3 Slave_TIMERx 互连事件 3 生成通道 0 置位请求
请参考 CH0SSTEV0 说明。

14 CH0SSTEV2 Slave_TIMERx 互连事件 2 生成通道 0 置位请求
请参考 CH0SSTEV0 说明。

13 CH0SSTEV1 Slave_TIMERx 互连事件 1 生成通道 0 置位请求

请参考 CH0SSTEV0 说明。

12	CH0SSTEV0	Slave_TIMERx 互连事件 0 生成通道 0 置位请求 该位置 1 时，Slave_TIMERx 互连事件 0 可以产生置位请求。具体请参考***表 19-5. Slave_TIMER 内部连接事件***。 0：该事件不生成置位请求。 1：该事件生成置位请求。
11	CH0SMTCMP3	Master_TIMER 比较 3 事件生成通道 0 置位请求 该位置 1 时，Master_TIMER 比较 3 事件可以产生置位请求。 0：Master_TIMER 比较 3 事件不生成置位请求。 1：Master_TIMER 比较 3 事件生成置位请求。
10	CH0SMTCMP2	Master_TIMER 比较 2 事件生成通道 0 置位请求 该位置 1 时，Master_TIMER 比较 2 事件可以产生置位请求。 0：Master_TIMER 比较 2 事件不生成置位请求。 1：Master_TIMER 比较 2 事件生成置位请求。
9	CH0SMTCMP1	Master_TIMER 比较 1 事件生成通道 0 置位请求 该位置 1 时，Master_TIMER 比较 1 事件可以产生置位请求。 0：Master_TIMER 比较 1 事件不生成置位请求。 1：Master_TIMER 比较 1 事件生成置位请求。
8	CH0SMTCMP0	Master_TIMER 比较 0 事件生成通道 0 置位请求 该位置 1 时，Master_TIMER 比较 0 事件可以产生置位请求。 0：Master_TIMER 比较 0 事件不生成置位请求。 1：Master_TIMER 比较 0 事件生成置位请求。
7	CH0SMTPER	Master_TIMER 周期事件生成通道 0 置位请求 连续模式下，Master_TIMER 计数器的翻转事件可以产生置位请求。在单脉冲模式下，Master_TIMER 计数器的复位事件可以产生置位请求。 0：该事件不生成置位请求。 1：该事件生成置位请求。
6	CH0SCMP3	Slave_TIMERx 比较 3 事件生成通道 0 置位请求 该位置 1 时，Slave_TIMERx 比较 3 事件可以产生置位请求。 0：Slave_TIMERx 比较 3 事件不生成置位请求。 1：Slave_TIMERx 比较 3 事件生成置位请求。
5	CH0SCMP2	Slave_TIMERx 比较 2 事件生成通道 0 置位请求 该位置 1 时，Slave_TIMERx 比较 2 事件可以产生置位请求。 0：Slave_TIMERx 比较 2 事件不生成置位请求。 1：Slave_TIMERx 比较 2 事件生成置位请求。
4	CH0SCMP1	Slave_TIMERx 比较 1 事件生成通道 0 置位请求 该位置 1 时，Slave_TIMERx 比较 1 事件可以产生置位请求。 0：Slave_TIMERx 比较 1 事件不生成置位请求。 1：Slave_TIMERx 比较 1 事件以生成置位请求。

3	CH0SCMP0	Slave_TIMERx 比较 0 事件生成通道 0 置位请求 该位置 1 时，Slave_TIMERx 比较 0 事件可以产生置位请求。 0：Slave_TIMERx 比较 0 事件不生成置位请求。 1：Slave_TIMERx 比较 0 事件生成置位请求。
2	CH0SPER	Slave_TIMERx 周期事件生成通道 0 置位请求 该位置 1 时，Slave_TIMERx 周期事件可以产生置位请求。 0：Slave_TIMERx 周期事件不生成置位请求。 1：Slave_TIMERx 周期事件生成置位请求。
1	CH0SRST	Slave_TIMERx 复位事件生成通道 0 置位请求 该位置 1 时，由软件和同步输入引起的 Slave_TIMERx 复位事件，生成通道 0 置位请求。 0：该事件不生成置位请求。 1：该事件生成置位请求。 注意：该位置 1 时，其他的定时器复位事件不会影响输出。
0	CH0SSEV	软件事件生成通道 0 置位请求 该位由软件置 1，由硬件自动清除。该位置 1 时，生成通道 0 置位请求。 0：该事件不生成置位请求 1：该事件生成置位请求 注意：该位不进行预装载。

Slave_TIMERx 通道 0 复位请求寄存器 (SHRTIMER_STxCH0RST)

地址偏移：0x40
复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH0RSUP	CH0RSEXEV9	CH0RSEXEV8	CH0RSEXEV7	CH0RSEXEV6	CH0RSEXEV5	CH0RSEXEV4	CH0RSEXEV3	CH0RSEXEV2	CH0RSEXEV1	CH0RSEXEV0	CH0RSSTEV8	CH0RSSTEV7	CH0RSSTEV6	CH0RSSTEV5	CH0RSSTEV4
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0RSSTEV3	CH0RSSTEV2	CH0RSSTEV1	CH0RSSTEV0	CH0RSMTCMP3	CH0RSMTCMP2	CH0RSMTCMP1	CH0RSMTCMP0	CH0RSMTPER	CH0RSCMP3	CH0RSCMP2	CH0RSCMP1	CH0RSCMP0	CH0RSPER	CH0RSRST	CH0RSSEV
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	CH0RSUP	更新事件生成通道 0 复位请求 该位置 1 时，更新事件可以产生复位请求。 0：更新事件不生成复位请求。 1：更新事件生成复位请求。
30	CH0RSEXEV9	外部事件 9 生成通道 0 复位请求 请参考 CH0RSEXEV0 说明。

29	CH0RSEXEV8	外部事件 8 生成通道 0 复位请求 请参考 CH0RSEXEV0 说明。
28	CH0RSEXEV7	外部事件 7 生成通道 0 复位请求 请参考 CH0RSEXEV0 说明。
27	CH0RSEXEV6	外部事件 6 生成通道 0 复位请求 请参考 CH0RSEXEV0 说明。
26	CH0RSEXEV5	外部事件 5 生成通道 0 复位请求 请参考 CH0RSEXEV0 说明。
25	CH0RSEXEV4	外部事件 4 生成通道 0 复位请求 请参考 CH0RSEXEV0 说明。
24	CH0RSEXEV3	外部事件 3 生成通道 0 复位请求 请参考 CH0RSEXEV0 说明。
23	CH0RSEXEV2	外部事件 2 生成通道 0 复位请求 请参考 CH0RSEXEV0 说明。
22	CH0RSEXEV1	外部事件 1 生成通道 0 复位请求 请参考 CH0RSEXEV0 说明。
21	CH0RSEXEV0	外部事件 0 生成通道 0 复位请求 当该位置 1 时，外部事件 0 可以产生复位请求。 0：该事件不生成复位请求。 1：该事件生成复位请求。
20	CH0RSSTEV8	Slave_TIMERx 互连事件 8 生成通道 0 复位请求 请参考 CH0RSSTEV0 说明。
19	CH0RSSTEV7	Slave_TIMERx 互连事件 7 生成通道 0 复位请求 请参考 CH0RSSTEV0 说明。
18	CH0RSSTEV6	Slave_TIMERx 互连事件 6 生成通道 0 复位请求 请参考 CH0RSSTEV0 说明。
17	CH0RSSTEV5	Slave_TIMERx 互连事件 5 生成通道 0 复位请求 请参考 CH0RSSTEV0 说明。
16	CH0RSSTEV4	Slave_TIMERx 互连事件 4 生成通道 0 复位请求 请参考 CH0RSSTEV0 说明。
15	CH0RSSTEV3	Slave_TIMERx 互连事件 3 生成通道 0 复位请求 请参考 CH0RSSTEV0 说明。
14	CH0RSSTEV2	Slave_TIMERx 互连事件 2 生成通道 0 复位请求 请参考 CH0RSSTEV0 说明。
13	CH0RSSTEV1	Slave_TIMERx 互连事件 1 生成通道 0 复位请求 请参考 CH0RSSTEV0 说明。

12	CH0RSSTEV0	Slave_TIMERx 互连事件 0 生成通道 0 复位请求 该位置 1 时，Slave_TIMERx 互连事件 0 可以产生复位请求。具体请参考***表 19-5. Slave_TIMER 内部连接事件***。 0：该事件不生成复位请求。 1：该事件生成复位请求。
11	CH0RSMTCMP3	Master_TIMER 比较 3 事件生成通道 0 复位请求 该位置 1 时，Master_TIMER 比较 3 事件可以产生复位请求。 0：Master_TIMER 比较 3 事件不生成复位请求。 1：Master_TIMER 比较 3 事件生成复位请求。
10	CH0RSMTCMP2	Master_TIMER 比较 2 事件生成通道 0 复位请求 该位置 1 时，Master_TIMER 比较 2 事件可以产生复位请求。 0：Master_TIMER 比较 2 事件不生成复位请求。 1：Master_TIMER 比较 2 事件生成复位请求。
9	CH0RSMTCMP1	Master_TIMER 比较 1 事件生成通道 0 复位请求 该位置 1 时，Master_TIMER 比较 1 事件可以产生复位请求。 0：Master_TIMER 比较 1 事件不生成复位请求。 1：Master_TIMER 比较 1 事件生成复位请求。
8	CH0RSMTCMP0	Master_TIMER 比较 0 事件生成通道 0 复位请求 该位置 1 时，Master_TIMER 比较 0 事件可以产生复位请求。 0：Master_TIMER 比较 0 事件不生成复位请求。 1：Master_TIMER 比较 0 事件生成复位请求。
7	CH0RSMTPER	Master_TIMER 周期事件生成通道 0 复位请求 连续模式下，Master_TIMER 计数器翻转事件可以产生复位请求。在单脉冲模式下，Master_TIMER 计数器复位事件可以产生复位请求。 0：该事件不生成复位请求。 1：该事件生成复位请求。
6	CH0RSCMP3	Slave_TIMERx 比较 3 事件生成通道 0 复位请求 该位置 1 时，Slave_TIMERx 比较 3 事件可以产生复位请求。 0：Slave_TIMERx 比较 3 事件不生成复位请求。 1：Slave_TIMERx 比较 3 事件生成复位请求。
5	CH0RSCMP2	Slave_TIMERx 比较 2 事件生成通道 0 复位请求 该位置 1 时，Slave_TIMERx 比较 2 事件可以产生复位请求。 0：Slave_TIMERx 比较 2 事件不生成复位请求。 1：Slave_TIMERx 比较 2 事件生成复位请求。
4	CH0RSCMP1	Slave_TIMERx 比较 1 事件生成通道 0 复位请求 该位置 1 时，Slave_TIMERx 比较 1 事件可以产生复位请求。 0：Slave_TIMERx 比较 1 事件不生成复位请求。 1：Slave_TIMERx 比较 1 事件生成复位请求。
3	CH0RSCMP0	Slave_TIMERx 比较 0 事件生成通道 0 复位请求 该位置 1 时，Slave_TIMERx 比较 0 事件可以产生复位请求。

		0：Slave_TIMERx 比较 0 事件不生成复位请求。 1：Slave_TIMERx 比较 0 事件生成复位请求。
2	CH0RSPER	Slave_TIMERx 周期事件生成通道 0 复位请求 该位置 1 时，Slave_TIMERx 周期事件可以产生复位请求。 0：Slave_TIMERx 周期事件不生成复位请求。 1：Slave_TIMERx 周期事件生成复位请求。
1	CH0RSRST	Slave_TIMERx 复位事件生成通道 0 复位请求 该位置 1 时，由软件和同步输入引起的 Slave_TIMERx 复位事件，生成通道 0 复位请求。 0：该事件不生成复位请求。 1：该事件生成复位请求。 注意：该位置 1 时，其他的定时器复位事件不会影响输出。
0	CH0RSSEV	软件事件生成通道 0 复位请求 该位由软件置 1，由硬件自动清除。该位置 1 时，生成通道 0 复位请求。 0：该事件不生成复位请求 1：该事件生成复位请求 注意：该位不进行预装载。

Slave_TIMERx 通道 1 置位请求寄存器 (SHRTIMER_STxCH1SET)

地址偏移：0x44

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH1SUP	CH1SEXEV9	CH1SEXEV8	CH1SEXEV7	CH1SEXEV6	CH1SEXEV5	CH1SEXEV4	CH1SEXEV3	CH1SEXEV2	CH1SEXEV1	CH1SEXEV0	CH1SSTEV8	CH1SSTEV7	CH1SSTEV6	CH1SSTEV5	CH1SSTEV4
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1SSTEV3	CH1SSTEV2	CH1SSTEV1	CH1SSTEV0	CH1SMTCMP3	CH1SMTCMP2	CH1SMTCMP1	CH1SMTCMP0	CH1SMTPER	CH1SCMP3	CH1SCMP2	CH1SCMP1	CH1SCMP0	CH1SPER	CH1SRST	CH1SSEV
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	CH1SUP	更新事件生成通道 1 置位请求 该位置 1 时，更新事件可以产生置位请求。 0：更新事件不生成置位请求。 1：更新事件生成置位请求。
30	CH1SEXEV9	外部事件 9 生成通道 1 置位请求 请参考 CH1SEXEV0 说明。
29	CH1SEXEV8	外部事件 8 生成通道 1 置位请求

请参考 CH1SEXEV0 说明。

28 CH1SEXEV7 外部事件 7 生成通道 1 置位请求
请参考 CH1SEXEV0 说明。

27 CH1SEXEV6 外部事件 6 生成通道 1 置位请求
请参考 CH1SEXEV0 说明。

26 CH1SEXEV5 外部事件 6 生成通道 1 置位请求
请参考 CH1SEXEV0 说明。

25 CH1SEXEV4 外部事件 4 生成通道 1 置位请求
请参考 CH1SEXEV0 说明。

24 CH1SEXEV3 外部事件 3 生成通道 1 置位请求
请参考 CH1SEXEV0 说明。

23 CH1SEXEV2 外部事件 2 生成通道 1 置位请求
请参考 CH1SEXEV0 说明。

22 CH1SEXEV1 外部事件 1 生成通道 1 置位请求
请参考 CH1SEXEV0 说明。

21 CH1SEXEV0 外部事件 0 生成通道 1 置位请求
当该位置 1 时，外部事件 1 可以产生置位请求。
0：该事件不生成置位请求。
1：该事件生成置位请求。

20 CH1SSTEV8 Slave_TIMERx 互连事件 8 生成通道 1 置位请求
请参考 CH1SSTEV0 说明。

19 CH1SSTEV7 Slave_TIMERx 互连事件 7 生成通道 1 置位请求
请参考 CH1SSTEV0 说明。

18 CH1SSTEV6 Slave_TIMERx 互连事件 6 生成通道 1 置位请求
请参考 CH1SSTEV0 说明。

17 CH1SSTEV5 Slave_TIMERx 互连事件 5 生成通道 1 置位请求
请参考 CH1SSTEV0 说明。

16 CH1SSTEV4 Slave_TIMERx 互连事件 4 生成通道 1 置位请求
请参考 CH1SSTEV0 说明。

15 CH1SSTEV3 Slave_TIMERx 互连事件 3 生成通道 1 置位请求
请参考 CH1SSTEV0 说明。

14 CH1SSTEV2 Slave_TIMERx 互连事件 2 生成通道 1 置位请求
请参考 CH1SSTEV0 说明。

13 CH1SSTEV1 Slave_TIMERx 互连事件 1 生成通道 1 置位请求
请参考 CH1SSTEV0 说明。

12 CH1SSTEV0 Slave_TIMERx 互连事件 0 生成通道 1 置位请求

		该位置 1 时，Slave_TIMERx 互连事件 0 可以产生置位请求。具体请参考*表 19-5. Slave_TIMER 内部连接事件*。 0：该事件不生成置位请求。 1：该事件生成置位请求。
11	CH1SMTCMP3	Master_TIMER 比较 3 事件生成通道 1 置位请求 该位置 1 时，Master_TIMER 比较 3 事件可以产生置位请求。 0：Master_TIMER 比较 3 事件不生成置位请求。 1：Master_TIMER 比较 3 事件生成置位请求。
10	CH1SMTCMP2	Master_TIMER 比较 2 事件生成通道 1 置位请求 该位置 1 时，Master_TIMER 比较 2 事件可以产生置位请求。 0：Master_TIMER 比较 2 事件不生成置位请求。 1：Master_TIMER 比较 2 事件生成置位请求。
9	CH1SMTCMP1	Master_TIMER 比较 1 事件生成通道 1 置位请求 该位置 1 时，Master_TIMER 比较 1 事件可以产生置位请求。 0：Master_TIMER 比较 1 事件不生成置位请求。 1：Master_TIMER 比较 1 事件生成置位请求。
8	CH1SMTCMP0	Master_TIMER 比较 0 事件生成通道 1 置位请求 该位置 1 时，Master_TIMER 比较 0 事件可以产生置位请求。 0：Master_TIMER 比较 0 事件不生成置位请求。 1：Master_TIMER 比较 0 事件生成置位请求。
7	CH1SMTPER	Master_TIMER 周期事件生成通道 1 置位请求 连续模式下，Master_TIMER 计数器翻转事件可以产生置位请求。在单脉冲模式下，Master_TIMER 计数器复位事件可以产生置位请求。 0：该事件不生成置位请求。 1：该事件生成置位请求。
6	CH1SCMP3	Slave_TIMERx 比较 3 事件生成通道 1 置位请求 该位置 1 时，Slave_TIMERx 比较 3 事件可以产生置位请求。 0：Slave_TIMERx 比较 3 事件不生成置位请求。 1：Slave_TIMERx 比较 3 事件生成置位请求。
5	CH1SCMP2	Slave_TIMERx 比较 2 事件生成通道 1 置位请求 该位置 1 时，Slave_TIMERx 比较 2 事件可以产生置位请求。 0：Slave_TIMERx 比较 2 事件不生成置位请求。 1：Slave_TIMERx 比较 2 事件生成置位请求。
4	CH1SCMP1	Slave_TIMERx 比较 1 事件生成通道 1 置位请求 该位置 1 时，Slave_TIMERx 比较 1 事件可以产生置位请求。 0：Slave_TIMERx 比较 1 事件不生成置位请求。 1：Slave_TIMERx 比较 1 事件生成置位请求。
3	CH1SCMP0	Slave_TIMERx 比较 0 事件生成通道 1 置位请求 该位置 1 时，Slave_TIMERx 比较 0 事件可以产生置位请求。 0：Slave_TIMERx 比较 0 事件不生成置位请求。

		1：Slave_TIMERx 比较 0 事件生成置位请求。
2	CH1SPER	Slave_TIMERx 周期事件生成通道 1 置位请求 该位置 1 时，Slave_TIMERx 周期事件可以产生置位请求。 0：Slave_TIMERx 周期事件不生成置位请求。 1：Slave_TIMERx 周期事件生成置位请求。
1	CH1SRST	Slave_TIMERx 复位事件生成通道 1 置位请求 该位置 1 时，由软件和同步输入引起的 Slave_TIMERx 复位事件，生成通道 1 置位请求。 0：该事件不生成置位请求。 1：该事件生成置位请求。 注意：该位置 1 时，其他的定时器复位事件不会影响输出。
0	CH1SSEV	软件事件生成通道 1 置位请求 该位由软件置 1，由硬件自动清除。该位置 1 时，生成通道 1 置位请求。 0：该事件不生成置位请求 1：该事件生成置位请求 注意：该位不进行预装载。

Slave_TIMERx 通道 1 复位请求寄存器 (SHRTIMER_STxCH1RST)

地址偏移：0x48

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH1RSUP	CH1RSEXEV9	CH1RSEXEV8	CH1RSEXEV7	CH1RSEXEV6	CH1RSEXEV5	CH1RSEXEV4	CH1RSEXEV3	CH1RSEXEV2	CH1RSEXEV1	CH1RSEXEV0	CH1RSSTEV8	CH1RSSTEV7	CH1RSSTEV6	CH1RSSTEV5	CH1RSSTEV4
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1RSSTEV3	CH1RSSTEV2	CH1RSSTEV1	CH1RSSTEV0	CH1RSMTCMP3	CH1RSMTCMP2	CH1RSMTCMP1	CH1RSMTCMP0	CH1RSMTPER	CH1RSCMP3	CH1RSCMP2	CH1RSCMP1	CH1RSCMP0	CH1RSPER	CH1RSRST	CH1RSSEV
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	CH1RSUP	更新事件生成通道 1 复位请求 该位置 1 时，更新事件可以产生复位请求。 0：更新事件不生成复位请求。 1：更新事件生成复位请求。
30	CH1RSEXEV9	外部事件 9 生成通道 1 复位请求 请参考 CH1RSEXEV0 说明。
29	CH1RSEXEV8	外部事件 8 生成通道 1 复位请求 请参考 CH1RSEXEV0 说明。

28	CH1RSEXEV7	外部事件 9 生成通道 1 复位请求 请参考 CH1RSEXEV0 说明。
27	CH1RSEXEV6	外部事件 6 生成通道 1 复位请求 请参考 CH1RSEXEV0 说明。
26	CH1RSEXEV5	外部事件 5 生成通道 1 复位请求 请参考 CH1RSEXEV0 说明。
25	CH1RSEXEV4	外部事件 4 生成通道 1 复位请求 请参考 CH1RSEXEV0 说明。
24	CH1RSEXEV3	外部事件 3 生成通道 1 复位请求 请参考 CH1RSEXEV0 说明。
23	CH1RSEXEV2	外部事件 2 生成通道 1 复位请求 请参考 CH1RSEXEV0 说明。
22	CH1RSEXEV1	外部事件 1 生成通道 1 复位请求 请参考 CH1RSEXEV0 说明。
21	CH1RSEXEV0	外部事件 0 生成通道 1 复位请求 当该位置 1 时，外部事件 0 可以产生复位请求。 0：该事件不生成复位请求。 1：该事件生成复位请求。
20	CH1RSSTEV8	Slave_TIMERx 互连事件 8 生成通道 1 复位请求 请参考 CH1RSSTEV0 说明。
19	CH1RSSTEV7	Slave_TIMERx 互连事件 7 生成通道 1 复位请求 请参考 CH1RSSTEV0 说明。
18	CH1RSSTEV6	Slave_TIMERx 互连事件 6 生成通道 1 复位请求 请参考 CH1RSSTEV0 说明。
17	CH1RSSTEV5	Slave_TIMERx 互连事件 5 生成通道 1 复位请求 请参考 CH1RSSTEV0 说明。
16	CH1RSSTEV4	Slave_TIMERx 互连事件 4 生成通道 1 复位请求 请参考 CH1RSSTEV0 说明。
15	CH1RSSTEV3	Slave_TIMERx 互连事件 3 生成通道 1 复位请求 请参考 CH1RSSTEV0 说明。
14	CH1RSSTEV2	Slave_TIMERx 互连事件 2 生成通道 1 复位请求 请参考 CH1RSSTEV0 说明。
13	CH1RSSTEV1	Slave_TIMERx 互连事件 1 生成通道 1 复位请求 请参考 CH1RSSTEV0 说明。
12	CH1RSSTEV0	Slave_TIMERx 互连事件 0 生成通道 1 复位请求 该位置 1 时，Slave_TIMERx 互连事件 1 可以产生复位请求。具体请参考***表 19-5.***

Slave_TIMER 内部连接事件。

0：该事件不生成复位请求。

1：该事件生成复位请求。

11 CH1RSMTCMP3 Master_TIMER 比较 3 事件生成通道 1 复位请求

该位置 1 时，Master_TIMER 比较 3 事件可以产生复位请求。

0：Master_TIMER 比较 3 事件不生成复位请求。

1：Master_TIMER 比较 3 事件生成复位请求。

10 CH1RSMTCMP2 Master_TIMER 比较 2 事件生成通道 1 复位请求

该位置 1 时，Master_TIMER 比较 2 事件可以产生复位请求。

0：Master_TIMER 比较 2 事件不生成复位请求。

1：Master_TIMER 比较 2 事件生成复位请求。

9 CH1RSMTCMP1 Master_TIMER 比较 1 事件生成通道 1 复位请求

该位置 1 时，Master_TIMER 比较 1 事件可以产生复位请求。

0：Master_TIMER 比较 1 事件不生成复位请求。

1：Master_TIMER 比较 1 事件生成复位请求。

8 CH1RSMTCMP0 Master_TIMER 比较 0 事件生成通道 1 复位请求

该位置 1 时，Master_TIMER 比较 0 事件可以产生复位请求。

0：Master_TIMER 比较 0 事件不生成复位请求。

1：Master_TIMER 比较 0 事件生成复位请求。

7 CH1RSMTPER Master_TIMER 周期事件生成通道 1 复位请求

连续模式下，Master_TIMER 计数器翻转事件可以产生复位请求。在单脉冲模式下，Master_TIMER 计数器复位事件可以产生复位请求。

0：该事件不生成复位请求。

1：该事件生成复位请求。

6 CH1RSCMP3 Slave_TIMERx 比较 3 事件生成通道 1 复位请求

该位置 1 时，Slave_TIMERx 比较 3 事件可以产生复位请求。

0：Slave_TIMERx 比较 3 事件不生成复位请求。

1：Slave_TIMERx 比较 3 事件生成复位请求。

5 CH1RSCMP2 Slave_TIMERx 比较 2 事件生成通道 1 复位请求

该位置 1 时，Slave_TIMERx 比较 2 事件可以产生复位请求。

0：Slave_TIMERx 比较 2 事件不生成复位请求。

1：Slave_TIMERx 比较 2 事件生成复位请求。

4 CH1RSCMP1 Slave_TIMERx 比较 1 事件生成通道 1 复位请求

该位置 1 时，Slave_TIMERx 比较 1 事件可以产生复位请求。

0：Slave_TIMERx 比较 1 事件不生成复位请求。

1：Slave_TIMERx 比较 1 事件生成复位请求。

3 CH1RSCMP0 Slave_TIMERx 比较 0 事件生成通道 1 复位请求

该位置 1 时，Slave_TIMERx 比较 0 事件可以产生复位请求。

0：Slave_TIMERx 比较 0 事件不生成复位请求。

		1：Slave_TIMERx 比较 0 事件生成复位请求。
2	CH1RSPER	Slave_TIMERx 周期事件生成通道 1 复位请求 该位置 1 时，Slave_TIMERx 周期事件可以产生复位请求。 0：Slave_TIMERx 周期事件不生成复位请求。 1：Slave_TIMERx 周期事件生成复位请求。
1	CH1RSRST	Slave_TIMERx 复位事件生成通道 1 复位请求 该位置 1 时，由软件和同步输入引起的 Slave_TIMERx 复位事件，生成通道 1 复位请求。 0：该事件不生成复位请求。 1：该事件生成复位请求。 注意：该位置 1 时，其他的定时器复位事件不会影响输出。
0	CH1RSSEV	软件事件生成通道 1 复位请求 该位由软件置 1，由硬件自动清除。该位置 1 时，生成通道 1 复位请求。 0：该事件不生成复位请求 1：该事件生成复位请求 注意：该位不进行预装载。

Slave_TIMERx 外部事件滤波配置寄存器 0 (SHRTIMER_STxEXEVFCFG0)

地址偏移：0x4C
复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			EXEV4FM[3:0]				EXEV4MEEN	保留	EXEV3FM[3:0]				EXEV3MEEN	保留	EXEV2FM[3]
			rw				rw		rw				rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXEV2FM[2:0]			EXEV0MEEN	保留	EXEV1FM[3:0]				EXEV1MEEN	保留	EXEV0FM[3:0]				EXEV0MEEN
rw			rw		rw				rw		rw				rw

位/位域	名称	描述
31:29	保留	必须保持复位值
28:25	EXEV4FM[3:0]	外部事件 4 滤波模式 请参考 EXEV0FM [3:0]的描述。
24	EXEV4MEEN	外部事件 4 存储功能使能 请参考 EXEV0MEEN 的描述。
23	保留	必须保持复位值
22:19	EXEV3FM[3:0]	外部事件 3 滤波模式

		请参考 EXEV0FM [3:0]的描述。
18	EXEV3MEEN	外部事件 3 存储功能使能 请参考 EXEV0MEEN 的描述。
17	保留	必须保持复位值
16:13	EXEV2FM[3:0]	外部事件 2 滤波模式 请参考 EXEV0FM [3:0]的描述。
12	EXEV2MEEN	外部事件 2 存储功能使能 请参考 EXEV0MEEN 的描述。
11	保留	必须保持复位值
10:7	EXEV1FM[3:0]	外部事件 1 滤波模式 请参考 EXEV0FM [3:0]的描述。
6	EXEV1MEEN	外部事件 1 存储功能使能 请参考 EXEV0MEEN 的描述。
5	保留	必须保持复位值
4:1	EXEV0FM[3:0]	外部事件 0 滤波模式 在消隐模式下，如果外部事件在消隐期间发生，则将其忽略。在窗口模式下，仅当外部事件发生在给定的时间窗口内时，才考虑该外部事件。 0000：滤波模式禁能。 0001：消隐模式。消隐时间是从计数器复位到 SHRTIMER_STxCMP0V 比较事件发生持续的时间。 0010：消隐模式。消隐时间是从计数器复位到 SHRTIMER_STxCMP1V 比较事件发生持续的时间。 0011：消隐模式。消隐时间是从计数器复位到 SHRTIMER_STxCMP2V 比较事件发生持续的时间。 0100：消隐模式。消隐时间是从计数器复位到 SHRTIMER_STxCMP3V 比较事件发生持续的时间。 0101：消隐模式。消隐时间为其他 Slave_TIMERy（除了 Slave_TIMERx）：STBLKSRC0。 0110：消隐模式。消隐时间为其他 Slave_TIMERy（除了 Slave_TIMERx）：STBLKSRC1。 0111：消隐模式。消隐时间为其他 Slave_TIMERy（除了 Slave_TIMERx）：STBLKSRC2。 1000：消隐模式。消隐时间为其他 Slave_TIMERy（除了 Slave_TIMERx）：STBLKSRC3。 1001：消隐模式。消隐时间为其他 Slave_TIMERy（除了 Slave_TIMERx）：STBLKSRC4。 1010：消隐模式。消隐时间为其他 Slave_TIMERy（除了 Slave_TIMERx）：STBLKSRC5。 1011：消隐模式。消隐时间为其他 Slave_TIMERy（除了 Slave_TIMERx）：STBLKSRC6。

		1100：消隐模式。消隐时间为其他 Slave_TIMERy（除了 Slave_TIMERx）：STBLKSRC7。 1101：窗口模式。窗口时间是从计数器复位到 SHRTIMER_STxCMP1V 比较事件发生持续的时间。 1110：窗口模式。窗口时间是从计数器复位到 SHRTIMER_STxCMP2V 比较事件发生持续的时间。 1111：窗口模式。窗口时间为其他 Slave_TIMERy（除了 Slave_TIMERx）：STWDSRC。 注意： （1）一旦计数器使能（STxCEN 位置 1），不得修改该位域。 （2）用于滤波的比较寄存器的值必须大于 0。
0	EXEV0MEEN	外部事件 0 存储功能使能 0：外部事件 0 存储功能禁能。 1：外部事件 0 存储功能使能。一旦消隐周期或窗口周期完成，便会产生存储事件。 注意： （1）一旦计数器使能（STxCEN 位置 1），不得修改该位域。 （2）该位置 1 时，可以在窗口模式下生成超时事件。

Slave_TIMERx 外部事件滤波配置寄存器 1 (SHRTIMER_STxEXEVFCFG1)

地址偏移：0x50

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			EXEV9FM[3:0]				EXEV9MEEN	保留	EXEV8FM[3:0]				EXEV8MEEN	保留	EXEV7FM[3]
			rw				rw		rw				rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXEV7FM[2:0]			EXEV7MEEN	保留	EXEV6FM[3:0]				EXEV6MEEN	保留	EXEV5FM[3:0]				EXEV5MEEN
rw			rw		rw				rw		rw				rw

位/位域	名称	描述
31:29	保留	必须保持复位值
28:25	EXEV9FM[3:0]	外部事件 9 滤波模式 请参考 SHRTIMER_STxEXEVFCFG0 寄存器中的 EXEV0FM [3:0]的描述。
24	EXEV9MEEN	外部事件 9 存储功能使能 请参考 SHRTIMER_STxEXEVFCFG0 寄存器中的 EXEV0MEEN 的描述。
23	保留	必须保持复位值
22:19	EXEV8FM[3:0]	外部事件 8 滤波模式

		请参考 SHRTIMER_STxEXEVFCFG0 寄存器中的 EXEV0FM [3:0]的描述。
18	EXEV8MEEN	外部事件 8 存储功能使能 请参考 SHRTIMER_STxEXEVFCFG0 寄存器中的 EXEV0MEEN 的描述。
17	保留	必须保持复位值
16:13	EXEV7FM[3:0]	外部事件 7 滤波模式 请参考 SHRTIMER_STxEXEVFCFG0 寄存器中的 EXEV0FM [3:0]的描述。
12	EXEV7MEEN	外部事件 7 存储功能使能 请参考 SHRTIMER_STxEXEVFCFG0 寄存器中的 EXEV0MEEN 的描述。
11	保留	必须保持复位值
10:7	EXEV6FM[3:0]	外部事件 6 滤波模式 请参考 SHRTIMER_STxEXEVFCFG0 寄存器中的 EXEV0FM [3:0]的描述。
6	EXEV6MEEN	外部事件 6 存储功能使能 请参考 SHRTIMER_STxEXEVFCFG0 寄存器中的 EXEV0MEEN 的描述。
5	保留	必须保持复位值
4:1	EXEV5FM[3:0]	外部事件 5 滤波模式 请参考 SHRTIMER_STxEXEVFCFG0 寄存器中的 EXEV0FM [3:0]的描述。
0	EXEV5MEEN	外部事件 5 存储功能使能 请参考 SHRTIMER_STxEXEVFCFG0 寄存器中的 EXEV0MEEN 的描述。

Slave_TIMERx 计数器复位寄存器(SHRTIMER_STxCNTRST)

地址偏移：0x54

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

For Slave_TIMER0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	ST4CMP3RST	ST4CMP1RST	ST4CMP0RST	ST3CMP3RST	ST3CMP1RST	ST3CMP0RST	ST2CMP3RST	ST2CMP1RST	ST2CMP0RST	ST1CMP3RST	ST1CMP1RST	ST1CMP0RST	EXEV9RST	EXEV8RST	EXEV7RST
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXEV6RST	EXEV5RST	EXEV4RST	EXEV3RST	EXEV2RST	EXEV1RST	EXEV0RST	MTCMP3RST	MTCMP2RST	MTCMP1RST	MTCMP0RST	MTPERRST	CMP3RST	CMP1RST	UPRST	保留
rw	rw	rw	rw	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	ST4CMP3RST	Slave_TIMER4 比较 3 事件复位计数器

请参考 ST4CMP0RST 说明。

29 ST4CMP1RST Slave_TIMER4 比较 1 事件复位计数器
请参考 ST4CMP0RST 说明。

28 ST4CMP0RST Slave_TIMER4 比较 0 事件复位计数器
该位用于配置 Slave_TIMER4 比较 0 事件复位计数器。
0：Slave_TIMER4 比较 0 事件不复位计数器。
1：Slave_TIMER4 比较 0 事件复位计数器。

27 ST3CMP3RST Slave_TIMER3 比较 3 事件复位计数器
请参考 ST3CMP0RST 说明。

26 ST3CMP1RST Slave_TIMER3 比较 1 事件复位计数器
请参考 ST3CMP0RST 说明。

25 ST3CMP0RST Slave_TIMER3 比较 0 事件复位计数器
该位用于配置 Slave_TIMER3 比较 0 事件复位计数器。
0：Slave_TIMER3 比较 0 事件不复位计数器。
1：Slave_TIMER3 比较 0 事件复位计数器。

24 ST2CMP3RST Slave_TIMER2 比较 3 事件复位计数器
请参考 ST2CMP0RST 说明。

23 ST2CMP1RST Slave_TIMER2 比较 1 事件复位计数器
请参考 ST2CMP0RST 说明。

22 ST2CMP0RST Slave_TIMER2 比较 0 事件复位计数器
该位用于配置 Slave_TIMER2 比较 0 事件复位计数器。
0：Slave_TIMER2 比较 0 事件不复位计数器。
1：Slave_TIMER2 比较 0 事件复位计数器。

21 ST1CMP3RST Slave_TIMER1 比较 3 事件复位计数器
请参考 ST1CMP0RST 说明。

20 ST1CMP1RST Slave_TIMER1 比较 1 事件复位计数器
请参考 ST1CMP0RST 说明。

19 ST1CMP0RST Slave_TIMER1 比较 0 事件复位计数器
该位用于配置 Slave_TIMER1 比较 0 事件复位计数器。
0：Slave_TIMER1 比较 0 事件不复位计数器。
1：Slave_TIMER1 比较 0 事件复位计数器。

18 EXEV9RST 外部事件 9 复位计数器
请参考 EXEV0RST 说明。

17 EXEV8RST 外部事件 8 复位计数器
请参考 EXEV0RST 说明。

16 EXEV7RST 外部事件 7 复位计数器
请参考 EXEV0RST 说明。

15	EXEV6RST	外部事件 6 复位计数器 请参考 EXEV0RST 说明。.
14	EXEV5RST	外部事件 5 复位计数器 请参考 EXEV0RST 说明。
13	EXEV4RST	外部事件 4 复位计数器 请参考 EXEV0RST 说明。
12	EXEV3RST	外部事件 3 复位计数器 请参考 EXEV0RST 说明。
11	EXEV2RST	外部事件 2 复位计数器 请参考 EXEV0RST 说明。
10	EXEV1RST	外部事件 1 复位计数器 请参考 EXEV0RST 说明。
9	EXEV0RST	外部事件 0 复位计数器 0：外部事件 0 不复位计数器。 1：外部事件 0 复位计数器。
8	MTCMP3RST	Master_TIMER 比较 3 事件复位计数器 请参考 MTCMP0RST 说明。
7	MTCMP2RST	Master_TIMER 比较 2 事件复位计数器 请参考 MTCMP0RST 说明。
6	MTCMP1RST	Master_TIMER 比较 1 事件复位计数器 请参考 MTCMP0RST 说明。
5	MTCMP0RST	Master_TIMER 比较 0 事件复位计数器 该位用于配置 Master_TIMER 比较 0 事件复位计数器。 0：Master_TIMER 比较 0 事件不复位计数器。 1：Master_TIMER 比较 0 事件复位计数器。
4	MTPERRST	Master_TIMER 周期事件复位计数器 该位用于配置 Master_TIMER 周期事件复位计数器。 0：Master_TIMER 周期事件不复位计数器。 1：Master_TIMER 周期事件复位计数器。
3	CMP3RST	Slave_TIMER0 比较 3 事件复位计数器 请参考 CMP1RST 说明。
2	CMP1RST	Slave_TIMER0 比较 1 事件复位计数器 该位用于配置比较 1 事件复位计数器。 0：比较 1 事件不复位计数器。 1：比较 1 事件复位计数器。
1	UPRST	Slave_TIMER0 更新事件复位计数器 该位用于配置更新事件复位计数器。

0：更新事件不复位计数器。
1：更新事件复位计数器。

0	保留	必须保持复位值

For Slave_TIMER1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	ST4CMP 3RST	ST4CMP 1RST	ST4CMP 0RST	ST3CMP 3RST	ST3CMP 1RST	ST3CMP 0RST	ST2CMP 3RST	ST2CMP 1RST	ST2CMP 0RST	ST0CMP 3RST	ST0CMP 1RST	ST0CMP 0RST	EXEV9 RST	EXEV8 RST	EXEV7 RST
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXEV6 RST	EXEV5 RST	EXEV4 RST	EXEV3 RST	EXEV2 RST	EXEV1 RST	EXEV0 RST	MTCMP3 RST	MTCMP2 RST	MTCMP1 RST	MTCMP0 RST	MTPER RST	CMP3RST	CMP1RST	UPRST	保留
rw	rw	rw	rw	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	ST4CMP3RST	Slave_TIMER4 比较 3 事件复位计数器 请参考 ST4CMP0RST 说明。
29	ST4CMP1RST	Slave_TIMER4 比较 1 事件复位计数器 请参考 ST4CMP0RST 说明。
28	ST4CMP0RST	Slave_TIMER4 比较 0 事件复位计数器 该位用于配置 Slave_TIMER4 比较 0 事件复位计数器。 0：Slave_TIMER4 比较 0 事件不复位计数器 1：Slave_TIMER4 比较 0 事件复位计数器
27	ST3CMP3RST	Slave_TIMER3 比较 3 事件复位计数器 请参考 ST3CMP0RST 说明。
26	ST3CMP1RST	Slave_TIMER3 比较 1 事件复位计数器 请参考 ST3CMP0RST 说明。
25	ST3CMP0RST	Slave_TIMER3 比较 0 事件复位计数器 该位用于配置 Slave_TIMER3 比较 0 事件复位计数器。 0：Slave_TIMER3 比较 0 事件不复位计数器 1：Slave_TIMER3 比较 0 事件复位计数器
24	ST2CMP3RST	Slave_TIMER2 比较 3 事件复位计数器 请参考 ST2CMP0RST 说明。
23	ST2CMP1RST	Slave_TIMER2 比较 1 事件复位计数器 请参考 ST2CMP0RST 说明。
22	ST2CMP0RST	Slave_TIMER2 比较 0 事件复位计数器 该位用于配置 Slave_TIMER2 比较 0 事件复位计数器。

		0：Slave_TIMER2 比较 0 事件不复位计数器 1：Slave_TIMER2 比较 0 事件复位计数器
21	ST0CMP3RST	Slave_TIMER0 比较 3 事件复位计数器 请参考 ST0CMP0RST 说明。
20	ST0CMP1RST	Slave_TIMER0 比较 1 事件复位计数器 请参考 ST0CMP0RST 说明。
19	ST0CMP0RST	Slave_TIMER0 比较 0 事件复位计数器 该位用于配置 Slave_TIMER0 比较 0 事件复位计数器。 0：Slave_TIMER0 比较 0 事件不复位计数器 1：Slave_TIMER0 比较 0 事件复位计数器
18	EXEV9RST	外部事件 9 复位计数器 请参考 EXEV0RST 说明。
17	EXEV8RST	外部事件 8 复位计数器 请参考 EXEV0RST 说明。
16	EXEV7RST	外部事件 7 复位计数器 请参考 EXEV0RST 说明。
15	EXEV6RST	外部事件 6 复位计数器 请参考 EXEV0RST 说明。
14	EXEV5RST	外部事件 5 复位计数器 请参考 EXEV0RST 说明。
13	EXEV4RST	外部事件 4 复位计数器 请参考 EXEV0RST 说明。
12	EXEV3RST	外部事件 3 复位计数器 请参考 EXEV0RST 说明。
11	EXEV2RST	外部事件 2 复位计数器 请参考 EXEV0RST 说明。
10	EXEV1RST	外部事件 1 复位计数器 请参考 EXEV0RST 说明。
9	EXEV0RST	外部事件 0 复位计数器 0：外部事件 0 不复位计数器 1：外部事件 0 复位计数器
8	MTCMP3RST	Master_TIMER 比较 3 事件复位计数器 请参考 MTCMP0RST 说明。
7	MTCMP2RST	Master_TIMER 比较 2 事件复位计数器 请参考 MTCMP0RST 说明。
6	MTCMP1RST	Master_TIMER 比较 1 事件复位计数器

		请参考 MTCMP0RST 说明。
5	MTCMP0RST	Master_TIMER 比较 0 事件复位计数器 该位用于配置 Master_TIMER 比较 0 事件复位计数器。 0：Master_TIMER 比较 0 事件不复位计数器 1：Master_TIMER 比较 0 事件复位计数器
4	MTPERRST	Master_TIMER 周期事件复位计数器 该位用于配置 Master_TIMER 周期事件复位计数器。 0：Master_TIMER 周期事件不复位计数器 1：Master_TIMER 周期事件复位计数器
3	CMP3RST	Slave_TIMER1 比较 3 事件复位计数器 请参考 CMP1RST 说明。
2	CMP1RST	Slave_TIMER1 比较 1 事件复位计数器 该位用于配置比较 1 事件复位计数器。 0：比较 1 事件不复位计数器 1：比较 1 事件复位计数器
1	UPRST	Slave_TIMER1 更新事件复位计数器 该位用于配置更新事件复位计数器。 0：更新事件不复位计数器 1：更新事件复位计数器
0	保留	必须保持复位值

For Slave_TIMER2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	ST4CMP 3RST	ST4CMP 1RST	ST4CMP 0RST	ST3CMP 3RST	ST3CMP 1RST	ST3CMP 0RST	ST1CMP 3RST	ST1CMP 1RST	ST1CMP 0RST	ST0CMP 3RST	ST0CMP 1RST	ST0CMP 0RST	EXEV9 RST	EXEV8 RST	EXEV7 RST
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXEV6 RST	EXEV5 RST	EXEV4 RST	EXEV3 RST	EXEV2 RST	EXEV1 RST	EXEV0 RST	MTCMP3 RST	MTCMP2 RST	MTCMP1 RST	MTCMP0 RST	MTPER RST	CMP3RST	CMP1RST	UPRST	保留
rw	rw	rw	rw	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	ST4CMP3RST	Slave_TIMER4 比较 3 事件复位计数器 请参考 ST4CMP0RST 说明。
29	ST4CMP1RST	Slave_TIMER4 比较 1 事件复位计数器 请参考 ST4CMP0RST 说明。

28	ST4CMP0RST	Slave_TIMER4 比较 0 事件复位计数器 该位用于配置 Slave_TIMER4 比较 0 事件复位计数器。 0：Slave_TIMER4 比较 0 事件不复位计数器 1：Slave_TIMER4 比较 0 事件复位计数器
27	ST3CMP3RST	Slave_TIMER3 比较 3 事件复位计数器 请参考 ST3CMP0RST 说明。
26	ST3CMP1RST	Slave_TIMER3 比较 1 事件复位计数器 请参考 ST3CMP0RST 说明。
25	ST3CMP0RST	Slave_TIMER3 比较 0 事件复位计数器 该位用于配置 Slave_TIMER3 比较 0 事件复位计数器。 0：Slave_TIMER3 比较 0 事件不复位计数器 1：Slave_TIMER3 比较 0 事件复位计数器
24	ST1CMP3RST	Slave_TIMER1 比较 3 事件复位计数器 请参考 ST1CMP0RST 说明。
23	ST1CMP1RST	Slave_TIMER1 比较 1 事件复位计数器 请参考 ST1CMP0RST 说明。
22	ST1CMP0RST	Slave_TIMER1 比较 0 事件复位计数器 该位用于配置 Slave_TIMER1 比较 0 事件复位计数器。 0：Slave_TIMER1 比较 0 事件不复位计数器 1：Slave_TIMER1 比较 0 事件复位计数器
21	ST0CMP3RST	Slave_TIMER0 比较 3 事件复位计数器 请参考 ST0CMP0RST 说明。
20	ST0CMP1RST	Slave_TIMER0 比较 1 事件复位计数器 请参考 ST0CMP0RST 说明。
19	ST0CMP0RST	Slave_TIMER0 比较 0 事件复位计数器 该位用于配置 Slave_TIMER0 比较 0 事件复位计数器。 0：Slave_TIMER0 比较 0 事件不复位计数器 1：Slave_TIMER0 比较 0 事件复位计数器
18	EXEV9RST	外部事件 9 复位计数器 请参考 EXEV0RST 说明。
17	EXEV8RST	外部事件 8 复位计数器 请参考 EXEV0RST 说明。
16	EXEV7RST	外部事件 7 复位计数器 请参考 EXEV0RST 说明。
15	EXEV6RST	外部事件 6 复位计数器 请参考 EXEV0RST 说明。
14	EXEV5RST	外部事件 5 复位计数器

		请参考 EXEV0RST 说明。
13	EXEV4RST	外部事件 4 复位计数器 请参考 EXEV0RST 说明。
12	EXEV3RST	外部事件 3 复位计数器 请参考 EXEV0RST 说明。
11	EXEV2RST	外部事件 2 复位计数器 请参考 EXEV0RST 说明。
10	EXEV1RST	外部事件 1 复位计数器 请参考 EXEV0RST 说明。
9	EXEV0RST	外部事件 0 复位计数器 0：外部事件 0 不复位计数器 1：外部事件 0 复位计数器
8	MTCMP3RST	Master_TIMER 比较 3 事件复位计数器 请参考 MTCMP0RST 说明。
7	MTCMP2RST	Master_TIMER 比较 2 事件复位计数器 请参考 MTCMP0RST 说明。
6	MTCMP1RST	Master_TIMER 比较 1 事件复位计数器 请参考 MTCMP0RST 说明。
5	MTCMP0RST	Master_TIMER 比较 0 事件复位计数器 该位用于配置 Master_TIMER 比较 0 事件复位计数器。 0：Master_TIMER 比较 0 事件不复位计数器。 1：Master_TIMER 比较 0 事件复位计数器。
4	MTPERRST	Master_TIMER 周期事件复位计数器 该位用于配置 Master_TIMER 周期事件复位计数器。 0：Master_TIMER 周期事件不复位计数器。 1：Master_TIMER 周期事件复位计数器。
3	CMP3RST	Slave_TIMER2 比较 3 事件复位计数器 请参考 CMP1RST 说明。
2	CMP1RST	Slave_TIMER2 比较 1 事件复位计数器 该位用于配置比较 1 事件复位计数器。 0：比较 1 事件不复位计数器。 1：比较 1 事件复位计数器。
1	UPRST	Slave_TIMER2 更新事件复位计数器 该位用于配置更新事件复位计数器。 0：更新事件不复位计数器 1：更新事件复位计数器
0	保留	必须保持复位值

For Slave_TIMER3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	ST4CMP3RST	ST4CMP1RST	ST4CMP0RST	ST2CMP3RST	ST2CMP1RST	ST2CMP0RST	ST1CMP3RST	ST1CMP1RST	ST1CMP0RST	ST0CMP3RST	ST0CMP1RST	ST0CMP0RST	EXEV9RST	EXEV8RST	EXEV7RST
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXEV6RST	EXEV5RST	EXEV4RST	EXEV3RST	EXEV2RST	EXEV1RST	EXEV0RST	MTCMP3RST	MTCMP2RST	MTCMP1RST	MTCMP0RST	MTPERRST	CMP3RST	CMP1RST	UPRST	保留
rw	rw	rw	rw	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	ST4CMP3RST	Slave_TIMER4 比较 3 事件复位计数器 请参考 ST4CMP0RST 说明。
29	ST4CMP1RST	Slave_TIMER4 比较 1 事件复位计数器 请参考 ST4CMP0RST 说明。
28	ST4CMP0RST	Slave_TIMER4 比较 0 事件复位计数器 该位用于配置 Slave_TIMER4 比较 0 事件复位计数器。 0：Slave_TIMER4 比较 0 事件不复位计数器。 1：Slave_TIMER4 比较 0 事件复位计数器。
27	ST2CMP3RST	Slave_TIMER2 比较 3 事件复位计数器 请参考 ST2CMP0RST 说明。
26	ST2CMP1RST	Slave_TIMER2 比较 1 事件复位计数器 请参考 ST2CMP0RST 说明。
25	ST2CMP0RST	Slave_TIMER2 比较 0 事件复位计数器 该位用于配置 Slave_TIMER2 比较 0 事件复位计数器。 0：Slave_TIMER2 比较 0 事件不复位计数器。 1：Slave_TIMER2 比较 0 事件复位计数器。
24	ST1CMP3RST	Slave_TIMER1 比较 3 事件复位计数器 请参考 ST1CMP0RST 说明。
23	ST1CMP1RST	Slave_TIMER1 比较 1 事件复位计数器 请参考 ST1CMP0RST 说明。
22	ST1CMP0RST	Slave_TIMER1 比较 0 事件复位计数器 该位用于配置 Slave_TIMER1 比较 0 事件复位计数器。 0：Slave_TIMER2 比较 0 事件不复位计数器。 1：Slave_TIMER1 比较 0 事件复位计数器。
21	ST0CMP3RST	Slave_TIMER0 比较 3 事件复位计数器

		请参考 ST0CMP0RST 说明。
20	ST0CMP1RST	Slave_TIMER0 比较 1 事件复位计数器 请参考 ST0CMP0RST 说明。
19	ST0CMP0RST	Slave_TIMER0 比较 0 事件复位计数器 该位用于配置 Slave_TIMER0 比较 0 事件复位计数器。 0：Slave_TIMER0 比较 0 事件不复位计数器。 1：Slave_TIMER0 比较 0 事件复位计数器。
18	EXEV9RST	外部事件 9 复位计数器 请参考 EXEV0RST 说明。
17	EXEV8RST	外部事件 8 复位计数器 请参考 EXEV0RST 说明。
16	EXEV7RST	外部事件 7 复位计数器 请参考 EXEV0RST 说明。
15	EXEV6RST	外部事件 6 复位计数器 请参考 EXEV0RST 说明。
14	EXEV5RST	外部事件 5 复位计数器 请参考 EXEV0RST 说明。
13	EXEV4RST	外部事件 4 复位计数器 请参考 EXEV0RST 说明。
12	EXEV3RST	外部事件 3 复位计数器 请参考 EXEV0RST 说明。
11	EXEV2RST	外部事件 2 复位计数器 请参考 EXEV0RST 说明。
10	EXEV1RST	外部事件 1 复位计数器 请参考 EXEV0RST 说明。
9	EXEV0RST	外部事件 0 复位计数器 该位用于配置外部事件 0 复位计数器。 0：外部事件 0 不复位计数器。 1：外部事件 0 复位计数器。
8	MTCMP3RST	Master_TIMER 比较 3 事件复位计数器 请参考 MTCMP0RST 说明。
7	MTCMP2RST	Master_TIMER 比较 2 事件复位计数器 请参考 MTCMP0RST 说明。
6	MTCMP1RST	Master_TIMER 比较 1 事件复位计数器 请参考 MTCMP0RST 说明。
5	MTCMP0RST	Master_TIMER 比较 0 事件复位计数器

		该位用于配置 Master_TIMER 比较 0 事件复位计数器。 0：Master_TIMER 比较 0 事件不复位计数器。 1：Master_TIMER 比较 0 事件复位计数器。
4	MTPERRST	Master_TIMER 周期事件复位计数器 该位用于配置 Master_TIMER 周期事件复位计数器。 0：Master_TIMER 周期事件不复位计数器。 1：Master_TIMER 周期事件复位计数器。
3	CMP3RST	Slave_TIMER3 比较 3 事件复位计数器 请参考 CMP1RST 说明。
2	CMP1RST	Slave_TIMER3 比较 1 事件复位计数器 该位用于配置比较 1 事件复位计数器。 0：比较 1 事件不复位计数器。 1：比较 1 事件复位计数器。
1	UPRST	Slave_TIMER3 更新事件复位计数器 该位用于配置更新事件复位计数器。 0：更新事件不复位计数器。 1：更新事件复位计数器。
0	保留	必须保持复位值

For Slave_TIMER4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	ST3CMP3RST	ST3CMP1RST	ST3CMP0RST	ST2CMP3RST	ST2CMP1RST	ST2CMP0RST	ST1CMP3RST	ST1CMP1RST	ST1CMP0RST	ST0CMP3RST	ST0CMP1RST	ST0CMP0RST	EXEV9RST	EXEV8RST	EXEV7RST
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXEV6RST	EXEV5RST	EXEV4RST	EXEV3RST	EXEV2RST	EXEV1RST	EXEV0RST	MTCMP3RST	MTCMP2RST	MTCMP1RST	MTCMP0RST	MTPERRST	CMP3RST	CMP1RST	UPRST	保留
rw	rw	rw	rw	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	ST3CMP3RST	Slave_TIMER3 比较 3 事件复位计数器 请参考 ST3CMP0RST 说明。
29	ST3CMP1RST	Slave_TIMER3 比较 1 事件复位计数器 请参考 ST3CMP0RST 说明。
28	ST3CMP0RST	Slave_TIMER3 比较 0 事件复位计数器 该位用于配置 Slave_TIMER3 比较 0 事件复位计数器。 0：Slave_TIMER3 比较 0 事件不复位计数器。 1：Slave_TIMER3 比较 0 事件复位计数器。

27	ST2CMP3RST	Slave_TIMER2 比较 3 事件复位计数器 请参考 ST2CMP0RST 说明。
26	ST2CMP1RST	Slave_TIMER2 比较 1 事件复位计数器 请参考 ST2CMP0RST 说明。
25	ST2CMP0RST	Slave_TIMER2 比较 0 事件复位计数器 该位用于配置 Slave_TIMER2 比较 0 事件复位计数器。 0：Slave_TIMER2 比较 0 事件不复位计数器。 1：Slave_TIMER2 比较 0 事件复位计数器。
24	ST1CMP3RST	Slave_TIMER1 比较 3 事件复位计数器 请参考 ST1CMP0RST 说明。
23	ST1CMP1RST	Slave_TIMER1 比较 1 事件复位计数器 请参考 ST1CMP0RST 说明。
22	ST1CMP0RST	Slave_TIMER1 比较 0 事件复位计数器 该位用于配置 Slave_TIMER1 比较 0 事件复位计数器。 0：Slave_TIMER1 比较 0 事件不复位计数器。 1：Slave_TIMER1 比较 0 事件复位计数器。
21	ST0CMP3RST	Slave_TIMER0 比较 3 事件复位计数器 请参考 ST0CMP0RST 说明。
20	ST0CMP1RST	Slave_TIMER0 比较 1 事件复位计数器 请参考 ST0CMP0RST 说明。
19	ST0CMP0RST	Slave_TIMER0 比较 0 事件复位计数器 该位用于配置 Slave_TIMER0 比较 0 事件复位计数器。 0：Slave_TIMER0 比较 0 事件不复位计数器。 1：Slave_TIMER0 比较 0 事件复位计数器。
18	EXEV9RST	外部事件 9 复位计数器 请参考 EXEV0RST 说明。
17	EXEV8RST	外部事件 8 复位计数器 请参考 EXEV0RST 说明。
16	EXEV7RST	外部事件 7 复位计数器 请参考 EXEV0RST 说明。
15	EXEV6RST	外部事件 6 复位计数器 请参考 EXEV0RST 说明。
14	EXEV5RST	外部事件 5 复位计数器 请参考 EXEV0RST 说明。
13	EXEV4RST	外部事件 4 复位计数器 请参考 EXEV0RST 说明。

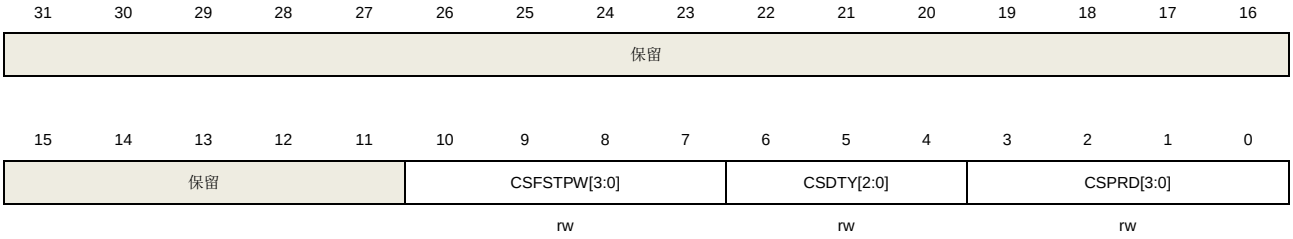
12	EXEV3RST	外部事件 3 复位计数器 请参考 EXEV0RST 说明。
11	EXEV2RST	外部事件 2 复位计数器 请参考 EXEV0RST 说明。
10	EXEV1RST	外部事件 1 复位计数器 请参考 EXEV0RST 说明。
9	EXEV0RST	外部事件 0 复位计数器 0：外部事件 0 不复位计数器 1：外部事件 0 复位计数器
8	MTCMP3RST	Master_TIMER 比较 3 事件复位计数器 请参考 MTCMP0RST 说明。
7	MTCMP2RST	Master_TIMER 比较 2 事件复位计数器 请参考 MTCMP0RST 说明。
6	MTCMP1RST	Master_TIMER 比较 1 事件复位计数器 请参考 MTCMP0RST 说明。
5	MTCMP0RST	Master_TIMER 比较 0 事件复位计数器 该位用于配置 Master_TIMER 比较 0 事件复位计数器。 0：Master_TIMER 比较 0 事件不复位计数器。 1：Master_TIMER 比较 0 事件复位计数器。
4	MTPERRST	Master_TIMER 周期事件复位计数器 该位用于配置 Master_TIMER 周期事件复位计数器。 0：Master_TIMER 周期事件不复位计数器。 1：Master_TIMER 周期事件复位计数器。
3	CMP3RST	Slave_TIMER4 比较 3 事件复位计数器 请参考 CMP1RST 说明。
2	CMP1RST	Slave_TIMER4 比较 1 事件复位计数器 该位用于配置比较 1 事件复位计数器。 0：比较 1 事件不复位计数器。 1：比较 1 事件复位计数器。
1	UPRST	Slave_TIMER4 更新事件复位计数器 该位用于配置更新事件复位计数器。 0：更新事件不复位计数器。 1：更新事件复位计数器。
0	保留	必须保持复位值

Slave_TIMERx 载波控制寄存器 (SHRTIMER_STxCSCTL)

地址偏移：0x58

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。



位/位域	名称	描述
31:11	保留	必须保持复位值
10:7	CSFSTPW[3:0]	第一个载波信号的脉冲宽度 该位域定义了在通道输出准备信号（CHxOPRE）上升沿之后的第一个载波信号的脉冲宽度。 $t_{CSFSTPW}$ = (CSFSTPW[3:0]+1) x $t_{SHRTIMER_CSGCK}$，$t_{SHRTIMER_CSGCK}$ =16 x $t_{SHRTIMER_CK}$. 0000：$t_{CSFSTPW}$ = $t_{SHRTIMER_CSGCK}$ 0001：$t_{CSFSTPW}$ = 2*$t_{SHRTIMER_CSGCK}$ … 1110：$t_{CSFSTPW}$ = 15*$t_{SHRTIMER_CSGCK}$ 1111：$t_{CSFSTPW}$ = 16*$t_{SHRTIMER_CSGCK}$
6:4	CSDTY[2:0]	载波信号占空比 该位域定义了载波信号（第一个脉冲除外）的占空比为 CSDTY [2:0] / 8。 000：0%（仅出现第一个脉冲）。 001：12.5% 010：25.0% 011：37.5% 100：50.0% 101：62.5% 110：75.0% 111：87.5%
3:0	CSPRD[3:0]	载波信号周期 该位域定义了载波信号的周期（第一个脉冲除外）。 t_{CSPRD} = (CSPRD[3:0]+1) x $t_{SHRTIMER_CSGCK}$, $t_{SHRTIMER_CSGCK}$ =16 x $t_{SHRTIMER_CK}$. 0000: 16 x $t_{SHRTIMER_CK}$ 0001: 32 x $t_{SHRTIMER_CK}$ … 1111: 256 x $t_{SHRTIMER_CK}$

Slave_TIMERx 捕获 0 触发寄存器 (SHRTIMER_STxCAP0TRG)

地址偏移：0x5C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CP0BST4CMP1	CP0BST4CMP0	CP0BST4NA	CP0BST4A	CP0BST3CMP1	CP0BST3CMP0	CP0BST3NA	CP0BST3A	CP0BST2MP1	CP0BST2CMP0	CP0BST2NA	CP0BST2A	CP0BST1CMP1	CP0BST1CMP0	CP0BST1NA	CP0BST1A
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CP0BST0CMP1	CP0BST0CMP0	CP0BST0NA	CP0BST0A	CP0BEXEV9	CP0BEXEV8	CP0BEXEV7	CP0BEXEV6	CP0BEXEV5	CP0BEXEV4	CP0BEXEV3	CP0BEXEV2	CP0BEXEV1	CP0BEXEV0	CP0BUP	CP0BSW
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	CP0BST4CMP1	Slave_TIMER4 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP0BST0CMP1 描述。
30	CP0BST4CMP0	Slave_TIMER4 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP0BST0CMP0 描述。
29	CP0BST4NA	ST4CH0_O 输出有效到无效的变化触发捕获 0 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP0BST0NA 描述。
28	CP0BST4A	ST4CH0_O 输出无效到有效的变化触发捕获 0 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP0BST0A 描述。
27	CP0BST3CMP1	Slave_TIMER3 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP0BST0CMP1 描述。
26	CP0BST3CMP0	Slave_TIMER3 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP0BST0CMP0 描述。
25	CP0BST3NA	ST3CH0_O 输出有效到无效的变化触发捕获 0 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP0BST0NA 描述。
24	CP0BST3A	ST3CH0_O 输出无效到有效的变化触发捕获 0 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP0BST0A 描述。
23	CP0BST2CMP1	Slave_TIMER2 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP0BST0CMP1 描述。
22	CP0BST2CMP0	Slave_TIMER2 的比较 0 事件触发捕获 0

		该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP0BST0CMP0 描述。
21	CP0BST2NA	ST2CH0_O 输出有效到无效的变化触发捕获 0 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP0BST0NA 描述。
20	CP0BST2A	ST2CH0_O 输出无效到有效的变化触发捕获 0 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP0BST0A 描述。
19	CP0BST1CMP1	Slave_TIMER1 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP0BST0CMP1 描述。
18	CP0BST1CMP0	Slave_TIMER1 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP0BST0CMP0 描述。
17	CP0BST1NA	ST1CH0_O 输出有效到无效的变化触发捕获 0 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP0BST0NA 描述。
16	CP0BST1A	ST1CH0_O 输出无效到有效的变化触发捕获 0 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP0BST0A 描述。
15	CP0BST0CMP1	Slave_TIMER0 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER0 的寄存器中保留。 0：Slave_TIMER0 的比较 1 事件不触发捕获 0。 1：Slave_TIMER0 的比较 1 事件触发捕获 0。
14	CP0BST0CMP0	Slave_TIMER0 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER0 的寄存器中保留。 0：Slave_TIMER0 的比较 0 事件不触发捕获 0。 1：Slave_TIMER0 的比较 0 事件触发捕获 0。
13	CP0BST0NA	ST0CH0_O 输出有效到无效的变化触发捕获 0 当 Slave_TIMER0 通道 0 的输出从有效电平转换为无效电平时，捕获 0 由 ST0CH0_O 触发。 该位仅在 Slave_TIMER0 中保留。 0：ST0CH0_O 输出有效到无效的变化不触发捕获 0。 1：ST0CH0_O 输出有效到无效的变化触发捕获 0。
12	CP0BST0A	ST0CH0_O 输出无效到有效的变化触发捕获 0 当 Slave_TIMER0 通道 0 的输出从无效电平转换为有效电平时，捕获 0 由 ST0CH0_O 触发。 该位仅在 Slave_TIMER0 中保留。 0：ST0CH0_O 输出无效到有效的变化不触发捕获 0。

		1：ST0CH0_O 输出无效到有效的变化触发捕获 0。
11	CP0BEXEV9	外部事件 9 触发捕获 0 请参考 CP0BEXEV0 描述。
10	CP0BEXEV8	外部事件 8 触发捕获 0 请参考 CP0BEXEV0 描述。
9	CP0BEXEV7	外部事件 7 触发捕获 0 请参考 CP0BEXEV0 描述。
8	CP0BEXEV6	外部事件 6 触发捕获 0 请参考 CP0BEXEV0 描述。
7	CP0BEXEV5	外部事件 5 触发捕获 0 请参考 CP0BEXEV0 描述。
6	CP0BEXEV4	外部事件 4 触发捕获 0 请参考 CP0BEXEV0 描述。
5	CP0BEXEV3	外部事件 3 触发捕获 0 请参考 CP0BEXEV0 描述。
4	CP0BEXEV2	外部事件 2 触发捕获 0 请参考 CP0BEXEV0 描述。
3	CP0BEXEV1	外部事件 1 触发捕获 0 请参考 CP0BEXEV0 描述。
2	CP0BEXEV0	外部事件 0 触发捕获 0 当该位置 1 时，外部事件 0 触发捕获 0。 0：外部事件 0 不触发捕获 0 1：外部事件 0 触发捕获 0
1	CP0BUP	更新事件触发捕获 0 当该位置 1 时，更新事件触发捕获 0。 0：更新事件不触发捕获 0 1：更新事件触发捕获 0
0	CP0BSW	软件触发捕获 0 该位由软件置 1，硬件自动清零。该位置 1 时，软件触发捕获 0。 0：软件不触发捕获 0 1：软件触发捕获 0

Slave_TIMERx 捕获 1 触发寄存器 (SHRTIMER_STxCAP1TRG)

地址偏移：0x60

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

CP1BST4 CMP1	CP1BST4 CMP0	CP1BST4 NA	CP1BST4 A	CP1BST3 CMP1	CP1BST3 CMP0	CP1BST3 NA	CP1BST3 A	CP1BST2 MP1	CP1BST2 CMP0	CP1BST2 NA	CP1BST2 A	CP1BST1 CMP1	CP1BST1 CMP0	CP1BST1 NA	CP1BST1 A
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CP1BST0 CMP1	CP1BST0 CMP0	CP1BST0 NA	CP1BST0 A	CP1BEX EV9	CP1BEX EV8	CP1BEX EV7	CP1BEX EV6	CP1BEX EV5	CP1BEX EV4	CP1BEX EV3	CP1BEX EV2	CP1BEX EV1	CP1BEX EV0	CP1BUP	CP1BSW
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	CP1BST4CMP1	Slave_TIMER4 的比较 1 事件触发捕获 1 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP1BST0CMP1 描述。
30	CP1BST4CMP0	Slave_TIMER4 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP1BST0CMP0 描述。
29	CP1BST4NA	ST4CH0_O 输出有效到无效的变化触发捕获 1 该位仅在 Slave_TIMER4 的寄存器中。 请参考 CP1BST0NA 描述。
28	CP1BST4A	ST4CH0_O 输出无效到有效的变化触发捕获 1 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP1BST0A 描述。
27	CP1BST3CMP1	Slave_TIMER3 的比较 1 事件触发捕获 1 该位仅存在于 Slave_TIMER3 的寄存器中保留。 请参考 CP1BST0CMP1 描述。
26	CP1BST3CMP0	Slave_TIMER3 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP1BST0CMP0 描述。
25	CP1BST3NA	ST3CH0_O 输出有效到无效的变化触发捕获 1 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP1BST0NA 描述。
24	CP1BST3A	ST3CH0_O 输出无效到有效的变化触发捕获 1 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP1BST0A 描述。
23	CP1BST2CMP1	Slave_TIMER2 的比较 1 事件触发捕获 1 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP1BST0CMP1 描述。
22	CP1BST2CMP0	Slave_TIMER2 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP1BST0CMP0 描述。

21	CP1BST2NA	ST2CH0_O 输出有效到无效的变化触发捕获 1 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP1BST0NA 描述。
20	CP1BST2A	ST2CH0_O 输出无效到有效的变化触发捕获 1 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP1BST0A 描述。
19	CP1BST1CMP1	Slave_TIMER1 的比较 1 事件触发捕获 1 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP1BST0CMP1 描述。
18	CP1BST1CMP0	Slave_TIMER1 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP1BST0CMP0 描述。
17	CP1BST1NA	ST1CH0_O 输出有效到无效的变化触发捕获 1 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP1BST0NA 描述。
16	CP1BST1A	ST1CH0_O 输出无效到有效的变化触发捕获 1 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP1BST0A 描述。
15	CP1BST0CMP1	Slave_TIMER0 的比较 1 事件触发捕获 1 该位仅在 Slave_TIMER0 的寄存器中保留。 0：Slave_TIMER0 的比较 1 事件不触发捕获 1。 1：Slave_TIMER0 的比较 1 事件触发捕获 1。
14	CP1BST0CMP0	Slave_TIMER0 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER0 的寄存器中保留。 0：Slave_TIMER0 的比较 0 事件不触发捕获 1。 1：Slave_TIMER0 的比较 0 事件触发捕获 1。
13	CP1BST0NA	ST0CH0_O 输出有效到无效的变化触发捕获 1 当 Slave_TIMER0 通道 0 的输出从有效电平转换为无效电平时，捕获 1 由 ST0CH0_O 触发。 该位仅在 Slave_TIMER0 中保留。 0：ST0CH0_O 输出有效到无效的变化不触发捕获 1。 1：ST0CH0_O 输出有效到无效的变化触发捕获 1。
12	CP1BST0A	ST0CH0_O 输出无效到有效的变化触发捕获 1 当 Slave_TIMER0 通道 0 的输出从无效电平转换为有效电平时，捕获 1 由 ST0CH0_O 触发。 该位仅在 Slave_TIMER0 中保留。 0：ST0CH0_O 输出无效到有效的变化不触发捕获 1。 1：ST0CH0_O 输出无效到有效的变化触发捕获 1。
11	CP1BEXEV9	外部事件 9 触发捕获 1

请参考 CP1BEXEV0 描述。

10	CP1BEXEV8	外部事件 8 触发捕获 1 请参考 CP1BEXEV0 描述。
9	CP1BEXEV7	外部事件 7 触发捕获 1 请参考 CP1BEXEV0 描述。
8	CP1BEXEV6	外部事件 6 触发捕获 1 请参考 CP1BEXEV0 描述。
7	CP1BEXEV5	外部事件 5 触发捕获 1 请参考 CP1BEXEV0 描述。
6	CP1BEXEV4	外部事件 4 触发捕获 1 请参考 CP1BEXEV0 描述。
5	CP1BEXEV3	外部事件 3 触发捕获 1 请参考 CP1BEXEV0 描述。
4	CP1BEXEV2	外部事件 2 触发捕获 1 请参考 CP1BEXEV0 描述。
3	CP1BEXEV1	外部事件 1 触发捕获 1 请参考 CP1BEXEV0 描述。
2	CP1BEXEV0	外部事件 0 触发捕获 1 当该位置 1 时，外部事件 0 触发捕获 1。 0：外部事件 0 不触发捕获 1 1：外部事件 0 触发捕获 1
1	CP1BUP	更新事件触发捕获 1 当该位置 1 时，更新事件触发捕获 1。 0：更新事件不触发捕获 1 1：更新事件触发捕获 1
0	CP1BSW	软件触发捕获 1 该位由软件置 1，硬件自动清零。该位置 1 时，软件触发捕获 1。 0：软件不触发捕获 1 1：软件触发捕获 1

Slave_TIMERx 通道输出控制寄存器 (SHRTIMER_STxCHOCTL)

地址偏移：0x64

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								BMCH1DTI	CH1CSEN	CH1FLTOS[1:0]		ISO1	BMCH1IEN	CH1P	保留

									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			DLYISCH[2:0]			DLYISMEN	DTEN	BMCH1DTI	CH0CSEN	CH0FLTOS[1:0]		ISO0	BMCH0IEN	CH0P	保留	
				rw		rw	rw	rw	rw	rw		rw	rw	rw		

位/位域	名称	描述
31:24	保留	必须保持复位值
23	BMCH1DTI	突发模式中通道 1 的死区时间 在突发模式下，可以在输出进入空闲状态之前插入死区时间。 0：输出立即为空闲状态。 1：输出在进入空闲状态之前插入死区时间。 注意： （1）一旦计数器使能（STxCEN 位置 1），就不得修改该位。 （2）仅在突发模式的空闲模式下，输出空闲状态之一为有效状态（ISOy = 1，y = 0,1），且死区时间值为正（DTFSPROT 位 / DTRSPROT 位为 0）时，才可以设置此位）。
22	CH1CSEN	通道 1 载波信号模式使能 0：通道 1 载波信号模式禁能。 1：通道 1 载波信号模式使能。 注意：一旦计数器使能（STxCEN 位置 1），就不得修改该位。
21:20	CH1FLTOS[1:0]	通道 1 故障输出状态 该位域配置了故障事件发生时，通道 1 的输出状态。 00：没有影响。发生故障事件时，输出处于运行模式。 01：输出为有效电平。 10：输出无效电平。 11：输出为高阻状态。 注意：如果 SHRTIMER_STxFLTCTL 寄存器中的 FLTyEN（y = 0..4）位置 1，或输出处于故障状态，,一旦计数器使能（将 STxCEN 位置 1），就不能修改该位域。
19	ISO1	通道 1 输出空闲状态 0：通道 1 输出空闲状态为无效电平。 1：通道 1 输出空闲状态为有效电平。 注意：必须在 SHRTIMER 控制输出前，配置该位。
18	BMCH1IEN	在突发模式中使能通道 1 空闲状态 该位用于配置在突发模式下，通道 1 输出空闲状态。 0：通道 1 输出不受突发模式影响。 1：在突发模式下，通道 1 的输出可以为空闲状态。 注意：该位已预加载，可以在运行时更改，但在突发模式下不得更改。
17	CH1P	通道 1 输出极性 该位确定通道 1 输出信号的极性。 0：通道 1 高电平有效。

		1：通道 1 低电平有效。 注意：一旦计数器使能（STxCEN 位置 1），就不得修改该位。
16:13	保留	必须保持复位值
12:10	DLYISCH[2:0]	延迟空闲的源和通道 该位域配置了延迟空闲模式（DLYISMEN = 1）使能时的源和通道。 在 SHRTIMER_STyCHOCTL（y = 0, 1, 2）寄存器中： 000：外部事件 5 到来时，通道 0 输出延迟的空闲状态。 001：外部事件 5 到来时，通道 1 输出延迟的空闲状态。 010：外部事件 5 到来时，通道 0 和通道 1 输出延迟的空闲状态。 011：在均衡模式下，外部事件 5 到来时，通道 0 和通道 1 输出均衡空闲状态（SHRTIMER_STyCTL0（y = 0, 1, 2）寄存器中的 BLNMEN = 1）。 100：外部事件 6 到来时，通道 0 输出延迟的空闲状态。 101：外部事件 6 到来时，通道 1 输出延迟的空闲状态。 110：外部事件 6 到来时，通道 0 和通道 1 输出延迟的空闲状态。 111：在均衡模式下，外部事件 6 到来时，通道 0 和通道 1 输出均衡空闲状态（SHRTIMER_STyCTL0（y = 0, 1, 2）寄存器中的 BLNMEN = 1）。 在 SHRTIMER_STyCHOCTL（y = 3,4）寄存器中： 000：外部事件 7 到来时，通道 0 输出延迟的空闲状态。 001：外部事件 7 到来时，通道 1 输出延迟的空闲状态。 010：外部事件 7 到来时，通道 0 和通道 1 输出延迟的空闲状态。 011：在均衡模式下，外部事件 7 到来时，通道 0 和通道 1 输出均衡空闲状态（SHRTIMER_STyCTL0（y = 3, 4）寄存器中的 BLNMEN = 1）。 100：外部事件 8 到来时，通道 0 输出延迟的空闲状态。 101：外部事件 8 到来时，通道 1 输出延迟的空闲状态。 110：外部事件 8 到来时，通道 0 和通道 1 输出延迟的空闲状态。 111：在均衡模式下，外部事件 8 到来时，通道 0 和通道 1 输出均衡空闲状态（SHRTIMER_STyCTL0（y = 3, 4）寄存器中的 BLNMEN = 1）。 注意：一旦延迟空闲模式（DLYISMEN 位置 1）使能，就不得修改此位域。
9	DLYISMEN	延迟空闲模式使能 0：延迟空闲模式禁能 1：延迟空闲模式使能 注意：一旦计数器使能（STxCEN 位置 1），就不得修改该位
8	DTEN	死区时间使能 0：通道 0 和通道 1 的输出是独立的。 1：通道 0 和通道 1 的输出是互补的，在通道 0 和通道 1 的输出之间插入死区时间。 注意：一旦计数器使能（STxCEN 位置 1），或其输出被其他定时器使能和控制，就不得修改该位。
7	BMCH0DTI	突发模式中通道 0 的死区时间 在突发模式下，可以在输出进入空闲状态之前插入死区时间。 0：输出立即为空闲状态。 1：输出在进入空闲状态之前插入死区时间。 注意：

（1）一旦计数器使能（STxCEN 位置 1），就不得修改该位。
（2）仅在突发模式的空闲模式下，输出空闲状态之一为有效状态（ISOy = 1，y = 0,1），且死区时间值为正（DTFSPROT 位 / DTRSPROT 位为 0）时，才可以设置此位）。

6	CH0CSEN	通道 0 载波信号模式使能 0：通道 0 载波信号模式禁能。 1：通道 0 载波信号模式使能。 注意：一旦计数器使能（STxCEN 位置 1），就不得修改该位。
5:4	CH0FLTOS[1:0]	通道 0 故障输出状态 该位域配置了在故障事件发生时，通道 0 的输出状态。 00：没有影响。发生故障事件时，输出处于运行模式。 01：输出为有效电平。 10：输出无效电平。 11：输出为高阻状态。 注意：如果 SHRTIMER_STxFLTCTL 寄存器中的 FLTyEN（y = 0..4）位置 1，或输出处于故障状态，一旦计数器使能（将 STxCEN 位置 1），就不能修改该位域。
3	ISO0	通道 0 输出空闲状态 0：通道 0 输出空闲状态为无效电平。 1：通道 0 输出空闲状态为有效电平。 注意：必须在 SHRTIMER 控制输出前，配置该位。
2	BMCH0IEN	在突发模式中使能通道 0 空闲状态 该位用于配置在突发模式下，通道 0 输出空闲状态 0：通道 0 输出不受突发模式影响。 1：在突发模式下，通道 0 的输出可以为空闲状态。 注意：该位已预加载，可以在运行时更改，但在突发模式下不得更改。
1	CH0P	通道 0 输出极性 该位确定通道 0 输出信号的极性。 0：通道 0 高电平有效 1：通道 0 低电平有效 注意：一旦计数器使能（STxCEN 位置 1），就不得修改该位。
0	保留	必须保持复位值

Slave_TIMERx 故障控制寄存器 (SHRTIMER_STxFLTCTL)

地址偏移：0x68
复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											FLT4EN	FLT3EN	FLT2EN	FLT1EN	FLT0EN
											rw	rw	rw	rw	rw

位/位域	名称	描述
31	FLTENPROT	故障保护使能 该位用于使能写保护，该位只能写一次，当它置 1 时，只能通过系统复位清零。 0：故障保护禁能。FLTyEN (y=0..4)可以写操作。 1：故障保护使能。FLTyEN (y=0..4)为只读。
30:5	保留	必须保持复位值
4	FLT4EN	故障 4 使能 0：故障 4 禁能 1：故障 4 使能
3	FLT3EN	故障 3 使能 0：故障 3 禁能 1：故障 3 使能
2	FLT2EN	故障 2 使能 0：故障 2 禁能 1：故障 2 使能
1	FLT1EN	故障 1 使能 0：故障 1 禁能 1：故障 1 使能
0	FLT0EN	故障 0 使能 0：故障 0 禁能 1：故障 0 使能

Slave_TIMERx 附加控制寄存器 (SHRTIMER_STxACTL)

地址偏移：0x7C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DTFCFG[15:9]							保留								
rw	rw	rw	rw	rw	rw	rw									

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

位/位域	名称	描述

15:9	DTFCFG[15:9]	下降沿死区值配置 该位域用于配置跟随输出准备信号（OyPRE，y = 0,1）下降沿之后的死区时间值。 DTF 值 = DTFCFG [15:0] x $t_{SHRTIMER_DTGCK}$，其中，$t_{SHRTIMER_DTGCK}$ = 1/ $f_{SHRTIMER_DTGCK}$。 写入该位域可以更改 DTFCFG [15:0]位域的高 7 位。 注意： （1）当 SHRTIMER_STxDTCTL 寄存器中的 DTFSVPROT 位置 1 时，无法修改此位域。 （2）该位是预装载的。
24:16	保留	必须保持复位值
15:9	DTRCFG[15:9]	上升沿死区值配置 该位域用于配置跟随输出准备信号（OyPRE，y = 0,1）上升沿之后的死区时间值。 DTR 值= DTRCFG[15:0] x $t_{SHRTIMER_DTGCK}$，其中，$t_{SHRTIMER_DTGCK}$ = 1/ $f_{SHRTIMER_DTGCK}$。 写入该位域可以更改 DTRCFG[15:0]位域的高 7 位。 注意： （1）当 SHRTIMER_STxDTCTL 寄存器中的 DTRSVPROT 位置 1 时，无法修改此位域。 （2）该位是预装载的。
8:4	保留	必须保持复位值
3	CNTCKDIV[3]	计数器时钟分频 该位位域由软件配置，确定超高分辨率时钟（SHRTIMER_HPCK）和计数器时钟（SHRTIMER_PSCCK）的分频比。 当 SHRTIMER_MTACTL 中的 CNTCKDIV [3]为 0 时，$f_{SHRTIMER_PSCCK}$ = $f_{SHRTIMER_HPCK}$ / $2^{CNTCKDIV\ [2:0]\ +1}$。 当 SHRTIMER_MTACTL 中的 CNTCKDIV [3]位为 1 且 CNTCKDIV [2:0]配置为 3’b000 时：$f_{SHRTIMER_PSSCK}$ = $f_{SHRTIMER_HPCK}$ 注意：一旦使能定时器，就不能修改 CNTCKDIV [3:0]位域。
2:0	保留	必须保持复位值

19.5.3. 通用寄存器

SHRTIMER 通用寄存器基地址：0x4001 7780

SHRTIMER 控制寄存器 0 (SHRTIMER_CTL0)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				ADTG3USRC[2:0]			ADTG2USRC [2:0]			ADTG1USRC [2:0]			ADTG0USRC [2:0]		
				rw			rw			rw			rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留										ST4UPDIS	ST3UPDIS	ST2UPDIS	ST1UPDIS	ST0UPDIS	MTUPDIS
										rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:28	保留	必须保持复位值
27:25	ADTG3USRC[2:0]	SHRTIMER_ADCTRIG3 更新源 该位域可以由软件配置，配置 SHRTIMER_ADCTRIGS3 寄存器的更新源。 000：Master_TIMER 更新事件 001：Slaver_TIMER0 更新事件 010：Slaver_TIMER1 更新事件 011：Slaver_TIMER2 更新事件 100：Slaver_TIMER3 更新事件 101：Slaver_TIMER4 更新事件 其他值保留。
24:22	ADTG2USRC[2:0]	SHRTIMER_ADCTRIG2 更新源 该位域可以由软件配置，配置 SHRTIMER_ADCTRIGS2 寄存器的更新源。 000：Master_TIMER 更新事件 001：Slaver_TIMER0 更新事件 010：Slaver_TIMER1 更新事件 011：Slaver_TIMER2 更新事件 100：Slaver_TIMER3 更新事件 101：Slaver_TIMER4 更新事件 其他值保留。
21:19	ADTG1USRC[2:0]	SHRTIMER_ADCTRIG1 更新源 该位域可以由软件配置，配置 SHRTIMER_ADCTRIGS1 寄存器的更新源。 000：Master_TIMER 更新事件 001：Slaver_TIMER0 更新事件 010：Slaver_TIMER1 更新事件 011：Slaver_TIMER2 更新事件 100：Slaver_TIMER3 更新事件 101：Slaver_TIMER4 更新事件 其他值保留。
18:16	ADTG0USRC[2:0]	SHRTIMER_ADCTRIG0 更新源 该位域可以由软件配置，配置 SHRTIMER_ADCTRIGS0 寄存器的更新源。 000：Master_TIMER 更新事件 001：Slaver_TIMER0 更新事件 010：Slaver_TIMER1 更新事件 011：Slaver_TIMER2 更新事件 100：Slaver_TIMER3 更新事件 101：Slaver_TIMER4 更新事件

其他值保留。

15:6	保留	必须保持复位值
5	ST4UPDIS	Slave_TIMER4 更新事件禁能 该位用于使能或禁能更新事件的生成。 0：更新事件使能。 1：更新事件禁能。
4	ST3UPDIS	Slave_TIMER3 更新事件禁能 该位用于使能或禁能更新事件的生成。 0：更新事件使能。 1：更新事件禁能。
3	ST2UPDIS	Slave_TIMER2 更新事件禁能 该位用于使能或禁能更新事件的生成。 0：更新事件使能。 1：更新事件禁能。
2	ST1UPDIS	Slave_TIMER1 更新事件禁能 该位用于使能或禁能更新事件的生成。 0：更新事件使能。 1：更新事件禁能。
1	ST0UPDIS	Slave_TIMER0 更新事件禁能 该位用于使能或禁能更新事件的生成。 0：更新事件使能。 1：更新事件禁能。
0	MTUPDIS	Master_TIMER 更新事件禁能 该位用于使能或禁能更新事件的生成。 0：更新事件使能。 1：更新事件禁能。

SHRTIMER 控制寄存器 1 (SHRTIMER_CTL1)

地址偏移：0x04

复位值：0x0000 0000

该寄存器只能进行字（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		ST4SRST	ST3SRST	ST2SRST	ST1SRST	ST0SRST	MTSRST	保留		ST4SUP	ST3SUP	ST2SUP	ST1SUP	ST0SUP	MTSUP
		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	ST4SRST	Slave_TIMER4 软件复位 该位可由软件置位，硬件自动清除。该位置 1 时，计数器复位。 0：无影响。 1：计数器复位。
12	ST3SRST	Slave_TIMER3 软件复位 该位可由软件置位，硬件自动清除。该位置 1 时，计数器复位。 0：无影响。 1：计数器复位。
11	ST2SRST	Slave_TIMER2 软件复位 该位可由软件置位，硬件自动清除。该位置 1 时，计数器复位。 0：无影响。 1：计数器复位。
10	ST1SRST	Slave_TIMER1 软件复位 该位可由软件置位，硬件自动清除。该位置 1 时，计数器复位。 0：无影响。 1：计数器复位。
9	ST0SRST	Slave_TIMER0 软件复位 该位可由软件置位，硬件自动清除。该位置 1 时，计数器复位。 0：无影响。 1：计数器复位。
8	MTSRST	Master_TIMER 软件复位 该位可由软件置位，硬件自动清除。该位置 1 时，计数器复位。 0：无影响。 1：计数器复位。
7:6	保留	必须保持复位值
5	ST4SUP	Slave_TIMER4 软件更新 该位可由软件置位，硬件自动清除。该位置 1 时，影子寄存器的内容被传送到有效寄存器，并且所有挂起的更新请求都被取消。 0：无影响。 1：更新生成。
4	ST3SUP	Slave_TIMER3 软件更新 该位可由软件置位，硬件自动清除。该位置 1 时，影子寄存器的内容被传送到有效寄存器，并且所有挂起的更新请求都被取消。 0：无影响。 1：更新生成。
3	ST2SUP	Slave_TIMER2 软件更新 该位可由软件置位，硬件自动清除。该位置 1 时，影子寄存器的内容被传送到有效寄存器，并且所有挂起的更新请求都被取消。

0：无影响。
1：更新生成。

2	ST1SUP	Slave_TIMER1 软件更新 该位可由软件置位，硬件自动清除。该位置 1 时，影子寄存器的内容被传送到有效寄存器，并且所有挂起的更新请求都被取消。 0：无影响。 1：更新生成。
1	ST0SUP	Slave_TIMER0 软件更新 该位可由软件置位，硬件自动清除。该位置 1 时，影子寄存器的内容被传送到有效寄存器，并且所有挂起的更新请求都被取消。 0：无影响。 1：更新生成。
0	MTSUP	Master_TIMER 软件更新 该位可由软件置位，硬件自动清除。该位置 1 时，影子寄存器的内容被传送到有效寄存器，并且所有挂起的更新请求都被取消。 0：无影响。 1：更新生成。

SHRTIMER 中断标志寄存器 (SHRTIMER_INTF)

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留										SYSFLTIF	FLT4IF	FLT3IF	FLT2IF	FLT1IF	FLT0IF
										r	r	r	r	r	r

位/位域	名称	描述
31:18	保留	必须保持复位值
17	BMPERIF	突发模式周期中断标志 突发模式的周期时间到达时，该位由硬件置位。可以通过软件写 1 清零。 0：突发模式周期中断未发生 1：突发模式周期中断发生
16	DLLCALIF	DLL 校准完成中断标志 DLL 校准完成后，此标志由硬件置 1。可以通过软件写 1 清零。 0：DLL 校准完成中断未发生

		1：DLL 校准完成中断发生
15:6	保留	必须保持复位值
5	SYSFLTIF	系统故障中断标志 系统故障发生时，此标志由硬件置 1。可以通过软件写 1 清零。 0：系统故障中断未发生 1：系统故障中断发生
4	FLT4IF	故障 4 中断标志 请参考 FLT0IF 描述。
3	FLT3IF	故障 3 中断标志 请参考 FLT0IF 描述。
2	FLT2IF	故障 2 中断标志 请参考 FLT0IF 描述。
1	FLT1IF	故障 1 中断标志 请参考 FLT0IF 描述。
0	FLT0IF	故障 0 中断标志 故障 0 发生时，此标志由硬件置 1。可以通过软件写 1 清零。 0：故障 0 中断未发生 1：故障 0 中断发生

SHRTIMER 中断标志清除寄存器 (SHRTIMER_INTC)

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留														BMPERIFC	DLLCALIFC
														w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留										SYSFLTIFC	FLT4IFC	FLT3IFC	FLT2IFC	FLT1IFC	FLT0IFC
										w	w	w	w	w	w

位/位域	名称	描述
31:18	保留	必须保持复位值
17	BMPERIFC	突发模式周期中断标志清除 该位软件写 1 可以清零 SHRTIMER_INTF 寄存器中的 BMPERIF 位。 0：无影响 1：突发模式周期中断标志清除

16	DLLCALIFC	DLL 校准完成中断标志清除 该位软件写 1 可以清零 SHRTIMER_INTF 寄存器中的 DLLCALIF 位。 0：无影响 1：DLL 校准完成中断标志清除
15:6	保留	必须保持复位值
5	SYSFLTIFC	系统故障中断标志清除 该位软件写 1 可以清零 SHRTIMER_INTF 寄存器中的 SYSFLTIF 位。 0：无影响 1：系统故障中断标志清除
4	FLT4IFC	故障 4 中断标志清除 该位软件写 1 可以清零 SHRTIMER_INTF 寄存器中的 FLT4IF 位。 0：无影响 1：故障 4 中断标志清除
3	FLT3IFC	故障 3 中断标志清除 该位软件写 1 可以清零 SHRTIMER_INTF 寄存器中的 FLT3IF 位。 0：无影响 1：故障 3 中断标志清除
2	FLT2IFC	故障 2 中断标志清除 该位软件写 1 可以清零 SHRTIMER_INTF 寄存器中的 FLT2IF 位。 0：无影响 1：故障 2 中断标志清除
1	FLT1IFC	故障 1 中断标志清除 该位软件写 1 可以清零 SHRTIMER_INTF 寄存器中的 FLT1IF 位。 0：无影响 1：故障 1 中断标志清除
0	FLT0IFC	故障 0 中断标志清除 该位软件写 1 可以清零 SHRTIMER_INTF 寄存器中的 FLT0IF 位。 0：无影响 1：故障 0 中断标志清除

SHRTIMER 中断使能寄存器 (SHRTIMER_INTEN)

地址偏移：0x10

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

保留	SYSFLTIE	保留
	rw	

位/位域	名称	描述
31:18	保留	必须保持复位值
17	BMPERIE	突发模式周期中断使能 0：禁能 1：使能
16	DLLCALIE	DLL 校准完成中断使能 0：禁能 1：使能
15:6	保留	必须保持复位值
5	SYSFLTIE	系统故障中断使能 0：禁能 1：使能
4:0	保留	必须保持复位值

SHRTIMER 通道输出使能寄存器 (SHRTIMER_CHOUTEN)

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能进行字（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						ST4CH1EN	ST4CH0EN	ST3CH1EN	ST3CH0EN	ST2CH1EN	ST2CH0EN	ST1CH1EN	ST1CH0EN	ST0CH1EN	ST0CH0EN
						rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

位/位域	名称	描述
31:10	保留	必须保持复位值
9	ST4CH1EN	Slave_TIMER4 通道 1 输出（ST4CH1_O）使能 请参考 ST0CH0EN 描述。 注意：禁能状态对应空闲和故障状态，由 SHRTIMER_CHOUTDISF 寄存器中的 ST4CH1DISF 位配置。
8	ST4CH0EN	Slave_TIMER4 通道 0 输出（ST4CH0_O）使能 请参考 ST0CH0EN 描述。

		注意：禁能状态对应空闲和故障状态，由 SHRTIMER_CHOUTDISF 寄存器中的 ST4CH0DISF 位配置。
7	ST3CH1EN	Slave_TIMER3 通道 1 输出（ST3CH1_O）使能 请参考 ST0CH0EN 描述。 注意：禁能状态对应空闲和故障状态，由 SHRTIMER_CHOUTDISF 寄存器中的 ST3CH1DISF 位配置。
6	ST3CH0EN	Slave_TIMER3 通道 0 输出（ST3CH0_O）使能 请参考 ST0CH0EN 描述。 注意：禁能状态对应空闲和故障状态，由 SHRTIMER_CHOUTDISF 寄存器中的 ST3CH0DISF 位配置。
5	ST2CH1EN	Slave_TIMER2 通道 1 输出（ST2CH1_O）使能 请参考 ST0CH0EN 描述。 注意：禁能状态对应空闲和故障状态，由 SHRTIMER_CHOUTDISF 寄存器中的 ST2CH1DISF 位配置。
4	ST2CH0EN	Slave_TIMER2 通道 0 输出（ST2CH0_O）使能 请参考 ST0CH0EN 描述。 注意：禁能状态对应空闲和故障状态，由 SHRTIMER_CHOUTDISF 寄存器中的 ST2CH0DISF 位配置。
3	ST1CH1EN	Slave_TIMER1 通道 1 输出（ST1CH1_O）使能 请参考 ST0CH0EN 描述。 注意：禁能状态对应空闲和故障状态，由 SHRTIMER_CHOUTDISF 寄存器中的 ST1CH1DISF 位配置。
2	ST1CH0EN	Slave_TIMER1 通道 0 输出（ST1CH0_O）使能 请参考 ST0CH0EN 描述。 注意：禁能状态对应空闲和故障状态，由 SHRTIMER_CHOUTDISF 寄存器中的 ST1CH0DISF 位配置。
1	ST0CH1EN	Slave_TIMER0 通道 1 输出（ST0CH1_O）使能 请参考 ST0CH0EN 描述。 注意：禁能状态对应空闲和故障状态，由 SHRTIMER_CHOUTDISF 寄存器中的 ST0CH1DISF 位配置。
0	ST0CH0EN	Slave_TIMER0 通道 0 输出（ST0CH0_O）使能 该位写 1 使能输出，写 0 无影响。 故障输入有效时，该位由硬件异步清除。 读该位内容，将返回输出使能或禁能的状态。 0：Slave_TIMER0 通道 0 输出 ST0CH0_O 禁能。输出处于故障状态或空闲状态。 1：Slave_TIMER0 通道 0 输出 ST0CH0_O 使能。 注意：禁能状态对应空闲和故障状态，由 SHRTIMER_CHOUTDISF 寄存器中的 ST0CH0DISF 位配置。

SHRTIMER 通道输出禁能寄存器 (SHRTIMER_CHOUTDIS)

地址偏移：0x18

复位值：0x0000 0000

该寄存器只能进行字（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						ST4CH1DIS	ST4CH0DIS	ST3CH1DIS	ST3CH0DIS	ST2CH1DIS	ST2CH0DIS	ST1CH1DIS	ST1CH0DIS	ST0CH1DIS	ST0CH0DIS
						w	w	w	w	w	w	w	w	w	w

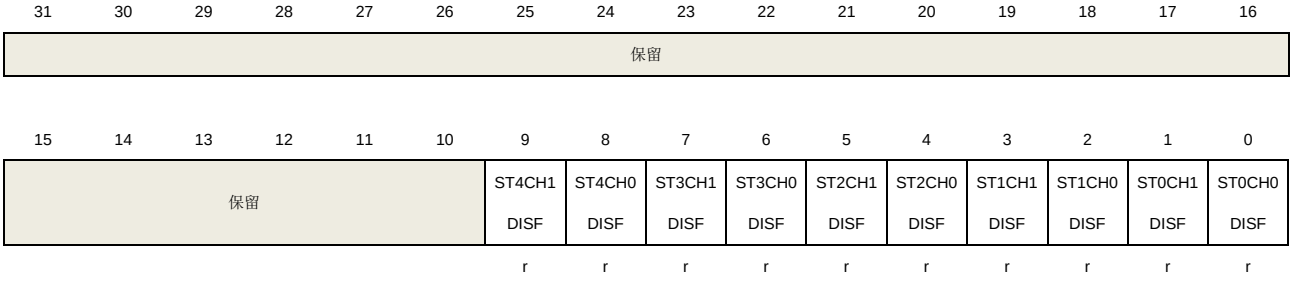
位/位域	名称	描述
31:10	保留	必须保持复位值
9	ST4CH1DIS	Slave_TIMER4 通道 1 输出（ST4CH1_O）禁能 请参考 ST0CH0DIS 描述。
8	ST4CH0DIS	Slave_TIMER4 通道 0 输出（ST4CH0_O）禁能 请参考 ST0CH0DIS 描述。
7	ST3CH1DIS	Slave_TIMER3 通道 1 输出（ST3CH1_O）禁能 请参考 ST0CH0DIS 描述。
6	ST3CH0DIS	Slave_TIMER3 通道 0 输出（ST3CH0_O）禁能 请参考 ST0CH0DIS 描述。
5	ST2CH1DIS	Slave_TIMER2 通道 1 输出（ST2CH1_O）禁能 请参考 ST0CH0DIS 描述。
4	ST2CH0DIS	Slave_TIMER2 通道 0 输出（ST2CH0_O）禁能 请参考 ST0CH0DIS 描述。
3	ST1CH1DIS	Slave_TIMER1 通道 1 输出（ST1CH1_O）禁能 请参考 ST0CH0DIS 描述。
2	ST1CH0DIS	Slave_TIMER1 通道 0 输出（ST1CH0_O）禁能 请参考 ST0CH0DIS 描述。
1	ST0CH1DIS	Slave_TIMER0 通道 1 输出（ST0CH1_O）禁能 请参考 ST0CH0DIS 描述。
0	ST0CH0DIS	Slave_TIMER1 通道 0 输出（ST1CH0_O）禁能 该位写 1 禁能输出，通道 0 进入空闲状态。写 0 无影响。 0：无影响。 1：Slave_TIMER1 通道 0 输出 ST0CH0_O 禁能。输出从故障状态或运行状态进入空闲状态。

SHRTIMER 通道输出禁能标志寄存器 (SHRTIMER_CHOUTDISF)

地址偏移：0x1C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。



位/位域	名称	描述
31:10	保留	必须保持复位值
9	ST4CH1DISF	Slave_TIMER4 通道 1 输出（ST4CH1_O）禁能标志 请参考 ST0CH0DISF 描述。
8	ST4CH0DISF	Slave_TIMER4 通道 0 输出（ST4CH0_O）禁能标志 请参考 ST0CH0DISF 描述。
7	ST3CH1DISF	Slave_TIMER3 通道 1 输出（ST3CH1_O）禁能标志 请参考 ST0CH0DISF 描述。
6	ST3CH0DISF	Slave_TIMER3 通道 0 输出（ST3CH0_O）禁能标志 请参考 ST0CH0DISF 描述。
5	ST2CH1DISF	Slave_TIMER2 通道 1 输出（ST2CH1_O）禁能标志 请参考 ST0CH0DISF 描述。
4	ST2CH0DISF	Slave_TIMER2 通道 0 输出（ST2CH0_O）禁能标志 请参考 ST0CH0DISF 描述。
3	ST1CH1DISF	Slave_TIMER1 通道 1 输出（ST1CH1_O）禁能标志 请参考 ST0CH0DISF 描述。
2	ST1CH0DISF	Slave_TIMER1 通道 0 输出（ST1CH0_O）禁能标志 请参考 ST0CH0DISF 描述。
1	ST0CH1DISF	Slave_TIMER0 通道 1 输出（ST0CH1_O）禁能标志 请参考 ST0CH0DISF 描述。
0	ST0CH0DISF	Slave_TIMER0 通道 0 输出（ST0CH0_O）禁能标志 读该位内容，将返回通道 0 输出禁能状态。输出使能时无效。 0：空闲状态下，Slave_TIMER0 通道 0 输出 ST0CH0_O 禁能。 1：故障状态下，Slave_TIMER0 通道 0 输出 ST0CH0_O 禁能。

SHRTIMER 突发模式控制寄存器 (SHRTIMER_BMCTL)

地址偏移：0x20

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BMOPTF	保留									BMST4	BMST3	BMST2	BMST1	BMST0	BMMT
rc_w0										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					BMSE	BMPSC[3:0]				BMCLKS[3:0]				BMCTN	BMEN
					rw	rw				rw				rw	rw

位/位域	名称	描述
31	BMOPTF	突发模式运行标志 突发模式正在运行时，该标志位由硬件置位。该位写 0 将停止突发模式。 0：常规运行，突发模式不起作用。 1：突发模式正在进行。
30:22	保留	必须保持复位值
21	BMST4	Slave_TIMER4 突发模式 0：Slave_TIMER4 计数器时钟（SHRTIMER_PSCCK）保持，且计数器正常运行。 1：Slave_TIMER4 计数器时钟（SHRTIMER_PSCCK）停止，且计数器复位。 注意： （1）突发模式使能时，无法更改此位。 （2）当均衡空闲模式有效（SHRTIMER_STxCHOCTL 寄存器中的 DLYISCH [2:0] = 3'bx11）时，不能将此位置位。
20	BMST3	Slave_TIMER3 突发模式 0：Slave_TIMER3 计数器时钟（SHRTIMER_PSCCK）保持，且计数器正常运行。 1：Slave_TIMER3 计数器时钟（SHRTIMER_PSCCK）停止，且计数器复位。 注意： （1）突发模式使能时，无法更改此位。 （2）当均衡空闲模式有效（SHRTIMER_STxCHOCTL 寄存器中的 DLYISCH [2:0] = 3'bx11）时，不能将此位置位。
19	BMST2	Slave_TIMER2 突发模式 0：Slave_TIMER2 计数器时钟（SHRTIMER_PSCCK）保持，且计数器正常运行。 1：Slave_TIMER2 计数器时钟（SHRTIMER_PSCCK）停止，且计数器复位。 注意： （1）突发模式使能时，无法更改此位。 （2）当均衡空闲模式有效（SHRTIMER_STxCHOCTL 寄存器中的 DLYISCH [2:0] = 3'bx11）时，不能将此位置位。
18	BMST1	Slave_TIMER1 突发模式 0：Slave_TIMER1 计数器时钟（SHRTIMER_PSCCK）保持，且计数器正常运行。

		1：Slave_TIMER1 计数器时钟（SHRTIMER_PSCCK）停止，且计数器复位。 注意： （1）突发模式使能时，无法更改此位。 （2）当均衡空闲模式有效（SHRTIMER_STxCHOCTL 寄存器中的 DLYISCH [2:0] = 3'bx11）时，不能将此位置位。
17	BMST0	Slave_TIMER0 突发模式 0：Slave_TIMER0 计数器时钟（SHRTIMER_PSCCK）保持，且计数器正常运行。 1：Slave_TIMER0 计数器时钟（SHRTIMER_PSCCK）停止，且计数器复位。 注意： （1）突发模式使能时，无法更改此位。 （2）当均衡空闲模式有效（SHRTIMER_STxCHOCTL 寄存器中的 DLYISCH [2:0] = 3'bx11）时，不能将此位置位。
16	BMMT	Master_TIMER 突发模式 0：Master_TIMER 计数器时钟（SHRTIMER_PSCCK）保持，且计数器正常运行。 1：Master_TIMER 计数器时钟（SHRTIMER_PSCCK）停止，且计数器复位。
15:11	保留	必须保持复位值
10	BMSE	突发模式影子寄存器使能 0：SHRTIMER_BMCMPV 和 SHRTIMER_BMCAR 寄存器的影子寄存器禁能。 1：SHRTIMER_BMCMPV 和 SHRTIMER_BMCAR 寄存器的影子寄存器使能。 注意：突发模式使能时不能更改此位。
9:6	BMPSC[3:0]	突发模式时钟分频 该位域可以由软件配置，当 SHRTIMER_BMCTL 寄存器中的 BMCLKS [3:0] = 4'b1010 时，确定超高分辨率时钟（SHRTIMER_HPCK）和突发模式计数器时钟（SHRTIMER_BMCNTCK）的分频比。 $f_{SHRTIMER_BMCNTCK} = f_{SHRTIMER_CK}/2^{BMPSC[3:0]}$ 0000：$f_{SHRTIMER_BMCNTCK} = f_{SHRTIMER_CK}$ 0001：$f_{SHRTIMER_BMCNTCK} = f_{SHRTIMER_CK}/2$ 0010：$f_{SHRTIMER_BMCNTCK} = f_{SHRTIMER_CK}/4$ 0011：$f_{SHRTIMER_BMCNTCK} = f_{SHRTIMER_CK}/8$ 0100：$f_{SHRTIMER_BMCNTCK} = f_{SHRTIMER_CK}/16$ 0101：$f_{SHRTIMER_BMCNTCK} = f_{SHRTIMER_CK}/32$ 0110：$f_{SHRTIMER_BMCNTCK} = f_{SHRTIMER_CK}/64$ 0111：$f_{SHRTIMER_BMCNTCK} = f_{SHRTIMER_CK}/128$ 1000：$f_{SHRTIMER_BMCNTCK} = f_{SHRTIMER_CK}/256$ 1001：$f_{SHRTIMER_BMCNTCK} = f_{SHRTIMER_CK}/512$ 1010：$f_{SHRTIMER_BMCNTCK} = f_{SHRTIMER_CK}/1024$ 1011：$f_{SHRTIMER_BMCNTCK} = f_{SHRTIMER_CK}/2048$ 1100：$f_{SHRTIMER_BMCNTCK} = f_{SHRTIMER_CK}/4096$ 1101：$f_{SHRTIMER_BMCNTCK} = f_{SHRTIMER_CK}/8192$ 1110：$f_{SHRTIMER_BMCNTCK} = f_{SHRTIMER_CK}/16384$ 1111：$f_{SHRTIMER_BMCNTCK} = f_{SHRTIMER_CK}/32768$

注意：突发模式使能时，该位不能修改。

5:2	BMCLKS[3:0]	突发模式时钟源 该位域用于配置突发模式计数器的时钟源。 0000：Master_TIMER 计数器复位/翻转事件。 0001：Slave_TIMER0 计数器复位/翻转事件。 0010：Slave_TIMER1 计数器复位/翻转事件。 0011：Slave_TIMER2 计数器复位/翻转事件。 0100：Slave_TIMER3 计数器复位/翻转事件。 0101：Slave_TIMER4 计数器复位/翻转事件。 0110：芯片内部信号 0，BMCLK0。 0111：芯片内部信号 1，BMCLK1。 1000：芯片内部信号 2，BMCLK2。 1001：芯片内部信号 3，BMCLK3。 1010：$F_{SHRTIMER_CK}$ 时钟预分频（根据 BMPRSC [3:0]设置）。 其他值保留。 注意： （1）突发模式使能时，无法更改此位 （2）BMCLKy（y = 0..3）：请参考*表 19-14. 突发模式的芯片内部信号*。
1	BMCTN	突发模式下的连续模式 0：单脉冲模式。BM 计数器达到 SHRTIMER_BMCAR 值时，由硬件停止。 1：连续模式。BM 计数器达到 SHRTIMER_BMCAR 值时，翻转到零并连续计数。
0	BMEN	突发模式使能 该位置 1 时，突发模式控制器已准备好接收突发模式启动触发。该位写 0 将终止突发模式。 0：突发模式禁能。 1：突发模式使能。

SHRTIMER 突发模式启动触发寄存器 (SHRTIMER_BMSTRG)

地址偏移：0x24

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CISGN	EXEV7	EXEV6	ST3EXEV7	ST0EXEV6	ST4CMP1	ST4CMP0	ST4REP	ST4RST	ST3CMP1	ST3CMP0	ST3REP	ST3RST	ST2CMP1	ST2CMP0	ST2REP
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST2RST	ST1CMP1	ST1CMP0	ST1REP	ST1RST	ST0CMP1	ST0CMP0	ST0REP	ST0RST	MTCMP3	MTCMP2	MTCMP1	MTCMP0	MTREP	MTRST	SWTRG
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述

31	CISGN	芯片内部信号触发突发模式 芯片内部信号（TIMER6_TRGO）启动突发模式。 0：对突发模式无影响。 1：芯片内部信号启动突发模式。
30	EXEV7	外部事件 7 触发突发模式 外部事件 7 启动突发模式。 0：对突发模式无影响。 1：外部事件 7 启动突发模式。
29	EXEV6	外部事件 6 触发突发模式 外部事件 6 启动突发模式。 0：对突发模式无影响。 1：外部事件 6 启动突发模式。
28	ST3EXEV7	外部事件 7 之后的 Slave_TIMER3 周期事件触发突发模式 0：对突发模式无影响。 1：外部事件 7 之后的 Slave_TIMER3 周期事件启动突发模式。
27	ST0EXEV6	外部事件 6 之后的 Slave_TIMER0 周期事件触发突发模式 0：对突发模式无影响。 1：外部事件 6 之后的 Slave_TIMER0 周期事件启动突发模式。
26	ST4CMP1	Slave_TIMER4 比较 1 事件触发突发模式。 请参考 MTCMP1 模式。
25	ST4CMP0	Slave_TIMER4 比较 0 事件触发突发模式。 请参考 MTCMP0 模式。
24	ST4REP	Slave_TIMER4 重复事件触发突发模式。 请参考 MTREP 模式。
23	ST4RST	Slave_TIMER4 复位事件触发突发模式。 请参考 MTRST 模式。
22	ST3CMP1	Slave_TIMER3 比较 1 事件触发突发模式。 请参考 MTCMP1 模式。
21	ST3CMP0	Slave_TIMER3 比较 0 事件触发突发模式。 请参考 MTCMP0 模式。
20	ST3REP	Slave_TIMER3 重复事件触发突发模式。 请参考 MTREP 模式。
19	ST3RST	Slave_TIMER3 复位事件触发突发模式。 请参考 MTRST 模式。
18	ST2CMP1	Slave_TIMER2 比较 1 事件触发突发模式。 请参考 MTCMP1 模式。
17	ST2CMP0	Slave_TIMER2 比较 0 事件触发突发模式。

		请参考 MTCMP0 模式。
16	ST2REP	Slave_TIMER2 重复事件触发突发模式。 请参考 MTREP 模式。
15	ST2RST	Slave_TIMER2 复位事件触发突发模式。 请参考 MTRST 模式。
14	ST1CMP1	Slave_TIMER1 比较 1 事件触发突发模式。 请参考 MTCMP1 模式。
13	ST1CMP0	Slave_TIMER1 比较 0 事件触发突发模式。 请参考 MTCMP0 模式。
12	ST1REP	Slave_TIMER1 重复事件触发突发模式。 请参考 MTREP 模式。
11	ST1RST	Slave_TIMER1 复位事件触发突发模式。 请参考 MTRST 模式。
10	ST0CMP1	Slave_TIMER0 比较 1 事件触发突发模式。 请参考 MTCMP1 模式。
9	ST0CMP0	Slave_TIMER0 比较 0 事件触发突发模式。 请参考 MTCMP0 模式。
8	ST0REP	Slave_TIMER0 重复事件触发突发模式。 请参考 MTREP 模式。
7	ST0RST	Slave_TIMER0 复位事件触发突发模式。 请参考 MTRST 模式。
6	MTCMP3	Master_TIMER 比较 3 事件触发突发模式。 请参考 MTCMP0 模式。
5	MTCMP2	Master_TIMER 比较 2 事件触发突发模式。 请参考 MTCMP0 模式。
4	MTCMP1	Master_TIMER 比较 1 事件触发突发模式。 请参考 MTCMP0 模式。
3	MTCMP0	Master_TIMER 比较 0 事件触发突发模式。 0：对突发模式无影响。 1：Master_TIMER 比较 0 事件启动突发模式。
2	MTREP	Master_TIMER 重复事件触发突发模式。 0：对突发模式无影响。 1：Master_TIMER 重复事件启动突发模式。
1	MTRST	Master_TIMER 复位事件触发突发模式。 0：对突发模式无影响。 1：Master_TIMER 复位事件启动突发模式。

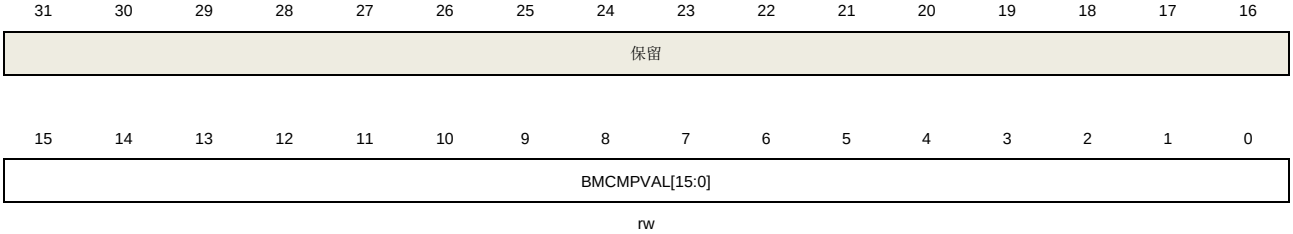
0	SWTRG	软件触发突发模式 该位由软件置 1，硬件自动清零。该位置 1 时，触发突发模式。如果突发模式未使能，则该位无效（SHRTIMER_BMCTL 寄存器中的 BMEN 位复位）。 0：对突发模式无影响。 1：软件触发启动突发模式。 注意：如果突发模式未使能（BMEN 位复位），则此位无效。

SHRTIMER 突发模式比较值寄存器 (SHRTIMER_BMCMPV)

地址偏移：0x28

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。



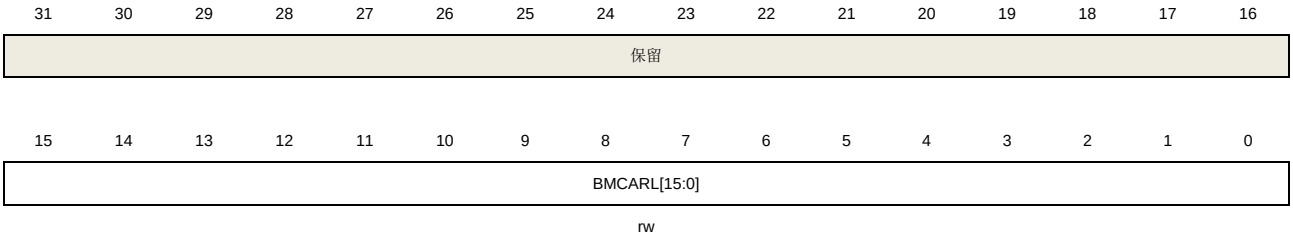
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	BMCMPVAL[15:0]	突发模式比较值 该位域包含了与BM计数器进行比较的值，并定义了空闲的持续时间。 该寄存器有影子寄存器，可以进行预装载。 注意：当预分频系数为0，$f_{SHRTIMER_CK}$时钟直接作为突发模式时钟源（BMCLKS [3:0] = 4'b1010和BMPSC [3:0] = 4）时，BMCMPVAL[15:0]不能设置为0x0000。

SHRTIMER 突发模式计数器自动重载寄存器 (SHRTIMER_BMCAR)

地址偏移：0x2C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。



位/位域	名称	描述
31:16	保留	必须保持复位值

15:0	BMCARL[15:0]	突发模式计数器自动重载值 该位域配置了 BM 计数器的自动重载值，并定义了突发模式的周期，该周期是空闲状态和运行状态持续的时间之和。 该寄存器具有影子寄存器，可以进行预加载。 注意：突发模式使能时，该位域不能为零。

SHRTIMER 外部事件配置寄存器 0 (SHRTIMER_EXEVCFG0)

地址偏移：0x30

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			EXEV4EG[1:0]		EXEV4P	EXEV4SRC[1:0]		保留	EXEV3EG[1:0]		EXEV3P	EXEV3SRC[1:0]		保留	EXEV2EG[1]
			rw		rw	rw			rw		rw	rw			rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXEV2EG[0]	EXEV2P	EXEV2SRC[1:0]		保留	EXEV1EG[1:0]		EXEV1P	EXEV1SRC[1:0]		保留	EXEV0EG[1:0]		EXEV0P	EXEV0SRC[1:0]	
rw	rw	rw			rw		rw	rw			rw		rw	rw	

位/位域	名称	描述
31:29	保留	必须保持复位值
28:27	EXEV4EG[1:0]	外部事件 4 有效沿 请参考 EXEV0EG [1:0]说明。
26	EXEV4P	外部事件 4 极性 请参考 EXEV0P 说明。
25:24	EXEV4SRC[1:0]	外部事件 4 的源 请参考 EXEV0SRC[1:0]说明。
23	保留	必须保持复位值
22:21	EXEV3EG[1:0]	外部事件 3 有效沿 请参考 EXEV0EG [1:0]说明。
20	EXEV3P	外部事件 3 极性 请参考 EXEV0P 说明。
19:18	EXEV3SRC[1:0]	外部事件 3 的源 请参考 EXEV0SRC[1:0]说明。
17	保留	必须保持复位值
16:15	EXEV2EG[1:0]	外部事件 2 有效沿 请参考 EXEV0EG [1:0]说明。

14	EXEV2P	外部事件 2 极性 请参考 EXEV0P 说明。
13:12	EXEV2SRC[1:0]	外部事件 2 的源 请参考 EXEV0SRC[1:0]说明。
11	保留	必须保持复位值
10:9	EXEV1EG[1:0]	外部事件 1 有效沿 请参考 EXEV0EG [1:0]说明。
8	EXEV1P	外部事件 1 极性 请参考 EXEV0P 说明。
7:6	EXEV1SRC[1:0]	外部事件 1 的源 请参考 EXEV0SRC[1:0]说明。
5	保留	必须保持复位值
4:3	EXEV0EG[1:0]	外部事件 0 有效沿 该位域配置了外部事件 0 的有效沿。 00：电平有效。有效电平由 EXEV0P 位定义。 01：上升沿有效，EXEV0P 位无效。 10：下降沿有效，EXEV0P 位无效 11：上升沿和下降沿均有效，EXEV0P 位无效。
2	EXEV0P	外部事件 0 的极性 当 EXEV0EG [1:0] = 2’b00 时，该位确定了外部事件 0 的有效电平。 0：外部事件 0 高电平有效。 1：外部事件 0 低电平有效。 注意：一旦 Slave_TIMERx 使能，就不能更改该位。
1:0	EXEV0SRC[1:0]	外部事件 0 的源 00：外部事件 0 的源为 EXEV0SRC 0。 01：外部事件 0 的源为 EXEV0SRC 1。 10：外部事件 0 的源为 EXEV0SRC 2。 11：外部事件 0 的源为 EXEV0SRC 3。 注意：一旦 Slave_TIMERx 使能，就不能更改该位。

SHRTIMER 外部事件配置寄存器 1 (SHRTIMER_EXEVCFG1)

地址偏移：0x34

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

15	14	13 12	11	10 9	8	7 6	5	4 3	2	1 0
EXEV7EG[0]	EXEV7P	EXEV7SRC[1:0]	保留	EXEV6EG[1:0]	EXEV6P	EXEV6SRC[1:0]	保留	EXEV5EG[1:0]	EXEV5P	EXEV5SRC[1:0]
rw	rw	rw		rw	rw	rw		rw	rw	rw

位/位域	名称	描述
31:29	保留	必须保持复位值
28:27	EXEV9EG[1:0]	外部事件 9 有效沿 请参考 SHRTIMER_EXEVCFG0 寄存器中的 EXEV0EG [1:0]位域说明。
26	EXEV9P	外部事件 9 极性 请参考 SHRTIMER_EXEVCFG0 寄存器中的 EXEV0P 位说明。
25:24	EXEV9SRC[1:0]	外部事件 9 的源 请参考 SHRTIMER_EXEVCFG0 寄存器中的 EXEV0SRC[1:0]位域说明。
23	保留	必须保持复位值
22:21	EXEV8EG[1:0]	外部事件 8 有效沿 请参考 SHRTIMER_EXEVCFG0 寄存器中的 EXEV0EG [1:0]位域说明。
20	EXEV8P	外部事件 8 极性 请参考 SHRTIMER_EXEVCFG0 寄存器中的 EXEV0P 位说明。
19:18	EXEV8SRC[1:0]	外部事件 8 的源 请参考 SHRTIMER_EXEVCFG0 寄存器中的 EXEV0SRC[1:0]位域说明。
17	保留	必须保持复位值
16:15	EXEV7EG[1:0]	外部事件 7 有效沿 请参考 SHRTIMER_EXEVCFG0 寄存器中的 EXEV0EG [1:0]位域说明。
14	EXEV7P	外部事件 7 极性 请参考 SHRTIMER_EXEVCFG0 寄存器中的 EXEV0P 位说明。
13:12	EXEV7SRC[1:0]	外部事件 7 的源 请参考 SHRTIMER_EXEVCFG0 寄存器中的 EXEV0SRC[1:0]位域说明。
11	保留	必须保持复位值
10:9	EXEV6EG[1:0]	外部事件 6 有效沿 请参考 SHRTIMER_EXEVCFG0 寄存器中的 EXEV0EG [1:0]位域说明。
8	EXEV6P	外部事件 6 极性 请参考 SHRTIMER_EXEVCFG0 寄存器中的 EXEV0P 位说明。
7:6	EXEV6SRC[1:0]	外部事件 6 的源 请参考 SHRTIMER_EXEVCFG0 寄存器中的 EXEV0SRC[1:0]位域说明。
5	保留	必须保持复位值

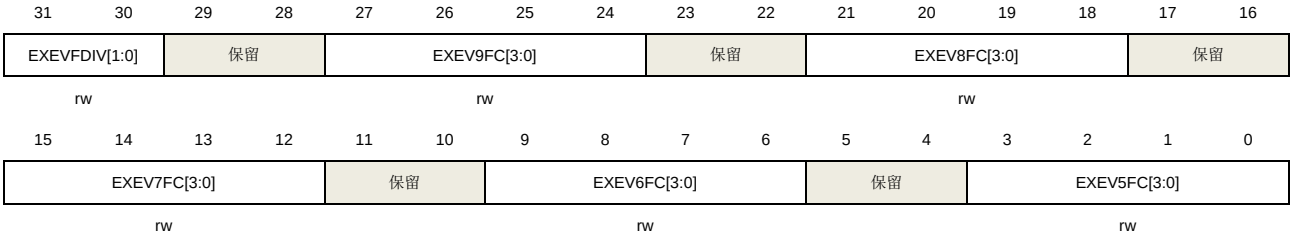
4:3	EXEV5EG[1:0]	外部事件 5 有效沿 请参考 SHRTIMER_EXEVCFG0 寄存器中的 EXEV0EG [1:0]位域说明。
2	EXEV5P	外部事件 5 极性 请参考 SHRTIMER_EXEVCFG0 寄存器中的 EXEV0P 位说明。
1:0	EXEV5SRC[1:0]	外部事件 5 的源 请参考 SHRTIMER_EXEVCFG0 寄存器中的 EXEV0SRC[1:0]位域说明。

SHRTIMER 外部事件数字滤波控制寄存器 (SHRTIMER_EXEVDFCTL)

地址偏移：0x38

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。



位/位域	名称	描述
31:30	EXEVFDIV[1:0]	外部事件数字滤波器时钟分频 该位域由软件配置，确定 SHRTIMER 时钟（SHRTIMER_CK）和外部事件数字滤波器时钟（SHRTIMER_EXEVFCK）之间的分频比。 $f_{SHRTIMER_EXEVFCK} = f_{SHRTIMER_CK}/2^{EXEVFDIV[2:0]}$。 00：$f_{SHRTIMER_EXEVFCK} = f_{SHRTIMER_CK}$。 01：$f_{SHRTIMER_EXEVFCK} = f_{SHRTIMER_CK}/2$。 10：$f_{SHRTIMER_EXEVFCK} = f_{SHRTIMER_CK}/4$。 11：$f_{SHRTIMER_EXEVFCK} = f_{SHRTIMER_CK}/8$。
29:28	保留	必须保持复位值
27:24	EXEV9FC[3:0]	外部事件 9 滤波控制 请参考 EXEV5FC [3:0]说明。
23:22	保留	必须保持复位值
21:18	EXEV8FC[3:0]	外部事件 8 滤波控制 请参考 EXEV5FC [3:0]说明。
17:16	保留	必须保持复位值
15:12	EXEV7FC[3:0]	外部事件 7 滤波控制 请参考 EXEV5FC [3:0]说明。
11:10	保留	必须保持复位值

9:6	EXEV6FC[3:0]	外部事件 5 滤波控制 请参考 EXEV5FC [3:0]说明。
5:4	保留	必须保持复位值
3:0	EXEV5FC[3:0]	外部事件 5 滤波控制 在数字滤波器中使用事件计数器，N 次输入事件后，输出才会发生转变。该位域用于确定采样外部事件的频率（f_{SAMP}）以及应用于外部事件的数字滤波器的长度。 0000：无滤波。 0001：$f_{SAMP} = f_{SHRTIMER_CK}$，N=2。 0010：$f_{SAMP} = f_{SHRTIMER_CK}$，N=4。 0011：$f_{SAMP} = f_{SHRTIMER_CK}$，N=8。 0100：$f_{SAMP} = f_{SHRTIMER_EXEVFCK}$ /2，N=6。 0101：$f_{SAMP} = f_{SHRTIMER_EXEVFCK}$ /2，N=8。 0110：$f_{SAMP} = f_{SHRTIMER_EXEVFCK}$ /4，N=6。 0111：$f_{SAMP} = f_{SHRTIMER_EXEVFCK}$ /4，N=8。 1000：$f_{SAMP} = f_{SHRTIMER_EXEVFCK}$ /8，N=6。 1001：$f_{SAMP} = f_{SHRTIMER_EXEVFCK}$ /8，N=8。 1010：$f_{SAMP} = f_{SHRTIMER_EXEVFCK}$ /16，N=5。 1011：$f_{SAMP} = f_{SHRTIMER_EXEVFCK}$ /16，N=6。 1100：$f_{SAMP} = f_{SHRTIMER_EXEVFCK}$ /16，N=8。 1101：$f_{SAMP} = f_{SHRTIMER_EXEVFCK}$ /32，N=5。 1110：$f_{SAMP} = f_{SHRTIMER_EXEVFCK}$ /32，N=6。 1111：$f_{SAMP} = f_{SHRTIMER_EXEVFCK}$ /32，N=8。

SHRTIMER ADC 触发源 0 寄存器 (SHRTIMER_ADCTRIGS0)

地址偏移：0x3C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TRG0ST4PER	TRG0ST4C3	TRG0ST4C2	TRG0ST4C1	TRG0ST3PER	TRG0ST3C3	TRG0ST3C2	TRG0ST3C1	TRG0ST2PER	TRG0ST2C3	TRG0ST2C2	TRG0ST2C1	TRG0ST1RST	TRG0ST1PER	TRG0ST1C3	TRG0ST1C2
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRG0ST1C1	TRG0ST0RST	TRG0ST0PER	TRG0ST0C3	TRG0ST0C2	TRG0ST0C1	TRG0EXEV4	TRG0EXEV3	TRG0EXEV2	TRG0EXEV1	TRG0EXEV0	TRG0MTPER	TRG0MTC3	TRG0MTC2	TRG0MTC1	TRG0MTC0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	TRG0ST4PER	Slave_TIMER4周期事件生成SHRTIMER_ADCTRIG0上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置周期事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER4周期事件不生成ADC触发事件。

		1：SHRTIMER Slave_TIMER4周期事件生成ADC触发事件。
30	TRG0ST4C3	Slave_TIMER4比较3事件生成SHRTIMER_ADCTRIG0上的ADC触发事件 请参考 TRG0ST4C1 描述。
29	TRG0ST4C2	Slave_TIMER4比较2事件生成SHRTIMER_ADCTRIG0上的ADC触发事件 请参考 TRG0ST4C1 描述。
28	TRG0ST4C1	Slave_TIMER4比较1事件生成SHRTIMER_ADCTRIG0上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置比较1事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER4比较1事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER4比较1事件生成ADC触发事件。
27	TRG0ST3PER	Slave_TIMER3周期事件生成SHRTIMER_ADCTRIG0上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置周期事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER3周期事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER3周期事件生成ADC触发事件。
26	TRG0ST3C3	Slave_TIMER3比较3事件生成SHRTIMER_ADCTRIG0上的ADC触发事件 请参考 TRG0ST3C1 描述。
25	TRG0ST3C2	Slave_TIMER3比较2事件生成SHRTIMER_ADCTRIG0上的ADC触发事件 请参考 TRG0ST3C1 描述。
24	TRG0ST3C1	Slave_TIMER3比较1事件生成SHRTIMER_ADCTRIG0上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置比较1事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER3比较1事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER3比较1事件生成ADC触发事件。
23	TRG0ST2PER	Slave_TIMER2周期事件生成SHRTIMER_ADCTRIG0上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置周期事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER2周期事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER2周期事件生成ADC触发事件。
22	TRG0ST2C3	Slave_TIMER3比较3事件生成SHRTIMER_ADCTRIG0上的ADC触发事件 请参考 TRG0ST2C1 描述。
21	TRG0ST2C2	Slave_TIMER3比较2事件生成SHRTIMER_ADCTRIG0 请参考 TRG0ST2C1 描述。
20	TRG0ST2C1	Slave_TIMER2比较1事件生成SHRTIMER_ADCTRIG0上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置比较1事件是否生成ADC触发事件。 0：SHRTIMER Slave_TIMER2比较1事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER2比较1事件生成ADC触发事件。

