GigaDevice Semiconductor Inc.

GD32A50x
Arm® Cortex®-M33 32-bit MCU

适用于 **GD32A503xx** 系列

用户手册

1.6 版本

(2025 年 2 月)

目录

目录 2
图索引 16
表索引 21
1. 系统及存储器架构 24
1.1. Arm® Cortex®-M33 处理器 24
1.2. 系统架构 25
1.3. 存储器映射 27
1.3.1. 片上 SRAM 存储器 31
1.3.2. 片上闪存 32
1.4. 引导配置 33
1.5. 系统配置控制器 33
1.6. 系统配置寄存器 34
1.6.1. 系统配置寄存器 0（SYSCFG_CFG0） 34
1.6.2. 系统配置寄存器 1（SYSCFG_CFG1） 34
1.6.3. EXTI 源选择寄存器 0（SYSCFG_EXTISS0） 35
1.6.4. EXTI 源选择寄存器 1（SYSCFG_EXTISS1） 36
1.6.5. EXTI 源选择寄存器 2（SYSCFG_EXTISS2） 38
1.6.6. EXTI 源选择寄存器 3（SYSCFG_EXTISS3） 39
1.6.7. 系统配置寄存器 1（SYSCFG_CFG2） 40
1.6.8. 系统状态寄存器（SYSCFG_STAT） 41
1.6.9. 系统配置寄存器 3（SYSCFG_CFG3） 42
1.6.10. TIMER 输入源选择寄存器（SYSCFG_TIMERINSEL） 43
1.7. 设备电子签名 45
1.7.1. 存储容量信息 45
1.7.2. 设备唯一 ID（96 位/位域） 46
2. 闪存控制器（FMC） 48
2.1. 简介 48
2.2. 主要特征 48
2.3. 功能说明 48
2.3.1. 闪存结构 48
2.3.2. 错误检查与纠正（ECC） 50
2.3.3. 读操作 51
2.3.4. 选项字节不可以用缓存。双 bank 边读边写特性 52
2.3.5. FMC_CTLx 寄存器解锁 53
2.3.6. 页擦除 53

2.3.7. 整片擦除 54
2.3.8. 主存储闪存块编程 55
2.3.9. 主存储闪存块快速编程 57
2.3.10. 查空命令 59
2.3.11. OTP 编程 59
2.3.12. 共享 RAM 59
2.3.13. 数据闪存操作 59
2.3.14. 选项字节 0 擦除 60
2.3.15. 选项字节编程 60
2.3.16. 选项字节说明 62
2.3.17. 页擦除 / 编程保护 64
2.3.18. 安全保护 65
2.3.19. 错误描述 66

2.4. FMC 寄存器 68
2.4.1. 等待状态寄存器（FMC_WS） 68
2.4.2. ECC 控制和状态寄存器（FMC_ECCCS） 69
2.4.3. 解锁寄存器 0（FMC_KEY0） 71
2.4.4. 状态寄存器 0（FMC_STAT0） 71
2.4.5. 控制寄存器 0（FMC_CTL0） 72
2.4.6. 地址寄存器 0（FMC_ADDR0） 73
2.4.7. 选项字节操作解锁寄存器（FMC_OBKEY） 74
2.4.8. 解锁寄存器 1（FMC_KEY1） 74
2.4.9. 状态寄存器 1（FMC_STAT1） 74
2.4.10. 控制寄存器 1（FMC_CTL1） 75
2.4.11. 地址寄存器 1（FMC_ADDR1） 77
2.4.12. 选项字节状态寄存器（FMC_OBSTAT） 78
2.4.13. 擦除 / 编程保护寄存器 0（FMC_WP0） 78
2.4.14. 擦除 / 编程保护寄存器 1（FMC_WP1） 79
2.4.15. 选项字节 1 控制和状态寄存器（FMC_OB1CS） 79
2.4.16. 产品 ID 寄存器（FMC_PID） 80

3. 电源管理单元（PMU） 81

3.1. 简介 81

3.2. 主要特征 81

3.3. 功能说明 81
3.3.1. V_{DD} / V_{DDA} 电源域 82
3.3.2. 1.1V 电源域 85
3.3.3. 省电模式 85

3.4. PMU 寄存器 88
3.4.1. 控制寄存器（PMU_CTL） 88
3.4.2. 电源控制和状态寄存器（PMU_CS） 89

4. 备份寄存器（BKP） 91

4.1. 简介 91
4.2. 主要特征 91
4.3. 功能说明 91
4.3.1. RTC 时钟校准 91
4.3.2. 侵入检测 91
4.4. BKP 寄存器 92
4.4.1. 备份数据寄存器（BKP_DATAx）（x= 0..9） 92
4.4.2. RTC 信号输出控制寄存器（BKP_OCTL） 92
4.4.3. 侵入引脚控制寄存器（BKP_TPCTL） 93
4.4.4. 侵入控制状态寄存器（BKP_TPCS） 94
5. 复位和时钟单元（RCU） 96
5.1. 复位控制单元（RCTL） 96
5.1.1. 简介 96
5.1.2. 功能描述 96
5.2. 时钟控制单元（CCTL） 97
5.2.1. 简介 97
5.2.2. 主要特性 98
5.2.3. 功能描述 99
5.3. RCU 寄存器 103
5.3.1. 控制寄存器（RCU_CTL） 103
5.3.2. 配置寄存器 0（RCU_CFG0） 105
5.3.3. 中断寄存器（RCU_INT） 108
5.3.4. APB2 复位寄存器（RCU_APB2RST） 111
5.3.5. APB1 复位寄存器（RCU_APB1RST） 112
5.3.6. AHB 使能寄存器（RCU_AHBEN） 114
5.3.7. APB2 使能寄存器（RCU_APB2EN） 116
5.3.8. APB1 使能寄存器（RCU_APB1EN） 118
5.3.9. 备份域控制寄存器（RCU_BDCTL） 119
5.3.10. 复位源/时钟寄存器（RCU_RSTSCK） 121
5.3.11. AHB 复位寄存器（RCU_AHBRST） 124
5.3.12. 配置寄存器 1（RCU_CFG1） 125
5.3.13. 配置寄存器 2（RCU_CFG2） 126
5.3.14. 电源解锁寄存器（RCU_VKEY） 127
5.3.15. 深度睡眠模式电压寄存器（RCU_DSV） 128
6. 中断事件控制器（EXTI） 129
6.1. 简介 129
6.2. 主要特性 129
6.3. 功能说明 129
6.4. 外部中断及事件（EXTI）框图 132

6.5. 外部中断及事件功能概述 132
6.6. EXTI 寄存器 135
6.6.1. 中断使能寄存器（EXTI_INTEN）..... 135
6.6.2. 事件使能寄存器（EXTI_EVEN）..... 135
6.6.3. 上升沿触发使能寄存器（EXTI_RTEN）..... 136
6.6.4. 下降沿触发使能寄存器（EXTI_FTEN）..... 136
6.6.5. 软件中断事件寄存器（EXTI_SWIEV）..... 136
6.6.6. 挂起寄存器（EXTI_PD）..... 137
7. 触发选择控制器（TRIGSEL）..... 138
7.1. 简介 138
7.2. 主要特征 138
7.3. 功能说明 138
7.4. 内部连接 139
7.5. TRIGSEL 寄存器 144
7.5.1. EXTOUT0 触发选择寄存器（TRIGSEL_EXTOUT0）..... 144
7.5.2. EXTOUT1 触发选择寄存器（TRIGSEL_EXTOUT1）..... 145
7.5.3. ADC0 触发选择寄存器（TRIGSEL_ADC0）..... 145
7.5.4. ADC1 触发选择寄存器（TRIGSEL_ADC1）..... 146
7.5.5. DAC 触发选择寄存器（TRIGSEL_DAC）..... 147
7.5.6. TIMER0_ITI 触发选择寄存器（TRIGSEL_TIMER0IN）..... 147
7.5.7. TIMER0_BRKIN 触发选择寄存器（TRIGSEL_TIMER0BRKIN）..... 148
7.5.8. TIMER7_ITI 触发选择寄存器（TRIGSEL_TIMER7IN）..... 149
7.5.9. TIMER7_BRKIN 触发选择寄存器（TRIGSEL_TIMER7BRKIN）..... 150
7.5.10. TIMER19_ITI 触发选择寄存器（TRIGSEL_TIMER19IN）..... 151
7.5.11. TIMER19_BRKIN 触发选择寄存器（TRIGSEL_TIMER19BRKIN）..... 151
7.5.12. TIMER20_ITI 触发选择寄存器（TRIGSEL_TIMER20IN）..... 152
7.5.13. TIMER20_BRKIN 触发选择控制寄存器（TRIGSEL_TIMER20BRKIN）..... 153
7.5.14. TIMER1_ITI 触发选择寄存器（TRIGSEL_TIMER1IN）..... 154
7.5.15. MFCOM register 触发选择寄存器（TRIGSEL_MFCOM）..... 155
7.5.16. CAN0 触发选择寄存器（TRIGSEL_CAN0）..... 156
7.5.17. CAN1 触发选择寄存器（TRIGSEL_CAN1）..... 156
8. 通用和备用输入/输出接口（GPIO 和 AFIO）..... 158
8.1. 简介 158
8.2. 主要特征 158
8.3. 功能说明 158
8.3.1. GPIO 引脚配置..... 159
8.3.2. 外部中断及事件..... 160
8.3.3. 备用功能（AF）..... 160
8.3.4. 附加功能..... 160
8.3.5. 输入配置..... 160

8.3.6. 输出配置 161
8.3.7. 模拟配置 161
8.3.8. 备用功能（AF）配置 162
8.3.9. GPIO 锁定功能 163
8.3.10. GPIO 单周期输出翻转功能 163

8.4. GPIO 寄存器 164
8.4.1. 端口控制寄存器（GPIOx_CTL，x=A..F） 164
8.4.2. 端口输出模式寄存器（GPIOx_OMODE，x=A..F） 165
8.4.3. 端口输出速度寄存器（GPIOx_OSPD，x=A..F） 167
8.4.4. 端口上拉/下拉寄存器（GPIOx_PUD，x=A..F） 169
8.4.5. 端口输入状态寄存器（GPIOx_ISTAT，x=A..F） 170
8.4.6. 端口输出控制寄存器（GPIOx_OCTL，x=A..F） 171
8.4.7. 端口位操作寄存器（GPIOx_BOP，x=A..F） 171
8.4.8. 端口配置锁定寄存器（GPIOx_LOCK，x=A..F） 172
8.4.9. 备用功能选择寄存器 0（GPIOx_AFSEL0，x=A..F） 173
8.4.10. 备用功能选择寄存器 1（GPIOx_AFSEL1，x=A..F） 174
8.4.11. 位清除寄存器（GPIOx_BC，x=A..F） 175
8.4.12. 端口位翻转寄存器（GPIOx_TG，x=A..F） 176

9. 多功能通信接口（MFCOM） 177

9.1. 简介 177

9.2. 主要特征 177

9.3. 框图 177

9.4. 功能说明 178
9.4.1. 时钟和复位 178
9.4.2. 移位器 178
9.4.3. 定时器 179
9.4.4. 引脚 181
9.4.5. 中断和 DMA 请求 182
9.4.6. 触发 182

9.5. MFCOM 寄存器 183
9.5.1. 控制寄存器（MFCOM_CTL） 183
9.5.2. 引脚数据寄存器（MFCOM_PINDATA） 183
9.5.3. 移位器状态寄存器（MFCOM_SSTAT） 184
9.5.4. 移位器错误寄存器（MFCOM_SERR） 184
9.5.5. 定时器状态寄存器（MFCOM_TMSTAT） 185
9.5.6. 移位器状态中断使能寄存器（MFCOM_SSIEN） 185
9.5.7. 移位器错误中断使能寄存器（MFCOM_SEIEN） 186
9.5.8. 定时器状态中断使能寄存器（MFCOM_TMSIEN） 186
9.5.9. 移位器状态 DMA 使能寄存器（MFCOM_SSDMAEN） 187
9.5.10. 移位器控制寄存器 x（MFCOM_SCTLx） 187
9.5.11. 移位器配置寄存器 x（MFCOM_SCFGx） 188
9.5.12. 移位缓冲区寄存器 x（MFCOM_SBUFx） 189

9.5.13. 移位缓冲区位交换寄存器 x（MFCOM_SBUFBISx） 190
9.5.14. 移位缓冲区字节交换寄存器 x（MFCOM_SBUFBYSx） 190
9.5.15. 移位缓冲区位字节交换寄存器 x（MFCOM_SBUFBBSx） 190
9.5.16. 定时器控制寄存器 x（MFCOM_TMCTLx） 191
9.5.17. 定时器配置寄存器 x（MFCOM_TMCFGx） 192
9.5.18. 定时器比较寄存器 x（MFCOM_TMCMPx） 194

10. 循环冗余校验计算单元（CRC） 196

10.1. 简介 196

10.2. 主要特征 196

10.3. 功能说明 197

10.4. CRC 寄存器 198

10.4.1. 数据寄存器（CRC_DATA） 198
10.4.2. 独立数据寄存器（CRC_FDATA） 198
10.4.3. 控制寄存器（CRC_CTL） 199
10.4.4. 初值寄存器（CRC_IDATA） 199
10.4.5. 多项式寄存器（CRC_POLY） 200

11. 直接存储器访问控制器（DMA） 201

11.1. 简介 201

11.2. 主要特征 201

11.3. 结构框图 202

11.4. 功能说明 202

11.4.1. DMA 操作 202
11.4.2. 外设握手 203
11.4.3. 仲裁 204
11.4.4. 地址生成 204
11.4.5. 循环模式 204
11.4.6. 存储器到存储器模式 205
11.4.7. 通道配置 205
11.4.8. 中断 205
11.4.9. DMA 请求映射 206

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） 208
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. DMA 请求多路复用器（DMAMUX） 212

12.1. 简介 212

12.2. 主要特征 212
12.3. 结构框图 213
12.4. 功能说明 213
12.4.1. DMAMUX 信号 214
12.4.2. DMAMUX 请求路由器 214
12.4.3. DMAMUX 请求生成器 216
12.4.4. 通道配置 217
12.4.5. 中断 217
12.4.6. DMAMUX 映射 218
12.5. DMAMUX 寄存器 222
12.5.1. 请求路由通道 x 配置寄存器（DMAMUX_RM_CHxCFG） 222
12.5.2. 请求路由通道中断标志位寄存器（DMAMUX_RM_INTF） 223
12.5.3. 请求路由通道中断标志位清除寄存器（DMAMUX_RM_INTC） 223
12.5.4. 请求生成通道 x 配置寄存器（DMAMUX_RG_CHxCFG） 224
12.5.5. 请求生成通道中断标志位寄存器（DMAMUX_RG_INTF） 224
12.5.6. 请求生成通道中断标志位清除寄存器（DMAMUX_RG_INTC） 225
13. 调试（DBG） 226
13.1. 简介 226
13.2. JTAG/SW 功能说明 226
13.2.1. 切换 JTAG/SW 接口 226
13.2.2. 引脚分配 226
13.2.3. JTAG 链状结构 227
13.2.4. 调试复位 227
13.2.5. JEDEC-106 ID code 227
13.3. 调试保持功能说明 227
13.3.1. 低功耗模式调试支持 227
13.3.2. TIMER，I2C，WWDGT 和 FWDGT 外设调试支持 228
13.4. DBG 寄存器 229
13.4.1. ID 寄存器（DBG_ID） 229
13.4.2. 控制寄存器（DBG_CTL） 229
14. 模数转换器（ADC） 232
14.1. 简介 232
14.2. 主要特征 232
14.3. 引脚和内部信号 233
14.4. 功能说明 233
14.4.1. 前置校准功能 234
14.4.2. ADC 时钟 234
14.4.3. ADC 使能 234
14.4.4. 常规序列 234

14.4.5. 运行模式 234
14.4.6. 转换结果阈值监测功能 237
14.4.7. 数据存储模式 237
14.4.8. 采样时间配置 238
14.4.9. 外部触发 239
14.4.10. DMA 请求 239
14.4.11. ADC 内部通道 239
14.4.12. 可编程分辨率(DRES) 240
14.4.13. 片上硬件过采样 240

14.5. ADC 同步模式 242
14.5.1. 独立模式 243
14.5.2. 常规并行模式 243
14.5.3. 常规快速跟随模式 244
14.5.4. 常规慢速跟随模式 244

14.6. 中断 245

14.7. ADC 寄存器 246
14.7.1. 状态寄存器（ADC_STAT） 246
14.7.2. 控制寄存器 0（ADC_CTL0） 247
14.7.3. 控制寄存器 1（ADC_CTL1） 248
14.7.4. 采样时间寄存器 0（ADC_SAMPT0） 250
14.7.5. 采样时间寄存器 1（ADC_SAMPT1） 251
14.7.6. 看门狗 0 高阈值寄存器（ADC_WDHT0） 252
14.7.7. 看门狗 0 低阈值寄存器（ADC_WDLT0） 252
14.7.8. 常规序列寄存器 0（ADC_RSQ0） 253
14.7.9. 常规序列寄存器 1（ADC_RSQ1） 253
14.7.10. 常规序列寄存器 2（ADC_RSQ2） 254
14.7.11. 常规数据寄存器（ADC_RDATA） 254
14.7.12. 过采样控制寄存器（ADC_OVSAMPCTL） 255
14.7.13. 看门狗 1 通道选择寄存器（ADC_WD1SR） 256
14.7.14. 看门狗 1 阈值寄存（ADC_WDT1） 257

15. 数模转换器（DAC） 258

15.1. 简介 258

15.2. 主要特征 258

15.3. 功能描述 259
15.3.1. DAC 使能 259
15.3.2. DAC 输出缓冲 259
15.3.3. DAC 数据配置 259
15.3.4. DAC 触发 259
15.3.5. DAC 转换 260
15.3.6. DAC 噪声波 260
15.3.7. DAC 输出电压 261
15.3.8. DMA 请求 261

15.4. DAC 寄存器 262
15.4.1. DACx 控制寄存器 (DAC_CTL0) 262
15.4.2. DACx 软件触发寄存器 (DAC_SWT) 263
15.4.3. DACx_OUT0 12 位右对齐数据保持寄存器 (DAC_OUT0_R12DH) 264
15.4.4. DACx_OUT0 12 位左对齐数据保持寄存器 (DAC_OUT0_L12DH) 264
15.4.5. DACx_OUT0 8 位右对齐数据保持寄存器 (DAC_OUT0_R8DH) 265
15.4.6. DACx_OUT0 数据输出寄存器 (DAC_OUT0_DO) 265
15.4.7. DACx 状态寄存器 0 (DAC_STAT0) 265
16. 看门狗定时器（WDGT） 267
16.1. 独立看门狗定时器（FWDGT） 267
16.1.1. 简介 267
16.1.2. 主要特征 267
16.1.3. 功能说明 267
16.1.4. FWDGT 寄存器 270
16.2. 窗口看门狗定时器（WWDGT） 273
16.2.1. 简介 273
16.2.2. 主要特征 273
16.2.3. 功能说明 273
16.2.4. WWDGT 寄存器 275
17. 实时时钟（RTC） 277
17.1. 简介 277
17.2. 主要特征 277
17.3. 功能说明 277
17.3.1. RTC 复位 278
17.3.2. RTC 读取 278
17.3.3. RTC 配置 278
17.3.4. RTC 标志位 279
17.4. RTC 寄存器 280
17.4.1. RTC 中断使能寄存器（RTC_INTEN） 280
17.4.2. RTC 控制寄存器（RTC_CTL） 280
17.4.3. RTC 预分频寄存器高位（RTC_PSCH） 281
17.4.4. RTC 预分频寄存器低位（RTC_PSCL） 281
17.4.5. RTC 分频器高位（RTC_DIVH） 282
17.4.6. RTC 分频器低位（RTC_DIVL） 282
17.4.7. RTC 计数寄存器高位（RTC_CNTH） 283
17.4.8. RTC 计数寄存器低位（RTC_CNTL） 283
17.4.9. RTC 闹钟寄存器高位（RTC_ALRMH） 283
17.4.10. RTC 闹钟寄存器低位（RTC_ALRML） 284
18. 定时器（TIMER） 285
18.1. 高级定时器（TIMERx,x=0,7,19.20） 286

18.1.1. 简介 286
18.1.2. 主要特征 286
18.1.3. 结构框图 286
18.1.4. 功能说明 287
18.1.5. TIMERx 寄存器（x=0,7,19,20) 319

18.2. 通用定时器 L0（TIMERx, x=1） 370
18.2.1. 简介 370
18.2.2. 主要特征 370
18.2.3. 结构框图 371
18.2.4. 功能说明 371
18.2.5. TIMERx 寄存器（x=1） 386

18.3. 基本定时器（TIMERx, x=5,6） 407
18.3.1. 简介 407
18.3.2. 主要特征 407
18.3.3. 结构框图 407
18.3.4. 功能说明 407
18.3.5. TIMERx 寄存器(x=5,6) 411

19. 通用同步异步收发器（USART） 416

19.1. 简介 416

19.2. 主要特征 416

19.3. 功能描述 417
19.3.1. USART 帧格式 418
19.3.2. 波特率发生 419
19.3.3. USART 发送器 419
19.3.4. USART 接收器 420
19.3.5. DMA 方式访问数据缓冲区 422
19.3.6. 硬件流控制 423
19.3.7. 多处理器通信 424
19.3.8. LIN 模式 425
19.3.9. 同步通信模式 426
19.3.10. 串行红外（IrDA SIR）编解码功能模块 427
19.3.11. 半双工通信模式 428
19.3.12. 智能卡（ISO7816-3）模式 428
19.3.13. ModBus 通信 429
19.3.14. 接收 FIFO 430
19.3.15. 从 Deepsleep 模式唤醒 430
19.3.16. USART 中断 431

19.4. USART 寄存器 433
19.4.1. USART 控制寄存器 0（USART_CTL0） 433
19.4.2. USART 控制寄存器 1（USART_CTL1） 435
19.4.3. USART 控制寄存器 2（USART_CTL2） 437
19.4.4. USART 波特率寄存器（USART_BAUD） 440

19.4.5. USART 保护时间和预分频器寄存器（USART_GP） 440
19.4.6. USART 接收超时寄存器（USART_RT） 441
19.4.7. USART 请求寄存器（USART_CMD） 442
19.4.8. USART 状态寄存器（USART_STAT） 442
19.4.9. USART 中断标志清除寄存器（USART_INTC） 446
19.4.10. USART 数据接收寄存器（USART_RDATA） 447
19.4.11. USART 数据发送寄存器（USART_TDATA） 447
19.4.12. USART 兼容性控制寄存器（USART_CHC） 448
19.4.13. USART 接收 FIFO 控制和状态寄存器（USART_RFCS） 448

20. 内部集成电路总线接口（I2C） 450

20.1. 简介 450

20.2. 主要特征 450

20.3. 功能说明 450

20.3.1. 时钟要求 451
20.3.2. I2C 通讯流程 452
20.3.3. 噪声滤波器 454
20.3.4. I2C 时序配置 454
20.3.5. I2C 复位 456
20.3.6. 数据传输 456
20.3.7. I2C 从机模式 458
20.3.8. I2C 主机模式 463
20.3.9. SMBus 支持 468
20.3.10. SMBus 模式 470
20.3.11. DMA 模式下数据传输 472
20.3.12. I2C 错误和中断 472
20.3.13. I2C 调试模式 473

20.4. I2C 寄存器 474

20.4.1. 控制寄存器 0（I2C_CTL0） 474
20.4.2. 控制寄存器 1（I2C_CTL1） 476
20.4.3. 从机地址寄存器 0（I2C_SADDR0） 478
20.4.4. 从机地址寄存器 1（I2C_SADDR1） 478
20.4.5. 时序寄存器（I2C_TIMING） 479
20.4.6. 超时寄存器（I2C_TIMEOUT） 480
20.4.7. 状态寄存器（I2C_STAT） 481
20.4.8. 状态清除寄存器（I2C_STATC） 484
20.4.9. PEC 寄存器（I2C_PEC） 485
20.4.10. 接收数据寄存器（I2C_RDATA） 485
20.4.11. 发送数据寄存器（I2C_TDATA） 485
20.4.12. 控制寄存器 2（I2C_CTL2） 486

21. 串行外设接口/片上音频接口（SPI/I2S） 487

21.1. 简介 487

21.2. 主要特性 487
21.2.1. SPI 主要特性 487
21.2.2. I2S 主要特性 487
21.3. SPI 功能说明 488
21.3.1. SPI 结构框图 488
21.3.2. SPI 信号线描述 488
21.3.3. SPI 时序和数据帧格式 489
21.3.4. NSS 功能 490
21.3.5. SPI 运行模式 491
21.3.6. DMA 功能 498
21.3.7. CRC 功能 498
21.3.8. SPI 中断 499
21.4. I2S 功能描述 500
21.4.1. I2S 结构框图 500
21.4.2. I2S 信号线描述 501
21.4.3. I2S 音频标准 501
21.4.4. I2S 时钟 508
21.4.5. 运行 509
21.4.6. DMA 功能 513
21.4.7. I2S 中断 513
21.5. SPI/I2S 寄存器 515
21.5.1. 控制寄存器 0（SPI_CTL0） 515
21.5.2. 控制寄存器 1（SPI_CTL1） 517
21.5.3. 状态寄存器（SPI_STAT） 518
21.5.4. 数据寄存器（SPI_DATA） 519
21.5.5. CRC 多项式寄存器（SPI_CRCPOLY） 520
21.5.6. 接收 CRC 寄存器（SPI_RCRC） 520
21.5.7. 发送 CRC 寄存器（SPI_TCRC） 521
21.5.8. I2S 控制寄存器（SPI_I2SCTL） 521
21.5.9. I2S 时钟预分频寄存器（SPI_I2SPSC） 523
21.5.10. SPI0 四路 SPI 控制寄存器（SPI_QCTL） 523
22. 比较器（CMP） 525
22.1. 简介 525
22.2. 主要特征 525
22.3. 功能描述 525
22.3.1. 比较器时钟 526
22.3.2. 比较器的 I / O 配置 526
22.3.3. 比较器供电模式 527
22.3.4. 比较器迟滞 527
22.3.5. 比较器寄存器写保护 528
22.3.6. 比较器输出消隐 528
22.3.7. 电压定标器功能 529

22.3.8. 比较器中断 529
22.4. CMP 寄存器 530
22.4.1. CMP 控制状态寄存器（CMPx_CS） 530
23. 控制器局域网络（CAN） 533
23.1. 简介 533
23.2. 主要特征 533
23.3. 功能说明 534
23.3.1. 邮箱描述符 535
23.3.2. 接收 FIFO 描述符 539
23.3.3. 通信模式 544
23.3.4. 省电模式 545
23.3.5. 数据发送 546
23.3.6. 数据接收 550
23.3.7. 在虚拟联网模式下的数据接收 556
23.3.8. CAN FD 操作 558
23.3.9. 错误和状态 561
23.3.10. 通信参数 563
23.3.11. 中断 565
23.4. 典型的 CAN 配置流程示例 566
23.5. CAN 寄存器 568
23.5.1. 控制寄存器 0（CAN_CTL0） 568
23.5.2. 控制寄存器 1（CAN_CTL1） 570
23.5.3. 计数器寄存器（CAN_TIMER） 572
23.5.4. 接收邮箱公有过滤寄存器（CAN_RMPUBF） 572
23.5.5. 错误寄存器 0（CAN_ERR0） 573
23.5.6. 错误寄存器 1（CAN_ERR1） 573
23.5.7. 中断使能寄存器（CAN_INTEN） 576
23.5.8. 状态寄存器（CAN_STAT） 577
23.5.9. 控制寄存器 2（CAN_CTL2） 578
23.5.10. 常规帧 CRC 寄存器（CAN_CRCC） 580
23.5.11. 接收 FIFO 共有过滤寄存器（CAN_RFIFOPUBF） 580
23.5.12. 接收 FIFO 标识符过滤元素匹配序号寄存器（CAN_RFIFOIFMN） 581
23.5.13. 位时间寄存器（CAN_BT） 581
23.5.14. 接收 FIFO/邮箱私有过滤 x 寄存器（CAN_RFIFOMPFx）（x=0..31） 582
23.5.15. 虚拟联网模式控制寄存器 0（CAN_PN_CTL0） 583
23.5.16. 虚拟联网模式超时寄存器（CAN_PN_TO） 584
23.5.17. 虚拟联网模式状态寄存器（CAN_PN_STAT） 584
23.5.18. 虚拟联网模式期望标识符 0 寄存器（CAN_PN_EID0） 585
23.5.19. 虚拟联网模式期望 DLC 寄存器（CAN_PN_EDLC） 586
23.5.20. 虚拟联网模式期望数据低字 0 寄存器（CAN_PN_EDL0） 586
23.5.21. 虚拟联网模式期望数据低字 1 寄存器（CAN_PN_EDL1） 587
23.5.22. 虚拟联网模式标识符过滤器 / 期望标识符 1 寄存器（CAN_PN_IFEID1） 587

23.5.23. 虚拟联网模式数据 0 过滤器 / 期望数据高字 0 寄存器（CAN_PN_DF0EDH0） 588
23.5.24. 虚拟联网模式数据 1 过滤器 / 期望数据高字 1 寄存器（CAN_PN_DF1EDH1） 589
23.5.25. 虚拟联网模式接收唤醒邮箱 x 控制状态信息寄存器（CAN_PN_RWMxCS）（x=0..3） . 589
23.5.26. 虚拟联网模式接收唤醒邮箱 x 标识符寄存器（CAN_PN_RWMxI）（x=0..3） 590
23.5.27. 虚拟联网模式接收唤醒邮箱 x 数据 0 寄存器（CAN_PN_RWMxD0）（x=0..3） 591
23.5.28. 虚拟联网模式接收唤醒邮箱 x 数据 1 寄存器（CAN_PN_RWMxD1）（x=0..3） 591
23.5.29. FD 控制寄存器（CAN_FDCTL） 591
23.5.30. FD 位时间寄存器（CAN_FDBT） 593
23.5.31. 常规帧和 FD 帧 CRC 寄存器（CAN_CRCCFD） 593

24. 附录 595

24.1. 寄存器表中使用的缩写列表 595

24.2. 术语表 595

24.3. 可用外设 595

25. 版本历史 596

图索引

图 1-1. Cortex®-M33 处理器结构框图 25
图 1-2. GD32A50x 系列器件的系统架构示意图 27
图 1-3. ECC 解码器示意图 32
图 2-1. 页擦除操作流程 54
图 2-2. 整片擦除操作流程 55
图 2-3. 字编程操作流程 56
图 2-4. 快速编程操作流程 58
图 3-1. 电源域概览 81
图 3-2. 上电/掉电复位波形图 83
图 3-3. 欠压复位波形图 83
图 3-4. LVD 阈值波形图 84
图 3-5. OVD 阈值波形图 84
图 5-1. 系统复位电路 97
图 5-2. 时钟树 98
图 5-3. HXTAL 时钟源 99
图 5-4. 旁路模式下 HXTAL 时钟源 99
图 8-1. GPIO 端口位的基本结构 159
图 8-2. 输入配置的基本结构 161
图 8-3. 输出配置的基本结构 161
图 8-4. 模拟配置的基本结构 162
图 8-5. 备用功能配置的基本结构 162
图 10-1. CRC 计算单元框图 196
图 11-1. DMA 结构框图 202
图 11-2. 握手机制 204
图 11-3. DMA 中断逻辑图 206
图 12-1. DMAMUX 结构框图 213
图 12-2. 同步模式 215
图 12-3. 通道事件输出 216
图 14-1. ADC 模块框图 233
图 14-2. 单次运行模式 235
图 14-3. 连续转换模式 235
图 14-4. 扫描运行模式，且连续转换模式禁能 236
图 14-5. 扫描转换模式，连续转换模式使能 236
图 14-6. 间断运行模式 237
图 14-7. 12 位数据存储模式 238
图 14-8. 10 位数据存储模式 238
图 14-9. 8 位数据存储模式 238
图 14-10. 6 位数据存储模式 238
图 14-11. 20 位到 16 位的结果截断 241
图 14-12. 右移 5 位和取整的数例 241
图 14-13. ADC 同步框图 243

图 14-14. 基于 16 个通道的常规并行模式 243
图 14-15. 一个采用连续转换模式通道上的常规快速跟随模式 244
图 14-16. 一个通道上的常规慢速跟随模式 245
图 15-1. DAC 结构框图 258
图 15-2. DAC LFSR 算法 260
图 15-3. DAC 三角噪声模式生成的波形 261
图 16-1. 独立看门狗定时器框图 268
图 16-2. 窗口看门狗定时器框图 273
图 16-3. 窗口看门狗定时器时序图 274
图 17-1. RTC 框图 278
图 17-2. RTC 秒信号及闹钟信号的波形(RTC_PSC = 3,RTC_ALRM = 2) 279
图 17-3. RTC 秒信号及溢出信号的波形(RTC_PSC = 3) 279
图 18-1. 高级定时器结构框图 287
图 18-2. 内部时钟分频为 1 时，计数器的时序图 288
图 18-3. 当 PSC 数值从 0 变到 2 时，计数器的时序图 289
图 18-4. 向上计数时序图，PSC=0/2 290
图 18-5. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 290
图 18-6. 向下计数时序图，PSC=0/2 291
图 18-7. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值 292
图 18-8. 中央计数模式计数器时序图 293
图 18-9. 中央计数模式下计数器重复时序图 294
图 18-10. 在向上计数模式下计数器重复时序图 294
图 18-11. 在向下计数模式下计数器重复时序图 295
图 18-12. 通道 0 输入捕获原理 296
图 18-13. 通道 0 互补输入捕获原理 296
图 18-14. 输出比较原理（当 MCHxMSEL = 2’00，x=0,1,2,3） 297
图 18-15. 输出比较原理（当 MCHxMSEL = 2’01，x=0,1,2,3） 298
图 18-16. 输出比较原理（当 MCHxMSEL = 2’11 时互补输出，x=0,1,2,3） 298
图 18-17. 三种输出比较模式 299
图 18-18. EAPWM 时序图 300
图 18-19. CAPWM 时序图 301
图 18-20 通道 x 输出 PWM（CHxVAL < CHxCOMVAL_ADD） 302
图 18-21 通道 x 输出 PWM（CHxVAL = CHxCOMVAL_ADD） 303
图 18-22. 通道 x 输出 PWM（CHxVAL > CHxCOMVAL_ADD） 303
图 18-23. 通道 x 输出 PWM（CHxVAL 或 CHxCOMVAL_ADD > CARL） 304
图 18-24. 通道 x 输出 PWM 占空比随着 CHxCOMVAL_ADD 值而改变 304
图 18-25. 复合 PWM 模式下四通道输出 305
图 18-26. 边沿对齐模式下 CHx_O 输出脉冲（CHxOMPSEL≠2’b00） 305
图 18-27. 中央对齐模式下 CHx_O 输出脉冲（CHxOMPSEL≠2’b00） 306
图 18-28. 带死区时间的通道互补输出 308
图 18-29 中止功能图 309
图 18-30. 通道响应中止输入（高电平有效）时，输出信号的行为 310
图 18-31. 在正交译码器模式 2 且 CI0FE0 极性不反相时计数器行为 311
图 18-32. 在正交译码器模式 2 且 CI0FE0 极性反相时计数器行为 311

图 18-33. 霍尔传感器用在 BLDC 电机控制中 312
图 18-34. 两个定时器之间的霍尔传感器时序图 313
图 18-35. 复位模式下的控制电路 314
图 18-36. 暂停模式下的控制电路 314
图 18-37. 事件模式下的控制电路 315
图 18-38. 单脉冲模式，TIMERx_CHxCV = 0x04 TIMERx_CAR=0x99 315
图 18-39. 定时器 0 主/从模式的例子 316
图 18-40. 用定时器 1 的使能信号触发定时器 0 317
图 18-41. 用定时器 1 的 CI0 输入来触发定时器 0 和定时器 1 317
图 18-42. 通用定时器 L0 结构框图 371
图 18-43. 内部时钟分频为 1 时，计数器的时序图 372
图 18-44. 当 PSC 数值从 0 变到 2 时，计数器的时序图 373
图 18-45. 向上计数时序图，PSC=0/2 374
图 18-46. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 374
图 18-47. 向下计数时序图，PSC=0/2 375
图 18-48. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值 376
图 18-49. 中央计数模式计数器时序图 377
图 18-50. 输入捕获原理 378
图 18-51. 输出比较原理（x=0,1,2,3） 379
图 18-52. 三种输出比较模式 380
图 18-53. EAPWM 时序图 381
图 18-54. CAPWM 时序图 381
图 18-55. 在正交译码模式 2 且 CI0FE0 极性不反相时计数器行为 382
图 18-56. 在正交译码模式 2 且 CI0FE0 极性反相时计数器行为 383
图 18-57. 复位模式下的控制电路 384
图 18-58. 暂停模式下的控制电路 384
图 18-59. 事件模式下的控制电路 384
图 18-60. 单脉冲模式，TIMERx_CHxCV = 0x04 TIMERx_CAR=0x99 385
图 18-61. 基本定时器结构框图 407
图 18-62. 内部时钟分频为 1 时，计数器的时序图 408
图 18-63. 当 PSC 数值从 0 变到 2 时，计数器的时序图 408
图 18-64. 向上计数时序图，PSC=0/2 409
图 18-65. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 410
图 19-1. USART 模块内部框图 418
图 19-2. USART 字符帧（8 数据位和 1 停止位） 418
图 19-3. USART 发送步骤 420
图 19-4. 过采样方式接收一个数据位（OSB=0） 421
图 19-5. 采用 DMA 方式实现 USART 数据发送配置步骤 422
图 19-6. 采用 DMA 方式实现 USART 数据接收配置步骤 423
图 19-7. 两个 USART 之间的硬件流控制 423
图 19-8. 硬件流控制 424
图 19-9. 空闲状态下检测断开帧 425
图 19-10. 数据传输过程中检测断开帧 426
图 19-11. 同步模式下的 USART 示例 426

图 19-12. 8-bit 格式的 USART 同步通信波形（CLEN=1） 426
图 19-13. IrDA SIR ENDEC 模块 427
图 19-14. IrDA 数据调制 427
图 19-15. ISO7816-3 数据帧格式 428
图 19-16. USART 接收 FIFO 结构 430
图 19-17. USART 中断映射框图 432
图 20-1. I2C 模块框图 451
图 20-2. 数据有效性 452
图 20-3. 开始和停止信号 452
图 20-4. 10 位地址的 I2C 通讯流程（主机发送） 453
图 20-5. 7 位地址的 I2C 通讯流程（主机发送） 453
图 20-6. 7 位地址的 I2C 通讯流程（主机接收） 453
图 20-7. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R=0） 454
图 20-8. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R=1） 454
图 20-9. 数据保持时间 455
图 20-10. 数据建立时间 455
图 20-11. 数据发送 457
图 20-12. 数据接收 457
图 20-13. I2C 从机初始化 460
图 20-14. I2C 从机发送编程模型（SS=0） 461
图 20-15. I2C 从机发送编程模型（SS=1） 462
图 20-16. I2C 从机接收编程模型 463
图 20-17. I2C 主机初始化 464
图 20-18. I2C 主机发送编程模型（N<=255） 465
图 20-19. I2C 主机发送编程模型（N>255） 466
图 20-20. I2C 主机接收编程模型（N<=255） 467
图 20-21. I2C 主机接收编程模型（N>255） 468
图 20-22. SMBus 主机发送器和从机接收器通信流程 471
图 20-23. SMBus 主机接收器和从机发送器通信流程 472
图 21-1. SPI 结构框图 488
图 21-2. 常规模式下的 SPI 时序图 489
图 21-3. SPI 四线模式下的 SPI 时序图(CKPL=1, CKPH=1, LF=0) 490
图 21-4. 典型的全双工模式连接 492
图 21-5. 典型的单工模式连接（主机：接收，从机：发送） 492
图 21-6. 典型的单工模式连接（主机：只发送，从机：接收） 493
图 21-7. 典型的双向线连接 493
图 21-8. 主机 TI 模式在不连续发送时的时序图 494
图 21-9. 主机 TI 模式在连续发送时的时序图 495
图 21-10. 从机 TI 模式时序图 495
图 21-11. NSS 脉冲模式时序图（主机连续发送） 496
图 21-12. SPI 四线模式四线写操作时序图 497
图 21-13. SPI 四路模式四路读操作时序图 497
图 21-14. I2S 结构框图 500
图 21-15. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=0, CKPL=0） 501

图 21-16. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=0, CKPL=1） 502
图 21-17. I2S 飞利浦标准时序图（DTLEN=10, CHLEN=1, CKPL=0） 502
图 21-18. I2S 飞利浦标准时序图（DTLEN=10, CHLEN=1, CKPL=1） 502
图 21-19. I2S 飞利浦标准时序图（DTLEN=01, CHLEN=1, CKPL=0） 502
图 21-20. I2S 飞利浦标准时序图（DTLEN=01, CHLEN=1, CKPL=1） 502
图 21-21. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=1, CKPL=0） 503
图 21-22. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=1, CKPL=1） 503
图 21-23. MSB 对齐标准时序图（DTLEN=00, CHLEN=0, CKPL=0） 503
图 21-24. MSB 对齐标准时序图（DTLEN=00, CHLEN=0, CKPL=1） 503
图 21-25. MSB 对齐标准时序图（DTLEN=10, CHLEN=1, CKPL=0） 503
图 21-26. MSB 对齐标准时序图（DTLEN=10, CHLEN=1, CKPL=1） 504
图 21-27. MSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=0） 504
图 21-28. MSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=1） 504
图 21-29. MSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=0） 504
图 21-30. MSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=1） 504
图 21-31. LSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=0） 505
图 21-32. LSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=1） 505
图 21-33. LSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=0） 505
图 21-34. LSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=1） 505
图 21-35. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=0） 506
图 21-36. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=1） 506
图 21-37. PCM 标准短帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=0） 506
图 21-38. PCM 标准短帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=1） 506
图 21-39. PCM 标准短帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=0） 506
图 21-40. PCM 标准短帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=1） 506
图 21-41. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=0） 507
图 21-42. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=1） 507
图 21-43. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=0） 507
图 21-44. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=1） 507
图 21-45. PCM 标准长帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=0） 507
图 21-46. PCM 标准长帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=1） 507
图 21-47. PCM 标准长帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=0） 508
图 21-48. PCM 标准长帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=1） 508
图 21-49. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=0） 508
图 21-50. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=1） 508
图 21-51. I2S 时钟生成结构框图 508
图 21-52. I2S 初始化流程 510
图 21-53. I2S 主机接收禁能流程 512
图 22-1. 比较器框图 526
图 22-2. 比较器迟滞 528
图 22-3. 比较器的输出消隐 528
图 23-1. CAN 模块结构框图 534
图 23-2. 传输延迟 561
图 23-3. CAN 位时间 564

表索引

表 1-1. AHB 互联矩阵的互联关系列表 25
表 1-2. GD32A50x 系列器件的存储器映射表 28
表 2-1. 384KB 闪存基地址和大小 48
表 2-2. 256KB 闪存基地址和大小 49
表 2-3. 128KB 闪存基地址和大小 50
表 2-4. WSCNT 与 AHB 时钟频率对应关系（LDO= 1.1V） 52
表 2-5. 选项字节 0 62
表 2-6. 选项字节 1 63
表 2-7. OB_BK0WP 页保护 64
表 2-8. OB_BK1WP 页保护 64
表 2-9. OB_DFWP 页保护 65
表 2-10. PGSERR 条件 66
表 2-11. PGAERR 条件 67
表 2-12. PGERR 条件 67
表 2-13. WPERR 条件 67
表 3-1. 节电模式总结 86
表 5-1. 时钟源的选择 101
表 5-2. 深度睡眠模式下内核电压选择 102
表 6-1. Cortex®-M33 中的 NVIC 异常类型 129
表 6-2. 中断向量表 130
表 6-3. EXTI 触发源 133
表 7-1. 触发输入位域选择 139
表 7-2. TRIGSEL 输入输出映射关系 141
表 8-1. GPIO 配置表 158
表 11-1. DMA 传输操作 203
表 11-2. 中断事件 205
表 12-1. DMAMUX 信号 214
表 12-2. 中断事件 217
表 12-3. DMAMUX 请求路由输入信号映射 218
表 12-4. 触发输入信号映射 220
表 12-5. 同步输入信号映射 221
表 13-1. 调试引脚分配 226
表 14-1. ADC 内部信号 233
表 14-2. ADC 引脚定义 233
表 14-3. 常规序列外部触发源 239
表 14-4. 不同分辨率对应的 t_{CONV} 时间 240
表 14-5. N 和 M 的最大输出值（灰色部分表示截断） 241
表 14-6. ADC 同步模式表 242
表 15-1. DAC 引脚 258

表 15-2. DAC 触发与输出 259
表 15-3. DAC 外部触发 260
表 16-1. 独立看门狗定时器在 40kHz（IRC40K）时的最小 / 最大超时周期 268
表 16-2. 在 50MHz（f_{PCLK1}）时的最大/最小超时值 274
表 18-1. 定时器（TIMERx）分为三种类型 285
表 18-2. 高级定时器通道介绍 287
表 18-3 复合 PWM 脉冲宽度 301
表 18-4. 由参数控制的互补输出表(MCHxMSEL =2'b11) 307
表 18-5. 不同译码器模式下的计数方向 311
表 18-6. 从模式例子列表 313
表 18-7. 不同译码器模式下的计数方向 382
表 18-8. 从模式列表和举例 383
表 19-1. USART 重要引脚描述 417
表 19-2. 停止位配置 418
表 19-3. USART 中断请求 431
表 20-1. I2C 总线术语说明（参考飞利浦 I2C 规范） 451
表 20-2. 数据建立时间和数据保持时间 456
表 20-3. 可关闭通信模式 457
表 20-4. I2C 错误标志 472
表 20-5. I2C 中断事件 472
表 21-1. SPI 信号描述 488
表 21-2. SPI 四线信号描述 489
表 21-3. 从机模式 NSS 功能 490
表 21-4. 主机模式 NSS 功能 491
表 21-5. SPI 运行模式 491
表 21-6. SPI 中断请求 500
表 21-7. I2S 比特率计算公式 509
表 21-8. 音频采样频率计算公式 509
表 21-9. 各种运行模式下 I2S 接口信号的方向 509
表 21-10. I2S 中断 514
表 22-1 CMP 的输入和输出总结 526
表 23-1. 64 字节数据的邮箱描述符 535
表 23-2. DLC 表示的数据字节长度 536
表 23-3. 接收邮箱 CODE 537
表 23-4. 发送邮箱 CODE 537
表 23-5. 邮箱数目 539
表 23-6. 接收 FIFO 描述符 539
表 23-7. 当本地优先级禁用时的邮箱仲裁值（32 位） 548
表 23-8. 当本地优先级使能时的邮箱仲裁值（35 位） 548
表 23-9. 接收邮箱匹配 554
表 23-10. Rx FIFO 匹配 555
表 23-11. 中断事件 566
表 23-12. Rx FIFO 标识符过滤表元素数目 579
表 24-1. 寄存器功能位访问属性 595

表 24-2. 术语 595
表 25-1. 版本历史 596

1. 系统及存储器架构

GD32A50x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佛结构，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

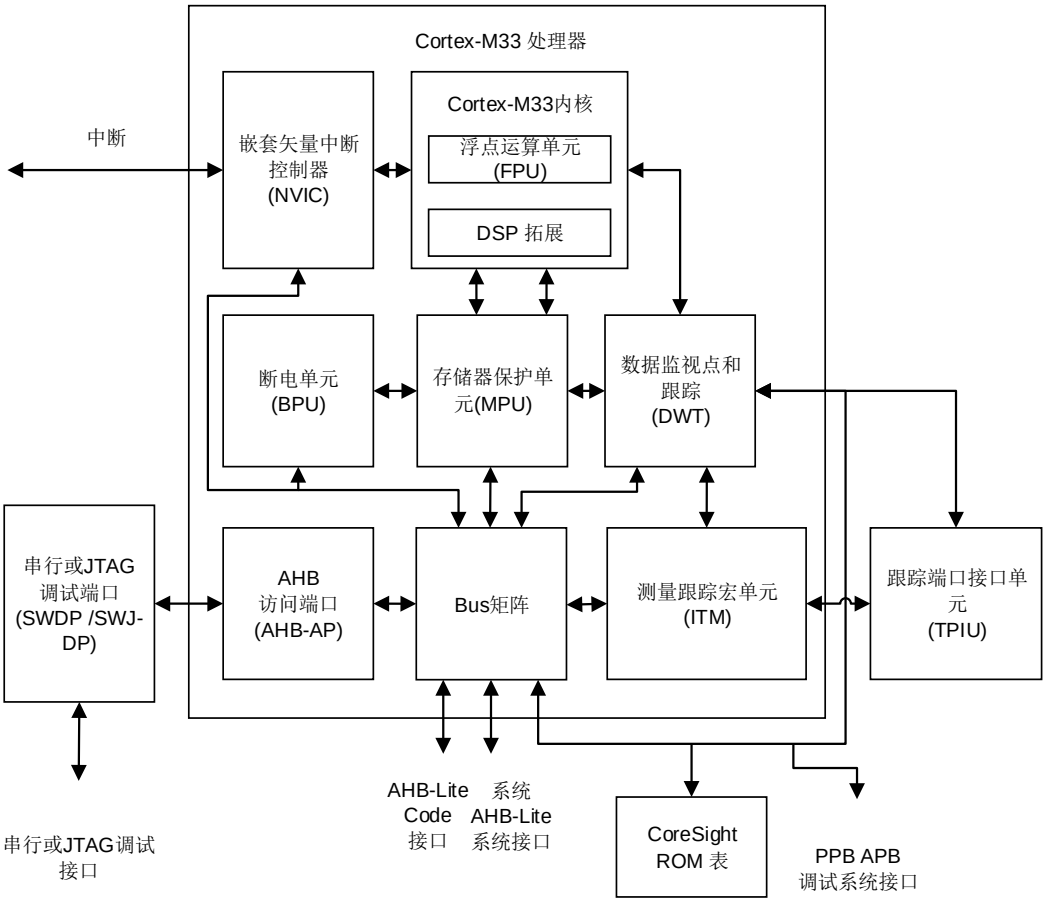
Cortex®-M33处理器是一个32位处理器，具有低中断延迟和低成本调试的特点。集成和先进的特性使Cortex®-M33处理器适合于需要高性能和低功耗微控制器的市场产品。Cortex®-M33处理器基于Armv8架构，支持强大的可扩展指令集，包括通用数据处理I/O控制任务、增强的数据处理位域操作和DSP。下面列出由Cortex®-M33提供的一些系统外围设：

- 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



注意：GD32A50x系列虽然有ITM模块，但不支持Trace数据的输出。

1.2. 系统架构

GD32A50x 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
FMC	1	0	1	1
SRAM	1	1	1	1
AHB1	0	1	1	1
AHB2	0	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 和 DMA1。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

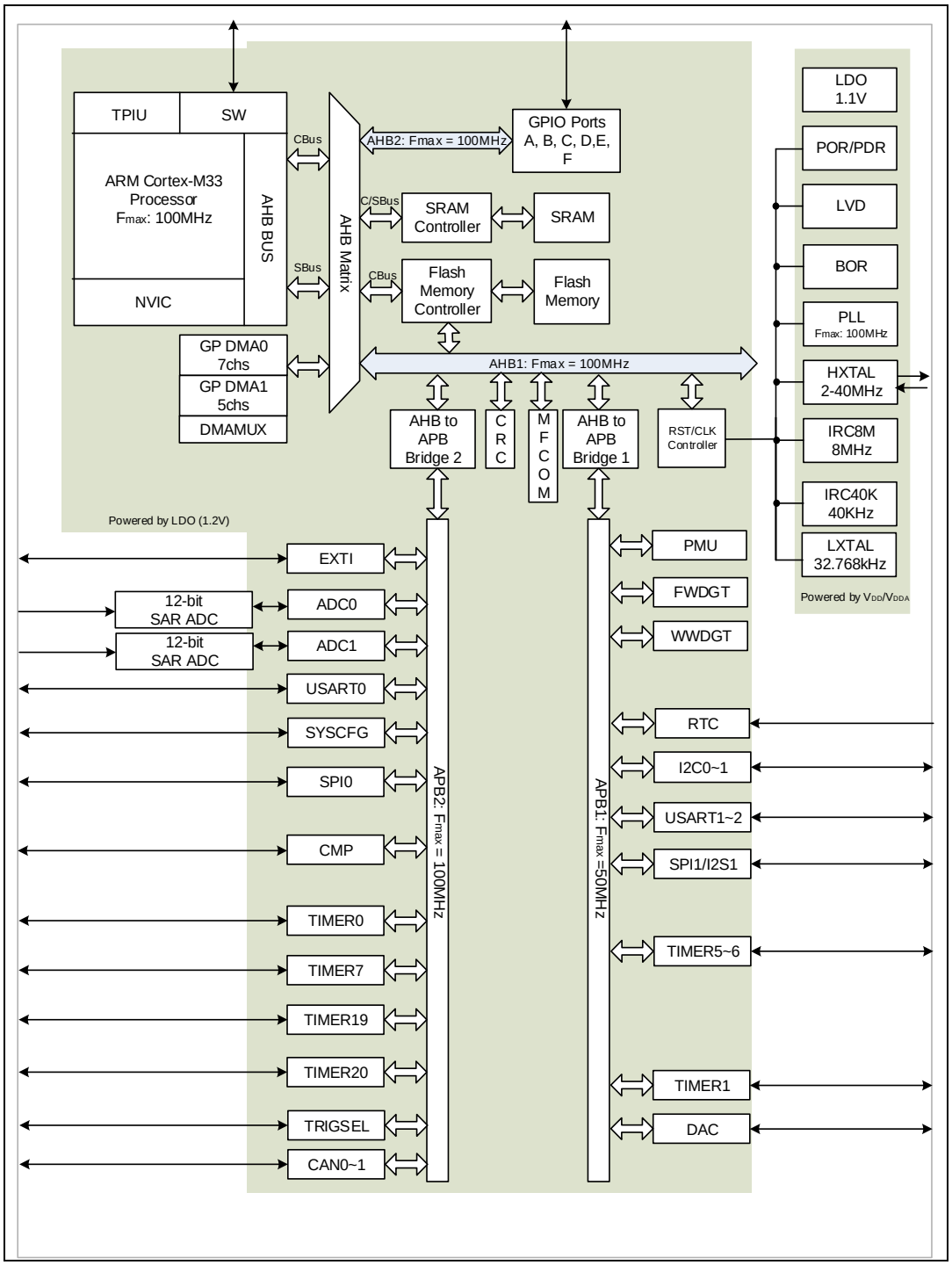
核的系统总线，用于指令和向量获取、数据加载和存储以及系统区域的调试访问。系统区域包括内部 SRAM 区域和外设区域。DMA0 和 DMA1 分别是 DMA0 和 DMA1 的存储器总线。

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。AHB1 是连接除 GPIO 之外所有 AHB 从机的 AHB 总线，AHB2 是连接 GPIO 的 AHB 总线。

APB1 和 APB2 是连接所有 APB 从机的两条 APB 总线。APB1 最高可达 50MHz，APB2 可以全速运行（最高可到 100MHz）。

这些是使用多层 AHB 总线架构互连的，如下图*图 1-2. GD32A50x 系列器件的系统架构示意图*所示。

图 1-2. GD32A50x 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.*

*GD32A50x 系列器件的存储器映射表*显示了 GD32A50x 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. GD32A50x 系列器件的存储器映射表

预先定义的地址空间	总线	地址	外设
		0xE004 4400 - 0xE00F FFFF	Cortex M33 内部外设
		0xE004 4000 – 0xE004 43FF	DBG
		0xE000 0000 – 0xE004 3FFF	Cortex M33 内部外设
外部 RAM		0x6000 0000 - 0x9FFF FFFF	保留
外设	AHB1	0x5000 0000 - 0x5FFF FFFF	保留
	AHB2	0x4800 1800 - 0x4FFF FFFF	保留
		0x4800 1400 - 0x4800 17FF	GPIOF
		0x4800 1000 - 0x4800 13FF	GPIOE
		0x4800 0C00 - 0x4800 0FFF	GPIOD
		0x4800 0800 - 0x4800 0BFF	GPIOC
		0x4800 0400 - 0x4800 07FF	GPIOB
		0x4800 0000 - 0x4800 03FF	GPIOA
	AHB1	0x4003 8C00 - 0x47FF FFFF	保留
		0x4003 8400 - 0x4003 8BFF	MFCOM
		0x4002 3400 - 0x4003 83FF	保留
		0x4002 3000 - 0x4002 33FF	CRC
		0x4002 2400 - 0x4002 2FFF	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	DMAMUX
		0x4002 0400 - 0x4002 07FF	DMA1
		0x4002 0000 - 0x4002 03FF	DMA0
	APB2	0x4001 C000 - 0x4001 FFFF	保留
		0x4001 B000 - 0x4001 BFFF	CAN1
		0x4001 A000 - 0x4001 AFFF	CAN0
		0x4001 8800 - 0x4001 9FFF	保留
		0x4001 8400 - 0x4001 87FF	TRIGSEL
		0x4001 8000 - 0x4001 83FF	保留
		0x4001 7C00 - 0x4001 7FFF	CMP
		0x4001 5800 - 0x4001 7BFF	保留
		0x4001 5400 - 0x4001 57FF	TIMER20
		0x4001 5000 - 0x4001 53FF	TIMER19

预先定义的地址空间	总线	地址	外设
		0x4001 4C00 - 0x4001 4FFF	保留
		0x4001 4800 - 0x4001 4BFF	保留
		0x4001 4400 - 0x4001 47FF	保留
		0x4001 4000 - 0x4001 43FF	保留
		0x4001 3C00 - 0x4001 3FFF	保留
		0x4001 3800 - 0x4001 3BFF	USART0
		0x4001 3400 - 0x4001 37FF	TIMER7
		0x4001 3000 - 0x4001 33FF	SPI0
		0x4001 2C00 - 0x4001 2FFF	TIMER0
		0x4001 2800 - 0x4001 2BFF	ADC1
		0x4001 2400 - 0x4001 27FF	ADC0
		0x4001 2000 - 0x4001 23FF	保留
		0x4001 1C00 - 0x4001 1FFF	保留
		0x4001 1800 - 0x4001 1BFF	保留
		0x4001 1400 - 0x4001 17FF	保留
		0x4001 1000 - 0x4001 13FF	保留
		0x4001 0C00 - 0x4001 0FFF	保留
		0x4001 0800 - 0x4001 0BFF	保留
		0x4001 0400 - 0x4001 07FF	EXTI
		0x4001 0000 - 0x4001 03FF	SYSCFG
		0x4000 DC00 - 0x4000 FFFF	保留
		0x4000 D800 - 0x4000 DBFF	保留
		0x4000 D400 - 0x4000 D7FF	保留
		0x4000 D000 - 0x4000 D3FF	保留
		0x4000 CC00 - 0x4000 CFFF	保留
		0x4000 C800 - 0x4000 CBFF	保留
		0x4000 C400 - 0x4000 C7FF	保留
		0x4000 C000 - 0x4000 C3FF	保留
		0x4000 8800 - 0x4000 BFFF	保留
		0x4000 8400 - 0x4000 87FF	保留
	APB1	0x4000 8000 - 0x4000 83FF	保留
		0x4000 7C00 - 0x4000 7FFF	保留
		0x4000 7800 - 0x4000 7BFF	保留
		0x4000 7400 - 0x4000 77FF	DAC0
		0x4000 7000 - 0x4000 73FF	PMU
		0x4000 6C00 - 0x4000 6FFF	BKP
		0x4000 6800 - 0x4000 6BFF	保留
		0x4000 6400 - 0x4000 67FF	保留
		0x4000 6000 - 0x4000 63FF	保留
		0x4000 5C00 - 0x4000 5FFF	保留
		0x4000 5800 - 0x4000 5BFF	I2C1

预先定义的地址空间	总线	地址	外设
		0x4000 5400 - 0x4000 57FF	I2C0
		0x4000 5000 - 0x4000 53FF	保留
		0x4000 4C00 - 0x4000 4FFF	保留
		0x4000 4800 - 0x4000 4BFF	USART2
		0x4000 4400 - 0x4000 47FF	USART1
		0x4000 4000 - 0x4000 43FF	保留
		0x4000 3C00 - 0x4000 3FFF	保留
		0x4000 3800 - 0x4000 3BFF	SPI1/I2S1
		0x4000 3400 - 0x4000 37FF	保留
		0x4000 3000 - 0x4000 33FF	FWDGT
		0x4000 2C00 - 0x4000 2FFF	WWDGT
		0x4000 2800 - 0x4000 2BFF	RTC
		0x4000 2400 - 0x4000 27FF	保留
		0x4000 2000 - 0x4000 23FF	保留
		0x4000 1C00 - 0x4000 1FFF	保留
		0x4000 1800 - 0x4000 1BFF	保留
		0x4000 1400 - 0x4000 17FF	TIMER6
		0x4000 1000 - 0x4000 13FF	TIMER5
		0x4000 0C00 - 0x4000 0FFF	保留
		0x4000 0800 - 0x4000 0BFF	保留
		0x4000 0400 - 0x4000 07FF	保留
		0x4000 0000 - 0x4000 03FF	TIMER1
SRAM		0x2000 D000 - 0x3FFF FFFF	保留
		0x2000 C000 - 0x2000 CFFF	共享 SRAM(4KB)
		0x2000 8000 - 0x2000 BFFF	SRAM(48KB)
		0x2000 6000 - 0x2000 7FFF	
		0x2000 4000 - 0x2000 5FFF	
		0x2000 0000 - 0x2000 3FFF	
代码		0x1FFF FC10 - 0x1FFF FFFF	保留
		0x1FFF FC00 - 0x1FFF FC0F	保留
		0x1FFF F818 - 0x1FFF BFFF	保留
		0x1FFF F800 - 0x1FFF F817	选项字节（24B）
		0x1FFF B000 - 0x1FFF F7FF	系统存储器（18KB）
		0x1FFF 7400 - 0x1FFF AFFF	保留
		0x1FFF 7000 - 0x1FFF 73FF	OTP(1KB)
		0x0A00 D000 - 0x1FFF 6FFF	保留
		0x0A00 C000 - 0x0A00 CFFF	共享 SRAM(4KB)
		0x0A00 8000 - 0x0A00 BFFF	SRAM(48KB)
		0x0A00 6000 - 0x0A00 7FFF	
		0x0A00 4000 - 0x0A00 5FFF	
		0x0A00 0000 - 0x0A00 3FFF	

预先定义的地址空间	总线	地址	外设
		0x08C0 1000 - 0x09FF FFFF	保留
		0x08C0 0000 - 0x08C0 0FFF	保留
		0x0881 0000 - 0x08BF FFFF	保留
		0x0880 0000 - 0x0880 FFFF	DFlash(64KB)
		0x0808 0000 - 0x0871 FFFF	保留
		0x0806 0000 - 0x0807 FFFF	保留
		0x0802 0000 - 0x0805 FFFF	主闪存
		0x0801 0000 - 0x0801 FFFF	
		0x0800 0000 - 0x0800 FFFF	
		0x0006 0000 - 0x07FF FFFF	保留
		0x0002 0000 - 0x0005 FFFF	别名为主闪存区域或 boot loader 区域
		0x0001 0000 - 0x0001 FFFF	
		0x0000 0000 - 0x0000 FFFF	

注意：0x20000000 - 0x2000BFFF和0x0A000000 - 0x0A00BFFF是同一块SRAM映射到的两个不同的逻辑地址。0x2000C000 - 0x2000CFFF和0x0A00C000 - 0x0A00CFFF是同一块共享SRAM映射到的两个不同的逻辑地址。

1.3.1. 片上 SRAM 存储器

GD32A50x 系列微控制器含有高达 48KB 的片上 SRAM。支持字节、半字（16 比特）和整字（32 比特）访问。

ECC

SRAM支持7比特的ECC功能。可纠错1比特，发现多比特（两比特）错误。

读之前必须先写入，否则很可能会导致ECC错误。非对齐的读操作会按照32比特的读操作来执行。非对齐的写操作会产生一个读改写的流程。例如，16比特写，首先会先读16比特，再和需要写入的16比特一起写入。所以初始化SRAM时，只能按照32位的来写入。

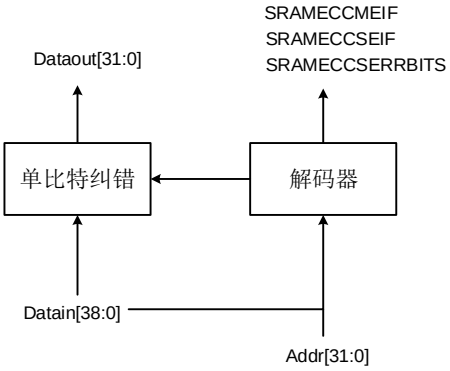
ECC 模块由编码器和解码器两部分构成：

编码器：在进行SRAM写操作时，会产生一个7比特的ECC码，和数据一起写入SRAM。

解码器：在进行 SRAM 读操作时，使用与编码器相同的算法，解码生成一个 7 比特的 ECC 码。ECC 码包括 ECC 错误状态和 32 位数据中哪位存在单比特位错误的信息。

解码器如下图所示：

图 1-3. ECC 解码器示意图



EEIC

EEIC（ECC Error Interrupt Control）模块提供了ECC错误状态管理和ECC中断配置的功能。

单比特可纠错事件

当检测到发生了单比特可纠错的事件时，EEIC：

(1) SYSCFG_STAT寄存器中的SRAMECCSEIF位置位，软件写1可以清除。
(2) SYSCFG_CFG3寄存器中记录发生单比特可纠错ECC事件的地址。

多比特（两比特）不可纠错事件

当检测到发生了多比特（两比特）不可纠错的事件时，EEIC：

(1) SYSCFG_STAT寄存器中的SRAMECCMEIF位置位，软件写1可以清除。
(2) SYSCFG_CFG3寄存器中记录发生两比特不可纠错ECC事件的地址。

单比特可纠错中断

在SYSCFG_CFG3寄存器中设置SRAMECCSEIE位，当检测到一个单比特可纠正错误事件时，将产生一个相应的中断。

多比特（两比特）不可纠错事件

在SYSCFG_CFG3寄存器中设置SRAMECCMEIE位。当检测到一个多比特（两比特）不可纠错错误事件时，将产生一个NMI中断。

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384KB的主快闪；
- 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

GD32A50x系列微控制器提供了最多两种引导源，可以通过BOOT0引脚来进行选择。该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的引脚电平。一旦这个引脚电平被采样，它可以被释放并用于其他用途。

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）。

嵌入式的Bootloader存放在系统存储空间，用于对FLASH存储器进行重新编程。在GD32A503xxxx和GD32A503xxxx0E设备中，Bootloader可以通过USART0（PA10和PA11），LIN（PA3和PA4），及CAN0（PB13和PB14）和外界交互。

注意：

（1）GD32A503xxxx支持从主FLASH、系统存储器启动。（由BOOT0决定，BOOT0 = 0时，从主FLASH启动；BOOT0 = 1时，从系统存储器启动。）

（2）GD32A503xxxx0E支持从主FLASH、系统存储器启动。（由BOOT0决定，BOOT0 = 0时，从主FLASH启动；BOOT0 = 1时，从系统存储器启动。）

（3）GD32A503xxxx1E仅支持从主FLASH启动。（固定从主FLASH启动，BOOT0可以为任意值 ）

1.5. 系统配置控制器

系统配置控制器（SYSCFG）的主要用途如下：

- 重新映射某些I/O口；
- 管理与外部中断线连接的GPIO。

1.6. 系统配置寄存器

SYSCFG基地址：0x4001 0000

1.6.1. 系统配置寄存器 0（SYSCFG_CFG0）

地址偏移：0x00
复位值：0x0000 000X（X表示BOOT_MODE能为任意值，取决于复位后BOOT0引脚的电平。）

该寄存器只能按字（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									BOOT0_PF0_RMP	保留	PA9_PA12_RMP	保留			BOOT_MODE
									rw		rw				r

位/位域	名称	描述
31:7	保留	必须保持复位值。
6	BOOT0_PF0_RMP	BOOT0 和 PF0 重映射使能 它控制 BOOT0 引脚映射到 BOOT0 或 PF0 功能。 0：无重映射（BOOT0 功能映射在 BOOT0 引脚上） 1：重新映射（PF0 功能映射在 BOOT0 引脚上）
5	保留	必须保持复位值。
4	PA9_PA12_RMP	小封装 PA9/PA12 重映射位控制位（32 管脚封装） 该位由软件设置和清除。它控制 PA9/12 或 PA10/11 针对在小引脚数封装上的映射。 0：不重映射（PA9/12 映射在引脚上） 1：重映射（PA9/12 映射而不是 PA10/11）
3:1	保留	必须保持复位值。
0	BOOT_MODE	引导模式 位 0 映射到 BOOT0 引脚 0：从片上闪存的主存引导启动 1：从片上闪存的系统存储器引导启动

1.6.2. 系统配置寄存器 1（SYSCFG_CFG1）

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
ADC0CH 9RMP	ADC0CH 8RMP	ADC1CH 15RMP	ADC1CH 14RMP	保留											
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	ADC0CH9RMP	ADC0 通道 9 重映射 0: ADC0_IN9 重映射到 PB1。 1: ADC0_IN9 重映射到 PC6。
30	ADC0CH8RMP	ADC0 通道 8 重映射 0: ADC0_IN8 重映射到 PB2。 1: ADC0_IN8 重映射到 PC7。
29	ADC1CH15RMP	ADC1 通道 15 重映射 0: ADC1_IN15 重映射到 PB13。 1: ADC1_IN15 重映射到 PD14。
28	ADC1CH14RMP	ADC1 通道 14 重映射 0: ADC1_IN14 重映射到 PB14。 1: ADC1_IN14 重映射到 PD15。
27:0	保留	必须保持复位值。

1.6.3. EXTI 源选择寄存器 0（SYSCFG_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 X000：PA3 引脚 X001：PB3 引脚 X010：PC3 引脚

		X011：PD3 引脚 X100：PE3 引脚 X101：PF3 引脚 X110：保留 X111：保留
11:8	EXTI2_SS[3:0]	EXTI 2 源选择 X000：PA2 引脚 X001：PB2 引脚 X010：PC2 引脚 X011：PD2 引脚 X100：PE2 引脚 X101：PF2 引脚 X110：保留 X111：保留
7:4	EXTI1_SS[3:0]	EXTI 1 源选择 X000：PA1 引脚 X001：PB1 引脚 X010：PC1 引脚 X011：PD1 引脚 X100：PE1 引脚 X101：PF1 引脚 X110：保留 X111：保留
3:0	EXTI0_SS[3:0]	EXTI 0 源选择 X000：PA0 引脚 X001：PB0 引脚 X010：PC0 引脚 X011：PD0 引脚 X100：PE0 引脚 X101：PF0 引脚 X110：保留 X111：保留

1.6.4. EXTI 源选择寄存器 1（SYSCFG_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 X000：PA7 引脚 X001：PB7 引脚 X010：PC7 引脚 X011：PD7 引脚 X100：PE7 引脚 X101：PF7 引脚 X110：保留 X111：保留
11:8	EXTI6_SS[3:0]	EXTI 6 源选择 X000：PA6 引脚 X001：PB6 引脚 X010：PC6 引脚 X011：PD6 引脚 X100：PE6 引脚 X101：PF6 引脚 X110：保留 X111：保留
7:4	EXTI5_SS[3:0]	EXTI 5 源选择 X000：PA5 引脚 X001：PB5 引脚 X010：PC5 引脚 X011：PD5 引脚 X100：PE5 引脚 X101：PF5 引脚 X110：保留 X111：保留
3:0	EXTI4_SS[3:0]	EXTI 4 源选择 X000：PA4 引脚 X001：PB4 引脚 X010：PC4 引脚 X011：PD4 引脚 X100：PE4 引脚 X101：PF4 引脚 X110：保留 X111：保留

1.6.5. EXTI 源选择寄存器 2（SYSCFG_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 X000：PA11 引脚 X001：PB11 引脚 X010：PC11 引脚 X011：PD11 引脚 X100：PE11 引脚 X101：保留 X110：保留 X111：保留
11:8	EXTI10_SS[3:0]	EXTI 10 源选择 X000：PA10 引脚 X001：PB10 引脚 X010：PC10 引脚 X011：PD10 引脚 X100：PE10 引脚 X101：保留 X110：保留 X111：保留
7:4	EXTI9_SS[3:0]	EXTI 9 源选择 X000：PA9 引脚 X001：PB9 引脚 X010：PC9 引脚 X011：PD9 引脚 X100：PE9 引脚 X101：保留 X110：保留 X111：保留
3:0	EXTI8_SS[3:0]	EXTI 8 源选择

X000：PA8 引脚
X001：PB8 引脚
X010：PC8 引脚
X011：PD8 引脚
X100：PE8 引脚
X101：保留
X110：保留
X111：保留

1.6.6. EXTI 源选择寄存器 3（SYSCFG_EXTISS3）

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
EXTI15_SS [3:0]				EXTI14_SS [3:0]				EXTI13_SS [3:0]				EXTI12_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	EXTI15_SS[3:0]	EXTI 15 源选择 X000：PA15 引脚 X001：PB15 引脚 X010：PC15 引脚 X011：PD15 引脚 X100：PE15 引脚 X101：保留 X110：保留 X111：保留
11:8	EXTI14_SS[3:0]	EXTI 14 源选择 X000：PA14 引脚 X001：PB14 引脚 X010：PC14 引脚 X011：PD14 引脚 X100：PE14 引脚 X101：保留 X110：保留 X111：保留
7:4	EXTI13_SS[3:0]	EXTI 13 源选择

		X000：PA13 引脚 X001：PB13 引脚 X010：PC13 引脚 X011：PD13 引脚 X100：PE13 引脚 X101：保留 X110：保留 X111：保留
3:0	EXTI12_SS[3:0]	EXTI 12 源选择 X000：PA12 引脚 X001：PB12 引脚 X010：PC12 引脚 X011：PD12 引脚 X100：PE12 引脚 X101：保留 X110：保留 X111：保留

1.6.7. 系统配置寄存器 1（SYSCFG_CFG2）

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													LVD_LOCK	SRAM_ECC_ERROR_LOCK	LOCKUP_LOCK
													rw	rw	rw

位/位域	名称	描述
31:3	保留	必须保持复位值。
2	LVD_LOCK	LVD 锁定 该位由软件置 1，在系统复位时才能清零。 0：LVD 中断从 TIMER0/7/19/20 的 break 输入端断开。PMU_CTL 寄存器的 LVDEN 和 LVDT[2:0]可以被设置 1：LVD 中断与 TIMER0/7/19/20 的 break 输入端连接。PMU_CTL 寄存器的 LVDEN 和 LVDT[2:0]仅仅可读
1	SRAM_ECC_ERROR_LOCK	SRAM ECC 校验错误锁定 该位由软件置 1，在系统复位时才能清零。

		0：SRAM ECC 校验错误从 TIMER0/7/19/20 的 break 输入端断开 1：SRAM ECC 校验错误与 TIMER0/7/19/20 的 break 输入端连接
0	LOCKUP_LOCK	Cortex-M33 LOCKUP 输出锁定 该位由软件置 1，在系统复位时才能清零。 0：Cortex-M33 LOCKUP 输出从 TIMER0/7/19/20 的 break 输入端断开 1：Cortex-M33 LOCKUP 输出与 TIMER0/7/19/20 的 break 输入端连接

1.6.8. 系统状态寄存器（SYSCFG_STAT）

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											NMIPINIF	CKMNMIIF	FLASHECCIF	SRAMECCSEIF	SRAMECCMEIF
											r	rc_w1	rc_w1	rc_w1	rc_w1

位/位域	名称	描述
31:5	保留	必须保持复位值。
4	NMIPINIF	外部 NMI 引脚中断标志 0：没有检测到外部 NMI 引脚中断标志。 1：检测到外部 NMI 引脚中断标志。
3	CKMNMIIF	HXTAL 时钟监测故障 NMI 中断状态 软件写 1 清零。 0：没有检测到 HXTAL 时钟监测故障。 1：检测到 HXTAL 时钟监测故障。
2	FLASHECCIF	Flash ECC 错误 NMI 中断标志 软件写 1 清零。 0：没有检测到 FLASH ECC 错误事件。 1：检测到 FLASH ECC 错误事件。
1	SRAMECCSEIF	SRAM 单比特可纠正错误中断标志 软件写 1 清零。 0：没有检测到 SRAM 单比特可纠正错误事件。 1：检测到 SRAM 单比特可纠正错误事件。
0	SRAMECCMEIF	SRAM 多比特（两比特）不可纠正错误中断标志 软件写 1 清零。 0：没有检测到 SRAM 多比特（两比特）不可纠正错误事件。 1：检测到 SRAM 多比特（两比特）不可纠正错误事件。

注意：当 SRAMECCMEIE 位置位时，SRAM 多比特（两比特）不可纠正错误会导致一个 NMI 中断。

1.6.9. 系统配置寄存器 3（SYSCFG_CFG3）

地址偏移：0x28

复位值：0xXXXX X00F

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRAMECCEADDR[13:0]														SRAMECCSERRBITS[5:4]	
r														r	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRAMECCSERRBITS[3:0]				保留							NMIPINIE	CKMNMIIE	FLASHECCIE	SRAMECCSEIE	SRAMECCMEIE
r											rw	rw	rw	rw	rw

位/位域	名称	描述
31:18	SRAMECCEADDR[13:0]	记录上一次发生 ECC 事件的 SRAM 故障系统地址（SRAMECCEADDR = 发生错误的 SRAM 地址[15:0] >> 2）。
17:12	SRAMECCSERRBITS[5:0]	发生 ECC 单比特可纠正错误的具体是哪个比特 0：无错误 1：比特 0 … 32：比特 31
11:5	保留	必须保持复位值。
4	NMIPINIE	外部 NMI 引脚中断使能 0：禁能外部 NMI 引脚中断。 1：使能外部 NMI 引脚中断。
3	CKMNMIIE	HXTAL 时钟检测故障 NMI 中断使能 0：禁能 HXTAL 时钟检测故障 NMI 中断。 1：使能 HXTAL 时钟检测故障 NMI 中断。
2	FLASHECCIE	Flash ECC 错误中断 NMI 使能 0：禁能 Flash ECC 错误 NMI 中断。 1：使能 Flash ECC 错误 NMI 中断。
1	SRAMECCSEIE	SRAM 单比特可纠正错误中断使能 0：禁能 SRAM 单比特可纠正错误中断。 1：使能 SRAM 单比特可纠正错误中断。
0	SRAMECCMEIE	SRAM 多比特（两比特）不可纠正错误 NMI 中断使能 0：禁能 SRAM 多比特（两比特）不可纠正错误 NMI 中断。

1：使能 SRAM 多比特（两比特）不可纠正错误 NMI 中断。

1.6.10. TIMER 输入源选择寄存器（SYSCFG_TIMERINSEL）

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
TIMER0_ETI_SEL [1:0]		TIMER7_ETI_SEL [1:0]		保留		TIMER19_ETI_SEL [1:0]		TIMER20_ETI_SEL [1:0]		TIMER0_ BRKIN0_ SEL	TIMER0_ BRKIN1_ SEL	TIMER0_ BRKIN2_ SEL	TIMER0_ BRKIN3_ SEL	TIMER7_ BRKIN0_ SEL	TIMER7_ BRKIN1_ SEL
rw		rw				rw		rw		rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER7_ BRKIN2_ SEL	TIMER7_ BRKIN3_ SEL	保留				TIMER19 _BRKIN0 _SEL	TIMER19 _BRKIN1 _SEL	TIMER19 _BRKIN2 _SEL	TIMER19 _BRKIN3 _SEL	TIMER20 _BRKIN0 _SEL	TIMER20 _BRKIN1 _SEL	TIMER20 _BRKIN2 _SEL	TIMER20 _BRKIN3 _SEL	保留	TIMER7_ CH0N_S EL
rw	rw					rw	rw	rw	rw	rw	rw	rw	rw		rw

位/位域	名称	描述
31:30	TIMER0_ETI_SEL[1:0]	TIMER0 外部输入源选择 00：定时器外部触发 0 01：定时器外部触发 1 10：定时器外部触发 2 11：保留
29:28	TIMER7_ETI_SEL[1:0]	TIMER7 外部输入源选择 00：定时器外部触发 0 01：定时器外部触发 1 10：定时器外部触发 2 11：保留
27:26	保留	必须保持复位值。
25:24	TIMER19_ETI_SEL[1:0]	TIMER19 外部输入源选择 00：定时器外部触发 0 01：定时器外部触发 1 10：定时器外部触发 2 11：保留
23:22	TIMER20_ETI_SEL[1:0]	TIMER20 外部输入源选择 00：定时器外部触发 0 01：定时器外部触发 1 10：定时器外部触发 2 11：保留
21	TIMER0_BRKIN0_S	TIMER0 刹车输入 0 选择

	EL	0：GPIO 引脚 1：TRIGSEL 模块
20	TIMER0_BRKIN1_SEL	TIMER0 刹车输入 1 选择 0：GPIO 引脚 1：TRIGSEL 模块
19	TIMER0_BRKIN2_SEL	TIMER0 刹车输入 2 选择 0：GPIO 引脚 1：TRIGSEL 模块
18	TIMER0_BRKIN3_SEL	TIMER0 刹车输入 3 选择 0：GPIO 引脚 1：TRIGSEL 模块
17	TIMER7_BRKIN0_SEL	TIMER7 刹车输入 0 选择 0：GPIO 引脚 1：TRIGSEL 模块
16	TIMER7_BRKIN1_SEL	TIMER7 刹车输入 1 选择 0：GPIO 引脚 1：TRIGSEL 模块
15	TIMER7_BRKIN2_SEL	TIMER7 刹车输入 2 选择 0：GPIO 引脚 1：TRIGSEL 模块
14	TIMER7_BRKIN3_SEL	TIMER7 刹车输入 3 选择 0：GPIO 引脚 1：TRIGSEL 模块
13:10	保留	必须保持复位值。
9	TIMER19_BRKIN0_SEL	TIMER19 刹车输入 0 选择 0：GPIO 引脚 1：TRIGSEL 模块
8	TIMER19_BRKIN1_SEL	TIMER19 刹车输入 1 选择 0：GPIO 引脚 1：TRIGSEL 模块
7	TIMER19_BRKIN2_SEL	TIMER19 刹车输入 2 选择 0：GPIO 引脚 1：TRIGSEL 模块
6	TIMER19_BRKIN3_SEL	TIMER19 刹车输入 3 选择 0：GPIO 引脚 1：TRIGSEL 模块
5	TIMER20_BRKIN0_SEL	TIMER20 刹车输入 0 选择 0：GPIO 引脚

		1：TRIGSEL 模块
4	TIMER20_BRKIN1_SEL	TIMER20 刹车输入 1 选择 0：GPIO 引脚 1：TRIGSEL 模块
3	TIMER20_BRKIN2_SEL	TIMER20 刹车输入 2 选择 0：GPIO 引脚 1：TRIGSEL 模块
2	TIMER20_BRKIN3_SEL	TIMER20 刹车输入 3 选择 0：GPIO 引脚 1：TRIGSEL 模块
1	保留	必须保持复位值。
0	TIMER7_CH0N_SEL	TIMER7 通道 0 互补输入选择 0：TIMER7_CH0N_IN 1：TIMER7_CH0_IN，TIMER7_CH0N_IN 和 TIMER0_CH0_IN 的异或

1.7. 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。

1.7.1. 存储容量信息

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。

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

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

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。

1.7.2. 设备唯一 ID（96 位/位域）

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。

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

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

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

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

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384K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384KB的片上闪存可用于存储指令或数据，高达1KB OTP，最大64KB扩展闪存。
  - bank0：256KB
  - bank1：128KB
  - 扩展块：64KB可用于数据闪存
  - OTP：1KB
  - 共享RAM：4KB用于基本SRAM或快速编程缓存区
- 双bank结构支持边读边写功能。
- ECC支持单个位错误纠正和双位错误检测。
- 在bank0 / bank1 / 数据闪存空间内，CPU执行指令或读数据需要0~3个等待时间。
- 预取缓存区以加速读操作。
- 1K字节的缓存，组织为4X64位的32缓存线。
- 闪存页大小为1KB。
- 支持64位双字编程，页擦除和整片擦除操作。
- 1KB OTP块（一次性编程），用于存储用户数据。
- 大小为24字节的选项字节可根据用户需求配置。
- 大小为4字节的选项字节1。
- 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。
- 支持快速编程。

2.3. 功能说明

2.3.1. 闪存结构

存储器包括一个高达384KB的主闪存，分为384页，每页大小为1KB、一个18KB的用于引导装载程序的信息块和高达64KB扩展闪存可用于数据闪存。主存储闪存的每页都可以单独擦除。基地址和大小如*表2-1. 384KB闪存基地址和大小*所示。

表 2-1. 384KB 闪存基地址和大小

闪存块		名称	地址范围	大小（字节）
主存储闪存块	bank0	第 0 页	0x0800 0000 - 0x0800 03FF	1KB

闪存块		名称	地址范围	大小（字节）
		第 1 页	0x0800 0400 - 0x0800 07FF	1KB
		第 2 页	0x0800 0800 - 0x0800 0BFF	1KB
		. . .	. . .	. . .
		第 255 页	0x0803 FC00 - 0x0803 FFFF	1KB
	bank1	第 256 页	0x0804 0000 - 0x0804 03FF	1KB
		第 257 页	0x0804 0400 - 0x0804 07FF	1KB
		第 258 页	0x0804 0800 - 0x0804 0BFF	1KB
		. . .	. . .	. . .
		第 383 页	0x0805 FC00 - 0x0805 FFFF	1KB
扩展闪存块		数据闪存(1)	0x0880 0000 - 0x0880 FFFF	64KB
共享 RAM		快速编程 SRAM	-	4KB
		基本 SRAM	参考*表 1-2. GD32A50x 系列器件的存储器映射表*	
信息块		引导装载程序(2)	0x1FFF B000 - 0x1FFF F7FF	18KB
选项字节块		选项字节 0	0x1FFF F800 - 0x1FFF F817	24B
		选项字节 1	0x4002 2068	4B
一次性编程块		OTP 字节(3)	0x1FFF 7000~0x1FFF 73FF	1KB

注意：1.当扩展闪存块用于数据闪存时，必须通过选项字节1进行配置。

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

3. 1KB（128双字）OTP（一次性编程）数据区域供用户使用，OTP区域仅由bank1寄存器操作。OTP数据不能被擦除，只能写一次。如果任何位被写为0，则该位所在的整个双字都不能被改写，即使值是0x0000 0000 0000 0000。