19	TRG0ST1RST	Slave_TIMER1复位事件生成SHRTIMER_ADCTRIG0上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置生成ADC触发事件。 0：SHRTIMER Slave_TIMER1复位事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER1复位事件生成ADC触发事件。
18	TRG0ST1PER	Slave_TIMER1周期事件生成SHRTIMER_ADCTRIG0上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置周期事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER1周期事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER1周期事件生成ADC触发事件。
17	TRG0ST1C3	Slave_TIMER1比较3事件生成SHRTIMER_ADCTRIG0 请参考 TRG0ST1C1 描述。
16	TRG0ST1C2	Slave_TIMER1比较2事件生成SHRTIMER_ADCTRIG0 请参考 TRG0ST1C1 描述。
15	TRG0ST1C1	Slave_TIMER1比较1事件生成SHRTIMER_ADCTRIG0上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置比较1事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER1比较1事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER1比较1事件生成ADC触发事件。
14	TRG0ST0RST	Slave_TIMER0复位事件生成SHRTIMER_ADCTRIG0上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置事件是否生成ADC触发事件。 0：SHRTIMER Slave_TIMER0复位事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER0复位事件生成ADC触发事件。
13	TRG0ST0PER	Slave_TIMER0周期事件生成SHRTIMER_ADCTRIG0上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置周期事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER0周期事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER0周期事件生成ADC触发事件。
12	TRG0ST0C3	Slave_TIMER0比较3事件生成SHRTIMER_ADCTRIG0上的ADC触发事件 请参考 TRG0ST0C1 描述。
11	TRG0ST0C2	Slave_TIMER0比较2事件生成SHRTIMER_ADCTRIG0上的ADC触发事件 请参考 TRG0ST0C1 描述。
10	TRG0ST0C1	Slave_TIMER0比较1事件生成SHRTIMER_ADCTRIG0上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置比较1事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER0比较1事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER0比较1事件生成ADC触发事件。
9	TRG0EXEV4	外部事件4生成生成SHRTIMER_ADCTRIG0上的ADC触发事件

请参考 TRG0EXEV0 描述。

8 TRG0EXEV3 外部事件3生成生成SHRTIMER_ADCTRIG0上的ADC触发事件
请参考 TRG0EXEV0 描述。

7 TRG0EXEV2 外部事件2生成生成SHRTIMER_ADCTRIG0上的ADC触发事件
请参考 TRG0EXEV0 描述。

6 TRG0EXEV1 外部事件1生成生成SHRTIMER_ADCTRIG0上的ADC触发事件
请参考 TRG0EXEV0 描述。

5 TRG0EXEV0 外部事件0生成SHRTIMER_ADCTRIG0上的ADC触发事件
SHRTIMER可以在SHRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置事件生成ADC触发事件。
0：外部事件0（EXEV0C）不生成ADC触发事件。
1：外部事件0（EXEV0C）生成ADC触发事件。

4 TRG0MTPER Master_TIMER周期事件生成SHRTIMER_ADCTRIG0上的ADC触发事件
SHRTIMER可以在SHRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置周期事件生成ADC触发事件。
0：SHRTIMER Master_TIMER周期事件不生成ADC触发事件。
1：SHRTIMER Master_TIMER周期事件生成ADC触发事件。

3 TRG0MTC3 Master_TIMER比较3事件生成SHRTIMER_ADCTRIG0上的ADC触发事件
请参考TRG0MTC0描述。

2 TRG0MTC2 Master_TIMER比较2事件生成SHRTIMER_ADCTRIG0上的ADC触发事件
请参考TRG0MTC0描述。

1 TRG0MTC1 Master_TIMER比较1事件生成SHRTIMER_ADCTRIG0上的ADC触发事件
请参考TRG0MTC0描述。

0 TRG0MTC0 Master_TIMER比较0事件生成SHRTIMER_ADCTRIG0上的ADC触发事件
SHRTIMER可以在SHRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置比较1事件生成ADC触发事件。
0：SHRTIMER Master_TIMER比较0事件不生成ADC触发事件。
1：SHRTIMER Master_TIMER比较0事件生成ADC触发事件。

SHRTIMER ADC 触发源 1 寄存器 (SHRTIMER_ADCTRIGS1)

地址偏移：0x40
复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TRG1ST4RST	TRG1ST4C3	TRG1ST4C2	TRG1ST4C1	TRG1ST3RST	TRG1ST3PER	TRG1ST3C3	TRG1ST3C2	TRG1ST3C1	TRG1ST2RST	TRG1ST2PER	TRG1ST2C3	TRG1ST2C2	TRG1ST2C1	TRG1ST1PER	TRG1ST1C3
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRG1ST1C2	TRG1ST1C1	TRG1ST0PER	TRG1ST0C3	TRG1ST0C2	TRG1ST0C1	TRG1EXEV9	TRG1EXEV8	TRG1EXEV7	TRG1EXEV6	TRG1EXEV5	TRG1MTPER	TRG1MTC3	TRG1MTC2	TRG1MTC1	TRG1MTC0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	TRG1ST4RST	Slave_TIMER4复位事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER4复位事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER4复位事件生成ADC触发事件。
30	TRG1ST4C3	Slave_TIMER4比较3事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 请参考 TRG1ST4C1 描述。
29	TRG1ST4C2	Slave_TIMER4比较2事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 请参考 TRG1ST4C1 描述。
28	TRG1ST4C1	Slave_TIMER4比较1事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG1上生成ADC触发事件。该位用于确定事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER4比较1事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER4比较1事件生成ADC触发事件。
27	TRG1ST3RST	Slave_TIMER3复位事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER3复位事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER3复位事件生成ADC触发事件。
26	TRG1ST3PER	Slave_TIMER3周期事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER3周期事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER3周期事件生成ADC触发事件。
25	TRG1ST3C3	Slave_TIMER3比较3事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 请参考 TRG1ST3C1 描述。
24	TRG1ST3C2	Slave_TIMER3比较2事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 请参考 TRG1ST3C1 描述。
23	TRG1ST3C1	Slave_TIMER3比较1事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER3比较1事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER3比较1事件生成ADC触发事件。

22	TRG1ST2RST	Slave_TIMER2复位事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER2复位事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER2复位事件生成ADC触发事件。
21	TRG1ST2PER	Slave_TIMER2周期事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER2周期事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER2周期事件生成ADC触发事件。
20	TRG1ST2C3	Slave_TIMER2比较3事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 请参考 TRG1ST2C1 描述。
19	TRG1ST2C2	Slave_TIMER2比较2事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 请参考 TRG1ST2C1 描述。
18	TRG1ST2C1	Slave_TIMER2比较1事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER2比较1事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER2比较1事件生成ADC触发事件。
17	TRG1ST1PER	Slave_TIMER1周期事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER1周期事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER1周期事件生成ADC触发事件。
16	TRG1ST1C3	Slave_TIMER1比较3事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 请参考 TRG1ST1C1 描述。
15	TRG1ST1C2	Slave_TIMER1比较2事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 请参考 TRG1ST1C1 描述。
14	TRG1ST1C1	Slave_TIMER1比较1事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER1比较1事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER1比较1事件生成ADC触发事件。
13	TRG1ST0PER	Slave_TIMER0周期事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER0周期事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER0周期事件生成ADC触发事件。
12	TRG1ST0C3	Slave_TIMER0比较3事件生成SHRTIMER_ADCTRIG1上的ADC触发事件

		请参考 TRG1ST0C1 描述。
11	TRG1ST0C2	Slave_TIMER0比较2事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 请参考 TRG1ST0C1 描述。
10	TRG1ST0C1	Slave_TIMER0比较1事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER0比较1事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER0比较1事件生成ADC触发事件。
9	TRG1EXEV9	外部事件9事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 请参考 TRG1EXEV5 描述。
8	TRG1EXEV8	外部事件8事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 请参考 TRG1EXEV5 描述。
7	TRG1EXEV7	外部事件7事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 请参考 TRG1EXEV5 描述。
6	TRG1EXEV6	外部事件6事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 请参考 TRG1EXEV5 描述。
5	TRG1EXEV5	外部事件5生成SHRTIMER_ADCTRIG1上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：外部事件5（EXEV5C）不生成ADC触发事件。 1：外部事件5（EXEV5C）生成ADC触发事件。
4	TRG1MTPER	Master_TIMER周期事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Master_TIMER周期事件不生成ADC触发事件。 1：SHRTIMER Master_TIMER周期事件生成ADC触发事件。
3	TRG1MTC3	Master_TIMER比较3事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 请参考TRG1MTC0描述。
2	TRG1MTC2	Master_TIMER比较2事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 请参考TRG1MTC0描述。
1	TRG1MTC1	Master_TIMER比较1事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 请参考TRG1MTC0描述。
0	TRG1MTC0	Master_TIMER比较0事件生成SHRTIMER_ADCTRIG1上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Master_TIMER比较0事件不生成ADC触发事件。 1：SHRTIMER Master_TIMER比较0事件生成ADC触发事件。

SHRTIMER ADC 触发源 2 寄存器 (SHRTIMER_ADCTRIGS2)

地址偏移：0x44

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TRG2ST4PER	TRG2ST4C3	TRG2ST4C2	TRG2ST4C1	TRG2ST3PER	TRG2ST3C3	TRG2ST3C2	TRG2ST3C1	TRG2ST2PER	TRG2ST2C3	TRG2ST2C2	TRG2ST2C1	TRG2ST1RST	TRG2ST1PER	TRG2ST1C3	TRG2ST1C2
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRG2ST1C1	TRG2ST0RST	TRG2ST0PER	TRG2ST0C3	TRG2ST0C2	TRG2ST0C1	TRG2EXEV4	TRG2EXEV3	TRG2EXEV2	TRG2EXEV1	TRG2EXEV0	TRG2MTPER	TRG2MTC3	TRG2MTC2	TRG2MTC1	TRG2MTC0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	TRG2ST4PER	Slave_TIMER4周期事件生成SHRTIMER_ADCTRIG2上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER4周期事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER4周期事件生成ADC触发事件。
30	TRG2ST4C3	Slave_TIMER4比较3事件生成SHRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2ST4C1 描述。
29	TRG2ST4C2	Slave_TIMER4比较2事件生成SHRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2ST4C1 描述。
28	TRG2ST4C1	Slave_TIMER4比较1事件生成SHRTIMER_ADCTRIG2上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER4比较1事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER4比较1事件生成ADC触发事件。
27	TRG2ST3PER	Slave_TIMER3周期事件生成SHRTIMER_ADCTRIG2上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER3周期事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER3周期事件生成ADC触发事件。
26	TRG2ST3C3	Slave_TIMER3比较3事件生成SHRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2ST3C1 描述。
25	TRG2ST3C2	Slave_TIMER3比较2事件生成SHRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2ST3C1 描述。
24	TRG2ST3C1	Slave_TIMER3比较1事件生成SHRTIMER_ADCTRIG2上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事

		件生成ADC触发事件。 0：SHRTIMER Slave_TIMER3比较1事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER3比较1事件生成ADC触发事件。
23	TRG2ST2PER	Slave_TIMER2周期事件生成SHRTIMER_ADCTRIG2上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER2周期事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER2周期事件生成ADC触发事件。
22	TRG2ST2C3	Slave_TIMER2比较3事件生成SHRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2ST2C1 描述。
21	TRG2ST2C2	Slave_TIMER2比较2事件生成SHRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2ST2C1 描述。
20	TRG2ST2C1	Slave_TIMER2比较1事件生成SHRTIMER_ADCTRIG2上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER2比较1事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER2比较1事件生成ADC触发事件。
19	TRG2ST1RST	Slave_TIMER1复位事件生成SHRTIMER_ADCTRIG2上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER1复位事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER1复位事件生成ADC触发事件。
18	TRG2ST1PER	Slave_TIMER1周期事件生成SHRTIMER_ADCTRIG2上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER1周期事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER1周期事件生成ADC触发事件。
17	TRG2ST1C3	Slave_TIMER1比较3事件生成SHRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2ST1C1 描述。
16	TRG2ST1C2	Slave_TIMER1比较2事件生成SHRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2ST1C1 描述。
15	TRG2ST1C1	Slave_TIMER1比较1事件生成SHRTIMER_ADCTRIG2上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER1比较1事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER1比较1事件生成ADC触发事件。
14	TRG2ST0RST	Slave_TIMER0复位事件生成SHRTIMER_ADCTRIG2上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。

		0：SHRTIMER Slave_TIMER0复位事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER0复位事件生成ADC触发事件。
13	TRG2ST0PER	Slave_TIMER0周期事件生成SHRTIMER_ADCTRIG2上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER0周期事件上生成ADC触发事件。 1：SHRTIMER Slave_TIMER0周期事件上成ADC触发事件。
12	TRG2ST0C3	Slave_TIMER0比较3事件生成SHRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2ST0C1 描述。
11	TRG2ST0C2	Slave_TIMER0比较2事件生成SHRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2ST0C1 描述。
10	TRG2ST0C1	Slave_TIMER0比较1事件生成SHRTIMER_ADCTRIG2上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER0比较1事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER0比较1事件生成ADC触发事件。
9	TRG2EXEV4	外部事件4生成SHRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2EXEV0 描述。
8	TRG2EXEV3	外部事件3生成SHRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2EXEV0 描述。
7	TRG2EXEV2	外部事件2生成SHRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2EXEV0 描述。
6	TRG2EXEV1	外部事件1生成SHRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2EXEV0 描述。
5	TRG2EXEV0	外部事件0生成SHRTIMER_ADCTRIG2上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：外部事件0（EXEV0C）不生成ADC触发事件。 1：外部事件0（EXEV0C）生成ADC触发事件。
4	TRG2MTPER	Master_TIMER周期事件生成SHRTIMER_ADCTRIG2上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Master_TIMER周期事件不生成ADC触发事件。 1：SHRTIMER Master_TIMER周期事件生成ADC触发事件。
3	TRG2MTC3	Master_TIMER比较3事件生成SHRTIMER_ADCTRIG2上的ADC触发事件 请参考TRG2MTC0描述。
2	TRG2MTC2	Master_TIMER比较2事件生成SHRTIMER_ADCTRIG2上的ADC触发事件 请参考TRG2MTC0描述。

1	TRG2MTC1	Master_TIMER比较31事件生成SHRTIMER_ADCTRIG2上的ADC触发事件 请参考TRG2MTC0描述。
0	TRG2MTC0	Master_TIMER比较0事件生成SHRTIMER_ADCTRIG2上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Master_TIMER比较0事件不生成ADC触发事件。 1：SHRTIMER Master_TIMER比较0事件生成ADC触发事件。

SHRTIMER ADC 触发源 3 寄存器 (SHRTIMER_ADCTRIGS3)

地址偏移：0x48

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TRG3ST4RST	TRG3ST4C3	TRG3ST4C2	TRG3ST4C1	TRG3ST3RST	TRG3ST3PER	TRG3ST3C3	TRG3ST3C2	TRG3ST3C1	TRG3ST2RST	TRG3ST2PER	TRG3ST2C3	TRG3ST2C2	TRG3ST2C1	TRG3ST1PER	TRG3ST1C3
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRG3ST1C2	TRG3ST1C1	TRG3ST0PER	TRG3ST0C3	TRG3ST0C2	TRG3ST0C1	TRG3EXEV9	TRG3EXEV8	TRG3EXEV7	TRG3EXEV6	TRG3EXEV5	TRG3MTPER	TRG3MTC3	TRG3MTC2	TRG3MTC1	TRG3MTC0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	TRG3ST4RST	Slave_TIMER4复位事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER4复位事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER4复位事件生成ADC触发事件。
30	TRG3ST4C3	Slave_TIMER4比较3事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 请参考 TRG3ST4C1 描述。
29	TRG3ST4C2	Slave_TIMER4比较2事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 请参考 TRG3ST4C1 描述。
28	TRG3ST4C1	Slave_TIMER4比较1事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER4比较1事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER4比较1事件生成ADC触发事件。
27	TRG3ST3RST	Slave_TIMER3复位事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER3复位事件不生成ADC触发事件。

		1：SHRTIMER Slave_TIMER3复位事件生成ADC触发事件。
26	TRG3ST3PER	Slave_TIMER3周期事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER3周期事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER3周期事件生成ADC触发事件。
25	TRG3ST3C3	Slave_TIMER3比较3事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 请参考 TRG3ST3C1 描述。
24	TRG3ST3C2	Slave_TIMER3比较2事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 请参考 TRG3ST3C1 描述。
23	TRG3ST3C1	Slave_TIMER3比较1事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER3比较1事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER3比较1事件生成ADC触发事件。
22	TRG3ST2RST	Slave_TIMER2复位事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER2复位事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER2复位事件生成ADC触发事件。
21	TRG3ST2PER	Slave_TIMER2周期事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER2周期事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER2周期事件生成ADC触发事件。
20	TRG3ST2C3	Slave_TIMER2比较3事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 请参考 TRG3ST2C1 描述。
19	TRG3ST2C2	Slave_TIMER2比较2事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 请参考 TRG3ST2C1 描述。
18	TRG3ST2C1	Slave_TIMER2比较1事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER2比较1事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER2比较1事件生成ADC触发事件。
17	TRG3ST1PER	Slave_TIMER1周期事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER1周期事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER1周期事件生成ADC触发事件。

16	TRG3ST1C3	Slave_TIMER1比较3事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 请参考 TRG3ST1C1 描述。
15	TRG3ST1C2	Slave_TIMER1比较2事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 请参考 TRG3ST1C1 描述。
14	TRG3ST1C1	Slave_TIMER1比较1事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER1比较1事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER1比较1事件生成ADC触发事件。
13	TRG3ST0PER	Slave_TIMER0周期事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER0周期事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER0周期事件生成ADC触发事件。
12	TRG3ST0C3	Slave_TIMER0比较3事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 请参考 TRG3ST0C1 描述。
11	TRG3ST0C2	Slave_TIMER0比较2事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 请参考 TRG3ST0C1 描述。
10	TRG3ST0C1	Slave_TIMER0比较1事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Slave_TIMER0比较1事件不生成ADC触发事件。 1：SHRTIMER Slave_TIMER0比较1事件生成ADC触发事件。
9	TRG3EXEV9	外部事件9事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 请参考 TRG3EXEV5 描述。
8	TRG3EXEV8	外部事件8事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 请参考 TRG3EXEV5 描述。
7	TRG3EXEV7	外部事件7事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 请参考 TRG3EXEV5 描述。
6	TRG3EXEV6	外部事件6事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 请参考 TRG3EXEV5 描述。
5	TRG3EXEV5	外部事件5生成SHRTIMER_ADCTRIG3上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：外部事件5（EXEV5C）不生成ADC触发事件。 1：外部事件5（EXEV5C）生成ADC触发事件。
4	TRG3MTPER	Master_TIMER周期事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事

		件生成ADC触发事件。 0：SHRTIMER Master_TIMER周期事件不生成ADC触发事件。 1：SHRTIMER Master_TIMER周期事件生成ADC触发事件。
3	TRG3MTC3	Master_TIMER比较3事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 请参考TRG3MTC0描述。
2	TRG3MTC2	Master_TIMER比较2事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 请参考TRG3MTC0描述。
1	TRG3MTC1	Master_TIMER比较1事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 请参考TRG3MTC0描述。
0	TRG3MTC0	Master_TIMER比较0事件生成SHRTIMER_ADCTRIG3上的ADC触发事件 SHRTIMER可以在SHRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：SHRTIMER Master_TIMER比较0事件不生成ADC触发事件。 1：SHRTIMER Master_TIMER比较0事件生成ADC触发事件。

SHRTIMER DLL 校准控制寄存器 (SHRTIMER_DLLCCTL)

地址偏移：0x4C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

位/位域	名称	描述
31:4	保留	必须保持复位值
3:2	CLBPER[1:0]	DLL 校准周期 该位域定义了 DLL 校准周期的长度。 00：1048576 * $t_{SHRTIMER_CK}$ 01：131072 * $t_{SHRTIMER_CK}$ 10：16384 * $t_{SHRTIMER_CK}$ 11：2048 * $t_{SHRTIMER_CK}$
1	CLBPEREN	DLL 定期校准使能 该位用于使能定期的 DLL 校准。CLBPER [1:0]位域用于设置校准周期。 0：DLL 定期校准禁能。 1：DLL 定期校准使能。

注意：不能同时设置 CLBPEREN 位和 CLBSTRT 位。

0	CLBSTRT	DLL 校准开始 当 CLBPEREN = 0 时，向该位写入 1 将启动 DLL 校准。该位只能进行写操作。 0：无效。 1：DLL 校准开始。 注意：不能同时设置 CLBPEREN 位和 CLBSTRT 位。

SHRTIMER 故障输入配置寄存器 0 (SHRTIMER_FLTINCFG0)

地址偏移：0x50

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLT3INPROT	FLT3INFC[3:0]				FLT3INSRC	FLT3INP	FLT3INEN	FLT2INPROT	FLT2INFC[3:0]				FLT2INSRC	FLT2INP	FLT2INEN
rwo	rw				rw	rw	rw	rwo	rw				rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLT1INPROT	FLT1INFC[3:0]				FLT1INSRC	FLT1INP	FLT1INEN	FLT0INPROT	FLT0INFC[3:0]				FLT0INSRC	FLT0INP	FLT0INEN
rwo	rw				rw	rw	rw	rwo	rw				rw	rw	rw

位/位域	名称	描述
31	FLT3INPROT	保护故障 3 输入配置 请参考 FLT0INPROT 描述。
30:27	FLT3INFC[3:0]	故障 3 输入滤波配置 请参考 FLT0INFC[3:0]描述。
26	FLT3INSRC	故障 3 输入源 请参考 FLT0INSRC 描述。
25	FLT3INP	故障 3 输入极性 请参考 FLT0INP 描述。
24	FLT3INEN	故障 3 输入使能 请参考 FLT0INEN 描述。
23	FLT2INPROT	保护故障 2 输入配置 请参考 FLT0INPROT 描述。
22:19	FLT2INFC[3:0]	故障 3 输入滤波控制 请参考 FLT0INFC[3:0]描述。
18	FLT2INSRC	故障 2 输入源 请参考 FLT0INSRC 描述。
17	FLT2INP	故障 2 输入极性

请参考 FLT0INP 描述。

16 FLT2INEN 故障 2 输入使能
请参考 FLT0INEN 描述。

15 FLT1INPROT 保护故障 1 输入配置
请参考 FLT0INPROT 描述。

14:11 FLT1INFC[3:0] 故障 1 输入滤波控制
请参考 FLT0INFC[3:0]描述。

10 FLT1INSRC 故障 1 输入源
请参考 FLT0INSRC 描述。

9 FLT1INP 故障 1 输入极性
请参考 FLT0INP 描述。

8 FLT1INEN 故障 1 输入使能
请参考 FLT0INEN 描述。

7 FLT0INPROT 保护故障 0 输入配置
该位域用于配置故障 0 输入配置的写保护功能。该位写一次有效。一旦由软件置位，只能通过系统复位将其清除。
0：保护禁能。FLT0INEN，FLT0INP，FLT0INSRC 和 FLT0INFC [3:0]是可写的。
1：保护使能。FLT0INEN，FLT0INP，FLT0INSRC 和 FLT0INFC [3:0]是只读的。

6:3 FLT0INFC[3:0] 故障 0 输入滤波控制
在数字滤波器中使用事件计数器，N 次输入事件后，输出才会发生转变。该位域用于确定采样外部事件的频率（f_{SAMP}）以及应用于外部事件的数字滤波器的长度。
0000：无滤波。
0001：$f_{SAMP} = f_{SHRTIMER_CK}$, N=2.
0010：$f_{SAMP} = f_{SHRTIMER_CK}$, N=4.
0011：$f_{SAMP} = f_{SHRTIMER_CK}$, N=8.
0100：$f_{SAMP} = f_{SHRTIMER_FLTFCK}$ /2, N=6.
0101：$f_{SAMP} = f_{SHRTIMER_FLTFCK}$ /2, N=8.
0110：$f_{SAMP} = f_{SHRTIMER_FLTFCK}$ /4, N=6.
0111：$f_{SAMP} = f_{SHRTIMER_FLTFCK}$ /4, N=8.
1000：$f_{SAMP} = f_{SHRTIMER_FLTFCK}$ /8, N=6.
1001：$f_{SAMP} = f_{SHRTIMER_FLTFCK}$ /8, N=8.
1010：$f_{SAMP} = f_{SHRTIMER_FLTFCK}$ /16, N=5.
1011：$f_{SAMP} = f_{SHRTIMER_FLTFCK}$ /16, N=6.
1100：$f_{SAMP} = f_{SHRTIMER_FLTFCK}$ /16, N=8.
1101：$f_{SAMP} = f_{SHRTIMER_FLTFCK}$ /32, N=5.
1110：$f_{SAMP} = f_{SHRTIMER_FLTFCK}$ /32, N=6.
1111：$f_{SAMP} = f_{SHRTIMER_FLTFCK}$ /32, N=8.
注意：
（1）仅当 FLT0INEN 位复位时，才能对该位进行写操作。
（2）当 FLT0INPROT 位配置时，不能修改该位域。

2	FLT0INSRC	故障 0 输入源 0：故障 0 的输入源是芯片外部引脚。 1：故障 0 的输入源是芯片内部信号（如比较器）。 注意：仅当 FLT0INEN 位复位时，才能对该位进行写操作。
1	FLT0INP	故障 0 输入极性 该位用于配置故障 0 的输入极性。 0：故障 0 输入为低电平有效。 1：故障 0 输入为高电平有效。
0	FLT0INEN	故障 0 输入使能 该位置 1 时，使能故障 0 输入。 0：故障 0 输入禁能。 1：故障 0 输入使能。

SHRTIMER 故障输入配置寄存器 1 (SHRTIMER_FLTINCFG1)

地址偏移：0x54

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						FLTFDIV[2:0]		保留							
						rw									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								FLT4INPROT	FLT4INFC[3:0]				FLT4INSRC	FLT4INP	FLT4INEN
								rwo	rw				rw	rw	rw

位/位域	名称	描述
31:26	保留	必须保持复位值
25:24	FLTFDIV[2:0]	故障输入数字滤波器时钟分频 该位域可由软件配置，以确定 SHRTIMER 时钟（SHRTIMER_CK）和故障输入数字滤波器时钟（SHRTIMER_FLTFCK）之间的分频比。 $f_{SHRTIMER_FLTFCK} = f_{SHRTIMER_CK}/2^{FLTFDIV[2:0]}$。 00：$f_{SHRTIMER_FLTFCK} = f_{SHRTIMER_CK}$。 01：$f_{SHRTIMER_FLTFCK} = f_{SHRTIMER_CK}/2$。 10：$f_{SHRTIMER_FLTFCK} = f_{SHRTIMER_CK}/4$。 11：$f_{SHRTIMER_FLTFCK} = f_{SHRTIMER_CK}/8$。 注意：必须在设置 FLTyINEN（y=0..4）之前配置该位。
23:8	保留	必须保持复位值
7	FLT4INPROT	保护故障 4 输入配置 请参考 SHRTIMER_FLTINCFG0 寄存器中的 FLT0INPROT 位描述。

6:3	FLT4INFC[3:0]	故障 4 输入滤波控制 请参考 SHRTIMER_FLTINCFG0 寄存器中的 FLT0INFC[3:0]位域描述。
2	FLT4INSRC	故障 4 输入源 请参考 SHRTIMER_FLTINCFG0 寄存器中的 FLT0INSRC 位描述。
1	FLT4INP	故障 4 输入极性 请参考 SHRTIMER_FLTINCFG0 寄存器中的 FLT0INP 位描述。
0	FLT4INEN	故障 4 输入使能 请参考 SHRTIMER_FLTINCFG0 寄存器中的 FLT0INEN 位描述。

SHRTIMER DMA 更新 Master_TIMER 寄存器 (SHRTIMER_DMAUPMTR)

地址偏移：0x58
复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						MTCMP3V	MTCMP2V	MTCMP1V	MTCMP0V	MTCREP	MTCAR	MTCNT	MTDMAINTEN	MTINTC	MTCTL0
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	MTACTL	通过 DMA 模式更新 SHRTIMER_MTACTL 寄存器 请参考 MTCTL0 描述。
30:10	保留	必须保持复位值
9	MTCMP3V	通过 DMA 模式更新 SHRTIMER_MTCMP3V 寄存器 请参考MTCTL0描述。
8	MTCMP2V	通过 DMA 模式更新 SHRTIMER_MTCMP2V 寄存器 请参考 MTCTL0 描述。
7	MTCMP1V	通过 DMA 模式更新 SHRTIMER_MTCMP1V 寄存器 请参考 MTCTL0 描述。
6	MTCMP0V	通过 DMA 模式更新 SHRTIMER_MTCMP0V 寄存器 请参考 MTCTL0 描述。
5	MTCREP	通过 DMA 模式更新 SHRTIMER_MTCREP 寄存器 请参考 MTCTL0 描述。
4	MTCAR	通过 DMA 模式更新 SHRTIMER_MTCAR 寄存器 请参考 MTCTL0 描述。

3	MTCNT	通过 DMA 模式更新 SHRTIMER_MTCNT 寄存器 请参考 MTCTL0 描述。
2	MTDMAINTEN	通过 DMA 模式更新 SHRTIMER_MTDMAINTEN 寄存器 请参考 MTCTL0 描述。
1	MTINTC	通过 DMA 模式更新 SHRTIMER_MTINTC 寄存器 请参考 MTCTL0 描述。
0	MTCTL0	通过 DMA 模式更新 SHRTIMER_MTCTL0 寄存器 该位用于配置通过 DMA 模式更新 SHRTIMER_MTCTL0 寄存器。 0：DMA 模式不能更新 SHRTIMER_MTCTL0 寄存器。 1：通过 DMA 模式更新 SHRTIMER_MTCTL0 寄存器。

SHRTIMER DMA 更新 Slavex_TIMER 寄存器 (SHRTIMER_DMAUPSTxR)(x=0..4)

地址偏移：0x5C + x * 0x4,(x=0..4)

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STxACTL	保留										STxFLTCTL	STxCHOCTL	STxCSCTL	STxCNTRST	STxEXEVFCFG1
rw											rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STxEXEVFCFG0	STxCH1RST	STxCH1SET	STxCH0RST	STxCH0SET	STxDTCTL	STxCMP3V	STxCMP2V	STxCMP1V	STxCMP0V	STxCREP	STxCAR	STxCNT	STxDMAINTEN	STxINTC	STxCTL0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	STxACTL	通过 DMA 模式更新 SHRTIMER_STxACTL 寄存器 请参考 STxCTL0 描述。
30:21	保留	必须保持复位值
20	STxFLTCTL	通过 DMA 模式更新 SHRTIMER_STxFLTCTL 寄存器 请参考 STxCTL0 描述。
19	STxCHOCTL	通过 DMA 模式更新 SHRTIMER_STxCHOCTL 寄存器 请参考 STxCTL0 描述。
18	STxCSCTL	通过 DMA 模式更新 SHRTIMER_STxCSCTL 寄存器 请参考 STxCTL0 描述。
17	STxCNTRST	通过 DMA 模式更新 SHRTIMER_STxCNTRST 寄存器 请参考 STxCTL0 描述。
16	STxEXEVFCFG1	通过 DMA 模式更新 SHRTIMER_STxEXEVFCFG1 寄存器

		请参考 STxCTL0 描述。
15	STxEXEVFCFG0	通过 DMA 模式更新 SHRTIMER_STxEXEVFCFG0 寄存器 请参考 STxCTL0 描述。
14	STxCH1RST	通过 DMA 模式更新 SHRTIMER_STxCH1RST 寄存器 请参考 STxCTL0 描述。
13	STxCH1SET	通过 DMA 模式更新 SHRTIMER_STxCH1SET 寄存器 请参考 STxCTL0 描述。
12	STxCH0RST	通过 DMA 模式更新 SHRTIMER_STxCH0RST 寄存器 请参考 STxCTL0 描述。
11	STxCH0SET	通过 DMA 模式更新 SHRTIMER_STxCH0SET 寄存器 请参考 STxCTL0 描述。
10	STxDTCTL	通过 DMA 模式更新 SHRTIMER_STxCHOCTL 寄存器 请参考 STxCTL0 描述。
9	STxCMP3V	通过 DMA 模式更新 SHRTIMER_STxDTCTL 寄存器 请参考 STxCTL0 描述。
8	STxCMP2V	通过 DMA 模式更新 SHRTIMER_STxCMP2V 寄存器 请参考 STxCTL0 描述。
7	STxCMP1V	通过 DMA 模式更新 SHRTIMER_STxCMP1V 寄存器 请参考 STxCTL0 描述。
6	STxCMP0V	通过 DMA 模式更新 SHRTIMER_STxCMP0V 寄存器 请参考 STxCTL0 描述。
5	STxCREP	通过 DMA 模式更新 SHRTIMER_STxCREP 寄存器 请参考 STxCTL0 描述。
4	STxCAR	通过 DMA 模式更新 SHRTIMER_STxCAR 寄存器 请参考 STxCTL0 描述。
3	STxCNT	通过 DMA 模式更新 SHRTIMER_STxCNT 寄存器 请参考 STxCTL0 描述。
2	STxDMAINTEN	通过 DMA 模式更新 SHRTIMER_STxDMAINTEN 寄存器 请参考 STxCTL0 描述。
1	STxINTC	通过 DMA 模式更新 SHRTIMER_STxINTC 寄存器 请参考 STxCTL0 描述。
0	STxCTL0	通过 DMA 模式更新 SHRTIMER_STxCTL0 寄存器 该位用于配置 DMA 模式更新 SHRTIMER_STxCTL0 寄存器。 0：DMA 模式不能更新 SHRTIMER_STxCTL0 寄存器。 1：通过 DMA 模式更新 SHRTIMER_STxCTL0 寄存器。

SHRTIMER DMA 传输缓冲寄存器 (SHRTIMER_DMATB)

地址偏移：0x70

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMATB[31:16]															
wo															

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

位/位域	名称	描述
31:0	DMATB[31:16]	DMA 传输缓冲区 对该寄存器进行写操作时，实际访问的寄存器是 SHRTIMER_DMAUPMTR 寄存器和 SHRTIMER_DMAUPSTxR（x = 0..4）寄存器中使能的寄存器。 寄存器指针的增量由硬件计算。

20. 通用同步异步收发器（USART）

20.1. 通用同步异步收发器（USARTx, x=0..4）

20.1.1. 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，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，以及同步单双工模式。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。

所有 USART 都支持 DMA 功能，以实现高速率的数据通信。

20.1.2. 主要特性

- NRZ标准格式（Mark/Space）；
- 全双工异步通信；
- 半双工单线通信；
- 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；
  - 8或16倍过采样；
  - 当时钟频率为180M，过采样为8，最高速度可到22.5MBits/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功能的部分，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。

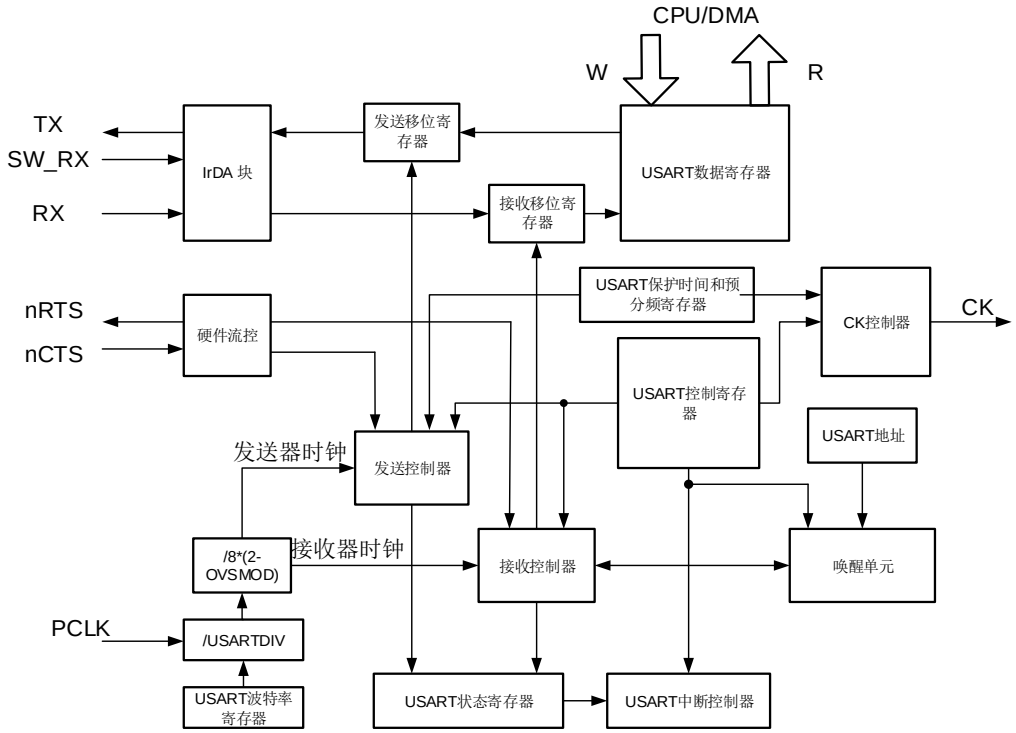
20.1.3. 功能描述

USART 接口通过*表 20-1. USART 重要引脚描述*中主要引脚从外部连接到其他设备。

表 20-1. USART 重要引脚描述

引脚	类型	描述
RX	输入	接收数据
TX	输出 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

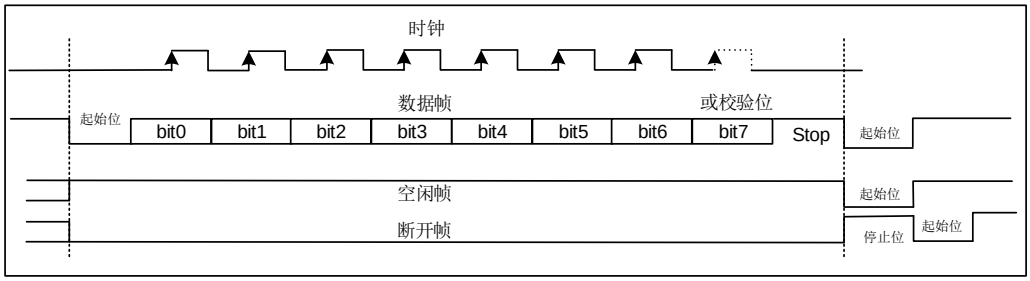
图 20-1. USART 模块内部框图



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 20-2. 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 20-2. 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 和单线模式
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 PCLK 时钟频率，波特率发生器的配置，以及过采样模式共同决定。

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。

波特率分频系数 （USARTDIV） 与系统时钟 UCLK 具有如下关系：

如果过采样率是 16，公式为：

$$USARTDIV = \frac{UCLK}{16 \times Baud\ Rate} \quad (20\text{-}1)$$

当过采样率是 8 时，公式为：

$$USARTDIV = \frac{UCLK}{8 \times Baud\ Rate} \quad (20\text{-}2)$$

例如，当过采样是 16：

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。

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。

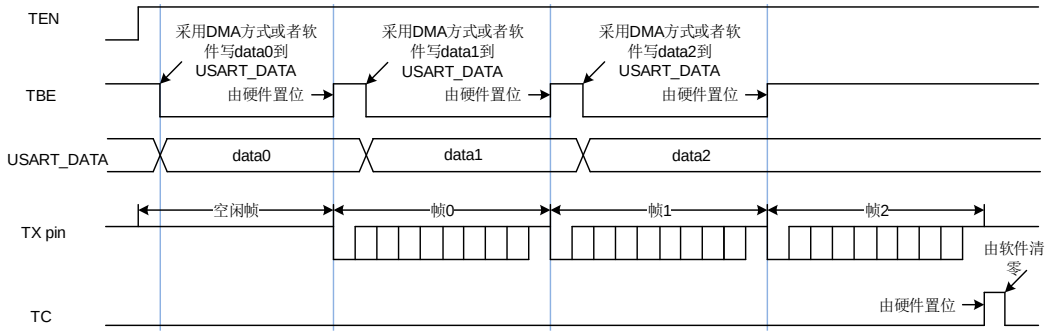
系统上电后，TBE 默认为高电平。在 USART_STAT0 寄存器中 TBE 置位时，数据可以在不覆盖前一个数据的情况下写入 USART_DATA 寄存器。当数据写入 USART_DATA 寄存器，TBE 位将被清 0。在数据由 USART_DATA 移入移位寄存器后，该位由硬件置 1。如果数据在一个发送过程正在进行时被写入 USART_DATA 寄存器，它将首先被存入发送缓冲区，在当前发送过程完成时传输到发送移位寄存器中。如果数据在写入 USART_DATA 寄存器时，没有发送过程正在进行，TBE 位将被清零然后迅速置位，原因是数据将立刻传输到发送移位寄存器。

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 20-3. 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 20-3. 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。

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。

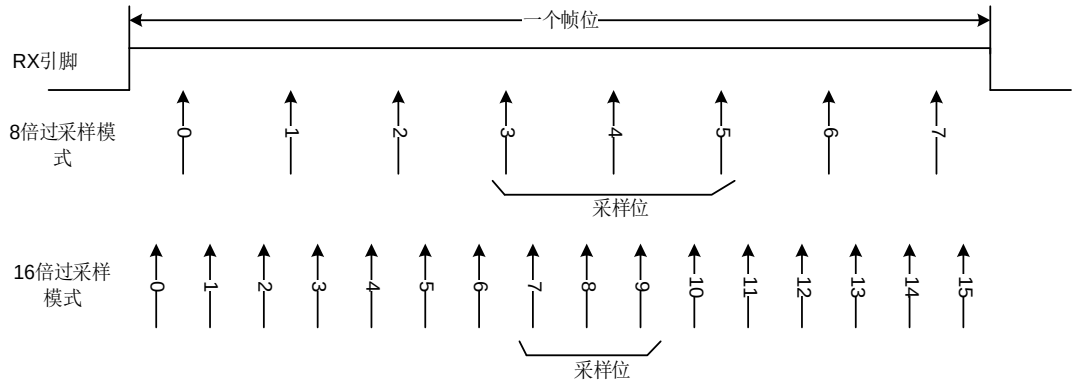
当接收到一个数据帧，USART_STAT0 寄存器中的 RBNE 置位，如果设置了 USART_CTL0 寄存器中相应的中断使能位 RBNEIE，将会产生中断。在 USART_STAT0 寄存器中可以观察接收状态标志。

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。如果是 8 倍过采样模式，选择第 3、4、5 个采样点；如果是 16 倍过采样模式，选择第 7、8、9 个采样点。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。如果在 USART_CTL2 中置位 OSB，接收器将仅获取一个采样点来估计一个数据位的值。在这种情况下将不会检测到噪声错误。

图 20-4. 过采样方式接收一个数据位（OSB=0）



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。

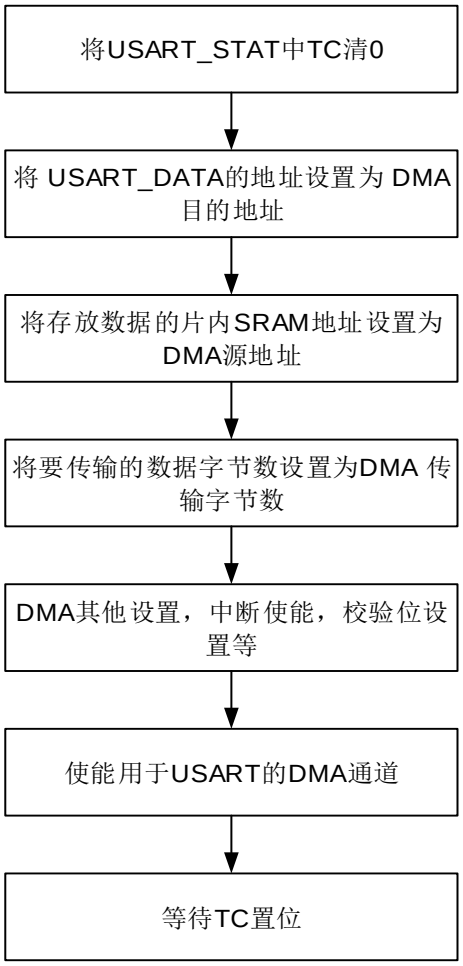
若接收过程中，产生了噪声错误（NERR）、校验错误（PERR）、帧错误（FERR）或溢出错误（ORERR），则 NERR、PERR、FERR 或 ORERR 将和 RBNE 同时置位。如果没有使能 DMA，RBNE 中断发生时，软件需检查是否有噪声错误、校验错误、帧错误或溢出错误产生。

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 20-5. 采用 DMA 方式实现 USART 数据发送配置步骤*所示。

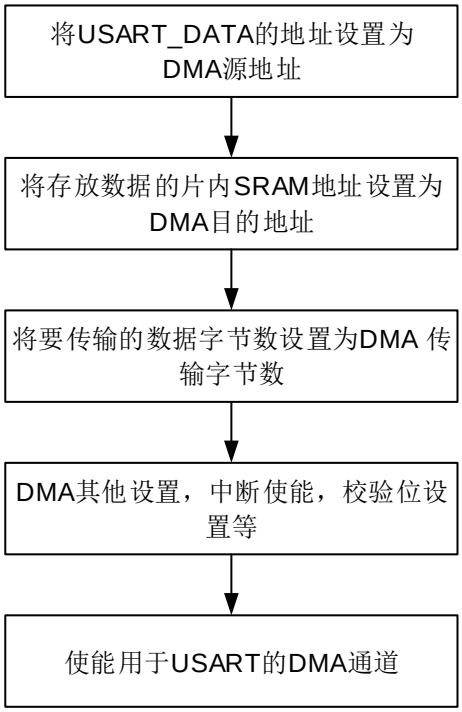
图 20-5. 采用 DMA 方式实现 USART 数据发送配置步骤



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 20-6. 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 20-6. 采用 DMA 方式实现 USART 数据接收配置步骤

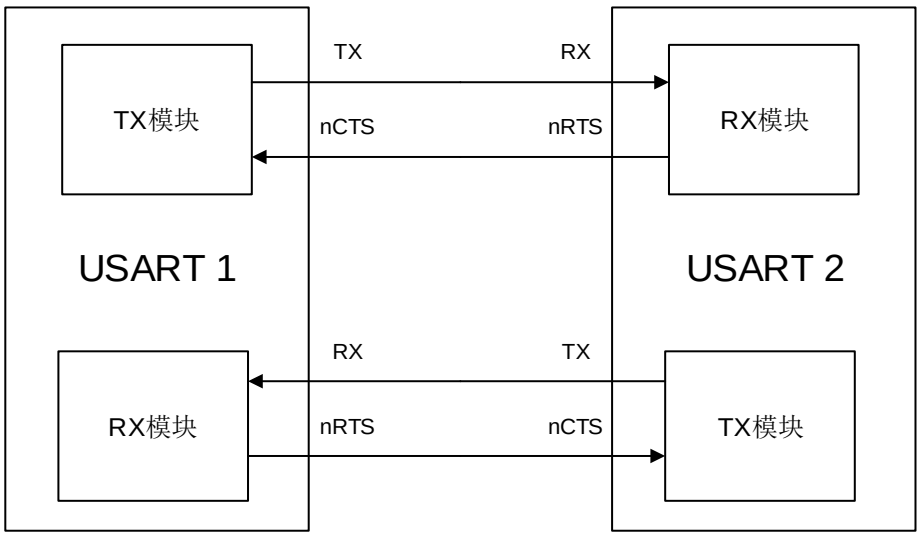


当 USART 接收到的数据数量达到了 DMA 传输数据数量，DMA 模块将产生传输完成中断。

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 20-7. 两个 USART 之间的硬件流控制



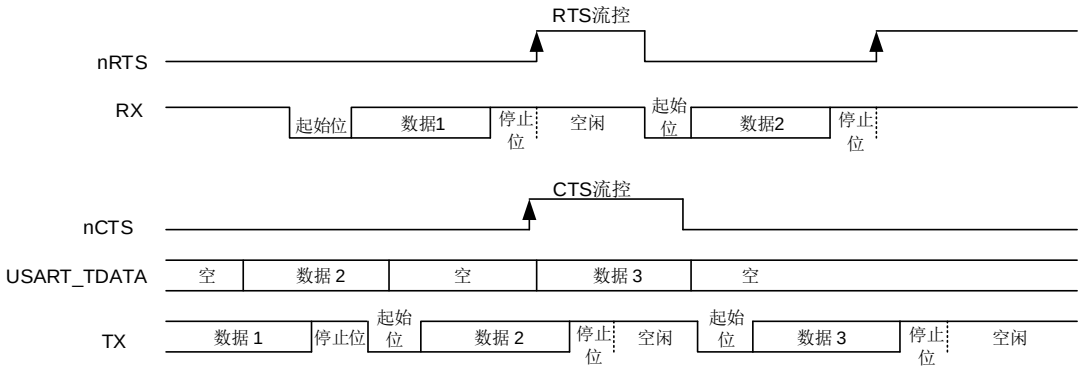
RTS 流控

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 20-8. 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。

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。

如果采用地址掩码检测，默认情况下，接收器对地址字节不做奇偶校验。如果 USART_CHC 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。

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，

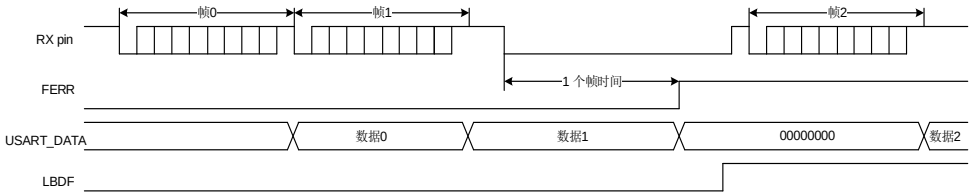
HDEN，IREN 位都应该被清 0。

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。当 USART_CTL0 寄存器中 SBKCMD 置位时，USART 在发送完一个停止位后会连续发送 13 个 0。

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。

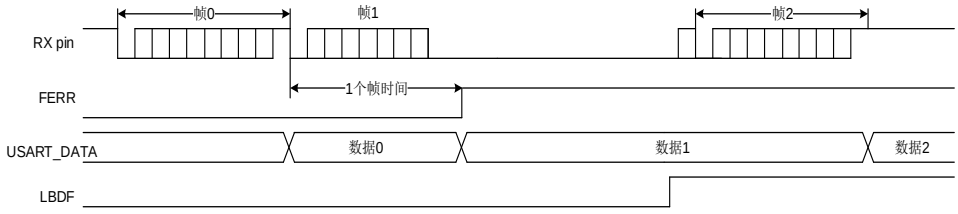
如*图 20-9. 空闲状态下检测断开帧*所示，如果断开帧发生在空闲状态下，USART 接收器会接收到一个全 0 数据帧，同时 FERR 置位。

图 20-9. 空闲状态下检测断开帧



如*图 20-10. 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 20-10. 数据传输过程中检测断开帧



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 20-11. 同步模式下的 USART 示例

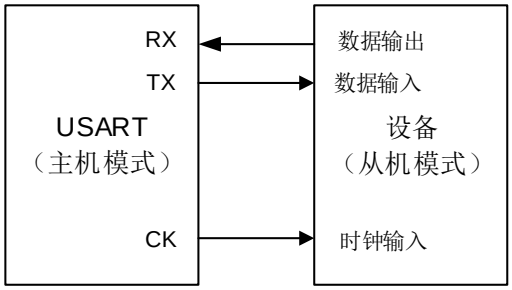
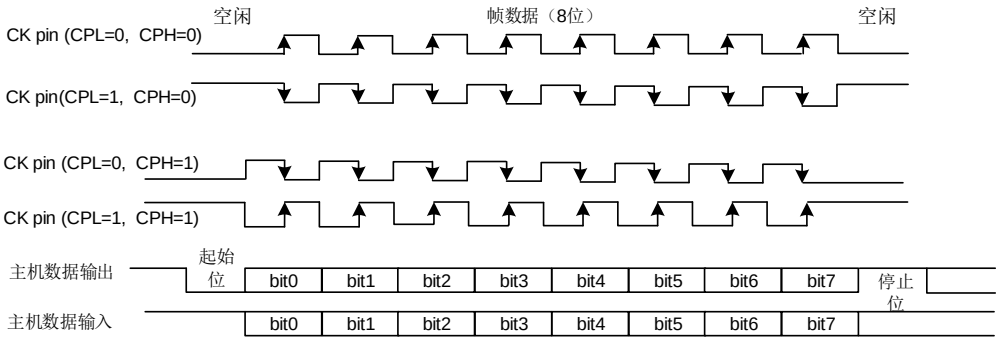


图 20-12. 8-bit 格式的 USART 同步通信波形（CLEN=1）

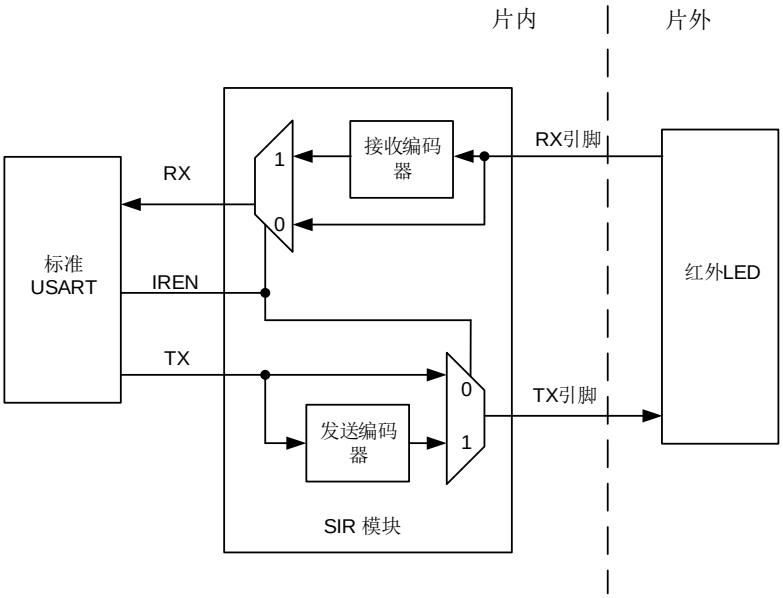


串行红外（IrDA SIR）编解码功能模块

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 发送正交译码器进行调制，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。对于正交译码器而言，波特率应小于 115200。

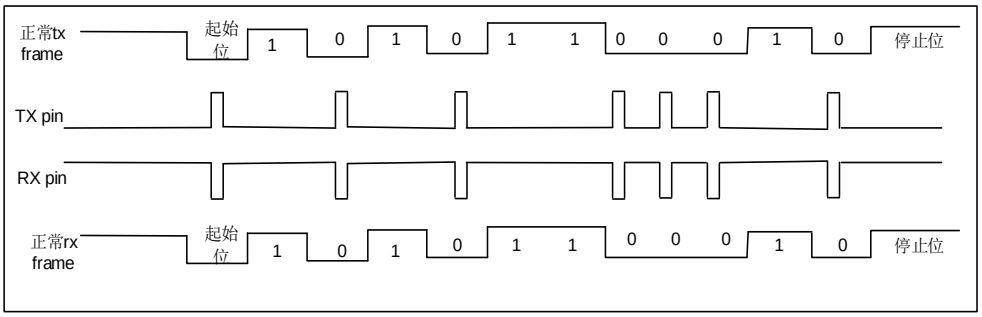
图 20-13. IrDA SIR ENDEC 模块



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 20-14. 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。

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。

在某些应用程序中可以启用冲突检测，例如多个发送设备所共享的一条数据线上的数据传输。通过将 USART_GDCTL 寄存器中的 CDEN 位置 1 使能该功能。当 txd 线上发生数据冲突时，CD 位置 1，如果 CDIE 位置 1，将会产生中断。

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 20-15. 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（默认值）。

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。

USART 中断

USART 中断事件和标志如*表 20-3. USART 中断请求*所示：

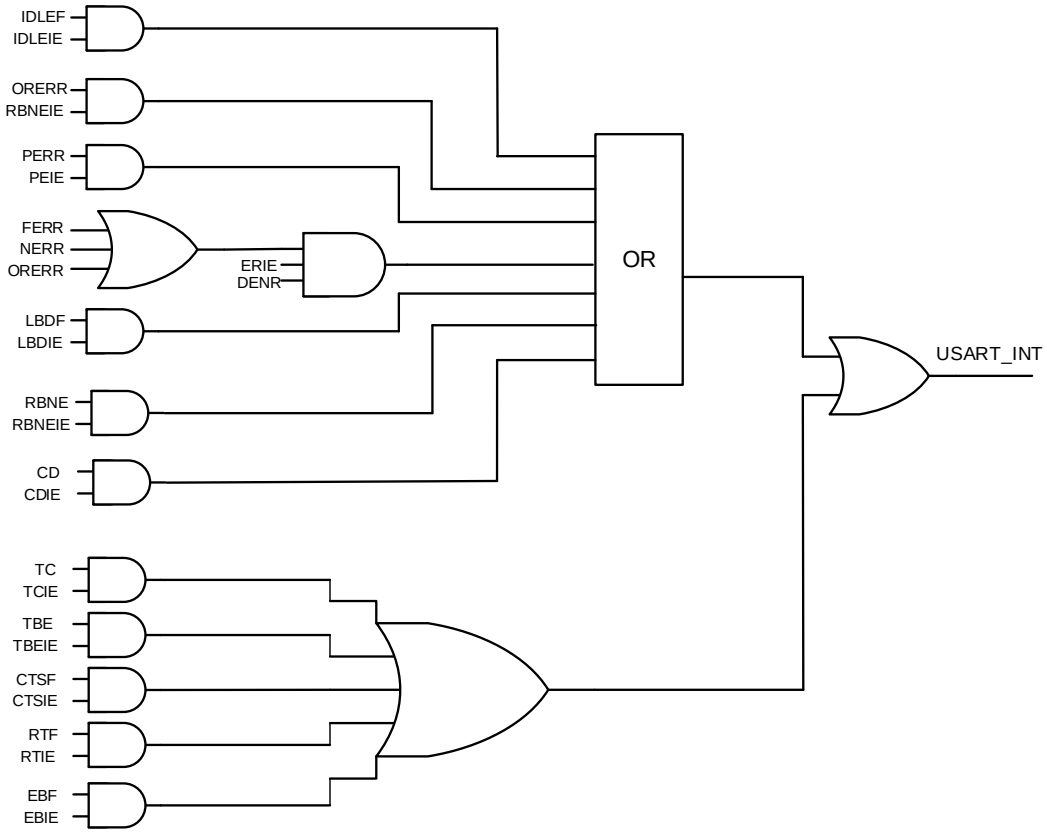
表 20-3. USART 中断请求

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		

中断事件	事件标志	控制寄存器	使能控制位
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
当 DMA 接收使能时，接收错误（噪声错误、溢出错误、帧错误）	NERR 或 ORERR 或 FERR	USART_CTL2	ERRIE
检测到冲突	CD	USART_GDCTL	CDIE

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 20-16. USART 中断映射框图



20.1.4. USART 寄存器

USART0 基地址：0x4001 3800

USART1 基地址：0x4000 4400

USART2 基地址：0x4000 4800

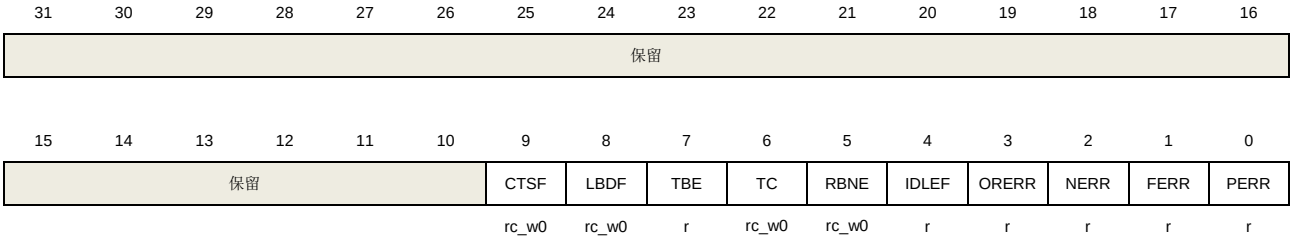
UART3 基地址：0x4000 4C00

UART4 基地址：0x4000 5000

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。



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。 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。 该位由软件清 0。 0：nCTS 状态线没有变化。 1：nCTS 状态线发生变化。 该位对 UART3/4 无效。
8	LBDF	LIN 断开检测标志。 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。 该位由软件清 0。 0：没有检测到 LIN 断开字符。 1：检测到 LIN 断开字符。
7	TBE	发送数据缓冲区空 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。 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。

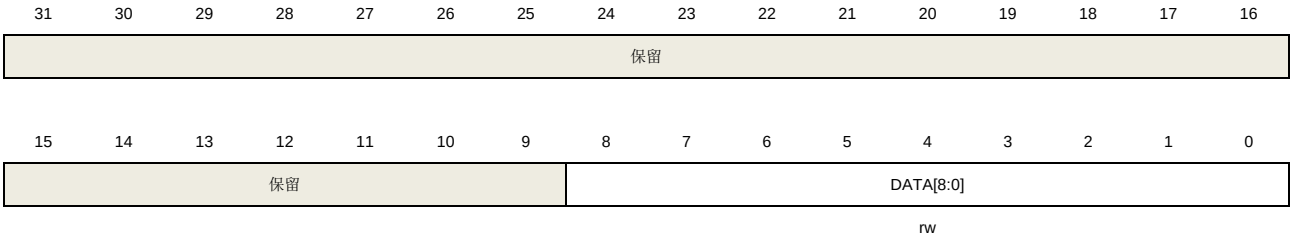
6	TC	发送完成 上电复位后，该位被置 1。如果 TBE 置位，在当前数据发送完成时该位置 1。USART_CTL0 寄存器中 TCIE 被置位将产生中断。 该位由软件清 0。 0：发送没有完成。 1：发送完成。
5	RBNE	读数据缓冲区非空。 当读数据缓冲区接收到来自移位寄存器的数据时，该位置 1。当寄存器 USART_CTL0 的 RBNEIE 位被置位，将会有中断产生。 软件可以通过对该位写 0 或读 USART_DATA 寄存器来将该位清 0。 0：读数据缓冲区为空。 1：读数据缓冲区不为空。
4	IDLEF	空闲线检测标志。 在一个帧时间内，在 RX 引脚检测到空闲状态，该位置 1。当寄存器 USART_CTL0 的 IDLEIE 位被置位，将会有中断产生。 软件先读 USART_STAT0，再读 USART_DATA 可清除该位。 0：未检测到空闲帧。 1：检测到空闲帧。
3	ORERR	溢出错误。 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_CTL2 的 ERRIE 位被置位，将会有中断产生。 软件先读 USART_STAT0，再读 USART_DATA 可清除该位。 0：没有检测到溢出错误。 1：检测到溢出错误。
2	NERR	噪声错误标志。 将 USART_CTL2 寄存器中 OSB 清 0，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。 软件先读 USART_STAT0，再读 USART_DATA 可清除该位。 0：没检测到噪声错误。 1：检测到噪声错误。
1	FERR	帧错误。 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。 软件先读 USART_STAT0，再读 USART_DATA 可清除该位。 0：未检测到帧错误。 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，再读或者写 USART_DATA 可清除该位。 0：没检测到校验错误。

1：检测到校验错误。

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。



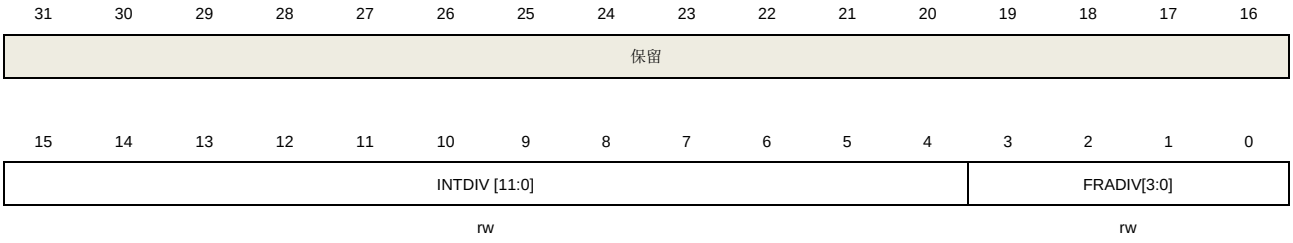
位/位域	名称	描述
31:9	保留	必须保持复位值。
8:0	DATA[8:0]	发送或接收的数据值。 软件可以通过写这些位来改变要发送的数据，或读这些位的值来获取接收到的数据。 如果使能了奇偶校验，当发送数据被写入寄存器，数据的最高位（第 7 位或第 8 位取决于 USART_CTL0 寄存器的 WL 位）将被校验位取代。

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。

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
OVSMOD	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	rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	OVSMOD	采样模式 0：16 倍采样 1：8 倍采样 如果 SCEN=1，IREN=1 或者 LMEN=1，OVSMOD 由硬件强制为 0。
14	保留	必须保持复位值。
13	UEN	USART 使能 0：USART 禁用 1：USART 使能
12	WL	字长 0：8 数据位 1：9 数据位
11	WM	从静默模式唤醒方法。 0：空闲线 1：地址掩码
10	PCEN	校验控制使能。 0：校验控制禁用。 1：校验控制使能。
9	PM	校验模式 0：偶校验 1：奇校验
8	PERRIE	校验错误中断使能。 如果该位置 1，USART_STAT0 寄存器中 PERR 被置位时产生中断。 0：校验错误中断禁用。 1：校验错误中断使能。
7	TBEIE	发送缓冲区空中断使能。

如果该位置 1，USART_STAT0 寄存器中 TBE 被置位时产生中断。
0：发送缓冲区空中断禁止。
1：发送缓冲区空中断使能。

6	TCIE	发送完成中断使能。 如果该位置 1，USART_STAT0 寄存器中 TC 被置位时产生中断。 0：发送完成中断禁用。 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。 如果该位置 1，USART_STAT0 寄存器中 RBNE 或 ORERR 被置位时产生中断。 0：读数据缓冲区非空中断和过载错误中断禁用。 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。 如果该位置 1，USART_STAT0 寄存器中 IDLEF 被置位时产生中断。 0：IDLE 线检测中断禁用。 1：IDLE 线检测中断使能。
3	TEN	发送器使能。 0：发送器禁用。 1：发送器使能。
2	REN	接收器使能。 0：接收器禁用。 1：接收器使能。
1	RWU	接收器从静默模式中唤醒。 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。 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。地址掩码模式下（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。 0：接收器处于正常工作模式。 1：接收器处于静默模式。
0	SBKCMD	发送断开帧。 软件通过发送断开帧将该位置 1。 断开帧传输结束由硬件清 0。 0：没有发送断开帧。 1：发送断开帧。

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。 0：LIN 模式禁用。 1：LIN 模式使能。
13:12	STB[1:0]	STOP 位长 00：1 停止位 01：0.5 停止位 10：2 停止位 11：1.5 停止位 对于 UART3/4，只有 1 位停止位和两位停止位是有效的。
11	CKEN	CK 引脚使能 0：CK 引脚禁用 1：CK 引脚使能 该位对于 UART3/4 无效。
10	CPL	时钟极性 该位用来设定在同步模式下 CK 引脚的极性。 0：CK 引脚不对外发送时保持为低电平。 1：CK 引脚不对外发送时保持为高电。 该位对于 UART3/4 无效。
9	CPH	时钟相位 该位用来设定在同步模式下 CK 引脚的相位。 0：在首个时钟边沿采样第一个数据 1：在第二个时钟边沿采样第一个数据 该位对于 UART3/4 无效。
8	CLEN	CK 信号长度。 该位用来设定在同步模式下 CK 信号的长度。 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。 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。 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。 如果该位置 1，当 USART_STAT0 寄存器中 LBDF 被置位时将产生中断。 0：断开信号检测中断禁用。

1：断开信号检测中断使能。

5	LBLEN	LIN 断开帧长度 该位用来设定在断开帧长度。 0：10 位 1：11 位
4	保留	必须保持复位值。
3:0	ADDR[3:0]	USART 地址 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。

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				OSB	CTSIE	CTSEN	RTSEN	DENT	DENR	SCEN	NKEN	HDEN	IRLP	IREN	ERRIE
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:12	保留	必须保持复位值。
11	OSB	一个数据位采样一次的方法。 该位用于选择采样方法。当该位被置位时，USART 对一个数据位取一个采样点，而不是一个数据位取三个采样点。当使用该方法时，噪声错误标志（NERR）需禁用。 0：三次采样方法。 1：一次采样方法。
10	CTSIE	CTS 中断使能。 如果该位置 1，当 USART_STAT0 寄存器中 CTSF 被置位时将产生中断。 0：CTS 中断禁用。 1：CTS 中断使能。 该位对于 UART3/4 无效。
9	CTSEN	CTS 使能 该位用于使能 CTS 硬件流控制功能。 0：CTS 硬件流控制禁用。 1：CTS 硬件流控制使能。 该位对于 UART3/4 无效。

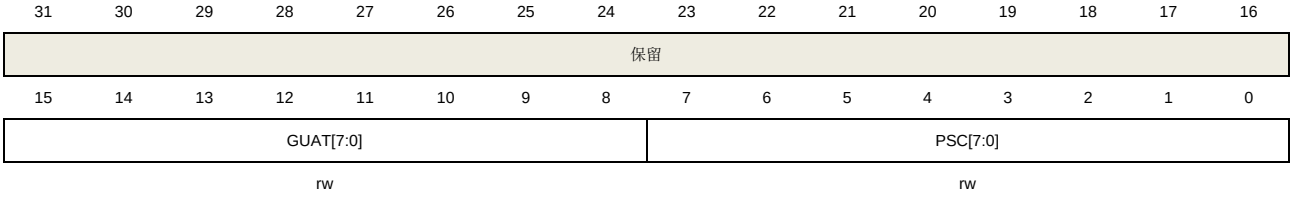
8	RTSEN	RTS 使能 该位用于使能 RTS 硬件流控制功能。 0：RTS 硬件流控制禁用。 1：RTS 硬件流控制使能。 该位对于 UART3/4 无效。
7	DENT	DMA 发送使能 0：DMA 发送模式禁用。 1：DMA 发送模式使能。
6	DENR	DMA 接收使能 0：DMA 接收模式禁用。 1：DMA 接收模式使能。
5	SCEN	智能卡模式使能 该位用于使能智能卡模式。 0：智能卡模式禁用。 1：智能卡模式使能。 该位对于 UART3/4 无效。
4	NKEN	在智能卡模式 NACK 使能。 该位用于智能卡模式在奇偶校验错误发生时使能 NACK 发送。 0：当出现校验错误时不发送 NACK。 1：当出现校验错误时发送 NACK。 该位对于 UART3/4 无效。
3	HDEN	半双工使能 该位用于使能半双工模式。 0：半双工模式禁用 1：半双工模式使能
2	IRLP	IrDA 低功耗模式。 该位用于为 IrDA 模式选择低功耗模式。 0：正常模式 1：低功耗模式
1	IREN	IrDA 模式使能 0：IrDA 禁用 1：IrDA 使能
0	ERRIE	错误中断使能。 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。 0：错误中断禁用。 1：错误中断使能。

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。



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。 TC 标志置位时间延时 GUAT[7:0]个波特时钟周期。 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。 00000000：保留 – 不要写入该值。 00000001：对源时钟 1 分频。 … 11111111：对源时钟 255 分频。 在 IrDA 正常模式下，PSC 只能设置成 00000001。 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。 00000：保留 – 不要写入该值。 00001：对源时钟 2 分频。 00010：对源时钟 4 分频。 … 11111：对源时钟 62 分频。 在智能卡模式下，PSC[7:5]保留。

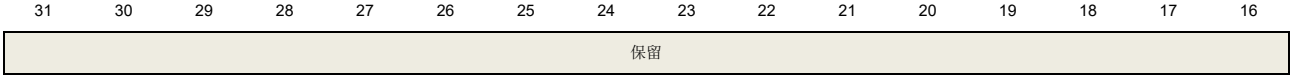
控制寄存器 3 （USART_CTL3）

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



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 该位用于设定数据在发送或接收时的顺序。 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。 USART 被使能（UEN=1）时，这一位不能被改写。
10	DINV	数据位反转 该位用于设定在发送或接收时数据位的极性。 0：数据位信号值没有反转。 1：数据位信号值被反转。 USART 被使能（UEN=1）时，这一位不能被改写。
9	TINV	TX 引脚电平反转。 该位用于设定 TX 引脚极性。 0：TX 引脚信号值没有反转。 1：TX 引脚信号值被反转。 USART 被使能（UEN=1）时，这一位不能被改写。
8	RINV	RX 引脚电平反转。 该位用于设定 RX 引脚极性。 0：RX 引脚信号值没有反转。 1：RX 引脚信号值被反转。 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 如果该位置 1，USART_STAT1 寄存器中 EBF 被置位时产生中断。 0：块中断禁用。 1：块中断使能。
4	RTIE	接收超时标志中断使能位。 如果该位置 1，USART_STAT1 寄存器中 RTF 被置位时产生中断。 0：接收超时中断禁用。 1：接收超时中断使能。
3:1	SCRTNUM[2:0]	智能卡自动重试次数寄存器。 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。 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。 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。
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。 该位用于使能 USART 接收超时。 0：接收器超时检测功能禁用。 1：接收器超时检测功能使能。

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。

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

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

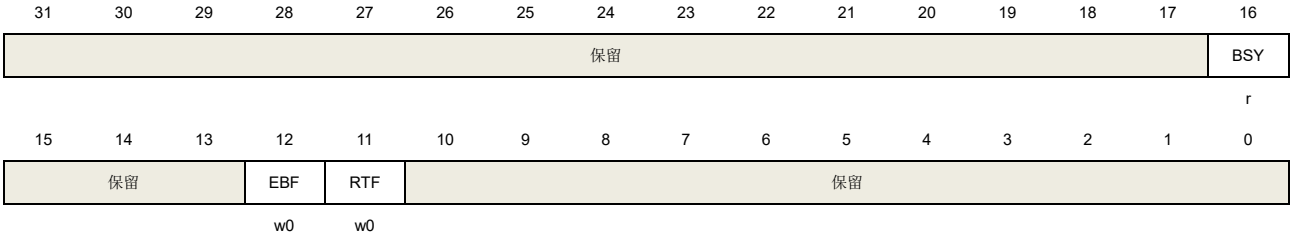
位/位域	名称	描述
31:24	BL[7:0]	块长度 这些位用于设定智能卡 T=1 的接收时，块的长度。它的值等于信息字节的长度+结束部分的长度（1-LEC/2-CRC） – 1。 这个值可以在块接收开始去设置（用于需要从块的序言提取块的长度的情形），这个值在每一个接收时钟周期只能设置一次。在智能卡模式下，当 TBE=0 时，块的长度计数器被清 0。 在其他模式下，当 REN=0 （禁用接收器）或者当 USART_STAT1 寄存器的 EBF 位被写 0 时，块的长度计数器被清 0。
23:0	RT[23:0]	接收器超时阈值。 该位域用于指定接收超时值，单位是波特时钟的时长。 标准模式下，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。 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。 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。 **注意：**当 16 倍过采样时，RT[23:0]不能被配置为 0xFFFFFF。

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

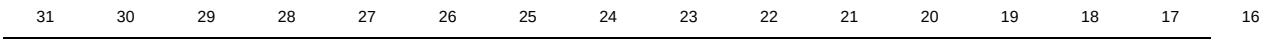


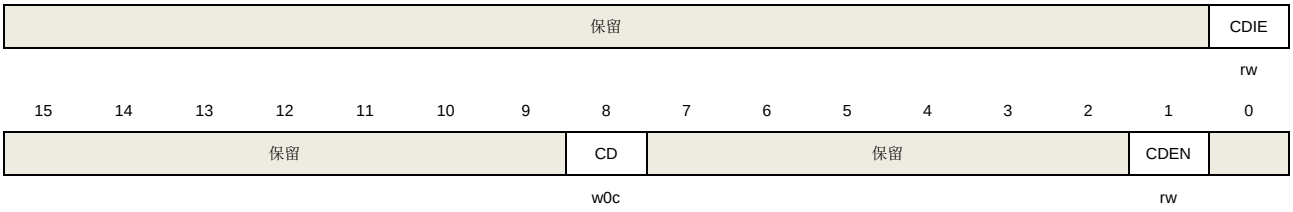
位/位域	名称	描述
31:17	保留	必须保持复位值。
16	BSY	忙标志 USART 接收一帧数据时被置位。 0：USART 接收通道空闲。 1：USART 接收通道忙。
15:13	保留	必须保持复位值。
12	EBF	块结束标志。 该位在接收字节数（从块开始开始计数，包含序言）等于或者大于 BLEN+4 时被置位。USART_CTL3 寄存器中 EBIE 被置位将产生中断。 软件可以通过写 0 清除该位。 0：块结束事件没有发生。 1：块结束事件发生。
11	RTF	接收超时标志。 该位在 RX 引脚空闲时间已经超过 RT 值时被置位。USART_CTL3 寄存器中 RTIE 被置位将产生中断。 软件可以通过写 0 清除该位。 0：接收器超时事件没有发生。 1：接收器超时事件发生。
10:0	保留	必须保持复位值。

GD 控制寄存器 （USART_GDCTL）

偏移地址：0xD0
复位值：0x0000 0000

该寄存器只能按字（32 位）访问





位/位域	名称	描述
31:17	保留	由硬件强制为 0。
16	CDIE	冲突检测中断使能。 0: 冲突检测中断禁用。 1: 冲突检测中断使能。
15:9	保留	由硬件强制为 0。
8	CD	冲突检测状态。 0: 未检测到冲突。 1: 半双工模式下检测到冲突。
7:2	保留	由硬件强制为 0。
1	CDEN	冲突检测使能。 0: 禁用 1: 使能
0	保留	由硬件强制为 0。

20.2. 通用同步异步收发器（USARTx, x=5）

20.2.1. 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（PCLK2、CK_USART5）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，智能卡模式，LIN，以及同步和单双工模式。它还支持多处理器通信。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。

所有 USART 都支持 DMA 功能，以实现高速率的数据通信。

20.2.2. 主要特性

- NRZ标准格式
- 全双工异步通信
- 半双工单线通信
- 接收FIFO功能
- 双时钟域：
  - 互为异步关系的APB时钟和USART时钟
  - 不依赖PCLK设置的波特率设置
- 可编程的波特率产生器，当时钟频率为180MHz，过采样为8，最高速度可达22.5MBits/s
- 完全可编程的串口特性：
  - 数据位（8或9位）低位或高位在前
  - 偶校验位，奇校验位，无校验位的生成或检测
  - 产生0.5，1，1.5或者2个停止位
- 可互换的Tx/Rx引脚
- 可配置的数据极性
- 可配置的多级缓存通信DMA访问数据缓冲区
- 发送器和接收器可分别使能
- 奇偶校验位控制：
  - 发送奇偶校验位
  - 检测接收的数据字节的奇偶校验位
- 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

■ 支持ModBus通信：
- 超时功能
- CR/LF字符识别

■ 从深度睡眠模式，深度睡眠模式1和深度睡眠模式2唤醒：
- 通过标准的RBNE中断
- 通过WUF中断

■ 多种状态标志：
- 传输检测标志：接收缓冲区不为空（RBNE），接收FIFO满（RFF），发送缓冲区为空（TBE），传输完成（TC）
- 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）
- 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）
- ModBus通信标志：地址/字符匹配（AMF），接收超时（RTF）
- 智能卡模式标志：块结束（EBF）和接收超时（RTF）
- 从深度睡眠模式唤醒标志
- 若相应的中断使能，这些事件发生将会触发中断

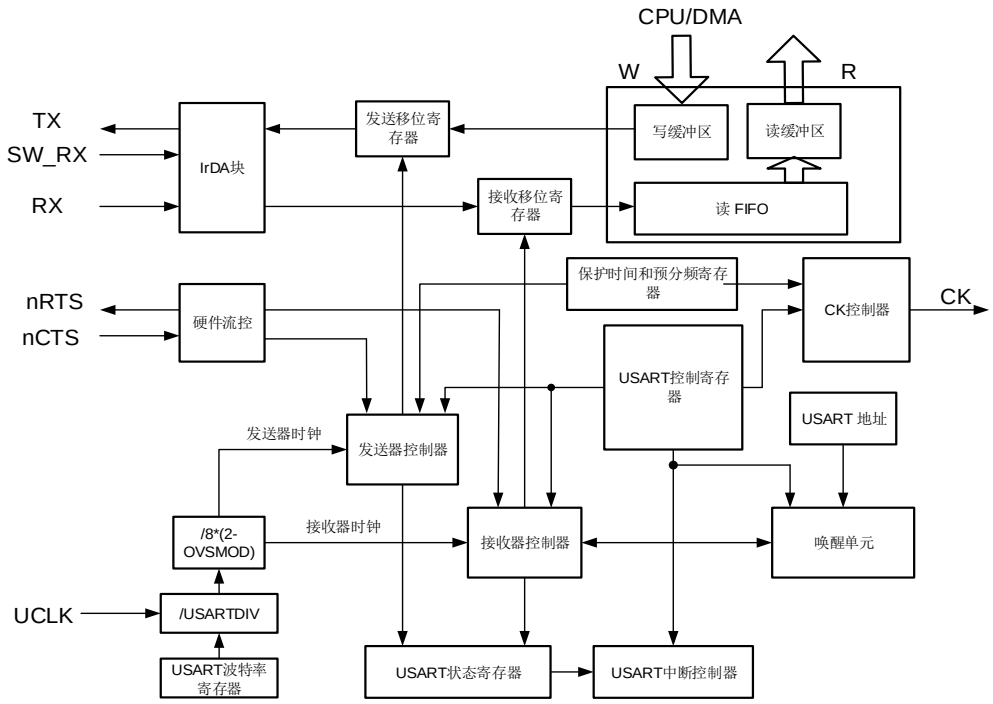
20.2.3. 功能描述

USART 接口通过*表 20-4. USART 重要引脚描述*中主要引脚从外部连接到其他设备。

表 20-4. USART 重要引脚描述

引脚	类型	描述
RX	输入	接收数据
TX	输出 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

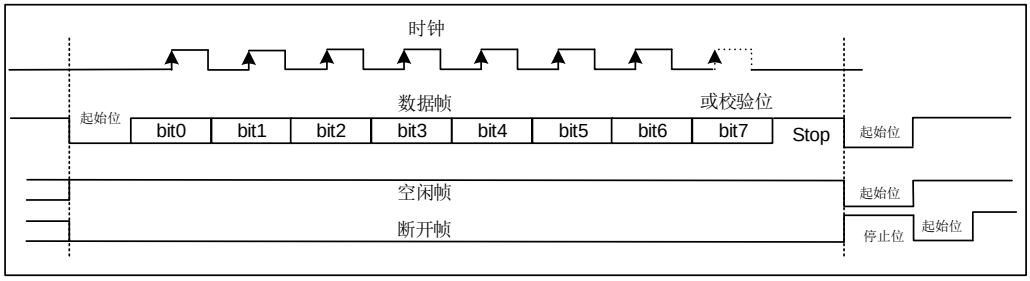
图 20-17. USART 模块内部框图



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 20-18. USART 字符帧（8 数据位和 1 停止位）



在发送和接收中，停止位可以在 USART_CTL1 寄存器中 STB[1:0]位域中配置。

表 20-5. 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 和单线模式
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。

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。

波特率分频系数（USARTDIV）与 UCLK 具有如下关系：

如果过采样率是 16，公式为：

$$USARTDIV = \frac{UCLK}{16 \times Baud\ Rate} \quad (20\text{-}3)$$

如果过采样是 8，公式为：

$$USARTDIV = \frac{UCLK}{8 \times Baud\ Rate} \quad (20\text{-}4)$$

例如，当过采样是 16：

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。

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_CTL1 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_STAT 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_TDATA 寄存器。当数据写入 USART_TDATA 寄存器，TBE 位将被清 0。在数据由 USART_TDATA 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_TDATA 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_TDATA 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，原因是数据被立刻传输到发送移位寄

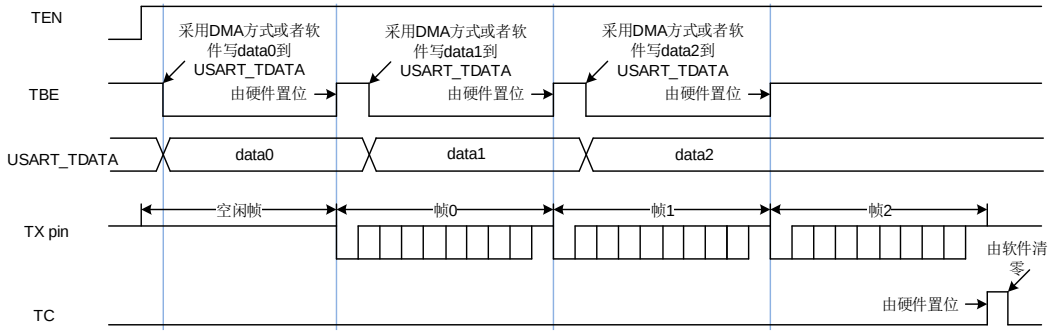
存器。

假如一帧数据已经被发送出去，并且 TBE 位已被置位，那么 USART_STAT 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 20-19. 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. 通过USART_CTL0寄存器的WL设置字长；
2. 在USART_CTL1寄存器中写STB[1:0]位来设置停止位的长度；
3. 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）；
4. 在USART_BAUD寄存器中设置波特率；
5. 在USART_CTL0寄存器中置位UEN位，使能USART；
6. 在USART_CTL0寄存器中设置TEN位；
7. 等待TBE置位；
8. 向USART_TDATA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 20-19. 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。通过向 USART_INTC 寄存器的 TCC 位写 1 可将 TC 位清 0。

当 SBKCMD 置位时，会发送一个断开帧，发送完成后，SBKCMD 将被清 0。

USART 接收器

上电后，按以下步骤使能 USART 接收器：

1. 写USART_CTL0寄存器的WL位去设置字长；
2. 在USART_CTL1寄存器中写STB[1:0]位来设置停止位的长度；
3. 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）
4. 在USART_BAUD寄存器中设置波特率；
5. 在USART_CTL0寄存器中置位UEN位，使能USART；
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_STAT 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_STAT 寄存器中可以观察接收

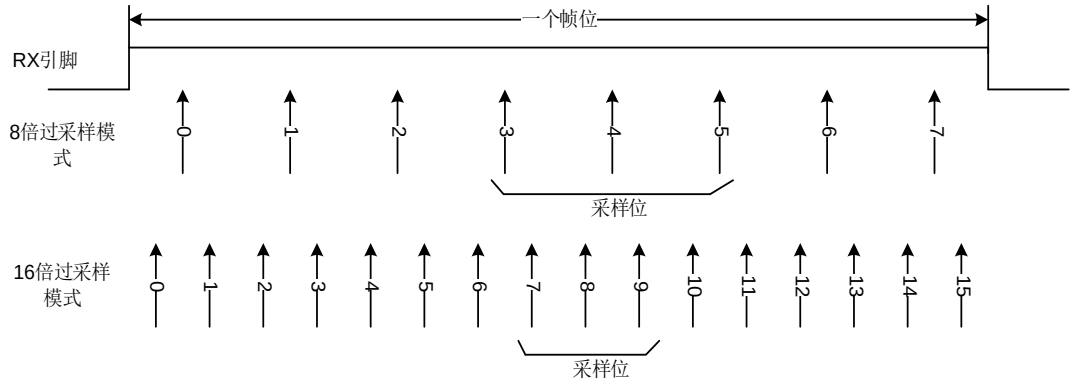
状态标志。

软件可以通过读 USART_RDATA 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_RDATA 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。如果是 8 倍过采样模式，选择第 3、4、5 个采样点；如果是 16 倍过采样模式，选择第 7、8、9 个采样点。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。如果在 USART_CTL2 中置位 OSB，接收器将仅获取一个采样点来估计一个数据位的值。在这种情况下将不会检测到噪声错误。

图 20-20. 过采样方式接收一个数据位（OSB=0）



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_STAT 寄存器中 PERR 被置位。如果置位了 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_STAT 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_STAT 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。

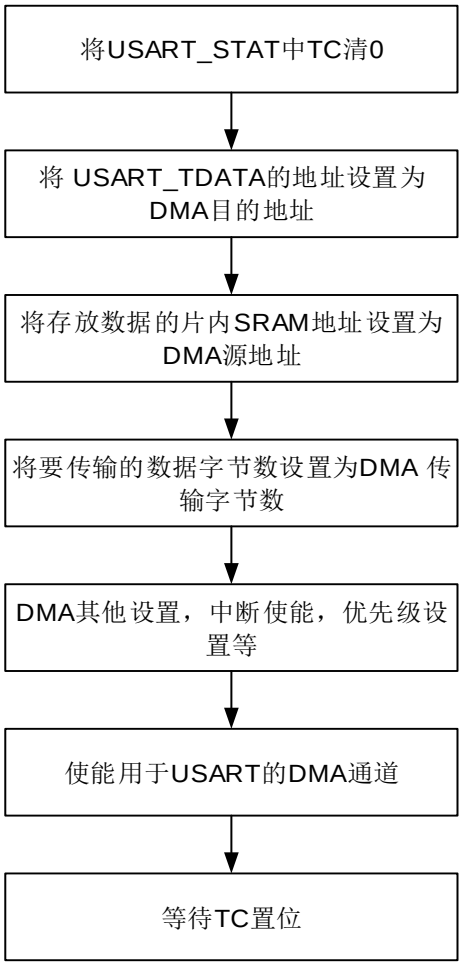
DMA 方式访问数据缓冲区

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 20-21. 采用 DMA 方式实现 USART 数据发送配置步骤*所示。

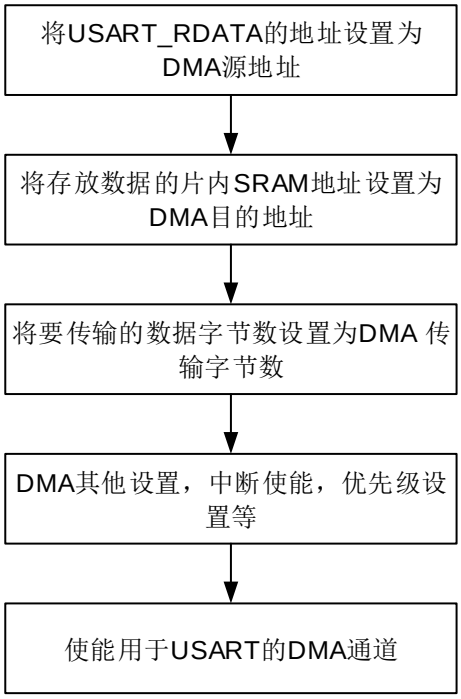
图 20-21. 采用 DMA 方式实现 USART 数据发送配置步骤



所有数据帧都传输完成后，USART_STAT 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 20-22. 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_STAT 寄存器中的错误标志位（FERR、ORERR 和 NERR）置位时将产生中断。

图 20-22. 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。

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_CMD 寄存器中 MMCMD 位置 1 使 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。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标记检测。

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_STAT 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。如果地址标志位是 0，该字节被认为是数据字节。如果地址字节的低 4 位或低 7 位与 USART_CTL1 寄存器中的 ADDR 位相同，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_STAT 寄存器中获取。如果地址字节的低 4 位或低 7 位与 USART_CTL1 寄存器中的 ADDR 位不相同，硬件会置位 RWU 并自动进入静默模式。在这种情况下，RBNE 不会被置位。

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。如果 ADDM 位被置位，且接收帧为 7 位的数据，其中最低的 6 位将与 ADDR[5:0]比较。如果 ADDM 位被置位，且接收帧为 9 位的数据，其中低 8 位将与 ADDR[7:0]进行比较。

LIN 模式

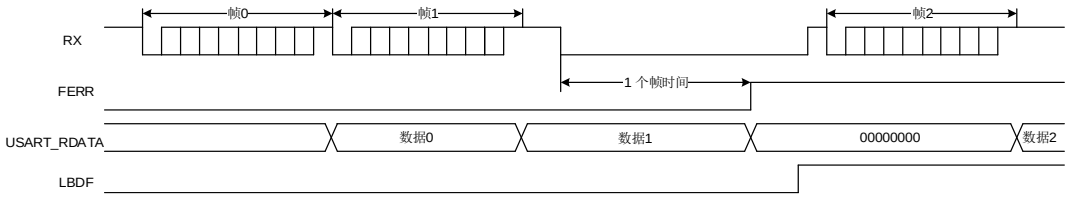
将 USART_CTL1 寄存器的 LMEN 置位即可使能本地互联网络模式。在 LIN 模式下，USART_CTL1 寄存器中 CKEN，STB[1:0]和 USART_CTL2 的 SCEN，HDEN，IREN 位都应该被清 0。

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。数据位的长度只能为 8。一个停止位后连续 13 个 0 为断开帧。

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 位可以选择断开帧的长度。如果在 RX 引脚检测到大于或等于与预期的断开帧长度的 0（LBLEN=0 时，10 个 0；LBLEN=1 时，11 个 0），USART_STAT 寄存器中 LBDF 置位。如果 USART_CTL1 寄存器中 LBDIE 被置位，将产生中断。

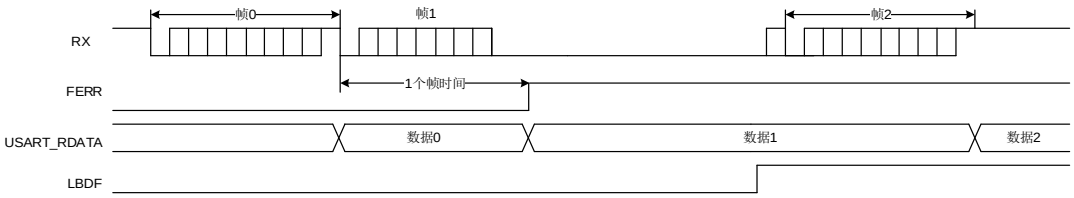
如*图 20-23. 空闲状态下检测断开帧*所示，如果断开帧发生在空闲状态下，USART 接收器会接收到一个全 0 数据帧，同时 FERR 置位。

图 20-23. 空闲状态下检测断开帧



如*图 20-24. 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 20-24. 数据传输过程中检测断开帧



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 位应被清 0。CK 引脚作为 USART 同步发送器的时钟输出，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。在空闲状态和断开帧的发送过程中，也不会有时钟信号产生。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。

时钟与已发送的数据同步。同步模式下的接收器按照发送器的时钟进行采样，并无任何过采样。

图 20-25. 同步模式下的 USART 示例

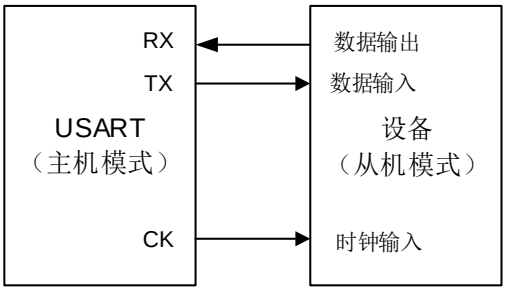
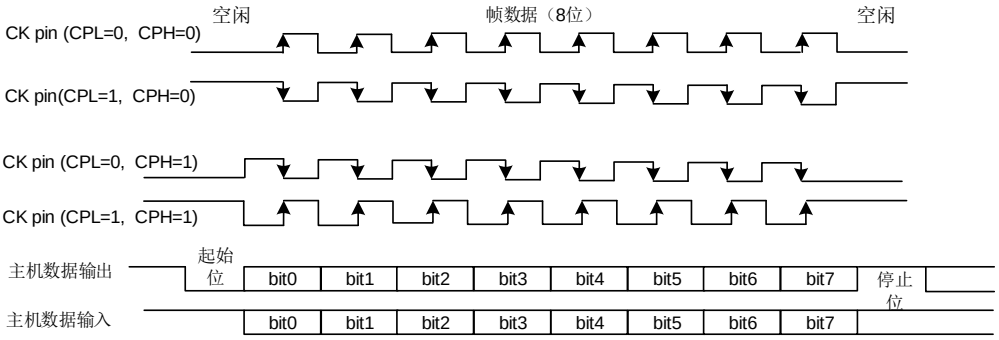


图 20-26. 8-bit 格式的 USART 同步通信波形（CLEN=1）

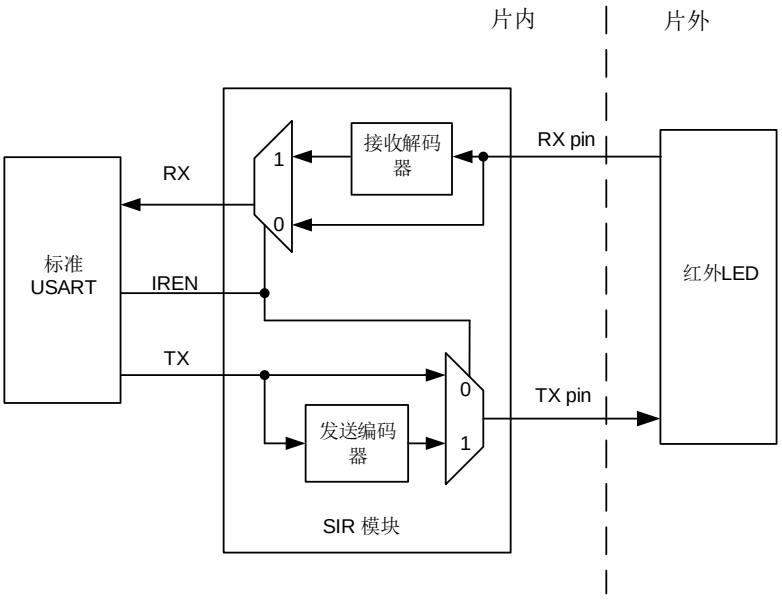


串行红外（IrDA SIR）编解码功能模块

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 位应被清 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。

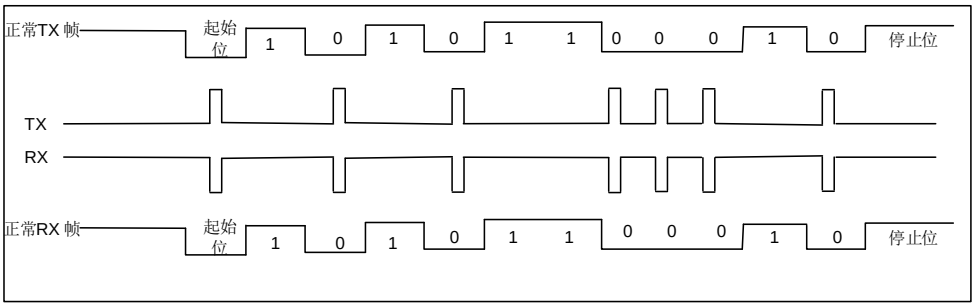
图 20-27. IrDA SIR ENDEC 模块



在 IrDA 模式下，TX 引脚与 RX 引脚电平不同。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 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 则无法可靠地检测到。

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 20-28. 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。接收解码器工作模式与正常 IrDA 模式相同。

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 位应被清零。

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。当 TEN 被置位时，在数据寄存器中的数据将会被发送。

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 将向智能卡提供一个时钟。该时钟可以分频用于其他用途。

智能卡模式下的帧格式为：1 起始位+9 数据位（包括 1 个奇偶校验位）+1.5 停止位。

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 引脚须被设置成开漏模式，这个引脚将会与智能卡驱动同一条双向连线。

图 20-29. 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 寄存器的 RT[23:0]位域设置成 BWT（块等待时间）-11 的值，并将 RBNEIE 置位。如果到了这个时间，还没有从智能卡收到应答，将引起超时中断。如果在超时之前收到了第一个字节，则会引起 RBNE 中断。块模式下，如果用 DMA 从智能卡读取数据，也只能在第一个字节接收完后再去使能 DMA。

在接收到第一个字节之后（RBNE 中断）必须将 USART_RT 寄存器设置为 CWT（字节等待时间）-11 之间的某个值（这个时间以波特时间作为单位），这是为了自动检测两个连续字符之间的最大等待时间。如果智能卡在前一个字符发送结束后到设定的 CWT 周期之间没有发送字符，USART 会通过 RTF 标志提醒软件，当 RTIE 被置位时，会引起中断。

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 寄存器的 BL[7:0]。当使用 DMA 模式时，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。

在中断驱动接收模式，块的长度可以由软件提取出来并做检测或者通过设置 BL 的值得到。但是在块开始之前，BL（0xFF）可以被设置为最大值。实际值则要在接收到第三个字节后写到寄存器中。

整个块的长度（包括序言区，收尾区和信息区）等于 BL+4。块尾通过 EBF 标志和相应中断提醒给软件（当 EBIE 位置 1 时）。如果块长度出错，将会引起一个 RT 中断。

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，MSBF 位和 DINV 位都应设置为 0（默认值）。

如果选择反向转换，从数据帧的最高位开始传输，TX 引脚低电平代表逻辑 ‘1’，偶校验。在这种情况下，MSBF 位和 DINV 位都应设置为 1。

ModBus 通信

通过实现块尾检测功能，USART 提供实现 ModBus/RTU 和 ModBus/ASCII 协议的基本支持。

在 ModBus/RTU 模式下，通过一个超过 2 个字符长度的空闲状态来识别块尾。这个功能是通过一个可编程的超时检测功能来实现的。

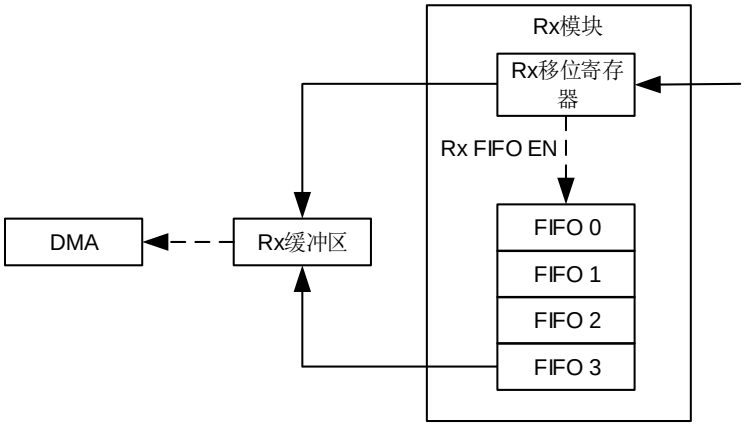
为了检测空闲状态，必须置位 USART_CTL1 寄存器的 RTEN 位和 USART_CTL0 寄存器的 RTIE 位。USART_RT 寄存器必须被设置成与 2 个字节超时所对应的值。在最后一个停止位被接收后，当接收线在这期间是空闲的，将产生一个中断，通知软件当前块接收已经完成。

在 ModBus/ASCII 模式下，块尾被认为是一个特定的字符（CR/LF）串。USART 用字符匹配机制实现这个功能。具体是通过将 LF 的 ASCII 码配置到 ADDR 区域并激活地址匹配中断（AMIE=1）来实现。软件将在收到 LF 或可以在 DMA 缓存中查找到 CR/LF 时得到提示。

接收 FIFO

通过将 USART_RFCS 寄存器的 RFEN 置位使能接收 FIFO，可以避免当 CPU 无法迅速响应 RBNE 中断时，发生过载错误。接收 FIFO 和接收缓存区可储存多至 5 帧的数据。若接收 FIFO 满，RFFINT 位将被置位。如果 RFFIE 被置位，将产生中断。

图 20-30. USART 接收 FIFO 结构



如果软件在响应 RBNE 中断时读数据接收缓冲区，在响应开始时，RBNEIE 位应清 0。当所有接收的数据被读出后，RBNEIE 位应置位。在读出接收的数据前，PERR，NERR，FERR，EBF 都应被清 0。

从 Deepsleep 模式唤醒

通过标准 RBNE 中断或 WUM 中断 USART 能从深度睡眠模式，深度睡眠模式 1 和深度睡眠模式 2 唤醒 MCU。

UESM 位必须置 1 并且 USART 时钟必须设置为 IRC8M 或 LXTAL （请参考 RCU 部分）。

当使用 RBNE 标准中断时，必须在进入深度睡眠模式，深度睡眠模式 1 和深度睡眠模式 2 前将 RBNEIE 位置位。

当使用 WUIE 中断时，WUIE 中断源可以通过 WUM 位来选择。

在进入深度睡眠模式，深度睡眠模式 1 和深度睡眠模式 2 前，必须禁用 DMA。在进入深度睡眠模式，深度睡眠模式 1 和深度睡眠模式 2 前，软件必须检测 USART 是否正在传送数据。这可以通过 USART_STAT 寄存器中的 BSY 标志来判断。REA 位必须被检测以确保 USART 是使能的。

当检测到唤醒事件时，无论 MCU 工作在深度睡眠模式还是正常模式，WUF 标志位通过硬件被置 1，并且在 WUIE 被置位的情况下，触发一个唤醒中断。

USART 中断

USART 中断事件和标志如*表 20-6. USART 中断请求*所示：

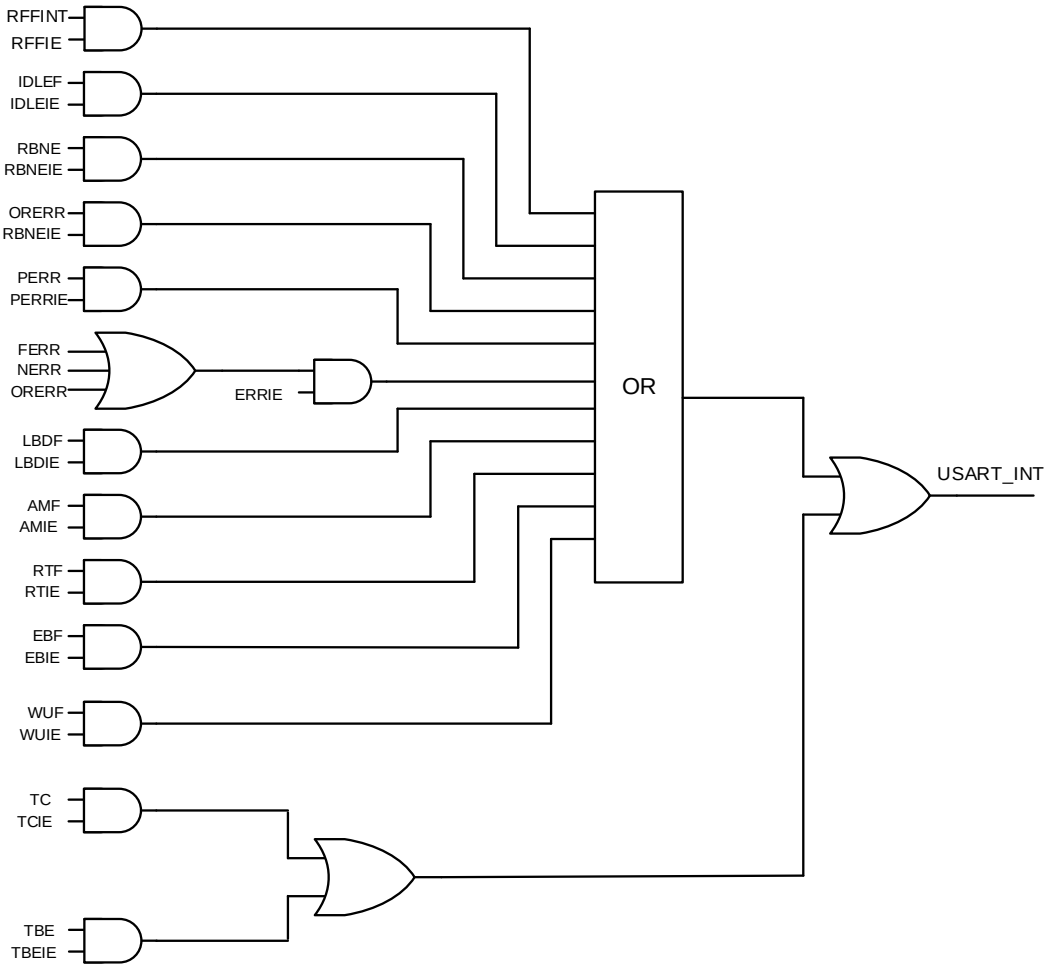
表 20-6. USART 中断请求

中断事件	事件标志	使能控制位
发送数据寄存器空	TBE	TBEIE
发送结束	TC	TCIE
接收到的数据可以读取	RBNE	RBNEIE
检测到过载错误	ORERR	
接收 FIFO 满	RFFINT	RFFIE
检测到线路空闲	IDLEF	IDLEIE

中断事件	事件标志	使能控制位
奇偶校验错误	PERR	PERRIE
LIN 模式下，检测到断开标志	LBDF	LBDIE
当 DMA 接收使能时，接收错误（噪声错误、溢出错误、帧错误）	NERR 或 ORERR 或 FERR	ERRIE
字符匹配	AMF	AMIE
接收超时错误	RTF	RTIE
发现块尾	EBF	EBIE
从 Deepsleep 模式唤醒	WUF	WUIE

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 20-31. USART 中断映射框图



20.2.4. USART 寄存器

USART5基地址：0x4001 7000

USART 控制寄存器 0 （USART_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				EBIE	RTIE	保留									
				rw	rw										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVSMOD	AMIE	MEN	WL	WM	PCEN	PM	PERRIE	TBEIE	TCIE	RBNEIE	IDLEIE	TEN	REN	UESM	UEN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:28	保留	必须保持复位值。
27	EBIE	块尾中断使能。 0：中断禁止 1：中断使能
26	RTIE	接收超时中断使能。 0：中断禁止 1：中断使能
25:16	保留	必须保持复位值。
15	OVSMOD	过采样模式。 0：16 倍过采样。 1：8 倍过采样。 在 LIN, IrDA 和智能卡模式，该位保持清 0。 当 USART 被使能（UEN=1）时，该位域不能被改写。
14	AMIE	ADDR 字符匹配中断使能。 0：ADDR 字符匹配中断禁用。 1：ADDR 字符匹配中断使能。
13	MEN	静默模式使能。 0：静默模式禁用。 1：静默模式被使能。
12	WL	字长 0：8 数据位 1：9 数据位 当 USART 被使能（UEN=1）时，该位域不能被改写。

11	WM	从静默模式唤醒方法。 0：空闲线 1：地址标记 当 USART 被使能（UEN=1）时，该位域不能被改写。
10	PCEN	校验控制使能。 0：校验控制禁用。 1：校验控制被使能。 当 USART 被使能（UEN=1）时，该位域不能被改写。
9	PM	校验模式 0：偶校验 1：奇校验 当 USART 被使能（UEN=1）时，该位域不能被改写。
8	PERRIE	校验错误中断使能。 0：校验错误中断禁用。 1：当 USART_STAT 寄存器的 PERR 位置位时，将触发中断。
7	TBEIE	发送寄存器空中断使能。 0：中断禁止。 1：当 USART_STAT 寄存器的 TBE 位置位时，将触发中断。
6	TCIE	发送完成中断使能。 如果该位置 1，USART_STAT 寄存器中 TC 被置位时产生中断。 0：发送完成中断禁用。 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。 0：读数据缓冲区非空中断和过载错误中断禁用。 1：当 USART_STAT 寄存器的 ORERR 或 RBNE 位置位时，将触发中断。
4	IDLEIE	IDLE 线检测中断使能。 0：IDLE 线检测中断禁用。 1：当 USART_STAT 寄存器的 IDLEF 位置位时，将触发中断。
3	TEN	发送器使能。 0：发送器关闭。 1：发送器打开。
2	REN	接收器使能。 0：接收器关闭。 1：接收器打开并且开始搜索起始位。
1	UESM	USART 在深度睡眠模式下使能。 0：USART 不能从深度睡眠模式唤醒 MCU。 1：USART 能从深度睡眠模式唤醒 MCU。条件是 USART 的时钟源必须是 IRC8M 或 LXTAL。
0	UEN	USART 使能。

0：USART 预分频器和输出禁用。

1：USART 预分频器和输出被使能。

USART 控制寄存器 1 （USART_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
ADDR[7:0]								RTEN	保留			MSBF	DINV	TINV	RINV
rw								rw				rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STRP	LMEN	STB[1:0]		CKEN	CPL	CPH	CLEN	保留	LBDIE	LBLEN	ADDM	保留			
rw	rw	rw		rw	rw	rw	rw		rw	rw	rw				

位/位域	名称	描述
31:24	ADDR[7:0]	USART 的节点地址。 这些位给出 USART 的节点地址。 在多处理器通信并且静默模式或者深度睡眠模式期间，这些位用来唤醒进行地址标记的检测。接收到的最高位为 1 的数据帧将和这些位进行比较。当 ADDM 位被清零时，仅仅 ADDR[3:0]被用来比较。 在正常的接收期间，这些位也用来进行字符检测。所有接收到的字符（8 位）与 ADDR[7:0]的值进行比较，如果匹配，AMF 标志将被置位。 当接收器（REN=1）和 USART（UEN=1） 被使能时，该位域不能被改写。
23	RTEN	接收器超时使能。 0：接收器超时功能禁用。 1：接收器超时功能被使能。
22:20	保留	必须保持复位值。
19	MSBF	高位在前 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。 USART 被使能（UEN=1）时，该位域不能被改写。
18	DINV	数据位反转。 0：数据位信号值没有反转。 1：数据位信号值被反转。 USART 被使能（UEN=1）时，该位域不能被改写。
17	TINV	TX 管脚电平反转。 0：TX 管脚信号值没有反转。 1：TX 管脚信号值被反转。 USART 被使能（UEN=1）时，该位域不能被改写。

16	RINV	RX 管脚电平反转。 0：RX 管脚信号值没有反转。 1：RX 管脚信号值被反转。 USART 被使能（UEN=1）时，该位域不能被改写。
15	STRP	交换 TX/RX 管脚。 0：TX 和 RX 管脚功能不被交换。 1：TX 和 RX 管脚功能被交换 。 当 USART 被使能（UEN=1）时，该位域不能改写。
14	LMEN	LIN 模式使能。 0：LIN 模式关闭。 1：LIN 模式开启。 USART 被使能（UEN=1）时，该位域不能被改写。
13:12	STB[1:0]	STOP 位长 00：1 停止位 01：0.5 停止位 10：2 停止位 11：1.5 停止位 USART 被使能（UEN=1）时，该位域不能被改写。
11	CKEN	CK 管脚使能。 0：CK 管脚禁用。 1：CK 管脚被使能。 USART 被使能（UEN=1）时，该位域不能被改写。
10	CPL	时钟极性 0：在同步模式下，CK 管脚不对外发送时保持为低电平。 1：在同步模式下，CK 管脚不对外发送时保持为高电平。 USART 被使能（UEN=1）时，该位域不能被改写。
9	CPH	时钟相位 0：在同步模式下，在首个时钟边沿采样第一个数据。 1：在同步模式下，在第二个时钟边沿采样第一个数据。 USART 被使能（UEN=1）时，该位域不能被改写。
8	CLEN	CK 长度 0：在同步模式下，最后一位（MSB）的时钟脉冲不输出到 CK 管脚。 1：在同步模式下，最后一位（MSB）的时钟脉冲输出到 CK 管脚。 USART 被使能（UEN=1）时，该位域不能被改写。
7	保留	必须保持复位值。
6	LBDIE	LIN 断开信号检测中断使能。 0：断开信号检测中断禁用。 1：当 USART_STAT 的 LBDF 位置位，将产生中断。
5	LBDL	LIN 断开帧长度。

0：检测 10 位断开帧。
1：检测 11 位断开帧。
USART 被使能（UEN=1）时，该位域不能被改写。

4	ADDM	地址检测模式。 该位用来选择 4 位地址检测或全位地址检测。 0：4 位地址检测。 1：全位地址检测。在 7 位，8 位和 9 位数据模式下，地址检测分别按 6 位，7 位和 8 位地址（ADDR[5:0], ADDR[6:0]和 ADDR[7:0]）执行。 USART 被使能（UEN=1）时，该位域不能被改写。
3:0	保留	必须保持复位值。

USART 控制寄存器 2 （USART_CTL2）

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									WUIE	WUM[1:0]		SCRTNUM[2:0]			保留
									rw	rw		rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		DDRE	OVRD	OSB	保留			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
31:23	保留	必须保持复位值。
22	WUIE	从深度睡眠模式唤醒中断使能。 0：从深度睡眠模式唤醒中断禁用。 1：从深度睡眠模式唤醒中断被使能。
21:20	WUM[1:0]	从深度睡眠模式唤醒模式 这个位域指定什么事件可以置位 USART_STAT 寄存器中的 WUF（从深度睡眠唤醒标志） 标志。 00：WUF 在地址匹配的时候置位。如何实现地址匹配在 ADDR 和 ADDM 中定义。 01:保留 10：WUF 在检测到起始位时置位。 11：WUF 在检测到 RBNE 时置位。 USART 被使能（UEN=1）时，该位域不能被改写。
19:17	SCRTNUM[2:0]	智能卡自动重试数目。 在智能卡模式下，这些位用来指定在发送和接收时重试的次数。在发送模式下，它指的是在产生发送错误（FERR 位置位） 之前自动重试的发送次数。 在接收模式下，它指的是在产生接收错误（RBNE 位和 PERR 位置位）之前自动重试的接收次数。

		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。 USART 被使能（UEN=1）时，该位域被清零，并停止重发。
16:14	保留	必须保持复位值。
13	DDRE	在接收错误时屏蔽 DMA 请求 0: 在发生接收错误的情况下，不禁用 DMA。所有的错误数据不会产生 DMA 请求，以确保错误的数据不会被传输，但是下一个接收到的正确的数据会被传输。在发生接收错误时，RBNE 位保持 0 以阻止过载错误，但是相应错误标志位会被置位。这种模式可用于智能卡模式。 1: 在接收错误的情况下，DMA 请求会被屏蔽，直到相应的标志位被清 0。RBNE 标志和相应的错误标志位会被置位。软件在清除错误标志前，必须首先失能 DMA 接收（DMAR = 0）或清 RBNE。 USART 被使能（UEN=1）时，该位域不能被改写。
12	OVRD	溢出禁止 0：溢出功能被使能。当接收到的数据在新数据到达前没有被读走，ORERR 错误标志位将被置位，并且新数据将会丢失。 1：溢出功能禁止。当接收到的数据在新数据到达前没有被读走，ORERR 错误标志位将不会被置位，新数据会将 USART_ RDATA 寄存器以前的内容覆盖。 USART 被使能（UEN=1）时，该位域不能被改写。
11	OSB	单次采样方式。 0：三次采样方法。 1：一次采样方法。 USART 被使能（UEN=1）时，该位域不能被改写。
10:8	保留	必须保持复位值。
7	DENT	DMA 发送使能。 0：关闭 DMA 发送模式。 1：开启 DMA 发送模式
6	DENR	DMA 接收使能。 0：关闭 DMA 接收模式。 1：开启 DMA 接收模式。
5	SCEN	智能卡模式使能。 0：智能卡模式禁用。 1：智能卡模式使能。 USART 被使能（UEN=1）时，该位域不能被改写。
4	NKEN	智能卡模式 NACK 使能。 0：当出现校验错误时不发送 NACK。 1：当出现校验错误时发送 NACK。 USART 被使能（UEN=1）时，该位域不能被改写。
3	HDEN	半双工使能 0：禁用半双工模式。

		1：开启半双工模式。 USART 被使能（UEN=1）时，该位域不能被改写。
2	IRLP	IrDA 低功耗模式 0：正常模式 1：低功耗模式 USART 被使能（UEN=1）时，该位域不能被改写。
1	IREN	IrDA 模式使能 0：IrDA 禁用 1：IrDA 被使能 USART 被使能（UEN=1）时，该位域不能被改写。
0	ERRIE	多级缓存通信模式的错误中断使能 0：禁用错误中断。 1：在多级缓存通信时，当 USART_STAT 寄存器的 FERR 位，ORERR 位或 NERR 位被置位时，会产生中断。

USART 波特率寄存器 （USART_ BAUD）

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

当 USART（UEN=1）被使能时，该寄存器不能被改写。

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:4	BRR[15:4]	波特率分频系数的整数部分。 INTDIV = BRR[15:4]
3:0	BRR[3:0]	波特率分频系数的小数部分。 如果 OVSMOD = 0，FRADIV = BRR [3:0]。 如果 OVSMOD = 1，FRADIV = BRR [2:0]，BRR [3]必须被置 0。

USART 保护时间和预分频器寄存器 （USART_GP）

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

USART 被使能（UEN=1）时，该寄存器不能被改写。

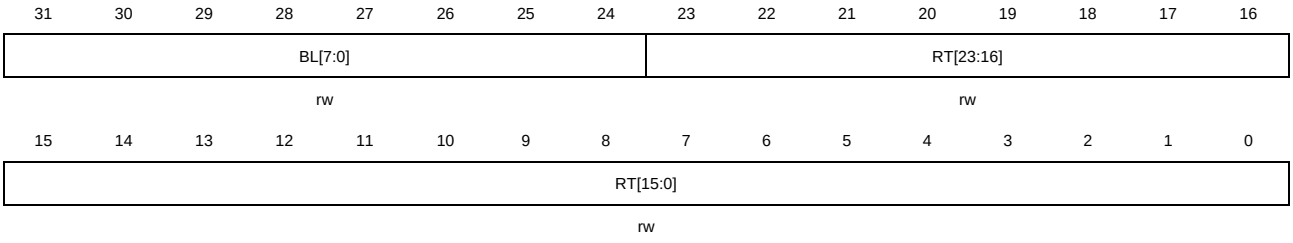
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	GUAT[7:0]	在智能卡模式下的保护时间值。 USART 被使能（UEN=1）时，该位域不能被改写。
7:0	PSC[7:0]	预分频器值 在红外低功耗模式下，对系统时钟进行分频已获得低功耗模式下的频率。寄存器的值是分频系数。 00000000：保留 – 不设置这个值 00000001：1 分频 00000010：2 分频 ... 在 IrDA 正常模式下的分频值 00000001：仅能设为这个值 在智能卡模式下，对系统时钟进行分频的值存于 PSC[4:0]位域中。PSC[7:5]位保持为复位值。分频系数是寄存器中值的两倍。 00000：保留 -不设置这个值 00001：2 分频 00010：4 分频 00011：6 分频 ... USART 被使能（UEN=1）时，该位域不能被改写。

USART 接收超时寄存器 （USART_RT）

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
31:24	BL[7:0]	块长度 这些位给出了智能卡 T=1 的接收时块的长度。它的值等于信息字节的长度+结束部分的长度（1-LEC/2-CRC） – 1。 这个值可以在块接收开始时设置（用于需要从块的序言提取块的长度的情形），这个只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）并且/或者当 EBC 位被写 1 时块的长度计数器被清 0。
23:0	RT[23:0]	接收器超时门限。 该位域指定接收超时值，单位是波特时钟的时长。 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，RTF 标志被置位。 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。 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。 **注意：**当 16 倍过采样时，RT[23:0]不能被配置为 0xFFFFFF。

USART 请求寄存器 （USART_CMD）

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											TXFCMD	RXFCMD	MMCMD	SBKCMD	保留
											w	w	w	w	

位/位域	名称	描述
31:5	保留	必须保持复位值。
4	TXFCMD	发送数据清空请求。 向该位写 1 去置位 TBE 标志位，以取消发送数据。
3	RXFCMD	接收数据清空请求。 向该位写 1 来清除 RBNE 标志位，以丢弃未读的接收数据。
2	MMCMD	静默模式请求。 向该位写 1 使 USART 进入静默模式并且置位 RWU 标志位。
1	SBKCMD	发送断开帧请求。

向该位写 1 置位 SBKF 标志并使 USART 在空闲时发送一个断开帧。

0 保留 必须保持复位值。

USART 状态寄存器 （USART_STAT）

地址偏移：0x1C

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									REA	TEA	WUF	RWU	SBF	AMF	BSY
									r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			EBF	RTF	保留		LBDF	TBE	TC	RBNE	IDLEF	ORERR	NERR	FERR	PERR
			r	r			r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:23	保留	必须保持复位值。
22	REA	接收使能通知标志。 这位反映了 USART 核心逻辑的接收使能状态，该位可以通过硬件设置。 0：USART 核心接收逻辑禁用。 1：USART 核心接收逻辑被使能。
21	TEA	发送使能通知标志。 该位反映了 USART 核心逻辑的发送使能状态，该位可以通过硬件设置。 0：USART 核心发送逻辑禁用。 1：USART 核心发送逻辑被使能。
20	WUF	从深度睡眠模式唤醒标志。 0：没有从深度睡眠模式唤醒。 1：已从深度睡眠模式唤醒，如果在 USART_CTL2 寄存器的 WUFIE=1 并且 MCU 处于深度睡眠模式，将引发一个中断。 当检测到一个唤醒事件时，该位通过硬件置位，这个事件在 WUM 位域被定义。 向 USART_INTC 寄存器中的 WUC 写 1，该位被清 0。 当 UESM 被清 0 时，该位清 0。
19	RWU	接收器从静默模式唤醒。 该位表示 USART 处于静默模式。 0：接收器在工作状态。 1：接收器在静默状态。 当在唤醒和静默模式切换时，它通过硬件清 0 或者置 1。静默模式控制 （地址帧还是空闲帧）是用通过 USART_CTL0 寄存器的 WAKE 位选择。 如果选择空闲信号唤醒，只能通过向 USART_CMD 寄存器的 MMCMD 位写 1 来将该位置位。

18	SBF	断开信号发送标识。 0：没发送断开字符。 1：将要发送断开字符。 该位表示一个断开发送信号被请求。 通过向 USART_CMD 寄存器的 SBKCMD 写 1 来置位。 在断开帧的停止位发送期间，硬件清 0。
17	AMF	ADDR 匹配标志。 0：ADDR 和接收到的字符不匹配。 1：ADDR 和接收到的字符匹配，如果 USART_CTL0 寄存器的 AMIE=1，将引发一个中断。 当接收到 ADDR[7:0]中定义的字符时，硬件置位。 通过向 USART_INTC 寄存器的 AMC 写 1 清 0。
16	BSY	忙标志 0：USART 处于空闲。 1：USART 正在接收。
15:13	保留	必须保持复位值。
12	EBF	块结束标志 0：块没有结束。 1：块结束已到（足够的字节数），如果 USART_CTL1 寄存器的 EBIE=1，将引发一个中断。 当接收到的字节数（从块开始，包括序言部分）等于或大于 BLEN + 4，硬件置位。 通过向 USART_INTC 寄存器的 EBC 写 1 清 0。
11	RTF	接收超时标志。 0：尚未超时。 1：已经超时，如果 USART_CTL1 寄存器的 RTIE 被置位，将会引发中断。 如果空闲的时间已经超过了在 USART_RT 寄存器中设定的 RT 值，通过硬件置 1。 通过向 USART_INTC 寄存器的 RTC 位写 1 清 0。 在智能卡模式，这个超时相当于 CWT 或 BWT 计时。
10:9	保留	必须保持复位值。
8	LBDF	LIN 断开检测标志。 0：没有检测到 LIN 断开字符。 1：检测到 LIN 断开字符。当 USART_CTL1 寄存器的 LBDIE 位被置位时，将会有中断产生。 当 LIN 断开帧被检测到的时候，硬件置位。 通过向 USART_INTC 寄存器的 LBDC 位写 1，清零该位。
7	TBE	发送数据寄存器空。 0：数据没有发送到移位寄存器。 1：数据发送到移位寄存器。如果 USART_CTL0 寄存器的 TBEIE 位置位，将会有中断产生。 当 USART_ TDATA 寄存器的内容已经被转移到移位寄存器或者向 USART_CMD 寄

存器的 TXFCMD 位写 1 时，由硬件置位。
通过向 USART_TDATA 寄存器中写数据来清 0。

6	TC	发送完成 0：发送没有完成。 1：发送完成。如果 USART_CTL0 寄存器的 TCIE 被置位，将会有中断产生。 如果一个包含数据的帧的发送完成且 TBE 被置位，该位由硬件置位。 通过向 USART_INTC 寄存器的 TCC 位写 1 清 0。
5	RBNE	读数据缓冲区非空。 0：没有接收到数据。 1：已接收到数据并且可以读取。当寄存器 USART_CTL0 的 RBNEIE 位被置位，将会有中断产生。 当接收移位寄存器的内容已经被转移到寄存器 USART_ RDATA，由硬件置位。 通过读 USART_RDATA 寄存器或向 USART_CMD 寄存器的 RXFCMD 位写 1 清 0。
4	IDLEF	空闲线检测标志。 0：没检测到空闲线。 1：检测到空闲线。如果 USART_CTL0 寄存器的 IDLEIE 位置 1，将会有中断产生。 当检测到空闲线时，通过硬件置位。直到 RBNE 位置位，否则它不会被再次置位。 向 USART_INTC 寄存器的 IDLEC 位写 1 清 0。
3	ORERR	溢出错误 0：未检测到溢出错误。 1：检测到溢出错误。在多级缓存通信中，如果寄存器 USART_CTL0 的 RBNEIE 位置位，将会引发中断。如果寄存器 USART_CTL2 的 ERRIE 位置位也会引发中断。 在 RBNE 置位的情况下，如果接收移位寄存器的数据传递给 USART_RDATA 寄存器，将会由硬件置位。 向 USART_INTC 寄存器的 OREC 位写 1 清 0。
2	NERR	噪声错误标志。 0：未检测到噪声错误。 1：检测到噪声错误。在多级缓存通信中，如果寄存器 USART_CTL2 的 ERRIE 位置位，将会有中断产生。 在接收帧的时候检测到噪声错误，将会由硬件置位。 向寄存器 USART_INTC 的 NEC 位写 1 清 0。
1	FERR	帧错误 0：未检测到帧错误。 1：检测到帧错误或者断开字符。在多级缓存通信中，如果寄存器 USART_CTL2 的 ERRIE 位置位，将会有中断产生。 当一个不同步，强噪声或者断开字符被检测到时，硬件置位。在智能卡模式下，当发送次数达到上限，仍然没有收到发送成功应答 （卡一直响应 NACKs），该位也将被置位。 向 USART_INTC 寄存器的 FEC 位写 1 清 0。
0	PERR	校验错误 0：未检测到校验错误。

1：检测到校验错误，在多级缓存通信中，如果寄存器 USART_CTL0 的 PERRIE 位置位，将会有中断产生。
当在接收模式的时候检测到校验错误，将会由硬件置位。
向 USART_INTC 寄存器的 PEC 位写 1 清 0。

USART 中断标志清除寄存器 （USART_INTC）

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											WUC	保留		AMC	保留
											w			w	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			EBC	RTC	保留		LBDC	保留	TCC	保留	IDLEC	OREC	NEC	FEC	PEC
			w	w			w		w		w	w	w	w	w

位/位域	名称	描述
31:21	保留	必须保持复位值。
20	WUC	从深度睡眠模式唤醒标志的清除。 向该位写 1 清除 USART_STAT 寄存器的 WUF 位。
19:18	保留	必须保持复位值。
17	AMC	ADDR 匹配标志清除。 向该位写 1 清除 USART_STAT 寄存器的 AMF 位。
16:13	保留	必须保持复位值。
12	EBC	块结束标志清除。 向该位写 1 清除 USART_STAT 寄存器的 EBF 位。
11	RTC	接收超时标志清除。 向该位写 1 清除 USART_STAT 寄存器的 RTF 标志。
10:9	保留	必须保持复位值。
8	LBDC	LIN 断开字符检测标志清除。 向该位写 1 清除 USART_STAT 寄存器的 LBDF 标志位。
7	保留	必须保持复位值。
6	TCC	发送完成标志清除。 向该位写 1 清除 USART_STAT 寄存器的 TC 位。
5	保留	必须保持复位值。
4	IDLEC	空闲线检测标志清除。

向该位写 1 清除 USART_STAT 寄存器的 IDLEF 位。

3	OREC	溢出标志清除。 向该位写 1 清除 USART_STAT 寄存器的 ORERR 位。
2	NEC	噪声检测清除。 向该位写 1 清除 USART_STAT 寄存器的 NERR 位。
1	FEC	帧格式错误标志清除。 向该位写 1 清除 USART_STAT 寄存器的 FERR 位。
0	PEC	校验错误标志清除。 向该位写 1 清除 USART_STAT 寄存器的 PERR 位。

USART 数据接收寄存器 （USART_RDATA）

地址偏移：0x24
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															

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

位/位域	名称	描述
31:9	保留	必须保持复位值。
8:0	RDATA[8:0]	接收数据的值。 包含接收到的数据字节。 如果接收到的数据打开了奇偶校验位（USART_CTL0 寄存器的 PCEN 置 1），那么接收到的数据的最高位（第 7 位或 8 位，取决于数据的长度）是奇偶校验位。

USART 数据发送寄存器 （USART_TDATA）

地址偏移：0x28
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															

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

位/位域	名称	描述
31:9	保留	必须保持复位值。
8:0	TDATA[8:0]	发送数据的值。 包含发送的数据字节。 如果发送到的数据打开了奇偶校验位（USART_CTL0 寄存器的 PCEN 置 1），那么发送的数据的最高位（第 7 位或 8 位取决于数据的长度）将会被奇偶校验位替代。 只有当 USART_STAT 寄存器的 TBE 位被置位时，这个寄存器才可以改写。

USART 兼容性控制寄存器 （USART_CHC）

地址偏移：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							
							rc_w0								

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。 在 RBNE 置位前，校验位被检测到时该标志置位。 软件写 0 可以清除该位。 0：没有检测到校验错误。 1：检测到校验错误。
7:0	保留	必须保持复位值。

USART 接收 FIFO 控制和状态寄存器 （USART_RFCS）

地址偏移：0xD0

复位值：0x0000 0400

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
RFFINT	RFCNT[2:0]			RFF	RFE	RFFIE	RFEN	保留							ELNACK
r_w0	r			r	r	rw	rw								rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	RFFINT	接收 FIFO 满中断标志。
14:12	RFCNT[2:0]	接收 FIFO 计数值。
11	RFF	接收 FIFO 满标志。 0：接收 FIFO 不为满。 1：接收 FIFO 满。
10	RFE	接收 FIFO 空标志。 0：接收 FIFO 不为空。 1：接收 FIFO 空。
9	RFFIE	接收 FIFO 满中断使能。 0：禁止接收 FIFO 满中断。 1：使能接收 FIFO 满中断。
8	RFEN	接收 FIFO 使能。 当 UESM=1，该位置位。 0：禁止使用接收 FIFO。 1：使能接收 FIFO。
7:1	保留	必须保持复位值。
0	ELNACK	若选择了智能卡模式，提前 NACK。 如果检测到校验位错误，NACK 脉冲提前 1/16 位的时间。 0：若选择了智能卡模式，禁止提前 NACK。 1：若选择了智能卡模式，使能提前 NACK。

21. 内部集成电路总线接口（I2C）

21.1. 内部集成电路总线接口（I2Cx, x=0, 1）

21.1.1. 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。

21.1.2. 主要特性

- 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（最高 100 kHz），快速（最高 400 kHz））和快速+ 模式（最高 1MHz）。
- 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。

21.1.3. 功能描述

I2C 接口的内部结构如*图 21-1. I2C 模块框图*所示。

图 21-1. I2C 模块框图

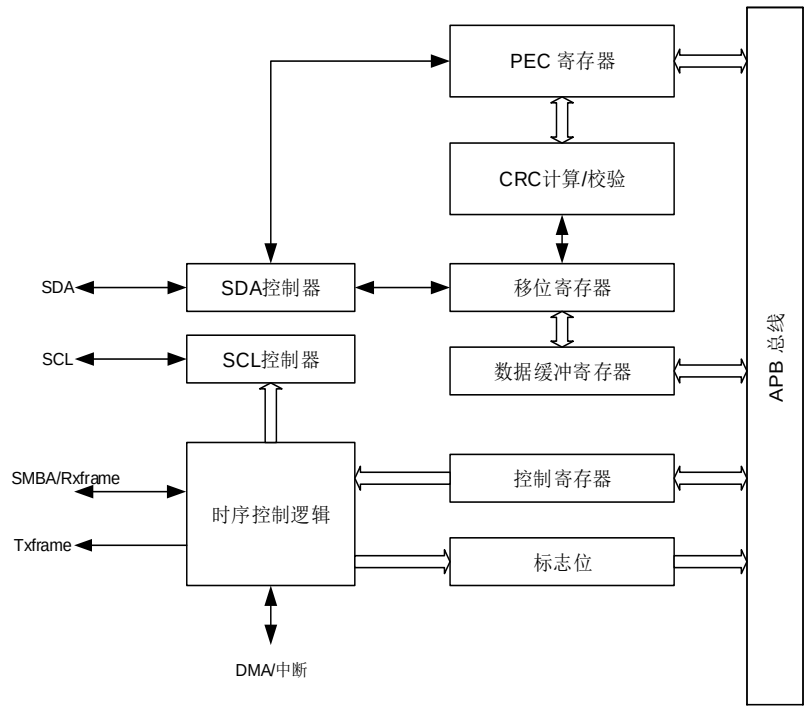


表 21-1. I2C 总线术语说明（参考飞利浦 I2C 规范）

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

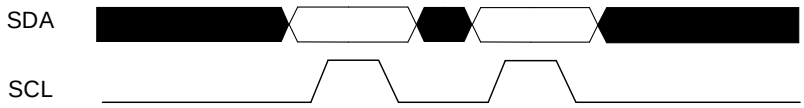
SDA 线和 SCL 线

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_CTL2 中 FMPEN 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。

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 21-2. 数据有效性*）。每个数据比特传输需要一个时钟脉冲。

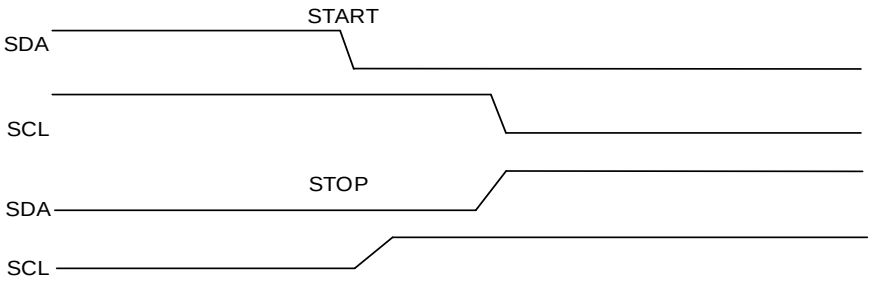
图 21-2. 数据有效性



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 21-3. 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 21-3. 起始和停止信号

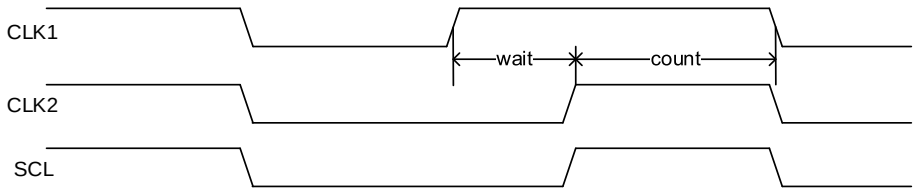


时钟同步

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 21-4. 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 21-4. 时钟同步



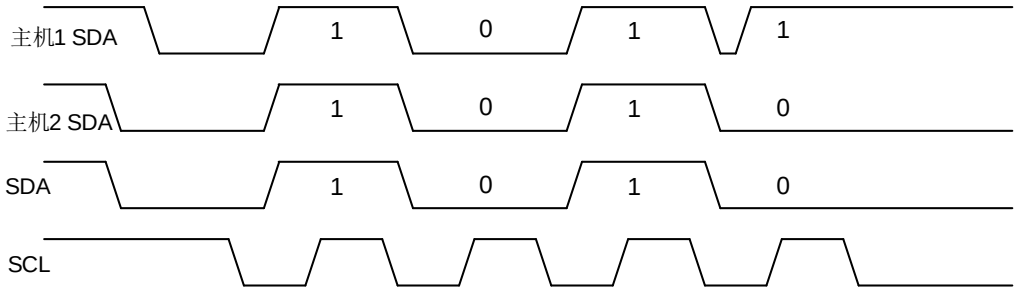
仲裁

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 21-5. SDA 线仲裁



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 21-6. 7 位地址的 I2C 通讯流程

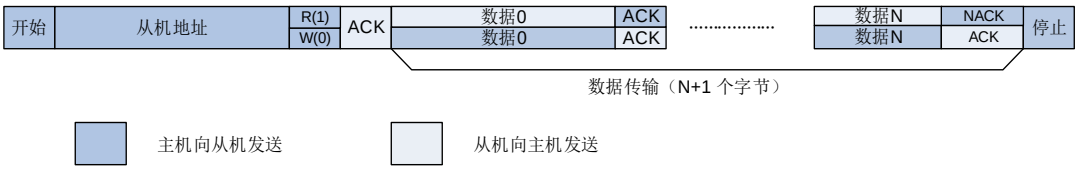


图 21-7. 10 位地址的 I2C 通讯流程（主机发送）

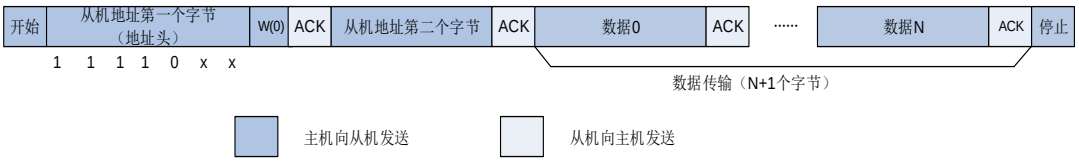
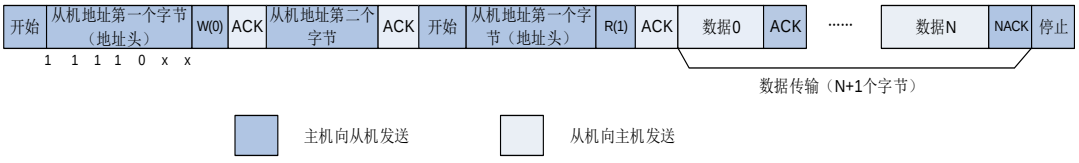


图 21-8. 10 位地址的 I2C 通讯流程（主机接收）



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 21-9. 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 21-9. 从机发送模式（10 位地址模式）

从机接收模式下的软件流程

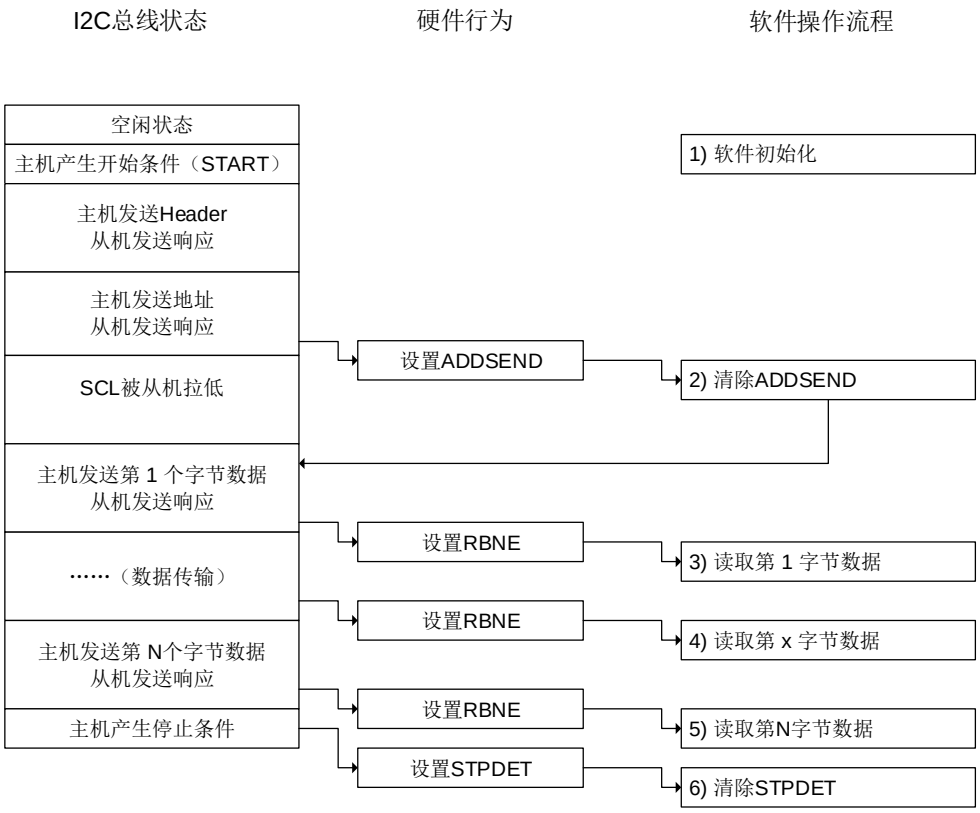
如*图 21-10. 从机接收模式（10 位地址模式）*所示，在从机模式下接收数据时，软件应该遵循这些步骤来操作：

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 21-10. 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 21-11. 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 21-11. 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。

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 21-12. 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 21-13. 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 21-13. 主机接收使用方案 B 模式（10 位地址模式）

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。

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。

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。

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。

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。

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 21-2. 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 21-3. 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

21.1.4. I2C 寄存器

I2C0 基地址：0x4000 5400

I2C1 基地址：0x4000 5800

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 0：I2C 未复位 1：I2C 复位
14	保留	必须保持复位值。
13	SALT	SMBus 警报 通过 SMBA 引脚发出警报。 软件置 1 和清 0，硬件清 0。 0：不通过 SMBA 发布警告 1：通过 SMBA 引脚发送警告
12	PECTRANS	PEC 传输 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。 0：不传输 PEC 值 1：传输的 PEC 值
11	POAP	ACK / PEC 的位置含义 软件置 1 和清 0，当 I2CEN=0 时，硬件清 0。 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。 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。

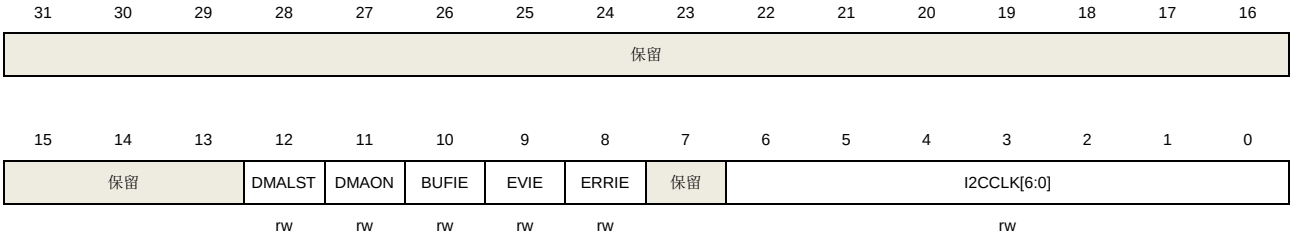
10	ACKEN	是否发送 ACK 软件置 1 和清 0，当 I2CEN=0 时硬件清 0。 0：不发送 ACK 1：发送 ACK
9	STOP	I2C 总线上产生一个 STOP 信号 软件置 1 和清 0，SMBus 超时时，硬件置 1，监测到 STOP 信号时，硬件清 0。 0：不发送 STOP 1：发送 STOP
8	START	I2C 总线上产生一个 START 信号 软件置 1 和清 0，当监测到 START 信号或 I2CEN=0 时由硬件清 0。 0：不发送 START 1：发送 START
7	SS	在从机模式下数据未就绪是否将 SCL 拉低 软件置 1 和清 0。 0：拉低 SCL 1：不拉低 SCL
6	GCEN	是否响应对地址（0x00）的广播呼叫 0：从机不响应广播呼叫 1：从机将响应广播呼叫
5	PECEN	PEC 计算使能 0：PEC 计算禁能 1：PEC 计算使能
4	ARPEN	SMBus 下 ARP 协议使能 0：ARP 禁能 1：ARP 使能
3	SMBSEL	SMBus 类型选择 0：从机 1：主机
2	保留	必须保持复位值。
1	SMBEN	SMBus / I2C 模式开关 0：I2C 模式 1：SMBus 模式
0	I2CEN	I2C 外设使能 0：禁能 I2C 1：使能 I2C

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 0：下一个 DMA EOT 不是最后传输 1：下一个 DMA EOT 是最后传输
11	DMAON	DMA 模式开关 0：DMA 模式关 1：DMA 模式开
10	BUFIE	缓冲区中断使能 0：禁用缓冲区中断，如果 EVIE=1，当 TBE = 1 或 RBNE = 1 时不产生中断。 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 0：禁用事件中断 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 0：禁用错误中断 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	保留	必须保持复位值
6:0	I2CCLK[6:0]	I2C 外设时钟频率 I2CCLK[6:0]应该是输入 APB1 时钟频率，最低 2MHz。 0000000 - 0000001：无时钟 0000010 - 1011010：2 MHz~90MHz0000010 - 1011010：2 MHz~90MHz 1011011 - 1111111：由于 APB1 时钟限制，无时钟 注意： 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。

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。

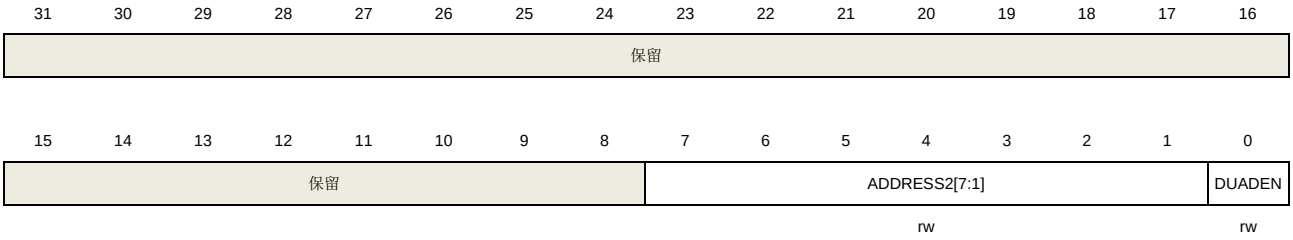
位/位域	名称	描述
31:16	保留	必须保持复位值。
15	ADDFORMAT	I2C 从机地址格式 0：7 位地址 1：10 位地址
14:10	保留	必须保持复位值。
9:8	ADDRESS[9:8]	10 位地址的最高两位
7:1	ADDRESS[7:1]	7 位地址或者 10 位地址的第 7-1 位
0	ADDRESS0	10 位地址的第 0 位

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

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。

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								TRB[7:0]							
								rw							

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

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。

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
SMBALT	SMBTO	保留	PECERR	OUERR	AERR	LOSTARB	BERR	TBE	RBNE	保留	STPDET	ADD10SEND	BTC	ADDSEND	SBSEND
rc_w0	rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	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	SMBALT	SMBus 警报状态 硬件置 1，软件写 0 清 0。 0：SMBA 引脚未被拉低（从机模式）或未监测到警报（主机模式） 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 硬件置 1，软件写 0 清 0。 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 硬件置 1，软件写 0 清 0。 0：接收到 PEC 且校验正确 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。 硬件置 1，软件写 0 清 0。 0：无上溢或下溢错误发生 1：发生上溢或下溢错误
10	AERR	应答错误 硬件置 1，软件写 0 清 0。 0：未发生应答错误 1：发生了应答错误
9	LOSTARB	主机模式下仲裁丢失 硬件置 1，软件写 0 清 0。 0：无仲裁丢失 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。 硬件置 1，软件写 0 清 0。 0：无总线错误 1：发生了总线错误
7	TBE	发送期间 I2C_DATA 为空 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） 0：I2C_DATA 非空 1：I2C_DATA 空，软件可以写
6	RBNE	接收期间 I2C_DATA 非空 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。 0：I2C_DATA 为空 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 该位由硬件置 1，软件读 I2C_STAT0 和写 I2C_DATA 清除此位。 0：主机模式下未发送 10 位地址的地址头 1：主机模式下发送 10 位地址的地址头
2	BTC	字节发送结束 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。 此位由硬件置 1。 可由以下三种方式清除： 1、软件清除：读 I2C_STAT0，然后读或写 I2C_DATA 寄存器清除此位 2、硬件清除：发送一个 STOP 或 START 信号 3、寄存器 I2C_CTL0 中 I2CEN=0 0：未发生 BTC 1：发生了 BTC
1	ADDSEND	主机模式下：成功发送了地址并收到 ACK 从机模式下：接收到的地址与自身的地址匹配 此位由硬件置 1，软件读 I2C_STAT0 寄存器和读 I2C_STAT1 清 0。 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 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 此位由硬件置 1，软件读 I2C_STAT0 和写 I2C_DATA 清 0。 0：未发送 START 信号 1：START 信号被发送

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

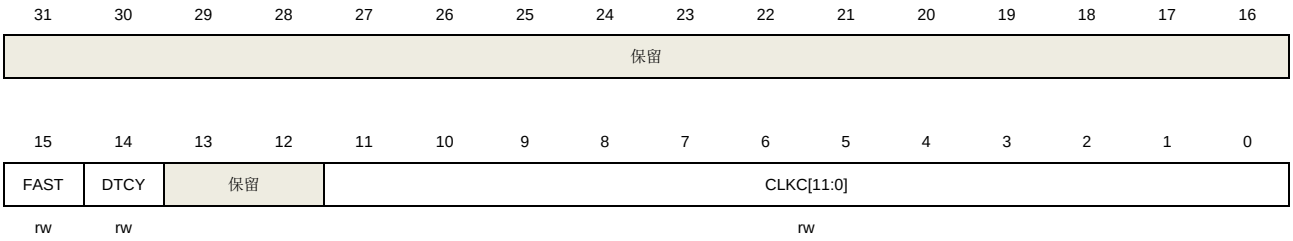
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	PECV[7:0]	当 PEC 使能后硬件计算出的 PEC 值。
7	DUMODF	从机模式下双标志位表明哪个地址和双地址模式匹配 STOP 或 START 信号产生后或 I2CEN=0 时此位由硬件清 0。 0：地址和 I2C_SADDR0 匹配 1：地址和 I2C_SADDR1 匹配
6	HSTSMB	从机模式下监测到 SMBus 主机地址头 STOP 或 START 信号产生后或 I2CEN=0 时此位由硬件清 0。 0：未监测到 SMBus 主机地址头 1：监测到 SMBus 主机地址头
5	DEFSMB	SMBus 设备缺省地址 STOP 或 START 信号产生后或 I2CEN=0 时此位由硬件清 0。 0：SMBus 设备没有接收到缺省地址 1：SMBus 设备接收到缺省地址
4	RXGC	是否接收到广播地址（0x00） STOP 或 START 信号产生后或 I2CEN=0 时此位由硬件清 0。 0：未接收到广播呼叫地址 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 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。 0：接收端 1：发送端
1	I2CBSY	忙标志 STOP 信号后硬件清 0。 0：无 I2C 通讯 1：I2C 正在通讯
0	MASTER	主机模式 表明 I2C 时钟在主机模式还是从机模式的标志位。 该位在 START 信号产生后由硬件置 1。 该位在 STOP 信号产生后或 I2CEN=0 或 LOSTARB=1 时此位由硬件清 0。 0：从机模式 1：主机模式

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。



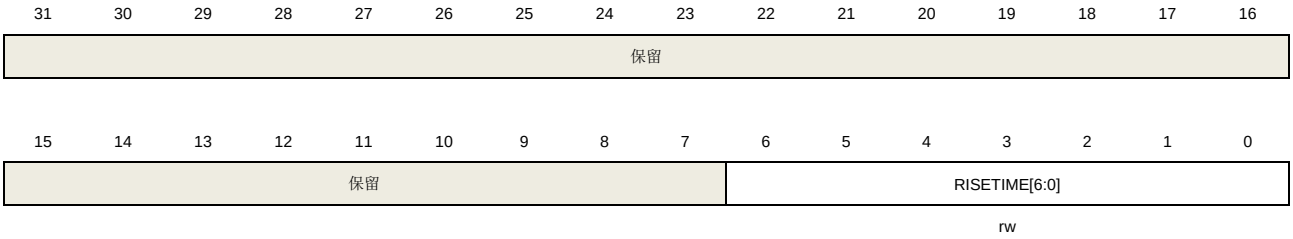
位/位域	名称	描述
31:16	保留	必须保持复位值。
15	FAST	主机模式下 I2C 速度选择 0：标准速度 1：快速
14	DTCY	快速模式或快速+ 模式下占空比 0：$T_{low}/T_{high}=2$ 1：$T_{low}/T_{high}=16/9$
13:12	保留	必须保持复位值。
11:0	CLKC[11:0]	主机模式下 I2C 时钟控制 标准速度模式下：$T_{high}=T_{low}=CLKC*T_{PCLK1}$ 如果 DTCY=0，快速模式或快速+ 模式下： $T_{high}=CLKC*T_{PCLK1}$，$T_{low}=2*CLKC*T_{PCLK1}$ 如果 DTCY=1，快速模式或快速+ 模式下： $T_{high}=9*CLKC*T_{PCLK1}$，$T_{low}=16*CLKC*T_{PCLK1}$ 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。

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

31:7	保留	必须保持复位值。
6:0	RISETIME[6:0]	主机模式下最大上升时间 RISETIME 值应该为 SCL 最大上升时间加 1

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 0：接收帧上升沿中断失能 1：接收帧上升沿中断使能
6	RFFIE	接收帧下降沿中断使能 0：接收帧下降沿中断失能 1：接收帧下降沿中断使能
5	TFRIE	发送帧上升沿中断使能 0：发送帧上升沿中断失能 1：发送帧上升沿中断使能
4	TFFIE	发送帧下降沿中断使能 0：发送帧下降沿中断失能 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 0：SAM_V 接口超时检测失能 1：SAM_V 接口超时检测使能
0	SAMEN	SAM_V 接口使能 0：SAM_V 接口失能 1：SAM_V 接口使能

控制寄存器 2（I2C_CTL2）

地址偏移：0x90

复位值：0x0000 FE00

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
ADDM[6:0]							RADD	保留			TOEN	保留		SETM	FMPEN
rw							rw				rw			rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:9	ADDM[6:0]	定义 ADDRESS[7:1]的哪些位和接收到的地址进行比较，哪些位不比较。ADDM[6:0]中设置为 1 的位使能 ADDRESS[7:1]中的相应位与接收到的地址进行比较，设置为 0 的位则忽视（此时接收到的地址在该位可以为 0 或 1）。
8	RADD	当该位为 1 时，接收到的 7 位从机地址或者 10 位从机地址的[7:0]位将会被记录在传输缓冲寄存器（I2C_DATA）里。
7:5	保留	必须保持复位值。
4	TOEN	超时计数使能。当没有使能 SMBA 模式时，该位用来使能 25ms SCL 低电平状态超时计数。 0：禁能 SCL 低电平状态超时计数 1：使能 SCL 低电平状态超时计数
3:2	保留	必须保持复位值。
1	SETM	启动提前终止模式。该位描述了当 I2C 在启动阶段之后，SCL 处于高电平状态时，如何处理接下来 SCL 的低电平状态。 0：终止 SCL 的高电平状态，开始计数 SCL 的低电平状态（和 I2C 标准协议一样） 1：处理方法和仲裁丢失时一样
0	FMPEN	快速+ 模式使能

当该位被置 1 时，I2C 设备支持高达 1MHz 的通信速率。

控制和状态寄存器（I2C_CS）

地址偏移：0x94

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
31:10	保留	必须保持复位值。
9	STPSENDIE	停止信号发送中断使能 0：禁能中断 1：使能中断
8	STLOIE	起始信号丢失中断使能 0：禁能中断 1：使能中断
7:2	保留	必须保持复位值。
1	STPSEND	主机模式下停止信号发送 此位被硬件置 1，软件写 0 清 0。 0：没有停止信号发送 1：停止信号发送
0	STLO	起始信号丢失标志位 如果 SETM 位设置为 1，则当 I2C 处于开始开始保持定时阶段（$t_{HD:STA}$）时，SCL 线的低电平发生时，该位置位。 此位被硬件置 1，软件写 0 清 0。 0：没有发生起始信号丢失 1：发生了起始信号丢失

状态清除寄存器（I2C_STATC）

地址偏移：0x98

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
SRCEN	保留										STOPFC	ADD10SENDC	BTCC	ADDSENDC	SBSENDC
rw											w	w	w	w	w

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	SRCEN	状态寄存器清除使能位 当该位置 1 时，STOPF、ADD10SEND、BTC、ADDSEND、SBSEND 状态位可以通过分别向 STOPFC、ADD10SENDC、BTCC、ADDSENDC、SBSENDC 位写 1 来清除。一旦 SRCEN 位置 1，那么 STOPF、ADD10SEND、BTC、ADDSEND、SBSEND 状态位不能被其它的软件序列清除。 0：禁能清除功能 1：使能清除功能
14:5	保留	必须保持复位值。
4	STOPFC	STOPF 状态清除位 当 SRCEN 位置 1 时，软件可以通过往该位写 1 来清除 I2C_STAT0 寄存器的 STPDET 位
3	ADD10SENDC	ADD10SEND 状态清除位 当 SRCEN 位置 1 时，软件可以通过往该位写 1 来清除 I2C_STAT0 寄存器的 ASS10SEND 位
2	BTCC	BTC 状态清除位 当 SRCEN 位置 1 时，软件可以通过往该位写 1 来清除 I2C_STAT0 寄存器的 BTC 位
1	ADDSENDC	ADDSEND 状态清除位 当 SRCEN 位置 1 时，软件可以通过往该位写 1 来清除 I2C_STAT0 寄存器的 ASSSEND 位
0	SBSENDC	起始发送状态清除位 当 SRCEN 位置 1 时，软件可以通过往该位写 1 来清除 I2C_STAT0 寄存器的 SBSEND 位

21.2. 内部集成电路总线接口（I2Cx, x=2）

21.2.1. 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）和 PMBus（电源管理总线）。此外，I2C 接口模块还支持多主机 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。

21.2.2. 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。
- 多个 7 位从机地址（两个地址，其中一个可配置地址位屏蔽）。
- 可编程的建立时间和保持时间。
- 支持 I2C 多主机模式。
- 支持标速（最高 100 kHz），快速（最高 400 kHz）和快速+ 模式（最高 1MHz）。
- 从机模式下可配置的 SCL 主动拉低。
- 支持 DMA 模式。
- 兼容 SMBus 3.0 和 PMBus 1.3。
- 可选择的 PEC（报文错误校验）生成和校验。
- 可编程模拟过滤器和数字过滤器。
- I2C 地址匹配时，由深度睡眠模式，深度睡眠模式 1 和深度睡眠模式 2 唤醒。
- 独立于 PCLK 的时钟。

21.2.3. 功能说明

I2C 接口的内部结构如*图 21-14. I2C 模块框图*所示。

图 21-14. I2C 模块框图

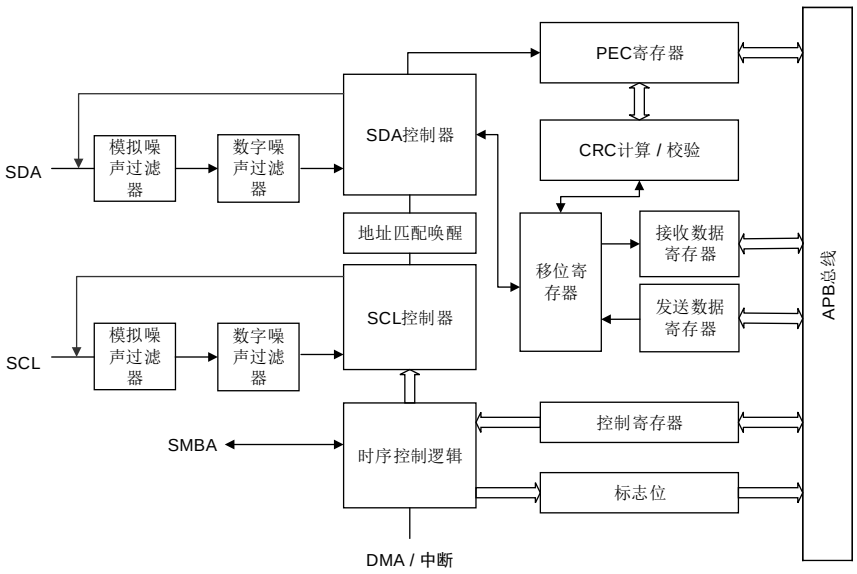


表 21-4. I2C 总线术语说明（参考飞利浦 I2C 规范）

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	不破坏信息的前提下同时控制总线的多个主机
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，保证上述的过程叫仲裁

时钟要求

I2C 时钟独立于 PCLK 时钟，因此可以独立操作 I2C。

I2C 时钟（I2CCLK）可以从以下三个时钟源中选择：

- APB1 时钟 PCLK1（默认值）
- 内部 8M RC 时钟 IRC8M
- 系统时钟 SYSCLK

I2C 时钟周期t_{I2CCLK}必须满足以下条件：

- $t_{I2CCLK}<(t_{LOW}-t_{filters})/4$
- $t_{I2CCLK}<t_{HIGH}$
-

其中：

t_{LOW}：SCL 低电平时间

t_{HIGH}：SCL 高电平时间

$t_{filters}$：在使能滤波器时，表示模拟滤波器和数字滤波器产生的延时总和。模拟滤波器产生的延时最大值为 160ns，数字滤波器产生的延时为DNF[3:0]×t_{I2CCLK}。

PCLK 时钟周期t_{PCLK}必须满足以下条件：

- $t_{PCLK}<4/3*t_{SCL}$

其中：

t_{SCL}：SCL 周期

注意：当 I2C 内核时钟由 PCLK 提供时，PCLK 必须符合t_{I2CCLK}的条件。

I2C 通讯流程

主机和从机都能实现数据收发，因此，I2C 可以实现四种工作模式：

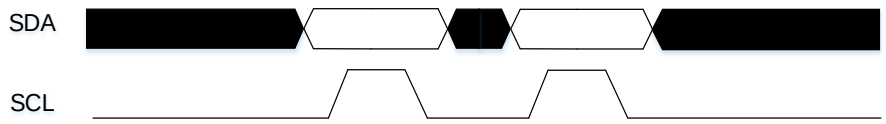
- 从机发送
- 从机接收

- 主机发送
- 主机接收

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 21-15. 数据有效性*。每个数据比特传输需要一个时钟脉冲。

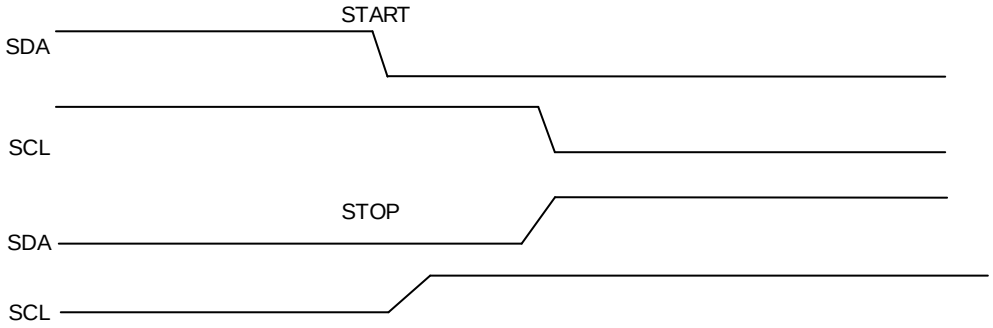
图 21-15. 数据有效性



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 21-16. 开始和停止信号*）。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 结束位定义为，在 SCL 为高时，SDA 线上出现一个从低到高的电平转换。

图 21-16. 开始和停止信号



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。在默认情况下，I2C 设备工作在从机模式下。当 START 信号产生时，I2C 设备由从机模式切换成主机模式。如果仲裁丢失或者 STOP 信号产生时，I2C 由主机模式切换成从机模式。支持 I2C 多主机模式。

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 模块支持 7 位和 10 位的地址模式。

数据和地址都是 8 位传输，高位在前。START 信号之后的字节（在 7 位地址模式下是一个字节，10 位地址模式下是两个字节）是主机发送的从机地址。

8 个时钟周期字节发送后，第 9 个时钟脉冲期间接收器会发送应答信号至发送器。是否产生 ACK 信号可以软件配置。

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。

在主机模式下，如果 AUTOEND=1，STOP 信号由硬件产生。如果 AUTOEND=0，STOP 信号由软件产生，或者主机可以产生 RESTART 信号来启动新的数据传输。

图 21-17. 10 位地址的 I2C 通讯流程（主机发送）

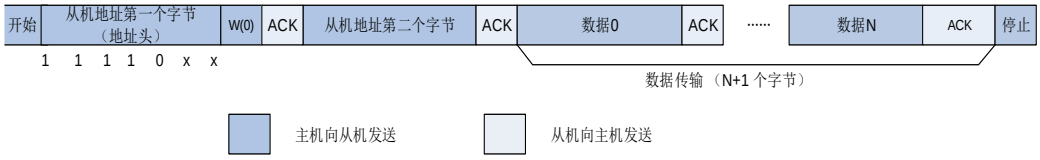


图 21-18. 7 位地址的 I2C 通讯流程（主机发送）

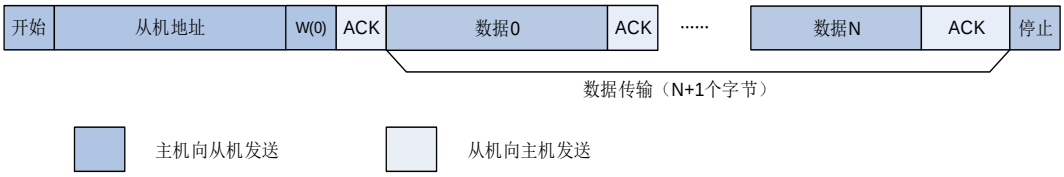
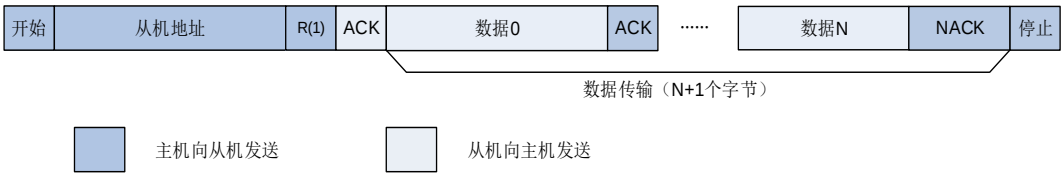


图 21-19. 7 位地址的 I2C 通讯流程（主机接收）



在 10 位寻址模式中，配置 HEAD10R 位可以选择执行完整的寻址序列或只发送地址头。当 HEAD10R=0，执行完整的 10 位地址寻址读序列 START+10 位地址头（写）+第二个地址字节 +RESTART+10 位地址头（读），如*图 21-20. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R=0）*所示。

在 10 位寻址模式中，如果主机接收是在主机发送结束后执行，读寻址序列可以是 RESTART+10 位地址头（读），如*图 21-21. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R=1）*所示

图 21-20. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R=0）

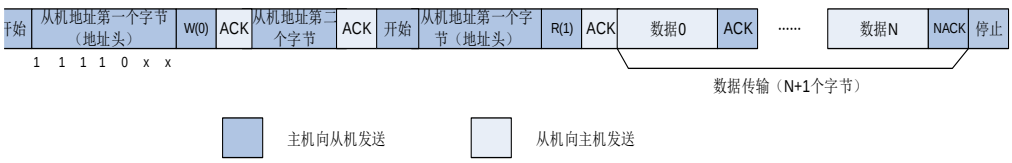
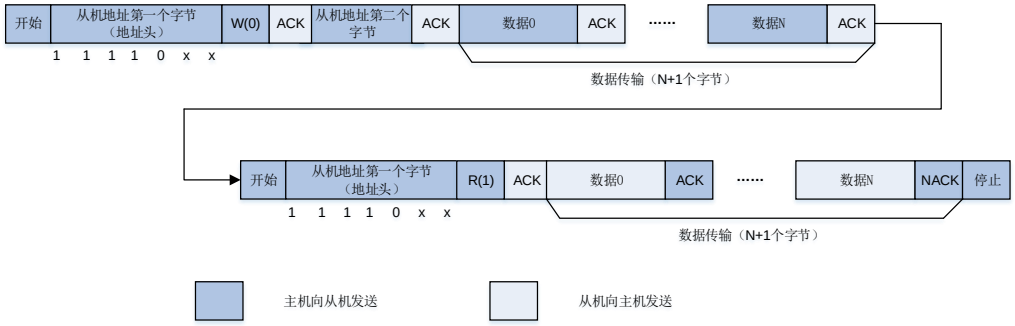


图 21-21. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R=1）



噪声滤波器

I2C 外设集成了模拟噪声滤波器和数字噪声滤波器，噪声滤波器可根据实际需要在 I2C 外设启用前进行配置。

将 I2C_CTL0 寄存器中 ANOFF 位置 1 可以禁用模拟噪声滤波器，将 ANOFF 位清 0 时使能模拟噪声滤波器。在快速模式和快速+ 模式下，模拟滤波器需要抑制脉冲宽度高达 50ns 的峰值。

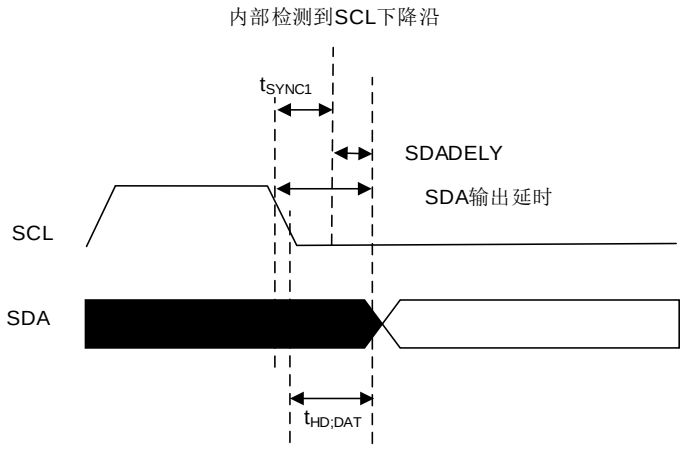
数字滤波器由 I2C_CTL0 寄存器中 DNF[3:0]位来配置。当数字滤波器使能时，SCL 和 SDA 电平保持稳定的时间大于DNF[3:0]×t_{I2CCLK}才会发生内部变化。抑制峰值宽度可由 DNF[3:0]配置。

I2C 时序配置

在 I2C 通信中，I2C_TIMING 寄存器中 PSC[3:0]，SCLDELY[3:0]和 SDADELY[3:0]用于保证正确的数据保持时间和数据建立时间。

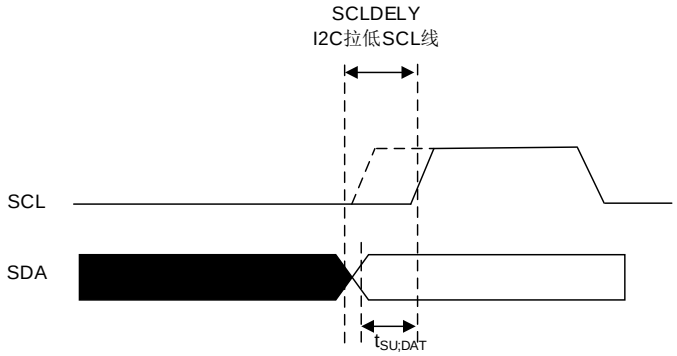
如果数据已经在 I2C_TDATA 寄存器中，在经历 SDADELY 延时后，数据由 SDA 发送，如*图 21-22. 数据保持时间*所示。

图 21-22. 数据保持时间



当数据经过 SDA 发送时，SCLDELY 计数器开启。如*图 21-23. 数据建立时间*所示。

图 21-23. 数据建立时间



当内部检测到 SCL 下降沿时，在 SDA 发送之前会插入一个延时。该延时为 $t_{SDADELY}$=SDADELY*t_{PSC}+t_{I2CCLK}，其中t_{PSC}=（PSC+1)*t_{I2CCLK}。$t_{SDADELY}$会影响$t_{HD;DAT}$。SDA 输出总延时为t_{SYNC1}+{[SDADELY*(PSC+1)+1]*t_{I2CCLK}}。t_{SYNC1}由 SCL 下降斜率，模拟滤波器延时，数字滤波器延时和 SCL 与 I2CCLK 时钟的同步延时共同决定。SCL 与 I2CCLK 时钟的同步延时为 2 至 3 个t_{I2CCLK}。

注意：当SDADELY=0时，$t_{SDADELY}$=t_{PSC}。

SDADELY 必须符合以下条件：

- SDADELY≥{t_f(max)+$t_{HD;DAT}$(min)-t_{AF}(min)-[（DNF+3)*t_{I2CCLK}]}/[(PSC+1)*t_{I2CCLK}]
- SDADELY≤{$t_{HD;DAT}$（max)-t_{AF}(max)-[(DNF+4)*t_{I2CCLK}]}/[(PSC+1)*t_{I2CCLK}]

注意：t_{AF}为模拟滤波器延时，$t_{HD;DAT}$必须小于$t_{VD;DAT}$的最大值。

当 SS=0 时，经过延时$t_{SDADELY}$，在数据写入 I2C_TDATA 寄存器之前，从机会拉低时钟线。在数据建立时间期间 SCL 保持低电平。数据建立时间$t_{SCLDELY}$=(SCLDELY+1)*t_{PSC}。$t_{SCLDELY}$影响$t_{SU;DAT}$。

SCLDELY 必须符合以下条件：

- SCLDELY≥{[t_r(max)+$t_{SU;DAT}$(min)]/[(PSC+1)*t_{I2CCLK}]}-1

在主机模式下，SCL 时钟高低电平由 I2C_TIMING 寄存器中 PSC[3:0]，SCLH[7:0]和 SCLL[7:0]控制。

当内部检测到 SCL 下降沿，在释放 SCL 输出之前会插入一个延时，该延时为 t_{SCLL}=(SCLL+1)*t_{PSC}，其中t_{PSC}=(PSC+1)*t_{I2CCLK}。t_{SCLL}影响 SCL 低电平持续时间t_{LOW}。

当内部检测到 SCL 上升沿，在将 SCL 拉低之前会插入一个延时，该延时为 t_{SCLH}=(SCLH+1)*t_{PSC}，其中t_{PSC}=(PSC+1)*t_{I2CCLK}。t_{SCLH}影响 SCL 高电平持续时间t_{HIGH}。

注意：时序配置和 SS 位在 I2C 外设使能时是不能改变的。

表 21-5. 数据建立时间和数据保持时间

符号	参数	标准模式		快速模式		快速 +模式		SMBus		单位
		最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值	
$t_{HD;DAT}$	数据保持时间	0	-	0	-	0	-	0.3	-	us
$t_{VD;DAT}$	数据有效时间	-	3.45	-	0.9	-	0.45	-	-	
$t_{SU;DAT}$	数据建立时间	250	-	100	-	50	-	250	-	ns

t_r	SCL 和 SDA 信号上升时间	-	1000	-	300	-	120	-	1000	
t_f	SCL 和 SDA 信号下降时间	-	300	-	300	-	120	-	300	

I2C 复位

清除 I2C_CTL0 寄存器中 I2CEN 位可以实现软件复位。当软件复位产生时，SCL 和 SDA 均被释放。通信控制位和状态位也还原成复位值。软件复位对配置寄存器无影响。受到影响的位为 I2C_CTL1 寄存器中 START，STOP 和 NACKEN，I2C_STAT 寄存器中 I2CBSY，TBE，TI，RBNE，ADDSEND，NACK，TCR，TC，STPDET，BERR，LOSTARB 和 OUERR。另外，如果支持 SMBus 模式，I2C_CTL1 寄存器中 PECTRANS 位，I2C_STAT 寄存器中 PECERR，TIMEOUT 和 SMBALT 位也会受到影响。

为了实现软件复位，I2CEN 必须在至少 3 个 APB 时钟周期内保持低电平。可以通过以下写软件序列来保证软件复位：

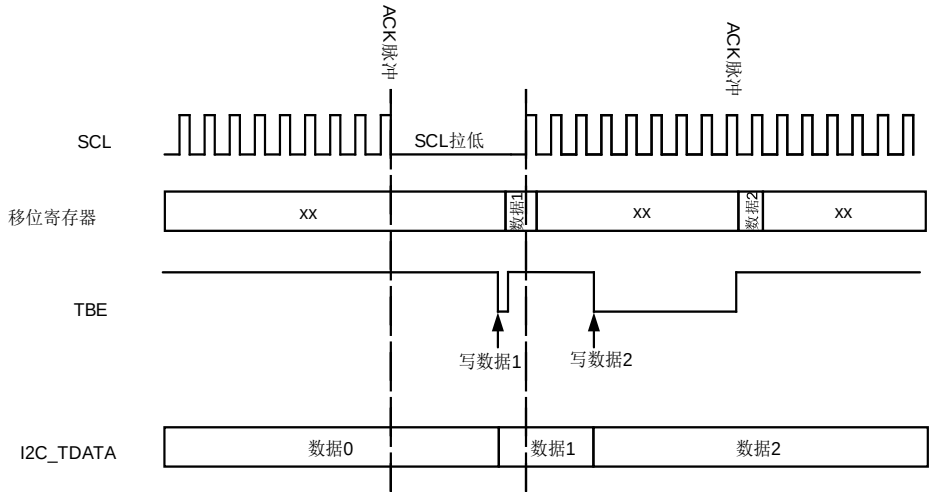
- I2CEN 写 0
- 检查 I2CEN 是否为 0
- I2CEN 写 1

数据传输

数据发送

在发送数据时，如果 TBE 为 0，表明 I2C_TDATA 寄存器非空，在第九个 SCL 脉冲（应答脉冲）后，I2C_TDATA 寄存器中的数据移入到移位寄存器。移位寄存器中的数据通过 SDA 线移出。如果 TBE 为 1，则表明 I2C_TDATA 寄存器为空，在 I2C_TDATA 不为空之前 SCL 将被拉低。SCL 拉低是在第九个 SCL 脉冲之后。

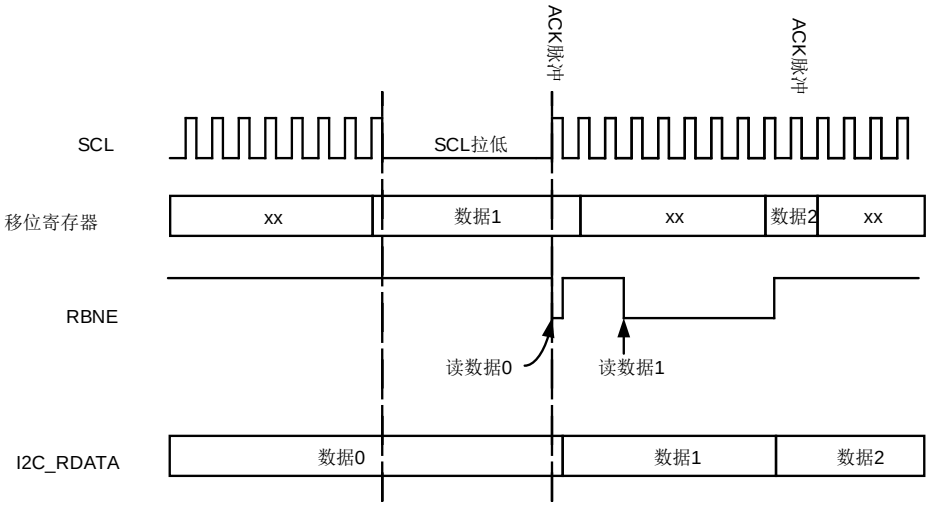
图 21-24. 数据发送



数据接收

在接收数据时，数据首先被接收到移位寄存器。如果 RBNE 为 0，移位寄存器中的数据将被移入 I2C_RDATA 寄存器。如果 RBNE 为 1，SCL 时钟将被拉低，直到之前接收到的数据字节被读取。这个时钟拉低被插入应答脉冲之前。

图 21-25. 数据接收



重载和自动结束模式

为了管理字节传输和中断如*表 21-6. 可关闭通信模式*所示几种通信模式，I2C 硬件嵌入了字节计数器。

表 21-6. 可关闭通信模式

工作模式	行为
主机模式	产生 NACK，STOP 和 RESTART
从机接收模式	ACK 控制
SMBus 模式	PEC 生成/校验

传输的字节数由 BYTENUM[7:0]在 I2C_CTL1 寄存器中配置。如果 BYTENUM 大于 255，或者处于从机字节控制模式，则必须通过将 I2C_CTL1 寄存器中 RELOAD 位置 1 来使能重载模式。在重载模式下，当 BYTENUM 计数到 0 时，TCR 位将置 1，如果 TCIE 位置 1 将产生中断。当 TCR 位置 1 时，SCL 将被拉低。在 BYTENUM 写一个非零值将清除 TCR 位。

注意：重载模式必须在 BYTENUM[7:0]最后一次重载后禁用。

当使能自动结束模式时，必须禁用重载模式。在自动结束模式下，当 BYTENUM[7:0]计数到 0 时，主机将自动发送一个 STOP 信号。

当重载模式和自动结束模式都被禁用时，I2C 通信进程需要由软件终止。如果 BYTENUM[7:0]中的字节数已经传输完成，软件应将 STOP 位置 1 来产生一个 STOP 信号，然后清除 TC。

I2C 从机模式

初始化

从机模式下，至少使能一个从机地址。第一个从机地址写在 I2C_SADDR0 寄存器中，第二个

从机地址写在 I2C_SADDR1 寄存器中。在使用从机地址时，必须相应地将 I2C_SADDR0 寄存器中 ADDRESSEN 位和 I2C_SADDR1 寄存器中 ADDRESS2EN 置 1。通过设置 I2C_SADDR0 寄存器中 ADDFORMAT 位可以选择 7 位地址或 10 位地址，该地址被写在 ADDRESS[9:0]。

ADDMSK2[2:0]用于屏蔽 I2C_SADDR1 寄存器中 ADDRESS2[7:1]，相关详细信息参考 I2C_SADDR1 寄存器 ADDMSK2[2:0]位域描述。

当 I2C 接收到的地址与使能的地址其中一个匹配成功时，ADDSEND 将被置 1，如果 ADDMIE 置位，将产生中断。I2C_STAT 寄存器 READDR[6:0]将会存储接收到的地址。在 ADDSEND 置位时，I2C_STAT 寄存器中 TR 位状态更新。TR 的状态指示从机是作为发送器还是接收器。

SCL 线控制

当 SS=0 时，时钟拉低功能默认用在从机模式下，在需要的时候 SCL 会被拉低。在下列情况下，SCL 会被拉低。

- 当 ADDSEND 置位时 SCL 线拉低，并在 ADDSEND 位清零之后释放。
- 在从机发送模式下，ADDSEND 清零之后，SCL 在第一个字节写入 I2C_TDATA 寄存器之前都是被拉低的。在前一个字节发送完成之后，新的字节写入 I2C_TDATA 寄存器之前，SCL 也是被拉低的。
- 在从机接收模式下，接收过程已完成但是 I2C_RDATA 寄存器中的数据还未被读取，SCL 将被拉低。
- 当 SBCTL=1 且 RELOAD=1 时，在最后一个字节传输结束后，TCR 置位。在 TCR 清除之前 SCL 将被拉低。
- SCL 下降沿被检测到之后，在[(SDADELY+SCLDELY+1)*(PSC+1)+1]*t_{I2CCLK}期间 SCL 被拉低。

SCL 线控制可以通过将 I2C_CTL0 寄存器中 SS 位置 1 来禁能。在下列情况下，SCL 不会被拉低。

- 在 ADDSEND 置位时 SCL 将不会被拉低。
- 在从机发送模式下，数据必须在它传输过程产生的第一个 SCL 脉冲之前写入 I2C_TDATA 寄存器。否则 I2C_STAT 寄存器中 OUERR 位将会置 1，如果 ERRIE 位也被置 1，将产生一个中断。当 STPDET 位置 1 并且第一个数据开始发送，I2C_STAT 寄存器中 OUERR 位也将置 1。
- 在从机接收模式下，数据必须在下一个字节接收产生的第九个 SCL 脉冲（ACK 脉冲）之前读取。否则 I2C_STAT 寄存器中 OUERR 位也将置 1。如果 ERRIE 位也被置 1，将产生一个中断。

从机字节控制模式

在从机接收模式下要实现字节 ACK 控制，可以通过将 I2C_CTL0 寄存器中 SBCTL 位置 1 来使能从机字节控制模式。当 SS=1 时，从机字节控制模式无效。

在使用从机字节控制模式时，必须通过置位 I2C_CTL1 寄存器中 RELOAD 位来使能重载模式。从机字节控制模式中，在 ADDSEND 中断服务程序中 I2C_CTL1 寄存器中 BYTENUM[7:0]必须配置为 1，并且在每个字节接收完成时重载为 1。当接收到一个字节时，I2C_STAT 寄存器中 TCR 位置 1，在第八个和第九个 SCL 时钟脉冲之间从机将 SCL 时钟拉低。然后数据可以

从 I2C_RDATA 寄存器中读取出来，通过配置 I2C_CTL1 寄存器中 NACKEN 位，从机可以决定发送 ACK 或者是 NACK。当在 BYTENUM[7:0]写入非零值时，从机释放 SCL 时钟线。

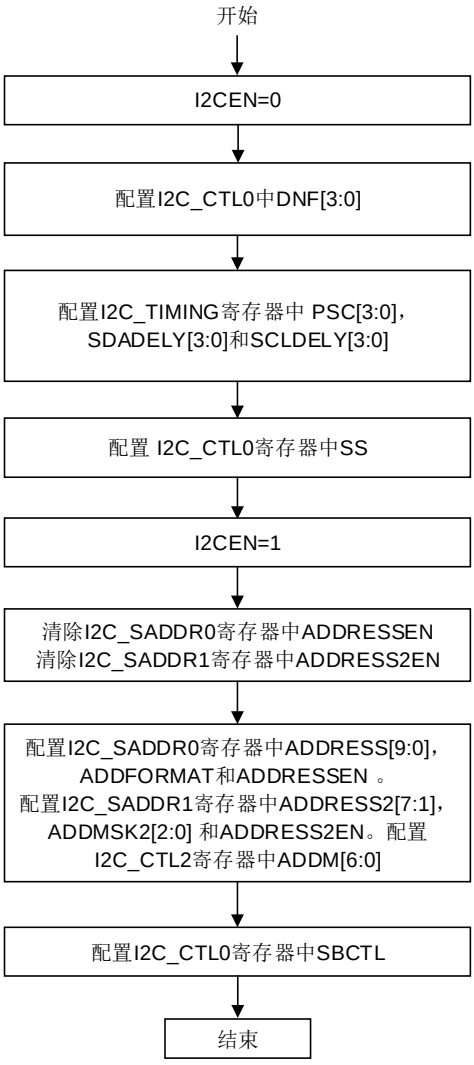
当 BYTENUM[7:0]大于 0x1 时，在 BYTENUM[7:0]数据接收期间，数据流是连续的。

注意：在下列情况下，可以配置 SBCTL 位：

1、 I2CEN=0。
2、 从机还未被寻址。
3、 ADDSEND=1。

当 ADDSEND=1，或者 TCR=1 时，RELOAD 才可以被修改。

图 21-26. I2C 从机初始化



从机发送模式下的软件流程

当 I2C_TDATA 寄存器为空，I2C_STAT 寄存器中 TI 位将会置位。如果 I2C_CTL0 寄存器中 TIE 位置 1，将产生中断。当接收到 NACK 时，I2C_STAT 寄存器中 NACK 位会置位。如果 I2C_CTL0 寄存器中 NACKIE 位置 1，将产生中断。当接收到 NACK 信号时，I2C_STAT 寄存器中 TI 位将不会置位。

当接收到 STOP 信号时，I2C_STAT 寄存器中 STPDET 位将置 1。如果 I2C_CTL0 寄存器中 STPDETIE 位置 1，将产生中断。

当 SBCTL=0 时，如果 ADDSEND=1，且 I2C_STAT 寄存器中 TBE 位为 0，可以选择发送 I2C_TDATA 寄存器中的数据或者是将 TBE 置 1 来清空 I2C_TDATA 寄存器。

当 SBCTL=1 时，从机工作在字节控制模式，BYTENUM[7:0]必须在 ADDSEND 中断服务程序中配置。TI 事件的数量与 BYTENUM[7:0]的值相等。

当 SS=1 时，I2C_STAT 寄存器中 ADDSEND 位置位时 SCL 时钟线不会被拉低。在这种情况下，I2C_TDATA 寄存器中数据不能在 ADDSEND 中断服务程序中清空。因此待发送的第一个字节应该在 ADDSEND 置位之前就被编程到 I2C_TDATA 寄存器。

- 该数据可以是上一次数据传输最后一次 TI 事件写入的数据。
- 如果该数据不是待发送数据，可通过将 TBE 位置 1 来刷新 I2C_TDATA 寄存器，从而编程新的数据。在数据发送开始时 STPDET 位必须为 0。否则 I2C_STAT 寄存器中 OUERR 位将置 1 并产生下溢错误。
- 从机发送模式下使用中断或者 DMA 时，如果需要一个 TI 事件，TI 位和 TBE 位都必须置 1。

图 21-27. I2C 从机发送编程模型（SS=0）

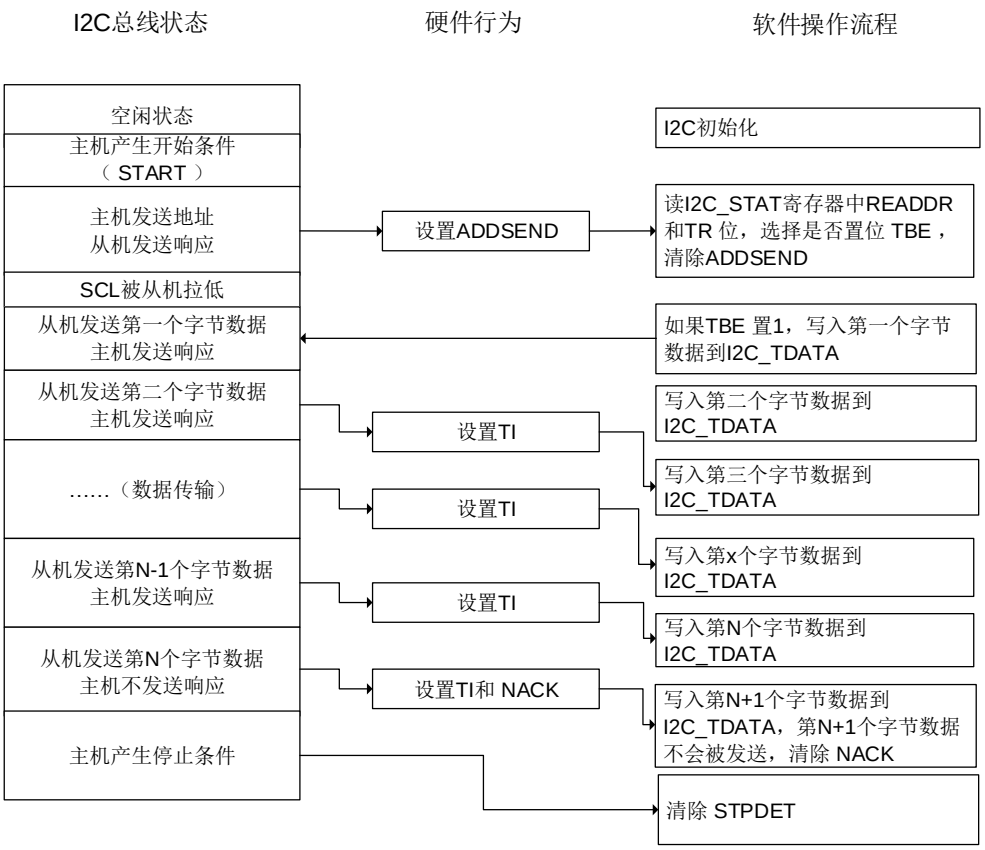
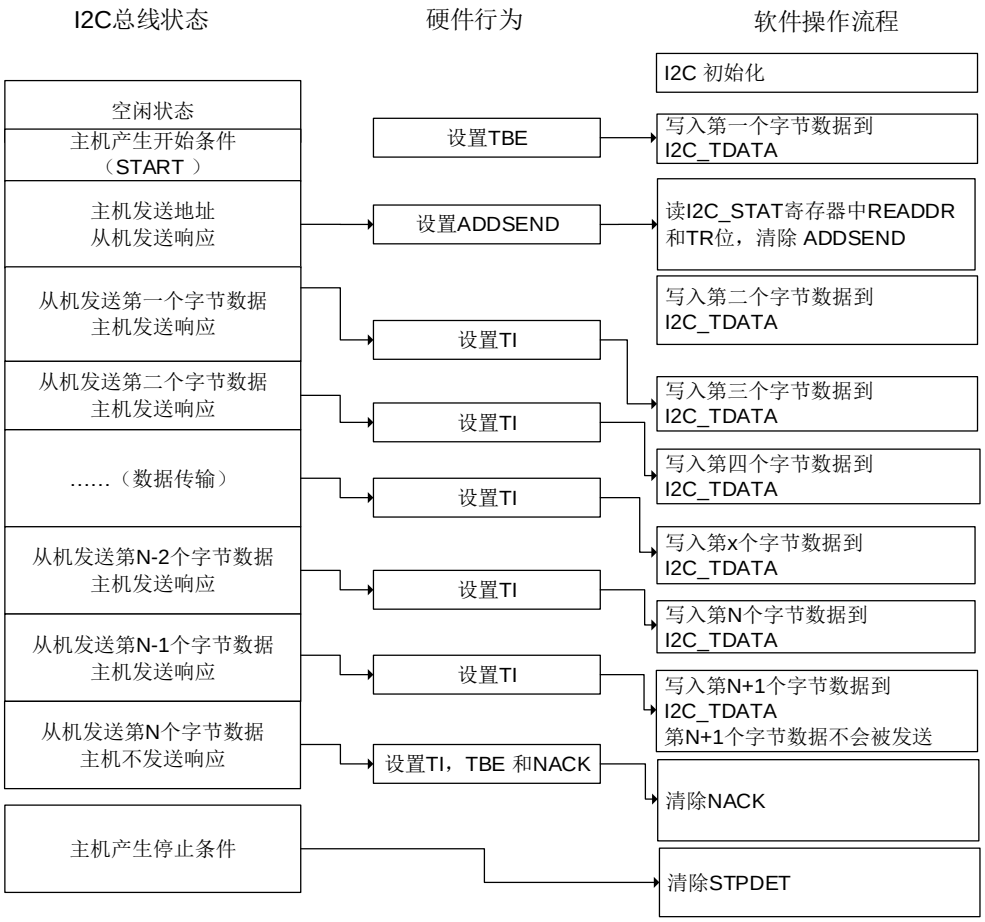


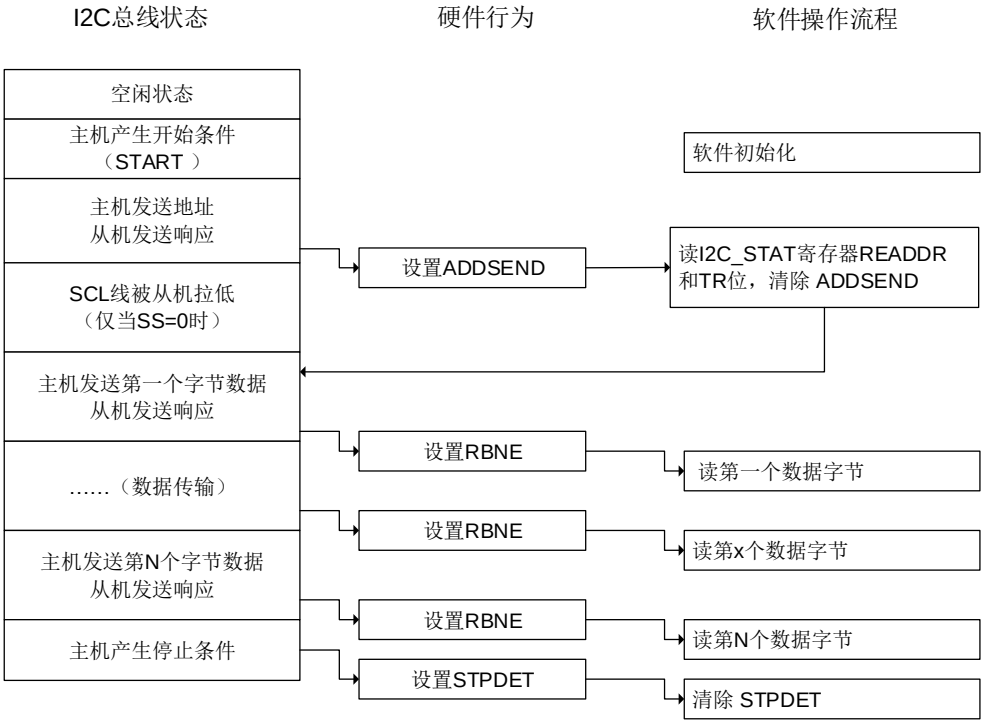
图 21-28. I2C 从机发送编程模型（SS=1）



从机接收模式下的软件流程

当 I2C_RDATA 寄存器非空，I2C_STAT 寄存器中 RBNE 位置 1，如果 I2C_CTL0 寄存器中 RBNEIE 位置 1，将产生中断。当接收到 STOP 信号时，I2C_STAT 寄存器中 STPDET 位将置 1。如果 I2C_CTL0 寄存器中 STPDETIE 置 1，将产生中断。

图 21-29. I2C 从机接收编程模型



I2C 主机模式

初始化

I2C_TIMING 寄存器中 SCLH[7:0]和 SCLL[7:0]必须在 I2CEN=0 时配置。为了支持多主机通信和从机时钟拉低，I2C 实现了时钟同步机制。

SCLL[7:0]和 SCLH[7:0]分别用于低电平计数和高电平计数。经过t_{SYNC1}延时后，当检测到 SCL 低电平时，SCLL[7:0]开始计数，如果 SCLL[7:0]计数器的值达到 I2C_TIMING 寄存器中 SCLL[7:0]时，I2C 将释放 SCL 时钟。经过t_{SYNC2}延时后，当检测到 SCL 高电平时，SCLH[7:0]开始计数，如果 SCLH[7:0]计数器的值达到 I2C_TIMING 寄存器中 SCLH[7:0]时，I2C 将拉低 SCL 时钟。

因此主机时钟周期为：$t_{SCL}=t_{SYNC1}+t_{SYNC2}+\{[(SCLH[7:0]+1)+(SLLL[7:0]+1)]*(PSC+1)*t_{I2CCLK}\}$。

t_{SYNC1}取决于 SCL 下降沿斜率，SCL 输入模拟和数字噪声滤波器延时以及 SCL 与 I2CCLK 时钟的同步产生的延时，一般为 2 到 3 个 I2CCLK 时钟周期。t_{SYNC2}取决于 SCL 上升沿斜率，SCL 输入模拟和数字噪声滤波器延时以及 SCL 与 I2CCLK 时钟的同步产生的延时，一般为 2 到 3 个 I2CCLK 时钟周期。数字噪声滤波器产生的延时为$DNF[3:0]*t_{I2CCLK}$。

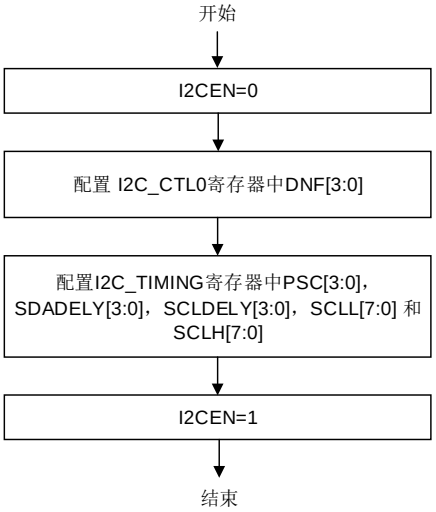
在主机模式下，必须配置 I2C_CTL1 寄存器中 ADD10EN，SADDRESS[9:0]以及 TRDIR 位。当在主机接收模式下使用 10 位寻址时，必须配置 HEAD10R 来选择是执行完整的地址寻址序列，还是只发送地址头。待传输的字节数在 I2C_CTL1 寄存器 BYTENUM[7:0]配置。如果待传输的字节数大于或者等于 255，必须将 BYTENUM[7:0]配置为 0xFF。然后主机发送 START 信号。以上提到的所有位必须在 START 位置 1 之前配置。START 信号发送完成之后，待 I2C_STAT 寄存器 I2CBSY 位为 0 时，发送从机地址。当仲裁丢失时，主机切换成从机模式，START 位由硬件清零。当从机地址发送完成时，START 位由硬件清零。

注意：BYTENUM[7:0]不能配置为 0。

在 10 位寻址模式下，在发送 10 位地址头之后，如果主机接收到 NACK，主机将重发 10 位地址头直到收到 ACK。将 ADDSENDC 置 1 可以停止重发从机地址。

如果 START 位置 1 时，I2C 作为从机被寻址成功，ADDSEND 置 1，主机将切换为从机模式。START 位将在 ADDSENDC 置 1 时清零。

图 21-30. I2C 主机初始化



主机发送模式下的软件流程

在主机发送模式下，每一个字节发送完成并接收到 ACK 信号之后，TI 位将置 1。如果 I2C_CTL0 寄存器中 TIE 位置 1，将产生中断。待发送的字节数编程在 I2C_CTL1 寄存器 BYTENUM[7:0]。如果发送字节数大于 255，必须通过将 I2C_CTL1 寄存器 RELOAD 位置 1 来使能重载模式。在重载模式下，当 BYTENUM[7:0]个字节传输完成，I2C_STAT 寄存器 TCR 位将置 1，并且在 BYTENUM[7:0]更新一个非零值之前，SCL 被拉低。

如果接收到 NACK，TI 位将不会置 1。

- 如果 BYTENUM[7:0]个字节传输完成且 RELOAD=0，将 I2C_CTL1 寄存器中 AUTOEND 置 1 可以自动产生 STOP 信号。当 AUTOEND=0 时，I2C_STAT 寄存器 TC 位将置 1 且 SCL 被拉低。在这种情况下，主机可以通过将 I2C_CTL1 寄存器中 STOP 位置 1 来产生 STOP 信号。或者产生 RESTART 信号来开始一个新的数据传输过程。将 START / STOP 置 1 可以清除 TC 位。
- 如果接收到 NACK 信号，I2C 将自动产生 STOP 信号。I2C_CTL0 寄存器中 NACK 将置 1，如果 NACKIE 位置 1，将产生中断。

注意：当 RELOAD=1 时，AUTOEND 位无效。

图 21-31. I2C 主机发送编程模型（N<=255）

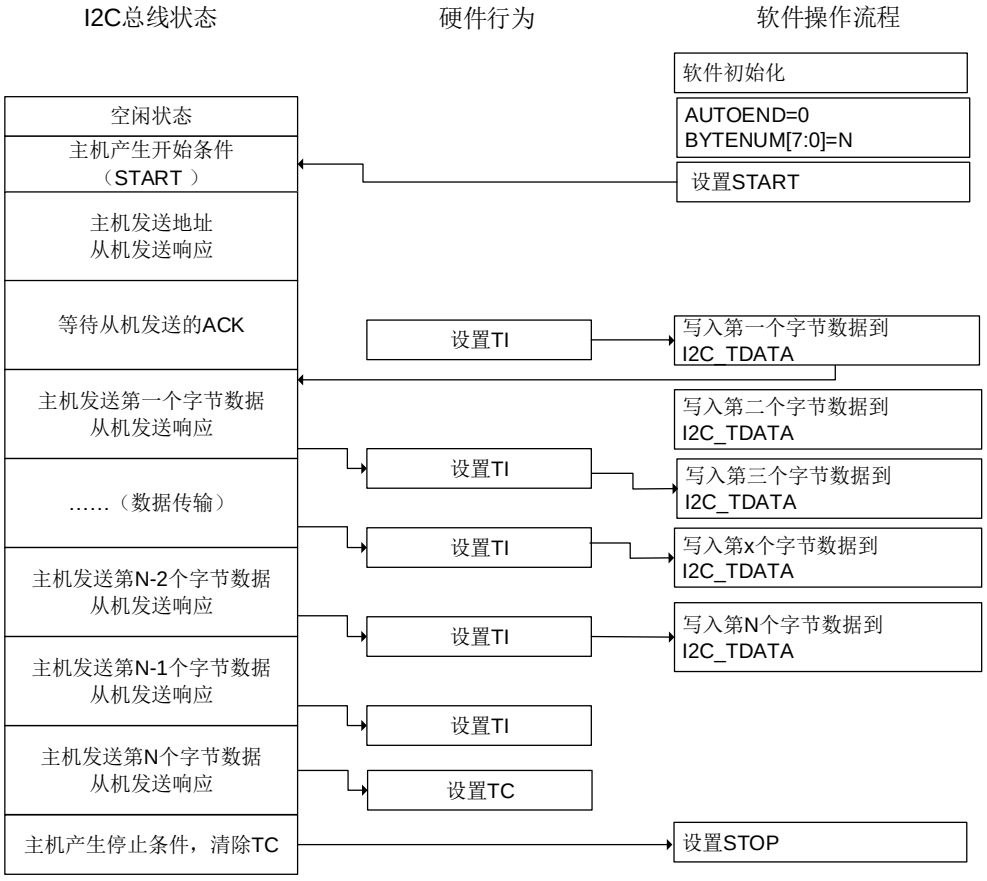
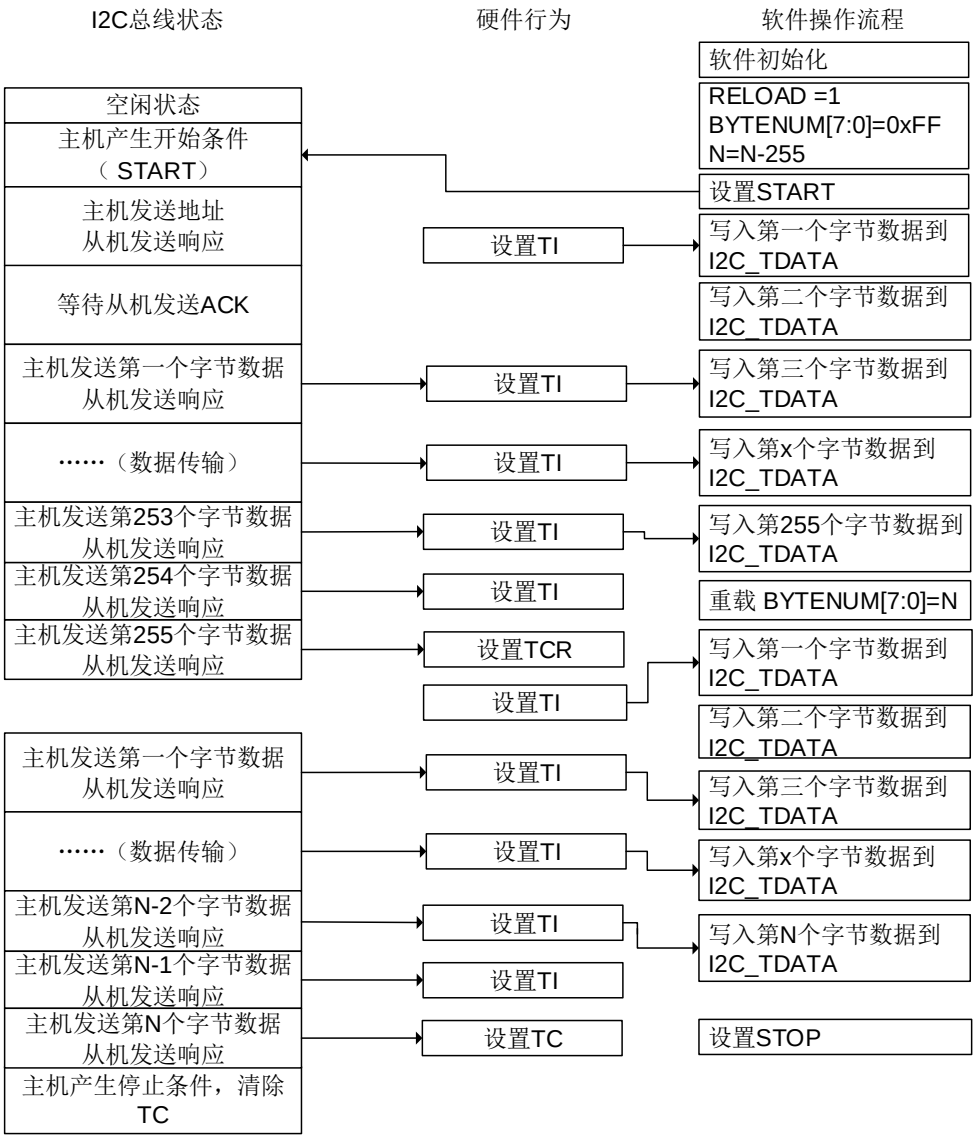


图 21-32. I2C 主机发送编程模型（N>255）



主机接收模式下的软件流程

在主机接收模式下，当接收到一个字节时，I2C_STAT 寄存器中 RBNE 位置 1。如果 I2C_CTL0 寄存器中 RBNEIE 置 1，将产生一个中断。如果待接收字节数大于 255，必须将 I2C_CTL1 寄存器中 RELOAD 位置 1 来使能重载模式。在重载模式下，当 BYTENUM[7:0]个字节传输完成，I2C_STAT 寄存器中 TCR 位将置 1，在 BYTENUM[7:0]中写入一个非零值之前，SCL 被拉低。

如果 BYTENUM[7:0]个字节传输完成且 RELOAD=0，将 I2C_CTL1 寄存器中 AUTOEND 置 1 可以自动产生 STOP 信号。当 AUTOEND=0 时，I2C_STAT 寄存器 TC 位将置 1 且 SCL 被拉低。在这种情况下，主机可以通过将 I2C_CTL1 寄存器中 STOP 位置 1 来产生 STOP 信号。或者产生 RESTART 信号来开始一个新的数据传输过程。将 START / STOP 置 1 可以清除 TC 位。

图 21-33. I2C 主机接收编程模型（N<=255）

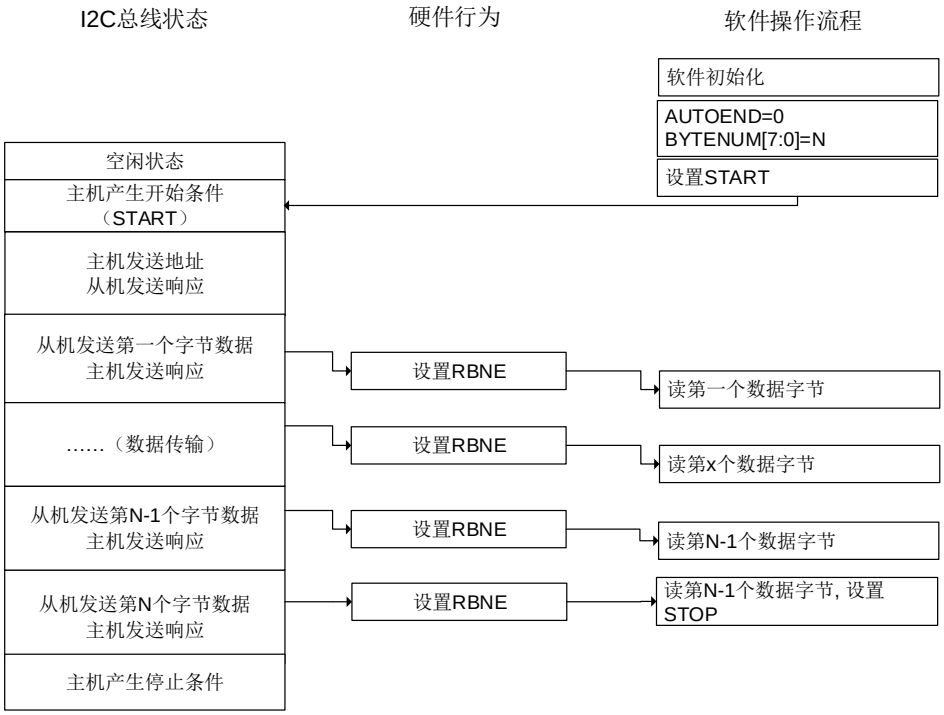
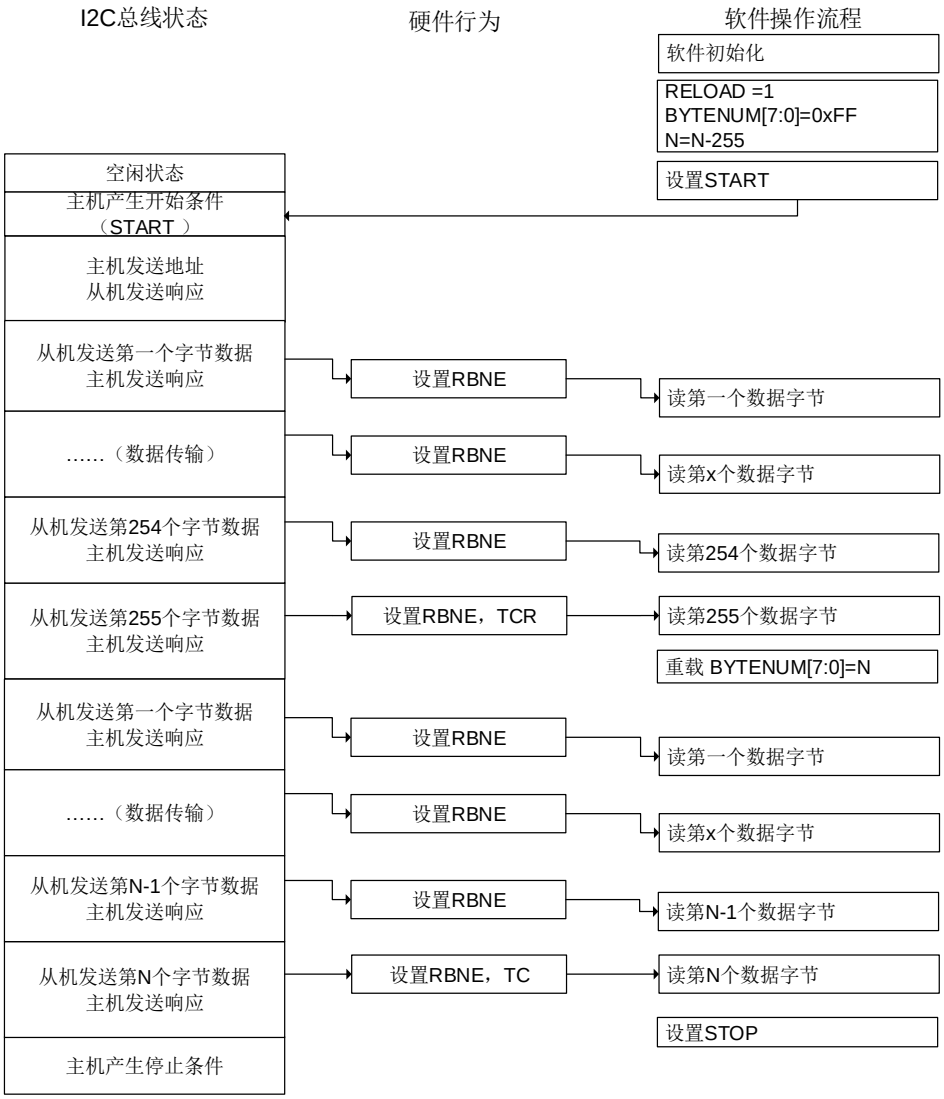


图 21-34. I2C 主机接收编程模型（N>255）



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 采用了 I2C 硬件以及 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界面。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。

SMBus 从机字节控制

SMBus 接收器从机字节控制与 I2C 一样。它允许 ACK 控制每个字节。必须能对接收到的命令或者数据进行 NACK 应答。通过将 I2C_CTL0 寄存器中 SBCTL 位置 1 来使能从机字节控制模式。

主机通知协议

通过将 I2C_CTL0 寄存器 SMBHAEN 位置 1，SMBus 可以支持主机通知协议。在该协议中，从设备作为主机，主设备作为从机，主机将应答 SMBus 主机地址。

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 25~35ms。按照 SMBus 协议的假定，如果某个会话耗时太久，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）状态。这样就并不允许从设备将时钟拉低太长时间。

将 I2C_TIMEOUT 寄存器中 TOEN 位和 EXTOEN 位置 1 可以使能超时检测。配置定时器必须保证在 SMBus 规范规定的时间最大值之前检测出超时情况。

在 BUSTOA[11:0]中编程的值被用来检查$t_{TIMEOUT}$参数。必须将 TOIDLE 位配置为 0，以检测 SCL 低电平超时。将 I2C_TIMEOUT 寄存器中 TOEN 位置 1 来使能定时器，在 TOEN 置 1 之后，BUSTOA[11:0] 和 TOIDLE 位不能被修改。如果 SCL 低电平时间大于(BUSTOA+1)*2048*t_{I2CCLK}，I2C_STAT 寄存器中 TIMEOUT 位将置 1。

BUSTOA[11:0]为从机校验$t_{LOW:SEXT}$，为主机校验$t_{LOW:MEXT}$。通过将 I2C_TIMEOUT 寄存器中 EXTOEN 位置 1 来使能定时器。在 EXTOEN 置 1 之后，BUSTOB[11:0]不能被修改。如果 SMBus 外设 SCL 拉低时间大于(BUSTOB+1)*2048*t_{I2CCLK}，并且达到了总线空闲检测章节中描述的超时时间间隔，I2C_STAT 寄存器中 TIMEOUT 位将置 1。

报文错误校验

I2C 模块中有一个 PEC 模块，它使用 CRC-8 计算器来执行 I2C 数据的报文校验。一个 PEC 字节（PEC 错误码）附加在每次传输结束。PEC 的计算方式是对所有消息字节（包含地址和读/写位）使用 CRC-8 计算校验和。CRC-8 多项式位x^8+x^2+x+1（CRC-8-ATM HEC 算法，初始化为 0）。

当 I2C 被禁用时，通过 I2C_CTL0 寄存器中的 PECEN 位置 1 可以使能 PEC。由于 PEC 传输是由 I2C_CTL1 寄存器中 BYTENUM[7:0]管理的，因此在从机模式下必须将 SBCTL 位置 1。当 PECTRANS 置 1，RELOAD 为 0 时，在 BYTENUM[7:0]-1 数据字节后发送 PEC。PEC 在

BYTENUM[7:0]-1 传输完成后发送。当 RELOAD 置 1 时 PECTRANS 无效。

SMBus 警报

SMBus 还有一个额外的共享的中断信号，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。主机会处理该中断，并通过报警响应地址，同时访问所有 SMBALERT#设备。如果 SMBALERT#电平被设备拉低，这些设备会应答报警响应地址。当配置为从设备（SMBHAEN=0）时，通过将 I2C_CTL0 寄存器中 SMBALTEN 置 1 可以将 SMBA 引脚电平拉低。同时也使能了报警响应地址。当配置为主设备（SMBHAEN=1），且 SMBALTEN 置 1 时，当在 SMBA 引脚检测到下降沿时，I2C_STAT 寄存器中 SMBALT 位将置 1。如果 I2C_CTL0 寄存器中 ERRIE 位置 1，将产生中断。当 SMBALTEN=0 时，即使外部 SMBA 引脚为低电平，ALERT 线也将被视为高电平。当 SMBALTEN=0 时，SMBA 引脚可用作标准 GPIO。

总线空闲检测

如果主机检测到时钟信号和数据信号的高电平持续时间大于$t_{HIGH,MAX}$，总线被视为空闲。

该时序参数已考虑到主机已动态添加至总线，但可能还未检测到 SMBCLK 或 SMBDAT 线上的状态转换的情况。在这种情况下，为了保证当前没有数据传输正在进行，主机必须等待足够长的时间。

要使能t_{IDLE}检查，必须将 BUSTOA[11:0]编程为定时器重载值，以获取t_{IDLE}参数。必须将 TIDLE 位置 1，以检测 SCL 和 SDA 高电平超时。然后通过将 I2C_TIMEOUT 寄存器中的 TOEN 位置 1 来使能定时器。TOEN 置 1 后，BUSTOA[11:0]和 TIDLE 不能被修改。如果 SCL 和 SDA 的高电平持续时间都大于(BUSTOA+1)*4*t_{I2CCLK}，I2C_STAT 寄存器中 TIMEOUT 位将置位。

SMBus 从机模式

SMBus 接收器必须能够对接收到的命令和数据进行 NACK 应答。对于从机模式下的 ACK 控制，通过将 I2C_CTL0 寄存器中 SBCTL 位置 1 可以使能从机字节控制模式。

必要时应使能特定的 SMBus 地址。通过将 I2C_CTL0 寄存器中 SMBDAEN 置 1 可以使能 SMBus 设备默认地址（0b1100 001）。通过将 I2C_CTL0 寄存器中 SMBHAEN 置 1 可以使能 SMBus 主机地址（0b0001 000）。通过将 I2C_CTL0 寄存器中 SMBALTEN 置 1 可以使能报警响应地址（0b0001 100）。

SMBus 模式

SMBus 主机发送器和从机接收器

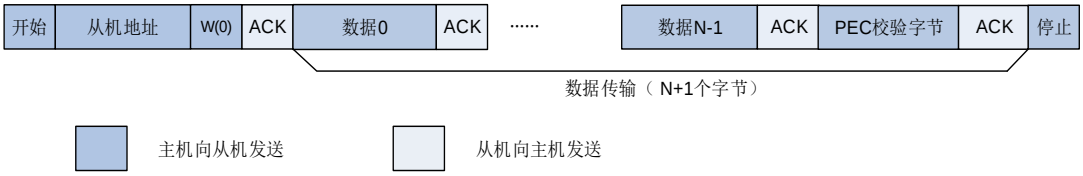
当 SMBus 主机发送 PEC 时，必须在 START 位置 1 前，将 PECTRANS 位置 1 并在 BYTENUM[7:0]位域中配置字节数。在这种情况下，总 TI 中断数为 BYTENUM-1。因此，如果 BYTENUM=0x1 且 PECTRANS 位置 1，则 I2C_PEC 寄存器的数据将自动发送。如果 AUTOEND 为 1，SMBus 主机在 PEC 字节发送完成之后将自动发送 STOP 信号。如果 AUTOEND 为 0，SMBus 主机可以在 PEC 字节发送完成之后发送 RESTART 信号。I2C_PEC 寄存器中的数据将在 BYTENUM -1 个字节发送完成后发送，PEC 字节发送完成后 TC 位将置 1。SCL 线被拉低。RESTART 位必须在 TC 中断服务程序中置 1。

SMBus 作为从机接收器时，为了在数据发送完成时进行 PEC 校验，SBCTL 位必须置 1。要对每个字节进行 ACK 控制，必须通过将 RELOAD 位置 1 来使能 RELOAD 模式。如果要校验

PEC 字节，必须将 RELOAD 位清零同时将 PECTRANS 置 1。在 BYTENUM-1 个字节接收完成后，接收的下一个字节将与 I2C_PEC 寄存器中的数据进行比较。如果校验值不匹配，将自动产生 NACK 信号；如果校验值匹配将自动产生 ACK 信号，将忽略 NACKEN 位的值。当接收到 PEC 字节时，PEC 字节会存到 I2C_RDATA 寄存器中，RBNE 位将置 1。如果 I2C_CTL0 寄存器中 ERRIE 位置 1，且 PEC 值不匹配，PECERR 将会置 1 并产生中断。如果无须使用 ACK 控制，PECTRANS 可以设置为 1，BYTENUM 可以根据待接收字节数来配置。

注意：在 RELAOD 位置 1 之后，PECTRANS 不可以被修改。

图 21-35. SMBus 主机发送器和从机接收器通信流程



SMBus 主机接收器和从机发送器

如果 SMBus 主机需要在数据传输完成后接收 PEC 字节，可以使能自动结束模式。在 START 信号发送之前，必须将 PECTRANS 位置 1，且配置好从机地址。在接收 BYTENUM-1 数据之后，接收的下一个字节将自动与 I2C_PEC 寄存器中的数据进行比较。在停止信号发送之前，接收 PEC 字节之后会给出 NACK 响应。

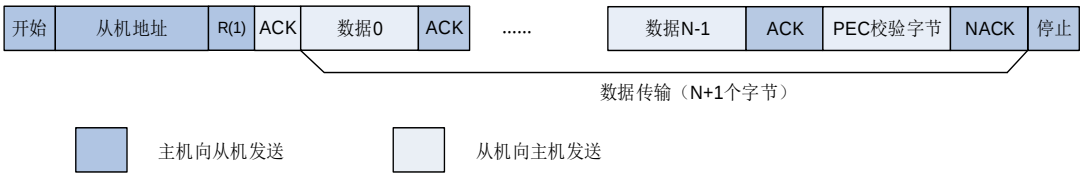
如果 SMBus 主机需要在接收到 PEC 字节之后产生 RESTART 信号，需要禁能自动结束模式。在 START 信号发送之前，PECTRANS 位必须置 1，且配置好从机地址。在接收 BYTENUM-1 数据之后，接收的下一个字节将自动与 I2C_PEC 寄存器中的数据进行比较。在 PEC 字节发送完成之后 TC 位将置 1，SCL 线被拉低。在 TC 中断服务程序中可将 RESTART 位置 1。

当 SMBus 作为从机发送器时，为了在 BYTENUM[7:0]个字节发送完成之后发送 PEC 字节，SBCTL 位必须置 1。如果 PECTRANS 置 1，字节数 BYTENUM[7:0]包含 PEC 字节。在这种情况下，如果主机请求接收的字节数大于 BYTENUM-1，总 TI 中断数为 BYTENUM-1，I2C_PEC 寄存器中的数据将自动发送。

注意：

1. PECTRANS 位在 RELOAD 置 1 之后不能被修改。
2. SMBus 作为从机发送器，并且使用 PEC 功能时，必须在检测到 START 信号前配置 BYTENUM[7:0]。

图 21-36. SMBus 主机接收器和从机发送器通信流程



从省电模式唤醒

当 I2C 地址匹配成功时，MCU 从深度睡眠模式，深度睡眠模式 1 和深度睡眠模式 2 被唤醒。为了将 MCU 从这些省电模式唤醒，I2C_CTL0 寄存器中 WUEN 位必须置 1，同时 I2CCLK 时钟源选择 IRC8M。在深度睡眠模式，深度睡眠模式 1 和深度睡眠模式 2 下，IRC8M 关闭。当 I2C 检测到 START 信号时，IRC8M 打开，I2C 会将 SCL 拉低直到 IRC8M 被唤醒。在接收地址期间，IRC8M 为 I2C 提供时钟。当地址匹配时，在 MCU 唤醒期间，I2C 的 SCL 线被拉低。当 ADDSEND 清除时，SCL 线被释放，数据传输过程恢复正常。如果检测到的地址不匹配，IRC8M 会再次关闭，MCU 将不会被唤醒。

只有地址匹配中断（ADDMIE=1）能唤醒 MCU。如果 I2C 的时钟源是系统时钟，或者 WUEN=0，IRC8M 在接收到 START 信号之后将不会打开。当从省电模式唤醒使能时，数字滤波器必须禁能，I2C_CTL0 寄存器中 SS 位也必须清 0。如果禁止从省电模式唤醒（WUEN=0），则在进入省电模式之前必须禁能 I2C 外设（I2CEN=0）。

DMA 模式下数据传输

如 I2C 从机模式和主机模式中描述，每当 TI 位和 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 TI 或 RBNE 位置 1 时，自动进行一次写或读操作。

将 I2C_CTL0 寄存器中 DENT 置 1 可以使能 DMA 发送请求。将 I2C_CTL0 寄存器中 DENR 置 1 可以使能 DMA 接收请求。在主机模式下，由软件写入从机地址，传输方向，待发送字节数和 START 位。DMA 必须在 START 位置 1 之前初始化。在 I2C_CTL1 寄存器 BYTENUM[7:0] 位配置待传输字节数。在从机模式下，DMA 必须在地址匹配事件发生之前或 ADDSEND 中断服务程序中清除 ADDSEND 标志之前完成初始化。

I2C 错误和中断

I2C 错误标志如*表 21-7. I2C 错误标志*所示。

表 21-7. I2C 错误标志

I2C 错误名称	描述
BERR	总线错误
LOSTARB	仲裁丢失
OUERR	上溢 / 下溢标志
PECERR	CRC 值不匹配
TIMEOUT	SMBus 模式下总线超时
SMBALT	SMBus 报警

I2C 中断和事件标志如*表 21-8. I2C 中断事件*所示。

表 21-8. I2C 中断事件

中断事件	事件标志	使能控制位
在接收期间 I2C_RDATA 非空	RBNE	RBNEIE
发送中断	TI	TIE
从机模式下检测到 STOP 信号	STPDET	STPDETIE

中断事件	事件标志	使能控制位
传输完成重载	TCR	TCIE
传输完成	TC	
地址匹配	ADDSEND	ADDMIE
接收到 NACK	NACK	NACKIE
总线错误	BERR	ERRIE
仲裁丢失	LOSTARB	
上溢 / 下溢错误	OUERR	
PEC 错误	PECERR	
超时错误	TIMEOUT	
SMBus 报警	SMBALT	

I2C 调试模式

当为控制器进入调试模式（Cortex®-M33 内核停止），SMBus 超时定时器会根据 DBG 模块中的 I2Cx_HOLD 配置位选择继续正常工作还是停止工作。

21.2.4. I2C 寄存器

I2C2 基地址：0x4000 C000

控制寄存器 0（I2C_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								PECEN	SMBALTEN	SMBDAEN	SMBHAEN	GCEN	WUEN	SS	SBCTL
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DENR	DENT	保留	ANOFF	DNF[3:0]				ERRIE	TCIE	STPDETIE	NACKIE	ADDMIE	RBNEIE	TIE	I2CEN
rw	rw		rw	rw				rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:24	保留	必须保持复位值。
23	PECEN	PEC 计算开关。 0：PEC 计算关闭。 1：PEC 计算打开。
22	SMBALTEN	SMBus 报警使能。 0：从机模式下 SMBA 引脚高电平或主机模式下 SMBus 报警引脚 SMBA 禁能。 1：从机模式下 SMBA 引脚低电平或主机模式下 SMBus 报警引脚 SMBA 使能。
21	SMBDAEN	SMBus 设备默认地址使能。 0：设备默认地址禁能，对默认地址 0b1100001x 进行 NACK 应答。 1：设备默认地址使能，对默认地址 0b1100001x 进行 ACK 应答。
20	SMBHAEN	SMBus 主机地址使能。 0：主机地址禁能，对地址 0b0001000x 进行 NACK 应答。 1：主机地址使能，对地址 0b0001000x 进行 ACK 应答。
19	GCEN	是否响应对地址（0x00）的广播呼叫。 0：从机不响应广播呼叫。 1：从机将响应广播呼叫。
18	WUEN	使能从省电模式中唤醒，包含深度睡眠模式，深度睡眠模式 1 和深度睡眠模式 2。 0：禁止从省电模式中唤醒。 1：使能从省电模式中唤醒。 **注意：**当 DNF[3:0] = 0 时，WUEN 才能被置 1。
17	SS	在从机模式下数据未就绪时是否将 SCL 拉低。

		软件置 1 和清 0。 0：拉低 SCL 1：不拉低 SCL **注意：**在主机模式下，该位必须为 0。该位只能在 I2CEN=0 时被修改。
16	SBCTL	从机模式下字节控制。 该位用于在从机模式下使能硬件字节控制。 0：从机模式下字节控制禁能。 1：从机模式下字节控制使能。
15	DENR	DMA 接收使能 0：DMA 接收禁能 1：DMA 接收使能
14	DENT	DMA 发送使能 0：DMA 发送禁能 1：DMA 发送使能
13	保留	必须保持复位值。
12	ANOFF	模拟噪声滤波器禁能 0：模拟噪声滤波器使能。 1：模拟噪声滤波器禁能。 **注意：**该位只有在 I2C 禁能（I2CEN=0）时被编程。
11:8	DNF[3:0]	数字噪声滤波器 0000：数字噪声滤波器禁能。 0001：数字噪声滤波使能并且可以滤除脉宽宽度不大于 1 t_{I2CCLK}的尖峰。 ... 1111：数字噪声滤波使能并且可以滤除脉宽宽度不大于 15 t_{I2CCLK}的尖峰。 这些位只能在 I2C 禁能（I2CEN = 0）时修改。
7	ERRIE	错误中断使能 0：错误中断禁能 1：错误中断使能，当 BERR，LOSTARB，OUERR，PECERR，TIMEOUT 或 SMBALT 位置 1 时，将产生中断。
6	TCIE	传输完成中断使能 0：传输完成中断禁能。 1：传输完成中断使能。
5	STPDETIE	停止信号检测中断使能 0：停止信号（STPDET）检测中断禁能。 1：停止信号（STPDET）检测中断使能。
4	NACKIE	接收到 NACK 应答中断使能 0：接收到 NACK 应答中断禁能。 1：接收到 NACK 应答中断使能。
3	ADDMIE	从机模式下地址匹配中断使能

0：地址匹配中断禁能。
1：地址匹配中断使能。

2 RBNEIE 接收中断使能
0：接收（RBNE）中断禁能。
1：接收（RBNE）中断使能。

1 TIE 发送中断使能
0：发送中断（TI）禁能。
1：发送中断（TI）使能。

0 I2CEN I2C 外设使能
0：I2C 禁能。
1：I2C 使能。

控制寄存器 1（I2C_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					PECTRANS	AUTOEND	RELOAD	BYTENUM[7:0]							
					rw	rw	rw	rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NACKEN	STOP	START	HEAD10R	ADD10EN	TRDIR	SADDRESS[9:0]									
rw	rw	rw	rw	rw	rw	rw									

位/位域	名称	描述
31:27	保留	必须保持复位值。
26	PECTRANS	PEC 传输 软件置 1 和清 0，硬件在以下条件下清除此位：PEC 传输完成或者 ADDSEND 置 1 或者检测到 STOP 信号或者 I2CEN=0。 0：不传输 PEC 值。 1：传输的 PEC 值。 **注意：**当 RELOAD=1 或者从机模式下 SBCTL=0 时，该位无效。
25	AUTOEND	主机模式下自动结束模式 0：当 BYTENUM[7:0]个字节传输完成后时，TC 位置 1。 1：当 BYTENUM[7:0]个字节传输完成后时，自动发送 STOP 信号。 **注意：**该位仅在 RELOAD=0 时有效。该位由软件置 1 和清 0。
24	RELOAD	重载模式使能 0：当 BYTENUM[7:0]个字节传输完成后时，传输结束。

		1：当 BYTENUM[7:0]个字节传输完成后时，传输未结束，重载新的 BYTENUM[7:0]。每次 BYTENUM[7:0]个字节传输完成，I2C_STAT 寄存器中 TCR 位将置 1。该位由软件置 1 和清 0。
23:16	BYTENUM[7:0]	待传输的字节数 这些用来编程待传输的字节数。当 SBCTL=0 时，这些位无效。 **注意：**当 START 位置 1 时，该位不能被修改，传输过程中软件不能修改该位。
15	NACKEN	从机模式下产生 NACK 0：在接收到新的字节时，发送 ACK。 1：在接收到新的字节时，发送 NACK。 **注意：**该位可由软件置 1，并在以下情况下由硬件清零：NACK 发送完成或检测到 STOP 信号或 ADDSEND 置 1，或 I2CEN=0。当 PEC 使能时，发送 ACK 还是 NACK 与 NACKEN 值无关。当 SS=1 时，且 OUERR 位置 1，NACKEN 的值会被忽略，并且发送 NACK。
14	STOP	I2C 总线上产生一个 STOP 结束信号。 该位由软件置 1，并在 I2CEN=0 或检测到 STOP 信号时由硬件清零。 0：不发送 STOP。 1：发送 STOP。
13	START	I2C 总线上产生一个 START 信号 该位由软件置 1，并在从机地址发送后由硬件清零。当仲裁丢失时，或发生超时错误，或 I2CEN=0 时，该位也可以由硬件清零。将 I2C_STATC 寄存器中 ADDSENDC 位置 1 可以软件清除该位。 0：不发送 START。 1：发送 START。
12	HEAD10R	在主机接收模式下仅执行 10 位地址头读操作。 0：主机发送 10 位从机地址读序列为 START + 10 位地址头（写） + 第二个地址字节 + RESTART + 10 位地址头（读）。 1：主机寻址读序列为 RESTART + 10 位地址头（读）。 **注意：**当 START 位置 1 时，该位不能被修改，传输过程中软件不能修改该位。
11	ADD10EN	主机模式下使能 10 位寻址模式 0：主机工作在 7 位寻址模式下。 1：主机工作在 10 位寻址模式下。 **注意：**当 START 位置 1 时，该位不能被修改，传输过程中软件不能修改该位。
10	TRDIR	主机模式下传输方向 0：主机发送 1：主机接收 **注意：**当 START 位置 1 时，该位不能被修改，传输过程中软件不能修改该位。
9:0	SADDRESS[9:0]	待发送的从机地址 SADDRESS[9:8]：从机地址 9:8 位。 如果 ADD10EN = 0，该位域无效。 如果 ADD10EN = 1，将该位域写入待发送从机地址的 9:8 位。

SADDRESS[7:1]：从机地址 7:1 位。

如果 ADD10EN = 0，在这些位写入待发送 7 位从机地址。

如果 ADD10EN = 1，在这些位写入待发送从机地址的 7:1 位。

SADDRESS0：从机地址 0 位。

如果 ADD10EN = 0，这些位无效。

如果 ADD10EN = 1，在这些位写入待发送从机地址的 0 位。

注意：当 START 位置 1 时，该位不能被修改，传输过程中软件不能修改该位。

从机地址寄存器 0（I2C_SADDR0）

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															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESSEN	保留				ADDFORMAT	ADDRESS[9:8]		ADDRESS[7:1]							ADDRESS0
rw					rw	rw		rw							rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	ADDRESSEN	I2C 地址使能 0：I2C 地址禁能。 1：I2C 地址使能。
14:11	保留	必须保持复位值。
10	ADDFORMAT	I2C 从机地址模式 0：7 位地址。 1：10 位地址。 **注意：**当 ADDRESSEN =1 时，该位不能被改写。
9:8	ADDRESS[9:8]	10 位地址的最高两位 **注意：**当 ADDRESSEN =1 时，该位不能被改写。
7:1	ADDRESS[7:1]	7 位地址或者 10 位地址的第 7-1 位 **注意：** 1. 当 ADDRESSEN =1 时，该位不能被改写。 2. 当从机地址设置为保留地址 0b1111000 时，如果地址匹配，从机不响应。
0	ADDRESS0	10 位地址的第 0 位 **注意：**当 ADDRESSEN =1 时，该位不能被改写。

从机地址寄存器 1（I2C_SADDR1）

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
31:16	保留	必须保持复位值。
15	ADDRESS2EN	I2C 第二个地址使能 0：I2C 第二个地址禁能 1：I2C 第二个地址使能
14:11	保留	必须保持复位值。
10:8	ADDMSK2[2:0]	ADDRESS2[7:1]掩码 定义接收到的地址哪些位需要与 ADDRESS2[7:1]进行比较，哪些位屏蔽（不比较）。 000：不屏蔽，所有的位都进行比较。 n（001~110）：ADDRESS2[n:1]屏蔽。ADDRESS2[7:n+1]需要进行比较。 111：ADDRESS2[7:1]屏蔽。对于接收到的所有 7 位地址都会进行 ACK 应答，保留地址（0b0000xxx 和 0b1111xxx）除外。 **注意：**当 ADDRESS2EN =1 时，该位不能被改写。如果 ADDMSK2 不等于 0，即使所有位都匹配，I2C 保留地址（0b0000xxx 和 0b1111xxx）也不会进行 ACK 应答。
7:1	ADDRESS2[7:1]	I2C 从机的第二个地址 **注意：** 1. 当 ADDRESS2EN =1 时，该位不能被改写。 2. 当从机地址设置为保留地址 0b1111000 时，如果地址匹配，从机不响应。
0	保留	必须保持复位值。

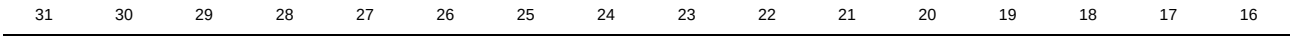
时序寄存器（I2C_TIMING）

地址偏移：0x10

复位值：0x0000 0000

当 I2C（I2CEN=1）被使能时，该寄存器不能被改写。

该寄存器只能按字（32 位）访问。



PSC[3:0]	保留	SCLDELY[3:0]	SDADELY[3:0]
rw		rw	rw

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

位/位域	名称	描述
31:28	PSC[3:0]	时序预分频 为了生成用于数据建立和数据保持的计数器的时钟周期t_{PSC}，这些位用于配置 I2CCLK 时钟预分频。t_{PSC}也用于 SCL 高电平和低电平计数器。 $t_{PSC}=(PSC+1)*t_{I2CCLK}$。
27:24	保留	必须保持复位值。
23:20	SCLDELY[3:0]	数据建立时间 这些位用于在 SDA 边沿和 SCL 上升沿之间生成延时$t_{SCLDELY}$。在主机模式下和在从机模式下 SS=0 时，在$t_{SCLDELY}$期间 SCL 线被拉低。 $t_{SCLDELY}=(SCLDELY+1)*t_{PSC}$。
19:16	SDADELY[3:0]	数据保持时间 这些位用于在 SCL 下降沿和 SDA 边沿之间生成延时$t_{SDADELY}$。在主机模式下和在从机模式下 SS=0 时，在$t_{SDADELY}$期间 SCL 线被拉低。 0x0：$t_{SDADELY}=t_{PSC}$ 0x1-0xF：$t_{SDADELY}=SDADELY*t_{PSC}$
15:8	SCLH[7:0]	SCL 高电平周期 SCL 高电平周期可以通过配置这些位来产生。 $t_{SCLH}=(SCLH+1)*t_{PSC}$。 **注意：**这些位只能用于主机模式。
7:0	SCLL[7:0]	SCL 低电平周期 SCL 低电平周期可以通过配置这些位来产生。 $t_{SCLL}=(SCLL+1)*t_{PSC}$。 **注意：**这些位只能用于主机模式。

超时寄存器（I2C_TIMEOUT）

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
EXTOEN	保留			BUSTOB[11:0]											
rw				rw											

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

位/位域	名称	描述
31	EXTOEN	时钟信号延展超时使能 当 SCL 累计拉低时间大于 $t_{LOW:EXT}$ 时，将会产生超时错误，$t_{LOW:EXT}$=(BUSTOB+1)*2048*t_{I2CCLK}。 0：时钟信号延展超时检测禁能。 1：时钟信号延展超时检测使能。
30:28	保留	必须保持复位值。
27:16	BUSTOB[11:0]	总线超时 B 配置累积时钟延展超时。在主机模式下，检测主机累计时钟低电平延展时间 $t_{LOW:MEXT}$。从机模式下，检测从机累计时钟低电平延展时间$t_{LOW:SEXT}$。 $t_{LOW:EXT}$=(BUSTOB+1)*2048*t_{I2CCLK}。 **注意：**该位域仅在 EXTOEN = 0 时可以被修改。
15	TOEN	时钟超时使能 当 TOIDLE = 0，SCL 拉低时间大于$t_{TIMEOUT}$或当 TOIDLE = 1，SCL 拉低时间大于 t_{IDLE}，将检测到超时错误。 0：SCL 超时检测禁能 1：SCL 超时检测使能
14:13	保留	必须保持复位值。
12	TOIDLE	空闲时钟超时检测 0：BUSTOA 用于检测 SCL 低电平超时。 1：BUSTOA 用于检测 SCL 和 SDA 高电平超时（总线空闲条件）。 **注意：**该位域仅在 TOEN = 0 时可以被改写。
11:0	BUSTOA[11:0]	总线超时 A 当 TOIDLE=0 时，$t_{TIMEOUT}$=(BUSTOA+1)*2048*t_{I2CCLK} 当 TOIDLE=1 时，t_{IDLE}=(BUSTOA+1)*4*t_{I2CCLK} **注意：**该位域仅在 TOEN = 0 时可以被改写。

状态寄存器（I2C_STAT）

地址偏移：0x18
复位值：0x0000 0001

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								READDR[6:0]							TR
											r				r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2CBSY	保留	SMBALT	TIMEOUT	PECERR	OUERR	LOSTARB	BERR	TCR	TC	STPDET	NACK	ADDSEND	RBNE	TI	TBE
r		r	r	r	r	r	r	r	r	r	r	r	r	rw	rw

位/位域	名称	描述
31:24	保留	必须保持复位值。
23:17	READDR[6:0]	从机模式下接收到的匹配地址 当 ADDSEND 置 1 时，这些位用于存储接收到的地址。在 10 位地址情况下，READDR[6:0]存储 10 位地址头和地址的最高两位。
16	TR	I2C 在从机模式下作为发送端还是接收端 该位在 ADDSEND 位置 1 时更新。 0：接收端 1：发送端
15	I2CBSY	忙标志 该位在硬件检测到 START 信号时置 1。在 STOP 信号后硬件清 0。当 I2CEN=0 时，由硬件清零。 0：无 I2C 通讯 1：I2C 正在通讯
14	保留	必须保持复位值。
13	SMBALT	SMBus 报警 当 SMBHAEN=1，SMBALTEN=1 且在 SMBA 引脚检测到 SMBALERT 事件（下降沿）时，该位由硬件置 1。SMBALTC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 0：在 SMBA 引脚上检测到 SMBALERT 事件。 1：在 SMBA 引脚上未检测到 SMBALERT 事件。
12	TIMEOUT	超时标志 当发生超时或延展时钟超时，该位将置 1。TIMEOUTC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 0：无超时或延展时钟超时发生。 1：发生超时或延展时钟超时。
11	PECERR	PEC 错误 当接收到的 PEC 字节与 I2C_PEC 寄存器中的内容不匹配时，该位置 1。然后将自动发生 NACK。PECERRC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 0：接收到 PEC 与 I2C_PEC 的内容匹配。 1：接收到 PEC 与 I2C_PEC 的内容不匹配，此时 I2C 将忽略 NACKEN 位的值，并直接发送 NACK。
10	OUERR	从模式下上溢 / 下溢错误 在从机模式下且 SS=1，当发生上溢 / 下溢错误时，该位置 1。OUERRC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 0：未发生上溢 / 下溢错误。 1：发生上溢 / 下溢错误。
9	LOSTARB	仲裁丢失

		LOSTARBC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 0：无仲裁丢失。 1：发生仲裁丢失，I2C 模块返回从机模式。
8	BERR	总线错误 当 I2C 总线上发生了预料之外的 START 信号或 STOP 信号时，将产生总线错误，该位将置 1。BERRC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 0：无总线错误。 1：发生了总线错误。
7	TCR	传输完成重载 当 RELOAD=1 且 BYTENUM[7:0]个字节传输完成时，该位置 1。在 BYTENUM[7:0]写入一个非零值可以软件清零该位。 0：当 RELOAD=1 时，BYTENUM[7:0]个字节传输未完成。 1：当 RELOAD=1 时，BYTENUM[7:0]个字节传输完成。
6	TC	主机模式下传输完成 当 RELOAD=0，AUTOEND=0 且 BYTENUM[7:0]个字节传输完成时，该位置 1。当 START 位或 STOP 位置 1 时该位清零。 0：BYTENUM[7:0]个字节传输未完成。 1：BYTENUM[7:0]个字节传输完成。
5	STPDET	总线上检测到 STOP 信号 当在总线上检测到 STOP 信号时，主机和从机的该位由硬件置 1。STPDETC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 0：未监测到 STOP 结束位。 1：监测到 STOP 结束位。
4	NACK	接收到 NACK 应答 当接收到 NACK 时，该位置 1。NACKC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 0：接收到 ACK。 1：接收到 NACK。
3	ADDSEND	从机模式下接收到的地址与自身地址匹配 当接收到的地址与使能的从机地址之一匹配时，该位由硬件置 1。ADDSENDC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 0：接收到的地址不匹配。 1：接收到的地址匹配。
2	RBNE	接收期间 I2C_RDATA 非空 当接收到的数据移入 I2C_RDATA 寄存器时，该位置 1。读 I2C_RDATA 可清除该位。 0：I2C_RDATA 空。 1：I2C_RDATA 非空，软件可以读。
1	TI	发送中断 当 I2C_TDATA 为空且 I2C 已经做好发送数据准备时，该位置 1。在下一个待发送字

节写入 I2C_TDATA 寄存器时该位清零。当 SS=1 时，可由软件将该位置 1 来产生 TI 事件（TIE=1 时为中断，DENT=1 时为 DMA 请求）。
0：I2C_TDATA 非空或者 I2C 还未做好发送数据准备。
1：I2C_TDATA 空且 I2C 已经做好发送数据准备。

0	TBE	发送期间 I2C_TDATA 空 当 I2C_TDATA 寄存器为空，该位置 1。当下一个待发送数据写入 I2C_TDATA 寄存器时，该位清零。可以软件将该位置 1 来清空 I2C_TDATA 寄存器。 0：I2C_TDATA 非空。 1：I2C_TDATA 空。

状态清除寄存器（I2C_STATC）

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
保留		SMBALTC	TIMEOUTC	PECERRC	OUERRC	LOSTARBC	BERRC	保留		STPDETC	NACKC	ADDSENDC	保留		
		w	w	w	w	w	w			w	w	w			

位/位域	名称	描述
31:14	保留	必须保持复位值。
13	SMBALTC	SMBus 报警标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 SMBALT 位。
12	TIMEOUTC	TIMEOUT 标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 TIMEOUT 位。
11	PECERRC	PEC 错误标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 PECERR 位。
10	OUERRC	上溢 / 下溢标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 OUERR 位。
9	LOSTARBC	仲裁丢失标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 LOSTARB 位。
8	BERRC	总线错误标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 BERR 位。
7:6	保留	必须保持复位值。
5	STPDETC	停止位检测标志清零

		软件对该位写 1 可以清除 I2C_STAT 寄存器中 STPDET 位。
4	NACKC	NACK 标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 NACK 位。
3	ADDSENDC	地址匹配标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 ADDSEND 位。
2:0	保留	必须保持复位值。

PEC 寄存器（I2C_PEC）

地址偏移：0x20

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

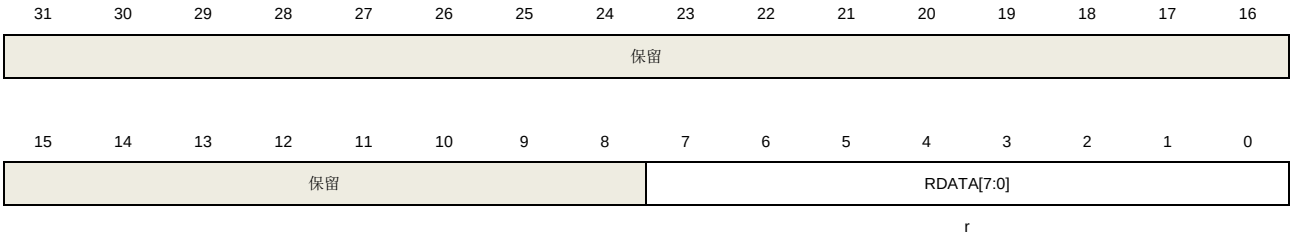
位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	PECV[7:0]	在 PEC 使能时，由硬件计算出来的 PEC 值。 当 I2CEN = 0 时，PECV 由硬件清零。

接收数据寄存器（I2C_RDATA）

地址偏移：0x24

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



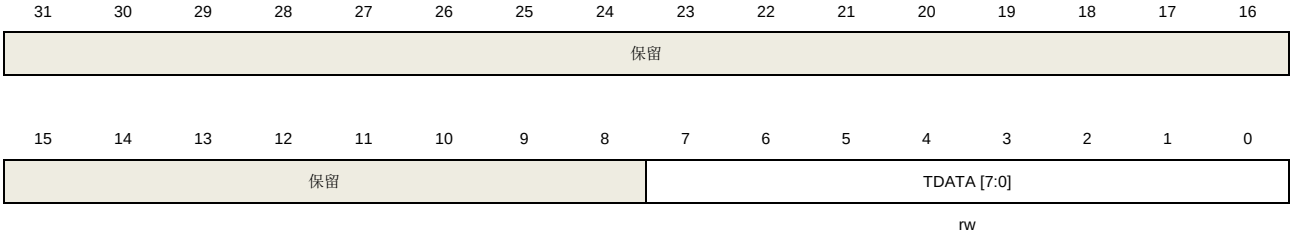
位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	RDATA[7:0]	接收到的数据

发送数据寄存器（I2C_TDATA）

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
31:8	保留	必须保持复位值。
7:0	TDATA[7:0]	发送的数据

22. 串行外设接口/片上音频接口（SPI/I2S）

22.1. 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。通过使用附加的 I2S 模块：I2S1_ADD 和 I2S2_ADD，SPI1 和 SPI2 支持 I2S 全双工模式。

22.2. 主要特性

22.2.1. 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）。

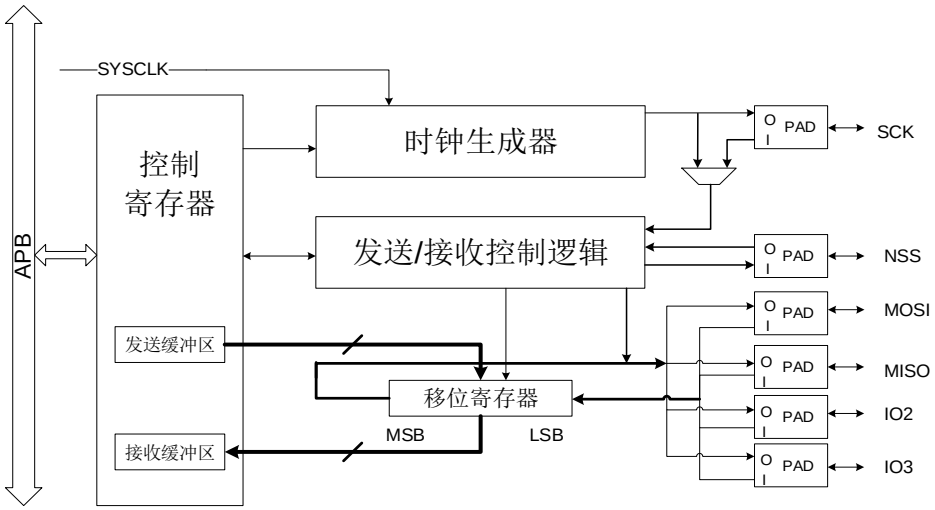
22.2.2. I2S 主要特性

- 具有发送和接收功能的主从操作。
- 具有全双工模式的主从操作（仅在SPI1和SPI2中）。
- 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。

22.3. SPI 功能说明

22.3.1. SPI 结构框图

图 22-1. SPI 结构框图



22.3.2. SPI 信号线描述

常规配置（非 SPI 四线模式）

表 22-1. SPI 信号描述

引脚名称	方向	描述
SCK	I/O	主机：SPI 时钟输出 从机：SPI 时钟输入
MISO	I/O	主机：数据接收线 从机：数据发送线 主机双向线模式：不使用 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 从机：数据接收线 主机双向线模式：数据发送和接收线 从机双向线模式：不使用
NSS	I/O	软件 NSS 模式：不使用 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。 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 默认配置为单路模式，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。

通过配置 SPI_QCTL 中的 IO23_DRV 位，在常规非四线 SPI 模式下，软件可以驱动 IO2 引脚和 IO3 引脚为高电平。

在 SPI 四线模式下，SPI 通过以下 6 个引脚与外部设备连接：

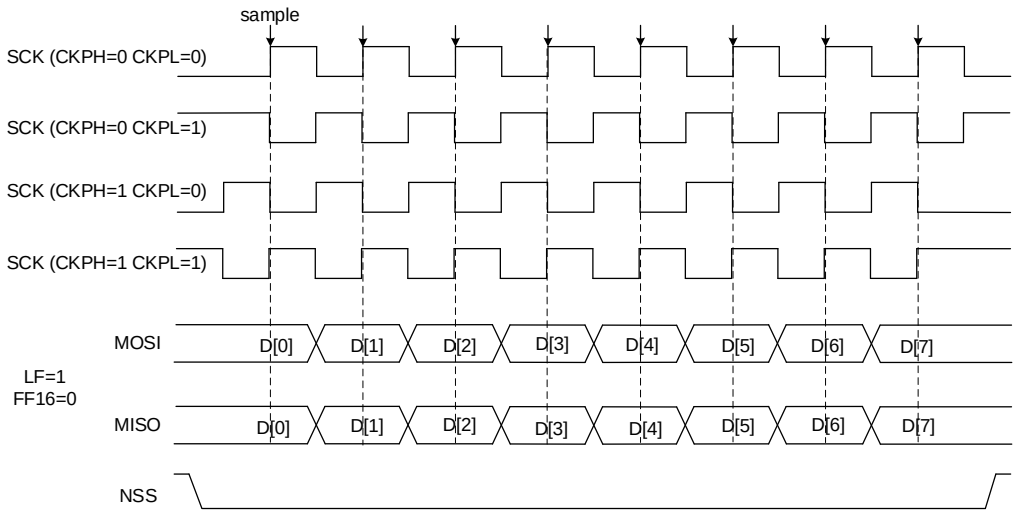
表 22-2. SPI 四线信号描述

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

22.3.3. 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。

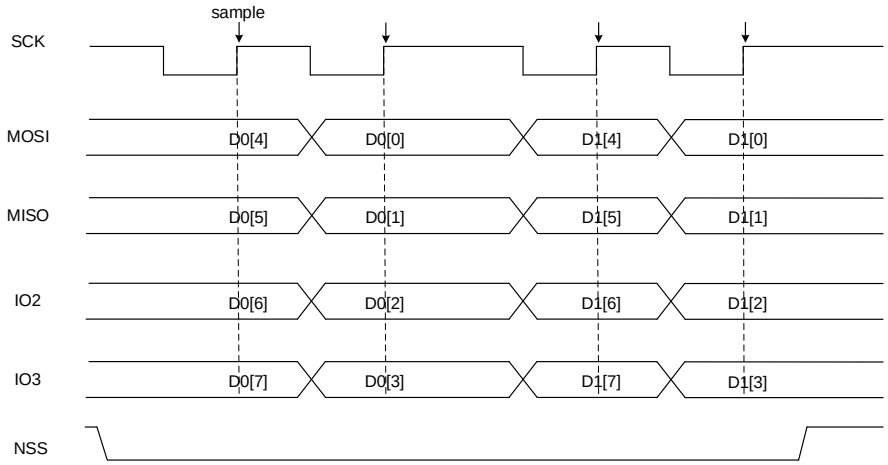
图 22-2. SPI 常规模式下的时序图



在 SPI 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。

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。

图 22-3. SPI 四线模式下的 SPI 时序图（CKPL=1, CKPH=1, LF=0）



在 SPI 四线模式下，数据长度固定为 8 位。

同样，通过设置 SPI_CTL0 中的 LF 位可以配置数据顺序，当 LF=1 时，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。

当访问 SPI_DATA 寄存器时，数据帧总是右对齐成一个字节（如果数据长度小于或等于一个字节）或一个半字。通讯时，只有数据长度内的位会随时钟输出。

22.3.4. 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 22-3. 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 SWNSSEN = 0	SPI 从机 NSS 电平从 NSS 引脚获取。
从机软件 NSS 模式	MSTMOD = 0 SWNSSEN = 1	SPI 从机 NSS 电平由 SWNSS 位决定。 SWNSS = 0：NSS 电平为低 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 22-4. 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 SWNSSEN = 0 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 SWNSSEN = 0 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 SWNSSEN = 1 SWNSS = 0 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 SWNSSEN = 1 SWNSS = 1 NSSDRV：不要求	从机可以使用硬件或软件 NSS 模式

22.3.5. SPI 运行模式

表 22-5. 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 RO = 0 BDEN = 0 BDOEN：不要求	MOSI：发送 MISO：接收
MTU	单向线连接主机发送模式	MSTMOD = 1 RO = 0 BDEN = 0 BDOEN：不要求	MOSI：发送 MISO：不使用
MRU	单向线连接主机接收模式	MSTMOD = 1 RO = 1 BDEN = 0 BDOEN：不要求	MOSI：不使用 MISO：接收
MTB	双向线连接主机发送模式	MSTMOD = 1 RO = 0 BDEN = 1 BDOEN = 1	MOSI：发送 MISO：不使用
MRB	双向线连接主机接收模式	MSTMOD = 1	MOSI：接收

模式	描述	寄存器配置	数据引脚用法
		RO = 0 BDEN = 1 BDOEN = 0	MISO：不使用
SFD	全双工从机模式	MSTMOD = 0 RO = 0 BDEN = 0 BDOEN：不要求	MOSI：接收 MISO：发送
STU	单向线连接从机发送模式	MSTMOD = 0 RO = 0 BDEN = 0 BDOEN：不要求	MOSI：不使用 MISO：发送
SRU	单向线连接从机接收模式	MSTMOD = 0 RO = 1 BDEN = 0 BDOEN：不要求	MOSI：接收 MISO：不使用
STB	双向线连接从机发送模式	MSTMOD = 0 RO = 0 BDEN = 1 BDOEN = 1	MOSI：不使用 MISO：发送
SRB	双向线连接从机接收模式	MSTMOD = 0 RO = 0 BDEN = 1 BDOEN = 0	MOSI：不使用 MISO：接收