表 2-2. 256KB 闪存基地址和大小

闪存块		名称	地址范围	大小（字节）
主存储闪存块	bank0	第 0 页	0x0800 0000 - 0x0800 03FF	1KB
		第 1 页	0x0800 0400 - 0x0800 07FF	1KB
		第 2 页	0x0800 0800 - 0x0800 0BFF	1KB
		. . .	. . .	. . .
		第 255 页	0x0803 FC00 - 0x0803 FFFF	1KB
扩展闪存块		数据闪存(1)	0x0880 0000 - 0x0880 FFFF	64KB
共享 RAM		快速编程 SRAM	-	4KB
		基本 SRAM	参考*表 1-2. GD32A50x 系列器件的存储器映射表*	

闪存块	名称	地址范围	大小（字节）
信息块	引导装载程序(2)	0x1FFF B000 - 0x1FFF F7FF	18KB
选项字节块	选项字节 0	0x1FFF F800 - 0x1FFF F817	24B
	选项字节 1	0x4002 2068	4B
一次性编程块	OTP 字节(3)	0x1FFF 7000~0x1FFF 73FF	1KB

注意：1.当扩展闪存块用于数据闪存时，必须通过选项字节1进行配置。

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

3. 1KB（128双字）OTP（一次性编程）数据区域供用户使用，OTP区域仅由bank1寄存器操作。OTP数据不能被擦除，只能写一次。如果任何位被写为0，则该位所在的整个双字都不能被改写，即使值是0x0000 0000 0000 0000。

表 2-3. 128KB 闪存基地址和大小

闪存块		名称	地址范围	大小（字节）
主存储闪存块	bank0	第 0 页	0x0800 0000 - 0x0800 03FF	1KB
		第 1 页	0x0800 0400 - 0x0800 07FF	1KB
		第 2 页	0x0800 0800 - 0x0800 0BFF	1KB
		. . .	. . .	. . .
		第 127 页	0x0801 FC00 - 0x0801 FFFF	1KB
扩展闪存块		数据闪存(1)	0x0880 0000 - 0x0880 7FFF	32KB
共享 RAM		快速编程 SRAM	-	2KB
		基本 SRAM	参考*表 1-2. GD32A50x 系列器件的存储器映射表*	
信息块		引导装载程序(2)	0x1FFF B000 - 0x1FFF F7FF	18KB
选项字节块		选项字节 0	0x1FFF F800 - 0x1FFF F817	24B
		选项字节 1	0x4002 2068	4B
一次性编程块		OTP 字节(3)	0x1FFF 7000~0x1FFF 73FF	1KB

注意：1. 当扩展闪存块用于数据闪存时，必须通过选项字节1进行配置。

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

3. 1KB（128双字）OTP（一次性编程）数据区域供用户使用，OTP区域仅由bank1寄存器操作。OTP数据不能被擦除，只能写一次。如果任何位被写为0，则该位所在的整个双字都不能被改写，即使值是0x0000 0000 0000 0000。

2.3.2. 错误检查与纠正（ECC）

ECC机制支持：

- 单个位错误检测与纠正；

- 双位错误检测。

当单个位错误被检测与纠正时：

- 当错误发生在选项字节0（系统复位后，选项字节加载到寄存器中） / 选项字节1时，错误被纠正时不会有任何提示；
- 当错误发生在其他储存空间（包含读取选项字节0地址0x1FFFF80x）时，FMC_ECCCS寄存器中ECCCOR位将置1。如果FMC_ECCCS寄存器中ECCCORIE置1，将产生中断。OTP_ECC / DF_ECC / SYS_ECC / BK1_ECC / OB0_ECC位表示出现错误的存储空间。ECCADDR表示各个存储空间出现错误的偏移地址。

当检测到双位错误时：

- 如果错误发生在选项字节0（系统复位后，选项字节加载到寄存器中），FMC_ECCCS寄存器中OB0ECCDET位将置1。如果FMC_ECCCS寄存器中ECCDETIE置1，将产生中断。数据返回全F。
- 如果错误发生在选项字节1，FMC_ECCCS寄存器中OB1ECCDET位将置1。如果FMC_ECCCS寄存器中ECCDETIE置1，将产生中断。数据返回全F。
- 如果错误发生在其他存储空间（包含读取选项字节0地址0x1FFFF80x），FMC_ECCCS寄存器中ECCDET位和SYSCFG_STAT寄存器中FLASHECCIF位将置1，如果SYSCFG_CFG3寄存器中FLASHECCIE位置1，将产生NMI中断。OTP_ECC / DF_ECC / SYS_ECC / BK1_ECC / OB0_ECC表示出现错误的存储空间，ECCADDR表示各个存储空间出现错误的偏移地址。数据返回全F。将FMC_ECCCS寄存器中ECCDET位或SYSCFG_STAT寄存器中FLASHECCIF位置1，FMC_ECCCS寄存器中ECCDET位和SYSCFG_STAT寄存器中FLASHECCIF位将清零。

注意：1. 闪存中的数据是72位存储的，每个双字（64位）后加8位纠错码。8位纠错码由硬件自动计算，用户不可访问。

2. 对于原始数据0xFF FFFF FFFF FFFF FFFF，不支持ECC。

OTP_ECC / DF_ECC / SYS_ECC / BK1_ECC / OB0_ECC / ECCADDR表示在单个位或者双位错误中首次出现错误的位置。如果ECCCOR或ECCDET置1，即使有新的错误产生，这些位的值将不会改变。

当上报ECC错误时，如果数据仍然存在于当前的缓存区 / 预取缓存区 / 缓存中，即使清除ECCCOR和ECCDET，重新读取该错误地址也不会产生ECC错误。

2.3.3. 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 BUS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2-4. WSCNT与AHB时钟频率对应关系*。

表 2-4. WSCNT 与 AHB 时钟频率对应关系（LDO= 1.1V）

AHB时钟频率	WSCNT配置
<= 25MHz	0（0等待状态增加）
<= 50MHz	1（1等待状态增加）
<= 75MHz	2（2等待状态增加）
<= 100MHz	3（3等待状态增加）

如果发生系统复位，AHB时钟频率为8MHz，此时WSCNT置为0。

注意：

1. 如果希望增加AHB时钟频率。首先，参考*表2-4. WSCNT与AHB时钟频率对应关系（LDO= 1.1V）*，根据目标AHB时钟频率配置WSCNT位。然后，增加AHB时钟频率至目标频率。禁止在配置WSCNT位之前增加AHB时钟频率。

2. 如果希望降低AHB时钟频率。首先，降低AHB时钟频率至目标频率。然后，参考*表2-4. WSCNT与AHB时钟频率对应关系（LDO= 1.1V）*，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：100MHz时需添加3个等待状态）。为了加速读操作，需要用到以下功能。

当前缓存区：

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

预取缓存区：

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

指令 / 数据代码缓存区

通过使能FMC_WS寄存器的IDCEN位，使能缓存区。缓存区大小为1KB，由32条缓存线构成，每条缓存线是4 x 64位。

如果数据在缓存中（缓存命中），CPU可以零等待地从缓存区获取数据。如果数据不在缓存中（缓存没有命中），且当前数据不在当前缓存区和预取缓存区，缓存线提取闪存数据并复制到缓存区中。如果所有的缓存区都已填满，将使用LRU（最近最少使用）策略更新缓存。

通过DMA读取flash时，不可以用缓存。

2.3.4. 选项字节不可以用缓存。双 bank 边读边写特性

闪存的双bank结构基于bank（256KB）0和bank1（高达128KB） / 引导装载程序 / 数据闪存

/ OTP / 选项字节，该结构支持RWW（边读边写）特性，即当一个bank上有读操作或编程操作时，另一个bank无需等待其操作完成，就可以进行读操作或编程操作。

引导装载程序 / 数据闪存 / OTP / 选项字节的RWW特性与bank1相同。

2.3.5. FMC_CTLx 寄存器解锁

复位后，除了用于重加载选项字节的OBRLD位, FMC_CTLx（x=0,1）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_KEYx寄存器写入0x45670123和0xCDEF89AB，可以使得FMC_CTLx寄存器解锁。两次写操作后，FMC_CTLx寄存器的LK位被硬件清0。可以通过软件设置FMC_CTLx寄存器的LK位为1再次锁定FMC_CTLx寄存器。任何对FMC_KEYx寄存器的错误操作都会将LK位置1，从而锁定FMC_CTLx寄存器，并引发一个总线错误。

FMC_CTLx寄存器的OB0PG位和OB0ER位在FMC_CTL1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_CTL1寄存器的OBWEN位置1。软件可以将FMC_CTL1的OBWEN位清0来锁定FMC_CTL1的OB0PG位和OB0ER位。

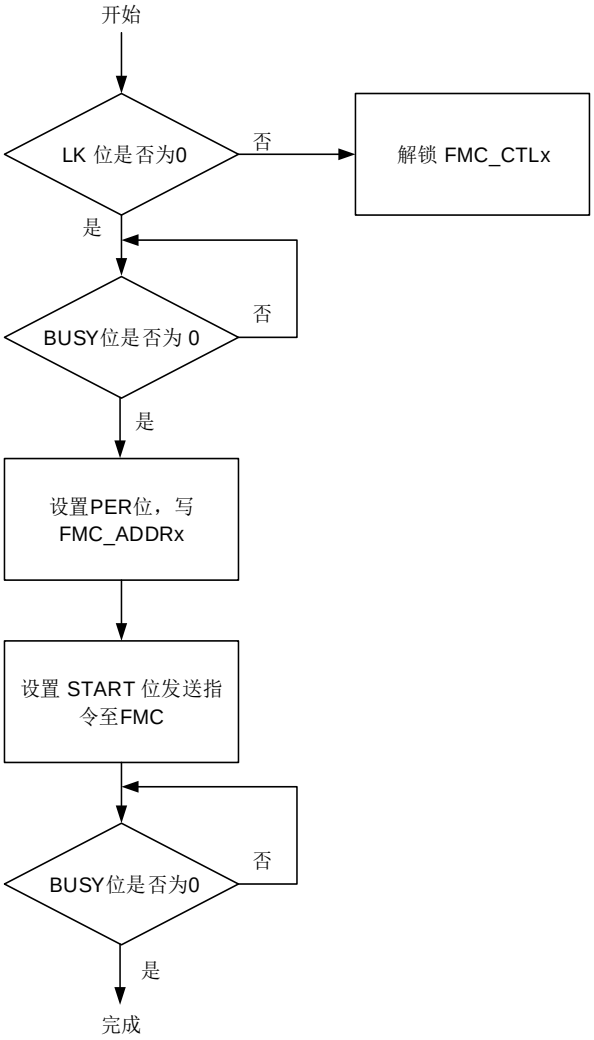
2.3.6. 页擦除

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_CTLx寄存器不处于锁定状态；
- 检查FMC_STATx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；
- 将待擦除页的绝对地址写到FMC_ADDRx寄存器；
- 置位FMC_CTLx寄存器的PER位；
- 通过将FMC_CTLx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_STATx寄存器的BUSY位清0；
- 如果需要，使用DBUS读并验证该页是否擦除成功。

当页擦除成功执行，FMC_STATx寄存器的ENDF位将置位。如果FMC_CTLx寄存器的ENDIE位置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不会有任何出错提示。另一方面，对擦写保护的页进行擦除操作将无效。如果FMC_CTLx寄存器的ERRIE位置1，将触发操作错误中断。中断服务程序可通过检测FMC_STATx寄存器的WPERR位来判断该中断是否发生。页擦除操作流程如*图2-1. 页擦除操作流程*所示。

图 2-1. 页擦除操作流程



2.3.7. 整片擦除

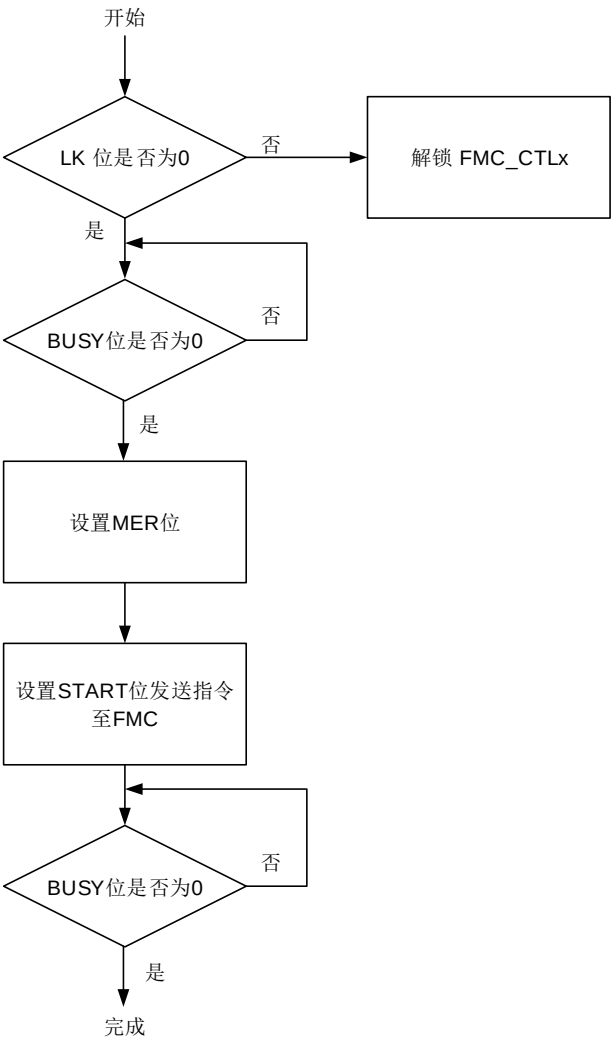
FMC提供了整片擦除功能可以初始化主存储闪存块的内容。整片擦除操作，寄存器设置具体步骤如下：

- 确保FMC_CTLx寄存器不处于锁定状态；
- 检查FMC_STATx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_CTLx寄存器的MER位；
- 通过将FMC_CTLx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_STATx寄存器的BUSY位清0；
- 如果需要，使用DBUS读并验证是否擦除成功。

当整片擦除成功执行，FMC_STATx寄存器的ENDF位置位。若FMC_CTLx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_CTLx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_STATx寄存器中的WPERR位来检测这种情况。整片擦除操作流程如*图2-2. 整片擦除操作流程*所示。

图 2-2. 整片擦除操作流程



注意：对bank1的全片擦除被硬件分为128个页擦除，所以擦除时间比bank0长。

2.3.8. 主存储闪存块编程

FMC提供了一个通过DBUS修改主存储闪存内容的32位整字编程功能。实际上，主存储闪存编程为64位。编程操作，寄存器设置具体步骤如下：

- 确保FMC_CTLx寄存器不处于锁定状态；
- 检查FMC_STATx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_CTLx寄存器的PG位；
- DBUS写数据到目的绝对地址（0x08XX XXXX）；
  DBUS写2次组成一个64位数据，即可将数据编程入闪存。待编程数据必须双字对齐。
- 等待编程指令执行完毕，FMC_STATx寄存器的BUSY位清0；

■ 如果需要，使用DBUS读并验证是否编程成功。

当主存储块编程成功执行，FMC_STATx寄存器的ENDF位置位。若FMC_CTLx寄存器的ENDIE位被置1，FMC将触发一个中断。双字编程操作之前需要检查目的地址是否已经被擦除。如果该地址没有被擦除，即使编程0x0，FMC_STATx寄存器的PGERR位也将被置1。另外，在被擦除 / 编程保护页上的编程操作会被忽略。如果若FMC_CTLx寄存器的ERRIE位被置1，将产生一个闪存操作错误中断。在中断服务程序中，软件可以通过检查FMC_STATx寄存器中的PGERR位来检测这种情况。

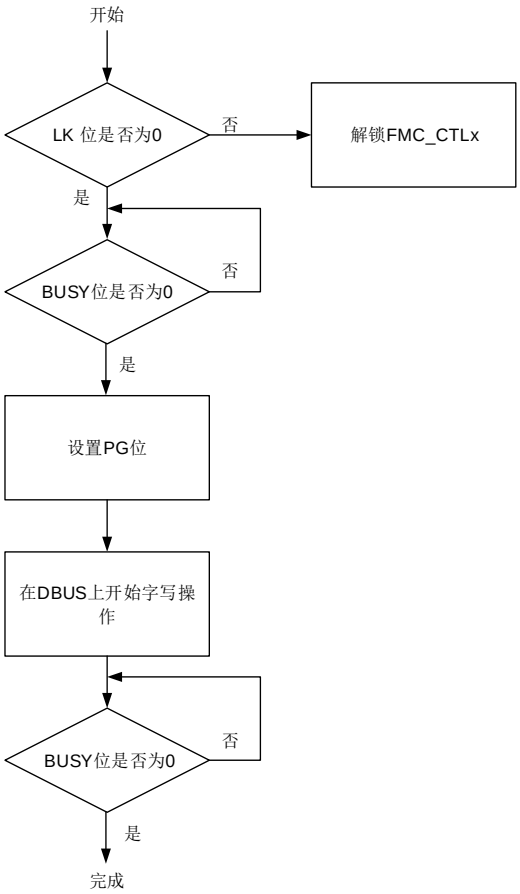
在下列情况下，FMC_STATx寄存器中的PGAERR位会被置位。

- DUBS不是按照32位编程。
- DBUS编程未对齐，如果采用DBUS 32位编程，DBUS第二次编程必须是双字对齐并且属于同一个双字地址范围。

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

在这些情况下，如果FMC_CTLx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_STATx寄存器中的PGERR位，PGAERR位或者WPERR位来检测发生了哪种错误。主存储块字编程操作流程如*图2-3. 字编程操作流程*所示。

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。

当编程一个双字时，从64位计算的ECC字节将添加在64位之后，这样，即使双字节是0xFFFF FFFF FFFF FFFF，每次编程的总位数也是72位。

如果编程 / 擦除操作被掉电、复位等意外中断，闪存中的内容将无法保证并处在一种不确定的状态。因此，应采取适当的措施，以避免由于程序中断 / 擦除而造成数据丢失。

2.3.9. 主存储闪存块快速编程

FMC通过DBUS总线提供快速编程功能，用于修改主闪存的内容。在这种模式下，可以在主闪存上编程一行（32个双字），消除编程前对闪存单元的验证，避免了每个字的高电压上升和下降时间，从而减少了页编程时间。

快速编程操作，寄存器设置具体步骤如下：

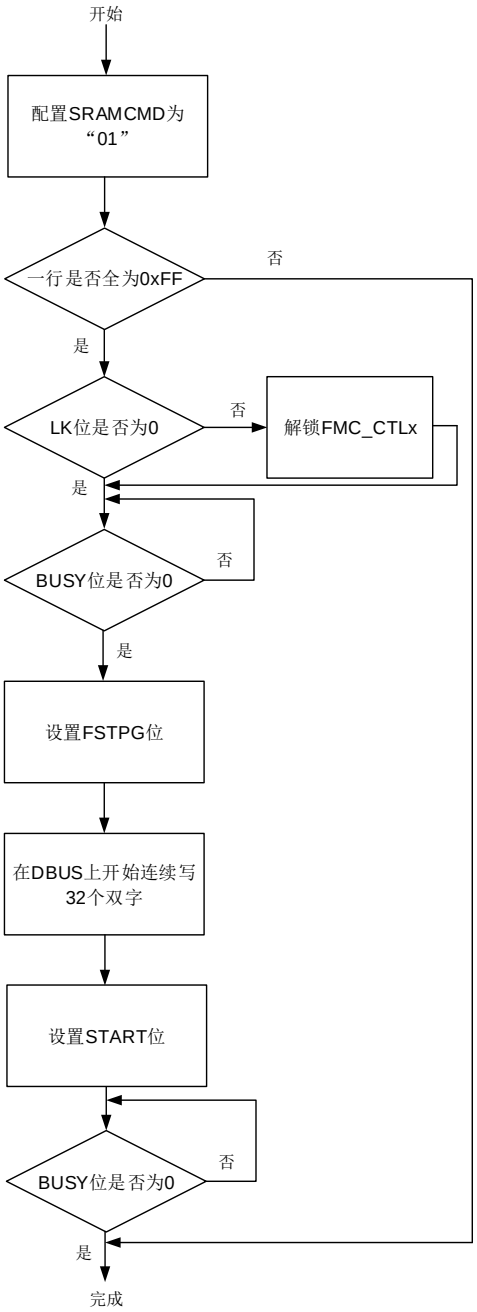
- 1、将SRAMCMD配置为“01”，使共享RAM进入快速编程模式；
- 2、检查闪存的一行（32个双字），确保为全FF，查空命令可用于检查行所在的页是否为全0xFF；
- 3、确保FMC_CTLx寄存器不处于锁定状态；
- 4、检查FMC_STATx寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；
- 5、置位FMC_CTLx寄存器的FSTPG位；
- 6、BUS写一行数据（32个双字）到目的绝对地址（0x08XX XXXX），这32个双字要连续写，否则，FMC_CTLx寄存器中PGERR位将在设置START位后置1；
- 7、设置START位启动闪存快速程序操作；
- 8、等待编程指令执行完毕，FMC_STATx寄存器的BUSY位清0；
- 9、如果需要，使用BUS读并验证是否编程成功。如果启用了缓存，建议清空缓存。

当主存储块快速编程成功执行，FMC_STATx寄存器的ENDF位置位。若FMC_CTLx寄存器的ENDIE位被置1，FMC将触发一个中断。在连续行快速编程时，如果确认行被擦除，重复6~8到快速编程下一行。

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

在这些情况下，如果FMC_CTLx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_STATx寄存器中的PGSERR / PGAERR / WPERR / PGERR位来检测发生了哪种错误。主存储块字编程操作流程如*图2-4. 快速编程操作流程*所示。

图 2-4. 快速编程操作流程



注意：

1. 32个双字必须连续写。

2. 32个双字必须对齐。

3. 由于快速编程无法通过硬件检查闪存是否为0xFF，因此软件必须首先检查闪存是否为0xFF，并且两次擦除之间不得对一行进行两次或多次编程。如果在两次擦除之间对一行编程两次或更多次，则可能会发生不可预测的结果。

4. 在设置FSTPG和START之间，读操作是允许的，但是读地址不应该是要编程的行地址，否则读出来的数据可能是旧数据。

5. 在进行快速编程之前必须清除cache。

2.3.10. 查空命令

查空命令用于检查FMC_ADDRx和FMC_CTLx寄存器中CBCMDLEN指定的闪存区域是否都是0xFF。通过配置FMC_CTLx寄存器的CBCMD位来配置查空命令，将FMC_CTLx寄存器中START位置1将发送查空命令，BUSY位将置1，硬件将检查闪存区域是否都是0xFF。如果闪存区域全部为0xFF，则设置ENDF，并清除BUSY位，否则CBCMDERR位将置1。仅bank0 / bank1 / 数据闪存支持查空命令。

注意：被检查的flash区域必须在一个页中，不超过1KB的边界。

2.3.11. 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。OTP区域仅能被bank1寄存器（FMC_KEY1 / FMC_STAT1 / FMC_CTL1 / FMC_ADDR1）操作。

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，闪存中的数据有很小可能性会出错。

2.3.12. 共享 RAM

4KB共享RAM可用于基本SRAM或者快速编程SRAM。

基本 SRAM

将SRAMCMD配置为“10”将发送基本SRAM命令。在发送基本SRAM命令后，共享RAM初始化为全“0”。如果基本SRAM准备就绪，FMC_WS寄存器中的BRAMRDY位将置1并且SRAMCMD位被清除。否则，等待命令发送完成。基本SRAM可以用作系统SRAM，无ECC纠错功能。

快速编程 SRAM

将SRAMCMD配置为“01”将发送基本快速编程SRAM命令。在发送快速编程SRAM命令后，共享RAM初始化为全“1”，如果快速编程SRAM准备就绪，FMC_WS寄存器中的PRAMRDY位将置1并且SRAMCMD位被清除。否则，等待命令发送完成。

注意：当配置SRAMCMD位时，必须要检查当前哪个SRAM可用。不能重复配置。例如，如果当前是基本SRAM，SRAMCMD就不能再配置为“10”。当前SRAM类型可以通过在FMC_WS寄存器中的BRAMRDY / PRAMRDY位来检查。

2.3.13. 数据闪存操作

数据闪存的大小通过选项字节1配置。

读

读数据闪存与读主闪存块方法一样。参考*读操作*。

编程

数据闪存编程与bank1编程方法一样。参考*主存储闪存块编程*。

页擦除

数据闪存页擦除与bank1页擦除方法一样。参考*页擦除*。

整片擦除

数据闪存整片擦除与bank1整片擦除方法一样。参考*整片擦除*。

注意：

1. 整片擦除被硬件分为多个页擦除，因此擦除时间比普通的整片擦除时间长。
2. 通过将FMC_CTL1寄存器中MERDF位置1启动整片擦除命令。

快速编程

数据闪存快速编程与bank1快速编程方法一样。参考*主存储闪存块快速编程*。

2.3.14. 选项字节 0 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0。选项字节0擦除过程如下所示。

- 确保FMC_CTL1寄存器不处于锁定状态；
- 检查FMC_STAT1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_CTL1寄存器的选项字节操作位OBWEN；
- 等待FMC_CTL1寄存器的OBWEN位置1；
- 置位FMC_CTL1寄存器的OB0ER位；
- 通过将FMC_CTL1寄存器的START位置1来发送选项字节0擦除命令到FMC；
- 等待擦除指令执行完毕，FMC_STAT1寄存器的BUSY位清0；
- 如果需要，使用DBUS读并验证是否擦除成功。

当选项字节0擦除成功执行，FMC_STAT1寄存器的ENDF位置位。若FMC_CTL1寄存器的ENDIE位被置1，FMC将触发一个中断。

2.3.15. 选项字节编程

选项字节 0 编程

FMC提供了64位双字编程功能，可用来修改选项字节0内容。编程操作过程如下：

- 确保FMC_CTL1寄存器不处于锁定状态；
- 检查FMC_STAT1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_CTL1寄存器的选项字节操作位OBWEN；
- 等待FMC_CTL1寄存器的OBWEN位置1；
- 置位FMC_CTL1寄存器的OB0PG位；
- DBUS写一个64位双字到目的地址；
- 等待编程指令执行完毕，FMC_STAT1寄存器的BUSY位清0；
- 如果需要，使用DBUS读并验证是否编程成功。

当选项字节0编程成功执行，FMC_STAT1寄存器的ENDF位置位。若FMC_CTL1寄存器的ENDIE位被置1，FMC将触发一个中断。注意在字编程操作之前检查待编程地址是否被擦除，如果未擦除，FMC_STAT1寄存器中的PGERR位将置1，即使编程数据为0x0。当编程操作完成时，FMC_STAT1寄存器中的ENDF位将置1。

选项字节 1 编程

编程操作过程如下：

- 确保FMC_CTL1寄存器不处于锁定状态；
- 检查FMC_STAT1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_CTL1寄存器的选项字节操作位OBWEN；
- 等待FMC_CTL1寄存器的OBWEN位置1；
- FMC_OB1CS寄存器中LKVAL / EPLOAD / EPSIZE / EFALC写入值；
- 将FMC_OB1CS寄存器中OB1START位置1；
- 等待编程指令执行完毕，FMC_STAT1寄存器的BUSY位清0；
- 在访问数据闪存之前执行系统复位或者将FMC_CTL1寄存器中OBRLD位置1；
- 如果需要，读并验证是否编程成功。

注意：

1. 在系统复位或者将FMC_CTL1寄存器中OBRLD位置1时，闪存存储器中的选项字节1加载到FMC_OB1CS寄存器中。

2. 修改选项字节1将在系统复位或者将FMC_CTL1寄存器中OBRLD位置1后生效。

3. 在系统复位或者将FMC_CTL1寄存器中OBRLD位置1后，如果LKVAL的值为0x33CC，且FMC_OB1CS寄存器中OB1LK位为1，选项字节1将不能再被修改。

4. 如果意料之外的值写入选项字节1，PGSERR位将在OB1START位置1之后被置1，并且该操作被忽略。

5. 在修改EPSIZE / EFALC时，扩展将先被硬件擦除，然后在修改选项字节1。

6. 在修改选项字节 1 后，强烈建议进行一次掉电复位。

2.3.16. 选项字节说明

选项字节 0 说明

每次系统复位或将FMC_CTL1寄存器中OBRLD位置1后，选项字节0被重加载到FMC_OBSTAT和FMC_WP0寄存器后，选项字节0生效。选项字节0的补字节具体为选项字节0取反。当选项字节0被重装载时，如果选项字节0的补字节和选项字节0不匹配，FMC_OBSTAT寄存器的OBERR位将被置1，选项字节0被强制设置为0xFF。选项字节0详情见*表2-5. 选项字节0*。

表 2-5. 选项字节 0

地址	名称	说明
0x1fff f800	OB_SPC	选项字节0安全保护值 0xA5：未保护状态 除0xA5和0xCC外的任何值：低级别保护 0xCC：高级别保护
0x1fff f801	OB_SPC_N	OB_SPC反码字节
0x1fff f802	OB_USER	[7]：BOR_TH（BOR复位阈值） 0：有BOR，复位阈值为2.6V（出厂值） 1：BOR关闭 [6:5]：保留 [4]: OTA 0: 当配置为从主存储块启动时，如果BB为0，bank1中所有数据将被拷贝到bank0，并从bank0启动 1: 无影响 [3]: BB 0：当OTA为1，并配置为从主存储块启动时，若bank1有启动程序，从bank1启动，否则从bank0启动 1：当配置为从主存储块启动时，从bank0启动 [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 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	OB_USER_N	OB_USER反码字节
0x1fff f804	OB_DATA[7:0]	用户定义数据7到0位
0x1fff f805	OB_DATA_N[7:0]	OB_DATA反码字节值的7到0位
0x1fff f806	OB_DATA[15:8]	用户定义数据15到8位
0x1fff f807	OB_DATA_N[15:8]	OB_DATA反码字节值的15到8位
0x1fff f808	OB_BK0WP[7:0]	bank0页擦除 / 编程保护值的7到0位

地址	名称	说明
		0：保护生效 1：未保护
0x1fff f809	OB_BK0WP_N[7:0]	OB_BK0WP反码字节值的7到0位
0x1fff f80a	OB_BK0WP[15:8]	bank0页擦除 / 编程的保护值的15到8位
0x1fff f80b	OB_BK0WP_N[15:8]	OB_BK0WP反码字节值的15到8位
0x1fff f80c	OB_BK0WP[23:16]	bank0页擦除 / 编程的保护值的23到16位
0x1fff f80d	OB_BK0WP_N[23:16]	OB_BK0WP反码字节值的23到16位
0x1fff f80e	OB_BK0WP[31:24]	bank0页擦除 / 编程的保护值的31到24位
0x1fff f80f	OB_BK0WP_N[31:24]	OB_BK0WP反码字节值的31到24位
0x1fff f810	OB_BK1WP[7:0]	bank1页擦除 / 编程保护值的7到0位
0x1fff f811	OB_BK1WP_N[7:0]	OB_BK1WP反码字节值的7到0位
0x1fff f812	OB_DFWP[7:0]	数据闪存页擦除 / 编程保护值的7到0位
0x1fff f813	OB_DFWP_N[7:0]	OB_DFWP反码字节值的7到0位
0x1fff f814	保留	必须保持0xFF。否则，可能会导致不可预知的后果。
0x1fff f815	保留	保存在地址0x1fff814值反码字节值的7到0位

选项字节 1 说明

选项字节1详情见*表2-6. 选项字节1*。

表 2-6. 选项字节 1

<table>
<tr><th>地址</th><th>名称</th><th>说明</th></tr>
<tr><td>[31:16]</td><td>LKVAL[15:0]</td><td>选项字节1锁定值
0x33CC：锁定选项字节1，选项字节1不能再被修改。
其他值：不锁定选项字节1。</td></tr>
<tr><td>[15]</td><td>EPLOAD</td><td>必须配置为0。否则，可能会导致不可预知的后果。</td></tr>
<tr><td>[11:8]</td><td>EPSIZE[3:0]</td><td>必须配置为0xF。否则，可能会导致不可预知的后果。</td></tr>
<tr><td>[7:4]</td><td>EFALC[3:0]</td><td>默认值为0xF。
当扩展内存块用于数据闪存时，这些位必须根据闪存容量进行如下配置。
如果配置了其他值，可能会导致不可预知的后果。
<table>
<tr><th>EFALC</th><th>闪存容量大小（KB）</th><th>数据闪存大小（KB）</th></tr>
<tr><td>0000</td><td>384/256</td><td>64</td></tr>
<tr><td>0001</td><td>128</td><td>32</td></tr>
<tr><td>1111</td><td colspan="2">默认值：扩展闪存没有配置有效的分区代码。</td></tr>
</table></td></tr>
</table>

注意：

1. 63:32 位为 31:0 位的反码字节。当加载选项字节时，如果选项字节与其反码不匹配，FMC_OB1CS 寄存器中 OB1ERR 位将置 1。

2. 当修改大小和分区时，整个扩展闪存将被硬件擦除。

3. 如果非法修改大小和分区，操作将被忽略并产生一个错误。

4. 建议在整个生命周期只对选项字节 1 进行一次配置。对选项字节 1 的修改将在系统复位或

FMC_CTL1 寄存器中 OBRLD 位置 1 后生效。

5. 在选项字节 1 的修改后建议进行掉电复位。

2.3.17. 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_STATx寄存器的WPERR位将被置1。如果WPERR位被置1且FMC_CTLx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。

bank0 页擦除 / 编程保护

bank0的页擦除 / 编程保护可以通过配置选项字节0的OB_BK0WP[31:0]某位为0来使能。如果对选项字节0进行页擦除操作，整个闪存的页擦除 / 编程保护都被禁能。将OB_BK0WP[31:0]置 1 或 者 清 0 都 需 要 软 件 将 FMC_CTL1 寄 存 器 中 OBRLD 位 置 1 或 系 统 复 位 重 载 OB_BK0WP[31:0]生效。设置OB_BK0WP[31:0]保护哪些页如*表2-7. OB_BK0WP页保护*所示。

表 2-7. OB_BK0WP 页保护

OB_BK0WP 位	页保护
OB_BK0WP[0]	BANK0_SIZE / 32
OB_BK0WP[1]	BANK0_SIZE / 32
OB_BK0WP[2]	BANK0_SIZE / 32
. . .	. . .
OB_BK0WP[30]	BANK0_SIZE / 32
OB_BK0WP[31]	BANK0_SIZE / 32

注意：BANK0_SIZE表示bank0的大小。

bank1 页擦除 / 编程保护

bank1的页擦除 / 编程保护可以通过配置选项字节0的OB_BK1WP[7:0]某位为0来使能。OB_BK1WP[7:0]每个位代表bank1的八分之一。如果对选项字节0进行页擦除操作，整个闪存的页擦除 / 编程保护都被禁能。将OB_BK1WP[7:0]置1或者清0都需要软件将FMC_CTL1寄存器中OBRLD位置1或系统复位重载OB_BK1WP[7:0]生效。设置OB_BK1WP[7:0]保护哪些页如*表2-8. OB_BK1WP页保护*所示。

表 2-8. OB_BK1WP 页保护

OB_BK1WP 位	页保护
OB_BK1WP[0]	BANK1_SIZE / 8
OB_BK1WP[1]	BANK1_SIZE / 8
. . .	. . .
OB_BK1WP[6]	BANK1_SIZE / 8

OB_BK1WP 位	页保护
OB_BK1WP[7]	BANK1_SIZE / 8

注意：1. BANK1_SIZE表示bank1的大小。

2. OTP写保护由OB_BK1WP[7]控制。

数据闪存页擦除 / 编程保护

数据闪存的页擦除 / 编程保护可以通过配置选项字节0的OB_DFWP[7:0]某位为0来使能。OB_DFWP[7:0]每个位代表数据闪存的八分之一。如果对选项字节0进行页擦除操作，整个闪存的页擦除 / 编程保护都被禁能。将OB_DFWP[7:0]置1或者清0都需要软件将FMC_CTL1寄存器中OBRLD位置1或系统复位重载OB_DFWP[7:0]生效。设置OB_DFWP[7:0]保护哪些页如*表2-9. OB_DFWP页保护*所示。

表 2-9. OB_DFWP 页保护

OB_DFWP 位	页保护
OB_DFWP[0]	DFLASH_SIZE / 8
OB_DFWP[1]	DFLASH_SIZE / 8
. . .	. . .
OB_DFWP[6]	DFLASH_SIZE / 8
OB_DFWP[7]	DFLASH_SIZE / 8

注意：DFLASH_SIZE表示数据闪存的大小。

2.3.18. 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。

安全保护等级划分为以下三种：

未保护状态：当将OB_SPC字节及其补字节被设置为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0可以被所有操作模式访问。

低等级保护：当设置OB_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。主存储块，OTP，数据闪存仅能被用户代码访问。在调试模式下，或从boot loader区启动时，对主存储块，OTP，数据闪存的操作都被禁止。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 / 擦除 / 查空操作，FMC_STATx寄存器的WPERR位将被置1。但这些模式下都可以对选项字节0进行操作。如果将OB_SPC字节及其补字节设置为0x5AA5，安全保护功能将失效，并自动触发一次整片擦除操作。

高等级保护：将OB_SPC字节及其补字节设置为0x33CC时，激活高等级安全保护。当编程选择该保护等级时，调试模式，或从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。OB_SPC字节及其补字节禁止再次编程。所以，如果高等级保护被激活，将不能再降回到低等级保护或未保护状态。

2.3.19. 错误描述

如果出现*表 2-10. PGSERR 条件*中一种情况，FMC_CTLx 寄存器中 PGSERR 位将置 1。

表 2-10. PGSERR 条件

模式	条件	操作
编程 / 快速编程	PG和FSTPG清零	写数据
快速编程	1. 不按地址顺序写 2. 不是从0写或没有写全32个双字 3. PRAMRDY未置1	设置START
编程	CBCMD / FSTPG / OB0ER / OB0PG / MERDF / MER / PER 未清除	设置PG
快速编程	CBCMD / PG / OB0ER / OB0PG / MERDF / MER / PER 未清除	设置FSTPG
修改选项字节1	CBCMD / FSTPG / OB0ER / OB0PG / MERDF / MER / PER / PG 未清除	设置OB1START
修改选项字节1	非有效EPSIZE / EFALC	设置OB1START
擦除选项字节0	CBCMD / FSTPG / OB0PG / MERDF / MER / PER / PG 未清除	设置OB0ER
编程选项字节0	CBCMD / FSTPG / OB0ER / MERDF / MER / PER / PG 未清除	设置OB0PG
全片擦除	CBCMD / OB0ER / OB0PG / MERDF / PER / PG 未清除	设置MER
数据闪存全片擦除	CBCMD / FSTPG / OB0ER / OB0PG / MER / PER / PG 未清除 无数据闪存	设置MERDF
页擦除	CBCMD / FSTPG / OB0ER / OB0PG / MERDF / MER / PG 未清除 FMC_ADDRx中数据无效	设置PER
查空	FMC_ADDRx和CBCMDLEN配置错误 1、超过1KB边界 2、无效地址	设置CBCMD
查空	FSTPG / OBER / OBPG / MERDF / MER / PER / PG 未清除	设置CBCMD

如果出现*表 2-11. PGAERR 条件*中一种情况，FMC_CTLx 寄存器中 PGAERR 位将置 1。

表 2-11. PGAERR 条件

模式	条件	操作
编程	1. DBUS编程不是32位写。 2. DBUS写未对齐。第一次DBUS写与第二次DBUS写必须双字对齐并且属于同一个双字地址范围。	写数据
快速编程	1. DBUS编程不是32位写。 2. DBUS写未对齐。第一次DBUS写与第二次DBUS写必须双字对齐并且属于同一个双字地址范围。	设置START

如果出现*表 2-12. PGERR 条件*中一种情况，FMC_CTLx 寄存器中 PGERR 位将置 1。

表 2-12. PGERR 条件

模式	条件	操作
编程	如果编程的地址未被擦除	写数据

如果出现*表 2-13. WPERR 条件*中一种情况，FMC_CTLx 寄存器中 WPERR 位将置 1。

表 2-13. WPERR 条件

模式	条件	操作
编程	1. 编程地址被选项字节0写保护。 2. 在调试模式下或从boot loader区启动时，低等级保护。	写数据
擦除	1. 编程地址被选项字节0写保护。 2. 在调试模式下或从boot loader区启动时，低等级保护。	设置START
查空	低级别保护并且程序从boot loader启动或在调试模式	设置START

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 0210

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	SLEEP_SLP	保留		IDRST	保留	IDCEN	保留				PFEN	保留	WSCNT[2:0]		
	rw			rw		rw					rw		rw		

位/位域	名称	描述
31:19	保留	必须保持复位值。
18	PRAMRDY	快速编程 SRAM 就绪标志。该位由硬件置 1，并且在 SRAMCMD 配置为基本 RAM 时清零。 0：快速编程 SRAM 未就绪。 1：快速编程 SRAM 就绪。
17	BRAMRDY	基本 SRAM 就绪标志。该位由硬件置 1，并且在 SRAMCMD 配置为快速编程 RAM 时清零。 0：基本 SRAM 未就绪。 1：基本 SRAM 就绪。
16:15	保留	必须保持复位值。
14	SLEEP_SLP	当 MCU 进入深度睡眠模式时，闪存进入睡眠模式还是掉电模式。 0：闪存进入掉电模式。 1：闪存进入睡眠模式。
13:12	保留	必须保持复位值。
11	IDRST	缓存复位 0：不复位缓存 1：当缓存失能时，复位缓存
10	保留	必须保持复位值。
9	IDCEN	缓存使能 0：缓存失能

		1：缓存使能
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 ~ 111：保留

2.4.2. ECC 控制和状态寄存器（FMC_ECCCS）

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
ECCDET	ECCCOR	保留		OB0ECCDET	OB1ECCDET	ECCDETIE	ECCCORIE	OTP_ECC	DF_ECC	SYS_ECC	BK1_ECC	OB0_ECC	保留		
rc_w1	rc_w1			rc_w1	rc_w1	rw	rw	r	r	r	r	r			

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

位/位域	名称	描述
31	ECCDET	检测到双位错误标志。当检测到双位错误时，该位置 1。当对 FMC_ECCCS 寄存器中 ECCDET 位或 SYSCFG_STAT 寄存器中 FLASHECCIF 位写 1 时，该位清零。 0：未检测到 ECC 双位错误。 1：检测到 ECC 双位错误。
30	ECCCOR	检测并纠正单个位错误标志 该位写 1 清零。 0：未检测并纠正 ECC 单个位错误。 1：检测并纠正 ECC 单个位错误。
29:28	保留	必须保持复位值。
27	OB0ECCDET	检测到选项字节 0 双位错误标志 该位写 1 清零。 0：未检测到选项字节 0 ECC 双位错误。

		1：检测到选项字节 0 ECC 双位错误。
26	OB1ECCDET	检测到选项字节 1 双位错误标志 该位写 1 清零。 0：未检测到选项字节 1 ECC 双位错误。 1：检测到选项字节 1 ECC 双位错误。
25	ECCDETIE	检测双位错误中断使能。当 EPECCDET，OB0ECCDET 或 OB1ECCDET 置 1，并且该位置 1，将产生一个中断。 0：检测双位错误中断失能 1：检测双位错误中断使能
24	ECCCORIE	纠正单个位中断使能 0：纠正单个位中断失能。 1：纠正单个位中断使能。
23	OTP_ECC	如果在 OTP 中检测到 ECC 错误，该位置 1。ECCADDR 会记录出错的 OTP 偏移地址。 0：在 OTP 中未检测到 ECC 错误。 1：在 OTP 中检测到 ECC 错误。
22	DF_ECC	如果在数据闪存中检测到 ECC 错误，该位置 1。ECCADDR 会记录出错的数据闪存偏移地址。 0：在数据闪存中未检测到 ECC 错误。 1：在数据闪存中检测到 ECC 错误。
21	SYS_ECC	如果在系统存储中检测到 ECC 错误，该位置 1。ECCADDR 会记录出错的系统存储偏移地址。 0：在系统存储中未检测到 ECC 错误。 1：在系统存储中检测到 ECC 错误。
20	BK1_ECC	如果在 bank1 中检测到 ECC 错误，该位置 1。ECCADDR 会记录出错的 bank1 偏移地址。 0：在 bank1 中未检测到 ECC 错误。 1：在 bank1 中检测到 ECC 错误。
19	OB0_ECC	如果在选项字节 0 中检测到 ECC 错误，该位置 1。ECCADDR 会记录出错的选项字节 0 偏移地址。 0：在选项字节 0 中未检测到 ECC 错误。 1：在选项字节 0 中检测到 ECC 错误。
18:15	保留	必须保持复位值。
14:0	ECCADDR[14:0]	检测到 ECC 错误的双字的偏移地址。 错误地址 = 基地址 + ECCADDR[14:0] * 8，基地址可以是 bank0，bank1，数据闪存，system 区，选项字节 0，选项字节 1 以及 OTP 的起始地址。详细信息参考 ***2.3.1***。

2.4.3. 解锁寄存器 0（FMC_KEY0）

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
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_CTL0 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_CTL0寄存器。

2.4.4. 状态寄存器 0（FMC_STAT0）

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
RSTERR	保留								CBCMDERR	ENDF	WPERR	PGAERR	PGERR	PGSERR	BUSY
rc_w1									rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	r

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	RSTERR	如果在编程或擦除时，电压低于 3.0V 或发生系统复位，将产生一个错误并且该位置 1。当错误发生时，当前地址的数据是不可靠的，需要重新擦除和编程。如果电压低于 BOR / POR，在 BOR / POR 复位后该位的值被复位，但是在发生系统复位后，该位的值保留。 **注意：**不建议在电压低于 3.0V 时进行 flash 编程 / 擦除操作。
14:7	保留	必须保持复位值。
6	CBCMDERR	查空命令检查的区域是否全为 0xFF。 0：检查的区域全为 0xFF。 1：检查的区域不是全为 0xFF。

5	ENDF	操作结束标志 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 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 当 DBUS 写数据不对齐时，此位被硬件置 1。软件写 1 清 0。
2	PGERR	编程错误标志 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	PGSERR	编程序列错误标志
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 0（FMC_CTL0）

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
CBCMDLEN[2:0]			保留												CBCMD
	rw														rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			ENDIE	保留	ERRIE	保留	FSTPG	LK	START	保留			MER	PER	PG
			rw		rw		rw	rs	rs				rw	rw	rw

位/位域	名称	描述
31:29	CBCMDLEN[2:0]	CBCMD 读数据长度 2^（CBCMDLEN）。 查空命令读数据长度。 读数据长度为 2^ CBCMDLEN 个双字。
28:17	保留	必须保持复位值。
16	CBCMD	查询选中区域是否为空命令。
15:13	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	保留	必须保持复位值。
8	FSTPG	主存储块快速编程命令位 软件置 1 和清 0 0：无作用 1：主存储块编程命令
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。
5:3	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 0（FMC_ADDR0）

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
ADDR[31:16]															
W															

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

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. 选项字节操作解锁寄存器（FMC_OBKEY）

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
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_CTL1 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_CTL1寄存器的选项字节命令。

2.4.8. 解锁寄存器 1（FMC_KEY1）

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
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_CTL1 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_CTL1寄存器。

2.4.9. 状态寄存器 1（FMC_STAT1）

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
RSTERR	保留								CBCMDERR	ENDF	WPERR	PGAERR	PGERR	PGSERR	BUSY
rc_w1									rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	r

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	RSTERR	如果在编程或擦除时，电压低于 3.0V 或发生系统复位，将产生一个错误并且该位置 1。当错误发生时，当前地址的数据是不可靠的，需要重新擦除和编程。如果电压低于 BOR / POR，在 BOR / POR 复位后该位的值被复位，但是在发生系统复位后，该位的值保留。 **注意：**不建议在电压低于 3.0V 时进行 flash 编程/擦除操作。
14:7	保留	必须保持复位值。
6	CBCMDERR	查空命令检查的区域是否全为 0xFF。 0：检查的区域全为 0xFF。 1：检查的区域不是全为 0xFF。
5	ENDF	操作结束标志 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 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 当 DBUS 写数据不对齐时，此位被硬件置 1。软件写 1 清 0。
2	PGERR	编程错误标志 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	PGSERR	编程序列错误标志
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。

2.4.10. 控制寄存器 1（FMC_CTL1）

地址偏移：0x50

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
CBCMDLEN[2:0]			保留			SRAMCMD[1:0]		保留							CBCMD

	rw						rw								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		OBRLD	ENDIE	保留	ERRIE	OBWEN	FSTPG	LK	START	OB0ER	OB0PG	MERDF	MER	PER	PG
		rw	rw		rw	rw	rw	rs	rs	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:29	CBCMDLEN[2:0]	CBCMD 读数据长度 2^（CBCMDLEN）。 查空命令读数据长度。 读数据长度为 2^ CBCMDLEN 个双字。
27:26	保留	必须保持复位值。
25:24	SRAMCMD[1:0]	共享 RAM 命令。这些位由软件设置，在 PRAMRDY 或 BRAMRDY 置 1 时清零。 00：无操作 01：设置快速编程 RAM 模式 10：设置基本 RAM 模式 11：保留。如果配置了该值，可能会导致不可预知的后果。
23:17	保留	必须保持复位值。
16	CBCMD	查询选中区域是否为空命令。
15:14	保留	必须保持复位值。
13	OBRLD	选项字节重加载位 软件置 1。 0：无作用 1：强制选项字节重装载。
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	FSTPG	主存储块快速编程命令位 软件置 1 和清 0 0：无作用 1：主存储块编程命令
7	LK	FMC_CTL1 寄存器锁定标志位

		当正确的序列写入 FMC_KEY1 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	OB0ER	选项字节 0 擦除命令位 软件置 1 和清 0 0：无作用 1：选项字节 0 擦除命令
4	OB0PG	选项字节 0 编程命令位 软件置 1 和清 0 0：无作用 1：选项字节 0 编程命令位
3	MERDF	数据闪存整片擦除命令位 软件置 1 和清 0 0：无作用 1：数据闪存整片擦除命令
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。

2.4.11. 地址寄存器 1（FMC_ADDR1）

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
ADDR[31:16]															
W															

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

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

2.4.12. 选项字节状态寄存器（FMC_OBSTAT）

地址偏移：0x5C
复位值：0x0XXX XX0X

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:16	DATA[15:0]	系统复位后保存选项字节的 OB_DATA[15:0]部分
15:8	USER[7:0]	系统复位后保存选项字节块的 OB_USER 字节
7:3	保留	必须保持复位值
2:1	PLEVEL[1:0]	安全保护级别 00：无保护 01：低等级保护 10：保留 11：高等级保护
0	OBERR	选项字节 0 错误位 当选项字节 0 和它的反码不匹配时此位由硬件置 1，并且选项字节被设置为 0xFF。

2.4.13. 擦除 / 编程保护寄存器 0（FMC_WP0）

地址偏移：0x60
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
BK0WP[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BK0WP[15:0]															

r

位/位域	名称	描述
31:0	BK0WP[31:0]	系统复位后保存选项字节 0 的 OB_BK0WP[31:0]部分

2.4.14. 擦除 / 编程保护寄存器 1（FMC_WP1）

地址偏移：0x64

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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DFWP[7:0]								BK1WP[7:0]							
r								r							

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	DFWP[7:0]	系统复位后保存选项字节 0 的 OB_DFWP[7:0]。
7:0	BK1WP[7:0]	系统复位后保存选项字节 0 的 OB_BK1WP[7:0]。

2.4.15. 选项字节 1 控制和状态寄存器（FMC_OB1CS）

地址偏移：0x68

复位值：0xXXXX XX0X

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPLOAD	保留			EPSIZE[3:0]				EFALC[3:0]				保留	OB1LK	OB1START	OB1ERR
rw				rw				rw					r	rw	r

位/位域	名称	描述
31:16	LKVAL[15:0]	系统复位后加载选项字节 1 的 LKVAL 值。当 OB1LK 为 0 时，该位域可写。
15	EPLOAD	系统复位后加载选项字节 1 的 EPLOAD 值。当 OB1LK 为 0 时，该位可写。
14:12	保留	必须保持复位值。

11:8	EPSIZE[3:0]	系统复位后加载选项字节 1 的 EPSIZE 值。当 OB1LK 为 0 时，该位域可写。
7:4	EFALC[3:0]	系统复位后加载选项字节 1 的 EFALC 值。当 OB1LK 为 0 时，该位域可写。
3	保留	必须保持复位值。
2	OB1LK	当 LKVAL 为 0x33CC 时，OB1LK 位置 1，FMC_OB1CS 将不能再被配置。
1	OB1START	发送 FMC 选项字节 1 修改命令。 当 BUSY 清零时，由软件置 1 或清零。
0	OB1ERR	选项字节读错误位 当选项字节和其反码字节不匹配时，该位由硬件置 1，并且选项字节 1 强制为 0xFFFF FFFF。

2.4.16. 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. 电源管理单元（PMU）

3.1. 简介

功耗设计是GD32A50x系列产品最注重的问题之一。电源管理单元提供了三种省电模式，包括睡眠模式、深度睡眠模式和待机模式。这些模式能减少电源能耗，且使得应用程序可以在CPU运行时间要求、速度和功耗的相互冲突中获得最佳折衷。如*图3-1. 电源域概览*所示，GD32A50x系列设备有两个电源域，包括V_{DD} / V_{DDA}域、1.1V域。V_{DD}域由电源直接供电。在V_{DD} / V_{DDA}域中嵌入了一个LDO，用来为1.1V域供电。

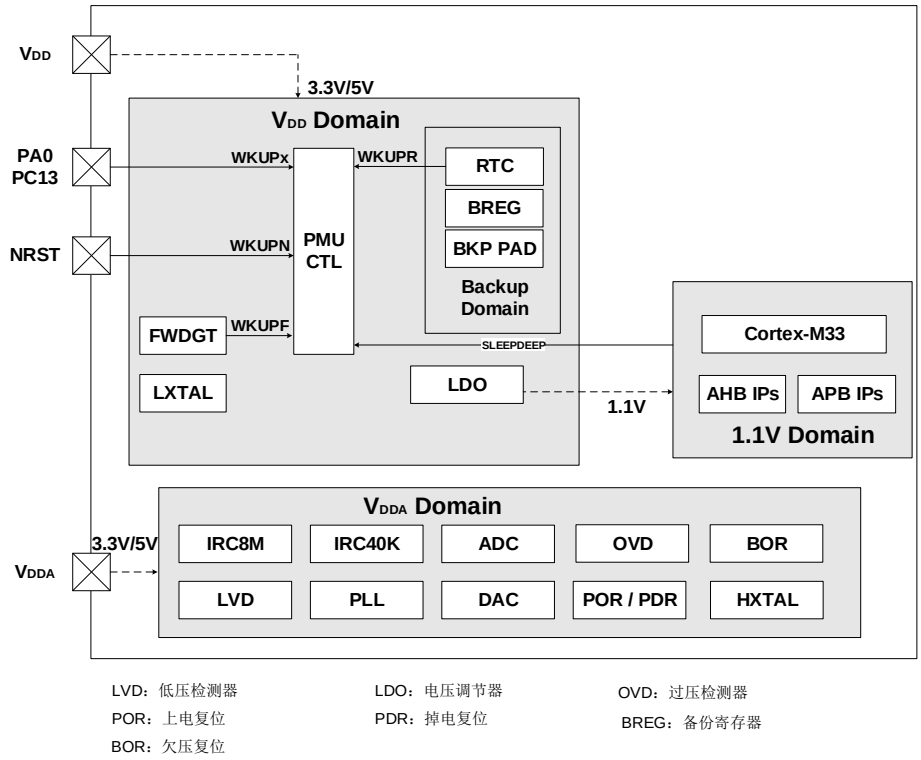
3.2. 主要特征

- 两个电源域：V_{DD} / V_{DDA}域和1.1V电源域；
- 三种省电模式：睡眠模式、深度睡眠模式和待机模式；
- 内部电压调节器（LDO）提供1.1V电源；
- 提供低电压检测器（LVD），当电压低于所设定的阈值时能发出中断或事件；
- 提供过电压检测器（OVD），当电压高于所设定的阈值时能发出中断或事件。

3.3. 功能说明

*图3-1. 电源域概览*提供了PMU及相关电源域的内部结构框图。

图3-1. 电源域概览



3.3.1. 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} 域包括备份域、LXTAL（低速外部晶体振荡器）、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。V_{DDA} 域包括 ADC / DAC（AD / DA 转换器）、IRC8M（内部 8M RC 振荡器）、IRC40K（内部 40KHz RC 振荡器）、PLL（锁相环）、HXTAL（高速外部晶体振荡器）、POR / PDR（上电/掉电复位）、LVD（低电压检测器）和 OVD（过电压检测器）等等。

V_{DD} 域

为 1.1V 域供电的 LDO（电压调节器），其复位后保持使能。可以被配置为三种不同的工作状态：包括睡眠模式（全供电状态）、深度睡眠模式（全供电或低功耗状态）和待机模式（关闭状态）。

电池备份域

电池备份域位于V_{DD}域。由V_{DD}为RTC（实时时钟）、LXTAL（低速外部晶体振荡器）、BPOR（备份域上电复位）、BREG（备份寄存器），以及PC13至PC15共3个BKP PAD供电。

应用软件可以通过设置RCU_BDCTL寄存器BKPRST位来触发备份域软件复位。

RTC的时钟源可以是低速内部RC振荡器（IRC40K）或低速外部晶体振荡器（LXTAL），或高速外部晶体振荡器（HXTAL）时钟的128分频。在通过WFI / WFE指令进入省电模式之前，Cortex®-M33需要通过RTC寄存器设置预期的唤醒时间并启用唤醒功能，以实现RTC定时器唤醒事件。进入省电模式一定时间之后，当经过的时间与预设的唤醒时间匹配时，RTC将唤醒设备。RTC的配置和操作的细节将在*实时时钟（RTC）*章节描述。

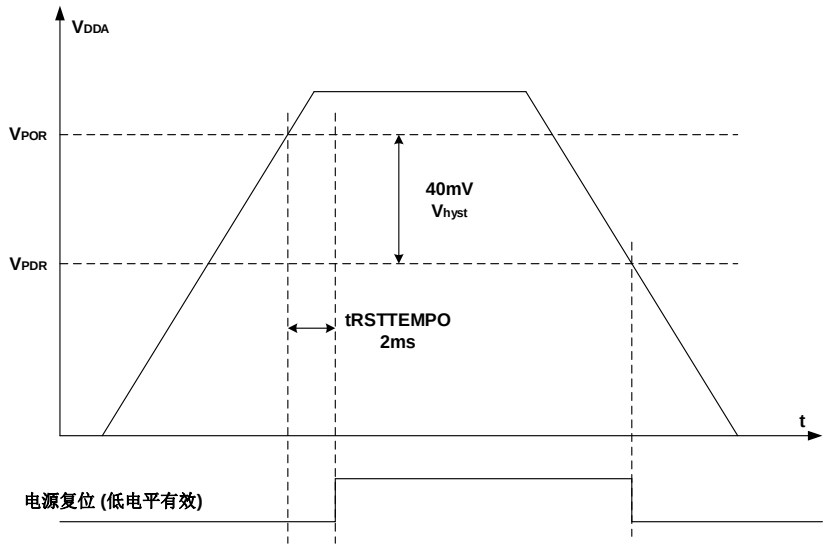
当备份域由V_{DD}供电时，以下功能可用：

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

V_{DDA} 域

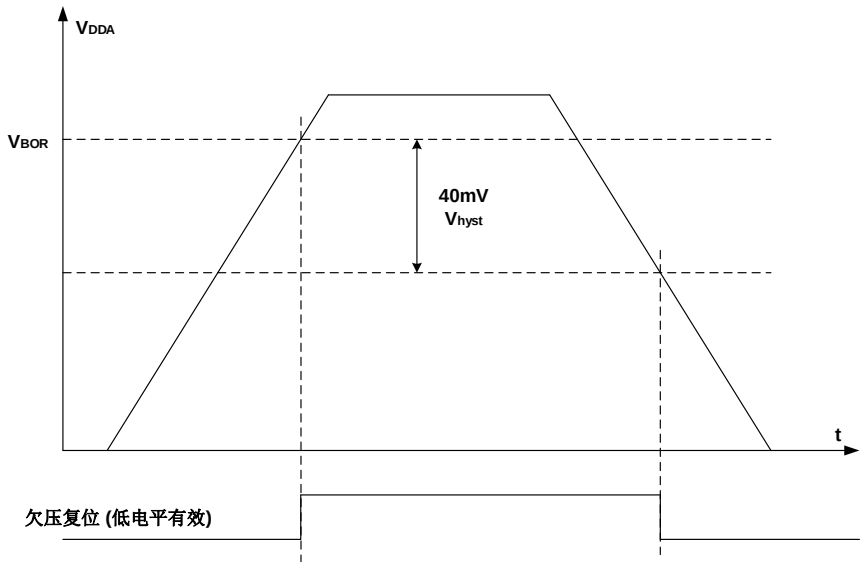
POR / PDR（上电/掉电复位）电路检测 V_{DDA} 并在电压低于特定阈值时产生电源复位信号复位整个芯片。*图 3-2. 上电/掉电复位波形图*显示了供电电压和电源复位信号之间的关系。V_{POR} 表示上电复位的阈值电压，V_{PDR} 表示掉电复位的阈值电压。迟滞电压 V_{hyst} 值约为 40mV。

图3-2. 上电/掉电复位波形图



BOR（欠压复位）电路检测 V_{DDA} 并在电压低于选项字节的 BOR_TH 定义的阈值时产生电源复位信号复位除备份域之外的整个芯片。*图 3-3. 欠压复位波形图*显示了供电电压和 BOR 复位信号之间的关系。V_{BOR} 表示欠压复位的阈值电压，该值在选项字节 BOR_TH 中定义。迟滞电压 V_{hyst} 值约为 40mV。

图3-3. 欠压复位波形图



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_CTL）中的 LVDT[2:0]位进行配置。LVD 通过 LVDEN 置位使能，位于电源控制和状态寄存器（PMU_CS）中的 LVDF 位表示低电压事件是否出现，该事件连接至 EXTI 的第 16 线，用户可以通过配置 EXTI 的第 16 线产生相应的中断。*图 3-4. 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）。该图也显示了供电电压与 LVD 信号的关系。迟滞电压 V_{hyst} 值为 100mV。

OVD 的功能是检测 V_{DDA} 供电电压是否高于电压检测阈值，该阈值由电源控制寄存器（PMU_CTL）中的 OVDT 位进行配置。OVD 通过 OVDEN 置位使能，位于电源控制和状态

寄存器（PMU_CS）中的 OVDF 位表示过电压事件是否出现，该事件连接至 EXTI 的第 24 线，用户可以通过配置 EXTI 的第 24 线产生相应的中断。*图 3-5. OVD 阈值波形图*显示了 V_{DDA} 供电电压和 OVD 输出信号的关系。（OVD 中断信号依赖于 EXTI 第 24 线的上升或下降沿配置）。该图也显示了供电电压与 OVD 信号的关系。迟滞电压 V_{hyst} 值为 25mV。

图3-4. LVD阈值波形图

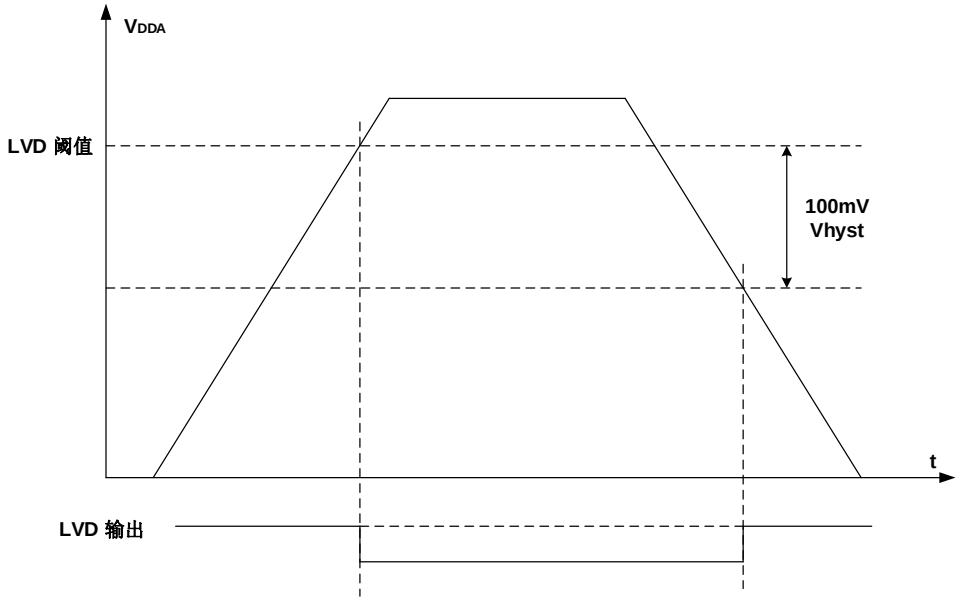
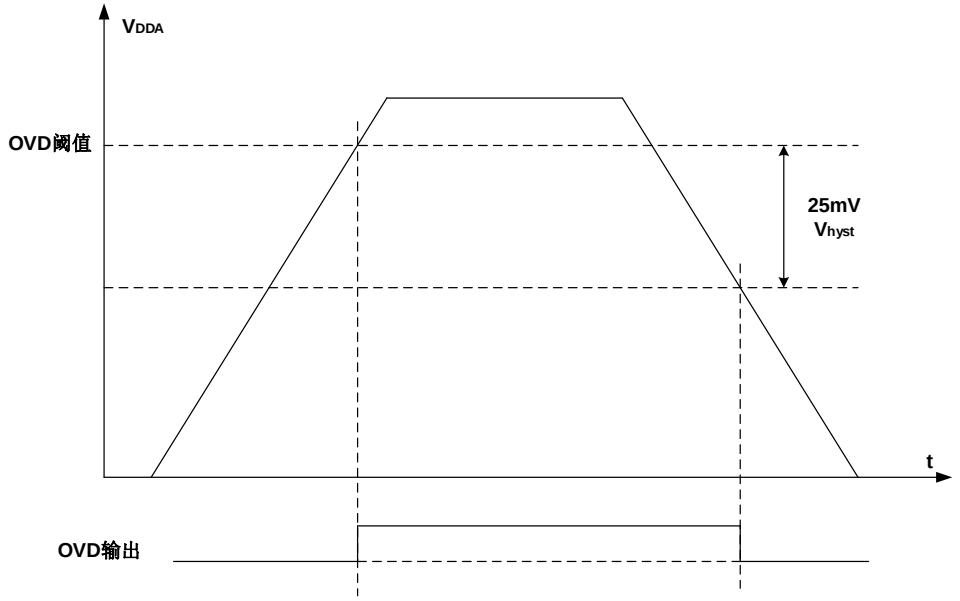


图3-5. OVD阈值波形图



一般来说，数字电路由 V_{DD} 供电，而大多数的模拟电路由 V_{DDA} 供电。为了提高 ADC 和 DAC 的转换精度，为 V_{DDA} 独立供电可使模拟电路达到更好的特性。为避免噪声，V_{DDA} 通过外部滤波电路连接至 V_{DD}，相应的 V_{SSA} 通过特定电路连接至 V_{SS}。否则，如果 V_{DDA} 和 V_{DD} 不同时，V_{DDA} 须高于 V_{DD}，但压差不超过 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 V_{REFP} / V_{REFN}。根据不同的封装，V_{REFP} 可被连接至 V_{DDA} 引脚，或者外部参考电压，外部参考电压的范围请参

考*表 14-2. ADC 引脚定义*和*表 15-1. DAC 引脚*，V_{REFN} 须被连接至 V_{SSA} 引脚，V_{REFP} 引脚仅存在于 100-pin 与 64-pin 的封装上，而在更少引脚的封装上不存在，因其内部已经连接至 V_{DDA}。V_{REFN} 内部直接连接至 V_{SSA}。

3.3.2. 1.1V 电源域

1.1V电源域为Cortex®-M33内核逻辑、AHB / APB外设和V_{DD} / V_{DDA}域的APB接口等供电。当1.1V电压上电后，POR将在1.1V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。

3.3.3. 省电模式

系统复位或电源复位后，GD32A50x 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：减慢系统时钟（HCLK，PCLK1，PCLK2），或关闭未使用的外设的时钟。此外，三种省电模式可以实现更低的功耗，它们是睡眠模式、深度睡眠模式和待机模式。

睡眠模式

睡眠模式与Cortex®-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，有两种睡眠进入的机制可选：

- 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，SRAM0（0KB~16KB）数据保持，1.1V 域中的所有时钟全部关闭，IRC8M、HXTAL 及 PLL 也全部被禁用。寄存器中的内容被保留。根据 PMU_CTL 寄存器中 SRAMSW1 与 SRAMSW2 位的配置，可分别控制 SRAM1（16KB~32KB）、SRAM2（32KB~48KB）在深度睡眠模式下的数据保持能力。根据 PMU_CTL 寄存器的 LDOLP 位的配置，可控制 LDO 工作在正常模式或低功耗模式。进入深度睡眠模式之前，先将 Cortex®-M33 系统控制寄存器的 SLEEPDEEP 位置 1，再清除 PMU_CTL 寄存器的 STBMOD 位，然后执行 WFI 或 WFE 指令即可进入深度睡眠模式。如果睡眠模式是通过执行 WFI 指令进入的，任何来自 EXTI 的中断可以将系统从深度睡眠模式中唤醒。如果睡眠模式是通过执行 WFE 指令进入的，任何来自 EXTI 的事件可以将系统从深度睡眠模式中唤醒（如果 SEVONPEND 为 1，任何来自 EXTI 的中断都可以唤醒系统，请

参考 Cortex®-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。

深度睡眠模式下的低驱动模式可通过配置 PMU_CLT0 中的 LDEN 位使能，低驱动模式具有低驱动能力，低功耗模式具有低功耗。

正常驱动/正常功耗：深度睡眠模式下，配置 PMU_CLT0 中的 LDEN 位为 0，系统将不进入低驱动模式；配置 PMU_CLT0 中的 LDOLP 位为 0，系统将不进入低功耗模式。

正常驱动/低功耗：深度睡眠模式下，配置 PMU_CLT0 中的 LDEN 位为 0，系统将不进入低驱动模式；配置 PMU_CLT0 中的 LDOLP 位为 1，系统将进入低功耗模式。

低驱动/正常功耗：深度睡眠模式下，配置 PMU_CLT0 中的 LDEN 位为 1，系统将进入低驱动模式；配置 PMU_CLT0 中的 LDOLP 位为 0，系统将不进入低功耗模式。

低驱动/低功耗：深度睡眠模式下，配置 PMU_CLT0 中的 LDEN 位为 1，系统将进入低驱动模式；配置 PMU_CLT0 中的 LDOLP 位为 1，系统将进入低功耗模式。

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）和相关外设标志位必须被复位，参考*表 6-3. 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。在进入深度睡眠模式之前需置位 FMC_WS 寄存器中的 SLEEP_SLP 位。

待机模式

待机模式也是基于 Cortex®-M33 的 SLEEPDEEP 模式实现的。在待机模式下，整个 1.1V 域全部停止供电，同时 LDO 和包括 IRC8M、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_CTL 寄存器的 STBMOD 位置 1，再清除 PMU_CS 寄存器的 WUF 位，然后执行 WFI 或 WFE 指令，系统进入待机模式，PMU_CS 寄存器的 STBF 位状态表示 MCU 是否已进入待机模式。待机模式有四个唤醒源，包括来自 NRST 引脚的外部复位，RTC 闹钟，FWDGT 复位，WKUP 引脚的上升沿。待机模式可以达到最低的功耗，但唤醒时间最长。另外，一旦进入待机模式，SRAM 和 1.1V 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 0x0000 0000 地址开始执行指令代码。

表 3-1. 节电模式总结

模式	睡眠	深度睡眠	待机
描述	仅关闭 CPU 时钟	1、关闭 1.1V 电源域的所有时钟 2、关闭 IRC8M、HXTAL 和 PLL	1、关闭 1.1V 电源域的供电 2、关闭 IRC8M、HXTAL 和 PLL
LDO 状态	开启开启（正常功耗模式、正常驱动模式）	开启（正常功耗模式或低功耗模式、正常驱动模式或低驱动模式）	关闭
配置	SLEEPDEEP = 0	SLEEPDEEP = 1 STBMOD = 0	SLEEPDEEP = 1 STBMOD = 1，WURST = 1
进入指令	WFI 或 WFE	WFI 或 WFE	WFI 或 WFE
唤醒	若通过 WFI 进入，则任	若通过 WFI 进入，来自	1、NRST 引脚

模式	睡眠	深度睡眠	待机
	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	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	2、WKUP 引脚 3、FWDGT 复位 4、RTC 闹钟
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	上电序列

注意：在待机模式下，除了 NRST 引脚，配置为 RTC 功能的 PC13，用作 LXTAL 晶振引脚的 PC14 和 PC15，使能的 WKUP 引脚，其他所有 I / O 都处于高阻态

3.4. PMU 寄存器

PMU基地址：0x4000 7000

3.4.1. 控制寄存器（PMU_CTL）

地址偏移：0x00
复位值：0x0000 8000（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										SRAMSW2	SRAMSW1	保留	LDEN	保留	
										rw	rw		rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVDT	OVDEN	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
31:22	保留	必须保持复位值。
21	SRAMSW2	深度睡眠模式下SRAM2（32KB~48KB）电源开关 0：深度睡眠模式下SRAM2电源打开且数据保持 1：深度睡眠模式下SRAM2电源关闭且数据丢失
20	SRAMSW1	深度睡眠模式下SRAM1（16KB~32KB）电源开关 0：深度睡眠模式下SRAM1电源打开且数据保持 1：深度睡眠模式下SRAM1电源关闭且数据丢失
19	保留	必须保持复位值。
18	LDEN	深度睡眠模式下低驱动模式使能 0：深度睡眠模式下进入正常驱动模式 1：深度睡眠模式下进入低驱动模式
17:16	保留	必须保持复位值。
15	OVDT	过压检测器阈值 0：5V 1：5.5V
14	OVDEN	过压检测器使能 0：关闭过电压检测器 1：开启过电压检测器
13:9	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 000：2.9V 001：3.1V 010：3.3V 011：3.5V 100：4.0V 101：4.2V 110：4.4V 111：4.6V
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 **注意：**当SYSCFG_CFG2寄存器里的LVD_LOCK位被置1时，LVDEN和LVDT[2:0]仅可读。
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 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时，LDO仍正常工作 1：当系统进入深度睡眠模式时，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。

3.4.2. 电源控制和状态寄存器（PMU_CS）

地址偏移：0x04

复位值：0x0000 0000（从待机模式唤醒后不复位）

该寄存器可以按半字（16位）或字（32位）访问。

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						WUPEN1	WUPEN0	保留				OVDF	LVDF	STBF	WUF
						rw	rw					r	r	r	r

位/位域	名称	描述
31:10	保留	必须保持复位值。
9	WUPEN1	WKUP引脚1唤醒使能（PC13） 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在进入待机模式之前置1，WKUP引脚1的上升沿会将系统从待机模式唤醒。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。当置位该控制位后，当输入为高的时候，将会触发一个唤醒事件。
8	WUPEN0	WKUP引脚0唤醒使能（PA0） 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在进入待机模式之前置1，WKUP引脚0的上升沿会将系统从待机模式唤醒。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。当置位该控制位后，当输入为高的时候，将会触发一个唤醒事件。
7:4	保留	必须保持复位值。
3	OVDF	过电压检测器状态标志 0：过电压事件没出现（V_{DDA}低于设定的OVD阈值） 1：过电压事件出现（V_{DDA}等于或高于OVD阈值） **注意：**OVD功能在待机模式被禁用。
2	LVDF	低电压检测器状态标志 0：低电压事件没出现（V_{DDA}高于设定的LVD阈值） 1：低电压事件出现（V_{DDA}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或通过设置PMU_CTL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 该位由系统复位或通过设置PMU_CTL寄存器的WURST位来清零。

4. 备份寄存器（BKP）

4.1. 简介

位于备份域中的备份寄存器由 V_{DD} 电源供电，备份寄存器有 10 个 16 位（20 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BKPEN位来打开电源和备份接口时钟，然后再通过设置PMU_CTL寄存器的BKPWEN位来使能对备份域中寄存器的写访问。

4.2. 主要特征

- 20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TAMPER引脚（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

4.3. 功能说明

4.3.1. RTC 时钟校准

为提高RTC时钟精度，MCU提供时钟输出校准功能。RTC时钟或者RTC时钟经64分频后作为输出至PC13。通过置位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的比例减慢或加快RTC时钟。

4.3.2. 侵入检测

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

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

4.4. BKP 寄存器

BKP 基地址：0x4000 6C00

4.4.1. 备份数据寄存器（BKP_DATAx）（x= 0..9）

地址偏移：0x04到0x28
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
DATA [15:0]															
rw															

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后，BKP_DATAx 寄存器的内容仍旧不会丢失。

4.4.2. RTC 信号输出控制寄存器（BKP_OCTL）

地址偏移：0x2C
复位值：0x0000 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
CALDIR	CCOSEL	保留				ROSEL	ASOEN	COEN	RCCV[6:0]						
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	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。

4.4.3. 侵入引脚控制寄存器（BKP_TPCTL）

地址偏移：0x30

复位值：0x0000 8000

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
PCSEL	保留													TPAL	TPEN
rw														rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	PCSEL	OSC32_IN 引脚选择 0：PC13 为 OSC32_IN 引脚

		1：PC14 为 OSC32_IN 引脚
14:2	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。

4.4.4. 侵入控制状态寄存器（BKP_TPCS）

地址偏移：0x34

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						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。

5. 复位和时钟单元（RCU）

5.1. 复位控制单元（RCTL）

5.1.1. 简介

GD32A50x复位控制包括三种复位控制：电源复位、系统复位和备份域复位。电源复位又称为冷复位，电源启动时复位除了备份域的所有系统。除了SW-DP控制器和备份域，系统复位将复位处理器内核和外设IP部分。备份域复位复位备份区域。复位被外部信号、内部事件和复位发生器触发。接下章节将详细介绍这些复位。

5.1.2. 功能描述

电源复位

当以下事件中之一发生时，产生电源复位：1、上电/掉电复位（POR/PDR复位）；2、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电压给GD32A50x产品时，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：

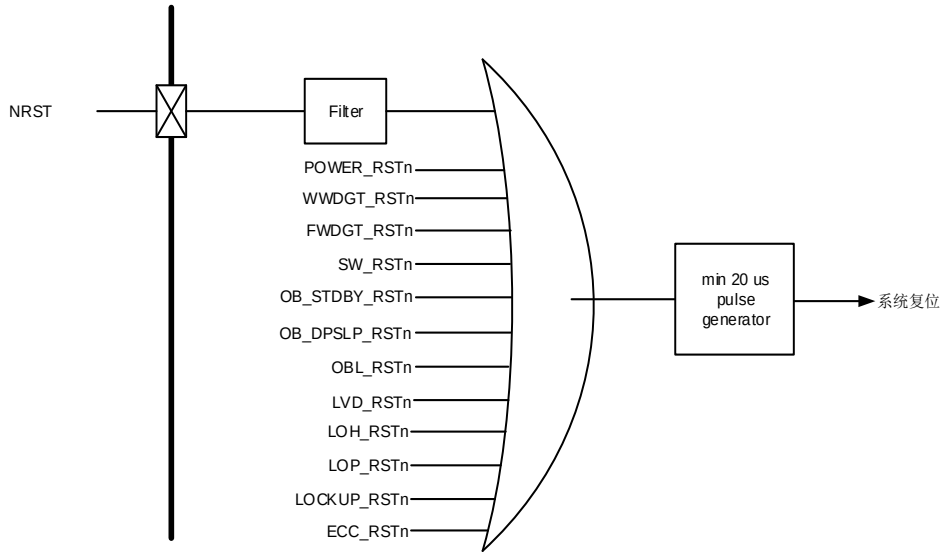
1. 电源复位（POWER_RSTn）；
2. 外部引脚复位（NRST）；
3. 窗口看门狗定时器计数终止（WWDGT_RSTn）；
4. 独立看门狗定时器计数终止（FWDGT_RSTn）；
5. Cortex®-M33的中断应用和复位控制寄存器中的SYSRESETREQ位置‘1’（SW_RSTn）；
6. 选项字节装载器复位（OBL_RSTn）；
7. 选项字节寄存器nRST_STDBY设置为0，并且进入待机模式时将产生复位（OB_STDBY_RSTn）；
8. 选项字节寄存器nRST_DPSLP设置为0，并且进入深度睡眠模式时将产生复位（OB_DPSLP_RSTn）；
9. 低电压检测复位（LVD_RSTn）；
10. HXTAL丢失复位（LOH_RSTn）；
11. PLL丢失复位（LOP_RSTn）；
12. CPU锁死复位（LOCKUP_RSTn）；
13. FLASH或SRAM 2-bit ECC错误复位（ECC_RSTn）。

LVD_RSTn / LOH_RSTn / LOP_RSTn / LOCKUP_RSTn / ECC_RSTn应通过软件启用。

除了SW-DP控制器和备份域，系统复位将复位处理器内核和外设IP部分。

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



注意：LVD_RSTn / LOH_RSTn / LOP_RSTn / LOCKUP_RSTn / ECC_RSTn产生复位是有条件的。复位源/时钟寄存器（RCU_RSTSCK）中的LVDRSTEN / LOHRSTEN / LOPRSTEN / LOCKUPRSTEN / ECCRSTEN位必须置位。

备份域复位

当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(V_{DD}重新上电)。

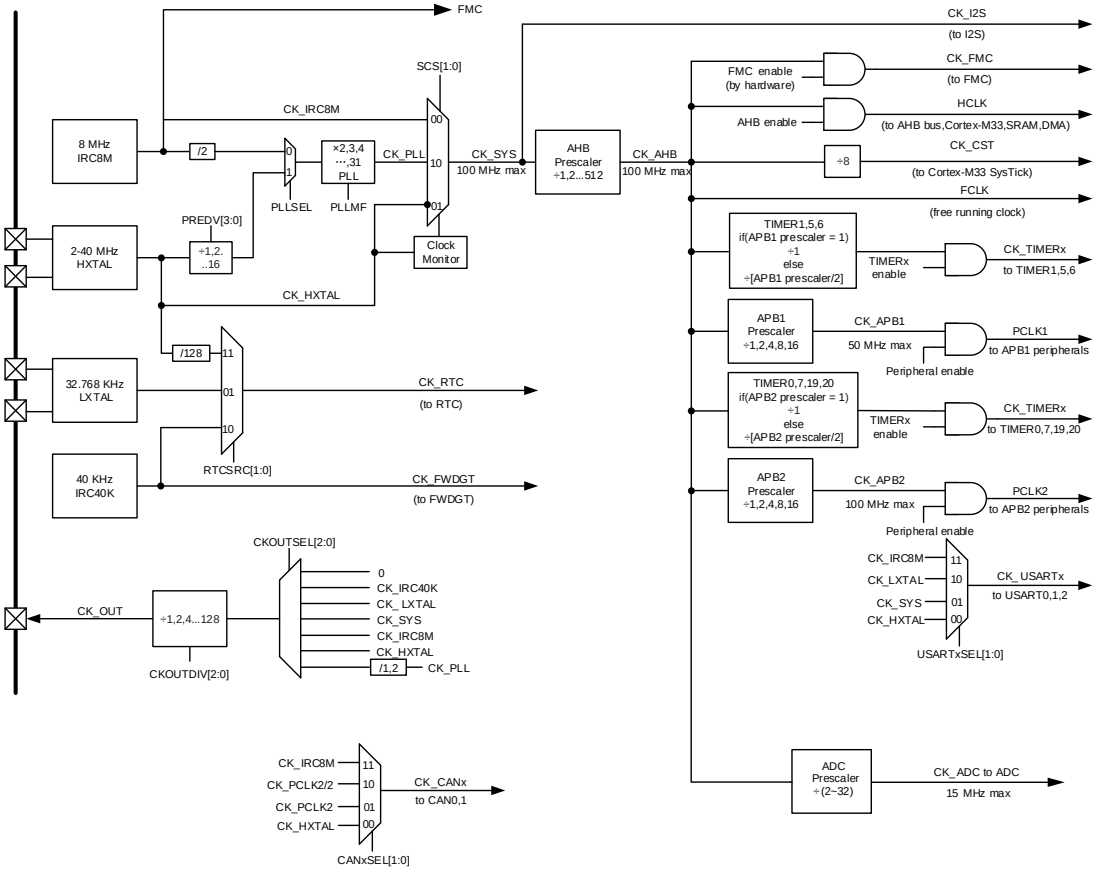
5.2. 时钟控制单元（CCTL）

5.2.1. 简介

时钟控制单元提供了一系列频率和时钟功能，包括一个内部8M RC振荡器时钟(IRC8M）、一个一个外部高速晶体振荡器时钟（HXTAL）、一个内部低速RC振荡器时钟（IRC40K）、一个外部低速晶体振荡器时钟（LXTAL）、一个锁相环（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选通电路。

AHB、APB和Cortex®-M33时钟都源自系统时钟（CK_SYS），系统时钟的时钟源为IRC8M、HXTAL或PLL。系统时钟的最大运行时钟频率可以达到100MHz。

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 和 APB1 域的最高时钟频率分别为 100MHz、100MHz 和 50MHz。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 控制与状态寄存器的设置，可选择上述时钟或 Cortex（HCLK）时钟作为 SysTick 时钟。

ADC时钟由AHB时钟经2~32分频获得，它们是通过设置配置寄存器2（RCU_CFG2）的ADCPSC位来选择的。USART0/1/2的时钟可以选择IRC8M时钟、LXTAL时钟、系统时钟或HXTAL时钟，通过设置配置寄存器2（RCU_CFG2）的USART0/1/2SEL位来选择。CAN0/1的时钟可以选择IRC8M时钟、PCLK2/2时钟、PCLK2时钟或HXTAL时钟，通过设置配置寄存器2（RCU_CFG2）的CAN0/1SEL位来选择。

RTC时钟可以选择LXTAL时钟、IRC40K时钟或HXTAL时钟128分频，通过设置备用域控制寄存器（RCU_BDCTL）的RTCSRC位域来选择。

FWDGT时钟可以选择IRC40K时钟，当FWDGT启动时强制选择。

如果APB时钟分频系数为1，定时器的时钟频率与所在AHB总线频率一致。否则，定时器的时钟频率被设为与其相连的APB总线频率的2倍。

5.2.2. 主要特性

- 2到40 MHz外部高速晶体振荡器（HXTAL）
- 8 MHz内部高速RC振荡器（IRC8M）

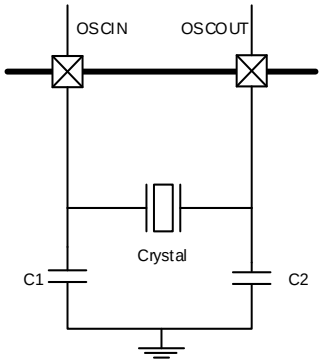
- 32,768 Hz外部低速晶体振荡器（LXTAL）
- 40 kHz内部低速RC振荡器（IRC40K）
- PLL时钟源可以是HXTAL或IRC8M
- HXTAL时钟监视

5.2.3. 功能描述

高速外部晶体振荡器时钟（HXTAL）

2到40MHz的外部振荡器可为系统提供更为精确的主时钟。带有特定频率的晶体必须靠近两个HXTAL的引脚。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。