图 22-4. 典型的全双工模式连接

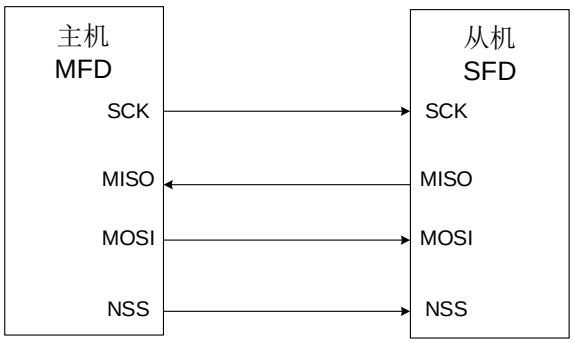


图 22-5. 典型的单工模式连接（主机：接收，从机：发送）

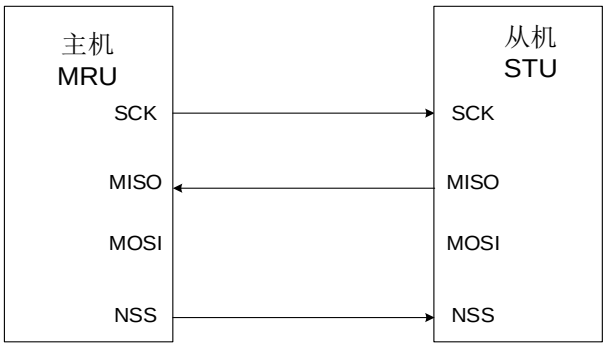


图 22-6. 典型的单工模式连接（主机：只发送，从机：接收）

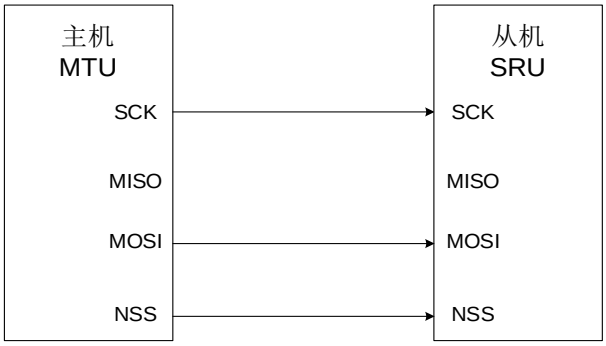
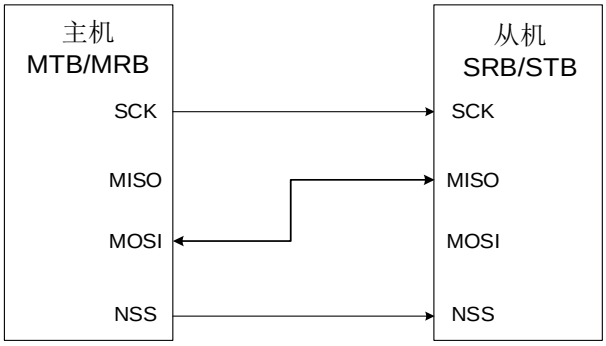


图 22-7. 典型的双向线连接



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22-5. SPI运行模式*，配置MSTMOD位、RO位、BDEN位和BDOEN位。
9. 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 22-8. 主机 TI 模式在不连续发送时的时序图

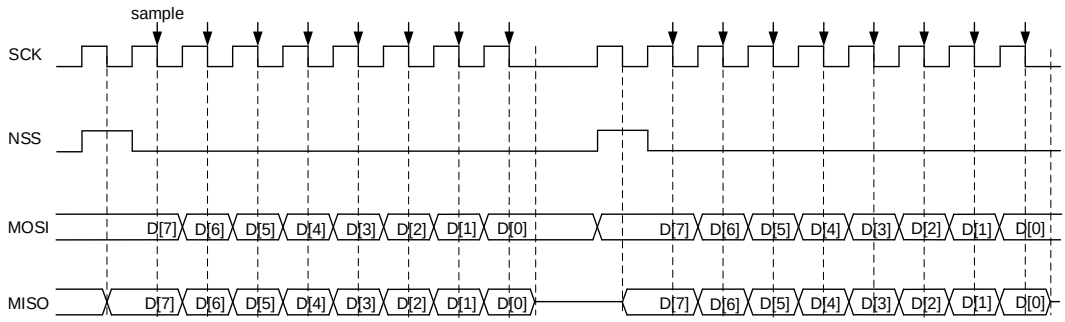
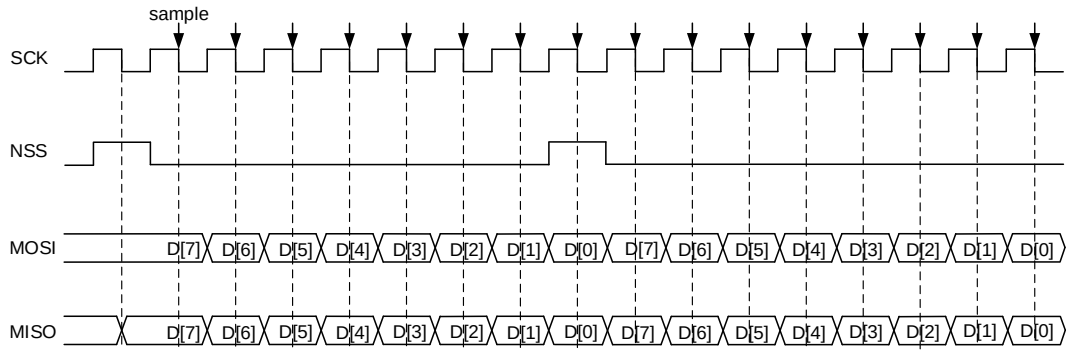
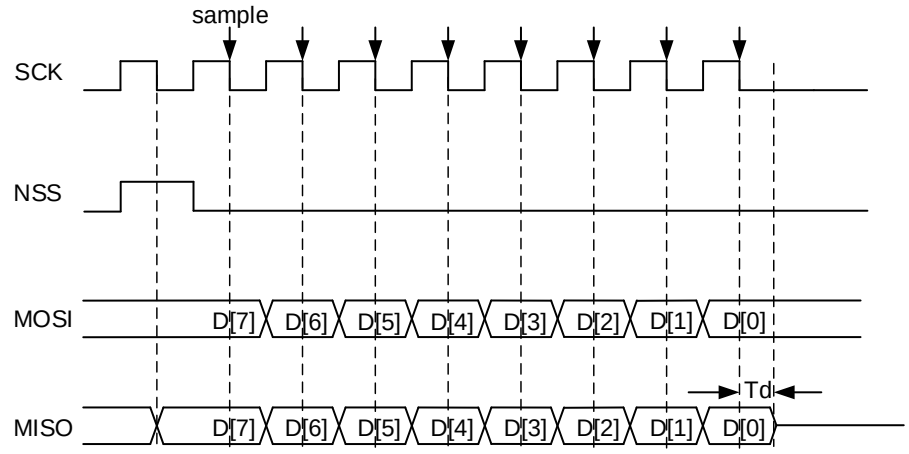


图 22-9. 主机 TI 模式在连续发送时的时序图



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 22-10. 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。

$$T_d=\frac{T_{bit}}{2}+5*T_{pclk} \quad (22\text{-}1)$$

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

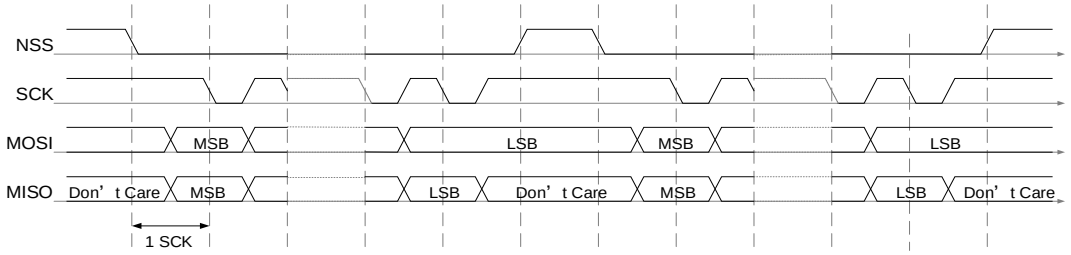
配置 SPI_CTL1 寄存器中的 NSSP 位使能该功能，为了确保使用该功能实现，需满足以下几个条件：配置设备为主机模式，使用普通 SPI 协议的数据帧格式，同时在第一个时钟跳变沿采样数据。

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 22-11. 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、CRCNT 位、RO 位和 LF 位保持清零，DZ[3:0]位域配置数据长度为 8 位，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。

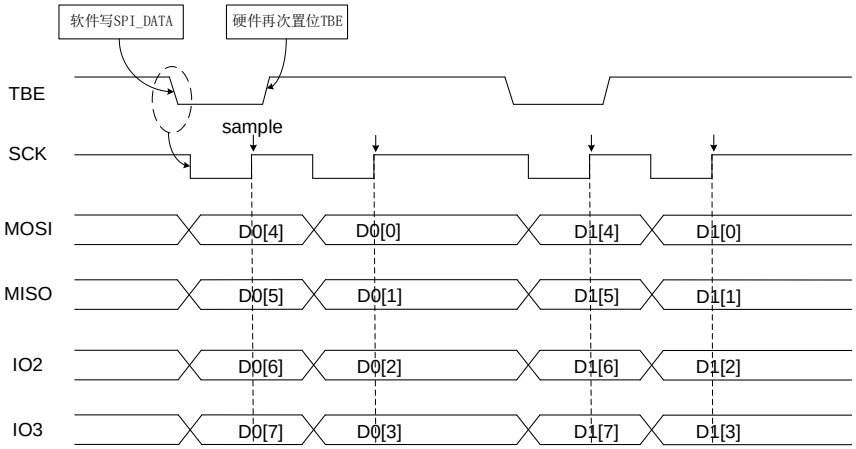
四线写模式

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 22-12. SPI 四线模式写操作时序图



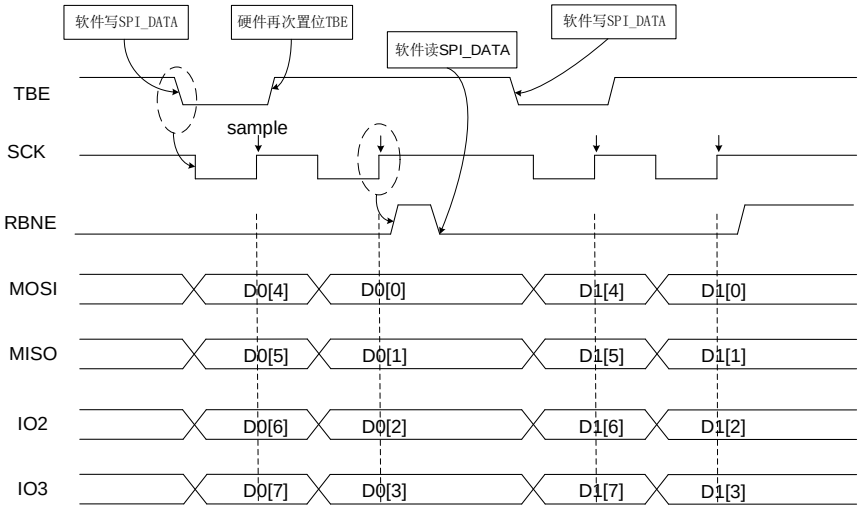
四线读模式

当 SPI_QCTL 寄存器中的 QMOD 位和 QRD 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 22-13. SPI 四线模式读操作时序图



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。

22.3.6. 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。

22.3.7. 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）。

22.3.8. 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后接收到的 CRC 值进行比较，当两者不同时，CRCERR 位将会置 1。

表 22-6. SPI 中断请求

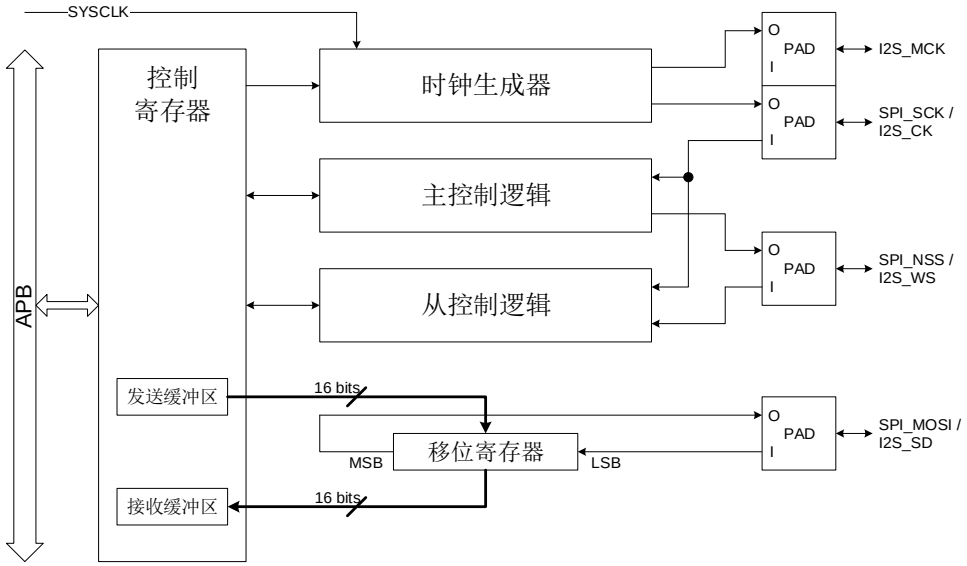
中断事件	描述	清除方式	中断使能位
TBE	发送缓冲区空	写SPI_DATA寄存器	TBEIE

中断事件	描述	清除方式	中断使能位
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	

22.4. I2S 功能说明

22.4.1. I2S 结构框图

图 22-14. 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。

22.4.2. 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，它最大可提供一个 256 倍于 Fs 的时钟频率，其中 Fs 是音频采样率。

22.4.3. 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 22-15. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=0, CKPL=0）

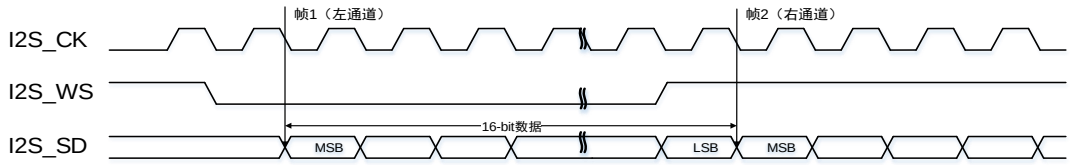
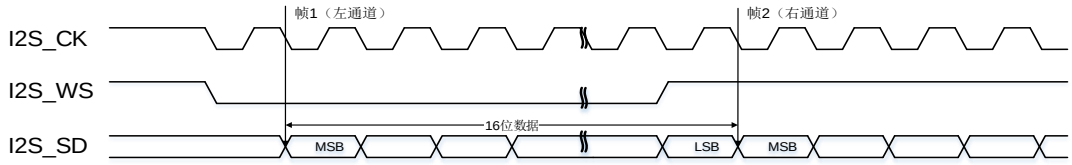


图 22-16. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=0, CKPL=1）



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 22-17. I2S 飞利浦标准时序图（DTLEN=10, CHLEN=1, CKPL=0）

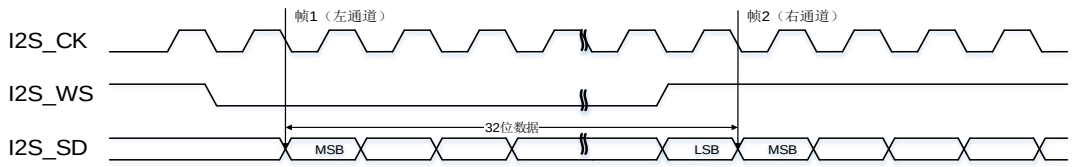
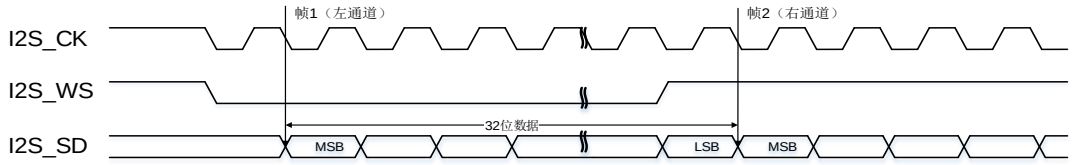


图 22-18. I2S 飞利浦标准时序图（DTLEN=10, CHLEN=1, CKPL=1）



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 22-19. I2S 飞利浦标准时序图（DTLEN=01, CHLEN=1, CKPL=0）

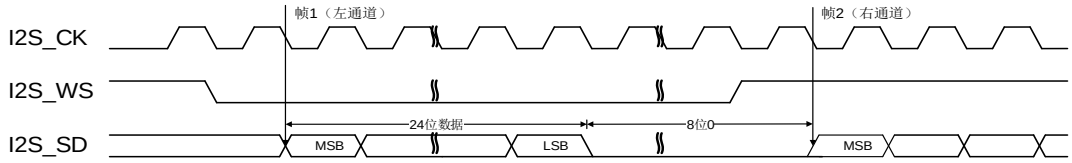
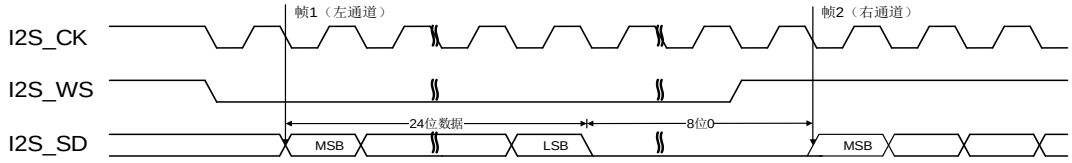


图 22-20. I2S 飞利浦标准时序图（DTLEN=01, CHLEN=1, CKPL=1）



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 22-21. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=1, CKPL=0）

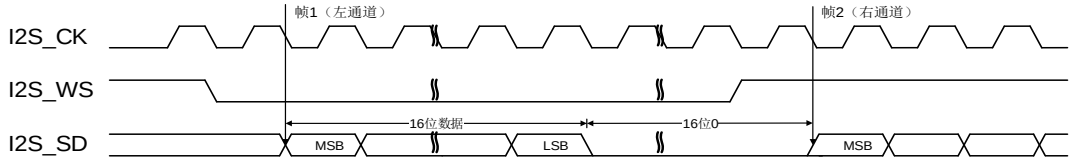
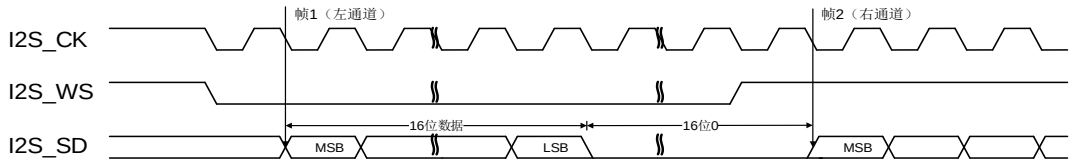


图 22-22. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=1, CKPL=1）



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 22-23. MSB 对齐标准时序图（DTLEN=00, CHLEN=0, CKPL=0）

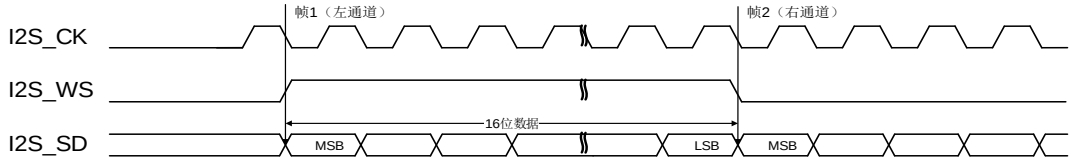


图 22-24. MSB 对齐标准时序图（DTLEN=00, CHLEN=0, CKPL=1）

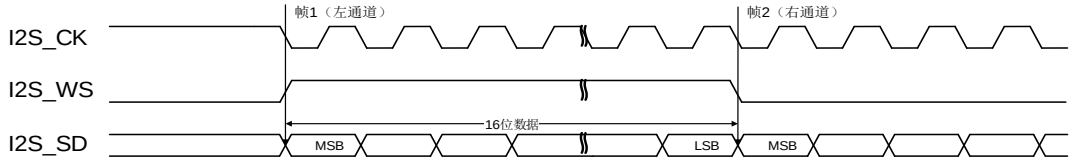


图 22-25. MSB 对齐标准时序图（DTLEN=10, CHLEN=1, CKPL=0）

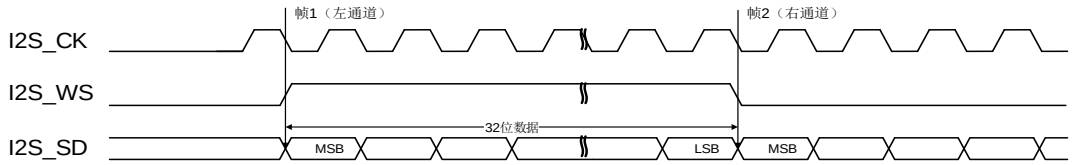


图 22-26. MSB 对齐标准时序图（DTLEN=10, CHLEN=1, CKPL=1）

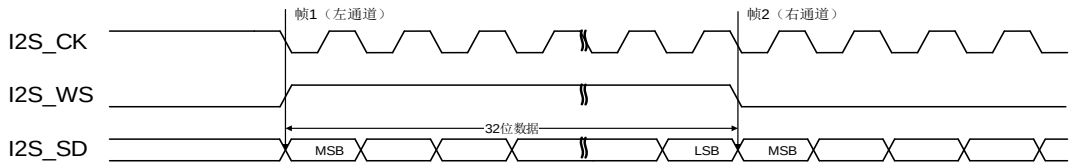


图 22-27. MSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=0）

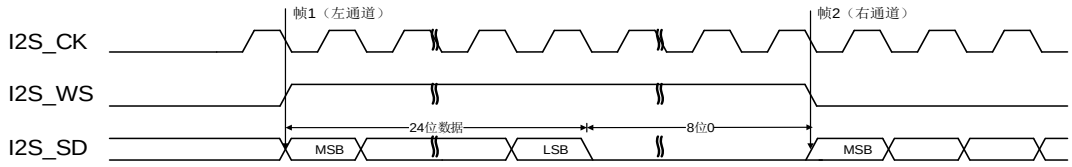


图 22-28. MSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=1）

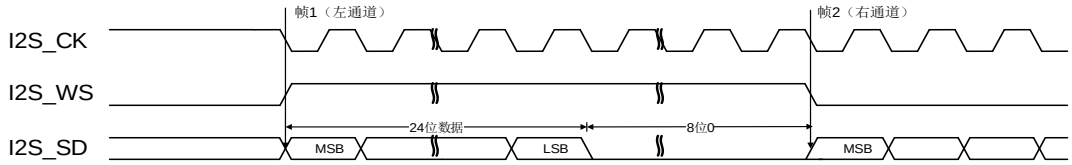


图 22-29. MSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=0）

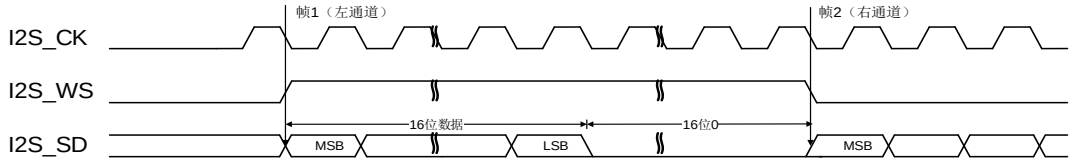
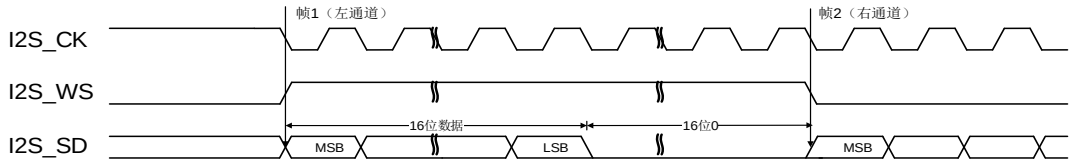


图 22-30. MSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=1）



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 22-31. LSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=0）

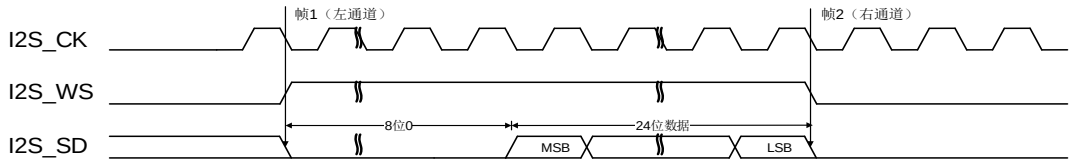
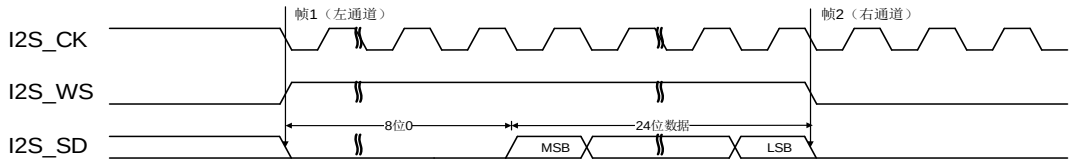


图 22-32. LSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=1）



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 22-33. LSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=0）

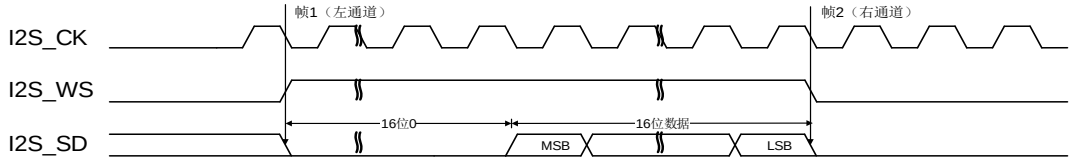
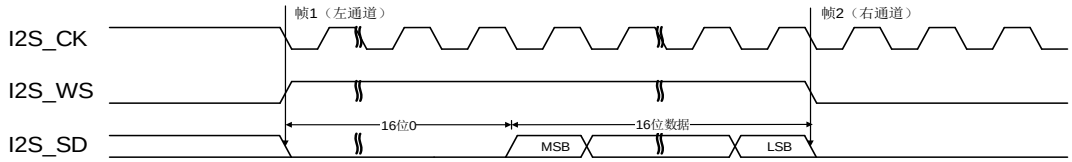


图 22-34. LSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=1）



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 22-35. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=0）

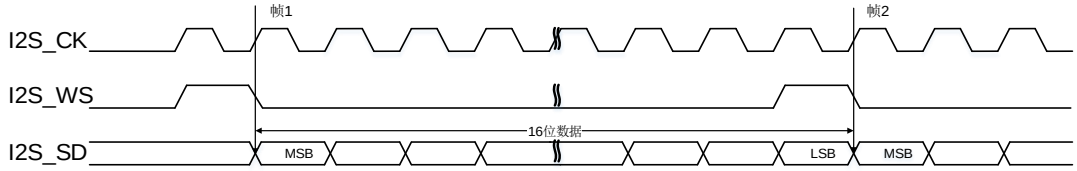


图 22-36. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=1）

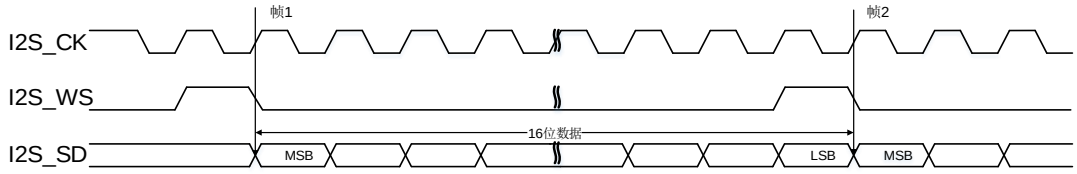


图 22-37. PCM 标准短帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=0）

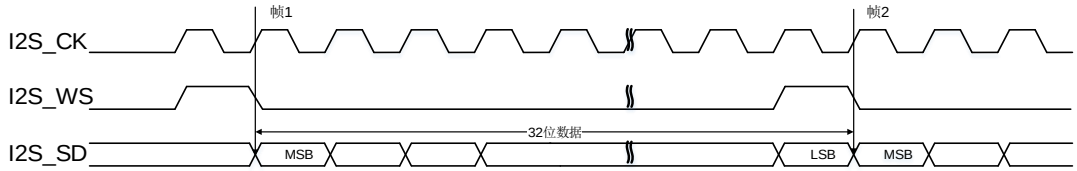


图 22-38. PCM 标准短帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=1）

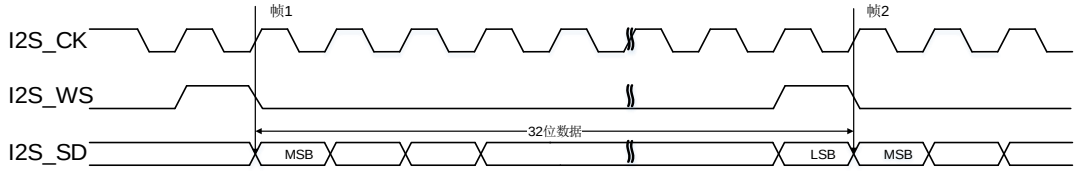


图 22-39. PCM 标准短帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=0）

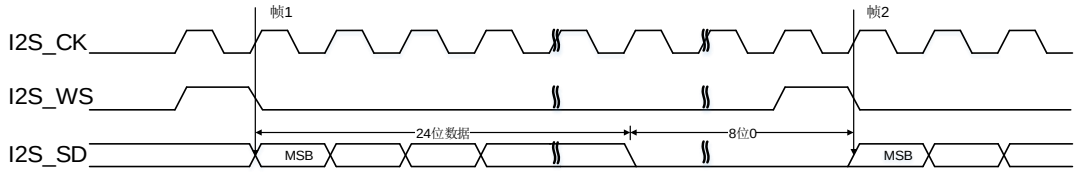


图 22-40. PCM 标准短帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=1）

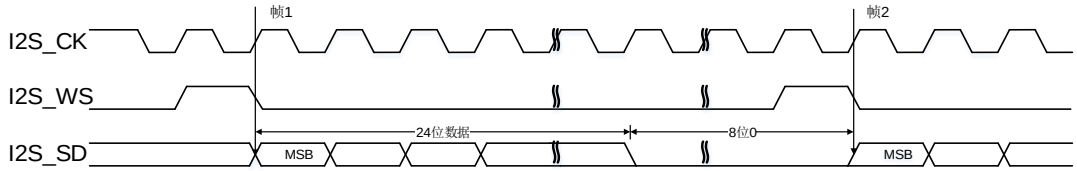


图 22-41. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=0）

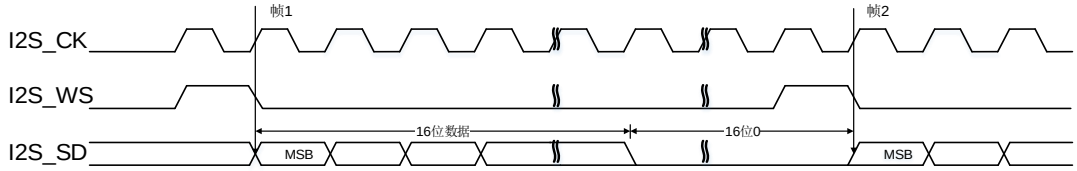
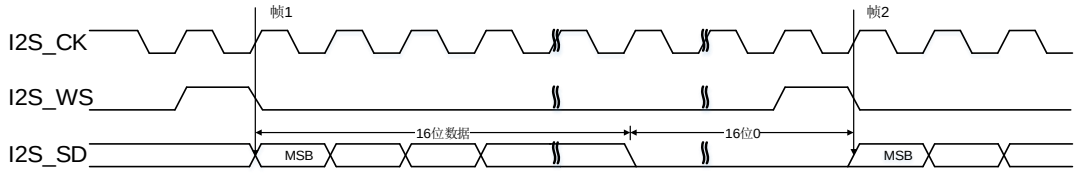


图 22-42. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=1）



长帧同步模式的各种配置情况时序图如下所示。

图 22-43. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=0）

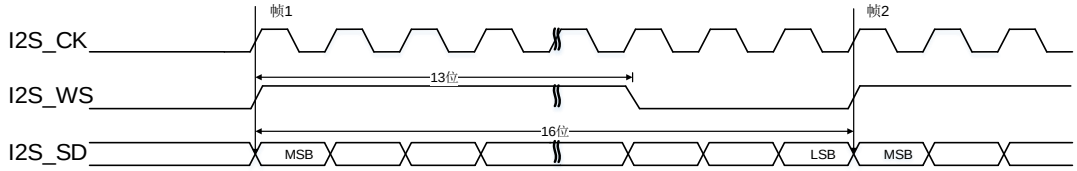


图 22-44. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=1）

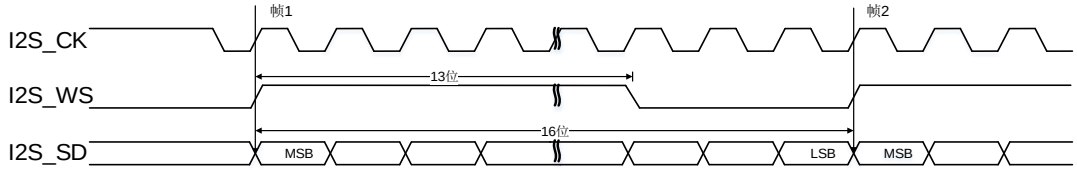


图 22-45. PCM 标准长帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=0）

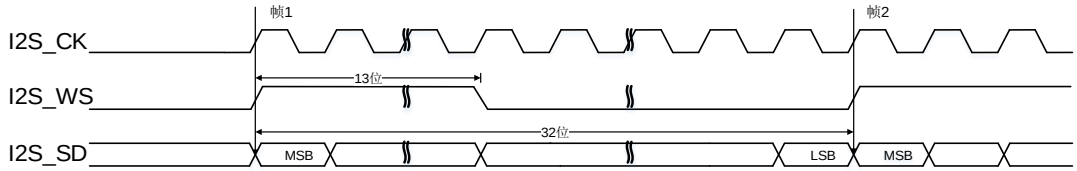


图 22-46. PCM 标准长帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=1）

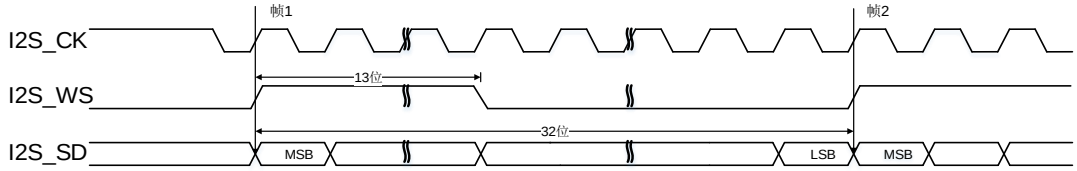


图 22-47. PCM 标准长帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=0）

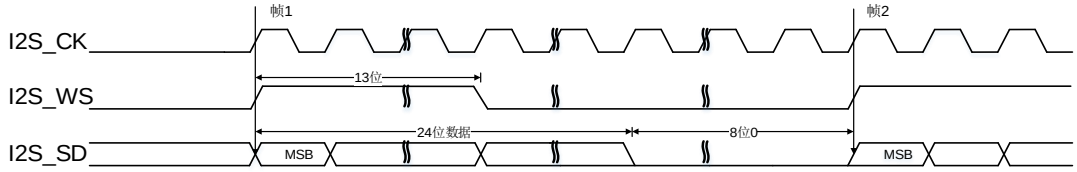


图 22-48. PCM 标准长帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=1）

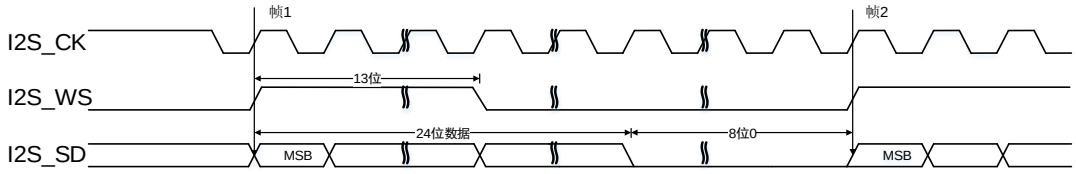


图 22-49. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=0）

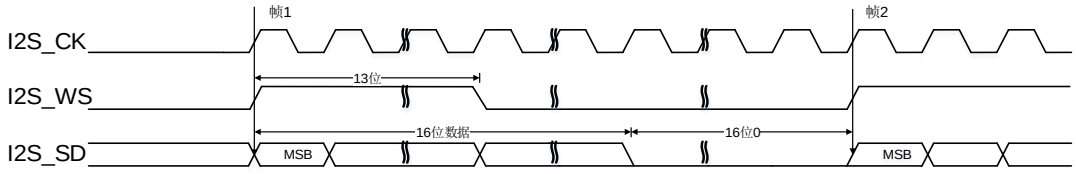
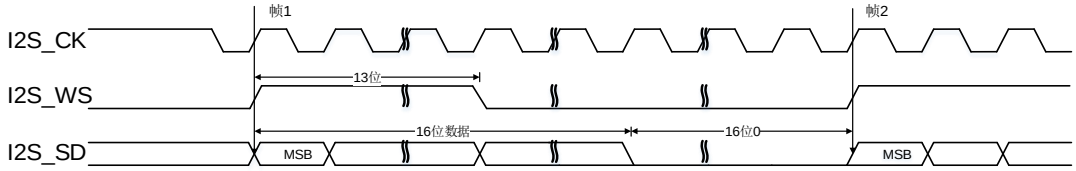
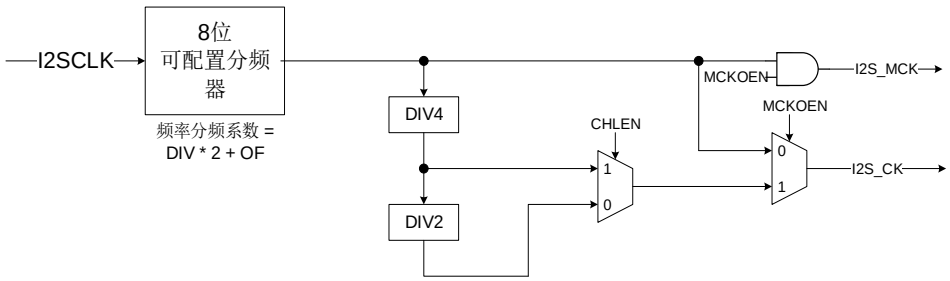


图 22-50. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=1）



22.4.4. I2S 时钟

图 22-51. I2S 时钟生成结构框图



I2S 时钟生成器框图如*图22-51. 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 或 PLL2*2。比特率可以通过*表 22-7. I2S 比特率计算公式*所示的公式计算。

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）。

表 22-7. 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 22-8. 音频采样频率计算公式*所列的公式进行配置。

表 22-8. 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))

22.4.5. 运行

运行模式

运行模式是通过 SPI_I2SCTL 寄存器的 I2SOPMOD[1:0]位域来选择的。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 22-9. 各种运行模式下 I2S 接口信号的方向*所示。

表 22-9. 各种运行模式下 I2S 接口信号的方向

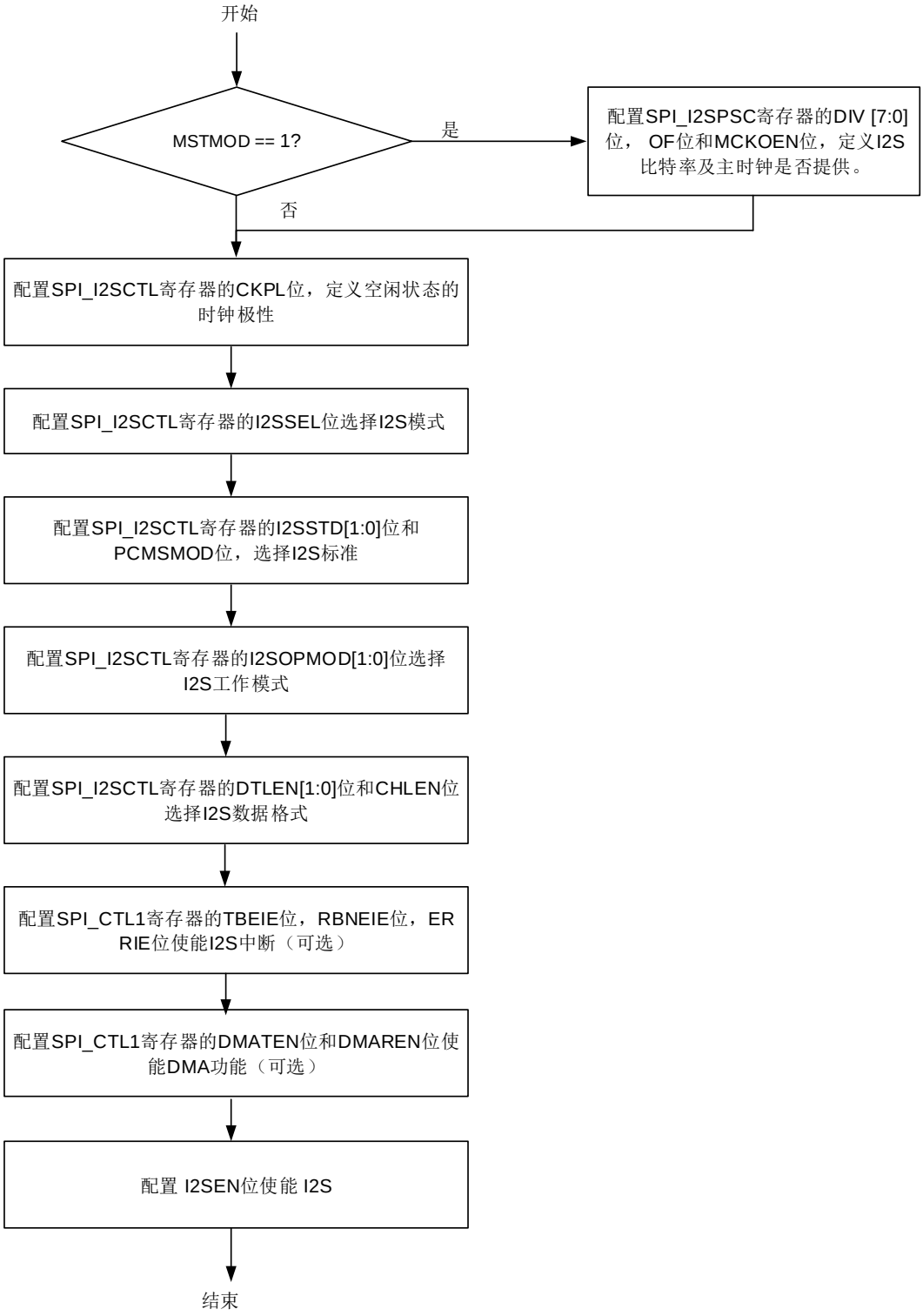
运行模式	I2S_MCK	I2S_CK	I2S_WS	I2S_SD	I2S_ADD_SD(2)
主机发送	输出或 NU(1)	输出	输出	输出	NU(1)
主机接收	输出或 NU(1)	输出	输出	输入	NU(1)
从机发送	输入或 NU(1)	输入	输入	输出	NU(1)
从机接收	输入或 NU(1)	输入	输入	输入	NU(1)
全双工	输出或 NU(1)	输出	输出	输出或输入	输入或输出

1. NU表示该引脚没有被I2S使用，可以用于其他功能。
2. I2S1和I2S2为了支持全双工运行模式，需要两个额外的片上I2S模块：I2S_ADD1和I2S_ADD2。I2S_ADD_SD引脚是I2S_ADD模块的数据引脚，在后面的章节将详细介绍全双工模式。

I2S 初始化流程

I2S 初始化过程如*图 22-52. I2S 初始化流程*所示。

图 22-52. 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 标志位置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。

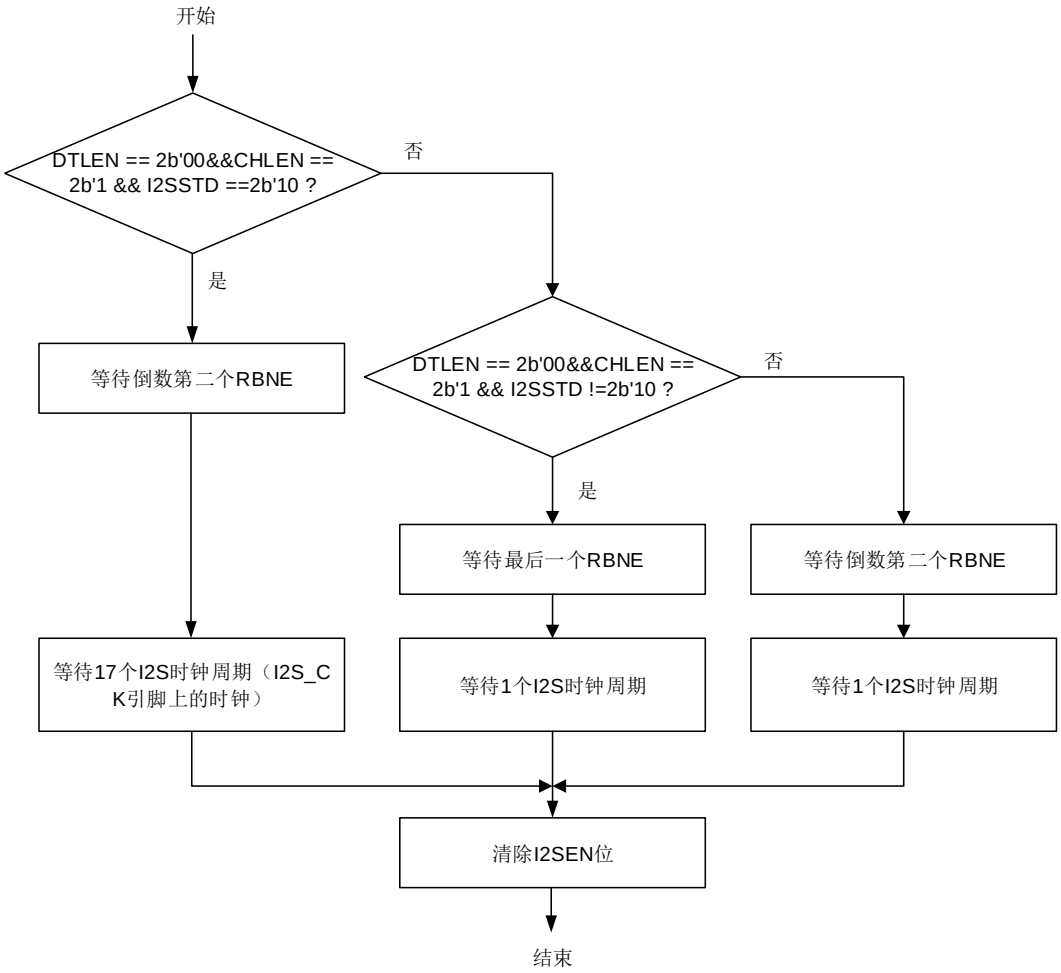
I2S 主机接收流程

RBNE 标志被用来控制接收序列。如前文所述，RBNE 标志置位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 22-53. I2S 主机接收禁能流程*所示。

图 22-53. 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。

为了关闭 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。

I2S 全双工模式

单个的 I2S 模块只支持单向传输：发送模式或接收模式，通过一个附加的 I2S 模块（I2S_ADD 模块）可以实现 I2S 的全双工模式。I2S_ADD 模块与 I2S 模块功能一样，但只工作在从模式。一共有两个 I2S_ADD 模块：I2S_ADD1 和 I2S_ADD2，所以只有 I2S1 和 I2S2 支持全双工模式。I2S_ADD 模块的 I2S_CK 和 I2S_WS 引脚分别与对应的 I2S 模块的相应引脚内部连接，I2S_ADD 模块的 I2S_SD 引脚映射到对应的 I2S 模块的 SPI_MISO 引脚。

为了工作在全双工模式，需要使能 I2S 模块和相应的 I2S_ADD 模块，I2S 模块支持两种全双工模式：主机模式和从机模式。

在主机全双工模式下，软件必须设置 I2S 为主机，I2S_ADD 为从机，I2S_ADD 模块的 WS 和 SCK 信号都由主机 I2S 模块提供。

在从机全双工模式下，软件必须设置 I2S 和 I2S_ADD 都为从机，I2S 模块和 I2S_ADD 模块的 WS 和 SCK 信号都由外部信号提供。

应用程序可以配置 I2S 模块为发送或接收模式，然后配置 I2S_ADD 为相反的模式。在发送过程中，软件同时操作 I2S 模块和 I2S_ADD 模块的寄存器和中断来实现全双工模式发送。

22.4.6. 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。

22.4.7. 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。该标志位不会产生任何中断。

- **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22-10. I2S中断*总结了I2S中断事件和相应的使能位。

表 22-10. 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	

22.5. 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

I2S1_ADD 基地址：0x4000 3400

I2S2_ADD 基地址：0x4000 4000

22.5.1. 控制寄存器 0（SPI_CTL0）

地址偏移：0x00
复位值：0x0000 0000

该寄存器可以按字节（8 位）或半字（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 0：2线单向传输模式。 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 当BDEN置位时，该位决定了数据的传输方向。 0：工作在只接收模式。 1：工作在只发送模式。
13	CRCEN	CRC计算使能 0：禁能CRC计算。 1：使能CRC计算。
12	CRCNT	下一次传输CRC 0：下一次传输值为数据。 1：下一次传输值为CRC值。 当数据传输由DMA管理时，CRC值由硬件传输，该位应该被清零。

		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。 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 0：8位数据帧格式 1：16位数据帧格式
10	RO	只接收模式 当BDEN清零时，该位决定了数据的传输方向。 0：全双工模式 1：只接收模式
9	SWNSSEN	NSS软件模式使能 0：NSS硬件模式，NSS电平取决于NSS引脚。 1：NSS软件模式，NSS电平取决于SWNSS位。 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 0：NSS引脚拉低。 1：NSS引脚拉高。 只有在SWNSSEN置位时，该位有效。 该位在SPI TI模式下没有意义。
7	LF	最低有效位先发模式 0：先发送最高有效位。 1：先发送最低有效位。 该位在SPI TI模式下没有意义。
6	SPIEN	SPI使能 0：禁能SP 1：使能SPI
5:3	PSC[2:0]	主时钟预分频选择 000：PCLK/2 001：PCLK/4 010：PCLK/8 011：PCLK/16 100：PCLK/32 101：PCLK/64 110：PCLK/128 111：PCLK/256 当使用SPI0时，PCLK=PCLK2。当使用SPI1或者SPI2时，PCLK=PCLK1。
2	MSTMOD	主从模式使能 0：从机模式 1：主机模式
1	CKPL	时钟极性选择 0：SPI为空闲状态时，CLK引脚拉低。

1：SPI为空闲状态时，CLK引脚拉高。

0	CKPH	时钟相位选择 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采集第一个数据。

22.5.2. 控制寄存器 1（SPI_CTL1）

地址偏移：0x04
复位值：0x0000 0000

该寄存器可以按字节（8 位）或半字（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								TBEIE	RBNEIE	ERRIE	TMOD	NSSP	NSSDRV	DMATEN	DMAREN
								rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:8	保留	必须保持复位值。
7	TBEIE	发送缓冲区空中断使能 0：禁能TBE中断 1：使能TBE中断。当TBE置位时，产生中断。
6	RBNEIE	接收缓冲区非空中断使能 0：禁能RBNE中断 1：使能RBNE中断。当RBNE置位时，产生中断。
5	ERRIE	错误中断使能 0：禁能错误中断 1：使能错误中断。当CRCERR位，CONFERR位，FERR位，RXORERR位或者TXURERR位置1时，产生中断。
4	TMOD	SPI TI模式使能 0：禁能SPI TI模式 1：使能SPI TI模式
3	NSSP	SPI NSS脉冲模式使能 0：禁能SPI NSS脉冲模式 1：使能SPI NSS脉冲模式
2	NSSDRV	NSS输出使能 0：禁能主机NSS输出 1：使能主机NSS输出
1	DMATEN	发送缓冲区DMA使能

0：禁能发送缓冲区DMA
1：使能发送缓冲区DMA。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 0：禁能接收缓冲区DMA 1：使能接收缓冲区DMA。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。

22.5.3. 状态寄存器（SPI_STAT）

地址偏移：0x08
复位值：0x0000 0002

该寄存器可以按字节（8 位）或半字（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 SPI TI 模式： 0：没有 TI 模式帧错误发生 1：TI 模式帧错误发生 I2S 模式： 0：没有 I2S 帧错误发生 1：I2S 帧错误发生 该位由硬件置位，可以通过写0清除。
7	TRANS	通信进行中标志 0：SPI或I2S空闲 1：SPI或I2S当前正在发送且/或接收数据。 该位由硬件置位和清除。
6	RXORERR	接收过载错误标志 0：没有接收过载错误发生 1：接收过载错误发生 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。

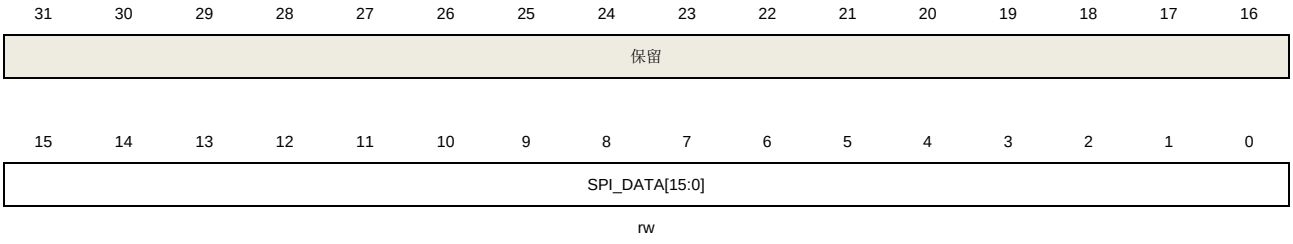
5	CONFERR	SPI配置错误 0：无配置错误发生 1：配置错误发生（主机模式下，在硬件NSS模式时NSS引脚被拉低，或者软件NSS模式时SWNSS位为0，都会产生CONFERR错误。） 该位由硬件置位，软件序列清零。软件序列为：读或写SPI_STAT寄存器，然后写SPI_CTL0寄存器。 I2S模式下不使用该位。
4	CRCERR	SPI CRC错误标志 0：SPI_RCRC值等于最后接收到的CRC值。 1：SPI_RCRC值不等于最后接收到的CRC值该位由硬件置位，可以通过写0清除。 I2S模式下不使用该位。
3	TXURERR	发送欠载错误标志 0：无发送欠载错误发生 1：发送欠载错误发生 该位由硬件置位，通过读SPI_STAT寄存器清除。 SPI模式下不使用该位。
2	I2SCH	I2S通道标志 0：下一个将要发送或接收的数据属于左通道 。 1：下一个要发送或接收的数据属于右通道 。 该位由硬件置位和清除。 SPI模式下不使用该位，I2S PCM模式下该位没有意义。
1	TBE	发送缓冲区空 0：发送缓冲区非空 1：发送缓冲区空
0	RBNE	接收缓冲区非空 0：接收缓冲区空 1：接收缓冲区非空

22.5.4. 数据寄存器（SPI_DATA）

地址偏移：0x0C

复位值：0x0000 0000

该寄存器可以按字节（8 位）或半字（16 位）或字（32 位）访问。



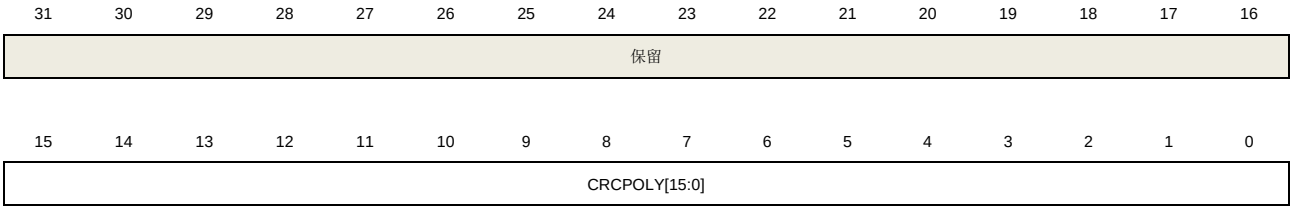
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	SPI_DATA[15:0]	数据传输寄存器值 硬件有两个缓冲区：发送缓冲区和接收缓冲区。向SPI_DATA写数据将会把数据存入发送缓冲区，从SPI_DATA读数据，将从接收缓冲区获得数据。当数据帧格式为8位时，SPI_DATA[15:8]强制为0，SPI_DATA[7:0]用来发送和接收数据，发送和接收缓冲区都是8位。如果数据帧格式为16位，SPI_DATA[15:0]用于发送和接收数据，发送和接收缓冲区也是16位。

22.5.5. CRC 多项式寄存器（SPI_CRCPOLY）

地址偏移：0x10

复位值：0x0000 0007

该寄存器可以按字节（8 位）或半字（16 位）或字（32 位）访问。



rw

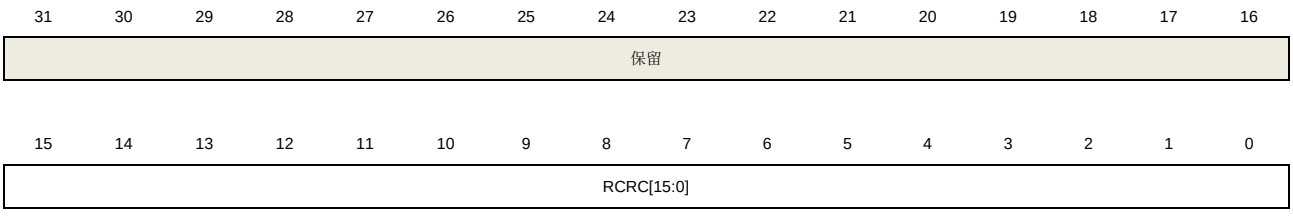
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CRCPOLY[15:0]	CRC多项式寄存器值 该值包含了CRC多项式，用于CRC计算，默认值为0007h。

22.5.6. 接收 CRC 寄存器（SPI_RCRC）

地址偏移：0x14

复位值：0x0000 0000

该寄存器可以按字节（8 位）或半字（16 位）或字（32 位）访问。



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 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。

22.5.7. 发送 CRC 寄存器（SPI_TCRC）

地址偏移：0x18

复位值：0x0000 0000

该寄存器可以按字节（8 位）或半字（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 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。 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。

22.5.8. I2S 控制寄存器（SPI_I2SCTL）

地址偏移：0x1C

复位值：0x0000 0000

该寄存器可以按字节（8 位）或半字（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]		PCMS	保留	I2SSTD[1:0]		CKPL	DTLEN[1:0]		CHLEN

				MOD					
	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 0：SPI模式 1：I2S模式 当SPI模式或I2S模式关闭时配置该位。
10	I2SEN	I2S使能 0：禁能I2S 1：使能I2S SPI模式不使用该位。
9:8	I2SOPMOD[1:0]	I2S运行模式 00：从机发送模式 01：从机接收模式 10：主机发送模式 11：主机接收模式 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 0：短帧同步 1：长帧同步 只有在PCM标准下，该位才有意义。 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 00：I2S飞利浦标准 01：MSB对齐标准 10：LSB对齐标准 11：PCM标准 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 0：I2S_CK空闲状态为低电平 1：I2S_CK空闲状态为高电平 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 00：16位 01：24位 10：32位 11：保留

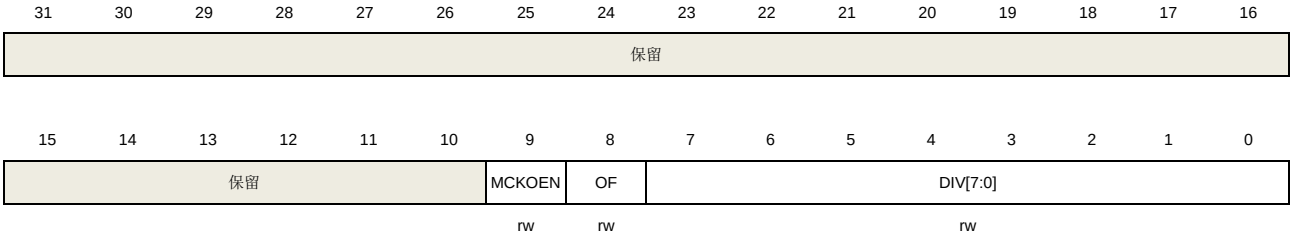
		当I2S关闭时配置该位。SPI模式不使用该位。
0	CHLEN	通道长度 0：16位 1：32位 通道长度必须大于或等于数据长度。 当I2S关闭时配置该位。SPI模式不使用该位。

22.5.9. I2S 时钟预分频寄存器（SPI_I2SPSC）

地址偏移：0x20

复位值：0x0000 0002

该寄存器可以按字节（8 位）或半字（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 0：禁能I2S_MCK输出 1：使能I2S_MCK输出 当I2S关闭时配置该位。 SPI模式不使用该位。
8	OF	预分频器的奇系数 0：实际分频系数为DIV * 2 1：实际分频系数为DIV * 2 + 1 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 实际分频系数是DIV * 2 + OF。 DIV不能为0。 当I2S关闭时配置该位。SPI模式下不使用该位。

22.5.10. SPI0 四线 SPI 控制寄存器（SPI_QCTL）

地址偏移：0x80

复位值：0x0000 0000

该寄存器可以按字节（8 位）或半字（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													IO23_DRV	QRD	QMOD
													rw	rw	rw

位/位域	名称	描述
31:3	保留	必须保持复位值。
2	IO23_DRV	IO2和IO3输出使能 0：单路模式下IO2和IO3输出关闭 1：单路模式下IO2和IO3输出高电平 该位仅适用于SPI0。
1	QRD	四线SPI模式读选择 0：SPI四线模式写操作 1：SPI四线模式读操作 该位仅能在SPI未通信时配置（TRANS位清零）。 该位仅适用于SPI0。
0	QMOD	四线SPI模式使能 0：SPI工作在单路模式 1：SPI工作在四线模式 该位仅能在SPI未通信时配置（TRANS位清零）。 该位仅适用于SPI0。

23. SPI/QPI（SQPI）

23.1. 简介

SQPI 接口是一个用于串行、双线、四线接口存储设备的控制器。例如，控制 SQPI-PSRAM 和 SQPI- FLASH。

利用 SQPI 接口可以像使用 SRAM 一样使用 SQPI 接口的存储器

23.2. 主要特性

- 两组独立的寄存器用于读操作和写操作；
- 支持ID长度配置；
- 读操作时SQPI时钟的采样边沿可配置；
- 支持命令阶段、地址阶段、和等待周期阶段的长度配置；
- 支持时钟输出由AHB时钟分频；
- 支持无地址阶段和数据阶段的特殊指令；
- 一个AHB读ID指令可以获取大于32位的ID数据；
- 支持AHB突发操作和8、16、32位的AHB指令；
- 支持256MB的外部存储空间，逻辑地址范围：0xB000 0000 - 0xBFFF FFFF；
- 支持六种模式，这些模式是不同的命令阶段，地址阶段，等待阶段，和数据阶段的组合。

23.3. 功能描述

23.3.1. SQPI 模式定义

模式定义的命名，第一个字符表示命令阶段的有效 IO 个数，第二个字符表示地址阶段的有效 IO 个数，第三个字符表示数据阶段的有效 IO 个数。对于每个字符，S 表示 1 个 IO 口，D 表示 2 个 IO 口，Q 表示 4 个 IO 口。

表 23-1. SQPI 控制器模式定义

引脚	方向	操作模式					
		SSS	SSQ	SQQ	QQQ	SSD	SDD
SQPI_CLK	输出	串行时钟					
SQPI_CSN	输出	片选 (低电平有效)					
命令阶段							
SQPI_D0	输出	O	O	O	O	O	O
SQPI_D1	输出	X	X	X	O	X	X
SQPI_D2	输出	0	0	0	O	0	0
SQPI_D3	输出	1	1	1	O	1	1

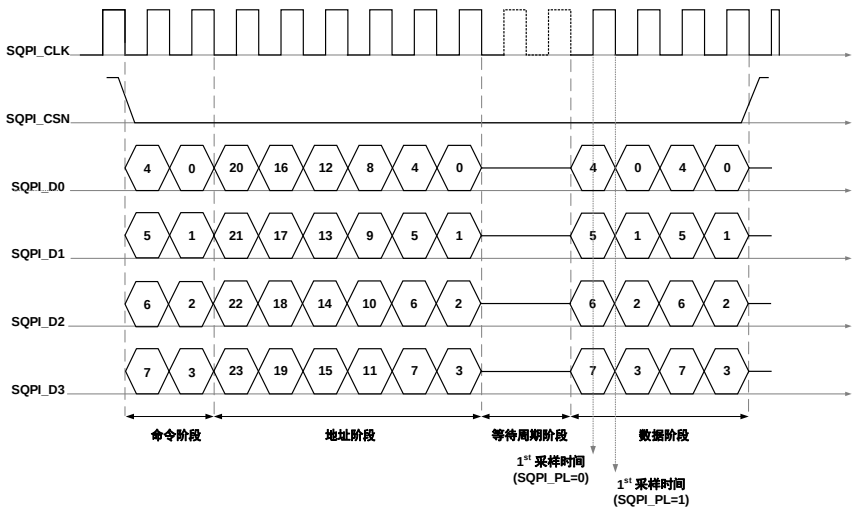
地址阶段							
SQPI_D0	输出	O	O	O	O	O	O
SQPI_D1	输出	X	X	O	O	X	O
SQPI_D2	输出	0	0	O	O	0	0
SQPI_D3	输出	1	1	O	O	1	1
等待周期阶段							
SQPI_D0	输入输出	X	X	X	X	X	X
SQPI_D1	输入输出	X	X	X	X	X	X
SQPI_D2	输入输出	0	X	X	X	0	0
SQPI_D3	输入输出	1	X	X	X	1	1
数据阶段							
SQPI_D0	输入输出	O	IO	IO	IO	IO	IO
SQPI_D1	输入输出	I	IO	IO	IO	IO	IO
SQPI_D2	输入输出	X	IO	IO	IO	X	X
SQPI_D3	输入输出	X	IO	IO	IO	X	X

Note: O – 输出, I – 输入, IO – 输入输出, 0 – 输出 0, 1 – 输出 1, X – 高阻态

23.3.2. SQPI 控制器采样极性

SQPI 控制器的在读操作时采用极性（SQPI_PL）选择功能支持用户改变控制器的采样时间。这个功能在 SQPI 时钟高的时候非常有用。示例如下：

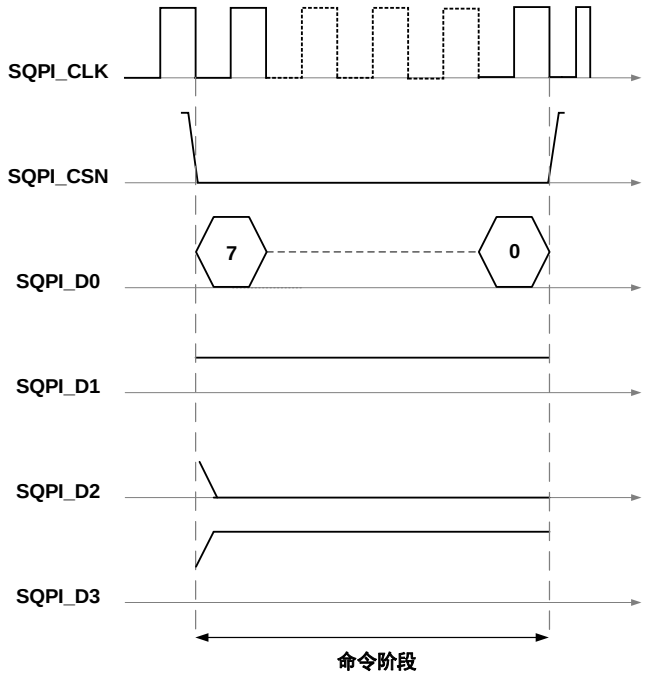
图 23-1. SQPI 采样极性



23.3.3. SQPI 控制器特殊指令

SQPI 控制器特殊命令功能可以只发送命令，而没有地址阶段，等待周期阶段和数据阶段。特殊指令功能由硬件强制使用 SSS 模式。如果设置 SQPI_SCMD 位为 1，需要在其他存储访问操作之前读取该位并等待被清 0，这样可以确保该操作被执行完成。

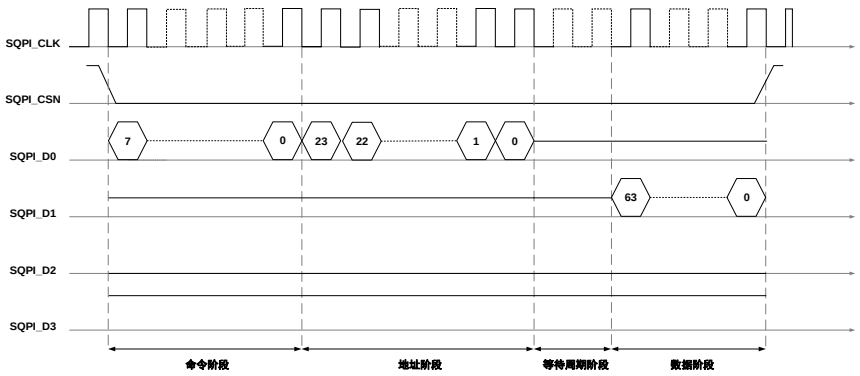
图 23-2. SQPI_SCMD 示例



23.3.4. SQPI 读 ID 命令

对于超过 32 位的 ID 数据，SQPI_RDID 功能也可以提供支持。使用该功能，首先需要设置 SQPI_IDLEN 位为 0x00(默认的 64 位)，然后设置 SQPI_RDID 为 1 并轮询该位直到被硬件清 0，最后读取 SQPI_IDL 和 SQPI_IDH 寄存器。该命令的执行被硬件强制使用 SSS 模式。

图 23-3. SQPI_RDID 示例(SQPI_IDLEN=00)



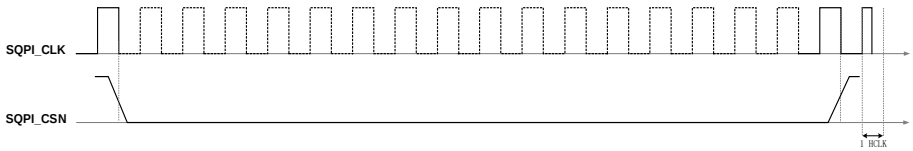
23.3.5. SQPI 控制器输出时钟配置

SQPI 时钟周期由 SQPI_CLKDIV 位配置，SQPI 的时钟频率公式如下：

$$\square_{\square\square\square\square_\square\square\square} = \frac{\square\ \square\square\square}{\square\square\square\square_\square\square\square\square\square\square+1} \qquad (23\text{-}1)$$

注意: SQPI_CLKDIV 不能为 0。当 SPI_CLKDIV 位域为偶数时，时钟输出的高电平时间要比低电平时间多一个 AHB 时钟周期。为了支持一些旧版本的 PSRAM,在 SQPI_CSN 引脚上升沿之后，在 SQPI_CLK 引脚上会有一个时钟周期的 AHB 时钟。

图 23-4. SQPI_CLK 示例



23.3.6. SQPI 控制器初始化

在开始配置时，用户可以编程初始化寄存器 SQPI_INIT。数据采样边沿可以通过 SQPI_PL 位配置，设备的 ID 长度可以通过 SQPI_IDLEN 位域配置，地址位数可以通过 SQPI_ADDRBIT 位域配置，命令位数可以通过 SQPI_CMDBIT 位域配置，时钟频率通过 SQPI_CLKDIV 位域配置。

23.3.7. 读 ID 命令流程

首先，用户需要通过 SQPI_RCMD 位域配置读 ID 命令。（例如 SQPIPSRAM 的读 ID 命令为 0x9F）并且在 SQPI_RCMD 寄存器中配置等待周期数。然后，设置 SQPI_RID 位为 1 并等待被清 0。最后，可以通过 SQPI_IDL 和 SQPI_IDH 寄存器获取 ID 值。

23.3.8. 读写操作流程

提供六种模式进行存储访问，需要在读写操作之前配置访问模式。读写操作的模式通过 SQPI_RMODE 和 SQPI_WMODE 位域进行配置。等待周期由 SQPI_RWAITCYCLE and SQPI_WWAITCYCLE 位域进行配置。通过 SQPI_RCMD 和 SQPI_WCMD 位域配置存储器规定的操作命令，读写操作的设置分别位于 SQPI_RCMD 和 SQPI_WCMD 寄存器

在存储设备的访问配置完成后，用户可以通过 SQPI 的逻辑地址像访问 SRAM 一样直接访问外部存储设备。

23.3.9. SQPI 控制器模式时序

QSPI 的读/写操作时序如下图，每次通过 AHB 读或者写访问 SQPI 逻辑地址时会发送下列时序。

图 23-5. SQPI SSS 模式时序(SPI)

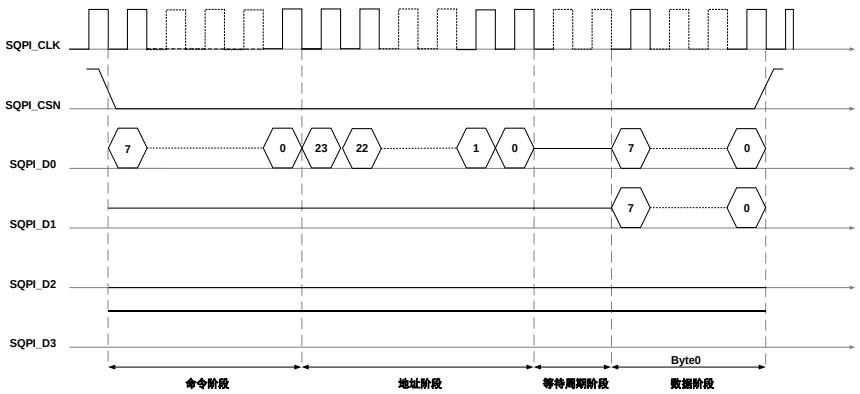


图 23-6. SQPI SSQ 模式时序

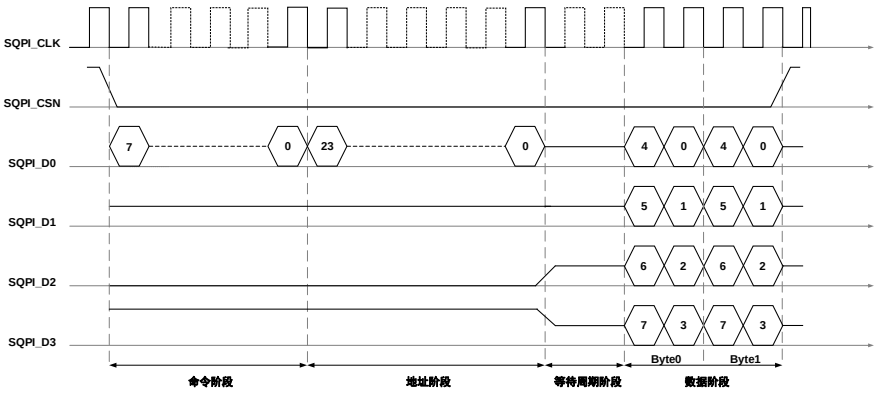


图 23-7. SQPI SQQ 模式时序(SQPI)

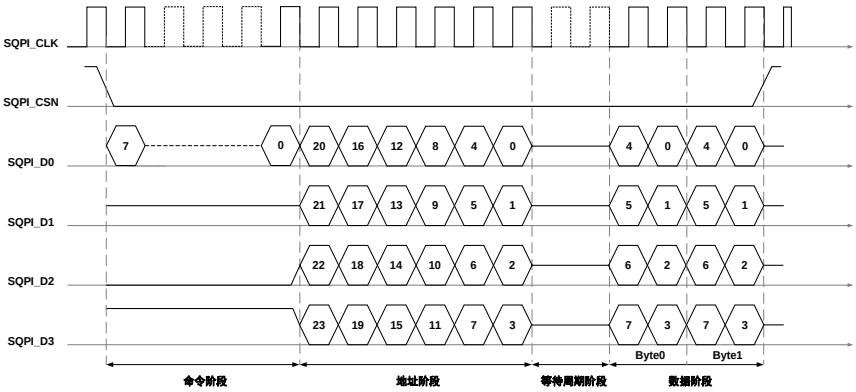


图 23-8. SQPI QQQ 模式时序(QPI)

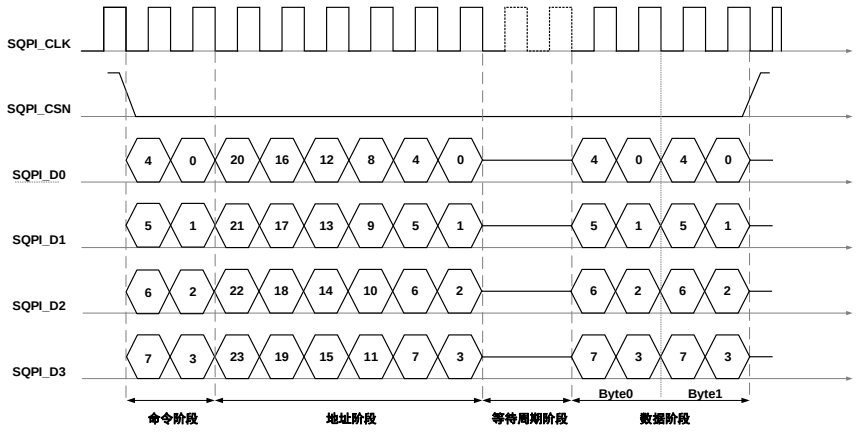


图 23-9. SQPI SSD 模式时序

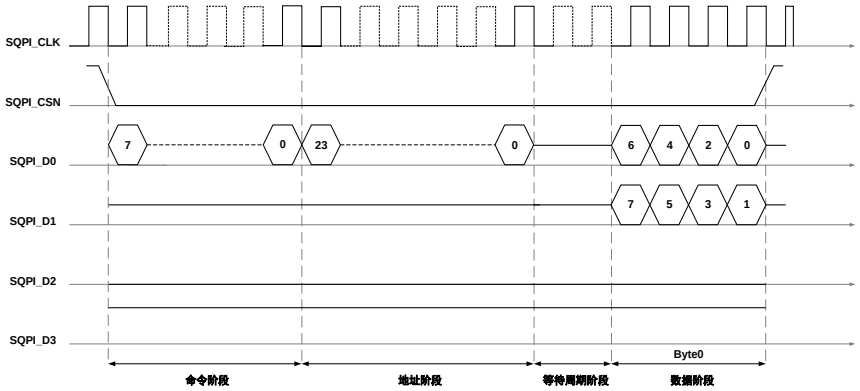
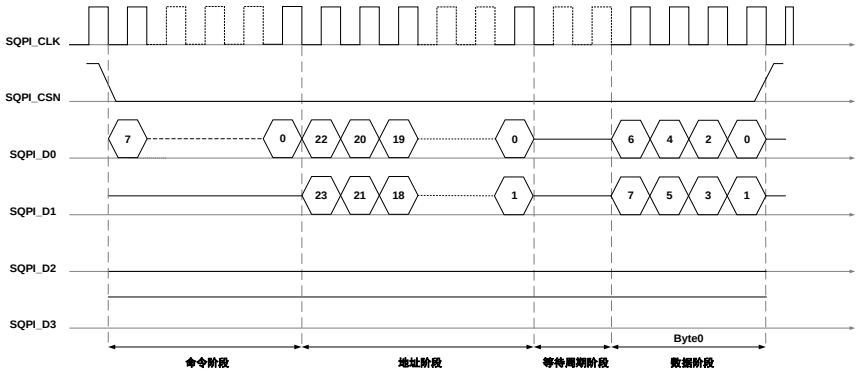


图 23-10. SQPI SDD 模式时序



23.4. SQPI 寄存器

SQPI 基地址：0xA000 1000

23.4.1. 初始化寄存器 (SQPI_INIT)

偏移地址：0x00

系统复位值：0x1805 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
SQPI_PL	SQPI_IDLEN[1:0]		SQPI_ADDRBIT[4:0]					SQPI_CLKDIV[5:0]						SQPI_CMDBIT[1:0]	
rw	rw		rw					rw						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	SQPI_PL	读数据采样极性 0：上升沿采样数据（默认）

		1：下降沿采样数据
30:29	SQPI_IDLEN[1:0]	外部存储器ID长度 00：64位 01：32位 10：16位 11：8 位
28:24	SQPI_ADDRBIT[4:0]	地址阶段的位数 默认：24 位
23:18	SQPI_CLKDIV[5:0]	SQPI时钟分频 0x0无效. 输出时钟频率等于 f_{hclk}/(SQPI_CLKDIV+1) 注意：当 SQPI_CLKDIV 为偶数时，输出时钟高电平比低电平多一个 AHB 时钟周期
17:16	SQPI_CMDBIT[1:0]	SQPI命令阶段的位数 00：4位 01：8位 (默认) 10：16位 11：保留
15:0	保留	必须保持复位值

23.4.2. 读命令寄存器 (SQPI_RCMD)

偏移地址： 0x04

系统复位值：0x0010 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
SQPI_RID	保留								SQPI_RMODE[2:0]			SQPI_RWAITCYCLE[3:0]			
rw									rw			rw			

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

位/位域	名称	描述
31	SQPI_RID	发送读 ID 命令，命令来自 SQPI_RCMD 位域
30:23	保留	必须保持复位值
22:20	SQPI_RMODE[2:0]	SQPI 读命令模式 000：SSQ 模式

001：SSS 模式
010：SQQ 模式
011：QQQ 模式
100：SSD 模式
101：SDD 模式

19:16	SQPI_RWAITCYCLE[3:0]	SQPI 在地址阶段之后的读命令等待周期个数
15:0	SQPI_RCMD[15:0]	用于 SQPI 读操作时的 AHB 总线发送的命令 当 SQPI_CMDBIT=00 时，SQPI_RCMD[3:0]有效 当 SQPI_CMDBIT=01 时，SQPI_RCMD[7:0]有效 当 SQPI_CMDBIT=10 时，SQPI_RCMD[15:0]有效

23.4.3. 写命令寄存器 (SQPI_WCMD)

偏移地址：0x08

系统复位：0x0010 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
SQPI_SCMD	保留								SQPI_WMODE [2:0]			SQPI_WWAITCYCLE [3:0]			
rs									rw			rw			

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

位/位域	名称	描述
31	SQPI_SCMD	发送没有地址阶段和数据阶段的特殊命令，命令来自 SQPI_WCMD.
30:23	保留	必须保持复位值
22:20	SQPI_WMODE[2:0]	SQPI 写命令模式: 000：SSQ 模式 001：SSS 模式 010：SQQ 模式 011：QQQ 模式 100：SSD 模式 101：SDD 模式
19:16	SQPI_WWAITCYCLE[3:0]	SQPI 在地址阶段之后的写命令等待周期个数
15:0	SPI_WCMD[15:0]	用于 SQPI 写操作时的 AHB 总线发送的命令

23.4.4. ID 低位寄存器 (SQPI_IDL)

偏移地址：0x0C

系统复位：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
SQPI_IDL [31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQPI_IDL [15:0]															
r															

位/位域	名称	描述
31:0	SQPI_IDL[31:0]	使用 SQPI 读 ID 命令时，返回的 ID 低位数据 当 SQPI_IDLEN=10 时，SQPI_IDL[15:0]有效 当 SQPI_IDLEN=11 时，SQPI_IDL[7:0]有效

23.4.5. ID 高位寄存器 (SQPI_IDH)

偏移地址： 0x10

系统复位：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
SQPI_IDH[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQPI_IDH[15:0]															
r															

位/位域	名称	描述
31:0	SQPI_IDH[31:0]	使用 SQPI 读 ID 命令时，返回的 ID 高位数据 只有当 SQPI_IDLEN = 00 时，该寄存器有效

24. 外部存储器控制器（EXMC）

24.1. 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，ROM，NOR Flash，NAND Flash，PC 卡。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。EXMC 的访问空间被划分为许多个块（Bank），每个块支持特定的存储器类型，用户可以通过对 Bank 的控制寄存器配置来控制外部存储器。

24.2. 主要特性

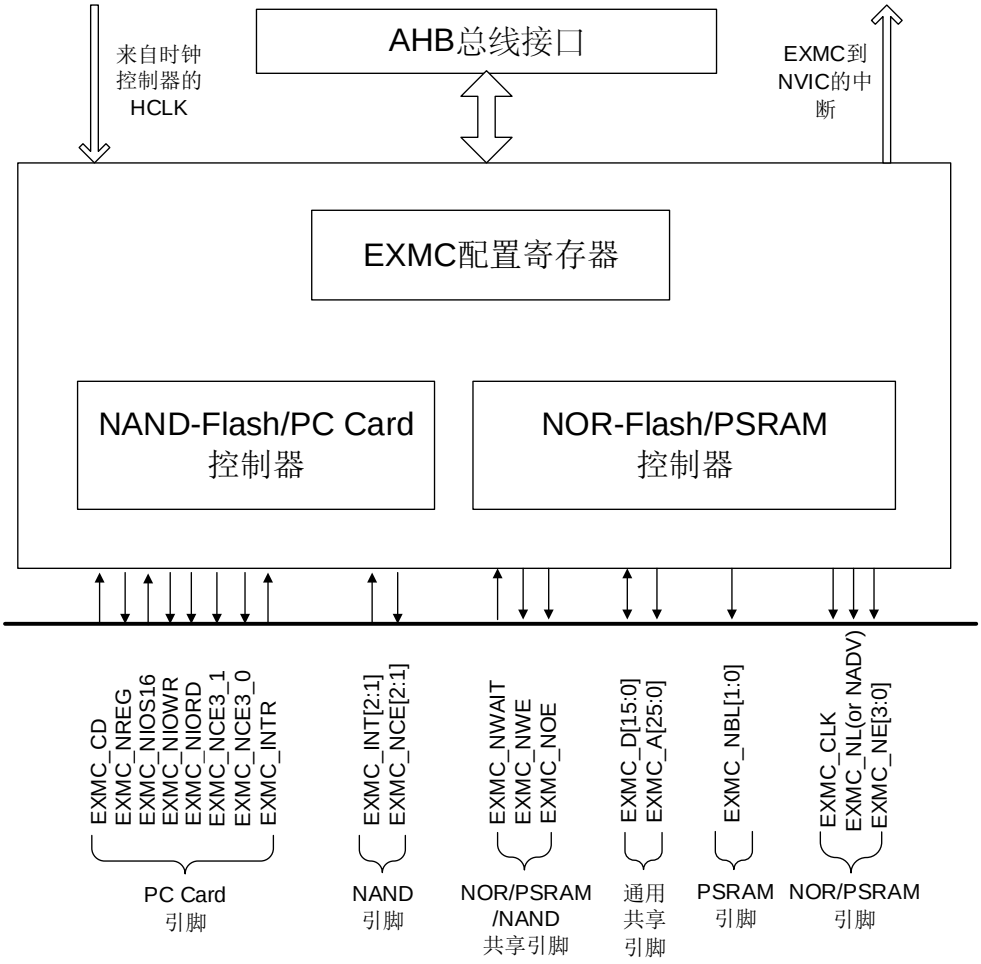
- 支持片外存储器类型：
  - SRAM；
  - PSRAM；
  - ROM；
  - NOR Flash；
  - 8 位或 16 位 NAND Flash；
  - 16 位 PC Card；
- AMBA 协议与各种片外存储器协议转换；
- 时序参数可编程可以满足用户特定需求；
- 每个 Bank 有独立的片选信号；
- 对于部分存储器类型支持独立的读写时序；
- 对于 NAND Flash 内置硬件 ECC；
- 支持 8 位，或 16 位总线带宽；
- NOR Flash 和 PSRAM 支持地址总线和数据总线的复用；
- 提供写使能和字节选择信号；
- 当 AMBA 总线宽度与外部存储器数据宽度不同时，会自动分割操作。

24.3. 功能描述

24.3.1. 结构框图

EXMC 由 5 个模块组成：AHB 总线接口，EXMC 配置寄存器，NOR/PSRAM 控制器，NAND/PC Card 控制器和外部设备接口。AHB 时钟（HCLK）是参考时钟。

图 24-1. 系统架构



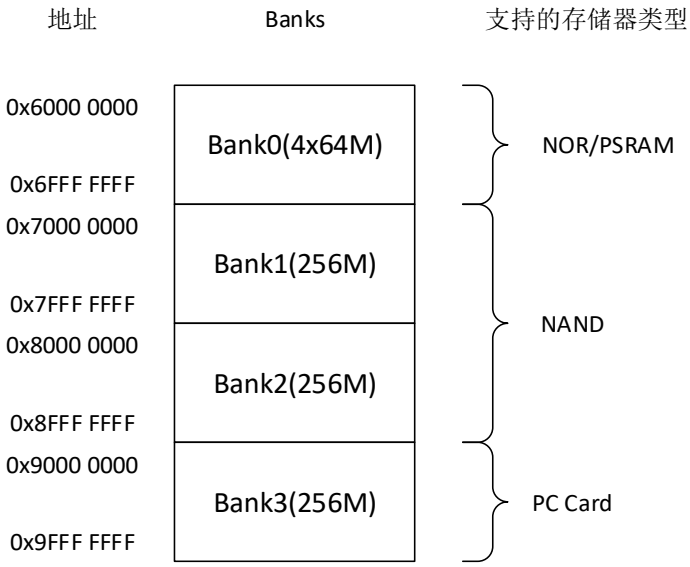
24.3.2. EXMC 访问基本规范

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。

24.3.3. 外部设备地址映射

图 24-2. EXMC Bank 划分



EXMC 将外部存储器分成多个 Bank，每个 Bank 占 256M 字节，其中 Bank0 又分为 4 个 Region，每个 Region 占 64M 字节。Bank1 和 Bank2 又都被分成 2 个 Section，分别是属性存储空间和通用存储空间。Bank3 分成 3 个 Section，分别是属性存储空间，通用存储空间和 I/O 存储空间。

每个 Bank 或 Region 都有独立的片选控制信号，也都能进行独立的配置。

Bank0 用于访问 NOR、PSRAM 设备。

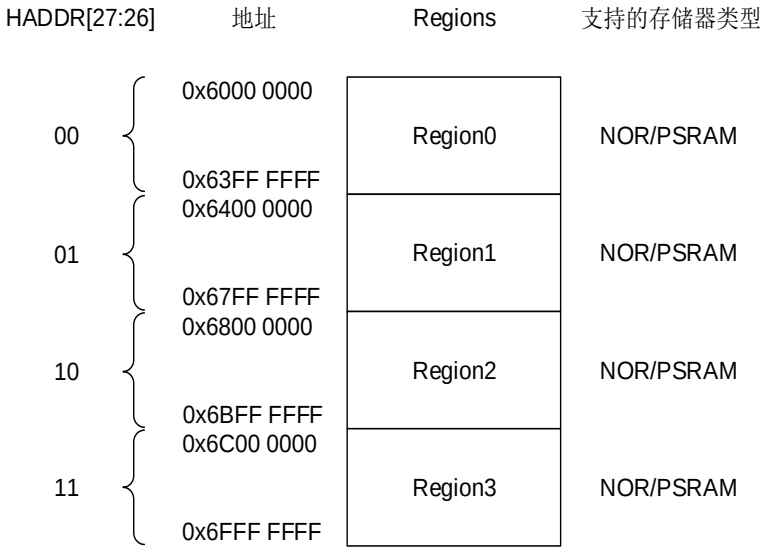
Bank1 和 Bank2 用于连接 NAND Flash，且每个 Bank 连接一个 NAND。

Bank3 用于连接 PC 卡。

NOR 和 PSRAM 的地址映射

*图24-3. Bank0地址映射*是Bank0四个Region的地址映射。AHB地址线HADDR[27:26]作为四个Region的片选信号。

图 24-3. Bank0 地址映射



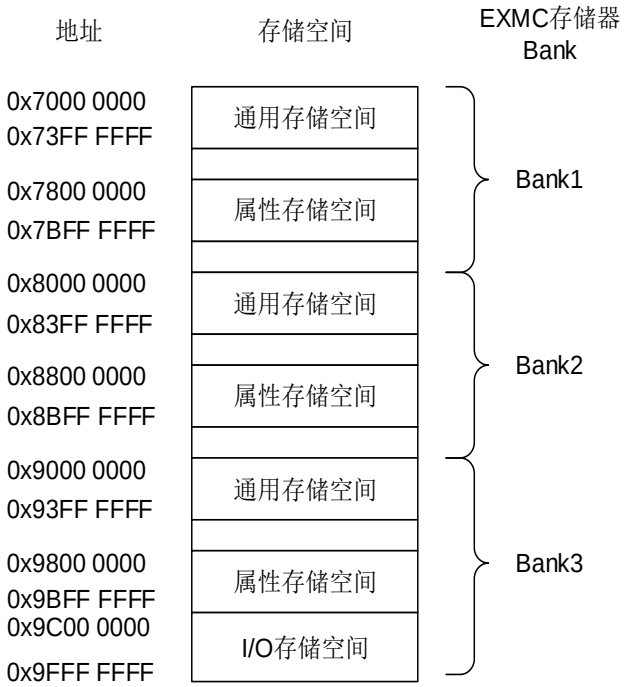
由于 HADDR[25:0]是字节地址，而外部存储器访问有可能不是按字节访问的，所以会出现地址不一致的情况，但 EXMC 能实现对 HADDR 的调整以适应外部存储器的数据宽度。具体规则如下：

- 如果外部存储器的数据宽度是 8 位按字节对齐，EXMC 内部将 HADDR[25:0]与 EXMC_A[25:0]相连，然后 EXMC_A[25:0]与外部存储器的地址线相连；
- 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。EXMC 内部将 HADDR[25:1]与 EXMC_A[24:0]相连，然后 EXMC_A[24:0]与外部存储器的地址线相连。

NAND/PC Card 地址映射

Bank1 和 Bank2 用来访问 NAND Flash，Bank3 用来访问 PC Card。每个 Bank 如*图 24-4. NAND/PC Card 地址映射*被分为多个存储空间。

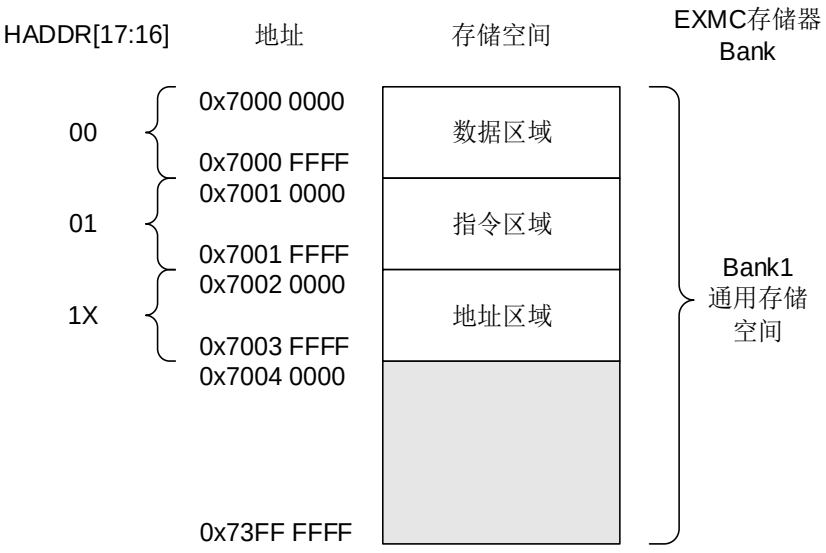
图 24-4. NAND/PC Card 地址映射



NAND 地址映射

对于 NAND Flash，通用和属性空间又可以细划分为 3 个区域。*图 24-5. Bank1 通用空间*为 Bank1 通用存储空间的数据区域，指令区域和地址区域的划分。

图 24-5. Bank1 通用空间



AHB 利用 HADDR[17:16]来实现对以上三个区的选择：

- HADDR[17:16]=00，即选择数据区；
- HADDR[17:16]=01，即选择指令区；
- HADDR[17:16]=1X，即选择地址区。

应用软件使用这 3 个区访问 NAND Flash。操作规则如下：

地址区：指定操作 NAND Flash 的地址，软件在地址区写入地址。在地址传输过程中，EXMC 会使能地址锁存信号（ALE），ALE 映射到 EXMC_A[17]。

指令区：指定 NAND Flash 将要执行的指令，软件在指令区写入指令。在指令传输过程中，EXMC 会使能命令锁存信号（CLE），CLE 映射到 EXMC_A[16]。

数据区：NAND Flash 读写数据，软件在数据区读出或写入数据。当 EXMC 在数据发送模式，软件需要在数据区写入数据，当 EXMC 在数据接收模式，软件需要在数据区读取数据。由于 NAND Flash 会自动累加其内部操作地址，故在读写时不需要软件修改操作地址。

24.3.4. NOR/PSRAM 控制器

EXMC 模块的 NOR/PSRAM 控制器控制 Bank0，它可以支持 NOR Flash、PSRAM、SRAM、ROM 和 CRAM 外部存储器。EXMC 对 Bank0 每个 Region 输出一个唯一的片选信号，NE[x](x=0..3)，用于在 4 个 Region 中进行片选，所有其他的信号都是共享的。每个 Region 都有专门的寄存器控制。

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 24-1. 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
Non-muxed EXMC_A[25:0]	输出	异步/同步	地址总线
Muxed 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[x]	输出	异步/同步	片选, x=0/1/2/3
EXMC_NOE	输出	异步/同步	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	地址有效

表 24-2. 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 引脚	传输方向	模式	功能描述
EXMC_NE[x]	输出	异步/同步	片选, x=0/1/2/3
EXMC_NOE	输出	异步/同步	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	地址锁存信号
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

*表24-3. EXMC的Bank0支持的所有处理*列出了EXMC对NOR，PSRAM和SRAM支持的访问模式。

表 24-3. EXMC 的 Bank0 支持的所有处理

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 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 NBL[1:0]
	同步	W	16	16	
	同步	W	32	16	分成 2 次 EXMC 访问
SRAM and 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 NBL[1:0]
	异步	W	16	8	

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
	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 24-4. 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	同步	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 24-5. 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 ASET	DSET ASET
	模式 2	0	NOR Flash	DSET ASET	DSET ASET
	模式 A	1	SRAM/PSRAM/CRAM 在数据阶段 EXMC_NOE 翻转	WDSET WASET	DSET ASET
	模式 B	1	NOR Flash	WDSET WASET	DSET ASET
	模式 C	1	NOR Flash 在数据阶段 EXMC_NOE 翻转	WDSET WASET	DSET ASET
	模式 D	1	有地址保持功能	WDSET WAHLD WASET	DSET AHLD ASET
	模式 AM	0	NOR Flash 数据/地址复用	DSET AHLD ASET BUSLAT	DSET AHLD ASET BUSLAT
同步	模式 E	0	NOR/PSRAM/CRAM 同步读 PSRAM/CRAM 同步写	DLAT CKDIV	DLAT CKDIV
	模式 SM	0	NOR Flash 数据/地址复用	DLAT CKDIV	DLAT CKDIV

如*表 24-5. 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 24-4. 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_SNCTLx 位 EXMODEN 置 1 使能扩展模式后，可以通过寄存器 EXMC_SNTCFGx 和 EXMC_SNWTCFGx 将读写配置成独立的时序。

异步访问时序

模式 1 – SRAM/CRAM

图 24-6. 模式 1 读时序

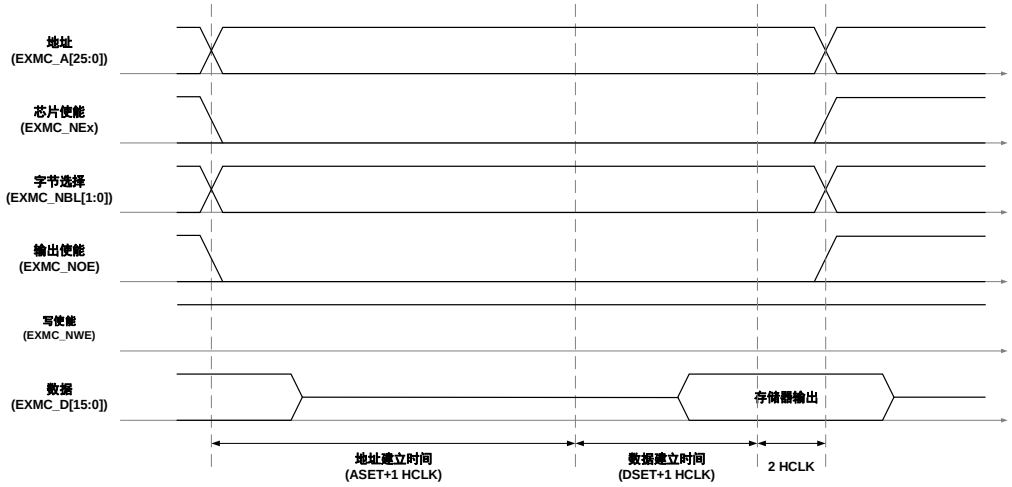


图 24-7. 模式 1 写时序

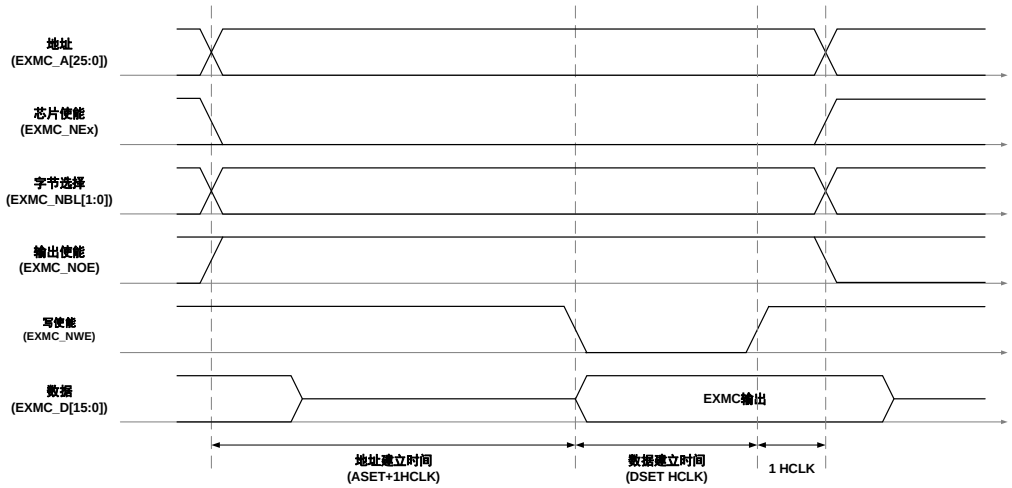


表 24-6. 模式 1 相关寄存器配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
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，Nor Flash：2
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFGx		
31-30	保留	0x0000
29-28	ASYNCMOD	无影响
27-24	DLAT	无影响
23-20	CKDIV	无影响
19-16	BUSLAT	EXMC_NE[x]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 24-8. 模式 A 读时序

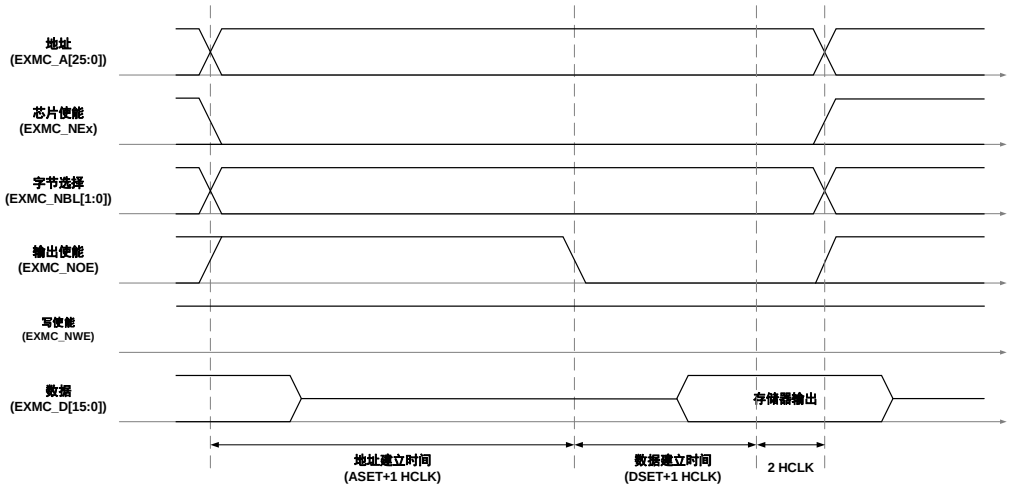
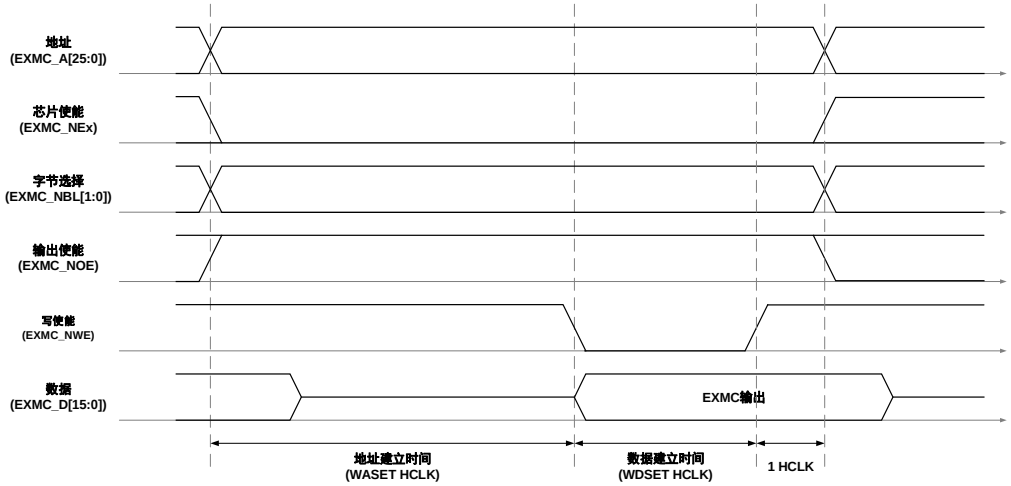


图 24-9. 模式 A 写时序



模式 A 和模式 1 的区别在于写时序，当两个模式的寄存器有相同的时序配置时，模式 A 的写时序独立于读时序。

表 24-7. 模式 A 相关寄存器配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
31-20	保留	0x000
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWAIT	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	无影响
5-4	NRW	取决于存储器
3-2	NRTP	取决于存储器，Nor Flash：2
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFGx(Read)		
31-30	保留	0x0
29-28	ASYNCMOD	0x0
27-24	DLAT	无影响
23-20	CKDIV	无影响
19-16	BUSLAT	EXMC_NE[x]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)

位域/位	名称	参考设定值
EXMC_SNCTLx		
7-4	AHLD	无影响
3-0	ASET	取决于存储器与用户
EXMC_SNWTCFGx(Write)		
31-30	保留	0x0
29-28	WASYNCMOD	0x0
27-20	保留	0x00
19-16	WBUSLAT	EXMC_NE[x]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 24-10. 模式 2/B 读时序

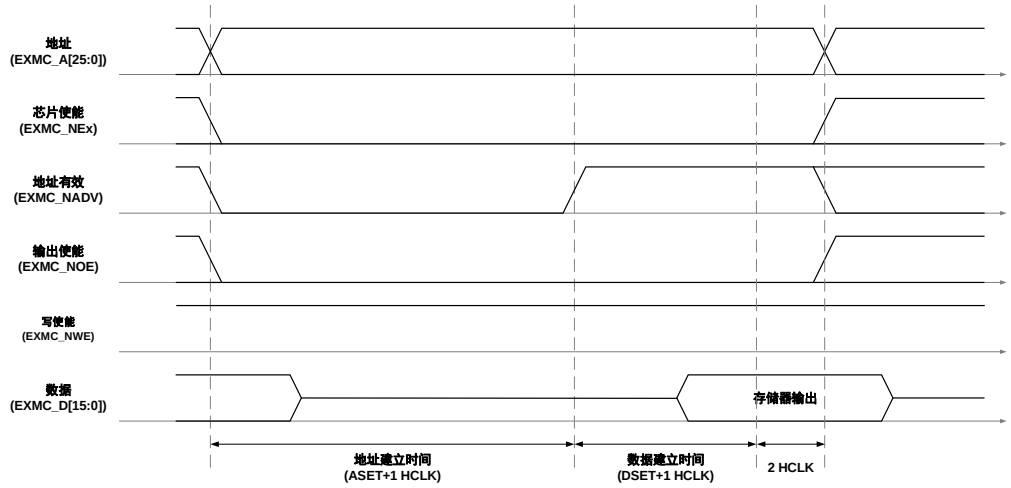


图 24-11. 模式 2 写时序

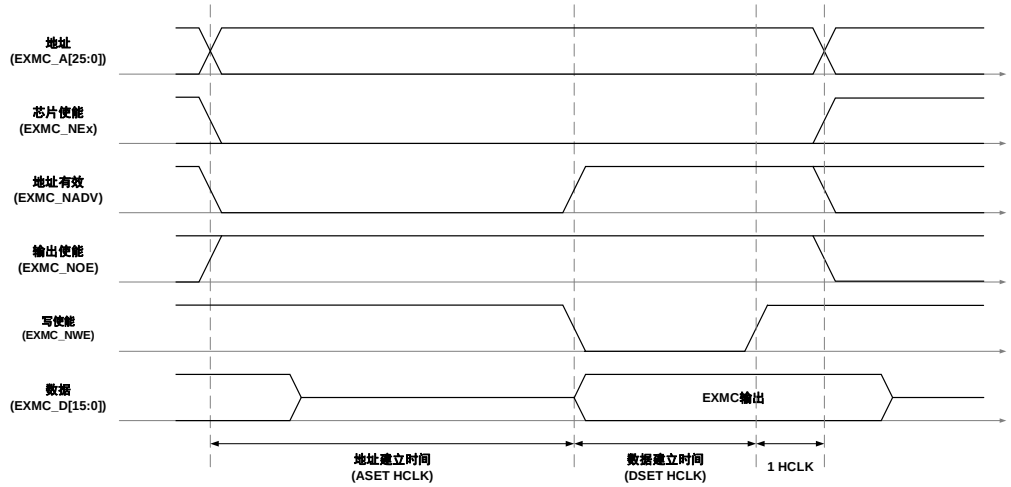


图 24-12. 模式 B 写时序

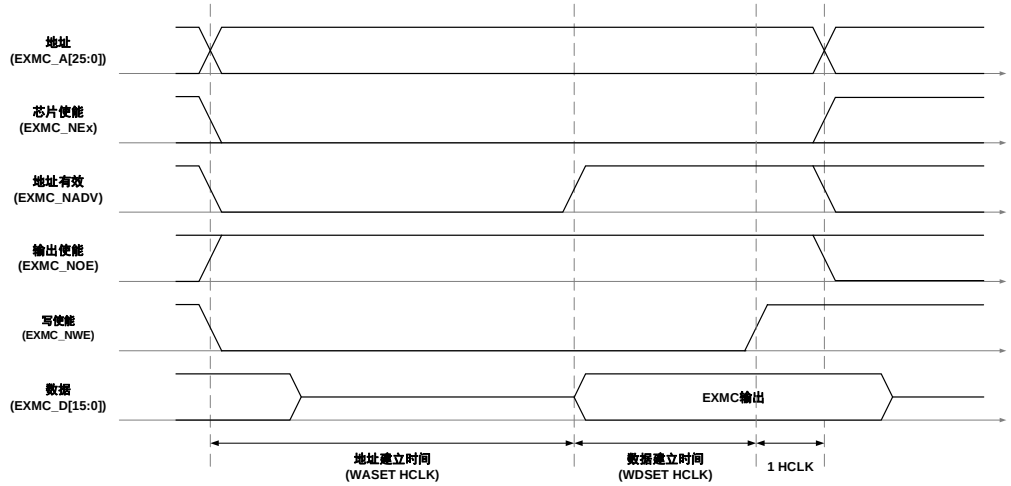


表 24-8. 模式 2/B 相关寄存器配置

位域/位	名称	参考设定值
EXMC_SNCTLx(模式 2，模式 B)		
31-20	保留	0x000
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWAIT	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
3-2	NRTP	Nor Flash：2
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFGx(模式 2 读/写操作，模式 B 读操作)		
31-30	保留	0x0000
29-28	ASYNCMOD	模式 B：0x1
27-24	DLAT	无影响
23-20	CKDIV	无影响
19-16	BUSLAT	EXMC_NE[x]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_SNWTCFGx(模式 B 写操作)		

位域/位	名称	参考设定值
EXMC_SNCTLx(模式 2，模式 B)		
31-30	保留	0x0000
29-28	WASYNCMOD	模式 B：0x1
27-20	保留	0x000
19-16	WBUSLAT	EXMC_NE[x]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 24-13. 模式 C 读时序

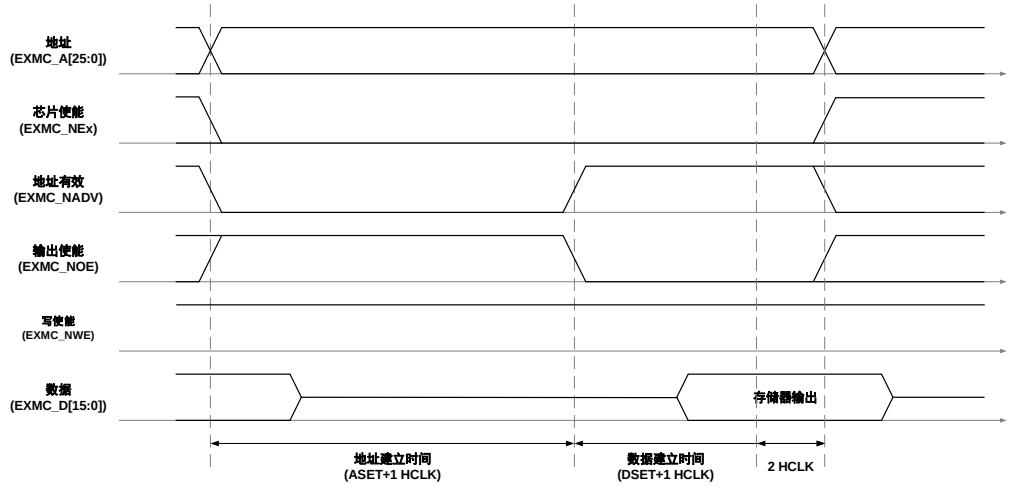
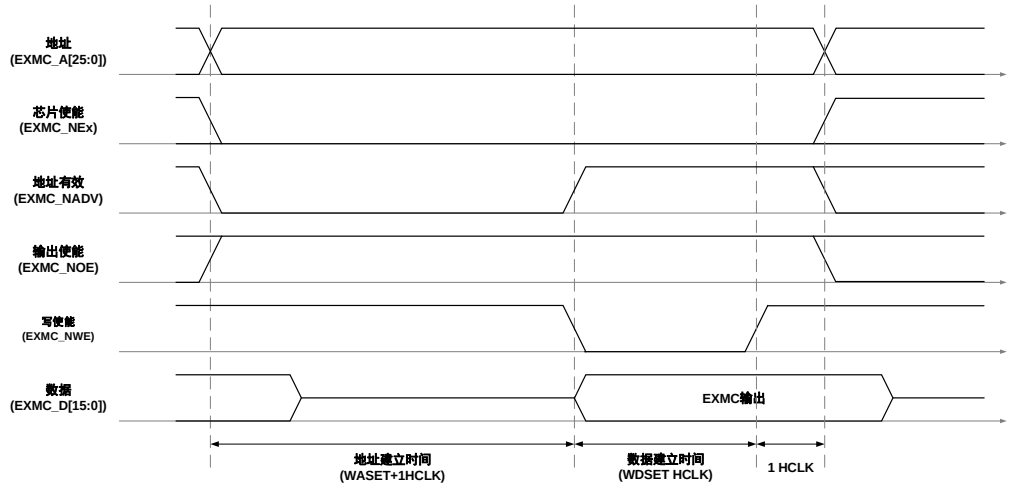


图 24-14. 模式 C 写时序



模式 C 和模式 1 的区别在于写时序，当两个模式的寄存器有相同的时序配置时，模式 C 的写时序独立于读时序，并且他们的 NOE 和 NADV 翻转变化也是不同的。

表 24-9. 模式 C 相关寄存器配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
31-20	保留	0x000
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWAIT	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：2
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFGx		
31-30	保留	0x0000
29-28	ASYNCMOD	模式 C：0x2
27-24	DLAT	无影响
23-20	CKDIV	无影响
19-16	BUSLAT	EXMC_NE[x]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_SNWTCFGx		
31-30	保留	0x0
29-28	WASYNCMOD	模式 C：0x2
27-20	保留	0x000
19-16	WBUSLAT	EXMC_NE[x]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 24-15. 模式 D 读时序

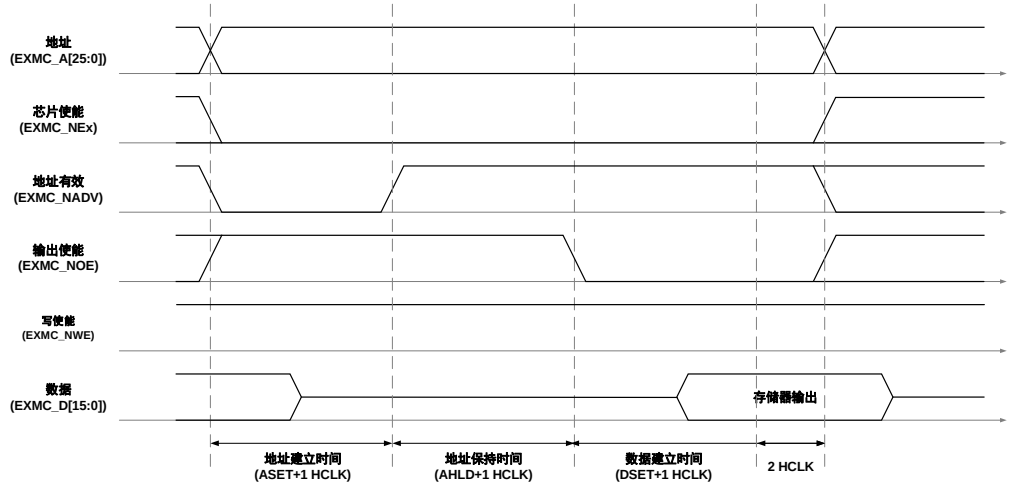


图 24-16. 模式 D 写时序

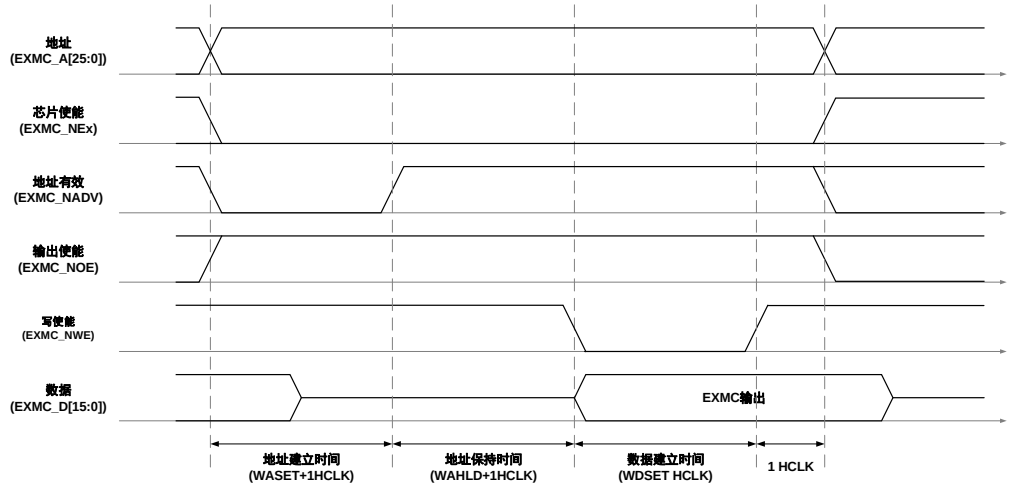


表 24-10. 模式 D 相关寄存器配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
31-20	保留	0x000
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWAIT	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_SNTCFGx		
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	EXMC_NE[x]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_SNWTCFGx		
31-30	保留	0x0
29-28	WASYNCMOD	模式 D：0x3
27-20	保留	0x00
19-16	WBUSLAT	EXMC_NE[x]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 WSET+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 24-17. 复用模式读时序

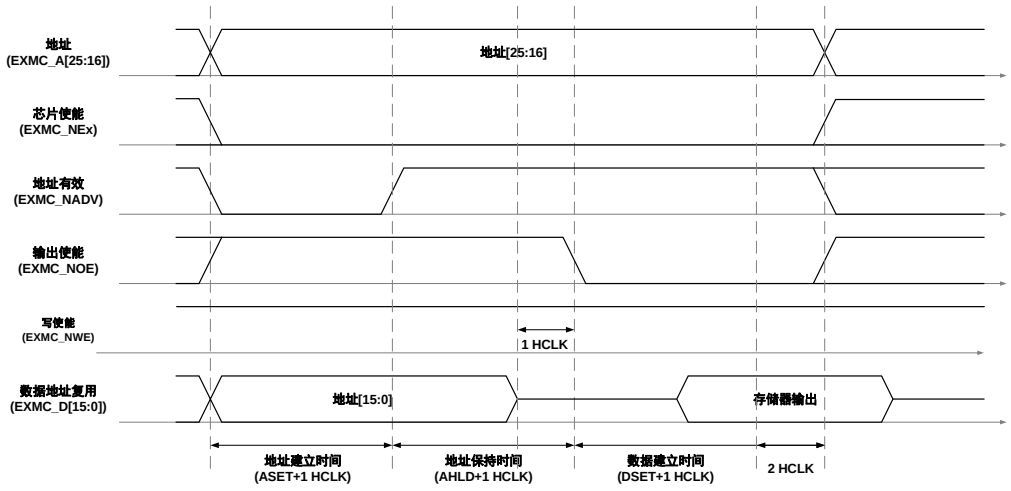


图 24-18. 复用模式写时序

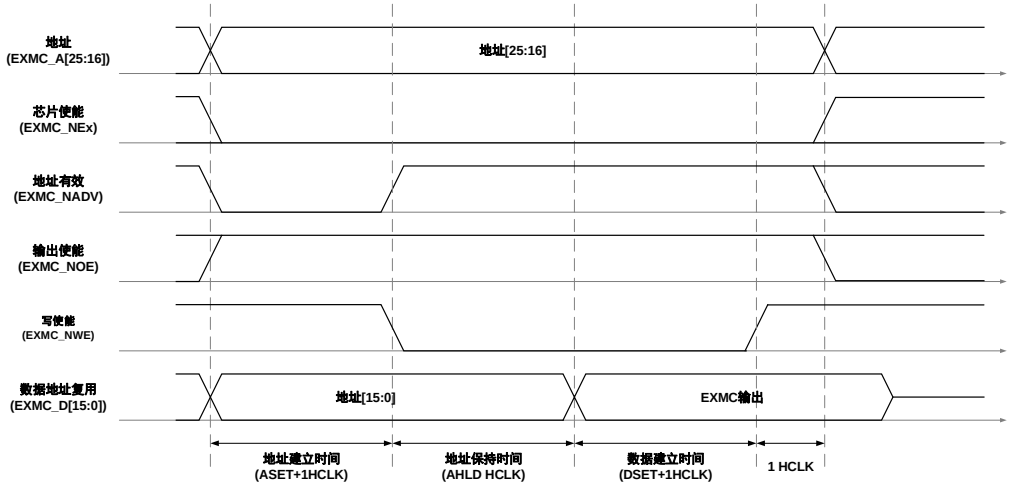


表 24-11. 复用模式相关寄存器配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
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
3-2	NRTP	0x2：NOR Flash
1	NRMUX	0x1
0	NRBKEN	0x1
EXMC_SNTCFGx		
31-30	保留	0x0
29-28	ASYNCMOD	0x0
27-24	DLAT	无影响
23-20	CKDIV	无影响
19-16	BUSLAT	EXMC_NE[x]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

异步通信的等待时间：

等待功能由寄存器 EXMC_SNCTLx 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：

$T_{DATA_SETUP} \geqslant maxT_{WAIT_ASSERTION}+4HCLK$ (25-1)

若存储器等待信号与 EXMC_NE 信号对齐：

如果

$maxT_{WAIT_ASSERTION} \geqslant T_{ADDRES_PHASE}+ T_{HOLD_PHASE}$ (25-2)

则

$T_{DATA_SETUP} \geqslant (maxT_{WAIT_ASSERTION}-T_{ADDRES_PHASE}-T_{HOLD_PHASE})+4HCLK$ (25-3)

否则

$T_{DATA_SETUP} \geqslant 4HCLK$ (25-4)

图 24-19. 异步等待有效时的读时序

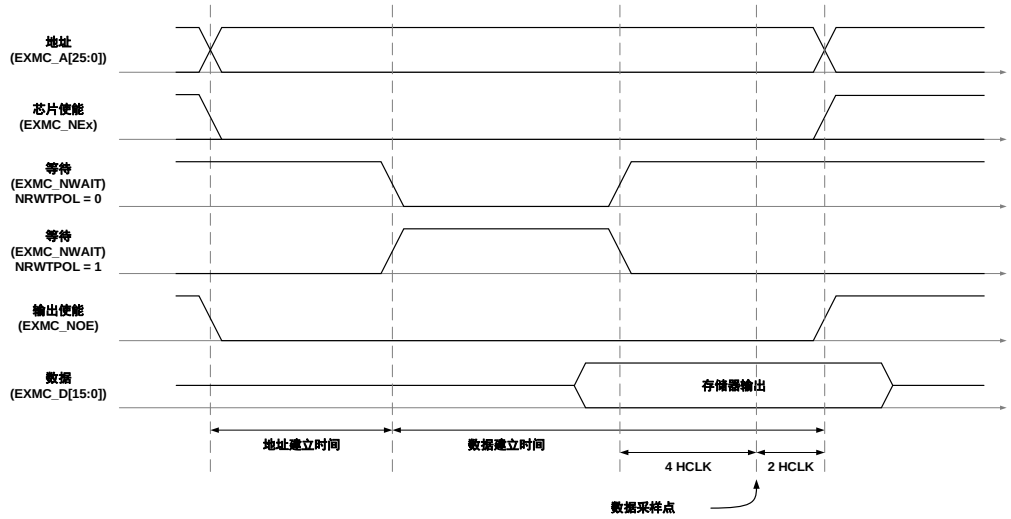
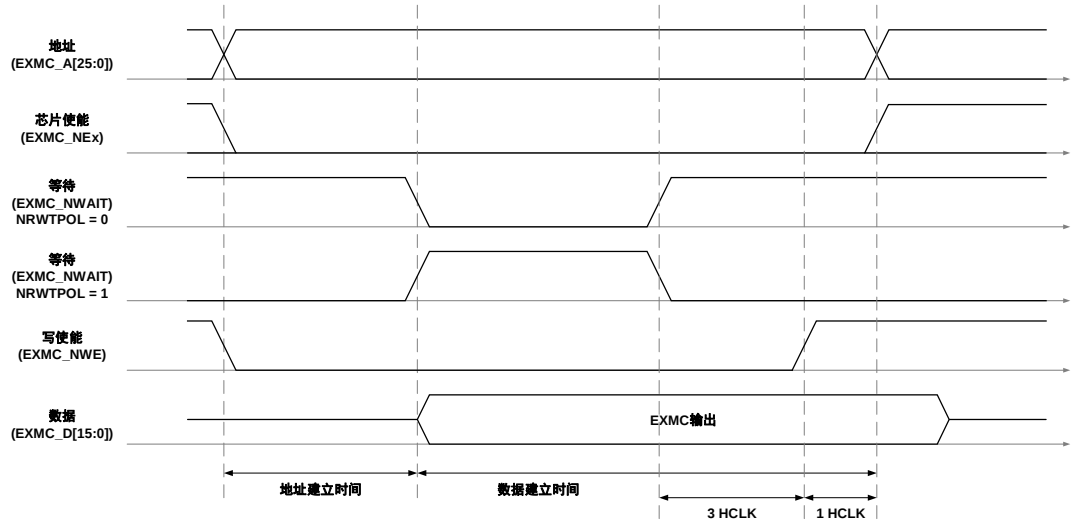


图 24-20. 异步等待有效时的写时序



同步访问时序

同步访问模式中，存储器时钟（EXMC_CLK）与系统时钟（HCLK）关系如下：

$$EXMC_CLK=\frac{HCLK}{CKDIV+1} \quad (25\text{-}5)$$

其中 CKDIV 是同步时钟分频比，通过配置寄存器 EXMC_SNTCFGx 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：

NOR 闪存延迟不包含 NADV，二者之间的关系为：

NOR 闪存延迟=DLAT+2 (25-6)

NOR 闪存延迟包含 NADV，二者之间的关系为：

NOR 闪存延迟=DLAT+3 (25-7)

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_SNCTLx 来设置，位 NRWTEN 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_SNCTLx 位 NRWTEN 置 1，在数据延迟之后检测到 EXMC_NWAIT 信号。如果 EXMC_NWAIT 有效，在 EXMC_NWAIT 无效之前会一直插入等待时钟。

- 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_SNCTLx 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 24-3. EXMC 的 Bank0 支持的所有处理*。

同步复用突发读时序 – NOR,PSRAM(CRAM)

图 24-21. 同步复用突发传输读时序

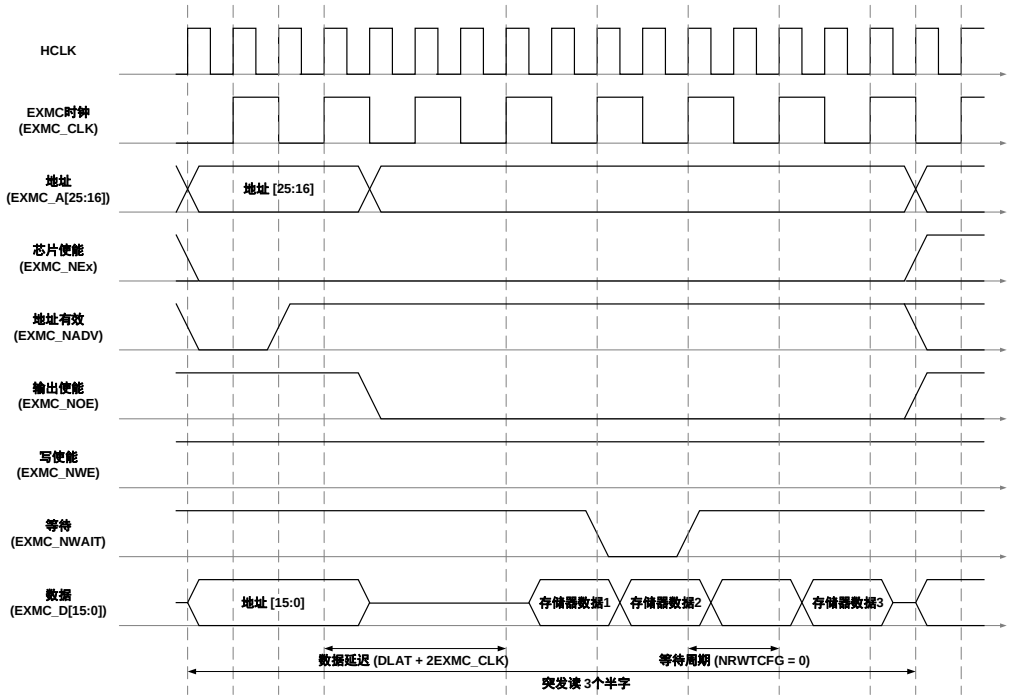


表 24-12. 同步复用模式读时序配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
31-20	保留	0x000
19	SYNCWR	无影响

位域/位	名称	参考设定值
18-16	CPS	0x0
15	ASYNCWAIT	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_SNTCFGx(Read)		
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	EXMC_NE[x]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

同步复用突发写时序 – NOR,PSRAM(CRAM)