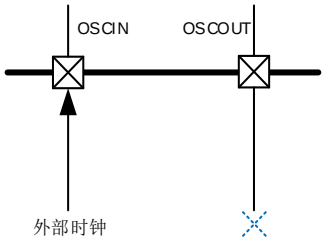
图5-3. HXTAL时钟源



HXTAL晶体可以通过设置时钟控制寄存器RCU_CTL的HXTALEN位来启动或关闭，在时钟控制寄存器RCU_CTL中的HXTALSTB位用来指示高速外部振荡器是否已稳定。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又称启动时间。当HXTAL时钟稳定后，如果在时钟中断寄存器RCU_INT中的相应中断使能位HXTALSTBIE位被置‘1’，将会产生相应中断。在这一点上，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HXTAKEN位置‘1’来使能外部旁路模式。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



通过使用控制寄存器（RCU_CTL）中的HXTALSCAL位选择HXTAL频率范围。如果HXTAL频率高于8MHz，则必须将HXTALSCAL位置1。

高速内部8MHz RC振荡器时钟（IRC8M）

高速内部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

认选择的时钟源就是IRC8M时钟。IRC8M RC振荡器能够在不需要任何外部器件的条件下提供更低成本类型的时钟源。IRC8M晶体可以通过设置时钟控制寄存器（RCU_CTL）中的IRC8MEN位被启动和关闭。时钟控制寄存器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。如果时钟中断寄存器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。用户需求、环境条件和成本将决定选择哪个时钟作为系统时钟源。

如果HXTAL或者PLL是系统时钟源，为了最大程度减小系统从深度睡眠模式启动的时间，系统从深度睡眠模式初始唤醒的时候硬件强制IRC8M时钟作为系统时钟。

锁相环（PLL）

内部锁相环PLL通过对输入参考频率为2~40MHz的时钟基准2 ~31倍频，可以提供16~100 MHz的时钟输出。

PLL可以通过设置时钟控制寄存器（RCU_CTL）中的PLLEN位被启动和关闭。时钟控制寄存器RCU_CTL中的PLLSTB位用来指示PLL时钟是否稳定。如果时钟中断寄存器RCU_INT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。

低速外部晶体振荡器时钟（LXTAL）

LXTAL晶体是一个32.768KHz的低速外部晶体或陶瓷谐振器。它为实时时钟电路提供一个低功耗且精确的时钟源。LXTAL时钟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。如果时钟中断寄存器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。

注意：在这个系列中，只能使用旁路模式（LXTALBPS位为1），由外部提供32.768KHz时钟。

低速内部RC振荡器时钟（IRC40K）

IRC40K RC振荡器时钟担当一个低功耗时钟源的角色，它的时钟频率大约40 kHz，为独立看门狗定时器和实时时钟电路提供时钟。IRC40K提供低成本的时钟源，因为不需要外部器件。IRC40K RC振荡器可以通过设置控制/状态寄存器RCU_RSTSCK中的IRC40KEN位被启动和关闭。控制/状态寄存器RCU_RSTSCK中的IRC40KSTB位用来指示IRC40K时钟是否已稳定。如果时钟中断寄存器RCU_INT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。

系统时钟（CK_SYS）选择

系统复位后，IRC8M时钟被选为系统时钟，改变时钟配置寄存器RCU_CFG0中的系统时钟变换位SCS可以切换系统时钟源为HXTAL或PLL。当SCS的值改变，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。

PLL时钟监控器（PLLM）

PLL时钟监控器功能通过控制寄存器(RCU_CTL)中的PLL时钟监控器使能位PLLMEN来使能。该功能应在PLL启动延迟后启用，并在PLL停止或频率降低时禁用。一旦检测到PLL故障，就会产生PLL丢失复位或中断。复位功能由寄存器RCU_RSTSCK中的LOPRSTEN位决定。中断功能由寄存器RCU_INT中的PLLMIE位决定。

PLL时钟由IRC8M监控。

如果LOPRSTEN位为0，将置位中断寄存器RCU_INT中的PLL丢失标志PLLMIF，并生成PLL故障中断。PLL将被自动禁用。如果选择PLL作为CK_SYS的时钟源，则PLL故障将迫使CK_SYS源为IRC8M，并且PLL将被自动禁用。

如果LOPRSTEN为1，则会产生PLL丢失复位。

HXTAL时钟监视器（CKM）

设置时钟控制寄存器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。一旦检测到HXTAL故障，HXTAL丢失复位由寄存器RCU_RSTSCK中的LOHRSTEN位决定是否产生。NMI中断是由SYSCFG_CFG3寄存器的CKMNMIIE位决定。

如果LOHRSTEN为0，则将时钟中断寄存器RCU_INT中的HXTAL时钟阻塞标志位CKMIF将被置‘1’，产生HXTAL故障事件。HXTAL将被自动禁用。此故障中断连接到Cortex®-M33的不可屏蔽中断NMI。如果选择HXTAL作为CK_SYS或PLL的时钟源，则HXTAL故障将迫使CK_SYS源为IRC8M，并且PLL将被自动禁用。

如果LOHRSTEN为1，则会生成HXTAL丢失复位。

LXTAL时钟监控器（LCKM）

可以通过软件在控制寄存器（RCU_CTL）中写入LCKMEN来激活LXTAL上的时钟监视器。在LXTAL和IRC40K使能并准备就绪之前，无法使能LCKMEN。

当LCKMEN使能时，4位加1计数器将在IRC40K域工作。如果LXTAL时钟卡在0/1错误或减慢约20KHz，计数器将溢出。LXTAL时钟故障将被发现。

时钟输出功能

时钟输出功能输出从32kHz到100MHz的时钟。通过设置时钟配置寄存器RCU_CFG0中的CK_OUT时钟源选择位CKOUTSEL能够选择不同的时钟信号。相应的GPIO引脚应该被配置成复用功能I/O（AFIO）模式来输出选择的时钟信号

表 5-1. 时钟源的选择

时钟源选择位	时钟源
000	无时钟
001	保留
010	CK_IRC40K
011	CK_LXTAL
100	CK_SYS
101	CK_IRC8M

时钟源选择位	时钟源
110	CK_HXTAL
111	CK_PLL or CK_PLL/2

通过配置时钟配置寄存器RCU_CFG0的CKOUTDIV[2:0]位，可以将输出时钟按比例分频，进而降低CK_OUT频率。

深度睡眠模式时钟控制

当MCU工作在深度睡眠模式时，USART0/1/2能唤醒MCU，前提是它们的时钟是由LXTAL时钟提供且LXTAL时钟被使能。

如果USART0/1/2时钟选择IRC8M时钟并且工作在深度睡眠模式，它们有能力开启或关闭IRC8M时钟，USART0/1/2采用IRC8M时钟作为工作时钟来唤醒深度睡眠模式。

电压控制

深度睡眠模式电压寄存器（RCU_DSV）中的DSLPVS[1:0]位可以控制内核在深度睡眠模式下的电压。

表 5-2. 深度睡眠模式下内核电压选择

DSLPVS[1:0]	深度睡眠模式电压(V)
00	0.8
01	0.9
10	1.0
11	1.1

RCU_DSV寄存器被电源解锁寄存器（RCU_VKEY）保护。只有在写0x1A2B3C4D到RCU_VKEY后，RCU_DSV寄存器才能被写入。

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						PLLSTB	PLLEN	保留	HXTALSCAL	LCKMEN	PLLMEN	CKMEN	HXTALBPS	HXTALSTB	HXTALEN
						r	rw		rw	rw	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
31:26	保留	必须保持复位值。
25	PLLSTB	PLL时钟稳定标志位 硬件置‘1’来指示PLL输出时钟是否稳定待用。 0：PLL没稳定 1：PLL 稳定
24	PLLEN	PLL使能 软件置位或复位。如果PLL时钟作为系统时钟的时候该位不能被复位。进入深度睡眠或待机模式时硬件自动复位。 0：PLL被关闭 1：PLL 被打开
23	保留	必须保持复位值。
22	HXTALSCAL	HXTAL 频率范围选择 仅当 HXTALEN 为 0 时，才能写入 HXTALSCAL 位。当 HXTAL 频率为 8MHz 时，HXTALSCAL 应被设为 0。仅在使用无源晶振（HXTALBPS 为 0）时需要配置该位。 0：HXTAL 频率范围为 2 ~ 8MHz 1：HXTAL 频率范围为 8 ~ 40MHz
21	LCKMEN	LXTAL 时钟监视器使能 0：禁能外部 32.768K LXTAL 时钟监视器 1：使能外部 32.768K LXTAL 时钟监视器 如果 LXTAL 时钟或 IRC40K 时钟被禁用，则该位不能设置为 1。 LCKMEN 使硬件能够检测到 LXTAL 时钟卡在低/高状态或减慢到约 20KHz。

20	PLLMEN	PLL 时钟监控器使能 0：禁能 PLL 时钟监控器 1：使能 PLL 时钟监控器 PLLMEN 使硬件能够检测到 PLL 时钟卡在低/高状态。 如果禁用了 PLL，则该位不能设置为 1。当进入深度睡眠或待机模式或检测到 PLL 解锁时，它会由硬件复位。
19	CKMEN	HXTAL时钟监视使能 0：禁止外部2 ~ 40 MHz 晶体振荡器（HXTAL）时钟监视器 1：使能外部2 ~ 40 MHz 晶体振荡器（HXTAL）时钟监视器 当硬件监测到HXTAL时钟一直停留在低或者高的状态，内部硬件将切换系统时钟到 IRC8M RC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	外部晶体振荡器（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	外部晶体振荡器（HXTAL）时钟稳定状态标志位 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	外部高速振荡器时钟使能 软件置‘1’或清‘0’。如果HXTAL时钟或者PLL输入时钟作为系统时钟，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：禁止外部2 ~ 40 MHz晶体振荡器 1：使能外部 2 ~ 40 MHz 晶体振荡器
15:8	IRC8MCALIB[7:0]	高速内部振荡器校准值寄存器 上电时自动加载这些位
7:3	IRC8MADJ[4:0]	高速内部振荡器时钟调整值 这些位由软件置位，最终调整值为 IRC8MADJ 当前值加上 IRC8MCALIB[7:0]位的值。最终调整值应该调整 IRC8M 到 8 MHz ± 1%。
2	保留	必须保持复位值。
1	IRC8MSTB	高速内部（IRC8M）时钟稳定状态标志位 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	高速内部振荡器使能 软件复位置位。如果IRC8M时钟用作系统时钟时该位不能被复位。当从待机和深度睡眠模式返回或用作系统时钟的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8 MHz RC振荡器关闭

1：内部 8 MHz RC 振荡器开启

5.3.2. 配置寄存器 0（RCU_CFG0）

地址偏移：0x04

复位值：0x0002 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
PLLDV	CKOUTDIV[2:0]			PLLMF[4]	CKOUTSEL[2:0]			保留		PLLMF[3:0]				DPLL	PLLSEL
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		APB2PSC[2:0]			APB1PSC[2:0]			AHBPSC[3:0]				SCSS[1:0]		SCS[1:0]	
		rw			rw			rw				r		rw	

位/位域	名称	描述
31	PLLDV	CK_PLL 1或2分频来用作CK_OUT 0：CK_PLL 2分频用作CK_OUT 1：CK_PLL 用作 CK_OUT
30:28	CKOUTDIV[2:0]	CK_OUT分频器，来降低CK_OUT频率 CK_OUT的选择参考RCU_CFG0的26:24位。 000：CK_OUT不分频 001：CK_OUT 2分频 010：CK_OUT 4分频 011：CK_OUT 8分频 100：CK_OUT 16分频 101：CK_OUT 32分频 110：CK_OUT 64分频 111：CK_OUT 128 分频
27	PLLMF[4]	PLLMF位域的位4 见 RCU_CFG0 的 21:18 位
26:24	CKOUTSEL[2:0]	CK_OUT时钟源选择 软件置位或清零。 000：没有时钟被选择 001：保留 010：选择内部40K RC振荡器时钟 011：选择外部低速振荡器时钟 100：选择系统时钟 101：选择内部8M RC振荡器时钟 110：选择外部高速振荡器时钟

		111：依赖于 PLLDV 选择（CK_PLL / 2）或 CK_PLL
23:22	保留	必须保持复位值。
21:18	PLLMF[3:0]	PLL倍频因子 软件写这些位包括RCU_CFG0的27位来确定PLL的倍频因子。 00000：（PLL时钟源 x 2） 00001：（PLL时钟源 x 3） 00010：（PLL时钟源 x 4） 00011：（PLL时钟源 x 5） 00100：（PLL时钟源 x 6） 00101：（PLL时钟源 x 7） 00110：（PLL时钟源 x 8） 00111：（PLL时钟源 x 9） 01000：（PLL时钟源 x 10） 01001：（PLL时钟源 x 11） 01010：（PLL时钟源 x 12） 01011：（PLL时钟源 x 13） 01100：（PLL时钟源 x 14） 01101：（PLL时钟源 x 15） 01110：（PLL时钟源 x 16） 01111：（PLL时钟源 x 16） 10000：（PLL时钟源 x 17） 10001：（PLL时钟源 x 18） 10010：（PLL时钟源 x 19） 10011：（PLL时钟源 x 20） 10100：（PLL时钟源 x 21） 10101：（PLL时钟源 x 22） 10110：（PLL时钟源 x 23） 10111：（PLL时钟源 x 24） 11000：（PLL时钟源 x 25） 11001：（PLL时钟源 x 26） 11010：（PLL时钟源 x 27） 11011：（PLL时钟源 x 28） 11100：（PLL时钟源 x 29） 11101：（PLL时钟源 x 30） 11110：（PLL时钟源 x 31） 11111：（PLL时钟源 x 31） **注意：**PLL 输出频率不能超过 100MHz。
17	DPLL	两倍 PLL 时钟 0：两倍 PLL 时钟 1：PLL 时钟
16	PLLSEL	PLL时钟源选择 软件置1或清0来控制PLL时钟源

		0：选择IRC8M二分频为PLL时钟源 1：选择 HXTAL 为 PLL 时钟源
15:14	保留	必须保持复位值。
13:11	APB2PSC[2:0]	APB2预分频选择 软件置1和清0来控制APB2时钟分频因子。 0xx：选择AHB时钟不分频 100：选择AHB时钟2分频 101：选择AHB时钟4分频 110：选择AHB时钟8分频 111：选择 AHB 时钟 16 分频
10:8	APB1PSC[2:0]	APB1预分频选择 软件设置和清除来控制APB1时钟分频因子。 0xx：选择AHB时钟不分频 100：选择AHB时钟2分频 101：选择AHB时钟4分频 110：选择AHB时钟8分频 111：选择 AHB 时钟 16 分频
7:4	AHBPSC[3:0]	AHB预分频选择 软件设置和清除来控制AHB时钟分频因子。 0xxx：选择CK_SYS系统时钟不分频 1000：选择CK_SYS系统时钟2分频 1001：选择CK_SYS系统时钟4分频 1010：选择CK_SYS系统时钟8分频 1011：选择CK_SYS系统时钟16分频 1100：选择CK_SYS系统时钟64分频 1101：选择CK_SYS系统时钟128分频 1110：选择CK_SYS系统时钟256分频 1111：选择 CK_SYS 系统时钟 512 分频
3:2	SCSS[1:0]	系统时钟转换状态 硬件设置和清除指示系统当前时钟源 00：选择CK_IRC8M作为CK_SYS系统时钟源 01：选择CK_HXTAL作为CK_SYS系统时钟源 10：选择CK_PLL作为CK_SYS系统时钟源 11：保留
1:0	SCS[1:0]	系统时钟转换 软件设置选择系统时钟源。由于CK_SYS的改变有固有的延迟，需要软件读SCSS位来确保转换是否结束。在从深度睡眠或待机模式中返回时，或作为系统时钟或PLL时钟源的HXTAL出现故障时，强制选择IRC8M作为系统时钟或PLL时钟。 00：选择IRC8M时钟作为CK_SYS系统时钟源 01：选择HXTAL时钟作为CK_SYS系统时钟源 10：选择PLL作为CK_SYS系统时钟源

11：保留

5.3.3. 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	PLLMIC	LCKMIC	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	PLLMIE	LCKMIE	PLL STBIE	HXTAL STBIE	IRC8M STBIE	LXTAL STBIE	IRC40K STBIE	CKMIF	PLLMIF	LCKMIF	PLL STBIF	HXTAL STBIF	IRC8M STBIF	LXTAL STBIF	IRC40K STBIF
	rw	rw	rw	rw	rw	rw	rw	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:24	保留	必须保持复位值。
23	CKMIC	HXTAL时钟阻塞中断清除 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	PLLMIC	PLL 时钟监控器中断清除 软件写 1 复位 PLLMIF 标志位。 0：不复位 PLLMIF 标志 1：重置 PLLMIF 标志
21	LCKMIC	LXTAL 时钟监控器中断清除 软件写 1 复位 LCKMIF 标志位。 0：不复位 LCKMIF 标志 1：重置 LCKMIF 标志
20	PLLSTBIC	PLL稳定中断清除 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时钟稳定中断清除 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时钟稳定中断清除 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时钟稳定中断清除 软件写1复位LXTALSTBIF标志位。 0：不复位LXTALSTBIF标志位 1：复位 LXTALRDYF 标志位
16	IRC40KSTBIC	IRC40K时钟稳定中断清除 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	PLLMIE	PLL 时钟监控器中断使能 软件置 1 和清 0 来使能/禁止 PLL 时钟监控器中断。 0：禁止 PLL 时钟监视器中断 1：使能 PLL 时钟监控器中断
13	LCKMIE	LXTAL 时钟监控器中断使能 软件置 1 和清 0 来使能/禁止 LXTAL 时钟监控器中断。 0：禁止 LXTAL 时钟监视器中断 1：使能 LXTAL 时钟监控器中断
12	PLLSTBIE	PLL时钟稳定中断使能 软件置1和清0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1和清0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1和清0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 LXTAL时钟稳定中断使能/禁止控制。 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 IRC40K时钟稳定中断使能/禁止控制。 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时钟阻塞时硬件置1。 软件置 CKMIC=1 时清除该位。 软件置 CKMNMIIE=1 时清除该位。 0：时钟运行正常 1：HXTAL 时钟阻塞
6	PLLMIF	PLL 时钟监视器中断标志 当 PLL 时钟卡住时，由硬件置 1。 软件置 PLLMIC=1 时清除该位。 0：PLL 时钟运行正常 1：PLL 时钟卡住
5	LCKMIF	LXTAL 时钟监视器中断标志 当 LXTAL 时钟卡住时，由硬件置 1。 软件置 LXTALMIC=1 时清除该位。 0：LXTAL 时钟运行正常 1：LXTAL 时钟卡住
4	PLLSTBIF	PLL稳定中断清除 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
3	HXTALSTBIF	HXTAL时钟稳定中断标志位 当外部2 ~ 40 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HXTALSTBIC=1时清除该位。 0：无HXTAL时钟稳定中断发生 1：发生 HXTAL 时钟稳定中断
2	IRC8MSTBIF	IRC8M时钟稳定中断标志位 当内部8 MHz RC振荡器时钟稳定且IRC8MSTBIE位被置1时由硬件置1。 软件置IRC8MSTBIC=1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 当外部32.768KHz晶体振荡器时钟稳定且LXTALSTBIE为被置1时由硬件置1。 软件置LXTALSTBIC=1时清除该位。 0：无LXTAL时钟稳定中断发生 1：发生 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IRC40KSTBIC =1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
CAN1RST	CAN0RST	保留								TIMER20 RST	TIMER19 RST	保留			
rw	rw	rw								rw	rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	USART0 RST	TIMER7 RST.	SPI0 RST	TIMER0 RST	ADC1 RST	ADC0 RST	保留							CMPRST	CFGRST
	rw	rw	rw	rw	rw	rw								rw	rw

位/位域	名称	描述
31	CAN1RST	CAN1 复位 由软件置 1 或清 0。 0：无复位 1：复位 CAN1
30	CAN0RST	CAN0 复位 由软件置 1 或清 0。 0：无复位 1：复位 CAN0
29:22	保留	必须保持复位值。
21	TIMER20RST	TIMER20 复位 由软件置 1 或清 0。 0：无复位 1：复位 TIMER20
20	TIMER19RST	TIMER19 复位 由软件置 1 或清 0。 0：无复位 1：复位 TIMER19
19:15	保留	必须保持复位值。
14	USART0RST	USART0 复位 由软件置 1 或清 0。 0：无复位 1：复位 USART0
13	TIMER7RST	TIMER7 复位 由软件置 1 或清 0。 0：无复位

		1：复位 TIMER7
12	SPI0RST	SPI0 复位 由软件置 1 或清 0。 0：无复位 1：复位 SPI0
11	TIMER0RST	TIMER0 复位 由软件置 1 或清 0。 0：无复位 1：复位 TIMER0
10	ADC1RST	ADC1 复位 由软件置 1 或清 0。 0：无复位 1：复位 ADC1
9	ADC0RST	ADC0 复位 由软件置 1 或清 0。 0：无复位 1：复位 ADC0
8:2	保留	必须保持复位值。
1	CMPRST	比较器复位 由软件置 1 或清 0。 0：无复位 1：复位比较器
0	CFGRST	系统配置复位 由软件置 1 或清 0。 0：无复位 1：复位系统配置

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.		DAC0 RST	PMU RST	保留					I2C1 RST	I2C0 RST	保留		USART2 RST	USART1 RST	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	SPI1 RST	保留		WWDGT RST	保留					TIMER6 RST	TIMER5 RST	保留			TIMER1 RST
	rw			rw						rw	rw				rw

位/位域	名称	描述
31:30	保留	必须保持复位值。
29	DAC0RST	DAC0 复位 由软件置 1 或清 0。 0：无复位 1：复位 DAC0
28	PMURST	电源控制复位 由软件置1或清0。 0：无复位 1：复位电源控制单元
27:23	保留	必须保持复位值。
22	I2C1RST	I2C1 复位 由软件置 1 或清 0。 0：无复位 1：复位 I2C1
21	I2C0RST	I2C0 复位 由软件置 1 或清 0。 0：无复位 1：复位 I2C0
20:19	保留	必须保持复位值。
18	USART2RST	USART2 复位 由软件置 1 或清 0。 0：无复位 1：复位 USART2
17	USART1RST	USART1 复位 由软件置 1 或清 0。 0：无复位 1：复位 USART1
16:15	保留	必须保持复位值。
14	SPI1RST	SPI1 复位 由软件置 1 或清 0。 0：无复位 1：复位 SPI1
13:12	保留	必须保持复位值。
11	WWDGTRST	窗口看门狗定时器复位 由软件置1或清0。 0：无复位

		1：复位窗口看门狗定时器
10:6	保留	必须保持复位值。
5	TIMER6RST	TIMER6 定时器复位 由软件置 1 或清 0。 0：无复位 1：复位 TIMER6 定时器
4	TIMER5RST	TIMER5定时器复位 由软件置1或清0。 0：无复位 1：复位 TIMER5 定时器
3:1	保留	必须保持复位值。
0	TIMER1RST	TIMER1定时器复位 由软件置1或清0。 0：无复位 1：复位 TIMER1 定时器

5.3.6. AHB 使能寄存器（RCU_AHBEN）

地址偏移：0x14

复位值：0x0000 0014

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留									PFEN	PCEN	PBEN	PCEN	PBEN	PAEN	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	MFCOMEN	保留							CRCEN	保留	FMCSPEN	DMAMUXEN	SRAMSPEN	DMA1EN	DMA0EN
	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	PFEN	GPIOF时钟使能 由软件置1或清0。 0：GPIOF时钟关闭 1：GPIOF 时钟开启
21	PEEN	GPIOE时钟使能 由软件置1或清0。 0：GPIOE时钟关闭 1：GPIOE 时钟开启

20	PDEN	GPIOD时钟使能 由软件置1或清0。 0：GPIOD时钟关闭 1：GPIOD 时钟开启
19	PCEN	GPIOC时钟使能 由软件置1或清0。 0：GPIOC时钟关闭 1：GPIOC 时钟开启
18	PBEN	GPIOB时钟使能 由软件置1或清0。 0：GPIOB时钟关闭 1：GPIOB 时钟开启
17	PAEN	GPIOA时钟使能 由软件置1或清0。 0：GPIOA时钟关闭 1：GPIOA 时钟开启
16:15	保留	必须保持复位值。
14	MFCOMEN	MFCOM 时钟使能 由软件置 1 或清 0。 0：MFCOM 时钟关闭 1：MFCOM 时钟开启
13:7	保留	必须保持复位值。
6	CRCEN	CRC 时钟使能 由软件置1或清0。 0：CRC时钟关闭 1：CRC 时钟开启
5	保留	必须保持复位值。
4	FMCSPEN	FMC时钟使能 由软件置1或清0来开启/关闭在睡眠模式下的FMC时钟。 0：关闭睡眠模式下的FMC时钟 1：开启睡眠模式下的 FMC 时钟
3	DMAMUXEN	DMAMUX 时钟使能 由软件置 1 或清 0。 0：DMAMUX时钟关闭 1：DMAMUX 时钟开启
2	SRAMSPEN	SRAM接口时钟使能 由软件置1或清0来开启/关闭在睡眠模式下的SRAM时钟。 0：关闭睡眠模式下的SRAM接口时钟

		1：开启睡眠模式下的 SRAM 接口时钟
1	DMA1EN	DMA1时钟使能 由软件置1或清0。 0：关闭DMA1时钟 1：开启 DMA1 时钟
0	DMA0EN	DMA0时钟使能 由软件置1或清0。 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
CAN1EN	CAN0EN	TRIGSELEN	保留							TIMER20EN	TIMER19EN	保留			
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	保留							CMPEN	CFGEN
	rw	rw	rw	rw	rw	rw								rw	rw

位/位域	名称	描述
31	CAN1EN	CAN1 时钟使能 由软件置 1 或清 0。 0：关闭CAN1时钟 1：开启 CAN1 时钟
30	CAN0EN	CAN0 时钟使能 由软件置 1 或清 0。 0：关闭CAN0时钟 1：开启 CAN0 时钟
29	TRIGSELEN	TRIGSEL 时钟使能 由软件置 1 或清 0。 0：关闭TRIGSEL时钟 1：开启 TRIGSEL 时钟
28:22	保留	必须保持复位值。
21	TIMER20EN	TIMER20定时器时钟使能 由软件置1或清0。

		0：关闭TIMER20定时器时钟 1：开启 TIMER20 定时器时钟
20	TIMER19EN	TIMER19定时器时钟使能 由软件置1或清0。 0：关闭TIMER19定时器时钟 1：开启 TIMER19 定时器时钟
19:15	保留	必须保持复位值。
14	USART0EN	USART0时钟使能 由软件置1或清0。 0：关闭USART0时钟 1：开启 USART0 时钟
13	TIMER7EN	TIMER7定时器时钟使能 由软件置1或清0。 0：关闭TIMER7定时器时钟 1：开启 TIMER7 定时器时钟
12	SPI0EN	SPI0时钟使能 由软件置1或清0。 0：关闭SPI0时钟 1：开启 SPI0 时钟
11	TIMER0EN	TIMER0定时器时钟使能 由软件置1或清0。 0：关闭TIMER0定时器时钟 1：开启 TIMER0 定时器时钟
10	ADC1EN	ADC1接口时钟使能 由软件置1或清0。 0：关闭ADC1接口时钟 1：开启 ADC1 接口时钟
9	ADC0EN	ADC0接口时钟使能 由软件置1或清0。 0：关闭ADC0接口时钟 1：开启 ADC0 接口时钟
8:2	保留	必须保持复位值。
1	CMPEN	比较器时钟使能 由软件置 1 或清 0。 0：关闭比较器时钟 1：开启比较器时钟
0	CFGEN	系统配置时钟使能 由软件置 1 或清 0。 0：关闭系统配置时钟

1：开启系统配置时钟

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		DAC0EN	PMUEN	BKPEN	保留				I2C1EN	I2C0EN	保留		USART2EN	USART1EN	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	SPI1EN	保留		WWDGTEN	保留					TIMER6EN	TIMER5EN	保留			TIMER1EN
	rw			rw						rw	rw				rw

位/位域	名称	描述
31:30	保留	必须保持复位值。
29	DAC0EN	DAC0 时钟使能 由软件置 1 或清 0。 0：关闭 DAC0 时钟 1：开启 DAC0 时钟
28	PMUEN	电源接口时钟使能 由软件置1或清0。 0：关闭电源接口时钟 1：开启电源接口时钟
27	BKPEN	BKP 接口时钟使能 由软件置 1 或清 0。 0：关闭 BKP 接口时钟 1：开启 BKP 接口时钟
26:23	保留	必须保持复位值。
22	I2C1EN	I2C1时钟使能 由软件置1或清0。 0：关闭I2C1时钟 1：开启 I2C1 时钟
21	I2C0EN	I2C0时钟使能 由软件置1或清0。 0：关闭I2C0时钟 1：开启 I2C0 时钟

20:19	保留	必须保持复位值。
18	USART2EN	USART2时钟使能 由软件置1或清0。 0:关闭USART2时钟 1:开启 USART2 时钟
17	USART1EN	USART1时钟使能 由软件置1或清0。 0:关闭USART1时钟 1:开启 USART1 时钟
16:15	保留	必须保持复位值。
14	SPI1EN	SPI1时钟使能 由软件置1或清0。 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	窗口看门狗定时器时钟使能 由软件置1或清0。 0：关闭窗口看门狗定时器时钟 1：开启窗口看门狗定时器时钟
10:6	保留	必须保持复位值。
5	TIMER6EN	TIMER6定时器时钟使能 由软件置1或清0。 0：关闭TIMER6定时器时钟 1：开启 TIMER6 定时器时钟
4	TIMER5EN	TIMER5定时器时钟使能 由软件置1或清0。 0：关闭TIMER5定时器时钟 1：开启 TIMER5 定时器时钟
3:1	保留	必须保持复位值。
0	TIMER1EN	TIMER1定时器时钟使能 由软件置1或清0。 0：关闭TIMER1定时器时钟 1：开启 TIMER1 定时器时钟

5.3.9. 备份域控制寄存器（RCU_BDCTL）

地址偏移：0x20

复位值：0x0000 0018，由备份域复位电路复位

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（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1或清0。 0：无复位 1：复位备份域
15	RTCEN	RTC时钟使能 由软件置1或清0。 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时钟入口选择 软件置位或清除来控制RTC时钟源。 00：没有时钟 01：选择LXTAL时钟作为RTC时钟源 10：选择IRC40K时钟作为RTC时钟源 11：选择 HXTAL 时钟 128 分频作为 RTC 时钟源
7:5	保留	必须保持复位值。
4:3	LXTALDRI[1:0]	LXTAL驱动能力 软件置位或清除。当复位备份域时，会重装载缺省值。 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 在旁路模式下无效
2	LXTALBPS	LXTAL旁路模式使能 软件置1和清0。 0：禁止LXTAL旁路模式

		1：使能LXTAL旁路模式 **注意：**在这个系列中，LXTALBPS 位必须被配置为 1。
1	LXTALSTB	外部低速振荡器稳定状态位 硬件置1来指示LXTAL输出时钟是否稳定待用。 0：LXTAL未稳定 1：LXTAL 已稳定
0	LXTALEN	LXTAL使能 软件置1和清0。 0：关闭LXTAL 1：开启 LXTAL

5.3.10. 复位源/时钟寄存器（RCU_RSTSCK）

地址偏移：0x24

复位值：0x0C80 0000，除复位标志外由系统复位清除，复位标志只能由电源复位清除

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	OBLRSTF	RSTFC	V11RSTF	LOPRSTF	LOHRSTF	ECCRSTF	LVDRSTF	LOCKUPRSTF	BORRSTF	保留
r	r	r	r	r	r	r	rw	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	LOPRSTEN	LOHRSTEN	ECCRSTEN	LVDRSTEN	LOCKUPRSTEN	保留								IRC40KSTB	IRC40KEN
	rw	rw	rw	rw	rw									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1。 由软件通过写1到RSTFC位来清除该位。 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。 由软件通过写1到RSTFC位来清除该位。 0：无窗口看门狗定时器复位发生 1：发生窗口看门狗定时器复位
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。 由软件通过写1到RSTFC位来清除该位。 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。 由软件通过写1到RSTFC位来清除该位。 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。 由软件通过写1到RSTFC位来清除该位。 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 当有外部引脚复位发生时由硬件置1。 由软件通过写1到RSTFC位来清除该位。 0：无外部引脚复位发生 1：发生外部引脚复位
25	OBLRSTF	选项字节装载器复位标志位 选项字节装载器装载字节时由硬件置1。 由软件通过写1到RSTFC位来清除该位。 0：无选项字节装载器复位发生 1：发生选项字节装载器复位
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：清除复位标志位
23	V11RSTF	1.1V域电源复位标志位 当有1.1V域电源复位发生时由硬件置1。 由软件通过写1到RSTFC位来清除该位。 0：无1.1V域电源复位发生 1：发生 1.1V 域电源复位
22	LOPRSTF	PLL 丢失错误复位标志 当产生 PLL 丢失错误复位时，由硬件置位。 由软件通过写1到RSTFC位来清除该位。 0：没有 PLL 丢失错误复位 1：PLL 丢失错误复位
21	LOHRSTF	HXTAL 丢失错误复位标志 当产生 HXTAL 丢失错误复位时，由硬件置位。 由软件通过写1到RSTFC位来清除该位。 0：没有 HXTAL 丢失错误复位 1：HXTAL 丢失错误复位
20	ECCRSTF	ECC 2 比特错误复位标志

		当产生 ECC 2 比特错误复位时，由硬件置位。 由软件通过写 1 到 RSTFC 位来清除该位。 0：未生成 ECC 2 比特错误复位 1：产生 ECC 2 比特错误复位
19	LVDRSTF	低电压检测错误复位标志 发生低电压检测错误复位时，由硬件置位。 **注意**：低电压复位功能，需要在 PMU 里使能 LVDEN，并且使能 LVDRSTEN 才能生效。 由软件通过写 1 到 RSTFC 位来清除该位。 0：未产生低电压检测错误复位 1：产生低电压检测错误复位
18	LOCKUPRSTF	CPU 锁死错误复位标志 发生 CPU 锁死错误复位时，由硬件置位。 由软件通过写 1 到 RSTFC 位来清除该位。 0：未产生 CPU 锁死错误复位 1：产生 CPU 锁死错误复位
17	BORRSTF	BOR 复位标志 发生 BOR 复位时，由硬件置位。 由软件通过写 1 到 RSTFC 位来清除该位。 0：未产生 BOR 复位 1：产生 BOR 复位
16:15	保留	必须保持复位值。
14	LOPRSTEN	PLL 丢失复位使能 0：如果 PLLMIE 为 1，则当检测到错误时，PLL 检测器会生成中断 1：当检测到错误时，PLL 检测器产生复位
13	LOHRSTEN	HXTAL 丢失复位使能 0：无复位产生 1：当检测到错误时，HXTAL 检测器产生复位
12	ECCRSTEN	ECC 2 比特错误复位使能 0：无复位产生 1：当检测到 ECC 2 比特错误时 ECC 产生复位
11	LVDRSTEN	低电压检测复位使能 0：无复位产生 1：当 V_{DDA} 低于预设值时，低电压检测会产生复位
10	LOCKUPRSTEN	CPU 锁死复位使能 0：无复位产生 1：CPU 锁死产生复位
9:2	保留	必须保持复位值。
1	IRC40KSTB	IRC40K时钟稳定状态位

		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 时钟已稳定
0	IRC40KEN	IRC40K时钟使能 软件置1和清0。 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
保留									PFRST	PERST	PDRST	PCRST	PBRST	PARST	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	MFCOMRST	保留							CRCRST	保留		DMAMUXRST	保留	DMA1RST	DMA0RST
	rw								rw			rw		rw	rw

位/位域	名称	描述
31:23	保留	必须保持复位值。
22	PFRST	GPIOF复位 由软件置1或清0。 0：无作用 1：复位 GPIOF 口
21	PERST	GPIOE复位 由软件置1或清0。 0：无作用 1：复位 GPIOE 口
20	PDRST	GPIOD复位 由软件置1或清0。 0：无作用 1：复位 GPIOD 口
19	PCRST	GPIOC复位 由软件置1或清0。 0：无作用 1：复位 GPIOC 口
18	PBRST	GPIOB复位

		由软件置1或清0。 0：无作用 1：复位 GPIOB 口
17	PARST	GPIOA复位 由软件置1或清0。 0：无作用 1：复位 GPIOA 口
16:15	保留	必须保持复位值。
14	MFCOMRST	MFCOM复位 由软件置1或清0。 0：无作用 1：复位 MFCOM
13:7	保留	必须保持复位值。
6	CRCRST	CRC复位 由软件置1或清0。 0：无作用 1：复位 CRC
5:4	保留	必须保持复位值。
3	DMAMUXRST	DMAMUX 复位 由软件置 1 或清 0。 0：无作用 1：复位 DMAMUX
2	保留	必须保持复位值。
1	DMA1RST	DMA1 复位 由软件置 1 或清 0。 0：无作用 1：复位 DMA1
0	DMA0RST	DMA0 复位 由软件置 1 或清 0。 0：无作用 1：复位 DMA0

5.3.12. 配置寄存器 1（RCU_CFG1）

地址偏移：0x2C

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

保留

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

位/位域	名称	描述
31:4	保留	必须保持复位值。
3:0	PREDV[3:0]	CK_HXTAL 作为 PLL 输入源分频因子 由软件置 1 或清 0。这些位仅能在 PLL 关闭时改写。 CK_HXTAL 时钟分频因子为（PREDV + 1）。 0000：PLL的输入，不分频 0001：PLL的输入2分频 0010：PLL的输入3分频 0011：PLL的输入4分频 0100：PLL的输入5分频 0101：PLL的输入6分频 0110：PLL的输入7分频 0111：PLL的输入8分频 1000：PLL的输入9分频 1001：PLL的输入10分频 1010：PLL的输入11分频 1011：PLL的输入12分频 1100：PLL的输入13分频 1101：PLL的输入14分频 1110：PLL的输入15分频 1111：PLL 的输入 16 分频

5.3.13. 配置寄存器 2（RCU_CFG2）

地址偏移：0x30
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
ADCPSC[4:0]					保留										
		rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAN1SEL[1:0]		CAN0SEL[1:0]		保留				USART2SEL[1:0]		USART1SEL[1:0]		保留		USART0SEL[1:0]	
rw		rw						rw		rw				rw	

位/位域	名称	描述
31:27	ADCPSC[4:0]	ADC 时钟预分频器选择

		由软件置1或清0。 **注意：**这些位不能设置为“11111”。 CK_ADC = CK_AHB / (ADCPSC + 2)
29:16	保留	必须保持复位值。
15:14	CAN1SEL[1:0]	CK_CAN1 时钟源选择 由软件置 1 或清 0。 00：CK_CAN1 选择 CK_HXTAL 01：CK_CAN1 选择 CK_PCLK2 10：CK_CAN1 选择 CK_PCLK2/2 11：CK_CAN1 选择 CK_IRC8M
13:12	CAN0SEL[1:0]	CK_CAN0 时钟源选择 由软件置 1 或清 0。 00：CK_CAN0 选择 CK_HXTAL 01：CK_CAN0 选择 CK_PCLK2 10：CK_CAN0 选择 CK_PCLK2/2 11：CK_CAN0 选择 CK_IRC8M
11:7	保留	必须保持复位值。
7:6	USART2SEL[1:0]	USART2时钟源选择 由软件置1或清0。 00：USART2时钟选择HXTAL时钟 01：USART2时钟选择系统时钟 10：USART2时钟选择LXTAL时钟 11：USART2 时钟选择 IRC8M 时钟
5:4	USART1SEL[1:0]	USART1时钟源选择 由软件置1或清0。 00：USART1时钟选择HXTAL时钟 01：USART1时钟选择系统时钟 10：USART1时钟选择LXTAL时钟 11：USART1 时钟选择 IRC8M 时钟
3:2	保留	必须保持复位值。
1:0	USART0SEL[1:0]	USART0时钟源选择 由软件置1或清0。 00：USART0时钟选择HXTAL时钟 01：USART0时钟选择系统时钟 10：USART0时钟选择LXTAL时钟 11：USART0 时钟选择 IRC8M 时钟

5.3.14. 电源解锁寄存器（RCU_VKEY）

地址偏移：0x100

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
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]	RCU_DSV寄存器解锁 这些位只能被软件写，读的话全是 0。只有在向 RCU_VKEY 寄存器写 0x1A2B3C4D 后，RCU_DSV 寄存器才能被写。

5.3.15. 深度睡眠模式电压寄存器（RCU_DSV）

地址偏移：0x134

复位值：0x0000 0003

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														DSLPVS[1:0]	
														rw	

位/位域	名称	描述
31:2	保留	必须保持复位值。
1:0	DSLPVS[1:0]	深度睡眠模式电压选择 这些位由软件置位和清除。 00：在深度睡眠模式下内核电压为0.8V 01：在深度睡眠模式下内核电压为0.9V 10：在深度睡眠模式下内核电压为1.0V 11：在深度睡眠模式下内核电压为 1.1V

6. 中断事件控制器（EXTI）

6.1. 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25个相互独立的边沿检测电路并且能够向处理器内核产生中断请求或唤醒事件。EXTI有三种触发类型：上升沿触发、下降沿触发和任意沿触发。EXTI中的每一个边沿检测电路都可以独立配置和屏蔽。

6.2. 主要特性

- Cortex®-M33系统异常；
- 71种可屏蔽的外设中断；
- 4位中断优先级配置位—16个中断优先等级；
- 高效的中断处理；
- 支持异常抢占和咬尾中断；
- 将系统从省电模式唤醒；
- EXTI中有25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。

6.3. 功能说明

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6-1. Cortex®-M33中的NVIC异常类型*和*表6-2. 中断向量表*列出了所有的异常类型。

表6-1. 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 6-2. 中断向量表

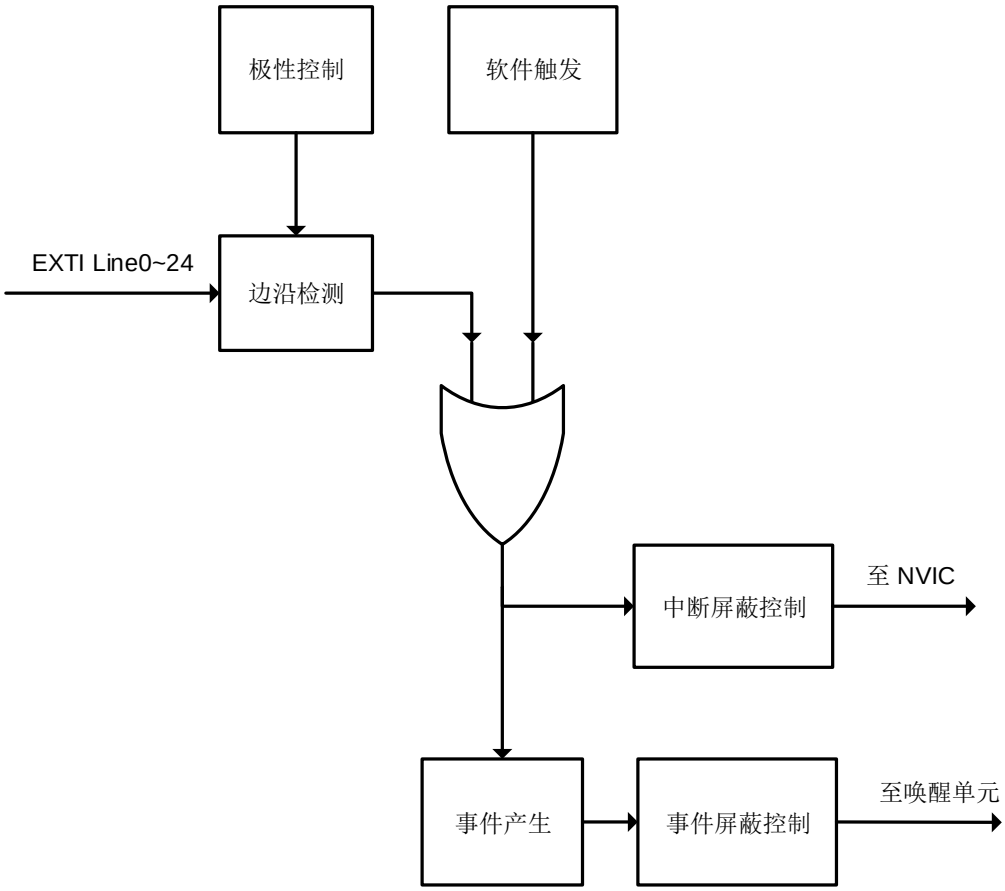
中断编号	向量编号	外设中断描述	向量地址
IRQ 0	16	窗口看门狗中断	0x0000_0040
IRQ 1	17	连接到 EXTI 线的 LVD（PVD）中断	0x0000_0044
IRQ 2	18	保留	0x0000_0048
IRQ 3	19	RTC 全局中断	0x0000_004C
IRQ 4	20	FMC 全局中断	0x0000_0050
IRQ 5	21	RCU 全局中断	0x0000_0054
IRQ 6	22	EXTI 线 0 中断	0x0000_0058
IRQ 7	23	EXTI 线 1 中断	0x0000_005C
IRQ 8	24	EXTI 线 2 中断	0x0000_0060
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 中断	0x0000_0088
IRQ 19	35	CAN0 消息缓冲区中断	0x0000_008C
IRQ 20	36	CAN0 总线关闭/总线关闭完成中断	0x0000_0090
IRQ 21	37	CAN0 错误中断	0x0000_0094
IRQ 22	38	CAN0 快速传输错误中断	0x0000_0098
IRQ 23	39	CAN0 发送警告中断	0x0000_009C
IRQ 24	40	CAN0 接收警告中断	0x0000_00A0
IRQ 25	41	连接到 EXTI 线的 CAN0 唤醒中断	0x0000_00A4

中断编号	向量编号	外设中断描述	向量地址
IRQ 26	42	TIMER0 中止，更新，触发和换相中断	0x0000_00A8
IRQ 27	43	TIMER0 捕获比较中断	0x0000_00AC
IRQ 28	44	TIMER1 全局中断	0x0000_00B0
IRQ 29	45	TIMER19 中止，更新，触发和换相中断	0x0000_00B4
IRQ 30	46	TIMER19 捕获比较中断	0x0000_00B8
IRQ 31	47	I2C0 事件中断	0x0000_00BC
IRQ 32	48	I2C0 错误中断	0x0000_00C0
IRQ 33	49	I2C1 事件中断	0x0000_00C4
IRQ 34	50	I2C1 错误中断	0x0000_00C8
IRQ 35	51	SPI0 全局中断	0x0000_00CC
IRQ 36	52	SPI1 全局中断	0x0000_00D0
IRQ 37	53	USART0 全局中断	0x0000_00D4
IRQ 38	54	USART1 全局中断	0x0000_00D8
IRQ 39	55	USART2 全局中断	0x0000_00DC
IRQ 40	56	EXTI 线[15:10]中断	0x0000_00E0
IRQ 41	57	EXTI 线[9:5]中断	0x0000_00E4
IRQ 42	58	BKP 篡改	0x0000_00E8
IRQ 43	59	TIMER20 中止，更新，触发和换相中断	0x0000_00EC
IRQ 44	60	TIMER20 捕获比较	0x0000_00F0
IRQ 45	61	TIMER7 中止，更新，触发和换相中断	0x0000_00F4
IRQ 46	62	TIMER7 捕获比较	0x0000_00F8
IRQ 47	63	DMA MUX 中断	0x0000_00FC
IRQ 48	64	SYSCFG SRAM ECC 单比特错误中断	0x0000_0100
IRQ 49	65	连接到 EXTI 线的 CMP 中断	0x0000_0104
IRQ 49	66	保留	0x0000_0108
IRQ 51	67	连接到 EXTI 线的过压检测中断	0x0000_010C
IRQ52	68	保留	0x0000_0110
IRQ53	69	保留	0x0000_0114
IRQ54	70	TIMER5 或 DAC0 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	保留	0x0000_0134
IRQ62	78	连接到 EXTI 线的 CAN1 唤醒中断	0x0000_0138
IRQ63	79	CAN1 消息缓冲区中断	0x0000_013C
IRQ64	80	CAN1 总线关闭/总线关闭完成中断	0x0000_0140
IRQ65	81	CAN1 错误中断	0x0000_0144
IRQ66	82	CAN1 快速传输错误中断	0x0000_014C
IRQ67	83	CAN1 传输警告中断	0x0000_0148

中断编号	向量编号	外设中断描述	向量地址
IRQ68	84	CAN1 接收警告中断	0x0000_0150
IRQ69	85	FPU 全局中断	0x0000_0154
IRQ70	86	MFCOM 中断	0x0000_0158

6.4. 外部中断及事件（EXTI）框图

图 6-1. EXTI 框图



6.5. 外部中断及事件功能概述

EXTI 包含多达 25 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 9 根线，具体细节参考*表 6-3. EXTI 触发源*。通过配置 SYSCFG_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*系统配置寄存器*。

除了中断，EXTI 还可以向处理器提供事件信号。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 SYSCFG 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 6-3. EXTI 触发源

EXTI 线编号	触发源
0	PA0 / PB0 / PC0 / PD0 / PE0 / PF0
1	PA1 / PB1 / PC1 / PD1 / PE1 / PF1
2	PA2 / PB2 / PC2 / PD2 / PE2 / PF2
3	PA3 / PB3 / PC3 / PD3 / PE3 / PF3
4	PA4 / PB4 / PC4 / PD4 / PE4 / PF4
5	PA5 / PB5 / PC5 / PD5 / PE5 / PF5
6	PA6 / PB6 / PC6 / PD6 / PE6 / PF6
7	PA7 / PB7 / PC7 / PD7 / PE7 / PF7
8	PA8 / PB8 / PC8 / PD8 / PE8
9	PA9 / PB9 / PC9 / PD9 / PE9
10	PA10 / PB10 / PC10 / PD10 / PE10
11	PA11 / PB11 / PC11 / PD11 / PE11
12	PA12 / PB12 / PC12 / PD12 / PE12
13	PA13 / PB13 / PC13 / PD13 / PE13
14	PA14 / PB14 / PC14 / PD14 / PE14
15	PA15 / PB15 / PC15 / PD15 / PE15
16	LVD
17	RTC 闹钟

EXTI 线编号	触发源
18	CAN0
19	CAN1
20	CMP 输出
21	USART0 唤醒
22	USART1 唤醒
23	USART2 唤醒
24	过压

6.6. EXTI 寄存器

EXTI 基地址：0x4001 0400

6.6.1. 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							INTEN24	INTEN23	INTEN22	INTEN21	INTEN20	INTEN19	INTEN18	INTEN17	INTEN16
							rw	rw	rw	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
31:25	保留	必须保持复位值。
24:0	INTENx	中断使能位 x（x = 0…24） 0：第 x 线中断被禁用 1：第 x 线中断被使能

6.6.2. 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							EVEN24	EVEN23	EVEN22	EVEN21	EVEN20	EVEN19	EVEN18	EVEN17	EVEN16
							rw	rw	rw	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
31:25	保留	必须保持复位值。
24:0	EVENx	事件使能位 x（x = 0…24） 0：第 x 线事件被禁用 1：第 x 线事件被使能

6.6.3. 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							RTEN24	RTEN23	RTEN22	RTEN21	RTEN20	RTEN19	RTEN18	RTEN17	RTEN16
							rw	rw	rw	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
31:25	保留	必须保持复位值
24:0	RTENx	上升沿触发使能x（x = 0…24） 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）

6.6.4. 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							FTEN24	FTEN23	FTEN12	FTEN21	FTEN20	FTEN19	FTEN18	FTEN17	FTEN16
							rw	rw	rw	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
31:25	保留	必须保持复位值
24:0	FTENx	下降沿触发使能 x（x = 0…24）。 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）

6.6.5. 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							SWIEV24	SWIEV23	SWIEV22	SWIEV21	SWIEV20	SWIEV19	SWIEV18	SWIEV17	SWIEV16
							rw	rw	rw	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
31:25	保留	必须保持复位值。
24:0	SWIEVx	中断/事件软件触发 x（x = 0…24） 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

6.6.6. 挂起寄存器（EXTI_PD）

地址偏移：0x14

复位值：0xXXXX XXXX X表示未定义

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							PD24	PD23	PD12	PD21	PD20	PD19	PD18	PD17	PD16
							rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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
31:25	保留	必须保持复位值
24:0	PDx	中断挂起状态 x（x = 0…24） 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。

7. 触发选择控制器（TRIGSEL）

7.1. 简介

触发选择控制器（TRIGSEL）可通过软件配置的方式，为各种外设选择触发输入信号。TRIGSEL提供了灵活的机制，可以为外设选择不同的触发输入。

使用TRIGSEL，每个外设最多可以配置4路TRIGSEL输出作为该外设的触发输入信号。每路TRIGSEL输出都可以从不同的TRIGSEL输入信号中进行选择。

7.2. 主要特征

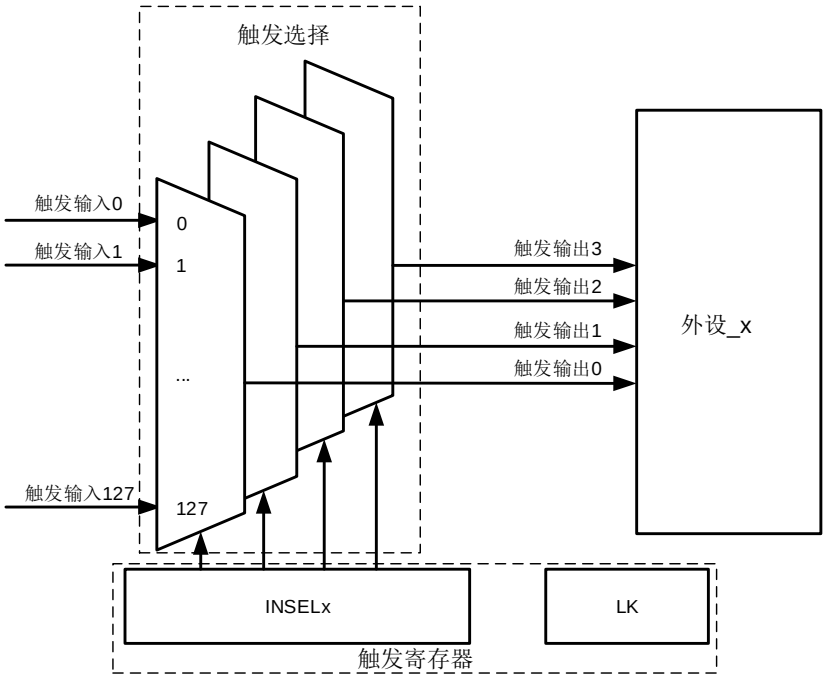
- 支持不同的可选触发输入；
- 每个外设都有专用的触发信号选择寄存器；
- 触发选择寄存器可配置多达4路输出到外设；
- 触发选择控制器的输入信号可来源于外部输入或外设输出；
- 触发选择控制器的输出信号可输出到外部输出或者到外设输入。

7.3. 功能说明

支持触发源选择的外设均具有专用TRIGSEL寄存器，用来为该外设选择不同的触发输入源。每个TRIGSEL寄存器可以配置多达4路输出，这些输出连接到外设的触发输入。每路输出均可从不同的触发输入源中选择。

*图7-1. TRIGSEL主要组成示例*显示了TRIGSEL的主要组成结构。

图 7-1. TRIGSEL 主要组成示例



7.4. 内部连接

TRIGSEL 允许软件方式为外设选择触发输入。*表 7-1. 触发输入位域选择*给出了触发输入寄存器的位域值对应的触发输入选择。

表 7-1. 触发输入位域选择

位域名称	位域值	触发输入选择
INSELx	0x00	0
	0x01	1
	0x02	TRIGSEL_IN0
	0x03	TRIGSEL_IN1
	0x04	TRIGSEL_IN2
	0x05	TRIGSEL_IN3
	0x06	TRIGSEL_IN4
	0x07	TRIGSEL_IN5
	0x08	TRIGSEL_IN6
	0x09	TRIGSEL_IN7
	0x0a	TRIGSEL_IN8
	0x0b	TRIGSEL_IN9
	0x0c	TRIGSEL_IN10
	0x0d	TRIGSEL_IN11
	0x0e	CMP_OUT
	0x0f	保留
	0x10	LXTAL_TRG

位域名称	位域值	触发输入选择
	0x11	TIMER1_CH0
	0x12	TIMER1_CH1
	0x13	TIMER1_CH2
	0x14	TIMER1_CH3
	0x15	TIMER1_TRGO
	0x16	TIMER0_CH0
	0x17	TIMER0_CH1
	0x18	TIMER0_CH2
	0x19	TIMER0_CH3
	0x1a	TIMER0_MCH0
	0x1b	TIMER0_MCH1
	0x1c	TIMER0_MCH2
	0x1d	TIMER0_MCH3
	0x1e	TIMER0_TRGO
	0x1f	TIMER7_CH0
	0x20	TIMER7_CH1
	0x21	TIMER7_CH2
	0x22	TIMER7_CH3
	0x23	TIMER7_MCH0
	0x24	TIMER7_MCH1
	0x25	TIMER7_MCH2
	0x26	TIMER7_MCH3
	0x27	TIMER7_TRGO
	0x28	TIMER19_CH0
	0x29	TIMER19_CH1
	0x2a	TIMER19_CH2
	0x2b	TIMER19_CH3
	0x2c	TIMER19_MCH0
	0x2d	TIMER19_MCH1
	0x2e	TIMER19_MCH2
	0x2f	TIMER19_MCH3
	0x30	TIMER19_TRGO
	0x31	TIMER20_CH0
	0x32	TIMER20_CH1
	0x33	TIMER20_CH2
	0x34	TIMER20_CH3
	0x35	TIMER20_MCH0
	0x36	TIMER20_MCH1
	0x37	TIMER20_MCH2
	0x38	TIMER20_MCH3
	0x39	TIMER20_TRGO

位域名称	位域值	触发输入选择
	0x3a	TIMER5_TRGO
	0x3b	TIMER6_TRGO
	0x3c	MFCOM_TRIG0
	0x3d	MFCOM_TRIG1
	0x3e	MFCOM_TRIG2
	0x3f	MFCOM_TRIG3
	0x40	RTC_Alarm
	0x41	RTC_Second
	0x42	TRIGSEL_IN12
	0x43	TRIGSEL_IN13
	0x44~0x7f	保留

如*表7-2. TRIGSEL 输入输出映射关系*所示，表明了 TRIGSEL 输入输出之间的连接关系。通过 TRIGSEL 寄存器的 INSELx[6:0]位域，可以给 TRIGSEL 的输出选择一个输入触发源。每个 TRIGSEL 寄存器配置多达 4 路输出，这些输出连接到对应的外设。

表 7-2. TRIGSEL 输入输出映射关系

触发源	触发选择	TRIGSEL 寄存器	TRIGSEL 输出	外设
1’b0	INSELx[6:0]	TRIGSEL_EXOUT0	output0	TRIGSEL_OUT0
1’b1			output1	TRIGSEL_OUT1
TRIGSEL_IN0			output2	TRIGSEL_OUT2
TRIGSEL_IN1			output3	TRIGSEL_OUT3
TRIGSEL_IN2		TRIGSEL_EXOUT1	output0	TRIGSEL_OUT4
TRIGSEL_IN3			output1	TRIGSEL_OUT5
TRIGSEL_IN4			output2	TRIGSEL_OUT6
TRIGSEL_IN5			output3	TRIGSEL_OUT7
TRIGSEL_IN6		TRIGSEL_ADC0	output0	ADC0_RTTRG
TRIGSEL_IN7				
TRIGSEL_IN8				
TRIGSEL_IN9				
TRIGSEL_IN10		TRIGSEL_ADC1	output0	ADC1_RTTRG
TRIGSEL_IN11				
CMP_OUT				
reserved				
LXTAL_TRG		TRIGSEL_DAC	output0	DAC_EXTRG
TIMER1_CH0				
TIMER1_CH1				
TIMER1_CH2				
TIMER1_CH3		TRIGSEL_TIMER0BRKIN	output0	TIMER0_BRKIN0
TIMER1_TRGO			output1	TIMER0_BRKIN1
TIMER0_CH0			output2	TIMER0_BRKIN2
TIMER0_CH1			output3	TIMER0_BRKIN3

触发源	触发选择	TRIGSEL 寄存器	TRIGSEL 输出	外设
TIMER0_CH2		TRIGSEL_TIMER7BRKIN	output0	TIMER7_BRKIN0
TIMER0_CH3			output1	TIMER7_BRKIN1
TIMER0_MCH0			output2	TIMER7_BRKIN2
TIMER0_MCH1			output3	TIMER7_BRKIN3
TIMER0_MCH2		TRIGSEL_TIMER19BRKIN	output0	TIMER19_BRKIN0
TIMER0_MCH3			output1	TIMER19_BRKIN1
TIMER0_TRGO			output2	TIMER19_BRKIN2
TIMER7_CH0			output3	TIMER19_BRKIN3
TIMER7_CH1		TRIGSEL_TIMER20BRKIN	output0	TIMER20_BRKIN0
TIMER7_CH2			output1	TIMER20_BRKIN1
TIMER7_CH3			output2	TIMER20_BRKIN2
TIMER7_MCH0			output3	TIMER20_BRKIN3
TIMER7_MCH1		TRIGSEL_MFCOM	output0	MFCOM_TRG_TIMER0
TIMER7_MCH2			output1	MFCOM_TRG_TIMER1
TIMER7_MCH3			output2	MFCOM_TRG_TIMER2
TIMER7_TRGO			output3	MFCOM_TRG_TIMER3
TIMER19_CH0		TRIGSEL_CAN0	output0	CAN0_EX_TIME_TICK
TIMER19_CH1				
TIMER19_CH2				
TIMER19_CH3				
TIMER19_MCH0		TRIGSEL_CAN1	output0	CAN1_EX_TIME_TICK
TIMER19_MCH1				
TIMER19_MCH2				
TIMER19_MCH3				
TIMER19_TRGO		TRIGSEL_TIMER0IN	output0	TIMER0_ITI0
TIMER20_CH0			output1	TIMER0_ITI1
TIMER20_CH1			output2	TIMER0_ITI2
TIMER20_CH2			output3	TIMER0_ITI3
TIMER20_CH3		TRIGSEL_TIMER7IN	output0	TIMER7_ITI0
TIMER20_MCH0			output1	TIMER7_ITI1
TIMER20_MCH1			output2	TIMER7_ITI2
TIMER20_MCH2			output3	TIMER7_ITI3
TIMER20_MCH3		TRIGSEL_TIMER19IN	output0	TIMER19_ITI0
TIMER20_TRGO			output1	TIMER19_ITI1
TIMER5_TRGO			output2	TIMER19_ITI2
TIMER6_TRGO			output3	TIMER19_ITI3
MFCOM_TRIG0		TRIGSEL_TIMER20IN	output0	TIMER20_ITI0
MFCOM_TRIG1			output1	TIMER20_ITI1
MFCOM_TRIG2			output2	TIMER20_ITI2
MFCOM_TRIG3			output3	TIMER20_ITI3
RTC_Alarm		TRIGSEL_TIMER1IN	output0	TIMER1_ITI0

触发源	触发选择	TRIGSEL 寄存器	TRIGSEL 输出	外设
RTC_Second			output1	TIMER1_ITI1
TRIGSEL_IN12			output2	TIMER1_ITI2
TRIGSEL_IN13			output3	TIMER1_ITI3

注意：除了TIMERx_ITIx和TIMERx_BRKINx外，其他所有的输出都可以选择所有输入作为触发源。TIMERx_ITIx无法选择CMP_OUT，LXTAL_TRG，其他定时器的CHx/MCHx信号和它们自己的信号作为触发源。TIMERx_BRKINx无法选择自己的信号作为触发信号。当为TIMERx_ITIx和TIMERx_BRKINx等输出选择了非法触发输入时，其输出将被强制选择为0。

触发输入选择 INSELx[6:0]位域值配置为 0 时，TRIGSEL 触发输入选择为低电平；配置为 1 时，TRIGSEL 触发输入选择为高电平。

7.5. TRIGSEL 寄存器

TRIGSEL 基地址：0x4001 8400

7.5.1. EXTOUT0 触发选择寄存器（TRIGSEL_EXTOUT0）

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
LK	INSEL3[6:0]							保留	INSEL2[6:0]						
rs	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	INSEL1[6:0]							保留	INSEL0[6:0]						
	rw								rw						

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_EXTOUT0将不能被修改。 0：TRIGSEL_EXTOUT0 寄存器可写可读 1：TRIGSEL_EXTOUT0 寄存器只读
30:24	INSEL3[6:0]	触发输出 3 的输入源选择 这些位用来选择连接到输出 3 的触发输入信号，输出 3 作为外部输出 3 的信号源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
23	保留	必须保持复位值。
22:16	INSEL2[6:0]	触发输出 2 的输入源选择 这些位用来选择连接到输出 2 的触发输入信号，输出 2 作为外部输出 2 的信号源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
15	保留	必须保持复位值。
14:8	INSEL1[6:0]	触发输出 1 的输入源选择 这些位用来选择连接到输出 1 的触发输入信号，输出 1 作为外部输出 1 的信号源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
7	保留	必须保持复位值。
6:0	INSEL0[6:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为外部输出 0 的信号源。关于具体配置请参考*表 7-1. 触发输入位域选择*。

7.5.2. EXTOUT1 触发选择寄存器（TRIGSEL_EXTOUT1）

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
LK	INSEL3[6:0]							保留	INSEL2[6:0]						
rs	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	INSEL1[6:0]							保留	INSEL0[6:0]						
	rw								rw						

位/位域	名称	描述
31	LK	TRIGSEL 锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_EXTOUT1将不能被修改。 0：TRIGSEL_EXTOUT1 寄存器可读可写 1：TRIGSEL_EXTOUT1 寄存器只读
30:24	INSEL3[6:0]	触发输出 3 的输入源选择 这些位用来选择连接到输出 3 的触发输入信号，输出 3 作为外部输出 7 的信号源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
23	保留	必须保持复位值。
22:16	INSEL2[6:0]	触发输出 2 的输入源选择 这些位用来选择连接到输出 2 的触发输入信号，输出 2 作为外部输出 6 的信号源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
15	保留	必须保持复位值。
14:8	INSEL1[6:0]	触发输出 1 的输入源选择 这些位用来选择连接到输出 1 的触发输入信号，输出 1 作为外部输出 5 的信号源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
7	保留	必须保持复位值。
6:0	INSEL0[6:0]	触发输出 0 的输入源选择 这些位用来选择连接外部输出 0 的触发输入信号，输出 0 作为外部输出 4 的信号源。关于具体配置请参考*表 7-1. 触发输入位域选择*。

7.5.3. ADC0 触发选择寄存器（TRIGSEL_ADC0）

地址偏移：0x08
复位值：0x0000 1E16

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									INSEL0[6:0]						
									rw						

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_ADC0将不能被修改。 0：TRIGSEL_ADC0 寄存器可读可写 1：TRIGSEL_ADC0 寄存器只读
30:7	保留	必须保持复位值。
6:0	INSEL0[6:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 ADC0_RTTRG（ADC0 常规组）的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。

7.5.4. ADC1 触发选择寄存器（TRIGSEL_ADC1）

地址偏移：0xC

复位值：0x0000 1E16

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									INSEL0[6:0]						
									rw						

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_ADC1将不能被修改。 0：TRIGSEL_ADC1 寄存器可读可写 1：TRIGSEL_ADC1 寄存器只读
30:7	保留	必须保持复位值。
6:0	INSEL0[6:0]	触发输出 1 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 ADC1_REGTRG（ADC1 常规组）的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。

7.5.5. DAC 触发选择寄存器（TRIGSEL_DAC）

地址偏移：0x10

复位值：0x0000 0015

该寄存器只能按位（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									INSEL0[6:0]						
												rw			

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_ DAC将不能被修改。 0：TRIGSEL_DAC 寄存器可读可写 1：TRIGSEL_DAC 寄存器制度
30:7	保留	必须保持复位值。
6:0	INSEL0[6:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 DAC_REGTRG（DAC 外部触发）的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。

7.5.6. TIMER0_ITI 触发选择寄存器（TRIGSEL_TIMER0IN）

地址偏移：0x14

复位值：0x2727 2727

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	INSEL3[6:0]							保留	INSEL2[6:0]						
rs				rw								rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	INSEL1[6:0]							保留	INSEL0[6:0]						
				rw								rw			

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER0IN将不能被修改。 0：TRIGSEL_TIMER0IN 寄存器可读可写 1：TRIGSEL_TIMER0IN 寄存器只读

30:24	INSEL3[6:0]	触发输出 3 的输入源选择 这些位用来选择连接到输出 3 的触发输入信号，输出 3 作为 TIMER0_ITI3 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
23	保留	必须保持复位值。
22:16	INSEL2[6:0]	触发输出 2 的输入源选择 这些位用来选择连接到输出 2 的触发输入信号，输出 2 作为 TIMER0_ITI2 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
15	保留	必须保持复位值。
14:8	INSEL1[6:0]	触发输出 1 的输入源选择 这些位用来选择连接到输出 1 的触发输入信号，输出 1 作为 TIMER0_ITI1 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
7	保留	必须保持复位值。
6:0	INSEL0[6:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER0_ITI0 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。

7.5.7. TIMER0_BRKIN 触发选择寄存器（TRIGSEL_TIMER0BRKIN）

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
LK	INSEL3[6:0]							保留	INSEL2[6:0]						
rs	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	INSEL1[6:0]							保留	INSEL0[6:0]						
	rw								rw						

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER0BRKIN将不能被修改。 0：TRIGSEL_TIMER0BRKIN 寄存器可读可写 1：TRIGSEL_TIMER0BRKIN 寄存器只读
30:24	INSEL3[6:0]	触发输出 3 输入源选择 这些位用来选择连接到输出 3 的触发输入信号，输出 3 作为 TIMER0_BRINK3 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
23	保留	必须保持复位值。
22:16	INSEL2[6:0]	触发输出 2 的输入源选择

		这些位用来选择连接到输出 2 的触发输入信号，输出 2 作为 TIMER0_BRINK2 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
15	保留	必须保持复位值。
14:8	INSEL1[6:0]	触发输出 1 输入源选择 这些位用来选择连接到输出 1 的触发输入信号，输出 1 作为 TIMER0_BRINK1 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
7	保留	必须保持复位值。
6:0	INSEL0[6:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER0_BRINK0 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。

7.5.8. TIMER7_ITI 触发选择寄存器（TRIGSEL_TIMER7IN）

地址偏移：0x1C
复位值：0x3030 3030

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	INSEL3[6:0]							保留	INSEL2[6:0]						
rs	rw								rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	INSEL1[6:0]							保留	INSEL0[6:0]						
	rw								rw						

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER7IN将不能被修改。 0：TRIGSEL_TIMER7IN 寄存器可读可写 1：TRIGSEL_TIMER7IN 寄存器只读
30:24	INSEL3[6:0]	触发输出 3 的输入源选择 这些位用来选择连接到输出 3 的触发输入信号，输出 3 作为 TIMER7_ITI3 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
23	保留	必须保持复位值。
22:16	INSEL2[6:0]	触发输出 2 的输入源选择 这些位用来选择连接到输出 2 的触发输入信号，输出 2 作为 TIMER7_ITI2 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
15	保留	必须保持复位值。
14:8	INSEL1[6:0]	触发输出 1 的输入源选择 这些位用来选择连接到输出 1 的触发输入信号，输出 1 作为 TIMER7_ITI1 的触发

		源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
7	保留	必须保持复位值。
6:0	INSEL0[6:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER7_ITI0 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。

7.5.9. TIMER7_BRKIN 触发选择寄存器（TRIGSEL_TIMER7BRKIN）

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
LK	INSEL3[6:0]							保留	INSEL2[6:0]						
rs	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	INSEL1[6:0]							保留	INSEL0[6:0]						
	rw								rw						

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER7BRKIN将不能被修改。 0：TRIGSEL_TIMER7BRKIN 寄存器可读可写 1：TRIGSEL_TIMER7BRKIN 寄存器只读
30:24	INSEL3[6:0]	触发输出 3 的输入源选择 这些位用来选择连接到输出 3 的触发输入信号，输出 3 作为 TIMER7_BRKIN3 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
23	保留	必须保持复位值。
22:16	INSEL2[6:0]	触发输出 2 的输入源选择 这些位用来选择连接到输出 2 的触发输入信号，输出 2 作为 TIMER7_BRKIN2 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
15	保留	必须保持复位值。
14:8	INSEL1[6:0]	触发输出 1 的输入源选择 这些位用来选择连接到输出 1 的触发输入信号，输出 1 作为 TIMER7_BRKIN1 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
7	保留	必须保持复位值。
6:0	INSEL0[6:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER7_BRKIN0 的触

发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。

7.5.10. TIMER19_ITI 触发选择寄存器（TRIGSEL_TIMER19IN）

地址偏移：0x24

复位值：0x3939 3939

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	INSEL3[6:0]							保留	INSEL2[6:0]						
rs	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	INSEL1[6:0]							保留	INSEL0[6:0]						
	rw								rw						

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER19IN将不能被修改。 0：TRIGSEL_TIMER19IN 寄存器可读可写 1：TRIGSEL_TIMER19IN 寄存器只读
30:24	INSEL3[6:0]	触发输出 3 的输入源选择 这些位用来选择连接到输出 3 的触发输入信号，输出 3 作为 TIMER19_ITI3 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
23	保留	必须保持复位值。
22:16	INSEL2[6:0]	触发输出 2 的输入源选择 这些位用来选择连接到输出 2 的触发输入信号，输出 2 作为 TIMER19_ITI2 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
15	保留	必须保持复位值。
14:8	INSEL1[6:0]	触发输出 1 的输入源选择 这些位用来选择连接到输出 1 的触发输入信号，输出 1 作为 TIMER19_ITI1 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
7	保留	必须保持复位值。
6:0	INSEL0[6:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER19_ITI0 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。

7.5.11. TIMER19_BRKIN 触发选择寄存器（TRIGSEL_TIMER19BRKIN）

地址偏移：0x28

复位值：0x3939 3939

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	INSEL3[6:0]							保留	INSEL2[6:0]						
rs	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	INSEL1[6:0]							保留	INSEL0[6:0]						
	rw								rw						

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER19BRKIN将不能被修改。 0：TRIGSEL_TIMER19BRKIN 寄存器可读可写 1：TRIGSEL_TIMER19BRKIN 寄存器只读
30:24	INSEL3[6:0]	触发输出 3 的输入源选择 这些位用来选择连接到输出 3 的触发输入信号，输出 3 作为 TIMER19_BRKIN3 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
23	保留	必须保持复位值。
22:16	INSEL2[6:0]	触发输出 2 的输入源选择 这些位用来选择连接到输出 2 的触发输入信号，输出 2 作为 TIMER19_BRKIN2 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
15	保留	必须保持复位值。
14:8	INSEL1[6:0]	触发输出 1 的输入源选择 这些位用来选择连接到输出 1 的触发输入信号，输出 1 作为 TIMER19_BRKIN1 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
7	保留	必须保持复位值。
6:0	INSEL0[6:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER19_BRKIN0 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。

7.5.12. TIMER20_ITI 触发选择寄存器（TRIGSEL_TIMER20IN）

地址偏移：0x2C

复位值：0x1515 1515

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	INSEL3[6:0]							保留	INSEL2[6:0]						
rs	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	INSEL1[6:0]							保留	INSEL0[6:0]						
	rw								rw						

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER20IN将不能被修改。 0：TRIGSEL_TIMER20IN 寄存器可读可写 1：TRIGSEL_TIMER20IN 寄存器只读
30:24	INSEL3[6:0]	触发输出 3 的输入源选择 这些位用来选择连接到输出 3 的触发输入信号，输出 3 作为 TIMER20_ITI3 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
23	保留	必须保持复位值。
22:16	INSEL2[6:0]	触发输出 2 的输入源选择 这些位用来选择连接到输出 2 的触发输入信号，输出 2 作为 TIMER20_ITI2 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
15	保留	必须保持复位值。
14:8	INSEL1[6:0]	触发输出 1 的输入源选择 这些位用来选择连接到输出 1 的触发输入信号，输出 1 作为 TIMER20_ITI1 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
7	保留	必须保持复位值。
6:0	INSEL0[6:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER20_ITI0 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。

7.5.13. TIMER20_BRKIN 触发选择控制寄存器（TRIGSEL_TIMER20BRKIN）

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
LK	INSEL3[6:0]							保留	INSEL2[6:0]						
rs	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	INSEL1[6:0]							保留	INSEL0[6:0]						
	rw								rw						

位/位域	名称	描述

31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER20BRINK将不能被修改。 0：TRIGSEL_TIMER20BRKIN 寄存器可读可写 1：TRIGSEL_TIMER20BRKIN 寄存器只读
30:24	INSEL3[6:0]	触发输出 3 的输入源选择 这些位用来选择连接到输出 3 的触发输入信号，输出 3 作为 TIMER20_BRINK3 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
23	保留	必须保持复位值。
22:16	INSEL2[6:0]	触发输出 2 的输入源选择 这些位用来选择连接到输出 2 的触发输入信号，输出 2 作为 TIMER20_BRINK2 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
15	保留	必须保持复位值。
14:8	INSEL1[6:0]	触发输出去 1 的输入源选择 这些位用来选择连接到输出 1 的触发输入信号，输出 1 作为 TIMER20_BRINK1 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
7	保留	必须保持复位值。
6:0	INSEL0[6:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER20_BRINK0 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。

7.5.14. TIMER1_ITI 触发选择寄存器（TRIGSEL_TIMER1IN）

地址偏移：0x34

复位值：0x1E1E 1E1E

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	INSEL3[6:0]							保留	INSEL2[6:0]						
rs	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	INSEL1[6:0]							保留	INSEL0[6:0]						
	rw								rw						

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER1IN将不能被修改。 0：TRIGSEL_TIMER1IN 寄存器可读可写 1：TRIGSEL_TIMER1IN 寄存器只读

30:24	INSEL3[6:0]	触发输出 3 的输入源选择 这些位用来选择连接到输出 3 的触发输入信号，输出 3 作为 TIMER1_ITI3 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
23	保留	必须保持复位值。
22:16	INSEL2[6:0]	触发输出 2 的输入源选择 这些位用来选择连接到输出 2 的触发输入信号，输出 2 作为 TIMER1_ITI2 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
15	保留	必须保持复位值。
14:8	INSEL1[6:0]	触发输出 1 的输入源选择 这些位用来选择连接到输出 1 的触发输入信号，输出 1 作为 TIMER1_ITI1 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
7	保留	必须保持复位值。
6:0	INSEL0[6:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER1_ITI0 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。

7.5.15. MFCOM register 触发选择寄存器（TRIGSEL_MFCOM）

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
LK	INSEL3[6:0]							保留	INSEL2[6:0]						
rs	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	INSEL1[6:0]							保留	INSEL0[6:0]						
	rw								rw						

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_MFCOM将不能被修改。 0：TRIGSEL_MFCOM 寄存器可读可写 1：TRIGSEL_MFCOM 寄存器只读
30:24	INSEL3[6:0]	触发输出 3 输入源选择 这些位用来选择连接到输出 3 的触发输入信号，输出 3 作为 MFCOM_TRG_TIMER3 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
23	保留	必须保持复位值。
22:16	INSEL2[6:0]	触发输出 2 的输入源选择

		这些位用来选择连接到输出 2 的触发输入信号，输出 2 作为 MFCOM_TRG_TIMER2 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
15	保留	必须保持复位值。
14:8	INSEL1[6:0]	触发输出 1 的输入源选择 这些位用来选择连接到输出 1 的触发输入信号，输出 1 作为 MFCOM_TRG_TIMER1 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。
7	保留	必须保持复位值。
6:0	INSEL0[6:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 MFCOM_TRG_TIMER0 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。

7.5.16. CAN0 触发选择寄存器（TRIGSEL_CAN0）

地址偏移：0x3C
复位值：0x0000 003A

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									INSEL0[6:0]						
												rw			

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_CAN0将不能被修改。 0：TRIGSEL_CAN0 寄存器可读可写 1：TRIGSEL_CAN0 寄存器只读
30:7	保留	必须保持复位值。
6:0	INSEL0[6:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出的触发输入信号，输出 0 作为 CAN0_EX_TIME_TICK 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。

7.5.17. CAN1 触发选择寄存器（TRIGSEL_CAN1）

地址偏移：0x40
复位值：0x0000 003A

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									INSEL0[6:0]						
												rw			

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_CAN1将不能被修改。 0：TRIGSEL_CAN1 寄存器可读可写 1：TRIGSEL_CAN1 寄存器只读
30:7	保留	必须保持复位值。
6:0	INSEL0[6:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出的触发输入信号，输出 0 作为 CAN1_EX_TIME_TICK 的触发源。关于具体配置请参考*表 7-1. 触发输入位域选择*。

8. 通用和备用输入/输出接口（GPIO 和 AFIO）

8.1. 简介

最多可支持 88 个通用 I/O 引脚（GPIO），分别为 PA0 ~ PA15，PB0 ~ PB15，PC0 ~ PC15，PD0 ~ PD15，PE0 ~ PE15，PF0 ~ PF7。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。片上设备 GPIO 引脚的外部中断由 EXTI 模块的寄存器控制和配置。

GPIO 端口和其他的备用功能（AFs）备用引脚，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引脚，备用功能输入/输出都可以。

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

8.2. 主要特征

- 输入/输出方向控制；
- 施密特触发输入功能使能控制；
- 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；
- 可编程的边沿触发外部中断-由 EXTI 寄存器配置；
- 模拟输入/输出配置；
- 备用功能输入/输出配置；
- 端口锁定配置；
- 单周期输出翻转功能。

8.3. 功能说明

每个通用I/O端口都可以通过32位控制寄存器（GPIOx_CTL）配置为GPIO输入，GPIO输出，AF功能或模拟模式。引脚AFIO输入/输出是通过AFIO功能使能来选择。当端口配置为输出（GPIO输出或AFIO输出）时，可以通过GPIO输出模式寄存器（GPIOx_OMODE）配置为推挽或开漏模式。输出端口的最大速度可以通过GPIO输出速度寄存器（GPIOx_OSPD）配置。每个端口可以通过GPIO上/下拉寄存器（GPIOx_PUD）配置为浮空（无上拉或下拉），上拉或下拉功能。

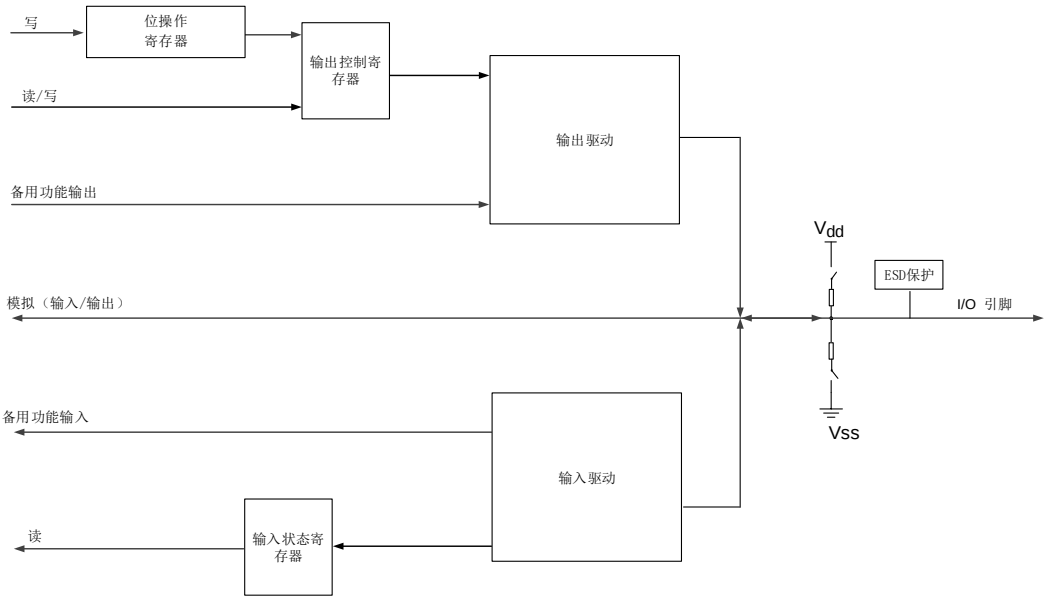
表 8-1. GPIO 配置表

PAD TYPE			CTLy	OMy	PUDy
GPIO 输入	X	悬空	00	X	00
		上拉			01
		下拉			10

PAD TYPE			CTLy	OMy	PUDy
GPIO 输出	推挽	悬空	01	0	00
		上拉			01
		下拉			10
	开漏	悬空		1	00
		上拉			01
		下拉			10
AFIO 输入	X	悬空	10	X	00
		上拉			01
		下拉			10
AFIO 输出	推挽	悬空	10	0	00
		上拉			01
		下拉			10
	开漏	悬空		1	00
		上拉			01
		下拉			10
ANALOG	X	X	11	X	XX

*图8-1. GPIO端口位的基本结构*为标准I/O端口位的基本结构图。

图 8-1. GPIO 端口位的基本结构



8.3.1. GPIO 引脚配置

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

PB7：JTDI为PU上拉模式
PB8：JTCK/SWCLK为PD下拉模式
PB9：JTMS/SWDIO为PU上拉模式

PB3：NJTRST为PU上拉模式
PB4：JTDO为输出模式

GPIO管脚可以配置为输入或输出。并且所有的GPIO管脚都有一个内部的弱上拉和弱下拉可以选择。当GPIO管脚可配置为输入管脚时，外部管脚上的数据在每个AHB时钟周期时都会装载到端口输入状态寄存器（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。端口输出控制寄存器（GPIOx_OCTL）的值将会从相应I/O引脚上输出。

当需要对GPIOx_OCTL进行按位写操作时不需关中断，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，或用于翻转操作的GPIOx_TG）修改一位或几位，该过程仅需要一个最小的AHB写访问周期，而其他位不受影响。

8.3.2. 外部中断及事件

所有的端口都有外部中断的能力，如果想使用端口的外部中断功能，需要配置为输入模式。

8.3.3. 备用功能（AF）

当端口配置为AFIO（设置GPIOx_CTL寄存器中的CTLy值为“0b10”）时，该端口用作外设备用功能。通过配置GPIO备用功能选择寄存器（GPIOx_AFSELy(y=0..1)），每个端口可以配置16个备用功能。端口备用功能分配的详细介绍见芯片数据手册。

8.3.4. 附加功能

有些引脚具有附加功能，它们优先于标准GPIO寄存器中的配置。当用作ADC，DAC， CMP或附加功能时，引脚必须配置成模拟模式。当引脚用作RTC、WKUPx和振荡器附加功能时，端口类型通过相关的RTC、PMU和RCU寄存器自动设置。当附加功能禁用时，这些端口可用作普通GPIO。

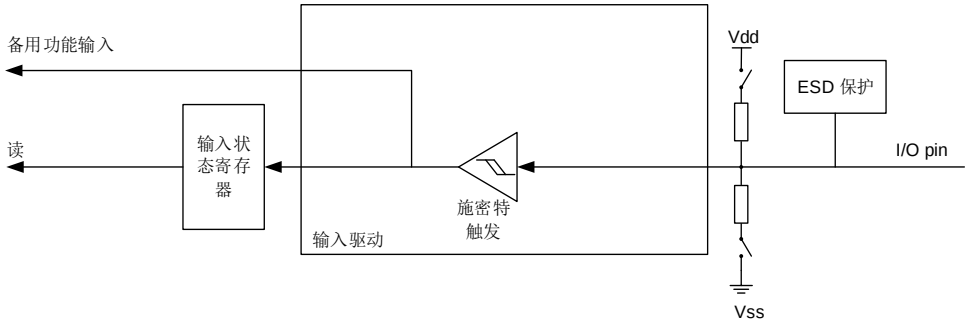
8.3.5. 输入配置

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AHB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*是I/O引脚的输入配置。

图 8-2. 输入配置的基本结构



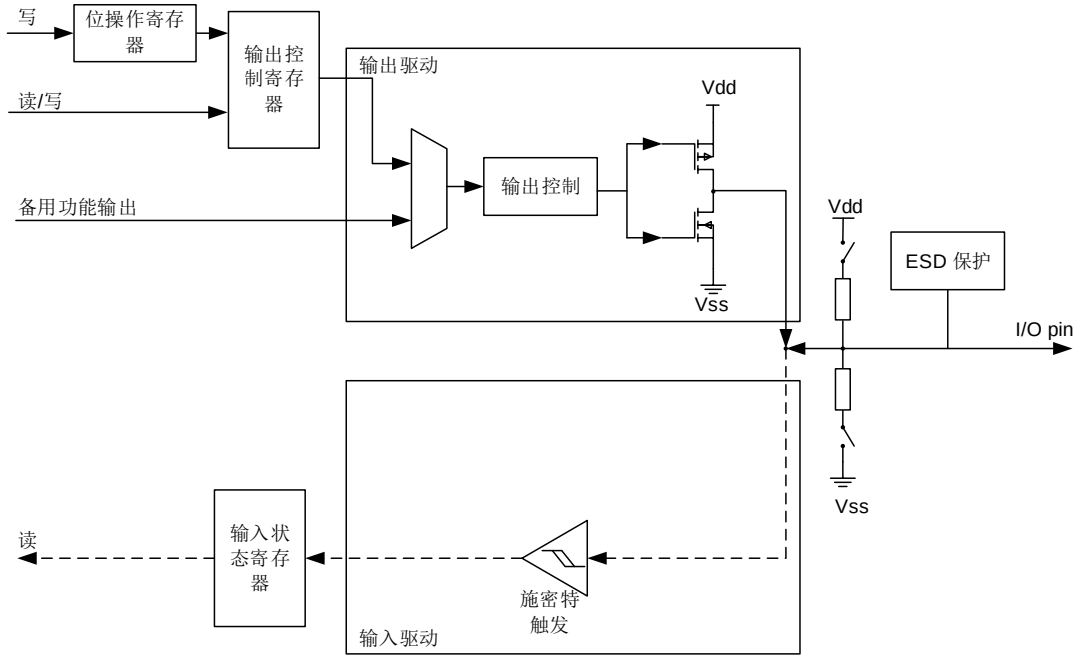
8.3.6. 输出配置

当GPIO配置为输出时：

- 施密特触发输入使能；
- 可选择的弱上拉和下拉电阻；
- 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；
- 在推挽模式下，对端口输出控制寄存器的读访问将返回上次写入的值；
- 在开漏模式下，对端口输入状态寄存器的读访问将返回I/O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



8.3.7. 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；
- 读端口输入状态寄存器返回“0”。

*图8-4. 模拟配置的基本结构*是I/O端口的模拟高阻配置。

图 8-4. 模拟配置的基本结构

8.3.8. 备用功能（AF）配置

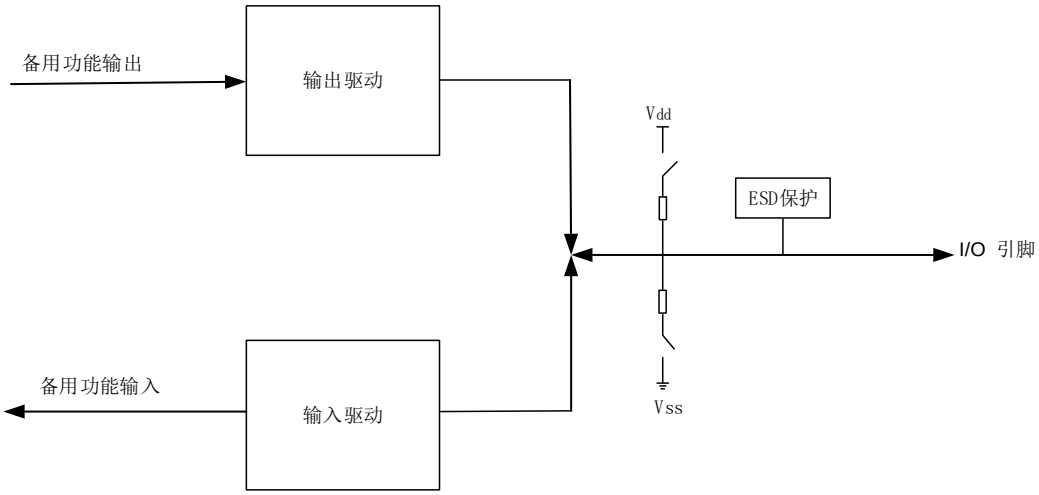
为了适应不同的器件封装，GPIO端口支持软件配置将一些备用功能应用到其他引脚上。

当引脚配置为备用功能时：

- 输出缓冲器启用开漏或者推挽功能；
- 输出缓冲器由外设驱动；
- 施密特触发输入使能；
- 可选择的弱上拉/下拉电阻；
- I/O引脚上的数据在每个AHB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



8.3.9. 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_CTL，GPIOx_OMODE，GPIOx_OSPD，GPIOx_PUD和GPIOx_AFSELy(y=0..1)。通过配置32位锁定寄存器（GPIOx_LOCK）可以锁定I/O端口的配置。当特定LOCK序列写到位于GPIOx_LOCK寄存器的LKK位上，并且LKy被置位，那么对应的端口配置直到下一次复位前将不能改变。建议在电源驱动模块驱动的配置时使用锁定功能。

8.3.10. GPIO 单周期输出翻转功能

通过将GPIOx_TG寄存器中对应的位写1，GPIO可以在一个AHB时钟周期内翻转I/O的输出电平。输出信号的频率可以达到AHB时钟的一半。

8.4. GPIO 寄存器

GPIOA基地址：0x4800 0000

GPIOB基地址：0x4800 0400

GPIOC基地址：0x4800 0800

GPIOD基地址：0x4800 0C00

GPIOE基地址：0x4800 1000

GPIOF基地址：0x4800 1400

8.4.1. 端口控制寄存器（GPIOx_CTL，x=A..F）

地址偏移：0x00
复位值：端口 B 0x000A 8280；其他端口 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
CTL15[1:0]	CTL14[1:0]	CTL13[1:0]	CTL12[1:0]	CTL11[1:0]	CTL10[1:0]	CTL9[1:0]	CTL8[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
CTL7[1:0]	CTL6[1:0]	CTL5[1:0]	CTL4[1:0]	CTL3[1:0]	CTL2[1:0]	CTL1[1:0]	CTL0[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。 参照CTL0[1:0]的描述
29:28	CTL14[1:0]	Pin 14配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
27:26	CTL13[1:0]	Pin 13配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
25:24	CTL12[1:0]	Pin 12配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
23:22	CTL11[1:0]	Pin 11配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
21:20	CTL10[1:0]	Pin 10配置位

		该位由软件置位和清除。 参照CTL0[1:0]的描述
19:18	CTL9[1:0]	Pin 9配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
17:16	CTL8[1:0]	Pin 8配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
15:14	CTL7[1:0]	Pin 7配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
13:12	CTL6[1:0]	Pin 6配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
11:10	CTL5[1:0]	Pin 5配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
9:8	CTL4[1:0]	Pin 4配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
7:6	CTL3[1:0]	Pin 3配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
5:4	CTL2[1:0]	Pin 2配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
3:2	CTL1[1:0]	Pin 1配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
1:0	CTL0[1:0]	Pin 0配置位 该位由软件置位和清除。 00：GPIO输入模式（复位值） 01：GPIO输出模式 10：备用功能模式 11：模拟模式（输入和输出）

8.4.2. 端口输出模式寄存器（GPIOx_OMODE，x=A..F）

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
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OM15	OM14	OM13	OM12	OM11	OM10	OM9	OM8	OM7	OM6	OM5	OM4	OM3	OM2	OM1	OM0
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	OM15	Pin 15输出模式位 该位由软件置位和清除。 参考OM0的描述
14	OM14	Pin 14输出模式位 该位由软件置位和清除。 参考OM0的描述
13	OM13	Pin 13输出模式位 该位由软件置位和清除。 参考OM0的描述
12	OM12	Pin 12输出模式位 该位由软件置位和清除。 参考OM0的描述
11	OM11	Pin 11输出模式位 该位由软件置位和清除。 参考OM0的描述
10	OM10	Pin 10输出模式位 该位由软件置位和清除。 参考OM0的描述
9	OM9	Pin 9输出模式位 该位由软件置位和清除。 参考OM0的描述
8	OM8	Pin 8输出模式位 该位由软件置位和清除。 参考OM0的描述
7	OM7	Pin 7输出模式位 该位由软件置位和清除。 参考OM0的描述

6	OM6	Pin 6输出模式位 该位由软件置位和清除。 参考OM0的描述
5	OM5	Pin 5输出模式位 该位由软件置位和清除。 参考OM0的描述
4	OM4	Pin 4输出模式位 该位由软件置位和清除。 参考OM0的描述
3	OM3	Pin 3输出模式位 该位由软件置位和清除。 参考OM0的描述
2	OM2	Pin 2输出模式位 该位由软件置位和清除。 参考OM0的描述
1	OM1	Pin 1输出模式位 该位由软件置位和清除。 参考OM0的描述
0	OM0	Pin 0输出模式位 该位由软件置位和清除。 0：输出推挽模式（复位值） 1：输出开漏模式

8.4.3. 端口输出速度寄存器（GPIOx_OSPD，x=A..F）

地址偏移：0x08

复位值：端口 B 0x000C 0000；其他端口 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
OSPD15[1:0]	OSPD14[1:0]	OSPD13[1:0]	OSPD12[1:0]	OSPD11[1:0]	OSPD10[1:0]	OSPD9[1:0]	OSPD8[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
OSPD7[1:0]	OSPD6[1:0]	OSPD5[1:0]	OSPD4[1:0]	OSPD3[1:0]	OSPD2[1:0]	OSPD1[1:0]	OSPD0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	OSPD15[1:0]	Pin 15输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述

29:28	OSPD14[1:0]	Pin 14输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
27:26	OSPD13[1:0]	Pin 13输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
25:24	OSPD12[1:0]	Pin 12输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
23:22	OSPD11[1:0]	Pin 11输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
21:20	OSPD10[1:0]	Pin 10输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
19:18	OSPD9[1:0]	Pin 9输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
17:16	OSPD8[1:0]	Pin 8输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
15:14	OSPD7[1:0]	Pin 7输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
13:12	OSPD6[1:0]	Pin 6输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
11:10	OSPD5[1:0]	Pin 5输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
9:8	OSPD4[1:0]	Pin 4输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
7:6	OSPD3[1:0]	Pin 3输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
5:4	OSPD2[1:0]	Pin 2输出最大速度位 该位由软件置位和清除。

		参考OSPD0[1:0]的描述
3:2	OSPD1[1:0]	Pin 1输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
1:0	OSPD0[1:0]	Pin 0输出最大速度位 该位由软件置位和清除。 x0：输出最大速度2M（复位值） 01：输出最大速度10M 11：输出最大速度50M

8.4.4. 端口上拉/下拉寄存器（GPIOx_PUD，x=A..F）

地址偏移：0x0C

复位值：端口 B 0x0006 4040；其他端口 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
PUD15[1:0]	PUD14[1:0]	PUD13[1:0]	PUD12[1:0]	PUD11[1:0]	PUD10[1:0]	PUD9[1:0]	PUD8[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
PUD7[1:0]	PUD6[1:0]	PUD5[1:0]	PUD4[1:0]	PUD3[1:0]	PUD2[1:0]	PUD1[1:0]	PUD0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	PUD15[1:0]	Pin 15上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
29:28	PUD14[1:0]	Pin 14上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
27:26	PUD13[1:0]	Pin 13上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
25:24	PUD12[1:0]	Pin 12上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
23:22	PUD11[1:0]	Pin 11上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
21:20	PUD10[1:0]	Pin 10上拉或下拉位

		该位由软件置位和清除。 参照PUD0[1:0]的描述
19:18	PUD9[1:0]	Pin 9上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
17:16	PUD8[1:0]	Pin 8上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
15:14	PUD7[1:0]	Pin 7上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
13:12	PUD6[1:0]	Pin 6上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
11:10	PUD5[1:0]	Pin 5上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
9:8	PUD4[1:0]	Pin 4上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
7:6	PUD3[1:0]	Pin 3上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
5:4	PUD2[1:0]	Pin 2上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
3:2	PUD1[1:0]	Pin 1上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
1:0	PUD0[1:0]	Pin 0上拉或下拉位 该位由软件置位和清除。 00：悬空模式，无上拉和下拉（复位值） 01：端口上拉模式 10：端口下拉模式 11：保留

8.4.5. 端口输入状态寄存器（GPIOx_ISTAT，x=A..F）

地址偏移：0x10

复位值：0x0000 XXXX

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
ISTAT15	ISTAT14	ISTAT13	ISTAT12	ISTAT11	ISTAT10	ISTAT9	ISTAT8	ISTAT7	ISTAT6	ISTAT5	ISTAT4	ISTAT3	ISTAT2	ISTAT1	ISTAT0
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) 0：引脚输入信号为低电平 1：引脚输入信号为高电平

8.4.6. 端口输出控制寄存器（GPIOx_OCTL，x=A..F）

地址偏移：0x14
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
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) 该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

8.4.7. 端口位操作寄存器（GPIOx_BOP，x=A..F）

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
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) 该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) 该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

8.4.8. 端口配置锁定寄存器（GPIOx_LOCK，x=A..F）

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															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	锁定键 该位只能通过Lock Key写序列置位，始终可读。 0：GPIOx_LOCK寄存器和端口配置没有锁定 1：直到下一次MCU复位前，GPIOx_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 y(y=0..15)的值必须保持。
15:0	LKy	端口锁定位y(y=0..15) 该位由软件置位和清除。 0：端口配置没有锁定

1：端口配置锁定

8.4.9. 备用功能选择寄存器 0（GPIOx_AFSEL0，x=A..F）

地址偏移：0x20

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
SEL7[3:0]				SEL6[3:0]				SEL5[3:0]				SEL4[3:0]			
rw				rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL3[3:0]				SEL2[3:0]				SEL1[3:0]				SEL0[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:28	SEL7[3:0]	Pin 7选择备用功能 该位由软件置位和清除。 参照SEL0 [3:0]的描述
27:24	SEL6[3:0]	Pin 6选择备用功能 该位由软件置位和清除。 参照SEL0 [3:0]的描述
23:20	SEL5[3:0]	Pin 5选择备用功能 该位由软件置位和清除。 参照SEL0 [3:0]的描述
19:16	SEL4[3:0]	Pin 4选择备用功能 该位由软件置位和清除。 参照SEL0 [3:0]的描述
15:12	SEL3[3:0]	Pin 3选择备用功能 该位由软件置位和清除。 参照SEL0 [3:0]的描述
11:8	SEL2[3:0]	Pin 2选择备用功能 该位由软件置位和清除。 参照SEL0 [3:0]的描述
7:4	SEL1[3:0]	Pin 1选择备用功能 该位由软件置位和清除。 参照SEL0 [3:0]的描述
3:0	SEL0[3:0]	Pin 0选择备用功能 该位由软件置位和清除。 0000：选择AF0功能（复位值）

0001：选择AF1功能
0010：选择AF2功能
0011：选择AF3功能
0100：选择AF4功能
0101：选择AF5功能
0110：选择AF6功能
0111：选择AF7功能
1000：选择AF8功能
1001：选择AF9功能

1010 ~ 1111：保留

8.4.10. 备用功能选择寄存器 1（GPIOx_AFSEL1，x=A..F）

地址偏移：0x24
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
SEL15[3:0]				SEL14[3:0]				SEL13[3:0]				SEL12[3:0]			
rw				rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL11[3:0]				SEL10[3:0]				SEL9[3:0]				SEL8[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:28	SEL15[3:0]	Pin 15选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
27:24	SEL14[3:0]	Pin 14选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
23:20	SEL13[3:0]	Pin 13选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
19:16	SEL12[3:0]	Pin 12选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
15:12	SEL11[3:0]	Pin 11选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述

11:8	SEL10[3:0]	Pin 10选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
7:4	SEL9[3:0]	Pin 9选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
3:0	SEL8[3:0]	Pin 8选择备用功能 该位由软件置位和清除。 0000：选择AF0功能（复位值） 0001：选择AF1功能 0010：选择AF2功能 0011：选择AF3功能 0100：选择AF4功能 0101：选择AF5功能 0110：选择AF6功能 0111：选择AF7功能 1000：选择AF8功能 1001：选择AF9功能 1010 ~ 1111：保留

8.4.11. 位清除寄存器（GPIOx_BC，x=A..F）

地址偏移：0x28

复位值：0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

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

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

8.4.12. 端口位翻转寄存器（GPIOx_TG，x=A..F）

地址偏移：0x2C

复位值：0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TG15	TG14	TG13	TG12	TG11	TG10	TG9	TG8	TG7	TG6	TG5	TG4	TG3	TG2	TG1	TG0
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	TGy	端口翻转位y（y=0..15） 该位由软件置位和清除。 0：相应OCTLy位没有改变 1：翻转相应的OCTLy位

9. 多功能通信接口（MFCOM）

9.1. 简介

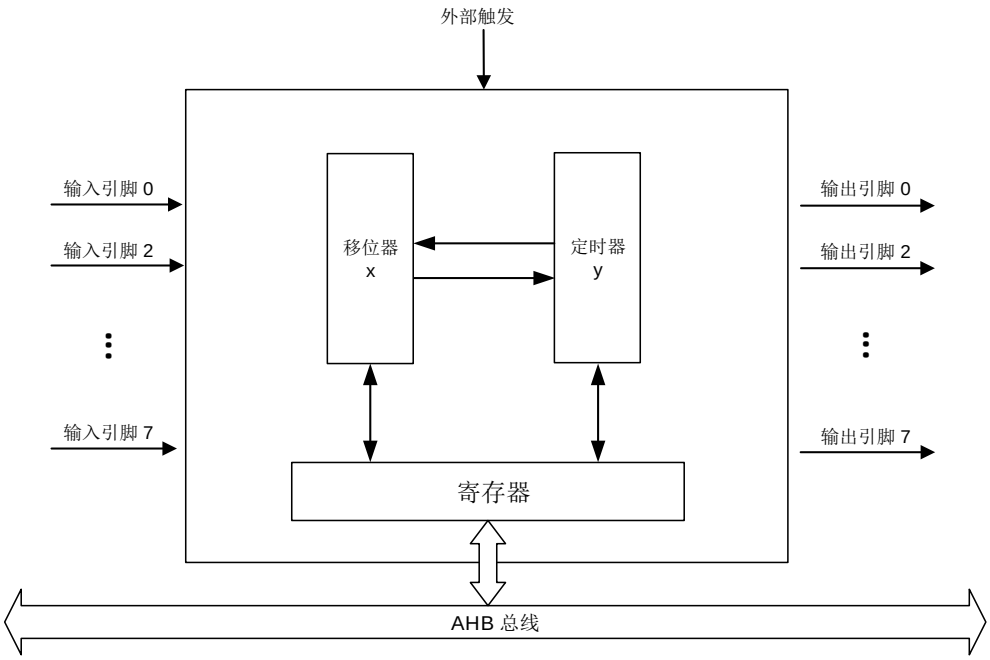
MFCOM 是一个高度可配置的模块，提供了多种串行通信协议的仿真和灵活的定时器配置。

9.2. 主要特征

- 连续数据传输配置；
- 双缓存区移位寄存器支持连续数据传输；
- 自动启动 / 停止控制；
- DMA、中断或轮询方式控制发送 / 接收；
- 高度灵活的16位定时器，支持多种内部或外部触发、复位、使能和禁用方式；
- 基于HCLK的波特率重编程；
- 结合引脚 / 移位 / 定时器功能以支持PWM波形生成，通信接口仿真，如UART，I2C，SPI，I2S等；

9.3. 框图

图 9-1. MFCOM 结构框



通过 AHB 总线时钟访问 MFCOM 寄存器，MFCOM 有 4 个定时器，4 个移位器和 8 个引脚。可以通过位域 SPSEL[2:0]选择移位器的引脚输入和输出。通过配置 INSRC 位，可以选择移位器的输入源（移位器的输出或引脚的输入）。

数据的加载、存储等事件可以通过使用 DMA/轮询/中断方法触发信号。通过配置定时器来选择触发源（外部触发，引脚或移位器状态标志），定时器为传输数据产生移位时钟，并且移位器可选择不同引脚输入 / 输出数据。

9.4. 功能说明

9.4.1. 时钟和复位

MFCOM 时钟

RCU 模块中的 MFCOMEN 位必须在访问任何 MFCOM 寄存器之前使能。MFCOM 时钟由 RCU 配置使能，选择 AHB 时钟（MFCOM 的时钟使能和复位由 RCU 寄存器控制）。

MFCOM 复位

MFCOM 允许通过设置 MFCOM_CTL 寄存器中的 SWRSTEN 位来进行软件复位。软件复位后除 MFCOM_CTL 寄存器外，MFCOM 的其余寄存器被重置为默认状态。系统复位将重置 MFOCM 的所有逻辑和寄存器。

MFCOM 模式

MFCOM 模块支持正常模式和调试模式。如果 DBG 模块的 DBG_CTL0 寄存器中的 MFCOM_HOLD 位置位，即内核停止时保持 MFCOM 计数器不变，用于调试。当 MFOCM_HOLD 位被清除时，MFCOM 将退出调试模式。

9.4.2. 移位器

移位器是用来缓存和传输 MFCOM 数据。通过 MFCOM_SCTLx 寄存器中的 TMSEL[1:0]位域来配置移位器所使用的定时器，从而控制移位、加载以及存储事件的时机。当处于发送模式时，如果停止位被启用，移位器在初始配置时则立即插入停止位。在接收模式下，在 MFCOM_SCFG 和 MFCOM_TCFG 寄存器中配置 SSTART、TMSTART 或 SSTOP，TMSTOP 检查起始 / 停止位可以在移位数据前 / 后启用。

图 9-2. 移位器微体系结构

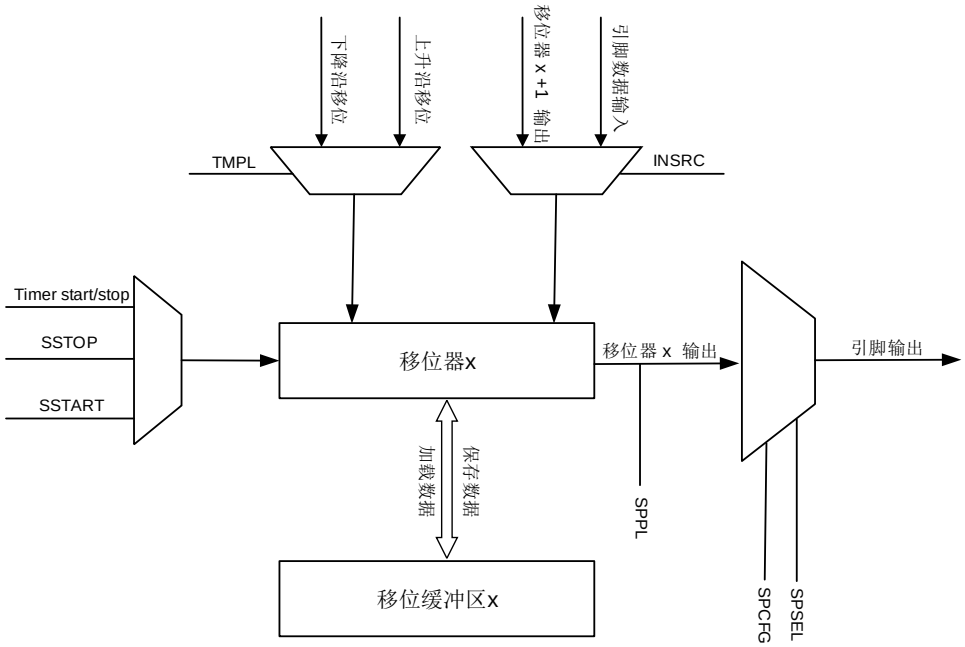


表 9-1. 移位器模式

模式	移位器工作原理	SSTAT、中断、DMA 请求设置条件	错误中断设置条件
发送	当一个加载事件被指定的定时器触发时，移位器将从移位缓冲区加载数据并将数据移出	数据已从移位缓冲区加载到移位器中，或当移位器刚配置为发送模式时	尝试从一个空的移位缓冲区加载数据（缓冲区未满）
接收	当一个存储事件被指定的定时器触发时，移位器将传输数据并将数据存储到移位缓冲区中	数据已从移位器存储到移位缓冲区	尝试将数据存储到满的移位缓冲区（缓冲区溢出）或在起始 / 停止位校验不匹配

9.4.3. 定时器

移位寄存器的加载、存储、开始和结束由 MFCOM 定时器控制，并在 MFCOM_CLK 上递减到零时，计数器加载比较寄存器的内容。可以设置起始条件和停止条件来配置定时器产生一个时钟或 PWM 波形。

定时器独立运行，但是定时器输出可以用来触发任何其他定时器，也可以配置后一个定时器来启用或禁用前一个定时器。定时器输出，引脚输入，移位状态标志或外部触发输入可以由每个定时器独立配置。在设置定时器配置之前，应该配置定时器配置寄存器（MFCOM_TMCFG）。定时器触发配置是一个引脚独立的配置，可以配置输入 / 输出数据，或输出使能。

8 位波特计数器配置

16 位计数器分为两个 8 位计数器，低 8 位用于配置移位时钟的波特率，当低 8 位减少到零时，

定时器输出翻转，低 8 位从比较寄存器重新加载。当低 8 位递减到 0 时，高 8 位用来配置传输中每个字节的位数。

在 8 位波特计数器配置下，定时器复位事件只复位低 8 位计数器，而不复位高 8 位计数器，如果配置定时器复位来更新定时器输出状态，则定时器复位事件会导致定时器输出状态的切换。当高 8 位递减到 0 且递减使能时，定时器比较事件发生，定时器状态标志置位。

注意：当位域 TMMOD[1:0]配置为 0b01，则位域 TMDEC[1:0]只能被配置为 0b00 或 0b01。

16 位计数器配置

传输中的移位时钟边沿数或移位时钟的波特率可以由一个 16 位计数器配置。当 16 位计数器为 0 且递减使能时，定时器输出翻转，且计数器从比较寄存器重新加载。当 16 位计数器减为 0 且递减有使能时，定时器比较事件触发，定时器状态标志位被置位。

8 位高 PWM 配置

16 位计数器分为两个 8 位计数器。低 8 位用于配置定时器输出高周期，高 8 位用于配置定时器输出低周期。当低 8 位减为零时，定时器输出被清除，低 8 位从比较寄存器重新加载数据。输出为低时，高 8 位递减。当减为 0 时，定时器输出为高，高 8 位将从比较寄存器重新加载数据。当高 8 位递减到 0 且递减有效时，定时器比较事件触发，定时器状态标志置位。如果位域 TMMOD[1:0]配置为 PWM 功能，则不支持配置位域 TMSTART / TMSTOP[1:0] / TMOUT[1:0]。

定时器使能和开始位

当 MFCOM_TMCTL 寄存器中的位域 TMMOD[1:0]被配置为所需的功能，并且根据定时器（TMEN[2:0]）的配置检测到相应的触发条件时，会发生以下事件。

1. 定时器的计数器将加载比较寄存器的当前值，并根据位域 TMDEC[1:0]的配置开始递减。
2. 根据位域 TMOUT[1:0]的配置，定时器输出会更新到初始状态。由这个定时器控制的移位器不会将这种状态看作定时器移位时钟的上升沿。
3. 根据位域 SSTART[1:0]的配置，从移位缓冲区加载到移位寄存器并输出第一个位，或者定时器控制的传输移位器将输出起始位。

当定时器起始位被使能时，定时器将在移位时钟的第一个上升沿上重新加载比较寄存器。在移位时钟上，如果在第一个上升沿（TMOUT=1）之前没有下降沿，移位器被配置为在下降沿上移位，则第一个移位时将会错误加载。

定时器递减和复位

定时器根据位域 TMMOD[1:0]和 TMDEC[1:0]，产生定时器输出和定时器移位时钟。（当 TMMOD[1:0]≠10 或 11）移位器时钟等于定时器输出或（当 TMDEC[1:0]≠10 或 11）等于递减时钟。当位域 TMDEC[1:0]配置为从引脚或触发源开始递减时，定时器可配置为在上升沿或下降沿递减。

在位域 TMRST[2:0]中，定时器被配置为计数器复位，然后计数器将加载比较寄存器的当前值。定时器移位时钟和定时器输出通过位域 TMOUT[1:0]可以配置定时器复位是否更新输出。如果定时器输出由于定时器复位而翻转，可能导致定时器时钟边沿移位。在 8 位波特计数器配置下，

计数器的高 8 位递减只会发生在低 8 位为 0 且递减有效时。

当定时器的计数器减为 0 时，会触发定时器比较事件。定时器比较事件的触发将导致定时器的计数器从比较寄存器里加载数据，或导致定时器输出翻转，任何配置为发送配置的移位器要从移位缓冲区中加载数据，任何配置为接收模式的移位器要存储数据到移位缓冲区中。定时器根据不同的配置状态标志被置位。

定时器禁用和停止位

将定时器配置为在每次比较中添加一个停止位，然后将发生以下事件。当定时器在每次比较发生时，必须配置第一次发送时，插入一个停止位。

1. 配置移位器为发送模式时，位域 SSTOP[1:0]可以控制自动插入停止位。
2. 配置移位器为接收模式时，通过位域 SSTOP[1:0]，可以将数据存储在移位缓冲区中。
3. 比较事件触发后，在移位时钟的第一个上升沿定时器的计数器将重新加载比较寄存器的当前值。

当位域（TMDIS[2:0]）配置的定时器的禁用条件被检测到时，将发生以下事件。

1. 定时器的计数器将从比较寄存器中加载当前值，根据位域 TMDEC[1:0]的配置递减。
2. 定时器输出将会被清除，不会将定时器输出视为定时器移位时钟上的下降沿，但可以产生一个移位事件。
3. 定时器控制的发送模式的移位器将输出它们的停止位。
4. 定时器控制的接收模式移位器将存储由 SSTOP 配置的移位缓冲区中的移位寄存器数据。

如果定时器停止位被启用，直到检测到移位时钟的下一个上升沿，定时器的计数器将继续递减。定时器移位时钟可以在停止位期间切换而不产生移位事件。定时器输出在停止位期间被强制关闭。

在与定时器禁用条件（停止位被禁用）相同的周期内，或在禁用条件（停止位被启用）后的移位时钟的第一个上升边沿上，可以检测到定时器启用条件。启用停止位时接收模式下的移位寄存器将数据存储到移位缓冲区中，并在定时器处于停止状态时验证所配置的移位边沿上的输入数据的状态。如果禁用定时器后和移位时钟的下一个上升沿之间没有配置停止位，则不会进行存储和验证。

9.4.4. 引脚

每个定时器和移位器可以配置为输入，输出数据，输出使能或级联引脚输入 / 输出数据。每个定时器和移位器的引脚可以配置为使用任何极性。配置为输出使能的引脚可以用作开漏或使能级联输入 / 输出。可以配置任何定时器或移位器来控制引脚的输出使能，引脚级联输出则由另一个定时器或移位器驱动。

引脚同步

当一个引脚被配置为输入时，输入信号在被定时器或移位器使用之前首先与 MFCOM 时钟同步。当使用外部引脚输入用于产生输出或控制移位器时，需要 0.5 到 1.5 个 MFCOM 时钟周期来同步。这将最大同步时间设置为 1.5 倍 MFCOM 时钟周期。如果一个输入被多个定时器或移位器使用，同步只发生一次，以确保使用输入的所有定时器和移位器在使用同一周期上的边沿。

MFCOM 引脚是互连的，配置一个移位器或定时器输出数据在一个未使用的引脚上将建立一个内部连接，其他移位器和定时器可以使用这个引脚作为输入。这意味着移位器输出可用于触发定时器或向定时器输出到移位器。这个过程需要与 MFCOM 时钟同步，所以有 1 个周期延迟。

同步延迟将发生在使用一个引脚输入作为定时器触发源，定时器时钟，或移位器接收数据。外部引脚需要 0.5 到 1.5 倍 MFCOM 时钟周期而内部引脚为 1 个 MFCOM 时钟周期去建立同步。

9.4.5. 中断和 DMA 请求

移位器状态标志，移位器错误标志和定时器状态标志可以产生 MFCOM 中断。具体的中断事件描述和 DMA 请求如*表 9-2. MFCOM 中断和 DMA 请求*所示。

表 9-2. MFCOM 中断和 DMA 请求

中断事件	描述	中断使能位	DMA 请求
SSTAT	移位器状态标志	SSIEN	Y
SERR	移位器错误标志	SEIEN	N
TMSTAT	定时器状态标志	TMSIEN	N

9.4.6. 触发

外部触发

MFCOM 外设触发源与其他外设的连接是特定的。

输入触发

支持多个外部输入触发。外部触发与 MFCOM_CLK 同步时，必须在至少两个时钟周期内被正确采样。输入触发可以用来触发一个或多个 MFCOM 定时器。

输出触发

每个 MFCOM 定时器的输出触发为定时器输出，且定时器输出不受定时器引脚极性配置的影响。

9.5. MFCOM 寄存器

MFCOM 基地址：0x4900 3400

9.5.1. 控制寄存器（MFCOM_CTL）

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														SWRSTEN	MFCOMEN
														rw	rw

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	SWRSTEN	使能软件复位 除控制寄存器外的寄存器访问将被忽略，直到该位被清除。 0：禁能软件复位 1：使能软件复位，除控制寄存器外的所有 MFCOM 寄存器都复位。
0	MFCOMEN	使能 MFCOM 0：禁能 MFCOM 模块 1：使能 MFCOM 模块

9.5.2. 引脚数据寄存器（MFCOM_PINDATA）

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								PDATA[7:0]							
											r				

位/位域	名称	描述
31:8	保留	必须保持复位值。

7:0	PDATA[7:0]	引脚数据 每个 MFCOM 引脚的输入/输出数据

9.5.3. 移位器状态寄存器（MFCOM_SSTAT）

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												SSTAT[3:0]			
														w1c	

位/位域	名称	描述
31:4	保留	必须保持复位值。
3:0	SSTAT[3:0]	移位器 x 状态标志 当发生以下事件之一时，移位器状态标志将置位: SMOD = 接收，当 MFCOM_SBUF 已经从移位器加载数据时，状态标志被置位，当 MFCOM_SBUF 寄存器被读取时，状态标志被清除。 SMOD = 发送，状态标志在 MFCOM_SBUF 数据被传输到移位器或初始配置为这种模式时置位，当状态标志在 MFCOM_SBUF 寄存器被写入数据时被清除。 0：移位器 x 状态标志没有置位 1：移位器 x 状态标志置位

9.5.4. 移位器错误寄存器（MFCOM_SERR）

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												SERR[3:0]			
														w1c	

位/位域	名称	描述
31:4	保留	必须保持复位值。
3:0	SERR[3:0]	移位器 x 错误标志 当发生以下事件之一时，移位错误标志将置位：

SMOD = 接收，MFCOM_SBUF 溢出，或接收的起始或停止位与期望值不匹配。
SMOD = 发送，MFCOM_SBUF 不足。
0：移位器 x 错误标志没有置位
1：移位器 x 错误标志置位

9.5.5. 定时器状态寄存器（MFCOM_TMSTAT）

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												TMSTAT[3:0]			
													w1c		

位/位域	名称	描述
31:4	保留	必须保持复位值。
3:0	TMSTAT[3:0]	定时器 x 状态标志 根据定时器配置，定时器状态标志在以下事件发生时置位。 在 8 位波特计数器配置下，当高 8 位计数器降为零且递减有效时，定时器状态标志将被置位。 在 8 位高 PWM 配置下，当高 8 位计数器下降到零且递减有效时，定时器状态标志将被置位。 在 16 位计数器配置下，当 16 位计数器减降为零且递减有效时，定时器状态标志将被置位。 该位可以通过软件写 1 清除。 0：定时器状态标志没有置位 1：定时器状态标志置位

9.5.6. 移位器状态中断使能寄存器（MFCOM_SSIEN）

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												SSIEN[3:0]			
													rw		

位/位域	名称	描述
31:4	保留	必须保持复位值。
3:0	SSIEN[3:0]	使能移位器状态中断 当位域 SSTAT[3:0]中的移位器 x 状态标志位被置位时使能中断 0：移位器状态标志不产生中断 1：移位器状态标志产生中断

9.5.7. 移位器错误中断使能寄存器（MFCOM_SEIEN）

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												SEIEN[3:0]			
												rw			

位/位域	名称	描述
31:4	保留	必须保持复位值。
3:0	SEIEN[3:0]	使能移位器错误中断 当位域 SERR[3:0]中的移位器 x 错误标志位被置位时使能中断 0：移位器错误标志不产生中断 1：移位器错误标志产生中断

9.5.8. 定时器状态中断使能寄存器（MFCOM_TMSIEN）

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												TMSIEN[3:0]			
												rw			

位/位域	名称	描述
31:4	保留	必须保持复位值。

3:0	TMSIEN[3:0]	使能定时器状态中断 当位域 TMSTAT[3:0]中的定时器 x 状态标志位被置位时使能中断 0：定时器状态标志不产生中断 1：定时器状态标志产生中断

9.5.9. 移位器状态 DMA 使能寄存器（MFCOM_SSDMAEN）

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												SSDMAEN[3:0]			
														rw	

位/位域	名称	描述
31:4	保留	必须保持复位值。
3:0	SSDMAEN[3:0]	使能移位器状态 DMA 当位域 SSTAT[3:0]中的定时器 x 状态标志位被置位时使能 DMA 0：移位器状态标志不产生 DMA 请求 1：移位器状态标志产生 DMA 请求

9.5.10. 移位器控制寄存器 x（MFCOM_SCTLx）

地址偏移：0x80 + 0x04 * x，（x = 0…3）
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						TMSEL[1:0]		TMPL	保留					SPCFG[1:0]	
						rw		rw						rw	

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

位/位域	名称	描述
31:26	保留	必须保持复位值
25:24	TMSEL[1:0]	定时器选择 选择用来产生移位时钟和控制移位逻辑的定时器。 00：选择定时器 0

		01：选择定时器 1 10：选择定时器 2 11：选择定时器 3
23	TMPL	定时器极性 0：在移位时钟的上升沿移位 1：在移位时钟的下降沿移位
22:18	保留	必须保持复位值
17:16	SPCFG[1:0]	移位器引脚配置 00：移位器引脚输入 01：移位器引脚开漏 10：移位器级联引脚输入 / 输出数据 11：移位器引脚输出
15:11	保留	必须保持复位值。
10:8	SPSEL[2:0]	移位器引脚选择 选择一个引脚用作移位器输入或输出
7	SPPL	移位器引脚极性 0：引脚高电平有效 1：引脚低电平有效
6:3	保留	必须保持复位值。
2:0	SMOD[2:0]	移位器模式 配置移位器模式 000：禁能 001：接收模式 010：发送模式 011：保留 100：保留 101：保留 110：保留 111：保留

9.5.11. 移位器配置寄存器 x（MFCOM_SCFGx）

地址偏移：0x100 + 0x004 * x，（x = 0…3）
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	INSRC	保留	SSTOP[1:0]	保留	SSTART[1:0]
	rw		rw		rw

位/位域	名称	描述
31:9	保留	必须保持复位值。
8	INSRC	移位器输入源选择 0：引脚 1：移位器 x+1 的输出（x<3）
7:6	保留	必须保持复位值。
5:4	SSTOP[1:0]	移位器停止位 00：禁用停止位 01：保留位 10：发送模式下停止位低电平有效，接收模式下停止位不是低电平，错误标志置位 11：发送模式下停止位高电平有效，接收模式下停止位不是高电平，错误标志置位 **注意**：发送模式下，若所选的定时器已启用停止位，该数据帧允许停止位自动插入。接收模式下，若所选的定时器已启用停止位，该数据帧允许自动校验停止位。
3:2	保留	必须保留复位值。
1:0	SSTART[1:0]	移位器起始位 00：禁用起始位，使能时发送数据 01：禁用起始位，首次移位时发送数据 10：发送模式下首次移位发送数据之前起始位为低电平，接收模式下起始位不是低电平，则错误标志置位 11：发送模式下首次移位发送数据之前起始位为高电平，接收模式下起始位不是高电平，则错误标志置位 **注意**：发送模式下，若所选的定时器已启用起始位，该数据帧允许起始位自动插入。接收模式下，若所选的定时器已启用起始位，该数据帧允许自动校验起始位。

9.5.12. 移位缓冲区寄存器 x（MFCOM_SBUFx）

地址偏移：0x200 + 0x004 * x，（x = 0…3）
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
SBUF[31:16]															
rw															

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

位/位域	名称	描述

31:0	SBUF[31:0]	移位缓冲区 当移位器为接收模式，移位器数据在定时器结束时存储到 MFCOM_SBUF 中。 当移位器为发送模式，SHIFTBUF 里的数据在定时器开始之前加载到移位器。

9.5.13. 移位缓冲区位交换寄存器 x（MFCOM_SBUFBISx）

地址偏移：0x280 + 0x004 * x，（x = 0…3）
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
SBUFBIS[31:16]															
rw															

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

位/位域	名称	描述
31:0	SBUFBIS[31:0]	移位缓冲区位交换 与 MFCOM_SBUF 寄存器同步，读/写寄存器是位交换的，并且读返回 SBUF[0:31]。

9.5.14. 移位缓冲区字节交换寄存器 x（MFCOM_SBUFBYSx）

地址偏移：0x300 + 0x004 * x，（x = 0…3）
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
SBUFBYS[31:16]															
rw															

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

位/位域	名称	描述
31:0	SBUFBYS[31:0]	移位缓冲区字节交换 与 MFCOM_SBUF 寄存器同步，读/写寄存器是字节交换的，并且读取返回 {SBUF[7:0]，SBUF[15:8]，SBUF[23:16]，SBUF[31:24]}。

9.5.15. 移位缓冲区位字节交换寄存器 x（MFCOM_SBUFBBSx）

地址偏移：0x380 + 0x004 * x，（x = 0…3）
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
SBUFBBS[31:16]															
rw															

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

位/位域	名称	描述
31:0	SBUFBBS[31:0]	移位缓冲区位字节交换 与 MFCOM_SBUF 寄存器同步，读/写寄存器是每个字节内位交换的，并读取返回 {SBUF[24:31]，SBUF[16:23]，SBUF[8:15]，SBUF[0:7]}。

9.5.16. 定时器控制寄存器 x（MFCOM_TMCTLx）

地址偏移：0x400 + 0x004 * x，（x = 0…3）
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				TRIGSEL[3:0]				TRIGPL	TRIGSRC	保留				TMPCFG[1:0]	
					rw			rw	rw					rw	

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

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:24	TRIGSEL[3:0]	触发选择 选择外部触发（TRIGSRC = 0）: 0001：外部触发 0 输入 0010：外部触发 1 输入 0100：外部触发 2 输入 1000：外部触发 3 输入 选择内部触发（TRIGSRC = 1）: 0000：引脚 0 0001：移位器 0 标志 0010：引脚 1 0011：定时器 0 触发 0100：引脚 2 0101：移位器 1 标志 0110：引脚 3 0111：定时器 1 触发 1000：引脚 4 1001：移位器 2 标志

		1010：引脚 5 1011：定时器 2 触发 1100：引脚 6 1101：移位器 3 标志 1110：引脚 7 1111：定时器 3 触发
23	TRIGPL	触发极性 0：高电平触发 1：低电平触发
22	TRIGSRC	触发源 0：选择外部触发 1：选择内部触发
21:18	保留	必须保持复位值。
17:16	TMPCFG[1:0]	定时器引脚配置 00：定时器引脚输入 01：定时器引脚开漏 10：定时器级联引脚输入/输出数据 11：定时器引脚输出
15:11	保留	必须保持复位值。
10:8	TMPSEL[2:0]	定时器引脚选择 选择要用于定时器输入/输出的引脚。
7	TMPPL	定时器引脚极性 0：引脚高电平有效 1：引脚低电平有效
6:2	保留	必须保持复位值。
1:0	TMMOD[1:0]	定时器配置 00：禁用定时器 01：8 位波特计数器配置 10：8 位高 PWM 配置 11：16 位计数器配置

9.5.17. 定时器配置寄存器 x（MFCOM_TMCFGx）

地址偏移：0x480 + 0x004 * x，（x = 0…3）
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						TMOUT[1:0]		保留		TMDEC[1:0]		保留	TMRST[2:0]		
						rw				rw			rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TMDIS[2:0]			保留	TMEN[2:0]			保留		TMSTOP[1:0]		保留		TMSTART	保留
	rw				rw					rw				rw	

位/位域	名称	描述
31:26	保留	必须保持复位值。
25:24	TMOUT[1:0]	定时器输出 配置定时器输出的初始状态以及是否受定时器复位影响。 00：定时器初始输出为 1，且不受定时器复位影响 01：定时器初始输出为 0，且不受定时器复位影响 10：定时器初始输出为 1，受定时器复位影响 11：定时器初始输出为 0，受定时器复位影响
23:22	保留	必须保持复位值。
21:20	TMDEC[1:0]	定时器递减 配置计数器递减参考时钟和移位时钟源 00：MFCOM 时钟上的计数器递减，移位时钟源为定时器输出 01：触发输入上的计数器递减，移位时钟源为定时器输出 10：计数器引脚输入上递减，移位时钟源为引脚输入 11：计数器触发输入上递减，移位时钟源为触发输入
19	保留	必须保持复位值。
18:16	TMRST[2:0]	定时器复位 配置定时器复位的条件。 000：定时器从不复位 001：保留 010：引脚等于定时器输出时定时器复位 011：触发等于定时器输出时定时器复位 100：定时器引脚上升沿时定时器复位 101：保留 110：上升沿触发时定时器复位 111：上升或下降沿触发时定时器复位 **注意：**在 8 位计数器配置下，定时器复位只会复位计数器低 8 位，其他配置下会复位 16 位。
15	保留	必须保持复位值。
14:12	TMDIS[2:0]	禁能定时器 000：永不禁能 001：在定时器 x-1 禁能时禁能 010：定时器比较事件发生时禁能 011：定时器比较事件发生和触发信号低电平时禁能 100：在引脚的上升或下降沿禁能 101：在引脚的上升或下降沿和提供触发信号高电平时禁能

		110：在触发信号的下降沿禁能 111：保留
11	保留	必须保持复位值。
10:8	TMEN[2:0]	定时器使能 配置启用定时器并开始递减的条件 000：永远使能 001：定时器 x-1 使能时使能 010：触发信号高电平时使能 011：触发信号高电平和引脚高电平时使能 100：引脚上升沿使能 101：引脚上升沿和触发信号高电平时使能 110：触发信号上升沿时使能 111：触发信号上升沿或下降沿时使能
7:6	保留	必须保持复位值。
5:4	TMSTOP[1:0]	定时器停止位 00：禁能停止位 01：在定时器比较事件发生时插入停止位 10：在定时器禁能时插入停止位 11：在定时器比较事件发生且禁能时插入停止位
3:2	保留	必须保持复位值。
1	TMSTART	定时器起始位 0：禁能起始位 1：使能起始位
0	保留	必须保持复位值。

9.5.18. 定时器比较寄存器 x（MFCOM_TMCMPx）

地址偏移：0x500 + 0x004 * x，（x = 0…3）
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
TMCVALUE[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。

15:0	TMCVALUE[15:0]	定时器比较值 当定时器第一次使能时，定时器比较值被加载到定时器计数器中，并且计数器减为零时定时器被重置。 8 位波特率配置： 低 8 位配置波特率分频器 =（TMCVALUE [7:0] + 1）* 2 高 8 位配置每个字的位数 =（TMCVALUE [15:8] + 1）/ 2 8 位高 PWM 配置： 低 8 位将输出的高电平周期配置为（TMCVALUE [7:0] + 1） 高 8 位将输出的低电平周期配置为（TMCVALUE [15:8] + 1） 16 位计数器配置： 波特率分频器 =（TMCVALUE [15:0] + 1）* 2 当移位时钟源为引脚或触发输入时，每个字的位数 =（TMCVALUE [15:0] + 1）/ 2

10. 循环冗余校验计算单元（CRC）

10.1. 简介

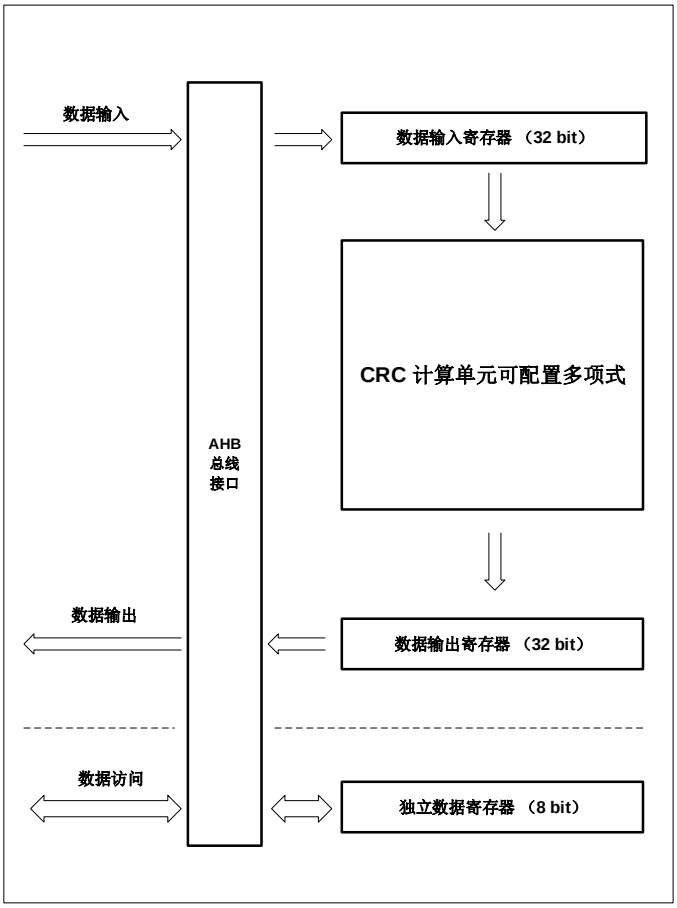
循环冗余校验码是一种用在数字网络和存储设备上的差错校验码，可以校验原始数据的偶然误差。

CRC 计算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。

10.2. 主要特征

- 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 10-1. CRC 计算单元框图



10.3. 功能说明

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

  如果不通过软件设置CRC_CTL寄存器的方式来清除CRC_DATA寄存器，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0x1A2B3C4D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：0x58D43CB2

  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：0xD458B23C

  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：0xB23CD458

  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0x22CC4488将被逆序成0x11223344。

- 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。

10.4. CRC 寄存器

CRC基地址：0x4002 3000

10.4.1. 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。

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

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

位/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。

10.4.2. 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。

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								FDATA[7:0]							
											rw				

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。

10.4.3. 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。

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								REV_O	REV_I[1:0]		PS[1:0]		保留		RST
								rw	rw		rw				rs

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。

10.4.4. 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。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDATA [31:16]															

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。

10.4.5. 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。

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传输方式，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。

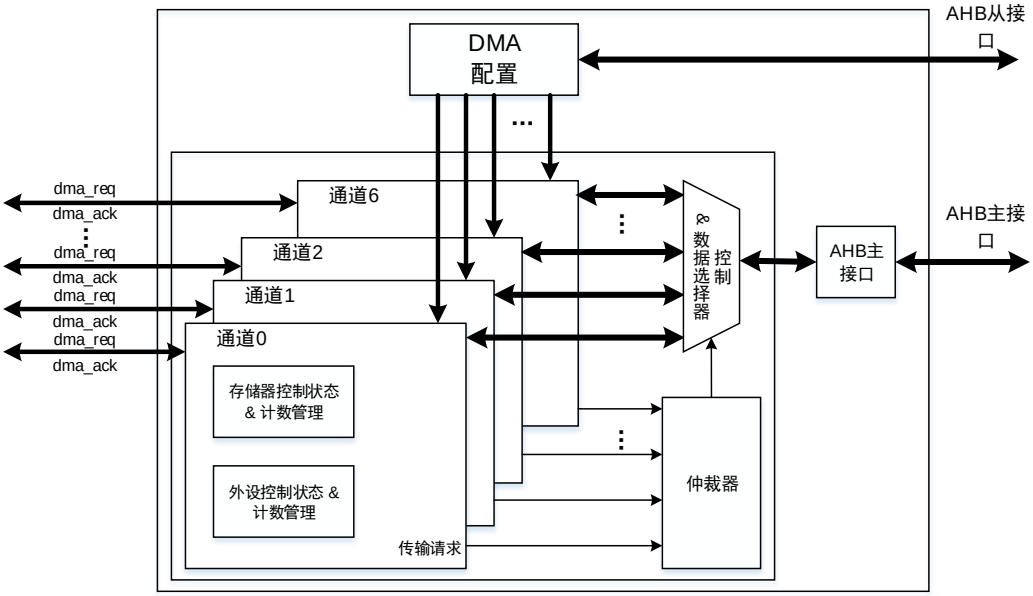
DMA 控制器和 Cortex®-M33 内核共享系统总线。当 DMA 和 CPU 访问同样的地址空间时，DMA 访问可能会阻挡 CPU 访问系统总线几个总线周期。总线矩阵中实现了循环仲裁算法来分配 DMA 与 CPU 的访问权，它可以确保 CPU 得到至少一半的系统总线带宽。

11.2. 主要特征

- 传输数据长度可编程配置，最大到 65536；
- 12 个通道（DMA0 有 7 个通道，DMA1 有 5 个通道），并且每个通道都可配置；
- 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 DMA 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；
- 支持中断使能和清除。

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，用于存储器访问和外设访问；
- 仲裁器进行 DMA 请求的优先级管理；
- 通道管理用于控制数据/地址选择和数据计数。

11.4. 功能说明

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 4: Read BFBEBDBC[31:0] @0xC	1: Write B1B0[15:0] @0x0 2: Write B5B4[15:0] @0x2 3: Write B9B8[15:0] @0x4 4: Write BDBC[15:0] @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，该位域控制传输的次数。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 DMA 传输将分两种情况：
  - 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，对相应通道的 DMA_CHxCNT、DMA_CHxPADDR 或 DMA_CHxMADDR 寄存器进行了操作，则 DMA 将开始一次新的传输。
- 若清零 CHEN 位时，DMA 传输已经完成，之后未对相应通道的 DMA_CHxCNT、DMA_CHxPADDR 或 DMA_CHxMADDR 寄存器进行操作前便使能 DMA 通道，则不会触发任何 DMA 传输。

11.4.2. 外设握手

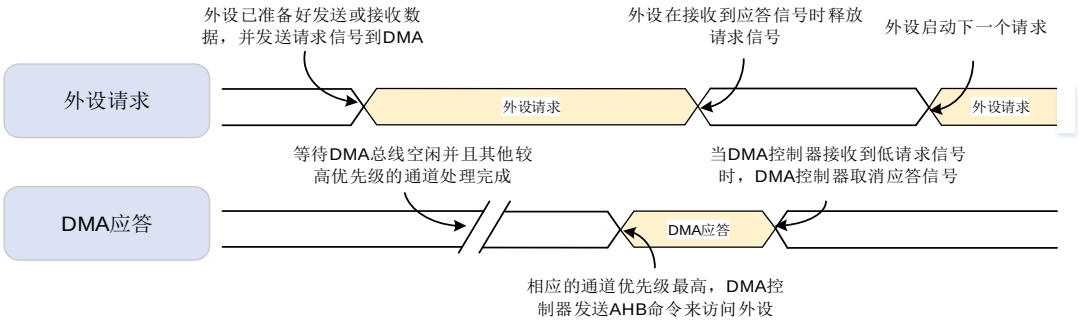
为了保证数据的有效传输，DMA控制器中引入了外设和存储器的握手机制，包括请求信号和应

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位分别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传输才会停止。

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。

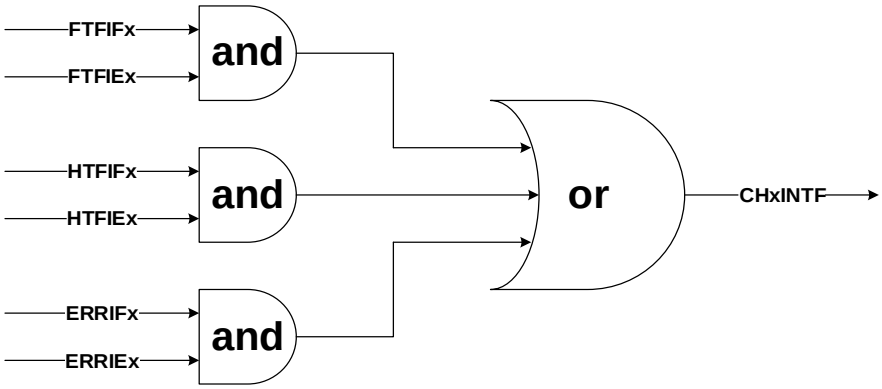
每一个中断事件在DMA_INTF寄存器中有专用的标志位，在DMA_INTC寄存器中有专用的清除位，在DMA_CHxCTL寄存器中有专用的使能位。*表11-2. 中断事件*描述了其对应关系。

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
传输半完成	HTFIF	HTFIFC	HTFIE
传输错误	ERRIF	ERRIFC	ERRIE

DMA中断逻辑如*图11-3. DMA 中断逻辑图*所示，任何类型中断使能时，产生了相应中断事件均会产生中断。

图 11-3. DMA 中断逻辑图



注意：“x”表示通道数（对应x=0…6）

11.4.9. DMA 请求映射

每个 DMA 通道的请求都连接至由 DMAMUX 请求复用器的对应通道输出来转发的 AHB/APB 外设请求，参考*表 12-3. DMAMUX 请求路由输入信号映射*。

11.5. DMA 寄存器

DMA0 基地址：0x4002 0000
DMA1 基地址：0x4002 0400
注意：DMA1 仅有 5 个通道（0 到 4 通道），所有相关寄存器中通道 5 和通道 6 的标志位不适用于 DMA1。

11.5.1. 中断标志位寄存器（DMA_INTF）

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

该寄存器只能按字（32 位）访问。

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

地址偏移：0x08 + 0x14 * x（x = 0...6, x 为通道序号）
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	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）

地址偏移：0x0C + 0x14 * x（x = 0...6, x 为通道序号）
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]	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）

地址偏移：0x10 + 0x14 * x（x = 0...6, x 为通道序号）
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
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）

地址偏移：0x14 + 0x14 * x（x = 0...6, x 为通道序号）
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
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. DMA 请求多路复用器（DMAMUX）

12.1. 简介

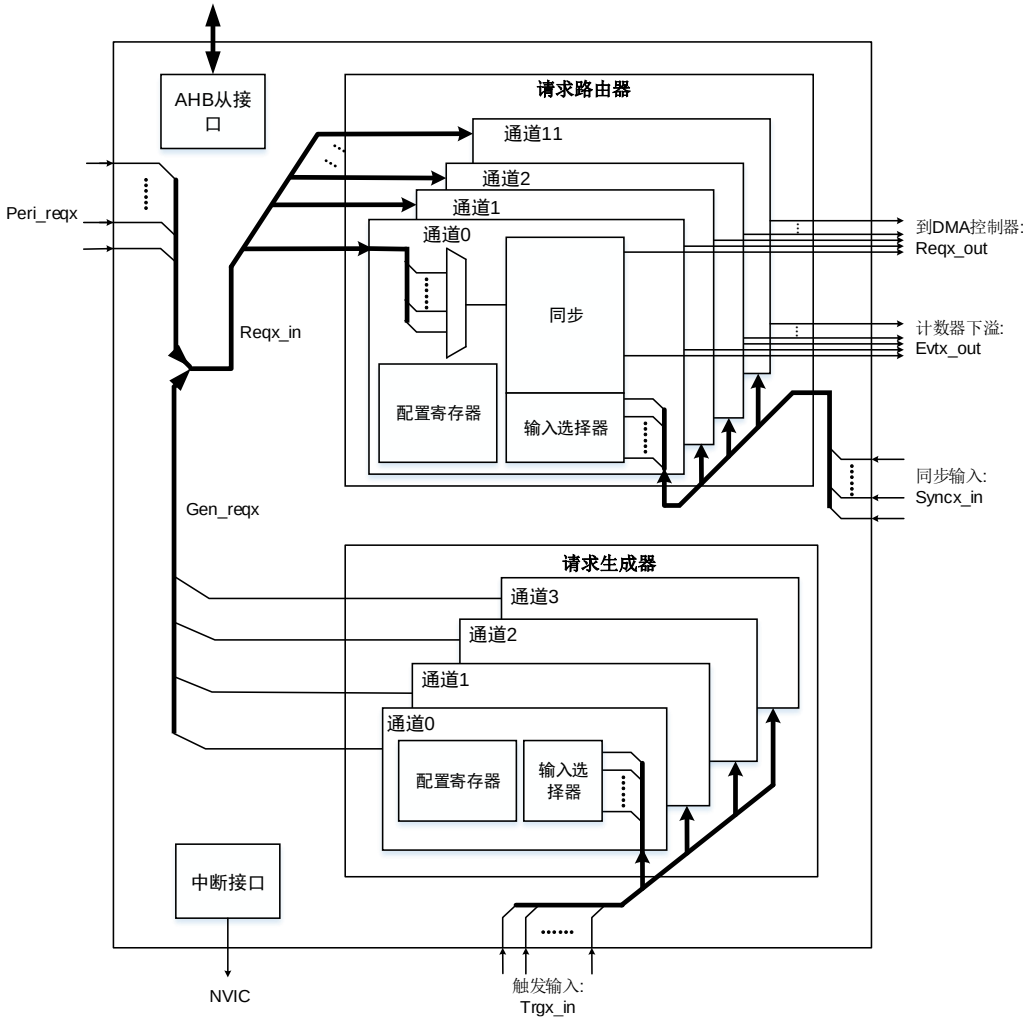
DMAMUX 是 DMA 请求的传输调度器。可编程的 DMA 请求多路复用器 DMAMUX，可在外设和 DMA 控制器之间路由 DMA 请求线路，或者 DMAMUX 也可以将可编程事件连入到输入触发信号上，作为一个 DMAMUX 请求发生器，再由 DMAMUX 请求路由器在 DMAMUX 请求生成器产生的 DMA 请求和 DMA 控制器之间路由 DMA 请求线路。每个 DMAMUX 请求路由通道选择一条唯一的 DMA 请求线路，无条件地或同步地从它的 DMAMUX 同步输入事件。DMA 请求信号会一直挂起，直到 DMA 控制器响应它，并且产生一个 DMA 确认信号，此时相应的 DMA 请求信号被释放。

12.2. 主要特征

- 12 个可配置的 DMAMUX 请求路由输出通道
- 4 个 DMAMUX 请求生成通道
- 27 路触发输入信号到 DMAMUX 请求生成器
- 27 路同步输入信号
- 每个 DMAMUX 请求生成通道：
  - DMAMUX 请求触发输入选择器
  - DMAMUX 请求生成计数器
  - 指示被选中的 DMAMUX 请求触发输入信号的事件溢出标志
- 每个 DMAMUX 请求路由输出通道：
  - 79 路外设 DMAMUX 请求输入信号
  - 同步输入信号选择器
  - 对应一条 DMA 请求路由输出线路
  - 一个路由事件输出信号，用于 DMA 请求级联
  - DMAMUX 请求路由计数器
  - 指示被选中的同步输入信号的事件溢出标志

12.3. 结构框图

图 12-1. DMAMUX 结构框图



12.4. 功能说明

如*图 11-1. DMA 结构框图*所示，DMAMUX 包含两个子模块：

- DMAMUX 请求路由器

  DMAMUX 请求路由器输入（Reqx_in）来自两部分：
  - 一部分来自外设请求（Peri_reqx）；
  - 另一部分来自 DMAMUX 请求生成器（Gen_reqx）。

  DMAMUX 请求路由输出到 DMA 控制器对应的通道（Reqx_out）。

  同步输入（Syncx_in）来自内部或外部信号。

- DMAMUX 请求生成器

  DMAMUX 请求触发输入（Trgx_in）来自内部或外部信号。

12.4.1. DMAMUX 信号

表 12-1. DMAMUX 信号

信号名称	描述
Reqx_in	DMAMUX 请求路由输入信号（来自外设的请求或者 DMAMUX 请求生成器生成的请求）
Peri_reqx	从外设输入到 DMAMUX 的 DMA 请求线路
Gen_reqx	DMAMUX 请求生成器生成输出的 DMA 请求信号
Reqx_out	DMAMUX 请求输出信号（到 DMA 控制器）
Trgx_in	DMAMUX 请求触发输入信号（到 DMAMUX 请求生成器）
Syncx_in	DMAMUX 同步输入信号（到 DMAMUX 请求路由器）
Evtx_out	DMAMUX 请求路由计数器下溢事件输出信号

12.4.2. DMAMUX 请求路由器

DMAMUX 请求路由器可在外设或者 DMAMUX 请求生成器，与 DMA 控制器之间路由 DMA 请求线路。DMAMUX 请求路由器由 DMAMUX 请求路由通道组成。DMA 请求输入信号并联至所有的 DMAMUX 请求路由通道。每个 DMAMUX 请求路由通道都有一个同步单元。同步输入信号并联至所有 DMAMUX 请求路由通道的同步单元。每个 DMAMUX 请求路由通道都有一个内部的 DMAMUX 请求路由计数器。

DMAMUX 请求路由通道

DMAMUX 请求路由通道 x 的请求路由输入由 DMAMUX_RM_CHxCFG 寄存器的 MUXID[6:0]位域来配置，请求路由输入可选为外设 DMA 请求，或者 DMAMUX 请求生成器产生的 DMA 请求，参考*表 12-3. DMAMUX 请求路由输入信号映射*。一个 DMAMUX 请求路由通道与对应的 DMA 控制器通道相连接。

注意： 当 MUXID[6:0]值为 0 时，没有 DMA 请求线路被映射到 DMAMUX 请求路由通道上。DMAMUX 不允许将同一个 DMA 请求线路（相同 MUXID[6:0]且非空）映射到两个不同的 DMAMUX 请求路由通道上。

当同步模式禁能时

每当连到 DMAMUX 的 DMA 请求被 DMA 控制器服务，这个 DMA 请求将取消挂起，内部的 DMAMUX 请求路由计数器将减 1。当 DMAMUX 请求路由计数器发生下溢时，DMAMUX_RM_CHxCFG 寄存器的 NBR[4:0]值将自动重装载到计数器中。如果将 EVGEN 位置位，使能通道事件输出，则通道事件输出前，DMA 请求数量为 NBR[4:0] + 1。

注意：只有当 DMAMUX 请求路由通道 x 的同步使能位 SYNCEN 位和通道事件输出使能位 EVGEN 位都为 0 时，才能配置其 NBR[4:0]位域。

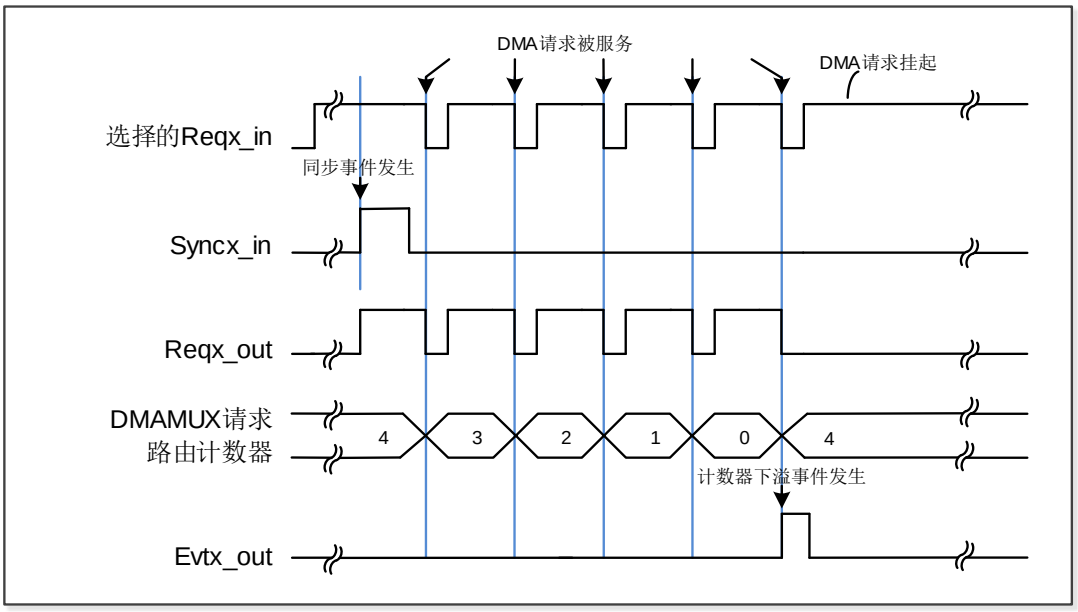
当同步模式使能时

如果 DMAMUX 请求路由通道 x 工作在同步模式下，当检测到选择的同步输入信号的上升沿或者下降沿时，挂起的 DMA 请求将被连到 DMAMUX 请求路由通道 x 的输出。每当连到 DMAMUX 的 DMA 请求被 DMA 控制器服务，这个 DMA 请求将取消挂起，内部的 DMAMUX 请求路由计

数器将减 1。当 DMAMUX 请求路由计数器发生下溢时，DMA 请求线路将断开与 DMAMUX 请求路由通道 x 的输出的连接，并且 DMAMUX_RM_CHxCFG 寄存器的 NBR[4:0]值将自动重装载到计数器中。一个同步事件可传输 NBR[4:0] + 1 个 DMA 请求到 DMAMUX 请求路由通道 x 的输出上。

*图 12-2. 同步模式*为当 NBR[4:0]=4，SYNCEN=1，EVGEN=1，SYNCP[1:0]=0b01 时的举例。

图 12-2. 同步模式



置位 DMAMUX_RM_CHxCFG 寄存器的 SYNCEN 位可使能 DMAMUX 请求路由通道 x 的同步模式。同步输入信号可由 DMAMUX_RM_CHxCFG 寄存器的 SYNCID[4:0]位域来配置，参考*表 12-5. 同步输入信号映射*。同步输入信号的有效边沿由 DMAMUX_RM_CHxCFG 寄存器的 SYNCP[1:0]位域来配置。

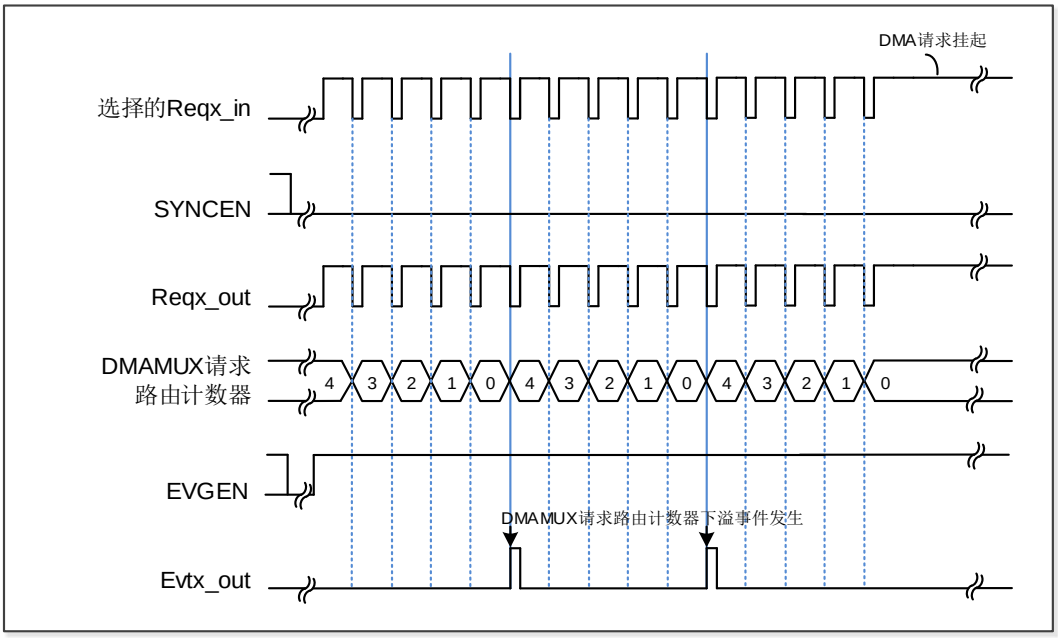
注意：如果同步输入事件发生时，DMAMUX 输入上没有挂起的 DMA 请求，则这个同步输入事件将被忽略，之后如有 DMA 请求被挂起，它将不会被连接到 DMAMUX 请求路由通道 x 的输出，直到发生下一个同步输入事件。

通道事件输出

每个 DMAMUX 请求路由通道都有一个通道事件输出信号 Evtx_out，用于 DMAMUX 请求路由计数器的下溢事件输出。Evt0_out ~ Evt3_out 信号可用于 DMA 请求级联。如果通过置位 DMAMUX_RM_CHxCFG 寄存器的 EVGEN 位来使能 DMAMUX 请求路由通道 x 的通道事件输出，当 DMAMUX 请求路由计数器自动重装载为 NBR[4:0]值时，发生一个通道事件，输出为一个 AHB 时钟周期脉冲。

*图 12-3. 通道事件输出*为当 NBR[4:0]=4，SYNCEN=0，EVGEN=1 时的举例。

图 12-3. 通道事件输出



注意：如果 EVGEN = 1 且 NBR[4:0] = 0，则每次 DMA 请求被服务时都会输出一个通道事件。

同步溢出

如果在 DMAMUX 请求路由计数器下溢之前又发生了新的同步事件，则 DMAMUX_RM_INTF 寄存器的同步溢出标志位 SOIFx 位将置位。

注意：建议在 DMA 控制器对应通道请求被取消时，配置 DMAMUX_RM_CHxCFG 寄存器的 SYNCEN 位为 0 来禁能 DMAMUX 请求路由通道 x 的同步模式。否则，当又发生一个新的同步事件时，由于接收不到 DMA 的响应信号将会发生同步溢出事件。

12.4.3. DMAMUX 请求生成器

DMAMUX 请求生成器在触发输入事件发生时会产生 DMA 请求。DMAMUX 请求生成器由 DMAMUX 请求生成通道组成。DMA 请求触发输入信号并联至所有 DMAMUX 请求生成通道。每个 DMAMUX 请求生成通道都有一个内部的 DMAMUX 请求生成计数器。

触发输入信号的有效边沿由 DMAMUX_RG_CHxCFG 寄存器的 RGTP[1:0]位域来配置。DMAMUX 请求生成通道 x 的触发输入信号由 DMAMUX_RG_CHxCFG 寄存器的 TID[4:0]位域来配置，参考*表 12-4. 触发输入信号映射*。置位 DMAMUX_RG_CHxCFG 寄存器的 RGEN 位来使能 DMAMUX 请求生成通道 x。

DMAMUX 请求生成通道

当发生触发输入事件时，对应的 DMAMUX 请求生成通道 x 开始产生 DMA 请求到通道的输出上，通道输出连到 DMAMUX 请求路由器的输入上。每当 DMAMUX 生成的 DMA 请求被 DMA 控制器服务，这个 DMA 请求将取消挂起，内部的 DMAMUX 请求生成计数器将减 1。当 DMAMUX 请求生成计数器发生下溢时，DMAMUX 请求生成通道将停止产生 DMA 请求，在下

一个触发输入事件发生时，DMAMUX 请求生成计数器将自动重装载为 DMAMUX_RG_CHxCFG 寄存器的 NBRG[4:0]位域值。

注意： 触发输入事件后产生的 DMA 请求数量为 NBRG[4:0] + 1。只有当 DMAMUX 请求生成通道 x 的 RGEN 位为 0 时才可以配置 NBRG[4:0]位域。

触发溢出

如果 RGEN 位为 1，DMAMUX 请求生成通道 x 被使能，当一个新的触发输入信号发生了，而此时 DMAMUX 请求生成计数器还未发生下溢，则 DMAMUX_RG_INTF 寄存器的 TOIFx 位将硬件置位以指示发生了触发溢出事件。

注意：建议在 DMA 控制器对应通道请求被取消时，配置 DMAMUX_RG_CHxCFG 寄存器的 RGEN 位为 0 来禁能 DMAMUX 请求生成通道 x。否则，当又发生一个新的触发输入事件时，由于接收不到 DMA 的响应信号将会发生触发溢出事件。

12.4.4. 通道配置

根据以下步骤来配置 DMAMUX 的通道 y 和对应的 DMA 通道 x：

1. 完整配置 DMA 通道 x 相关参数，除了 DMA 通道 x 的使能。
2. 完整配置 DMAMUX 通道 y 相关参数。
3. 设置 DMA_CHxCTL 寄存器的 CHEN 位 1 来使能 DMA 通道 x。

12.4.5. 中断

DMAMUX模块有两种类型的中断事件，包括DMAMUX请求路由通道的同步溢出事件，和DMAMUX请求生成通道的触发溢出事件。

每个中断事件都有一个专用的标志位，专用的清除位和专用的使能位。*表12-2. 中断事件*描述了其对应关系。

表 12-2. 中断事件

中断事件	标志位	清除位	使能位
DMAMUX 请求路由通道 x 上的同步溢出事件	SOIFx	SOIFCx	SOIE
DMAMUX 请求生成通道 y 上的触发溢出事件	TOIFy	TOIFCy	TOIE

触发溢出中断

当 DMAMUX 请求生成触发溢出标志位 TOIFx 置位，并且触发溢出中断使能位 TOIE 位置位，则会产生一个触发溢出中断。写 1 到 DMAMUX_RG_INTC 寄存器的对应触发溢出清除位 TOIFCx 将会清除触发溢出标志位 TOIFx。

同步溢出中断

当 DMAMUX 请求路由同步溢出标志位 SOIFx 置位，并且触发同步溢出中断使能位 SOIE 位置位，则会产生一个同步溢出中断。写 1 到 DMAMUX_RM_INTC 寄存器的对应同步溢出清除位 SOIFCx 将会清除同步溢出标志位 SOIFx。

12.4.6. DMAMUX 映射

DMAMUX 请求路由输入映射

DMAMUX 请求路由输入可来自于外设或者 DMAMUX 请求生成器，参考*表 12-3. DMAMUX 请求路由输入信号映射*，由 DMAMUX_RM_CHxCFG 寄存器的 MUXID[5:0]位域配置 DMAMUX 请求路由通道 x 的输入。

表 12-3. DMAMUX 请求路由输入信号映射

请求路由通道输入标识 MUXID[5:0]	来源
1	Gen_reqx0
2	Gen_reqx1
3	Gen_reqx2
4	Gen_reqx3
5	ADC
6	DAC_CH0
7	保留
8	I2C1_RX
9	I2C1_TX
10	I2C0_RX
11	I2C0_TX
12	MFCOM_SSTAT0
13	MFCOM_SSTAT1
14	MFCOM_SSTAT2
15	MFCOM_SSTAT3
16	SPI0_RX
17	SPI0_TX
18	SPI1_RX
19	SPI1_TX
20	TIMER0_CH0
21	TIMER0_CH1
22	TIMER0_CH2
23	TIMER0_CH3
24	TIMER0_TI
25	TIMER0_UP
26	TIMER0_CO
27	TIMER0_MCH0
28	TIMER0_MCH1
29	TIMER0_MCH2

请求路由通道输入标识 MUXID[5:0]	来源
30	TIMER0_MCH3
31	TIMER1_CH0
32	TIMER1_CH1
33	TIMER1_CH2
34	TIMER1_CH3
35	TIMER1_TI
36	TIMER1_UP
37	TIMER7_CH0
38	TIMER7_CH1
39	TIMER7_CH2
40	TIMER7_CH3
41	TIMER7_TI
42	TIMER7_UP
43	TIMER7_CO
44	TIMER7_MCH0
45	TIMER7_MCH1
46	TIMER7_MCH2
47	TIMER7_MCH3
48	CAN1
49	CAN0
50	USART0_RX
51	USART0_TX
52	USART1_RX
53	USART1_TX
54	USART2_RX
55	USART2_TX
56	TIMER5_UP
57	TIMER6_UP
58	TIMER19_CH0
59	TIMER19_CH1
60	TIMER19_CH2
61	TIMER19_CH3
62	TIMER19_TI
63	TIMER19_UP
64	TIMER19_CO
65	TIMER19_MCH0
66	TIMER19_MCH1
67	TIMER19_MCH2
68	TIMER19_MCH3
69	TIMER20_CH0
70	TIMER20_CH1

请求路由通道输入标识 MUXID[5:0]	来源
71	TIMER20_CH2
72	TIMER20_CH3
73	TIMER20_TI
74	TIMER20_UP
75	TIMER20_CO
76	TIMER20_MCH0
77	TIMER20_MCH1
78	TIMER20_MCH2
79	TIMER20_MCH3

触发输入映射

DMAMUX 请求生成通道 x 的触发输入可由 DMAMUX_RG_CHxCFG 寄存器的 TID[4:0]位域来配置，参考*表 12-4. 触发输入信号映射*。

表 12-4. 触发输入信号映射

触发输入标识TID[4:0]	来源
0	EXTI_0
1	EXTI_1
2	EXTI_2
3	EXTI_3
4	EXTI_4
5	EXTI_5
6	EXTI_6
7	EXTI_7
8	EXTI_8
9	EXTI_9
10	EXTI_10
11	EXTI_11
12	EXTI_12
13	EXTI_13
14	EXTI_14
15	EXTI_15
16	Evtx_out0
17	Evtx_out1
18	Evtx_out2
19	Evtx_out3
20	保留
21	保留
22	TIMER20_CH0_O
23	保留
24	保留

触发输入标识TID[4:0]	来源
25	保留
26	保留

同步输入映射

同步输入由 DMAMUX_RM_CHxCFG 寄存器的 SYNCID[4:0]位域来配置，参考*表 12-5. 同步输入信号映射*。

表 12-5. 同步输入信号映射

同步输入标识SYNCID[4:0]	来源
0	EXTI_0
1	EXTI_1
2	EXTI_2
3	EXTI_3
4	EXTI_4
5	EXTI_5
6	EXTI_6
7	EXTI_7
8	EXTI_8
9	EXTI_9
10	EXTI_10
11	EXTI_11
12	EXTI_12
13	EXTI_13
14	EXTI_14
15	EXTI_15
16	Evtx_out0
17	Evtx_out1
18	Evtx_out2
19	Evtx_out3
20	保留
21	保留
22	TIMER20_CH0_O
23	保留
24	保留
25	保留
26	保留

12.5. DMAMUX 寄存器

DMAMUX基地址：0x4002 0800

12.5.1. 请求路由通道 x 配置寄存器（DMAMUX_RM_CHxCFG）

地址偏移：0x00 + 0x04 * x（x = 0...11，其中 x 为通道序号）
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			SYNCID[4:0]					NBR[4:0]					SYNCP[1:0]		SYNCEN
					rw					rw			rw		rw

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

位/位域	名称	描述
31:29	保留	必须保持复位值。
28:24	SYNCID[4:0]	同步输入标识 选择同步输入源。
23:19	NBR[4:0]	传递的DMA请求数量 在同步输入事件之后，或者通道事件输出之前，将传递到DMA控制器的DMA请求数量为NBR[4:0] + 1。 该位域只能在SYNCEN位和EVGEN位都禁能时才能配置。
18:17	SYNCP[1:0]	同步输入极性 00：不检测事件 01：上升沿 10：下降沿 11：上升和下降沿
16	SYNCEN	同步模式使能 0：禁能同步模式 1：使能同步模式
15:10	保留	必须保持复位值。
9	EVGEN	事件输出使能 0：禁能事件输出 1：使能事件输出
8	SOIE	同步溢出中断使能 0：禁能中断 1：使能中断

7	保留	必须保持复位值。
6:0	MUXID[6:0]	请求路由标识 选择DMAMUX请求路由通道的DMA请求输入源。

12.5.2. 请求路由通道中断标志位寄存器（DMAMUX_RM_INTF）

地址偏移：0x80
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				SOIF11	SOIF10	SOIF9	SOIF8	SOIF7	SOIF6	SOIF5	SOIF4	SOIF3	SOIF2	SOIF1	SOIF0
				r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:12	保留	必须保持复位值。
11:0	SOIFx	请求路由通道x（x=0..11）的同步溢出事件标志位 当DMAMUX请求路由通道x发生了同步输入事件，而此时DMAMUX请求路由计数器值小于NBR[4:0]，则该通道的同步溢出标志位置位。 通过对DMAMUX_RM_INTC寄存器的SOIFCx位写1来清除相应通道的同步溢出标志。

12.5.3. 请求路由通道中断标志位清除寄存器（DMAMUX_RM_INTC）

地址偏移：0x084
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				SOIFC11	SOIFC10	SOIFC9	SOIFC8	SOIFC7	SOIFC6	SOIFC5	SOIFC4	SOIFC3	SOIFC2	SOIFC1	SOIFC0
				w	w	w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:12	保留	必须保持复位值。
11:0	SOIFCx	请求路由通道x（x=0..11）的同步溢出事件标志清除位 写1可清除相应通道在DMAMUX_RM_INTF寄存器的同步溢出标志SOIFx。

12.5.4. 请求生成通道 x 配置寄存器（DMAMUX_RG_CHxCFG）

地址偏移：0x100 + 0x04 * x（x = 0...3，其中 x 为通道序号）
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								NBRG[4:0]					RGTP[1:0]		RGEN
										rw			rw		rw

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

位/位域	名称	描述
31:24	保留	必须保持复位值。
23:19	NBRG[4:0]	待产生的DMA请求数量 在触发输入事件之后，待产生的DMA请求数量为NBRG[4:0] + 1。 **注意：**只有当RGEN位为0时才能写该位域。
18:17	RGTP[1:0]	DMAMUX请求生成触发输入极性 00：不检测事件 01：上升沿 10：下降沿 11：上升沿和下降沿
16	RGEN	DMAMUX请求生成通道x使能 0：禁能DMAMUX请求生成通道x 1：使能DMAMUX请求生成通道x
15:9	保留	必须保持复位值。
8	TOIE	触发溢出中断使能 0：禁能中断 1：使能中断
7:5	保留	必须保持复位值。
4:0	TID[4:0]	触发输入标识 选择DMAMUX请求生成通道的触发输入源。