图 24-22. 同步复用突发传输写时序

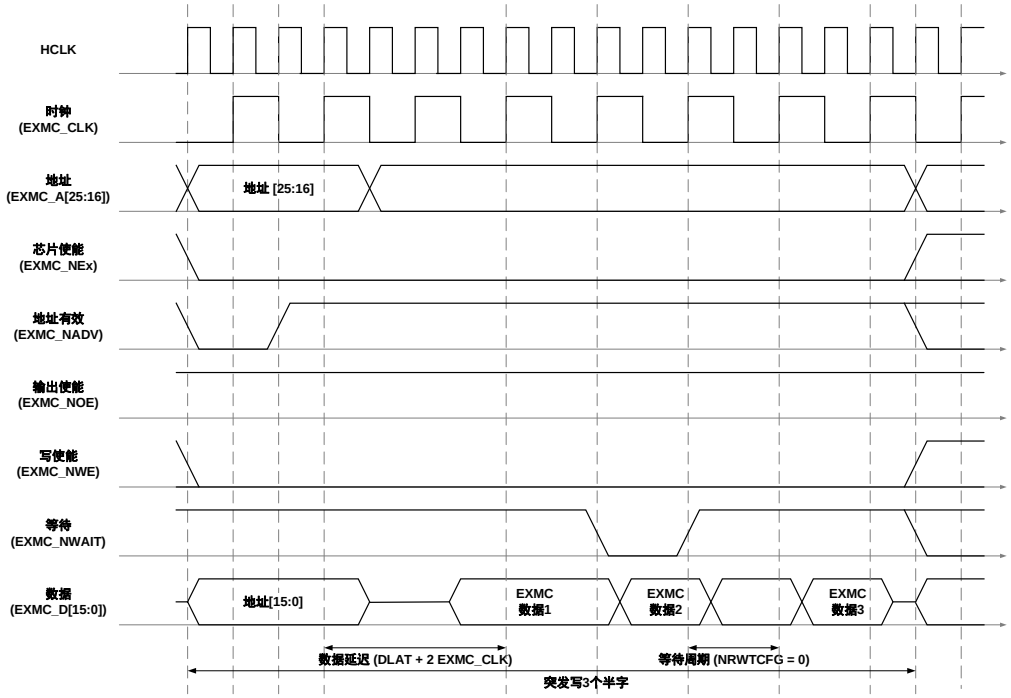


表 24-13. 同步复用模式写时序配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
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
1	NRMUX	0x1，取决于用户
0	NRBKEN	0x1
EXMC_SNTCFGx(Write)		
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	EXMC_NE[x]上升沿到下降沿的时间

位域/位	名称	参考设定值
15-8	DSET	无影响
7-4	AHLD	无影响
3-0	ASET	无影响

24.3.5. NAND Flash 或 PC Card 控制器

EXMC 模块 Bank1、Bank2 支持 NAND Flash，Bank3 支持 PC Card 设备。对于每个 Bank，EXMC 提供独立的寄存器来配置访问时序，支持 8 位、16 位的 NAND Flash 以及 16 位 PC 卡。对于 NAND Flash，EXMC 还提供 ECC 计算模块，保证数据传输和保存的鲁棒性。

NAND Flash/PC Card 接口功能

表 24-14. 8 位/16 位 NAND 接口信号描述

EXMC 引脚	传输方向	功能描述
EXMC_A[17]	输出	NAND Flash 地址锁存（ALE）
EXMC_A[16]	输出	NAND Flash 命令锁存（CLE）
EXMC_D[7:0]/ EXMC_D[15:0]	输入/输出	8 位复用，双向地址/数据总线
		16 位复用，双向地址/数据总线
EXMC_NCE[x]	输出	片选, x = 1,2
EXMC_NOE(NRE)	输出	输出使能
EXMC_NWE	输出	写使能
EXMC_NWAIT/ EXMC_INT[x]	输入	NAND Flash 就绪/忙输入信号 EXMC，x=1,2

表 24-15. 16 位 PC Card 接口信号描述

EXMC 引脚	传输方向	功能描述
EXMC_A[10:0]	输出	地址总线
EXMC_NIOS16	输入	仅适合 16 位传输的 I/O 空间的数据传输宽度(必须接地)
EXMC_NIORD	输出	I/O 空间输出使能
EXMC_NIOWR	输出	I/O 空间写使能
EXMC_NREG	输出	决定访问通用空间还是属性空间
EXMC_D[15:0]	输入/输出	双向数据总线
EXMC_NCE3_x	输出	片选(x=0,1)
EXMC_NOE	输出	输出使能
EXMC_NWE	输出	写使能
EXMC_NWAIT	输入	PC Card 等待信号
EXMC_INTR	输入	PC Card 中断输入信号
EXMC_CD	输入	PC Card 卡存在检测信号，高有效

支持的存储器访问模式

表 24-16. Bank1/2/3 支持的访问模式

存储器	模式	读/写	AHB	注释

			传输宽度	
8 位 NAND	异步	R	8	
	异步	W	8	
	异步	R	16	分成 2 次 EXMC 访问
	异步	W	16	
	异步	R	32	分成 4 次 EXMC 访问
	异步	W	32	
16 位 NAND/PC Card	异步	R	8	
	异步	W	8	不支持此操作
	异步	R	16	
	异步	W	16	
	异步	R	32	分成 2 次 EXMC 访问
	异步	W	32	

NAND Flash/PC Card 的控制时序

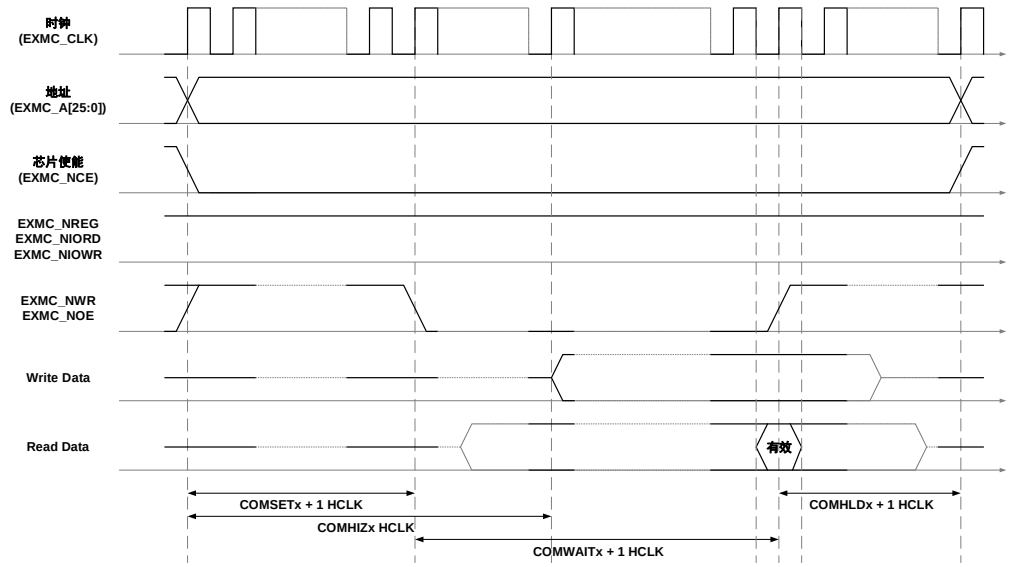
EXMC 能够为 NAND Flash、PC 卡等设备产生合适的时序信号。每个 Bank 都有相应的寄存器来对外部存储器进行管理和控制，EXMC_NPCTLx、EXMC_NPINTENx、EXMC_NPCTCFGx、EXMC_NPATCFGx、EXMC_PIOTCFG3、EXMC_NECCx，其中寄存器 EXMC_NPINTENx、EXMC_NPCTCFGx、EXMC_NPATCFGx 都可以配置 4 个时序参数，可以根据用户需求和外部存储器的特性来进行相应的配置。

表 24-17. NADN/PC Card 可编程参数

参数	读/写	单位	功能描述	NAND Flash/ PC Card	
				最小值	最大值
存储器数据总线高阻时间（HIZ）	W/R	HCLK	启动写操作之后保持数据总线为高阻态的时间	0	255
存储器保持时间（HLD）	W/R	HCLK	在发送命令结束后保持地址的(HCLK)时钟周期数目，写操作时也是数据的保持时间	1	254
存储器等待时间（WAIT）	W/R	HCLK	发出命令的最短持续时间(HCLK)时钟周期数目	2	256
存储器建立时间（SET）	W/R	HCLK	发出命令之前建立地址的(HCLK)时钟周期数目	1	255

*图 24-23. PC Card 通用空间操作时序*给出了在通用存储空间中操作的可编程参数定义，属性存储空间和 I/O 空间(只适用于 PC Card)中操作与此相似。

图 24-23. PC Card 通用空间操作时序



NAND Flash 操作

EXMC 在对 NAND Flash 发送命令或地址时，需要利用其命令锁存信号（EXMC_A[16]）或地址锁存信号（EXMC_A[17]）这两条地址线，即 CPU 需要在特定的地址进行写操作。

示例：NAND Flash 读操作步骤：

1) 配置 EXMC_NPCTLx、EXMC_NPCTCFGx，若需要预等待功能，还需配置 EXMC_NPATCFGx；

2) 往通用空间写入 NAND Flash 读数据命令，即在 EXMC_NCE 和 EXMC_NWE 有效期间，EXMC_CLE（EXMC_A[16]）变为有效电平（高），则被 NAND 认为写入命令；

3) 往通用空间写入读操作的起始地址，即在 EXMC_NCE 和 EXMC_NWE 有效期间，EXMC_ALE（EXMC_A[17]）变为有效电平（高），则被 NAND 认为写入地址；

4) 等待 NAND 就绪信号，NAND 控制器会在这期间将和 EXMC_NCE 一直保持有效；

5) 从通用空间的数据区逐字节的读出数据；

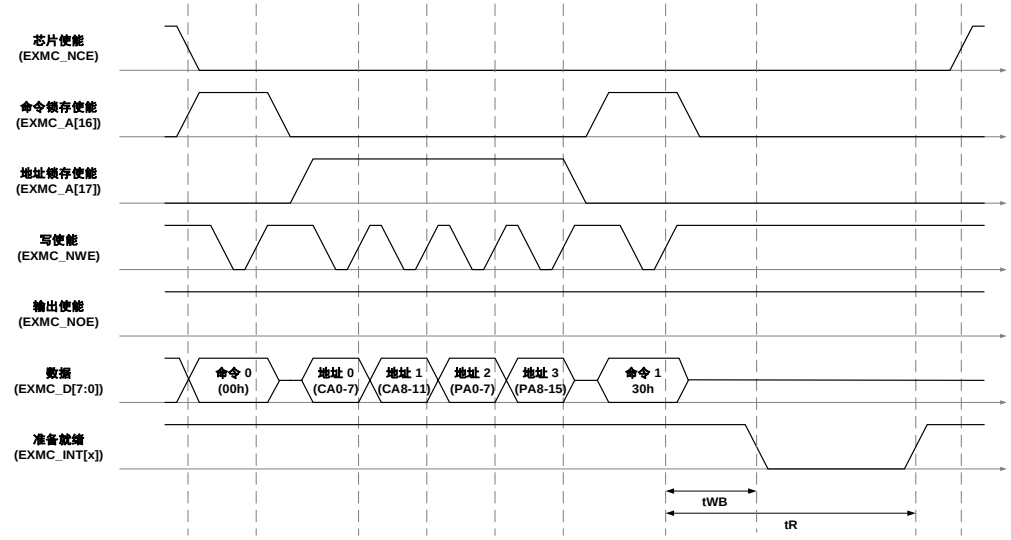
6) 在不写入新的命令和地址，可以自动读出 NAND 下一页数据；或转到 3）写入新的地址进行下一页的读取；或转到 2）写入新的命令和地址。

NAND Flash 预等待功能

某些 NAND Flash 要求在输入最后一个地址字节后，控制器等待 NAND Flash 就绪，并且还有一些对 EXMC_NCE 敏感型的 NAND Flash 还要求在其就绪前 NCE 必须保持有效。

下面以 TOSHIBA128M*8bit NADN Flash 为例：

图 24-24. NCE 敏感 NAND Flash 访问时序



1) 往 NAND 的通用空间命令区写入命令 CMD0

2) 往 NAND 的通用空间地址区写入操作地址 ADD0

3) 往 NAND 的通用空间地址区写入操作地址 ADD1

4) 往 NAND 的通用空间地址区写入操作地址 ADD2

5) 往 NAND 的通用空间地址区写入操作地址 ADD3

6) 往 NAND 的属性空间命令区写入命令 CMD1

在 6)中写命令操作，EXMC 使用的是寄存器 EXMC_NPATCFGx 定义的时序。经过 ATTHLD 时间后，NAND Flash 等待 EXMC_INTx 信号，ATTHLD 要大于 t_{WB}（EXMC_NWE 高到 EXMC_INTx 低）。对于那些对 EXMC_NCE 敏感的 NAND Flash，在地址字节之后的第一个地址字节输入后，要求片选信号 NCE 一直保持低电平，直到 EXMC_INTx 由低到高。这里可以通过配置属性存储空间的 ATTHLD 的值来满足 t_{WB} 的时序，这样 CPU 只有在地址字节之后写入第一个命令字节时才使用属性存储空间的时序，而在其他时候都使用通用存储空间的时序。

NAND Flash 的 ECC 计数模块

EXMC 模块中的 Bank1 和 Bank2 各有一个 ECC 计算的硬件模块，用户可以根据 EXMC_NPCTLx 中的 ECCSZ 来选择 ECC 计算的页面大小，通过 ECC 计算可以矫正 1 个 bit 的错误并且能检测 2 个 bit 的错误。

当 NAND 存储器块使能，ECC 模块就会检测 EXMC_D[15:0]以及 EXMC_NCE、EXMC_NWE 信号。当已经完成 ECCSZ 大小字节的读写操作时，软件必须读出 EXMC_NECCx 中的结果值。如果需要再次开始 ECC 计算，软件需要先将 EXMC_NPCTLx 中 ECCEN 清 0 来清除 EXMC_NECCx 中的值，再将 ECCEN 置 1 来重新启动 ECC 计算。

PC/CF Card 访问

EXMC 的 Bank3 用来访问 PC/CF Card，同时支持存储器和 IO 模式。Bank3 分为 3 个子空间，分别为存储空间，属性空间和 IO 空间。

EXMC_NCE3_0 和 EXMC_NCE3_1 是字节选择信号，当仅有 EXMC_NCE3_0 有效时，低字

节或高字节的选择取决于 EXMC_A[0]，当仅有 EXMC_NCE3_1 有效时，硬件不支持，当 EXMC_NCE3_0 和 EXMC_NCE3_1 都有效时，16 位操作。复位 NDTP 来选择 PC/CF Card 作为外部存储器，寄存器 EXMC_NPCTLx 位 NDW 必须设置为 01 来保证 EXMC 的正确操作。

下面是对不同空间的访问：

1. 通用空间：EXMC_NCE3_x(x= 0,1)是片选信号，表示同时支持 8 位和 16 位的访问操作。在 EXMC_NREG 为高电平时，EXMC_NWE 为低电平时表明正在执行写操作，EXMC_NOE 为低电平时表明正在执行读操作。
2. 属性空间：EXMC_NCE3_x(x= 0,1)是片选信号，表示同时支持 8 位和 16 位的访问操作。在 EXMC_NREG 为低电平时，EXMC_NWE 为低电平时表明正在执行写操作，EXMC_NOE 为低电平时表明正在执行读操作。
3. IO 空间：EXMC_NCE3_x(x= 0,1)是片选信号，表示同时支持 8 位和 16 位的访问操作。在 EXMC_NREG 为低电平时，EXMC_NIOWR 为低电平时表明正在执行写操作，EXMC_NIORD 为低电平时表明正在执行读操作。

AHB 访问 16 位的 PC/CF Card：

1. 通用空间：数据存储的位置，支持字节和半字访问，奇地址禁止字节访问。当 AHB 进行字访问，EXMC 会自动分成两次连续的半字操作。在 EXMC_NREG 为高电平时，EXMC_NWE 为低电平时写操作，EXMC_NOE 为低电平时读操作。
2. 属性空间：配置信息存储的位置，仅偶地址支持字节访问，半字访问会被转换为单次字节操作，字访问会被转换为两次字节访问。半字与字访问时，只有 EXMC_NCE3_0 有效。在 EXMC_NREG 为低电平时，EXMC_NWE 为低电平时写操作，EXMC_NOE 为低电平时读操作。
4. IO 空间：同时支持字节和半字访问，EXMC_NREG 为低电平时，EXMC_NIOWR 为低电平时写操作，EXMC_NIORD 为低电平时读操作。

24.4. EXMC 寄存器

EXMC 基地址：0xA000 0000

24.4.1. NOR/PSRAM 控制器寄存器

SRAM/NOR Flash 控制寄存器 (EXMC_SNCTLx) (x=0, 1, 2, 3)

偏移地址：0x00 + 8 * x, (x = 0, 1, 2, 3)
复位值：0x0000 30DX

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 0：异步写操作 1：同步写操作
18:16	CPS[2:0]	CRAM页大小 000：页边界自动突发分割 001：128字节 010：256字节 011：512字节 100：1024字节 其他：保留
15	ASYNCWAIT	异步等待功能使能位 0：禁用异步等待功能 1：使能异步等待功能
14	EXMODEN	扩展模式使能 0：禁用扩展模式 1：使能扩展模式
13	NRWTEN	NWAIT信号使能 对于存储器的突发模式访问，该位使能/禁用等待状态插入NWAIT信号功能 0：禁用NWAIT信号

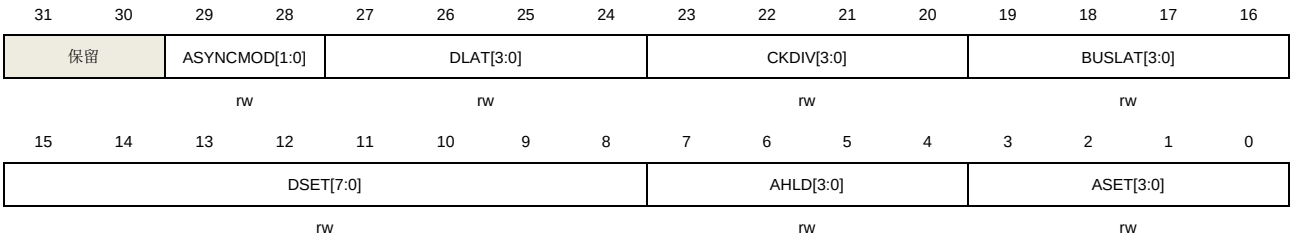
		1：使能NWAIT信号
12	WREN	写操作使能 0：禁止EXMC对外部存储器的写操作，否则产生一个AHB错误 1：允许EXMC对外部存储器的写操作（复位缺省值）
11	NRWTCFG	NWAIT信号配置，只在同步模式有效 0：NWAIT信号在等待状态前的一个数据周期有效 1：NWAIT信号在等待状态期间有效
10	WRAPEN	非对齐成组模式使能 0：禁止非对齐成组操作 1：允许非对齐成组操作
9	NRWTPOL	NWAIT信号极性 0：NWAIT低电平有效 1：NWAIT高电平有效
8	SBRSTEN	同步突发模式使能 0：禁止同步突发模式 1：使能同步突发模式
7	保留	必须保持复位值
6	NREN	NOR闪存访问使能 0：禁止NOR Flash访问 1：允许NOR Flash访问
5:4	NRW[1:0]	存储器数据宽度 00：8位 01：16位(复位缺省值) 10/11：保留
3:2	NRTP[1:0]	存储器类型 00：SRAM(region1~region3复位之后的默认值) 01：PSRAM(CRAM) 10：NOR Flash(region0复位之后的默认值) 11：保留
1	NRMUX	数据线/地址线复用 0：禁用地址/数据复用功能 1：允许地址/数据复用功能
0	NRBKEN	存储块使能 0：禁用对应的存储器块 1：使能对应的存储器块

SRAM/NOR Flash 时序寄存器 (EXMC_SNTCFGx) (x=0, 1, 2, 3)

偏移地址：0x04 + 8 * x, (x = 0, 1, 2, 3)

复位值：0x0FFF FFFF

该寄存器只能按字（32 位）访问。



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 该位只有在扩展模式中使用 00：模式A 01：模式B 10：模式C 11：模式D
27:24	DLAT[3:0]	NOR Flash数据延时，仅在同步模式有效 0x0：第一数据的保持时间为2个EXMC_CLK时钟周期 0x1：第一数据的保持时间为3个EXMC_CLK时钟周期 …… 0xF：第一数据的保持时间为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 0x0：保留 0x1：EXMC_CLK周期=2个HCLK周期 …… 0xF：EXMC_CLK周期=16个HCLK周期
19:16	BUSLAT[3:0]	总线延迟时间 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 0x0：总线延迟=1个HCLK周期 0x1：总线延迟=2个HCLK周期 …… 0xF：总线延迟=16个HCLK周期
15:8	DSET[7:0]	异步数据建立时间 该位域仅在异步模式有效 0x00：保留 0x01：数据建立时间=2个HCLK周期 …… 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 0x0：保留 0x1：地址建立时间=2个HCLK …… 0xF：地址建立时间=16个HCLK
3:0	ASET[3:0]	异步地址建立时间 该位域设置地址建立时间 注意：该位域仅在SRAM,ROM,NOR Flash的异步模式有效 0x0：地址建立时间= 1个HCLK …… 0xF：地址建立时间= 16个HCLK

SRAM/NOR Flash 写时序寄存器 (EXMC_SNWTCFGx) (x=0, 1, 2, 3)

偏移地址：0x104 + 8 * x, (x = 0, 1, 2, 3)
复位值：0x0FFF FFFF

该寄存器只能按字（32 位）访问。

该寄存器仅在扩展模式使能（寄存器 EXMC_SNCTL 位 EXMODEN 置 1）后有效。

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 该位只有在扩展模式（EXMC_SNCTLx寄存器的EXMODEN位为1）中使用 00：模式A 01：模式B 10：模式C 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 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。 0x0：总线延迟=1个HCLK周期 0x1：总线延迟=2个HCLK周期 …… 0xF：总线延迟=16个HCLK周期

15:8	WDSET[7:0]	异步数据建立时间 该位域仅在异步模式有效 0x00：保留 0x01：数据建立时间=2个HCLK周期 …… 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 该位域设置地址保持时间，仅在模式D与复用模式有效 0x0：保留 0x1：地址建立时间=2个HCLK …… 0xF：地址建立时间=16个HCLK
3:0	WASET[3:0]	异步地址建立时间 该位域设置地址建立时间 注意：该位域仅在SRAM,ROM,NOR Flash的异步模式有效 0x0：地址建立时间= 1个HCLK 0x1：地址建立时间= 2个HCLK …… 0xF：地址建立时间= 16个HCLK

24.4.2. NAND Flash/PC Card 控制器寄存器

NAND Flash/PC Card 控制寄存器 (EXMC_NPCTLx) (x=1, 2, 3)

偏移地址：0x40 + 0x20 * x, (x = 1, 2, 3)
复位值：0x0000 0018

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												ECCSZ[2:0]			ATR[3]
													rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATR[2:0]			CTR[3:0]				保留		ECCEN	NDW[1:0]		NDTP	NDBKEN	NDWTEN	保留
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
19:17	ECCSZ[2:0]	ECC块大小 000：256字节 001：512字节 010：1024字节 011：2048字节

		100：4096字节 101：8192字节
16:13	ATR[3:0]	ALE至RE的延迟 0x0：1个HCLK …… 0xF：16个HCLK
12:9	CTR[3:0]	CLE至RE的延迟 0x0：1个HCLK 0x1：2个HCLK …… 0xF：16个HCLK
8:7	保留	必须保持复位值
6	ECCEN	ECC使能 0：关闭ECC，并复位EXMC_NECCx 1：打开ECC
5:4	NDW[1:0]	外部存储器数据宽度 00：8位 01：16位 其他：保留 注意：对于PC/CF Card，数据宽度必须选择16位
3	NDTP	外部存储器的类型 0：PC Card, CF Card, PCMCIA 1：NAND Flash
2	NDBKEN	存储块使能 0：禁用对应的存储器块 1：使能对应的存储器块
1	NDWTEN	NWAIT信号使能位 0：关闭等待功能 1：使能等待功能
0	保留	必须保持复位值

NAND Flash/PC Card 中断使能寄存器 (EXMC_NPINTENx) (x=1, 2, 3)

偏移地址：0x44 + 0x20 * x, (x = 1, 2, 3)
复位值：0x0000 0046（对于 bank1 和 bank2），0x0000 0040（对于 bank3）

该寄存器只能按字（32 位）访问。

除了中断控制比特位，该寄存器还包含一个 FIFO 空状态位，该位主要用于 ECC。当写外部存储器时，FIFO 可以容纳来自 AHB 访问的 2 个字，使得 AHB 总线可以暂时被释放而用于其他外设。ECC 计算是基于从 FIFO 传递的数据。为了得到正确的 ECC 值，用户应该在 FIFO 空

状态标志位为 1 时才去读 ECC 寄存器。

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									FFEPT	INTFEN	INTHEN	INTREN	INTFS	INTHS	INTRS
									r	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:7	保留	必须保持复位值
6	FFEPT	FIFO空标志位 0：FIFO非空 1：FIFO空
5	INTFEN	中断下降沿检测使能 0：禁用中断下降沿检测 1：使能中断下降沿检测
4	INTHEN	中断高电平检测使能 0：禁用中断高电平检测 1：使能中断高电平检测
3	INTREN	中断上升沿中断检测使能 0：禁用中断上升沿检测 1：使能中断上升沿检测
2	INTFS	中断下降沿状态 0：没有检测到中断下降沿 1：检测到中断下降沿
1	INTHS	中断高电平状态 0：没有检测到中断高电平 1：检测到中断高电平
0	INTRS	中断上升沿状态 0：没有检测到中断上升沿 1：检测到中断上升沿

NAND Flash/PC Card 通用空间时序寄存器 (EXMC_NPCTCFGx) (x=1, 2, 3)

偏移地址：0x48 + 0x20 * x, (x = 1, 2, 3)
复位值：0xFCFC FCFC

该寄存器只能按字（32 位）访问。

这些操作适用于以下类型的外部存储器的通用存储空间 16 位的 PC Card，CF card 和 NAND Flash。

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

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

位/位域	名称	描述
31:24	COMHIZ[7:0]	通用空间数据总线的高阻时间 定义在通用空间进行写操作后数据总线保持高阻态时间 0x00：1个HCLK …… 0xFE：255个HCLK 0xFF：保留
23:16	COMHLD[7:0]	通用空间的保持时间 在发送地址后的地址保持时间，在写操作时，也作为数据信号保持的时间 0x00：保留 0x01：1个HCLK …… 0xFE：254个HCLK 0xFF：保留
15:8	COMWAIT[7:0]	通用空间的等待时间 定义了保持命令的最小时间 0x00：保留 0x01：2个HCLK (加上NWAIT时钟周期) …… 0xFE：255个HCLK (加上NWAIT时钟周期) 0xFF：保留
7:0	COMSET[7:0]	通用空间的建立时间 定义地址信号的建立时间 0x00：1个HCLK …… 0xFE：255个HCLK 0xFF：保留

NAND Flash/PC Card 属性空间时序寄存器 (EXMC_NPATCFGx) (x=1, 2, 3)

偏移地址：0x4C + 0x20 * x, (x = 1, 2, 3)
复位值：0xFCFC FCFC

该寄存器只能按字（32 位）访问。

用于 8 位访问 PC 卡的属性空间，或是用另一种时序来对 Nand Flash 的最后地址进行写操作。

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

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

位/位域	名称	描述
31:24	ATTHIZ[7:0]	属性空间数据总线的高阻时间 定义在属性空间进行写操作后数据总线保持高阻态时间 0x00：1个HCLK …… 0xFE：255个HCLK 0xFF：保留
23:16	ATTHLD[7:0]	属性空间的保持时间 在发送地址后的地址保持时间，在写操作时，也作为数据信号保持的时间 0x00：保留 0x01：1个HCLK …… 0xFE：254个HCLK 0xFF：保留
15:8	ATTWAIT[7:0]	属性空间的等待时间 定义了保持命令的最小时间 0x00：保留 0x01：2个HCLK (加上NWAIT时钟周期) …… 0xFE：255个HCLK (加上NWAIT时钟周期) 0xFF：保留
7:0	ATTSET[7:0]	属性空间的建立时间 定义地址信号的建立时间 0x00：1个HCLK …… 0xFE：255个HCLK 0xFF：保留

PC Card I/O 空间时序寄存器 (EXMC_PIOTCFG3)

偏移地址：0xB0
复位值：0xFCFC FCFC

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

IOHIZ[7:0]	IOHLD[7:0]
rw	rw

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

位/位域	名称	描述
31:24	IOHIZ[7:0]	I/O空间数据总线的高阻时间 定义在IO空间进行写操作后数据总线保持高阻态时间 0x00：0个HCLK …… 0xFF：255个HCLK
23:16	IOHLD[7:0]	I/O空间的保持时间 在发送地址后的地址保持时间，在写操作时，也作为数据信号保持的时间0x00：保留 0x01：1个HCLK …… 0xFF：255个HCLK
15:8	IOWAIT[7:0]	I/O空间的等待时间 定义了保持命令的最小时间 0x00：保留 0x01：2个HCLK (加上NWAIT时钟周期) …… 0xFF：256个HCLK (加上NWAIT时钟周期)
7:0	IOSET[7:0]	I/O空间的建立时间 定义地址信号的建立时间 0x00：1个HCLK …… 0xFF：256个HCLK

NAND Flash ECC 结果寄存器 (EXMC_NECCx) (x=1, 2)

偏移地址：0x54+0x20 * x, (x =1, 2)
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
ECC[31:16]															
r															

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

位/位域	名称	描述
31:0	ECC[31:0]	ECC计算结果

ECCSZ[2:0]	NAND Flash 页大小（字节）	ECC 位
0b000	256	ECC[21:0]
0b001	512	ECC[23:0]
0b010	1024	ECC[25:0]
0b011	2048	ECC[27:0]
0b100	4096	ECC[29:0]
0b101	8192	ECC[31:0]

25. 控制器局域网络（CAN）

25.1. 简介

CAN（Controller Area Network）总线是一种可以在无主机情况下实现微处理器或者设备之间相互通信的总线标准。

GD32A508xx 系列的 CAN 网络接口遵循 CAN 总线协议 2.0A、2.0B、ISO11898-1:2015 和 BOSCH CAN-FD 等规范。CAN 总线控制器可以处理总线上的数据收发，CAN0 和 CAN1 共享 28 个过滤器，而 CAN2 拥有 14 个独立的过滤器。用户可以通过 3 个发送邮箱将待发送数据传输至总线，邮箱发送的顺序由发送调度器决定。并通过 2 个深度为 3 的接收 FIFO 获取总线上的数据，接收 FIFO 的管理完全由硬件控制。同时 CAN 总线控制器硬件支持时间触发通信（Time-trigger communication）功能。

25.2. 主要特征

- 支持 CAN 总线协议 2.0A 和 2.0B；
- 常规帧：通信波特率最大为 1Mbit/s；
- 支持时间触发通信（Time-triggered communication）；
- 中断使能和清除；
- 支持 CAN-FD 帧；
- CAN-FD 帧：通信波特率最大为 6Mbit/s；
- 支持传输延迟补偿。

发送功能

- 3 个发送邮箱；
- 支持发送优先级；
- 支持发送时间戳。

接收功能

- 2 个深度为 3 的接收 FIFO；
- CAN0 和 CAN1 具有 28 个标识符过滤器；
- CAN2 拥有 14 个独立的标识符过滤器；
- FIFO 锁定功能。

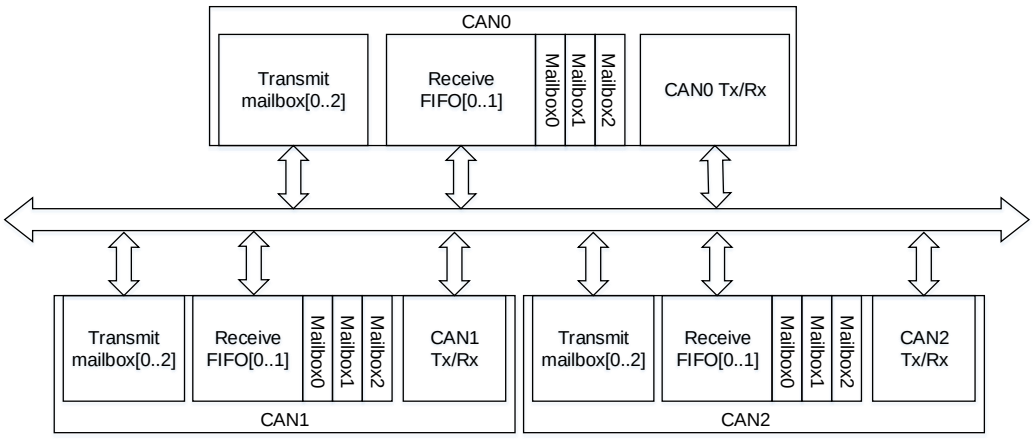
时间触发通信

- 在时间触发通信模式下禁用自动重传；
- 16 位定时器；
- 接收时间戳；
- 发送时间戳。

25.3. 功能说明

CAN 模块结构框图如*图 25-1. CAN 模块结构框图*所示。

图 25-1. CAN 模块结构框图



25.3.1. 工作模式

CAN 总线控制器有 3 种工作模式：

- 睡眠工作模式；
- 初始化工作模式；
- 正常工作模式。

睡眠工作模式

芯片复位后，CAN 总线控制器处于睡眠工作模式。该模式下 CAN 总线控制器的时钟停止工作并处于一种低功耗状态。

将 CAN_CTL 寄存器的 SLPWMOD 位置 1，可以使 CAN 总线控制器进入睡眠工作模式。当进入睡眠工作模式后，CAN_STAT 寄存器的 SLPWS 位将被硬件置 1。

将 CAN_CTL 寄存器的 AWU 位置 1，并当 CAN 检测到总线活动时，CAN 总线控制器将自动退出睡眠工作模式。将 CAN_CTL 寄存器的 SLPWMOD 位清 0，也可以退出睡眠工作模式。

由睡眠模式进入初始化工作模式：将 CAN_CTL 寄存器的 IWMOD 位置 1，SLPWMOD 位清 0。

由睡眠模式进入正常工作模式：将 CAN_CTL 寄存器的 IWMOD 位和 SLPWMOD 位清 0。

初始化工作模式

如果需要配置 CAN 总线通信参数，CAN 总线控制器必须进入初始化工作模式。将 CAN_CTL 寄存器的 IWMOD 位置 1，使 CAN 总线控制器进入初始化工作模式，将其清 0 则离开初始化工作模式。在进入初始化工作模式后，CAN_STAT 寄存器的 IWS 位将被硬件置 1。

由初始化模式进入睡眠模式：CAN_CTL 寄存器的 SLPWMOD 位置 1，IWMOD 位清 0。

由初始化模式进入正常工作模式：CAN_CTL 寄存器的 SLPWMOD 位和 IWMOD 位清 0。

正常工作模式

在初始化工作模式中配置完 CAN 总线通信参数后，将 CAN_CTL 寄存器的 IWMOD 位清 0 可以进入正常工作模式并与 CAN 总线网络中的节点进行正常通信。

由正常工作模式进入睡眠工作模式：CAN_CTL 寄存器的 SLPWMOD 位置 1，并等待当前数据收发过程结束。

由正常工作模式初始化工作模式：CAN_CTL 寄存器的 IWMOD 位置 1，并等待当前数据收发过程结束。

25.3.2. 通信模式

CAN 总线控制器有 4 种通信模式：

- 静默（Silent）通信模式；
- 回环（Loopback）通信模式；
- 回环静默（Loopback and Silent）通信模式；
- 正常（Normal）通信模式。

静默（Silent）通信模式

在静默通信模式下，可以从 CAN 总线接收数据，但不向总线发送任何数据。将 CAN_BT 寄存器中的 SCMOD 位置 1，使 CAN 总线控制器进入静默通信模式，将其清 0 可以退出静默通信模式。

静默通信模式可以用来监控 CAN 网络上的数据传输。

回环（Loopback）通信模式

在回环通信模式下，由 CAN 总线控制器发送的数据可以被自己接收并存入接收 FIFO，同时这些发送数据也送至 CAN 网络。将 CAN_BT 寄存器中的 LCMOD 位置 1，使 CAN 总线控制器进入回环通信模式，将其清 0 可以退出回环通信模式。

回环通信模式通常用来进行 CAN 通信自测。

回环静默（Loopback and Silent）通信模式

在回环静默通信模式下，CAN 的 RX 和 TX 引脚与 CAN 网络断开。CAN 总线控制器既不从 CAN 网络接收数据，也不向 CAN 网络发送数据，其发送的数据仅可以被自己接收。将 CAN_BT 寄存器中的 LCMOD 位和 SCMOD 位置 1，使 CAN 总线控制器进入回环静默通信模式，将它们清 0 可以退出回环静默通信模式。

回环静默通信模式通常用来进行 CAN 通信自测。对外 TX 引脚保持隐性状态（逻辑 1），RX 引脚保持高阻态。

正常（Normal）通信模式

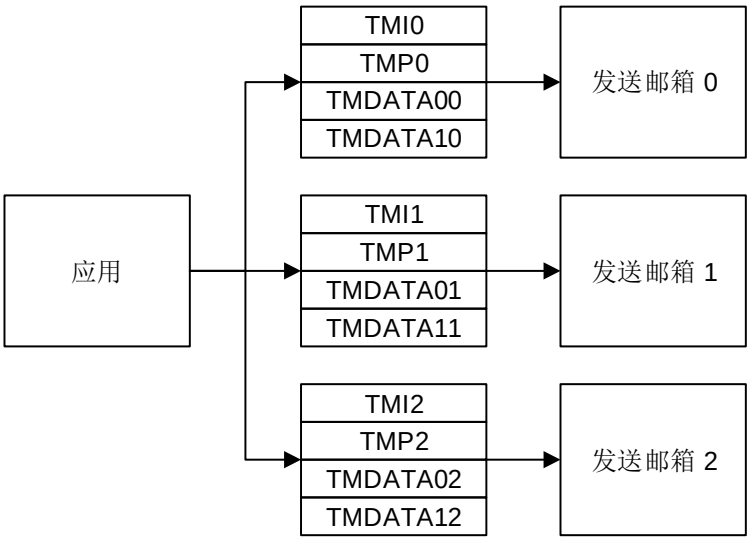
CAN 总线控制器通常工作在正常通信模式下，可以从 CAN 总线接收数据，也可以向 CAN 总线发送数据。这时需要将 CAN_BT 寄存器的 LCMOD 位和 SCMOD 位清 0。

25.3.3. 数据发送

发送寄存器

数据发送通过 3 个发送邮箱进行，可以通过寄存器 CAN_TMIx，CAN_TMPx，CAN_TMDATA0x 和 CAN_TMDATA1x 对发送邮箱进行配置。如*图 25-2. 发送寄存器*所示。

图 25-2. 发送寄存器

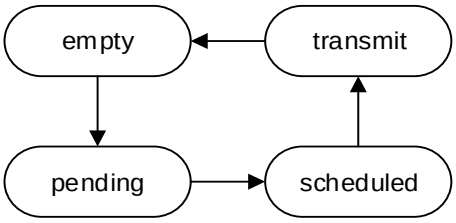


如果想要发送 CAN-FD 帧，使用发送邮箱 x（x 等于 0 到 2）时，仅需要写对应的 TMDATA0x（x 等于 0 到 2）寄存器。例如软件想要使用发送邮箱 0 发送 64 字节数据，需要通过写 TMDATA00 寄存器 16 次将待发送数据写入内部专用 SRAM 区。

发送邮箱状态转换

当发送邮箱处于 **empty** 状态时，应用程序才可以对邮箱进行配置。当邮箱被配置完成后，可以将 CAN_TMIx 寄存器的 TEN 位置 1，从而向 CAN 总线控制器提交发送请求，这时发送邮箱处于 **pending** 状态。当超过 1 个邮箱处于 **pending** 状态时，需要对多个邮箱进行调度，这时发送邮箱处于 **scheduled** 状态。当调度完成后，发送邮箱中的数据开始向 CAN 总线上发送，这时发送邮箱处于 **transmit** 状态。当数据发送完成，邮箱变为空闲，可以再次交给应用程序使用，这时发送邮箱重新变为 **empty** 状态。如*图 25-3. 发送邮箱状态转换*所示。

图 25-3. 发送邮箱状态转换



发送状态和错误信息

CAN_TSTAT 寄存器中的 MTF，MTFNERR，MAL 和 MTE 位用来说明发送状态和错误信息。

- MTF：发送完成标志位。当数据发送完成时，MTF 置 1。
- MTFNERR：无错误发送完成标志位。当数据发送完成且没有错误时，MTFNERR 置 1。
- MAL：仲裁失败标志位。当发送数据过程中出现仲裁失败时，MAL置1。
- MTE：发送错误标志位。当发送过程中检测到总线错误时，MTE置1。

数据发送步骤

数据发送步骤如下：

第 1 步：选择一个空闲发送邮箱；

第 2 步：根据应用程序要求，配置 4 个发送寄存器；

第 3 步：将 CAN_TMIx 寄存器的 TEN 置 1；

第 4 步：检测发送状态和错误信息。典型情况是检测到 MTF 和 MTFNERR 置 1，说明数据被成功发送。

发送选项

中止数据发送

将 CAN_TSTAT 寄存器的 MST 置 1，可以中止数据发送。

当发送邮箱处于 **pending** 和 **scheduled** 状态，CAN_TSTAT 寄存器的 MST 置 1 可以立即中止数据发送。

当发送邮箱处于 **transmit** 状态，则面临两种情况。一种情况是数据发送被成功地完成，MTF 和 MTFNERR 为 1，这时发送邮箱将转换为 **empty** 状态。相对的，如果数据发送过程中出现了问题，这时发送邮箱将转换为 **scheduled** 状态，这时数据发送被中止。

发送优先级

当有 2 个及其以上发送邮箱等待发送时，寄存器 CAN_CTL 的 TFO 位的值可以决定发送顺序。

当 TFO 为 1，所有等待发送的邮箱按照先来先发送（FIFO）的顺序进行。

当 TFO 为 0，具有最小标识符（Identifier）的邮箱最先发送。如果所有的标识符（Identifier）相等，具有最小邮箱编号的邮箱最先发送。

25.3.4. 数据接收

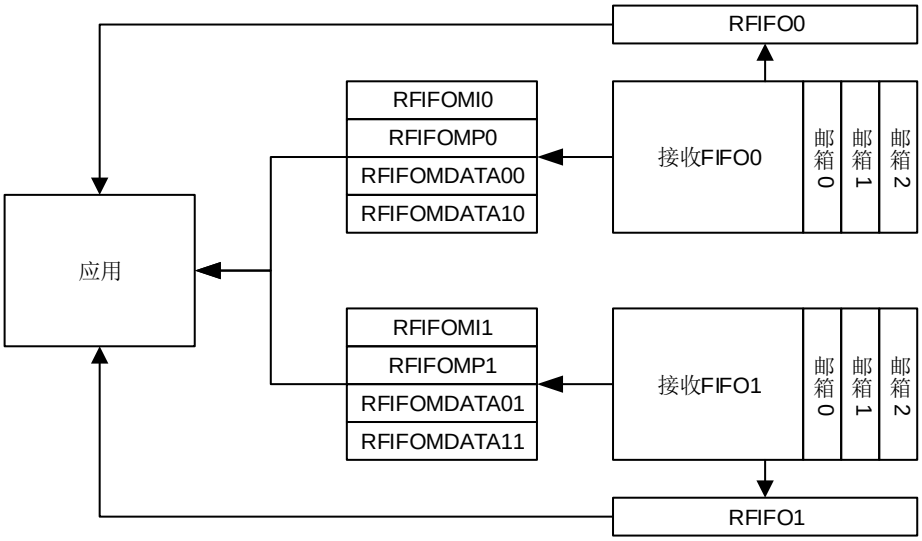
接收寄存器

应用程序通过 2 个深度为 3 的 FIFO 接收来自 CAN 网络的数据。

寄存器 CAN_RFIFOx 可以操作 FIFO，也包含 FIFO 状态。寄存器 CAN_RFIFOMIx，CAN_RFIFOMPx，CAN_RFIFOMDATA0x 和 CAN_RFIFOMDATA1x 用于接收数据帧。

如*图 25-4. 接收寄存器*所示。

图 25-4. 接收寄存器



接收 FIFO

每个接收 FIFO 包含 3 个接收邮箱，用来接收存储数据帧。这些邮箱按照先进先出方式进行组织，最早从 CAN 网络接收的数据，最早被应用程序处理。

寄存器 CAN_RFIFOx 包含 FIFO 状态信息和帧的数量。当 FIFO 中包含数据时，可以通过寄存器 CAN_RFIFOMIx，CAN_RFIFOMPx，CAN_RFIFOMDATA0x 和 CAN_RFIFOMDATA1x 读取数据，之后将寄存器 CAN_RFIFOx 的 RFD 位置 1 释放邮箱，并且等待其由硬件自动清 0。

如果接收到 CAN-FD 帧，数据将被存储到内部专用 SRAM 中，并通过多次读 CAN_RFIFOMDATA0x 寄存器将数据取出。FIFO0 使用 CAN_RFIFOMDATA00，FIFO1 使用 CAN_RFIFOMDATA01 寄存器。例如，如果软件想要从 FIFO0 中读 64 字节数据，需要通过读 CAN_RFIFOMDATA00 寄存器 16 次将数据全部读出。

接收 FIFO 状态信息

接收 FIFO 状态信息包含在寄存器 CAN_RFIFOx 中。

RFL：FIFO 中包含的帧数量。FIFO 为空时，RFL 为 0；FIFO 为满时，RFL 为 3。

RFF：FIFO 满状态标志位。这时 RFL 为 3。

RFO：FIFO 溢出标志位。当 FIFO 已经包含了 3 个数据帧时，新的数据帧到来使 FIFO 发生溢出。如果 CAN_CTL 寄存器的 RFOD 位被置 1，新的数据帧将丢弃。如果该位被清 0，新的数据帧将覆盖接收 FIFO 中最后一帧数据。

数据接收步骤

第 1 步：查看 FIFO 中帧的数量。

第 2 步：通过 CAN_RFIFOMIx，CAN_RFIFOMPx，CAN_RFIFOMDATA0x 和 CAN_RFIFOMDATA1x 读取数据。

第 3 步：将寄存器 CAN_RFIFOx 的 RFD 置 1 释放邮箱，并且等待其由硬件自动清 0。

25.3.5. 过滤功能

一个待接收的数据帧会根据其标识符（Identifier）进行过滤：硬件会将通过过滤的帧送至接收 FIFO，并丢弃没有通过过滤的帧。

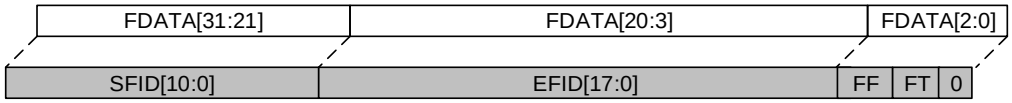
过滤器位宽

CAN0 和 CAN1 的过滤器包含 28 个单元，它们是 bank0 到 bank27。CAN2 的过滤器包含 14 个独立的单元。

每一个过滤器单元有 2 个寄存器 CAN_FxDATA0 和 CAN_ FxDATA1，它们可以配置为 2 种位宽：32-bit 位宽和 16-bit 位宽。

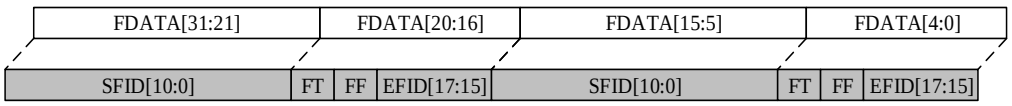
32-bit 位宽：CAN_FxDATAy 包含字段 SFID[10:0]，EFID[17:0]，FF 和 FT。如*图 25-5. 32-bit 位宽过滤器*所示。

图 25-5. 32-bit 位宽过滤器



16-bit 位宽：CAN_FxDATAy 包含字段：SFID[10:0]，FT，FF 和 EFID[17:15]。如*图 25-6. 16-bit 位宽过滤器*所示。

图 25-6. 16-bit 位宽过滤器



掩码模式

对于一个待过滤的数据帧的标识符（Identifier），掩码模式用来指定哪些位必须与预设的标识符相同，哪些位无需判断。

一个 32-bit 位宽掩码模式过滤器如*图 25-7. 32-bit 位宽掩码模式过滤器*所示。

图 25-7. 32-bit 位宽掩码模式过滤器

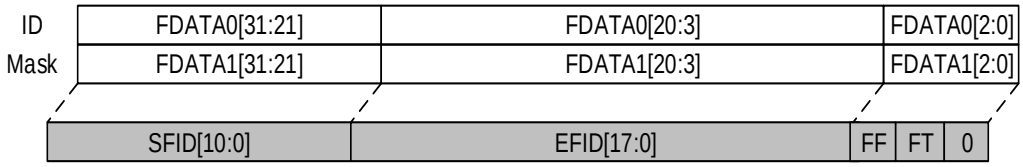
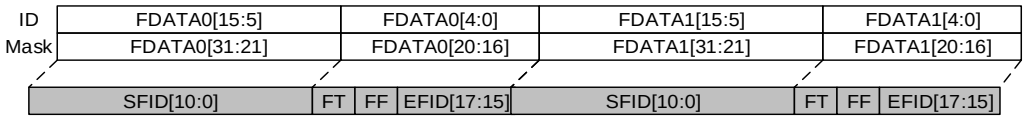


图 25-8. 16-bit 位宽掩码模式过滤器



列表模式

对于一个待过滤的数据帧的标识符（Identifier），列表模式用来表示与预设的标识符列表中能够匹配则通过，否则丢弃。

一个 32-bit 位宽列表模式过滤器如*图 25-9. 32-bit 位宽列表模式过滤器*所示。

图 25-9. 32-bit 位宽列表模式过滤器

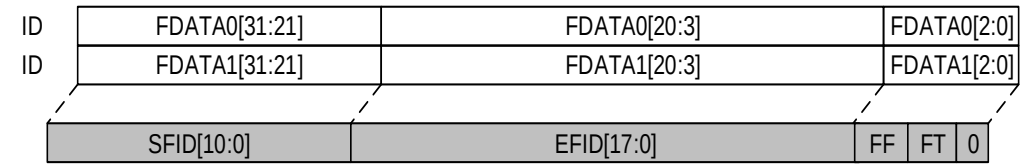
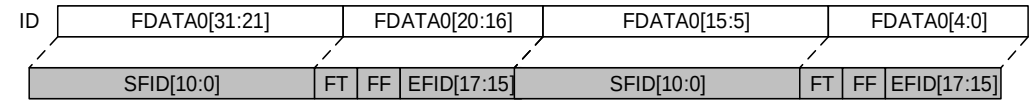


图 25-10. 16-bit 位宽列表模式过滤器



过滤序号

过滤器由若干过滤单元（Bank）组成，每个过滤单元因为位宽和模式的选择不同，而具有不同的过滤效果。例如*表 25-1. 32-bit 过滤序号*所示的 2 个过滤单元，Bank0 是 32-bit 位宽掩码模式，Bank1 是 32-bit 位宽列表模式。

表 25-1. 32-bit 过滤序号

过滤单元	过滤器数据寄存器	过滤序号
0	F0DATA0-32bit-ID	0
	F0DATA1-32bit-Mask	
1	F1DATA0-32bit-ID	1
	F1DATA1-32bit-ID	2

过滤器关联的 FIFO

28 个过滤单元均可以关联接收 FIFO0 或接收 FIFO1。一旦一个过滤单元关联到接收 FIFO，

只有通过这个过滤单元的帧才会被传送到接收 FIFO 中存储。

过滤器激活控制

一个过滤单元如果被应用程序用到，就必须激活。通过 CAN_FW 寄存器可以进行配置。

过滤索引

一个包含过滤序号（Fiter Number）N 的过滤单元通过了某个帧，则该帧数据的过滤索引（Filtering Index）为 N。这时 CAN_RFIFOMPx 中 FI 的值为 N。

在*表 25-2. 过滤索引*中，如果一个帧通过了 FIFO0 中过滤序号 10（Filter Number=10）的过滤单元，那么该帧的过滤索引为 10。这时 CAN_RFIFOMPx 中 FI 的值为 10。

过滤序号不关心对应的过滤单元（Bank）是否处于工作状态。例如 Bank3 被关联到 FIFO0，且为“不激活”状态，但它仍然包含过滤序号 3 和 4。

表 25-2. 过滤索引

过滤单元	FIFO0	激活	过滤序号	过滤单元	FIFO1	激活	过滤序号
0	F0DATA0-32bits-ID	是	0	2	F2DATA0[15:0]-16bits-ID	是	0
	F0DATA1-32bits-Mask				F2DATA0[31:16]-16bits-Mask		
1	F1DATA0-32bits-ID	是	1		F2DATA1[15:0]-16bits-ID		1
	F1DATA1-32bits-ID		2		F2DATA1[31:16]-16bits-Mask		
3	F3DATA0[15:0]-16bits-ID	否	3	4	F4DATA0-32bits-ID	否	2
	F3DATA0[31:16]-16bits-Mask				F4DATA1-32bits-Mask		
	F3DATA1[15:0]-16bits-ID		4	5	F5DATA0-32bits-ID	否	3
	F3DATA1[31:16]-16bits-Mask				F5DATA1-32bits-ID		4
7	F7DATA0[15:0]-16bits-ID	否	5	6	F6DATA0[15:0]-16bits-ID	是	5
	F7DATA0[31:16]-16bits- ID		6		F6DATA0[31:16]-16bits- ID		6
	F7DATA1[15:0]-16bits-ID		7		F6DATA1[15:0]-16bits-ID		7
	F7DATA1[31:16]-16bits- ID		8		F6DATA1[31:16]-16bits- ID		8
8	F8DATA0[15:0]-16bits-ID	是	9	10	F10DATA0[15:0]-16bits-ID	否	9
	F8DATA0[31:16]-16bits- ID		10		F10DATA0[31:16]-16bits-Mask		

过滤单元	FIFO0	激活	过滤序号	过滤单元	FIFO1	激活	过滤序号
	F8DATA1[15:0]-16bits-ID		11		F10DATA1[15:0]-16bits-ID		10
	F8DATA1[31:16]-16bits- ID		12		F10DATA1[31:16]-16bits-Mask		
9	F9DATA0[15:0]-16bits-ID	是	13	11	F11DATA0[15:0]-16bits-ID	否	11
	F9DATA0[31:16]-16bits-Mask				F11DATA0[31:16]-16bits-ID		12
	F9DATA1[15:0]-16bits-ID		14		F11DATA1[15:0]-16bits-ID		13
	F9DATA1[31:16]-16bits-Mask				F11DATA1[31:16]-16bits-ID		14
12	F12DATA0-32bits-ID	是	15	13	F13DATA0-32bits-ID	是	15
	F12DATA1-32bits-Mask				F13DATA1-32bits- ID		16

优先级

过滤器优先级规则如下：

1、32-bits 位宽模式高于 16-bits 位宽模式；

2、列表模式高于掩码模式；

3、较小的过滤序号（Filter Number）具有较高的优先级。

25.3.6. 时间触发通信

时间触发通信是 CAN 数据链路层应用协议。CAN 网络中的所有节点都按照一个预先设定的时间序列进行通信，尤其适合于时间周期性应用和时间确定性应用。

在这种通信模式下，一个内部的 16-bit 计数器开始工作，在每一个 CAN 位时间（Bit time）增 1。这个内部计数器为数据发送和数据接收提供时间戳，这些时间戳存放在寄存器 CAN_RFIFOMPx 和 CAN_TMPx 中。

在这种通信模式下，自动重发功能是禁止的。

25.3.7. 通信参数

自动重发禁止模式

在时间触发通信模式下，要求自动重发必须是禁止的，可以通过将 CAN_CTL 寄存器的 ARD 位置 1 满足要求。

在这种模式下，数据只会被发送一次，如果因为仲裁失败或者总线错误而导致发送失败，CAN 总线控制器不会像通常那样进行数据自动重发。

发送结束时，寄存器 CAN_TSTAT 的 MTF 位被硬件置 1，而发送状态信息可以通过 MTFNERR，MAL 和 MTE 获得。

位时序（Bit time）

CAN 协议采用位同步传输方式。这种方式不仅增大了传输容量，而且意味着需要一种复杂的位同步方法。面向字节传输的位同步方式适用于接收在每个字节前都有起始位的情况，而同步传输协议只要求数据帧的最开始有一个起始位。为保证接收器能正确读取信息，需要不断地进行重新同步。因此，在相位缓冲段采样点前面和后面都应该插入一个帧间隔。

可以通过位操作仲裁方式访问 CAN 总线。信号从发送器到接收器，再回到发送器必须在一个位时间内完成。为了达到同步的目的，除了相位缓冲段外，还需要一个传输延时段。在信号传输过程中，传输延时段被视为发送或接收延时。

CAN 总线控制器将位时间分为 3 个部分。

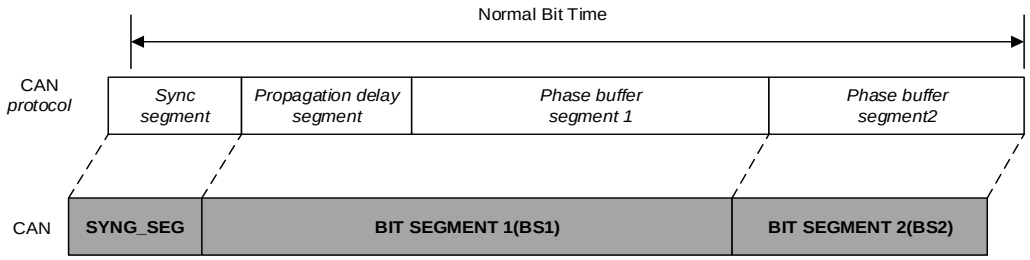
同步段（Synchronization segment），记为 SYNC_SEG。该段占用 1 个时间单元（*1* × □□）。

位段 1（Bit segment 1），记为 BS1。该段占用 1 到 16 个时间单元。相对于 CAN 协议而言，BS1 相当于传播时间段（Propagation delay segment）和相位缓冲段 1（Phase buffer segment 1）。

位段 2（Bit segment 2），记为 BS2。该段占用 1 到 8 个时间单元。相对于 CAN 协议而言，BS2 相当于相位缓冲段 2（Phase buffer segment 2）。

对比与 CAN 协议，位时序如*图 25-11. 位时序*所示。

图 25-11. 位时序



再同步补偿宽度 SJW（resynchronization Jump Width）对 CAN 网络节点同步误差进行补偿，占用 1 到 4 个时间单元。

有效跳变沿定义为只有当在前一个采样点检测到的总线状态为隐性（recessive）时，一个位时间内从隐性位到显性位的第一次转变。

如果有效跳变在 BS1 期间被检测到，而不是 SYNC_SEG 期间，BS1 将会最多被延长 SJW，因此采样点延时。

相反，如果有效跳变在 BS2 期间被检测到，而不是 SYNC_SEG 期间，BS2 将会最多被缩短 SJW，因此采样点提前。

注：有关硬同步和再同步的详细说明，请参考ISO 11898标准。

波特率

波特率计算公式如下：

$$BaudRate=\frac{1}{Normal\ Bit\ Time} \tag{25-1}$$

$$Normal\ Bit\ Time=t_{SYNC_SEG}+t_{BS1}+t_{BS2} \tag{25-2}$$

其中：

$$t_{SYNC_SEG}=1\times t_q \tag{25-3}$$

$$t_{BS1}=(1+BT.BS1)\times t_q \tag{25-4}$$

$$t_{BS2}=(1+BT.BS2)\times t_q \tag{25-5}$$

$$t_q=(1+BT.BAUDPSC)\times t_{PCLK1} \tag{25-6}$$

25.3.8. CAN-FD 操作

通过将 CAN_FDCTL 寄存器的 FDEN 位置 1，可以使能 CAN FD (CAN with Flexible Data rate) 功能。如果 FDEN 位被清 0，CAN 总线控制器仅支持常规帧（标准帧和扩展帧）的收发，若 FDEN 位被置 1，则 CAN 总线控制器同时支持常规帧（标准帧和扩展帧）以及 FD 帧的收发。根据协议，当前帧是否为 FD 帧是通过帧的 FDF 位来判断（在常规帧中该位为保留位）。如果 FDF 位为隐性，表示是 CAN FD 帧；如果为显性，表示是常规帧。

通过配置 CAN_FDCTL 寄存器的 NISO 位，可以选择 CAN-FD 功能支持 ISO11898-1 或 BOSCH CAN FD 规范 V1.0。

在 CAN-FD 帧的帧结构中，FDF 位之后是保留位和 BRS 位。BRS 位决定数据位速率，当 BRS 位为显性时，表示不能通过配置 CAN_DBT 寄存器来切换数据位速率。当 BRS 位为隐性时，可以通过配置 CAN_DBT 寄存器使得数据段（从 BRS 位到 ACK 场之前）的位速率高于仲裁段的位速率。详情请参考 ISO11898-1 或 BOSCH CAN FD 规范 V1.0。

通过将 CAN_FDCTL 寄存器的 PRED 位清 0，可以使能协议异常处理功能。此时，在接收帧数据过程中检测到隐性的保留位时，该功能将使操作状态转变为 IDLE 并在下一个采样点中止当前帧。反之，将 PRED 位置 1，该功能将被禁止，隐性的保留位将被视为格式错误，并当做错误帧来进行处理，同时 CAN_FDSTAT 寄存器的 PRE 位将被置 1。

ISO11898-1 或 BOSCH CAN FD 规范 V1.0 规定的发送 ESI 位（该位位于 CAN FD 帧的 DLC 位域之前）功能通过 CAN_FDCTL 寄存器的 ESIMOD 位和 CAN_TMPx 寄存器的 ESI 位来实现。如果将 ESIMOD 位清 0，当 CAN 总线控制器处于被动错误状态时，该位为隐性；当处于主动错误状态时，该位为显性。若将 ESIMOD 位置 1，将根据 CAN_TMPx 寄存器的 ESI 位的值来决定该位为显性还是隐性。

发送帧 FDF 位和 BRS 位的总线电平逻辑由 CAN_TMPx 寄存器的 FDF 位和 BRS 位的值决定。

25.3.9. 传输延迟补偿

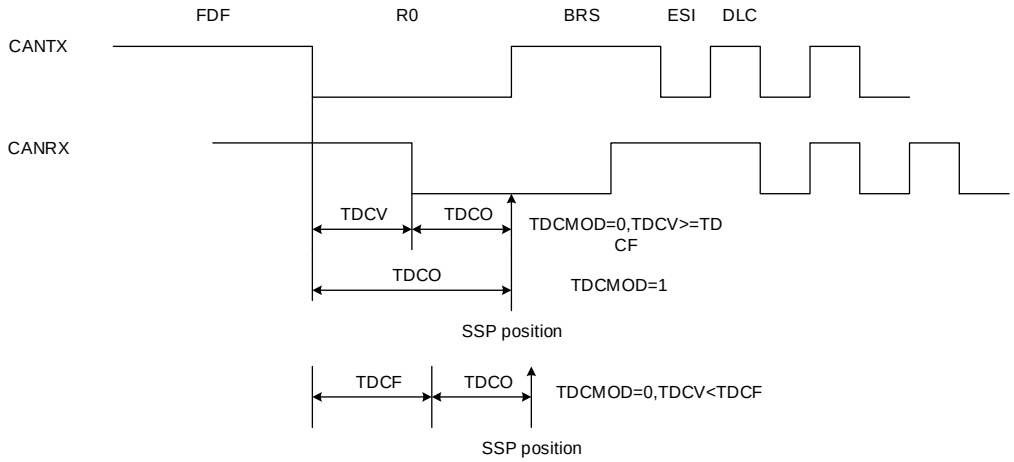
CAN-FD 协议支持传输延迟补偿机制。由于 CAN 收发器存在回路延迟，因此当发送 CAN-FD 帧的高速数据段的位时间长度小于收发器内部回路延迟的限定值时，该机制可以避免当采样点到来时发送节点还没有收到其自己发出的位，从而报错的情况发生。关于传输延迟补偿的具体定义，请参考 ISO11898-1 或 BOSCH CAN FD 规范 V1.0。

将 CAN_FDCTL 寄存器的 TDCEN 位置 1 将使能传输延迟补偿。

传输延迟补偿机制可以调节次级采样点（SSP）的位置。SSP_Delay 被定义为 CANTX 上的信号到 SSP 采样点的延迟。如果 CAN_FDCTL 寄存器的 TDCMOD 位被置 1，SSP_Delay 的值由 CAN_FDTDC 寄存器的 TDCO 位域软件配置决定。如果 TDCMOD 位被清 0，硬件将自动计算位速率转换之前的 FDF 位到 RES0 位的下降沿在 CAN_TX 与 CAN_RX 上出现的延迟，并将计算值存入 CAN_FDSTAT 寄存器的 TDCV 位域。由于存在信号毛刺，可能导致硬件自动计算的 SSP 位置比预期的提前，为了避免 TDCV 的值过小，可以使用 CAN_FDTDC 寄存器的 TDCF 位域。如果 TDCV 的值大于 TDCF，SSP_Delay 的值被定义为 TDCO 加上 TDCV，否则 SSP_Delay 的值被定义为 TDCO 加上 TDCF。

SSP_Delay 的值不能大于 3 个数据位时间。

图 25-12 传输延迟测量



25.3.10. 错误标志

CAN 总线的状态可以通过 CAN_ERR 寄存器的发送错误计数值（Transmit Error Counter，TECNT）和接收错误计数值（Receive Error Counter， RECNT）反映，其值会根据错误的情况由硬件增加或减少，软件可以通过这些值判断 CAN 网络的稳定性。关于错误计数值的详细信息请参考 CAN 协议相关章节。

通过使能 CAN_INTEN 寄存器中的相应位（ERRIE 等），软件可以在检测到错误时产生相应中断。

离线恢复

当 TECNT 大于 255 时，CAN 总线控制器进入离线状态，这时寄存器 CAN_ERR 中的 BOERR 置 1，并且发送和接收失效。

根据寄存器 CAN_CTL 中的 ABOR 配置，离线恢复（变为主动错误状态）有 2 种方式。这两种方式都要求处于离线状态的 CAN 总线控制器检测到 CAN 协议所定义的离线恢复序列（在 CAN_RX 检测到 128 次连续 11 个位的隐性位）时，才会自动恢复。

如果 ABOR 为 1，将在检测到离线恢复序列后自动恢复。

如果 ABOR 为 0，则必须先将 CAN_CTL 中的 IWMOD 置 1 进入初始化工作模式，然后进入正常工作模式并在检测到离线恢复序列后恢复。

25.3.11. 中断

CAN 总线控制器占用 4 个中断向量，通过寄存器 CAN_INTEN 进行控制。

这 4 个中断向量对应 4 类中断源：

- 发送中断；
- FIFO0 中断；
- FIFO1 中断；
- 错误和状态改变中断。

发送中断

发送中断包括：

- 寄存器CAN_TSTAT中的MTF0置1：发送邮箱0变为空闲。
- 寄存器CAN_TSTAT中的MTF1置1：发送邮箱1变为空闲。
- 寄存器CAN_TSTAT中的MTF2置1：发送邮箱2变为空闲。

FIFO0 中断

FIFO0 中断包括：

- FIFO0中包含待接收数据：寄存器CAN_RFIFO0中的RFL0不为0，CAN_INTEN寄存器中RFNEIE0被置位；
- FIFO0满：寄存器CAN_RFIFO0中的RFF0为1，CAN_INTEN寄存器中RFFIE0被置位；
- FIFO0溢出：寄存器CAN_RFIFO0中的RFO0为1，CAN_INTEN寄存器中RFOIE0被置位。

FIFO1 中断

FIFO1 中断包括：

- FIFO1中包含待接收数据：寄存器CAN_RFIFO1中的RFL1不为0，CAN_INTEN寄存器中RFNEIE1被置位；
- FIFO1满：寄存器CAN_RFIFO1中的RFF1为1，CAN_INTEN寄存器中RFFIE1被置位；
- FIFO1溢出：寄存器CAN_RFIFO1中的RFO1为1，CAN_INTEN寄存器中RFOIE1被置位。

错误和工作模式改变（EWMC）中断

错误和工作模式改变中断可由以下条件触发：

- 错误：CAN_STAT寄存器的ERRIF和CAN_INTEN寄存器的ERRIE被置位，请参考CAN_STAT寄存器中ERRIF位描述；
- 唤醒：CAN_STAT寄存器中的WUIF和CAN_INTEN寄存器的WIE被置位；
- 进入睡眠模式：CAN_STAT寄存器中的SLPIF和CAN_INTEN寄存器的SLPWIE被置

CAN 总线控制器的中断产生条件可参考*表 25-3. CAN 事件/中断标志*。

表 25-3. CAN 事件/中断标志

中断事件	事件/中断标志		使能控制位	
发送中断	发送邮箱 0 空闲标志 MTF0		TMEIE	
	发送邮箱 1 空闲标志 MTF1			
	发送邮箱 2 空闲标志 MTF2			
FIFO0 中断	接收 FIFO0 中帧的数量 RFL0[1:0]		RFNEIE0	
	接收 FIFO0 满 RFF0		RFFIE0	
	接收 FIFO0 溢出 RFO0		RFOIE0	
FIFO1 中断	接收 FIFO1 中帧的数量 RFL1[1:0]		RFNEIE1	
	接收 FIFO1 满 RFF1		RFFIE1	
	接收 FIFO1 溢出 RFO1		RFOIE1	
EWMC 中断	警告错误 WERR	错误中断标志 ERRIF	WERRIE	ERRIE
	被动错误 PERR		PERRIE	
	离线错误 BOERR		BOIE	
	错误种类 1<= ERRN[2:0] <= 6		ERRNIE	
	从睡眠工作模式唤醒的状态改变中断标志WUIF		WIE	
	进入睡眠工作模式的状态改变中断标志SLPIF		SLPWIE	

25.4. CAN 寄存器

CAN0 基地址：0x4000 6400

CAN1 基地址：0x4000 6800

CAN2 基地址：0x4000 CC00

25.4.1. 控制寄存器（CAN_CTL）

地址偏移：0x00
复位值：0x0001 0002

该寄存器只能按字（32 位）访问。

位/位域	名称	描述
31:17	保留	必须保持复位值。
16	DFZ	调试冻结 如果DBG_CTL寄存器中CANx_HOLD被置位，该位用来定义CAN控制器工作在调试冻结或正常工作状态。如果DBG_CTL寄存器中CANx_HOLD被清零，该位无效。 0：处于Debug时，CAN接收和发送正常工作 1：处于Debug时，CAN接收和发送停止
15	SWRST	软件复位 0：正常操作 1：复位CAN并进入睡眠工作模式。该位会自动清0。
14:8	保留	必须保持复位值。
7	TTC	时间触发通信 0：禁用时间触发通信 1：使能时间触发通信
6	ABOR	自动离线恢复 0：通过软件手动地从离线状态恢复 1：通过硬件自动的从离线状态恢复
5	AWU	自动唤醒 一旦自动唤醒后，CAN_CTL寄存器的SLPWMOD位将自动被清0。

0：通过软件手动的从睡眠工作模式唤醒
1：通过硬件自动的从睡眠工作模式唤醒

4	ARD	自动重发禁止 0：使能自动重发 1：禁用自动重发
3	RFOD	禁用接收FIFO满时覆盖 0：使能接收FIFO满时覆盖。当接收FIFO满时，FIFO中的数据被新来的数据覆盖 1：禁用接收FIFO满时覆盖。当接收FIFO满时，新来的数据被丢弃，FIFO中的数据保持不变，不会被覆盖
2	TFO	发送FIFO顺序 0：标识符（Identifier）较小的帧先发送 1：所有等待发送的邮箱按照先进先出（FIFO）的顺序发送
1	SLPWMOD	睡眠工作模式 如果软件将该位置1，CAN将会在当前发送或接收完成时进入睡眠工作模式。该位可由软件或者硬件清0。如果CAN_CTL寄存器中AWU被置位，当检测到CAN总线工作时，该位被清0。 0：禁用睡眠工作模式 1：使能睡眠工作模式
0	IWMOD	初始化工作模式 0：禁用初始化工作模式 1：使能初始化工作模式

25.4.2. 状态寄存器（CAN_STAT）

地址偏移：0x04
复位值：0x0000 0C02

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				RXL	LASTRX	RS	TS	保留			SLPIF	WUIF	ERRIF	SLPWS	IWS
				r	r	r	r				rc_w1	rc_w1	rc_w1	r	r

位/位域	名称	描述
31:12	保留	必须保持复位值。
11	RXL	RX引脚电平
10	LASTRX	RX引脚最近一次的采样值
9	RS	接收状态

0：CAN当前不是接收器
1：CAN当前是接收器

8	TS	发送状态 0：CAN当前不是发送器 1：CAN当前是发送器
7:5	保留	必须保持复位值。
4	SLPIF	进入睡眠工作模式的状态改变中断标志 该位在进入睡眠工作模式时由硬件置位。当CAN不再处于睡眠工作模式时由硬件清零。该位也可以由软件写1清0。 0：CAN没有进入睡眠工作模式 1：CAN进入睡眠工作模式。如果相应的中断使能位为1，则发生中断
3	WUIF	从睡眠工作模式唤醒的状态改变中断标志 该位在睡眠工作模式时检测到CAN总线上的活动时由硬件置位。该位由软件写1清0。 0：没有检测到唤醒信号 1：发现唤醒信号。如果相应的中断使能位为1，则发生中断。
2	ERRIF	错误中断标志 该位由以下事件置位。CAN_ERR寄存器中BOERR位和CAN_INTEN寄存器中BOIE位都置位。或CAN_ERR寄存器中PERR位和CAN_INTEN寄存器中PERRIE位都置位。或CAN_ERR寄存器中WERR位和CAN_INTEN寄存器中WERRIE位都置位。或CAN_ERR寄存器中ERRN位域的值不为0且CAN_INTEN寄存器中ERRNIE位置位。该位由软件写1清零。 0：没有错误 1：发生错误。如果相应的中断使能位为1，则发生中断
1	SLPWS	睡眠工作状态 将CAN_CTL寄存器中SLPWMOD位置位进入睡眠工作模式后该位由硬件置位。当CAN由正常通信模式切换到睡眠工作模式，需等待当前发送过程或者接收过程完成。当CAN离开睡眠工作模式（清除CAN_CTL寄存器中SLPWMOD位或是在CAN_CTL寄存器中AWU置位时检测到CAN总线上的活动）时，该位由硬件清零。如果由睡眠工作模式切换到正常工作模式，该位在CAN接收到来自总线的连续11个隐性位后被清0。 0：CAN没有处于睡眠工作状态 1：CAN处于睡眠工作状态
0	IWS	初始化工作状态 将CAN_CTL寄存器中IWMOD位置位进入初始化工作模式后该位由硬件置位。当CAN由正常通信模式切换到初始化工作模式，需等待当前发送过程或者接收过程完成。在清除CAN_CTL寄存器中IWMOD位离开初始化模式后，该位由硬件清0。如果由初始化工作模式切换到正常工作模式，该位在CAN接收到来自总线的连续11个隐性位后被清0。 0：CAN没有处于初始化工作状态 1：CAN处于初始化工作状态

25.4.3. 发送状态寄存器（CAN_TSTAT）

地址偏移：0x08

复位值：0x1C00 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
TMLS2	TMLS1	TMLS0	TME2	TME1	TME0	NUM[1:0]		MST2	保留			MTE2	MAL2	MTFNERR2	MTF2
r	r	r	r	r	r	r		rs				rc_w1	rc_w1	rc_w1	rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MST1	保留			MTE1	MAL1	MTFNERR1	MTF1	MST0	保留			MTE0	MAL0	MTFNERR0	MTF0
rs				rc_w1	rc_w1	rc_w1	rc_w1	rs				rc_w1	rc_w1	rc_w1	rc_w1

位/位域	名称	描述
31	TMLS2	在发送FIFO中邮箱2最后发送 该位为1表明，当有2个及其以上帧等待发送时，发送邮箱2具有最后的发送顺序。
30	TMLS1	在发送FIFO中邮箱1最后发送 该位为1表明，当有2个及其以上帧等待发送时，发送邮箱1具有最后的发送顺序。
29	TMLS0	在发送FIFO中邮箱0最后发送 该位为1表明，当有2个及其以上帧等待发送时，发送邮箱0具有最后的发送顺序。
28	TME2	发送邮箱2空 0：发送邮箱2不为空 1：发送邮箱2空
27	TME1	发送邮箱1空 0：发送邮箱1不为空 1：发送邮箱1空
26	TME0	发送邮箱0空 0：发送邮箱0不为空 1：发送邮箱0空
25:24	NUM[1:0]	当发送FIFO不满时，NUM表示下一个将要发送的邮箱号。 当发送FIFO满时，NUM表示最后一个将要发送的邮箱号。
23	MST2	邮箱2停止发送 将其置1，将停止邮箱2的发送过程。 当邮箱2变为emtpy状态时，该位被硬件自动清0。
22:20	保留	必须保持复位值。
19	MTE2	邮箱2发送错误 当发生发送错误时，该位由硬件置1。由软件写1清0或对CAN_TSTAT寄存器中MTF2写1清0。也可以在下一次发送开始时由硬件清0。

		当发生错误时该位被置1。
18	MAL2	邮箱2仲裁失败 当发生发送错误时，该位由硬件置1。由软件写1清0或对CAN_TSTAT寄存器中MTF2写1清0。也可以在下一次发送开始时由硬件清0。
17	MTFNERR2	邮箱2无错发送完成 当发送结束并且没有错误产生时，该位由硬件置1。由软件写1清0或对CAN_TSTAT寄存器中MTF2写1清0。也可以在无错传输结束时由硬件清0。 0：传输结束时发生了错误 1：传输结束且没有错误
16	MTF2	邮箱2发送完成 当发送完成或被中止时，该位由硬件置1。由软件写1清0，或当CAN_TMI2寄存器的TEN被置位时清0。 0：发送邮箱2正在发送 1：发送邮箱2完成发送
15	MST1	邮箱1停止发送 将其置1，将停止邮箱1的发送过程。 当邮箱1变为emtpy状态时，该位被硬件自动清0。
14:12	保留	必须保持复位值。
11	MTE1	邮箱1发送错误 当发生发送错误时，该位由硬件置1。由软件写1清0或对CAN_TSTAT寄存器中MTF1写1清0。也可以在下一次发送开始时由硬件清0。 当发生错误时该位被置1。
10	MAL1	邮箱1仲裁失败 当发生发送错误时，该位由硬件置1。由软件写1清0或对CAN_TSTAT寄存器中MTF1写1清0。也可以在下一次发送开始时由硬件清0。 当发生仲裁失败时该位被置1。
9	MTFNERR1	邮箱1无错发送完成 当发送结束并且没有错误产生时，该位由硬件置1。由软件写1清0或对CAN_TSTAT寄存器中MTF1写1清0。也可以在无错传输结束时由硬件清0。 0：传输结束时发生了错误 1：传输结束且没有错误
8	MTF1	邮箱1发送完成 当发送完成或被中止时，该位由硬件置1。由软件写1清0，或当CAN_TMI1寄存器的TEN被置位时清0。 0：发送邮箱1正在发送 1：发送邮箱1完成发送
7	MST0	邮箱0停止发送 将其置1，将停止邮箱0的发送过程。 当邮箱0变为emtpy状态时，该位被硬件自动清0。

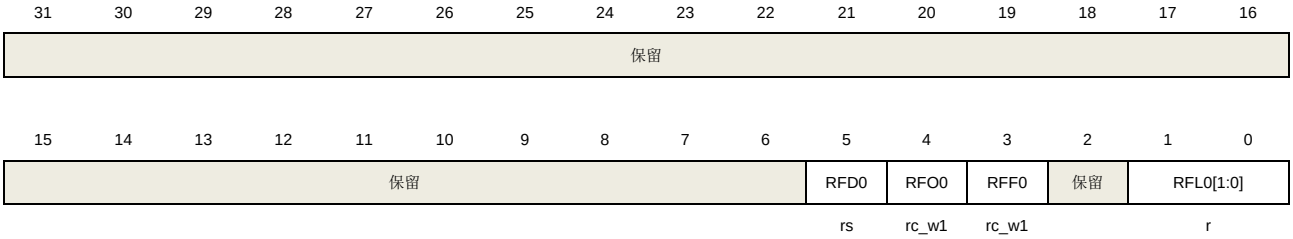
6:4	保留	必须保持复位值。
3	MTE0	邮箱0发送错误 当发生发送错误时，该位由硬件置1。由软件写1清0或对CAN_TSTAT寄存器中MTF0写1清0。也可以在下一次发送开始时由硬件清0。 当发生错误时该位被置1。
2	MAL0	邮箱0仲裁失败 当发生发送错误时，该位由硬件置1。由软件写1清0或对CAN_TSTAT寄存器中MTF0写1清0。也可以在下一次发送开始时由硬件清0。 当发生仲裁失败时该位被置1。
1	MTFNERR0	邮箱0无错发送完成 当发送结束并且没有错误产生时，该位由硬件置1。由软件写1清0或对CAN_TSTAT寄存器中MTF0写1清0。也可以在无错传输结束时由硬件清0。 0：传输结束时发生了错误 1：传输结束且没有错误
0	MTF0	邮箱0发送完成 当发送完成或被中止时，该位由硬件置1。由软件写1清0，或当CAN_TMI0寄存器的TEN被置位时清0。 0：发送邮箱0正在发送 1：发送邮箱0完成发送

25.4.4. 接收 FIFO0 寄存器（CAN_RFIFO0）

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
31:6	保留	必须保持复位值。
5	RFD0	释放一次FIFO0中的数据 该位被置1， 将释放FIFO0中的一帧数据。 FIFO释放相应的数据空间后，该位被清0。
4	RFO0	接收FIFO0溢出 当接收FIFO0溢出时被置位，由软件写1清0。 0：接收FIFO0没有溢出

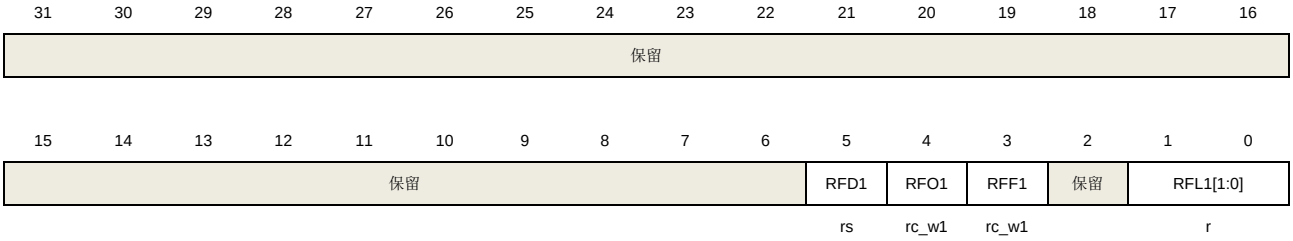
		1：接收FIFO0溢出
3	RFF0	接收FIFO0满 当接收FIFO0满时被置位，由软件写1清0。 0：接收FIFO0不满 1：接收FIFO0满
2	保留	必须保持复位值。
1:0	RFL0[1:0]	接收FIFO0中帧的数量

25.4.5. 接收 FIFO1 寄存器（CAN_RFIFO1）

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
31:6	保留	必须保持复位值。
5	RFD1	释放一次FIFO1中的数据 该位被置1， 将释放FIFO1中的一帧数据。 FIFO释放相应的数据空间后，该位被清0。
4	RFO1	接收FIFO1溢出 当接收FIFO1溢出时被置位，由软件写1清0。 0：接收FIFO1没有溢出 1：接收FIFO1溢出
3	RFF1	接收FIFO1满 当接收FIFO1满时被置位，由软件写1清0。 0：接收FIFO1不满 1：接收FIFO1满
2	保留	必须保持复位值。
1:0	RFL1[1:0]	接收FIFO1中帧的数量

25.4.6. 中断使能寄存器（CAN_INTEN）

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														SLPWIE	WIE
														rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIE	保留			ERRNIE	BOIE	PERRIE	WERRIE	保留	RFOIE1	RFFIE1	RFNEIE1	RFOIE0	RFFIE0	RFNEIE0	TMEIE
rw				rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:18	保留	必须保持复位值。
17	SLPWIE	睡眠中断使能 0：禁用睡眠中断 1：使能睡眠中断
16	WIE	唤醒中断使能 0：禁用唤醒中断 1：使能唤醒中断
15	ERRIE	错误中断使能 0：禁用错误中断 1：使能错误中断
14:12	保留	必须保持复位值。
11	ERRNIE	错误种类中断使能 0：禁用错误种类中断 1：使能错误种类中断
10	BOIE	离线中断使能 0：禁用离线中断 1：使能离线中断
9	PERRIE	被动错误中断使能 0：禁用被动错误 1：使能被动错误
8	WERRIE	警告错误中断使能 0：禁用警告错误中断 1：使能警告错误中断
7	保留	必须保持复位值。
6	RFOIE1	接收FIFO1溢出中断使能 0：禁用接收FIFO1溢出中断 1：使能接收FIFO1溢出中断
5	RFFIE1	接收FIFO1满中断使能

		0：禁用接收FIFO1满中断 1：使能接收FIFO1满中断
4	RFNEIE1	接收FIFO1非空中断使能 0：禁用接收FIFO1非空中断 1：使能接收FIFO1非空中断
3	RFOIE0	接收FIFO0溢出中断使能 0：禁用接收FIFO0溢出中断 1：使能接收FIFO0溢出中断
2	RFFIE0	接收FIFO0满中断使能 0：禁用接收FIFO0满中断 1：使能接收FIFO0满中断
1	RFNEIE0	接收FIFO0非空中断使能 0：禁用接收FIFO0非空中断 1：使能接收FIFO0非空中断
0	TMEIE	发送邮箱空中断使能 0：禁用发送邮箱空中断 1：使能发送邮箱空中断

25.4.7. 错误寄存器（CAN_ERR）

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
RECNT[7:0]								TECNT[7:0]							
r								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									ERRN[2:0]			保留	BOERR	PERR	WERR
									rw				r	r	r

位/位域	名称	描述
31:24	RECNT[7:0]	接收错误计数值
23:16	TECNT[7:0]	发送错误计数值
15:7	保留	必须保持复位值。
6:4	ERRN[2:0]	错误种类 ERRN由硬件更新，可以反映位传输过程中的错误情况。当位传输成功没有错误时，ERRN为0。软件可以设置ERRN为0b111。 000：无错误 001：填充错误

010：格式错误
011：ACK错误
100：位隐性错
101：位显性错误
110：CRC错误
111：软件设置值

3	保留	必须保持复位值。
2	BOERR	离线错误 当TEC 上溢（超过 255）时，CAN总线控制器进入离线状态，该位被置1。
1	PERR	被动错误 当TECNT或者RECNT大于127时，该位由硬件置1。
0	WERR	警告错误 当TECNT或RECNT大于等于96时，该位由硬件置1。

25.4.8. 位时序寄存器（CAN_BT）

地址偏移：0x1C
复位值：0x0123 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
SCMOD	LCMOD	保留	SJW[4:0]					保留	BS2[2:0]			BS1[3:0]			
rw	rw		rw						rw			rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	BS2[4:3]		BS1[6:4]												
	rw		rw												

位/位域	名称	描述
31	SCMOD	静默通信模式 0：禁用静默通信模式 1：使能静默通信模式
30	LCMOD	回环通信模式 0：禁用回环通信模式 1：使能回环通信模式
29	保留	必须保持复位值。
28:24	SJW[4:0]	再同步补偿宽度 再同步补偿占用的时间单元数量= SJW[4:0]+1
23	保留	必须保持复位值。
22:20	BS2[2:0]	位段2的2:0位

		位段2占用的时间单元数量=BS2[4:0]+1
19:16	BS1[3:0]	位段1的3:0位 位段1占用的时间单元数量=BS1[6:0]+1
15	保留	必须保持复位值。
14:13	BS2[4:3]	位段2的4:3位 参考CAN_BT寄存器的22:20位 当CAN_FDCTL寄存器中的FDEN=1时，此位域才能被修改。
12:10	BS1[6:4]	位段1的6:4位 参考CAN_BT寄存器的19:16位 当CAN_FDCTL寄存器中的FDEN=1时，此位域才能被修改。
9:0	BAUDPSC[9:0]	波特率分频系数

25.4.9. FD 控制寄存器（CAN_FDCTL）

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									ESIMOD	TDCMOD	TDCEN	NSIO	PRED	保留	FDEN
									rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:7	保留	必须保持复位值。
6	ESIMOD	错误状态指示模式 0：发送帧的ESI位始终显示节点错误状态。主动错误状态节点发送显性位，被动错误状态节点发送隐性位 1：当节点处于主动错误状态时，通过CAN_TMPx寄存器的ESI位值决定发送帧的ESI位显示的错误状态。当节点处于被动错误状态时，发送帧该位发送隐性位。
5	TDCMOD	传输延迟补偿模式 0：测量值加软件偏移 1：仅使用软件偏移
4	TDCEN	传输延迟补偿使能 0：传输延迟补偿禁止 1：传输延迟补偿使能
3	NSIO	ISO/BOSCH

		0：ISO 1：BOSCH
2	PRED	协议异常事件检测禁用 0：协议异常事件检测使能（当接收到隐性保留位时转为IDLE状态） 1：协议异常事件检测禁用（当接收到隐性保留位时，将此视为帧格式错误）
1	保留	必须保持复位值。
0	FDEN	FD功能使能 0：CAN-FD功能禁止 1：CAN-FD功能使能

25.4.10. FD 状态寄存器（CAN_FDSTAT）

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															PRE
															rc_w1

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

位/位域	名称	描述
31:17	保留	必须保持复位值。
16	PRE	协议异常事件 当协议异常事件被检测到时该位由硬件置1，软件写1可清0该位。
15:7	保留	必须保持复位值。
6:0	TDCV[6:0]	传输延迟补偿值 这些位由硬件控制，显示硬件自动计算的传输延迟补偿值。

25.4.11. FD 传输延迟补偿寄存器（CAN_FDTDC）

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

保留	TDCO[6:0]	保留	TDCF[6:0]
	rw		rw

位/位域	名称	描述
31:15	保留	必须保持复位值。
14:8	TDCO[6:0]	传输延迟补偿偏移 这些位被用于设置次级采样点（SSP）基于测量的补偿时间的偏移，测量的补偿时间是由硬件计算信号从CAN_TX发出到从CAN_RX接收到的过程的延迟时间得出的。
7	保留	必须保持复位值。
6:0	TDCF[6:0]	传输延迟补偿过滤器 这些位定义了次级采样点（SSP）的最小值。当配置的TDCF值大于TDCV时，这些位的值用于替代TDCV，以过滤脉冲毛刺引起的测量误差导致的SSP采样点过早的情况发生。

25.4.12. 数据位时序寄存器（CAN_DBT）

地址偏移：0x2C
复位值：0x0123 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					DSJW[2:0]			保留	DBS2[2:0]			DBS1[3:0]			
					rw				rw			rw			

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

位/位域	名称	描述
31:27	保留	必须保持复位值。
26:24	DSJW[2:0]	再同步补偿宽度 再同步补偿占用的时间单元数量= SJW[2:0]+1
23	保留	必须保持复位值。
22:20	DBS2[2:0]	位段2的2:0位 位段2占用的时间单元数量=BS2[2:0]+1
19:16	DBS1[3:0]	位段1的3:0位 位段1占用的时间单元数量=BS1[3:0]+1
15:10	保留	必须保持复位值。
9:0	DBAUDPSC[9:0]	波特率分频系数

25.4.13. 发送邮箱标识符寄存器（CAN_TMIx）（x = 0…2）

地址偏移：0x180 + 0x10 * x

复位值：0xXXXX XXXX （bit0 = 0）

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
SFID[10:0]/EFID[28:18]											EFID[17:13]				
rw											rw				

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

位/位域	名称	描述
31:21	SFID[10:0]/EFID[28:18]	标识符 SFID[10:0]：标准格式帧标识符 EFID[28:18]：扩展格式帧标识符
20:16	EFID[17:13]	标识符 EFID[17:13]：扩展格式帧标识符
15:3	EFID[12:0]	标识符 EFID[12:0]：扩展格式帧标识符
2	FF	帧格式 0：标准格式帧 1：扩展格式帧
1	FT	帧种类 0：数据帧 1：遥控帧
0	TEN	发送使能 当应用程序想要发送数据时，该位被置1将启动发送过程。当发送结束，发送邮箱为空时，该位由硬件清0。 0：禁用发送 1：使能发送

25.4.14. 发送邮箱属性寄存器（CAN_TMPx）（x = 0...2）

地址偏移：0x184 + 0x10 * x

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
TS[15:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							TSEN	FDF	保留	BRS	ESI	DLENC[3:0]			
							rw	rw		rw	rw	rw			

位/位域	名称	描述
31:16	TS[15:0]	时间戳 发送时间戳
15:9	保留	必须保持复位值。
8	TSEN	时间戳使能 0：禁用时间戳 1：使能时间戳。时间戳TS[15:0]将放在寄存器CAN_TMDATA1的DATA6和DATA7中 只有当寄存器CAN_CTL中的TTC为1时，该位才有效。
7	FDF	CAN-FD帧标志位 0：常规帧（标准帧或扩展帧） 1：CAN-FD帧
6	保留	必须保持复位值。
5	BRS	位速率转换开关 0：不转换速率 1：转换可变速率
4	ESI	错误状态指示 仅当CAN_FDCTL寄存器的ESIMOD位为1时，该位有效 0：发送帧中的ESI位将发送显性位 1：发送帧中的ESI位将发送隐性位
3:0	DLENC[3:0]	数据长度，DLENC[3:0]表示帧内数据长度。

25.4.15. 发送邮箱 data0 寄存器（CAN_TMDATA0x）（x = 0...2）

地址偏移：0x188 + 0x10 * x
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
DB3[7:0]								DB2[7:0]							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB1[7:0]								DB0[7:0]							
rw								rw							

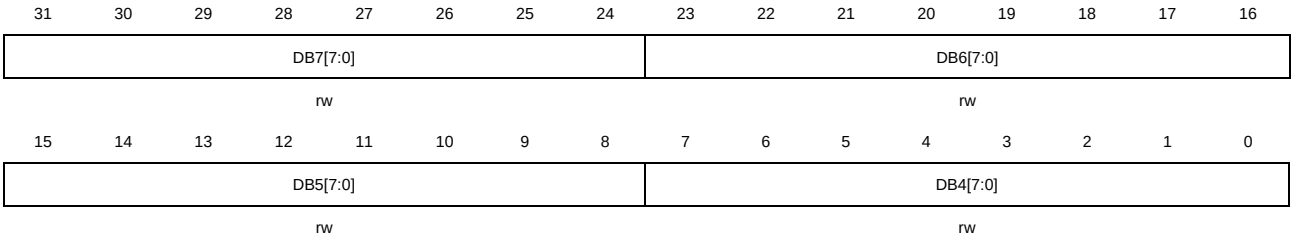
位/位域	名称	描述

31:24	DB3[7:0]	字节3
23:16	DB2[7:0]	字节2
15:8	DB1[7:0]	字节1
7:0	DB0[7:0]	字节0

25.4.16. 发送邮箱 data1 寄存器（CAN_TMDATA1x）（x = 0...2）

地址偏移：0x18C + 0x10 * x
复位值：0xXXXX XXXX

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:24	DB7[7:0]	字节7
23:16	DB6[7:0]	字节6
15:8	DB5[7:0]	字节5
7:0	DB4[7:0]	字节4

25.4.17. 接收 FIFO 邮箱标识符寄存器（CAN_RFIFOMIx）（x = 0,1）

地址偏移：0x1B0 + 0x10 * x
复位值：0xXXXX XXXX

该寄存器只能按字（32 位）访问。

位/位域	区域	说明
31:21	SFID[10:0]/EFID[28:18]	标识符 SFID[10:0]：标准格式帧标识符

		EFID[28:18]：扩展格式帧标识符
20:16	EFID[17:13]	标识符 EFID[17:13]：扩展格式帧标识符
15:3	EFID[12:0]	标识符 EFID[12:0]：扩展格式帧标识符
2	FF	帧格式 0：标准格式帧 1：扩展格式帧
1	FT	帧种类 0：数据帧 1：遥控帧
0	保留	必须保持复位值。

25.4.18. 接收 FIFO 邮箱属性寄存器（CAN_RFIFOMPx）（x = 0,1）

地址偏移：0x1B4 + 0x10 * x
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
TS[15:0]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FI[7:0]								FDF	保留	BRS	ESI	DLENC[3:0]			
r								rw		rw	rw	r			

位/位域	名称	描述
31:16	TS[15:0]	时间戳 接收时间戳
15:8	FI[7:0]	过滤索引 帧通过过滤器时的过滤序号
7	FDF	CAN-FD帧标志位 0：常规帧（标准帧或扩展帧） 1：CAN-FD帧
6	保留	必须保持复位值。
5	BRS	位速率转换开关 0：不转换速率 1：转换可变速率
4	ESI	错误状态指示

仅当CAN_FDCTL寄存器的ESIMOD位为1时，该位有效
0：发送帧中的ESI位将发送显性位
1：发送帧中的ESI位将发送隐性位

3:0	DLENC[3:0]	数据长度 DLENC[3:0]表示帧内数据长度。

25.4.19. 接收 FIFO 邮箱 data0 寄存器（CAN_RFIFOMDATA0x）（x = 0,1）

地址偏移：0x1B8 + 0x10 * x
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
DB3[7:0]								DB2[7:0]							
r								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB1[7:0]								DB0[7:0]							
r								r							

位/位域	名称	描述
31:24	DB3[7:0]	字节3
23:16	DB2[7:0]	字节2
15:8	DB1[7:0]	字节1
7:0	DB0[7:0]	字节0

25.4.20. 接收 FIFO 邮箱 data1 寄存器（CAN_RFIFOMDATA1x）（x = 0,1）

地址偏移：0x1BC + 0x10 * x
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
DB7[7:0]								DB6[7:0]							
r								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB5[7:0]								DB4[7:0]							
r								r							

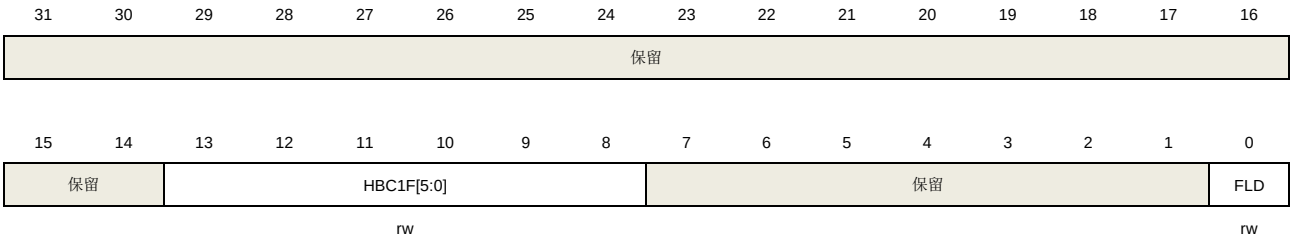
位/位域	名称	描述
31:24	DB7[7:0]	字节7

23:16	DB6[7:0]	字节6
15:8	DB5[7:0]	字节5
7:0	DB4[7:0]	字节4

25.4.21. 过滤器控制寄存器（CAN_FCTL）

地址偏移：0x200
复位值：0x2A1C 0E01

该寄存器只能按字（32 位）访问。



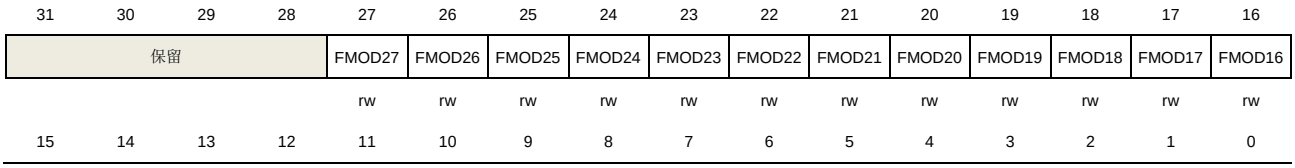
位/位域	名称	描述
31:14	保留	必须保持复位值。
13:8	HBC1F[5:0]	CAN1过滤器单元起始位置 这些位用来定义CAN1过滤器起始位置。CAN0可以用编号为0~HBC1F-1过滤器，CAN1可以用编号为HBC1F~27过滤器。当这些位的值为0，CAN0将没有过滤器可以使用。当这些位的值为28时，CAN1将没有过滤器可以使用。 此位域在CAN2中为保留位。
7:1	保留	必须保持复位值。
0	FLD	过滤器锁禁用 0：使能过滤器锁 1：禁用过滤器锁

25.4.22. 过滤器模式配置寄存器（CAN_FMCFG）

地址偏移：0x204
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。修改此寄存器的值，需要将 CAN_FCTL 寄存器的 FLD 位置 1。

在 CAN0 和 CAN1 中的 CAN_FMCFG 寄存器：



FMOD15	FMOD14	FMOD13	FMOD12	FMOD11	FMOD10	FMOD9	FMOD8	FMOD7	FMOD6	FMOD5	FMOD4	FMOD3	FMOD2	FMOD1	FMOD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:0	FMODx	过滤器模式 0：掩码模式 1：列表模式

在 CAN2 中的 CAN_FMCFG 寄存器：

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		FMOD13	FMOD12	FMOD11	FMOD10	FMOD9	FMOD8	FMOD7	FMOD6	FMOD5	FMOD4	FMOD3	FMOD2	FMOD1	FMOD0
		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。
13:0	FMODx	过滤器模式 0：掩码模式 1：列表模式

25.4.23. 过滤器位宽配置寄存器（CAN_FSCFG）

地址偏移：0x20C
复位值：0x0000 0000

该寄存器只能按字（32 位）访问，修改此寄存器的值，需要将 CAN_FCTL 寄存器的 FLD 位置 1。

在 CAN0 和 CAN1 中的 CAN_FSCFG 寄存器：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				FS27	FS26	FS25	FS24	FS23	FS22	FS21	FS20	FS19	FS18	FS17	FS16
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FS15	FS14	FS13	FS12	FS11	FS10	FS9	FS8	FS7	FS6	FS5	FS4	FS3	FS2	FS1	FS0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:0	FSx	过滤器位宽 0：16-bit位宽

1：32-bit位宽

在 CAN2 中的 CAN_FSCFG 寄存器：

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		FS13	FS12	FS11	FS10	FS9	FS8	FS7	FS6	FS5	FS4	FS3	FS2	FS1	FS0
		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。
13:0	FSx	过滤器位宽 0：16-bit位宽 1：32-bit位宽

25.4.24. 过滤器关联 FIFO 寄存器（CAN_FAFIFO）

地址偏移：0x214
复位值：0x0000 0000

该寄存器只能按字（32 位）访问，修改此寄存器的值，需要将 CAN_FCTL 寄存器的 FLD 位置 1。

在 CAN0 和 CAN1 中的 CAN_FAFIFO 寄存器：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				FAF27	FAF26	FAF25	FAF24	FAF23	FAF22	FAF21	FAF20	FAF19	FAF18	FAF17	FAF16
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAF15	FAF14	FAF13	FAF12	FAF11	FAF10	FAF9	FAF8	FAF7	FAF6	FAF5	FAF4	FAF3	FAF2	FAF1	FAF0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:0	FAFx	过滤器关联FIFO 0：关联FIFO0 1：关联FIFO1

在 CAN2 中的 CAN_FSCFG 寄存器：

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	FAF13	FAF12	FAF11	FAF10	FAF9	FAF8	FAF7	FAF6	FAF5	FAF4	FAF3	FAF2	FAF1	FAF0
	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。
13:0	FAFx	过滤器关联FIFO 0：关联FIFO0 1：关联FIFO1

25.4.25. 过滤器激活寄存器（CAN_FW）

地址偏移：0x21C
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

在 CAN0 和 CAN1 中的 CAN_FAFIFO 寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				FW27	FW26	FW25	FW24	FW23	FW22	FW21	FW20	FW19	FW18	FW17	FW16
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FW15	FW14	FW13	FW12	FW11	FW10	FW9	FW8	FW7	FW6	FW5	FW4	FW3	FW2	FW1	FW0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:0	FWx	过滤器激活 0：没有激活 1：激活工作

在 CAN2 中的 CAN_FSCFG 寄存器：

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		FW13	FW12	FW11	FW10	FW9	FW8	FW7	FW6	FW5	FW4	FW3	FW2	FW1	FW0
		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。
13:0	FWx	过滤器激活

0：没有激活

1：激活工作

25.4.26. 过滤器（x）数据（y）寄存器（CAN_FxDATAy）（x = 0…27, y = 0,1）

地址偏移：0x240+8 * x+4 * y, (在 CAN2 中：x = 0…13, y = 0,1）

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
FD31	FD30	FD29	FD28	FD27	FD26	FD25	FD24	FD23	FD22	FD21	FD20	FD19	FD18	FD17	FD16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:0	FDx	过滤器数据 掩码模式下： 0：标识符的Bit(x)不需参与比较 1：标识符的Bit(x)需要参与比较 列表模式下： 0：标识符的Bit(x)必须为0 1：标识符的Bit(x)必须为1

26. 以太网（ENET）

26.1. 简介

本章描述以太网外设模块。该以太网模块包含 10/100Mbps 以太网 MAC（媒体访问控制器），采用 DMA 优化数据帧的发送与接收性能，支持 MII（媒体独立接口）与 RMII（简化的媒体独立接口）两种与物理层(PHY)通讯的标准接口，实现以太网数据帧的发送与接收。以太网模块遵守 IEEE 802.3-2002 标准和 IEEE 1588-2008 标准。

26.2. 主要特性

MAC 特性

- 支持10Mbps或100Mbps数据传输速率；
- 支持MII和RMII接口；
- 支持调试用回环模式；
- 支持符合CSMA/CD协议的半双工背压通讯；
- 支持符合IEEE 802.3x的流控通讯。在当前帧发送完毕后，根据接收的暂停帧中暂停时间延迟发送。在全双工/半双工模式下，MAC根据RxFIFO的填充程度自动发送暂停帧/背压信号；
- 支持符合IEEE 802.3x的全双工流控通讯，当输入流控信号失效时，自动发送零时间片暂停帧。支持符合IEEE 802.3x的半双工流控通讯，支持根据RxFIFO的填充程度（直通模式）自动发送背压信号；
- 可选择在发送操作时自动生成校验/填充位；
- 可选择在接收操作时自动去除校验/填充位；
- 帧长度可配置；
- 帧间隙可配置；
- 支持多种模式的接收过滤；
- 支持检测接收帧的IEEE 802.1Q VLAN标签；
- 支持强制网络统计标准（RFC2819/RFC2665）；
- 支持两种唤醒帧检测：LAN远程唤醒帧和AMD的Magic Packet™帧；
- 支持校验和检查（IPv4报头，IPv4或IPv6数据格式封装的TCP、UDP或ICMP）；
- 支持IEEE 1588-2008标准定义的以太网帧时间戳，并将其按64位记录于帧状态中；
- 相互独立的两个FIFO分别用于发送与接收；
- 在延迟冲突、过度冲突、过度顺延和下溢情况下丢弃帧；
- 帧传输时，支持存储转发模式下的硬件校验和的计算和插入。

DMA 特性

- 支持环结构或链结构两种形式的描述符列表；
- 每个描述符可以传输最高为8192字节的数据；
- 中断可配置，适用于多种工作状态；
- 支持轮询或固定优先级两种方式仲裁DMA发送和接收控制器的请求。

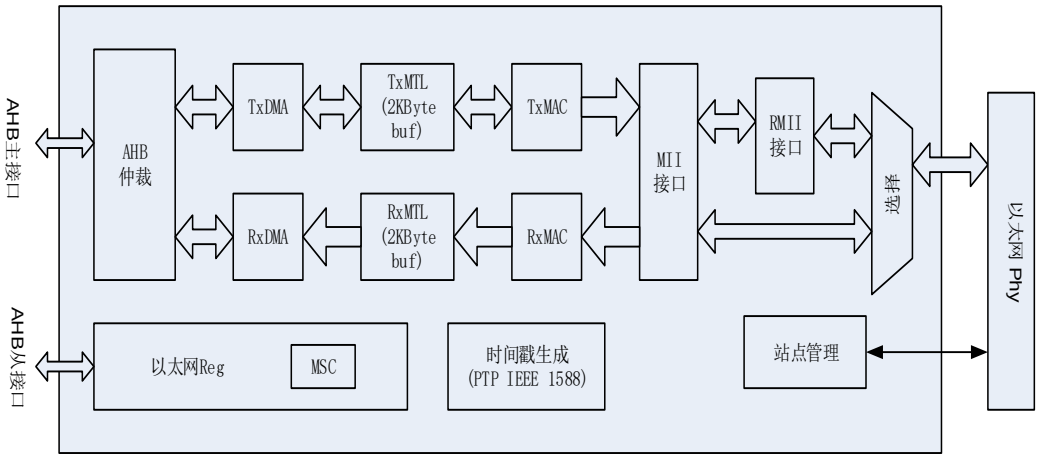
PTP 特性

- 支持符合IEEE1588的时间同步功能；
- 支持粗/精调两种校正方法；
- 输出秒脉冲；
- 达到预设目标时间时触发中断。

26.2.1. 模块框图

以太网模块由 MAC（介质访问控制器）模块、MII/RMII 模块和一个以描述符形式控制的 DMA 模块组成。在使用以太网模块时，用户应确保配置的时钟频率不小于 25MHz。

图 26-1. 以太网模块框图



MAC 模块通过 MII 或 RMII 与片外 PHY 连接。通过对 AFIO_PCF0 寄存器的 ENET_PHY_SEL 位进行设置，可以选择使用哪种接口。SMI（站点管理接口）用于配置和管理外部 PHY。

发送数据模块包括：

- TxDMA控制器，用于从存储器中读取描述符和数据，以及将状态写入存储器；
- TxMTL，用于对发送数据的控制，管理和存储。TxMTL内含TxFIFO，用于缓存待MAC发送的数据；
- MAC发送控制寄存器组，用于管理和控制数据帧的发送。

接收数据模块包括：

- RxDMA控制器，用于从存储器中读取描述符，以及将数据与状态写入存储器；
- RxMTL，用于对接收数据的控制，管理和存储。RxMTL实现了RxFIFO，用于存储待转发到系统存储的帧数据；
- MAC接收控制寄存器组，用于管理数据帧的接收和标示接收状态。MAC内含接收过滤器，采用多种过滤机制，滤除特定的以太网帧。

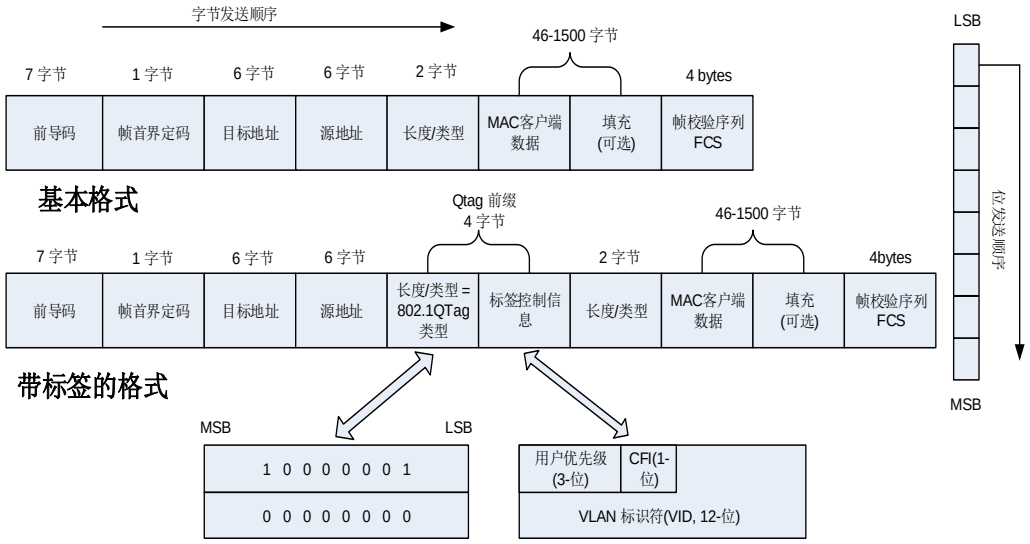
26.2.2. MAC 802.3 以太网数据包描述

MAC 的数据通信可使用两种帧格式：

- 基本帧格式；
- 带标签的帧格式。

图27-2. MAC/带标签的MAC帧格式描述了帧结构（基本的和带标签的）：

图 26-2. MAC/带标签的 MAC 帧格式



注意：除了帧校验序列，以太网控制器发送每个字节时都按照低位先出的次序进行传输。

CRC 计算包括帧数据的所有字节除去前导码和帧首界定码域。以太网帧的 32 位 CRC 生成多项式为 0x04C11DB7，且此多项式用于以太网模块中所有的 32 位 CRC 计算，如下式所示：

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

26.2.3. 以太网信号描述

表26-1. 以太网信号（MII默认）、*表26-2. 以太网信号（MII重映射）*、*表26-3. 以太网信号（RMII默认）*和*表26-4. 以太网信号（RMII重映射）*列出了MAC模块所用引脚在MII/RMII模式下默认及重映射的功能和具体配置。

表 26-1. 以太网信号（MII 默认）

信号	引脚	引脚模式
MDC	PC1	推挽复用输出
MII_TXD2	PC2	推挽复用输出
MII_TX_CLK	PC3	浮空输入(复位状态)
MII_CRS	PA0	浮空输入(复位状态)
MII_RX_CLK	PA1	浮空输入(复位状态)
MDIO	PA2	推挽复用输出
MII_COL	PA3	浮空输入(复位状态)

信号	引脚	引脚模式
MII_RX_DV	PA7	浮空输入(复位状态)
MII_RXD0	PC4	浮空输入(复位状态)
MII_RXD1	PC5	浮空输入(复位状态)
MII_RXD2	PB0	浮空输入(复位状态)
MII_RXD3	PB1	浮空输入(复位状态)
PPS_OUT	PB5	推挽复用输出
MII_TXD3	PB8	推挽复用输出
MII_RX_ER	PB10	浮空输入(复位状态)
MII_TX_EN	PB11	推挽复用输出
MII_TXD0	PB12	推挽复用输出
MII_TXD1	PB13	推挽复用输出

表 26-2. 以太网信号（MII 重映射）

信号	引脚	引脚模式
MDC	PC1	推挽复用输出
MII_TXD2	PC2	推挽复用输出
MII_TX_CLK	PC3	浮空输入(复位状态)
MII_CRS	PA0	浮空输入(复位状态)
MII_RX_CLK	PA1	浮空输入(复位状态)
MDIO	PA2	推挽复用输出
MII_COL	PA3	浮空输入(复位状态)
MII_RX_DV	PD8	浮空输入(复位状态)
MII_RXD0	PD9	浮空输入(复位状态)
MII_RXD1	PD10	浮空输入(复位状态)
MII_RXD2	PD11	浮空输入(复位状态)
MII_RXD3	PD12	浮空输入(复位状态)
PPS_OUT	PB5	推挽复用输出
MII_TXD3	PB8	推挽复用输出
MII_RX_ER	PB10	浮空输入(复位状态)
MII_TX_EN	PB11	推挽复用输出
MII_TXD0	PB12	推挽复用输出
MII_TXD1	PB13	推挽复用输出

表 26-3. 以太网信号（RMII 默认）

信号	引脚	引脚模式
MDC	PC1	推挽复用输出
REF_CLK	PA1	浮空输入(复位状态)
MDIO	PA2	推挽复用输出
CRS_DV	PA7	浮空输入(复位状态)
RMII_RXD0	PC4	浮空输入(复位状态)
RMII_RXD1	PC5	浮空输入(复位状态)
PPS_OUT	PB5	推挽复用输出
RMII_TX_EN	PB11	推挽复用输出

信号	引脚	引脚模式
RMII_TXD0	PB12	推挽复用输出
RMII_TXD1	PB13	推挽复用输出

表 26-4. 以太网信号（RMII 重映射）

信号	引脚	引脚模式
MDC	PC1	推挽复用输出
REF_CLK	PA1	浮空输入(复位状态)
MDIO	PA2	推挽复用输出
CRS_DV	PD8	浮空输入(复位状态)
RMII_RXD0	PD9	浮空输入(复位状态)
RMII_RXD1	PD10	浮空输入(复位状态)
PPS_OUT	PB5	推挽复用输出
RMII_TX_EN	PB11	推挽复用输出
RMII_TXD0	PB12	推挽复用输出
RMII_TXD1	PB13	推挽复用输出

26.3. 功能描述

26.3.1. 接口配置

以太网模块通过 MII/RMII 接口与片外 PHY 连接，传送与接收以太网包。MII 或 RMII 模式由软件选择并通过 SMI 接口对 PHY 进行管理。

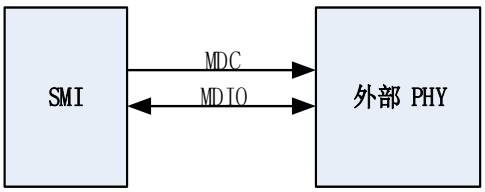
SMI：站点管理接口

SMI 用于访问和设置 PHY 的配置。

站点管理接口（SMI）通过 MDC 时钟线与 MDIO 数据线与外部 PHY 通讯，可以通过其访问任意 PHY 的任意寄存器。SMI 接口支持的最大 PHY 数量为 32，但在同一时刻只能访问一个 PHY 的一个寄存器。MDC 时钟线和 MDIO 数据线具体作用如下：

- MDC：最高频率为2.5MHz的时钟信号，在空闲状态下该引脚保持为低电平状态。在传输数据时该信号的高电平和低电平的最短保持时间为160ns，信号的最小周期为400ns；
- MDIO：用于与PHY之间的数据传输，与MDC时钟线配合，接收/发送数据。

图 26-3. 站点管理接口信号



写操作

应用程序需将要传输的数据写入 ENET_MAC_PHY_DATA 寄存器中，并对 ENET_MAC_PHY_CTL 寄存器相关位进行操作：

1. 设置PHY设备地址和将要操作的PHY寄存器地址，并将PW位置为1，使能写模式；
2. 将PB位置1开始传输。在传输过程中PB位一直为高，直到传输完成，硬件将会自动清除PB位。

应用程序可以通过 PB 位判断传输是否完成。在 PB 位置 1 期间，由于操作正在运行，因此不能修改 PHY 控制寄存器和 PHY 数据寄存器的内容。在将 PB 位置位之前，应用程序必须确保该位读出为’0’。

读操作

应用程序对 ENET_MAC_PHY_CTL 寄存器相关位进行操作：

1. 设置PHY设备地址和将要操作的PHY寄存器地址，并将PW位置为0，使能读模式；
2. 将PB位置1开始数据接收。在接收过程中PB位一直为高，直到接收完成，硬件将会自动清除PB位。

应用程序可以通过 PB 位判断传输是否完成。在 PB 位置 1 期间，由于操作正在运行，因此不能修改 PHY 控制寄存器和 PHY 数据寄存器的内容。在将 PB 位置位之前，应用程序必须确保该位读出为’0’。

注意：由于 PHY 寄存器地址 16-31 的寄存器内容由各厂商自由定义，所以在访问不同 PHY 设备的这部分寄存器时，需要根据厂商手册对应用程序进行不同的设置。GD32A508xx 固件库当前支持的 PHY 设备详情请参考固件库相关手册说明。

时钟配置

SMI 接口的时钟源由 AHB 时钟分频得到。为了保证 MDC 时钟频率不超过 2.5MHz，需根据 AHB 时钟频率对 PHY 控制寄存器中相关位进行设置，选择合适的分频系数。*表 26-5. 时钟范围*列出了对应 AHB 时钟范围的分频系数的选择。

表 26-5. 时钟范围

AHB时钟	MDC 时钟	ENET_MAC_PHY_CTL 位 CLR[2:0]
150~180MHz	AHB clock/102	0x4
35~60MHz	AHB clock/26	0x3
20~35MHz	AHB clock/16	0x2
100~150 MHz	AHB clock/62	0x1
60~100MHz	AHB clock/42	0x0

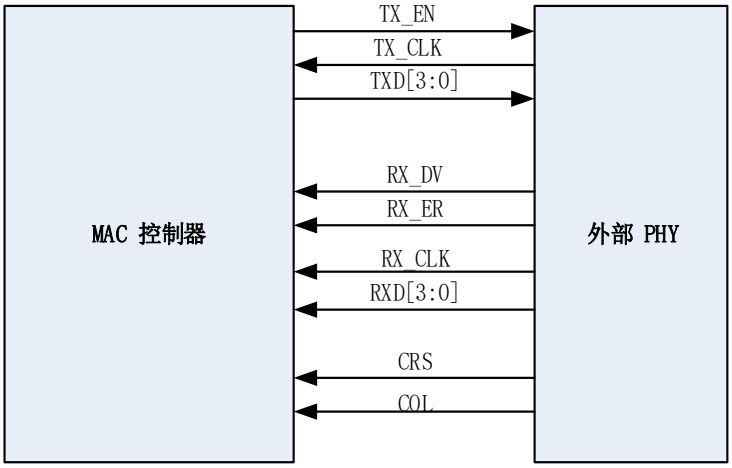
MII/RMII 的选择

当使能以太网控制器时钟前或以太网控制器处于复位状态时，应用程序可通过配置 AFIO_PCF0 寄存器中的 ENET_PHY_SEL 位来选择使用 MII 或者 RMII 模式，默认为 MII 模

式。

MII：媒体独立接口

图 26-4. 媒体独立接口（MII）信号线



- **MII_TX_CLK**：发送数据使用的时钟信号，对于 10Mbit/s 的数据传输，此时钟为 2.5MHz，对于 100M bit/s 的数据传输，此时钟为 25MHz。

- **MII_RX_CLK**：接收数据使用的时钟信号，对于 10Mbit/s 的数据传输，此时钟为 2.5MHz，对于 100M bit/s 的数据传输，此时钟为 25MHz。

- **MII_TX_EN**：发送使能信号，，当数据前导码的起始位出现时，该信号必须有效，并且需要在传输完毕前保持有效。

- **MII_TXD[3:0]**：发送数据线，每次传输 4 位数据，数据在 MII_TX_EN 信号有效时有效。当 MII_TX_EN 信号无效时，PHY 将忽略传输的数据。

- **MII_CRS**：载波侦听信号，仅工作在半双工模式下，由 PHY 控制。该信号不需要与 MII_TX_CLK 和 MII_RX_CLK 保持同步。当它处于有效状态时，意味着发送或接收介质不处于空闲状态。MII_CRS 信号一直保持有效，直到发送和接收介质都处于空闲状态。

- **MII_COL**：冲突检测信号，仅工作在半双工模式下，由 PHY 控制。该信号不需要与 MII_TX_CLK 和 MII_RX_CLK 保持同步。当检测到介质发生冲突时，此信号有效，并且在整个冲突的持续时间内，保持此信号有效。

- **MII_RXD[3:0]**：接收数据线，每次接收 4 位数据，数据在 MII_RX_DV 信号有效时有效。根据 MII_RX_DV 和 MII_RX_ER 信号的状态，MII_RXD[3:0]数据值可被用来传达一些特定信息（请参考*表 26-6. 接收接口信号编码*）。

- **MII_RX_DV**：接收数据使能信号，由 PHY 控制，当 PHY 准备好数据供 MAC 接收时，该信号有效。当数据帧的第一个 4 位出现时，该信号必须有效，并且需要在传输完毕前保持有效。在传送最后 4 位数据后的第一个时钟之前，此信号必须变为无效状态。为确保正确地接收帧，MII_RX_DV 信号应该在 SFD 字段出现之前有效。

- **MII_RX_ER**：接收出错信号，为了表明 MAC 在接收过程中检测到错误，MII_RX_ER 信号必须在一个或多个时钟周期（MII_RX_CLK）内保持有效。具体错误原因需结合 MII_RX_DV 的

状态及 MII_RXD[3:0]的数据值，详见*表 26-6. 接收接口信号编码*。

表 26-6. 接收接口信号编码

信号	正常的帧间隔		正常的数据接收	载波错误指示	数据接收出错
MII_RX_ER	0	1	0	1	1
MII_RX_DV	0	0	1	0	1
MII_RXD[3:0]	0000 to 1111	0000	0000 to 1111	1110	0000 to 1111

MII 时钟源

用户需要给外部 PHY 提供一个外部的 25MHz 时钟来产生 TX_CLK 和 RX_CLK 时钟信号。该时钟不需要与 MAC 时钟相同。可以使用外部的 25MHz 晶振或者微控制器的时钟输出引脚 CK_OUT0 提供这一时钟。当时钟来源为 CK_OUT0 引脚时需配置合适的 PLL，保证 CK_OUT0 引脚输出的时钟为 25MHz。

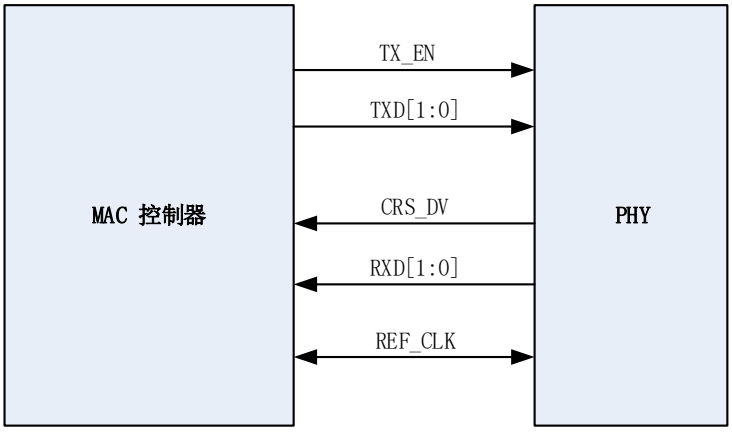
RMII：精简媒体独立接口

精简媒体独立接口（RMII）规范减少了以太网通信所需要的引脚数。根据 IEEE 802.3 标准，MII 接口需要 16 个引脚用于数据和控制信号，而 RMII 标准则将引脚数减少到了 7 个。

RMII 特性：

- 只有一个时钟信号，且该时钟信号需要提高到50MHz；
- MAC和外部的以太网PHY需要使用同样的时钟源；
- 使用2位宽度的数据收发。

图 26-5. 精简媒体独立接口（RMII）信号线



RMII 时钟源

通过将相同的时钟源接到 MAC 和以太网 PHY 的 REF_CLK 引脚保证两者时钟源的同步。可以通过外部的 50MHz 信号或者微控制器的 CK_OUT0 引脚提供这一时钟。当时钟来源 CK_OUT0 引脚时需配置合适的 PLL，保证 CK_OUT0 引脚输出的时钟为 50MHz。

MII/RMII 位传输顺序

不论选择的是 MII 还是 RMII 接口，发送接收的次序都是低位先出。

MII 和 RMII 之间的区别主要是数据位数和发送次数的不同。MII 上是先发送/接收低 4 位数据，再发送/接收高 4 位。在 RMII 上则是先发送/接收最低 2 位数据，再次低 2 位数据，次高 2 位数据，和最高 2 位数据。

例如：一个字节数据为 10011101b（从左到右顺序：高位到低位）

使用 MII 发送需 2 个时钟周期：1101 -> 1001（从左到右顺序：高位到低位，其中 1101 对应 MII_T/RXD[3] - MII_T/RXD[0]）

使用 RMII 发送需 4 个时钟周期：01 -> 11 -> 01 -> 10（从左到右顺序：高位到低位，其中 01 对应 RMII_T/RXD[1] - RMII_T/RXD[0]）

26.3.2. MAC 功能简介

MAC 模块可以在两种模式（半双工模式和全双工模式）下工作。半双工模式下，通过 CSMA/CD 算法来抢占对物理介质的访问，在同一时间只有一个传输方向的两个站点有效。全双工模式下，满足以下条件时，可同时进行收发而不发生冲突：1）物理介质支持同时进行收发操作。2）只有两个站点接入 LAN，且两个站点都配置为全双工模式。

MAC 模块能够实现以下功能：1）数据封装（发送和接收），包括检测/解码帧、帧边界定界、寻址（管理源地址和目的地址）、错误检测。2）半双工模式下的介质访问管理，包括介质分配和冲突解决。

MAC 的发送流程

所有的发送均由以太网模块中专用 DMA 控制器和 MAC 控制。在收到应用程序发送指令后，DMA 将发送帧从系统存储区读出并存入深度为 2K 的 TxFIFO 中，之后根据选择的模式（直通或者存储转发模式，具体定义请查看下段）将数据取出到 MAC 控制器，通过 MII/RMII 接口发送到以太网 PHY，并可以选择配置使 MAC 控制器自动将硬件计算的 CRC 值添加到数据帧的帧校验序列中。当 MAC 控制器收到来自 TxFIFO 的帧结束信号后，完成整个传输过程。传输完毕后，传输状态信息将会由 MAC 控制器生成并写回到 DMA 控制器中，应用程序可以通过 DMA 当前发送描述符查询发送状态。

TxFIFO 取出数据到 MAC 控制器的操作有两种模式：

- 在直通模式下，一旦FIFO中的数据字节数等于或超过设置的阈值，或者描述符中的帧结束标志被写入，FIFO中的数据就会被送入到MAC控制器中。用户可通过ENET_DMA_CTL中的TTHC[2:0]配置阈值。
- 在存储转发模式下，只有当一个完整的帧写入FIFO之后，FIFO中的数据才会被送入到MAC控制器中。但还有一种情况，帧没有被完整写入FIFO，FIFO也会取出数据。这种情况为TxFIFO的大小小于要发送的以太网帧长度，那么在TxFIFO即将全满时，数据会被送入到MAC控制器。

特殊情况处理

在传输过程中，如果空闲的 DMA 发送描述符不足，或者误操作了 ENET_DMA_CTL 的 FTF 位清空了 FIFO（此位置 1 时将清空 TxFIFO 中的数据并将 FIFO 的指针复位，清空操作完成后由硬件将此位清零），则将导致不能及时连续的发送数据，此时 MAC 控制器会标识数据下溢

状态。对于只收到帧起始信号却没有收到帧结束信号的情况，MAC 会忽略第二帧数据的帧起始，而将第二帧作为前一帧的延续。

若被发送的一帧占用两个 DMA 发送描述符，则第一个描述符的首段位（FSG）和末段位（LSG）应为 10b，第二个描述符的应为 01b。若第一个描述符与第二个描述符的 FSG 位都置位了，且第一个描述符的 LSG 位复位了，则将忽略第二个描述符的 FSG 位，并认为这两个描述符为只发送一个帧。

若发送 MAC 帧的数据域长度小于 46 或者带标签的 MAC 帧的数据域长度小于 42，可以选择配置 MAC 控制器自动填充内容为 0 的数据，使帧数据域的长度符合 IEEE 802.3 规范的相关定义。若执行了自动填充 0 功能，则 MAC 将忽略 DMA 描述符 DCRC 位的配置，自动计算并添加 CRC 值到帧的帧校验序列中。

MAC 的发送管理

Jabber 定时器

为了防止出现一个站点长时间占用 PHY 的情况，以太网内置的 Jabber 定时器会在以太网帧发送超过 2048 字节后终止发送。默认情况下，Jabber 定时器是使能的，因此当以太网帧发送超过 2048 字节，则 MAC 将只发送 2048 字节，并丢弃剩余的帧数据。

冲突处理机制：重发

在半双工模式下，MAC 发送数据帧时可能会发生冲突。当发生冲突事件的时候如果 FIFO 中只有不超过 96 个字节的帧数据被取出到了 MAC 中，那么帧重发功能将被激活。重发功能激活后，MAC 会中止当前的传输，然后重新从 FIFO 中读取数据并发送。当发生冲突事件的时候如果已有超过 96 个字节的数据从 FIFO 中取出到 MAC 中，那么 MAC 会中止当前的传输但不会激活重发功能，然后在描述符中置位 LCO 以通知应用程序。

清空 TxFIFO 操作

将 ENET_DMA_CTL 寄存器的 FTF 位置 1 将清空 TxFIFO，并将 FIFO 数据指针复位。无论 TxFIFO 是否正在取出数据到 MAC 中，清空操作都会立刻执行。因此这也将导致 MAC 控制器产生数据下溢事件，并终止发送当前帧，同时返回该帧的状态信息和发送状态信息字到应用程序。并标记数据下溢位和清空位（发送描述符 0 的 FRMF 和 UFE 位）。在应用程序（DMA）接收到所有被清空帧的状态信息字以后，清空操作完成。清空操作完成后，ENET_DMA_CTL 寄存器的 FTF 位将自动清’0’。当收到清空操作指令，所有从 FIFO 取出到 MAC 的数据都将被丢弃，直到收到 FSG 位为 1 的描述符。

帧间隔管理

MAC 管理两个帧之间的时间间隔。两个帧之间的时间间隔称为帧间隙时间。在全双工模式下，在完成帧发送后，或者 MAC 进入空闲状态时，帧间隙计数器开始计数。如果在帧间隙时间未到达 ENET_MAC_CFG 寄存器中 IGBS 位所配置的值时，来了新的发送帧，则这个发送帧将被延迟发送直到达到帧间隙时间值。若这个新的发送帧在帧间隙时间之后到达，则会立即发送该帧。在半双工模式下，MAC 遵循截断二进制指数退让算法，简要来说，就是在前一个发送帧发送完成之后，或者 MAC 进入空闲状态时，帧间隙计数器开始计数。在帧间隙时间内，可能会有 3 种情况会发生：

1. 如果在帧间隙时间的前2/3时间检测到载波信号，帧间隙计数器将复位并重新计数；
2. 如果在帧间隙时间的后1/3时间里检测到载波信号，帧间隙计数器不会复位，将继续计数，当帧间隙时间到达后，MAC发送新的帧；
3. 如果在整个帧间隙时间内都没有检测到载波信号，则在到达帧间隙时间后停止帧时隙计数器，并在之前有帧被延迟的情况下立即发送新的帧。