12.5.5. 请求生成通道中断标志位寄存器（DMAMUX_RG_INTF）

地址偏移：0x140
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												TOIF3	TOIF2	TOIF1	TOIF0
												r	r	r	r

位/位域	名称	描述
31:4	保留	必须保持复位值。
3:0	TOIFx	DMAMUX请求生成通道x（x=0..3）的触发溢出标志位 如果在DMAMUX请求生成计数器（通过DMAMUX_RG_CHxCFG寄存的NBRG[4:0]位域配置）发生下溢之前，DMAMUX请求生成通道x发生了一个新的触发输入事件，则该标志位置位。 通过对DMAMUX_RG_INTC寄存器的TOIFCx位写1来清除相应通道的触发溢出标志。

12.5.6. 请求生成通道中断标志位清除寄存器（DMAMUX_RG_INTC）

地址偏移：0x144

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												TOIFC3	TOIFC2	TOIFC1	TOIFC0
												w	w	w	w

位/位域	名称	描述
31:4	保留	必须保持复位值。
3:0	TOIFCx	请求生成通道x（x=0..3）的触发溢出事件标志清除位 写1可清除相应通道在DMAMUX_RG_INTF寄存器的触发溢出标志TOIFx。

13. 调试（DBG）

13.1. 简介

GD32A50x 系列产品提供了各种各样的调试和测试功能。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。调试功能集成在 Arm® Cortex®-M33 内核中。调试系统支持串行（SW）调试功能，也支持 JTAG 调试。调试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、CAN 和 MFCOM。

13.2. JTAG/SW 功能说明

调试工具可以通过串行（SW）调试接口或者 JTAG 调试接口来访问调试功能。

13.2.1. 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。

13.2.2. 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）。

表 13-1. 调试引脚分配

引脚	调试接口
PB7	JTDI

引脚	调试接口
PB8	JTCK/SWCLK
PB9	JTMS/SWDIO
PB3	NJTRST
PB4	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 PB3 可以用作普通 GPIO 功能（NJTRST 硬件拉高）。如果切换到 SW 调试模式，PB7/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*通用和备用输入/输出接口（GPIO 和 AFIO）*。

13.2.3. JTAG 链状结构

Cortex®-M33 内核的 JTAG TAP 和边界扫描（BSD）TAP 串行连接。边界扫描（BSD）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 BSD 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 BSD JTAG 已处在 BYPASS 模式。

BSD JTAG ID 代码是 0x06418041。

13.2.4. 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。

13.2.5. 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。

13.3. 调试保持功能说明

13.3.1. 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。

13.3.2. 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；

对于 MFCOM 外设，计数器时钟停止并进行调试。

13.4. DBG 寄存器

DEBUG 基地址：0xE004 4000

13.4.1. 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。

13.4.2. 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
TIMER19_HOLD	TIMER20_HOLD	保留						CAN1_HOLD	CAN0_HOLD	MFCOM_HOLD	TIMER6_HOLD	TIMER5_HOLD	保留	TIMER7_HOLD	I2C1_HOLD
rw	rw							rw	rw	rw	rw	rw		rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C0_HOLD	保留.			TIMER1_HOLD	TIMER0_HOLD	WWDGT_HOLD	FWDGT_HOLD	保留					STB_HOLD	DSLP_HOLD	SLP_HOLD
rw				rw	rw	rw	rw	rw		rw			rw	rw	rw

位/位域	名称	描述
31	TIMER19_HOLD	TIMER19 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 19 计数器不变，用于调试。
30	TIMER20_HOLD	TIMER20 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 20 计数器不变，用于调试。
29:24	保留	必须保持复位值
23	CAN1_HOLD	CAN1 保持寄存器 该位由软件置位和复位。 0：无影响 1：当内核停止时保持 CAN1 计数器不变，用于调试。
22	CAN0_HOLD	CAN0 保持寄存器 该位由软件置位和复位。 0：无影响 1：当内核停止时保持 CAN0 计数器不变，用于调试。
21	MFCOM_HOLD	MFCOM 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 MFCOM 计数器不变，用于调试。
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	保留	必须保持复位值
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。
14:12	保留	必须保持复位值
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。
7:3	保留	必须保持复位值
2	STB_HOLD	待机模式保持寄存器 该位由软件置位和复位。 0：无影响 1：在待机模式下，系统时钟和 AHB 时钟由 CK_IRC8M 提供（由软件配置），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。

14. 模数转换器（ADC）

14.1. 简介

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

对于电机、电源等对ADC有更高需求的应用，可以联系我们的销售，获取更多的ADC详细资料。

14.2. 主要特征

- 高性能：
  - ADC采样分辨率：12位、10位、8位、或者6位分辨率；
  - 前置校准功能；
  - 可编程采样时间；
  - 数据存储模式：最高有效位对齐和最低有效位对齐；
  - 支持DMA请求。
- 模拟输入通道：
  - 16个外部模拟输入通道；
  - 1个内部温度传感器通道(V_{SENSE})；
  - 1个内部参考电压输入通道(V_{REFINT})。
- 转换开始的发起：
  - 软件；
  - TRIGSEL。
- 运行模式：
  - 转换单个通道，或者扫描一序列的通道；
  - 单次运行模式，每次触发转换一次选择的输入通道；
  - 连续运行模式，连续转换所选择的输入通道；
  - 间断运行模式；
  - 同步模式（适用于具有两个或多个ADC的设备）。
- 转换结果阈值监测功能：模拟看门狗。
- 中断产生：
  - 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。
- ADC输入范围：$V_{REFN} \leq V_{IN} \leq V_{REFP}$。

14.3. 引脚和内部信号

*图14-1. ADC模块框图*给出了ADC模块框图。*表14-1. ADC内部信号*给出了ADC内部信号。*表14-2. ADC引脚定义*给出了ADC引脚说明。

表 14-1. ADC 内部信号

内部信号名称	说明
V_{SENSE}	内部温度传感器输出电压
V_{REFINT}	内部参考输出电压

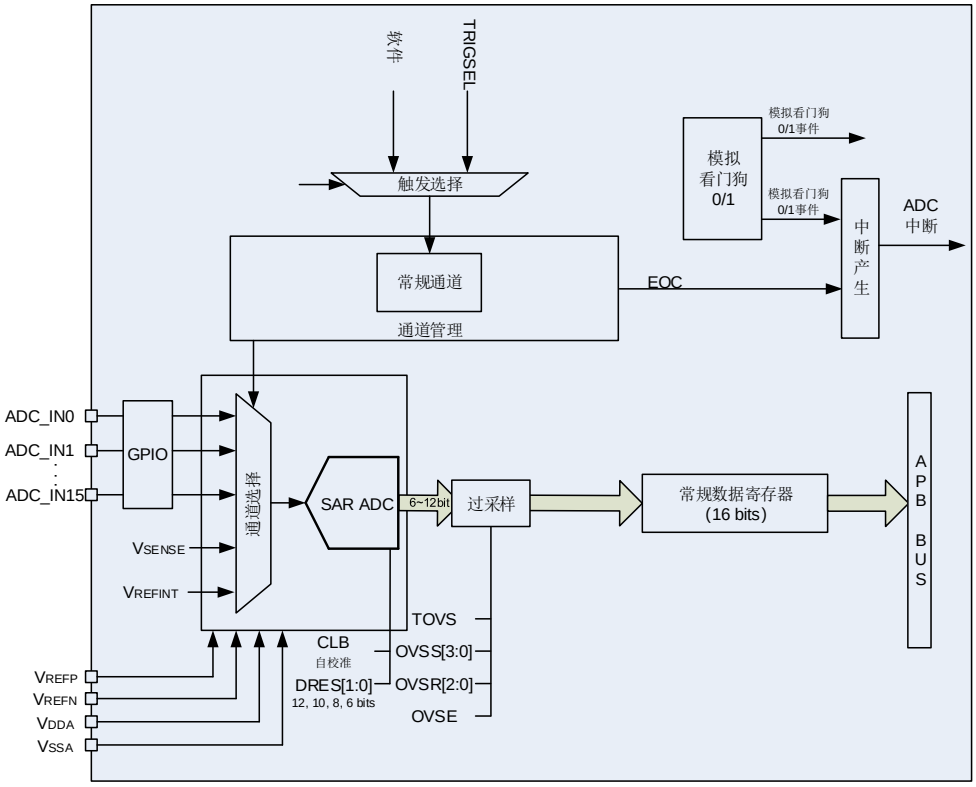
表 14-2. ADC 引脚定义

名称	注释
V_{DDA}	模拟电源输入，等于V_{DD}
V_{SSA}	模拟地，等于V_{SS}
V_{REFP}	ADC正参考电压
V_{REFN}	ADC负参考电压
ADCx_IN[15:0]	多达16路外部通道

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

14.4. 功能说明

图 14-1. ADC 模块框图



14.4.1. 前置校准功能

ADC带有一个前置校准功能。在校准期间，ADC计算一个校准系数，这个系数是应用于ADC内部的，它直到ADC下次掉电才无效。在校准期间，应用不能使用ADC，必须等到校准完成。在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，直到校准完成，该位由硬件清0。

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

内部的模拟校准可以通过设置ADC_CTL1寄存器的RSTCLB位来重置。

软件校准过程：

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

14.4.2. ADC 时钟

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

14.4.3. ADC 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}数值详见芯片相关型号Datasheet。

14.4.4. 常规序列

通道管理电路把采样通道组织成一个序列：常规序列。

常规序列支持高达 16 个通道，每个通道称为常规通道。ADC_RSQ0 寄存器的 RL[3:0]位规定了整个常规序列的长度。ADC_RSQ0~ADC_RSQ2 寄存器规定了常规序列的通道选择。

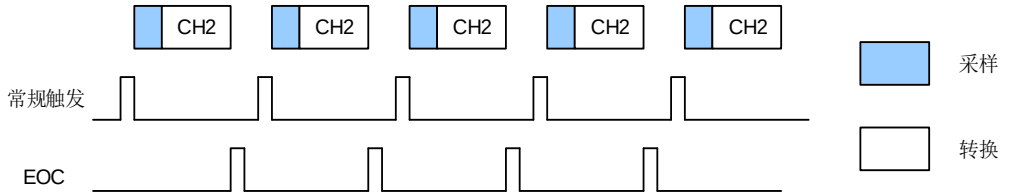
注意：尽管 ADC 支持 18 个通道，但常规序列一次最多转换 16 个通道。

14.4.5. 运行模式

单次运行模式

该模式能够运行在常规序列。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 14-2. 单次运行模式



常规序列的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。

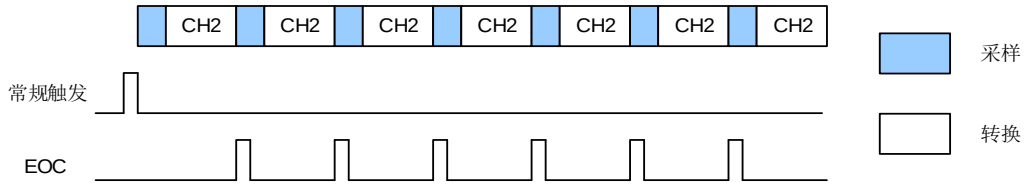
常规序列单次转换模式的软件流程：

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. 从ADC_RDATA寄存器中读ADC转换结果；
8. 写0清除EOC标志位。

连续运行模式

该模式能够运行在常规序列。将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 14-3. 连续转换模式



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. 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。

可以使用DMA来传输转换数据，不需循环查询EOC标志位，软件流程如下：

1. 设置ADC_CTL1寄存器的CTN位为1；

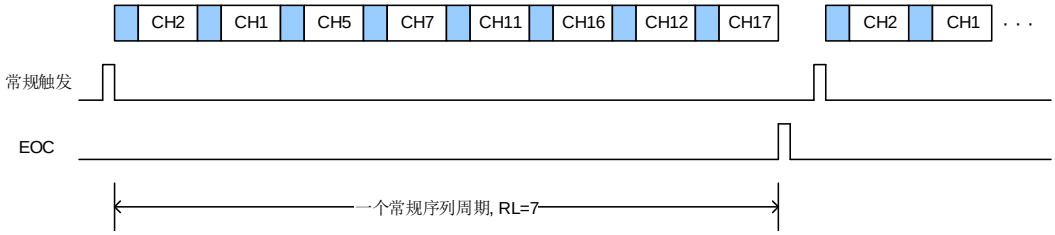
2. 根据模拟通道编号配置RSQ0；
3. 配置ADC_SAMPTx寄存器；
4. 如果有需要，配置ADC_CTL1寄存器的ETERC和ETSRC位；
5. 准备DMA模块，用于传输来自ADC_RDATA的数据；
6. 设置SWRCST位，或者给常规序列产生一个外部触发。

扫描运行模式

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

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

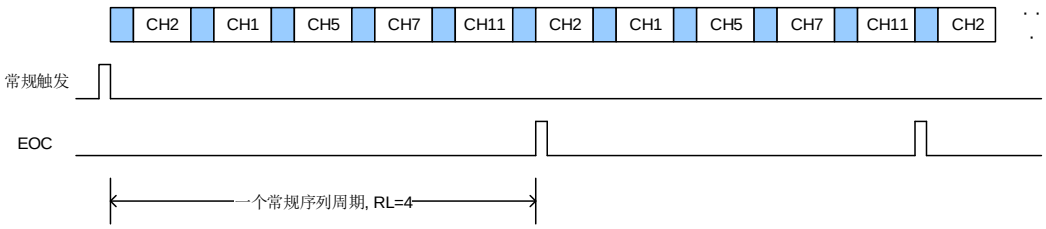
图 14-4. 扫描运行模式，且连续转换模式禁能



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 14-5. 扫描转换模式，连续转换模式使能

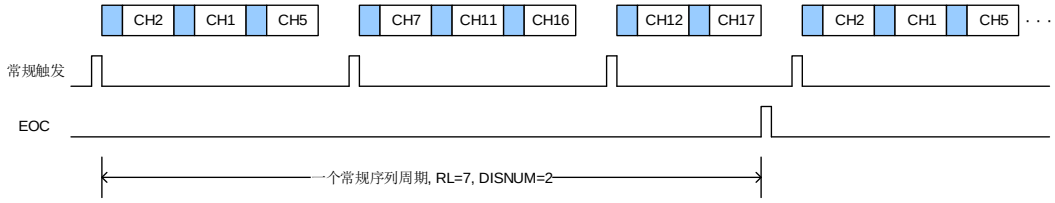


间断运行模式

对于常规序列，当 ADC_CTL0 寄存器的 DISRC 位置 1 时，常规序列间断转换模式使能。该模式下可以执行一次 n 个通道的短序列转换(n<=8)，这个短序列是 ADC_RSQ0~RSQ2 寄存器所选择的转换序列的一部分。数值 n 由 ADC_CTL0 寄存器的 DISCNUM[2:0]位给出。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 寄存器所选择通道

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 位将置 1。如果 EOCIE 位置 1，则将产生一个中断。

图 14-6. 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。

14.4.6. 转换结果阈值监测功能

模拟看门狗 0

ADC_CTL0 寄存器的 RWD0EN 位置 1 将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 位将被置 1。如果 WDE0IE 位被置 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

模拟看门狗 1 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。模拟看门狗 1 的高/低阈值可在 ADC_WDT1 寄存器中进行配置。

14.4.7. 数据存储模式

ADC_CTL1 寄存器的 DAL 位确定转换后数据存储的对齐方式。

图 14-7. 12 位数据存储模式

常规序列数据

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

DAL=0

常规序列数据

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

DAL=1

图 14-8. 10 位数据存储模式

常规序列数据

0	0	0	0	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0

DAL=0

常规序列数据

D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0

DAL=1

图 14-9. 8 位数据存储模式

常规序列数据

0	0	0	0	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0

DAL=0

常规序列数据

D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0	0	0

DAL=1

6 位分辨率的数据存储模式不同于 12 位/10 位/8 位分辨率数据存储模式，如*图 14-10. 6 位数据存储模式*所示。

图 14-10. 6 位数据存储模式

常规序列数据

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

DAL=0

常规序列数据

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

DAL=1

14.4.8. 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]位更改。每个通道可以用不同的时间采样。在 12 位分辨率的情况下，总采样转换时间=采样时间+12.5 个 CK_ADC 周期。

例如：

CK_ADC = 15MHz，采样时间为 2.5 个周期，那么总的转换时间为：“2.5+12.5”个 CK_ADC 周期，即 1us。

14.4.9. 外部触发

TRIGSEL 或者 SWRCST 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 位控制。

表 14-3. 常规序列外部触发源

ETSRC	触发源	触发类型
0	TRIGSEL	来自TRIGSEL的信号
1	SWRCST	软件触发

14.4.10. DMA 请求

DMA 请求，可以通过设置 ADC_CTL1 寄存器的 DMA 位来使能，它用于传输常规序列多个通道的转换结果。ADC 在常规序列的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。

注意：ADC1 无 DMA 请求。

14.4.11. ADC 内部通道

将 ADC_CTL1 寄存器的 TSVEN 位置 1，可以使能温度传感器通道（ADC_IN16）。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 t_{s_temp} μs（具体数值请参考 Datasheet）。温度传感器不用时，复位 TSVEN 位可以将其置于掉电模式。

温度传感器的输出电压随温度线性变化，由于生产过程的多样化，温度变化曲线的偏移在不同的芯片上会有不同（具体数值请参考 Datasheet）。

使用温度传感器：

1. 配置 ADC 时钟（不超过 5MHz）；
2. 配置温度传感器通道（ADC0_IN16）的转换序列和采样时间大于 t_{s_temp} μs；
3. 置位 ADC_CTL1 寄存器中的 TSVEN 位，使能温度传感器；
4. 置位 ADC_CTL1 寄存器的 ADCON 位，或者由外部触发 ADC 转换；
5. 读取内部温度传感器输出电压 $V_{temperature}$，并由下面公式计算出实际温度：

$$温度\ (°C) = \frac{V_{temperature} - V_{30}}{Avg_Slope} + 30 \tag{14-1}$$

$V_{temperature}$：温度传感器的输出电压。

V_{30}：内部温度传感器在 30°C 时的输出电压，典型值请参考相关型号 Datasheet。。芯片出厂时记录了温度传感器在 30°C 下对应的 ADC 转换结果（VDDA 为 5.0V 时）。这个出厂校准值存储在 FLASH 中的只读区域，具体存储地址请见数据手册。

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

注意：

1）温度传感器使能后，需等待至少 3 个采样周期，ADC 转换码值才认为有效，前 3 个转换数

据应舍弃；

2）可通过硬件过采样或软件求均值的方式提高温度传感器采样精度。如果需要更高精度的温度，必须多次（推荐 50 次以上）采样求平均。

将 ADC_CTL1 寄存器的 INREFEN 位置 1，可以使能 VREFINT 通道（ADC_IN17）。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 ADC_IN17 输入通道。

14.4.12. 可编程分辨率(DRES)

对寄存器 ADC_OVSAMPCTL 中的 DRES[1:0]位进行编程即可配置分辨率为 6、8、10、12 位。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 位为 0 时，才能修改 DRES[1:0]的值。ADC 转换的结果只有 12 位，其余没有被用到的低位读出来都是为 0。较低的分辨率能够减少逐次逼近步骤所需的转换时间，如*表 14-4. 不同分辨率对应的 tCONV 时间*所示。

表 14-4. 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}=15MHz	t_{SMPL}(min) (ADC clock cycles)	t_{ADC} (ADC clock cycles)	t_{ADC}(ns) at f_{ADC}=15MHz
12	12.5	833 ns	2.5	15	1000 ns
10	10.5	700 ns	2.5	13	867 ns
8	8.5	567ns	2.5	11	733 ns
6	6.5	433 ns	2.5	9	600 ns

14.4.13. 片上硬件过采样

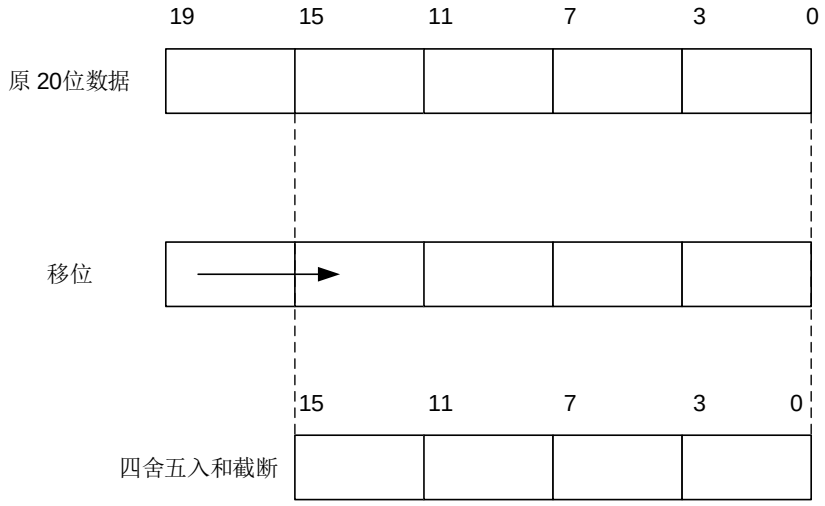
片上硬件过采样单元执行数据预处理以减轻 CPU 负担。它能够处理多个转换，并将多个转换的结果取平均，得出一个 16 位宽的数据。其结果根据如下公式计算得出，其中 N 和 M 的值可以被调整，$D_{out}(n)$是指 ADC 输出的第 n 个数字信号：

$$\text{Result}=\frac{1}{M}*\sum_{n=0}^{N-1}D_{out}(n) \qquad (14\text{-}2)$$

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 14-11. 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 14-12. 右移 5 位和取整的数例*描述一个从原始 20 位的累积数值处理成 16 位结果值。

图 14-12. 右移 5 位和取整的数例

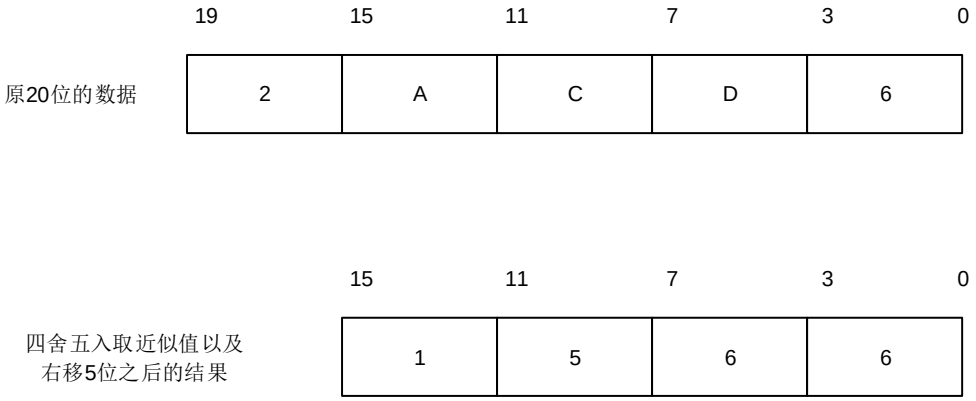


*表 14-5. N 和 M 的最大输出值（灰色部分表示截断）*给出了 N 和 M 各种组合的数据格式，初始转换值为 0xFFF。

表 14-5. N 和 M 的最大输出值（灰色部分表示截断）

过采样率	最大原始数据	不移位 OVSS=0000	移1位 OVSS=0001	移2位 OVSS=0010	移3位 OVSS=0011	移4位 OVSS=0100	移5位 OVSS=0101	移6位 OVSS=0110	移7位 OVSS=0111	移8位 OVSS=1000
2x	0x1FFE	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080	0x0040	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

过采样率	最大原始数据	不移位 OVSS=0000	移1位 OVSS=0001	移2位 OVSS=0010	移3位 OVSS=0011	移4位 OVSS=0100	移5位 OVSS=0101	移6位 OVSS=0110	移7位 OVSS=0111	移8位 OVSS=1000
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：

$$N \times t_{ADC} = N \times (t_{SMPL} + t_{CONV}) \quad (14\text{-}3)$$

14.5. ADC 同步模式

在具有两个ADC的设备上，可以使用ADC同步模式。

在ADC同步模式中，通过ADC0的触发器来同步ADC1的转换。根据ADC_CTL0寄存器中SYNCM[3:0]位来选择两个ADC按并行模式还是交替模式进行转换。

在同步模式下，ADC0配置为外部触发转换，ADC1必须配置为软件触发。此外，主从ADC的外部触发都必须使能。

ADC同步模式如*表14-6. ADC同步模式表*示。

在ADC同步模式下，即使不使用DMA功能，也要将DMA置位。ADC1的转换数据可以通过ADC0的常规数据寄存器（ADC0_RDATA）读取。

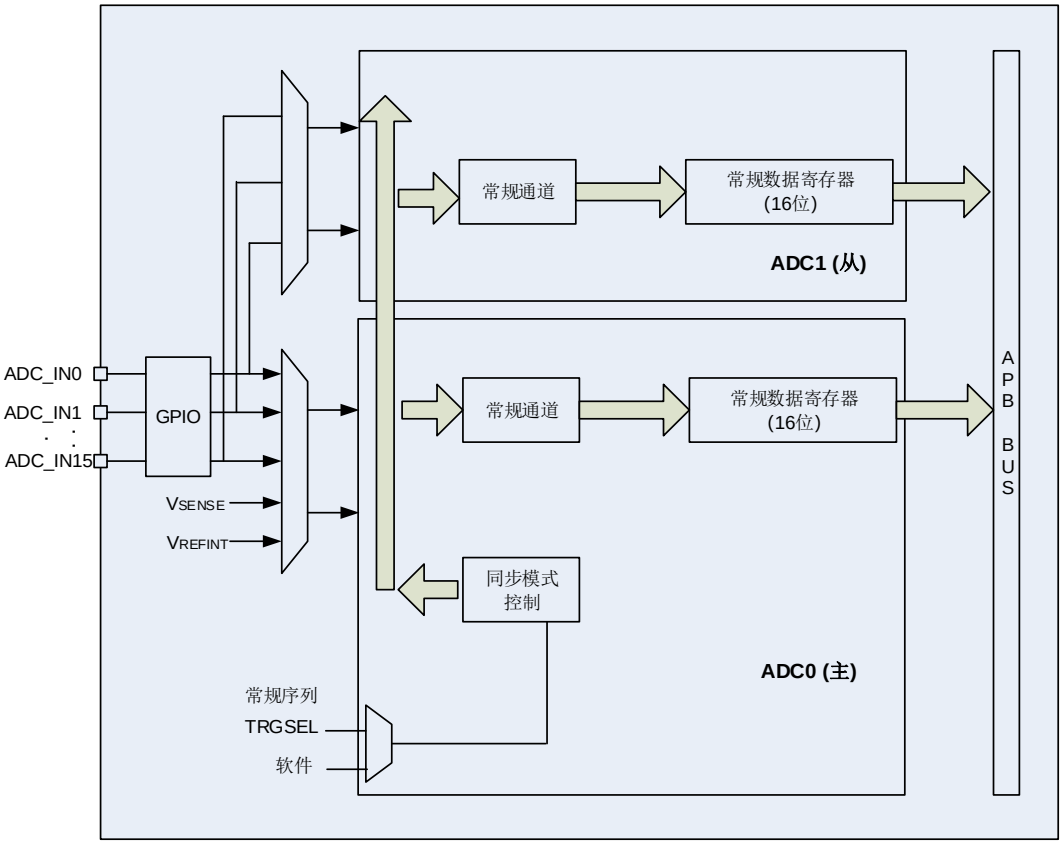
表 14-6. ADC 同步模式表

SYNCM[3: 0]	模式
0000	独立模式
0110	常规并行模式
0111	常规快速跟随模式
1000	常规慢速跟随模式

当 ADC 工作在同步模式，而非独立模式时，如果需要再将 ADC 配置成其他同步模式，则需要在配置成其他同步模式前，首先将 ADC 配置成独立模式。

ADC 同步框图如*图 14-13. ADC 同步框图*所示。

图 14-13. ADC 同步框图



14.5.1. 独立模式

在这种模式下，ADC 同步是忽略的，每个 ADC 都独立工作。

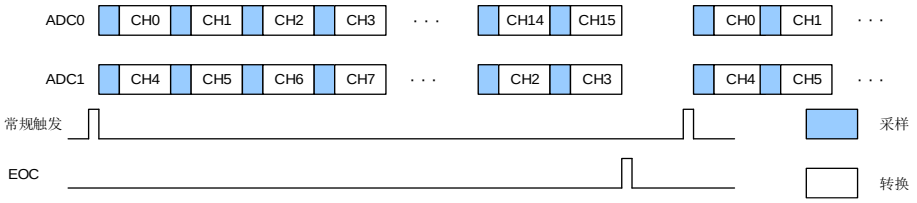
14.5.2. 常规并行模式

设置 ADC0_CTL0 寄存器的 SYNCM[3:0]位为 4b'0110，使能常规并行模式。在常规并行模式中，根据 ADC0 中选择的外部触发，所有的 ADC 并行的转换常规通道。触发选择由 ADC0 的 ADC_CTL1 寄存器 ETSRC 位进行配置。

在转换结束时产生 EOC 中断(如果 ADC 接口使能了该中断)。常规并行模式的行为如*图 14-14. 基于 16 个通道的常规并行模式*所示。

32 位 ADC_RDATA 寄存器包含上半字（由 ADC1 转换的数据构成）和下半字（由 ADC0 转换的数据构成），32 位的 DMA 被用来将 ADC_RDATA 中的数据传送到 SRAM。

图 14-14. 基于 16 个通道的常规并行模式



注意：

1. 在一个给定的时间，两个 ADC 不能同时转换同一个通道。（当转换同一通道时，不能覆盖采样时间）
2. 确保在没有任何一个 ADC 在进行转换的时候才触发 ADC。

14.5.3. 常规快速跟随模式

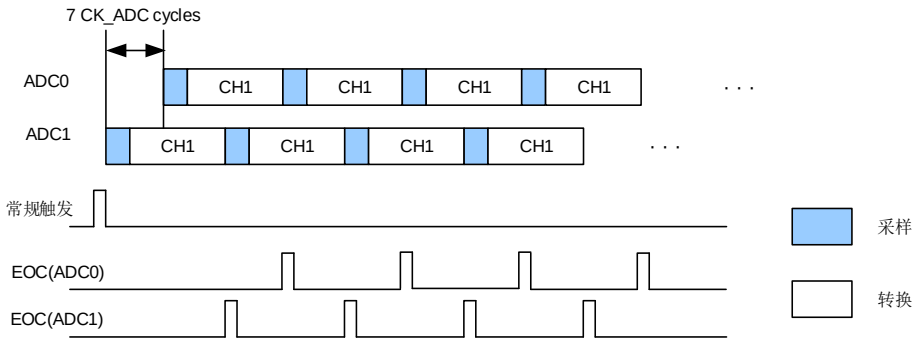
设置 ADC0_CTL0 寄存器的 SYNCM[3:0]位为 4b'0111，使能常规快速跟随模式。在常规快速跟随模式中，根据选择的外部触发，ADC1 开始转换常规序列。外部触发选择由 ADC0 的 ADC_CTL1 寄存器 ETSRC 位进行配置。经过 7 个 ADC 时钟周期之后，ADC0 开始转换常规序列。以上描述中提到的常规序列只能包含一个通道。

如果 ADC_CTL1 寄存器的 CNT 位置 1，选择的常规序列会被连续的转换。在转换事件结束时产生 EOC 中断（如果 ADC 使能了该中断）。常规跟随模式的行为如*图 14-15. 一个采用连续转换模式通道上的常规快速跟随模式*所示。

32 位 ADC_RDATA 寄存器包含上半字（由 ADC1 转换的数据构成）和下半字（由 ADC0 转换的数据构成），32 位的 DMA 被用来将 ADC_RDATA 中的数据传送到 SRAM。

注意：可允许的最大采样时间必须小于 7 个 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 14-15. 一个采用连续转换模式通道上的常规快速跟随模式



14.5.4. 常规慢速跟随模式

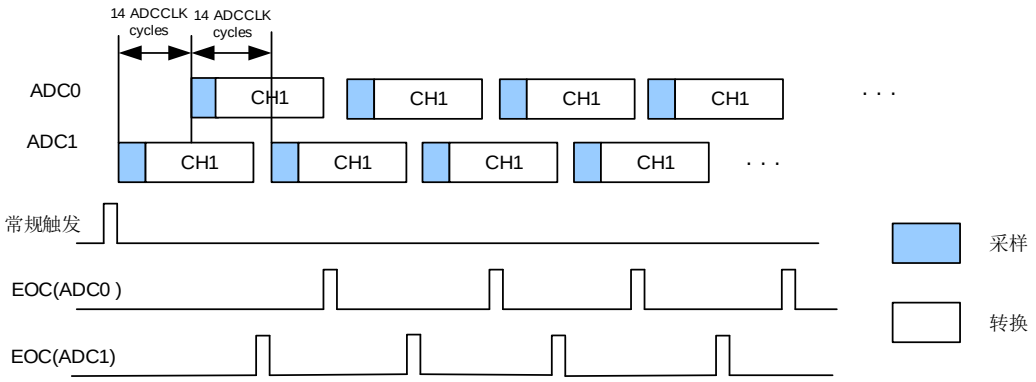
设置 ADC0_CTL0 寄存器的 SYNCM[3:0]位为 4b'1000，使能常规慢速跟随模式。在常规慢速跟随模式中，根据选择的外部触发，ADC1 开始转换常规序列。外部触发选择由 ADC0 的 ADC_CTL1 寄存器 ETSRC 位进行配置。经过 14 个 ADC 时钟周期之后，ADC0 开始转换常规序列。以上描述中提到的常规序列只能包含一个通道。

在这种模式下，不能使用连续转换模式，因为在这种模式下所选的常规通道在两个 ADC 中被不停的转换，如*图 14-16. 一个通道上的常规慢速跟随模式*。

如果置位 EOCIE 被置 1，在 ADC0 转换结束事件时会产生 EOC 中断。也可以可通过 32 位 DMA 将 ADC_RDATA 中数据传送到 SRAM。32 位 ADC_RDATA 寄存器包含上半字（由 ADC1 转换的数据构成）和下半字（由 ADC0 转换的数据构成）。

注意：可允许的最大采样时间必须小于 14 个 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 14-16. 一个通道上的常规慢速跟随模式



14.6. 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0/1事件。

单独的中断使能位可使得使用更灵活。

ADC0和ADC1都被映射到同一个中断向量IRQ18。

14.7. ADC 寄存器

ADC0基地址：0x4001 2400

ADC1基地址：0x4001 2800

14.7.1. 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。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	WDE1	保留													
	rc_w0														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											STRC	保留		EOC	WDE0
											rc_w0			rc_w0	rc_w0

位/位域	名称	描述
31	保留	必须保持复位值。
30	WDE1	模拟看门狗1事件标志 0：没有模拟看门狗事件 1：产生模拟看门狗事件 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 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：没有模拟看门狗事件 1：产生模拟看门狗事件

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。

14.7.2. 控制寄存器 0（ADC_CTL0）

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	WDE1IE	保留						RWD0EN	保留			SYNCM[3:0]			
	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	保留	必须保持复位值。
30	WDE1IE	WDE1中断使能 0：WDE1中断禁能 1：WDE1中断使能
29:24	保留	必须保持复位值。
23	RWD0EN	常规通道模拟看门狗0使能 0：常规通道模拟看门狗0禁能 1：常规通道模拟看门狗0使能
22:20	保留	必须保持复位值。
19:16	SYNCM[2:0]	同步模式选择 当ADC同步模式已经使能，如果要将同步模式修改为其他值，必须先将这些位设置为00000。 0000：独立模式。所有的ADC都独立工作。 0110：ADC0和ADC1工作在常规并行模式 0111：ADC0和ADC1工作在常规快速跟随模式 1000：ADC0和ADC1工作在常规慢速跟随模式 其他值保留。
15:13	DISNUM[2:0]	间断模式下的转换数目 触发后常规序列中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：间断运行模式禁能 1：间断运行模式使能

10	保留	必须保持复位值。
9	WD0SC	扫描模式下，模拟看门狗0在单通道有效 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：扫描运行模式禁能 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：WDE0中断禁能 1：WDE0中断使能
5	EOCIE	EOC中断使能 0：EOC中断禁能 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：** 1. 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}。 2. ADC1的模拟输入通道16和通道17内部都连接到V_{SSA}。

14.7.3. 控制寄存器 1（ADC_CTL1）

地址偏移：0x08

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留							INREFEN	TSVEN	SWRCST	SWICST	ETERC	保留		ETSRC	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.				DAL	保留.		DMA	保留				RSTCLB	CLB	CTN	ADCON
				rw			rw					rw	rw	rw	rw

位/位域	名称	描述
31:25	保留	必须保持复位值。
24	INREFEN	使能ADC0的通道17（内部参考电压） 0：ADC0的通道17禁能 1：ADC0的通道17使能
23	TSVEN	使能ADC0的通道16（温度传感器） 0：ADC0的通道16禁能 1：ADC0的通道16使能
22	SWRCST	常规序列软件启动转换 如果ETSRC是1，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：常规序列外部触发禁能 1：常规序列外部触发使能
19:18	保留	必须保持复位值。
17	ETSRC	常规序列外部触发选择 0：TRIGSEL 1：软件触发（SWRCST）
16:12	保留	必须保持复位值。
11	DAL	数据对齐 0：最低有效位对齐 1：最高有效位对齐
10:9	保留	必须保持复位值。
8	DMA	DMA请求使能 0：DMA请求禁能 1：DMA请求使能 **注意：**仅ADC0有该位。

7:4	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：禁能连续运行模式 1：使能连续运行模式
0	ADCON	开启ADC 该位从0变成1将唤醒ADC，并且ADC需要一段稳定时间t_{SU}。 当该位已经置高，再给该位写“1”将会启动ADC转换。 为了省电，当该位为0时，模拟子模块将会进入掉电模式。 0：禁能ADC，并进入掉电模式 1：使能ADC

14.7.4. 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2.5周期 001：通道采样时间为14.5周期 010：通道采样时间为27.5周期 011：通道采样时间为55.5周期 100：通道采样时间为83.5周期 101：通道采样时间为111.5周期 110：通道采样时间为143.5周期 111：通道采样时间为479.5周期

14.7.5. 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2.5周期 001：通道采样时间为14.5周期 010：通道采样时间为27.5周期

011：通道采样时间为55.5周期
100：通道采样时间为83.5周期
101：通道采样时间为111.5周期
110：通道采样时间为143.5周期
111：通道采样时间为479.5周期

14.7.6. 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。

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				WDHT0[11:0]											
				rw											

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的高侧阈值。

14.7.7. 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。

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				WDLT0[11:0]											
				rw											

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的低侧阈值。

14.7.8. 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。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								RL[3:0]				RSQ15[4:1]			
								rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSQ15[0]	RSQ14[4:0]					RSQ13[4:0]					RSQ12[4:0]				
rw	rw					rw					rw				

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

14.7.9. 常规序列寄存器 1（ADC_RSQ1）

地址偏移：0x30
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	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

14.7.10. 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序列的第n个转换的通道。

14.7.11. 常规数据寄存器（ADC_RDATA）

地址偏移：0x4C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADC1RDTR[15:0]															
r															

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

位/位域	名称	描述

31:16	ADC1RDTR[15:0]	ADC1常规通道数据 在ADC0中：在同步模式下，这些位包含着ADC1的常规通道数据。 在ADC1中：这些位保留。
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。

14.7.12. 过采样控制寄存器（ADC_OVSAMPCTL）

地址偏移：0x80
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	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	过采样触发 该位通过软件设置和清除。 0：在一次触发后连续执行过采样通道的所有转换 1：对于过采样通道的每次转换都需要一次触发，触发次数由过采样率（OVSR[2:0]）决定。 **注意：**只有在ADCON=0的时候才允许通过软件对该位进行写(确保没有转换正在执行)。
8:5	OVSS[3:0]	过采样移位 该位通过软件置位和清除。 0000：不移位 0001：右移1位 0010：右移2位 0011：右移3位 0100：右移4位

		0101：右移5位 0110：右移6位 0111：右移7位 1000：右移8位 其余值都保留。 **注意：**只有在ADCON=0的时候才允许通过软件对该位进行写(确保没有转换正在执行)。
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：**只有在ADCON=0的时候才允许通过软件对该位进行写(确保没有转换正在执行)。
1	保留	必须保持复位值。
0	OVSEN	过采样使能 该位通过软件置位和清除。 0：过采样禁能 1：过采样使能 **注意：**只有在ADCON=0的时候才允许通过软件对该位进行写（确保没有转换正在执行）。

14.7.13. 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
Reserved														AWD1CS[17:16]	
														rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWD1CS[15:0]															
rw															

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AWD1CS[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寄存器中配置的通道； 2）只有在ADC禁能（ADCON=0）时，才能软件写这些位。

14.7.14. 看门狗 1 阈值寄存（ADC_WDT1）

地址偏移：0xA8
复位值：0x00FF 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
Reserved								WDHT1[7:0]							
											rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								WDLT1[7:0]							
											rw				

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。

15. 数模转换器（DAC）

15.1. 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。

15.2. 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}；
- 噪声波生成（LSFR 噪声模式和三角噪声模式）。

*图 15-1. DAC 结构框图*为 DAC 的结构框图，*表 15-1. DAC 引脚*给出了引脚描述。

图 15-1. DAC 结构框图

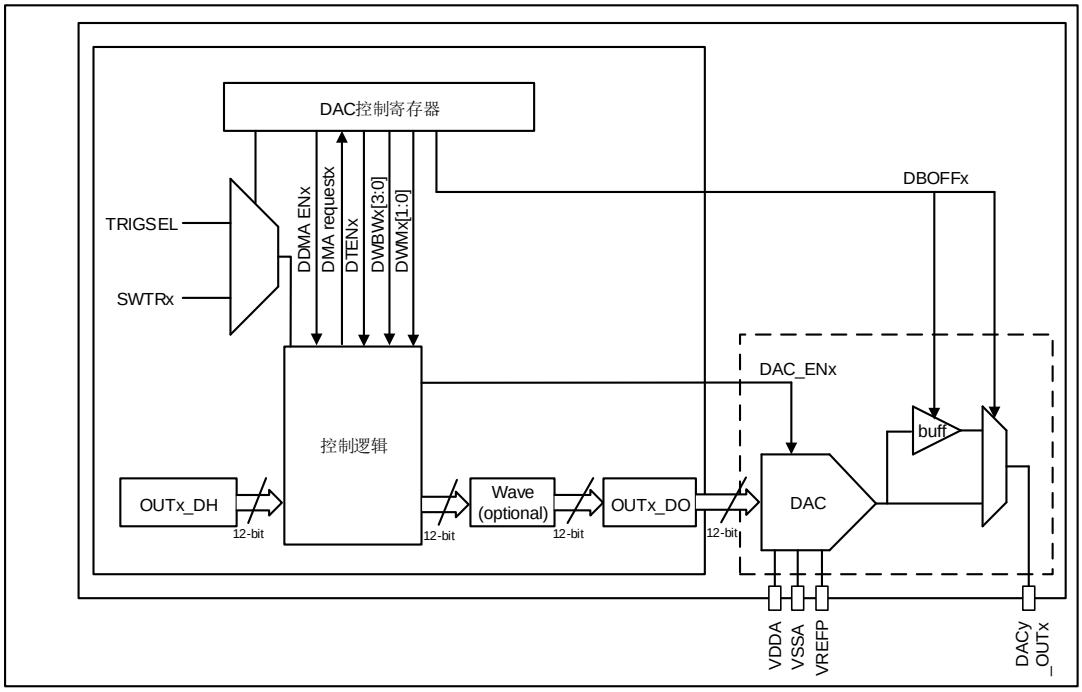


表 15-1. DAC 引脚

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 15-2. DAC 触发与输出

	DAC0
通道	通道 0
DAC 输出 I/O	PA7
DAC 输出 BUFFER 功能	●
TRIGSEL 触发功能	●
软件触发功能	●

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

15.3. 功能描述

15.3.1. 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。

15.3.2. 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。

当 DAC 输出缓存功能关闭，DAC 可以通过设置 DAC_CTL0 寄存器中 DDISC 位，实现独立连接到芯片的 CMP 外设中。

15.3.3. 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。

15.3.4. 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 15-3.*

*DAC 外部触发*所示。

表 15-3. DAC 外部触发

DTSELx[1:0]	触发源	触发类型
2b'00	TRIGSEL	硬件触发
2b'01	SWTR	软件触发
2b'10	保留	保留
2b'11		

外部触发信号由触发选择控制器(TRIGSEL)产生，而软件触发是通过设置 DAC_SWT 寄存器的 SWTRx 位生成的。

15.3.5. 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）。

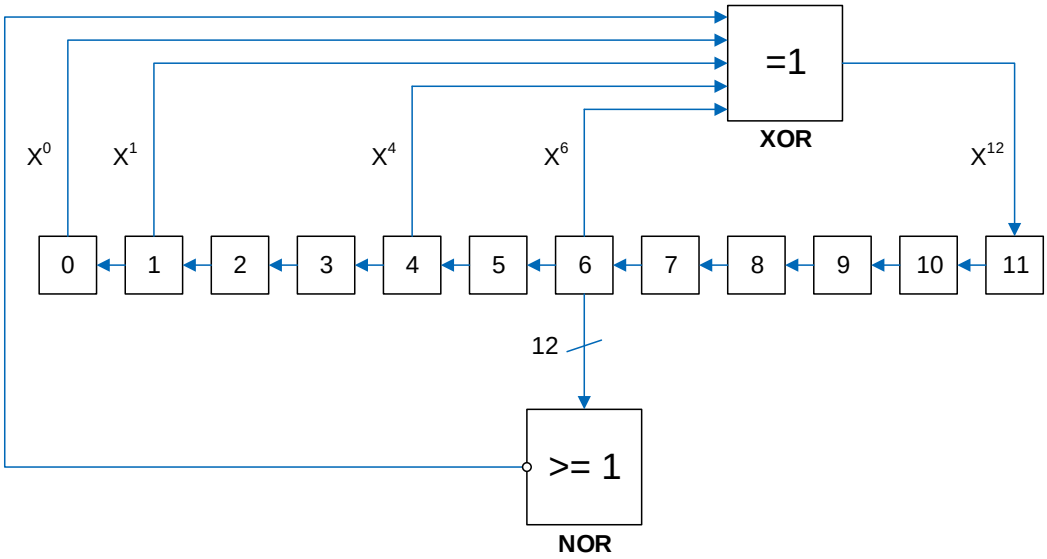
当 DAC 保持数据（OUTx_DH）加载到 DAC_OUTx_DO 寄存器时，经过 $t_{SETTLING}$ 时间之后，模拟输出变得有效，$t_{SETTLING}$ 的值与电源电压和模拟输出负载有关。

15.3.6. 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。

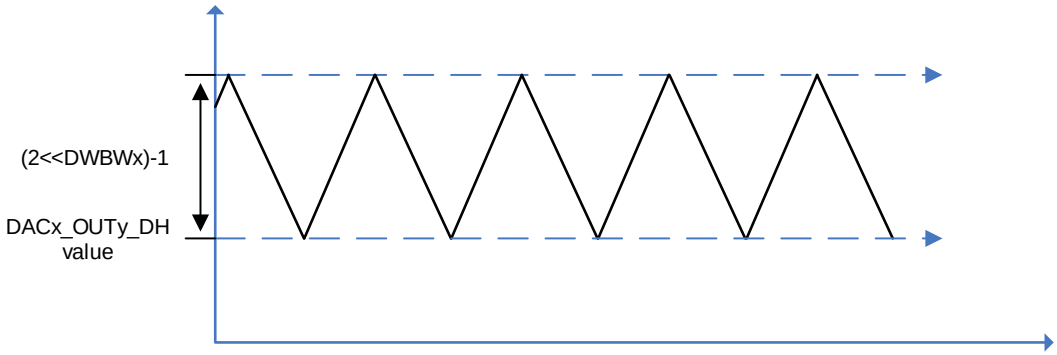
LFSR 噪声模式：在 DAC 控制逻辑中有一个线性反馈移位寄存器（LFSR）。在此模式下，LFSR 的值与 OUTx_DH 值相加后，被写入到 DAC 数据输出寄存器（DAC_OUTx_DO）。当配置的 DAC 噪声波位宽小于 12 时，LFSR 的值等于 LFSR 寄存器最低的 DWBWx 位，高位被屏蔽。

图 15-2. 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 15-3. DAC 三角噪声模式生成的波形



15.3.7. DAC 输出电压

DAC 引脚上的模拟输出电压取决于下面的等式：

$$V_{DAC_OUT}=V_{REFP}*OUTx_DO/4096 \quad (15\text{-}1)$$

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}。

15.3.8. 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。

15.4. DAC 寄存器

DAC0 基地址：0x4000 7400

15.4.1. 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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	DDISC0	DDUDRIE0	DDMAEN0	DWBW0[3:0]				DWM0[1:0]		保留	DTSEL0[1:0]		DTEN0	DBOFF0	DEN0
	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	DDISC0	DACx_OUT0 连接 GPIO 选择 0：DACx_OUT0 连接到外部管脚和片上外设（CMP） 1：当输出缓冲区关闭时（DBOFF=1）仅连接到片上外设（CMP），缓冲区打开时连接到外部管脚和片上外设（CMP）
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	保留	必须保持复位值
4:3	DTSEL0[1:0]	DACx_OUT0 触发选择 这些位仅在 DTEN=1 并选择用于触发 DAC 的外部事件时使用。 00：EXTRIG(外部触发来自 TRIGSEL) 01：软件触发 其他值：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

15.4.2. 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															SWTR0
															w

位/位域	名称	描述
31:1	保留	必须保持复位值

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

15.4.3. 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。

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

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:4	OUT0_DH[11:0]	DACx_OUT0 12 位左对齐数据 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

15.4.5. 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。

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[7:0]							
											rw				

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。

15.4.6. DACx_OUT0 数据输出寄存器 (DAC_OUT0_DO)

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				OUT0_DO [11:0]											
											r				

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。

15.4.7. DACx 状态寄存器 0 (DAC_STAT0)

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		DDUDR0	保留												
		rc_w1													

位/位域	名称	描述
31:14	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。

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的门限时，会触发一个复位（对于窗口看门狗定时器来说，会产生一个中断）。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或计数器的值大于窗口寄存器的值时刷新计数器，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。

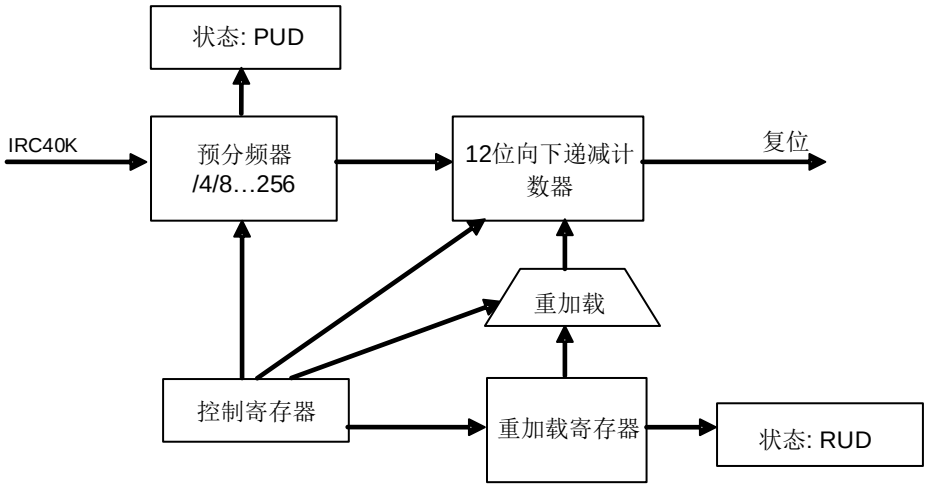
16.1.2. 主要特征

- 自由运行的12位向下计数器；
- 使能看门狗定时器，有以下两种情况下会产生系统复位：
  - 当计数器到0时产生系统复位；
  - 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。参考*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。

独立看门狗定时器也能够工作在窗口看门狗定时器模式下，只要在窗口寄存器FWDGT_WND寄存器中设置适当的值即可。当重加载操作执行时，看门狗定时器计数器的值大于FWDGT_WND中存储的值，将会引起系统复位。FWDGT_WND的默认值是0x00000FFF，所以如果没有改写它，那么窗口选项默认是关闭的。窗口值一旦改变，立即就会引起看门狗定时器计数器的一次重加载动作，将向下递减计数器置为FWDGT_RLD中的值，并复位预分频计数器。

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。写其他任何值到控制寄存器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

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 / 128	101	0.025	13104.025
1 / 256	110或111	0.025	26208.025

通过校准IRC40K可以使自由看门狗定时器超时更加精确。

注意：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。

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。

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
CMD[15:0]															
w															

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、FWDGT_RLD和FWDGT_WND的写保护 0xCCCC：开启独立看门狗定时计数器。计数减到0时产生系统复位 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。

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													PSC[2:0]		
														rw	

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（在进入省电模式前需等待PUD值清零）。

重装载寄存器（FWDGT_RLD）

地址偏移：0x08

复位值：0x0000 0FFF

该寄存器可以按半字（16 位）或字（32 位）访问。

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

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

位/位域	名称	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，改变重加载值之前必须等到RUD位被清0。更新了重加载寄存器的值后，在代码持续执行之前不必等待RUD值被清零（在进入省电模式前需等待RUD值清零）。

状态寄存器（FWDGT_STAT）

地址偏移：0x0C

复位值：0x0000 0000

该寄存器可以按半字（16 位）或字（32 位）访问。

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

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

位/位域	名称	说明
31:3	保留	必须保持复位值。
2	WUD	独立看门狗定时计数器窗口值更新 FWDGT_WND寄存器写操作时，该位被置1，此时读取FWDGT_WND寄存器的任何值都是无效的。
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。

窗口寄存器（FWDGT_WND）

地址偏移：0x10

复位值：0x0000 0FFF

该寄存器可以按半字（16 位）或字（32 位）访问。

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

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

位/位域	名称	说明
31:12	保留	必须保持复位值。
11:0	WND	独立看门狗定时器计数器窗口值。这些位将用来将窗口值的上限值与向下递减计数器进行比较。当计数值大于WMD[11:0]中值，重装载操作会引起复位，若要改变重装载值，FWDGT_STAT寄存器中的WUD位必须保持复位状态。 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。 如果应用需要使用几个窗口值，改变窗口值之前必须等到WUD位被清0。除了在进入低功耗模式下，更新了窗口值后，在代码持续执行之前不必等待WUD值被清零。

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。

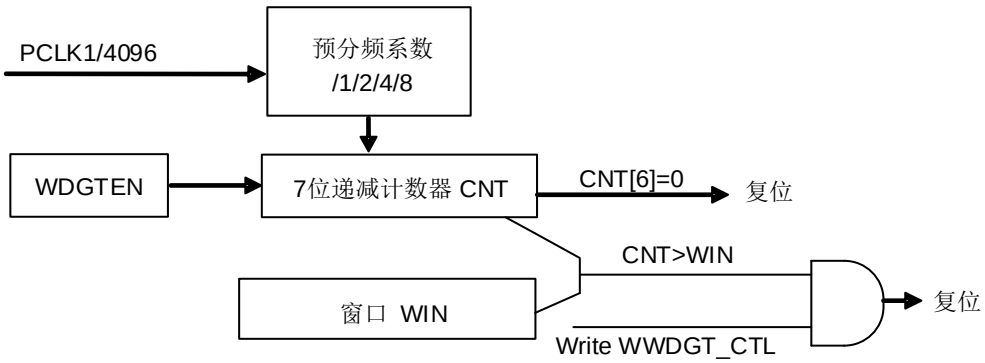
16.2.2. 主要特征

- 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）。

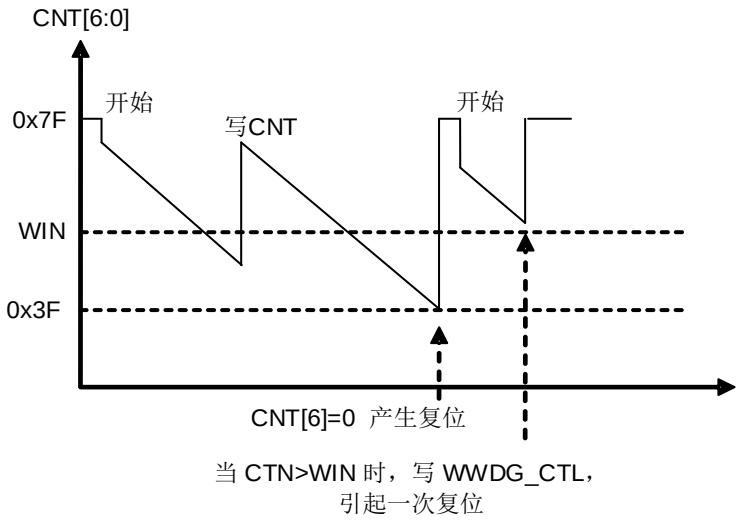
配置寄存器（WWDGT_CFG）中的WIN[6:0]位用来设定窗口值。当计数器的值小于窗口值，且

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：

$$t_{WWDGT}=t_{PCLK1} \times 4096 \times 2^{PSC} \times (CNT[5:0]+1) \quad (ms) \qquad (16\text{-}1)$$

其中：

t_{WWDGT}：窗口看门狗定时器的超时时间

t_{PCLK1}： APB1以ms为单位的时钟周期

t_{WWDGT} 的最大值和最小值请参考*表 16-2. 在 50MHz（fPCLK1）时的最大/最小超时值*。

表 16-2. 在 50MHz（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	81.92μs	5.24ms
1 / 2	01	163.84μs	10.49ms
1 / 4	10	327.68μs	20.97ms
1 / 8	11	655.36μs	41.94ms

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。

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								WDGTEN	CNT[6:0]						
								rs	rw						

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。

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
保留						EWIE	PSC[1:0]		WIN[6:0]						
						rs	rw		rw						

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软件复位来清0。写0没有任何作用。
8:7	PSC[1:0]	预分频器，看门狗定时器的时间基准。 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。

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															EWIF
															rc_w0

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. 主要特征

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

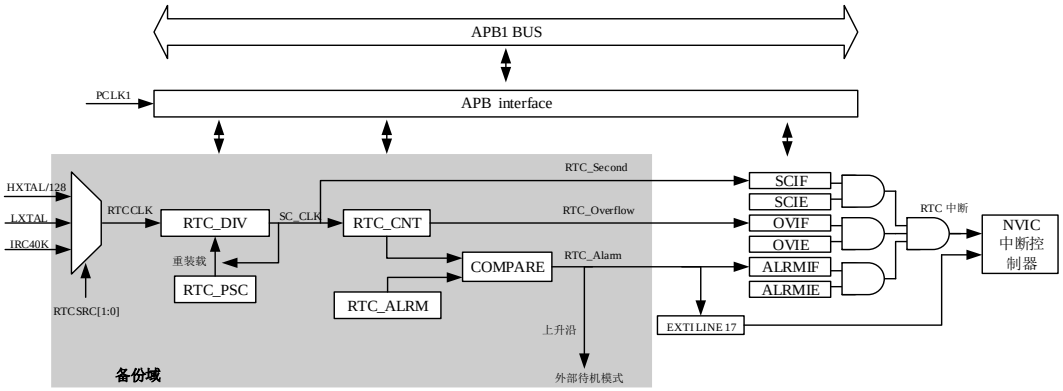
17.3. 功能说明

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

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

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

图 17-1. RTC 框图



注意：在 48-pin 和 32-pin 封装中不支持使用 LXTAL 作为 RTC 时钟源。

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)

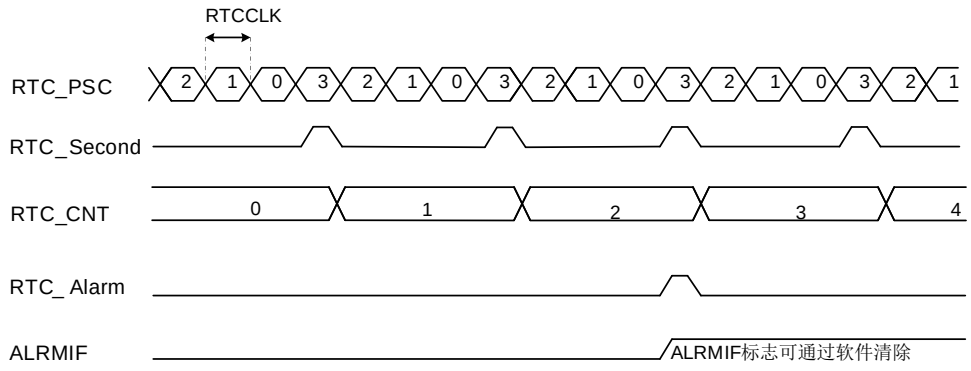
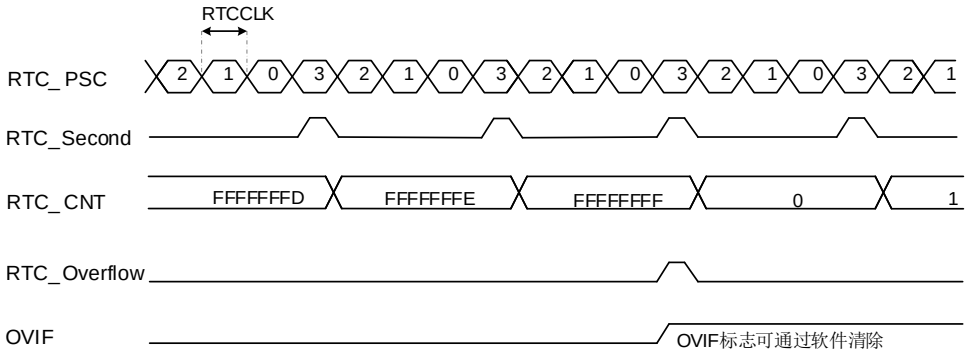


图 17-3. RTC 秒信号及溢出信号的波形(RTC_PSC = 3)



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）

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
Reserved															

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

地址偏移：0x04
复位值：0x0000 0020

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）

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。

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												PSC[19:16]			
														w	

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）

地址偏移：0x0C

复位值：0x0000 8000

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
PSC[15:0]															
w															

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）

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												DIV[19:16]			
												r			

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）

地址偏移：0x14

复位值：0x0000 8000

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
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）

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
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）

地址偏移：0x1C

复位值：0x0000 0000

该寄存器可以按半字（16 位）或字（32 位）访问。

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

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

地址偏移：0x20

复位值：0x0000 FFFF

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
ALRM[31:16]															
w															

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）

地址偏移：0x24

复位值：0x0000 FFFF

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
ALRM[15:0]															
w															

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）分为三种类型

定时器	定时器 0/7/19/20	定时器 1	定时器 5/6
类型	高级	通用（L0）	基本
预分频器	16 位	16 位	16 位
计数器	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
可重复性	●	×	×
捕获/比较通道数	8	4	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)	●(1)	TRGO TO DAC
DMA	●	●	●(2)
Debug 模式	●	●	●

(1) 详细信息请*触发选择控制器（TRIGSEL）*。

(2) 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。

18.1. 高级定时器（TIMERx,x=0,7,19.20）

18.1.1. 简介

高级定时器（TIMER0/7/19/20）是八通道定时器，支持输入捕获和输出比较。可以产生 PWM 信号控制电机和电源管理。高级定时器含有一个 16 位无符号计数器。

高级定时器是可编程的，可以用于计数，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. 主要特征

- 总通道数：8；
- 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：用来追踪运动和分辨旋转方向和位置；
- 霍尔传感器接口：用来进行三相电机控制；
- 可编程的预分频器：16位，运行时可以被改变；
- 每个通道相互独立且可配置：输入捕获模式，输出比较模式，可编程的PWM模式，单脉冲模式和触发输出；
- 可编程的死区时间和独立的死区时间配置；
- 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，换相事件和中止事件；
- 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，*表 18-2. 高级定时器通道介绍*介绍了通道输入和输出情况。

图 18-1. 高级定时器结构框图

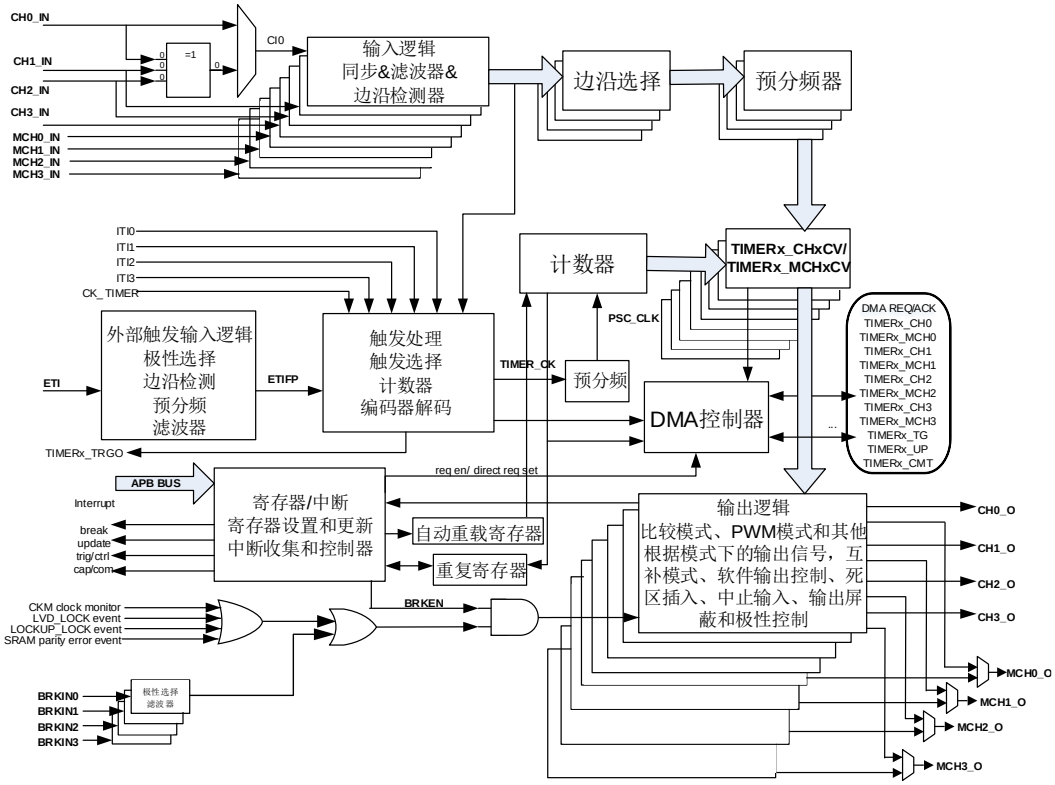


表 18-2. 高级定时器通道介绍

通道名称 （x=0..3）	MCHxMSEL[1:0]=00 独立模式	MCHxMSEL[1:0]=01 镜像模式	MCHxMSEL[1:0]=11 互补模式
CHx （通道 x）	CHx 和 MCHx 可独立输入捕获、独立比较输出	CHx 和 MCHx 输出相同 （仅用于输出）	只有 CHx 可用于输入， CHx 和 MCHx 输出互补
MCHx （多模式通道 x）			

18.1.4. 功能说明

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。

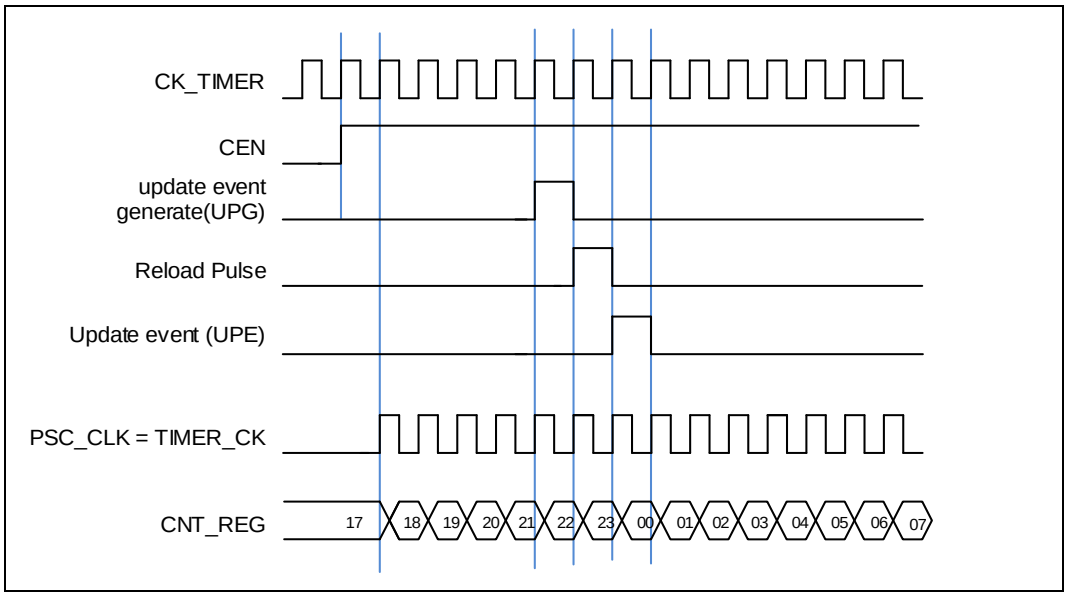
■ SMC[2:0]=3’b000，定时器选择内部时钟源（连接到RCU模块的CK_TIMER）

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 [3:0]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

驱动计数器预分频器计数的 TIMER_CK 可以在 TIMERx_CHn/ TIMERx_MCHn（n=0..3）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 0x4~0x6 和 0x8~0xD 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 [3:0]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

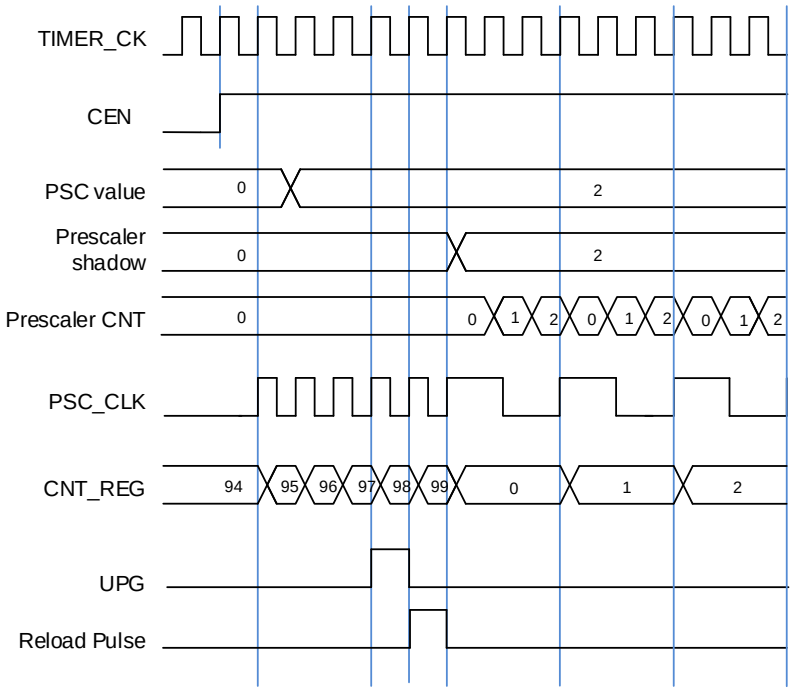
驱动计数器预分频器计数的 TIMER_CK 可以在外部引脚 ETI 的每个上升沿或下降沿计数。这种模式可以通过设置 TIMERx_SMCFG 寄存器中的 SMC1 位为 1 来选择。另一种选择 ETI 信号作为时钟源方式是，设置 SMC [2:0]为 0x7 同时设置 TRGS [3:0]为 0x7。注意 ETI 信号是通过数字滤波器采样 ETI 引脚得到的。如果选择 ETI 信号为时钟源，触发控制器包括边沿监测电路将在每个 ETI 信号上升沿产生一个时钟脉冲来为计数器预分频器提供时钟。

注意：ETI 引脚可从 TIMER_ETIx(x=0..2)引脚中任选一个，每个高级定时器（TIMER0/7/19/20）只能使用 1 个，具体请参考 ***TIMER 输入源选择寄存器（SYSCFG_TIMERINSEL）***。

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计数器自动加载值（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，并且产生上溢事件。另外，在(TIMERx_CREP+1)次上溢后将会产生更新事件。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

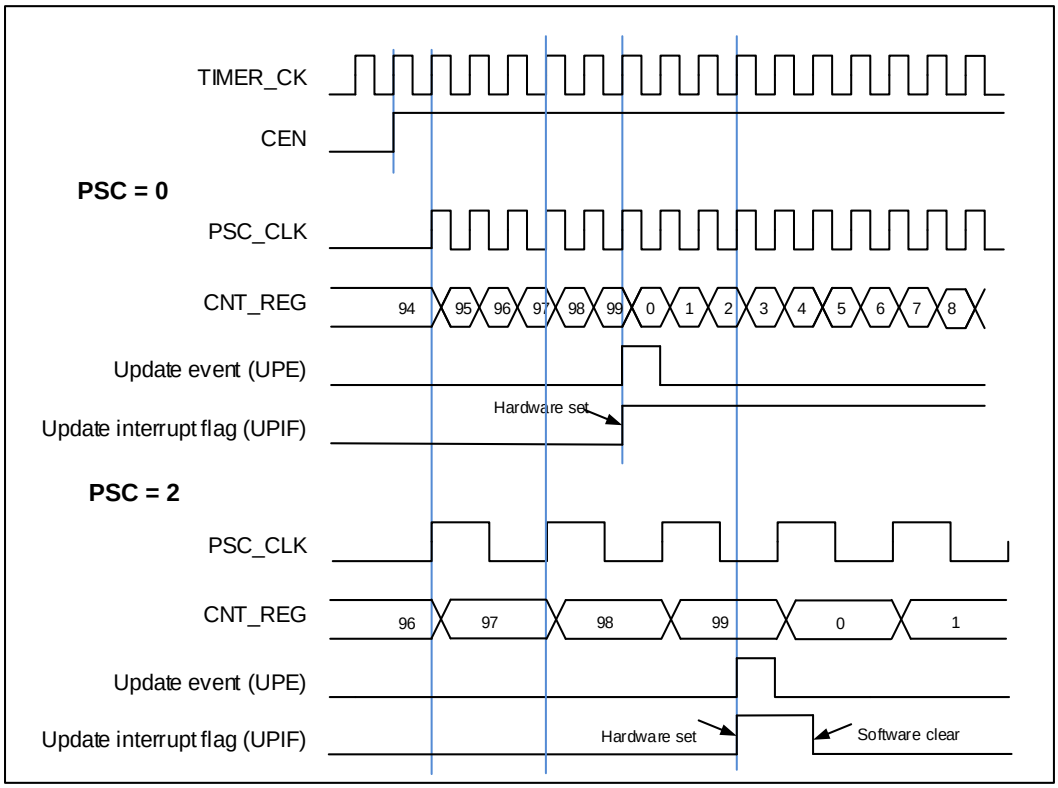
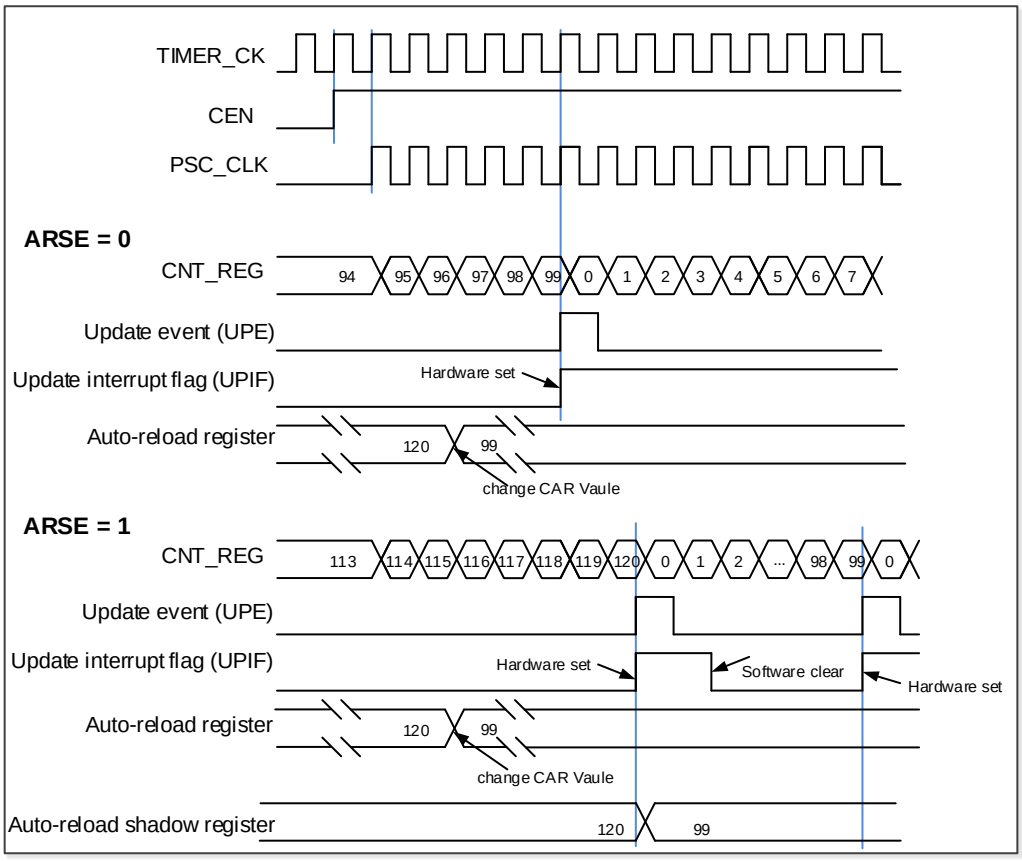


图 18-5. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



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并且产生下溢事件。另外，在（TIMERx_CREP+1）次下溢后将会产生更新事件。在向下计数模式中，TIMERx_CTL0寄存器中的计数方向控制位DIR应该被设置成1。

当通过 TIMERx_SWEVG 寄存器的 UPG 位置 1 来设置更新事件时，计数值会被初始化为自动加载值，并产生更新事件。

如果 TIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有影子寄存器（重复计数器，计数器自动重载寄存器，预分频寄存器）都将被更新。

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

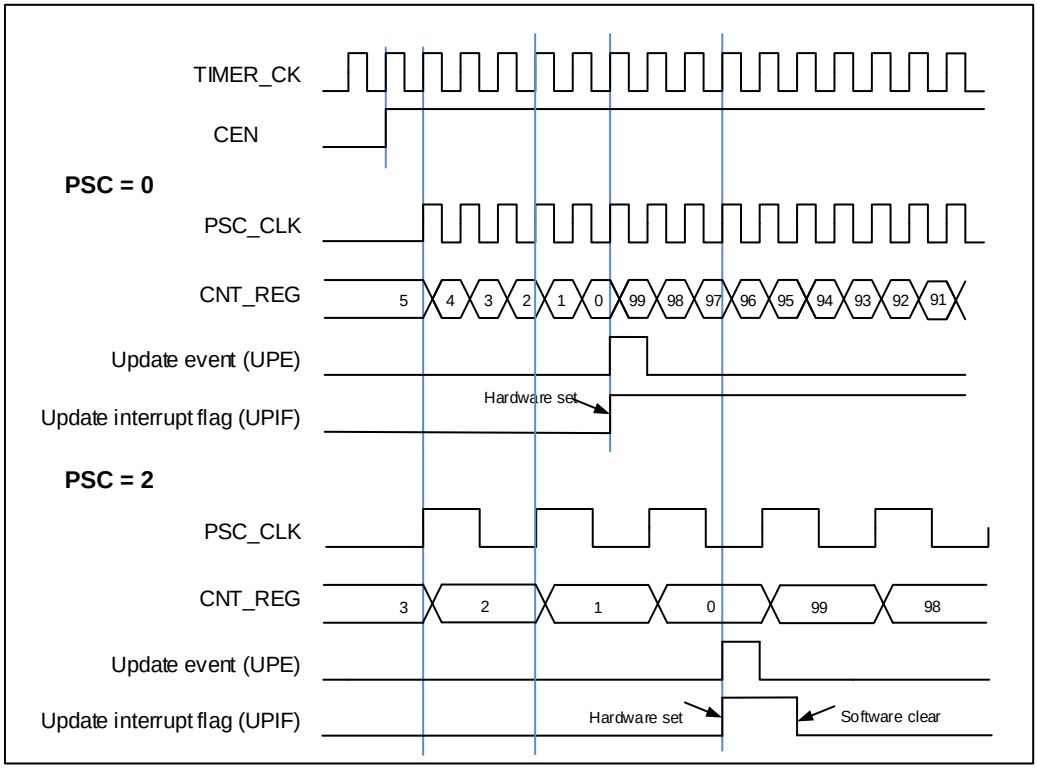
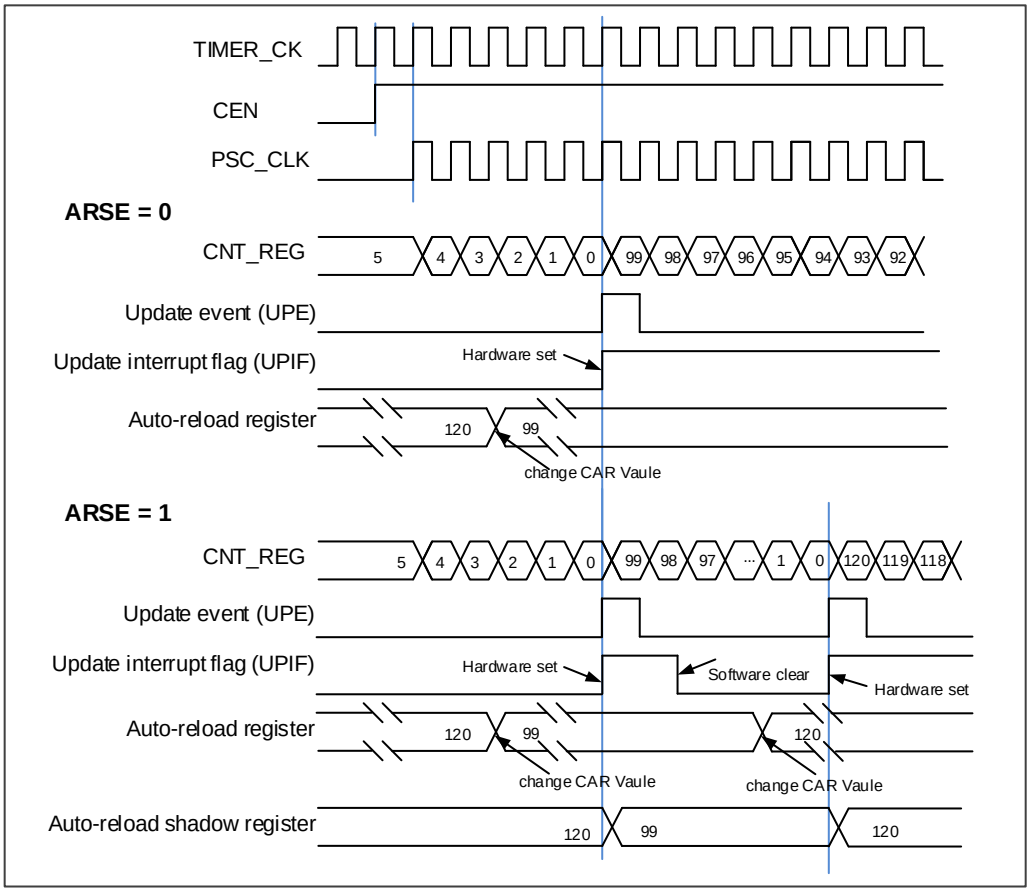


图 18-7. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值



计数器中央对齐模式

在中央对齐计数模式下，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。另外，在(TIMERx_CREP+1)次上溢和下溢事件后将会产生更新事件。在中央对齐计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 只读，指示了当前的计数方向。

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。

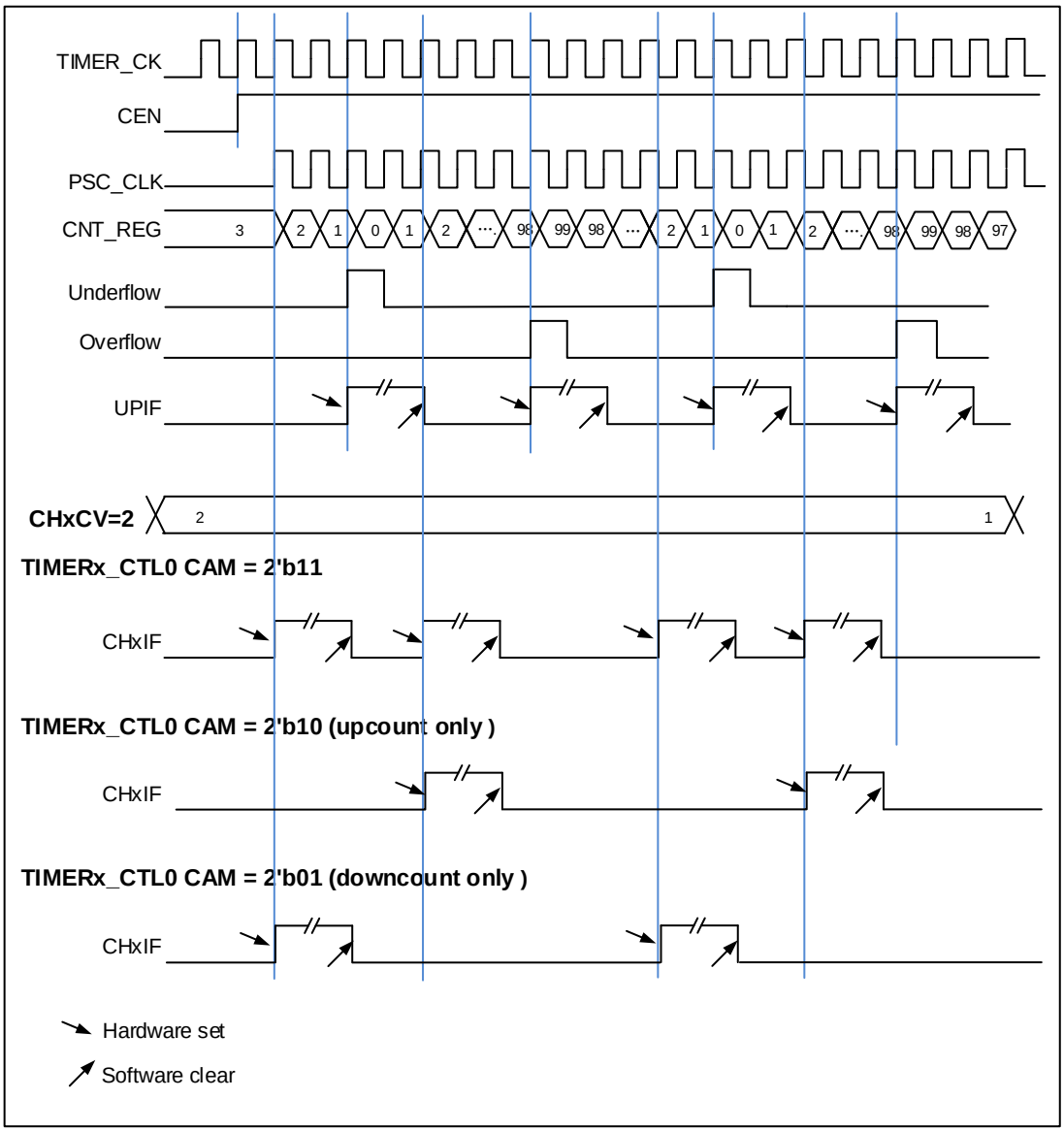
上溢或者下溢时，TIMERx_INTF 寄存器中的 UPIF 位都会被置 1，然而 CHxIF 位置 1 与 TIMERx_CTL0 寄存器中 CAM 的值有关。具体细节参考*图18-8. 中央计数模式计数器时序图*。