地址过滤模块

MAC 过滤分为错误过滤（诸如过短帧、CRC 错误以及坏帧的过滤）和地址过滤。此部分主要讨论地址过滤。

地址过滤利用静态物理地址（MAC 地址）过滤和多播 HASH 列表过滤实现。若 ENET_MAC_FRMF 过滤器寄存器的 FAR 位为’0’（默认值），则只有通过地址过滤的帧才会被接收。该功能会根据应用程序设定的参数(帧过滤器寄存器)对单播帧或多播帧的目的与/或源地址进行过滤（通过目标地址的 I/G 位可判断是单播帧还是多播帧）并报告相应的地址过滤结果,所有不能通过过滤器的帧将被丢弃。

注意：若 ENET_MAC_FRMF 过滤器寄存器的 FAR 位为’1’，则所有帧都会被接收。在这种情况下，帧过滤结果仍会更新到接收描述符中，但帧过滤结果不会影响到帧是否会被过滤。

单播目标地址过滤器

通过对 ENET_MAC_FRMF 寄存器 HUF 位的设置，可以选择使用静态物理地址（HUF 位为’0’）或者 HASH 列表（HUF 位为’1’）的方式实现单播过滤。

- 静态物理地址过滤

MAC 控制器支持多达 4 个 MAC 地址对单播地址进行完美过滤。在这种方式下，MAC 会把接收到帧的 6 个字节单播地址与设好的 MAC 地址寄存器逐位比较，检查是否相符。对于 MAC 地址 0 寄存器始终使能，对于 MAC 地址 1-MAC 地址 3 寄存器分别有对应的使能位。MAC 地址 1-MAC 地址 3 寄存器的每一个字节都可以通过相应 MAC 地址的高寄存器的屏蔽字节控制位（MB 位）来设置是否与接收帧的目标地址相应字节比较。

- HASH列表过滤

这种过滤使用一种 HASH 机制。MAC 利用 64 位的 HASH 列表对单播地址进行不完美过滤。这种过滤模式遵循以下两个过滤步骤：

1. MAC计算接收帧的目标地址的CRC值
2. 取CRC计算结果高6位作为索引检索HASH列表。如果CRC值对应的HASH列表上的相应位为’1’，则该帧能通过HASH过滤器，反之则该帧不能通过HASH过滤器。

这种类型过滤器的优点是可以仅用一个小表就覆盖任何可能的地址。缺点是过滤不完全，即有时应该丢弃的帧也会被接收。

单播源地址过滤器

使能 MAC 地址 1-MAC 地址 3 寄存器，并设置其对应 MAC 地址高寄存器的 SAF 位为’1’，MAC 可以将 MAC 地址 1-MAC 地址 3 寄存器中设置的物理（MAC）地址与接收帧的源地址进行比较并过滤。MAC 也支持对源地址的成组过滤。若设置帧过滤寄存器 ENET_MAC_FRMF

的 SAFLT 位为’1’，MAC 会丢弃没能通过源地址过滤的帧，同时过滤结果会通过 DMA 接收描述符 0 的 SAFF 位反映出来。当 SAFLT 位为’1’的同时，目标地址过滤器也在工作，此时 MAC 控制器以两个滤波器结果的逻辑“与”形式判定帧是否通过。这意味着，只要帧没能通过其中一个过滤器，就会被丢弃。MAC 只会把通过全部过滤器的帧转发给应用程序。

多播目标地址过滤器

将帧过滤寄存器 ENET_MAC_FRMF 的 MFD 位清零，以开启 MAC 多播地址过滤功能。此时根据帧过滤寄存器 ENET_MAC_FRMF 的 HMF 位的取值可以选择类似于单播目标地址过滤的两种方式进行地址过滤。

广播地址过滤器

默认情况下，MAC 无条件的接收任何广播帧。但当设置帧过滤寄存器 ENET_MAC_FRMF 的 BFRMD 位为’1’时，MAC 将丢弃接收到的所有广播帧。

HASH 或者完美地址过滤器

设置帧过滤器寄存器 ENET_MAC_FRMF 的 HPFLT 位为’1’，并设置相应的 HUF 位（对单播帧）或者 HMF 位（对多播帧）为’1’，则可以将过滤器配置成只要接收帧的目标地址匹配 HASH 过滤器或者物理地址过滤器之一，就令帧通过。

逆转过滤操作

无论是目标地址过滤还是源地址过滤，都能在过滤器输出端逆转过滤结果。即地址与过滤器匹配时，帧不通过；不匹配时帧通过。通过设置帧过滤寄存器 ENET_MAC_FRMF 的 DAIFLT 位和 SAIFLT 位为’1’可以实现这一功能。DAIFLT 位作用于单播和多播帧的目标地址的过滤结果，SAIFLT 位作用于单播和多播帧的源地址的过滤结果。

下面*表 26-7. 目标地址过滤器结果列表*和*表 26-8. 源地址过滤器结果列表*总结了目标地址和源地址过滤器在不同设置下的工作状态。

表 26-7. 目标地址过滤器结果列表

帧类型	PM	HPFLT	HUF	DAIFLT	HMF	MFD	BFRMD	目标地址过滤器操作
广播帧	1	-	-	-	-	-	-	通过
	0	-	-	-	-	-	0	通过
	0	-	-	-	-	-	1	不通过
单播帧	1	-	-	-	-	-	-	所有帧通过
	0	-	0	0	-	-	-	匹配完美/组过滤器时通过
	0	-	0	1	-	-	-	匹配完美/组过滤器时不通过
	0	0	1	0	-	-	-	匹配HASH过滤器时通过
	0	0	1	1	-	-	-	匹配HASH过滤器时不通过
	0	1	1	0	-	-	-	匹配HASH或者完美/组过滤器时通过
	0	1	1	1	-	-	-	匹配HASH或者完美/组过滤器时不通过
多播帧	1	-	-	-	-	-	-	所有帧通过
	-	-	-	-	-	1	-	所有帧通过

帧类型	PM	HPFLT	HUF	DAIFLT	HMF	MFD	BFRMD	目标地址过滤器操作
	0	-	-	0	0	0	-	匹配完美/组过滤器时通过，如果PCFRM = 0x，丢弃暂停控制帧
	0	0	-	0	1	0	-	匹配HASH过滤器时通过，如果PCFRM = 0x，丢弃暂停控制帧
	0	1	-	0	1	0	-	匹配HASH或者完美/组过滤器时通过，如果PCFRM = 0x，丢弃暂停控制帧
	0	-	-	1	0	0	-	匹配完美/组过滤器时不通过，如果PCFRM = 0x，丢弃暂停控制帧
	0	0	-	1	1	0	-	匹配HASH过滤器时不通过，如果PCFRM = 0x，丢弃暂停控制帧
	0	1	-	1	1	0	-	匹配HASH或者完美/组过滤器时不通过，如果PCFRM = 0x，丢弃暂停控制帧

表 26-8. 源地址过滤器结果列表

帧类型	PM	SAIFLT	SAFLT	源地址过滤器操作
单播帧	1	-	-	所有帧通过
	0	0	0	匹配完美/组过滤器时返回通过状态，不匹配时状态为不通过，但不丢弃不通过的帧
	0	1	0	匹配完美/组过滤器时返回不通过状态，但不丢弃帧
	0	0	1	匹配完美/组过滤器时通过，丢弃不通过的帧
	0	1	1	匹配完美/组过滤器时不通过，丢弃不通过的帧

混杂模式

若设置 ENET_MAC_FRMF 寄存器的 PM 位为’1’将使能混杂模式，此时地址过滤器无效，所有帧均可通过过滤器。同时接收状态信息的目标地址/源地址错误位总是为’0’。

暂停控制帧过滤

MAC 会检测接收到的控制帧内的 6 字节目标地址域，若 ENET_MAC_FCTL 寄存器的 UPFDT 位设为 0，则判断目标地址域的值是否符合 IEEE 802.3 规范控制帧的唯一值(0x0180 C200 0001)。若 ENET_MAC_FCTL 寄存器的 UPFDT 位设为 1，则在与 IEEE 802.3 规范定义的唯一值比较外，同时与控制器所设置的 MAC 地址逐位比较。如果目标地址域比较通过且接收流控制被使能（ENET_MAC_FCTL 的 RFCEN 位被置 1），则相应暂停控制帧功能将被触发。这个通过过滤的暂停帧是否会被转发给应用取决于 ENET_MAC_FRMF 寄存器的 PCFRM[1:0]位设置。

MAC 的接收流程

MAC 接收到的帧都会被送入 RxFIFO 中。MAC 接收到帧后会剥离其前导码和帧首界定码，并从帧首界定码后的第一个字节（目标地址）开始向 FIFO 发送帧数据。如果使能了 IEEE 1588 时间戳，MAC 会在检测到帧的帧首界定码的时候记录下系统的当前时间。如果这个帧通过地址过滤器的检查，MAC 会把这个时间戳通过接收描述符一并发给应用程序。

若 ENET_MAC_CFG 寄存器的 APCD 位置位，且接收到的帧长度/类型域的值小于 0x600 时，

MAC 将自动剥离填充域和帧校验序列。MAC 会在向 RxFIFO 发送完帧长度/类型域规定字节数后，丢弃包括帧校验序列在内的余下字节。如果长度/类型域的值大于或等于 0x600，则忽略 APCD 位，由 TFCD 位来确定是否自动剥离帧校验序列。

若看门狗定时器被使能（ENET_MAC_CFG 寄存器中的 WDD 位被复位），当帧长度超过 2048 字节时将被切断。即使看门狗定时器被禁能，MAC 仍然会切断长度大于 16384 字节的帧。

当 RxFIFO 工作于直通模式时，如果 FIFO 中的数据量大于门限值（可通过 ENET_DMA_CTL 寄存器的 RTHC 位设置），就开始从 FIFO 中取出数据，并通知 DMA 接收。当 FIFO 完成取出整个帧后，MAC 控制器将接收状态信息字发送给 DMA 控制器以回写到接收描述符中。在这种模式下，假如一个帧开始由 FIFO 取出由 DMA 发送到应用程序，则即使检测到错误，帧也会一直接收直到整个帧接收完毕。由于错误信息也要等到此时才会发送给 DMA 控制器，此时帧的前部分已经被 DMA 接收，所以在这种模式下将 MAC 设置成将所有错误帧丢弃将无效。

当 RxFIFO 工作于存储转发模式（通过 ENET_DMA_CTL 寄存器的 RSFD 位设置）时，DMA 只在 RxFIFO 完整地收到一帧后，才将其读出。此模式下，如果 MAC 设置成将所有错误帧丢弃，那么 DMA 只会读出合法的帧，并转发给应用程序。一旦 MAC 在接口上检测到帧首界定码就会启动接收过程。MAC 控制器在处理帧之前会剥离前导码和帧首界定码。会通过过滤器检查帧的报头，并用帧校验序列核对帧的 CRC 值。如果帧没能通过地址滤波器，MAC 控制器就会丢弃该帧。

MAC 的接收管理

多个帧的接收处理

与 TxFIFO 不同，由于帧的状态信息紧随在帧数据之后，MAC 可以判断接受帧的状态，因此第二个接收帧的传送是紧接着第一个接收帧的数据与状态信息的，只要 RxFIFO 未满，就可以存放任意数量的帧。

错误处理

- 在从MAC接收到EOF之前，RxFIFO已满。则MAC控制器会将整个帧丢弃并返回一个溢出状态。同时将溢出计数器加1；
- 若RxFIFO设置成存储转发模式，MAC可以过滤并丢弃所有的错误帧。但根据ENET_DMA_CTL寄存器的FERF和FUF位的设置，RxFIFO仍可以接收错误帧和长度低于最小帧长的帧；
- 若RxFIFO设置成直通模式，并不能将所有的错误帧都丢弃，仅当DMA从RxFIFO读出帧的SOF时，RxFIFO也已获得了该帧的错误状态时可以丢弃错误帧。

流控模块

MAC 控制器主要通过背压（半双工模式）和暂停控制帧（全双工模式）来管理帧的发送流控。

- 半双工模式流控：背压

当 MAC 采用半双工模式进行通讯时，如果设置了发送流控使能位(ENET_MAC_FCTL 寄存器的 TFCEN 位)，有两种情况可以触发背压流控。背压流控是通过发送一个 32 位的堵塞信号 0x5555 5555，通知所有其他站点发生了冲突。两种触发情况中，第一种是通过置位 ENET_MAC_FCTL 寄存器的 FLCB/BKPA 位来使能发送流控。第二种情况在接收帧时发生，

MAC 在接收帧的过程中，RxFIFO 中字节数不断增大，当接收数目超过流控激活阈值（ENET_MAC_FCTH 寄存器中的 RFA 位），MAC 将置位背压挂起标志。若背压挂起标志置位了，且又有新的帧到来，MAC 将发送堵塞信号以延迟一段背压时间再接收帧。在背压时间结束后，PHY 会重新发送这个新的帧。若在背压期间，RxFIFO 中字节数大于等于流控失活阈值（ENET_MAC_FCTH 寄存器中的 RFD 位），则 MAC 会再次发送背压信号；反之，则 MAC 将复位背压挂起标志，并可以接收新的帧，不再发送堵塞信号。

- 全双工模式流控：暂停帧

对于全双工模式，MAC 控制器使用“暂停帧”进行流控制。这种方式可以使接收端能够命令发送端暂停一段时间再发送，如当接收缓冲区快要溢出的情况。如果设置了发送流控使能位(ENET_MAC_FCTL 寄存器的 TFCEN 位)，在全双工模式下，MAC 会在以下两种情况下产生并发送暂停帧。两种情况分别为：

1) 应用程序把 ENET_MAC_FCTL 寄存器的 FLCB/BKPA 位置位，将立即发送一个暂停帧。这个暂停帧指定的暂停时间为ENET_MAC_FCTL 寄存器中 PTM 位配置好的暂停时间值。如果应用程序前面要求了一段时间的暂停，但在这段时间内，应用程序准备好了，可以不需要剩余的暂停时间了，这时应用程序需要发一个零时间片暂停帧来通知发送方可以继续发送了。零时间片暂停帧是通过设置 ENET_MAC_FCTL 寄存器中的 PTM 位为 0，并将 FLCB/BKPA 位置位来发送的；
2) 在 RxFIFO 满足一定的条件下，MAC 会自动发送暂停帧。在接收过程中，RxFIFO 不停地有数据进来，同时 RxFIFO 也取出数据给 RxDMA，如果 RxFIFO 取出数据的频率小于其接收数据的频率，RxFIFO 中的数据就会越来越多。一旦 RxFIFO 中的数据量超过了流控的激活阈值（ENET_MAC_FCTH 寄存器中的 RFA 位），MAC 将发送一个暂停时间为 PTM 位定义的值的暂停帧。发送暂停帧之后，MAC 将启动一个计数器，计数器的时间由 ENET_MAC_FCTL 寄存器的 PLTS 位定义，当到了计数器规定的时间，MAC 将重新检查 RxFIFO。此时若 RxFIFO 中的数据量仍然大于流控激活阈值，MAC 将再次发送一个暂停帧。若 RxFIFO 中的数据量小于流控失活阈值，并且 ENET_MAC_FCTL 寄存器中的 DZQP 位被复位，则 MAC 将发送一个零时间片暂停帧。这个零时间片暂停帧用于指示远程站点结束暂停，本地缓存区已经准备好接收新的数据帧。

MAC 通过如下方式管理帧的接收流控：

在全双工模式下，MAC 能够检测暂停帧，并按照暂停帧中的暂停时间域参数，在暂停一定时间后再发送数据。可以通过设置 ENET_MAC_FCTL 寄存器的 RFCEN 位，使能或者取消暂停帧检测功能。如果没有使能该功能，则 MAC 会忽略接收到的暂停帧。若使能了该功能，MAC 将能够对接收到的暂停帧进行解码。类型域、操作数域和暂停时间域都将能够被 MAC 识别。在暂停期间，如果收到一个新的暂停帧，则新的暂停时间将立即被加载到暂停时间计数器中。如果接到收的暂停时间域值为 0，则 MAC 会停止暂停时间计数器，恢复数据的发送。通过配置 ENET_MAC_FRMF 寄存器的 PCFRM 位值，来处理这些接收到的控制帧。

校验和引擎

以太网控制器具有发送校验和的功能，支持计算校验和，并在发送时插入计算结果，以及在接收时侦测校验和错误。

如下描述了发送帧的校验和的操作功能。

注意：只有将 ENET_DMA_CTL 寄存器的 TSFD 位置为’1’（TxFIFO 设置成存储转发模式），同时必须保证 TxFIFO 的深度足够容纳将要发送的完整帧时，才能使能此功能。若 FIFO 深度小于帧长度，则仅仅计算和插入 IPv4 报头的校验和域。

欲了解 IPv4、TCP、UDP、ICMP、IPv6 和 ICMPv6 报头的规范，请分别查阅 IETF 规范 RFC 791、RFC 793、RFC 768、RFC 792、RFC 2460 和 RFC 4443。

- IP头校验和

若以太网帧的类型域值为 0x0800 同时 IP 数据包的版本域值为 0x4，则校验和模块标记其为 IPv4 数据包并会用计算结果取代帧的校验和域的内容。IPv6 的报头不包含校验和域，因此校验和模块不会改变 IPv6 报头的值。IP 头校验和计算完毕之后，其结果会写到发送描述符 0 的 IPHE 位。当发生下述情况时，IPHE 错误状态位会被硬件置’1’：

1) 对于IPv4数据帧：
   a) 接收到的以太网类型域值为0x0800，但IP报头版本域的值不等于0x4；
   b) IPv4报头长度域的值大于帧的总长度；
   c) IPv4报头长度域值小于IP报头总长0x5（20字节）。

2) 对于IPv6数据帧：
   a) 接收到的以太网类型域值为0x86dd，但IP报头版本域的值不等于0x6；
   b) 帧在完全接收IPv6报头或者扩展报头之前结束。IPv6标准报头长度为40字节，扩展报头包含相应的报头长度字段。

- TCP/UDP/ICMP校验和

校验和模块通过分析 IPv4 或 IPv6 报头(包括扩展报头)来判断帧的类型（TCP、UDP 或 ICMP）。

当帧发生以下情况时，将绕过校验和功能，校验和模块不对这些帧进行处理：

1) 不完整的IPv4或IPv6帧；
2) 包含安全功能的IP帧(如验证报头或者封装有安全数据)；
3) 非TCP/UDP/ICMPv4/ICMPv6数据的IP帧；
4) 带路由报头的IPv6帧。

校验和模块会对 TCP、UDP 或者 ICMP 的数据进行计算，并插入报头的相应域。它有以下 2 种工作模式：

1) 校验和计算不包括TCP、UDP或者ICMPv6的伪首部。并假定输入帧的校验和字段已有值。校验和字段包含在校验和计算中，在计算完成后插入并替换原校验和域的值；
2) 校验和计算包括TCP、UDP或者ICMPv6的伪首部。将传输帧的校验和字段清零。进行校验和的计算，计算完成后插入传输帧的原校验和域。

校验和计算完毕之后，其结果会写到发送描述符 0 的 IPPE 位。当发生下述情况时，IPPE 错误状态位会被硬件置’1’：

1) 在存储转发模式下，帧未被完整写入FIFO之前就被转发给MAC控制器；
2) 帧已发送完毕，但MAC从FIFO中取出的数据包字节数小于IP报头中数据长度域标明的字节数。

如果数据包长度大于标明的长度，不会报告错误，之后的数据会被当成填充字节而丢弃。如果

检测到第一类错误情况，校验和的值不会插入 TCP、UDP 或者 ICMP 报头。如果检测到第二类错误情况，仍然会把校验和计算结果插入报头的相应域。

注意：无论采用哪种模式，对于 IPv4 上的 ICMP 数据包，由于这类数据包没有定义伪报头，为正确计算其校验和，校验和域内容必须为 0x0000。

接收帧校验和的操作功能描述如下所述。

置位 ENET_MAC_CFG 寄存器的 IPFCO 位，可以使能接收校验和模块。接收校验和模块可以计算 IPv4 报头的校验和，并检查它是否与 IPv4 报头的校验和域的内容相匹配以外。MAC 可根据检查接收到的以太网帧类型域是 0x0800 还是 0x86dd，来判别是 IPv4 帧还是 IPv6 帧，这个方法也用于带 VLAN 标签的帧识别。DMA 接收描述符的报头校验和错误位（接收描述符 0 中的 IPHERR 位）反映了对报头的校验和结果，该位在接收到的 IP 报头出现下述错误时被置 1：

- 计算的IPv4报头的校验和值与其校验和域的内容不匹配；
- 以太网类型域值指示的数据类型与IP报头版本域不匹配；
- 接收到的帧长少于IPv4报头长度域指示的长度，或者IPv4/IPv6报头少于20字节。

接收校验和模块还能识别 IP 数据包的数据类型是 TCP、UDP 还是 ICMP，并按照 TCP、UDP 或 ICMP 的规范计算它们的校验和。计算过程包括 TCP/UDP/ICMPv6 伪报头的数据。DMA 接收描述符（接收描述符 0 的 PCERR 位）的数据校验和错误位反映了对数据的校验和结果，该位在接收到的 IP 数据包数据出现下述错误时被置 1：

- 计算的TCP、UDP或ICMP校验和与其帧的TCP、UDP或ICMP校验和域值不匹配；
- 收到的TCP、UDP或者ICMP数据长度与IP报头给出的长度不符。

接收校验和模块不计算下列情况：不完整的 IP 数据包、带安全功能的 IP 数据包、IPv6 路由报头以及数据类型不是 TCP、UDP 或者 ICMP 的数据包。

MAC 回环模式

通常地，回环模式用于应用程序对系统硬件和软件的测试与调试。通过将 ENET_MAC_CFG 寄存器的 LBM 位置’1’，可以使能 MAC 回环模式。在该模式下，MAC 发射端把帧发送到自身的接收端上。该模式默认为关闭。

26.3.3. DMA 控制器描述

为了减少 CPU 的干预，设计了以太网专用 DMA 控制器，用于实现 FIFO 和系统存储之间的帧数据传输。CPU 和 DMA 之间的的通讯通过 2 种数据结构实现。分别是：1）描述符列表（链结构或环结构）和数据缓存；2）控制和状态寄存器。应用程序需要开辟存储描述符列表及数据缓存用到的物理内存。在存储器里，描述符是以指向缓存的指针的形式存放。有 2 个描述符队列，一个用作发送，另一个用作接收。两个队列的基地址分别存放在 ENET_DMA_TDTADDR 寄存器和 ENET_DMA_RDTADDR 寄存器中。当 DFM 位为 0 时，发送描述符由四个描述符字发送描述符 0-4）组成，当 DFM 位为 1 时，发送描述符由八个描述符字（发送描述符 0-8）组成。同样的，当 DFM 位为 0 时，接收描述符由四个描述符字（接收描述符 0-4）组成，当 DFM 位为 1 时，接收描述符由八个描述符字（接收描述符 0-8）组成。每个描述符可以指向最多 2 个缓存用来存储帧的数据。根据描述符列表类型是环结构还是链结构，来决定第二个缓存是被

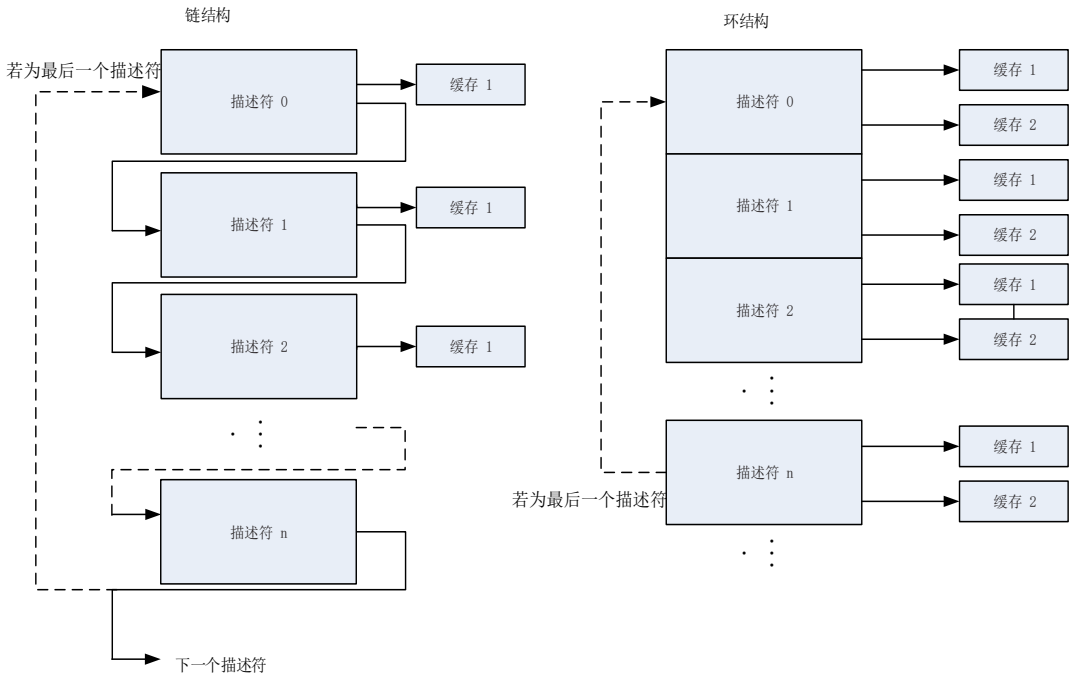
配置为第二个数据存储地址，还是下一个描述符地址。数据缓存存放在 MCU 的物理内存里，可以存放一个帧的全部或者部分，但是不允许存放不属于同一个帧的数据。描述符队列可以是显性（链结构）或者隐性（环结构）的方式前向连接的。通过设置接收描述符 1 的 RCHM 位和发送描述符的 TCHM 位为’1’，可以实现描述符的显性连接，此时接收描述符 2 及发送描述符 2 中将存放缓存地址，接收描述符 3 及发送描述符 3 中将存放下一个描述符的地址，这种链接的描述符也可以称为描述符的链结构。通过设置接收描述符 1 的 RCHM 位和发送描述符的 TCHM 位为’0’，可以实现描述符的隐性连接，此时接收描述符 2/发送描述符 2，接收描述符 3/发送描述符 3 中都将存放缓存地址，这种链接的描述符也可以称为描述符的环结构。在使用当前的描述符所指向的缓存地址时，描述符指针就指向下一个描述符。当使用链结构时，描述符指针指向的是第二个缓存。当使用环结构，根据下式计算描述符指针下一个所指向的地址：

DFM=0：下个描述符地址 = 当前描述符地址 + 16 + DPSL * 4

DFM=1：下个描述符地址 = 当前描述符地址 + 32 + DPSL * 4

若当前描述符是描述符列表的最后一个描述符，环结构下必须设置发送描述符 0 的位 21 或接收描述符 1 的位 15 以标识当前描述符为列表的最后一个。此时下一个描述符又指向描述符列表的第一个。链结构下还可以通过设置发送描述符 3 或接收描述符 3 的值指向描述符列表中第一个的地址。DMA 一旦检测到帧结束就会跳到下一个帧的缓存。

图 26-6. 描述符的环结构和链结构



数据缓存地址对齐

以太网 DMA 控制器支持所有对齐类型：字节对齐，半字对齐，字对齐。这意味着应用程序可将发送和接收数据缓存地址配置到任意地址。但是，在 DMA 发起传输的时候，总是以字对齐的方式访问地址。对于读和写缓存的访问也不一样。示例如下：

- 读缓存示例：如果发送缓存的地址为0x2000 0AB2，并需要传输15字节。在开始读操作后，

DMA实际会从地址0x2000 0AB0，0x2000 0AB4，0x2000 0AB8，0x2000 0ABC和0x2000 0AC0先读5个字，但是在往FIFO发送数据的时候，会丢弃头2个字节(0x2000 0AB0和0x2000 0AB1)和最后3个字节（0x2000 0AC1，0x2000 0AC2和0x2000 0AC3）。

- 写缓存示例：如果接收缓存的地址为0x2000 0CD2，并需要传输16字节。在开始写操作后，DMA实际会从地址0x2000 0CD0到0x2000 0CE0先写5个32位数据。但是头2个字节（0x2000 0CD0和0x2000 0CD1）和末尾的2个字节（0x2000 0CE2和0x2000 0CE3）会用虚拟字节替代。

注意：DMA 控制器不会写任何数据到定义的缓存区之外的地址。

缓冲区有效长度

发送帧的过程中，TxDMA 会传输与发送描述符 1 中标明的缓存有效长度的字节给 MAC 控制器。如前所述，一个发送帧可以用多个描述符来描述一个帧，即一个帧的数据可以处于多个不同的缓存中。如果 DMA 控制器读取的发送描述符 0 的 FSG 位为’1，那么 DMA 就明确了当前缓存存储的是一个新的帧，并标记发送的第一个字节是帧首。如果 DMA 控制器读取的发送描述符 0 的 LSG 位为’1’，则 DMA 就明确了当前缓存存储的是当前帧的最后一部分数据。通常来说一个帧只存在一个缓存里（因为缓存的大小对于一个正常的帧来说足够大了），因此 FSG 和 LSG 位会在一个相同的描述符中同时置位。

接收帧的过程中，接收帧的缓存长度域值必须是字对齐的。对于字对齐或非字对齐的缓存地址，接收操作与发送操作不大相同。如果接收缓存地址是字对齐的，则与发送流程是类似的，缓存的有效长度为由接收描述符 1 中配置的值。如果接收缓存地址是非字对齐的，则缓存的有效长度将小于接收描述符 1 中配置的值。缓存有效长度值应为接收描述符 1 中配置的值减去缓存地址的低 2 位值。例如，假设缓存的总大小为 2048 字节，缓存地址为 0x2000 0001，地址的低 2 位值为 0b01，那么缓存有效长度为 2047 个字节，范围从 0x2000 0001（帧首）到 0x2000 07FF。

当收到了一个帧起始 SOF，则 DMA 控制器将 FSG 位置位，当收到一个帧结束 EOF 时，则 LSG 位被置位。如果接收缓存长度域值配置的足够大，能放下整个帧，则 FSG 和 LSG 位将在同个描述符中被置位。实际接收的帧长度可从接收描述符 0 的 FRML 位域获取。从而应用程序可计算未被使用的缓存空间。RxDMA 总是用新的描述符来接收下一帧。

TxDMA 和 RxDMA 的仲裁器

DMA 的仲裁器设计了两种仲裁方式用于提高 DMA 发送与接收控制器的效率：固定和轮询优先级。设置 ENET_DMA_BCTL 寄存器 DAB 位为’0’，选择轮询优先级，在 TxDMA 和 RxDMA 同时要求访问数据总线的时候，按照 ENET_DMA_BCTL 寄存器 RTPR 位设定的比例对其访问进行分配。设置 DAB 位为’1’选择固定优先级，此时 RxDMA 和 TxDMA 同时要求访问时，RxDMA 总是对总线拥有更高的访问优先级。

DMA 错误状态

若 DMA 在传输过程中出现了错误的总线响应，那么 DMA 控制器认为发生了一个致命错误，会立刻停止所有操作，并更新状态寄存器 ENET_DMA_STAT。在发生类似的致命错误（响应错误）之后，应用程序必须复位以太网外设并重新初始化 DMA，DMA 才能恢复操作。

TxDMA 与 RxDMA 控制器的初始化

在使用 DMA 控制器之前，必须按如下步骤对 DMA 进行初始化：

1. 对ENET_DMA_BCTL寄存器进行总线访问参数的相关设置；
2. 对ENET_DMA_INTEN寄存器进行设置，屏蔽不需要的中断源；
3. 将发送描述符列表和接收描述符列表的基地址分别写入ENET_DMA_TDTADDR寄存器与ENET_DMA_RDTADDR寄存器中；
4. 对相关的寄存器进行期望的过滤器配置；
5. 根据从PHY读出的自协商的结果，设置SPD位和DPM位的值，来选择通讯模式（半/全双工）及通讯速度（10Mbit/s或100Mbit/s）。将ENET_MAC_CFG寄存器的TEN和REN位置’1’，使能MAC的发送和接收操作；
6. 设置ENET_DMA_CTL寄存器的位STE和位SRE为’1’，使能DMA发送和接收器。

注意：如果 HCLK 频率过低，应用程序可以先使能 DMA 接收器，再将 ENET_MAC_CFG 寄存器的 REN 位置’1’，以避免 RxFIFO 在启动的时候溢出。

DMA 发送帧处理

如前所述，一个帧可以分散在不同缓存内，这意味着需要多个描述符。当 FSG 位置位，表示当前描述符指向的缓存为帧头，当 LSG 位置位，表示当前描述符指向的缓存为帧尾。对于当前帧其他描述符（LSG 位为’0’的描述符），TxDMA 控制器仅修改清零其 DAV 位。在这最后一个缓存的数据发送完毕以后，DMA 会将整个帧的发送状态信息，写入最后一个的发送描述符 0 并返回。将数据从系统存储传输到 FIFO，开始发送数据，但实际上真正的数据发送是由 TxDMA 模式决定的：直通模式和存储转发模式。直通模式在 FIFO 中的字节数大于所配置的阈值时，数据将取出到 MAC 发送。存储转发模式在整个帧数据都传入 FIFO 后或 FIFO 快要填满时再取出数据给 MAC 进行发送。

DMA 发送管理

发送缓存区中第二帧操作

如果 ENET_DMA_CTL 寄存器中 OSF 位为’0’，则发送顺序为：首先读取发送描述符，然后从系统存储读取数据写到 FIFO，再将帧数据通过 MAC 放到接口上，最后等待数据发送完毕后将发送状态写回描述符。

上述是 TxDMA 的标准发送流程，但当 HCLK 远远大于 TX_CLK 时，在发送两个帧时发送效率将显著降低。

为避免上述提及的情况，应用程序可将 OSF 位置位。在此情况下，第二帧的数据可以不等待第一帧的描述符状态信息被写回，就先读取内存里的第二帧数据，并把它们送进 FIFO。OSF 功能仅在两相邻帧之间起作用。

TxDMA 操作模式（A）（默认）：非 OSF

在默认模式下，TxDMA 控制器的工作流程如下：

1. 初始化帧数据到发送缓存，并对发送描述符0-3进行设置，置发送描述符0的DAV位为’1’；
2. 将ENET_DMA_CTL寄存器的STE位置为’1’，使能TxDMA控制器；

3. TxDMA控制器开始轮询发送描述符列表来获取待发送的帧。如果TxDMA检测到发送描述符0的位DAV为0，或者发生了错误，则控制器就会终止传输进入挂起状态，并设置ENET_DMA_STAT寄存器的发送缓存不可用位（位2）和正常中断汇总位（位16）为’1’。如果处于挂起状态，则发送控制器操作跳至步骤8；
4. 如果取到的描述符标志位显示该描述符由DMA占有（DAV位被置’1’），那么DMA从描述符中解析出所配置的发送帧以及发送数据缓存的地址；
5. DMA从内存中取出数据并将数据存入TxFIFO；
6. TxDMA控制器会一直轮询描述符列表直到帧结尾被传送出去（LSG位置位）。如果当前描述符的LSG位为’0’，则在所有缓存数据送入TxFIFO之后，将DAV位清零以关闭这个描述符。然后TxDMA控制器等待写回描述符状态，以及IEEE 1588时间戳值（如果使能了时间戳功能）；
7. 在整个帧发送完成以后，仅当发送描述符0位INTC为’1’时，发送状态位（ENET_DMA_STAT寄存器中的TS位）会被置位。此时若使能了DMA中断，将进入相应中断。然后DMA控制器返回步骤3，继续处理下一帧；
8. 在挂起状态下，如果向发送查询使能寄存器ENET_DMA_TPEN写入任意值，并清除发送溢出标志位，TxDMA将重新回到运行状态，尝试重新获取描述符。发送控制器操作回到步骤3。

TxDMA 操作模式(B)：OSF

在操作第二帧(OSF)模式下，TxDMA 可以不必等到前一帧的状态信息写回，就发送下一帧。如果系统时钟频率远远大于 MAC 频率（10Mbit/s 或 100Mbit/s），这种情况 OSF 模式可以提高发送效率。设置 ENET_DMA_CTL 寄存器的位 OSF 为’1’，进入此模式。DMA 在发送完前一帧数据后，不必等到前一阵的状态写回，而是立即查询第二帧的发送描述符，如果第二帧发送描述符的 DAV 位与 FSG 位都置 1，那么 TxDMA 立即读取第二帧的帧数据并将其存入 MAC FIFO。

在 OSF 模式下，TxDMA 的操作流程如下：

1. 按照TxDMA默认模式的步骤1-6操作；
2. DMA不等到关闭前一帧的最后一个描述符（LSG位为’1’），就直接取下一个描述符；
3. 如果取到的描述符标明被DMA占有（DAV位为’1’），那么就从解析的发送缓存地址中读取下一帧的数据。如果DAV位为’0’即DMA不占有这个描述符，则TxDMA进入挂起状态并跳到步骤7；
4. TxDMA控制器会一直轮询描述符列表直到帧结尾被传送出去。如果一个帧由多个描述符描述，则中间描述符会在获取之后就被关闭；
5. TxDMA等待前一帧的发送状态信息和时间戳（如果使能了时间戳功能），在接收到状态信息后，DMA会把DAV位为’0’的状态信息写入发送描述符0，将该描述符的占有权交还给CPU进行操作；
6. 在整个帧发送完成以后，仅当发送描述符0位INTC为’1’时，发送状态位（ENET_DMA_STAT寄存器中的TS位）会被置位。此时若使能了DMA中断，将进入相应中断。如果前一个帧返回的状态信息正常则跳到步骤3。若显示有数据下溢错误，TxDMA进入挂起状态，并跳到步骤7；
7. 在挂起状态下，如果TxDMA收到一个发送帧的待处理的状态信息和时间戳（若使能了时间戳），则TxDMA将这些信息写入发送描述符，并将相应描述符的的DAV位清零。随后设置相关的中断标志位并回到暂停状态；

8. 在挂起状态下，如果向发送查询使能寄存器ENET_DMA_TPEN写入任意值并将溢出中断标志位清零，TxDMA将回到运行状态，尝试重新获取描述符。发送控制器操作根据是否有待处理的状态信息跳到步骤1或者步骤2。

发送帧格式

根据前述的 IEEE 802.3 规范，一个正常的发送帧应该由以下及部分构成：前导码，帧首界定码 SFD，目标地址 DA，源地址 SA，QTAG 前缀（可选），长度/类型域 LT，数据，PAD 填充域（可选），和帧校验序列 FCS。

前导码和帧首界定码都是由 MAC 自动生成的，因此应用程序只需要存储目标地址，源地址，QTAG（若需要），长度/类型，数据，填充域（若需要），帧校验序列（若需要）。如果帧需要填充位，即缓存中没有存储填充位和帧校验序列部分，则应用程序可配置自动生成帧校验序列和填充位功能。如果帧仅需帧校验序列，即缓存中没有存储帧校验序列部分，则应用程序可配置自动生成帧校验序列。DPAD 位和 DCRC 位用于配置填充位和帧校验序列的自动生成。

发送查询挂起后的处理

当传输开始后 DMA 会不断对发送描述符进行查询，当发生如下情况时，会导致 DMA 进入挂起状态，并暂停发送。此时当前描述符固定为暂停前的最后一个描述符。

- DMA检测到发送描述符0的DAV位为0，此时CPU占有描述符，则会进入挂起状态，并暂停查询。同时设置ENET_DMA_STAT寄存器的正常中断总结位NI和发送缓存不可用位TBU为’1’；
- 当接口在发送帧的过程中MAC FIFO为空，意味着检测到了数据下溢错误。在此情况下，设置ENET_DMA_STAT寄存器的异常中断总结位AI和发送数据下溢位TU为’1’，同时把该信息写入发送描述符0。

带 IEEE 1588 时间戳的 TxDMA 描述符格式

如果设置 TTSEN 位为’1’，则使能了 IEEE1588 功能。TxDMA 控制器会在帧发送完成后，将时间戳写入描述符同时设置 TTMSS 位为’1’。写时间戳值的地址由 ENET_DMA_BCTL 寄存器的 DFM 位决定。如果 DFM 位为’0’即描述符格式为常规模式，则时间戳将覆盖写到发送描述符 2 和发送描述符 3 上。如果 DFM 位为’1’即描述符格式为增强型模式，则时间戳将写到发送描述符 6 和发送描述符 7 上，发送描述符 2 和发送描述符 3 中的值保持不变。

常规 TxDMA 描述符

常规 TxDMA 描述符结构体包含 4 个 32 位字，发送描述符 0~发送描述符 3。发送描述符 0~发送描述符 3 的的位定义如下：

注意：若一个帧由多个描述符表示，则对于描述符的控制位（除了 INTC 位）只有第一个描述符的才有效。状态信息和时间戳（若使能了时间戳功能）只写回到最后一个描述符。

图 26-7. 常规发送描述符

■ 发送描述符0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DAV	INTC	LSG	FSG	DCRC	DPAD	TTSEN	保留	CM[1:0]		TERM	TCHM	保留		TTMSS	IPHE
rw	rw	rw	rw	rw	rw	rw		rw		rw	rw			rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ES	JT	FRMF	IPPE	LCA	NCA	LCO	ECO	VFRM	COCNT[3:0]				EXD	UFE	DB
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				rw	rw	rw

位/位域	名称	描述
31	DAV	DAV位 DMA会在将帧完整传输或者描述符指向的缓存里的数据全部被读出以后把该位清'0'。当一个帧位于多个缓存中时，第一个缓存描述符的DAV位，必须在后面缓存描述符的DAV位全部置'1'以后，才能置'1'。 0：表示CPU占有描述符 1：表示DMA占有描述符
30	INTC	完成时中断位 LSG位置位后，此位才有效。 0：帧发送完成时，ENET_DMA_STAT寄存器的TS位不被置位 1：帧发送完成时，ENET_DMA_STAT寄存器的TS位被置位
29	LSG	最后分块位 此位指示缓存是否包含帧的最后一个分块。 0：该描述符缓存中没有存放帧的最后一个分块 1：该描述符缓存中存放有帧的最后一个分块
28	FSG	第一分块位 此位指示缓存是否包含帧的第一个分块。 0：该描述符缓存中没有存放帧的第一个分块 1：该描述符缓存中存放有帧的第一个分块
27	DCRC	不计算CRC位 只有在FSG位置位时，此位才有效。 0：MAC 在传输帧末尾自动插入 CRC 域 1：MAC不在传输帧末尾自动插入CRC域

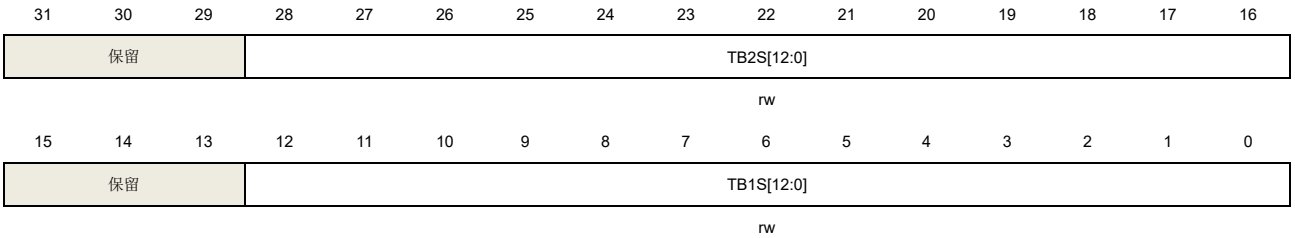
26	DPAD	不填充位 只有在 FSG 位置位时，此位才有效。 0：DMA 对传输帧自动添加填充字节，并且插入 CRC 数值。发生填充时，CRC 会被插入，忽略 DCRC 位的值。 1：MAC 不对传输帧自动填充字节 注意：此处的传输帧小于64字节。
25	TTSEN	使能发送时间戳位 只有在 FSG 位置位时，此位才有效。 0：发送时间戳功能失能 1：当ENET_PTP_TSCTL寄存器的TMSEN位为'1'时，传输帧的IEEE1588硬件时间戳功能使能。
24	保留	必须保持复位值。
23:22	CM[1:0]	校验和插入模式位 0x0：不插入校验和 0x1：只使能硬件IP报头的校验和计算和插入 0x2：使能硬件IP报头和数据域的校验和计算和插入，但是不计算伪报头的校验和 0x3：使能硬件IP报头和数据域的校验和计算和插入，也计算伪报头的校验和
21	TERM	环形发送结束模式位 该位仅在环模式下使用，且比TCHM位具有更高优先级。 0：当前描述符还不是描述符队列的最后一个 1：当前描述符到达描述符队列的最后一个，DMA返回列表的基地址
20	TCHM	第二地址链表模式位 该位在链模式下使用。该位为'1'时，忽略TB2S[12:0]的值。 0：描述符里的第二个地址是第二缓存的地址 1：描述符里的第二个地址是下一个描述符的地址，而不是第二个缓存的地址
19:18	保留	必须保持复位值。
17	TTMSS	发送时间戳状态位 只有在LSG位置位时，此位才有效。 0：还未记录帧的时间戳信息 1：记录下了描述符对应的帧时间戳，记录的时间戳放在发送描述符2（或发送描述符6，假如DFM=1）和发送描述符3（或发送描述符7，假如DFM=1）处。
16	IPHE	IP报头错误位 发生下列任意一种情况，则产生IP报头错误： IPv4帧： 1）报头长度域值小于0x5。 2）报头长度域值与报头的长度不符。 3）报头版本域值与帧长度/类型域值不匹配。 IPv6帧： 1）主报头长度不足40字节。 2）报头版本域值与帧长度/类型域值不匹配。

0：未发现IP数据包报头的错误
1：MAC发送端发现了IP数据包报头的错误

15	ES	错误汇总 该位为下列位的逻辑“或”： IPHE：IP 报头错误 JT：Jabber 超时 FRMF：帧清空 IPPE：IP 数据错误 LCA：载波丢失 NCA：无载波 LCO：延迟冲突 ECO：过度冲突 EXD：过度顺延 UFE：数据下溢错误
14	JT	Jabber超时位 该位仅当JBD位复位时才会被置’1’。 0：未发生Jabber超时 1：MAC发送端发生了Jabber超时
13	FRMF	帧清空位 置1时，清空TxFIFO中的数据。
12	IPPE	IP数据错误位 发送端会核对IPv4或者IPv6报头的数据长度域值与实际收到的TCP、UDP和ICMP数据数目，不符合就置’1’报错。 0：未发生IP数据错误 1：MAC发送端发现了IP数据包的TCP、UDP或者ICMP的IP数据错误。
11	LCA	载波丢失位 在发送时，如果CRS信号在一个或一个以上发送时钟周期中为无效状态，并且没有发生冲突，则载波丢失将概率性发生。 该位只有在半双工模式下有效。 0：未发生载波丢失 1：帧发送的时候发生了载波丢失
10	NCA	无载波位 0：PHY的载波侦听信号有效 1：帧发送的时候PHY的载波侦听信号无效
9	LCO	延迟冲突位 如果冲突在64字节（包括前导符）发送之后发生，则这种情况称作延迟冲突。 0：未发生延迟冲突 1：发生了延迟冲突 注意：如果溢出错误位UFE置’1’，该位无效。

8	ECO	过度冲突位 如果MAC设置寄存器的RTD（不进行重试）位为'1'，那么在发生一次冲突后，该位就置'1'。 如果MAC设置寄存器的RTD（不进行重试）位为'0'，那么在连续发生16次冲突后，该位置'1'。 若该位置位，则中止当前帧的发送。 0：未发生过度冲突 1：发生了过度冲突
7	VFRM	VLAN帧位 0：发送帧为普通帧 1：发送的帧是VLAN帧
6:3	COCNT[3:0]	冲突计数位 只有在 ECO 位为 0 时，此位才有效。 该4位计数值记录了帧发送出去前出现的冲突次数。
2	EXD	过度顺延位 当MAC设置寄存器的顺延位DFC为'1'时有效。 0：未发生过度顺延 1：由于顺延超过3036字节的时间而结束发送
1	UFE	数据下溢错误位 数据下溢错误表示由于从系统存储传输数据到FIFO的速度过慢，导致DMA在发送帧的时候遇到了空的缓存。发送过程进入挂起状态，并将ENET_DMA_STAT寄存器的发送数据下溢位TU（位5）和发送状态位TS（位0）都置'1'。 0：未发生数据下溢错误 1：发生了数据下溢错误，MAC中止帧的发送。
0	DB	顺延位 该位指示了是否由于载波侦听信号CRS在MAC发送帧之前被占用，而导致发生帧的顺延。 该位只在半双工模式下有效。 0：未发生发送顺延 1：MAC发生了顺延，推迟发送。

■ 发送描述符1



位/位域	名称	描述
31:29	保留	必须保持复位值。

28:16	TB2S[12:0]	发送缓存 2 大小 第二个数据缓存的大小（以字节记）。
15:13	保留	必须保持复位值。
12:0	TB1S[12:0]	发送缓存 1 大小 第一个数据缓存的大小（以字节记）。

■ 发送描述符2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TB1AP/TTSL[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TB1AP/TTSL[15:0]															
rw															

位/位域	名称	描述
31:0	TB1AP/TTSL[31:0]	发送缓存1地址指针/发送帧时间戳低32位 在发送帧之前，应用程序必须对这些位进行配置发生缓存1地址（TB1AP），等到数据发送完后，DMA可以用它们存放帧的时间戳低32位（若DFM=0）。但若DFM=1，这些位将不会被修改，保持为缓存地址。 当这些位的值表示缓存1的物理地址（TTSL）时，对缓存的地址对齐不做限制。 当这些位的值表示时间戳低32位（TB1AP）时，当前描述符的TTSEN位和LSG位必须置位。

■ 发送描述符3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TB2AP/TTSH[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TB2AP/TTSH[15:0]															
rw															

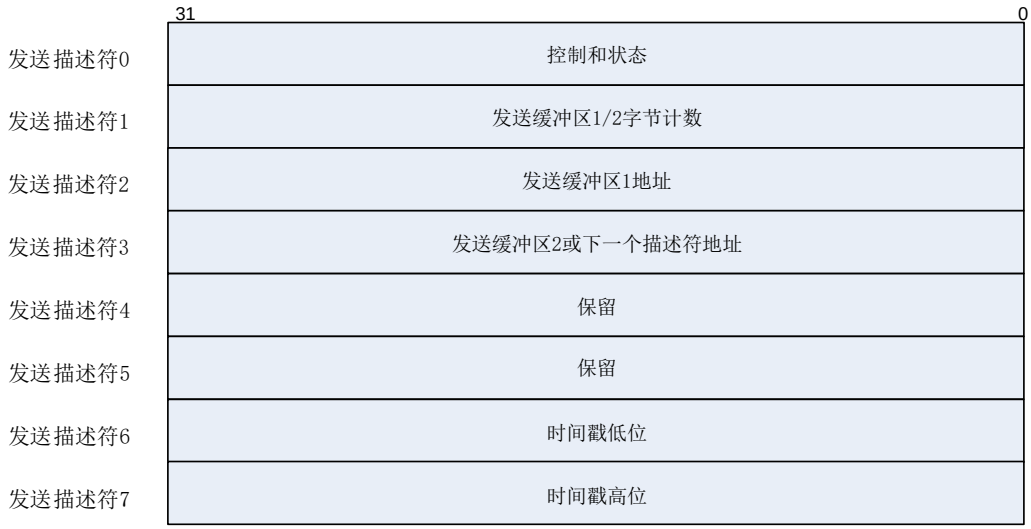
位/位域	名称	描述
31:0	TB2AP/TTSH[31:0]	发送缓存 2 地址指针（下个描述符地址）/发送帧时间戳高 32 位 在发送帧之前，应用程序必须对这些位进行配置发生缓存 2 地址（TB2AP），或者配置下一个描述符地址（由描述符类型是链型还是环型决定）。等到数据发送完后，DMA 可以用它们存放帧的时间戳高 32 位 TTSH（若 DFM=0，且 TTSEN=1）。但若 DFM=1 或 TTSEN=0，这些位将不会被修改。 当这些位的值表示缓存 2 的物理地址时（TCHM=0），对缓存的地址对齐不做限制。 当这些位的值表示下个描述符地址时（TCHM=1），这些位必须是字对齐的。 当这些位的值表示时间戳高32位时，当前描述符的TTSEN位和LSG位必须置位。

增强 TxDMA 描述符

增强 TxDMA 描述符结构体包含 8 个 32 位字：发送描述符 0~7。发送描述符 0~3 的位定义与常规 TxDMA 描述符相同，发送描述符 4~7 的位定义如下：

注意：若一个帧由多个描述符表示，则对于描述符的控制位（除了 INTC 位）只有第一个描述符的才有效。状态信息和时间戳（若使能了时间戳功能）只写回到最后一个描述符。

图 26-8. 增强发送描述符



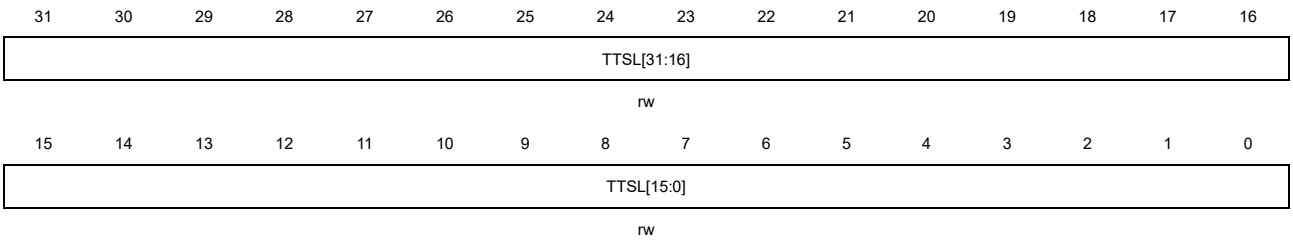
■ 发送描述符4

所有位保留。

■ 发送描述符5

所有位保留。

■ 发送描述符6



位/位域	名称	描述
31:0	TTSL[31:0]	发送时间戳低 32 位 当 TTSEN 位和 LSG 位都为 1 时，这些位用于记录当前发送帧的时间戳低 32 位值。

■ 发送描述符7

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

位/位域	名称	描述
31:0	TTSH[31:0]	发送时间戳高 32 位 当 TTSEN 位和 LSG 位都为 1 时，这些位用于记录当前发送帧的时间戳高 32 位值。

DMA 接收帧处理

当接口上出现一个帧的时候，MAC 开始接收帧。同时，地址过滤模块开始工作，如果这个帧没有通过地址过滤，则 MAC RxFIFO 将忽略该帧，不会将其通过 RxDMA 转发给接收缓存。如果这个帧通过了地址过滤，则其在不同的转发条件满足时会被转发给接收缓存。在直通模式下，这个转发条件是指接收的帧长大于等于设好的接收阈值。在存储转发模式下，这个转发条件是指 FIFO 里存入了完整的帧时。在接收帧的过程中，当发生以下任意一种情况时，将丢弃 RxFIFO 中的数据，并且不转发数据：

- RxFIFO中数据少于64字节；
- 在接收过程中发生了冲突；
- 提前终止接收帧。

当满足转发条件时，RxDMA 控制器开始将数据从 RxFIFO 中传输到接收缓存中。若当前缓存中包含了帧起始，则在 RxDMA 控制器写回帧接收状态的时候会将接收描述符 0 中的 FDES 位置位，以表明这个描述符中存储的是帧的第一部分。若当前缓存中包含了帧结尾，则在 RxDMA 控制器写回帧接收状态的时候会将接收描述符 0 中的 LDES 位置位，以表明这个描述符中存储的是帧的最后一部分。通常当接收缓存大小大于接收帧的长度时，FDES 位和 LDES 位会在同一个描述符中置位。当缓存接收到了帧结尾，或者当前描述符的缓存不足以存储整个帧时，RxDMA 将获取下一个接收描述符，并将上一个描述符的接收描述符 0 的 DAV 位清零以关闭上个描述符。当 LDES 位置位时，描述符其他状态也会更新，并且 ENET_DMA_STAT 寄存器中的 RS 位将置位（当 DINTC=0 时立刻置位，当 DINTC=1 时延迟置位）。当接收到一个新的帧时，如果描述符的 DAV 位为’1’，则重复上述的 RxDMA 控制器操作。如果描述符的 DAV 位为’0’，则 DMA 控制器进入挂起状态，并设置 ENET_DMA_STAT 寄存器的 RBU 位为’1’。记录描述符列表地址指针当前值，并在退出挂起状态后作为描述符开始的地址。

DMA 接收管理

RxDMA 控制器的工作流程如下：

1. DMA接收描述符初始化，置接收描述符0的DAV位为’1’；
2. 将ENET_DMA_CTL寄存器的SRE位置为’1’，使能RxDMA控制器。DMA进入运行状态后，会从ENET_DMA_RDTADDR寄存器配置的描述符列表基地址获取接收描述符。如果获取的描述符DAV位为1，则当前描述符开始接收帧。但如果检测到取到的描述符正在被

CPU操作而不可用（DAV=0），则DMA进入挂起状态，跳到步骤9；

3. 如果获取的描述符显示描述符由DMA占有（DAV=1），那么该描述符的控制位和缓存地址就会被DMA所记录；
4. 处理接收到的帧，并从RxFIFO将数据传输到接收缓存；
5. 如果缓存被填满或者帧传输结束，接收控制器会从描述符队列中获取下一个接收描述符；
6. 如果当前帧传输结束，DMA操作跳到步骤7。如果当前帧传输没有结束(未接收到帧尾EOF)，则可能发生两种情况：
   - 下一个描述符的DAV位为'0'。如果接收帧清空功能使能，则RxDMA控制器将接收描述符0的描述符错误位DERR位置位。然后RxDMA控制器将当前描述符的DAV位清零以关闭描述符，并根据帧清空功能是否使能来确定是否置位LSG位（若使能则置位LSG，反之则不置位LSG）。之后DMA操作跳到步骤8；
   - 下一个描述符的DAV位为'1'。那么RxDMA将DAV位清零以关闭当前描述符，之后操作退回步骤4；
7. 如果使能了IEEE 1588时间戳功能，在接收帧完成后DMA控制器会把获取的时间戳的低位和高位（如果接收帧符合需要记录时间戳的帧的条件），分别写入当前描述符的接收描述符2和接收描述符3（当DFM=0时），或者写入当前描述符的接收描述符6和接收描述符7（当DFM=1时）。同时DMA把从MAC处返回的接收状态信息写入接收描述符0，并把DAV位清'0'，把LSG位置'1'；
8. 如果新获取的描述符DAV位为'1'，则RxDMA控制器操作跳动步骤4。如果DAV位为'0'，则RxDMA控制器进入挂起状态，并设ENET_DMA_STAT寄存器的RBU位为1。如果使能了接收帧清空功能，则在DMA进入挂起状态之前，控制器会清空接收帧；
9. 在挂起状态下，有两种方法退出该状态。一种方法是向接送查询使能寄存器ENET_DMA_RPEN中写入任意值。另一种方法是RxFIFO收到下一帧数据，这意味着在直通模式下时，帧数据字节数需要大于设置的阈值，或者在存储转发模式下，需要收到整个帧。当DMA退出暂停状态后，DMA会获取下一个描述符，并跳到步骤2。

获取接收描述符

只要满足下列条件任意一个或多个，DMA 就会尝试获取接收描述符：

- 在寄存器ENET_DMA_CTL的接收开始/停止位SRE从'0'变为'1'，使DMA控制器进入运行状态的时候；
- 当前描述符的整个缓存大小（对于链结构为缓存1，对于环结构为缓存1和2）不足以接收整个帧，也就是说在接收到帧的结尾之前，当前描述符的缓存已满；
- 在一个完整的帧传送到接收缓存之后，并在当前描述符关闭之前；
- 在挂起状态时，MAC接收到新的帧；
- 向接送查询使能寄存器ENET_DMA_RPEN写入任意值。

挂起状态时接收到新的帧时的处理

在挂起状态时，当接收到一个新的帧，并且满足转发条件时（转发条件如上所述），RxDMA 将获取帧的描述符。如果接收描述符 0 的 DAV 位为'1'，则 RxDMA 控制器退出挂起状态，返回运行状态开始接收帧。但当接收描述符 0 的 DAV 位为'0'，则应用程序可以通过配置 ENET_DMA_CTL 寄存器中 DAFRF 位来选择是否清空 RxFIFO 中的帧。如果 DAFRF=0，则 RxDMA 控制器将丢弃 FIFO 所接收的帧数据，并将丢失帧计数器 MSFC 加 1。若 DAFRF=1，

则可以阻止丢弃 RxFIFO 顶部的帧，除非 RxFIFO 满，丢失帧计数器 MSFC 的值不会增加。在 DAV 位为’0’时，ENET_DMA_STAT 寄存器中的 RBU 位将被置位，RxDMA 控制器仍处于挂起状态。

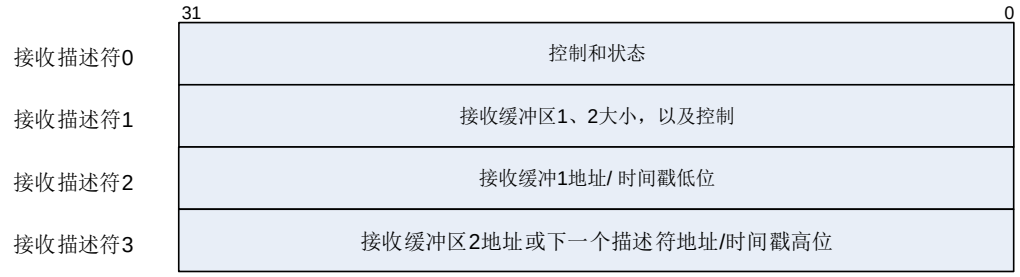
带 IEEE1588 时间戳的 RxDMA 描述符格式

如果使能了 IEEE 1588 功能，则 MAC 控制器会在带时间戳的帧接收完成之后，DMA 关闭描述符之前（DAV 位清’0’），将时间戳写入接收描述符 2 和接收描述符 3(DFM=0)，或接收描述符 6 和接收描述符 7(DFM=1)。

常规 RxDMA 描述符

常规 RxDMA 描述符结构体包含 4 个 32 位字，接收描述符 0~接收描述符 3。接收描述符 0~接收描述符 3 的位定义如下：

图 26-9. 常规接收描述符



■ 接收描述符0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DAV	DAFF	FRML[13:0]													
rw	rw	rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRS	DERR	SAFF	LERR	OERR	VTAG	FDES	LDES	IPHERR/ TSV	LCO	FRMT	RWDT	RERR	DBERR	CERR	PCERR/ EXSV
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	DAV	描述符转发条件位 该位指示了DMA控制器是否可以占有该描述符。DMA会在帧完整接收或者描述符指向的缓存被填满以后把该位清’0’。 0：表示CPU占有描述符 1：表示DMA占有描述符
30	DAFF	未通过目标地址过滤器位 0：接收帧通过目标地址过滤器 1：接收帧没有通过目标地址过滤器

29:16	FRML	帧长位 这些位表示了传送到输入缓存的接受帧字节长度（如果输入帧不是类型帧则包括CRC部分，如果输入帧是类型帧，则是否包括CRC部分由ENET_MAC_CFG寄存器的TFCD位决定）。只有LDES位为1且DERR位为0时，此位才有效。若LDES位和错误汇总位ERRS位都为'0'，则这些位表示当前接收帧已经传送到内存里的累计字节数。 注意：帧长度值为0意味着由于某种原因（例如FIFO溢出或在接收过程中动态地修改了过滤器的值，导致未通过过滤器的情况等）造成写入FIFO的帧不完整。
15	ERRS	错误汇总位 该位只在LDES位为'1'时有效。 当DFM = 0时，该位为下列位的逻辑"或"： DERR：描述符错误 OERR：溢出错误 LCO：延迟冲突 RWDT：看门狗超时 RERR：接收错误 CERR：CRC错误 IPHERR = 0, FRMT = 1且PCERR = 1：数据校验和错误 IPHERR = 1, FRMT = 1且PCERR = 0：报头校验和错误 IPHERR = 1, FRMT = 1且PCERR = 1：报头及数据校验和错误 当DFM = 1时，该位为下列位的逻辑"或"： IPPLDERR：IP帧数据错误 IPHERR：IP帧报头错误 DERR：描述符错误 OERR：溢出错误 LCO：延迟冲突 RWDT：看门狗超时 RERR：接收错误 CERR：CRC错误
14	DERR	描述符错误位 只有在LDES位置位时，此位才有效。 当当前描述符的缓存大小小于接收帧帧长同时DMA又无法占有下一个描述符，将发生描述符错误。 0：未发生描述符错误 1：发生了描述符错误
13	SAFF	未通过源地址过滤器位 0：未发生未通过源地址过滤器事件 1：帧没有通过源地址过滤器
12	LERR	长度错误位 只有在 FRMT 位为'0'时，此位才有效。 该位指示了接收到的以太网帧头长度/类型域的值是否与接收帧的实际长度匹配。 0：未发生长度错误

1：发生了长度错误

11	OERR	溢出错误位 当RxFIFO发生了溢出，而接收帧已有部分被传送到输入缓存时，溢出错误位置位。 0：未发生溢出错误 1：发生了RxFIFO溢出，帧数据无效
10	VTAG	VLAN标签位 0：接收帧非VLAN帧 1：当前的接收帧为VLAN帧
9	FDES	第一个描述符 该位表示当前描述符是否存放了接受帧的第一部分。 0：当前描述符未存放帧的第一部分 1：当前描述符存放了帧的第一部分
8	LDES	最后一个描述符 该位表示当前描述符是否存放了接受帧最后一部分。 0：当前描述符未存放帧的最后一部分 1：当前描述符存放了帧的最后一部分
7	IPHERR/TSV	IP帧报头错误/时间戳有效位 当DFM=0时，则将位7结合位5和位0，查询错误状态表，判断发生了什么错误。 当DFM=1时，则表示已获取时间戳并将时间戳值写到了接收描述符6和接收描述符中，该位仅在LDES位置位时有效。
6	LCO	延迟冲突位 延迟冲突表示在接收到64字节数据后是否发生了冲突。 该位仅在半双工模式下有效 。 0：未发生延迟冲突 1：接收帧的过程发生了延迟冲突
5	FRMT	帧类型 当DFM=0时，则将位7结合位5和位0，查询错误状态表，判断发生了什么错误。 当DFM=1时，则表示接受帧为以太网类型帧或标签帧。 当接收帧为过短帧时，该位无效。 0：接收到的帧是IEEE802.3帧 1：接收到的帧是以太网类型的帧（以太网帧头长度/类型域的值大于等于0x0600，或是一个标签帧）。
4	RWDT	接收看门狗超时 当WDD=0，该位表示已接收到超过2048字节的帧数据。 当WDD=1，该位表示已接收到超过16384字节的帧数据。 0：未发生接收看门狗超时 1：在接收帧的过程中发生了看门狗超时，当前接收帧将被截断。
3	RERR	接收错误 该位表示帧接收过程中，在RX_DV信号有效时是否收到有效接口信号RX_ER。 0：未发生接收错误

1：发生了接收错误

2	DBERR	Dribble位错误 该位表示接收到的数据帧中存在半字节，只有 MII 模式时该位有效。 0：未发生Dribble位错误 1：发生了Dribble位错误
1	CERR	CRC错误 该位表示接受帧的帧校验序列是否与硬件计算结果不匹配，只有在 LDES 位为'1'时该位有效。 0：未发生CRC错误 1：检测到接收帧发生了CRC错误
0	PCERR/EXSV	数据校验和错误/扩展状态有效位 当DFM=0时，则将位7结合位5和位0，查询错误状态表，判断发生了什么错误。 当DFM=1时，该位表示接收描述符4有效。 该位仅在LDES位置位时有效。 0：接收描述符4无效 1：接收描述符4有效

*表26-9. 接收描述符0错误状态描述，仅适用于常规描述符（DFM=0）*显示了接收描述符0中位IPHERR、FRMT、PCERR取值的含义：

表 26-9. 接收描述符 0 错误状态描述，仅适用于常规描述符（DFM=0）

位7: IPHERR	位5: FRMT	位0: PCERR	帧状态
0	0	0	IEEE802.3类型帧（长度域值小于0x0600或非标签帧）
0	0	1	IPv4/IPv6类型帧，未检测到IP报头校验和错误；由于不支持的数据格式，未执行数据校验和检测
0	1	0	IPv4/IPv6类型帧，未检测到校验和错误
0	1	1	IPv4/IPv6类型帧，检测到数据校验和错误。 该错误可能原因： 1）计算的校验和值与接收帧中的校验和域值不匹配 2）接收帧长度域值与实际接收帧字节数不符
1	0	0	保留
1	0	1	类型帧（长度/类型域值大于等于0x600）或标签帧，但非IPv4/IPv6帧，不检测数据校验和
1	1	0	IPv4/IPv6类型帧，检测到报头校验和错误。 该错误可能原因： 1) 类型域值与数据包中的版本值不一致 2) 计算的报头校验和值与报头校验和域值不匹配 3）IP报头字节数不足
1	1	1	IPv4/IPv6类型帧，检测到IP报头校验和错误和数据校验和错误

■ 接收描述符1

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

DINTC	保留	RB2S[12:0]
rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RERM	RCHM	保留	RB1S[12:0]												
Rw	rw		rw												

位/位域	名称	描述
31	DINTC	关闭接收完成中断 0：接收完成后ENET_DMA_STAT寄存器的RS位（位6）会立即置1，此时若使能了相应中断则会触发中断。 1：在接收帧成功接收完成后，ENET_DMA_STAT寄存器的RS位（位6）不会立即置1，但会在可配置的延时（由ENET_DMA_RSWDC寄存器配置）之后置位。
30:29	保留	必须保持复位值。
28:16	RB2S[12:0]	接收缓存2大小 表示接收缓存2的大小（以字节为单位）。缓存大小必须被设为4的倍数。这些位在RCHM位为'1'时被忽略。
15	RERM	接收描述符环形结构结尾 该位表示到达的是描述符列表中的最后一个描述符，下个描述符自动返回列表的基地址。 0：当前描述符不是最后一个描述符 1：到达描述符列表的最后一个描述符
14	RCHM	第二地址链表 0：描述符里的第二个地址指向第二缓存的地址 1：描述符里的第二个地址是下一个描述符的地址，该位仅在在链模式下使用。忽略RB2S [12:0]的值。 注意：当RERM=1，则即使该位置位，下个描述符也将返回列表基地址。
13	保留	必须保持复位值。
12:0	RB1S[12:0]	接收缓存1大小 表示接收缓存1的大小（以字节为单位）。缓存大小必须被设为4的倍数。

■ 接收描述符2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RB1AP/RTSL[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RB1AP/RTSL[15:0]															
rw															

位/位域	名称	描述
31:0	RBAP1/RTSL[31:0]	接收缓存1地址指针/发送帧时间戳低32位

这些位有2个功能：缓冲区1的地址指针（RB1AP）和时间戳低32位（RTSL）。
RB1AP：在DMA控制器获取该描述符之前，配置该位为缓存区1的地址。如果位RB1S不为'0'，则用该位的地址来存储接收的数据帧。对缓存的地址对齐不做限制。
RTSL：当使能了时间戳功能同时LDES位为'1'时，如果接收帧通过了地址过滤，并且置位了对应的帧类型使能位，则DMA会将时间戳低32位写入这些位。如果接收帧没有置位对应的帧类型使能位，则这些位会保持RB1AP的值。

■ 接收描述符3

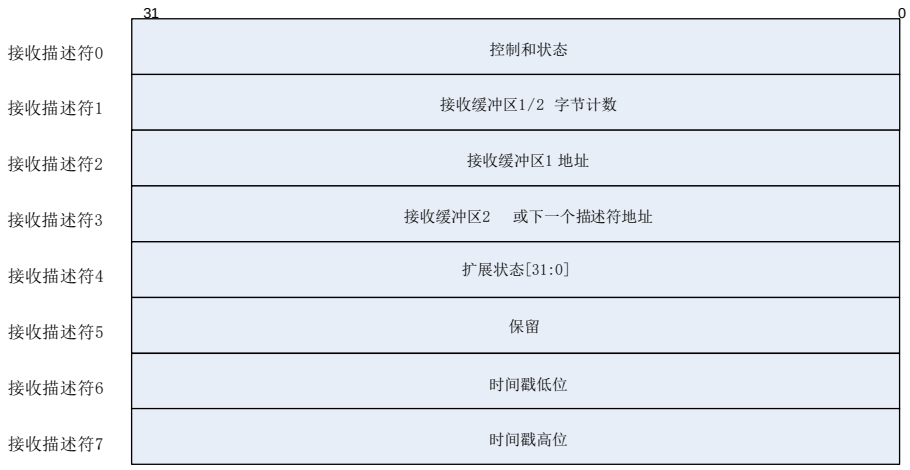
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RB2AP/RTSH[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RB2AP/RTSH[15:0]															
rw															

位/位域	名称	描述
31:0	RB2AP/RTSH[31:0]	接收缓存2地址指针（或者下一个描述符地址指针）/接收帧时间戳高32位 这些位有2个功能：数据接收时这些位存放缓冲区2的地址或下个描述符的地址（RB2AP），或帧时间戳高32位（RTSH）。 RB2AP：在DMA控制器获取该描述符之前，配置该位为缓存区2的地址（RCHM=0）或下个描述符地址（RCHM=1）。如果RCHM=1，且RERM=1，则忽略这些位。如果该地址表示下个描述符地址，则地址需为字对齐的，没有其他的限制条件。 RTSH：当使能了时间戳功能同时帧的最后部分LDES位为'1'时，如果接收帧通过了地址过滤，并且置位了对应的帧类型使能位，则DMA会将时间戳高32位写入这些位。如果接收帧没有置位对应的帧类型使能位，则这些位会保持RB2AP的值。

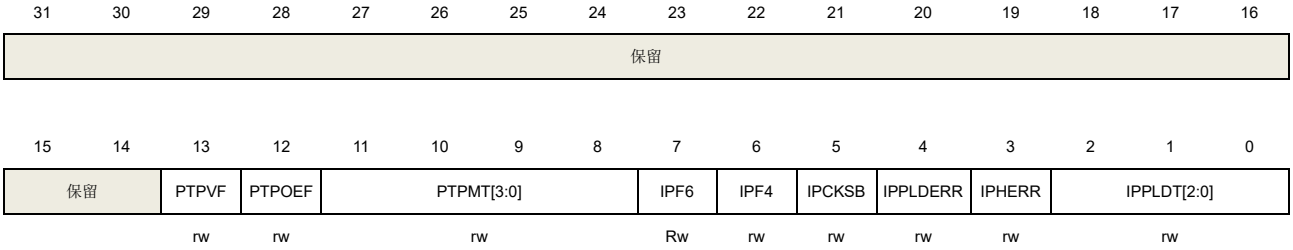
增强 RxDMA 描述符

增强 RxDMA 描述符结构体包含 8 个 32 位字：接收描述符 0~接收描述符 7。接收描述符 0~接收描述符 3 的位定义与常规 RxDMA 描述符相同，接收描述符 4~接收描述符 7 的位定义如下：

图 26-10. 增强接收描述符



■ 接收描述符4



位/位域	名称	描述
31:14	保留	必须保持复位值。
13	PTPVF	PTP 版本格式位 0：版本1格式 1：版本2格式
12	PTPOEF	以太网PTP帧 0：当PTPMT不为0时，接收的PTP帧是IP-UDP帧。 1：接收的PTP帧是IEEE802.3以太网帧
11:8	PTPMT[3:0]	PTP消息类型 PTP消息类型解码为以下几种： 0x0：未收到PTP消息 0x1：SYNC 0x2：FOLLOW_UP 0x3：DELAY_REQ 0x4：DELAY_RESP 0x5：对于点对点透明时钟：PDELAY_REQ 对于普通时钟或边界时钟：ANNOUNCE 0x6：对于点对点透明时钟：PDELAY_RESP 对于普通时钟或边界时钟：MANAGEMENT 0x7：对于点对点透明时钟：PDELAY_RESP_FOLLOW_UP 对于普通时钟或边界时钟：SIGNALING
7	IPF6	IPv6帧位 0：接收帧不是IPv6帧 1：接收帧是IPv6帧
6	IPF4	IPv4帧位 0：接收帧不是IPv4帧 1：接收帧是IPv4帧
5	IPCKSB	绕过IP帧校验和 该位仅在接收帧为IPv6或IPv4帧时有效。 0：没有绕过接收帧校验和功能 1：绕过了接收帧校验和功能
4	IPPLDERR	IP帧数据错误位

在以下任意情形中，该位会被置位：1）硬件计算的校验和域TCP，UDP或ICMP帧校验和域值不匹配 2）IP报头数据长度域值与接收到的帧数据长度不符。
0：在接收帧中没有发生帧数据错误
1：在接收帧中发生了帧数据错误

3	IPHERR	IP报头错误 在以下任意情形中，该位会被置位：1）硬件计算的校验和值与IP报头校验和域值不匹配 2）以太网帧类型域值与IP数据包中版本域值不一致（例如类型域值为0x800，但版本域值不为0x4；类型域值为0x86dd，但版本域值不为0x6）。 0：没有发生IP报头错误 1：发生了IP报头错误
2:0	IPPLDT[2:0]	IP帧数据类型位 仅在IPFCO=1，IPHERR=0且LDES=1时，这些位有效。 0x0：不支持的数据类型或忽略的IP数据类型 0x1：数据类型为UDP 0x2：数据类型为TCP 0x3：数据类型为ICMP 0x4~0x7：保留

- 接收描述符5

所有位保留。

- 接收描述符6

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

位/位域	名称	描述
31:0	RTSL[31:0]	接收时间戳低 32 位 当使能了时间戳功能同时帧的最后部分 LDES 位为'1'时，如果接收帧通过了地址过滤，并且置位了对应的帧类型使能位，则 DMA 会将时间戳低 32 位写入这些位。

- 接收描述符7

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

位/位域	名称	描述
31:0	RTSH[31:0]	接收时间戳高 32 位 当使能了时间戳功能同时帧的最后部分 LDES 位为'1'时，如果接收帧通过了地址过滤，并且置位了对应的帧类型使能位，则 DMA 会将时间戳高 32 位写入这些位。

26.3.4. MAC 统计计数器：MSC

为了了解发送和接收帧的统计情况，利用一组计数器来收集相关的统计数据。这些 MAC 计数器被称为 MAC 统计计数器（MSC）。在章节"ENET 寄存器"中有这些寄存器的详细功能说明。

当发送帧没有出现帧下溢、没有载波、载波丢失、顺延(Deferral)过多、延迟冲突，过度冲突和 jabber 超时等情况时，可以称为"好帧"，MSC 发送计数器会自动更新。

- 当接收帧没有出现对齐错误、CRC错误、过短帧、长度错误、超出范围和MII_RX_ER引脚上的错误信号有效等情况时，可以称为"好帧"，MSC接收计数器会自动更新。其中，CRC错误是指CRC计算结果与帧校验序列值不一致，过短帧表示帧长少于64字节，长度错误表示长度域值与实际接收到的字节数不符，超出范围表示长度域值超过IEEE 802.3所规定的最大值，即对于非标签帧最大值为1518字节，对于VLAN标签帧最大值为1522字节。

注意：当被丢弃的帧是长度小于 6 字节的过短帧（没有完整接收到目标地址）时，MSC 接收计数器也会更新。

26.3.5. 唤醒管理：WUM

以太网模块支持两种将系统从深度睡眠模式唤醒的方法。分别为远程唤醒帧和 Magic Packet 唤醒帧。为了减小功耗，可以使主机系统和以太网模块进入低功耗状态，从而可以停止由 HCLK 驱动的电路以及发送时钟。但由接收时钟驱动的电路将继续工作，以监听唤醒帧。如果将 ENET_MAC_WUM 寄存器的 PWD 位置 1，则以太网模块进入低功耗状态。在低功耗状态下，MAC 会丢弃所有的帧，直到退出低功耗状态。此时可以采用上述的两种方式能够退出低功耗状态。将 ENET_MAC_WUM 寄存器的 WFEN 置 1，以设置当收到远程唤醒帧时唤醒以太网模块，或将 ENET_MAC_WUM 寄存器的 MPEN 置 1，以设置当收到 Magic Packet 唤醒帧时唤醒以太网模块。当任一个唤醒功能被使能，一旦 MAC 接收到相应的唤醒帧，以太网模块将产生一个唤醒中断，并退出低功耗状态。

远程唤醒帧检测

将 ENET_MAC_WUM 寄存器的 WFEN 置 1 可以使能远程唤醒检测。当 MAC 处于低功耗状态，且远程唤醒使能位为'1'时，MAC 会进行唤醒帧过滤。如果输入帧通过了过滤器命令的地址过滤，而且过滤器 CRC-16 与被检查的输入帧匹配，则认为接收到唤醒帧，随后 MAC 即恢复正常工作。即便唤醒帧的长度超过了 512 字节，只要该帧有正确的 CRC 值，它仍然被认为是有效的。在接收到远程唤醒帧时还会将 ENET_MAC_WUM 寄存器的 WUFR 位置 1。如果远程唤醒中断没有被屏蔽，那么此时还将产生一个 WUM 中断。

Magic Packet 检测

另一种唤醒方法是检测 Magic Packet 唤醒帧（见 AMD 公司的“Magic Packet 技术”）。一个 Magic Packet 帧是一种特殊构成的数据包，专门用于唤醒。这种包可以被以太网模块接收、分析和识别，并用于触发一个唤醒事件。设置 ENET_MAC_WUM 寄存器的 MPEN 位为’1’可以使能此功能。这种类型的帧格式如下：目的和源地址域之后的任何位置连续 6 字节全 1（0xFFFF FFFF FFFF），接着是在没有任何中断和暂停的情况下有 16 个重复的 MAC 地址；如果这 16 次重复间有任何的间断，则需要重新在输入帧里检测 0xFFFF FFFF FFFF。WUM 模块会持续监视每一个发向本节点的帧，那些通过地址过滤的 Magic Packet 帧，MAC 会进一步检测其是否符合 Magic Packet 的格式，一旦通过检测将会使 MAC 从低功耗状态下唤醒。设备也接受多播帧作为 Magic Packet 帧。

下面是一个站地址为 0xAABB CCDD EEFF 的 Magic Packet 帧实例（MISC 表示包内各种附加的数据字节）：

<DESTINATION><SOURCE><MISC>

……FF FF FF FF FF FF

AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF

AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF

AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF

AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF

<MISC><FCS>

一旦检测到 Magic Packet 帧，ENET_MAC_WUM 寄存器的位 MPKR 会被置 1。如果使能了 Magic Packet 中断，此时还将产生对应中断。

系统在低功耗期间注意事项

在 MCU 处于深度睡眠模式时，若使能外部中断线 19，则以太网的 WUM 模块仍能够检测帧。由于 MAC 在低功耗状态也需要进行 Magic Packet/远程唤醒帧检测，因此 ENET_MAC_CFG 寄存器的 REN 位必须保持为’1’。在低功耗状态时需要把 ENET_MAC_CFG 寄存器的 TEN 位清’0’来关闭发送功能。此外，由于不需要把 Magic Packet/远程唤醒帧转发给应用，因此在低功耗状态时也要关闭以太网 DMA 模块，可以通过设置 ENET_DMA_CTL 寄存器的 STE 位和 SRE 位（分别对应 TxDMA 和 RxDMA）为’0’来关闭以太网 DMA。

推荐的进入低功耗状态和唤醒步骤如下：

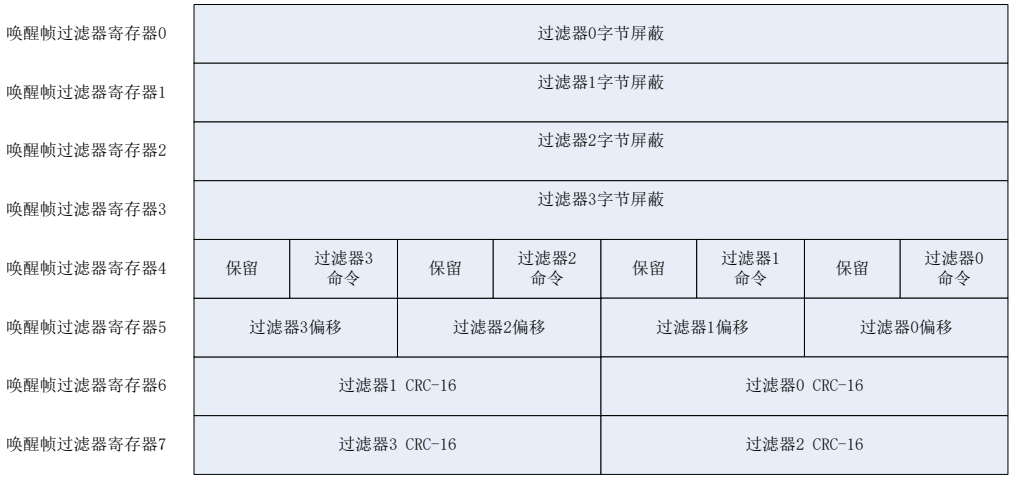
1. 等待当前帧发送完毕，然后将ENET_DMA_CTL寄存器的STE位复位来关闭TxDMA；
2. 把ENET_MAC_CFG寄存器的TEN位和REN位清’0’，来关闭MAC发射器和MAC接收器；
3. 观察ENET_DMA_STAT寄存器位RS，等待RxDMA把RxFIFO里的所有帧读出，再关闭RxDMA；
4. 配置并使能外部中断线19，使其能产生事件或者中断。如果配置了外部中断线19产生中断，则还需要编写中断处理程序ENET_WKUP_IRQ，在其中清除外部中断线19的中断标志位；

5. 设置ENET_MAC_WUM寄存器的MPEN或WFEN位（或两位）为'1'，使能Magic Packet/远程唤醒帧检测（或两种功能）；
6. 设置ENET_MAC_WUM寄存器的PWD位为'1'，使能低功耗模式；
7. 设置ENET_MAC_CFG寄存器的REN位为'1'，打开MAC接收器；
8. 设置使MCU进入深度睡眠模式；
9. 在接收到有效的唤醒帧后，以太网模块退出低功耗状态；
10. 读取ENET_MAC_WUM寄存器来清除电源管理事件标志位，打开MAC发送器，以及TxDMA和RxDMA；
11. 设置系统时钟：使能HXTAL并配置RCU时钟参数。

远程唤醒帧过滤器寄存器

唤醒帧过滤器寄存器一共有 8 个，但这些寄存器共用一个相同的偏移地址。在完成对某个过滤器寄存器的读或写的时候，内部的指针会自动指到下一个过滤器寄存器。不论是读还是写操作，强烈建议连续 8 次的操作。也就是说，对其设置时需要将设置的值分为 8 次逐一写入唤醒帧过滤器寄存器地址，读取的时候也是需要连续读 8 次唤醒帧过滤器寄存器，才能将所有值读出。

图 26-11. 唤醒帧过滤器寄存器



■ 过滤器n字节屏蔽

该寄存器定义了过滤器 n（n=0，1，2，3）使用帧的哪些字节来检查判断是否为唤醒帧。其第 31 位必须为'0',位[30:0]是字节屏蔽位。如果过滤器 n（n=0，1，2，3）的第 m 位（m=0~30）为'1'，则唤醒帧检测的 CRC 模块会处理输入帧的第[过滤器 n 偏移+m]字节，否则忽略之。

■ 过滤器n命令

共 4 位控制过滤器 n 的工作模式。最高位位 3 为地址类型选择，如果该位为'1'，则只检测多播帧；如果该位为'0'，则只检测单播帧。位 2 和位 1 必须保持为 0。位 0 是过滤器 n 的使能位，置位时使能过滤器 n，反之禁能过滤器 n。

■ 过滤器n偏移

与过滤器 n 字节屏蔽配合使用。该寄存器定义了过滤器 n 要检查的首字节在帧内的偏移量。最小允许取值是 12，代表了帧的第 13 个字节（偏移值为 0 表示帧的第 1 个字节）。

- 过滤器n CRC-16

该寄存器包含了预先写入的 CRC-16 码，用于与帧数据计算的 CRC-16 值进行比较。

26.3.6. 精确时间协议：PTP

协议的大部分是通过 UDP 层之上的应用程序软件实现的。MAC 的 PTP 模块主要是支持记录 PTP 包从以太网端口发出和收到的准确时间，并将其返回给应用程序。

关于精确时间协议(PTP)的具体内容可参见 IEEE 1588™相关文档。

基准时钟源

IEEE 1588 协议规定，通过一个 64 位寄存器来获得系统基准时间，其中高 32 位提供秒级的时间信息，低 32 位提供亚秒级的时间信息。

PTP 基准时钟输入用来生成系统基准时间（也称为系统时间），以及获取 PTP 帧的时间戳值。其频率必须大于或等于时间戳计数器的分辨率。主节点和从节点之间的时间同步精度在 0.1us 左右。

同步精度

时间同步的精度取决于以下几个因素：

- PTP基准时钟输入的频率；
- 所用晶体振荡器的特性（频漂）；
- 同步流程的执行频度。

系统时间校准

64 位 PTP 系统时间由 PTP 输入基准时钟来更新。这个 PTP 系统时间用来作为记录发送/接收时间戳的依据。该系统时间的初始化和校准支持两种模式：粗调和精调。校准的目的是纠正频率偏移。

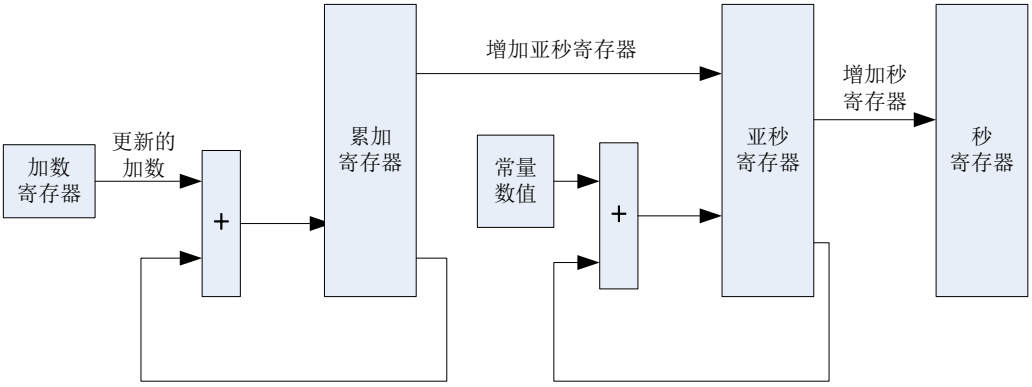
若选择了粗调的方式，则可通过配置 PTP 时间戳更新寄存器(ENET_PTP_TSUH 和 ENET_PTP_TSUL)，来进行系统时间初始化和校准。如果 TMSSTI 位被置位，则 PTP 时间戳更新寄存器被用于初始化。如果 TMSSTU 位被置位，则 PTP 时间戳更新寄存器被用于系统时间的调整，加上或者减去这个寄存器值进行校准。

若选择了精调的方式，则需要一段时间才能完成。由应用程序确定精调的频率，以确保从时钟能线性地同步于主时钟，避免不可预知的大的抖动。

这种方法是指，在每个 HCLK 周期把加数寄存器 ENET_PTP_TSADDEND 中的值加入累加器。当累加器溢出时会产生脉冲令时间戳低寄存器 ENET_PTP_TSL 的值增加。增加的值由亚秒递增寄存器 ENET_PTP_SSINC 中的值决定。

*图26-12. 系统时钟精细校准方法*演示了精调算法的流程：

图 26-12. 系统时钟精细校准方法



下面是一个具体例子用于说明精调方式如何更新系统时间：

假设系统时钟更新电路的精度需要达到 25ns，即更新的频率为 40MHz。假设基准时钟 HCLK 是 72MHz，计算频率比得 72 / 40 = 1.8。写入 ENET_PTP_TSADDEND 寄存器的值应当是 2^{32} / 1.8，等于 0x8E38 E38E。如果基准时钟频率漂低，假设降至 68MHz，此时频率比变成 68/40 = 1.7，写入 ENET_PTP_TSADDEND 寄存器的值应当是 2^{32} / 1.7 = 0x9696 9697。如果基准时钟漂高，假设升到 76MHz，写入到 ENET_PTP_TSADDEND 寄存器的值应当是 2^{32} / 1.9 = 0x86BC A1AF。。初始时，将加数寄存器设为从时钟频率 Clock Addend Value(0)，该值按上述进行计算。除了配置加数计数器之外，还需对亚秒递增寄存器进行设置才能保证达到 20ns 的精度。每次累加寄存器溢出后，该寄存器的值会对时间戳低寄存器进行更新。因为 ENET_PTP_TSL 寄存器中的 STMSS[30:0]位表示系统时间的亚秒值，其精度为 10^{9}ns / 2^{31} = 0.46ns。所以为了使系统时间精度达到 25ns，亚秒递增寄存器的值应该设为 25 / 0.46 = 0d54。

注意：下文描述的算法是以主从设备之间传输的时延 Master-to-Slave-Delay 恒定为基础的，通过该算法在若干个 Sync 周期内确定同步频率比。

算法如下：

- 定义主设备发送一个SYNC消息到从设备时的时间：MSYNCT(n)

定义从设备的本地时间：SLOCALT（n）

定义主设备的本地时间：MLOCALT（n）

计算：MLOCALT（n）= MSYNCT（n）+ Master-to-Slave-Delay（n）

- 定义发送两次SYNC消息之间的主设备时钟计数：MCLOCKC（n）

计算：MCLOCKC（n）= MLOCALT（n） - MLOCALT（n-1）

定义接收两次 SYNC 消息之间的从设备时钟计数：SCLOCKC（n）

计算：SCLOCKC（n）= SLOCALT（n）- SLOCALT（n-1）

- 定义两个计算之间的差值：DIFFCC（n）

计算：DIFFCC（n）= MCLOCKC（n）- SCLOCKC（n）

- 定义从时钟频率调整系数：SCFAF（n）

计算：SCFAF（n）=（MCLOCKC（n）+ DIFFCC（n））/ SCLOCKC（n）

■ 定义加数寄存器的时钟加数值：Clock Addend Value（n）

Clock Addend Value（n）= SCFAF（n）* Clock Addend Value（n-1）

注意：实际操作中，可能需要多个 SYNC 消息来完成主从设备的同步。

系统时间初始化流程

设置 ENET_PTP_TSCTL 寄存器的位 TMSEN 为’1’，可以使能时间戳功能。不过在把该位置’1’以后，必须首先初始化时间戳计数器来开始时间戳操作。初始化步骤如下：

1. 置位ENET_MAC_INTMSK寄存器的TMSTIM位，以屏蔽时间戳触发中断；
2. 置位ENET_PTP_TSCTL寄存器TMSEN位，以使能时间戳；
3. 根据期望时钟精度配置亚秒递增寄存器；
4. 若希望采用精调校准方式，则配置时间戳加数寄存器，并置位ENET_PTP_TSCTL寄存器的TMSARU位。若希望采用粗调校准方式，则忽略第4-6步，直接跳至第7步；
5. 轮询ENET_PTP_TSCTL寄存器的位TMSARU，直到其变为’0’；
6. 将ENET_PTP_TSCTL寄存器的TMSFCU位置位，来选择使用精调校准方式；
7. 把希望设置的系统时间值写入时间戳更新高寄存器和时间戳更新低寄存器；
8. 置位ENET_PTP_TSCTL寄存器的TMSSTI位，以初始化时间戳；
9. 一旦初始化成功后，时间戳计数器就开始工作。

系统时间更新步骤

粗调方式

1. 在时间戳更新高寄存器和时间戳更新低寄存器中写入偏移值（可以是负值）；
2. 置位ENET_PTP_TSCTL寄存器的TMSSTU位，以更新时间戳寄存器；
3. 轮询TMSSTU位，直到其被清’0’后完成。

精调方式

1. 利用前述”*系统时间校准*”介绍的算法，计算出期望的系统时钟频率所对应的加数寄存器的值；
2. 将值写入加数计数器，并设置ENET_PTP_TSCTL寄存器的TMSARU位为’1’将该值更新到PTP模块；
3. 把要求的期望时间写入期望时间高和期望时间低寄存器，并设置ENET_MAC_INTMSK寄存器的TMSTIM位为’0’来允许时间戳中断；
4. 设置ENET_PTP_TSCTL寄存器的TMSITEN位为’1’使能时间戳中断；
5. 在这个事件产生中断时，读出ENET_MAC_INTF寄存器的值以清除相应的中断标志位；
6. 重新用旧值编写时间戳加数寄存器，并设置ENET_PTP_TSCTL寄存器的TMSARU位为’1’将值更新到PTP模块。

带 PTP 功能的帧的发送与接收

在使能了 IEEE 1588（PTP）时间戳功能后，在发送帧的帧首界定码从 MAC 输出或者 MAC 接收到帧的帧首界定码的时候，时间戳值被记录。每一个等待发送的帧在 DMA 发送描述符中都

有一个标志，指示是否需要记录这个帧的时间戳，这与发送的帧是否为 PTP 帧无关。若 ENET_PTP_TSCTL 寄存器的位 ARFSEN 为’1’，则所有接收到的帧的时间戳都将被记录。如果 ARFSEN 位为’0’，则通过地址过滤的接收帧需要与 ENET_PTP_TSCTL 寄存器的配置进行匹配。换句话说，只有与 PTP 配置相匹配的帧才标记为一个 PTP 帧，并且同时记录时间戳值到描述符中。鉴别接收帧是否为 PTP 帧，接收帧中的 PTP 版本需要与 PFSV 位相符，并且对应的帧类型使能位（寄存器 ENET_PTP_TSCTL 中的位 13 到位 11）要置位。特殊地，对于非 IP 数据包的 PTP 帧（普通 802.3 以太网帧的 PTP），以太网帧的目标地址域需要为特殊的 MAC 地址（比如对于 PDELAY_REQ/PDELAY_RESP/PDELAY_RESP_FOLLOW_UP 消息类型，目标地址应为 0x0E00 00C2 8001，对于其他的消息类型，目标地址应为 0x0000 0019 1B01，详细信息请参考 IEEE1588-2008 规范）。同时，为了增强灵活性，如果 MAFEN 位置位，除了上述两个特殊地址以外，MAC1-3 寄存器中 SAF 位为复位状态的地址也可以认为是特殊地址。

记录下来的时间戳会和帧的发送/接收状态信息一起，存放在相应的发送/接收描述符里。64 位的发送帧时间戳写入 DMA 发送描述符，64 位的接受帧时间戳写入 DMA 接收描述符。具体描述见后续的”*带 IEEE 1588 时间戳的 TxDMA 描述符格式*”和”*带 IEEE1588 时间戳的 RxDMA 描述符格式*”。

内部连接触发

MAC 可以在系统时间大于等于期望时间的时候提供触发中断。使用中断会引入一段已知的中断时延再加上不确定的命令执行时间。为了计算这部分已知的中断时延时间，在系统时间大于期望值的时候，PTP 会将一个输出信号置高。将 AFIO_PCF0 寄存器的 TIMER1ITI1_REMAP 位设为 0，可将此输出信号内部连接到 TIMER1 的 ITI1 输入上。利用这个信号，由于 TIMER1 的时钟与 PTP 基准时钟（HCLK）是同步的，因此不再有任何不确定的误差。

PPS 输出信号

将 AFIO_PCF0 寄存器中的 PTP_PPS_REMAP 位置 1，可以使能 PPS 输出功能。该功能可以输出脉冲宽度为默认 125ms（其他宽度详见 *PTP PPS 控制寄存器（ENET_PTP_PPSCTL）*）的脉冲，用于检查网络全部节点之间的同步。为了测试本地从时钟和主时钟之间的差别，可以把主从设备的 PPS（秒脉冲）输出都连接到示波器，以测量 2 个时钟之间的差别。

26.3.7. 典型的以太网配置流程示例

在上电复位或系统复位之后，应用程序可按以下的典型操作流程来配置并启动以太网模块：

- 使能以太网时钟：

配置 RCU 模块来使能 HCLK 时钟和以太网发送/接收时钟。

- 配置通讯接口：

配置 AFIO_PCF0，选择接口模式（MII 或 RMII）；

配置 GPIO 模块，将相应的功能脚映射到复用功能 11（AF11）上。

- 等待复位完成：

轮询 ENET_DMA_BCTL 寄存器直到 SWR 位复位（SWR 位在上电复位后或系统复位后默认

置位）。

- 获取并配置PHY寄存器参数：

根据 HCLK 频率，配置 SMI 时钟频率，并访问 PHY 寄存器获取 PHY 的信息（例如是否支持半/全双工，是否支持 10M/100Mbit 速度等等）。根据外部 PHY 支持的模式，配置 ENET_MAC_CFG 寄存器使与 PHY 寄存器信息一致。

- 初始化以太网DMA模块用于数据传输：

配置 ENET_DMA_BCTL，ENET_DMA_RDTADDR，ENET_DMA_TDTADDR 和 ENET_DMA_CTL 寄存器，完成 DMA 模块初始化（详细信息请参考 ***DMA 控制器描述***章节）。

- 初始化用于存放描述符列表以及数据缓存的物理内存空间：

根据 ENET_DMA_RDTADDR 和 ENET_DMA_TDTADDR 寄存器中的地址，初始化发送和接收描述符(DAV=1)，以及数据缓存。

- 使能MAC和DMA模块，开始发送和接收：

置位 ENET_MAC_CFG 寄存器中的 TEN 和 REN 位，开启 MAC 发送器和接收器。置位 ENET_DMA_CTL 寄存器中的 STE 位和 SRE 位，使能 DMA 的发送和接收。

- 如果有帧要发送：
  1. 选择一个或多个描述符发送描述符，将发送帧数据写到发送描述符中指定的缓存地址中；
  2. 将这些发送帧描述符中的 DAV 位置位；
  3. 写入任意值到 ENET_DMA_TPEN 寄存器中，使 TxDMA 退出挂起模式，开始发送数据；
  4. 有两种方法来确定当前帧是否发送完毕。第一种方法为轮询当前描述符的 DAV 位直到其复位；第二种方法仅适用于当 INTC 位为 1 的情况，应用程序可以轮询 ENET_DMA_STAT 寄存器的 TS 位直到其置位。
- 如果有帧要接收：
  1. 查看描述符列表中的第一个接收描述符（其地址在 ENET_DMA_RDTADDR 寄存器中配置）；
  2. 如果接收描述符 0 的 DAV 位复位，则说明描述符已被使用过，且接收缓存空间已存储了接收帧；
  3. 处理接收帧数据；
  4. 置位当前描述符的 DAV 位，以复用当前描述符接收新的帧；
  5. 查看列表中的下一个描述符，跳到步骤 2。

26.3.8. 以太网中断

以太网部分一共有 2 个中断向量，一个用于以太网正常操作，另一个用于映射到 EXTI 线 19 的以太网唤醒事件（检测唤醒帧或者 Magic Packet）。

第一个中断向量用于由 MAC 和 DMA 产生的中断。关于 MAC 中断和 DMA 中断的详细介绍请看下文。

第二个中断向量用于 WUM 模块在唤醒事件时产生的中断，唤醒事件为远程唤醒帧接收事件或 Magic Packet 唤醒帧接收事件。唤醒事件映射到 EXTI 线 19 上，若使能了 EXTI 线 19 的上升沿中断，则唤醒事件可以使微控制器退出深度睡眠模式。此外，若使能了 WUM 中断，则 EXTI

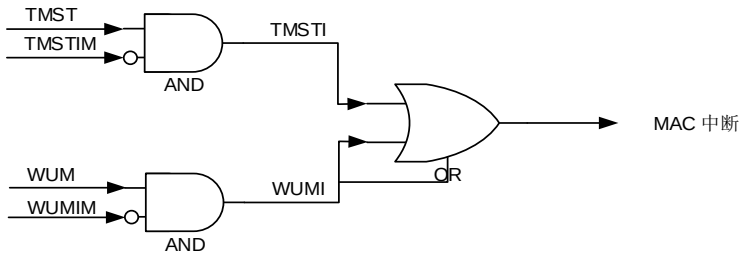
线 19 中断和以太网中断都会被触发。

注意：由于 WUM 寄存器位于 RX_CLK 域，在应用程序读 WUM 寄存器后，到这些标志位被清除可能会有可观的延迟（延迟由 HCLK 和 RX_CLK 时钟频率之间差异决定）。为避免两次进入同一个中断，强烈建议应用程序在中断里等待唤醒帧接收标志位 WUFR 和 Magic Packet 接收标志位 MPKR 变为 0 后，再退出中断服务程序。

MAC 中断

MAC 控制器有多个中断触发源。ENET_MAC_INTF 寄存器描述了所有可产生的 MAC 中断类型，每个位都有对应的中断屏蔽位来防止某一事件引发中断。MAC 中断信号为 MAC 所有中断的逻辑或。

图 26-13. MAC 控制器中断示意图



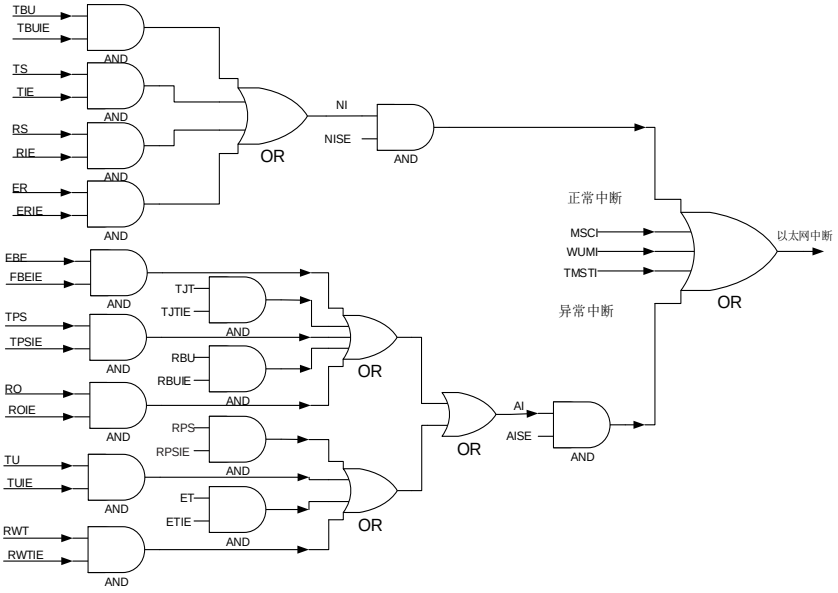
DMA 中断

DMA 控制器有两种类型中断事件：正常类和异常类。

无论什么类型的中断事件，都具有相应的中断使能位（屏蔽位）来控制是否产生中断。当所有中断事件都被清除，或中断使能位被清除，则相应的中断汇总位也被清除。如果正常类和异常类中断都被清除，则 DMA 中断将被清除。

*图26-14. 以太网中断示意图*示意了以太网模块的中断连接：

图 26-14. 以太网中断示意图


TBU
TBUIE
AND
TS
TIE
AND
RS
RIE
AND
ER
ERIE
AND
OR
NI
NISE
AND
正常中断
MSCI
WUMI
TMSTI
以太网中断
OR
异常中断
EBE
FBEIE
AND
TJT
TJTIE
AND
TPS
TPSIE
AND
RBU
RBUIE
AND
OR
RO
ROIE
AND
RPS
RPSIE
AND
OR
AI
AISE
AND
TU
TUIE
AND
ET
ETIE
AND
OR
RWT
RWTIE
AND

26.4. ENET 寄存器

ENET 基地址：0x4002 8000

支持字节，半字和字访问方式。

26.4.1. MAC 配置寄存器（ENET_MAC_CFG）

地址偏移：0x0000
复位值：0x0000 8000

MAC 配置寄存器是 MAC 的工作模式寄存器。它定义了接收和发送的工作模式。

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						TFCD	保留	WDD	JBD	保留		IGBS[2:0]			CSD
						rw		rw	rw				rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	SPD	ROD	LBM	DPM	IPFCO	RTD	保留	APCD	BOL[1:0]		DFC	TEN	REN	保留	
	rw	rw	rw	rw	rw	rw		rw	rw		rw	rw	rw		

位/位域	名称	描述
31:26	保留	必须保持复位值。
25	TFCD	类型帧CRC剥离位 0：帧的帧校验序列（最后4字节）在转发之前不会被剥离 1：帧的帧校验序列（最后4字节）在转发之前会被剥离 注意：该位仅在帧的LT域大于0x0600时有效。
24	保留	必须保持复位值。
23	WDD	关闭看门狗 该位表示已接收到了最大字节数的数据，超过的部分将被切断。 0：MAC允许接收小于或等于2048字节的帧 1：MAC关闭接收看门狗定时器，此时最多可接收16384字节的帧。
22	JBD	不检测Jabber 该位表示发送帧最大允许的发送字节数，超过的部分将被截断。 0：MAC允许的最大发送字节数为2048字节 1：MAC关闭发送Jabber定时器，此时最多可发送16384字节的帧。
21:20	保留	必须保持复位值。
19:17	IGBS[2:0]	帧间间隙选择位 这些位用于选择2个相邻发送帧之间的最短发送间隙。 0x0：96位时间 0x1：88位时间 0x2：80位时间

0x3：72位时间
0x4：64位时间
0x5：56位时间（半双工模式下不可用）
0x6：48位时间（半双工模式下不可用）
0x7：40位时间（半双工模式下不可用）

16	CSD	关闭载波侦听功能 0：MAC载波信号错误时会报错，并终止发送。 1：在半双工模式下，MAC在发送帧过程中忽略MII的CRS信号，发送过程中载波丢失或者没有载波都不会报错。
15	保留	必须保持复位值。
14	SPD	快速以太网速度 该位表示快速以太网模式下的速度： 0：10 Mbit/s 1：100 Mbit/s
13	ROD	关闭自接收功能 该位在全双工模式下可忽略 0：MAC在发送时接收所有来自PHY的数据包 1：MAC在半双工模式下不接受帧
12	LBM	回环模式 0：MAC在普通模式下工作 1：MAC在MII的回环模式下工作
11	DPM	双工模式 0：半双工模式使能 1：全双工模式使能
10	IPFCO	IP帧数据校验和 0：禁止接收端TCP/UDP/ICMP报头的校验和检验功能 1：使能接收端的帧数据校验和检测功能
9	RTD	不尝试重试 全双工模式下该位可被忽略 0：MAC会在发生冲突后按照BOL位的设定重发高达16次 1：帧仅发送一次
8	保留	必须保持复位值。
7	APCD	自动填充/CRC剥离 该位仅在非标签帧，且其长度域值小于等于1536时有效。 0：MAC会转发所有接收到的帧，而不改变帧的内容。 1：MAC会去除帧的填充字节和CRC域
6:5	BOL[1:0]	退后限制 在全双工模式下这些位可被忽略 在发生冲突后，MAC在重发当前帧之前需要延迟一段时间。这个延迟时间（dt）的时

		基单元称为时间间隙，一个时间间隙为512位时间。这个延迟时间（dt）是由下式计算得的随机整数值：0≤dt <2^k。 0x0：k = min (n, 10) 0x1：k = min (n, 8) 0x2：k = min (n, 4) 0x3：k = min (n, 1) 其中n = 重发次数。
4	DFC	顺延检验 在全双工模式下这些位可被忽略 0：禁止MAC顺延检验功能。MAC会延迟发送直到CRS信号失效。 1：MAC顺延检验功能使能。如果延迟超过24288位时间，则会发生过度顺延错误，并且MAC将中止发送。但如果在顺延时间内检测到有效的CRS（载波侦听）信号，则会将顺延计数器重置为0，重新启动顺延计时。
3	TEN	使能发送器 0：MAC关闭发送状态机，若当前帧正在发送则在完成发送后关闭。 1：MAC使能发送状态机
2	REN	使能接收器 0：MAC关闭接收状态机，若当前帧正在接收则在接收完成后关闭。 1：MAC使能接收状态机
1:0	保留	必须保持复位值。

26.4.2. MAC 帧过滤器寄存器（ENET_MAC_FRMF）

地址偏移：0x0004
复位值：0x0000 0000

MAC 帧过滤器寄存器包含了接收帧的过滤模式位。

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
FAR	保留														
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					HPFLT	SAFLT	SAIFLT	PCFRM[1:0]		BFRMD	MFD	DAIFLT	HMF	HUF	PM
					rw	rw	rw	rw		rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	FAR	接收所有帧 该位控制帧过滤器功能。 0：只有通过了地址过滤器的接收帧才会被转发给应用程序 1：所有接收到的帧都会被转发给应用程序，但过滤的结果会反映在更新接收描述符状态信息的相应标志位。

30:11	保留	必须保持复位值。
10	HPFLT	HASH或者完美过滤 0：如果HMF位或者HUF位置'1'，符合HASH过滤器的帧才能通过接收地址过滤。 1：如果HMF位或者HUF位置'1'，接收帧通过HASH过滤器或者完美过滤器中任一种，就认为通过接收地址过滤。
9	SAFLT	源地址过滤器 除了目标地址过滤之外，使能源地址过滤器。 过滤器将接收帧的源地址域值与使能的源地址寄存器中配置的值进行比较。如果源地址值相匹配，则接收描述符中的源地址匹配状态位将置位。 0：源地址过滤器关闭 1：源地址过滤器使能
8	SAIFLT	源地址过滤结果逆转 该位将源地址比较结果逆转。 0：仅在源地址过滤器结果逆转 1：使能源地址过滤器结果逆转，所有源地址符合源地址寄存器的帧会被标记为未通过源地址过滤。
7:6	PCFRM[1:0]	控制帧转发位 这些位用于设置所有控制帧的转发条件（包括单播和多播暂停帧）。 对于是否处理暂停控制帧，只取决于RFCEN位（ENET_MAC_FCTL[2]）的值。 0x0：MAC不转发任何控制帧给应用程序 0x1：MAC转发除了暂停帧以外的其他控制帧给应用程序 0x2：MAC转发所有的控制帧给应用程序，即使是没通过地址过滤器的控制帧。 0x3：MAC转发通过地址过滤器的控制帧给应用程序
5	BFRMD	不接收广播帧 0：过滤器接收所有广播帧 1：过滤器不接收所有广播帧
4	MFD	关闭多播过滤器 0：是否对多播帧进行过滤，取决于HMF位的取值。 1：所有的带多播目标地址的帧（帧的目标地址域中第一位为'1'，但不是所有位都为'1'）都能通过过滤器。
3	DAIFLT	目标地址过滤结果逆转 该位将目标地址过滤结果逆转。 0：禁用目标地址过滤结果逆转 1：使能目标地址过滤结果逆转
2	HMF	多播HASH过滤器 0：MAC会将接收到的多播帧的目标地址域和目标地址寄存器的设定值比较 1：MAC根据HASH列表对接收到的多播帧进行目标地址过滤
1	HUF	单播HASH过滤器 0：MAC会将接收到的单播帧目标地址域和目标地址寄存器的设定值比较

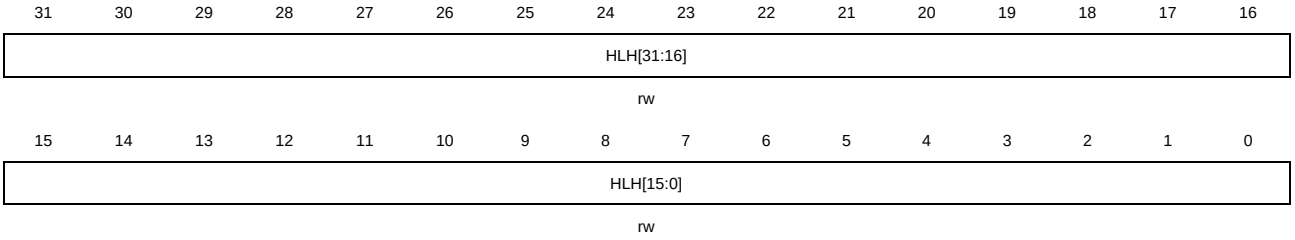
1：MAC根据HASH列表对接收到的单播帧进行目标地址过滤

0	PM	混杂模式 该位使地址过滤器无效，这意味着所有帧均可通过过滤器，同时接收描述符中状态信息的目标地址/源地址错误位总是为'0'。 0：禁用混杂模式 1：使能混杂模式

26.4.3. MAC hash 列表高寄存器（ENET_MAC_HLH）

地址偏移：0x0008
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。

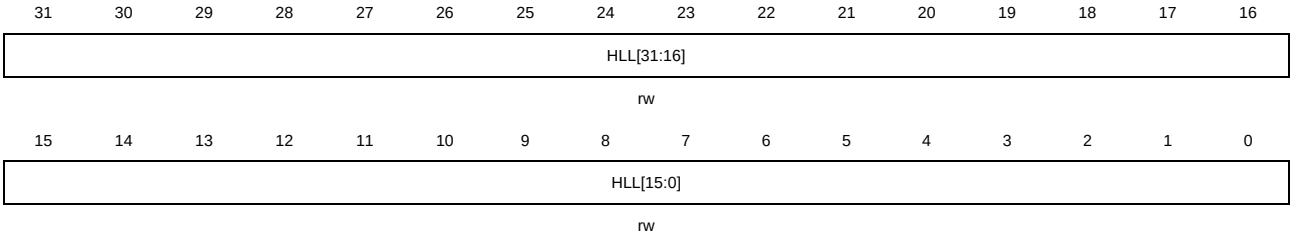


位/位域	名称	描述
31:0	HLH[31:0]	HASH列表高位 这些位是HASH列表的高32位。

26.4.4. MAC hash 列表低寄存器（ENET_MAC_HLL）

地址偏移：0x000C
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。

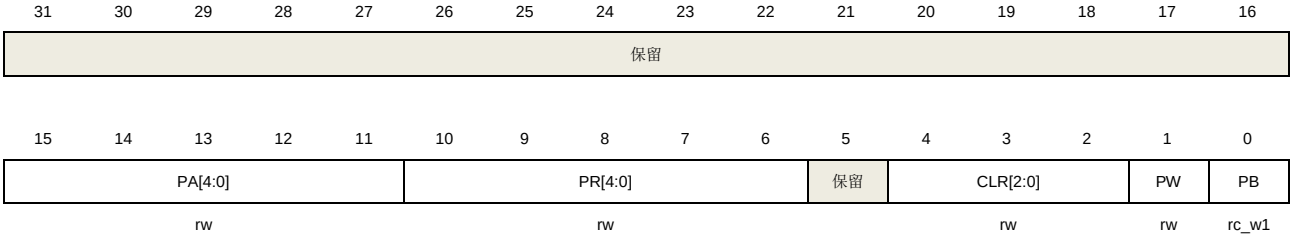


位/位域	名称	描述
31:0	HLL[31:0]	HASH列表低位 这些位是HASH列表的低32位。

26.4.5. MAC PHY 控制寄存器（ENET_MAC_PHY_CTL）

地址偏移：0x0010
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
31:16	保留	必须保持复位值。
15:11	PA[4:0]	PHY地址 这些位选择想要访问的PHY地址。
10:6	PR[4:0]	PHY寄存器 这些位选择想要访问的PHY寄存器。
5	保留	必须保持复位值。
4:2	CLR[2:0]	时钟范围 根据HCLK的频率来决定MDC的时钟分频系数。 0x0：HCLK/42（HCLK范围：60-100 MHz） 0x1：HCLK/62（HCLK范围：100-150 MHz） 0x2：HCLK/16（HCLK范围：20-35 MHz） 0x3：HCLK/26（HCLK范围：35-60 MHz） 0x4：HCLK/102（HCLK范围：150-180 MHz） 其他：保留。
1	PW	PHY写 该位指示了PHY的操作模式。 0：对PHY进行读操作 1：对PHY进行写操作
0	PB	PHY忙 该位指示了对PHY操作的状态。由应用程序置’1’后开始对PHY的进行读或者写操作，并需等到该位在操作完成后由硬件清’0’。在写ENET_MAC_PHY_CTL寄存器和读ENET_MAC_PHY_DATA寄存器之前，该位应当为’0’。

26.4.6. MAC PHY 数据寄存器（ENET_MAC_PHY_DATA）

地址偏移：0x0014
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
PD[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	PD[15:0]	PHY数据位 对于读操作，这些位为从PHY中读取的数据。对于写操作，这些位为将要写到PHY中的数据。

26.4.7. MAC 流控寄存器（ENET_MAC_FCTL）

地址偏移：0x0018
复位值：0x0000 0000

该寄存器用于配置控制帧的生成和接收。

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
PTM[15:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								DZQP	保留	PLTS[1:0]		UPFDT	RFCEN	TFCEN	FLCB/BKPA
								rw		rw		rw	rw	rw	rw

位/位域	名称	描述
31:16	PTM[15:0]	暂停时间 这些位用来设置暂停控制帧时间域的值。
15:8	保留	必须保持复位值。
7	DZQP	关闭零时间片暂停功能 0：打开零时间片暂停控制帧自动生成功能 1：关闭零时间片暂停控制帧的自动生成
6	保留	必须保持复位值。
5:4	PLTS[1:0]	暂停低阈值 这些位设置了自动重发暂停帧的定时器阈值。这个阈值应当大于0，小于位[31:16]定义的暂停时间。低阈值的计算公式为PTM-PLTS。例如，PTM = 0x80（128个时间间

隙），PLTS = 0x1（28个时间间隙），那么在第一个暂停帧发出100（128-28）个时间间隙后，将自动重发第二个暂停帧。
0x0：暂停时间 – 4个时间间隙
0x1：暂停时间 –28个时间间隙
0x2：暂停时间 – 144个时间间隙
0x3：暂停时间 – 256个时间间隙
注意：一个时间间隙是指MII接口发送512位（64字节）数据所需要的时间。

3	UPFDT	单播暂停帧检测 0：MAC只接收符合IEEE802.3规范定义的唯一多播地址的暂停帧 1：除了唯一多播地址的暂停帧，MAC同时还会使用MAC0地址（ENET_MAC_ADDR0H寄存器和ENET_MAC_ADDR0L寄存器）来检测暂停帧。
2	RFCEN	接收流控使能位 0：MAC不解析暂停帧 1：MAC解析并处理接收到的暂停帧。MAC关闭发送器一段指定的时间（接收帧中的暂停时间域值）。
1	TFCEN	发送流控使能位 0：MAC关闭发送流控功能。在全双工模式下，MAC不发送暂停帧；在半双工模式下，MAC关闭背压功能。 1：MAC开启发送流控功能。在全双工模式下，MAC使能暂停帧发送；在半双工模式下，MAC使能背压功能。
0	FLCB/BKPA	流控忙/背压激活 该位仅在TFCEN位置位时有效。 在全双工模式下，该位可发送暂停帧；在半双工模式下，该位可激活背压功能。 在全双工模式下，应用程序要确保在写ENET_MAC_FCTL寄存器之前该位为'0'。置位该位后，MAC将发送一个暂停帧到接口，在发送控制帧的过程中，该位始终为'1'，直到暂停控制帧发送完成以后，MAC将该位重置为'0'。 在半双工模式下，设置该位为'1'可以激活背压功能。在背压功能有效时，如果MAC接收到新的帧，就会在发送端发送阻塞信号，通知有冲突发生。

26.4.8. MAC VLAN 标签寄存器（ENET_MAC_VLT）

地址偏移：0x001C
复位值：0x0000 0000

该寄存器包含了用来识别 VLAN 帧的 IEEE802.1Q VLAN 标签。MAC 把接收到帧的第 13，14 字节（长度/类型域）与 0x8100 比较，再把之后的 2 个字节（第 15，16 字节）和 VLAN 标签比较。

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

VLTI[15:0]
rw

位/位域	名称	描述
31:17	保留	必须保持复位值。
16	VLTC	12位VLAN标签比较位 该位选择用12位或16位VLAN标签来进行比较。 0：接收到的VLAN帧的全部16位数据（第15和16字节）都用来与VLTI位比对 1：仅用VLTI位[11:0]12位数据和接收到VLAN帧的相应域比对
15:0	VLTI[15:0]	VLAN标签标识符位 这些位用来识别VLAN帧的802.1Q VLAN标签格式。格式如下： VLTI[15:13]：UP（用户优先级） VLTI[12]：CFI（标准格式指示符） VLTI[11:0]：VID（VLAN标识符） 如果比较的位（当VLTC=1，则为VLTI[11:0]；当VLTC=0，则为VLTI[15:0]）值是全'0'，则MAC不再比对检验VLAN帧的第15、16字节，并将接收帧的类型域值是0x8100的帧都直接视为VLAN帧。 如果用于比较的位不是全为'0'，则使用VLTI[11:0]（VLTC=1）或VLTI[15:0]（VLTC=0）进行比较。

26.4.9. MAC 远程唤醒帧过滤器寄存器（ENET_MAC_RWFF）

地址偏移：0x0028
复位值：0x0000 0000

该寄存器实质上是指向 8 个不透明的唤醒帧过滤器寄存器的指针（使用同一个偏移地址）。对该寄存器地址(偏移为 0x0028)的 8 次连续写操作，可以写入全部 8 个唤醒帧过滤器寄存器；对该寄存器地址（偏移为 0x0028）的 8 次连续读操作，可以读出全部 8 个唤醒帧过滤器寄存器。

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

图 26-15. 远程唤醒帧过滤器寄存器

	31							0
Wakeup frame filter reg0	Byte Mask of Filter-0							
Wakeup frame filter reg1	Byte Mask of Filter-1							
Wakeup frame filter reg2	Byte Mask of Filter-2							
Wakeup frame filter reg3	Byte Mask of Filter-3							
Wakeup frame filter reg4	Reserved	Filter 3 Command	Reserved	Filter 2 Command	Reserved	Filter 1 Command	Reserved	Filter 0 Command
Wakeup frame filter reg5	Offset of Filter 3		Offset of Filter 2		Offset of Filter 1		Offset Filter 0	
Wakeup frame filter reg6	Filter 1 CRC - 16				Filter 0 CRC - 16			
Wakeup frame filter reg7	Filter 3 CRC - 16				Filter 2 CRC - 16			

26.4.10. MAC 唤醒管理寄存器（ENET_MAC_WUM）

地址偏移：0x002C
复位值：0x0000 0000

该寄存器设置并监控唤醒事件。

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
WUFFRPR	保留														
rs															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						GU	保留		WUFR	MPKR	保留		WFEN	MPEN	PWD
						rw			rc_r	rc_r			rw	rw	rs

位/位域	名称	描述
31	WUFFRPR	唤醒帧过滤器寄存器指针复位 向该位写'1'，将会把远程唤醒帧过滤器寄存器指针ENET_MAC_RWFF复位，该位在指针复位完成后自动清'0'。 0：无作用 1：复位ENET_MAC_RWFF寄存器指针
30:10	保留	必须保持复位值。
9	GU	全局单播 向该位写1，所有能通过MAC地址过滤器的单播帧，都被认为是唤醒帧。 0：不是所有接收的单播帧都被认为是唤醒帧 1：所有能通过MAC地址过滤器的单播帧，都被认为是唤醒帧。
8:7	保留	必须保持复位值。
6	WUFR	接收到唤醒帧

		读本寄存器可以清'0'该位。 0：没有接收到唤醒帧 1：接收到唤醒帧，并发生唤醒事件。
5	MPKR	接收到Magic Packet 读本寄存器可以清'0'该位。 0：没有接收到Magic Packet 1：接收到Magic Packet帧，并发生唤醒事件。
4:3	保留	必须保持复位值。
2	WFEN	唤醒帧使能位 0：禁能在接收到唤醒帧时产生唤醒事件 1：使能在接收到唤醒帧时产生唤醒事件
1	MPEN	Magic Packet使能位 0：禁能在接收到Magic Packet唤醒帧时产生唤醒事件 1：使能在接收到Magic Packet唤醒帧时产生唤醒事件
0	PWD	低功耗位 该位由软件置位，由硬件复位。当该位置位，MAC丢弃所有接收到的帧。当发生了唤醒事件，使得退出低功耗模式，硬件会自动将该位清'0'。

26.4.11. MAC 调试寄存器（ENET_MAC_DBG）

地址偏移：0x0034
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						TXFF	TXFNE	保留	TXFW	TXFRS[1:0]		PCS	SOMT[1:0]		MTNI
						ro	ro		ro	ro		ro	ro		ro

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						RXFS[1:0]		保留	RXFRS[1:0]		RXFW	保留	RXAFS[1:0]		MRNI
						ro			ro		ro		ro		ro

位/位域	名称	描述
31:26	保留	必须保持复位值。
25	TXFF	TxFIFO满标志位 0：TxFIFO未满 1：TxFIFO已满
24	TXFNE	TxFIFO非空标志位 0: TxFIFO空 1: TxFIFO不为空

23	保留	必须保持复位值。
22	TXFW	正在写TxFIFO 0: 没有向TxFIFO写帧数据 1: 正在向TxFIFO写帧数据
21:20	TXFRS[1:0]	TxFIFO读操作状态 0x0：空闲状态 0x1：读状态 0x2：等待MAC发送器返回Tx状态 0x3：写发送描述符状态，或清空TxFIFO。
19	PCS	暂停状态 0：MAC发送器不处于暂停状态 1：MAC发送器处于暂停状态，并暂停发送帧。
18:17	SOMT[1:0]	MAC发送器状态 0x0：空闲状态 0x1：等待前一帧的状态返回或IFG/BACKOFF周期结束 0x2：对于全双工模式，表示正在发送暂停控制帧。 0x3：从FIFO读取待发送的帧
16	MTNI	MAC发送器不空闲位 0：MAC发送器处于空闲状态 1：MAC发送器处于非空闲状态
15:10	保留	必须保持复位值。
9:8	RXFS	RxFIFO状态 0x0：RxFIFO空 0x1：RxFIFO中字节数低于流控下阈值 0x2：RxFIFO中字节数高于流控上阈值 0x3：RxFIFO满
7	保留	必须保持复位值。
6:5	RXFRS[1:0]	RxFIFO读操作状态 0x0：空闲状态 0x1：读数据帧状态 0x2：读帧状态（包括时间戳） 0x3：清空帧
4	RXFW	正在写RxFIFO 0：没有向RxFIFO中写帧数据 1：正在向RxFIFO写帧数据
3	保留	必须保持复位值。
2:1	RXAFS[1:0]	Rx异步FIFO状态 RXAFS[1]：Rx异步FIFO在HCLK时钟域进行读操作（MAC取出数据）

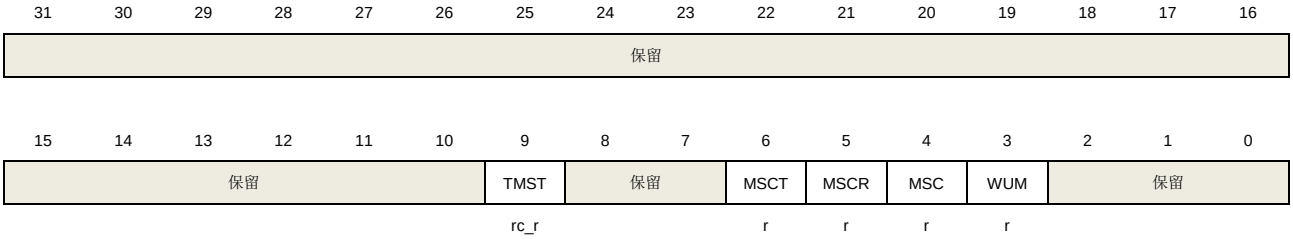
RXAFS[0]：Rx异步FIFO在MAC RX_CLK时钟域进行写操作（MAC存入数据）

0	MRNI	MAC接收器不空闲位 0：MAC接收器处于空闲状态 1：MAC接收器处于非空闲状态

26.4.12. MAC 中断状态寄存器（ENET_MAC_INTF）

地址偏移：0x0038
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。



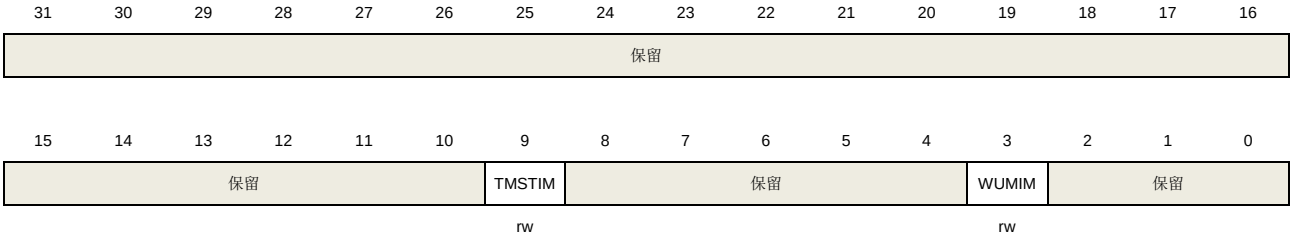
位/位域	名称	描述
31:10	保留	必须保持复位值。
9	TMST	时间戳触发状态 读 ENET_PTP_TSF 寄存器可以清'0'该位。 0：系统时间值小于期望时间值 1：系统时间值等于或者超过期望时间值
8:7	保留	必须保持复位值。
6	MSCT	MSC 发送状态 0：没有产生任一 ENET_MSC_TINTF 寄存器中的中断 1：产生任一 ENET_MSC_TINTF 寄存器中的中断
5	MSCR	MSC 接收状态 0：没有产生任一 ENET_MSC_RINTF 寄存器中的中断 1：产生任一 ENET_MSC_RINTF 寄存器中的中断
4	MSC	MSC 状态 该位为 MSCT 位与 MSCR 位的逻辑或。 0：MSCT 位和 MSCR 位均为'0' 1：MSCT 位和 MSCR 位中有位为'1'
3	WUM	WUM 状态 该位为 ENET_MAC_WUM 寄存器中的 WUFR 和 MPKR 位的逻辑或。 0：未接收到唤醒帧或者 Magic Packet 帧 1：在低功耗模式下，接收到唤醒帧或者 Magic Packet。