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 的值。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，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

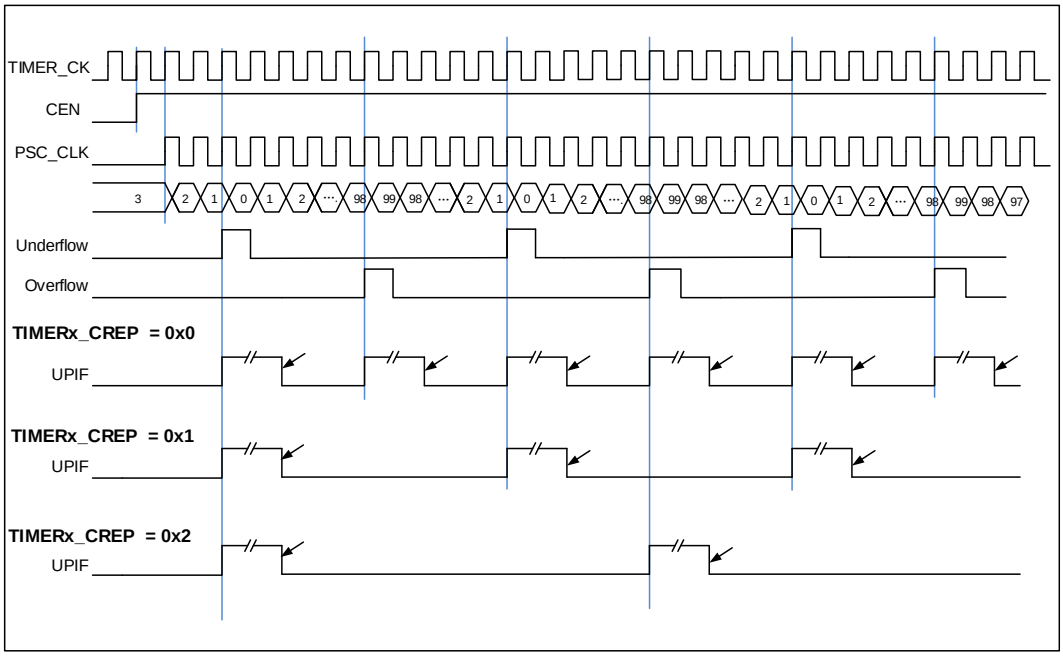


图 18-10. 在向上计数模式下计数器重复时序图

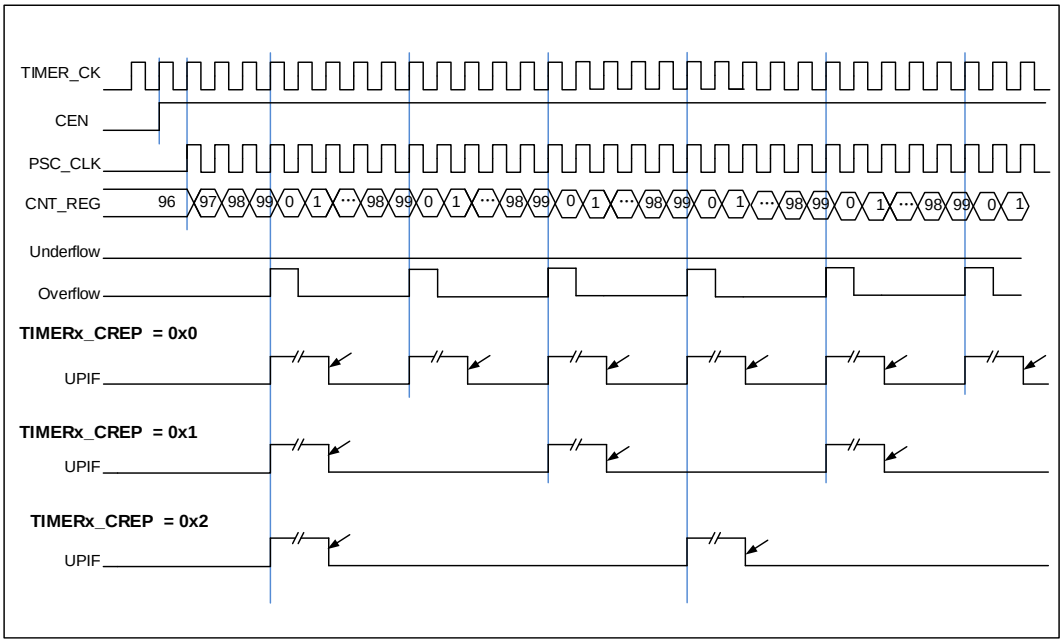
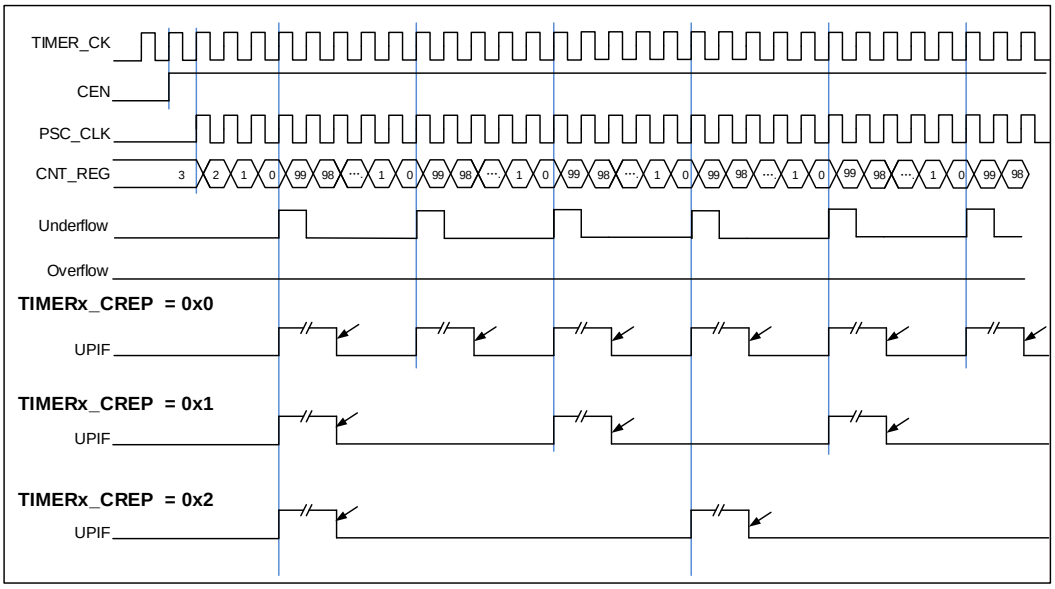


图 18-11. 在向下计数模式下计数器重复时序图



输入捕获/输出比较通道

高级定时器拥有 8 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。

当通道用于输入时，通道 x 和多模式通道 x 可独立进行输入捕获；当通道用于比较输出时，通道 x 和多模式通道 x 可输出独立、镜像和互补。

■ 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/ TIMERx_MCHxCV(x=0..3)寄存器会捕获计数器当前的值，同时 CHxIF/ MCHxIF (x=0..3)位置 1，如果 CHxIE/ MCHxIE =1(x=0..3)，则产生相应的通道中断。

图 18-12. 通道 0 输入捕获原理

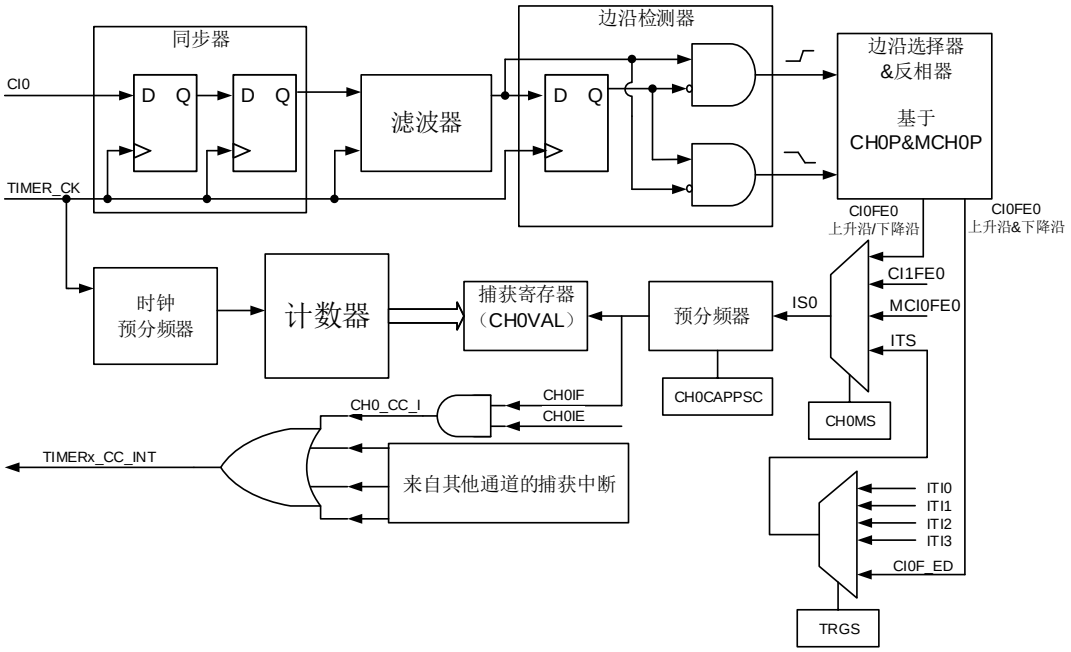
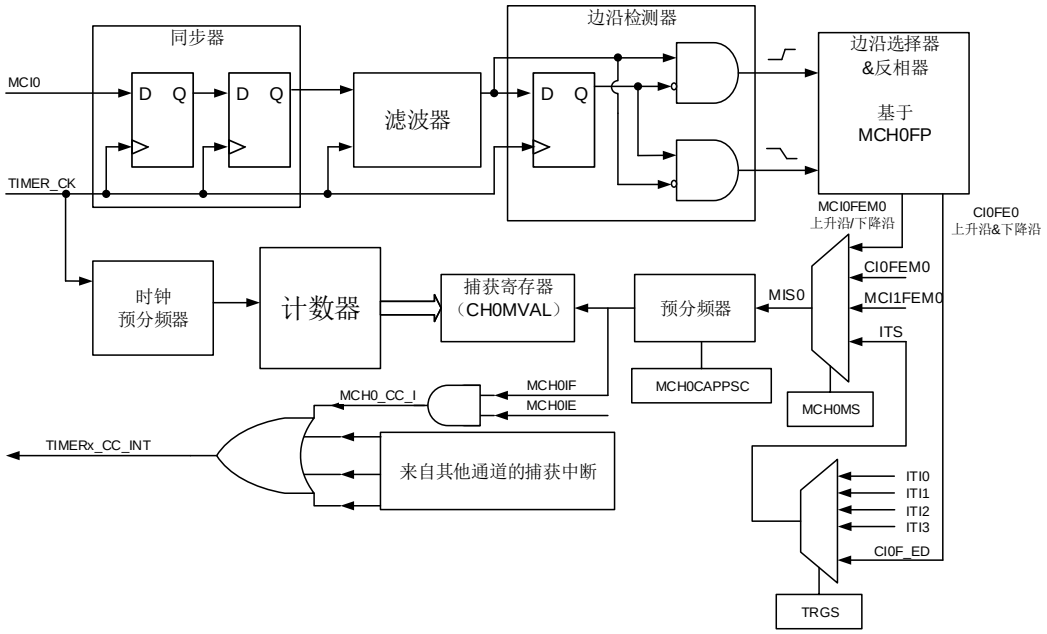


图 18-13. 通道 0 互补输入捕获原理



通道输入信号 CIx/ MCIx 有两种选择，一种是 TIMERx_CHx/ TIMERx_MCHxCV 信号，另一种是 TIMERx_CH0，TIMERx_CH1 和 TIMERx_CH2 异或之后的信号（仅限于 CI0）。

通道输入信号 CIx/ MCIx 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/ MCHxP、MCHxFP 选择使用上升沿或者下降沿。配置 CHxMS/ MCHxMS，可以选择其他通道的输入信号或内部触发信号。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/ TIMERx_MCHxCV 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位和TIMERx_MCHCTL0寄存器中CHxMCAPFLT）：

根据输入信号和请求信号的质量，配置相应的CHxCAPFLT/ CHxMCAPFLT位。

第二步：边沿选择（TIMERx_CHCTL2寄存器中CHxP和MCHxP位，TIMERx_MCHCTL2寄存器中MCHxFP[1:0]位域）：

配置CHxP和MCHxP位或MCHxFP位域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、TIMERx_MCHCTL0寄存器中MCHxMS）：

一旦通过配置CHxMS/ MCHxMS选择输入捕获源，必须确保通道配置在输入模式（CHxMS!=0x000或MCHxMS!=0x000），而且TIMERx_CHxCV/TIMERx_MCHxCV寄存器不能再被写。

第四步：中断使能（TIMERx_DMAINTEN寄存器中CHxIE、CHxDEN位和MCHxIE、MCHxDEN位）：

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/ MCHxEN）。

结果：当期望的输入信号发生时，TIMERx_CHxCV/ TIMERx_MCHxCV被设置成当前计数器的值，CHxIF/ MCHxIF位置1。如果CHxIF/ MCHxIF位已经为1，则CHxOF/ MCHxOF位置1。根据TIMERx_DMAINTEN寄存器中CHxIE、CHxDEN位和MCHxIE、MCHxDEN位的配置，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 和 TIMERx_MCHx 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 3’b001，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 3’b010，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。

■ 输出比较模式

图 18-14. 输出比较原理（当 MCHxMSEL = 2’00，x=0,1,2,3），*图 18-15. 输出比较原理（当 MCHxMSEL = 2’01，x=0,1,2,3）*和*图 18-16. 输出比较原理（当 MCHxMSEL = 2’11 时互补输出，x=0,1,2,3）*给出了通道的输出比较原理。

图 18-14. 输出比较原理（当 MCHxMSEL = 2’00，x=0,1,2,3）

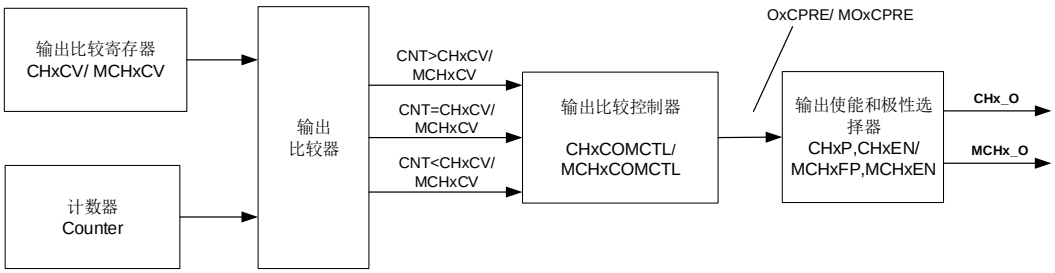


图 18-15. 输出比较原理（当 MCHxMSEL = 2’01，x=0,1,2,3）

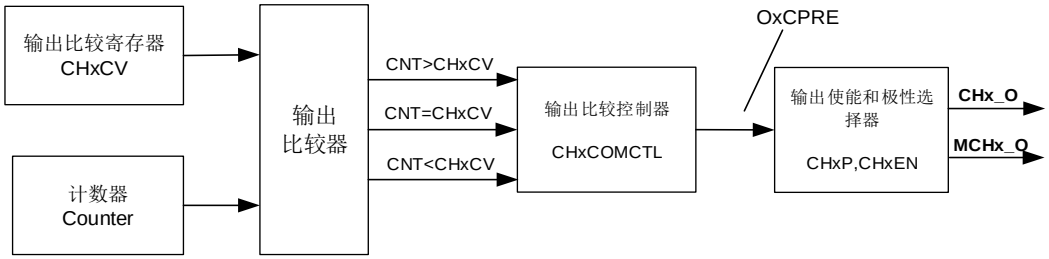
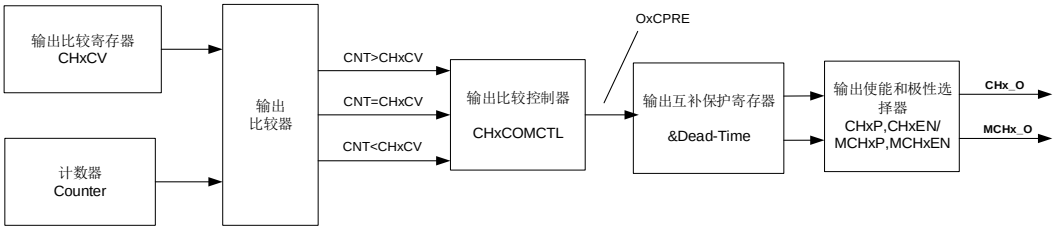


图 18-16. 输出比较原理（当 MCHxMSEL = 2’11 时互补输出，x=0,1,2,3）



通道输出信号CHx_O/MCHx_O与OxCPRE/ MOxCPRE信号（详情请见*通道输出准备信号*）的关系描述如下(OxCPRE/ MOxCPRE信号高电平有效)：

- 当MCHxMSEL=2’b00（TIMERx_CTL2寄存器中），MCHx_O输出与CHx_O输出相互独立。CHx_O 输出电平取决于OxCPRE信号、CHxP位和CHxEN位（详细内容参考TIMERx_CHCTL2寄存器）。MCHx_O输出电平取决于MOxCPRE信号、MCHxFP[1:0]位和MCHxEN位（详细内容参考TIMERx_CHCTL2寄存器）。请参考*图18-14. 输出比较原理（当MCHxMSEL = 2’00，x=0,1,2,3）*。
- 当MCHxMSEL=2’b01，MCHx_O输出和CHx_O输出相同。CHx_O和MCHx_O输出电平取决于OxCPRE信号、CHxP和CHxEN位，对MCHx_O的配置无效。请参考*图18-15. 输出比较原理（当MCHxMSEL = 2’01，x=0,1,2,3）*。
- 当MCHxMSEL=2’b11，MCHx_O输出和CHx_O输出互补。CHx_O/MCHx_O输出电平取决于OxCPRE信号、CHxP/ MCHxP位和CHxEN/ MCHxEN位。请参考*图18-16. 输出比较原理（当MCHxMSEL = 2’11时互补输出，x=0,1,2,3）*。

例如（MCHx_O输出与CHx_O输出相互独立）：

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MCHxP=1（MCHx_O低电平有效，与MOxCPRE输出极性相反）、MCHxEN=1（MCHx_O输出使能）时：

若MOxCPRE输出有效（高）电平，则MCHx_O输出有效（低）电平；

若MOxCPRE输出无效（低）电平，则MCHx_O输出无效（高）电平。

当MCHxMSEL=2'b11，CHx_O和MCHx_O同时输出时，CHx_O和MCHx_O的具体输出情况还与TIMERx_CCHP寄存器中的相关位（ROS、IOS、POEN和DTCFG等位）有关。详情请见*通道输出互补PWM*。

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/ TIMERx_MCHxCV 寄存器与计数器的值匹配时，根据 CHxCOMCTL/ MCHxCOMCTL 的配置，这个通道的输出可以被置高电平，被置低电平或者翻转。当计数器的值与 TIMERx_CHxCV/ TIMERx_MCHxCV 寄存器的值匹配时，CHxIF/ MCHxIF 位被置 1，如果 CHxIE/ MCHxIE = 1 则会产生中断，如果 CHxDEN/ MCHxDEN =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/ MCHxCOMSEN位来配置输出比较影子寄存器；
- 设置CHxCOMCTL/ MCHxCOMCTL位来配置输出模式（置高电平/置低电平/翻转）；
- 设置CHxP/ MCHxP/ MCHxFP位来选择有效电平的极性；
- 设置CHxEN/MCHxEN使能输出。

第三步：通过CHxIE/ MCHxIE/ CHxDEN/ MCHxDEN位配置中断/DMA请求使能。

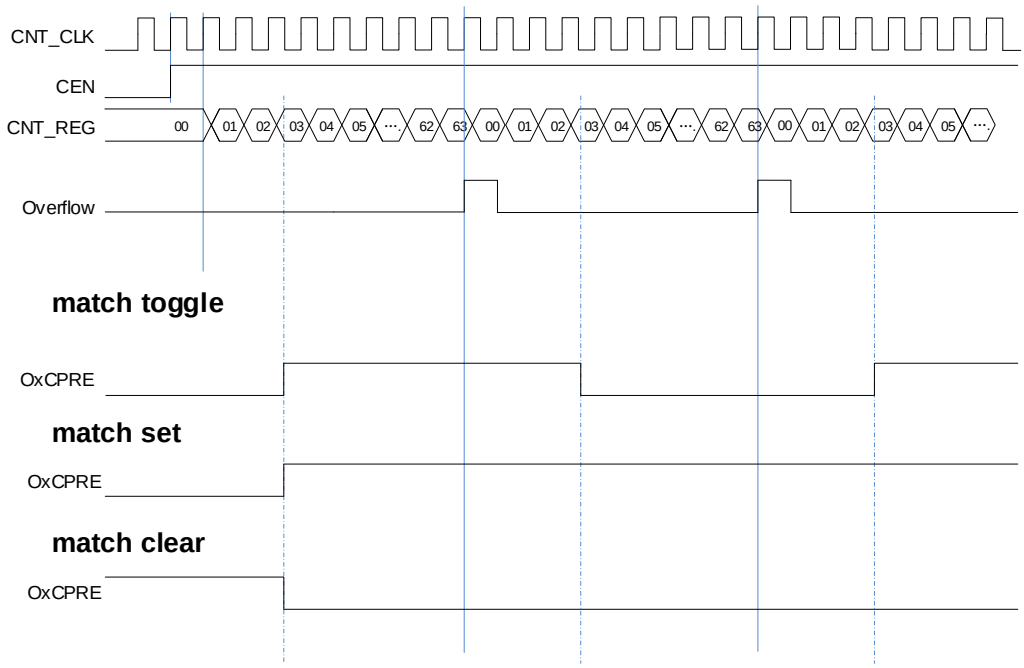
第四步：通过TIMERx_CAR寄存器和TIMERx_CHxCV寄存器配置输出比较时基：

TIMERx_CHxCV/ TIMERx_MCHxCV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18-17. 三种输出比较模式*显示了三种比较输出模式：翻转/置高电平/置低电平，CAR=0x63, CHxVAL=0x3。

图 18-17. 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/ MCHxCOMCTL 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/ TIMERx_MCHxCV 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/ TIMERx_MCHxCV寄存器值决定。*图18-18. 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/ TIMERx_MCHxCV寄存器值）决定。*图18-19. CAPWM时序图*显示了CAPWM的输出波形和中断。

当计数器向上计数时，在PWM0模式下(CHxCOMCTL/ MCHxCOMCTL =3'b110)，如果TIMERx_CHxCV/ TIMERx_MCHxCV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；PWM1模式下(CHxCOMCTL/ MCHxCOMCTL =3'b111)，如果TIMERx_CHxCV/ TIMERx_MCHxCV寄存器的值大于TIMERx_CAR寄存器的值，通道输出一直为无效电平。

图 18-18. EAPWM 时序图

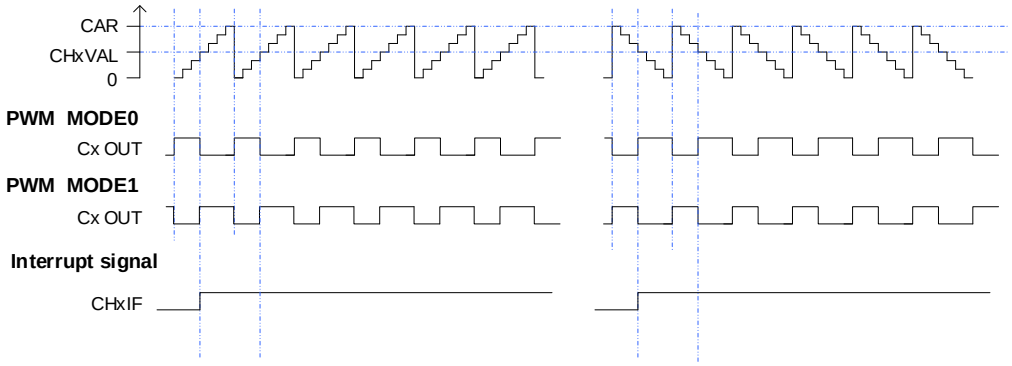
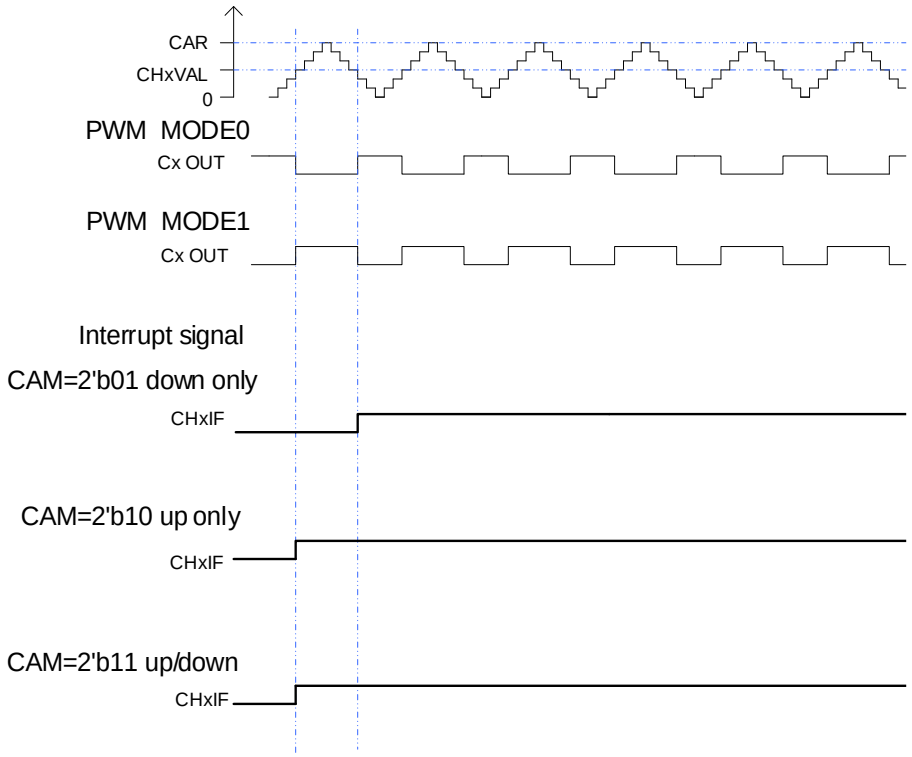


图 18-19. CAPWM 时序图



复合 PWM 模式

在复合 PWM 中（CHxCPWMEN = 1′b1，CHxMS[2:0] = 3′b000 和 CHxCOMCTL=3′b110、3′b111），通道 x(x=0,1,2,3)上的 PWM 输出信号由 CHxVAL 和 CHxCOMVAL_ADD 位确定。

如果 CHxCOMCTL = 3′b110（PWM 模式 0）且 DIR = 1′b0（向上计数模式），或者 CHxCOMCTL = 3′b111（PWM 模式 1）且 DIR = 1′b1（向下计数模式），当计数器和 CHxVAL 的值相匹配时通道 x 输出强制为低。当计数器与 CHxCOMVAL_ADD 的值相匹配时，通道 x 输出强制为高。

如果 CHxCOMCTL = 3′b111（PWM 模式 1）且 DIR = 1′b0（向上计数模式），或者 CHxCOMCTL = 3′b110（PWM 模式 0）且 DIR = 1′b1（向下计数模式），当计数器和 CHxVAL 的值相匹配时通道 x 输出强制为高。当计数器与 CHxCOMVAL_ADD 的值相匹配时，通道 x 输出强制为低。

PWM 的周期取决于（CARL + 0x0001），PWM 脉冲宽度可以下*表 18-3 复合 PWM 脉冲宽度*计算。

表 18-3 复合 PWM 脉冲宽度

条件	模式	PWM 脉冲宽度
CHxVAL < CHxCOMVAL_ADD ≤ CARL	PWM 模式 0	（CARL + 0x0001） + （CHxVAL – CHxCOMVAL_ADD）
	PWM 模式 1	（CHxCOMVAL_ADD – CHxVAL）
CHxCOMVAL_ADD < CHxVAL ≤ CARL	PWM 模式 0	（CHxVAL - CHxCOMVAL_ADD）
	PWM 模式 1	（CARL + 0x0001） + （CHxCOMVAL_ADD – CHxVAL）
（CHxVAL = CHxCOMVAL_ADD ≤	PWM 模式 0（向上计数）	100%

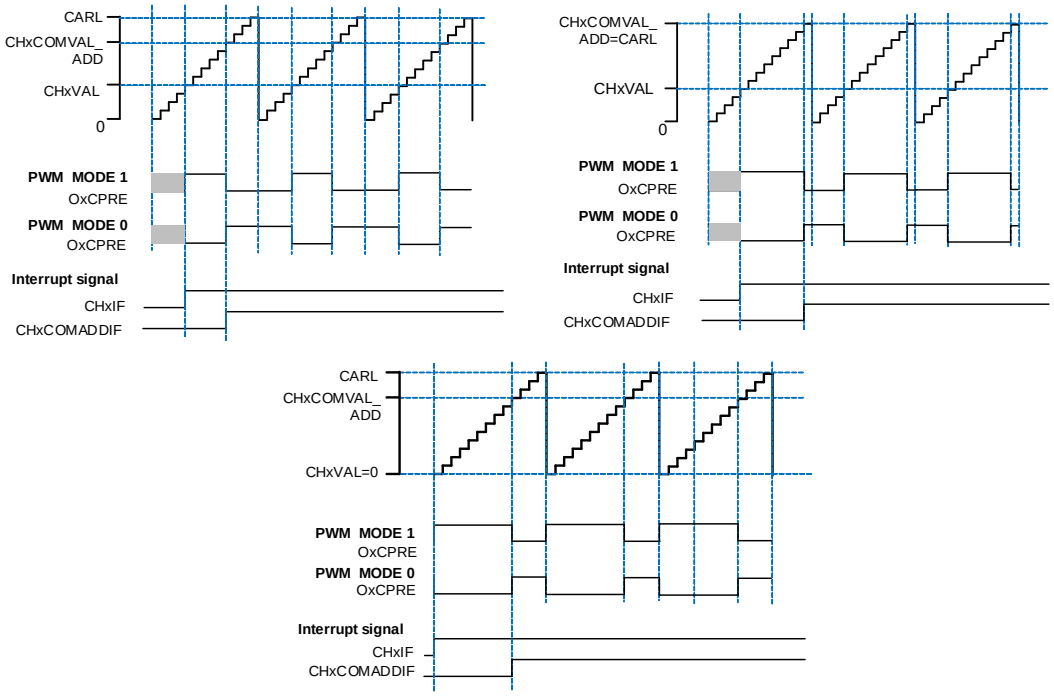
条件	模式	PWM 脉冲宽度
CARL）或 （CHxVAL > CARL > CHxCOMVAL_ADD）	或 PWM 模式 1（向下计数）	
	PWM 模式 0（向下计数） 或 PWM 模式 1（向上计数）	0%
CHxCOMVAL_ADD > CARL > CHxVAL	PWM 模式 0（向上计数） 或 PWM 模式 1（向下计数）	0%
	PWM 模式 0（向下计数） 或 PWM 模式 1（向上计数）	100%
（CHxVAL>CARL）且 （CHxCOMVAL_ADD > CARL）	-	CHx_O 输出保持

当计数器计数到CHxVAL，CHxIF位置1且如果CHxIE=1通道x产生中断，如果CHxDEN=1，则产生DMA请求。当计数器计数到CHxCOMVAL_ADD时，CHxCOMADDIF位置1（该中断标志位只在复合PWM模式有效，CHxCPWMEN=1），如果CHxCOMADDIE = 1通道x附加比较中断产生（只有中断产生，没有DMA请求响应）。

根据CHxVAL，CHxCOMVAL_ADD和CARL之间的关系，可以分为四种情况：

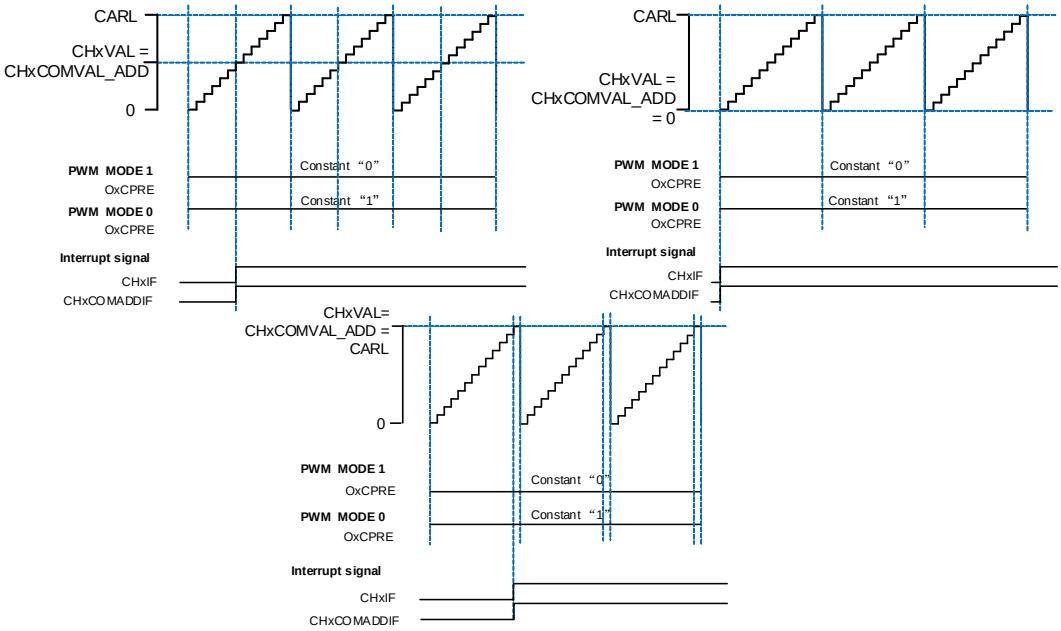
1） CHxVAL < CHxCOMVAL_ADD，CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 18-20 通道 x 输出 PWM（CHxVAL < CHxCOMVAL_ADD）



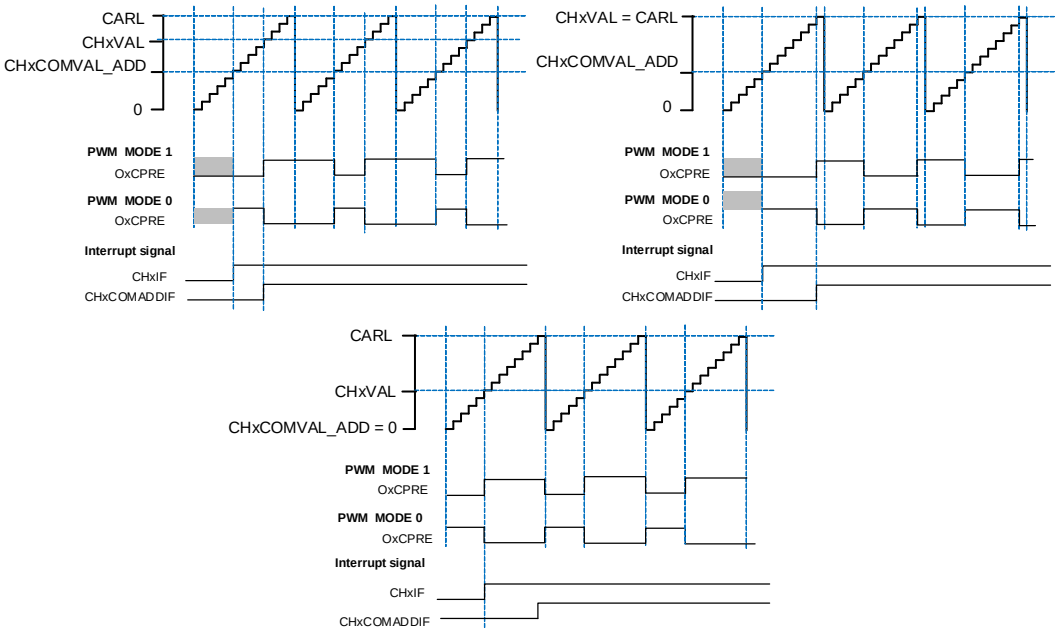
2） CHxVAL = CHxCOMVAL_ADD, CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 18-21 通道 x 输出 PWM（CHxVAL = CHxCOMVAL_ADD）



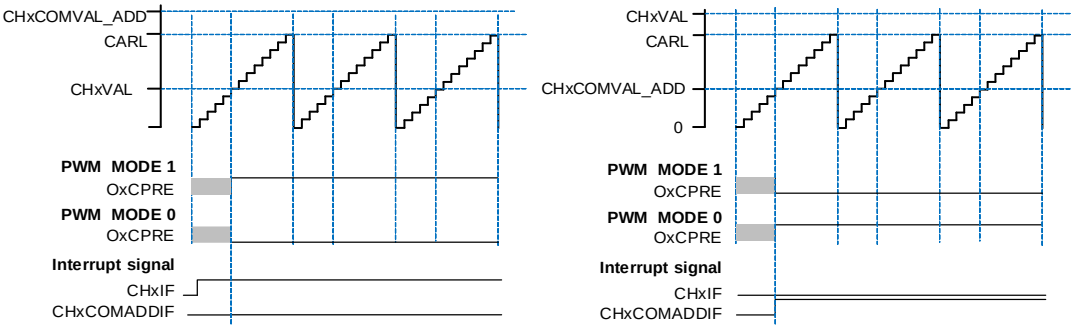
3） CHxVAL > CHxCOMVAL_ADD, CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 18-22. 通道 x 输出 PWM（CHxVAL > CHxCOMVAL_ADD）



4） CHxVAL或CHxCOMVAL_ADD值大于CARL。

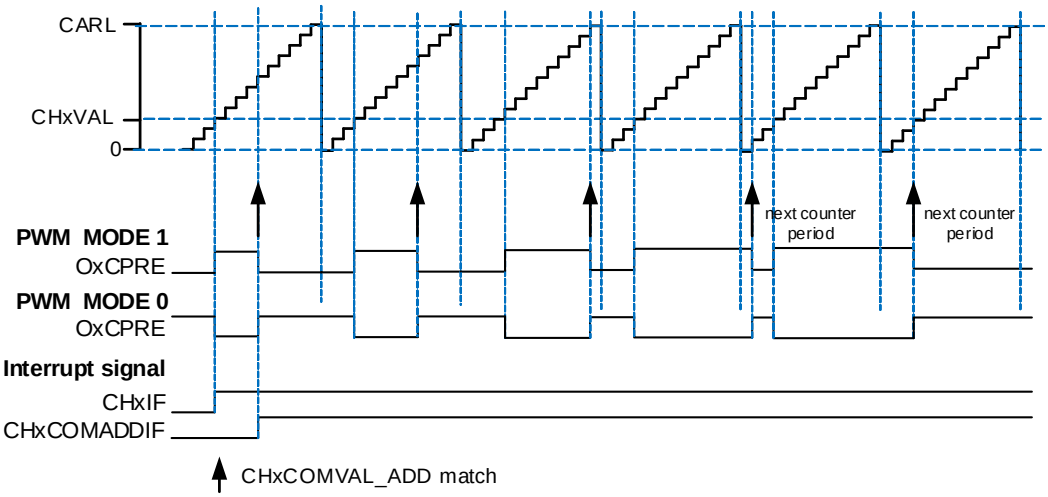
图 18-23. 通道 x 输出 PWM（CHxVAL 或 CHxCOMVAL_ADD > CARL）



复合PWM模式支持不修改周期只修改占空比的PWM信号的生成。*图18-24. 通道x输出PWM占空比随着CHxCOMVAL_ADD值而改变*显示PWM输出和中断波形。

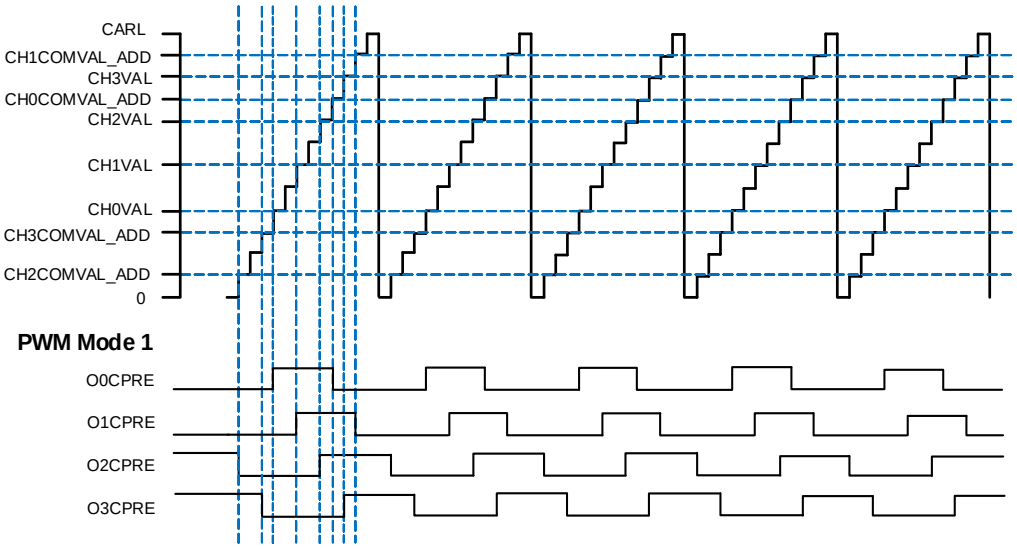
在某些情况下，CHxCOMVAL_ADD的匹配事件可以发生在下一个计数周期（CHxCOMVAL_ADD值在计数器到达CHxVAL值之后被写入，且CHxCOMVAL_ADD值小于或者等于CHxVAL值）。

图 18-24. 通道 x 输出 PWM 占空比随着 CHxCOMVAL_ADD 值而改变



如果多个通道配置为复合PWM模式，可以为每对通道x的匹配边沿设定一个偏移量（相对于其它通道）。这种特性在产生照明PWM控制信号时非常有用，因为在这种情况下，希望彼此边缘不重合，以消除噪声的产生。CHxVAL寄存器值是PWM脉冲相对于计数器周期开始的偏移。

图 18-25. 复合 PWM 模式下四通道输出



输出匹配脉冲选择

当发生匹配事件时，CHx_O（x=0..3）的输出由CHxCOMCTL[2:0]（x=0..3）位设置，通过配置CHxOMPSEL[1:0]（x=0..3）位，可选择CHx_O（x=0..3）的输出信号正常或者脉冲。

当匹配事件发生时，CHxOMPSEL[1:0]（x=0..3）用于选择OxCPRE信号输出（驱动CHx_O）：

- CHxOMPSEL = 2’b00，OxCPRE信号根据CHxCOMCTL[2:0]位的配置正常输出；
- CHxOMPSEL = 2’b01，只有在计数器向上计数，发生匹配事件时，OxCPRE信号输出一个脉冲，且脉冲宽度为一个CK_TIMER时钟周期；
- CHxOMPSEL = 2’b10，只有在计数器向下计数，发生匹配事件时，OxCPRE信号输出一个脉冲，且脉冲宽度为一个CK_TIMER时钟周期；
- CHxOMPSEL = 2’b11，无论计数器向上计数还是向下计数，发生匹配事件时，OxCPRE信号输出一个脉冲，且脉冲宽度为一个CK_TIMER时钟周期；

图 18-26. 边沿对齐模式下 CHx_O 输出脉冲（CHxOMPSEL≠2’b00）

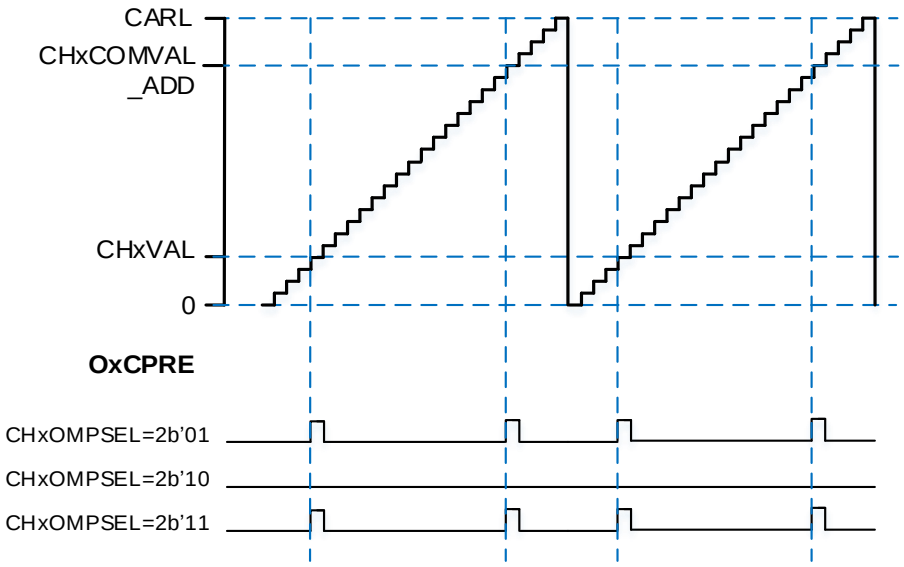
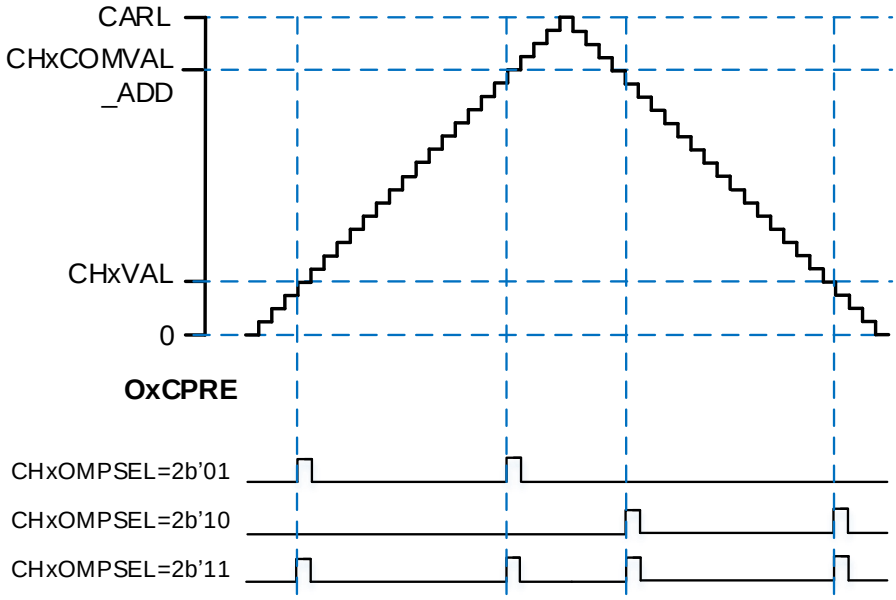


图 18-27. 中央对齐模式下 CHx_O 输出脉冲（CHxOMPSEL≠2'b00）



通道输出准备信号

如*图 18-14. 输出比较原理（当 MCHxMSEL = 2'00，x=0,1,2,3）*和*图 18-16. 输出比较原理（当 MCHxMSEL = 2'11 时互补输出，x=0,1,2,3）*所示，当 TIMERx 用于输出匹配比较模式下，在通道输出信号之前将产生一个中间信号，即 OxCPRE 或 MOxCPRE 信号（通道 x 或其互补输出参考信号）。

OxCPRE 和 MOxCPRE 信号有若干类型的输出功能，通过配置 CHxCOMCTL 位定义 OxCPRE 信号类型，通过配置 MCHxCOMCTL 位定义 MOxCPRE 信号类型。

下面以 OxCPRE 为例进行说明，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 MCHx_O 的输出具有三种情况：

- MCHxMSEL=2'b00：MCHx_O输出独立于CHx_O输出。
- MCHxMSEL=2'b01：MCHx_O 输出与 CHx_O 输出相同，且 MCHx_O 的输出不由

CHxMOMCTL位配置。

- MCHxMSEL=2’b11：MCHx_O 输出与 CHx_O 输出互补，且 MCHx_O 的输出不由 CHxMOMCTL位配置。

当 CHx_O 和 MCHx_O 输出互补时，这两个信号不能同时有效。TIMERx 有 4 对通道，所有 4 对通道都具有此功能。互补信号 CHx_O 和 MCHx_O 是由一组参数来决定：TIMERx_CHCTL2 寄存器中的 CHxEN 和 MCHxEN 位，TIMERx_CCHP 寄存器中和 TIMERx_CTL1 寄存器中的 POEN、ROS、IOS、ISOx 和 ISOxN 位（当 CHx_O 和 MCHx_O 具有独立的死区时间和中止功能时，请参考*表 18-4. 由参数控制的互补输出表(MCHxMSEL =2’b11)*）。输出极性由 TIMERx_CHCTL2 寄存器中的 CHxP 和 MCHxP 位来决定。

表 18-4. 由参数控制的互补输出表(MCHxMSEL =2’b11)

<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>MCHxEN</th><th>CHx_O</th><th>MCHx_O</th></tr>
<tr><td rowspan="5">0</td><td rowspan="5">0/1</td><td rowspan="4">0</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(1)</td></tr>
<tr><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 rowspan="2">1</td><td>0</td></tr>
<tr><td>1</td></tr>
<tr><td>1</td><td>x</td><td>x</td><td colspan="2">CHx_O/CHx_ON输出关闭状态：
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="8">1</td><td rowspan="4">0</td><td rowspan="8">0/1</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O/MCHx_O = LOW
CHx_O/MCHx_O输出禁用</td></tr>
<tr><td>1</td><td>CHx_O = LOW
CHx_O输出禁用</td><td>MCHx_O=OxCPRE⊕(4)MCHxP
MCHx_O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>MCHx_O = LOW
MCHx_O输出禁用.</td></tr>
<tr><td>1</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O输出使能</td><td>MCHx_O=(!OxCPRE) (5) ⊕ MCHxP
MCHx_O输出使能</td></tr>
<tr><td rowspan="4">1</td><td rowspan="2">0</td><td>0</td><td>CHx_O = CHxP
CHx_O输出禁用</td><td>MCHx_O = MCHxP
MCHx_O输出禁用</td></tr>
<tr><td>1</td><td>CHx_O = CHxP
CHx_O输出使能</td><td>MCHx_O=OxCPRE⊕ MCHxP
MCHx_O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>MCHx_O = MCHxP
MCHx_O输出使能</td></tr>
<tr><td>1</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O输出使能</td><td>MCHx_O=(!OxCPRE) ⊕ MCHxP
MCHx_O输出使能</td></tr>
</table>

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 MCHxMSEL=2'b11，CHxEN 和 MCHxEN 为 1'b1，同时设置 POEN=1，就可以使能死区插入功能。DTCFG 位域定义了死区时间，死区时间对所有通道有效。死区时间设置的细节请参考*互补通道保护寄存器(TIMERx_CCHP)*。

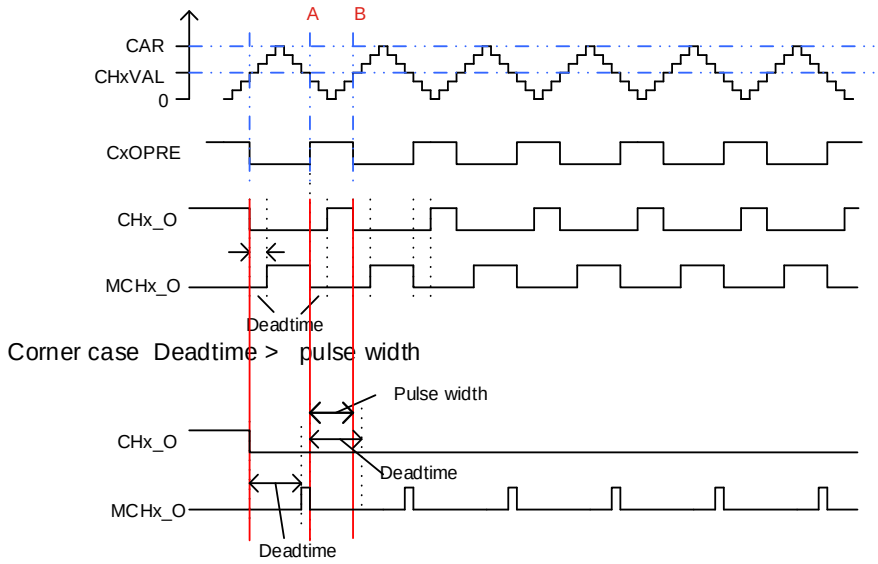
死区时间的插入，确保了通道互补的两路信号不会同时有效。

在 PWM0 模式，当通道 x 匹配发生时（TIMERx 计数器=TIMERx_CHxCV），OxCPRE 翻转。在*图 18-28. 带死区时间的通道互补输出*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 MCHx_O 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，MCHx_O 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_O信号的占空比，CHx_O信号一直为无效值。（如*图18-28. 带死区时间的通道互补输出*所示）

图 18-28. 带死区时间的通道互补输出



CHx_O和MCHx_O通道可以具有独立的死区时间，具体请参考*独立的死区时间插入和中止功能*。

通过配置TIMERx_CTL2寄存器中的DTIENCHx（x = 0..3）位，可实现对每对通道的死区插入功能的独立控制。当DTIENCHx（x = 0..3）位为“0”时，相应的通道CHx_O和MCHx_O将不会插入死区。

中止模式

使用中止功能时，CHx_O 和 MCHx_O 信号的输出电平由以下位控制：TIMERx_CCHP 寄存器的 POEN、IOS 和 ROS 位，TIMERx_CTL1 寄存器的 ISOx 和 ISOxN 位。

当 MCHxMSEL = 2’b11（MCHx_O 的输出不使用 CHxMOMCTL 位配置）时，MCHx_O 输出与 CHx_O 输出互补。在这种情况下，CHx_O 和 MCHx_O 信号不能同时设置为有效电平。中止源是系统事件和其他事件。当发生中止事件时，在死区持续时间之后，输出将以预定的电平（有效或无效）强制输出。中止功能通过将 TIMERx_CCHP 寄存器中的 BRKEN 位置 1 来使能。中止输入极性由 TIMERx_CCHP 寄存器中的 BRKP 位配置。

中止事件是所有源逻辑或运算的结果。中止功能可以处理两种类型的事件源：

- 外部信号源：来自BRKINx(x=0..3)输入，并带有数字滤波器和极性选择；
- 内部信号源：系统源（例如：由RCU中的时钟监视器CKM生成的HXTAL卡住事件，LVD锁定事件，Cortex®-M33锁定事件或SRAM ECC校验错误事件）和片上比较器事件（在TRIGSEL中配置，由BRKIN0输入）。

中止事件也可以由软件置位TIMERx_SWEVG寄存器中的BRKG位产生。

图 18-29 中止功能图

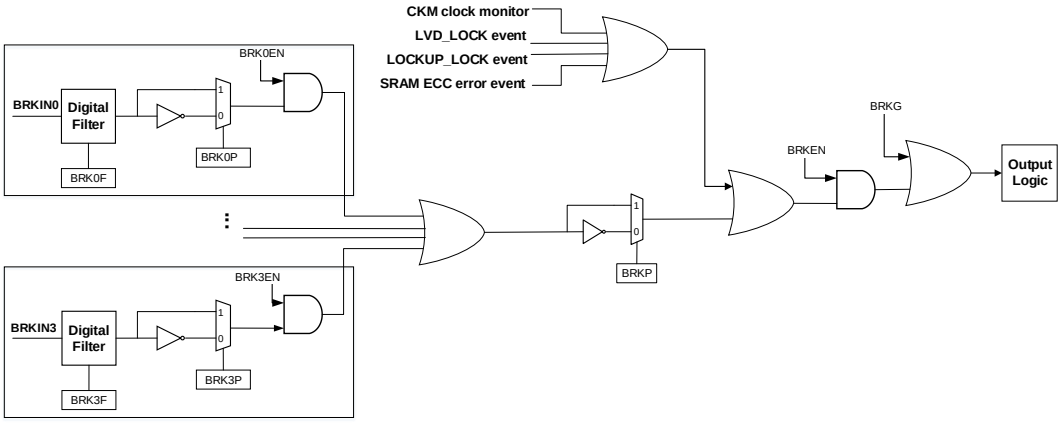
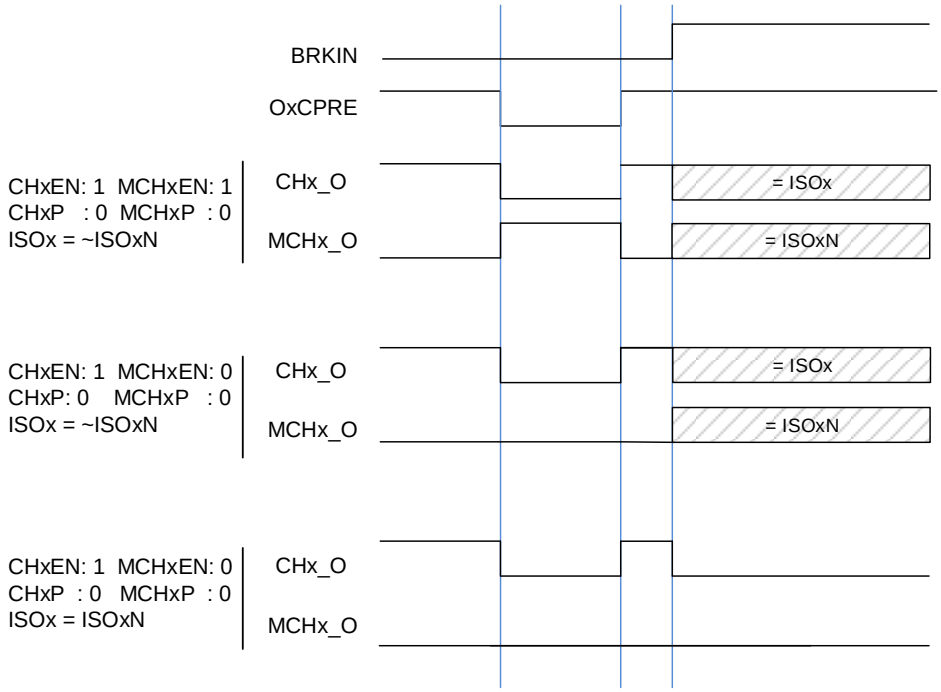


*图 18-29 中止功能图*中的 BRKINx(x=0..3)可以从 TRIGSEL 模块或者 GPIO 引脚选择通过 ***TIMER 输入源选择寄存器（SYSCFG_TIMERINSEL）***。

当 MCHxMSEL = 2’b11 且发生中止事件时，POEN 位被异步清除，一旦 POEN 位为 0，CHx_O 和 MCHx_O 的输出由 TIMERx_CTL1 寄存器中的 ISOx 位和 ISOxN 位确定。如果 IOS=0，定时器释放输出使能，否则输出使能仍然为高。首先通道互补输出为复位状态，然后死区时间发生器重新被激活，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。

发生中止事件时，TIMERx_INTF 寄存器的 BRKIF 位被置 1。如果 BRKIE=1，中断产生。

图 18-30. 通道响应中止输入（高电平有效）时，输出信号的行为



CHx_O和MCHx_O通道可以具有独立中止功能时，请参考*独立的死区时间插入和中止功能*。

通过配置TIMERx_CTL2寄存器中的BEKENCHx（x = 0..3）位，可实现对每对通道的中止功能进行独立控制。当BEKENCHx（x = 0..3）位为“0”且发生中止事件时，相应的通道CHx_O和MCHx_O输出保持不变。

独立的死区时间插入和中止功能

CHx_O和MCHx_O具有独立的死区时间插入和中止功能，允许每对通道具有自己的死区时间和中止功能。在此功能中，CHx_O和MCHx_O实际上由TIMERx_FCCHPy（y = 0..3）寄存器中的IOS位、ROS位和DTCFG[7:0]控制。

通过配置TIMERx_FCCHPy(y=0..3)寄存器中的FCCHPyEN位，可以选择每对通道是否采用独立的死区时间插入和中止功能控制：当FCCHPyEN=0时，TIMERx_CCHP寄存器中的ROS、IOS和DTCFG[7:0]有效；当FCCHPyEN=1时，TIMERx_FCCHPy寄存器中的ROS、IOS和DTCFG[7:0]有效，使能独立的死区时间插入和中止功能。

正交译码器

正交译码器功能使用由TIMERx_CH0和TIMERx_CH1引脚生成的CI0FE0和CI1FE1正交信号各自相互作用产生计数值。在每个输入源改变期间，DIR位会发生改变。输入源可以是只有CI0FE0，可以只有CI1FE1，或着可以同时有CI0FE0和CI1FE1，通过设置SMC=0x01，0x02或0x03来选择使用哪种模式。计数器计数方向改变的机制*表18-5. 不同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 18-5. 不同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 18-31. 在正交译码器模式 2 且 CI0FE0 极性不反相时计数器行为

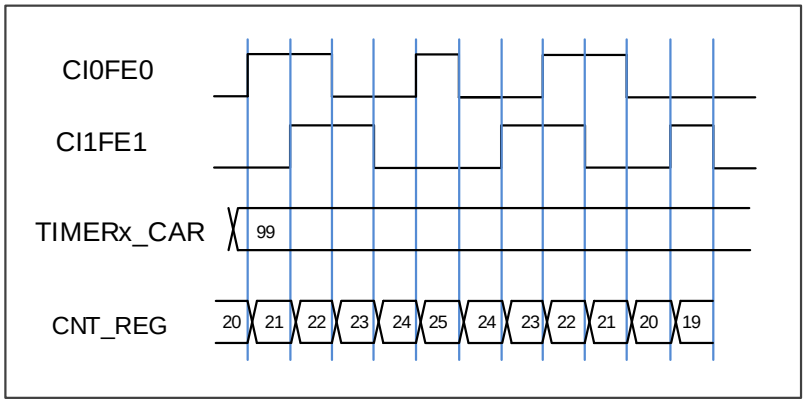
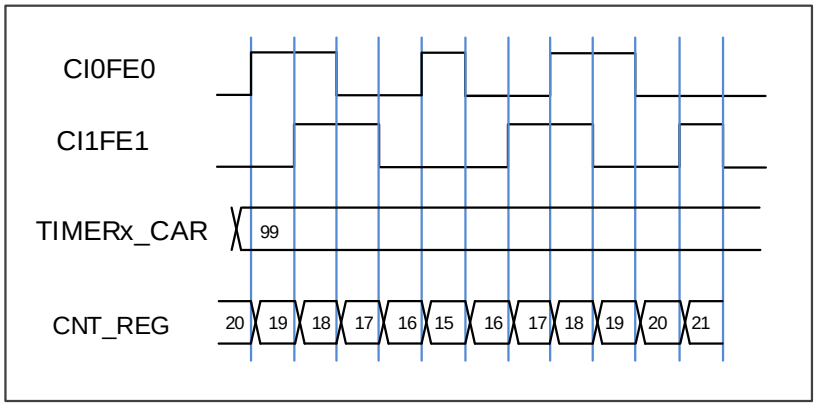


图 18-32. 在正交译码器模式 2 且 CI0FE0 极性反相时计数器行为



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 18-33. 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。TIMER_in 定时器（可以是高级定时器或者通用 L0 定时器）接收来自电机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，可以通过 TRIGSEL 模块，选择互连的定时器，例如：

TIMER_in (TIMER0) -> TIMER_out (TIMER7 ITI0)

TIMER_in (TIMER1) -> TIMER_out (TIMER0 ITI1)

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都会翻转，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 18-33. 霍尔传感器用在 BLDC 电机控制中

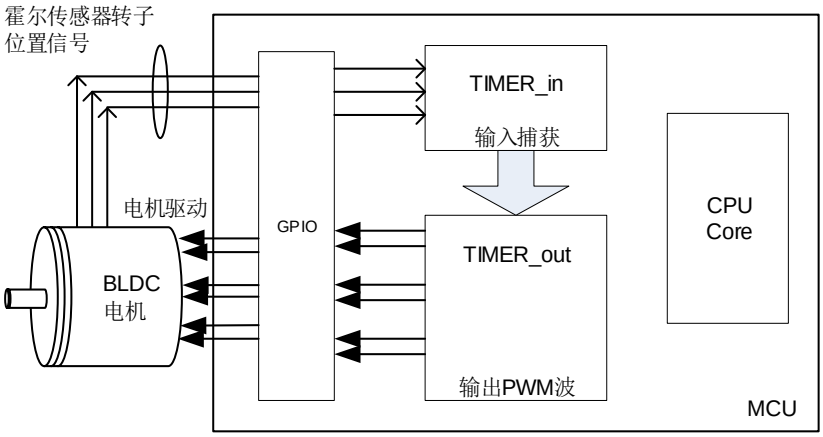
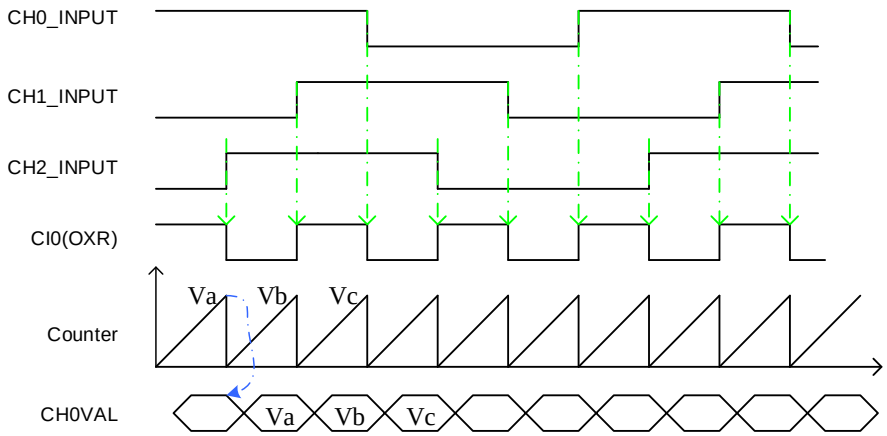
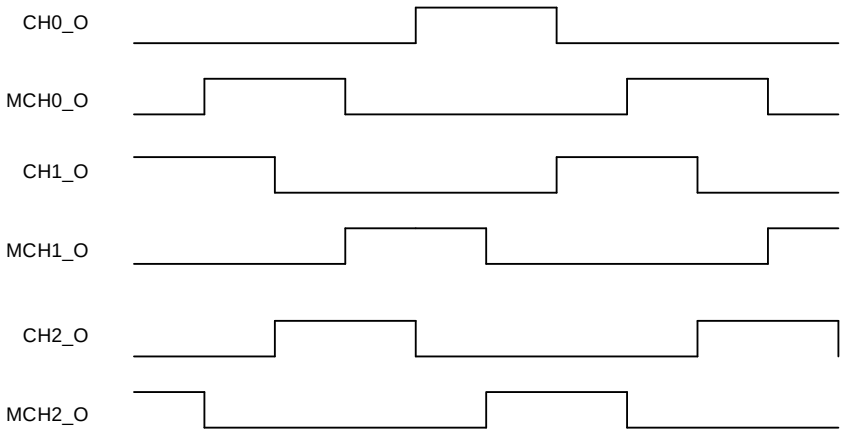


图 18-34. 两个定时器之间的霍尔传感器时序图

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 [3:0]来选择。

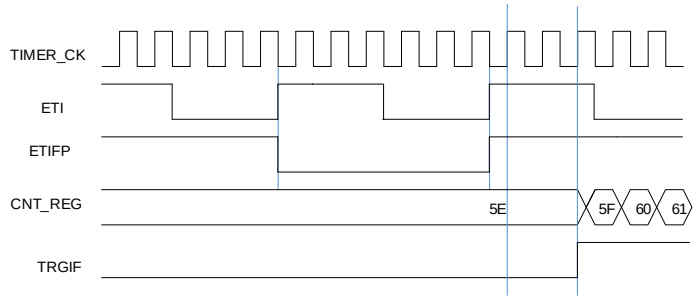
表 18-6. 从模式例子列表

	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[3:0] 0000：ITI0 0001：ITI1 0010：ITI2 0011：ITI3 0100：CI0F_ED 0101：CI0FE0 0110：CI1FE1 0111：ETIFP(1)	如果触发源是CI0FE0或者CI1FE1，配置CHxP和MCHxP来选择极性和反相。 如果触发源是ETIFP（滤波后的外部触发输入ETI），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/ MCIx，配置CHxCAPFLT设置滤波，分频不可用 触发源是ETIFP，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
		1000：CI2FE2 1001：CI3FE3 1010：MCI0FEM0 1011：MCI1FEM1 1100：MCI2FEM2 1101：MCI3FEM3		
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 18-35. 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	TI0S=0（非异或） [MCH0P=0,CH0P=0] CI0FE0不反相，在上升沿捕获	在这个例子中滤波被旁路
	图 18-36. 暂停模式下的控制电路			
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 18-37. 事件模式下的控制电路



(1) ETI 引脚可从 TIMER_ETIx(x=0..2)引脚中任选一个，每个高级定时器（TIMER0/7/19/20）只能使用 1 个，具体请参考 ***TIMER 输入源选择寄存器（SYSCFG_TIMERINSEL）***。

单脉冲模式

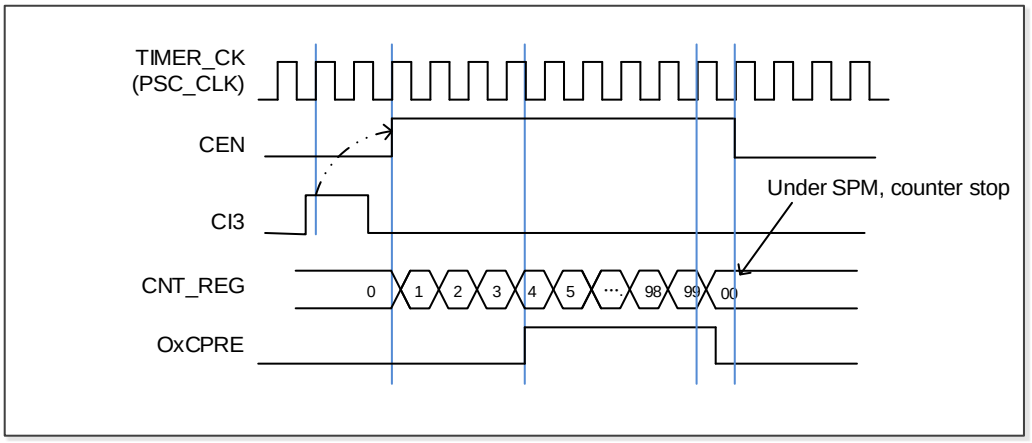
单脉冲模式与重复模式是相反的，设置 TIMERx_CTL0 寄存器的 SPM 位置 1，可使能单脉冲模式。当 SPM 置 1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置 CHxCOMCTL/ MCHxCOMCTL 配置 TIMERx 为 PWM 模式或者比较模式。

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。单脉冲模式下，触发上升沿产生之后，OxCPRE/ MOxCPRE 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。

单脉冲模式也同样适用于复合 PWM 模式（CHxCPWMEN = 1’b1 和 CHxMS[2:0] = 3’b000）

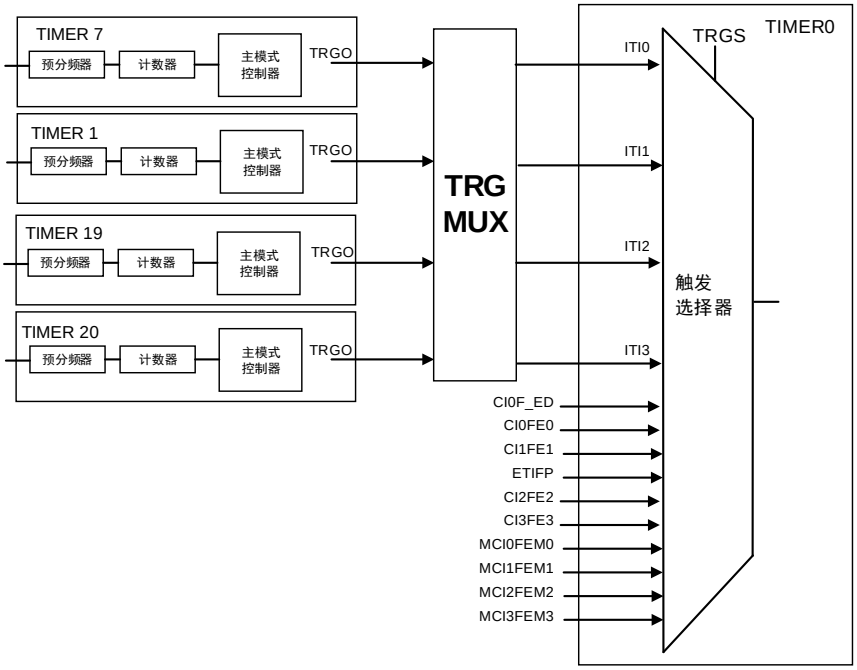
图 18-38. 单脉冲模式，TIMERx_CHxCV = 0x04 TIMERx_CAR=0x99



定时器互连

定时器之间可以内部级联或者同步，通过配置一个定时器工作在主模式另一个定时器工作在从模式来实现。*图 18-39. 定时器 0 主/从模式的例子*显示了一些主从模式触发选择的例子。

图 18-39. 定时器 0 主/从模式的例子



其他互连的例子：

■ 定时器1作为定时器0的预分频器

参考*图 18-39. 定时器 0 主/从模式的例子*连接配置定时器 1 为定时器 0 的预分频器，步骤如下：

1. 配置定时器1为主模式，选择其更新事件（UPE）为触发输出（配置TIMER1_CTL1寄存器的MMC=3’b010）。定时器1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1周期（TIMER1_CAR寄存器）；
3. 选择定时器0输入触发源为定时器1（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1（TIMER1_CTL0寄存器）。

■ 用定时器1的使能信号来启动定时器0

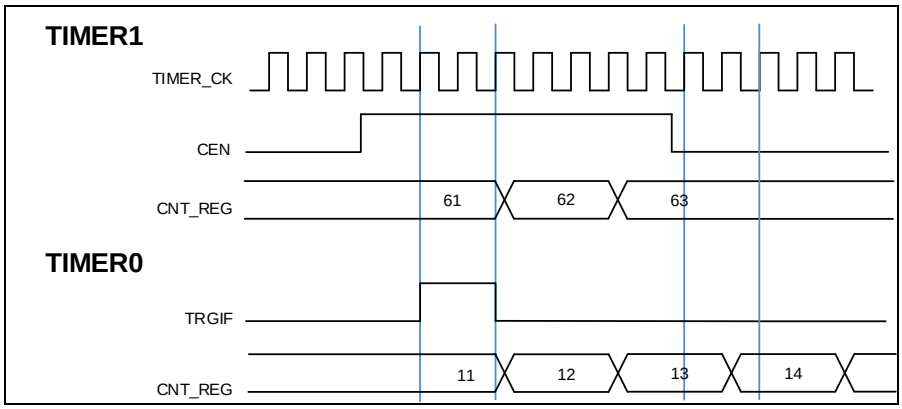
用定时器 1 的使能信号来启动定时器 0，见*图 18-40. 用定时器 1 的使能信号触发定时器 0*。在定时器 1 使能信号输出后，定时器 0 按照分频后的内部时钟从当前值开始计数。

当定时器 0 接收到触发信号，它的 CEN 位置 1，计数器计数直到禁能定时器 0。两个定时器的计数器频率都是 TIMER_CK 经过预分频器 3 分频后频率（f_{CNT_CLK} = f_{TIMER_CK}/3）。步骤如下：

1. 配置定时器1为主模式，发送它的使能信号作为触发输出（配置TIMER1_CTL1寄存器的MMC=3’b001）；
2. 配置定时器0选择输入触发来自定时器1（配置TIMERx_SMCFG寄存器的TRGS=3’b010）；

3. 配置定时器0在事件模式（配置TIMERx_SMCFG寄存器的SMC=3’b110）；
4. 写1到CEN来开启定时器1（TIMER1_CTL0寄存器）。

图 18-40. 用定时器 1 的使能信号触发定时器 0



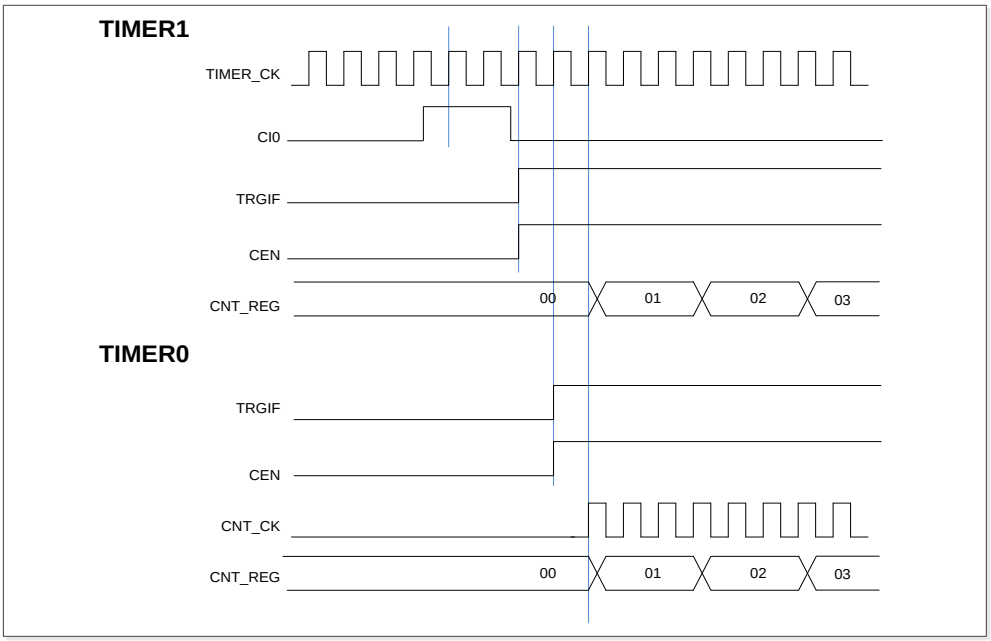
■ 使用一个外部触发来同步两个定时器

配置定时器1的使能信号触发定时器0的开启，配置定时器1的CI0输入信号上升沿来触发定时器1。为了确保两个定时器同步开启，定时器1必须配置在主/从模式。步骤如下：

1. 配置定时器1工作在从模式来获取来自CI0的触发输入CI0F_ED（配置TIMER1_SMCFG寄存器的TRGS=3’b100）；
2. 配置定时器1工作在事件模式（配置TIMER1_SMCFG寄存器的SMC=3’b110）；
3. 写MSM=1（TIMER1_SMCFG寄存器）来配置定时器1工作在主/从模式；
4. 配置定时器0的触发输入来自定时器1（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1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 18-41. 用定时器 1 的 CI0 输入来触发定时器 0 和定时器 1



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。通过使能 DMA 请求，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_CTL寄存器中的TIMERx_HOLD位置1时，定时器的计数器停止计数。

18.1.5. TIMERx 寄存器（x=0,7,19,20)

TIMER0基地址：0x4001 2C00

TIMER7基地址：0x4001 3400

TIMER19基地址：0x4001 5000

TIMER20基地址：0x4001 5400

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	自动重载影子寄存器使能 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模式或译码器模式时，该位为只读。
3	SPM	单脉冲模式 0：更新事件发生后，计数器继续计数 1：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，如果 UPG 位被置 1 或者复位模式不会产生更新事件，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

ISO3N	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		rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	ISO3N	多模式通道 3 的互补输出空闲状态 参考 ISO0N 位。
14	ISO3	通道 3 的空闲状态输出 参考 ISO0 位。
13	ISO2N	多模式通道 2 的互补输出空闲状态 参考 ISO0N 位。
12	ISO2	通道 2 的空闲状态输出 参考 ISO0 位。
11	ISO1N	多模式通道 1 的互补输出空闲状态 参考 ISO0N 位。
10	ISO1	通道 1 的空闲状态输出 参考 ISO0 位。
9	ISO0N	多模式通道 0 的互补输出空闲状态 0：当 POEN 复位，MCH0_O 输出低电平 1：当 POEN 复位，MCH0_O 输出高电平 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 输出低电平 1：当 POEN 复位，CH0_O 输出高电平 如果 MCH0_O 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 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	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、MCHxEN 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	换相控制影子寄存器使能 0：影子寄存器 CHxEN、MCHxEN 和 CHxCOMCTL 位禁能 1：影子寄存器 CHxEN、MCHxEN 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
TRGS[3]	保留														
rw															

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	TRGS[3]	触发选择 参考 TRGS[2:0]描述。
30:16	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 同时被配置，外部时钟的输入是 ETIFP。

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 ETI 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 ETIP 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	滤波禁能	
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。 0000：内部触发输入 0（ITI0） 0001：内部触发输入 1（ITI1） 0010：内部触发输入 2（ITI2） 0011：内部触发输入 3（ITI3） 0100：CI0 的边沿标志位（CI0F_ED） 0101：滤波后的通道 0 输入（CI0FE0） 0110：滤波后的通道 1 输入（CI1FE1） 0111：滤波后的外部触发输入（ETIFP） 1000：滤波后的通道 2 输入（CI2FE2） 1001：滤波后的通道 3 输入（CI3FE3） 1010：滤波后的多模式通道 0 输入（MCI0FEM0） 1011：滤波后的多模式通道 1 输入（MCI1FEM1） 1100：滤波后的多模式通道 2 输入（MCI2FEM2） 1101：滤波后的多模式通道 3 输入（MCI3FEM3） 1110：保留 1111：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：正交译码模式 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
CH3COM	CH2COM	CH1COM	CH0COM	MCH3	MCH2	MCH1	MCH0	MCH3IE	MCH2IE	MCH1IE	MCH0IE	保留			

ADDIE	ADDIE	ADDIE	ADDIE	DEN	DEN	DEN	DEN								
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	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	CH3COMADDIE	通道 3 附加比较中断使能 0：禁止通道 3 附加比较中断 1：使能通道 3 附加比较中断 **注意**：此中断使能位仅用于复合 PWM 模式（CH3CPWMEN=1，CH3MS[2:0] = 3’b000 和 CH3COMCTL=3’b110、3’b111）。
30	CH2COMADDIE	通道 2 附加比较中断使能 0：禁止通道 2 附加比较中断 1：使能通道 2 附加比较中断 **注意**：此中断使能位仅用于复合 PWM 模式（CH2CPWMEN=1，CH2MS[2:0] = 3’b000 和 CH2COMCTL=3’b110、3’b111）。
29	CH1COMADDIE	通道 1 附加比较中断使能 0：禁止通道 1 附加比较中断 1：使能通道 1 附加比较中断 **注意**：此中断使能位仅用于复合 PWM 模式（CH1CPWMEN=1，CH1MS[2:0] = 3’b000 和 CH1COMCTL=3’b110、3’b111）。
28	CH0COMADDIE	通道 0 附加比较中断使能 0：禁止通道 0 附加比较中断 1：使能通道 0 附加比较中断 **注意**：此中断使能位仅用于复合 PWM 模式（CH0CPWMEN=1，CH0MS[2:0] = 3’b000 和 CH0COMCTL=3’b110、3’b111）。
27	MCH3DEN	多模式通道 3 比较/捕获 DMA 请求使能 0：禁止多模式通道 3 比较/捕获 DMA 请求 1：使能多模式通道 3 比较/捕获 DMA 请求 **注意**：此 DMA 使能位仅用于多模式通道输入和输出独立模式（当 MCH3MSEL[1:0] = 2b’00 时）。
26	MCH2DEN	多模式通道 2 比较/捕获 DMA 请求使能 0：禁止多模式通道 2 比较/捕获 DMA 请求 1：使能多模式通道 2 比较/捕获 DMA 请求 **注意**：此 DMA 使能位仅用于多模式通道输入和输出独立模式（当 MCH2MSEL[1:0] = 2b’00 时）。
25	MCH1DEN	多模式通道 1 比较/捕获 DMA 请求使能 0：禁止多模式通道 1 比较/捕获 DMA 请求 1：使能多模式通道 1 比较/捕获 DMA 请求 **注意**：此 DMA 使能位仅用于多模式通道输入和输出独立模式（当 MCH1MSEL[1:0]

		= 2b’00 时）。
24	MCH0DEN	多模式通道 0 比较/捕获 DMA 请求使能 0：禁止多模式通道 0 比较/捕获 DMA 请求 1：使能多模式通道 0 比较/捕获 DMA 请求 **注意**：此 DMA 使能位仅用于多模式通道输入和输出独立模式（当 MCH0MSEL[1:0] = 2b’00 时）。
23	MCH3IE	多模式通道 3 比较/捕获中断使能 0：禁止多模式通道 3 中断 1：使能多模式通道 3 中断 **注意**：此中断使能位仅用于多模式通道输入和输出独立模式（当 MCH3MSEL[1:0] = 2b’00 时）。
22	MCH2IE	多模式通道 2 比较/捕获中断使能 0：禁止多模式通道 2 中断 1：使能多模式通道 2 中断 **注意**：此中断使能位仅用于多模式通道输入和输出独立模式（当 MCH2MSEL[1:0] = 2b’00 时）。
21	MCH1IE	多模式通道 1 比较/捕获中断使能 0：禁止多模式通道 1 中断 1：使能多模式通道 1 中断 **注意**：此中断使能位仅用于多模式通道输入和输出独立模式（当 MCH1MSEL[1:0] = 2b’00 时）。
20	MCH0IE	多模式通道 0 比较/捕获中断使能 0：禁止多模式通道 0 中断 1：使能多模式通道 0 中断 **注意**：此中断使能位仅用于多模式通道输入和输出独立模式（当 MCH0MSEL[1:0] = 2b’00 时）。
19:15	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
CH3COMADDIF	CH2COMADDIF	CH1COMADDIF	CH0COMADDIF	MCH3OF	MCH2OF	MCH1OF	MCH0OF	MCH3IF	MCH2IF	MCH1IF	MCH0IF	保留			
rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0				

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	CH3COMADDIF	通道 3 附加比较中断标志 参见 CH0COMADDIF 描述。
30	CH2COMADDIF	通道 2 附加比较中断标志 参见 CH0COMADDIF 描述。
29	CH1COMADDIF	通道 1 附加比较中断标志 参见 CH0COMADDIF 描述。
28	CH0COMADDIF	通道 0 附加比较中断标志 此标志由硬件置 1 软件清 0。当通道 0 用于输出模式时，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 **注意**：此标志仅用于复合 PWM 模式（CH0CPWMEN=1，CH0MS[2:0] = 3’b000 和 CH0COMCTL=3’b110、3’b111）。
27	MCH3OF	多模式通道 3 捕获溢出标志 参见 MCH0OF 描述。
26	MCH2OF	多模式通道 2 捕获溢出标志 参见 MCH0OF 描述。
25	MCH1OF	多模式通道 1 捕获溢出标志 参见 MCH0OF 描述。
24	MCH0OF	多模式通道 0 捕获溢出标志 当通道 0 被配置为输入模式时，在 MCH0IF 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：捕获溢出中断发生
23	MCH3IF	多模式通道 3 比较/捕获中断标志 参见 MCH0IF 描述。
22	MCH2IF	多模式通道 2 比较/捕获中断标志 参见 MCH0IF 描述。
21	MCH1IF	多模式通道 1 比较/捕获中断标志 参见 MCH0IF 描述。

20	MCH0IF	多模式通道 0 比较/捕获中断标志 此标志由硬件置 1 软件清 0。当多模式通道 0 用于输入模式时，捕获事件发生时此标志位置 1；当多模式通道 0 用于输出模式时，此标志位在一个比较事件发生时置 1。 当多模式通道 0 在输入模式下时，通过读 TIMERx_MCH0CV 寄存器可以清零该位。 0：无多模式通道 0 中断发生 1：多模式通道 0 中断发生
19:13	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：捕获溢出中断发生
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。 当通道 0 在输入模式下时，通过读 TIMERx_CH0CV 寄存器可以清零该位。 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
CH3COMADDG	CH2COMADDG	CH1COMADDG	CH0COMADDG	保留				MCH3G	MCH2G	MCH1G	MCH0G	保留			
w	w	w	w					w	w	w	w				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BRKG	TRGG	CMTG	CH3G	CH2G	CH1G	CH0G	UPG
								w	w	w	w	w	w	w	w

位/位域	名称	描述
31	CH3COMADDG	通道 3 附加比较事件发生 参见 CH0COMADDG 描述。
30	CH2COMADDG	通道 2 附加比较事件发生 参见 CH0COMADDG 描述。
29	CH1COMADDG	通道 1 附加比较事件发生 参见 CH0COMADDG 描述。
28	CH0COMADDG	通道 0 附加比较事件发生 该位由软件置 1，用于在通道 0 产生一个比较事件，由硬件自动清 0。当此位被置 1，CH0COMADDIF 标志位被置 1，若开启对应的中断和 DMA，则发出相应的中断请求。 0：不产生通道 0 附加比较事件 1：发生通道 0 附加比较事件 **注意**：此位仅用于复合 PWM 模式（CH0CPWMEN=1，CH0MS[2:0] = 3’b000

		和 CH0COMCTL=3'b110、3'b111）。
27:24	保留	必须保持复位值。
23	MCH3G	多模式通道 3 捕获或比较事件发生 参见 MCH0G 描述。
22	MCH2G	多模式通道 2 捕获或比较事件发生 参见 MCH0G 描述。
21	MCH1G	多模式通道 1 捕获或比较事件发生 参见 MCH0G 描述。
20	MCH0G	多模式通道 0 互补捕获或比较事件发生 该位由软件置 1，用于在多模式通道 0 产生一个捕获/比较事件，由硬件自动清 0。当此位被置 1，MCH0IF 标志位被置 1，若开启相应的中断和 DMA，则发出相应的中断和 DMA 请求。此外，如果多模式通道 0 配置为输入模式，计数器的当前值被 TIMERx_MCH0CV 寄存器捕获，如果 MCH0IF 标志位已经为 1，则 MCH0OF 标志位被置 1。 0：不产生多模式通道 0 捕获或比较事件 1：发生多模式通道 0 捕获或比较事件
19:8	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、MCHxEN 和 CHxCOMCTL 位）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
CH1MS[2]	CH0MS[2]	CH1COMADDSEN	CH0COMADDSEN	保留											
		保留	保留												
rw	rw	rw	rw												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COMCEN	CH1COMCTL[2:0]			CH1COMSEN	保留	CH1MS[1:0]		CH0COMCEN	CH0COMCTL[2:0]			CH0COMSEN	保留	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	CH1MS[2]	通道 1 I/O 模式选择 参考 CH1MS[1:0]描述。
30	CH0MS[2]	通道 0 I/O 模式选择 参考 CH0MS[1:0]描述。
29	CH1COMADDSEN	通道 1 附加输出比较影子寄存器使能 参考 CH0COMADDSEN 描述。
28	CH0COMADDSEN	通道 0 附加输出比较影子寄存器使能 当此位被置 1，TIMERx_CH0COMV_ADD 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 附加比较输出影子寄存器 1：使能通道 0 附加比较输出影子寄存器 仅在单脉冲模式下（TIMERx_CTL0 寄存器的 SPM =1），可以在未确认预装载寄存器情况下使用 PWM 模式。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =000 时此位不能被改变。
27:16	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	保留	必须保持复位值。
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（当 MCH1MSEL[1:0] = 2b’00 时，TIMERx_CHCTL2 寄存器的 CH1EN 位清 0；当 MCH1MSEL[1:0] = 2b’01 或 2b’11 时，TIMERx_CHCTL2 寄存器的 CH1EN、MCH1EN 位清 0）时，这些位才可以写。 000：通道 1 配置为输出 001：通道 1 配置为输入，IS1 映射在 CI1FE1 上 010：通道 1 配置为输入，IS1 映射在 CI0FE1 上 011：通道 1 配置为输入，IS1 映射在 ITS 上，此模式仅工作在内部触发源输入被选中时（由 TIMERx_SMCFGFG 寄存器的 TRGS 位选择）。 100：通道1配置为输入，IS1映射在MCI1FE1上 101~111：保留
7	CH0COMCEN	通道 0 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，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 的动作，而 O0CPRE 决定了 CH0_O 的值。O0CPRE 高电平有效，而 CH0_O 的有效电平取决于 CH0P 位。 **注意**：当多模式通道 0 配置为输出模式，且 MCH0MSEL[1:0] = 2b’11 时，此位定义了输出准备信号 O0CPRE 的动作，而 O0CPRE 决定了 CH0_O、MCH0_O 的值。O0CPRE 高电平有效，CH0_O、MCH0_O 的有效电平取决于 CH0P、MCH0P 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 为有效电平，否则为无效电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为无效电平，否则为有效电平。 111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为无效电平，否则为有效电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为有效电平，否则为无效电平。 **注意**：在复合 PWM 模式下（CH0CPWMEN = 1’b1 和 CH0MS = 3’b000），通道 0 的 PWM 输出信号由 TIMERx_CH0CV 和 TIMERx_CH0COMV_ADD 寄存器共同确定。详细信息请参考***复合 PWM 模式***。 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 =000（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情况下使用 PWM 模式。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =000 时此位不能被改变。
2	保留	必须保持复位值。
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（当 MCH0MSEL[1:0] = 2b’00 时，TIMERx_CHCTL2 寄存器的 CH1EN 位清 0；当 MCH0MSEL[1:0] = 2b’01 或 2b’11 时，TIMERx_CHCTL2 寄存器的 CH0EN、MCH0EN 位清 0）时，CH0MS[2:0]才可写。 000：通道 0 配置为输出 001：通道 0 配置为输入，IS0 映射在 CI0FE0 上 010：通道 0 配置为输入，IS0 映射在 CI1FE0 上 011：通道 0 配置为输入，IS0 映射在 ITS 上。此模式仅工作在内部触发输入被选中时（通过设置 TIMERx_SMCFGFG 寄存器的 TRGS 位）。 100：通道 0 配置为输入，IS0 映射在 MCI0FE0 上 101~111：保留

输入捕获模式：

位/位域	名称	描述

31 CH1MS[2] 通道 1 模式选择
与输出模式相同。

30 CH0MS[2] 通道 0 模式选择
与输出模式相同。

29:16 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：输入捕获预分频器系数为 2
10：输入捕获预分频器系数为 4
11：输入捕获预分频器系数为 8

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
CH3MS[2]	CH2MS[2]	CH3COMADDSEN	CH2COMADDSEN	保留											
		保留	保留												
rw	rw	rw	rw												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3COMCEN	CH3COMCTL[2:0]			CH3COMSEN	保留	CH3MS[1:0]		CH2COMCEN	CH2COMCTL[2:0]			CH2COMSEN	保留	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	CH3MS[2]	通道 3 I/O 模式选择 参考 CH3MS[1:0]描述。
30	CH2MS[2]	通道 2 I/O 模式选择 参考 CH2MS[1:0]描述。
29	CH3COMADDSEN	通道 3 附加输出比较影子寄存器使能 参考 CH2COMADDSEN 描述。
28	CH2COMADDSEN	通道 2 附加输出比较影子寄存器使能 当此位被置 1，TIMERx_CH2COMV_ADD 寄存器的影子寄存器使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 2 附加输出/比较影子寄存器 1：使能通道 2 附加输出/比较影子寄存器 仅在单脉冲模式下(TIMERx_CTL0 寄存器的 SPM =1)，可以在未确认预装载寄存器情况下使用 PWM 模式。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH2MS =000 时此位不能被改变。
27:16	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	保留	必须保持复位值。
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（当 MCH3MSEL[1:0] = 2b'00 时，TIMERx_CHCTL2 寄存器的 CH3EN 位清 0；当 MCH3MSEL[1:0] = 2b'01 或 2b'11 时，TIMERx_CHCTL2 寄存器的 CH3EN、MCH3EN 位清 0）时，这些位才可以写。 000：通道 3 配置为输出 001：通道 3 配置为输入，IS3 映射在 CI3FE3 上 010：通道 3 配置为输入，IS3 映射在 CI2FE3 上 011：通道 3 配置为输入，IS3 映射在 ITS 上，此模式仅工作在内部触发源输入被选中时（由 TIMERx_SMCFGFG 寄存器的 TRGS 位选择）。 100：通道3配置为输入，IS3映射在MCI3FE3上。 101~111：保留
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 的动作，而 O2CPRE 决定了 CH2_O 的值。O2CPRE 高电平有效，而 CH2_O 的有效电平取决于 CH2P 位。 **注意**：当多模式通道 2 配置为输出模式，且 MCH2MSEL[1:0] = 2b'11，此位定义了输出准备信号 O2CPRE 的动作，而 O2CPRE 决定了 CH2_O、MCH2_O 的值。O2CPRE 高电平有效，而 CH2_O、MCH2_O 的有效电平取决于 CH2P、MCH2P 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 为有效电平，否则为无效电平。在向下计数时，一旦计数器的值大于 TIMERx_CH2CV 时，O2CPRE 为无效电平，否则为有效电平。 111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH2CV 时，O2CPRE 为无效电平，否则为有效电平。在向下计数时，一旦计数器的值大于 TIMERx_CH2CV 时，O2CPRE 为有效电平，否则为无效电平。 **注意**：在复合 PWM 模式下（CH2CPWMEN = 1'b1 和 CH2MS = 3'b000），通道 2

		的 PWM 输出信号由 TIMERx_CH2CV 和 TIMERx_CH2COMV_ADD 寄存器共同确定。详细信息请参考*复合 PWM 模式*。 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 =000（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 =000 时此位不能被改变。
2	保留	必须保持复位值。
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（当 MCH2MSEL[1:0] = 2b’00 时，TIMERx_CHCTL2 寄存器的 CH2EN 位清 0；当 MCH1MSEL[1:0] = 2b’01 或 2b’11 时，TIMERx_CHCTL2 寄存器的 CH2EN、MCH2EN 位清 0）时，这些位才可写。 000：通道 2 配置为输出 001：通道 2 配置为输入，IS2 映射在 CI2FE2 上 010：通道 2 配置为输入，IS2 映射在 CI3FE2 上 011：通道 2 配置为输入，IS2 映射在 ITS 上。此模式仅工作在内部触发输入被选中时（通过设置 TIMERx_SMCFGFG 寄存器的 TRGS 位）。 100：通道2配置为输入，IS2映射在MCI2FE2上。 101~111：保留

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：输入捕获预分频器系数为 2

10：输入捕获预分频器系数为 4

11：输入捕获预分频器系数为 8

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
MCH3P	MCH3EN	CH3P	CH3EN	MCH2P	MCH2EN	CH2P	CH2EN	MCH1P	MCH1EN	CH1P	CH1EN	MCH0P	MCH0EN	CH0P	CH0EN
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	MCH3P	多模式通道 3 捕获/比较极性

		参考 MCH0P 描述。
14	MCH3EN	多模式通道 3 捕获/比较使能 参考 MCH0EN 描述。
13	CH3P	通道 3 捕获/比较极性 参考 CH0P 描述。
12	CH3EN	通道 3 捕获/比较使能 参考 CH0EN 描述。
11	MCH2P	多模式通道 2 捕获/比较极性 参考 MCH0P 描述。
10	MCH2EN	多模式通道 2 捕获/比较使能 参考 MCH0EN 描述。
9	CH2P	通道 2 捕获/比较极性 参考 CH0P 描述。
8	CH2EN	通道 2 捕获/比较使能 参考 CH0EN 描述。
7	MCH1P	多模式通道 1 捕获/比较极性 参考 MCH0P 描述。
6	MCH1EN	多模式通道 1 捕获/比较使能 参考 MCH0EN 描述。
5	CH1P	通道 1 捕获/比较极性 参考 CH0P 描述。
4	CH1EN	通道 1 捕获/比较使能 参考 CH0EN 描述。
3	MCH0P	多模式通道 0 捕获/比较极性 当通道 0 配置为输出模式，且 MCH0MSEL[1:0] = 2b’11 时，此位定义了多模式通道 0 输出信号 MCH0_O 的极性。 0：多模式通道 0 高电平有效 1：多模式通道 0 低电平有效 当通道 0 配置为输入模式时，此位和 CH0P 联合使用，作为通道 0 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	MCH0EN	多模式通道 0 捕获/比较使能 当多模式通道 0 配置为输出模式时，将此位置 1 使能 MCH0_O 信号有效。当多模式通道 0 配置为输入模式时，将此位置 1 使能多模式通道 0 上的捕获事件。 0：禁止多模式通道 0 1：使能多模式通道 0
1	CH0P	通道 0 捕获/比较极性

		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 高电平有效 1：通道 0 低电平有效 当通道 0 配置为输入模式时，此位定义了通道 0 输入信号的极性。[MCH0P, CH0P] 用于选择通道 0 输入信号信号有效边沿或者捕获极性。 00：把通道 0 输入信号的上升沿作为捕获或者从模式下触发的有效信号，且通道 0 输入信号不会被翻转。 01：把通道 0 输入信号的下降沿作为捕获或者从模式下触发的有效信号，且通道 0 输入信号会被翻转。 10：保留。 11：把通道 0 输入信号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，且通道 0 输入信号不翻转。 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
Reserved															

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 的值被装入当前预分频寄存器。

计数器自动重载寄存器(TIMERx_CAR)

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。 **注意：**在定时器被配置为输入捕获模式时，该寄存器需要被配置成一个大于用户期望值的非 0 值(例如 0xFFFF)。

重复计数寄存器(TIMERx_CREP)

地址偏移：0x30
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	CREP[7:0]	重复计数器的值 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。

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/比较寄存器(TIMERx_CH3CV)

地址偏移：0x40

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH3VAL[15:0]	通道 3 的捕获或比较值 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 –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 位，MCHxEN 位），则使能 CHx_O 和 MCHx_O 得输出。 0：禁止通道输出或强制为空闲状态 1：通道输出使能 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	BRKINx（x = 0..3）输入信号极性 此位定义了中止输入信号 BRKINx（x = 0..3）的极性。 0：BRKINx（x = 0..3）中止输入低电平有效 1：BRKINx（x = 0..3）中止输入高电平有效 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。
12	BRKEN	BRKINx（x = 0..3）输入信号使能 此位置 1 使能 BRKINx（x = 0..3）中止事件和 CKM 时钟失败事件输入。 0：中止输入禁能 1：中止输入使能 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。

11	ROS	运行模式下"关闭状态"使能 当 POEN 位被置 1（运行模式），此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出"关闭状态"。参见*表 18-4. 由参数控制的互补输出表(MCHxMSEL =2'b11)*。 0：输出"关闭状态"禁能。当 CHxEN 或者 CHxNEN 位被清零，对应通道为输出"禁能状态"。 1：输出"关闭状态"使能。当 CHxEN 或者 CHxNEN 位被清零，对应通道为输出"关闭状态"。 此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
10	IOS	空闲模式下"关闭状态"使能 当 POEN 位被清 0（空闲模式），此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出"关闭状态"。参见*表 18-4. 由参数控制的互补输出表(MCHxMSEL =2'b11)*。 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、TIMERx_BRKCFG 寄存器中 BRKxP/ BRKxEN（x = 0..3）位、TIMERx_FCCHPx（x = 0..3）寄存器中 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/ MCHxP 位（如果相应通道配置为输出模式），TIMERx_CCHP 寄存器中 ROS/IOS 位和 TIMERx_FCCHPx（x = 0..3）寄存器中 ROS/IOS 位。 11：PROT 模式 2。除了 PROT 模式 1 下的寄存器写保护外，还有 TIMERx_CHCTLR0/ 1 及 TIMERx_MCHCTL0/ 1 寄存器中 CHxCOMCTL/ CHxCOMSEN/ CHxCOMADDSEN/ MCHxCOMCTL/ MCHxCOMSEN 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。

多模式通道控制寄存器 0（TIMERx_MCHCTL0）

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
MCH1MS[2]	MCH0MS[2]	保留													
rw	rw														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCH1COMCEN	MCH1COMCTL[2:0]			MCH1COMSEN	保留	MCH1MS[1:0]		MCH0COMCEN	MCH0COMCTL[2:0]			MCH0COMSEN	保留	MCH0MS[1:0]	
MCH1CAPFLT[3:0]				MCH1CAPPSC[1:0]				MCH0CAPFLT[3:0]				MCH0CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

输出比较模式：

位/位域	名称	描述
31	MCH1MS[2]	多模式通道 1 I/O 模式选择 参考 MCH1MS[1:0]描述。
30	MCH0MS[2]	多模式通道 0 I/O 模式选择 参考 MCH0MS[1:0]描述。
29:16	保留	必须保持复位值。
15	MCH1COMCEN	多模式通道 1 输出比较清 0 使能 参见 MCH0COMCEN 描述。
14:12	MCH1COMCTL[2:0]	多模式通道 1 输出比较模式 参见 MCH0COMCTL 描述。
11	MCH1COMSEN	多模式通道 1 输出比较影子寄存器使能 参见 MCH0COMSEN 描述。
10	保留	必须保持复位值。
9:8	MCH1MS[1:0]	多模式通道 1 I/O 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 MCH1EN 位清 0）时这些位才可以写。 000：多模式通道 1 配置为输出 001：多模式通道1配置为输入，MIS1映射在MCI1FEM1上 010：多模式通道1配置为输入，MIS1映射在MCI0FEM1上 011：多模式通道 1 配置为输入，MIS1 映射在 ITS 上。此模式仅工作在内部触发源输入被选中时（由 TIMERx_SMCFGFG 寄存器的 TRGS 位选择）。 100：多模式通道1配置为输入，MIS1映射在CI1FEM1上。

		101~111：保留
7	MCH0COMCEN	多模式通道 0 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，MO0CPRE 参考信号被清 0。 0：多模式通道 0 输出比较清零禁止 1：多模式通道 0 输出比较清零使能
6:4	MCH0COMCTL[2:0]	多模式通道 0 输出比较模式 当多模式通道 0 配置为输出模式，并且 MCH0MSEL[1:0] = 2b'00，此位定义了输出准备信号 MO0CPRE 的动作，而 MO0CPRE 决定了 MCH0_O 的值。MO0CPRE 高电平有效，而 MCH0_O 的有效电平取决于 MCH0FP[1:0]位。 **注意**：当多模式通道 0 配置为输出模式，且 MCH0MSEL[1:0] = 2b'11 时，CH0COMCTL[2:0]位定义了输出准备信号 O0CPRE 的动作，而 O0CPRE 决定了 CH0_O、MCH0_O 的值。O0CPRE 高电平有效，CH0_O、MCH0_O 的有效电平取决于 CH0P、MCH0P 位。 000：时基。输出比较寄存器 TIMERx_MCH0CV 与计数器 TIMERx_CNT 间的比较对 MO0CPRE 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_MCH0CV 相同时，强制 MO0CPRE 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_MCH0CV 相同时，强制 MO0CPRE 为低。 011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_MCH0CV 相同时，强制 MO0CPRE 翻转。 100：强制为低。强制 MO0CPRE 为低电平 101：强制为高。强制 MO0CPRE 为高电平 110：PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_MCH0CV 时，MO0CPRE 为有效电平，否则为无效电平。在向下计数时，一旦计数器的值大于 TIMERx_MCH0CV 时，MO0CPRE 为无效电平，否则为有效电平。 111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_MCH0CV 时，MO0CPRE 为无效电平，否则为有效电平。在向下计数时，一旦计数器的值大于 TIMERx_MCH0CV 时，MO0CPRE 为有效电平，否则为无效电平。 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，MO0CPRE 电平才改变。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 MCH0MS =00（比较模式）时此位不能被改变。
3	MCH0COMSEN	多模式通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_MCH0CV 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：禁止多模式通道 0 输出/比较影子寄存器 1：使能多模式通道 0 输出/比较影子寄存器 仅在单脉冲模式下（TIMERx_CTL0 寄存器的 SPM =1），可以在未确认预装载寄存器情况下使用 PWM 模式。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS=00 时此位不能被改变。
2	保留	必须保持复位值。

1:0	MCH0MS[1:0]	多模式通道 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 MCH0EN 位清 0）时，MCH0MS[2:0]才可写。 000：多模式通道 0 配置为输出 001：多模式通道 0 配置为输入，MIS0 映射在 MCI0FEM0 上 010：多模式通道 0 配置为输入，MIS0 映射在 MCI1FEM0 上 011：多模式通道 0 配置为输入，MIS0 映射在 ITS 上。此模式仅工作在内部触发输入被选中时（通过设置 TIMERx_SMCFGFG 寄存器的 TRGS 位）。 100：多模式通道0配置为输入，MIS0映射在CI0FEM0上。 101~111：保留

输入捕获模式：

位/位域	名称	描述
31	MCH1MS[2]	多模式通道 1 I/O 模式选择 参考 MCH1MS[1:0]描述。
30	MCH0MS[2]	多模式通道 0 I/O 模式选择 参考 MCH0MS[1:0]描述。
29:16	保留	必须保持复位值。
15:12	MCH1CAPFLT[3:0]	多模式通道 1 输入捕获滤波控制 参见 MCH0CAPFLT 描述。
11:10	MCH1CAPPSC[1:0]	多模式通道 1 输入捕获预分频器 参见 MCH0CAPPSC 描述。
9:8	MCH1MS[1:0]	多模式通道 1 I/O 模式选择 与输出模式相同。
7:4	MCH0CAPFLT[3:0]	通道 0 输入捕获滤波控制 数字滤波器由一个事件计数器组成，它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 MCI0 输入信号的采样频率和数字滤波器的长度。 0000：无滤波器，$f_{SAMP}= f_{DTS}$，N=1 0001：$f_{SAMP} = f_{CK_TIMER}$，N=2 0010：$f_{SAMP} = f_{CK_TIMER}$，N=4 0011：$f_{SAMP} = f_{CK_TIMER}$，N=8 0100：$f_{SAMP}=f_{DTS}/2$，N=6 0101：$f_{SAMP}=f_{DTS}/2$，N=8 0110：$f_{SAMP}=f_{DTS}/4$，N=6 0111：$f_{SAMP}=f_{DTS}/4$，N=8 1000：$f_{SAMP}=f_{DTS}/8$，N=6 1001：$f_{SAMP}=f_{DTS}/8$，N=8 1010：$f_{SAMP}=f_{DTS}/16$，N=5 1011：$f_{SAMP}=f_{DTS}/16$，N=6 1100：$f_{SAMP}=f_{DTS}/16$，N=8 1101：$f_{SAMP}=f_{DTS}/32$，N=5

1110：$f_{SAMP}=f_{DTS}/32$，N=6

1111：$f_{SAMP}=f_{DTS}/32$，N=8

3:2	MCH0CAPPSC[1:0]	多模式通道 0 输入捕获预分频器 这 2 位定义了多模式通道 0 输入的预分频系数。当 TIMERx_CHCTL2 寄存器中的 MCH0EN =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	MCH0MS[1:0]	多模式通道 0 模式选择 与输出比较模式相同

多模式通道控制寄存器 1（TIMERx_MCHCTL1）

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
MCH3MS[2]	MCH2MS[2]	保留													
rw	rw														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCH3COMCEN	MCH3COMCTL[2:0]			MCH3COMSEN	保留	MCH3MS[1:0]		MCH2COMCEN	MCH2COMCTL[2:0]			MCH2COMSEN	保留	MCH2MS[1:0]	
MCH3CAPFLT[3:0]				MCH3CAPPSC[1:0]				MCH2CAPFLT[3:0]				MCH2CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

输出比较模式：

位/位域	名称	描述
31	MCH3MS[2]	多模式通道 1 I/O 模式选择 参考 MCH3MS[1:0]描述。
30	MCH2MS[2]	多模式通道 0 I/O 模式选择 参考 MCH2MS[1:0]描述。
29:16	保留	必须保持复位值。
15	MCH3COMCEN	多模式通道 3 输出比较清 0 使能 参见 MCH2COMCEN 描述。
14:12	MCH3COMCTL[2:0]	多模式通道 3 输出比较模式 参见 MCH2COMCTL 描述。
11	MCH3COMSEN	多模式通道 3 输出比较影子寄存器使能 参见 MCH2COMSEN 描述。
10	保留	必须保持复位值。

9:8	MCH3MS[1:0]	多模式通道 3 I/O 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 MCH3EN 位清 0）时，这些位才可以写。 000：多模式通道 3 配置为输出 001：多模式通道 3 配置为输入，MIS3 映射在 MCI3FEM3 上 010：多模式通道 3 配置为输入，MIS3 映射在 MCI2FEM3 上 011：多模式通道 3 配置为输入，MIS3 映射在 ITS 上，此模式仅工作在内部触发源输入被选中时（由 TIMERx_SMCFGFG 寄存器的 TRGS 位选择）。 100：多模式通道3配置为输入，MIS3映射在CI3FEM3上。 101~111：保留
7	MCH2COMCEN	多模式通道 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，MO2CPRE 参考信号被清 0 0：多模式通道 2 输出比较清零禁止 1：多模式通道 2 输出比较清零使能
6:4	MCH2COMCTL[2:0]	多模式通道 2 输出比较模式 当多模式通道 2 配置为输出模式，并且 MCH2MSEL[1:0] = 2b’00，这些位定义了输出准备信号 MO2CPRE 的动作，而 MO2CPRE 决定了 MCH2_O 的值。MO2CPRE 高电平有效，而 MCH2_O 的有效电平取决于 MCH2FP[1:0]位。 **注意**：当多模式通道 2 配置为输出模式，且 MCH2MSEL[1:0] = 2b’11 时，CH2COMCTL[2:0]位定义了输出准备信号 O2CPRE 的动作，而 O2CPRE 决定了 CH2_O、MCH2_O 的值。O2CPRE 高电平有效，CH2_O、MCH2_O 的有效电平取决于 CH2P、MCH2P 位。 000：时基。输出比较寄存器 TIMERx_CHN2CV 与计数器 TIMERx_CNT 间的比较对 MO2CPRE 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_MCH2CV 相同时，强制 MO2CPRE 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_MCH2CV 相同时，强制 MO2CPRE 为低。 011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_MCH2CV 相同时，强制 MO2CPRE 翻转。 100：强制为低。强制 MO2CPRE 为低电平 101：强制为高。强制 MO2CPRE 为高电平 110：PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_MCH2CV 时，MO2CPRE 为有效电平，否则为无效电平。在向下计数时，一旦计数器的值大于 TIMERx_MCH2CV 时，MO2CPRE 为无效电平，否则为有效电平。 111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_MCH2CV 时，MO2CPRE 为无效电平，否则为有效电平。在向下计数时，一旦计数器的值大于 TIMERx_MCH2CV 时，MO2CPRE 为有效电平，否则为无效电平。 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，MO2CPRE 电平才改变。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 MCH2MS =00（比较模式）时此位不能被改变。
3	MCH2COMSEN	多模式通道 2 输出比较影子寄存器使能

		当此位被置 1，TIMERx_MCH2CV 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：禁止多模式通道 2 输出/比较影子寄存器 1：使能多模式通道 2 输出/比较影子寄存器 仅在单脉冲模式下（TIMERx_CTL0 寄存器的 SPM =1），可以在未确认预装载寄存器情况下使用 PWM 模式。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 MCH2MS =00 时此位不能被改变。
2	保留	必须保持复位值。
1:0	MCH2MS[1:0]	多模式通道 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 MCH2EN 位清 0）时，这些位才可写。 000：多模式通道 2 配置为输出 001：多模式通道 2 配置为输入，MIS2 映射在 MCI2FEM2 上 010：多模式通道 2 配置为输入，MIS2 映射在 MCI3FE2M 上 011：多模式通道 2 配置为输入，MIS2 映射在 ITS 上。此模式仅工作在内部触发输入被选中时（通过设置 TIMERx_SMCFGFG 寄存器的 TRGS 位）。 100：多模式通道2配置为输入，MIS3映射在CI2FEM2上。 101~111：保留

输入捕获模式：

位/位域	名称	描述
31	MCH3MS[2]	多模式通道 1 I/O 模式选择 参考 MCH3MS[1:0]描述。
30	MCH2MS[2]	多模式通道 0 I/O 模式选择 参考 MCH2MS[1:0]描述。
29:16	保留	必须保持复位值。
15:12	MCH3CAPFLT[3:0]	多模式通道 3 输入捕获滤波控制 参见 MCH2CAPFLT 描述。
11:10	MCH3CAPPSC[1:0]	多模式通道 3 输入捕获预分频器 参见 MCH2CAPPSC 描述。
9:8	MCH3MS[1:0]	多模式通道 3 I/O 模式选择 与输出模式相同。
7:4	MCH2CAPFLT[3:0]	多模式通道 2 输入捕获滤波控制 数字滤波器由一个事件计数器组成，它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 MCI2 输入信号的采样频率和数字滤波器的长度。 0000：无滤波器，$f_{SAMP}= f_{DTS}$，N=1 0001：$f_{SAMP} = f_{CK_TIMER}$，N=2 0010：$f_{SAMP} = f_{CK_TIMER}$，N=4 0011：$f_{SAMP} = f_{CK_TIMER}$，N=8 0100：$f_{SAMP}=f_{DTS}/2$，N=6

0101：$f_{SAMP}=f_{DTS}/2$，N=8
0110：$f_{SAMP}=f_{DTS}/4$，N=6
0111：$f_{SAMP}=f_{DTS}/4$，N=8
1000：$f_{SAMP}=f_{DTS}/8$，N=6
1001：$f_{SAMP}=f_{DTS}/8$，N=8
1010：$f_{SAMP}=f_{DTS}/16$，N=5
1011：$f_{SAMP}=f_{DTS}/16$，N=6
1100：$f_{SAMP}=f_{DTS}/16$，N=8
1101：$f_{SAMP}=f_{DTS}/32$，N=5
1110：$f_{SAMP}=f_{DTS}/32$，N=6
1111：$f_{SAMP}=f_{DTS}/32$，N=8

3:2	MCH2CAPPSC[1:0]	多模式通道 2 输入捕获预分频器 这 2 位定义了多模式通道 2 输入的预分频系数。当 TIMERx_CHCTL2 寄存器中的 MCH2EN =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	MCH2MS[1:0]	多模式通道 2 I/O 模式选择 与输出比较模式相同。

多模式通道控制寄存器 2（TIMERx_MCHCTL2）

地址偏移：0x50
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								MCH3FP[1:0]		MCH2FP[1:0]		MCH1FP[1:0]		MCH0FP[1:0]	
								rw		rw		rw		rw	

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:6	MCH3FP[1:0]	多模式通道 3 捕获/比较独立极性控制 参考 MCH0FP[1:0]描述。
5:4	MCH2FP[1:0]	多模式通道 2 捕获/比较独立极性控制 参考 MCH0FP[1:0]描述。
3:2	MCH1FP[1:0]	多模式通道 1 捕获/比较独立极性控制 参考 MCH0FP[1:0]描述。

1:0	MCH0FP[1:0]	多模式通道 0 捕获/比较独立极性控制 当多模式通道 0 配置为输出模式时，且 MCH0MSEL[1:0] = 2b'00，此位定义了输出信号极性。 00：多模式通道 0 高电平有效 01：多模式通道 0 低电平有效 10：保留 11：保留 当通道 0 配置为输入模式时，此位定义了多模式通道 0 输入信号的极性。 MCH0FP[1:0]将选择多模式通道 0 输入信号的有效边沿或者捕获极性。 00：把多模式通道 0 输入信号的上升沿作为捕获或者从模式下触发的有效信号，且多模式通道 0 输入信号不会被翻转。 01：把多模式通道 0 输入信号的下降沿作为捕获或者从模式下触发的有效信号，且多模式通道 0 输入信号会被翻转。 10：保留。 11：把多模式通道 0 输入信号的上升沿或下降沿作为捕获或者从模式下触发的有效信号，并且多模式通道 0 输入信号不会被翻转。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。

多模式通道 0 捕获/比较寄存器(TIMERx_MCH0CV)

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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCH0VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	MCH0VAL[15:0]	多模式通道 0 的捕获或比较值 当多模式通道 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，且本寄存器为只读。 当多模式通道 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。

多模式通道 1 捕获/比较寄存器(TIMERx_MCH1CV)

地址偏移：0x58

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
MCH1VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	MCH1VAL[15:0]	多模式通道 1 的捕获或比较值 当多模式通道 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，且本寄存器为只读。 当多模式通道 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。

多模式通道 2 捕获/比较寄存器(TIMERx_MCH2CV)

地址偏移：0x5C

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
MCH2VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	MCH2VAL[15:0]	多模式通道 2 的捕获或比较值 当多模式通道 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，且本寄存器为只读。 当多模式通道 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。

多模式通道 3 捕获/比较寄存器(TIMERx_MCH3CV)

地址偏移：0x60

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
MCH3VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	MCH3VAL[15:0]	多模式通道 3 的捕获或比较值 当多模式通道 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，且本寄存器为只读。 当多模式通道 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 0 附加比较寄存器(TIMERx_CH0COMV_ADD)

地址偏移：0x64

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
CH0COMVAL_ADD[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH0COMVAL_ADD [15:0]	通道0附加比较值 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。 **注意：**该寄存器仅用于复合PWM模式（当CH0CPWMEN=1时）。

通道 1 附加比较寄存器(TIMERx_CH1COMV_ADD)

地址偏移：0x68

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
CH0COMVAL_ADD[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH1COMVAL_ADD [15:0]	通道1附加比较值 当通道1附加配置为输出模式时，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。 **注意**：该寄存器仅用于复合PWM模式（当CH1CPWMEN=1时）。

通道 2 附加比较寄存器(TIMERx_CH2COMV_ADD)

地址偏移：0x6C
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
CH0COMVAL_ADD[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH2COMVAL_ADD [15:0]	通道2附加比较值 当通道2附加配置为输出模式时，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。 **注意**：该寄存器仅用于复合PWM模式（当CH2CPWMEN=1时）。

通道 3 附加比较寄存器(TIMERx_CH3COMV_ADD)

地址偏移：0x70
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
CH0COMVAL_ADD[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值。

15:0	CH3COMVAL_ADD [15:0]	通道3附加比较值 当通道3附加配置为输出模式时，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。 **注意**：该寄存器仅用于复合PWM模式（当CH2CPWMEN=1时）。

控制寄存器 2（TIMERx_CTL2）

地址偏移：0x74

复位值：0x0FF0 00FF

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3C PWMEN	CH2C PWMEN	CH1C PWMEN	CH0C PWMEN	MCH3MSEL[1:0]		MCH2MSEL[1:0]		MCH1MSEL[1:0]		MCH0MSEL[1:0]		保留			
rw	rw	rw	rw	rw		rw		rw		rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3OMPSEL[1:0]		CH2OMPSEL[1:0]		CH1OMPSEL[1:0]		CH0OMPSEL[1:0]		BRKEN CH3	BRKEN CH2	BRKEN CH1	BRKEN CH0	DTIEN CH3	DTIEN CH2	DTIEN CH1	DTIEN CH0
rw		rw		rw		rw		rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	CH3CPWMEN	通道 3 复合 PWM 模式使能 0：通道 3 复合 PWM 模式禁能 1：通道 3 复合 PWM 模式使能
30	CH2CPWMEN	通道 2 复合 PWM 模式使能 0：通道 2 复合 PWM 模式禁能 1：通道 2 复合 PWM 模式使能
29	CH1CPWMEN	通道 1 复合 PWM 模式使能 0：通道 1 复合 PWM 模式禁能 1：通道 1 复合 PWM 模式使能
28	CH0CPWMEN	通道 0 复合 PWM 模式使能 0：通道 0 复合 PWM 模式禁能 1：通道 0 复合 PWM 模式使能
27:26	MCH3MSEL[1:0]	多模式通道 3 模式选择 00：独立模式，MCH3 独立于 CH3 01：镜像模式，MCH3 输出与 CH3 输出相同（仅用于输出） 10：保留 11：互补模式，只有 CH3 可用于输入，MCH3 输出与 CH3 输出互补
25:24	MMCH2SEL[1:0]	多模式通道 2 模式选择 00：独立模式，MCH2 独立于 CH2 01：镜像模式，MCH2 输出与 CH2 输出相同（仅用于输出） 10：保留

		11：互补模式，只有 CH2 可用于输入，MCH2 输出与 CH2 输出互补
23:22	MCH1MSEL[1:0]	多模式通道 1 模式选择 00：独立模式，MCH1 输出独立于 CH1 输出 01：镜像模式，MCH1 输出与 CH1 输出相同（仅用于输出） 10：保留 11：互补模式，只有 CH1 可用于输入，MCH1 输出与 CH1 输出互补
21:20	MCH0MSEL[1:0]	多模式通道 0 模式选择 00：独立模式，MCH0 独立于 CH0 01：镜像模式，MCH0 输出与 CH0 输出相同（仅用于输出） 10：保留 11：互补模式，只有 CH0 可用于输入，MCH0 输出与 CH0 输出互补
19:16	Reserved	必须保持复位值。
15:14	CH3OMPSEL[1:0]	通道 3 输出匹配脉冲选择 当匹配事件发生时，该位用于选择准备输出信号 O3CPRE（用来驱动 CH3_O 信号）。 00：O3CPRE 信号根据 CH3COMCTL[2:0]位的配置输出。 01：只有在计数器向上计数，匹配事件发生时，O3CPRE 信号输出一个脉冲，且脉冲宽度是一个 CK_TIMER 时钟周期。 10：只有在计数器向下计数，匹配事件发生时，O3CPRE 信号输出一个脉冲，且脉冲宽度是一个 CK_TIMER 时钟周期。 11：在计数器向上计数或向下计数，匹配事件发生时，O3CPRE 信号输出一个脉冲，且脉冲宽度是一个 CK_TIMER 时钟周期。
13:12	CH2OMPSEL[1:0]	通道 2 输出匹配脉冲选择 当匹配事件发生时，该位用于选择准备输出信号 O2CPRE（用来驱动 CH2_O 信号）。 00：O2CPRE 信号根据 CH2COMCTL[2:0]位的配置正常输出。 01：只有在计数器向上计数，匹配事件发生时，O2CPRE 信号输出一个脉冲，且脉冲宽度是一个 CK_TIMER 时钟周期。 10：只有在计数器向下计数，匹配事件发生时，O2CPRE 信号输出一个脉冲，且脉冲宽度是一个 CK_TIMER 时钟周期。 11：在计数器向上计数或者向下计数，匹配事件发生时，O2CPRE 信号输出一个脉冲，且脉冲宽度是一个 CK_TIMER 时钟周期。
11:10	CH1OMPSEL[1:0]	通道 1 输出匹配脉冲选择 当匹配事件发生时，该位用于选择准备输出信号 O1CPRE（用来驱动 CH1_O 信号）。 00：O1CPRE 信号根据 CH1COMCTL[2:0]位的配置正常输出。 01：只有在计数器向上计数，匹配事件发生时，O1CPRE 信号输出一个脉冲，且脉冲宽度是一个 CK_TIMER 时钟周期。 10：只有在计数器向下计数，匹配事件发生时，O1CPRE 信号输出一个脉冲，且脉冲宽度是一个 CK_TIMER 时钟周期。 11：在计数器向上计数或者向下计数，匹配事件发生时，O1CPRE 信号输出一个脉冲，且脉冲宽度是一个 CK_TIMER 时钟周期。

9:8	CH0OMPSEL[1:0]	通道 0 输出匹配脉冲选择 当匹配事件发生时，该位用于选择准备输出信号 O0CPRE（用来驱动 CH0_O 信号）。 00：O0CPRE 信号根据 CH0COMCTL[2:0]位的配置正常输出。 01：只有在计数器向上计数，匹配事件发生时，O0CPRE 信号输出一个脉冲，并且脉冲宽度是一个 CK_TIMER 时钟周期。 10：只有在计数器向下计数，匹配事件发生时，O0CPRE 信号输出一个脉冲，脉冲宽度是一个 CK_TIMER 时钟周期。 11：在计数器向上计数或者向下计数，匹配事件发生时，O0CPRE 信号输出一个脉冲，脉冲宽度是一个 CK_TIMER 时钟周期。
7	BRKENCH3	通道 3 中止控制使能 0：通道 3 中止控制禁能 1：通道 3 中止控制使能
6	BRKENCH2	通道 2 中止控制使能 0：通道 2 中止控制禁能 1：通道 2 中止控制使能
5	BRKENCH1	通道 1 中止控制使能 0：通道 1 中止控制禁能 1：通道 1 中止控制使能
4	BRKENCH0	通道 0 中止控制使能 0：通道 0 中止控制禁能 1：通道 0 中止控制使能
3	DTIENCH3	通道 3 死区时间插入使能 在 MCH3_O 和 CH3_O 输出中使能死区时间插入。 0：通道 3 死区时间插入禁能 1：通道 3 死区时间插入使能
2	DTIENCH2	通道 2 死区时间插入使能 在 MCH2_O 和 CH2_O 输出中使能死区时间插入。 0：通道 2 死区时间插入禁能 1：通道 2 死区时间插入使能
1	DTIENCH1	通道 1 死区时间插入使能 在 MCH1_O 和 CH1_O 输出中使能死区时间插入。 0：通道 1 死区时间插入禁能 1：通道 1 死区时间插入使能
0	DTIENCH0	通道 0 死区时间插入使能 在 MCH0_O 和 CH0_O 输出中使能死区时间插入。 0：通道 0 死区时间插入禁能 1：通道 0 死区时间插入使能

中止配置寄存器(TIMERx_BRKCFG)

地址偏移：0x78

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
BRK3P	BRK3EN	BRK2P	BRK2EN	BRK1P	BRK1EN	BRK0P	BRK0EN	保留							
rw	rw	rw	rw	rw	rw	rw	rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRK3F[3:0]				BRK2F[3:0]				BRK1F[3:0]				BRK0F[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31	BRK3P	BRKIN3输入信号极性 该位定义了BRKIN3输入信号极性。 0：BRKIN3输入低电平有效 1：BRKIN3输入高电平有效 只有当TIMERx_CCHP寄存器中PROT[1:0] = 00时，该位可以被修改。
30	BRK3EN	BRKIN3输入信号使能 该位置1使能BRKIN3输入。 0：中止输入禁能 1：中止输入使能 只有当TIMERx_CCHP寄存器中PROT[1:0] = 00时，该位可以被修改。
29	BRK2P	BRKIN2输入信号极性 该位定义了BRKIN2输入信号极性。 0：BRKIN2输入低电平有效 1：BRKIN2输入高电平有效 只有当TIMERx_CCHP寄存器中PROT[1:0] = 00时，该位可以被修改。
28	BRK2EN	BRKIN2输入信号使能 该位置1使能BRKIN2输入。 0：中止输入禁能 1：中止输入使能 只有当TIMERx_CCHP寄存器中PROT[1:0] = 00时，该位可以被修改。
27	BRK1P	BRKIN1输入信号极性 该位定义了BRKIN1输入信号极性。 0：BRKIN1输入低电平有效 1：BRKIN1输入高电平有效 只有当TIMERx_CCHP寄存器中PROT[1:0] = 00时，该位可以被修改。
26	BRK1EN	BRKIN1输入信号使能 该位置1使能BRKIN1输入。 0：中止输入禁能 1：中止输入使能

		只有当TIMERx_CCHP寄存器中PROT[1:0] = 00时，该位可以被修改。
25	BRK0P	BRKIN0输入信号极性 该位定义了BRKIN0输入信号极性。 0：BRKIN0输入低电平有效 1：BRKIN0输入高电平有效 只有当TIMERx_CCHP寄存器中PROT[1:0] = 00时，该位可以被修改。
24	BRK0EN	BRKIN0输入信号使能 该位置1使能BRKIN0输入。 0：中止输入禁能 1：中止输入使能 只有当TIMERx_CCHP寄存器中PROT[1:0] = 00时，该位可以被修改。
23:16	保留	必须保持复位值。
15:12	BRK3F[3:0]	BRKIN3输入信号滤波器 数字滤波器由一个事件计数器组成，它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 BRKIN3 输入信号的采样频率和数字滤波器的长度。 0000：无滤波器，BRKIN3 异步动作，N=1 0001：$f_{SAMP} = f_{CK_TIMER}$，N=2 0010：$f_{SAMP} = f_{CK_TIMER}$，N=4 0011：$f_{SAMP} = f_{CK_TIMER}$，N=8 0100：$f_{SAMP} = f_{DTS}/2$，N=6 0101：$f_{SAMP} = f_{DTS}/2$，N=8 0110：$f_{SAMP} = f_{DTS}/4$，N=6 0111：$f_{SAMP} = f_{DTS}/4$，N=8 1000：$f_{SAMP} = f_{DTS}/8$，N=6 1001：$f_{SAMP} = f_{DTS}/8$，N=8 1010：$f_{SAMP} = f_{DTS}/16$，N=5 1011：$f_{SAMP} = f_{DTS}/16$，N=6 1100：$f_{SAMP} = f_{DTS}/16$，N=8 1101：$f_{SAMP} = f_{DTS}/32$，N=5 1110：$f_{SAMP} = f_{DTS}/32$，N=6 1111：$f_{SAMP} = f_{DTS}/32$，N=8 只有当TIMERx_CCHP寄存器中PROT[1:0] = 00时，该位可以被修改。
11:8	BRK2F[3:0]	BRKIN2输入信号滤波器 数字滤波器由一个事件计数器组成，它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 BRKIN2 输入信号的采样频率和数字滤波器的长度。 0000：无滤波器，BRKIN2 异步动作，N=1 0001：$f_{SAMP} = f_{CK_TIMER}$，N=2 0010：$f_{SAMP} = f_{CK_TIMER}$，N=4 0011：$f_{SAMP} = f_{CK_TIMER}$，N=8 0100：$f_{SAMP} = f_{DTS}/2$，N=6 0101：$f_{SAMP} = f_{DTS}/2$，N=8 0110：$f_{SAMP} = f_{DTS}/4$，N=6

		0111：$f_{SAMP} = f_{DTS}/4$，N=8 1000：$f_{SAMP} = f_{DTS}/8$，N=6 1001：$f_{SAMP} = f_{DTS}/8$，N=8 1010：$f_{SAMP} = f_{DTS}/16$，N=5 1011：$f_{SAMP} = f_{DTS}/16$，N=6 1100：$f_{SAMP} = f_{DTS}/16$，N=8 1101：$f_{SAMP} = f_{DTS}/32$，N=5 1110：$f_{SAMP} = f_{DTS}/32$，N=6 1111：$f_{SAMP} = f_{DTS}/32$，N=8 只有当TIMERx_CCHP寄存器中PROT[1:0] = 00时，该位可以被修改。
7:4	BRK1F[3:0]	BRKIN1输入信号滤波器 数字滤波器由一个事件计数器组成，它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 BRKIN1 输入信号的采样频率和数字滤波器的长度。 0000：无滤波器，BRKIN1 异步动作，N=1 0001：$f_{SAMP} = f_{CK_TIMER}$，N=2 0010：$f_{SAMP} = f_{CK_TIMER}$，N=4 0011：$f_{SAMP} = f_{CK_TIMER}$，N=8 0100：$f_{SAMP} = f_{DTS}/2$，N=6 0101：$f_{SAMP} = f_{DTS}/2$，N=8 0110：$f_{SAMP} = f_{DTS}/4$，N=6 0111：$f_{SAMP} = f_{DTS}/4$，N=8 1000：$f_{SAMP} = f_{DTS}/8$，N=6 1001：$f_{SAMP} = f_{DTS}/8$，N=8 1010：$f_{SAMP} = f_{DTS}/16$，N=5 1011：$f_{SAMP} = f_{DTS}/16$，N=6 1100：$f_{SAMP} = f_{DTS}/16$，N=8 1101：$f_{SAMP} = f_{DTS}/32$，N=5 1110：$f_{SAMP} = f_{DTS}/32$，N=6 1111：$f_{SAMP} = f_{DTS}/32$，N=8 只有当TIMERx_CCHP寄存器中PROT[1:0] = 00时，该位可以被修改。
3:0	BRK0F[3:0]	BRKIN0输入信号滤波器 数字滤波器由一个事件计数器组成，它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 BRKIN0 输入信号的采样频率和数字滤波器的长度。 0000：无滤波器，BRKIN0 异步动作，N=1 0001：$f_{SAMP} = f_{CK_TIMER}$，N=2 0010：$f_{SAMP} = f_{CK_TIMER}$，N=4 0011：$f_{SAMP} = f_{CK_TIMER}$，N=8 0100：$f_{SAMP} = f_{DTS}/2$，N=6 0101：$f_{SAMP} = f_{DTS}/2$，N=8 0110：$f_{SAMP} = f_{DTS}/4$，N=6 0111：$f_{SAMP} = f_{DTS}/4$，N=8 1000：$f_{SAMP} = f_{DTS}/8$，N=6 1001：$f_{SAMP} = f_{DTS}/8$，N=8

1010：$f_{SAMP} = f_{DTS}/16$，N=5

1011：$f_{SAMP} = f_{DTS}/16$，N=6

1100：$f_{SAMP} = f_{DTS}/16$，N=8

1101：$f_{SAMP} = f_{DTS}/32$，N=5

1110：$f_{SAMP} = f_{DTS}/32$，N=6

1111：$f_{SAMP} = f_{DTS}/32$，N=8

只有当TIMERx_CCHP寄存器中PROT[1:0] = 00时，该位可以被修改。

独立互补通道保护寄存器 0（TIMERx_FCCHP0）

地址偏移：0x7C

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

该寄存器用于配置CH0_O / MCH0_O的输出。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FCCHP0EN	保留														
rw															

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

位/位域	名称	描述
31	FCCHP0EN	独立互补通道寄存器 0 使能位 0：TIMERx_CCHP 寄存器中的 ROS、IOS 和 DTCFG[7:0]有效 1：TIMERx_FCCHP0 寄存器中的 ROS、IOS 和 DTCFG[7:0]有效 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]=00 时才可修改。
30:12	保留	必须保持复位值。
11	ROS	运行模式下“关闭状态”配置 当 POEN 位被置 1，此位定义了通道（带有互补输出且配置为输出模式）的输出状态。 0：当 POEN 位被置 1，通道输出信号（CH0_O/ MCH0_O）被禁止 1：当 POEN 位被置 1，通道输出信号（CH0_O/ MCH0_O）被使能，和 TIMER0_CHCTL2 寄存器 CH0EN/ MCH0EN 位有关。 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”配置 当 POEN 位被清 0，此位定义了已经配置为输出模式的通道的输出状态。 0：当 POEN 位被清 0，通道输出信号（CH0_O/ MCH0_O）被禁止 1：当 POEN 位被清 0，通道输出信号（CH0_O/ MCH0_O）被使能，和 TIMERx_CHCTL2 寄存器 CH0EN/ MCH0EN 位有关。 此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。

9:8	保留	必须保持复位值。
7:0	DTCFG[7:0]	死区时间控制 这些位定义了插入互补输出之间的死区持续时间。DTCFG 值和死区时间的关系如下： DTCFG [7:5] =3’b0xx：DTvalue = DTCFG [7:0]x t_{DT}, t_{DT}=t_{DTS} DTCFG [7:5] =3’b10x：DTvalue = (64+DTCFG [5:0])xt_{DT}, t_{DT} =t_{DTS}*2 DTCFG [7:5] =3’b110：DTvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT}=t_{DTS}*8 DTCFG [7:5] =3’b111：DTvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT} =t_{DTS}*16 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]=00 时才可修改。

独立互补通道保护寄存器 1（TIMERx_FCCHP1）

地址偏移：0x80
复位值：0x0000 0000

该寄存器只能按字（32位）访问。
该寄存器用于配置CH1_O / MCH1_O的输出。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FCCHP1EN	保留														
rw															

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

位/位域	名称	描述
31	FCCHP1EN	独立互补通道寄存器 1 使能位 0：TIMERx_CCHP 寄存器中的 ROS、IOS 和 DTCFG[7:0]有效 1：TIMERx_FCCHP1 寄存器中的 ROS、IOS 和 DTCFG[7:0]有效 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]=00 时才可修改。
30:12	保留	必须保持复位值。
11	ROS	运行模式下“关闭状态”配置 当 POEN 位被置 1，此位定义了通道（带有互补输出且配置为输出模式）的输出状态。 0：当 POEN 位被置 1，通道输出信号（CH1_O / MCH1_O）被禁止 1：当 POEN 位被置 1，通道输出信号（CH1_O / MCH1_O）被使能，和 TIMER0_CHCTL2 寄存器 CH1EN/ MCH1EN 位有关 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”配置 当 POEN 位被清 0，此位定义了已经配置为输出模式的通道的输出状态。 0：当 POEN 位被清 0，通道输出信号（CH1_O / MCH1_O）被禁止 1：当 POEN 位被清 0，通道输出信号（CH1_O / MCH1_O）被使能，和 TIMERx_CHCTL2 寄存器 CH1EN/MCH1EN 位有关

		此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
9:8	保留	必须保持复位值。
7:0	DTCFG[7:0]	死区时间控制 这些位定义了插入互补输出之间的死区持续时间。DTCFG 值和死区时间的关系如下： DTCFG [7:5] =3’b0xx：DTvalue = DTCFG [7:0]x t_{DT}, t_{DT}=t_{DTS}. DTCFG [7:5] =3’b10x：DTvalue = (64+DTCFG [5:0])xt_{DT}, t_{DT} =t_{DTS}*2. DTCFG [7:5] =3’b110：DTvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT}=t_{DTS}*8. DTCFG [7:5] =3’b111：DTvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT} =t_{DTS}*16. 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]=00 时才可修改。

独立互补通道保护寄存器 2（TIMERx_FCCHP2）

地址偏移：0x84

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

该寄存器用于配置CH2_O / MCH2_O的输出。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FCCHP2EN	保留														
rw															

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

位/位域	名称	描述
31	FCCHP2EN	独立互补通道寄存器 2 使能位 0：TIMERx_CCHP 寄存器中的 ROS、IOS 和 DTCFG[7:0]有效 1：TIMERx_FCCHP2 寄存器中的 ROS、IOS 和 DTCFG[7:0]有效 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]=00 时才可修改。
30:12	保留	必须保持复位值。
11	ROS	运行模式下“关闭状态”配置 当 POEN 位被置 1，此位定义了通道（带有互补输出且配置为输出模式）的输出状态。 0：当 POEN 位被置 1，通道输出信号（CH2_O / MCH2_O）被禁止 1：当 POEN 位被置 1，通道输出信号（CH2_O / MCH2_O）被使能，和 TIMER0_CHCTL2 寄存器 CH2EN/ MCH2EN 位有关 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”配置 当 POEN 位被清 0，此位定义了已经配置为输出模式的通道的输出状态。 0：当 POEN 位被清 0，通道输出信号（CH2_O / MCH2_O）被禁止

		1：当 POEN 位被清 0，通道输出信号（CH2_O / MCH2_O）被使能，和 TIMERx_CHCTL2 寄存器 CH2EN/ MCH2EN 位有关 此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
9:8	保留	必须保持复位值。
7:0	DTCFG[7:0]	死区时间控制 这些位定义了插入互补输出之间的死区持续时间。DTCFG 值和死区时间的关系如下： DTCFG [7:5] =3’b0xx：DTvalue = DTCFG [7:0]x t_{DT}, t_{DT}=t_{DTS}. DTCFG [7:5] =3’b10x：DTvalue = (64+DTCFG [5:0])xt_{DT}, t_{DT} =t_{DTS}*2. DTCFG [7:5] =3’b110：DTvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT}=t_{DTS}*8. DTCFG [7:5] =3’b111：DTvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT} =t_{DTS}*16. 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]=00 时才可修改。

独立互补通道保护寄存器 3（TIMERx_FCCHP3）

地址偏移：0x88

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

该寄存器用于配置CH3_O / MCH3_O的输出。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FCCHP3EN	保留														
rw															

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

位/位域	名称	描述
31	FCCHP3EN	独立互补通道寄存器 3 使能位 0：TIMERx_CCHP 寄存器中的 ROS、IOS 和 DTCFG[7:0]有效 1：TIMERx_FCCHP3 寄存器中的 ROS、IOS 和 DTCFG[7:0]有效 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]=00 时才可修改。
30:12	保留	必须保持复位值。
11	ROS	运行模式下“关闭状态”配置 当 POEN 位被置 1，此位定义了通道（带有互补输出且配置为输出模式）的输出状态。 0：当 POEN 位被置 1，通道输出信号（CH3_O / MCH3_O）被禁止 1：当 POEN 位被置 1，通道输出信号（CH3_O / MCH3_O）被使能，和 TIMER0_CHCTL2 寄存器 CH3EN/ MCH3EN 位有关 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”配置

		当 POEN 位被清 0，此位定义了已经配置为输出模式的通道的输出状态。 0：当 POEN 位被清 0，通道输出信号（CH3_O / MCH3_O）被禁止 1：当 POEN 位被清 0，通道输出信号（CH3_O / MCH3_O）被使能，和 TIMERx_CHCTL2 寄存器 CH3EN/ MCH3EN 位有关 此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
9:8	保留	必须保持复位值。
7:0	DTCFG[7:0]	死区时间控制 这些位定义了插入互补输出之间的死区持续时间。DTCFG 值和死区时间的关系如下： DTCFG [7:5] =3’b0xx：DTvalue = DTCFG [7:0]x t_{DT}, t_{DT}=t_{DTS}. DTCFG [7:5] =3’b10x：DTvalue = (64+DTCFG [5:0])xt_{DT}, t_{DT} =t_{DTS}*2. DTCFG [7:5] =3’b110：DTvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT}=t_{DTS}*8. DTCFG [7:5] =3’b111：DTvalue =（32+DTCFG [4:0])xt_{DT}, t_{DT} =t_{DTS}*16. 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：0xE0

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[5:0]						保留		DMATA [5:0]					
		rw								rw					

位/位域	名称	描述
31:14	保留	必须保持复位值。
13:8	DMATC [5:0]	DMA 传输计数 该位域定义了 DMA 访问（读/写）TIMERx_DMATB 寄存器的次数 n，n = (DMATC [5:0] +1)。DMATC [5:0]从 6’b00000 到 6’b100010.
7:6	保留	必须保持复位值。
5:0	DMATA [5:0]	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。 总之：起始地址 = TIMERx_CTL0 + DMATA*4

DMA 发送缓冲区寄存器（TIMERx_DMATB）

地址偏移：0xE4

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]															

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。

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	OUTSEL
														rw	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	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）

18.2.1. 简介

通用定时器 L0（TIMER1）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 的计数器是 16 位无符号计数器。

通用定时器 L0 是可编程的，可以被用来计数，其外部事件可以驱动其他定时器。

定时器和定时器之间是相互独立，但是它们的计数器可以被同步在一起形成一个更大的定时器。

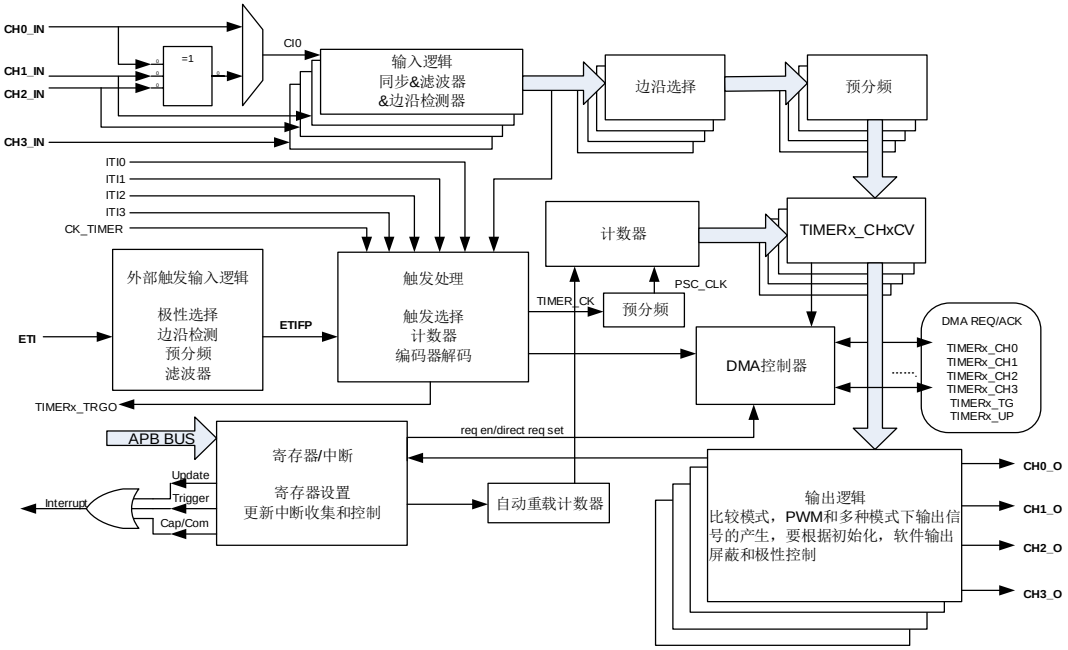
18.2.2. 主要特征

- 总通道数：4；
- 计数器宽度：16位；
- 时钟源可选：内部时钟，内部触发，外部输入，外部触发；
- 多种计数模式：向上计数，向下计数和中央计数；
- 正交译码器接口：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 18-42. 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 18-42. 通用定时器 L0 结构框图



18.2.4. 功能说明

时钟源配置

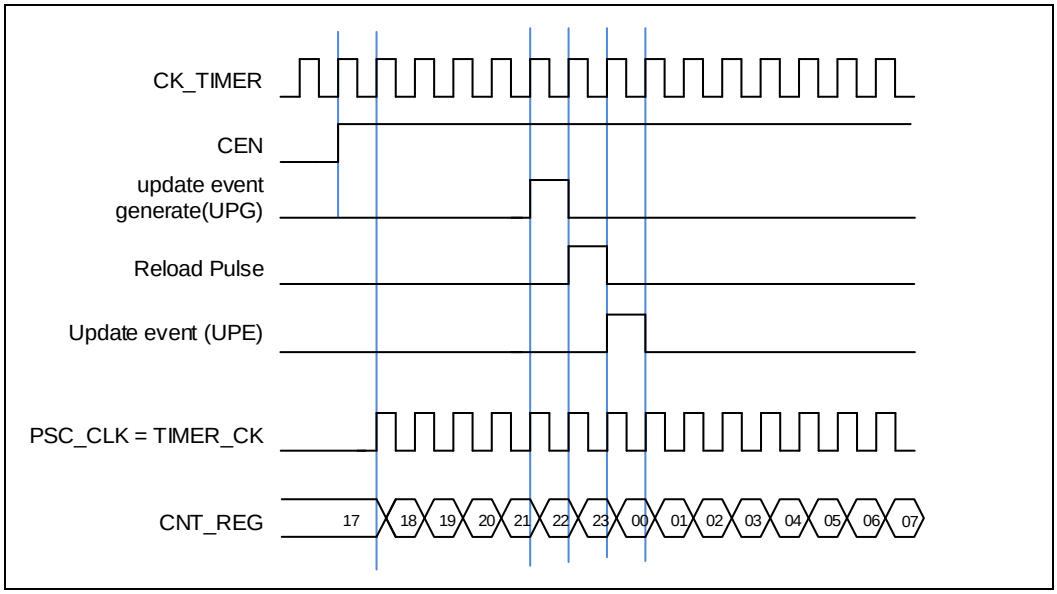
通用定时器 L0 可以由内部时钟源 CK_TIMER 或者由 SMC（TIMERx_SMCFG 寄存器位[2:0]）控制的复用时钟源驱动。

- 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 18-43. 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 0x3。

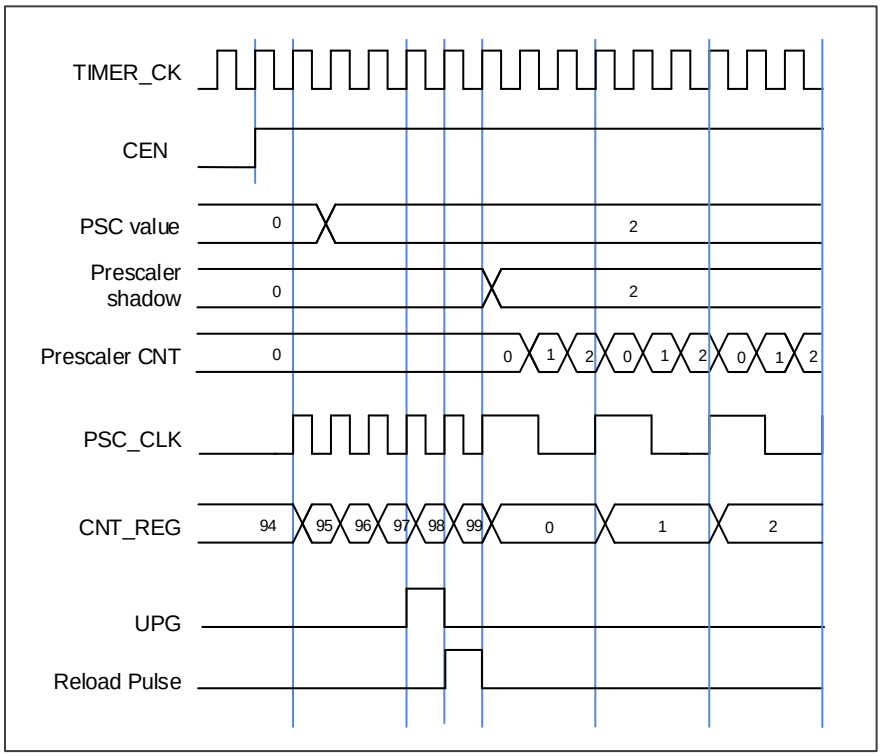
■ SMC1= 1’b1（外部时钟模式1），定时器选择外部输入引脚ETI作为时钟源

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 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 18-44. 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 18-45. 向上计数时序图，PSC=0/2 和 *图 18-46. 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 18-45. 向上计数时序图，PSC=0/2

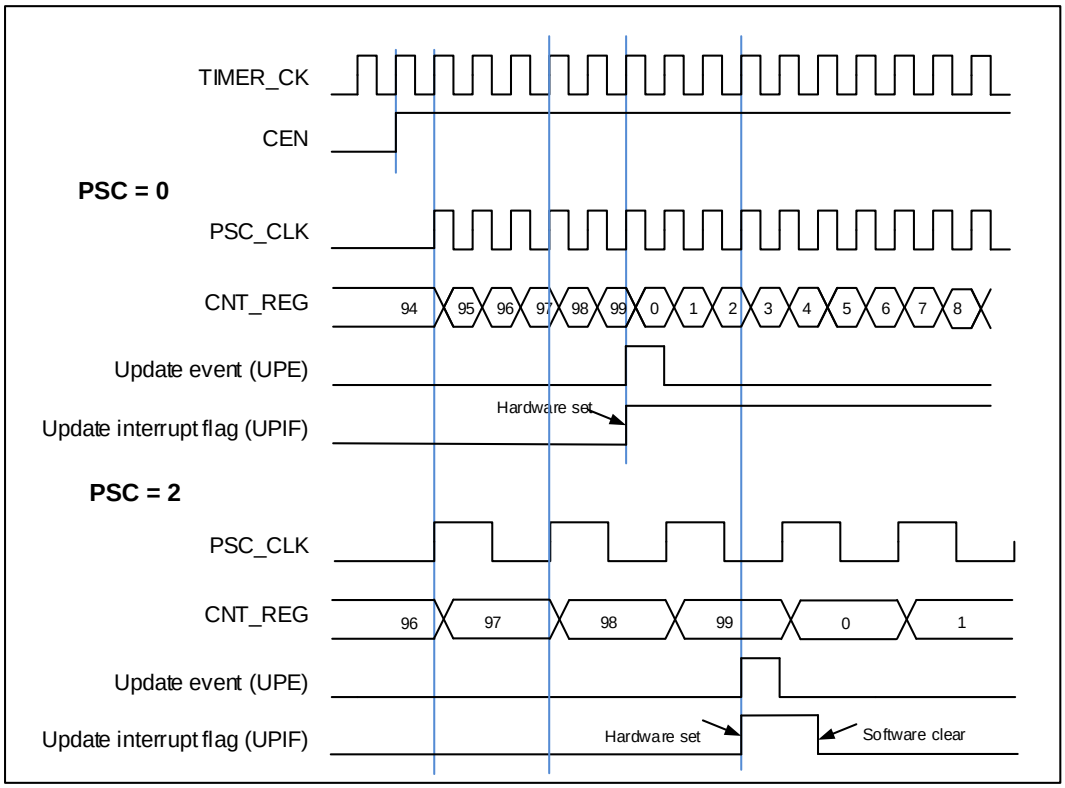
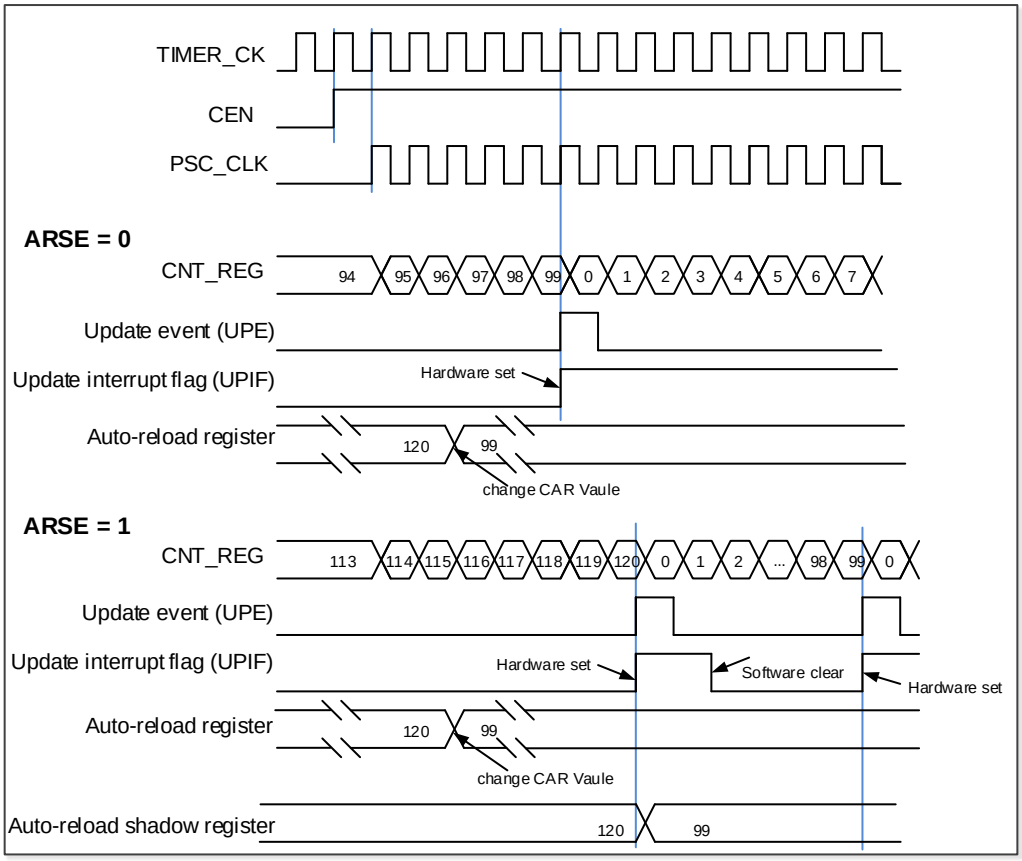


图 18-46. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



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 18-47. 向下计数时序图，PSC=0/2 和 *图 18-48. 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 18-47. 向下计数时序图，PSC=0/2

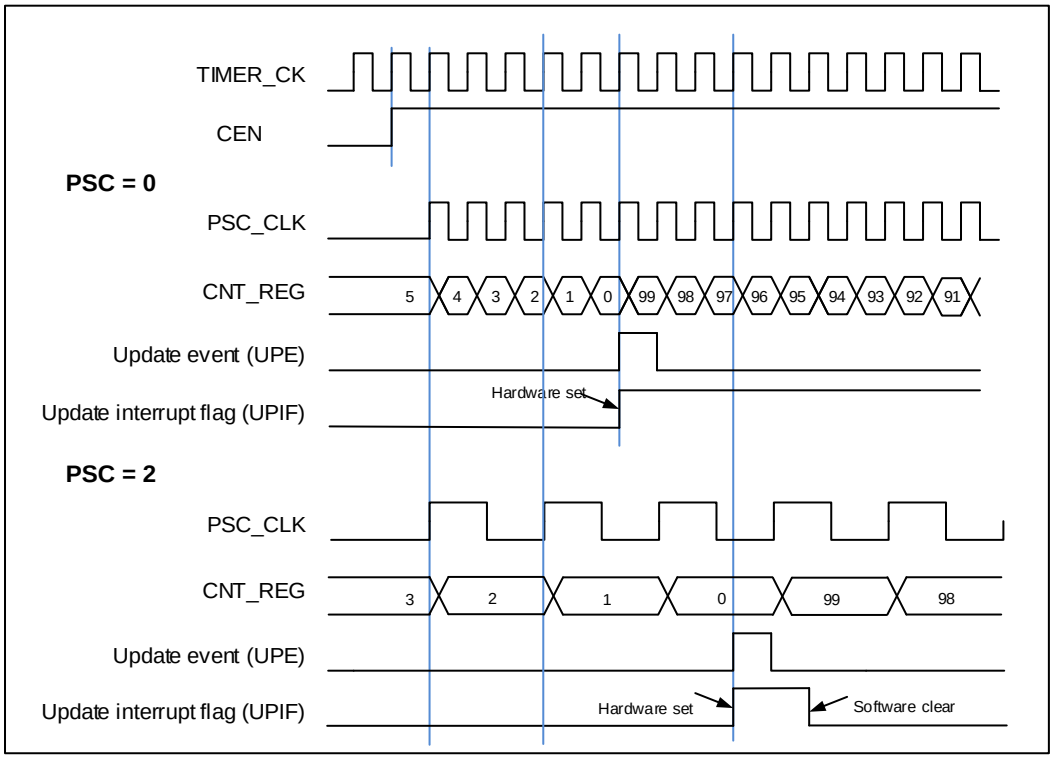
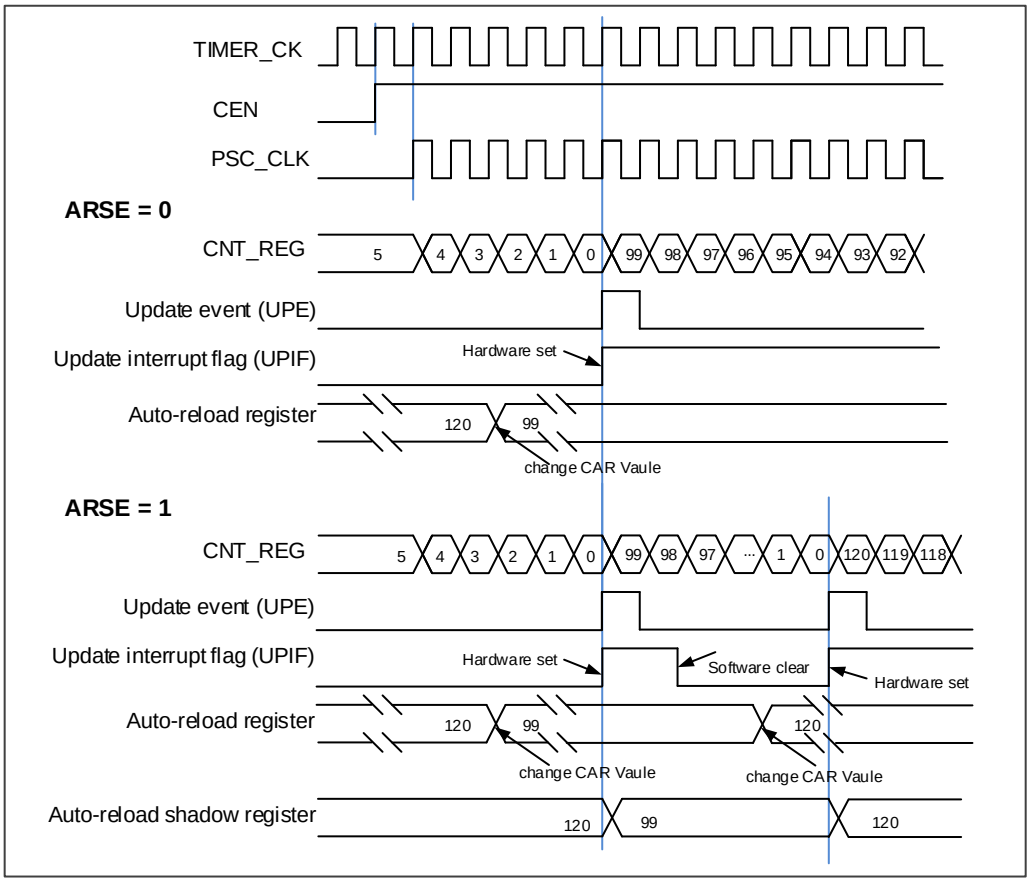


图 18-48. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值



计数器中央对齐模式

在中央对齐计数模式下，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。在中央对齐计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 只读，指示了当前的计数方向。

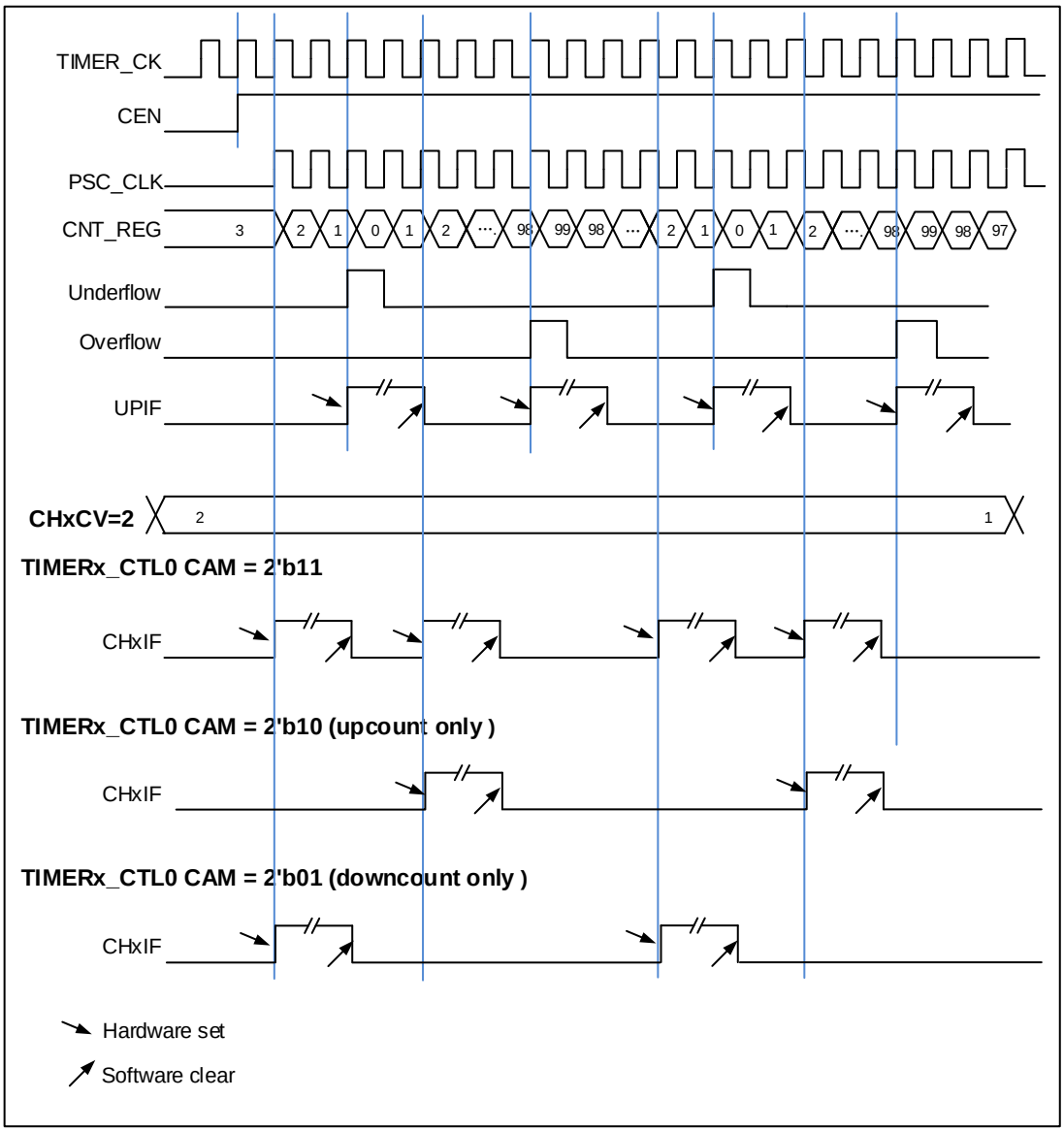
将 TIMERx_SWEVG 寄存器的 UPG 位置 1 可以初始化计数值为 0，并产生一个更新事件，而无需考虑计数器在中央模式下是向上计数还是向下计数。

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 18-49. 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 18-49. 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 18-49. 中央计数模式计数器时序图



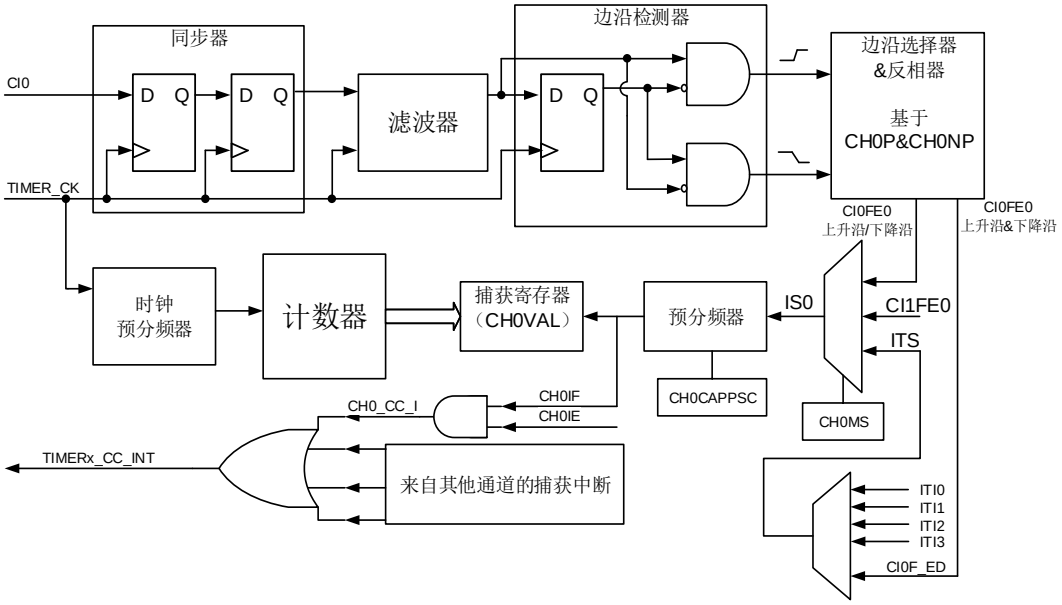
输入捕获/输出比较通道

通用定时器 L0 拥有四个独立的通道用于捕获输入或比较输出是否匹配。每个通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

■ 通道输入捕获功能

通道输入捕获功能允许通道测量一个波形时序，频率，周期，占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，TIMERx_CHxCV 寄存器会捕获计数器当前的值，同时 CHxIF 位被置 1，如果 CHxIE = 1 则产生通道中断。

图 18-50. 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（仅限于 CI0）。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 18-51. 输出比较原理（x=0,1,2,3）

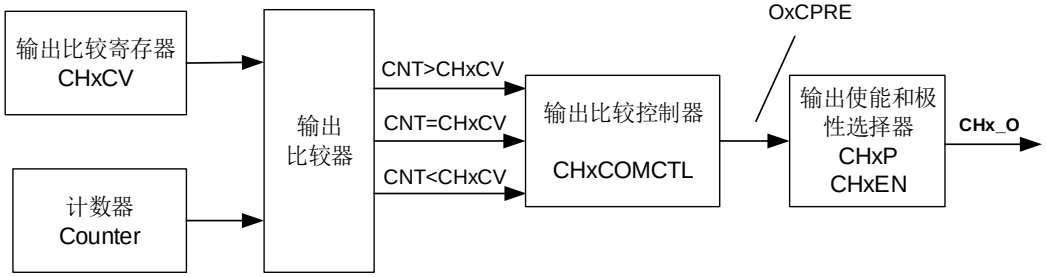


*图18-51. 输出比较原理（x=0,1,2,3）*给出了输出比较的原理电路。通道输出信号CHx_O与OxCPRE信号（详情请见*通道输出准备信号*）的关系描述如下：OxCPRE信号高电平有效，CHx_O 的输出情况与 OxCPRE 信号，CHxP 位和 CHxEN 位有关（具体情况请见 TIMERx_CHCTL2寄存器中的描述）。例如，当设置CHxP=0（CHx_O高电平有效，与OxCPRE输出极性相同）、CHxEN=1（CHx_O输出使能）时：