2:0	保留	必须保持复位值。

26.4.13. MAC 中断屏蔽寄存器（ENET_MAC_INTMSK）

地址偏移：0x003C
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。

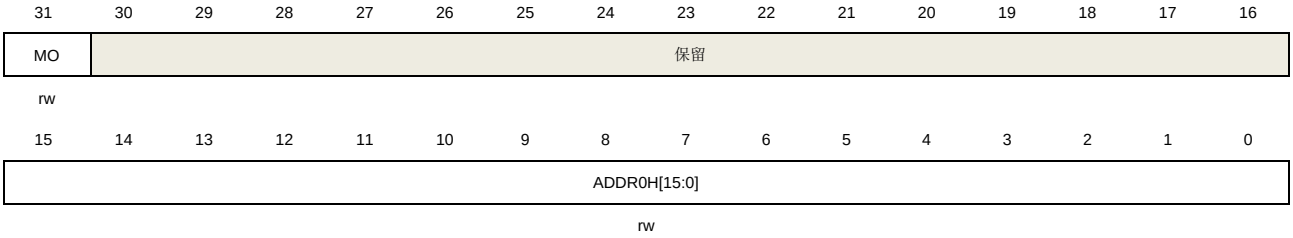


位/位域	名称	描述
31:10	保留	必须保持复位值。
9	TMSTIM	时间戳触发中断屏蔽位 0：允许产生时间戳中断 1：禁止产生时间戳中断
8:4	保留	必须保持复位值。
3	WUMIM	WUM中断屏蔽位 0：允许由于ENET_MAC_INTF寄存器的WUM状态位置位而引发的中断 1：禁止由于 ENET_MAC_INTF 寄存器的 WUM 状态位置 1 而引发的中断
2:0	保留	必须保持复位值。

26.4.14. MAC 地址 0 高寄存器（ENET_MAC_ADDR0H）

地址偏移：0x0040
复位值：0x8000 FFFF

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	MO	该位总是为'1'。

30:16	保留	必须保持复位值。
15:0	ADDR0H[15:0]	MAC地址0高16位 这些位包含了 6 字节 MAC 地址 0 的高 16 位，这些位用于作为接收帧的地址过滤，还用于发送流控中发送暂停帧时插入作为帧的源地址。

26.4.15. MAC 地址 0 低寄存器（ENET_MAC_ADDR0L）

地址偏移：0x0044
复位值：0xFFFF FFFF

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
ADDR0L[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR0L[15:0]															
rw															

位/位域	名称	描述
31:0	ADDR0L[31:0]	MAC地址0低32位 这些位包含了 6 字节 MAC 地址 0 的低 32 位，这些位用于作为接收帧的地址过滤，还用于发送流控中发送暂停帧时插入作为帧的源地址。

26.4.16. MAC 地址 1 高寄存器（ENET_MAC_ADDR1H）

地址偏移：0x0048
复位值：0x0000 FFFF

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
AFE	SAF	MB[5:0]						保留							
rw	rw	rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR1H[15:0]															
rw															

位/位域	名称	描述
31	AFE	地址过滤使能 0：不使用 MAC 地址 1 进行地址过滤 1：地址过滤器使用 MAC 地址 1 来进行完美过滤
30	SAF	源地址过滤器 0：MAC 地址 1[47:0]用来和接收帧的目标地址进行比对

		1：MAC 地址 1[47:0]用来和接收帧的源地址进行比对
29:24	MB[5:0]	屏蔽字节位 当某个位置'1'时，MAC 不再把接收帧目标地址/源地址的对应字节与 MAC 地址 1 的相应字节进行比较。每个控制位对应的 MAC 地址字节如下： MB[5]：ENET_MAC_ADDR1H [15:8] MB[4]：ENET_MAC_ADDR1H [7:0] MB[3]：ENET_MAC_ADDR1L [31:24] MB[2]：ENET_MAC_ADDR1L[23:16] MB[1]：ENET_MAC_ADDR1L[15:8] MB[0]：ENET_MAC_ADDR1L [7:0]
23:16	保留	必须保持复位值。
15:0	ADDR1H[15:0]	MAC 地址 1 高[47:32]位 这些位包含了 6 字节的 MAC 地址 1 的高 16 位。

26.4.17. MAC 地址 1 低寄存器（ENET_MAC_ADDR1L）

地址偏移：0x004C
复位值：0xFFFF FFFF

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
ADDR1L[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR1L[15:0]															
rw															

位/位域	名称	描述
31:0	ADDR1L[31:0]	MAC 地址 1 低 32 位 这些位包含了 6 字节 MAC 地址 1 的低 32 位。

26.4.18. MAC 地址 2 高寄存器（ENET_ MAC_ADDR2H）

地址偏移：0x0050
复位值：0x0000 FFFF

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
AFE	SAF	MB[5:0]						保留							
rw	rw	rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR2H[15:0]															
rw															

位/位域	名称	描述
31	AFE	地址过滤使能 0：不使用 MAC 地址 2 进行地址过滤 1：地址过滤器使用 MAC 地址 2 来进行完美过滤
30	SAF	源地址过滤器 0：MAC 地址 2[47:0]用来和接收帧的目标地址进行比对 1：MAC 地址 2[47:0]用来和接收帧的源地址进行比对
29:24	MB[5:0]	屏蔽字节位 当某个位置'1'时，MAC 不再把接收帧目标地址/源地址的对应字节与 MAC 地址 2 的相应字节进行比较。每个控制位对应的 MAC 地址字节如下： MB[5]：ENET_MAC_ADDR2H [15:8] MB[4]：ENET_MAC_ADDR2H [7:0] MB[3]：ENET_MAC_ADDR2L [31:24] MB[2]：ENET_MAC_ADDR2L[23:16] MB[1]：ENET_MAC_ADDR2L[15:8] MB[0]：ENET_MAC_ADDR2L [7:0]
23:16	保留	必须保持复位值。
15:0	ADDR2H[15:0]	MAC 地址 2 高[47:32]位 这些位包含了 6 字节的 MAC 地址 2 的高 16 位。

26.4.19. MAC 地址 2 低寄存器（ENET_MAC_ADDR2L）

地址偏移：0x0054
复位值：0xFFFF FFFF

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
ADDR2L[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR2L[15:0]															
rw															

位/位域	名称	描述
31:0	ADDR2L[31:0]	MAC 地址 2 低 32 位 这些位包含了 6 字节 MAC 地址 2 的低 32 位。

26.4.20. MAC 地址 3 高寄存器（ENET_MAC_ADDR3H）

地址偏移：0x0058
复位值：0x0000 FFFF

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
AFE	SAF	MB[5:0]						保留							
rw	rw	rw													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR3H[15:0]															
rw															

位/位域	名称	描述
31	AFE	地址过滤使能 0：不使用 MAC 地址 3 进行地址过滤 1：地址过滤器使用 MAC 地址 3 来进行完美过滤
30	SAF	源地址过滤器 0：MAC 地址 3[47:0]用来和接收帧的目标地址进行比对 1：MAC 地址 3[47:0]用来和接收帧的源地址进行比对
29:24	MB[5:0]	屏蔽字节位 当某个位置’1’时，MAC 不再把接收帧目标地址/源地址的对应字节与 MAC 地址 3 的相应字节进行比较。每个控制位对应的 MAC 地址字节如下： MB[5]：ENET_MAC_ADDR3H [15:8] MB[4]：ENET_MAC_ADDR3H [7:0] MB[3]：ENET_MAC_ADDR3L [31:24] MB[2]：ENET_MAC_ADDR3L[23:16] MB[1]：ENET_MAC_ADDR3L[15:8] MB[0]：ENET_MAC_ADDR3L [7:0]
23:16	保留	必须保持复位值。
15:0	ADDR3H[15:0]	MAC 地址 3 高[47:32]位 这些位包含了 6 字节的 MAC 地址 3 的高 16 位。

26.4.21. MAC 地址 3 低寄存器（ENET_MAC_ADDR3L）

地址偏移：0x005C

复位值：0xFFFF FFFF

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

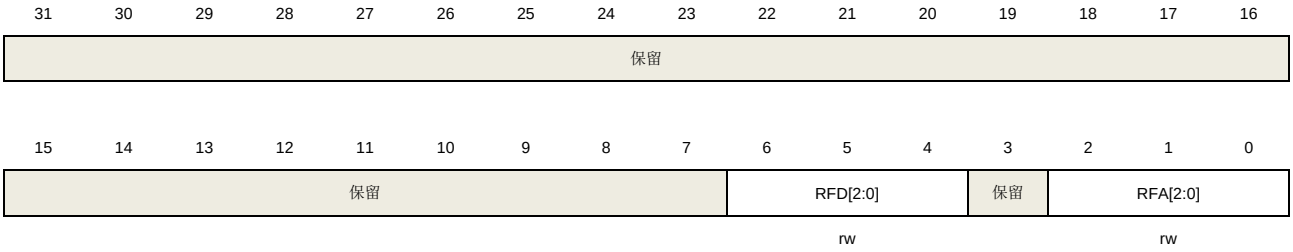
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR3L[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR3L[15:0]															
rw															

位/位域	名称	描述
31:0	ADDR3L[31:0]	MAC 地址 3 低 32 位 这些位包含了 6 字节 MAC 地址 3 的低 32 位。

26.4.22. MAC 流控阈值寄存器（ENET_MAC_FCTH）

地址偏移：0x1080
复位值：0x0000 0015

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。



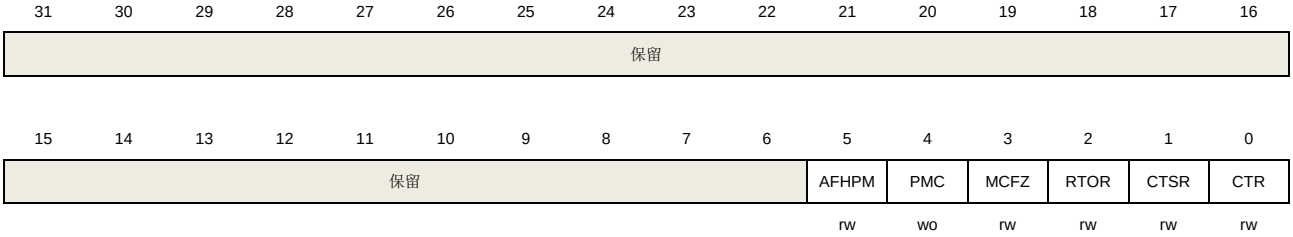
位/位域	名称	描述
31:7	保留	必须保持复位值。
6:4	RFD[2:0]	流控失效阈值 这些位设置了流控失效的阈值。这个值应当小于位[2:0]定义的流控激活阈值。当 RxFIFO 中未处理的数据低于这些位所设置的值，流控功能将自动失效。 0x0：256 字节 0x1：512 字节 0x2：768 字节 0x3：1024 字节 0x4：1280 字节 0x5：1536 字节 0x6，0x7：1792 字节
3	保留	必须保持复位值。
2:0	RFA[2:0]	流控激活阈值 这些位设置了流控激活的阈值。若使能了流控功能，当 RxFIFO 中未处理的数据超过了这些位所设置的值，流控功能将被激活。 0x0：256 字节 0x1：512 字节 0x2：768 字节 0x3：1024 字节 0x4：1280 字节 0x5：1536 字节 0x6，0x7：1792 字节

26.4.23. MSC 控制寄存（ENET_MSC_CTL）

地址偏移：0x0100

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。



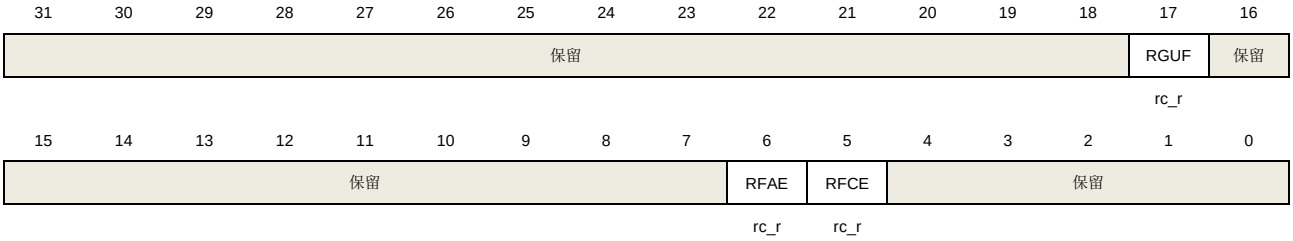
位/位域	名称	描述
31:6	保留	必须保持复位值。
5	AFHPM	近似全值或半值预设模式 0：预设 MSC 计数器的值为近似半值（0x7FFF FFF0） 1：预设 MSC 计数器的值为近似全值（0xFFFF FFF0） 注意：该位仅在 PMC 位置位时有效。
4	PMC	MAC 计数器预设位 0：无作用 1：将 MSC 计数器预设为一个预设值。预设值取决于 AFHPM 位。
3	MCFZ	MSC 计数器冻结位 0：MSC 计数器正常工作 1：冻结 MSC 计数器，保持它们的当前值。RTOR 位可在计数器冻结状态时工作。
2	RTOR	读时复位 0：读 MSC 计数器后，计数器不复位。 1：读 MSC 计数器后，计数器复位。
1	CTSR	计数器停止回转 0：计数器在计数到最大值后，会重新从 0 开始计数。 1：计数器在计数到最大值后，不会重新从 0 开始计数。
0	CTR	计数器复位 该位置位后，会在 1 个时钟周期后由硬件自动清零。 0：无作用 1：复位所有计数器

26.4.24. MSC 接收中断状态寄存器（ENET_MSC_RINTF）

地址偏移：0x0104

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。



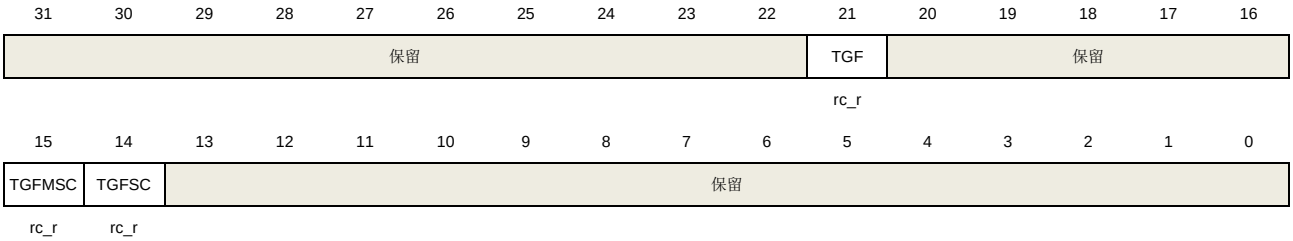
位/位域	名称	描述
31:18	保留	必须保持复位值。
17	RGUF	接收到”好”的单播帧 0：接收“好”单播帧计数器值小于最大值的一半 1：接收“好”单播帧计数器值达到最大值的一半
16:7	保留	必须保持复位值。
6	RFAE	接收到帧对齐错误 0：对齐错误接收帧计数器值小于最大值的一半 1：对齐错误接收帧计数器值达到最大值的一半
5	RFCE	接收到帧 CRC 错误 0：CRC 错误接收帧计数器值小于最大值的一半 1：CRC 错误接收帧计数器值达到最大值的一半
4:0	保留	必须保持复位值。

26.4.25. MSC 发送中断状态寄存器（ENET_MSC_TINTF）

地址偏移：0x0108

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
31:22	保留	必须保持复位值。
21	TGF	发送”好”的帧 0：发送“好”单播帧计数器值小于最大值的一半 1：发送“好”单播帧计数器值达到最大值的一半

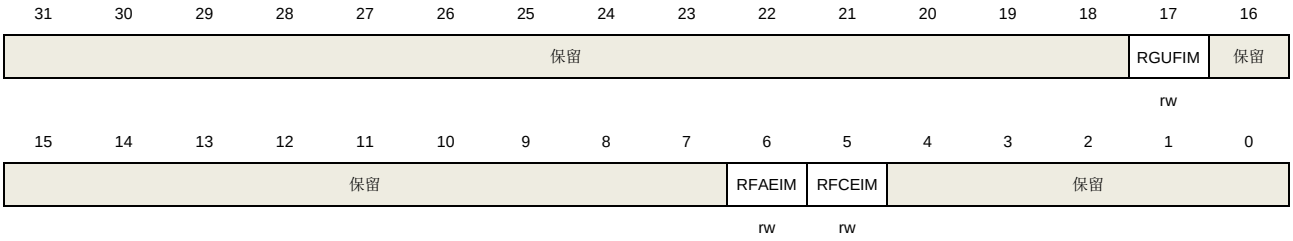
20:16	保留	必须保持复位值。
15	TGFMSC	发送”好”的帧时遇到 1 个以上冲突 0：1 次以上冲突后发送”好”帧计数器值小于最大值的一半 1：1 次以上冲突后发送”好”帧计数器值达到最大值的一半
14	TGFSC	发送”好”的帧时仅遇到 1 个冲突 0：1 次冲突后发送”好“帧计数器值小于最大值的一半 1：1 次冲突后发送”好”帧计数器值达到最大值的一半
13:0	保留	必须保持复位值。

26.4.26. MSC 接收中断屏蔽寄存器（ENET_MSC_RINTMSK）

地址偏移：0x010C
复位值：0x0000 0000

该寄存器包含当接收统计计数器达到其最大值的一半时所产生的中断的屏蔽位。

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
31:18	保留	必须保持复位值。
17	RGUFIM	接收到”好”的单播帧的中断屏蔽位 0：不屏蔽当 RGUF 位为’1’时发生的中断 1：屏蔽当 RGUF 位为’1’时发生的中断
16:7	保留	必须保持复位值。
6	RFAEIM	接收帧对齐错误中断屏蔽位 0：不屏蔽当 RFAE 位为’1’时发生的中断 1：屏蔽当 RFAE 位为’1’时发生的中断
5	RFCEIM	接收帧 CRC 错误中断屏蔽位 0：不屏蔽当 RFCE 位为’1’时发生的中断 1：屏蔽当 RFCE 位为’1’时发生的中断
4:0	保留	必须保持复位值。

26.4.27. MSC 发送中断屏蔽寄存器（ENET_MSC_TINTMSK）

地址偏移：0x0110
复位值：0x0000 0000

该寄存器可以设置相应中断的屏蔽位。

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

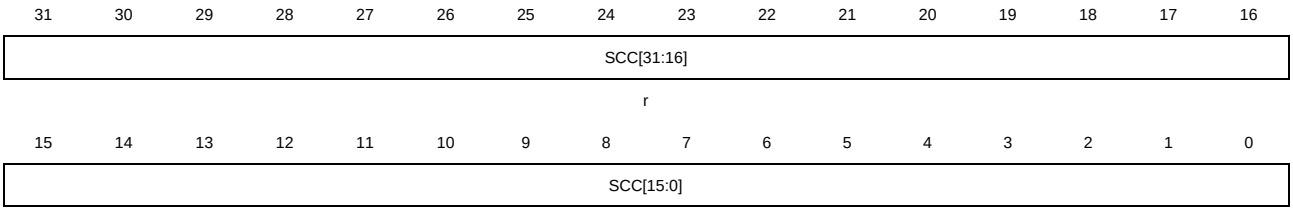
位/位域	名称	描述
31:22	保留	必须保持复位值。
21	TGFIM	发送”好”的帧的中断屏蔽位 0：不屏蔽当 TGF 位为’1’时发生的中断 1：屏蔽当 TGF 位为’1’时发生的中断
20:16	保留	必须保持复位值。
15	TGFMSCIM	遇到 1 个以上冲突后发送”好”帧中断屏蔽位 0：不屏蔽当 TGFMSC 位为’1’时发生的中断 1：屏蔽当 TGFMSC 位为’1’时发生的中断
14	TGFSCIM	仅遇到 1 个冲突后发送”好”帧中断屏蔽位 0：不屏蔽当 TFGSC 位为’1’时发生的中断 1：屏蔽当 TFGSC 位为’1’时发生的中断
13:0	保留	必须保持复位值。

26.4.28. MSC 1 次冲突后发送”好”帧的计数器寄存器（ENET_MSC_SCCNT）

地址偏移：0x014C
复位值：0x0000 0000

该寄存器统计在半双工模式下，在只遇到一次冲突后发送帧成功时的帧的数目。

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。



r

位/位域	名称	描述
31:0	SCC[31:0]	1 次冲突后发送好帧计数器 这些位是 1 次冲突后发送的”好”帧的计数器。

26.4.29. MSC 1 次以上冲突后发送”好”帧的计数器寄存器（ENET_MSC_MSCCNT）

地址偏移：0x0150
复位值：0x0000 0000

该寄存器统计在半双工模式下，遇到一次以上冲突后发送帧成功时的帧的数目。

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
MSCC[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSCC[15:0]															
r															

位/位域	名称	描述
31:0	MSCC[31:0]	1 次以上冲突后发送”好”帧计数器 这些位是 1 次以上冲突后发送”好”帧的计数器。

26.4.30. MSC 发送”好”帧计数器寄存器（ENET_MSC_TGFCNT）

地址偏移：0x0168
复位值：0x0000 0000

该寄存器统计发送”好”帧的数目。

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
TGF[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TGF[15:0]															
r															

位/位域	名称	描述
31:0	TGF[31:0]	发送”好”帧计数器 这些位是发送“好”帧的计数器。

26.4.31. MSC CRC 错误接收帧计数器寄存器（ENET_MSC_RFCECNT）

地址偏移：0x0194
复位值：0x0000 0000

该寄存器统计接收帧中有 CRC 错误的帧的数目。

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
RFCER[31:16]															
r															

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

位/位域	名称	描述
31:0	RFCER[31:0]	CRC 错误接收帧计数器 这些位是接收帧中有 CRC 错误的帧的计数器。

26.4.32. MSC 对齐错误接收帧计数器寄存器（ENET_MSC_RFAECNT）

地址偏移：0x0198
复位值：0x0000 0000

该寄存器统计接收帧中有对齐错误帧的数目。

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
RFAER[31:16]															
r															

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

位/位域	名称	描述
31:0	RFAER[31:0]	对齐错误接收帧计数器 这些位是接收帧中有对齐错误的帧的计数器。

26.4.33. MSC“好”单播帧接收帧计数器寄存器（ENET_MSC_RGUFCNT）

地址偏移：0x01C4
复位值：0x0000 0000

该寄存器统计接收到”好”单播帧的数目。

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
RGUF[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RGUF[15:0]															
r															

位/位域	名称	描述
31:0	RGUF[31:0]	“好”单播帧接收帧计数器 这些位是接收到”好“的单播帧的计数器。

26.4.34. PTP 时间戳控制寄存器（ENET_PTP_TSCTL）

地址偏移：0x0700

复位值：0x0000 2000

该寄存器用于配置时间戳的产生和更新。

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													MAFEN	CKNT[1:0]	
													rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MNMSEN	ETMSEN	IP4SEN	IP6SEN	ESEN	PFSV	SCROM	ARFSEN	保留		TMSARU	TMSITEN	TMSSTU	TMSSTI	TMSFCU	TMSEN
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	MAFEN	PTP 帧 MAC 地址过滤使能 0：无作用 1：当接收帧的类型域值为 0x88f7，则使能 MAC 地址 1-3 用于 PTP 帧过滤。
17:16	CKNT[1:0]	时间戳时钟节点类型 0x0：普通类型时钟 0x1：边界类型时钟 0x2：端对端透明类型时钟 0x3：点对点透明类型时钟
15	MNMSEN	接收主节点消息时时间戳快照使能 该位仅在 CKNT=0x0 或 0x1 时有效。 0：从节点消息时间戳快照使能 1：主节点消息时间戳快照使能
14	ETMSEN	接收事件类型的消息时时间戳快照使能 0：接收到除了 Announce，Management 和 Signaling 以外的所有其他类型的消息

		时，时间戳快照使能。 1：只有接收到事件类型的消息（SYNC，DELAY_REQ，PDELAY_REQ 和 PDELAY_RESP）时，时间戳快照使能。
13	IP4SEN	接收 IPv4 帧时时间戳使能 0：接收到 IPv4 帧时，时间戳失能。 1：接收到 IPv4 帧时，时间戳使能。
12	IP6SEN	接收 IPv6 帧时时间戳使能 0：接收到 IPv6 帧时，时间戳失能。 1：接收到 IPv6 帧时，时间戳使能。
11	ESEN	接收以太网帧时时间戳使能 0：接受到非类型帧时，时间戳失能。 1：接受到非类型帧时，时间戳使能。
10	PFSV	监听 PTP 帧版本 0：版本 1（版本为 IEEE STD. 1588-2002/1588-2008） 1：版本 2（版本为 IEEE STD. 1588-2008）
9	SCROM	亚秒计数器回转模式 0：二进制回转模式，亚秒计数器在达到 0x7FFF FFFF 以后重新从 0 计数。 1：十进制回转模式，亚秒计数器在达到 0x3B9A C9FF（0d999 999 999）以后重新从 0 计数。
8	ARFSEN	所有接收帧时间戳快照使能 0：不对所有接收帧使能时间戳功能 1：对所有接收帧使能时间戳功能
7:6	保留	必须保持复位值。
5	TMSARU	时间戳加数寄存器更新位 该位在更新完成后清’0’。该位在置位前必须确保读出为'0’。 0：不将时间戳加数寄存器的值更新到 PTP 模块进行精调 1：将时间戳加数寄存器的值更新到 PTP 模块进行精调
4	TMSITEN	时间戳中断触发使能 0：禁止时间戳中断 1：使能时间戳中断，当系统时间超过期望时间寄存器的值时将会产生中断。 注意：产生时间戳中断后，该位将会清’0’。
3	TMSSTU	时间戳系统时间更新位 置位该位之前，必须确保 TMSSTU 位和 TMSSTI 位读出为’0’。 0：系统时间保持不变 1：更新系统时间，在原有系统时间上加上或者减去时间戳高和低更新寄存器的值。 完成更新后，硬件将会清除该位。
2	TMSSTI	时间戳系统时间初始化位 置位该位之前，必须确保该位读出为’0’。 0：系统时间保持不变

1：初始化系统时间，将原有系统时间替换为时间戳高和低更新寄存器的值。在初始化完成后，硬件将会清除该位。

1 TMSFCU 时间戳粗调或者精调更新位
0：用粗调的方式更新系统时间戳
1：用精调的方式更新系统时间戳

0 TMSEN 时间戳使能位
0：禁止时间戳功能
1：使能接收和发送帧的时间戳功能
注意：每次设置该位为'1'后，都需要重新初始化系统时间。

表 26-10. 支持的 PTP 时间戳及其寄存器配置

CKNT (位 17:16)	0X			10		11	
MNMSEN (位 15)	X(*)	1	0	X			
ETMSEN (位 14)	0	1	1	0	1	0	1
支持的时间戳消息类型	SYNC, FOLLOW_UP, DELAY_REQ, DELAY_RESP	DELAY_REQ	SYNC	SYNC, FOLLOW_UP, DELAY_REQ, DELAY_RESP	SYNC, FOLLOW_UP	SYNC, FOLLOW_UP, DELAY_REQ, DELAY_RESP, PDELAY_REQ, PDELAY_RESP	SYNC, PDELAY_REQ, PDELAY_RESP

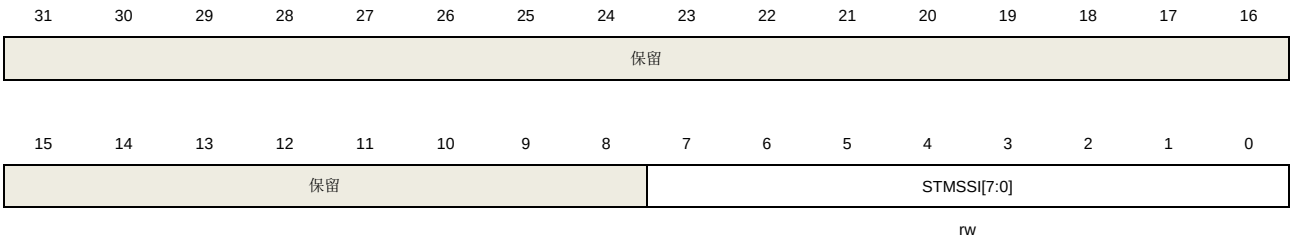
*：指无关值。

26.4.35. PTP 亚秒递增寄存器（ENET_PTP_SSINC）

地址偏移：0x0704
复位值：0x0000 0000

该寄存器用于配置亚秒递增寄存器的 8 位递增值。在粗调模式下，每个 HCLK 时钟周期，系统时间就加一次该寄存器的值。在精调模式下，在累加器溢出时，系统时间才加一次该寄存器的值。

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
31:8	保留	必须保持复位值。
7:0	STMSSI[7:0]	系统时间亚秒递增 在每次系统时间递增时，把这些位的值加到系统时间的亚秒值上。

26.4.36. PTP 时间戳高寄存器（ENET_PTP_TSH）

地址偏移：0x0708

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
STMS[31:16]															
r															

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

位/位域	名称	描述
31:0	STMS[31:0]	系统时间秒位 这些位表示了当前系统时间的秒值。

26.4.37. PTP 时间戳低寄存器（ENET_PTP_TSL）

地址偏移：0x070C

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
STS	STMSS[30:16]														
r	r														

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

位/位域	名称	描述
31	STS	系统时间符号位 0：时间值是正的 1：时间值是负的
30:0	STMSS[30:0]	系统时间亚秒位 这些位表示了当前系统时间的亚秒值，亚秒精度为 0.46ns。

26.4.38. PTP 时间戳高更新寄存器（ENET_PTP_TSUH）

地址偏移：0x0710
复位值：0x0000 0000

使用该寄存器的值对当前系统时间替换、加或减。时间戳高和低更新寄存器可以用来初始化或更新 MAC 的当前系统时间。应当先写这 2 个寄存器，再置位时间戳控制寄存器的 TMSSTI 位或 TMSSTU 位。

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
TMSUS[31:16]															
rw															

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

位/位域	名称	描述
31:0	TMSUS[31:0]	时间戳秒更新位 这些位表示的值在初始化时用于替换系统时间，在更新时表示在系统时间上加上或减去的秒值。

26.4.39. PTP 时间戳低更新寄存器（ENET_PTP_TSUL）

地址偏移：0x0714
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
TMSUPNS	TMSUSS[30:16]														
rw	rw														

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

位/位域	名称	描述
31	TMSUPNS	时间戳更新正或者负符号位 TMSSTI 位置’1’时，该位应当为’0’。 0：在系统时间上加上时间戳更新值 1：从系统时间中减去时间戳更新值
30:0	TMSUSS[30:0]	时间戳更新亚秒位 这些位表示的值在初始化时用于替换系统时间，在更新时表示在系统时间上加上或减

去的亚秒值。

26.4.40. PTP 时间戳加数寄存器（ENET_PTP_TSADDEND）

地址偏移：0x0718
复位值：0x0000 0000

该寄存器只用于系统时间更新方式为精调模式。该寄存器的值在每个时钟周期都会累加到 32 位累加器上，一旦该累加器溢出就更新系统时间。

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
TMSA[31:16]															
rw															

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

位/位域	名称	描述
31:0	TMSA[31:0]	时间戳加数 这些位用于时钟同步时加到累加器上的值，以实现时间同步。

26.4.41. PTP 期望时间高寄存器（ENET_PTP_ETH）

地址偏移：0x071C
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
ETSH[31:16]															
rw															

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

位/位域	名称	描述
31:0	ETSH[31:0]	期望时间戳高位 这些位表示了期望时间的秒值。

26.4.42. PTP 期望时间低寄存器（ENET_PTP_ETL）

地址偏移：0x0720
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
ETSL[31:16]															
rw															

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

位/位域	名称	描述
31:0	ETSL[31:0]	期望时间戳低位 这些位表示了期望时间的纳秒值。

26.4.43. PTP 时间戳标志寄存器（ENET_PTP_TSF）

地址偏移：0x0728
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														TTM	TSSCO
														ro	ro

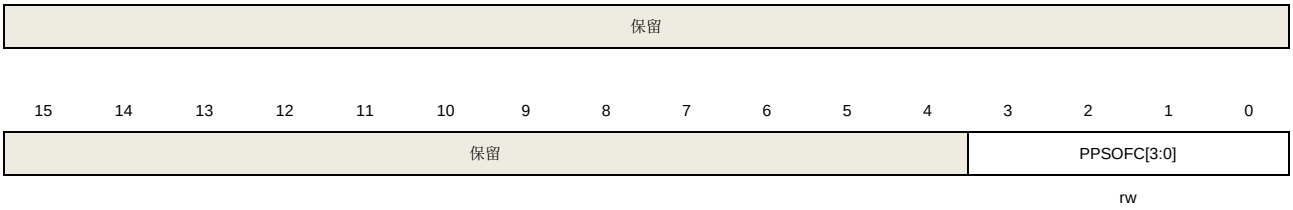
位/位域	名称	描述
31:2	保留	必须保持复位值。
1	TTM	期望时间比较位 0：系统时间小于期望时间 1：系统时间大于或等于期望时间 注意：读 ENET_PTP_TSF 寄存器将清除该位。
0	TSSCO	时间戳秒计数器上溢位 0：时间戳秒计数器没有发生上溢 1：时间戳秒计数器值大于 0xFFFF FFFF

26.4.44. PTP PPS 控制寄存器（ENET_PTP_PPSCTL）

地址偏移：0x072C
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



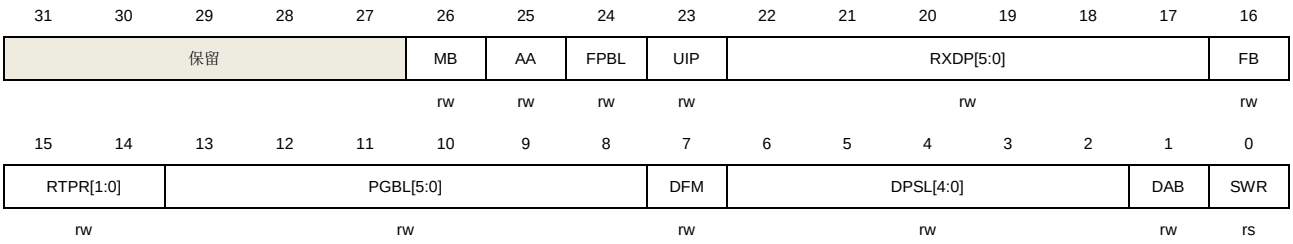
位/位域	名称	描述
31:4	保留	必须保持复位值。
3:0	PPSOFC[3:0]	PPS 输出频率配置位 0x0：1Hz（脉冲宽度：二进制回转模式下为 125ms，十进制回转模式下为 100ms） 0x1：2Hz（脉冲宽度：二进制回转模式下 50%占空比） 0x2：4Hz（脉冲宽度：二进制回转模式下 50%占空比） …. 0xF：32768（2^{15}）Hz（脉冲宽度：二进制回转模式下 50%占空比） 注意： 如果选择的是十进制回转模式，则建议仅使用 PPSOFC=0。

26.4.45. DMA 总线控制寄存器（ENET_DMA_BCTL）

地址偏移：0x1000

复位值：0x0002 0101

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
31:27	保留	必须保持复位值。
26	MB	混合传输位 0：AHB 主接口仅传输小于或等于 16 固定长度的传输 1：AHB 主接口将以 INCR 传输大于 16 长度的传输 注意：MB 和 FB 位应当且必须只有其中一位为‘1’。
25	AA	地址对齐 0：关闭传输地址对齐功能 1：使能传输地址对齐，如果 FB 位为’1’，AHB 接口对齐所有连续传输至起始地址的 LS 位（位 1 到位 0）。如果 FB 位为’0’，除第一次 AHB 访问的地址（访问数据缓存的起始地址）不对齐，后续的传输与地址均对齐。
24	FPBL	4×PGBL 模式

		0：PGBL 值（位[22:17]和位[13:8]）作为 DMA 传输长度值 1：PGBL 值（位[22:17]和位[13:8]）乘以 4 作为 DMA 传输长度值
23	UIP	使用分散 PGBL 0：PGBL 值（位[13:8]）对 DMA 接收和发送控制器都有效 1：RXDP[5:0]位用于 RxDMA 的传输长度值，PGBL[5:0]位用于 TxDMA 的传输长度值。
22:17	RXDP[5:0]	RxDMA PGBL 位 如果 UIP=0，则这些位无效。仅当 UIP=1 时，这些位定义了一次 DMA 转发的最大数据传输次数。 0x01：最大数据传输次数为 1 0x02：最大数据传输次数为 2 0x04：最大数据传输次数为 4 0x08：最大数据传输次数为 8 0x10：最大数据传输次数为 16 0x20：最大数据传输次数为 32 其他：保留。
16	FB	固定传输位 0：AHB 在连续传输时，只用 SINGLE 和 INCR 数据传输操作。 1：AHB 在连续传输时，用 SINGLE，INCR4，INCR8 和 INCR16 数据传输操作。 注意：MB 和 FB 位应当且必须只有其中一位为'1'。
15:14	RTPR[1:0]	接收发送优先级比率 这些位表示 RxDMA 和 TxDMA 之间的访问优先级比率。 0x0：RxDMA：TxDMA = 1：1 0x1：RxDMA：TxDMA = 2：1 0x2：RxDMA：TxDMA = 3：1 0x3：RxDMA：TxDMA = 4：1 注意：该位只在 DMA 仲裁模式为循环模式（DAB=0）时有效。
13:8	PGBL[5:0]	可编程的数据传输长度位 这些位定义了一次 DMA 转发的最大数据传输次数。如果 UIP=1，则这些位仅用于 TxDMA 传输。如果 UIP=0 时，则这些位同时用于 TxDMA 和 RxDMA 传输。 0x01：最大数据传输次数为 1 0x02：最大数据传输次数为 2 0x04：最大数据传输次数为 4 0x08：最大数据传输次数为 8 0x10：最大数据传输次数为 16 0x20：最大数据传输次数为 32 其他：保留。
7	DFM	描述符模式 0：常规描述符模式 1：增强描述符模式
6:2	DPSL[4:0]	描述符跳跃长度

		这些位仅对于环模式的两个描述符有效，定义了两个无链接的描述符之间从当前描述符的结尾到下一个描述符开头的地址差值，单位为字（32 位）。若 DPSL 域为 0 则 DMA 认为描述符是相邻地连续排列的。
1	DAB	DMA 仲裁位 该位指示了 TxDMA 和 RxDMA 之间的仲裁模式。 0：根据 RTPR 位的值以循环方式仲裁 1：固定模式，接收的优先级高于发送
0	SWR	软件复位 在所有时钟域的复位操作完成之后，该位将由硬件清零。 注意：在写任何 MAC 的寄存器前，应当确保该位为'0'。 0：MAC 内部寄存器正常工作 1：复位 MAC 所有内核寄存器

26.4.46. DMA 发送查询使能寄存器（ENET_DMA_TPEN）

地址偏移：0x1004
复位值：0x0000 0000

该寄存器用于 TxDMA 对发送描述符列表的查询。TxDMA 通常因为发送帧的数据下溢错误或者描述符被 CPU 占有(DAV=0)而进入暂停状态。可以对该寄存器写任意值使能发送查询。

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
TPE[31:16]															
rw_wt															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPE[15:0]															
rw_wt															

位/位域	名称	描述
31:0	TPE[31:0]	发送查询使能位 对这些位写任意值，DMA 使能发送查询，将查询当前描述符（描述符地址在 ENET_DMA_CTDADDR 寄存器中）是否被 CPU 占有。如果不是（DAV=1），则描述符可用，DMA 退出暂停状态并恢复工作。如果是（DAV=0），则 TxDMA 回到暂停状态，并把 ENET_DMA_STAT 的位 TBU 置'1'。

26.4.47. DMA 接收查询使能寄存器（ENET_DMA_RPEN）

地址偏移：0x1008
复位值：0x0000 0000

该寄存器用于 RxDMA 对接收描述符列表的查询。对该寄存器写任意值可以使能接收查询。

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
RPE[31:16]															
rw_wt															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RPE[15:0]															
rw_wt															

位/位域	名称	描述
31:0	RPE[31:0]	接收查询使能位 对这些位写任意值，DMA 使能接收查询，将查询当前描述符（描述符地址在 ENET_DMA_CRDADDR 寄存器中）是否被 CPU 占有。如果不是（DAV=1），则描述符可用，DMA 退出暂停状态并恢复工作。如果是（DAV=0），则 TxDMA 回到暂停状态，并把 ENET_DMA_STAT 的位 RBU 置'1'。

26.4.48. DMA 接收描述符列表地址寄存器（ENET_DMA_RDTADDR）

地址偏移：0x100C
复位值：0x0000 0000

接收描述符列表寄存器指向接收描述符队列的开始。描述符队列位于物理内存，并且其地址必须字对齐。只有在接收停止的时候，才允许写该寄存器。在开启 RxDMA 接收流程之前，必须正确配置该寄存器。

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
SRT[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRT[15:0]															
rw															

位/位域	名称	描述
31:0	SRT[31:0]	接收队列基址 这些位包含了接收描述符队列第一个描述符的地址。SRT[1:0]的取值默认为'0'， 因此 SRT[1:0]这两个最低位是只读的。

26.4.49. DMA 发送描述符列表地址寄存器（ENET_DMA_TDTADDR）

地址偏移：0x1010
复位值：0x0000 0000

该寄存器指向发送描述符队列的起始。描述符队列位于物理内存，并且其地址必须字对齐。只有在发送停止的时候，才允许写该寄存器。在开启 TxDMA 发送流程之前，必须正确配置该寄存器。

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
STT[31:16]															
rw															

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

位/位域	名称	描述
31:0	STT[31:0]	发送队列基址 这些位包含了发送描述符列表第一个描述符的地址。STT[1:0]的取值默认为’0’，因此 STT[1:0]这两个最低位是只读的。

26.4.50. DMA 状态寄存器（ENET_DMA_STAT）

地址偏移：0x1014
复位值：0x0000 0000

该寄存器表示 DMA 的状态位。读 ENET_DMA_STAT 寄存器并不能清除其中的标志位。对寄存器位[16:0]（除保留位）需要写’1’才能清除，而写’0’是无效的。通过设置 ENET_DMA_INTEN 寄存器里的相应位，可以屏蔽位[16:0]触发的中断。

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		TST	WUM	MSC	保留	EB[2:0]			TP[2:0]			RP[2:0]			NI
		r	r	r		r			r			r			rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AI	ER	FBE	保留		ET	RWT	RPS	RBU	RS	TU	RO	TJT	TBU	TPS	TS
rc_w1	rc_w1	rc_w1			rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位/位域	名称	描述
31:30	保留	必须保持复位值。
29	TST	时间戳触发状态 该位指示发生了一个时间戳中断事件。通过清除 TMST 标志，可以清零该位。当该位置’1’时，如果相应中断屏蔽位复位，则产生中断。 0：未发生时间戳中断事件 1：发生了时间戳中断事件
28	WUM	WUM 状态 该位指示发生了一个 WUM 事件。当两个事件触发源状态都被清除时，可以清零该位。当该位置’1’时，如果相应中断屏蔽位复位，则产生中断。 0：WUM 模块未发生中断事件 1：WUM 模块发生了中断事件

27	MSC	MSC 状态 该位指示发生了一个 MSC 事件。当所有事件触发源状态都被清除时，可以清零该位。 当该位置'1'时，如果相应中断屏蔽位复位，则产生中断。 0：MSC 模块未发生中断事件 1：MSC 模块发生了中断事件
26	保留	必须保持复位值。
25:23	EB[2:0]	错误位状态 当 FBE=1 时，这些位将对 AHB 总线上的总线响应错误进行错误类型解析。 EB[0] 1：TxDMA 传输数据时出错 0：RxDMA 传输数据时出错 EB[1] 1：读数据转发时出错 0：写数据转发时出错 EB[2] 1：访问描述符时出错 0：访问数据缓存时出错
22:20	TP[2:0]	发送流程状态 这些位表示 TxDMA 的状态。 0x0：停止，接到复位或者停止发送命令。 0x1：运行，正在取发送描述符。 0x2：运行，正在等待状态信息。 0x3：运行，正在读取内存中数据并存入 TxFIFO 中。 0x4，0x5：保留。 0x6：暂停，发送描述符不可用或发送缓存数据下溢。 0x7：运行，正在关闭发送描述符。
19:17	RP[2:0]	接收流程状态 这些位表示 RxDMA 的状态。 0x0：停止，接到复位或者停止接收命令。 0x1：运行，正在取接收描述符。 0x2：保留。 0x3：运行，正在等待接收数据包。 0x4：暂停，接收描述符不可用。 0x5：运行，正在关闭接收描述符。 0x6：保留。 0x7：运行，正在把接收到数据包从 RxFIFO 转发到内存中。
16	NI	正常中断汇总 该位是下列位在相应中断使能位（ENET_DMA_INTEN 寄存器）使能了的情况下，其各取值的逻辑或： TS：发送中断 TBU：发送缓存不可用 RS：接收中断

		ER：提前接收中断 注意：该位置'1'后，只有把造成该位置'1'的位清'0'（写'1'），才能把该位清'0'。
15	AI	异常中断汇总 该位是下列位在相应中断使能位（ENET_DMA_INTEN 寄存器）使能了的情况下，其各取值的逻辑或： TPS：发送流程停止 TJT：发送 Jabber 超时 RO：RxFIFO 上溢 TU：发送数据下溢 RBU：接收缓存不可用 RPS：接收流程停止 RWT：接收看门狗超时 ET：提前发送中断 FBE：总线致命错误 注意：该位置'1'后，只有把造成该位置'1'的位清'0'（写'1'），才能把该位清'0'。
14	ER	提前接收状态 在接收中断位 RS 置'1'时，该位自动清'0'。 0：未接收到帧数据 1：接收到的数据帧已由 DMA 填满了第一个缓存
13	FBE	总线致命错误状态 该位指示发生了一个 AHB 接口响应错误，其错误类型可以由 EB[2:0]位进行解释。 0：未发生总线错误 1：发生了总线错误，相应的 DMA 控制器停止所有操作。
12:11	保留	必须保持复位值。
10	ET	提前发送状态 0：发送的帧还未完全传输到 TxFIFO 中 1：发送的帧已经完全传输到 TxFIFO 中
9	RWT	接收看门狗超时状态 0：接收到的帧长度小于 2048 字节 1：接收到的帧长度超过 2048 字节
8	RPS	接收流程停止状态 0：接收流程未停止 1：接收流程进入停止状态
7	RBU	接收缓存不可用状态 0：下一个接收描述符的 DAV 位为'1' 1：下一个接收描述符的 DAV 位为'0'，RxDMA 进入暂停状态。
6	RS	接收状态 0：帧接收未完成 1：帧接收完成

5	TU	发送数据下溢状态 0：未发生发送数据下溢错误 1：发送帧的过程中发生数据下溢，同时发送进入暂停状态。
4	RO	接收上溢状态 0：未发生接收数据上溢错误 1：接收帧的过程中发生上溢错误。如果已有一部分帧数据转发到内存，则设置接收描述符 0 的上溢错误位 OERR 为'1'。
3	TJT	发送 Jabber 超时状态 0：未发生发送 Jabber 定时器超时事件 1：发送 Jabber 定时器超时。此时中止发送进程并进入停止状态，同时设置发送描述符 0 的 Jabber 超时位 JT 为'1'。
2	TBU	发送缓存不可用状态 0：下一个发送描述符的 DAV 位为'1' 1：下一个发送描述符的 DAV 位为'0'，TxDMA 进入暂停状态。
1	TPS	发送流程停止状态 0：发送未停止 1：发送停止
0	TS	发送状态 该位仅在发送描述符 0 中 LSG 和 INTC 位都置位时，才可被置位。 0：当前帧发送未完成 1：当前帧发送完成

26.4.51. DMA 控制寄存器（ENET_DMA_CTL）

地址偏移：0x1018
复位值：0x0000 0000

该寄存器设定了接收和发送的工作模式和命令。在整个 DMA 的初始化流程中，应当最后写该寄存器。

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					DTCERFD	RSFD	DAFRF	保留		TSFD	FTF	保留			TTHC[2]
					rw	rw	rw			rw	rs				rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTHC[1:0]		STE	保留					FERF	FUF	保留	RTHC[1:0]		OSF	SRE	保留
rw		rw						rw	rw		rw		rw	rw	

位/位域	名称	描述
31:27	保留	必须保持复位值。

26	DTCERFD	不丢弃 TCP/IP 校验和错误帧 0：当 FERF 位为'0'，MAC 丢弃所有有错的帧。 1：当 FERF 位为'1'，接收到帧仅有校验和错误时，MAC 不会丢弃该帧。
25	RSFD	接收存储转发 0：RxFIFO 工作在直通模式，转发阈值由 RTHC 位决定。 1：RxFIFO 工作在存储转发模式，只有在帧完整写入 RxFIFO 后，RxDMA 才会把它转发给应用程序，此时 RTHC 位的取值会被忽略。
24	DAFRF	不清空接收帧 0：当接收描述符不可用时，RxDMA 就清空 RxFIFO 里的接收帧。 1：RxDMA 不清空接收帧，即使接收描述符不可用。
23:22	保留	必须保持复位值。
21	TSFD	发送存储转发 0：TxFIFO 工作在直通模式，发送阈值由 ENET_DMA_CTL 寄存器中的 TTHC 位决定。 1：TxFIFO 工作在存储转发模式，只有在帧完整写入 TxFIFO 后，MAC 才会把其发送出去，TTHC 位的取值会被忽略。 注意：在发送处于停止状态时，可以修改该位。
20	FTF	清空 TxFIFO 当此位为 1 时，TxFIFO 控制逻辑电路被复位到初始状态，TxFIFO 里所有的数据被清空/丢失。在清空操作完成后该位被自动清'0'。 注意：在该位为'0'之前，不允许写 ENET_DMA_CTL 寄存器。
19:17	保留	必须保持复位值。
16:14	TTHC[2:0]	发送阈值控制 这三位控制直通模式下 TxFIFO 的阈值。 当 TSFD=1 时，忽略这些位。 0x0：64 0x1：128 0x2：192 0x3：256 0x4：40 0x5：32 0x6：24 0x7：16
13	STE	开始/停止发送 0：在发送完当前帧或 TxDMA 进入暂停状态后，发送进程进入停止模式。保存发送描述符队列里下一发送描述符的位置，在传输重新开始时，这个描述符就变成当前描述符。 1：TxDMA 进入运行状态。DMA 获取当前发送描述符，发送帧描述符可从 ENET_DMA_TDTADDR 基址获取，若上一次发送为停止状态，则也可从发送描述符队列的指针位置获取。如果当前描述符的 DAV 位为'0'，则 TxDMA 进入暂停状态，

		并设置 TBU 位为'1'。如果在未设置完其他 DMA 寄存器的情况下就置位该位，则会引起不可预料的后果。
12:8	保留	必须保持复位值。
7	FERF	转发错误帧 0：当 RxFIFO 工作于直通模式（RSFD=0）时，如果在将 RxFIFO 数据转发到内存之前检测到了帧错误（CRC 错误、冲突错误、校验和错误、看门狗超时、溢出），则 RxFIFO 会丢弃这个错误的帧。但如果在将 RxFIFO 数据转发到内存之后才检测到了帧错误，则就不会丢弃该帧。当 RxFIFO 工作于存储转发模式时，在接收过程中一旦检测到帧错误，就会丢弃该帧。 1：除了过短帧外的所有帧都会转发给 DMA
6	FUF	转发长度不够的"好"帧 0：RxFIFO 丢弃所有长度小于 64 字节的帧，但如果在检测到过短帧之前，帧已开始转发该帧给应用程序（例如在直通模式下，帧长小于接收阈值），则将转发整个帧。 1：RxFIFO 把长度不够的"好"帧（帧长小于 64 字节但没有错误）转发给应用程序
5	保留	必须保持复位值。
4:3	RTHC[1:0]	接收阈值控制 这两位设置了在直通模式下 RxFIFO 的阈值。 注意：只有在 RSFD 位（位 21）为'0'时，这些位才有效。在 RSFD 位为'1'时忽略这些位。 0x0：64 0x1：32 0x2：96 0x3：128
2	OSF	操作第二帧 0：TxDMA 仅在接收到前一个帧的发送状态信息后，才开始发送下一个帧的数据。 1：TxDMA 在前一帧数据全部存入到 TxFIFO 之后，在接收到前一个帧的发送状态信息前，就开始发送下一个帧的数据。
1	SRE	开始/停止接收 0：在转发完当前接收帧后，RxDMA 进入停止模式。保存接收描述符队列里下一接收描述符的位置，在传输重新开始时，这个描述符就变成当前描述符。只有在接收运行时或接收暂停时，可"停止接收"。 1：把接收进程置为运行状态，DMA 检查接收描述符队列的当前位置，用来处理下一个收到的帧。接收帧描述符可从 ENET_DMA_RDTADDR 基址获取，若上一次接收为停止状态，则也可从接收描述符队列的指针位置获取。如果获取的描述符 DAV=0，那么接收进程进入暂停状态，并设置 RBU 位为'1'。只有在接收停止时或接收暂停时，"开始接收"命令才有效。在未设置完所有其他 DMA 寄存器之前发出"开始接收"命令，会引起不可预料的后果。
0	保留	必须保持复位值。

26.4.52. DMA 中断使能寄存器（ENET_DMA_INTEN）

地址偏移：0x101C
复位值：0x0000 0000

该寄存器可以使能 ENET_DMA_STAT 寄存器反映的中断。

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															NIE
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AIE	ERIE	FBEIE	保留		ETIE	RWTIE	RPSIE	RBUIE	RIEN	TUIE	ROIE	TJTIE	TBUIE	TPSIE	TIE
rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:17	保留	必须保持复位值。
16	NIE	正常中断汇总使能 0：屏蔽正常中断 1：使能正常中断 该位使能下列位： TS：发送中断 TBU：发送缓存不可用 RS：接收中断 ER：提前接收中断
15	AIE	异常中断汇总使能 0：屏蔽异常中断 1：使能异常中断 该位使能下列位： TPS：发送流程停止 TJT：发送 Jabber 超时 RO：RxFIFO 上溢 TU：发送下溢 RBU：接收缓存不可用 RPS：接收流程停止 RWT：接收看门狗超时 ET：提前发送中断 FBE：总线致命错误
14	ERIE	提前接收中断使能 0：屏蔽提前接收中断 1：使能早接收中断
13	FBEIE	总线致命错误中断使能 0：屏蔽总线致命错误中断

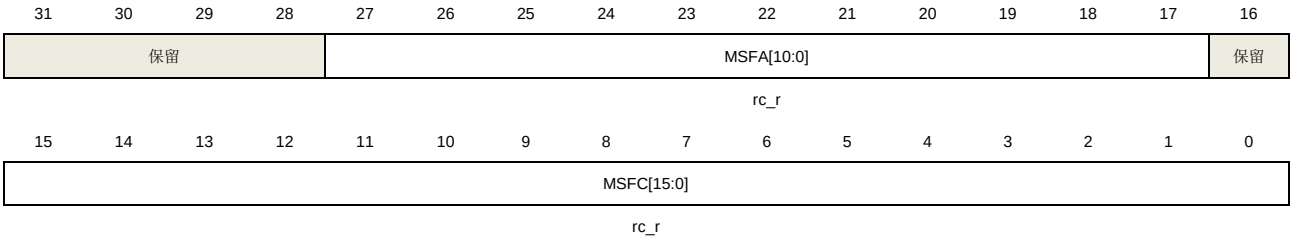
		1：使能总线致命错误中断
12:11	保留	必须保持复位值。
10	ETIE	提前发送中断使能 0：屏蔽提前发送中断 1：使能提前发送中断
9	RWTIE	接收看门狗超时中断使能 0：屏蔽接收看门狗超时中断 1：使能接收看门狗超时中断
8	RPSIE	接收流程停止中断使能 0：屏蔽接收流程停止中断 1：使能接收流程停止中断
7	RBUIE	接收缓存不可用中断使能 0：屏蔽接收缓存不可用中断 1：使能接收缓存不可用中断
6	RIE	接收中断使能 0：屏蔽接收中断 1：使能接收中断
5	TUIE	发送下溢中断使能 0：屏蔽发送数据下溢中断 1：使能发送下溢中断
4	ROIE	接收上溢中断使能 0：屏蔽接收上溢中断 1：使能接收上溢中断
3	TJTIE	发送 Jabber 超时中断使能 0：屏蔽发送 Jabber 超时中断 1：使能发送 Jabber 超时中断
2	TBUIE	发送缓存不可用中断使能 0：屏蔽发送缓存不可用中断 1：使能发送缓存不可用中断
1	TPSIE	发送流程停止中断使能 0：屏蔽发送流程停止中断 1：使能发送流程停止中断
0	TIE	发送中断使能 0：屏蔽发送中断 1：使能发送中断

26.4.53. DMA 丢失帧和缓存溢出计数器寄存器（ENET_DMA_MFBOCNT）

地址偏移：0x1020
复位值：0x0000 0000

DMA 有 2 个计数器，用来统计接收过程中丢失帧的数目。可通过读本寄存器来获取计数器的当前值。这个计数器通常用作故障诊断。

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。



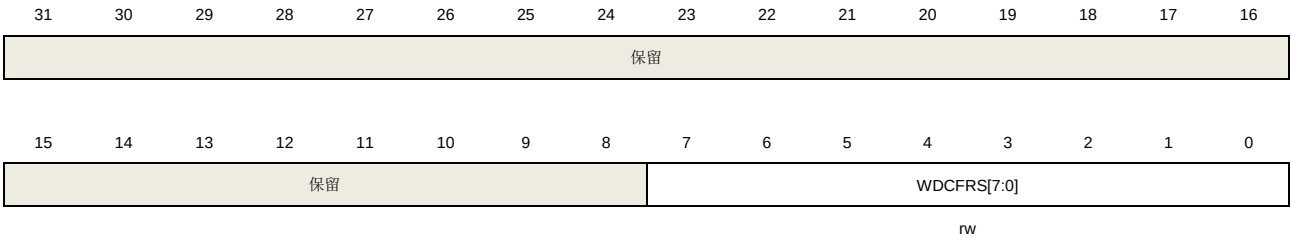
位/位域	名称	描述
31:28	保留	必须保持复位值。
27:17	MSFA[10:0]	应用程序丢失的帧 这些位指示了 RxFIFO 丢失的帧数目。
16	保留	必须保持复位值。
15:0	MSFC[15:0]	控制器丢失的帧 这些位表示了由于 MCU 的接收缓存不可用而导致 RxDMA 丢失的帧的数目。每当 DMA 清空一个输入帧时，这个计数器加 1。

26.4.54. DMA 接收状态看门狗计数器寄存器（ENET_DMA_RSWDC）

地址偏移：0x1024
复位值：0x0000 0000

向该寄存器写入一个值，可在延时一段可配置的时间之后，使能针对 RS 位（ENET_DMA_STAT 寄存器）的看门狗定时器。

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
31:8	保留	必须保持复位值。

7:0	WDCFRS[7:0]	接收状态看门狗计数器 这些位仅在接收描述符 1 的 DINTC 位置位时有效。当 DINTC=1 时，并接收到一个帧，则 RS 位会在接收完毕后延时 WDCFRS*256 个 HCLK 时钟周期再置位。

26.4.55. DMA 当前发送描述符地址寄存器（ENET_DMA_CTDADDR）

地址偏移：0x1048
复位值：0x0000 0000

当前发送描述符寄存器指向 TxDMA 正在读取的发送描述符起始地址。

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
TDAP[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDAP[15:0]															
r															

位/位域	名称	描述
31:0	TDAP[31:0]	发送描述符地址指针 这些位在复位时清'0'，由 TxDMA 在操作过程中自动更新。

26.4.56. DMA 当前接收描述符地址寄存器（ENET_DMA_CRDADDR）

地址偏移：0x104C
复位值：0x0000 0000

当前接收描述符寄存器指向 RxDMA 正在读取的接收描述符起始地址。

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
RDAP[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDAP[15:0]															
r															

位/位域	名称	描述
31:0	RDAP[31:0]	接收描述符地址指针 这些位在复位时清'0'，由 RxDMA 在操作过程中自动更新。

26.4.57. DMA 当前发送缓存地址寄存器（ENET_DMA_CTBADDR）

地址偏移：0x1050
复位值：0x0000 0000

该寄存器指向 TxDMA 正在读取的发送缓存的地址。

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
TBAP[31:16]															
r															

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

位/位域	名称	描述
31:0	TBAP[31:0]	发送缓存地址指针 这些位在复位时清'0'，由 TxDMA 在工作过程中更新。

26.4.58. DMA 当前接收缓存地址寄存器（ENET_DMA_CRBADDR）

地址偏移：0x1054
复位值：0x0000 0000

该寄存器地址指向 RxDMA 正在读取的接收缓存地址。

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
RBAP[31:16]															
r															

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

位/位域	名称	描述
31:0	RBAP[31:0]	接收缓存地址指针 这些位在复位时清'0'，由 RxDMA 在工作过程中更新。

27. 通用串行总线高速接口（USBHS）

27.1. 概述

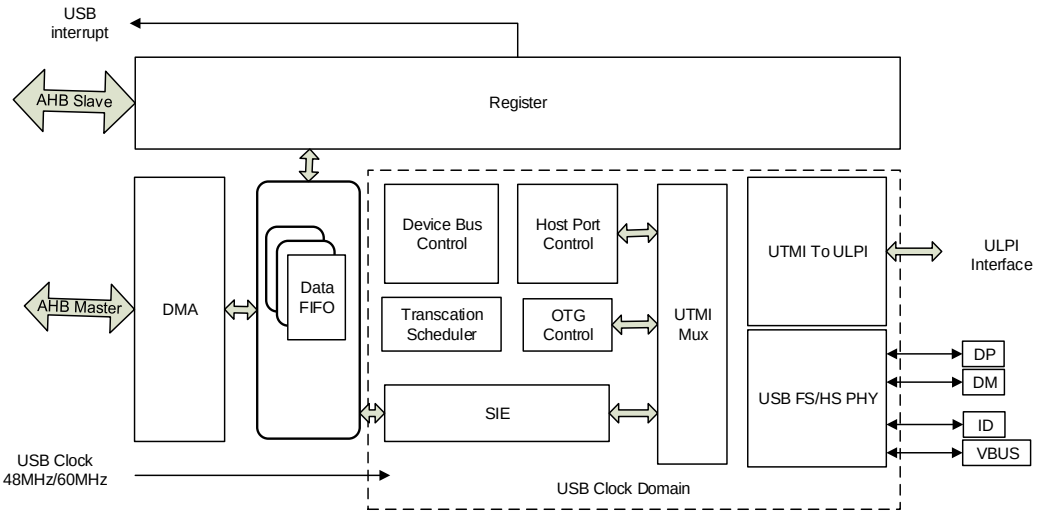
USB 高速（USBHS）控制器为便携式设备提供了一套 USB 互联解决方案。USBHS 不仅支持主机模式和设备模式，也支持遵循 HNP（主机协商协议）和 SRP（会话请求协议）的 OTG 模式。USBHS 包含了一个内部的 USB PHY，可以配置成全速或高速，并且不再需要外部 PHY 芯片。USBHS 可以支持 USB 2.0 协议所定义的所有四种传输方式（控制传输、批量传输、中断传输和同步传输）。另外，在 USBHS 内部还有一个 DMA 引擎操作，可作为 AHB 总线主机在 USBHS 和系统之间加速数据传输。对于全速设备的操作，还支持电池充电检测（BCD）、附加检测协议（ADP）和链路层电源管理（LPM）。

27.2. 主要特性

- 支持USB 2.0高速（480Mb/s）/全速（12Mb/s）/低速（1.5Mb/s）主机模式；
- 支持USB 2.0高速（480Mb/s）/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；
- 支持高带宽中断和同步传输；
- 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4KB的FIFO RAM；
- 在主机模式下，支持12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6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）；
- 在主机模式下，若在高速模式下操作，支持PING协议；
- 在设备模式下，支持6个OUT端点和6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 OTG协议的USB PHY；
- 包含一个内部DMA调度器和引擎，每个应用请求都可在USBHS和系统之间执行数据拷贝；
- 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；
- 支持1.2版电池充电规范中描述的电池充电检测（BCD）；
- 支持2.0版USB OTG补充协议中描述的附加检测协议（ADP）
- 支持USB 2.0链路层电源管理附录和USB2.0工程变更通知单勘误表中描述的链路电源管理（LPM）。

27.3. 结构框图

图 27-1. USBHS 结构框图



27.4. 信号线描述

表 27-1. USBHS 信号线描述

I/0 端口	类型	描述	注意
VBUS	输入	总线电源端口	仅内部 PHY 使用
DM	输入/输出	差分信号线-端口	仅内部 PHY 使用
DP	输入/输出	差分信号线+端口	仅内部 PHY 使用
ID	输入	USB 识别：微连接器识别接口	仅内部 PHY 使用
ULPI_D[7:0]	输入/输出	ULPI 数据线	外部 ULPI PHY 使用
ULPI_NXT	输入	ULPI 下个信号线	外部 ULPI PHY 使用
ULPI_DIR	输入	ULPI 方向	外部 ULPI PHY 使用
ULPI_STP	输出	ULPI 停止	外部 ULPI PHY 使用
ULPI_CLK	输入	ULPI 时钟	外部 ULPI PHY 使用

27.5. 功能描述

27.5.1. USBHS PHY 选择、时钟及工作模式

USBHS 可以作为一个主机、一个设备或者一个 DRD（双角色设备），并且支持两种连接类型：内部嵌入式 PHY 和外部 ULPI PHY。根据用户需求，应用可以选择两种连接类型的任何一种。

应用可以在主机模式下使用 USBHS_HCTL 寄存器内的 SPDFSLS 控制位和在设备模式下使用 USBHS_DCFG 寄存器内的 DS[1:0]控制位将内部 PHY 和外部 ULPI PHY 的最大速度限制

至全速。

表 27-2. USBHS 支持速度列表

寄存器配置		主机支持速度	设备支持速度
EMBPHY_FS=1 EMBPHY_HS=0 （内部FS PHY）		全速 低速	全速
EMBPHY_FS=0 EMBPHY_HS=1 （内部 HS PHY）	DS =01（设备模式） SPDFSLS=1（主机模式）	全速 低速	全速
	DS =00（设备模式） SPDFSLS=0（主机模式）	高速 全速 低速	高速 全速
EMBPHY_FS=0 EMBPHY_HS=0 （外部 ULPI PHY）	DS = 01（设备模式） SPDFSLS = 1（主机模式）	全速 低速	全速
	DS = 00（设备模式） SPDFSLS = 0（主机模式）	高速 全速 低速	高速 全速

应用可以使用 USBHS_GUSBCS 寄存器中的 FHM 和 FDM 控制位选择 USBHS 的工作模式：主机模式(FHM=1)或设备模式(FDM=1)。当这两个控制位被清除时，USBHS 工作在 OTG 模式，即系统复位后的默认模式。

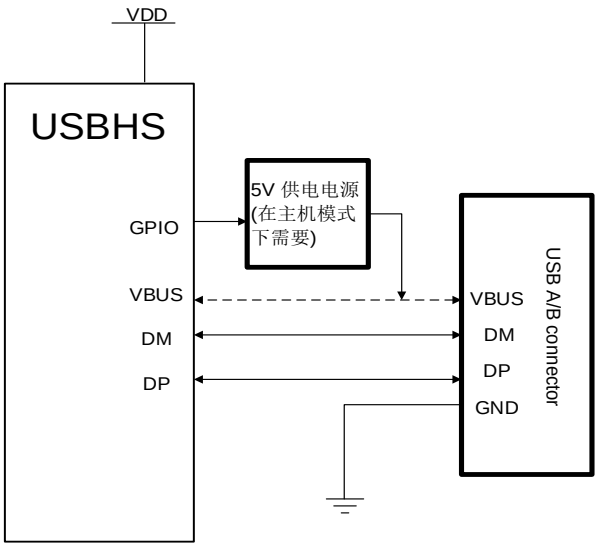
内部嵌入式 PHY

USBHS 包含一个内部嵌入式 PHY，该内部嵌入式 PHY 支持主机模式下的高速、全速和低速、设备模式下高速和全速，以及具备 HNP 和 SRP 的 OTG 协议。软件需要置位 USBHS_GUSBCS 寄存器中的 EMBPHY_FS 控制位并清除 USBHS_GUSBCS 寄存器中的 EMBPHY_HS 控制位来使用该内部嵌入式 PHY 的全速模式；或者清除 EMBPHY_FS 控制位并置位 EMBPHY_HS 控制位来使用该内部嵌入式 PHY 的高速式。如果内部全速 PHY 被选择，USBHS 在全速模式下所使用的 USB 时钟需要配置为 48MHz，在高速模式需要配置为 60MHz。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，而 60MHz USB 时钟由 480MHz PLLUSB 产生。

注意：在配置 480MHz 时钟的过程中，不建议在寄存器 RCU_ADDCFG 的位 PLLUSBPRESEL 选择 CK_IRC48M.

上拉或下拉电阻已经集成在内部全速 PHY 的内部，并且 USBHS 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 PHY 的典型连接示意图如*图 27-2. 在主机或设备模式下连接示意图*所示。

图 27-2. 在主机或设备模式下连接示意图

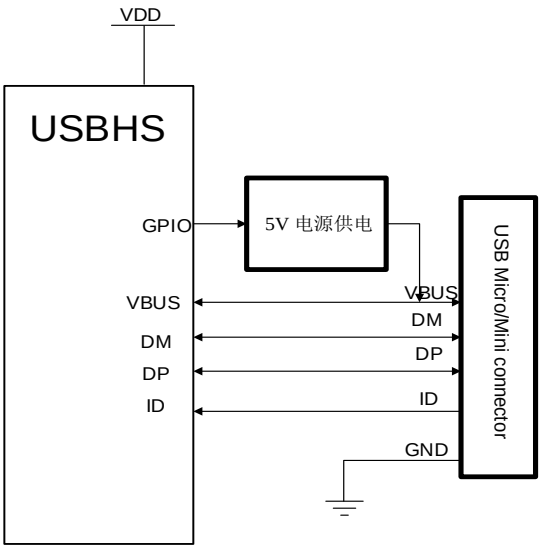


当 USBHS 工作在主机模式下时（FHM 控制位置位、FDM 控制位清除），VBUS 为 USB 协议所定义的 5V 电源检测引脚。内部 PHY 不能提供 5V VBUS 电源，仅在 VBUS 信号线上具有电压比较器和充电、放电电路。所以，如果应用需要提供 VBUS 电源，那么则需要一个外部的供电电源 IC。在主机模式下，USBHS 和 USB 连接头之间的 VUBS 连接可以被忽略，这是由于 USBHS 并不检测 VBUS 引脚的电平状态，并假定 5V 供电电源一直存在。

当 USBHS 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 USBHS_GCCFG 寄存器中的 VDEN 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 VDEN 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 USBHS 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 27-3. OTG 模式下使用内部嵌入式 PHY 连接示意图*所示。当 USBHS 工作在 OTG 模式下时，USBHS_GUSBCS 寄存器内的 FHM、FDM 控制位和 USBHS_GCCFG 寄存器的 VDEN 位都应该被清除。在这种模式下，USBHS 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。USBHS 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，USBHS 控制上拉和下拉电阻。

图 27-3. OTG 模式下使用内部嵌入式 PHY 连接示意图

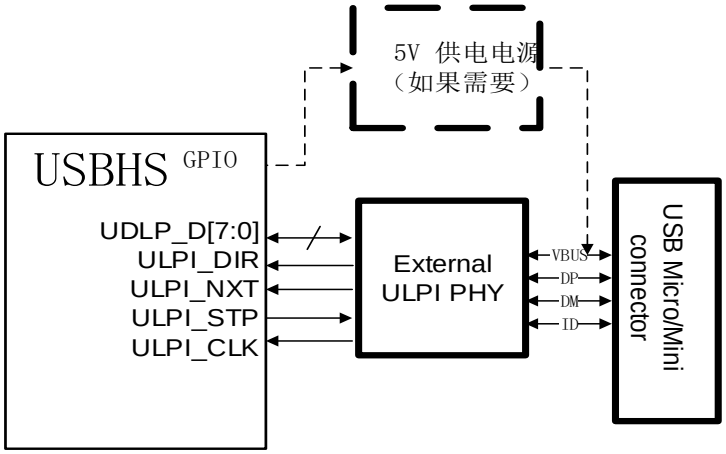


外部 ULPI PHY

USBHS 为外部 PHY 提供了一个 ULPI 接口。如果需要使用 USBHS 模块完成高速 USB 应用，那么则需要一个外部高速 ULPI PHY。结合外部 ULPI PHY，USBHS 支持高速主机和设备，也支持前文中内部嵌入式全速 PHY 所描述的所有模式。

软件需要清除 USBHS_GUSBCS 寄存器中的 EMBPHY_FS 和 EMBPHY_HS 控制位以使能 ULPI 接口。当 ULPI 模式能使，USB 时钟需要配置到 60MHz，并且需要从 ULPI_CLK 引脚引入。软件可以在 RCU 模块中打开或关闭该 60MHz ULPI 时钟。

图 27-4. 使用外部 ULPI PHY 的连接示意图

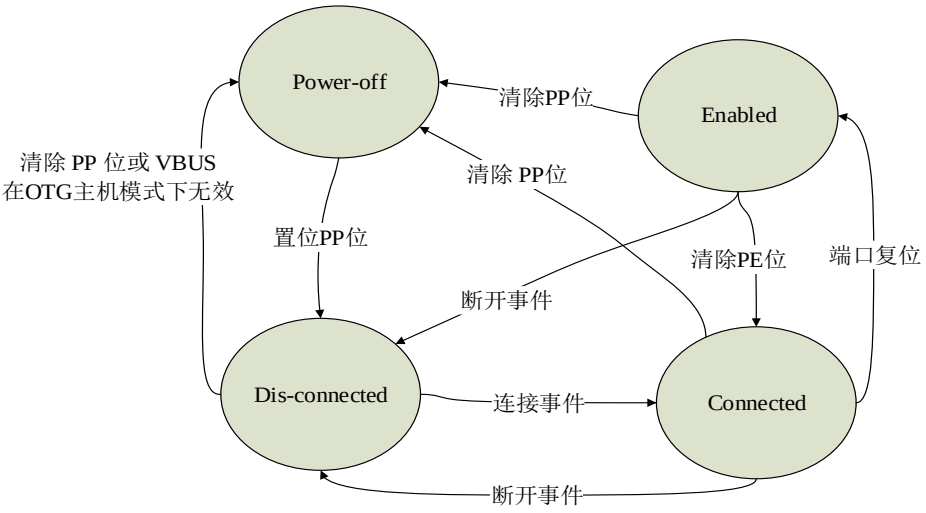


27.5.2. USB 主机功能

USB 主机端口状态

主机应用可以通过 USBHS_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，USB PHY（内部或外部）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 27-5. 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，USBHS 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。

USBHS 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 USBHS_HPCS 寄存器的 PS 位域中。

如果最大支持速度被配置为全速（SPDFSLS=1），USBHS 仅仅在设备连接的过程中执行速度识别，并且从 DM 或 DP 的电平状态决定设备速度。就像 USB 协议中所描述的那样，全速设备上拉 DP 信号线，而低速设备上拉 DM 信号线。

如果最大支持速度被配置为高速（SPDFSLS=0），USBHS 首先在连接的过程中执行速度检测，如果检测到全速设备连接，USBHS 会在连接事件后的每个 USB 复位序列中，尝试执行高速检测（USB2.0 协议中所描述的 CHIRP 序列）。所以，在主机上的应用应该在一个连接事件后提供一个 USB 复位，并且再次检查 PS[1:0]标志位，以确定其连接的是否为高速设备。

挂起和复位

USBHS 支持挂起和复位状态，当 USBHS 端口在使能状态时，向 USBHS_HPCS 寄存器的 PSP 控制位写 1，USBHS 会进入到挂起状态。在挂起状态下，USBHS 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

USBHS_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 USBHS_GINTF 寄存器的 WKUPIF 标志位，并且触发 USBHS 唤醒中断。

SOF 产生器

在主机模式下，USBHS 向 USB 总线发送 SOF 令牌包。如 USB2.0 协议所描述，全速连接下，SOF 令牌包每 1ms 产生一次(由主机控制器或者 HUB 事务转换器产生)；高速连接下，SOF 令牌包将在接下来的七个 125 μs 周期后产生。

每当 USBHS 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 USBHS_HFT 寄存器中的 FRI[15:0]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 USBHS 所使用的 USB 时钟频率计算该值。FRT[14:0]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。

USBHS 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 USBHS_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

USBHS 在主机模式下包含 12 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 USBHS_HCHxCTL 和 USBHS_HCHxLEN 寄存器。

USBHS 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，USBHS 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。

在无 DMA 模式下，如果应用想要在 USB 总线上启动一个 OUT 事务，应用需要通过 AHB 寄存器接口向数据 FIFO 中写入数据包。USBHS 硬件会在应用写完整包数据后，自动产生一个事务请求并进入请求队列。在 DMA 模式下，应用仅需要配置通道属性和通道数据缓冲区地址，USBHS 内部的 DMA 引擎会执行数据包拷贝和请求条目的产生工作。当应用使能 IN 通道时，USBHS 自动产生 IN 请求条目。

请求队列中的请求条目通过事务控制模块按顺序处理。USBHS 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，USBHS 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 USBHS 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，USBHS 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，USBHS 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，USBHS 会停止处理任何队列，并等待直到当前帧结束。

LPM

USBHS 模块添加了电源管理状态（LPM 状态）和机制，这种机制影响主机和集线器用于有效管理总线和系统电源的状态更改。LPM 只是添加了一个新特性和总线状态休眠状态（L1），它与 USB2.0 定义的 suspend（L2）/resume 共存。

L1 类似于 L2，但是使用起来比 L2 更加细致。进入到转换成 L1 是通过对集线器或主机端口的请求启动的。LPM 事务被发送到下游设备，该事务请求的转换只能在设备响应了 ACK 握手时才发生。通过远程唤醒、恢复信令、重置信令或断开连接从 L1 退出。主机或设备可以在 L1 中启动恢复信令。尽管 resume 的信号等级与 L2 相同，但与 L1、L0（活动状态）转换相关的信号和过渡延迟的持续时间要短得多。

27.5.3. USB 设备功能

USB 设备连接

在设备模式下，USBHS 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 USBHS_GCCFG 寄存器中 VDEN 控制位，USBHS 将进入供电状态。USBHS 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，USBHS 会为软件触发一个复位中断。

如果最大支持速度被配置为全速（USBHS_DCFG 寄存器内 DS[1:0] = 01），USBHS 会以全速设备操作，然而如果最大支持速度被配置为高速（USBHS_DCFG 寄存器内 DS[1:0] = 00），在复位序列中，USBHS 设备会尝试和主机启动一个速度识别（USB2.0 协议中描述的一个 CHIRP 序列）。如果和主机的 CHIRP 序列握手成功，设备将会进入高速模式，否则，仍然停留在全速模式。

在复位序列和速度识别过程完成后，USBHS 将会触发 USBHS_GINTF 寄存器中的 ENUMF 标志/中断，并且利用 USBHS_DSTAT 寄存器内的 ES 标志位反映当前枚举设备速度。所以，如果软件想要实现一个高速设备，必须等待 ENUMF 中断，然后读取 ES[1:0]控制位以获得速度识别结果。

如 USB2.0 协议所需要，USBHS 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。USBHS 检测到恢复信号后，将置位 USBHS_GINTF 寄存器的 WKUPIF 标志位并且触发 USBHS 唤醒中断。

在挂起设备模式，USBHS 也能够远程唤醒 USB 总线。软件可以通过置位 USBHS_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

USBHS 支持软件断开。设备进入到供电状态后，USBHS 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 USBHS_DCTL 寄存器中 SD 控制位进行强制断开。SD 控制位置位后，如果当前设备速度为高速，USBHS 会首先返回到全速设备，然后关闭 DP 信号线上的上拉电阻；如果当前设备速度为全速，USBHS 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 USBHS 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 USBHS_DSTAT 寄存器的 FNRSOF[13:0]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），USBHS 会触发 USBHS_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。

BCD

支持第 1.2 版电池充电规范中描述的充电端口检测（BCD）。为了使 PD（便携式设备）确定允许从上游 USB 端口吸取多少电流，需要 PD 有区分标准下游端口和充电端口的机制。

在 BCD 机制中，包括 USB VBUS 检测（VD）、数据接触检测（DCD）、主检测（PD）和次检测（SD）。关于 BCD 的控制和配置位在 USBHS_GCCFG 寄存器中描述。

27.5.4. OTG 功能概述

USBHS 支持 OTG 协议 1.3/2.0 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。USBHS 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 USBHS_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 OTG1.3/2.0 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 USBHS 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 USBHS_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 USBHS_GOTGCS 寄存器的 HNPREQ 控制位，然后 USBHS 会开始在总线上执行 HNP 协议，最后，HNP 的结果会反馈在 USBHS_GOTGCS 寄

存器的 HNPS 状态位。另外，软件总能从 USBHS_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 USBHS_GOTGCS 寄存器的 ASV 和 BSV 状态位中。

当 USBHS 工作在 B 设备 OTG 模式时，软件可以通过置位 USBHS_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，USBHS 会在 USBHS_GOTGCS 寄存器中产生一个成功标志位 SRPS。

当 USBHS 工作在 OTG A 设备模式且从 B 设备检测到一个 SRP 请求时，USBHS 将会置位 USBHS_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。

ADP

附加检测协议（ADP）是一种允许本地设备检测远程设备何时被连接或分离的协议。远程设备可以是任何 USB 设备。ADP 通过检测两个设备连接或分离时 VBUS 电容的变化来工作。电容的检测方法是先对 VBUS 线放电，然后用已知的电流源测量 VBUS 充电到已知电压所需的时间。通过寻找充电时间的变化来检测电容的变化。

软件可以设置 ADPMEN、ADPEN 和 ENAPRB 位来执行 ADP 探测，并且应至少执行一个 ADP 探测周期，以便在首次启动具有 ADP 功能的 A 设备或 B 设备时获得 TADP_RISE 的初始值。对于 B 设备，可以通过设置位 ENASNS 来执行 ADP sense。如果 USBHS_ADPCTL 寄存器中的 RITM 发生变化，则表明远程设备已连接或分离。

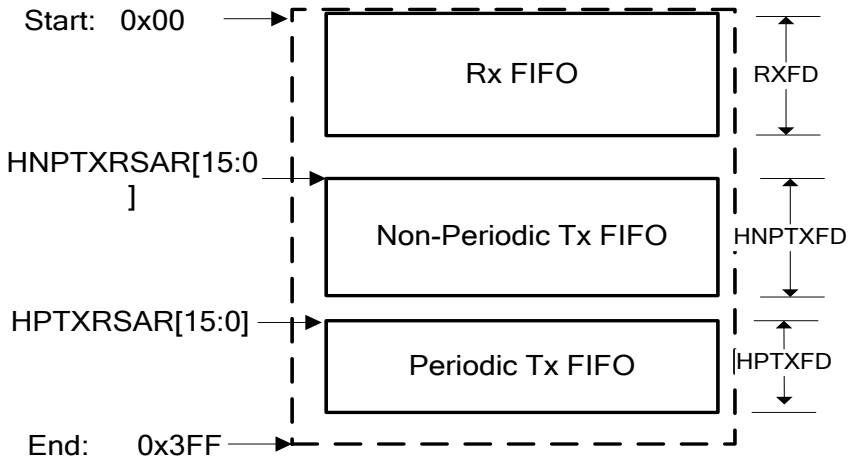
27.5.5. 数据 FIFO

USBHS 中采用 4K 字节数据 FIFO 存储包数据，数据 FIFO 是通过 USBHS 的内部 SRAM 实现的。

主机模式

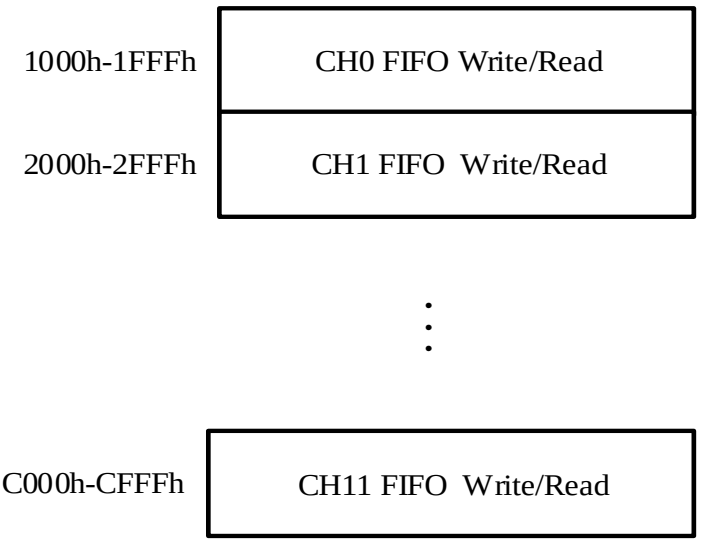
主机模式下，数据 FIFO 空间分为三个部分，分别是：用于接收数据包的 Rx FIFO、用于非周期性发送数据包的非周期性 Tx FIFO 和用于周期性发送数据包的周期性 Tx FIFO。所有的 IN 通道通过共享 Rx FIFO 接收数据。所有的周期性 OUT 通道通过共享周期性 Tx FIFO 来发送数据，所有的非周期性 OUT 通道通过共享非周期性 Tx FIFO 来发送数据。通过寄存器 USBHS_GRFLEN、USBHS_HNPTFLEN 和 USBHS_HPTFLEN，软件可以配置以上数据 FIFO 的大小和起始偏移地址。*图 27-6. 主机模式 FIFO 空间*所描述的是 SRAM 中各 FIFO 的结构，图中的数值是按照 32 位为单位写的。

图 27-6. 主机模式 FIFO 空间



在 DMA 模式下，DMA 负责系统存储区和数据 FIFO 之间的数据包传输。在非 DMA 模式下，程序将包数据写入数据 FIFO 或从数据 FIFO 读取包数据。USBHS 为程序提供了专有寄存器空间来读写数据 FIFO。*图 27-7. 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 USBHS 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 USBHS_GRXTATR/USBHS_GRSTATP 来访问数据包所从属的 Rx FIFO。

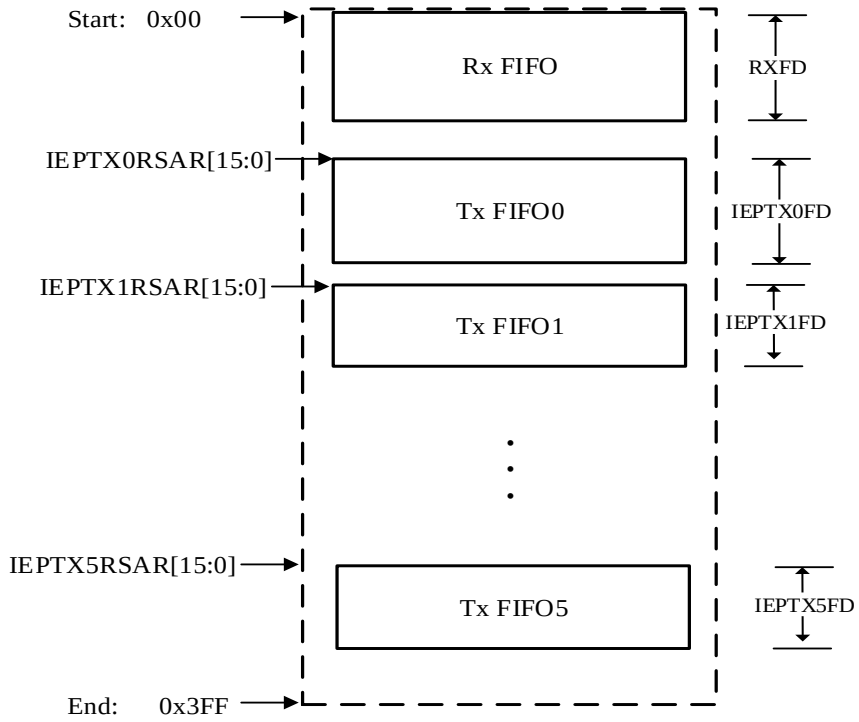
图 27-7. 主机模式 FIFO 访问寄存器映射表



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 6 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 USBHS_GRFLEN 和 USBHS_DIEPxTFLEN （x=0…5），程序可配置数据 FIFO 的大小和起始偏移地址。*图 27-8. 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 27-8. 设备模式 FIFO 空间



在 DMA 模式下，DMA 负责系统存储区和数据 FIFO 之间的数据包传输。在非 DMA 模式下，程序将包数据写入数据 FIFO 或从数据 FIFO 读取包数据。USBHS 为程序提供了专有寄存器空间来读写数据 FIFO。*图 27-9. 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 USBHS_GRXTATR/USBHS_GRSTATP 来访问 Rx FIFO。

图 27-9. 设备模式 FIFO 访问寄存器映射表

27.5.6. DMA 功能

该部分描述 USBHS 的 DMA 调度器和引擎。

DMA 请求和调度器

DMA 功能通过置位寄存器 USBHS_GAHBCS 的位 DMAEN 获得使能。当一个 IN/OUT 通道或 IN 端点被适当地配置和使能，或 Rx FIFO 非空，USBHS 将生成 DMA 请求。USBHS 的 DMA 调度器负责应答这些 DMA 请求。

当同时存在多个请求时，DMA 调度器负责仲裁这些请求。这些请求分为三类：Rx FIFO DMA 请求、周期性传输 DMA 请求和非周期性传输 DMA 请求。在仲裁中，Rx FIFO DMA 请求是最高优先级，周期性传输 DMA 请求是中级优先级，非周期性传输 DMA 请求是最低优先级。在处理周期性和非周期性传输 DMA 请求中，DMA 调度器实行循环仲裁方法。

综上所述，DMA 将自动处理 Rx FIFO 非空事件，所以，在 DMA 模式下，程序中可以忽略寄存器 USBHS_GINTF 的 RXFNEIF 标志位。

DMA 引擎

接收：

在主机或设备模式下，当 Rx FIFO DMA 请求获得仲裁后，DMA 驱动器开始从 Rx FIFO 读取包数据或状态条目。对于包数据而言，DMA 将数据写到特定的系统地址，该地址配置在寄存器 USBHS_HCHxDMAADDR 或 USBHS_DIEPxDMAADDR / USBHS_DIEPxDMAADDR。对于表目状态而言，在相关的通道或端点，DMA 将生成特定的标志位或中断。

主机传输：

当一个 IN 周期性或非周期性通道 DMA 请求获得仲裁后，DMA 将 IN 请求条目写入周期性或非周期性请求队列。当一个预期的 IN 传输完成，或一个 AHB/USB 总线错误发生后，DMA 停止特定的通道，生成寄存器 USBHS_HCHxINTF 的 TF 和 CH 标志位。如上文所述，在 Rx FIFO DMA 请求生成后，在 IN 传输的过程中所接受的包数据被复制到系统存储区。

当一个 OUT 周期性或非周期性通道 DMA 请求获得仲裁后，DMA 从系统存储区读取包数据，或将包数据写到内部的 Tx FIFO。当每次完成包数据复制后，DMA 总是将 OUT 请求条目写入请求队列。当一个预期的 OUT 传输完成，或一个 AHB/USB 总线错误发生后，DMA 停止特定的通道，生成寄存器 USBHS_HCHxINTF 的 TF 和 CH 标志位。

设备传输：

在设备模式下，当一个 IN 端点 DMA 请求获得仲裁后，DMA 从系统存储区读取包数据，或将包数据写到端点的 Tx FIFO。当 USBHS 获取 IN 端点的 IN 令牌后，将发送 DMA 引擎所复制的包数据。

27.5.7. 操作手册

该部分描述的是 USBHS 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，如是否使能DMA、DMA的传输类型、Tx FIFO的空阈值等，设置寄存器USBHS_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、ULPI和USB协议，设置寄存器USBHS_GUSBCS；
3. 根据应用的需求，设置寄存器USBHS_GCCFG；
4. 根据应用的需求，设置寄存器USBHS_GRFLEN、USBHS_HNPTFLEN_DIEP0TFLEN、USBHS_HPTFLEN，配置数据FIFO；
5. 通过设置寄存器USBHS_GINTEN使能模式错误和主机端口中断，置位USBHS_GAHBCS寄存器的GINTEN位使能全局中断；
6. 通过设置寄存器USBHS_HCTL的SPDFSLS位，判断是否将设备速度限制为全速。
7. 设置寄存器USBHS_HPCS，置位PP位；
8. 等待设备连接，当设备连接后，触发寄存器USBHS_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；
9. 等待USBHS_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USBHS_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USBHS_HCHxCTL，在设置期间，要保证位CEN和CDIS保持清除；
2. 设置寄存器USBHS_HCHxINTEN，设置期望的中断使能位；
3. 在DMA使能的前提下，设置寄存器USBHS_HCHxDMAADDR；
4. 设置寄存器USBHS_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；
5. 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USBHS_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；
6. 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；
7. 置位寄存器USBHS_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，USBHS 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，USBHS 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，USBHS 将清除 CEN 和 CDIS 位。

对于 IN 通道而言，USBHS 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，USBHS 将清除 CEN 和 CDIS 位。

IN 传输操作顺序（DMA 除能）：

1. 初始化USBHS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，USBHS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，USBHS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，USBHS在USB总线上开始IN事务；
6. 当IN事务结束时（收到ACK握手包），USBHS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，USBHS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。USBHS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。

IN 传输操作顺序（DMA 使能）：

1. 初始化USBHS全局寄存器；
2. 初始化并使能相应通道；
3. 在通过软件使能IN通道后，USBHS在相应请求队列中生成一个Rx请求条目；
4. USBHS逐一处理IN请求队列中的请求条目，并将它们所指示的IN事务发到USB总线上；
5. 当一个IN事务获得NAK握手包时，DMA可以自动地再发IN令牌直至USBHS获得预期的数据包的数目；
6. 在USBHS获取寄存器USBHS_HCHxTLEN的位PCNT中期望数据包数目后，USBHS生成TF和CH标志来表示传输成功完成，相应通道除能。如果在这些事务期间发生USB总线错误或DMA取值错误，DMA将触发相关的错误标志，停止该通道的操作，最后除能该通道，触发CH标志。

注意：在 DMA 模式下，因为 DMA 将自动处理 Rx FIFO，程序不再使能或处理 RXFNEIF 中断。

OUT 传输操作顺序（DMA 除能）：

1. 初始化USBHS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，USBHS在相应的请求队列中产生一个Tx请求条目，并且将USBHS_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，USBHS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，USBHS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USBHS_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，USBHS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。

OUT 传输操作顺序（DMA 使能）：

1. 初始化USBHS全局寄存器；
2. 初始化并使能相应通道；
3. USBHS的DMA开始从寄存器USBHS_HCHxDMAADDR的位DMAADDR中所指示的地址取包数据，并且将数据写入相应通道的Tx FIFO（周期性Tx FIFO或非周期性Tx FIFO）。每当一个完整的包数据被写入FIFO中，USBHS在相应的请求队列中生成一个Tx请求条目，并减少寄存器USBHS_HCHxTLEN的位TLEN的数值，所减少的数值与所完成写操作的包大小相同；
4. USBHS逐一处理请求队列中的请求条目，并将它们所指示的事务发到USB总线上；
5. 当一个事务获得NAK或PING握手包时，DMA可以再取或是再发数据包，在执行PING协议时也会自动像这样执行；
6. 如果所有的事务都被成功发送到USB总线上，USBHS生成TF和CH标志来表示传输成功完成，相应通道除能。如果在这些事务期间发生USB总线错误或DMA取值错误，DMA将触发相关的错误标志，停止该通道的操作，最后除能该通道，触发CH标志。

注意：在 DMA 模式下，因为 DMA 将自动处理 Rx FIFO，程序不再使能或处理 RXFNEIF 中断。

设备模式

全局寄存器初始化顺序：

1. 根据应用的需求，如是否使能DMA、DMA的传输类型、Tx FIFO的空阈值等，设置寄存器USBHS_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、ULPI和USB协议，设置寄存器USBHS_GUSBCS；
3. 根据应用的需求，设置寄存器USBHS_GCCFG；
4. 根据应用的需求，设置寄存器USBHS_GRFLEN、USBHS_HNPTFLEN_DIEP0TFLEN、USBHS_HPTFLEN，配置数据FIFO；
5. 通过设置寄存器USBHS_GINTEN使能模式错误、挂起、SOF、枚举完成和USB复位中断，置位USBHS_GAHBCS寄存器的GINTEN位使能全局中断；
6. 根据应用的需求，如设备的地址和设备的速度等，设置寄存器USBHS_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USBHS_GINTF的RST中断；
8. 等待寄存器USBHS_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 USBHS_DIEPnCTL 或 USBHS_DOEPxCTL；
2. 设定寄存器 USBHS_DIEPINTEN 或 USBHS_DOEPINTEN，置位相应中断使能位；
3. 如果 DMA 使能，设定寄存器 USBHS_DIEPxDMAADDR 或 USBHS_DOEPxDMAADDR；
4. 设定寄存器 USBHS_DIEPxLEN 或 USBHS_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；
5. 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 USBHS_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；

6. 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；
7. 置位 USBHS_DIEPxCTL 或 USBHS_DOEPxCTL 寄存器 EPEN 位使能端点。

端点除能顺序

当 USBHS_DIEPnCTL 或 USBHS_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（DMA 除能）：

1. 初始化USBHS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，USBHS减少USBHS_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，USBHS发送数据包，在USB总线上的事务完成后，USBHS_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，USBHS生成一个TF标志位以表明传输成功结束，除能相应IN端点。

IN 传输操作顺序（DMA 使能）：

1. 初始化USBHS全局寄存器；
2. 初始化并使能相应端点；
3. 将数据包写入端点的Tx FIFO，每当包数据写入FIFO，USBHS减少USBHS_DIEPxLEN寄存器的TLEN域的数值，其减少的数值等于已写的包数据大小；
4. 当IN令牌接收后，USBHS发送数据包，在USB总线上的事务完成后，USBHS_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，USBHS生成一个TF和EPDIS标志位表明传输成功结束，除能相应IN端点。如果在事务期间出现USB总线错误或DMA取值错误，DMA将触发相关错误标志。

注意：在 DMA 模式下，因为 DMA 将自动处理 Rx FIFO，程序不再使能或处理 RXFNEIF 中断。

OUT 传输操作顺序（DMA 除能）：

1. 初始化USBHS全局寄存器；
2. 初始化和使能端点；
3. 当OUT令牌接收后，USBHS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（USBHS接收并保存数据到Rx FIFO，发送ACK握手包），USBHS_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，USBHS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。USBHS生成TF标志来指示传输成功结束。USBHS生成一个TF标志位以表明传输成功结束，除能相应OUT端点。

OUT 传输操作顺序（DMA 使能）：

1. 初始化USBHS全局寄存器；
2. 初始化并使能相应OUT端点；
3. 当OUT令牌接收后，USBHS接收包数据或基于Rx FIFO状态和寄存器配置回复NAK握手包。如果事务成功完成（USBHS接收并保存数据到Rx FIFO，发送ACK握手包），USBHS_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发送，USBHS生成一个TF和EPDIS标志位表明传输成功结束，除能相应端点。如果在事务期间出现USB总线错误或DMA取值错误，DMA将触发相关错误标志。

注意：在 DMA 模式下，因为 DMA 将自动处理 Rx FIFO，程序不再使能或处理 RXFNEIF 中断。

27.6. 中断

OTG 有四种中断：全局中断、唤醒中断、端点 1 IN 中断和端点 1 OUT 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 USBHS_GINTF 寄存器读取，列举在*表 27-3. USBHS 全局中断*中。

表 27-3. USBHS 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
LPMIF	LPM 中断标志	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 / 同步OUT传输未完成中断标志	主机或设备模式
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

中断标志	描述	运行模式
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 USBHS 处于挂起状态时触发，即使 USBHS 的时钟停止。寄存器 USBHS_GINTF 的位 WKUPIF 是唤醒源。

端点 1 IN/OUT 中断是适用于端点 1 的两个特殊中断，程序可通过这两个中断快速回应端点 1 的事件。这两个是中断通过寄存器 USBHS_DEP1INT 各自使能，这两个中断源来自于寄存器 USBHS_DIEP1INTF 和 USBHS_DOEP1INTF，其中断使能位定义在寄存器 USBHS_DIEP1INTEN 和 USBHS_DOEP1INTEN。

27.7. USBHS 寄存器

USBHS 基地址：0x5000 0000

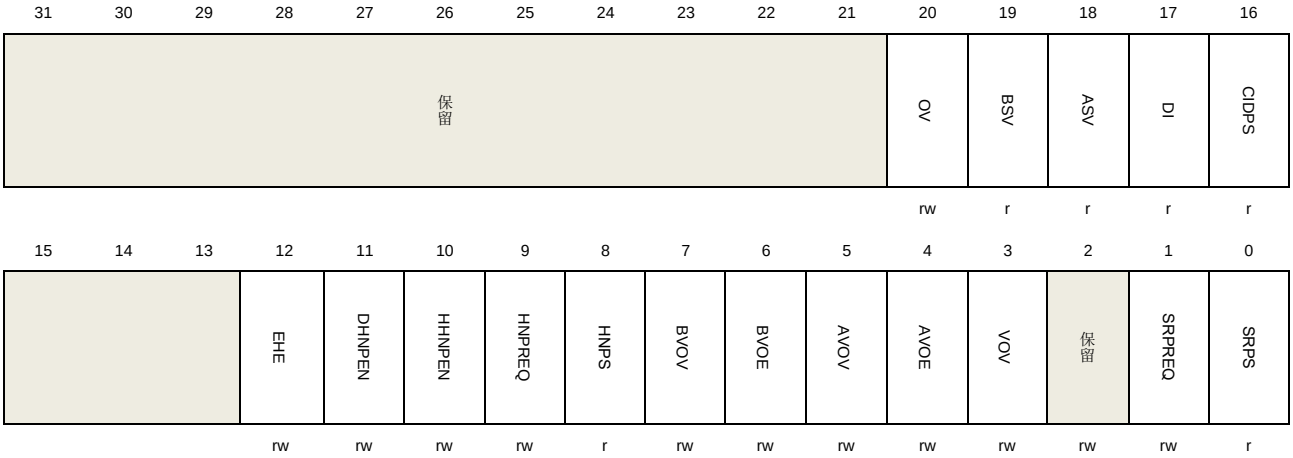
27.7.1. 全局控制与状态寄存器组

全局 OTG 控制和状态寄存器（USBHS_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
31:21	保留	必须保持复位值
20	OV	OTG版本选择 0：1.3版本，SRP支持数据线脉冲和VBUS脉冲 1：2.0版本，SRP仅支持数据线脉冲
19	BSV	B会话有效（在OTG协议中描述） 0：OTG B设备VBUS电压水平低于VBSESSVLD 1：OTG B设备VBUS电压水平不低于VBSESSVLD 注意：仅在OTB B设备模式下可访问
18	ASV	A会话有效 A主机模式收发器状态 0：OTG A设备VBUS电压水平低于VASESSVLD 1：OTG A设备VBUS电压水平不低于VASESSVLD 在会话的开始，A设备默认是主机。 注意：仅在OTG A设备模式下可访问
17	DI	去抖动间隔 检测到连接的去抖动间隔。

		0：当USB总线上发生插入和连接时，表示长去抖动间隔 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 连接器ID引脚的电压水平 0：USBHS工作在A设备模式 1：USBHS工作在B设备模式 注意：在设备和主机模式下均可访问
15:13	保留	必须保持复位值
12	EHE	嵌入式主机使能 0：选择OTG A设备状态 1：选择嵌入式主机状态
11	DHNPEN	设备HNP使能 使能B设备HNP功能。如果该控制位清除，当应用置位USBHS_GOTGCS寄存器中的HNPREQ控制位c时，USBHS并不启动HNP协议。 0：HNP功能不使能 1：HNP功能使能 注意：仅在设备模式下访问
10	HHNPEN	主机HNP使能 使能A设备HNP功能。如果该控制位清除，USBHS不能够响应B设备的HNP请求。 0：HNP功能不使能 1：HNP功能使能 注意：仅在主机模式下访问
9	HNPREQ	HNP请求 软件通过置位该控制位在USB总线上启动一个HNP。当USBHS_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USBHS_GOTGINTF寄存器中的HNPEND控制位来清除该控制位。 0：不发送HNP请求 1：发送HNP请求 注意：仅在设备模式下访问
8	HNPS	HNP成功标志位 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。 0：HNP失败 1：HNP成功 注意：仅在设备模式下访问
7	BVOV	B外设会话有效覆盖值 0：BVOE=1时，B外设会话有效值为0 1：BVOE=1时，B外设会话有效值为1 注意：仅在设备模式下可访问
6	BVOE	B外设会话有效覆盖使能

0：覆盖禁用，从PHY接收内部B外设会话有效值
1：覆盖使能，从PHY接收的内部B外设会话有效值被BVOV值覆盖
注意：仅在设备模式下可访问

5 AVOV A外设会话有效覆盖值
0：AVOE=1时，A外设会话有效值为0
1：AVOE=1时，A外设会话有效值为1
注意：仅在设备模式下可访问

4 AVOE A外设会话有效覆盖使能
0：覆盖禁用，从PHY接收内部A外设会话有效值
1：覆盖使能，从PHY接收的内部A外设会话有效值被AVOV值覆盖
注意：仅在设备模式下可访问

3 VOV VBUS有效覆盖值
0：VOE=1时，VBUS有效值为0
1：VOE=1时，VBUS有效值为1
注意：仅在设备模式下可访问

2 VOE VBUS有效覆盖使能
0：覆盖禁用，从PHY接收VBUS有效值
1：覆盖使能，从PHY接收的VBUS有效值被VOV值覆盖
注意：仅在设备模式下可访问

1 SRPREQ SRP请求
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USBHS_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USBHS_GOTGINTF寄存器中的SRPEND控制位来清除该控制位。
0：没有会话请求
1：会话请求
注意：仅在设备模式下访问

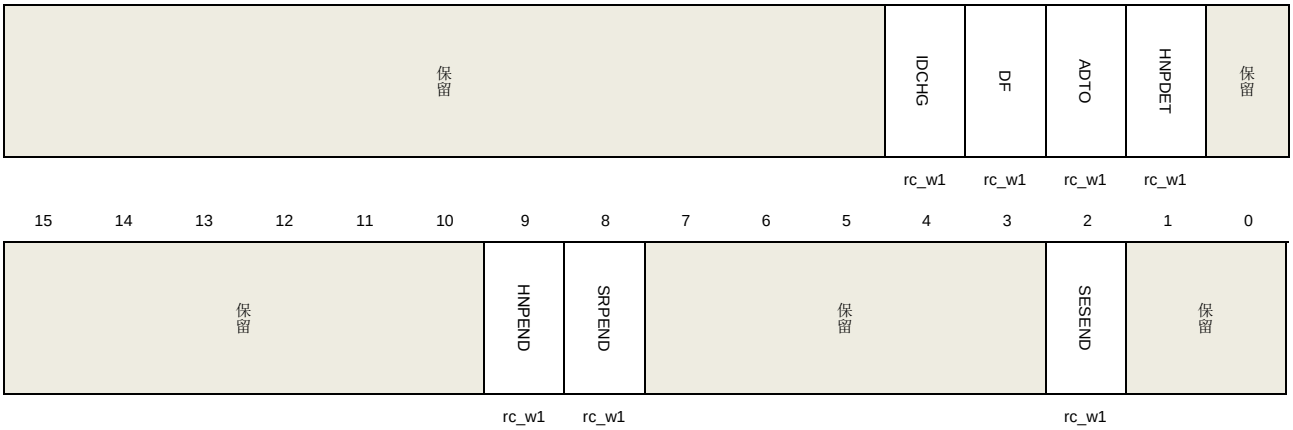
0 SRPS SRP会话请求成功
当SRP会话请求成功时，该标志位由内核置位。当SRPREQ控制位被置位时，该标志位被清除。
0：SRP会话请求失败
1：SRP会话请求成功
注意：仅在设备模式下访问

全局 OTG 中断状态寄存器（USBHS_GOTGINTF）

地址偏移：0x0004
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



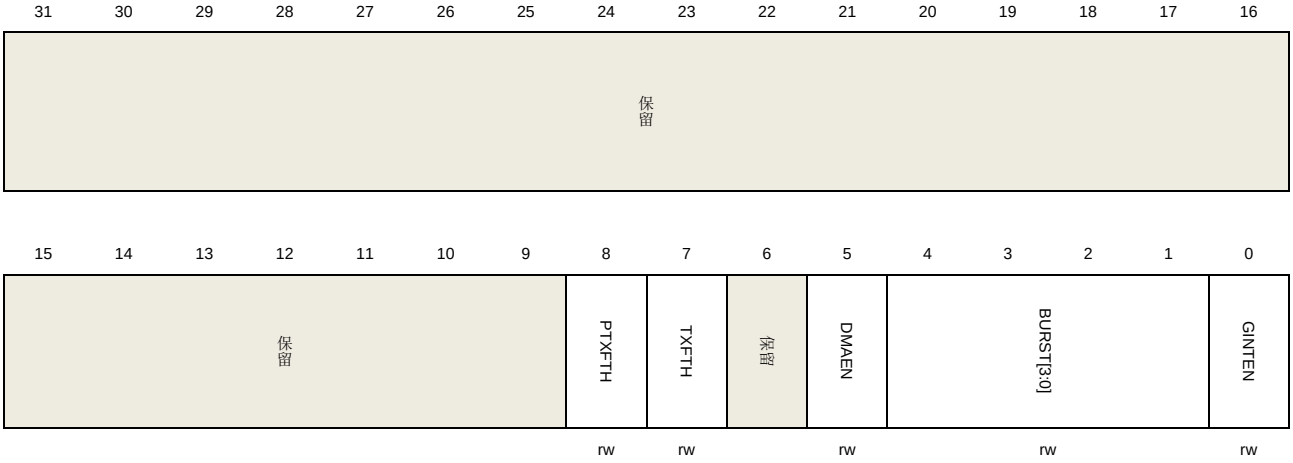
位/位域	名称	描述
31:21	保留	必须保持复位值
20	IDCHG	ID输入的值有变化
19	DF	去抖动完成 当设备连接去抖动完成时，USBHS置位该控制位 注意：仅在主机模式下可访问
18	ADTO	A设备超时 当A设备等待B设备连接发生超时，USBHS置位该控制位 注意：在设备和主机模式下，均可访问
17	HNPDET	检测到主机协商请求 当A设备检测到一个HNP请求时，USBHS置位该标志位 注意：在设备和主机模式下，均可访问
16:10	保留	必须保持复位值
9	HNPEND	HNP结束 当一个HNP结束时，内核置位该标志位。软件应该读取USBHS_GOTGCS寄存器中HNPS标志位，以获取HNP结果。 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USBHS_GOTGCS寄存器中SRPS标志位，以获取SRP结果。 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 当VBUS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（USBHS_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 0：当周期性发送FIFO半空时，将触发PTXFEIF标志位 1：当周期性发送FIFO全空时，将触发PTXFEIF标志位 注意：只在主机模式下访问
7	TXFTH	Tx FIFO阈值 设备模式： 0：当IN端点发送FIFO半空时，将触发TXFEIF标志位 1：当IN端点发送FIFO全空时，将触发TXFEIF标志位 主机模式： 0：当非周期性发送FIFO半空时，将触发NPTXFEIF标志位 1：当非周期性发送FIFO全空时，将触发NPTXFEIF标志位
6	保留	必须保持复位值
5	DMAEN	DMA功能使能 0：DMA功能使能 1：DMA功能禁用
4:1	BURST[3:0]	DMA使用的AHB突发类型 0000：单次 0001：INCR 0011：INCR4 0101：INCR8 0111：INCR16

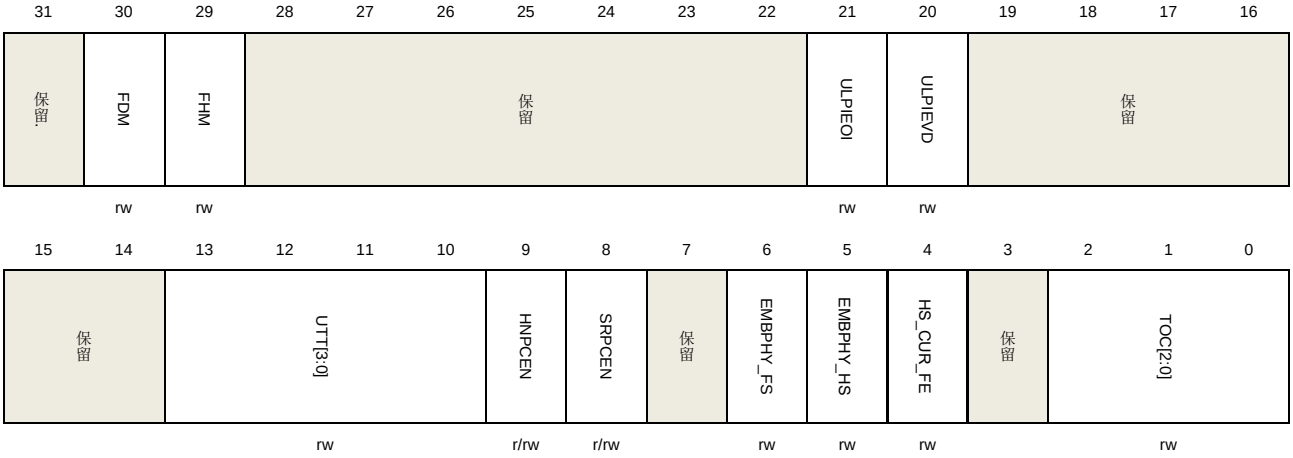
0	GINTEN	全局中断使能 0：全局中断不使能 1：全局中断使能 注意：在主机和设备模式下，均可访问

全局 USB 控制和状态寄存器（USBHS_GUSBCS）

地址偏移：0x000C

复位值：0x0000 0A00

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 通过置位该控制位，可强制USB内核为设备模式，并且忽略USBHS ID引脚的输入状态 0：正常模式 1：设备模式 设置该控制位后，应用必须等待至少25ms，让变化产生作用。 注意：在设备和主机模式下，均可访问。
29	FHM	强制主机模式 通过置位该控制位，可强制USB内核为主机模式，并且忽略USBHS ID引脚的输入状态 0：正常模式 1：主机模式 设置该控制位后，应用必须等待至少25ms，让变化产生作用。 注意：在设备和主机模式下，均可访问
28:22	保留	必须保持复位值
21	ULPIEOI	ULPI外部过流指示器 ULPI PHY使用该控制位决定使用内部或者外部过流指示器。该控制位只在外部ULPI

		PHY被使用时（本寄存器中EMBPHY_HS和EMBPHY_FS控制位为0），才有效。 0：ULPI PHY使用内部过流指示器 1：ULPI PHY使用外部过流指示器
20	ULPIEVD	ULPI外部VBUS驱动器 ULPI PHY使用该控制位决定是由ULPI PHY还是外部电源驱动。该控制位仅在外部ULPI PHY被使用时（本寄存器中EMBPHY_HS和EMBPHY_FS控制位为0），才有效。 0：VBUS由ULPI PHY驱动 1：VBUS由外部电源驱动
19:14	保留	必须保持复位值
13:10	UTT[3:0]	USB运转时间 以物理时钟数来设定运转时间 注意：仅在设备模式下访问
9	HNPCEN	HNP能力使能 控制HNP能力是否使能 0：HNP能力禁用 1：HNP能力使能 注意：在设备和主机模式下，均可访问
8	SRPCEN	SRP能力使能 控制SRP能力是否使能 0：SRP能力禁用 1：SRP能力使能 注意：在设备和主机模式下，均可访问
7	保留	必须保持复位值
6	EMBPHY_FS	嵌入式全速PHY选择 0：嵌入式全速PHY禁用 1：嵌入式全速PHY使能 注意：该位仅在EMBPHY_HS为0时才能置1，在设备和主机模式下，均可访问
5	EMBPHY_HS	嵌入式高速PHY选择 0：嵌入式高速PHY禁用 1：嵌入式高速PHY使能 注意：该位仅在EMBPHY_FS为0时才能置1，在设备和主机模式下，均可访问
4	HS_CUR_FE	HS当前软件使能 0：释放HS模式，TX当前使能 1：强制HS模式，TX当前使能
3	保留	必须保持复位值
2:0	TOC[2:0]	超时校准 当等待一个包时，USBHS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时钟频率由使用的PHY所决定：

48MHz（内部嵌入式PHY）和60MHz（外部ULPI PHY）。

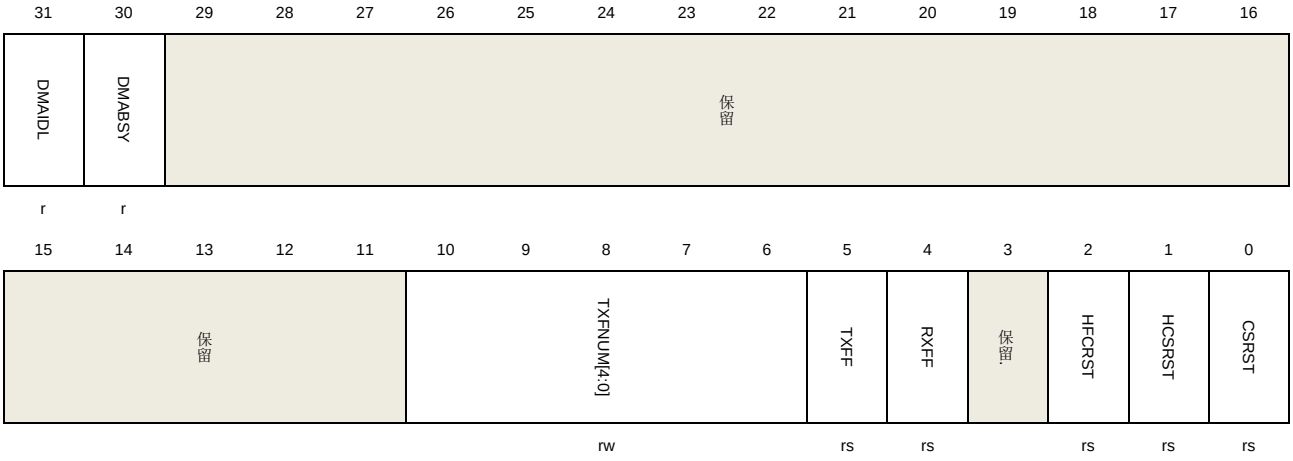
全局复位控制寄存器（USBHS_GRSTCTL）

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	DMAIDL	DMA空闲状态 该标志位体现了DMA是否在空闲状态 0：DMA在空闲状态 1：DMA不在空闲状态 注意：在设备和主机模式下，均可访问
30	DMABSY	DMA忙标志位 该标志位体现了DMA是否忙 0：DMA不忙 1：DMA忙 注意：在设备和主机模式下，均可访问
29:11	保留	必须保持复位值
10:6	TXFNUM[4:0]	Tx FIFO数目 当本寄存器中TXFF控制位置位时，该标志位决定那个Tx FIFO会被冲刷 主机模式： 00000：仅非周期性Tx FIFO被冲刷 00001：仅周期性Tx FIFO被冲刷 1xxxx：周期性和非周期性Tx FIFO均被冲刷 其他：没有数据被冲刷 设备模式： 00000：仅Tx FIFO0被冲刷

		00001：仅Tx FIFO1被冲刷 … 00101：仅Tx FIFO5被冲刷 1XXXX：所有的Tx FIFO均被冲刷 其他：没有数据被冲刷
5	TXFF	Tx FIFO冲刷控制位 应用通过置位该控制位来冲刷Tx FIFO数据，并且TXFNUM[4:0]决定冲刷的FIFO数目。当冲刷完成后，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USBHS不应有其他操作。 注意：在设备和主机模式下，均可访问
4	RXFF	Rx FIFO冲刷控制位 应用通过置位该控制位来冲刷Rx FIFO数据。当冲刷完成后，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USBHS不应有其他操作。 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 应用通过置位该控制位来复位USBHS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USBHS不应有其他操作。 注意：仅在主机模式下访问
1	HCSRST	HCLK软件复位 应用通过置位该控制位来复位ABH时钟域电路 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，应用应该等待该控制位清除，并且，在此之前USBHS不应有其他操作。 注意：在设备和主机模式下，均可访问
0	CSRST	USB内核软件复位 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（USBHS_GINTF）

地址偏移：0x0014

复位值：0x0400 0021

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
WKUPIF	SESIF	DISCIF	IDPSC	LPMIF	PTXFEIF	HCIF	HPIF	保留		PXNCIF/ISOONCIF	ISOINCIF	OEPIF	IEPIF	保留	
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	r	r	r			rc_w1	rc_w1	r	r		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

EOPIF	ISOOPDIF	ENUMF	RST	SP	ESP	保留	GONAK	GNPINAK	NPTXFEIF	RXFNEIF	SOF	OTGIF	MFIF	COPM
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1		r	r	r	r	rc_w1	r	rc_w1	r

位/位域	名称	描述
31	WKUPIF	唤醒中断标志位 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。 注意：在设备和主机模式下，均可访问
30	SESIF	会话中断标志位 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 注意：在设备和主机模式下，均可访问
29	DISCIF	断开中断标志位 当设备断开后，将触发该标志位。 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 当ID引脚状态改变时，内核将置位该标志位 注意：在设备和主机模式下，均可访问
27	LPMIF	LPM中断标志位 在主机模式下，当设备以ACK、NYET或STALL响应LPM事务，或者主机已经发送了RECNT（USBHS_LPMCFG寄存器）次LPM事务，中断会被触发 在从机模式下，当设备已经接收到LPM事务并且以ACK、NYET或STALL进行了响应，中断会被触发
26	PTXFEIF	周期性Tx FIFO空中断标志位 当周期性发送FIFO半空或全空时，将触发该标志位。空阈值由USBHS_GAHBCS寄存器中周期性Tx FIFO空等级控制位（PTXFTH）决定。 注意：仅在主机模式下访问
25	HCIF	主机通道中断标志位 当在主机模式下其中一个通道挂起一个中断时，USBHS将置位该标志位。软件应该首先读取USBHS_HACHINT寄存器以获取通道号，然后读取相应的USBHS_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。 注意：仅在主机模式下访问
24	HPIF	主机端口中断标志位 当USBHS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USBHS_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。

		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 在当前帧内，当帧结束时，周期性传输未完成，USBHS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 在周期性帧结束时（由USBHS_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，USBHS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 在周期性帧结束时（由USBHS_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，USBHS将置位该标志位（设备模式）。 注意：仅在设备模式下访问
19	OEPIF	OUT端点中断标志位 当在设备模式下，其中一个OUT端点挂起一个中断时，USBHS将置位该中断标志位。软件应该首先读取USBHS_DAEPINT寄存器以获取设备号，然后读取相应的USBHS_ 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。 注意：仅在设备模式下访问
18	IEPIF	IN端点中断标志位 当在设备模式下，其中一个IN端点挂起一个中断时，USBHS将置位该标志位。软件应该首先读取USBHS_DAEPINT寄存器以获取设备号，然后读取相应的USBHS_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 当一帧内USB总线时间已经达到USBHS_DCFG寄存器中EOPFT控制位所定义的数值时，USBHS将置位该中断标志位。 注意：仅在设备模式下访问
14	ISOOPDIF	同步OUT包丢失中断标志位 如果USBHS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，USBHS将置位该标志位。 注意：仅在设备模式下访问
13	ENUMF	枚举完成中断标志位 在速度枚举完成后，USBHS将置位该中断标志位。软件能够读取USBHS_DSTAT寄存器以获取当前设备速度。 注意：仅在设备模式下访问
12	RST	USB复位中断标志位 当USBHS在USB总线上检测到一个USB复位信号后，USBHS将置位该中断标志位。

		注意：仅在设备模式下访问
11	SP	USB挂起中断标志位 当USBHS检测到USB总线空闲3ms并且进入挂起状态，USBHS将置位该中断标志位。 注意：仅在设备模式下访问
10	ESP	早期挂起中断标志位 当USBHS检测到USB总线空闲3ms时，USBHS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 软件能够向USBHS_DCTL寄存器的SGONAK控制位写1，并且USBHS将会在SGONAK写入有效后，置位GONAK标志位。软件可通过向USBHS_DCTL寄存器的CGONAK控制位写1，清除该标志位 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 软件能够向USBHS_DCTL寄存器中的SGINAK控制位写1，并且USBHS将会在SGINAK写入有效后，置位GNPINAK标志位。软件可通过向USBHS_DCTL寄存器的CGINAK控制位写1，清除该标志位 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 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USBHS_GAHBCS寄存器中的非周期Tx FIFO空等级控制位（TXFTH）决定。 注意：仅在主机模式下访问
4	RXFNEIF	Rx FIFO非空中断标志位 当至少有一个包或状态条目在Rx FIFO中时，USBHS将置位该标志位。 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 主机模式： 当准备在USB总线上发送一个SOF或保持有效信号，USBHS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。 设备模式： 当USBHS接收到一个SOF令牌包后，USBHS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。 注意：在设备和主机模式下，均可访问
2	OTGIF	OTG中断标志位 当USBHS_GOTGINTF寄存器中标志位产生一个中断时，USBHS置位该中断标志位。软件应该读取USBHS_GOTGINTF寄存器以获取产生该中断的信号源，当USBHS_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。 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 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，USBHS将置位该中断标志位。这些错误操作不会产生作用。 注意：在主机和设备模式下，均可访问
0	COPM	当前操作模式 0：设备模式 1：主机模式 注意：在主机和设备模式下，均可访问

全局中断使能寄存器（USBHS_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（USBHS_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	LPMIE	PTXFEIE	HCIE	HPIE	保留		PXNCIE/ISOONCIE	ISOINCIE	OEPIE	IEPIE	保留	
rw	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 0：禁用唤醒中断 1：使能唤醒中断 注意：在主机和设备模式下，均可访问
30	SESIE	会话中断使能 0：禁用会话中断 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 0：禁用断开中断 1：使能断开中断

		注意：仅在设备模式下使用
28	IDPSCIE	ID引脚状态改变中断使能 0：禁用连接器ID引脚状态中断 1：使能连接器ID引脚状态中断 注意：在主机和设备模式下，均可访问
27	LPMIE	LPM中断使能 0：禁用LPM中断 1：使能LPM中断 注意：在主机和设备模式下，均可访问
26	PTXFEIE	周期性Tx FIFO空中断使能 0：禁用周期性Tx FIFO空中断 1：使能周期性Tx FIFO空中断 注意：仅在主机模式下访问
25	HCIE	主机通道中断使能 0：禁用主机通道中断 1：使能主机通道中断 注意：仅在主机模式下访问
24	HPIE	主机端口中断使能 0：禁止主机端口中断 1：使能主机端口中断 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 0：禁止周期性未完成传输中断 1：使能周期性未完成传输中断 注意：仅在主机模式下访问
	ISOONCIE	同步OUT传输未完成中断使能 0：禁止同步OUT传输未完成中断 1：使能同步OUT传输未完成中断 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 0：禁止同步IN传输未完成中断 1：使能同步IN传输未完成中断 注意：仅在设备模式下访问
19	OEPIE	OUT端点中断使能 0：禁止OUT端点中断 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

		0：禁止IN端点中断 1：使能IN端点中断 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 0：禁止周期性帧结束中断 1：使能周期性帧结束中断 注意：仅在设备模式下访问
14	ISOOPDIE	同步OUT包丢失中断使能 0：禁止同步OUT包丢失中断 1：使能同步OUT包丢失中断 注意：仅在设备模式下访问
13	ENUMFIE	枚举完成中断使能 0：禁止枚举完成中断 1：使能枚举完成中断 注意：仅在设备模式下访问
12	RSTIE	USB复位中断使能 0：禁止USB复位中断 1：使能USB复位中断 注意：仅在设备模式下访问
11	SPIE	USB挂起中断使能 0：禁止USB挂起中断 1：使能USB挂起中断 注意：仅在设备模式下访问
10	ESPIE	早期挂起中断使能 0：禁止早期挂起中断 1：使能早期挂起中断 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 0：禁止全局OUT NAK有效中断 1：使能全局OUT NAK有效中断 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 0：禁止全局非周期性IN NAK有效中断 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 0：禁止非周期性发送FIFO空中断

		1：使能非周期性发送FIFO空中断 注意：仅在主机模式下访问
4	RXFNEIE	接收FIFO非空中断使能 0：禁止接收FIFO非空中断 1：使能接收FIFO非空中断 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 0：禁止帧首中断 1：使能帧首中断 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 0：禁止OTG中断 1：使能OTG中断 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 0：禁止模式错误中断 1：使能模式错误中断 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/接收状态读取和弹出寄存器（USBHS_GRSTATR/USBHS_GRSTATP）

读地址偏移：0x001C

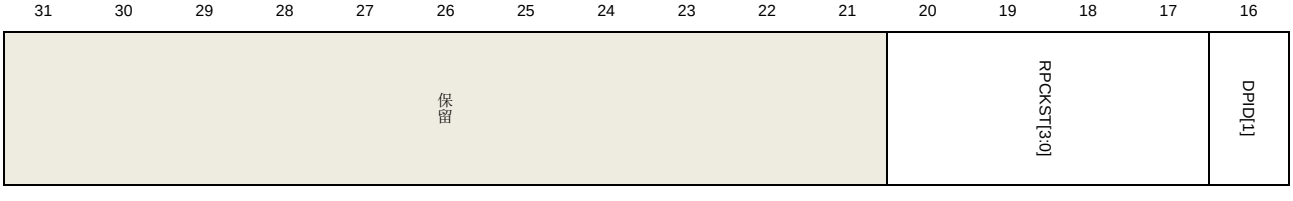
弹出地址偏移：0x0020
复位值：0x0000 0000

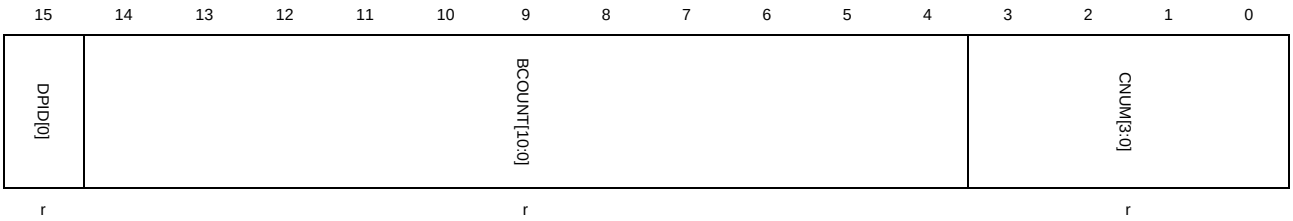
对接收状态读寄存器的读操作，将返回接收 FIFO 中顶部的条目。对接收状态读取和弹出寄存器的读操作，将额外的弹出 Rx FIFO 的顶部条目。

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（USBHS_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

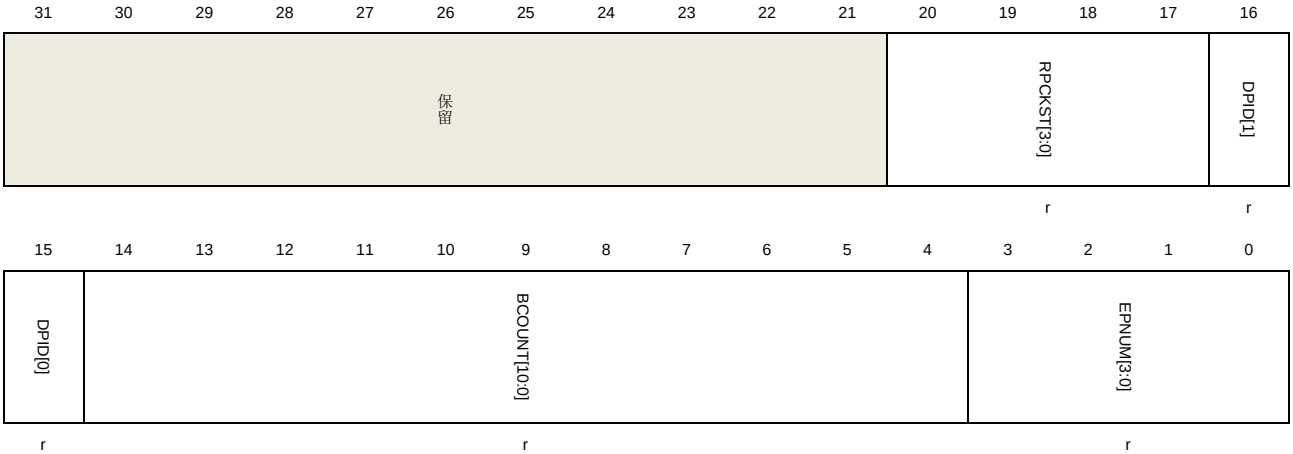
主机模式：





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 0010：接收到IN数据包 0011：IN传输完成（如果取出，触发一个中断） 0101：数据翻转错误（如果取出，触发一个中断） 0111：通道中止（如果取出，触发一个中断） 其他：保留
16:15	DPID[1:0]	数据PID 接收包的数据PID 00：DATA0 10：DATA1 01：DATA2 11：MDATA
14:4	BCOUNT[10:0]	字节数 接收IN数据包字节数。
3:0	CNUM[3:0]	通道数 当前接收包所属通道编号。

设备模式：



位/位域	名称	描述

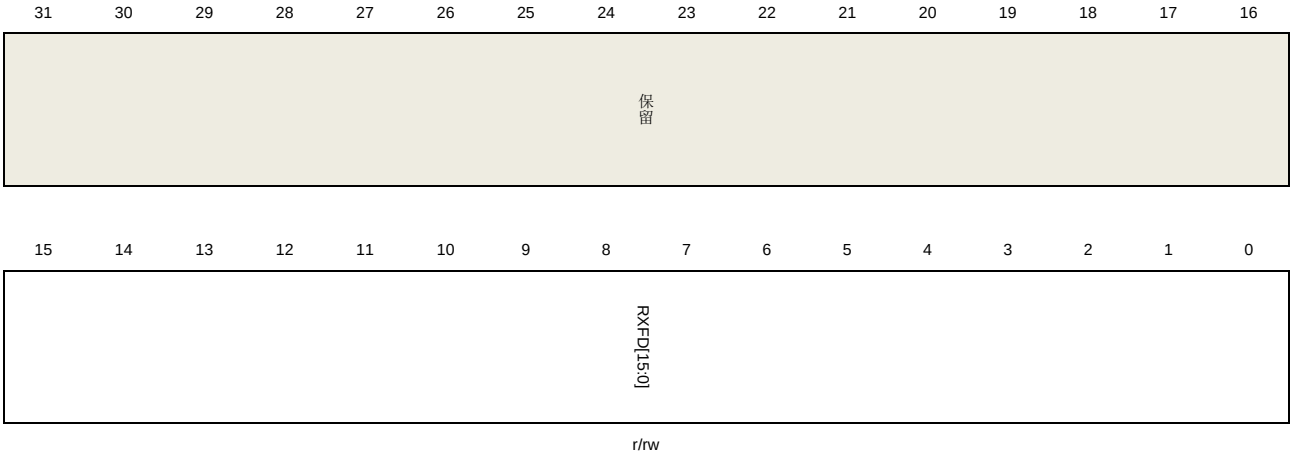
31:21	保留	必须保持复位值
20:17	RPCKST[3:0]	接收包状态 0001：全局OUT NAK（产生一个中断） 0010：接收到OUT数据包 0011：OUT传输完成（产生一个中断） 0100：SETUP传输完成（产生一个中断） 0110：接收到SETUP数据包 其他：保留
16:15	DPID[1:0]	数据PID 接收到OUT数据包的数据PID 00：DATA0 10：DATA1 01：DATA2 11：MDATA
14:4	BCOUNT[10:0]	字节数 接收数据包的字节数
3:0	EPNUM[3:0]	端点号 当前接收包所属端点编号

全局接收 FIFO 长度寄存器（USBHS_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 以32位字计数

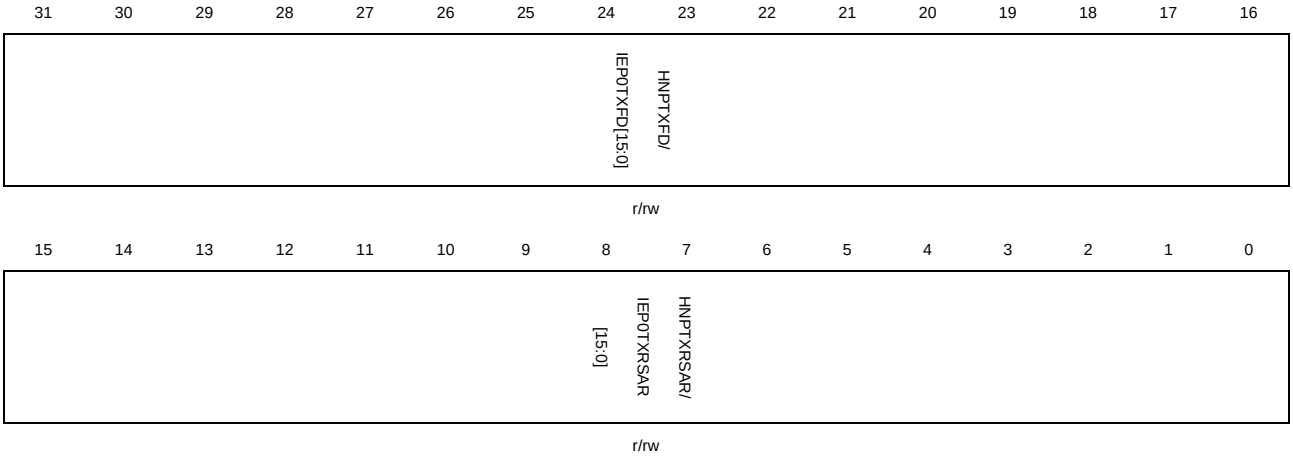
1≤RXFD≤1024

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（USBHS_HNPTFLEN_DIEP0TFLEN）

地址偏移：0x0028

复位值：0x0200 0200

该寄存器只能按字（32 位）访问



主机模式下：

位/位域	名称	描述
31:16	HNPTXFD[15:0]	主机非周期性Tx FIFO深度 以32位字计数 1≤HNPTXFD≤1024
15:0	HNPTXRSAR[15:0]	主机非周期性Tx RAM起始地址 非周期性发送FIFO RAM的起始地址

设备模式下：

位/位域	名称	描述
31:16	IEP0TXFD[15:0]	输入端点0 Tx FIFO深度 以32位字计数 16≤IEP0TXFD≤140
15:0	IEP0TXRSAR[15:0]	输入端点0 TX RAM起始地址 端点0发送FIFO RAM的起始地址

主机非周期性发送 FIFO/队列状态寄存器（USBHS_HNPTFQSTAT）

地址偏移：0x002C

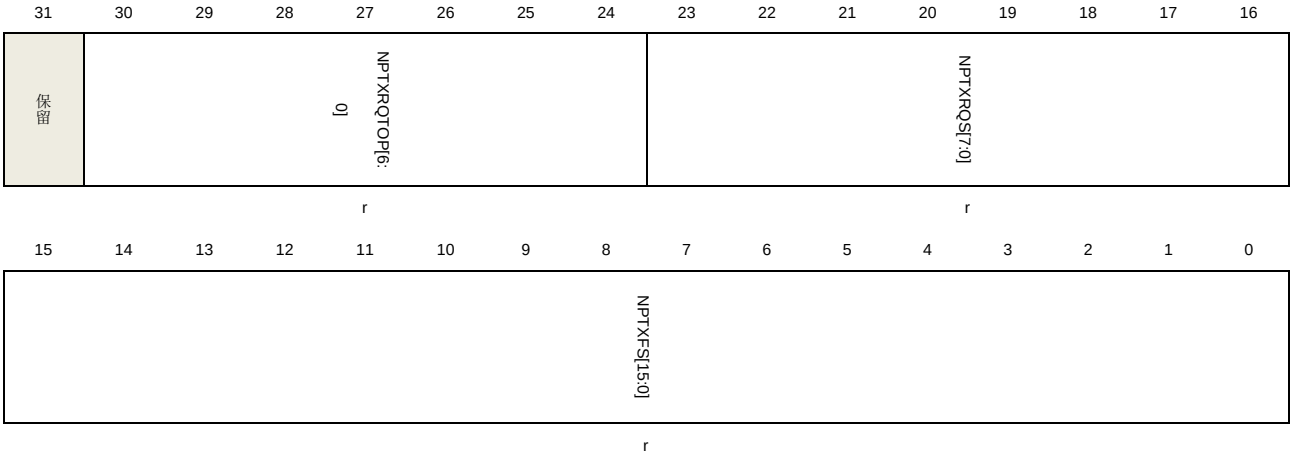
复位值：0x0008 0200

该寄存器只能按字（32 位）访问

该寄存器反映了非周期性 Tx FIFO 和请求队列的当前状态。

请求队列包括在主机模式下的 IN、OUT 或其他请求条目。

注意：在设备模式下，该寄存器不可用。



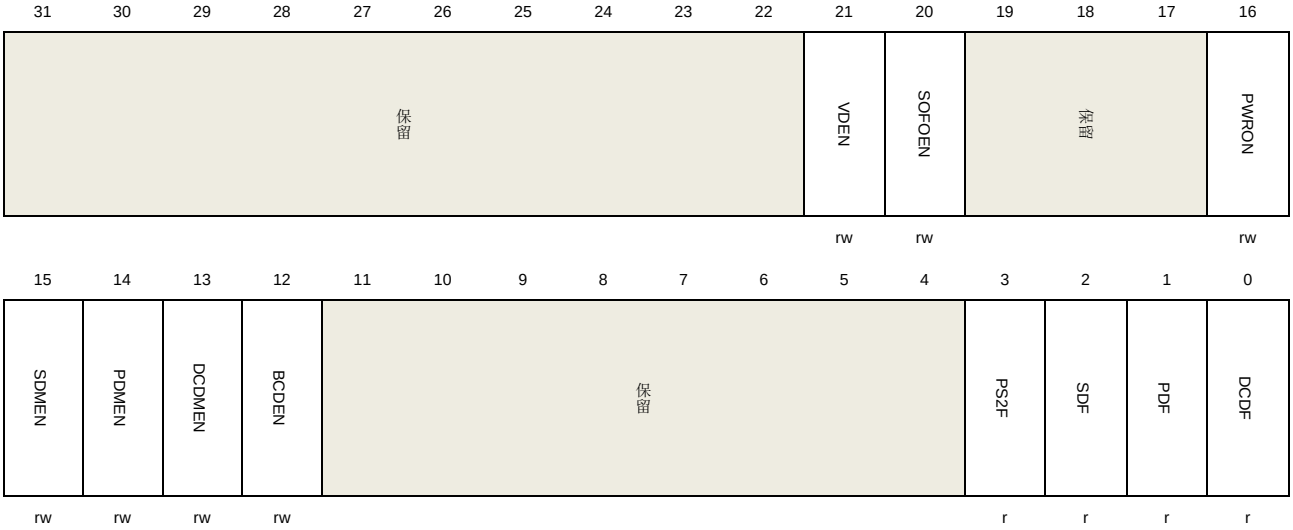
位/位域	名称	描述
31	保留	必须保持复位值
30:24	NPTXRQTOP[6:0]	非周期性发送请求队列的顶部条目 在非周期性传输请求队列中的条目。 位30:27：通道号 位26:25： – 00：IN/OUT令牌 – 01：0长度OUT包 – 11：通道中止请求 位24：结束标志位，表明所选通道的最后一个条目
23:16	NPTXRQS[7:0]	非周期性发送请求队列空间 非周期性请求队列的剩余空间 0：请求队列空 1：1个条目 2：2个条目 … n：n个条目 (0≤n≤8) 其他：保留
15:0	NPTXFS[15:0]	非周期性Tx FIFO空间 非周期性发送FIFO剩余空间 以32位字计数 0：非周期性Tx FIFO为空 1：1个字 2：2个字 … n：n个字(0≤n≤NPTXFD) 其他：保留

全局内核配置寄存器（USBHS_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	VDEN	启用VBUS感应比较器检测VBUS有效。如果支持HNP或SRP，则自动启用VBUS比较器。 0：禁用V_{BUS}检测 1：使能V_{BUS}检测
20	SOFOEN	SOF输出使能 0：SOF脉冲输出禁止 1：SOF脉冲输出使能
19:17	保留	必须保持复位值
16	PWRON	上电 该控制位为内部嵌入式全速PHY的电源开关 0：嵌入式全速PHY掉电 1：嵌入式全速PHY上电
15	SDMEN	二次检测模式使能 0：二次检测模式禁止 1：二次检测模式使能
14	PDMEN	主检测模式使能 0：主检测模式禁止 1：主检测模式使能
13	DCDMEN	数据连接检测模式使能

0：数据连接检测模式禁止
1：数据连接检测模式使能

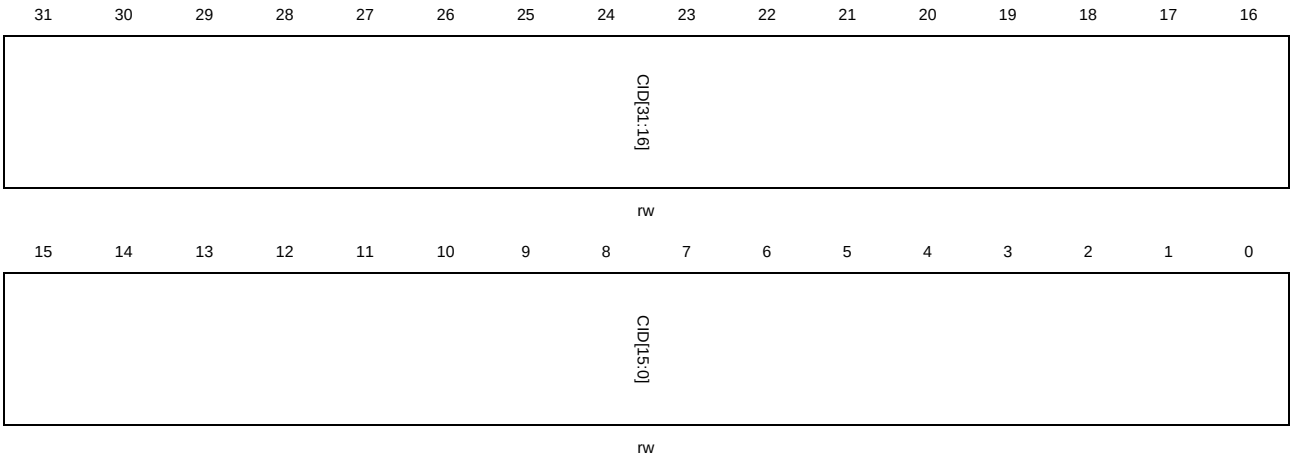
12	BCDEN	电池充电检测使能 0：电池充电检测禁止 1：电池充电检测使能
11:4	保留	必须保持复位值
3	PS2F	PS2检测状态，仅在主检测模式下激活 0：检测到普通端口 1：检测到PS2端口
2	SDF	二次检测状态 0：检测到CDP 1：检测到DCP
1	PDF	主检测状态 0：检测到BCD支持 1：检测到BCD支持
0	DCDF	数据连接检测状态 0：未检测到数据线连接 1：检测到数据线连接

内核 ID 寄存器（USBHS_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



位/位域	名称	描述
31:0	CID[31:0]	内核ID

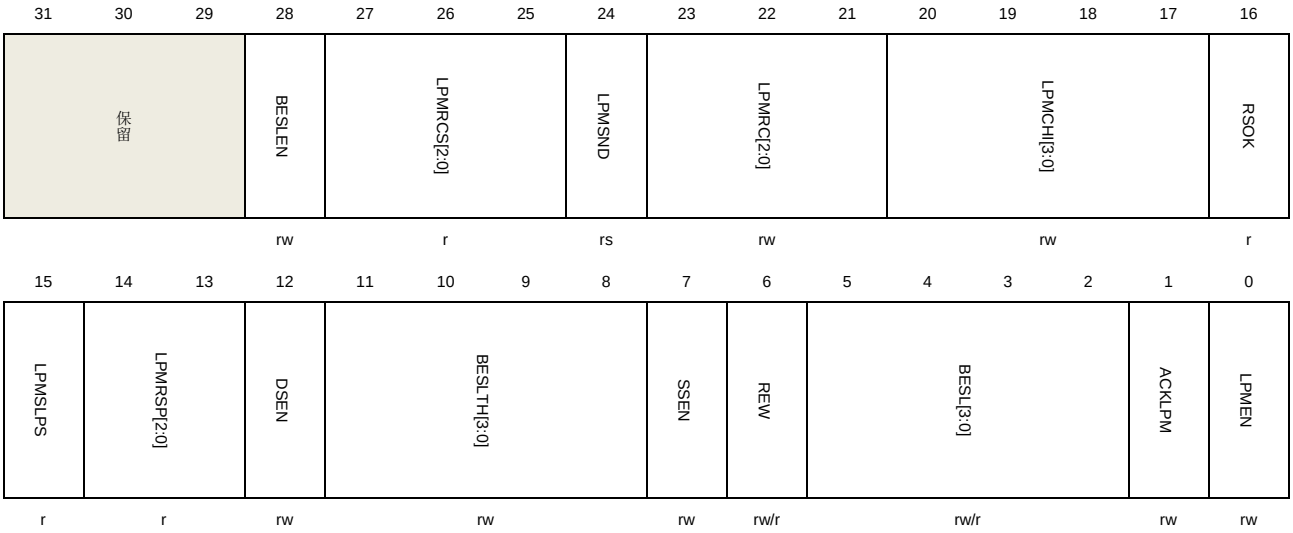
软件能够写入或读取该域值，并利用该域值为应用产生一个唯一ID。

全局内核 LPM 配置寄存器（USBHS_GLPMCFG）

地址偏移：0x0054

复位值：0x0000 0000

该寄存器只能按字（32 位）访问



位/位域	名称	描述
31:29	保留	必须保持复位值
28	BESLEN	使能 LPM 勘误表选项 0：选择 USB2.0 规范的链路电源管理附录工程变更通知 1：选择 USB2.0 规范的链路电源管理工程变更通知的勘误表
27:25	LPMRCS[2:0]	LPM 重试计数状态 注意：仅在主机模式下访问
24	LPMSND	发送 LPM 事务 当收到 ACK、STALL 或 NYET 响应，或者 LPM 发送次数达到重试计数，由硬件清除该位 注意：仅在主机模式下访问
23:21	LPMRC[2:0]	LPM 重试计数 接收到错误响应时的重试计数，直到接收到 ACK、STALL 或 NYET 响应 注意：仅在主机模式下访问
20:17	LPMCHI[3:0]	发送 LPM 事务时的通道号索引 注意：仅在主机模式下访问
16	RSOK	睡眠状态可以发送唤醒信号 主机或设备可以在进入睡眠状态 50us 后发送唤醒（TI1Residency） 当 LPMSLPS 为 0 时该位为 0 1：睡眠状态可以启动唤醒

		0：睡眠状态不能启动唤醒
15	LPMSLPS	睡眠状态 主机模式： 主机在接收到 ACK 响应后转换到睡眠状态。 从机模式： 发送 ACK 响应后，设备进入睡眠状态，TL1 令牌重发计时器已过期。 1：内核在睡眠状态 0：内核不在睡眠状态
14:13	LPMRSP[1:0]	LPM 响应 11：ACK 10：NYET 01：STALL 00：ERROR（没有响应）
12	DSEN	深度睡眠使能 使能在深度睡眠模式下挂起 PHY
11:8	BESLTH[3:0]	BESL 阈值 设备模式： 当 BESL 大于或等于 BESLTH 值时，设备进入深低功耗模式。 主机模式： BESLTH 表示当检测到设备初始化的恢复时，恢复信号（TL1HubDrvResume2）的持续时间。 0000：75us 0001：100us 0010：150us 0011：250us 0101：450us 0110：950us
7	SSEN	浅睡眠下使能 在浅睡眠模式下使能挂起 PHY
6	REW	b-远程唤醒值 主机模式： LPM 事务中发送的远程唤醒值 设备模式（只读）： 当 ACK、STALL 或 NYET 已经发送，用接收到的 LPM 事务中的 b-远程唤醒值更新
5:2	BESL[3:0]	最大限度的服务延迟 主机模式： 要在 LPM 事务中发送的 BESL 值。这也是主机初始化 resume 时 resume（TL1HubDrvResume1）的持续时间。 设备模式： 当 ACK、STALL 或 NYET 已经发送，用接收到的 LPM 事务中的 BESL 值更新 0000：125us

0001：150us
0010：200us
0011：300us
0100：400us
0101：500us
0110：1000us
0111：2000us
1000：3000us
1001：4000us
1010：5000us
1011：6000us
1100：7000us
1101：8000us
1110：9000us
1111：10000us

1	ACKLPM	在 LPM 事务中使能 ACK 1：ACK 设备仅在成功的 LPM 事务上使用 ACK 进行响应 - LPM 事务中无错误 - 无数据挂起错误 - bLinkState = 0001 在接收的 LPM 事务中 0：NYET 设备使用 NYET 进行响应 - 接收的 bLinkState 值不是 0001 - 接收 LPM 事务时产生了一个错误 注意：仅在设备模式下访问
0	LPMEN	LPM 使能 1：使能 LPM 0：禁用 LPM

断电寄存器（USBHS_PWRD）

地址偏移：0x0058

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								ADPF	保留						
								rc_w1							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

位/位域	名称	描述
31:24	保留	必须保持复位值
23	ADPF	ADP 事件中断标志位
22:1	保留	必须保持复位值
0	ADPMEN	ADP 模块使能 1：ADP 模块使能 0：ADP 模块禁用

ADP 控制和状态寄存器（USBHS_ADPCTL）

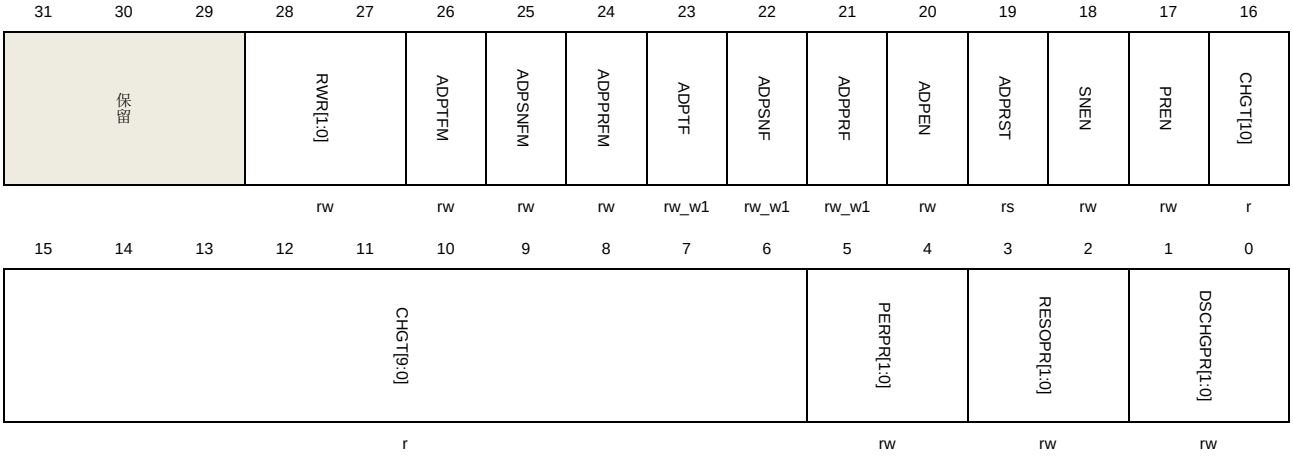
地址偏移：0x0060

复位值：0x0000 0000

为了写入该寄存器，需要写入 RWR 为 10，并保持轮询直到 RWR=00。

为了读取该寄存器，需要等待任何 ADP 标志被置位或用 01 写入 RWR 并保持轮询直到 RWR=00。

该寄存器只能按字（32 位）访问



位/位域	名称	描述
31:29	保留	必须保持复位值
28:27	RWR[1:0]	读和写请求 00：读和写有效（通过内核更新） 01：读请求 10：写请求

26	ADPTFM	ADP 超时中断标志的掩码
25	ADPSNFM	ADP 检测中断标志的掩码
24	ADPPRFM	ADP 探测中断标志的掩码
23	ADPTF	ADP 超时中断标志
22	ADPSNF	ADP 检测中断标志
21	ADPPRF	ADP 探测中断标志
20	ADPEN	ADP 使能 1：ADP 使能 0：ADP 禁用
19	ADPRST	ADP 复位 复位完成后自动清除
18	SNEN	ADP 检测使能 1：检测使能 0：检测禁止
17	PREN	ADP 探测使能 1：探测使能 0：探测禁止
16:6	CHGT[10:0]	VBUS 从 VADPSINK 渐变到 VADPPRB 的最新时间。这些位以 32 kHz 时钟周期为单位定义。 000：1 个周期 001：2 个周期 002：3 个周期 003：4 个周期 … 7ff：2048 个周期
5:4	PERPR[1:0]	探测周期 00：0.625s 到 0.925s 01：1.25s 到 1.85s 10：1.9s 到 2.6s
3:2	RESOPR[1:0]	CHGT 值的分辨率。这些位以 32 kHz 时钟周期为单位定义。如果选择 10，则 CHGT 每 3 个 32 kHz 时钟周期递增一次。 00：1 个周期 01：2 个周期 10：3 个周期 11：4 个周期
1:0	DSCHGPR[1:0]	探测放电时间 00：4 ms 01：8 ms

10：16 ms

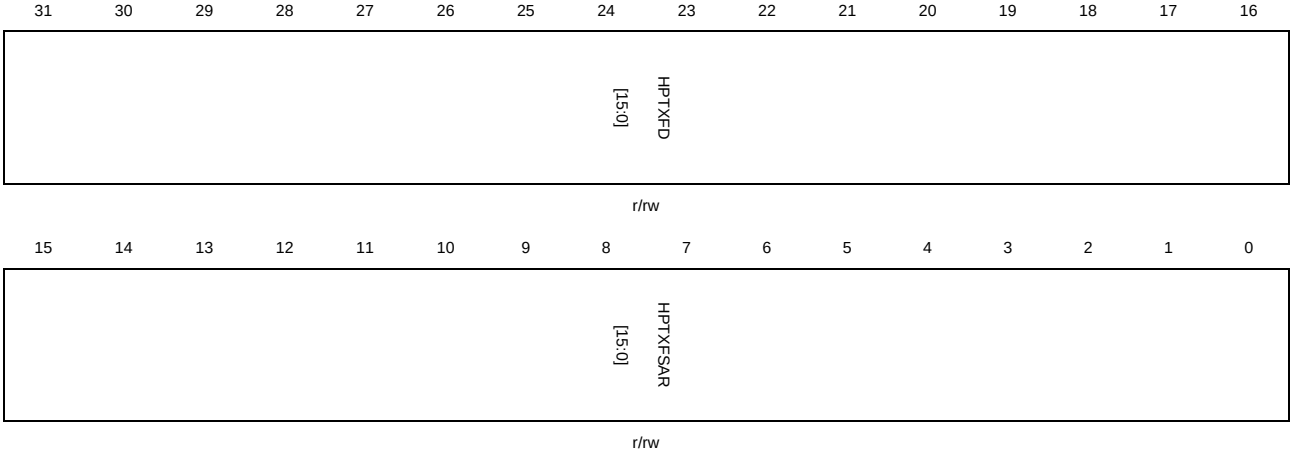
11：32 ms

主机周期性发送 FIFO 长度寄存器（USBHS_HPTFLEN）

地址偏移：0x0100

复位值：0x0200 0600

该寄存器只能按字（32 位）访问



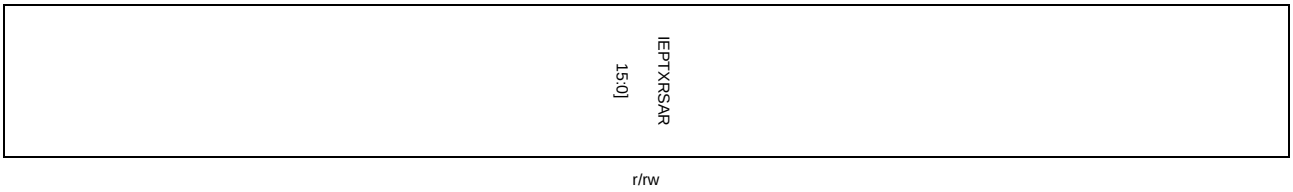
位/位域	名称	描述
31:16	HPTXFD[15:0]	主机周期性Tx FIFO深度 以32位字计数 1≤HPTXFD≤1024
15:0	HPTXFSAR[15:0]	主机周期性Tx RAM起始地址 主机周期性发送FIFO RAM起始地址

设备 IN 端点发送 FIFO 长度寄存器（USBHS_DIEPxTFLEN）（x = 1..3， 其中 x 为 FIFO 编号）

地址偏移：0x0104 + (FIFO 编号 － 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 以32位字计数 1≤HPTXFD≤1024
15:0	IEPTXRSAR[15:0]	IN端点FIFOx Tx RAM起始地址 以32位字为单位的IN端点发送FIFOx起始地址

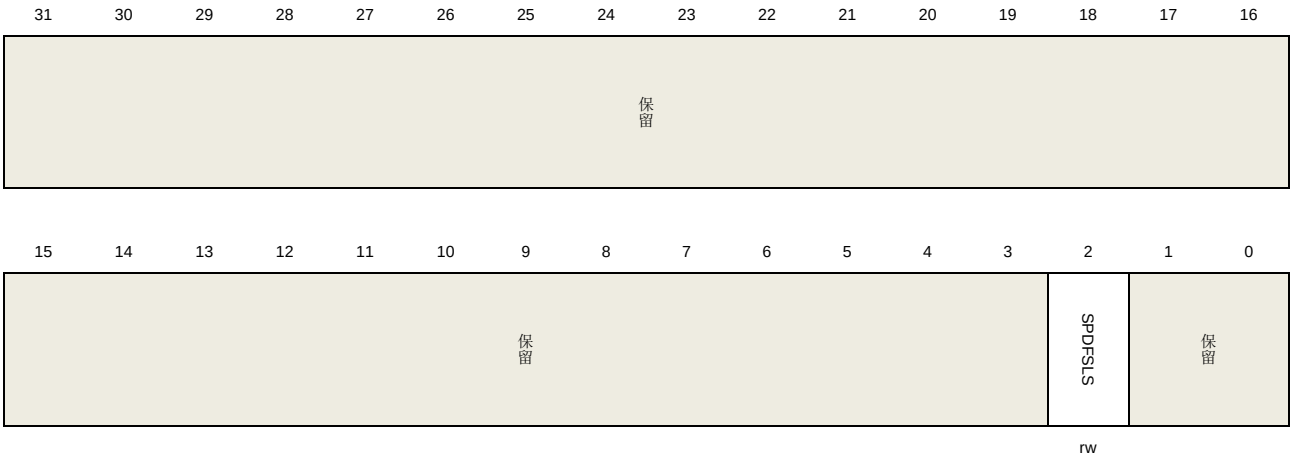
27.7.2. 主机控制和状态寄存器

主机控制寄存器（USBHS_HCTL）

地址偏移：0x0400
复位值：0x0000 0000

在主机模式下，上电后，该寄存器有 USB 内核配置。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
31:3	保留	必须保持复位值
2	SPDFSLS	限制速度为FS和LS 软件可以利用该控制位限制USBHS的枚举速度为FS/LS，并且使USBHS在复位的过程中不执行高速枚举。 0：不限制速度 1：限制速度仅为FS/LS

1:0	保留	必须保持复位值

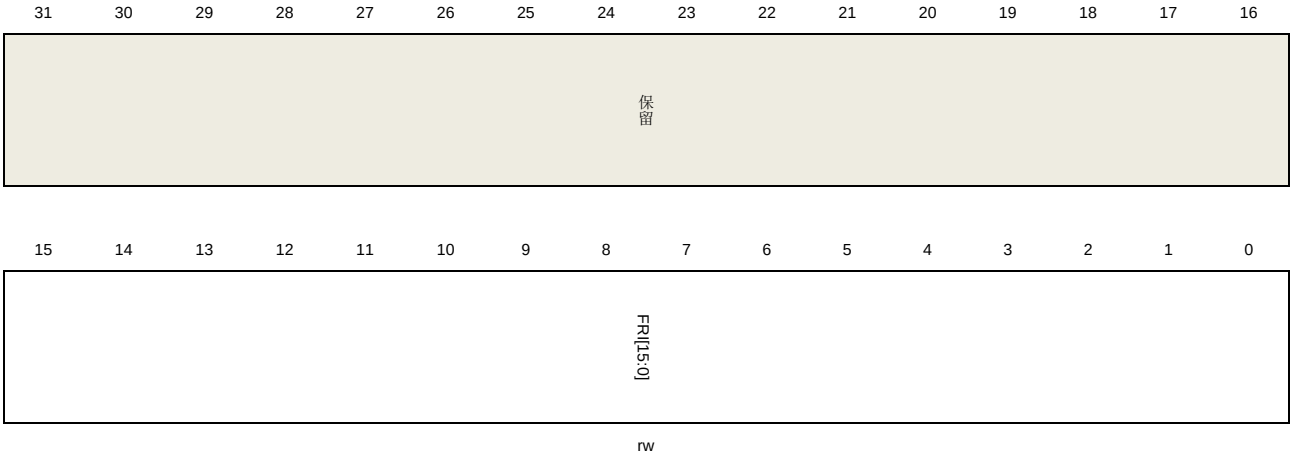
主机帧间隔寄存器（USBHS_HFT）

地址偏移：0x0404

复位值：0x0000 BB80

当 USBHS 控制器正在枚举中时，该寄存器为当前枚举速度设置帧间隔。

该寄存器只能按字（32 位）访问



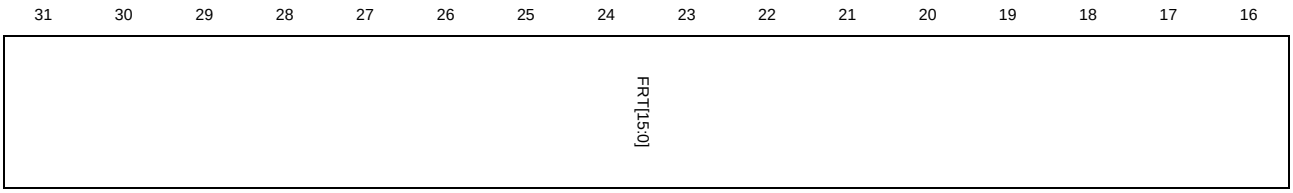
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	FRI[15:0]	帧间隔 该值描述了以PHY时钟为单位的帧周期。每次端口复位操作后，端口被使能，USBHS根据当前速度，采用一个固有值，并且软件可以向该位域写值以改变该固有值。该值需要采用以下描述的频率来进行计算： 内部嵌入式PHY 高速：60MHz 全速/低速：48MHz 外部ULPI PHY 60MHz

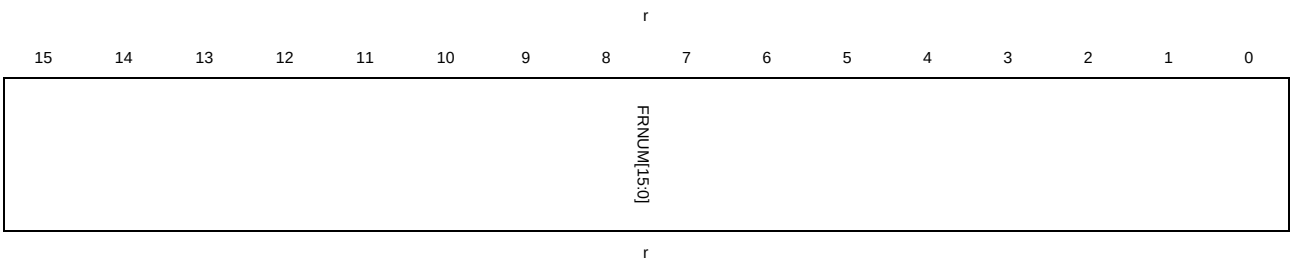
主机帧信息保持寄存器（USBHS_HFINFR）

地址偏移：0x0408

复位值：0xBB80 0000

该寄存器只能按字（32 位）访问





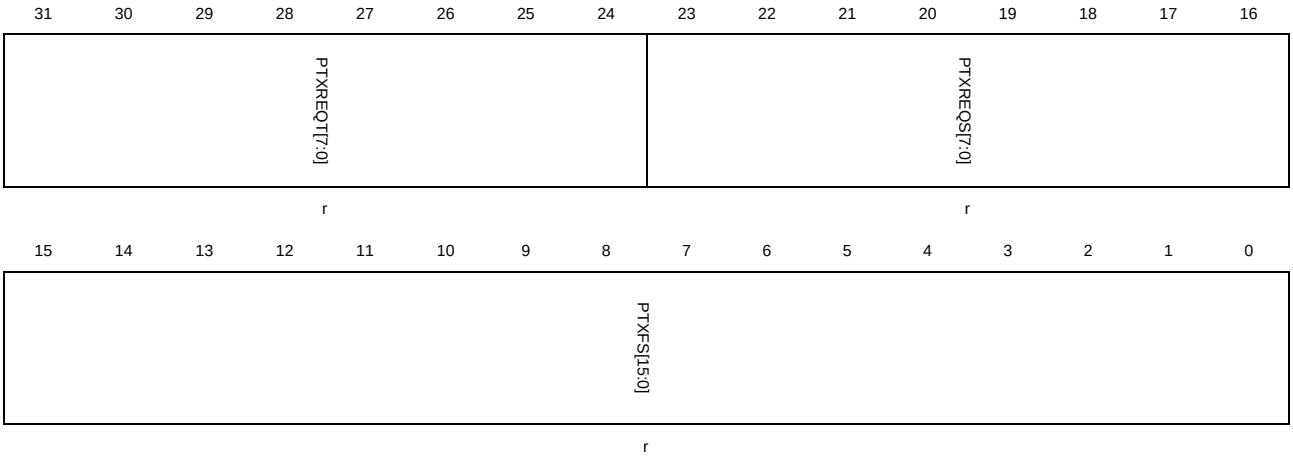
位/位域	名称	描述
31:16	FRT[15:0]	帧剩余时间 该位域以PHY时钟为单位反映了当前帧剩余时间。
15:0	FRNUM[15:0]	帧号 该位域反映了当前帧的帧号，当其增加到0x3FFF后，其值变为0。

主机周期性发送 FIFO/队列状态寄存器（USBHS_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Tx 请求队列的顶部条目 在周期性发送请求队列中的条目 位31：奇偶帧 – 0：奇数帧 – 1：偶数帧 位30:27：通道号 位26:25： – 00：IN/OUT 令牌

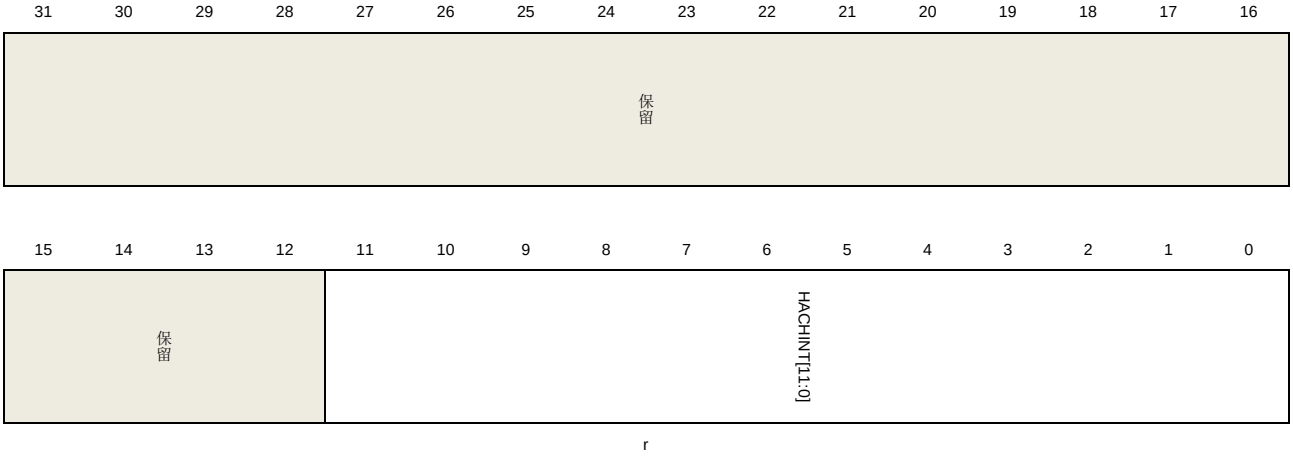
		– 01：0长度OUT包 – 11：通道中止请求 位24：中止标志，指示所选通道的最后一个条目
23:16	PTXREQS[7:0]	周期性发送请求队列空间 周期性发送请求队列剩余空间 0：请求队列为空 1：1个条目 2：2个条目 … n：n个条目 (0≤n≤8) 其他：保留
15:0	PTXFS[15:0]	周期性发送FIFO空间 周期性发送FIFO剩余空间 以32位字计数 0：周期性发送FIFO为空 1：1个字 2：2个字 … n：n个字（0≤n≤PTXFD)） 其他：保留

主机所有通道中断寄存器（USBHS_HACHINT）

地址偏移：0x0414
复位值：0x0000 0000

当触发一个通道中断时，USBHS 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

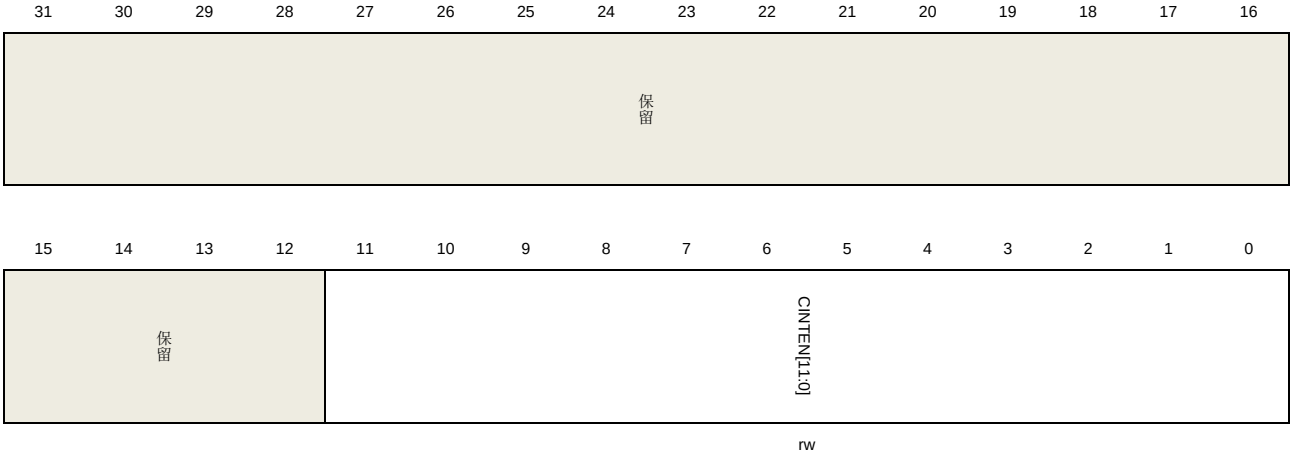
31:12	保留	必须保持复位值
11:0	HACHINT[11:0]	主机所有通道中断 每一位表示一个通道：位0代表通道0，位1表示通道1

主机所有通道中断使能寄存器（USBHS_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。只有该寄存器中相应通道的中断使能控制位被置位，USBHS_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
31:12	保留	必须保持复位值
11:0	CINTEN	通道中断使能 0：禁用通道n中断 1：使能通道n中断 每一位表示一个通道：位0代表通道0，位11代表通道11

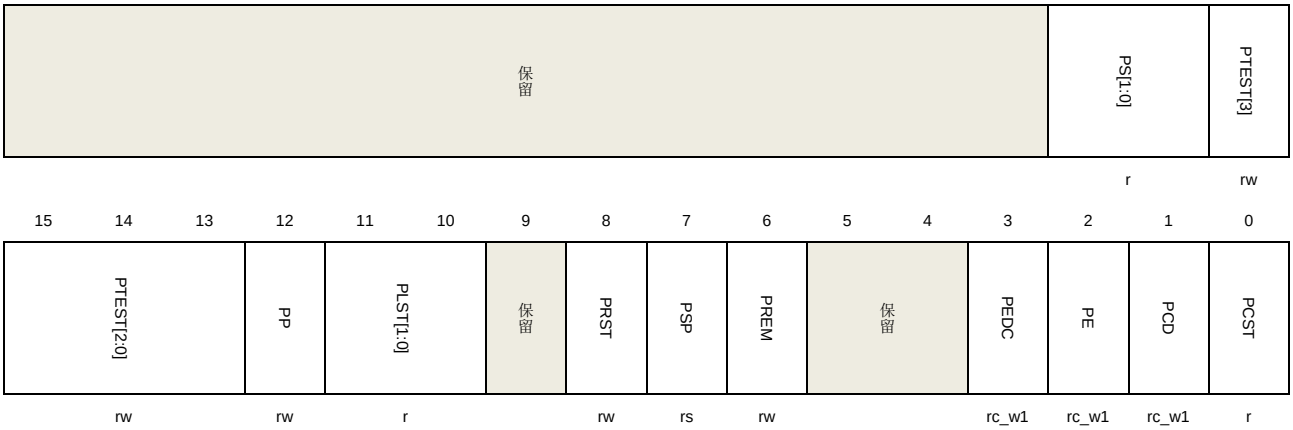
主机端口控制和状态寄存器（USBHS_HPCS）

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 USBHS 置位的话，USBHS_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

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



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 反映连接到该端口的设备的枚举速度。 00：高速 01：全速 10：低速 其他：保留
16:13	PTEST[3:0]	端口测试控制 软件向该字段写入一个非零值以使端口进入测试模式，相应的模式发送到端口上。使用测试模式时，还应设置USBHS_GUSBCS寄存器中的HS_CUR_FE位。 0000：测试模式禁止 0001：Test_J模式 0010：Test_K模式 0011：Test_SE0_NAK模式 0100：Test_Packet模式 0101：强制测试使能 其他：保留
12	PP	端口供电 在端口被使用后，该控制位应该被置位。由于USBHS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。 0：端口掉电 1：端口供电
11:10	PLST[1:0]	端口线状态 反映USB数据线当前状态 位10：DP线状态 位11：DM线状态
9	保留	必须保持复位值

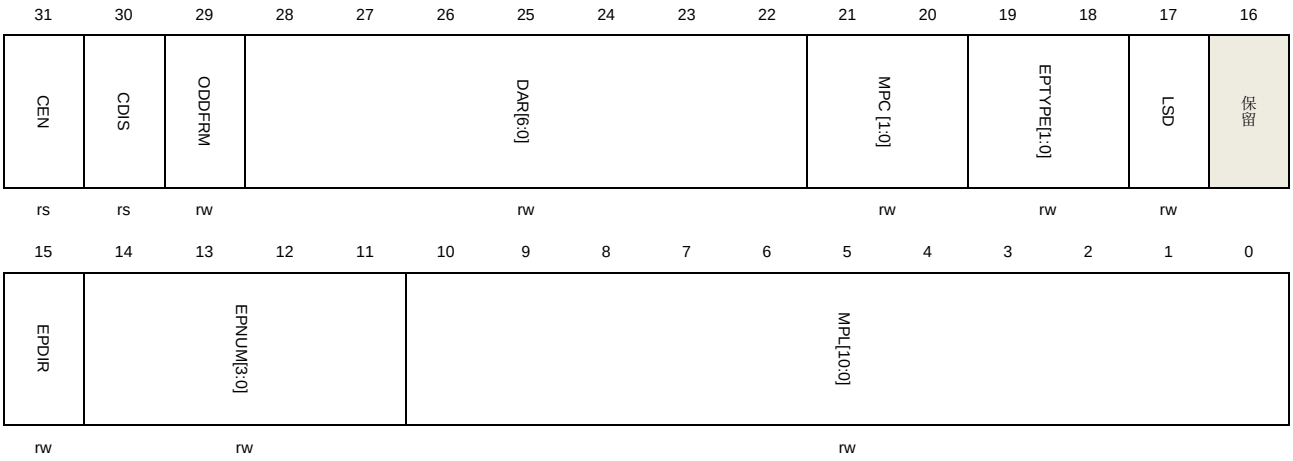
8	PRST	端口复位 应用通过设置该控制位以在USB端口上启动一个复位信号。当应用希望停止复位信号时，应用应该清除该控制位。 0：端口不在复位状态 1：端口处于复位状态
7	PSP	端口挂起 应用设置该控制位来将端口进入挂起状态。当该控制位被置位后，端口停止发送SOF令牌包。该控制位只能够通过以下操作清除。 – 应用置位该寄存器中的PRST控制位 – 置位该寄存器中的PREM控制位 – 检测到一个远程唤醒信号 – 检测到一个设备断开 0：端口不在挂起状态 1：端口处于挂起状态
6	PREM	端口恢复 应用通过置位该控制位以在USB端口上启动一个恢复信号。当应用希望停止恢复信号时，应用可以清除该控制位。 0：无恢复驱动 1：恢复驱动 当应用程序在睡眠状态将PREM置1时，内核继续驱动恢复信号，直到计时器计数到BESLTH指定的时间。当内核检测到USB远程唤醒时，开始驱动恢复信号，并在恢复结束时自动将其清除。
5:4	保留	必须保持复位值
3	PEDC	端口使能/禁止更改 当该寄存器中的位2端口使能控制位更改时，USB内核置位该标志位。
2	PE	端口使能 当USB复位信号完成后，USBHS自动置位该位，并且该位不可由软件置位。 该位可通过以下事件清除： – 一个断开状态 – 软件清除该位 0：端口禁止 1：端口使能
1	PCD	端口连接检测 当检测到设备连接时，USBHS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 0：设备没有连接到该端口 1：设备连接到该端口

主机通道 x 控制寄存器（USBHS_HCHxCTL）（x = 0..11， 其中 x 为主机通道号）

地址偏移：0x0500 + （主机通道号 × 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 由应用设置，并且由USBHS清除 0：通道禁止 1：通道使能 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 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 对于周期性传输（中断或同步传输），该位控制将要处理的通道事务为奇数帧还是偶数帧。
28:22	DAR[6:0]	设备地址 与该通道通信的USB设备地址。
21:20	MPC[1:0]	多包计数 对于周期性传输，该位域指定主机每个微帧必须执行的事务数量。对于非周期性传输，该位域指定在内部DMA引擎更改仲裁之前，DMA为此通道获取或写入的包数量。 00：保留 01：每微帧发出1个事务 10：每微帧发出2个事务 11：每微帧发出3个事务
19:18	EPTYPE[1:0]	端点类型 与该通道通信的端点的传输类型 00：控制 01：同步 10：批量

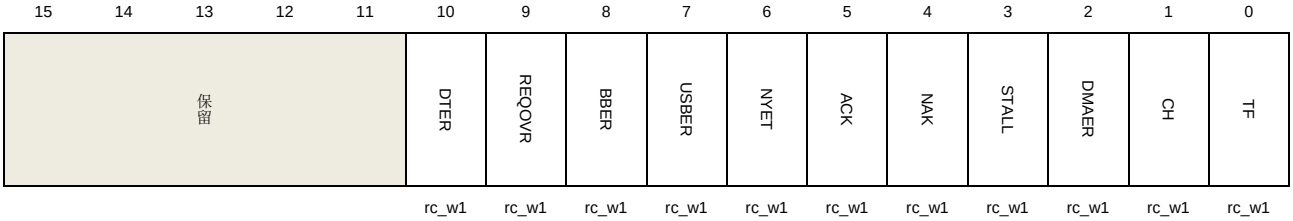
		11：中断
17	LSD	低速设备 与该通道通信的设备是一个低速设备。
16	保留	必须保持复位值
15	EPDIR	端点方向 与该通道通信的端点的传输方向 0：OUT 1：IN
14:11	EPNUM[3:0]	端点号 与该通道通信的端点号
10:0	MPL	最大包长 目标端点的最大包长

主机通道 x 中断标志寄存器（USBHS_HCHxINTF）（x = 0..11， 其中 x 为主机通道号）

地址偏移：0x0508 + （主机通道号 × 0x20）
复位值：0x0000 0000

该寄存器包含一个通道的状态和事件，当软件获取一个通道中断时，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



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 IN事务获取一个数据包，但是该包的PID和USBHS_HCHxLEN寄存器中的DPID[1:0]控制位不匹配。
9	REQOVR	请求队列上溢

		当软件启动新的传输时，请求队列上溢。
8	BBER	串扰错误 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 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： – 接收包有一个错误的CRC域 – 在USB总线上检测到填充错误 – 当等待一个响应包时，超时
6	NYET	NYET 接收到一个NYET响应包（在高速模式下）
5	ACK	ACK 接收或者发送一个ACK响应包
4	NAK	NAK 接收到一个NAK响应包
3	STALL	STALL 接收到一个STALL响应包
2	DMAER	DMA 错误 当DMA尝试为当前通道获取或写如包数据时，产生一个错误
1	CH	通道中止 当DMA未被使能时： 通道被当前请求禁用。 当DMA使能时： 通道被DMA禁用，可能由于该通道的所有事务成功完成或者发生一个USB错误。
0	TF	发送完成 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USBHS_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RxFIFO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（USBHS_HCHxINTEN）（x = 0..11， 其中 x 为主机通道号）

地址偏移：0x050C + （主机通道号 × 0x20）
复位值：0x0000 0000

该寄存器包含 USBHS_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，USBHS_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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					DTERIE	REQOVRIE	BBERIE	USBERIE	NYETIE	ACKIE	NAKIE	STALLIE	DMAERIE	CHIE	TFIE
					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	DTERIE	数据切换错误中断使能 0：禁用数据切换错误中断 1：使能数据切换错误中断
9	REQOVRIE	请求队列上溢中断使能 0：禁用请求队列上溢中断 1：使能请求队列上溢中断
8	BBERIE	串扰错误中断使能 0：禁用串扰错误中断 1：使能串扰错误中断
7	USBERIE	USB总线错误中断使能 0：禁用USB总线错误中断 1：使能USB总线错误中断
6	NYETIE	NYET中断使能 0：禁用NYET中断 1：使能NYET中断
5	ACKIE	ACK中断使能 0：禁用ACK中断 1：使能ACK中断
4	NAKIE	NAK中断使能 0：禁用NAK中断 1：使能NAK中断
3	STALLIE	STALL中断使能 0：禁用STALL中断 1：使能STALL中断
2	DMAERIE	DMA 错误中断使能

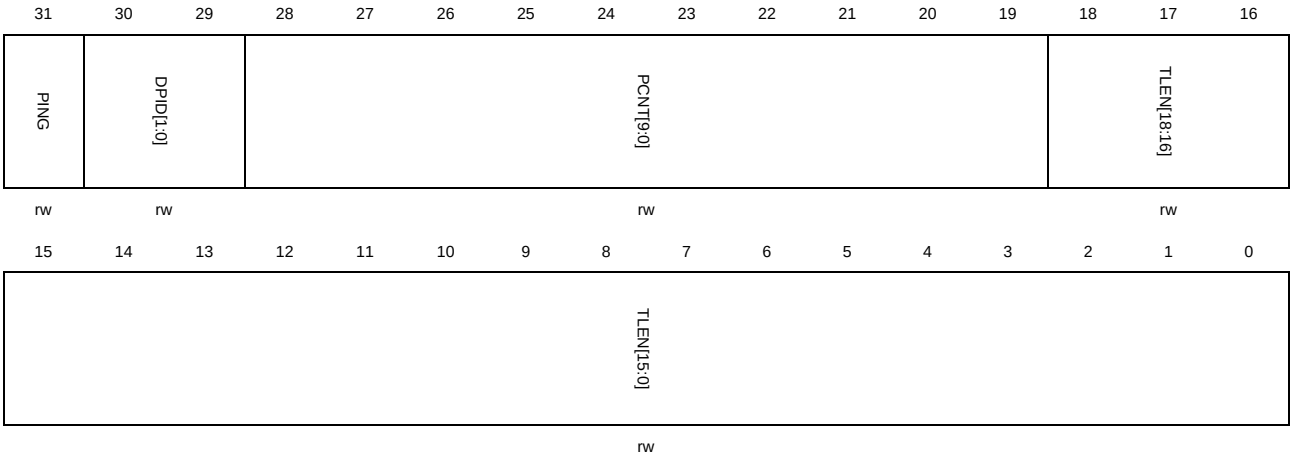
		0：禁用DMA错误中断 1：使能DMA错误中断
1	CHIE	通道中止中断使能 0：禁用通道中止中断 1：使能通道中止中断
0	TFIE	传输完成中断使能 0：禁用传输完成中断 1：使能传输完成中断

主机通道 x 长度寄存器（USBHS_HCHxLEN）（x = 0..11，其中 x 为主机通道号）

地址偏移：0x0510 + （主机通道号 × 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	PING	PING 令牌请求 对于OUT传输，如果软件置位该控制位，USBHS会执行PING协议。当OUT事务接收到一个NAK或NYET握手包时，USBHS会自动置位该控制位。不要为IN传输置位该控制位。
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，USBHS遵循USB协议自动改变和切换该位域。 00：DATA0 01：DATA2 10：DATA1 11：MDATA（非控制）/SETUP（控制）
28:19	PCNT[9:0]	包计数

		在一个传输中希望发送（OUT）或接收（IN）的数据包个数。 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USBHS正确传输每个数据包后，自动减少。
18:0	TLEN[18:0]	传输长度 一次传输的总数据字节数。 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。当软件或DMA正确向通道的数据FIFO中写入一个包时，该位域以包中字节大小进行减少。 对于IN传输，每次软件或DMA从RxFIFO中读取一个包后，该位域也以包中字节大小进行减少。

主机通道 x DMA 地址寄存器（USBHS_HCHxDMAADDR）（x = 0..11， 其中 x 主机通道号）

地址偏移：0x0514 + （主机通道号 × 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
DMAADDR[31:16]															
rw															

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

位/位域	名称	描述
31:0	DMAADDR[31:0]	DMA地址 该位域定义了端点DMA地址。DMA使用该地址来为该通道取出或写入包数据。

27.7.3. 设备控制和状态寄存器

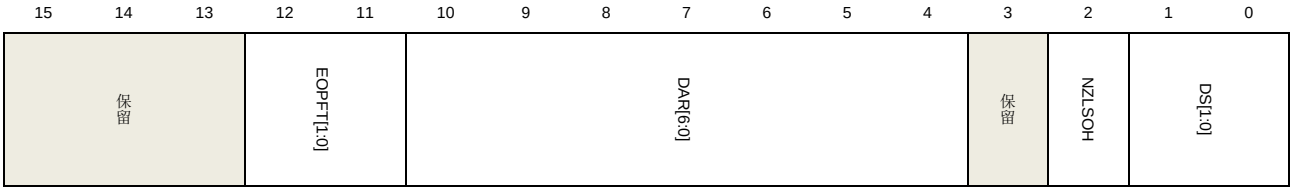
设备配置寄存器（USBHS_DCFG）

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



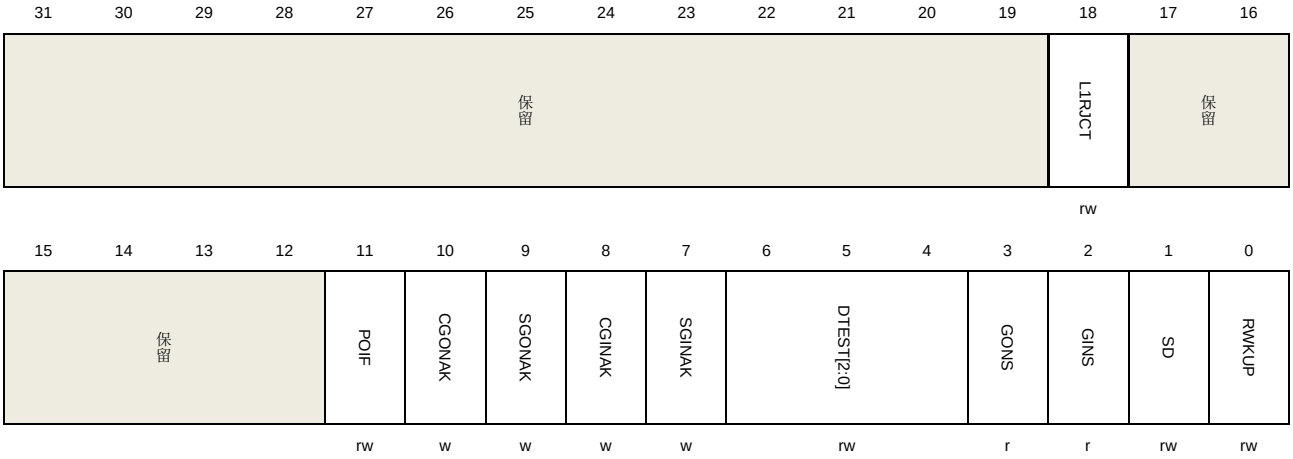
位/位域	名称	描述
31:13	保留	必须保留为复位值
12:11	EOPFT[1:0]	周期性帧尾时间 该域定义周期性帧时间的帧尾标志触发的时间点 00：80%的帧时间 01：85%的帧时间 10：90%的帧时间 11：95%的帧时间
10:4	DAR[6:0]	设备地址 该位定义USB设备地址，USBHS采用该位匹配接收的设备令牌地址域，在接收到来自主机的设置地址的命令后，软件设置该域
3	保留	必须保留为复位值
2	NZLSOH	非零长度OUT状态阶段握手 在控制传输的OUT状态阶段，当USB设备接收到一个非零长度数据包时，该域控制控制USBHS是接收该包，还是用STALL握手信号拒绝该包。 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 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 该域控制设备连入主机后的设备速度 00：高速 01：全速 其他：保留

设备控制寄存器（USBHS_DCTL）

地址偏移：0x0804

复位值：0x0000 0000

该寄存器采用字（32 位）访问



位/位域	名称	描述
31:19	保留	必须保留为复位值
18	L1RJCT	禁止深度睡眠 当这一位被设置，内核以NYET响应LPM事务，同时BESL大于BSELTH
17:12	保留	必须保留为复位值
11	POIF	上电初始化完成 软件通过设置该位，通知USBHS寄存器在从掉电模式下唤醒，然后完成初始化。
10	CGONAK	清零全局OUT NAK 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 软件设置该位从而实现该寄存器的位GONS置位。 当GONS位为零，设置该位会引起USBHS_GINTF寄存器的GONAK标志触发，软件应该在再写该位前清除GONAK标志。
8	CGINAK	清零全局IN NAK 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 软件设置该位从而实现该寄存器的位GINS置位 当GINS位为零，设置该位会引起USBHS_GINTF寄存器的GINAK标志触发，软件应该在再写该位前清除GINAK标志。
6:4	DTEST[2:0]	设备测试控制 软件向该字段写入一个非零值以使端口进入测试模式，相应的模式发送到端口上。使用测试模式时，还应设置USBHS_GUSBCS寄存器中的HS_CUR_FE位。 0000：测试模式禁止 0001：Test_J模式 0010：Test_K模式 0011：Test_SE0_NAK模式

0100：Test_Packet模式
0101：强制测试使能
其他：保留

3	GONS	全局OUT NAK状态 0：USBHS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NAKS、STALL位确定。 1：USBHS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 0：USBHS回复IN事务的握手信号由Tx FIFO状态、端点的NAKS、STALL位确定。 1：USBHS通常回复IN事务NAK握手信号
1	SD	软断开 软件可实现USB总线上的软断开，在置1该位后，如果当前是高速模式，USBHS先退回到全速模式，然后在关掉DP线上的上拉电阻，从而引起主机检测设备的断开。 0：没有软断开生成 1：生成软断开
0	RWKUP	远程唤醒 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 0：没有远程唤醒信号生成 1：生成远程唤醒信号 当核心已启用LPM且处于睡眠状态时，如果设置了此位，则核心将继续驱动它，并在50us后自动清除它（TL1DevDrvResume）。当从LPM事务接收的bRemoteWake值为零时，应用程序无法设置此位。

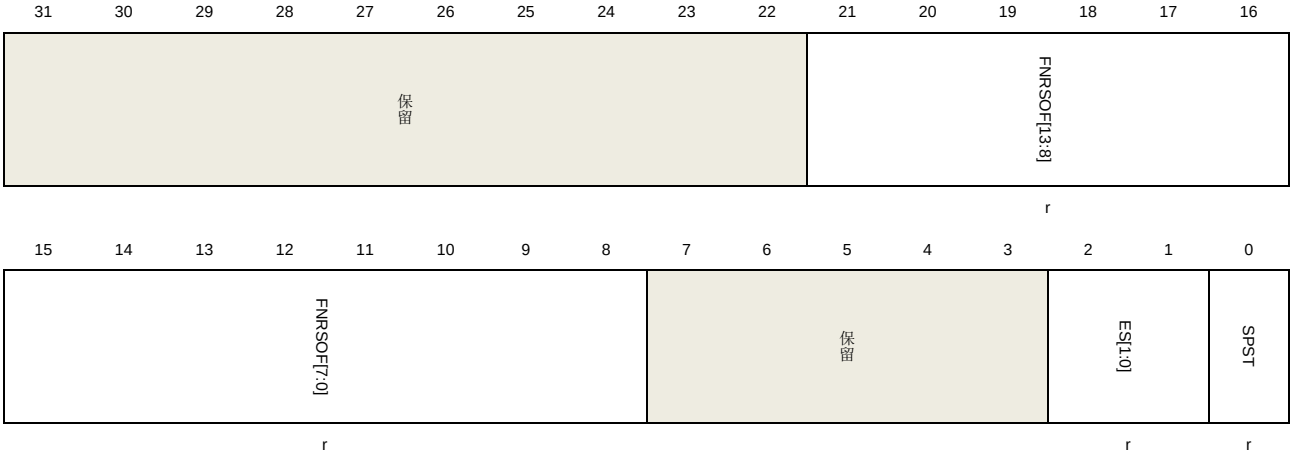
设备状态寄存器（USBHS_DSTAT）

地址偏移：0x0808

复位值：0x0000 0000

该寄存器包含设备模式下的 USBHS 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	必须保留为复位值
21:8	FNRSOF[13:0]	所接收的SOF帧编号 USBHS会在接收到一个SOF令牌后更新该域。
7:3	保留	必须保留为复位值
2:1	ES[1:0]	枚举速度 该域指示所枚举的设备速度，在寄存器USBHS_GINTF的ENUMF标志触发后，软件可以读取该域。 00：高速 01：全速 其他：保留
0	SPST	挂起状态 该位指示设备是否处于挂起状态。 0：设备不在挂起状态 1：设备在挂起状态

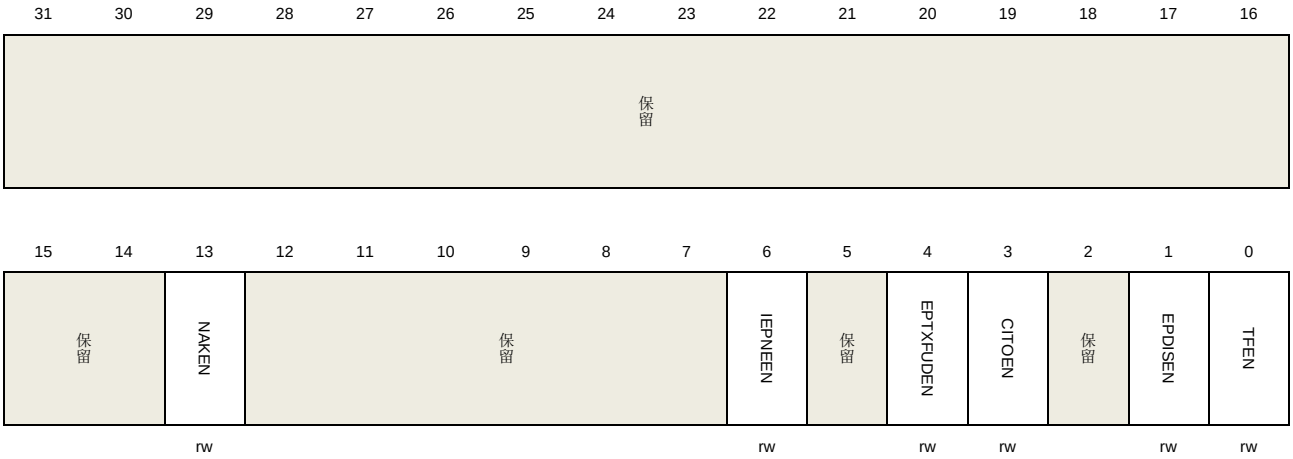
设备 IN 端点通用中断使能寄存器（USBHS_DIEPINTEN）

地址偏移：0x810

复位值：0x0000 0000

该寄存器包含寄存器 USBHS_DIEPxINTF 中的标志的中断使能位，如果软件置 1 某位，其在寄存器 USBHS_DIEPxINTF 中对应的位可以触发一个寄存器 USBHS_DAEPINT 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
31:14	保留	必须保留为复位值
13	NAKEN	发送NAK握手中断使能位 0：除能中断

		1：使能中断
12:7	保留	必须保留为复位值
6	IEPNEEN	IN端点NAK有效中断使能位 0：除能中断 1：使能中断
5	保留	必须保留为复位值
4	EPTXFUDEN	端点Tx FIFO下溢中断使能位 0：除能中断 1：使能中断
3	CITOEN	控制IN事务超时中断使能位 0：除能中断 1：使能中断
2	保留	必须保留为复位值
1	EPDISEN	端点除能中断使能位 0：除能中断 1：使能中断
0	TFEN	传输完成中断使能位 0：除能中断 1：使能中断

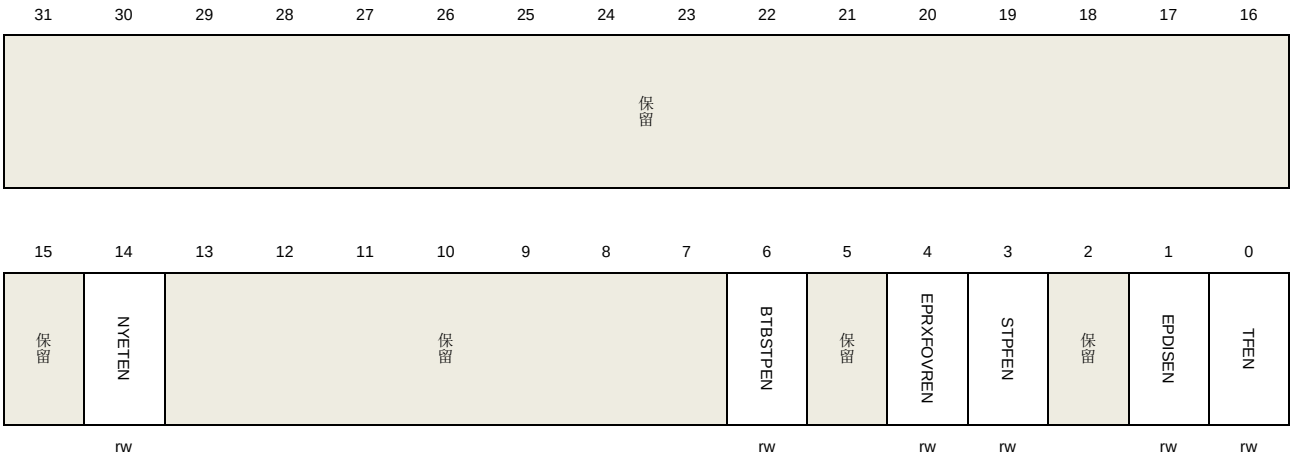
设备 OUT 端点通用中断使能寄存器（USBHS_DOEPINTEN）

地址偏移：0x0814

复位值：0x0000 0000

该寄存器包含寄存器 USBHS_DOEPxINTF 中的标志的中断使能位，如果软件置 1 某位，其在寄存器 USBHS_DOEPxINTF 中对应的位可以触发一个寄存器 USBHS_DAEPINT 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
31:15	保留	必须保留为复位值
14	NYETEN	发送NYET握手中断使能位 0：除能中断 1：使能中断
13:7	保留	必须保留为复位值
6	BTBSTPEN	连续SETUP包中断使能位（仅适用于控制OUT端点） 0：除能中断 1：使能中断
5	保留	必须保留为复位值
4	EPRXFOVREN	端点Rx FIFO上溢中断使能位 0：除能中断 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） 0：除能中断 1：使能中断
2	保留	必须保留为复位值
1	EPDISEN	端点除能中断使能位 0：除能中断 1：使能中断
0	TFEN	传输完成中断使能位 0：除能中断 1：使能中断

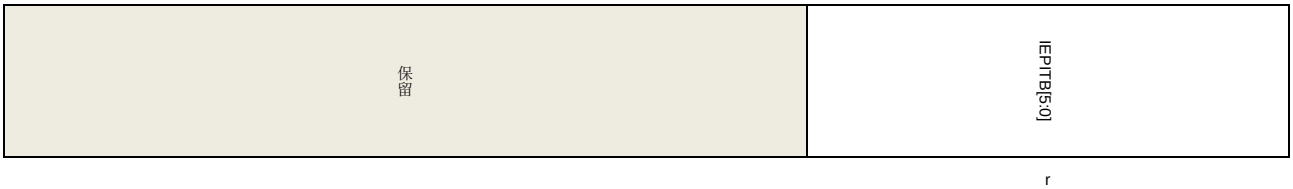
设备端点中断寄存器（USBHS_DAEPINT）

地址偏移：0x0818

复位值：0x0000 0000

当一个端点的中断被触发，USBHS 置 1 该寄存器的相应位，软件可通过该寄存器知道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
31:22	保留	必须保留为复位值
21:16	OEPITB[5:0]	设备OUT端点中断位 每个位代表一个OUT端点：Bit16代表OUT端点0，Bit21代表OUT端点5
15:6	保留	必须保留为复位值
5:0	IEPITB[5:0]	设备IN端点中断位 每个位代表一个IN端点：Bit0代表IN端点0，Bit5代表IN端点5

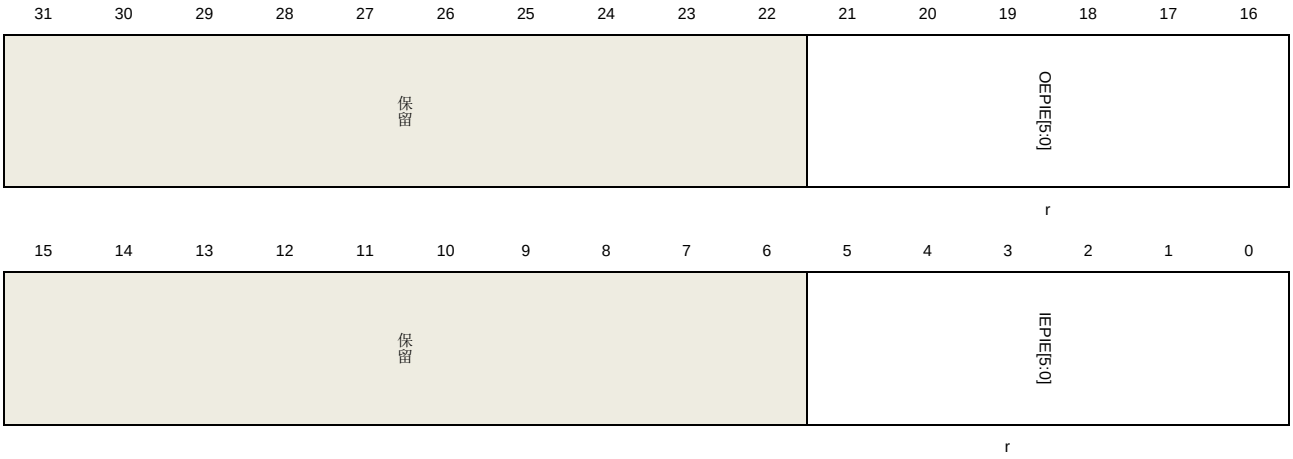
设备端点中断使能寄存器（USBHS_DAEPINTEN）

地址偏移：0x081C

复位值：0x0000 0000

该寄存器可通过软件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 USBHS_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
31:22	保留	必须保留为复位值
21:16	OEPIE[5:0]	OUT端点中断使能位 0：除能OUT端点n中断 1：使能OUT端点n中断 每个位代表一个OUT端点：Bit16对应OUT端点0，Bit21对应OUT端点5
15:6	保留	必须保留为复位值

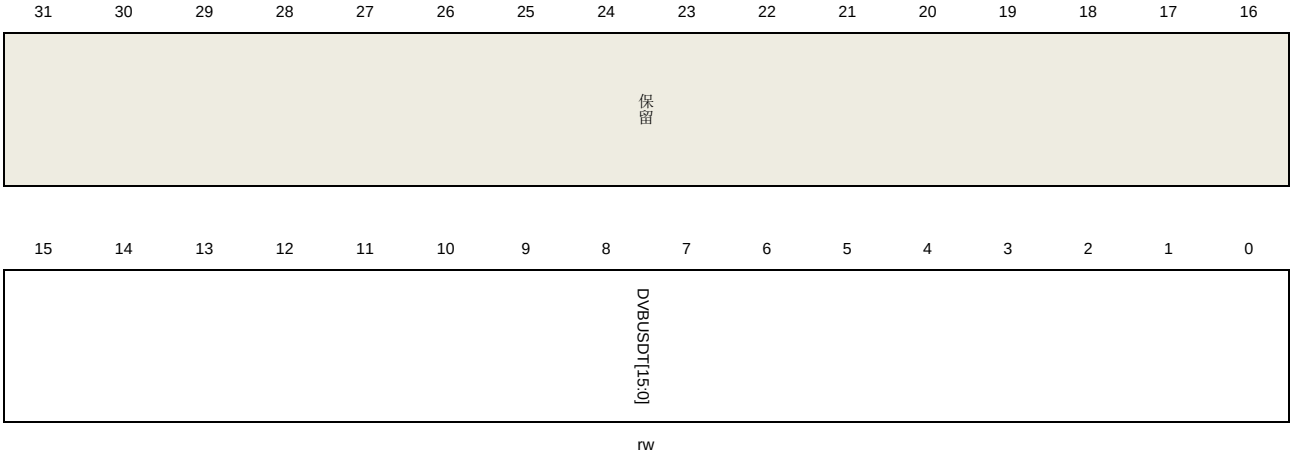
5:0	IEPIE[5:0]	IN端点中断使能位 0：除能IN端点n中断 1：使能IN端点n中断 每个位代表一个IN端点：Bit0对应IN端点0，Bit5对应IN端点5

设备 VBUS 放电时间寄存器（USBHS_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	必须保留为复位值
15:0	DVBUSDT[15:0]	设备V_{BUS}放电时间 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（USBHS_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	必须保留为复位值
11:0	DVBUSPT[11:0]	设备V_{BUS}脉冲时间 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

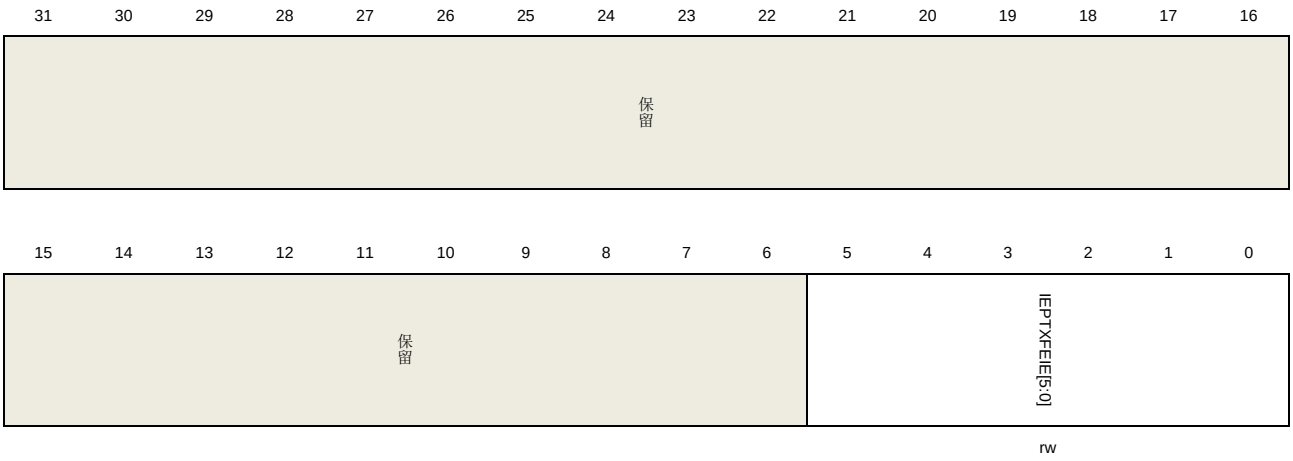
设备 IN 端点 FIFO 空中断使能寄存器（USBHS_DIEPFEINTEN）

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
31:6	保留	必须保留为复位值
5:0	IEPTXFEIE[5:0]	IN端点Tx FIFO空中断的使能位 该域控制着USBHS_DIEPxINTF寄存器的TXFE位能否生成一个寄存器USBHS_DAEPINT的端点中断位 Bit0对应IN端点0，Bit5对应IN端点5 0：除能FIFO空中断 1：使能FIFO空中断

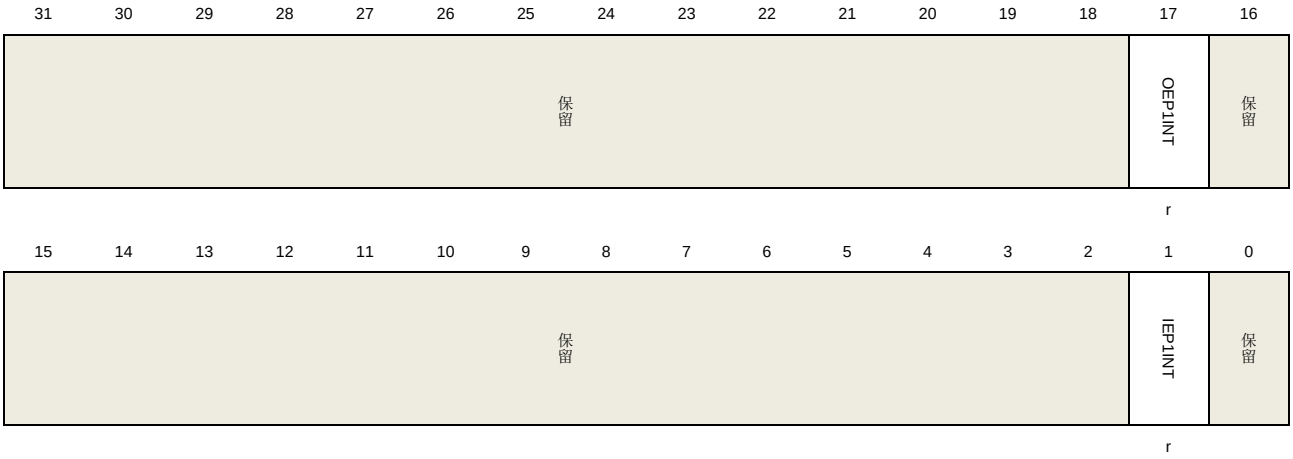
设备端点 1 中断寄存器（USBHS_DEP1INT）

地址偏移：0x0838

复位值：0x0000 0000

当端点 1 OUT 或 IN 被触发，USBHS 置位该寄存器的相应位，软件可通过读取该位知道端点 1 的 IN 或 OUT 被触发。

该寄存器采用字（32 位）访问



位/位域	名称	描述
31:18	保留	必须保留为复位值
17	OEP1INT	OUT端点1中断
16:2	保留	必须保留为复位值
1	IEP1INT	IN端点1中断
0	保留	必须保留为复位值

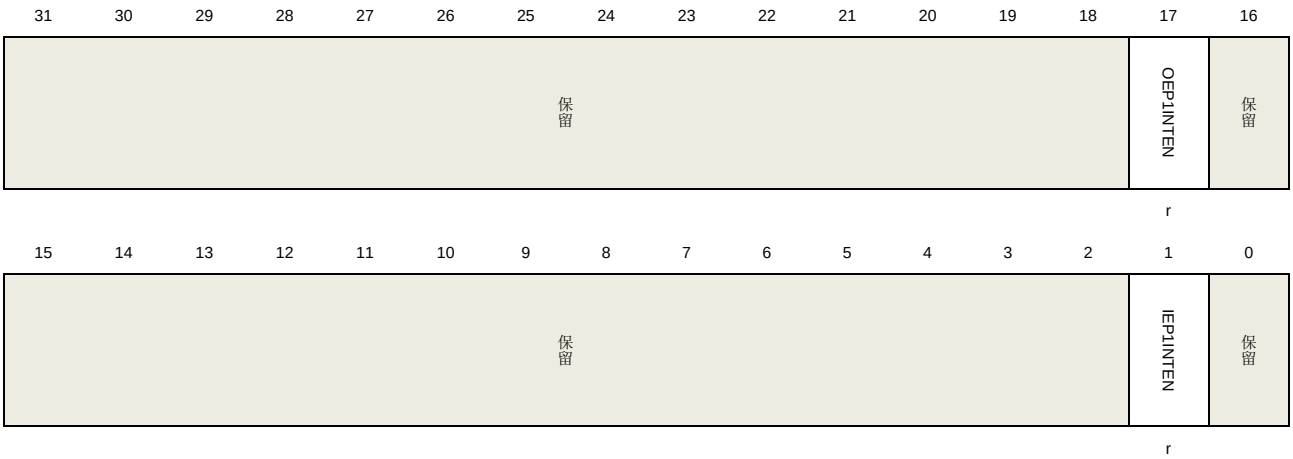
设备端点 1 中断使能寄存器（USBHS_DEP1INTEN）

地址偏移：0x083C

复位值：0x0000 0000

该寄存器可以用软件使能或除能端点 1 的中断，只有该寄存器中的相应位被置位才可以引起端点 1 IN 或者 OUT 的中断。

该寄存器采用字（32 位）访问



位/位域	名称	描述
31:18	保留	必须保留为复位值
17	OEP1INTEN	OUT端点1中断使能
16:2	保留	必须保留为复位值
1	IEP1INTEN	IN端点1中断使能
0	保留	必须保留为复位值

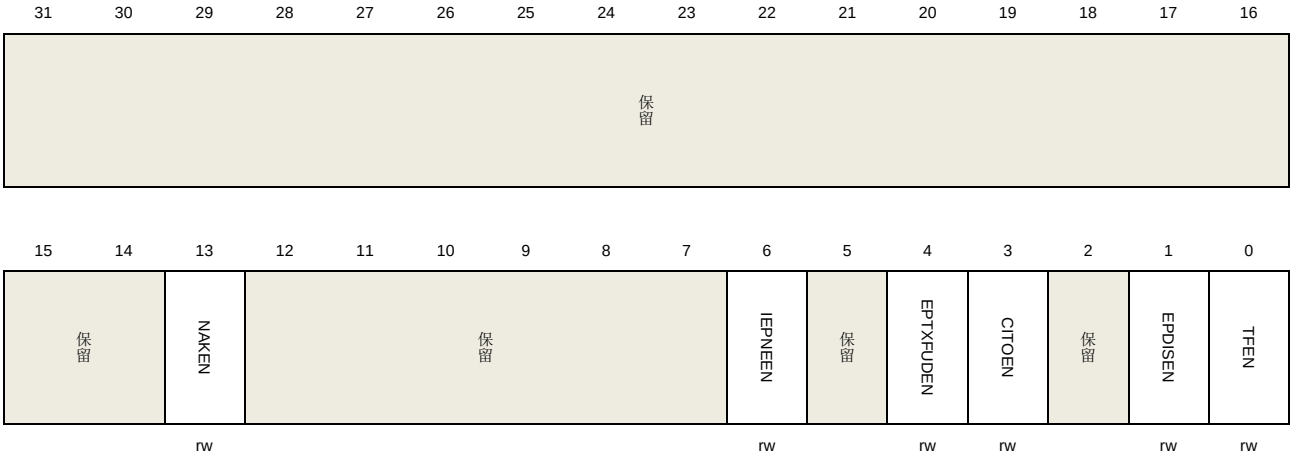
设备 IN 端点 1 中断使能寄存器（USBHS_DIEP1INTEN）

地址偏移：0x844

复位值：0x0000 0000

该寄存器包含对应寄存器 USBHS_DIEP1INTF 标志的中断使能位，如果软件置位该寄存器的某一位，其在寄存器 USBHS_DIEP1INTF 中对应的位会触发一个寄存器 USBHS_DEP1INT 中的端点中断，该寄存器的位可以通过软件置位或清零。

该寄存器采用字（32 位）访问



位/位域	名称	描述
31:14	保留	必须保留为复位值
13	NAKEN	发送NAK握手中断使能位 0：除能中断 1：使能中断
12:7	保留	必须保留为复位值
6	IEPNEEN	IN端点NAK有效中断使能位 0：除能中断 1：使能中断
5	保留	必须保留为复位值
4	EPTXFUD	端点Tx FIFO下溢中断使能位

		0：除能中断 1：使能中断
3	CITOEN	控制传输IN超时中断使能位 0：除能中断 1：使能中断
2	保留	必须保留为复位值
1	EPDISEN	端点除能中断使能位 0：除能中断 1：使能中断
0	TFEN	传输完成中断使能位 0：除能中断 1：使能中断

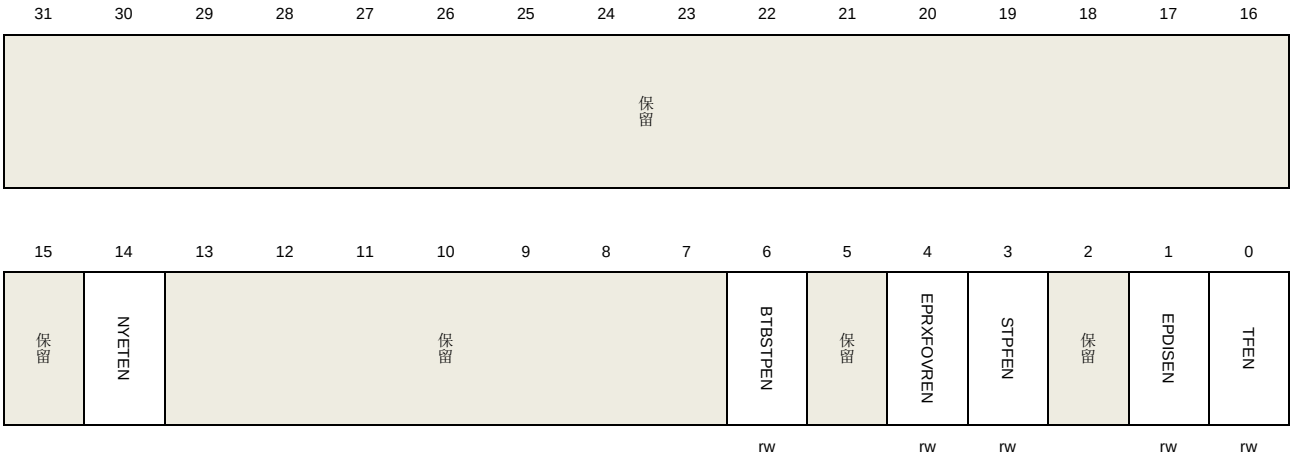
设备 OUT 端点 1 中断使能寄存器（USBHS_DOEP1INTEN）

地址偏移：0x0884

复位值：0x0000 0000

该寄存器包含对应寄存器 USBHS_DOEP1INTF 标志的中断使能位，如果软件置位该寄存器的某一位，其在寄存器 USBHS_DOEP1INTF 中对应的位会触发一个寄存器 USBHS_DEP1INT 中的端点中断，该寄存器的位可以通过软件置位或清零。

该寄存器采用字（32 位）访问



位/位域	名称	描述
31:15	保留	必须保留为复位值
14	NYETEN	发送NYET握手中断使能位 0：除能中断 1：使能中断
13:7	保留	必须保留为复位值
6	BTBSTPEN	连续SETUP包中断使能位（仅适用于控制OUT端点）

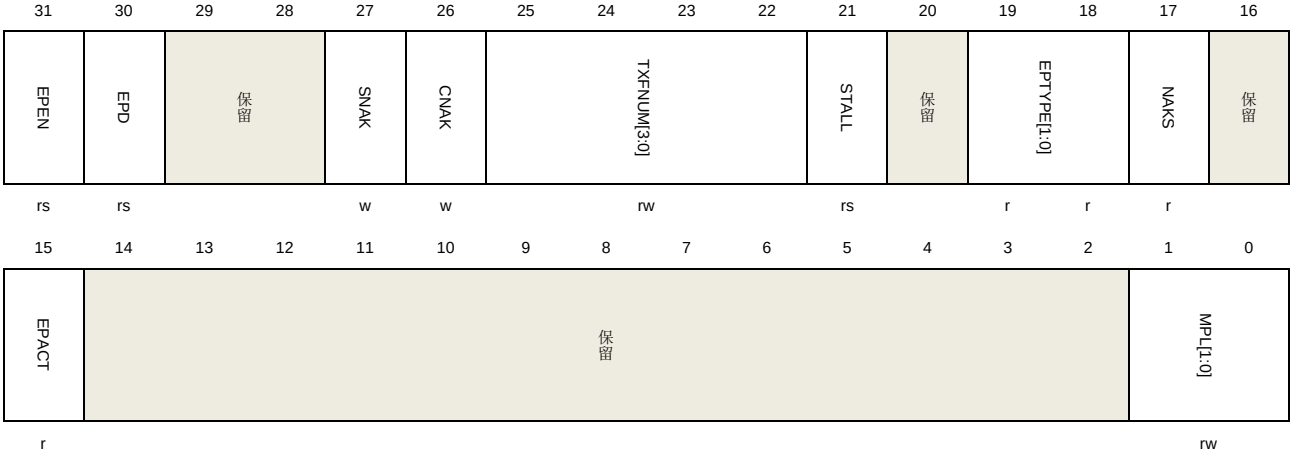
		0：除能中断 1：使能中断
5	保留	必须保留为复位值
4	EPRXFOVREN	端点Rx FIFO上溢中断使能位 0：除能中断 1：使能中断
3	STPFEN	SETUP阶段完成中断使能位（仅适用于控制OUT端点） 0：除能中断 1：使能中断
2	保留	必须保留为复位值
1	EPDISEN	端点除能中断使能位 0：除能中断 1：使能中断
0	TFEN	传输完成中断使能位 0：除能中断 1：使能中断

设备 IN 端点 0 控制寄存器（USBHS_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 软件置位、USBHS清零 0：端点除能 1：端点使能 软件应该按照操作指南使能或除能端点

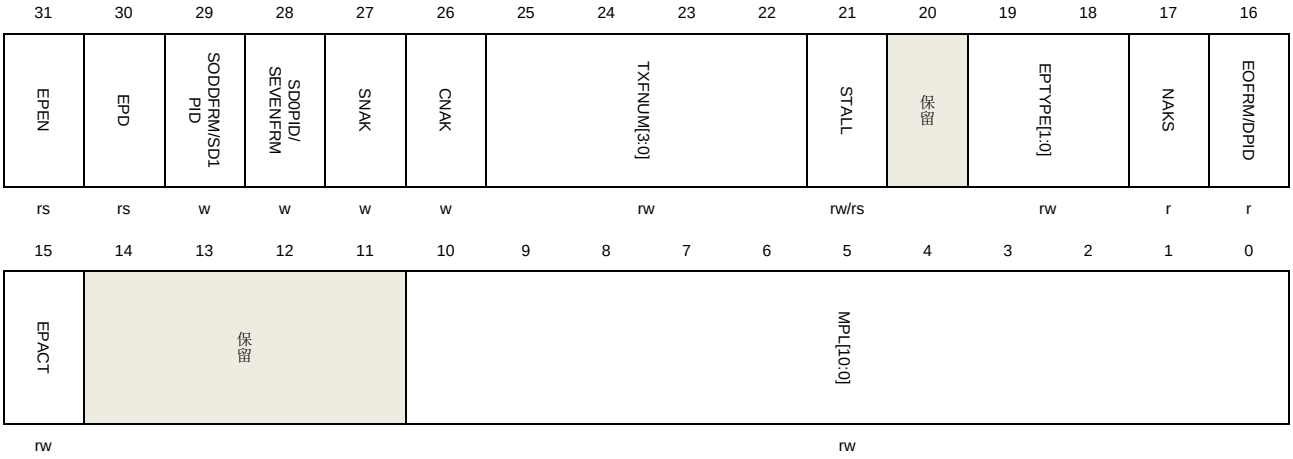
30	EPD	端点除能 软件可通过置位该位从而除能端点，软件应该按照操作指南使能或除能端点。
29:28	保留	必须保留为复位值
27	SNAK	置位NAK 软件置位该位来设置该寄存器的NAKS位
26	CNAK	清零NAK 软件置位该位来清零该寄存器的NAKS位
25:22	TXFNUM[3:0]	Tx FIFO编号 定义IN端点0的Tx FIFO编号
21	STALL	STALL握手 当接收IN令牌时，软件可以通过置1该位发送STALL握手包，对于相应的OUT端点0，在接收SETUP令牌后，USBHS清除此位。该位比该寄存器的NAKS位和寄存器USBHS_DCTL的GINS位优先级要高，如果STALL和NAKS位都被置位，STALL位生效。
20	保留	必须保留为复位值
19:18	EPTYPE[1:0]	端点类型 该域固定为’00’,控制端点。
17	NAKS	NAK状态 当该寄存器的STALL位和寄存器USBHS_DCTL的位GINS被清零，该位控制USBHS的NAK状态。 0：根据端点Tx FIFO的状态，USBHS发送数据或握手包 1：USBHS总为IN令牌发送NAK握手包 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	必须保留为复位值
15	EPACT	端点激活 对于端点0来说，该域固定为‘1’
14:2	保留	必须保留为复位值
1:0	MPL[1:0]	最大包长 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： 00：64字节 01：32字节 10：16字节 11：8字节

设备 IN 端点 x 控制寄存器（USBHS_DIEPxCTL）（x = 1..5， x 是设备端点编号）

地址偏移：0x0900 + (设备端点编号 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 软件置位，USBHS清零 0：端点除能 1：端点使能 软件应该按照操作指南使能或除能端点
30	EPD	端点除能 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） 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) 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) 软件通过置1该位清零该寄存器的EOFRM位
	SD0PID	设置DATA0 PID(适用于中断和大容量IN端点) 软件可通过置1该位清零该寄存器的DPID位
27	SNAK	设置NAK 软件置1该位置1该寄存器的NAKS位
26	CNAK	清零NAK 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 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USBHS_DCTL的GINS位优先级要高，如果STALL和NAKS位都被置位，STALL位生效。

		对于控制IN端点： 当对应的OUT端点接收到SETUP令牌时，只有USBHS可以清零此位，软件不可清除此位。 对于中断或大容量IN端点： 只有软件可以清零此位。
20	保留	必须保留为复位值
19:18	EPTYPE[1:0]	端点类型 该域定义端点的传输类型： 00：控制 01：同步 10：大容量 11：中断
17	NAKS	NAK状态 当该寄存器的STALL位和寄存器USBHS_DCTL的位GINS被清零，该位控制USBHS的NAK状态： 0：根据端点Tx FIFO的状态，USBHS发送数据或握手包 1：USBHS总为IN令牌发送NAK握手包 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） 对于同步传输，软件通过使用该位控制USBHS只在奇数帧或偶数帧为IN事务发送数据包，如果当前帧号的奇偶性不匹配该位，USBHS回复一个零长度的包： 0：只在偶数帧发送数据 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） 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，USBHS在传输过程中保持该位。 0：数据包的PID是DATA0 1：数据包的PID是DATA1
15	EPACT	端点激活 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	必须保留为复位值
10:0	MPL[10:0]	该域定义最大包长

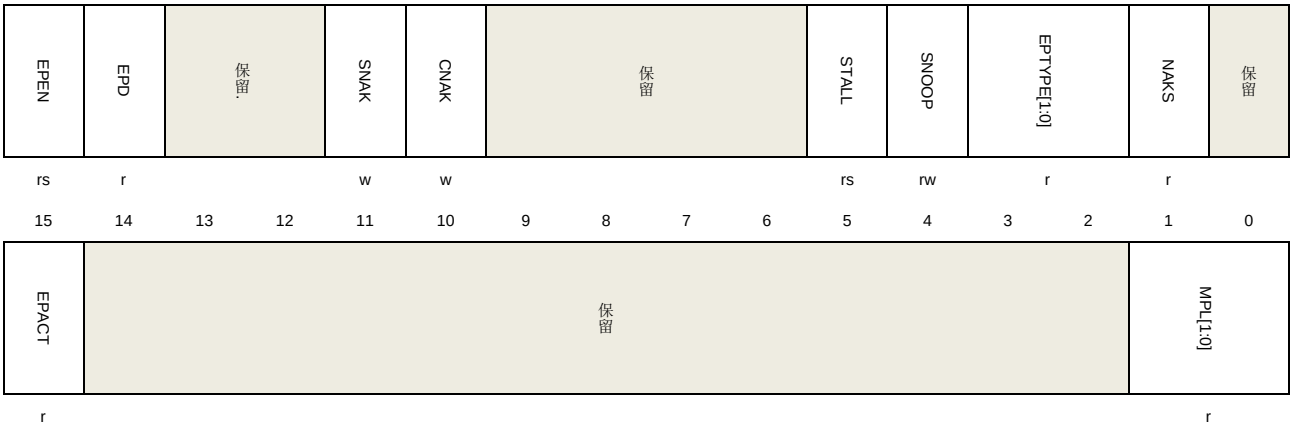
设备 OUT 端点 0 控制寄存器（USBHS_DOEP0CTL）

地址偏移：0x0B00

复位值：0x0000 8000

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



位/位域	名称	描述
31	EPEN	端点使能 软件置位，USBHS清零 0：端点除能 1：端点使能 软件应该按照操作指南使能或除能端点。
30	EPD	端点除能 对于OUT端点0，该位固定为0
29:28	保留	必须保留为复位值
27	SNAK	设置NAK 软件置1该位置1该寄存器的NAKS位
26	CNAK	清零NAK 软件置1该位清零该寄存器的NAKS位
25:22	保留	必须保留为复位值
21	STALL	STALL握手 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，USBHS清除此位。该位比该寄存器的NAKS位和寄存器USBHS_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 该位控制OUT端点的调查模式，在调查模式中，USBHS不再检查接收数据包的CRC值 0：调查模式除能 1：调查模式使能
19:18	EPTYPE[1:0]	端点类型 对于控制端点，该位固定为”00“
17	NAKS	NAK状态 当该寄存器的STALL位和寄存器USBHS_DCTL的位GINS被清零，该位控制

		USBHS的NAK状态： 0：根据端点Rx FIFO的状态，USBHS发送数据或握手包 1：USBHS为OUT事务发NAK握手包 该位是只读位，通过该寄存器的CNAK和SNAK位控制该位
16	保留	必须保留为复位值
15	EPACT	端点激活 对于端点0，该域固定为1
14:2	保留	必须保留为复位值
1:0	MPL[1:0]	最大包长 该位是只读位，其数值来自于寄存器USBHS_DIEP0CTL的位MPL： 00：64字节 01：32字节 10：16字节 11：8字节

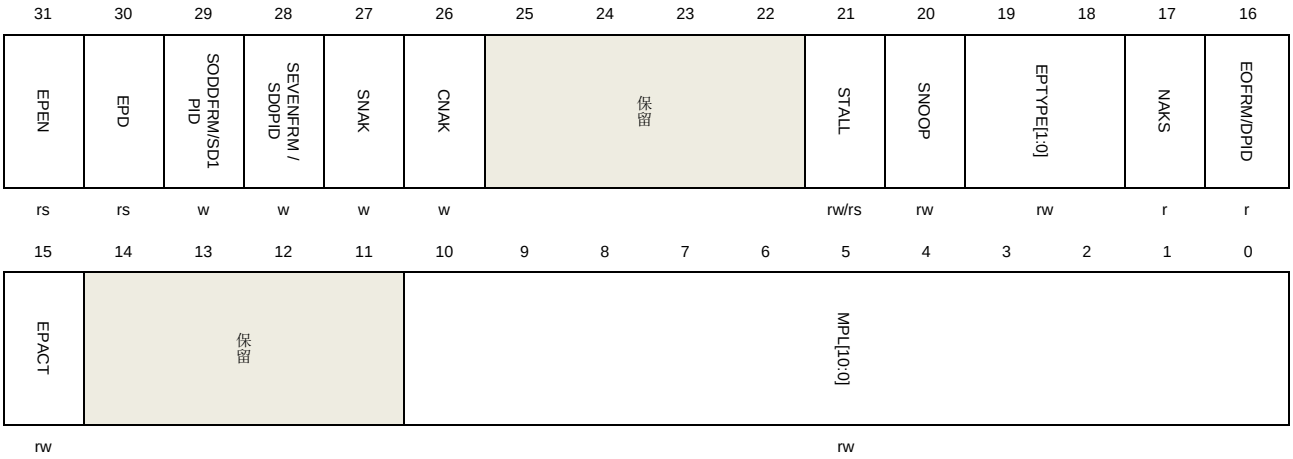
设备 OUT 端点 x 控制寄存器（USBHS_DOEPxCTL）（x= 1..5， x 是设备端点编号）

地址偏移：0x0B00 + (设备端点编号 x 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



位/位域	名称	描述
31	EPEN	端点使能 软件置位，USBHS清零 0：端点除能 1：端点使能 软件应该按照操作指南使能或除能端点。
30	EPD	端点除能

		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） 该位只针对同步OUT端点有效 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) 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） 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) 软件置1该位来清零该寄存器的DPID位
27	SNAK	设置NAK 软件置1该位从而置1该寄存器的NAKS位
26	CNAK	清零NAK 软件置1该位从而清零该寄存器的NAKS位
25:22	保留	必须保留为复位值
21	STALL	STALL握手 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USBHS_DCTL的GINS位优先级要高，如果STALL和NAKS位都被置位，STALL位生效。 对于控制OUT端点： 当OUT端点接收SETUP令牌时，只有USBHS可以清零该位，软件不可清零此位。 对于中断或大容量OUT端点 只有软件可以清零该位
20	SNOOP	调查模式 该位控制OUT端点的调查模式，在调查模式中，USBHS不再检查接收数据包的CRC值 0：调查模式除能 1：调查模式使能
19:18	EPTYPE[1:0]	端点类型 该域定义端点的传输类型 00：控制 01：同步 10：大容量 11：中断
17	NAKS	NAK状态 当该寄存器的STALL位和寄存器USBHS_DCTL的位GONS被清零，该位控制USBHS的NAK状态： 0：根据端点的Rx FIFO的状态，发送握手包 1：USBHS为OUT事务发送NAK握手

		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） 对于同步传输，软件通过使用该位控制USBHS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，USBHS不保存数据包 0：只在偶数帧发送数据 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） 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，USBHS在传输过程中保持该位。 0：数据包PID是DATA0 1：数据包PID是DATA1
15	EPACT	端点激活 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	必须保留为 复位值
10:0	MPL[10:0]	该位定义最大包长

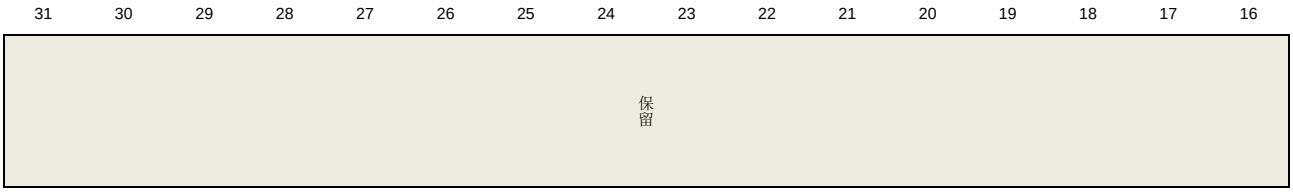
设备 IN 端点 x 中断标志寄存器（USBHS_DIEPxINTF）（x = 0..5， x 是设备端点编号）

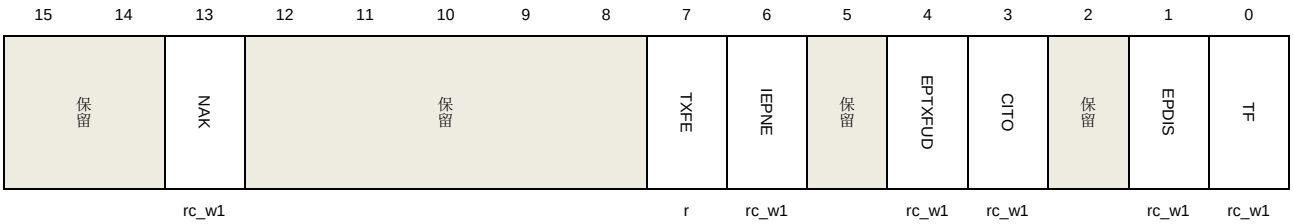
地址偏移：0x0908 + (设备端点编号 x 0x20)

复位值：0x0000 0080

该寄存器包含 IN 端点的状态和事件，当获得一个 IN 端点的中断时，应该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，除了 TXFE 位，各位写 1 清零。

该寄存器采用字（32 位）访问





位/位域	名称	描述
31:14	保留	必须保留为复位值
13	NAK	发送NAK握手 USBHS在发出一个NAK握手包后置该位，因为寄存器USBHS_DIEPxCTL的NAKS位

		被置位，或端点的Tx FIFO里没有包数据。
12:8	保留	必须保留为复位值
7	TXFE	发送FIFO空 端点的Tx FIFO达到寄存器USBHS_GAHBCS的位TXFTH定义的空阈值。
6	IEPNE	IN端点NAK有效 寄存器USBHS_DIEPxCTL的位SNAK的设置生效，该位可以通过写1清零或设置CNAK位
5	保留	必须保留为复位值
4	EPTXFUD	端点Tx FIFO下溢 如果当IN令牌被接收后，Tx FIFO没有包数据，该标志被触发。
3	CITO	控制IN事务超时中断 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	必须保留为复位值
1	EPDIS	端点除能 端点除能时，该标志位被触发
0	TF	传输完成 当该端点的所有IN事务完成，该标志位被触发。

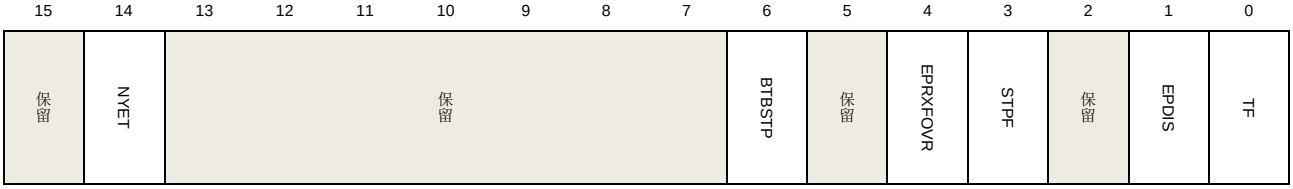
设备 OUT 端点 x 中断标志寄存器（USBHS_DOEPxINTF）（x = 0..5， x 是设备端点编号）

地址偏移：0x0B08 + (设备端点编号 x 0x20)

复位值：0x0000 0000

该寄存器包含 OUT 端点的状态和事件，当获得一个 OUT 端点的中断时，应该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



位/位域	名称	描述

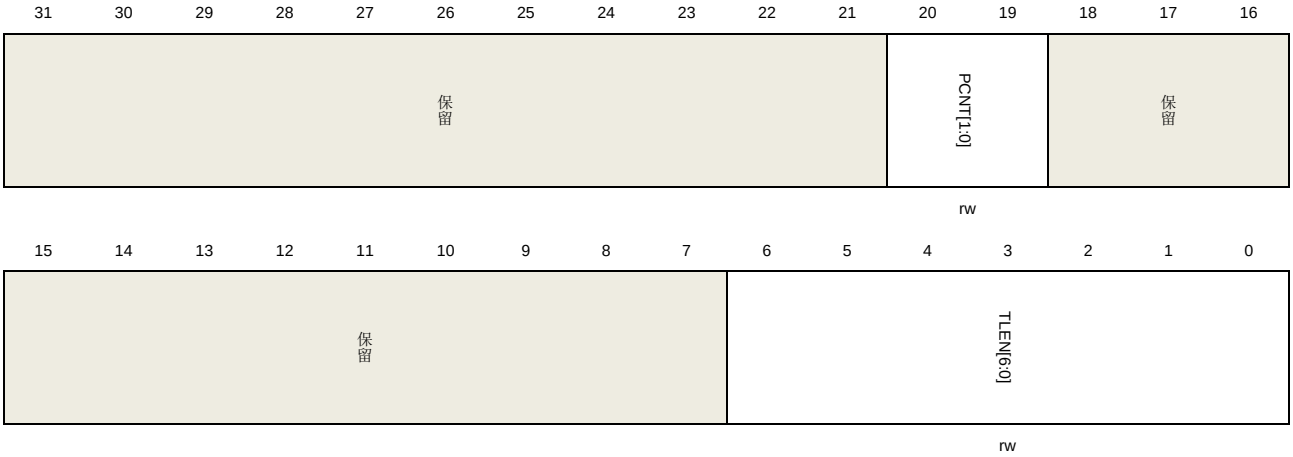
31:15	保留	必须保留为复位值
14	NYET	发送NYET握手 发送NYET握手包时，该标志被触发
13:7	保留	必须保留为复位值
6	BTBSTP	连续SETUP包（适用于控制OUT端点） 当一个控制OUT端点接收超过连续3个SETUP包时，该标志被触发。
5	保留	必须保留为复位值
4	EPRXFOVR	端点Rx FIFO上溢 当OUT令牌被接收时，如果OUT端点的Rx FIFO没有足够的空间存放数据包，该位被触发。在这种情况下，USBHS不能接收OUT数据包，发送一个NAK握手包。
3	STPF	SETUP阶段完成（适用于控制OUT端点） 当一个SETUP阶段完成，也就是USBHS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	必须保留为复位值
1	EPDIS	端点除能 端点除能时，该标志位被触发
0	TF	传输完成 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（USBHS_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	必须保留为复位值
20:19	PCNT[1:0]	包数

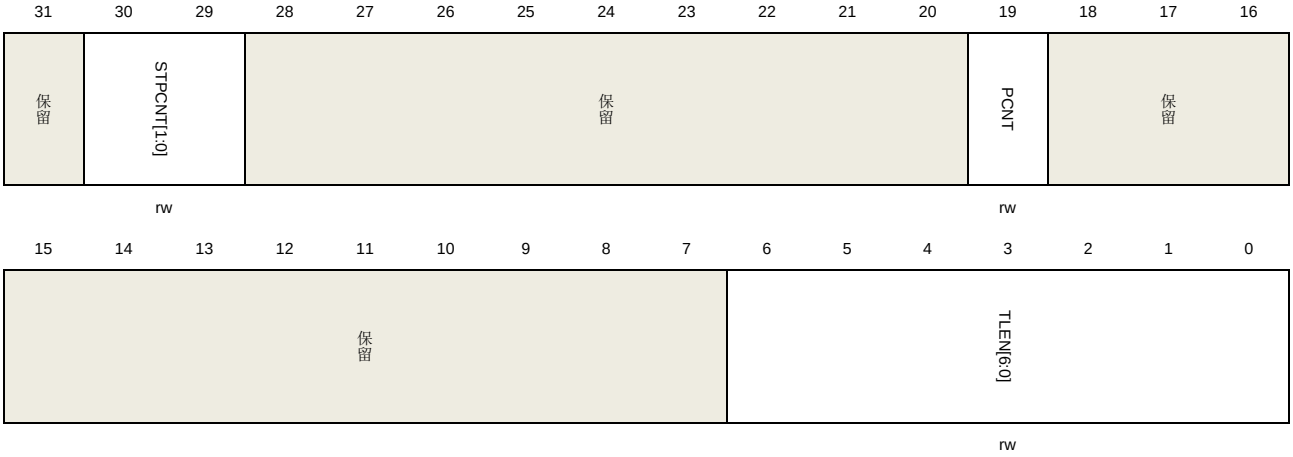
		传输中被发送的数据包数量 在端点使能之前，软件设置该位，在传输开始后，该域在每次数据包成功发送后自动减少。
18:7	保留	必须保留为复位值
6:0	TLEN[6:0]	传输长度 一次传输的数据总字节数 该域是IN传输中需要发送的包数据的总字节数，在端点使能之前，软件设置该位，在软件或DMA成功地将包数据写入端点的Tx FIFO中，该域减少与包数据大小相同的数值。

设备 OUT 端点 0 传输长度寄存器（USBHS_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	必须保留为复位值
30:29	STPCNT[1:0]	SETUP包计数 该域定义端点可以接受的最大连续SETUP包数量 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USBHS_DOEP0INTF的BTBSTP标志被触发。 00：0个包 01：1个包 10：2个包 11：3个包
28:20	保留	必须保留为复位值
19	PCNT	包计数 一次传输中应该接收到包数量。 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，该域数值自

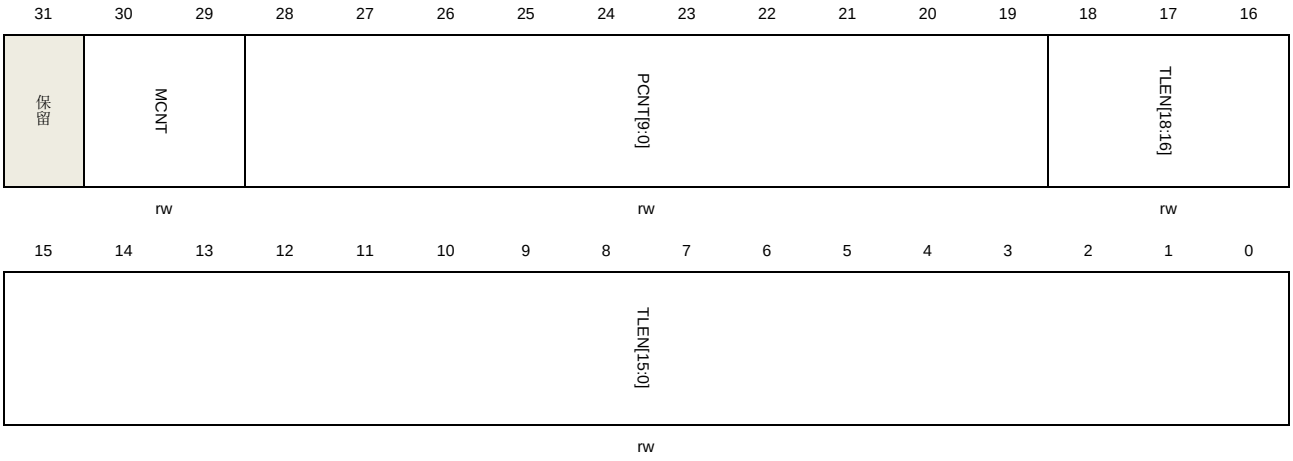
		动减少。
18:7	保留	必须保留为复位值
6:0	TLEN[6:0]	传输长度 传输中数据总字数。 该域是OUT传输中需要接收的包数据的总字节数，在端点使能之前，软件设置该位，在软件或DMA成功地将包数据读取端点的Rx FIFO中，该域减少与包数据大小相同的数值。

设备 IN 端点 x 传输长度寄存器（USBHS_DIEPxLEN）（x = 1..5， x 是设备端点编号）

地址偏移：0x910 + (设备端点编号 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
31:29	保留	必须保留为复位值
30:29	MCNT[1:0]	多包个数 该域描述在一帧内需要传输的包的个数 01:1个包 10:2个包 11:3个包
28:19	PCNT[9:0]	包数量 传输中被发送的数据包数量 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自动减少。

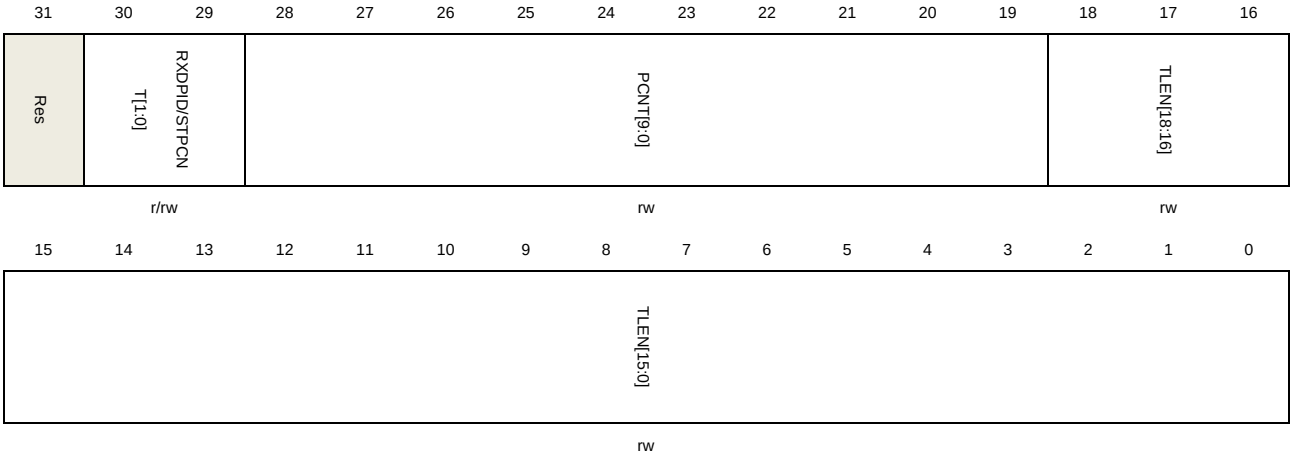
18:0	TLEN[18:0]	传输长度 传输的数据总字节数 该域是IN传输中需要发送的包数据的总字节数，在端点使能之前，软件设置该位，在软件或DMA成功地将包数据写入端点的Tx FIFO中，该域减少与包数据大小相同的数值。

设备 OUT 端点 x 传输长度寄存器（USBHS_DOEPxLEN）（x = 1..5， x 是设备端点编号）

地址偏移：0x0B10 + (设备端点编号 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	保留	必须保留为复位值
30:29	RXDPID[1:0]	接收数据PID（适用于同步OUT端点) 该域保存该端点该数据包所接受的最后一个数据包的PID 00：DATA0 01：DATA2 10：DATA1 11：MDATA
	STPCNT[1:0]	SETUP包数（适用于控制OUT端点) 该位定义该端点可以接受连续SETUP最大包数 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USBHS_DOEP0INTF的BTBSTP标志被触发。 00：0个包 01：1个包 10：2个包 11：3个包
28:19	PCNT[9:0]	包数 传输中应该接收到包数量 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，该域数值自动

减少。

18:0	TLEN[18:0]	传输长度 传输中数据总字数 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，在软件或DMA成功地将包数据读取端点的Rx FIFO中，该域减少与包数据大小相同的数值。

设备 IN 端点 x DMA 地址寄存器（USBHS_DIEPxDMAADDR）/设备 OUT 端点 x DMA 地址寄存器（USBHS_DOEPxDMAADDR）（x = 0..5， x 是设备端点编号）

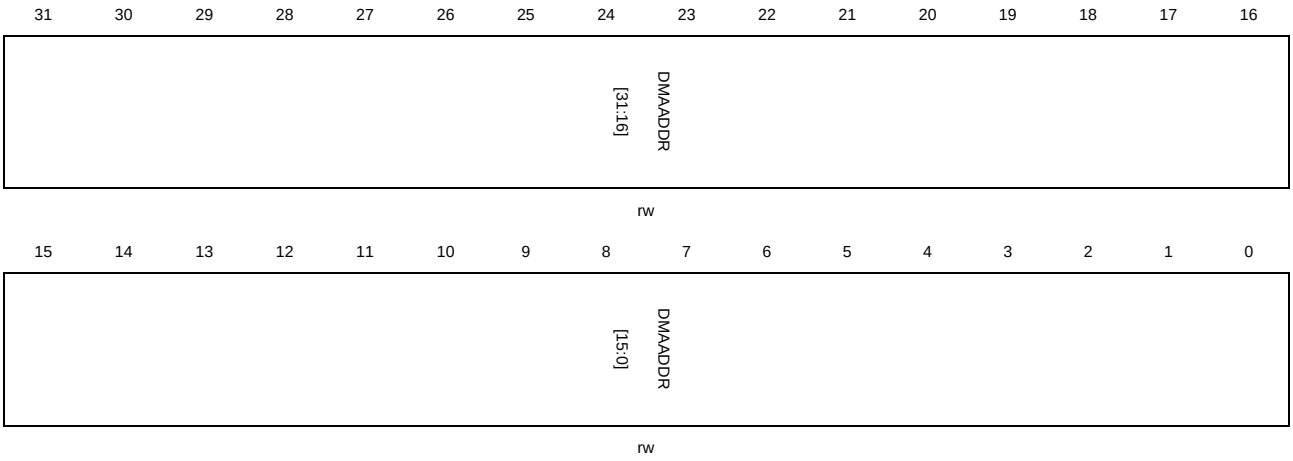
地址偏移：

IN endpoint：0x0914 + (设备端点编号 x 0x20)

OUT endpoint：0x0B14 + (设备端点编号 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
31:0	DMAADDR[31:0]	DMA地址 该域定义端点的DMA地址，DMA使用该地址为IN端点提取包数据，或为OUT端点写入包数据。

设备 IN 端点 x 发送 FIFO 状态寄存器（USBHS_DIEPxTFSTAT）（x = 0..5， x 是设备端点编号）

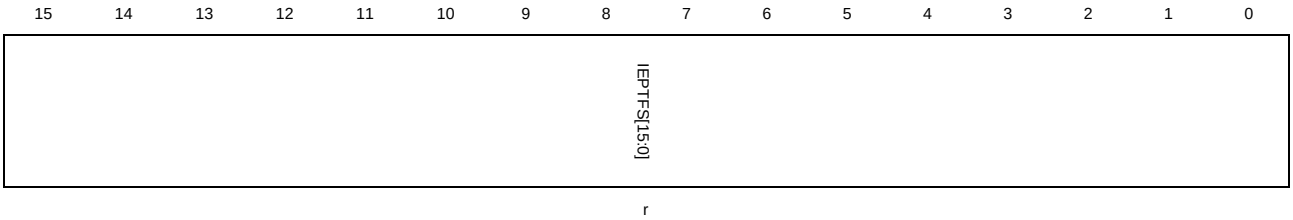
地址偏移：0x0918 + (设备端点编号 x 0x20)

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

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



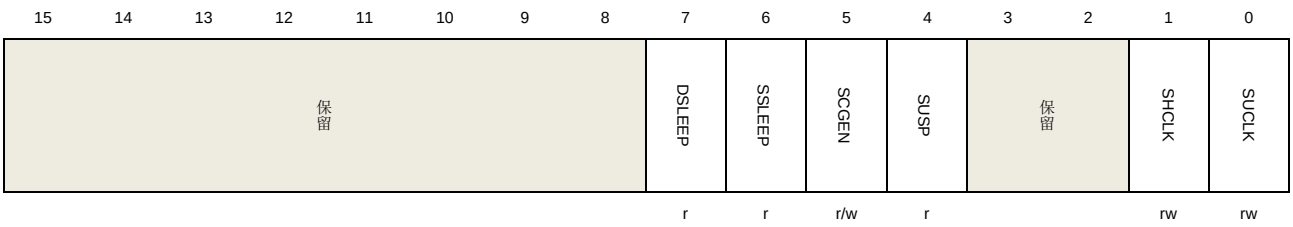
位/位域	名称	描述
31:16	保留	必须保留为复位值
15:0	IEPTFS[15:0]	IN端点的Tx FIFO可用空间 IN端点的Tx FIFO可用空间用32位字为单位 0：FIFO是满的 1：1字可用 … n：n字可用

27.7.4. 电源和时钟控制寄存器（USBHS_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
31:8	保留	必须保留为复位值
7	DSLEEP	PHY处于深度睡眠状态

6	SSLEEP	PHY处于浅睡眠状态
5	SCGEN	当该位被设置时，内部门控时钟被启用。
4	SUSP	PHY处于挂起状态
3:2	保留	必须保留为复位值
1	SHCLK	停止HCLK 停止HCLK，节省电量 0：HCLK未停止 1：HCLK停止
0	SUCLK	停止USB时钟 停止USB时钟，节省电量 0：USB时钟未停止 1：USB时钟停止

28. 附录

28.1. 寄存器表中使用的缩写列表

表 28-1. 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。

28.2. 术语表

表 28-2. 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

28.3. 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。

29. 版本历史

表 29-1. 版本历史

版本号	描述	日期
1.0	初稿发布	2023 年 4 月 7 日
1.1	1. 名称更改为 A508xx。 2. 更正 ADC 外设数量。	2023 年 4 月 25 日
1.2	1. 格式修改。 2. 修改 SHRTIMER 章节的***图 19 46. ADC 触发源选择图***。	2023 年 6 月 20 日
1.3	1. 修改 GPIO 章节的***表 8 9. CAN0/1 备用功能重映射***。 2. 修改 PMU 章节的***图 4 1. 电源域概览***。	2023 年 11 月 15 日
1.4	1. 格式修改。 2. 修改 SHRTIMER 章节的 ***19.4.1. Master TIMER 单元 / 更新事件和影子寄存器***、***19.4.2. Slave_TIMERx(x=0..4) 单元 / 空闲控制***、***19.4.2. Slave_TIMERx(x=0..4) 单元/ 更新事件和影子寄存器***。 3. 修改 I2C 章节，***模拟滤波器产生的延时最大值由 260ns 改为 160ns***。 4. 修改 19.4.12.DMA 模式的***图 19 48. DMA 模式运行流程图***。 5. 修改以太网章节的***图 26 8. 增强发送描述符***和***图 26 10. 增强接收描述符***。 6. 删除 FMC、PMU 和 RCU 章节的 ***BOR 相关内容***。 7. 修改 CMP 章节描述 ***15.3.2. 比较器的 I / O 配置***。	2024 年 7 月 15 日
1.5	1. 修改***13.4.6. 运行模式***章节，增加描述“注意：当 EOC 置 1 后，需延迟一个 CK_ADC 再读取 ADC 转换结果”。 2. 修改 RCU 章节 5.2.1 节，增加 ***SHRTIMER 高精度模式时钟选择部分***描述。 3. ***13.4.13. 可编程分辨率（DRES）***章节 tADC 的单位改成 ns. 4. PMU 章节删除 ***Vhyst 电压值***描述。	2024 年 12 月 25 日
1.6	1. 格式修改。 2. 修改 4.3 小节，***POR/PDR 和 1.1V 域/COREOFF0 域/COREOFF1 域***相关内容。 3. 修改 ***21.2.3 主机发送模式下的软件流程&主机接收模式下的软件流程***描述。 4. 修改 ***27.7.3 节的设备状态寄存器（USBHS_DSTAT）***。 5. 更新 ***Important Notice***。	2025 年 3 月 21 日
1.7	1. ***SHRTIMER 章节***中对表 19-1 中的最小值和最大值	2025 年 08 月 08 日

	的描述进行补充更新 2. 将 ***19.4.2 节*** 下计数器时钟小节内的 SHRTIMER_MTACTL 改为 SHRTIMER_STxACTL， SHRTIMER_MTCTL0 改为 SHRTIMER_STxCTL0 3. ***USART 20.1.3, 20.2.3 章节***中修改描述。 4. 修改 ***21.1.3 超时特性***节和 ***21.2.4 I2C_SADDR1*** 节描述 5. ***I2S 22.4.4*** 节增加描述。 6. 修改 ***8.5.1 GPIOx_CTL0 和 8.5.2 GPIOx_CTL1*** 复位值。	
1.8	1. USBHS 章节中，修改 ***USBHS_HFT 寄存器的 FRI 的 bit 位***的描述 2. PMU 章节中，***1.2V 域描述修改为 VCORE 域*** 3. I2C 章节中，修改 ***I2C_CTL1 、I2C_TIMING 、I2C_SADDR0、I2C_SADDR1、I2C_TIMING 寄存器*** 描述，***I2C 时序配置***小节增加注意描述，***SMBus 主机接收器***和***从机发送器***小节增加注意描述， 4. USART 章节中，***USART 接收器***小节删除部分触发噪声错误描述，***智能卡（ISO7816-3）模式***小节中，修改重发数据帧位数描述，***串行红外***小节增加注意描述， 5. SHRTIMER 章节，***HRTIMER_EXEVCFG1 寄存器*** 的 EXEV9EG 改为 EXEV7EG 6. DBG 章节，***12.2.3. JTAG 链状结构***小节中修改 BSD TAP 为 MCU JTAG TAP 7. FMC 章节中，***2.2. 主要特征***小节增加“闪存支持 128 位宽数据读取”的主要特征 8. FWDGT 章节中，***16.1.3 功能描述***增加注意描述 9. RCU 章节中，***RCU_AHBEN 寄存器***中删除 FMCSPEN 位 *10.* CAN 章节，***25.3.7.通信参数***小节，修改 CAN 协议同步边沿的描述	2026 年 01 月 20 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, aeronautic or aerospace applications, pollution control or hazardous substance management; (ii) life-support systems, other medical equipment or systems (including life support equipment and surgical implants); and/or (iii) 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.

While the Company has implemented advanced security features, the Product may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on Customer's applications and products, and to the maximum extent permitted by applicable law, the Company accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

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