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 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，被置低电平或者翻转。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/翻转）；

设置CHxP位来选择有效电平的极性；

设置CHxENN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：

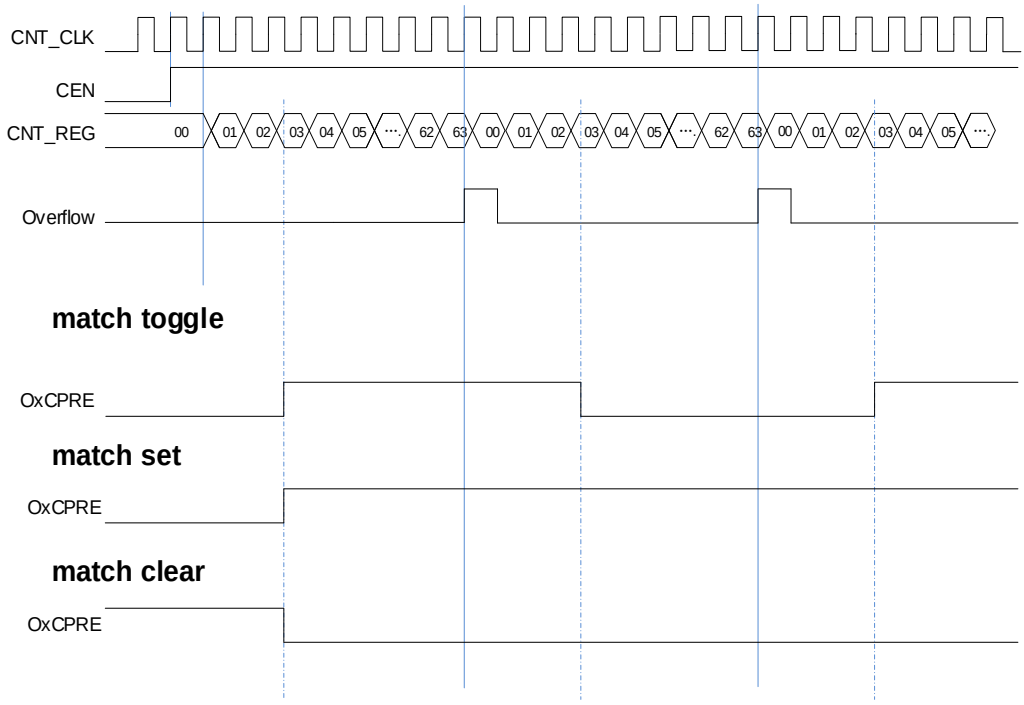
CHxVAL可以在运行时根据你所期望的波形而改变。

第五步：设置CEN位使能定时器。

*图 18-52. 三种输出比较模式*显示了三种比较输出模式：翻转/置高电平/置低电平，CAR=0x63,

CHxVAL=0x3。

图 18-52. 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 18-53. 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 18-54. 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。如果 TIMERx_CHxCV 寄存器的值等于 0，通道输出一直为无效电平。

图 18-53. EAPWM 时序图

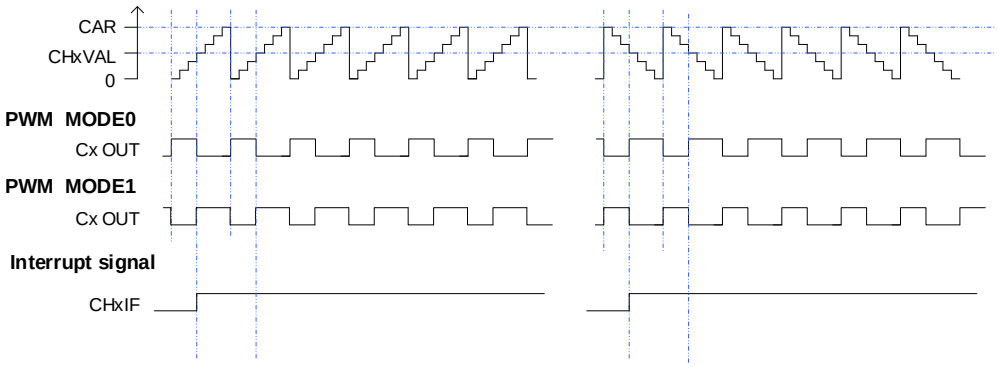
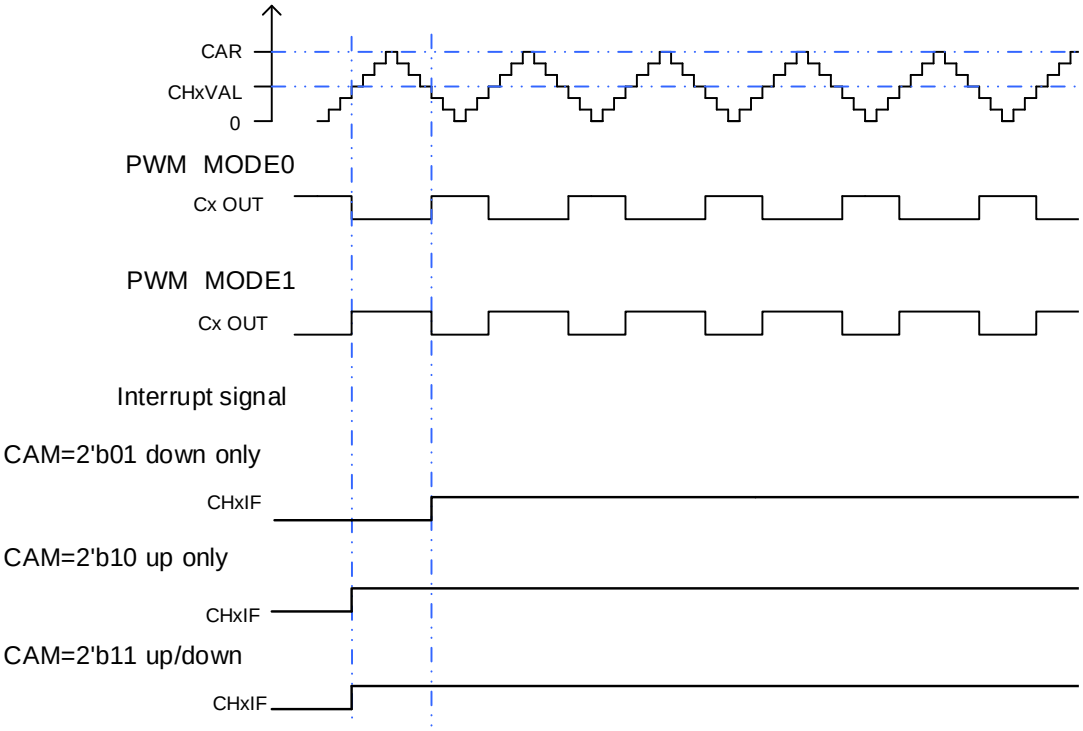


图 18-54. CAPWM 时序图



通道输出准备信号

根据*图 18-51. 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

正交译码器功能使用由 TIMERx_CH0 和 TIMERx_CH1 引脚生成的 CI0FE0 和 CI1FE1 正交信号各自相互作用产生计数值。在每个输入源改变期间，DIR 位会发生改变。输入源可以是只有 CI0FE0，可以只有 CI1FE1，或着可以同时有 CI0FE0 和 CI1FE1，通过设置 SMC=0x01，0x02 或 0x03 来选择使用哪种模式。计数器计数方向改变的机制如*表 18-7. 不同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 18-7. 不同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
正交译码模式0 SMC[2:0]=3’b001	CI1FE1=1	向下	向上	-	-
	CI1FE1=0	向上	向下	-	-
正交译码模式1 SMC [2:0]=3’b010	CI0FE0=1	-	-	向上	向下
	CI0FE0=0	-	-	向下	向上
正交译码模式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 18-55. 在正交译码模式 2 且 CI0FE0 极性不反相时计数器行为

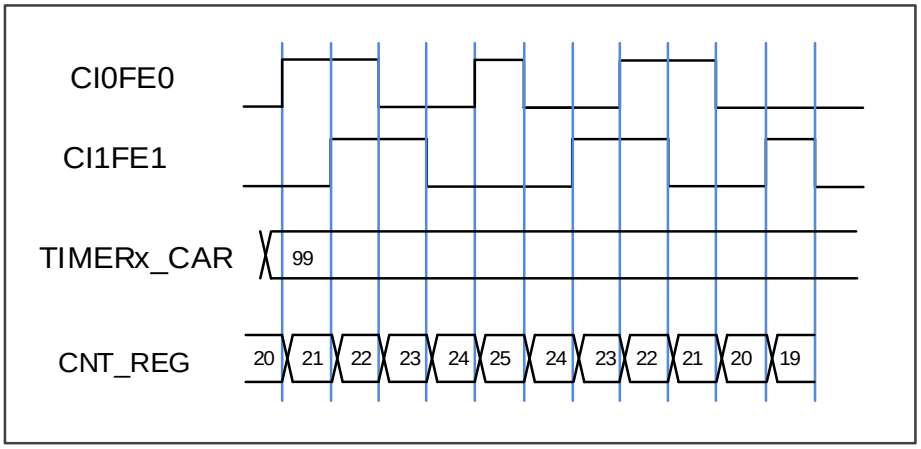
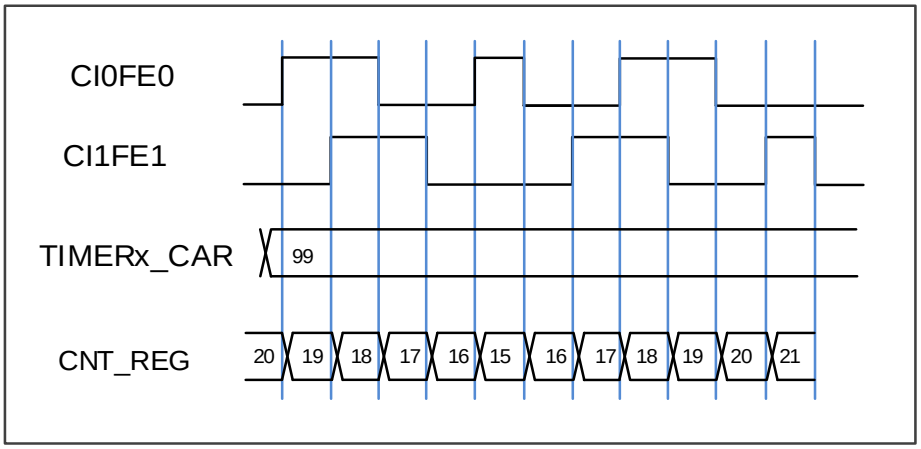


图 18-56. 在正交译码模式 2 且 CI0FE0 极性反相时计数器行为



霍尔传感器接口功能

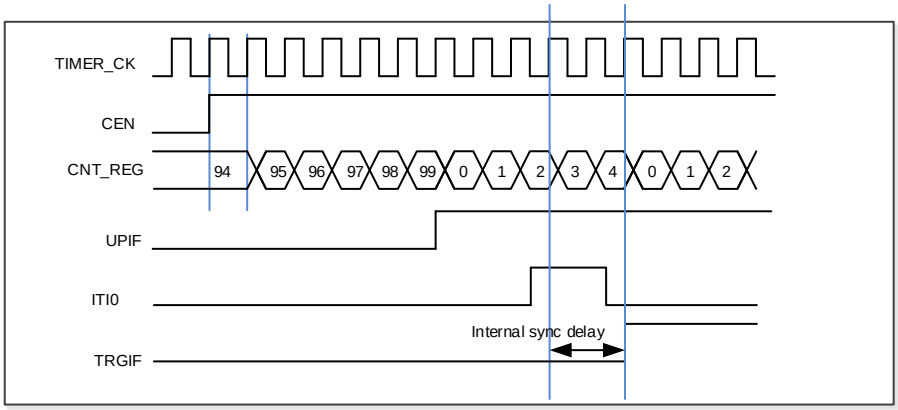
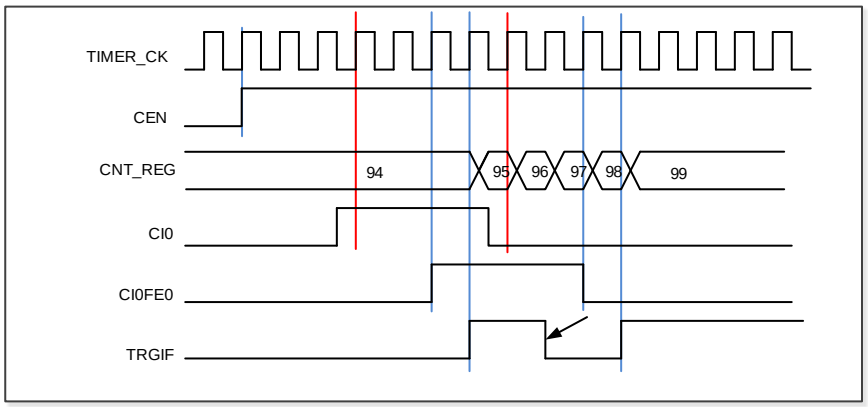
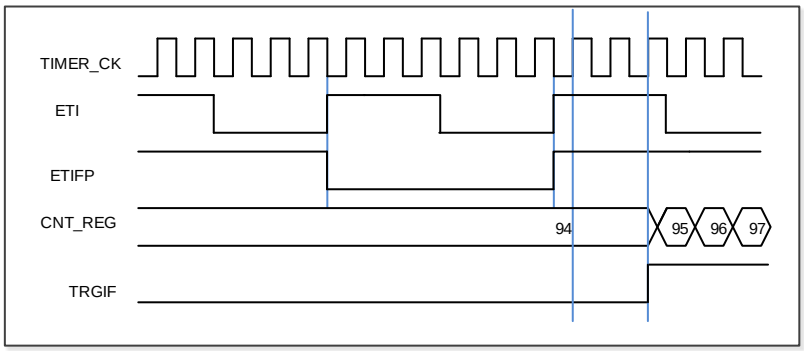
参考*高级定时器(TIMERx,x=0,7)霍尔传感器接口功能*。

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 18-8. 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(1)	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ETIFP（滤波后的外部触发输入ETI），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

<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></td><td colspan="4">图 18-57. 复位模式下的控制电路
 
</td></tr>
<tr><td rowspan="2">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）
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 18-58. 暂停模式下的控制电路
 
</td></tr>
<tr><td rowspan="2">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>
<tr><td colspan="4">图 18-59. 事件模式下的控制电路
 
</td></tr>
</table>

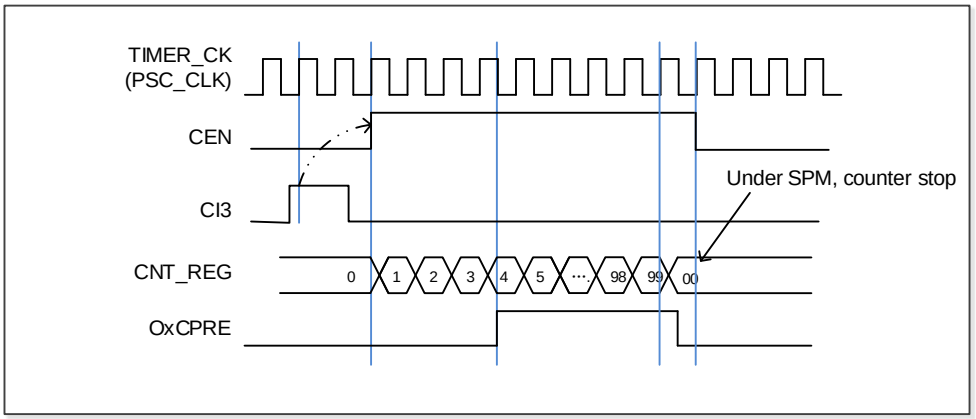
单脉冲模式

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。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。

图 18-60. 单脉冲模式，TIMERx_CHxCV = 0x04 TIMERx_CAR=0x99



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。通过使能 DMA 请求，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）

TIMER1基地址：0x4000 0000

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 状态。
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模式或译码器模式时，该位为只读。
3	SPM	单脉冲模式 0：更新事件发生后，计数器继续计数 1：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，如果 UPG 位被置 1 或者复位模式不会产生更新事件，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 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 位域。

13:12 ETPSC[1:0]

外部触发预分频

外部触发信号 ETI 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 ETIP 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：内部触发输入 0（ITI0） 001：I 内部触发输入 1（ITI1） 010：内部触发输入 2（ITI2） 011：内部触发输入 3（ITI3） 100：CI0 的边沿标志位（CI0F_ED） 101：滤波后的通道 0 输入（CI0FE0） 110：滤波后的通道 1 输入（CI1FE1） 111：滤波后的外部触发输入（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。 当通道 0 在输入模式下时，通过读 TIMERx_CH0CV 寄存器可以清零该位。 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
CH1COM CEN	CH1COMCTL[2:0]			CH1COM SEN	保留	CH1MS[1:0]		CH0COM CEN	CH0COMCTL[2:0]			CH0COM SEN	保留	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	保留	必须保持复位值。
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，此模式仅工作在内部触发源输入被选中时（由 TIMERx_SMCFGFG 寄存器的 TRGS 位选择）。
7	CH0COMCEN	通道 0 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，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 此位定义了输出参考信号 O0CPRE 的动作，而 O0CPRE 决定了 CH0_O 的值。O0CPRE 高电平有效，而 CH0_O 的有效电平取决于 CH0P 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 为有效电平，否则为无效电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为无效电平，否则为有效电平。 111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为无效电平，否则为有效电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为有效电平，否则为无效电平。 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	保留	必须保持复位值。
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. 此模式仅工作在内部触发输入被选中时（通过设置 TIMERx_SMCFGFG 寄存器的 TRGS 位）

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：输入捕获预分频器系数为 2 10：输入捕获预分频器系数为 4 11：输入捕获预分频器系数为 8
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	保留	CH3MS[1:0]		CH2COMCEN	CH2COMCTL[2:0]			CH2COMSEN	保留	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	保留	必须保持复位值。
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，此模式仅工作在内部触发源输入被选中时（由 TIMERx_SMCFGFG 寄存器的 TRGS 位选择）。
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 此位定义了输出参考信号 O2CPRE 的动作，而 O2CPRE 决定了 CH2_O 的值。O2CPRE 高电平有效，而 CH2_O 的有效电平取决于 CH2P 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_CH0CV 时，O0CPRE 为有效电平，否则为无效电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为无效电平，否则为有效电平。 111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为无效电平，否则为有效电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为有效电平，否则为无效电平。 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	保留	必须保持复位值。
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。此模式仅工作在内部触发输入被选中时（通过设置 TIMERx_SMCFGFG 寄存器的 TRGS 位）

输入捕获模式：

位/位域	名称	描述
31:16	保留	必须保持复位位值。
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_{TIMER_CK}
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：输入捕获预分频器系数为 2 10：输入捕获预分频器系数为 4 11：输入捕获预分频器系数为 8
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。 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 高电平有效 1：通道 0 低电平有效 当通道 0 配置为输入模式时，此位定义了通道 0 输入信号极性。[CH0NP，CH0P] 将选择 CI0FE0 或者 CI1FE0 的有效边沿或者捕获极性。 00：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。 01：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。 10：保留。 11：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。 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 的值被装入当前预分频寄存器。

计数器自动重载寄存器(TIMERx_CAR)

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。 **注意：**在定时器被配置为输入捕获模式时，该寄存器需要被配置成一个大于用户期望值的非 0 值(例如 0xFFFF)。

通道 0 捕获/比较值寄存器(TIMERx_CH0CV)

地址偏移：0x34
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值 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。

通道 2 捕获/比较值寄存器(TIMERx_CH2CV)

地址偏移：0x3C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

保留

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH2VAL[15:0]	通道 2 的捕获或比较值 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）

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’b00000 到 5’b10001.
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。 起始地址 = TIMERx_CTL0 + DMATA*4。

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。

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。

通道输入重映射寄存器(TIMERx_IRMP)

地址偏移：0x50
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														CI0_RMP[1:0]	
														rw	

位/位域	名称	描述
31:2	保留	必须保持复位值。
1:0	CI0_RMP[1:0]	通道0输入重映射 00：通道0输入连接到GPIO(TIMER1_CH0) 01：通道0输入连接到LXTAL 10：通道0输入连接到HXTAL/128时钟 11：通道 0 输入连接到 CK_OUT

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。

18.3. 基本定时器（TIMERx, x=5,6）

18.3.1. 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。

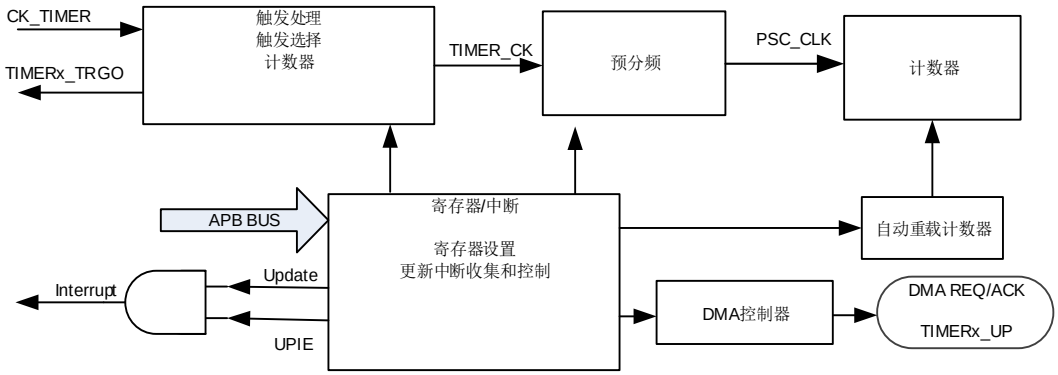
18.3.2. 主要特征

- 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

18.3.3. 结构框图

*图 18-61. 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 18-61. 基本定时器结构框图



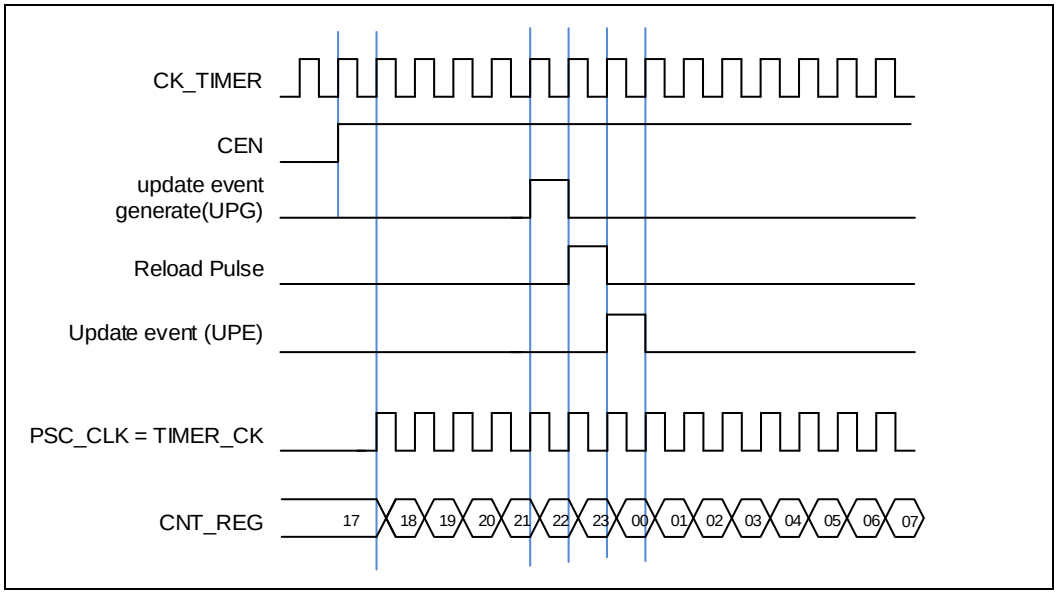
18.3.4. 功能说明

时钟源配置

基本定时器只能由内部时钟源CK_TIMER驱动（来自RCU模块）。

TIMER_CK用来驱动计数器预分频器。当CEN置位，TIMER_CK经过预分频器（预分频值由TIMERx_PSC寄存器确定）产生PSC_CLK。

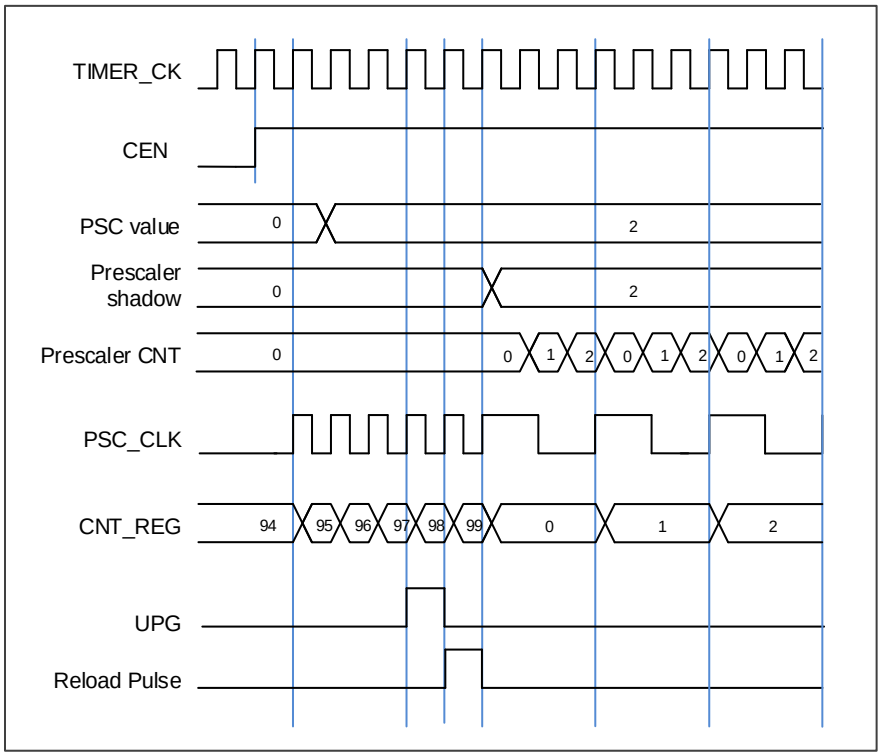
图 18-62. 内部时钟分频为 1 时，计数器的时序图



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 18-63. 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 18-64. 向上计数时序图，PSC=0/2

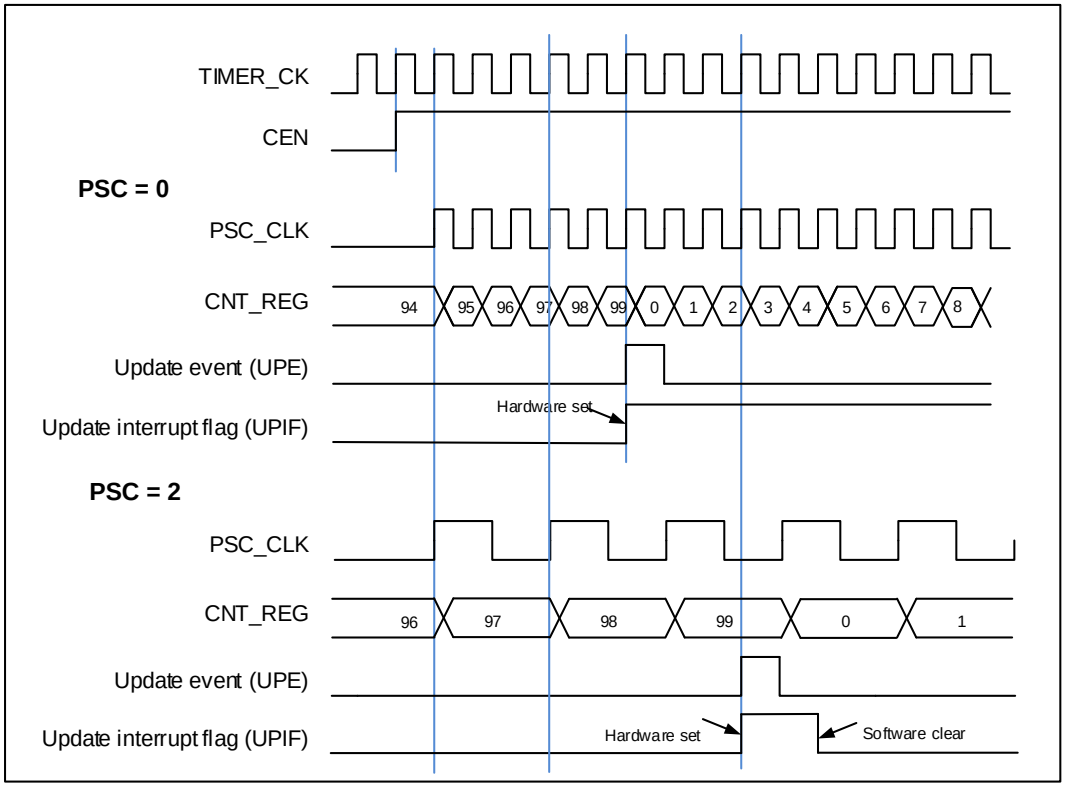
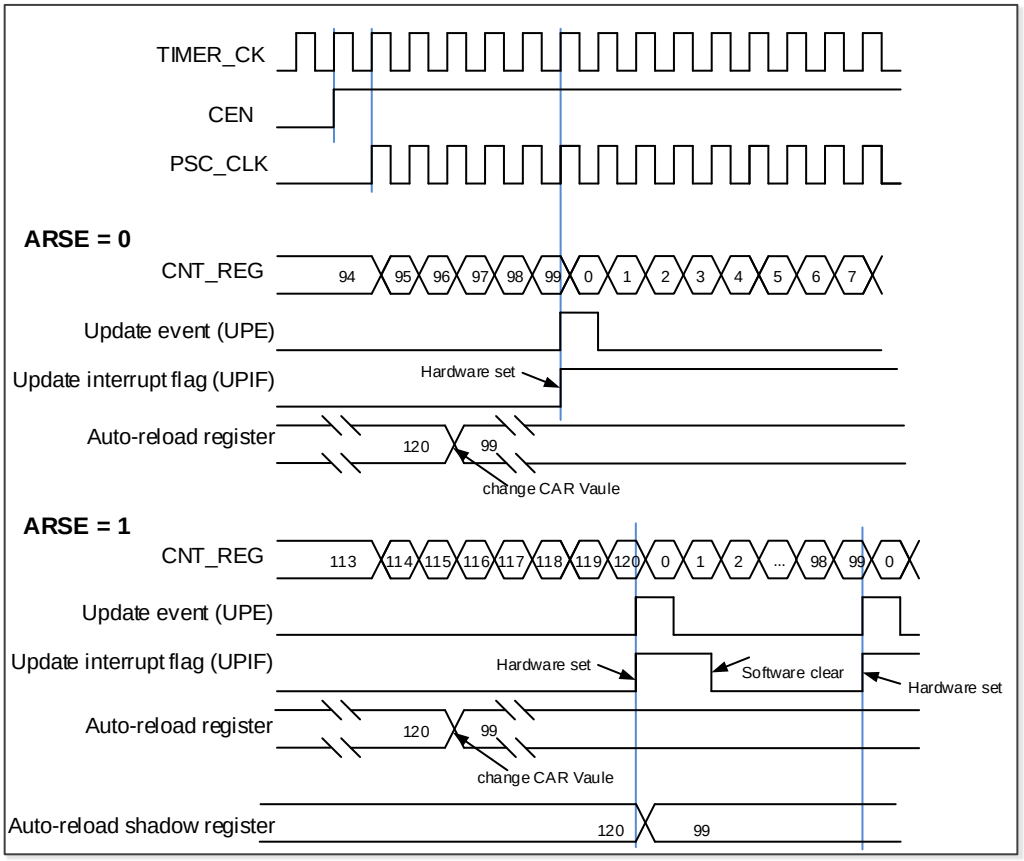


图 18-65. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



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。

18.3.5. 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。

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								ARSE	保留			SPM	UPS	UPDIS	CEN
								rw				rw	rw	rw	rw

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：更新事件发生后，计数器继续计数 1：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 – 复位模式产生一个更新事件

		1：更新事件禁能。 注意：当该位被置 1 时，如果 UPG 位被置 1 或者复位模式不会产生更新事件，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]	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 100~111：保留。
3:0	保留	必须保持复位值。

DMA 和中断使能寄存器（TIMERx_DMAINTEN）

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

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。

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
															rc_w0

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。

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															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 的值被装入当前预分频寄存器。

计数器自动重载寄存器(TIMERx_CAR)

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

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

19. 通用同步异步收发器（USART）

19.1. 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（PCLK, CK_SYS, LXTAL, IRC16M）时钟进行分频产生USART发送和接收所需的特定频率。

USART不仅支持标准的异步收发模式，还实现了一些其他类型的串行数据交换模式，如红外编码规范，SIR，智能卡协议，LIN，半双工以及同步模式。它还支持多处理器通信和硬件流控操作（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。

19.2. 主要特征

- 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PCLK和独立于PCLK时钟的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100MHz，过采样为8，最高速度可达12.5MBits/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
- 支持硬件Modem流控操作（CTS/RTS）和RS485驱动使能
- 借助集中式DMA，实现可配置的多级缓存通信
- 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标记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）
  - 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）
  - 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

USART0，USART1和USART2完全实现上述功能。

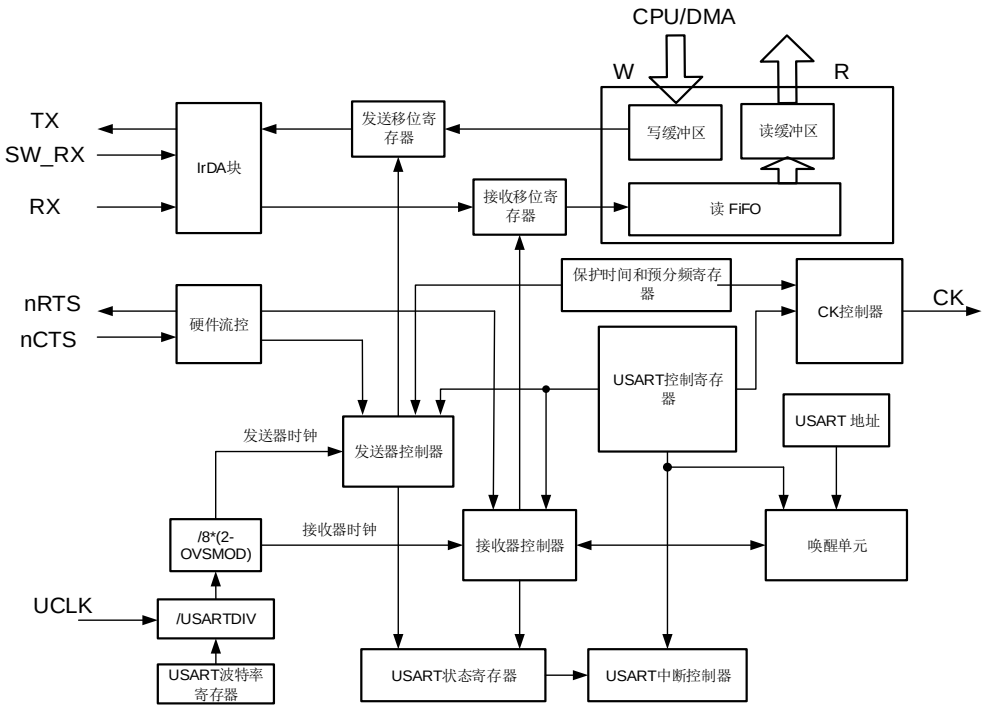
19.3. 功能描述

USART 接口通过*表 19-1. USART 重要引脚描述*中主要引脚从外部连接到其他设备。

表 19-1. 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

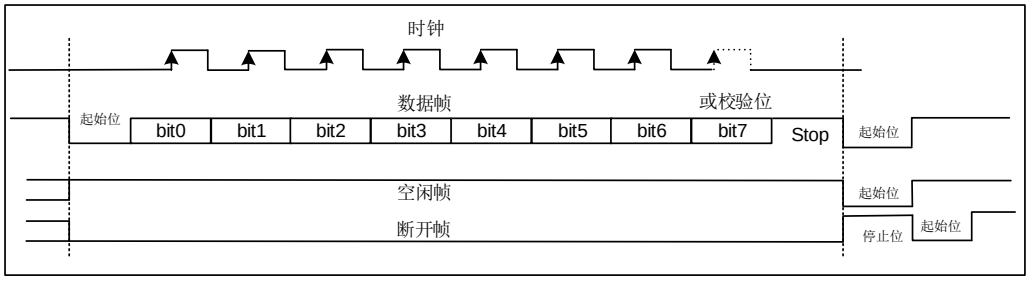
图 19-1. USART 模块内部框图



19.3.1. USART 帧格式

USART数据帧开始于起始位，结束于停止位。USART_CTL0寄存器中WL位可以设置数据长度。将USART_CTL0寄存器中PCEN置位，最后一个数据位可以用作校验位。若WL位为0，第七位为校验位。若WL位置1，第八位为校验位。USART_CTL0寄存器中PM位用于选择校验位的计算方法。

图19-2. USART字符帧（8数据位和1停止位）



在发送和接收中，停止位可以在USART_CTL1寄存器中STB[1:0]位域中配置。

表 19-2. 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。

19.3.2. 波特率发生

波特率分频系数是一个16位的数字，包含12位整数部分和4位小数部分。波特率发生器使用这两部分组合所得的数值来确定波特率。由于具有小数部分的波特率分频系数，将使USART能够产生所有标准波特率。

波特率分频系数（USARTDIV）与系统时钟具有如下关系：

如果过采样率是16，公式为：

$$USARTDIV=\frac{UCLK}{16\times Baud\ Rate} \quad (19\text{-}1)$$

如果过采样是8，公式为：

$$USARTDIV=\frac{UCLK}{8\times Baud\ Rate} \quad (19\text{-}2)$$

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。

19.3.3. USART 发送器

如果USART_CTL0寄存器的发送使能位（TEN）被置位，当发送数据缓冲区不为空时，发送器将会通过TX引脚发送数据帧。TX引脚的极性可以通过USART_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。

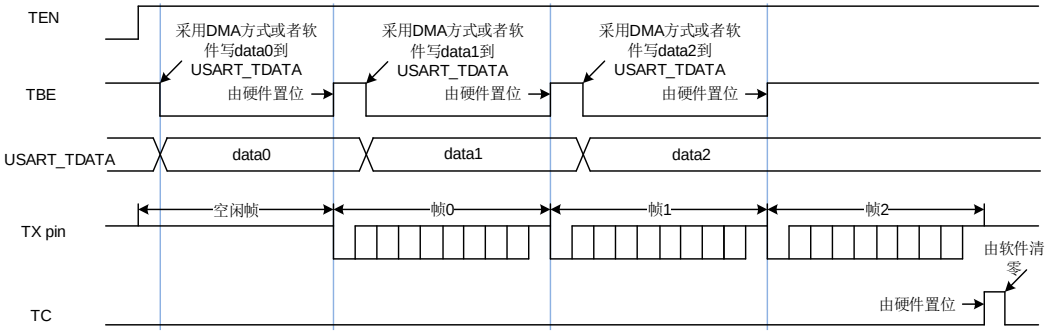
系统上电后，TBE默认为高电平。在USART_STAT寄存器中TBE置位时，数据可以在不覆盖前一个数据的情况下写入USART_TDATA寄存器。当数据写入USART_TDATA寄存器，TBE位将被清0。在数据由USART_TDATA移入移位寄存器后，该位由硬件置1。如果数据在一个发送过程正在进行时被写入USART_TDATA寄存器，它将首先被存入发送缓冲区，在当前发送过程完成时传输到发送移位寄存器中。如果数据在写入USART_TDATA寄存器时，没有发送过程正在进行，TBE位将被清零然后迅速置位，原因是数据被立刻传输到发送移位寄存器。

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 19-3. 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. 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 19-3. 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。通过将USART_INTC寄存器的TCC位置1可以将TC位清零。

当SBKCMD置位时，会发送一个断开帧，发送完成后，SBKCMD将被清0。

19.3.4. 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。

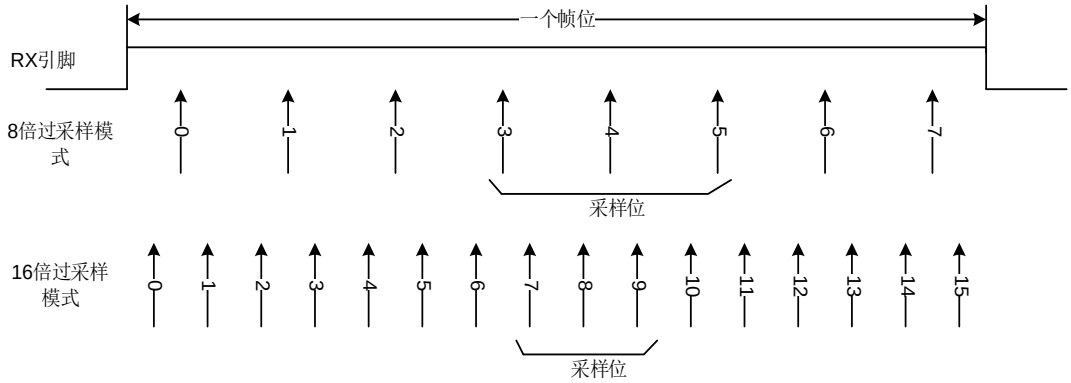
当接收到一个数据帧，USART_STAT寄存器中的RBNE置位，如果设置了USART_CTL0寄存器中相应的中断使能位RBNEIE，将会产生中断。在USART_STAT寄存器中可以观察接收状态标志。

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，不管是起始位，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 19-4. 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_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。根据停止位的配置，有以下几种情形：

- 0.5个停止位：0.5个停止位时，停止位不采样。
- 1个停止位：1个停止位时，在停止位的中间进行采样。
- 1.5个停止位：1.5个停止位时，1.5个停止位可以分为两个部分：0.5个停止位的部分不采样和1个停止位的中间进行采样。
- 2个停止位：2个停止位时，如果在第一个停止位期间检测到帧错误，帧错误标志置位，则第二个停止位不检测帧错误。如果第一个停止位期间没有检测到帧错误，则在第二个停止位继续检测帧错误。

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。如果置位RBNEIE或使能DMA的情况下并置位USART_CTL2寄存器中ERRIE位，将产生中断。

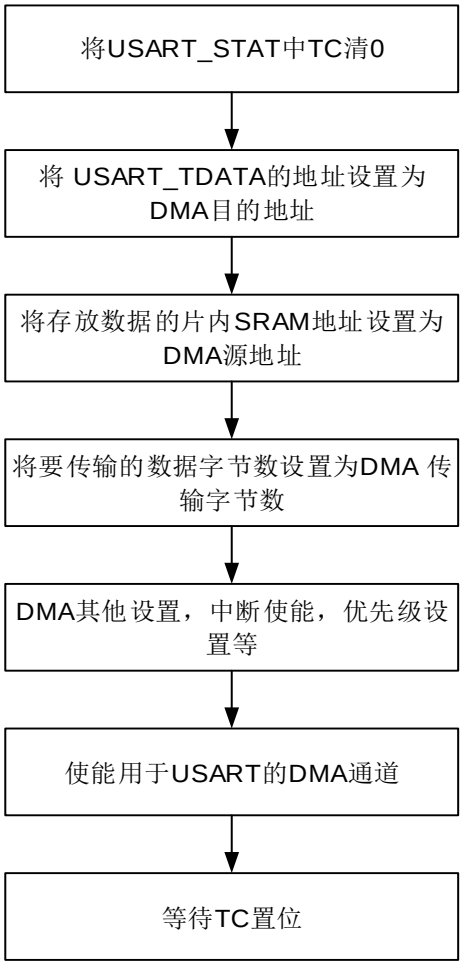
在一个接收过程中，NERR、PERR、FERR、ORERR总是分别和RBNE同时置位。如果没有使能DMA，软件需检查RBNE中断是否由NERR、PERR、FERR或者ORERR置位产生。

19.3.5. DMA 方式访问数据缓冲区

为减轻处理器的负担，可以采用DMA访问发送缓冲区或者接收缓冲区。置位USART_CTL2寄存器中DENT位可以使能DMA发送，置位USART_CTL2寄存器中DENR位可以使能DMA接收。

当 DMA 用于 USART 发送时，DMA 将数据从片内 SRAM 传送到 USART 的数据缓冲区。配置步骤如*图 19-5. 采用 DMA 方式实现 USART 数据发送配置步骤*所示。

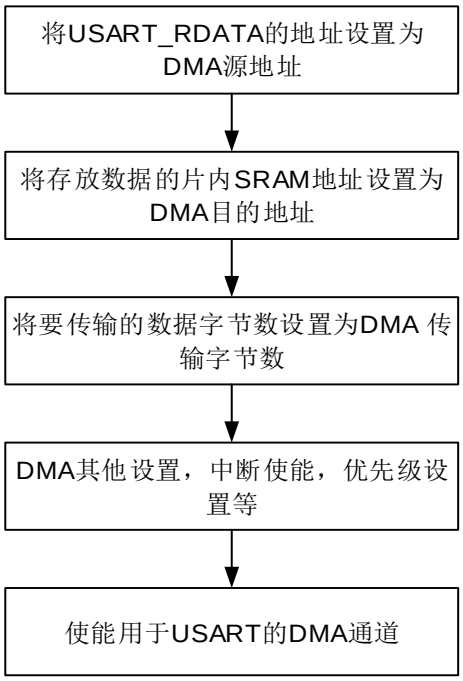
图 19-5. 采用 DMA 方式实现 USART 数据发送配置步骤



所有数据帧都传输完成后，USART_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 19-6. 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_STAT 寄存器中的错误标志位（FERR、ORERR 和 NERR）置位时将产生中断。

图 19-6. 采用 DMA 方式实现 USART 数据接收配置步骤

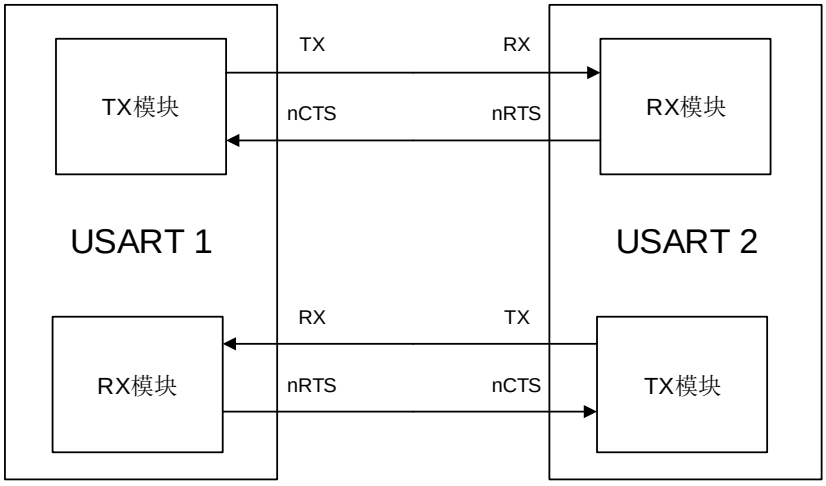


当USART接收到的数据数量达到了DMA传输数据数量，DMA模块将产生传输完成中断。

19.3.6. 硬件流控制

硬件流控制功能通过nCTS和nRTS引脚来实现。通过将USART_CTL2寄存器中RTSEN位置1来使能RTS流控，将USART_CTL2寄存器中CTSEN位置1来使能CTS流控。

图 19-7. 两个 USART 之间的硬件流控制



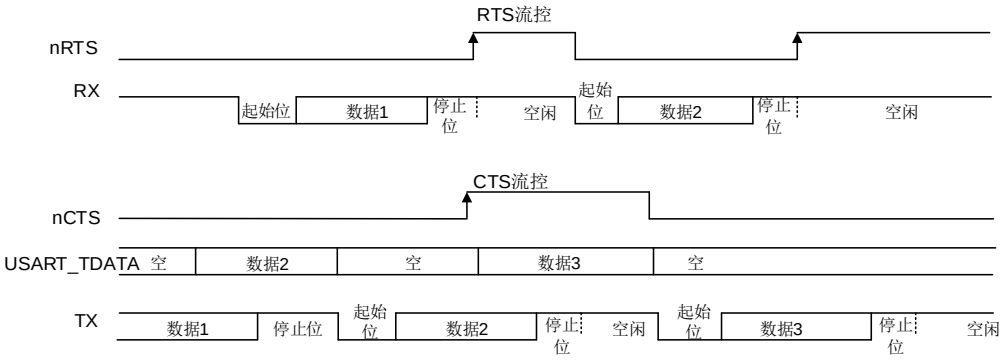
RTS 流控

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。

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_STAT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 19-8. 硬件流控制



RS485 驱动使能

驱动使能功能通过设置USART_CTL2控制寄存器的DEM位来打开。它允许用户通过DE（Driver Enable）信号激活外部收发器控制。提前时间是驱动使能信号和第一个字节的起始位之间的时间间隔。这个时间可以在USART_CTL0控制器的DEA[4:0]位域中进行设置。滞后时间是一个发送信息最后一个字节的停止位与释放DE信号之间的时间间隔。这个时间可以在USART_CTL0控制寄存器的DED[4:0]位域中进行设置。DE信号的极性可以通过USART_CTL2控制寄存器的DEP位进行设置。

19.3.7. 多处理器通信

在多处理器通信中，多个USART被连接成一个网络。对于一个设备来说，监视所有来自RX引脚的消息，是一种巨大的负担。为减轻设备负担，软件可以通过将USART_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匹配检测。

设备默认使用空闲总线检测方法唤醒USART。如果RWU位为0，RX引脚检测到空闲帧，USART_STAT寄存器中的IDLEF位会置位。如果RWU位置位，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，当它是被空闲帧唤醒时，USART_STAT寄存器中IDLEF位不会被置1。

当USART_CTL0寄存器中WM[1:0]为0b01，USART通过地址匹配方式被唤醒。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（通过USART_CTL1寄存器的ADDM位配置）与USART_CTL1寄存器中的ADDR_DATA位域相同，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_DATA位域不相同，硬件会置位RWU并自动进入静默模式。在这种情况下，RBNE不会置位。

当USART_CTL0寄存器中WM[1:0]为0b1x，USART通过地址匹配方式被唤醒，在这种情况下，ADDM位必须置位。如果数据帧和USART_CTL1寄存器中ADDR_DATA[7:0]位域相同，硬件会将RWU位清零，并退出静默模式。接收到将USART唤醒的数据帧，RBNE将置位。如果数据帧与USART_CTL1寄存器中的ADDR_DATA位域不相同，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。当USART通过地址匹配唤醒，如果ADDM位置位，且接收帧为7位数据，其中低6位将与ADDR_DATA[5:0]比较。如果ADDM位置位，且接收帧为9位数据，其中低8位将与ADDR_DATA[7:0]比较。当USART通过数据匹配唤醒，如果接收帧为7位数据，其中低6位将与ADDR_DATA[5:0]比较。如果接收帧为9位数据，其中低8位将与ADDR_DATA[7:0]比较。

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。

19.3.8. LIN 模式

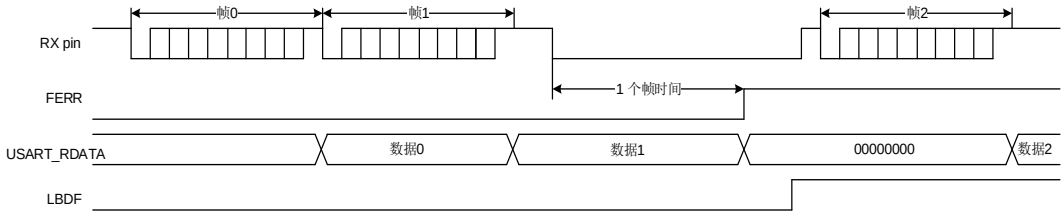
将USART_CTL1寄存器的LMEN置位即可使能本地互联网络模式。在LIN模式下，USART_CTL1寄存器中CKEN，STB[1:0]和USART_CTL2的SCEN，HDEN，IREN位都应该被清0。

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。

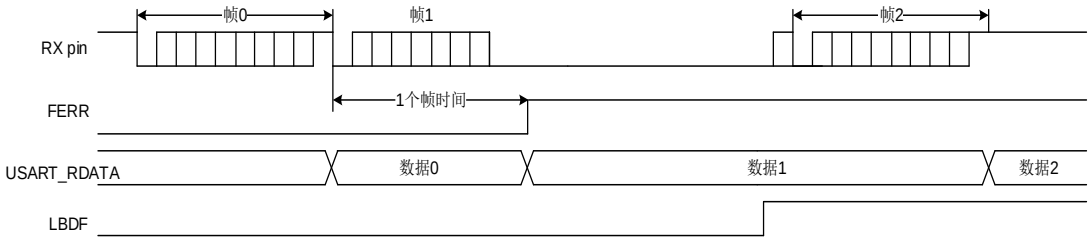
如*图 19-9. 空闲状态下检测断开帧*所示，如果断开帧发生在空闲状态下，USART 接收器会接收到一个全 0 数据帧，同时 FERR 置位。

图 19-9. 空闲状态下检测断开帧



如*图 19-10. 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 19-10. 数据传输过程中检测断开帧



19.3.9. 同步通信模式

USART支持主机模式下的全双工同步串行通信，可以通过置位USART_CTL1的CKEN位来使能。在同步模式下，USART_CTL1的LMEN和USART_CTL2的SCEN，HDEN，IREN位应被清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 19-11. 同步模式下的 USART 示例

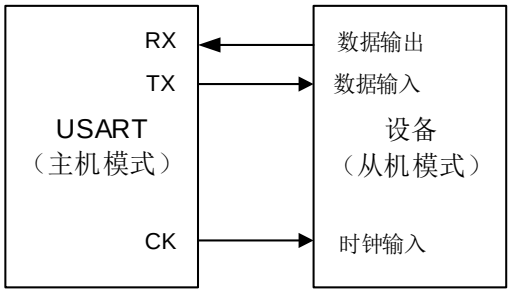
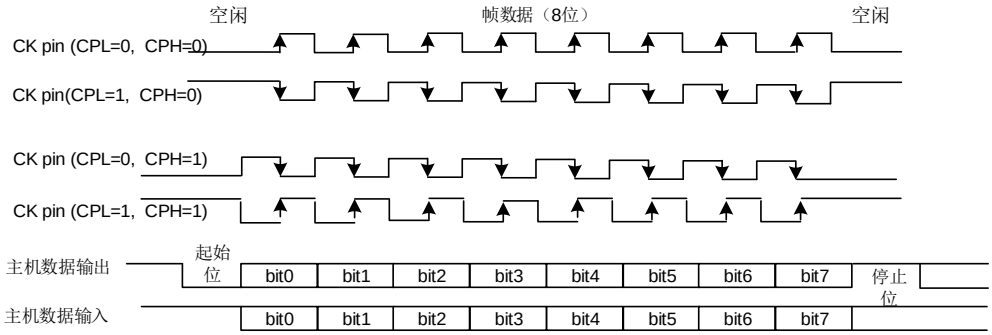


图 19-12. 8-bit 格式的 USART 同步通信波形（CLEN=1）

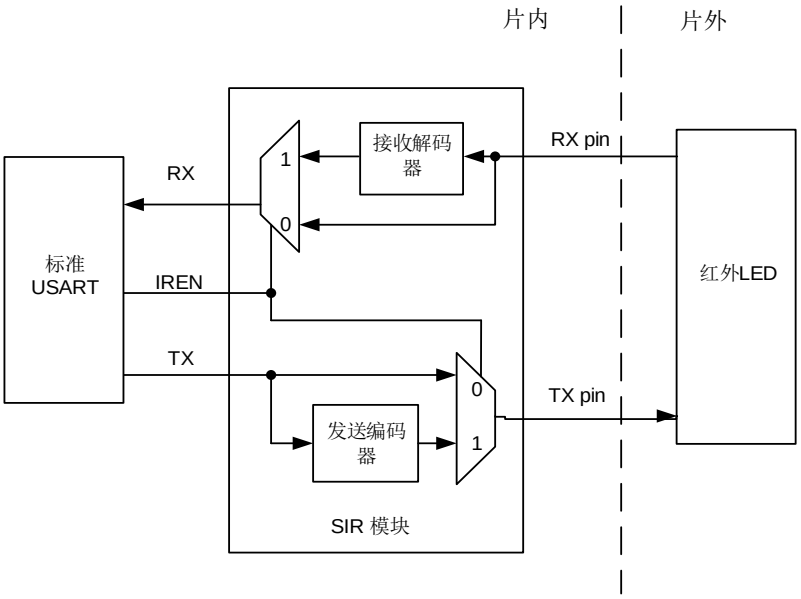


19.3.10. 串行红外（IrDA SIR）编解码功能模块

串行红外编解码功能通过置位USART_CTL2寄存器中IREN使能。在IrDA模式下，USART_CTL1寄存器的LMEN，STB[1:0]，CKEN位和USART_CTL2寄存器的HDEN，SCEN位应被清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。

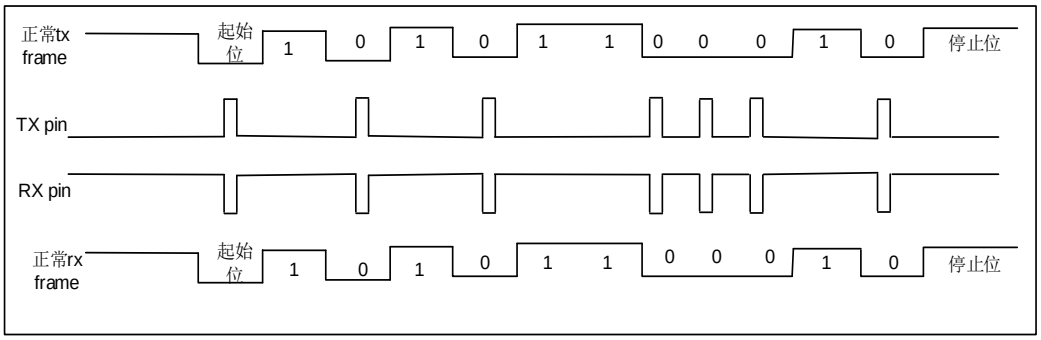
图 19-13. IrDA SIR ENDEC 模块



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 19-14. 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。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。

19.3.11. 半双工通信模式

通过设置USART_CTL2寄存器的HDEN位，可以使能半双工模式。在半双工通信模式下，USART_CTL1寄存器的LMEN，CKEN位和USART_CTL2寄存器的SCEN，IREN位应被清零。

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，TX引脚应被配置为IO管脚。通信冲突应由软件处理。当TEN被置位时，在数据寄存器中的数据将会被发送。

19.3.12. 智能卡（ISO7816-3）模式

智能卡模式是一种异步通信模式，支持ISO7816-3协议。支持字节模式（T=0）和块模式（T=1）。将USART_CTL2寄存器的SCEN位置1，即可使能智能卡模式。在智能卡模式下，USART_CTL1寄存器的LMEN位和USART_CTL2的HDEN，IREN位应该清0。

如果CKEN位被置位，USART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 19-15. 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2.5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。

19.3.13. 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

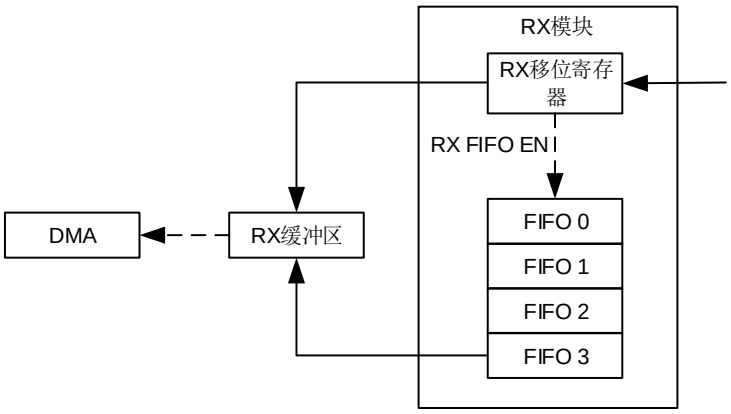
位。USART_RT寄存器必须被设置成与2个字节超时所对应的值。在最后一个停止位被接收后，当接收线在这期间是空闲的，将产生一个中断，通知软件当前块接收已经完成。

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。

19.3.14. 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 19-16. 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。

19.3.15. 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*复位和时钟单元（RCU）*）。

当使用RBNE标准中断时，必须在进入深度睡眠模式前将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前，必须禁用DMA。在进入深度睡眠模式前，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。

19.3.16. USART 中断

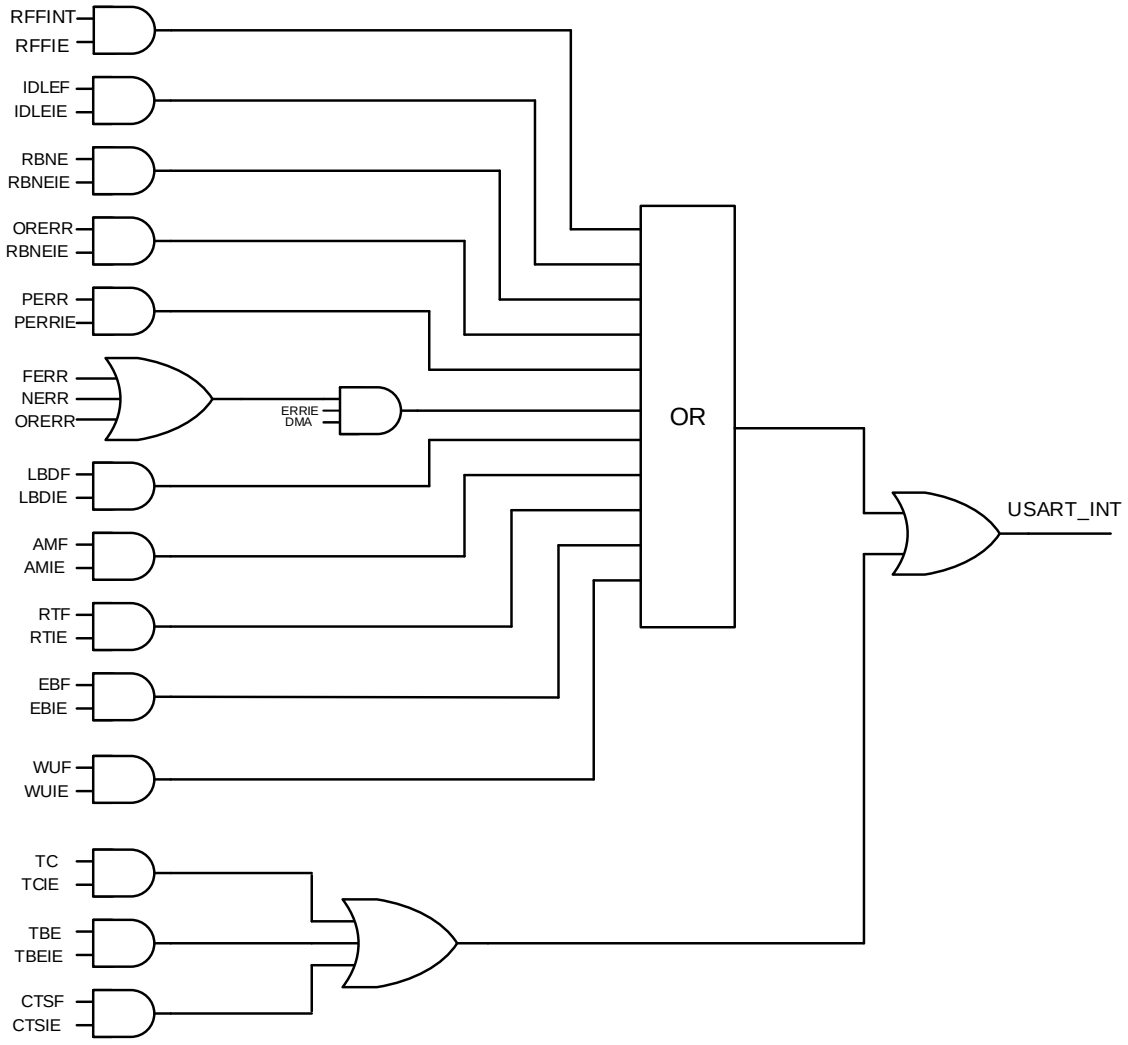
USART 中断事件和标志如*表 19-3. USART 中断请求*所示：

表 19-3. 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
CTS标志	CTSF	CTSIE
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
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，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 19-17. USART 中断映射框图



19.4. USART 寄存器

USART0基地址：0x4001 3800
USART1基地址：0x4000 4400
USART2基地址：0x4000 4800

19.4.1. 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			WM1	EBIE	RTIE	DEA[4:0]					DED[4:0]				
			rw	rw	rw			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。
28	WM1	从静默模式唤醒方法，该位和bit[11]共同决定唤醒方法。 00：空闲线。 01：地址匹配。 1x：数据匹配。 当USART被使能（UEN=1）时，该位域不能被改写。
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。
25:21	DEA[4:0]	驱动使能置位时间。 这些数字用来定义DE（驱动使能）信号的置位与第一个字节的起始位之间的时间间隔。它以采样时间为单位（1/8或1/16位时间），可以通过OVSMOD位来配置。 当USART被使能（UEN=1）时，该位域不能被改写。
20:16	DED[4:0]	驱动使能置低时间。 这些位用来定义一个发送信息最后一个字节的停止位与置低DE（驱动使能）信号之间的时间间隔。它以采样时间为单位（1/8或1/16位时间），可以通过OVSMOD位来配置。 当USART被使能（UEN=1）时，该位域不能被改写。

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	WM0	从静默模式唤醒方法，该位和bit[28]位共同决定唤醒方法。 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。

19.4.2. 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_DATA[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_DATA[7:0]	USART的节点数据或地址。 这些位给出USART的节点数据或地址。 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或数据匹配检测。接收到的数据帧将和这些位进行比较。 当WM[1:0] = 01，如果ADDM位为零，只有ADDR[3:0]位的值用于比较。 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_DATA[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	LBLEN	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。

19.4.3. 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
DEP	DEM	DDRE	OVRD	OSB	CTSIE	CTSEN	RTSEN	DENT	DENR	SCEN	NKEN	HDEN	IRLP	IREN	ERRIE
rw	rw	rw	rw	rw	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	保留	必须保持复位值。
15	DEP	驱动使能的极性选择模式。 0：DE信号高有效。 1：DE信号低有效。 USART被使能（UEN=1）时，该位域不能被改写。
14	DEM	驱动使能模式。 用户使能该位以后，可以通过DE信号对外部收发器进行控制。DE信号是从RTS管脚输出的。 0：DE功能禁用。 1：DE功能开启。 USART被使能（UEN=1）时，该位域不能被改写。
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，直到相应的错误标志位被清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	CTSIE	CTS中断使能。 0：CTS中断屏蔽。 1：当USART_STAT的CTS位置位时，会产生中断。
9	CTSEN	CTS使能。 0：CTS 硬件流控禁用。 1：CTS 硬件流控被使能。 USART被使能（UEN=1）时，该位域不能被改写。
8	RTSEN	RTS使能。 0：RTS硬件流控禁用。 1：RTS硬件流控被使能，只有当接收缓冲区有空间的时候，才会请求下一个数据。 USART被使能（UEN=1）时，该位域不能被改写。
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。

19.4.4. 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。

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
BRR [15:4]												BRR[3:0]			
rw												rw			

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。

19.4.5. 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。

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
GUAT[7:0]								PSC[7:0]							
rw								rw							

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	GUAT[7:0]	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。

19.4.6. 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。

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。

19.4.7. 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。

19.4.8. 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		保留	EBF	RTF	CTS	CTSF	LBDF	TBE	TC	RBNE	IDLEF	ORERR	NERR	FERR	PERR
			r	r	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WM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	CTS	CTS电平。 这个值等于nCTS输入引脚电平的反向拷贝。 0：nCTS输入引脚高电平。 1：nCTS输入引脚低电平。
9	CTSF	CTS变化标志。 0：nCTS状态线没有变化。 1：nCTS状态线发生变化，如果USART_CTL2寄存器的CTSIE位置位，将引发中断。 当nCTS输入变化时，由硬件置位。 通过向USART_INTC寄存器的CTSC位写1，清零该位。
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。

19.4.9. 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	CTSC	LBDC	保留	TCC	保留	IDLEC	OREC	NEC	FEC	PEC
			w	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	保留	必须保持复位值。
9	CTSC	CTS变化标志清除。 向该位写1清除USART_STAT寄存器的CTSF位。
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。

19.4.10. 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。

19.4.11. 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。

19.4.12. 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							HCM
							rc_w0								rw

位/位域	名称	描述
31:9	保留	必须保持复位值。
8	EPERR	校验错误超前检测标志。 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。
7:1	保留	必须保持复位值。
0	HCM	硬件流控制兼容性模式。 0：nRTS信号等于RBNE状态寄存器。 1：当最后一个数据位（PCE置位时的奇偶位）被采样时，nRTS信号置位。

19.4.13. 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。

20. 内部集成电路总线接口（I2C）

20.1. 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。

20.2. 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。

20.3. 功能说明

I2C 接口的内部结构如*图 20-1. I2C 模块框图*所示。

图 20-1. I2C 模块框图

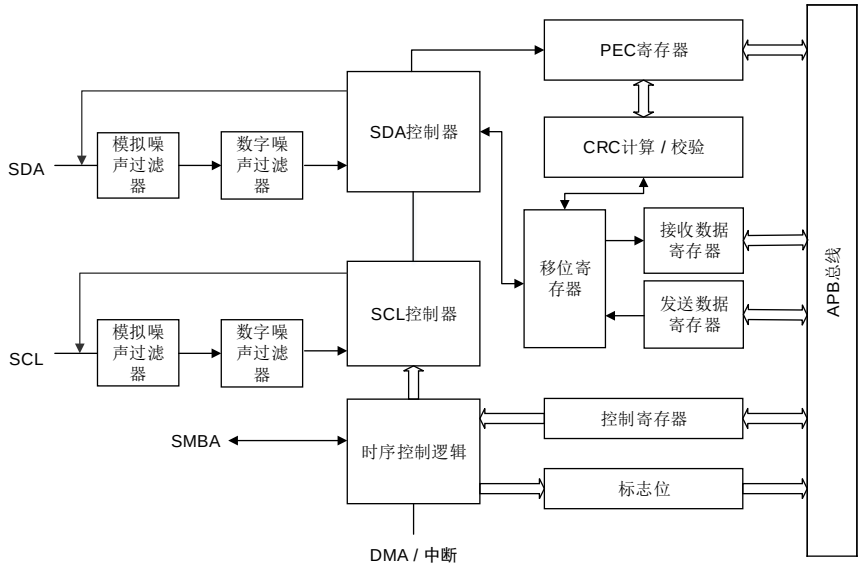


表 20-1. I2C 总线术语说明（参考飞利浦 I2C 规范）

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	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

20.3.1. 时钟要求

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 260ns，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。

20.3.2. I2C 通讯流程

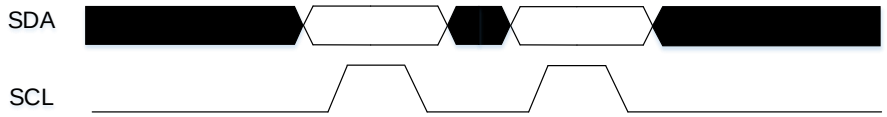
主机和从机都能实现数据收发，因此，I2C 可以实现四种工作模式：

- 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 20-2. 数据有效性*）。每个数据比特传输需要一个时钟脉冲。

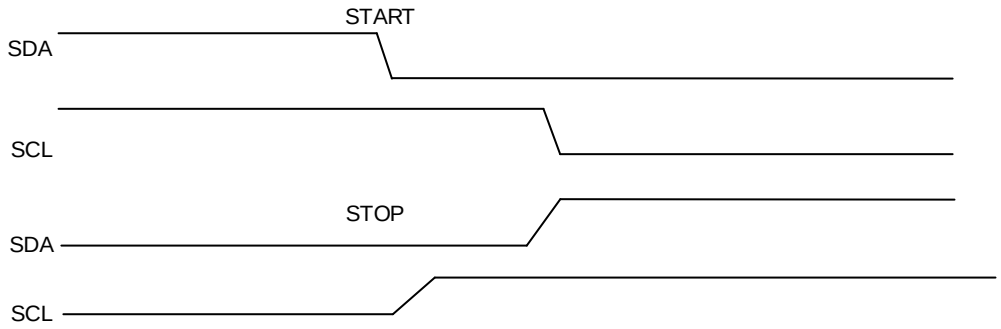
图 20-2. 数据有效性



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 20-3. 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 20-3. 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 20-4. 10 位地址的 I2C 通讯流程（主机发送）

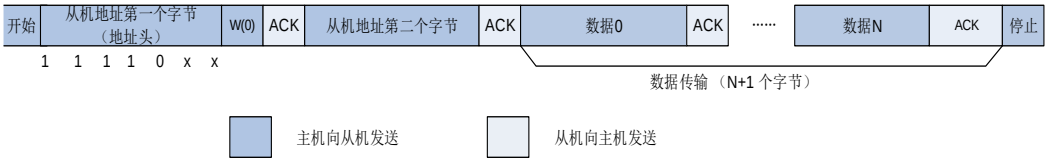


图 20-5. 7 位地址的 I2C 通讯流程（主机发送）

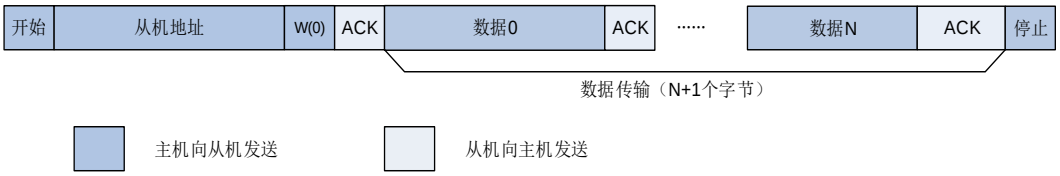
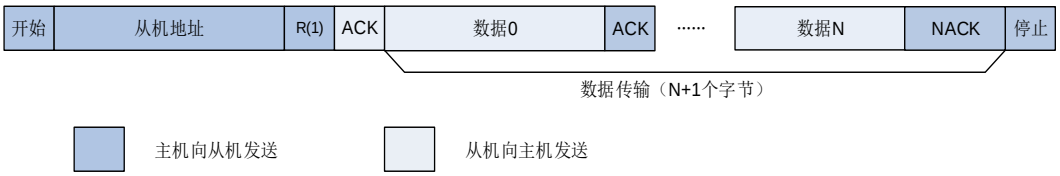


图 20-6. 7 位地址的 I2C 通讯流程（主机接收）



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 20-7. 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 20-8. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R=1）*所示。

图 20-7. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R=0）

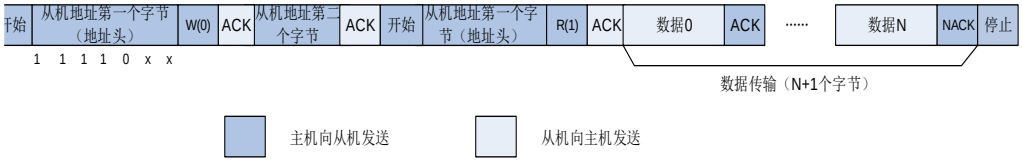
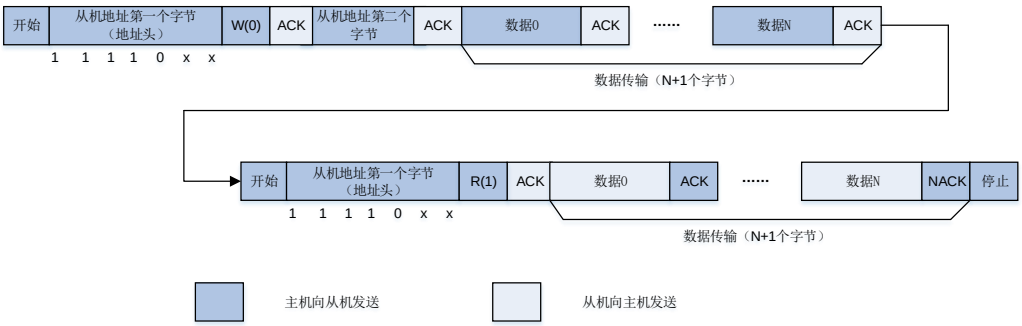


图 20-8. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R=1）



20.3.3. 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。

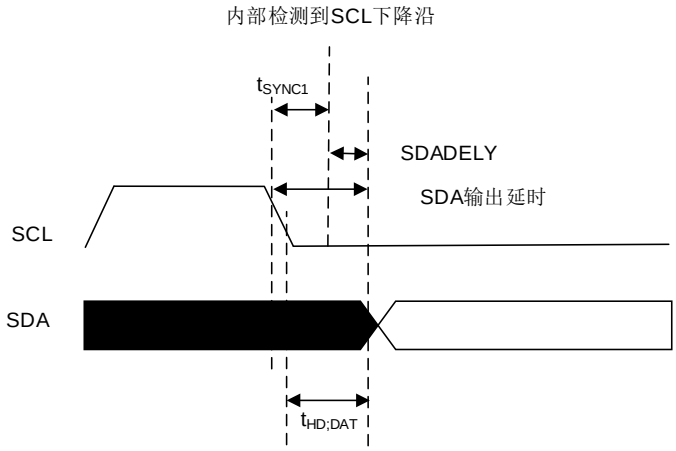
数字滤波器由 I2C_CTL0 寄存器中 DNF[3:0]位来配置。当数字滤波器使能时，SCL 和 SDA 电平保持稳定的时间大于DNF[3:0]×t_{I2CCLK}才会发生内部变化。抑制峰值宽度可由 DNF[3:0]配置。

20.3.4. 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。

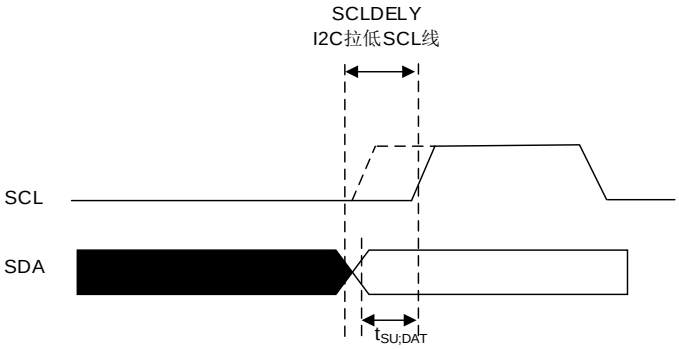
如果数据已经在 I2C_TDATA 寄存器中，在经历 SDADELY 延时后，数据由 SDA 发送，如*图 20-9. 数据保持时间*所示。

图 20-9. 数据保持时间



当数据经过 SDA 发送时，SCLDELY 计数器开启。如图*图 20-10. 数据建立时间*所示。

图 20-10. 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}。

SDADELY 必须符合以下条件：

- SDADELY≥{t_r(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 20-2. 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	

20.3.5. 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：

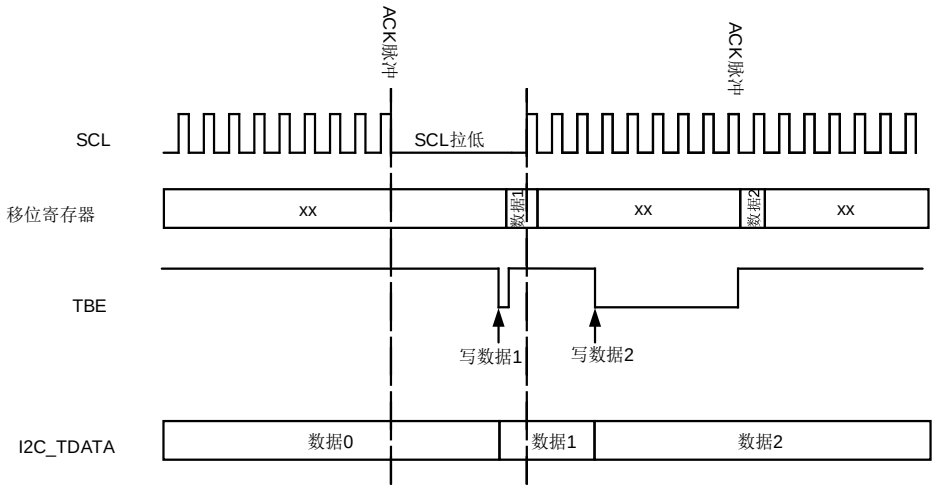
- I2CEN 写 0
- 检查 I2CEN 是否为 0
- I2CEN 写 1

20.3.6. 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。

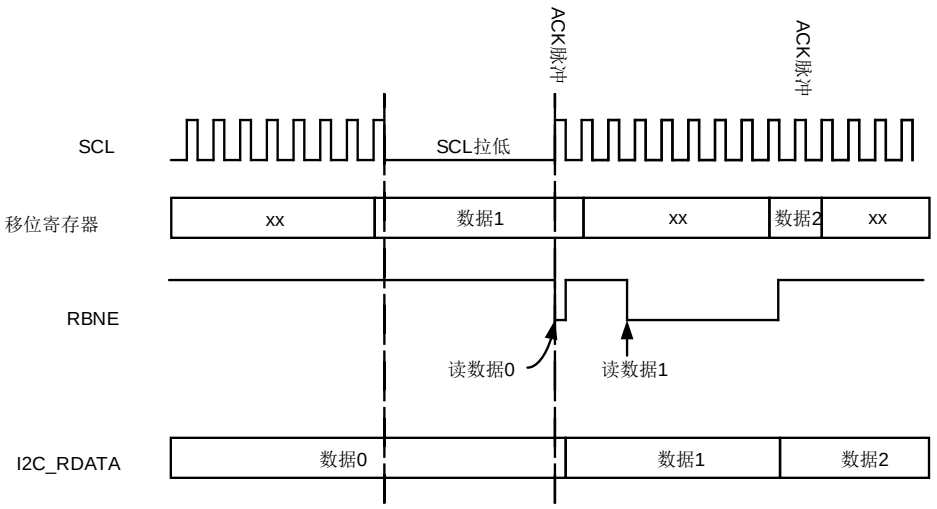
图 20-11. 数据发送



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 20-12. 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 20-3. 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 20-3. 可关闭通信模式

工作模式	行为
主机模式	产生 NACK，STOP 和 RESTART

工作模式	行为
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。

20.3.7. 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]。

I2C_CTL2 寄存器中 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。

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：

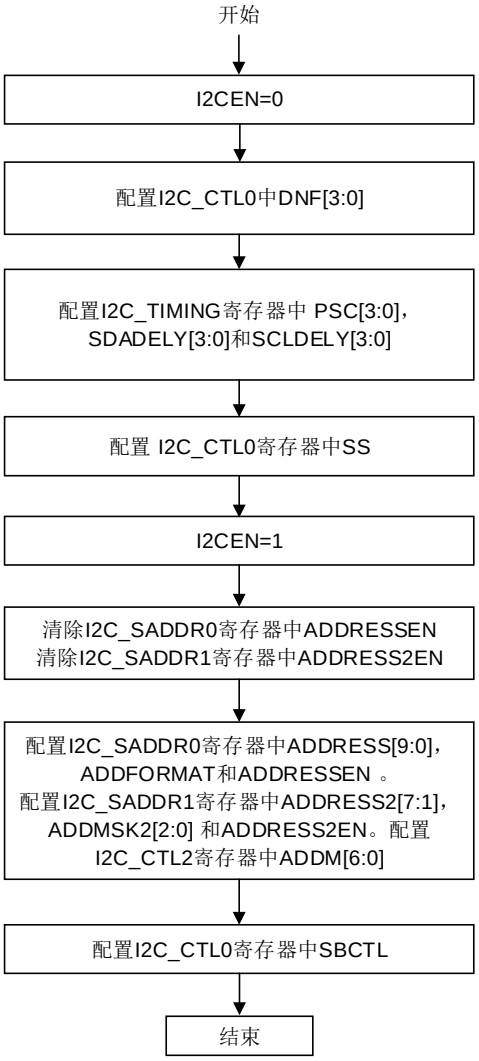
1、 I2CEN=0。

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 20-13. 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 20-14. I2C 从机发送编程模型（SS=0）

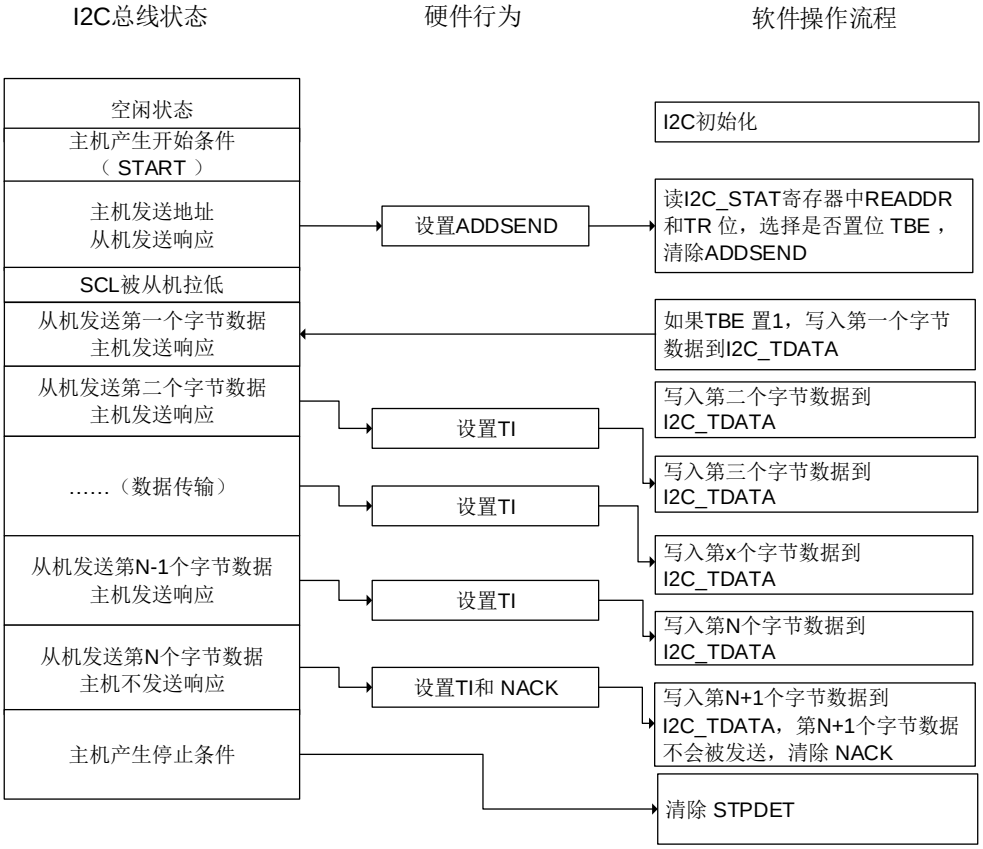
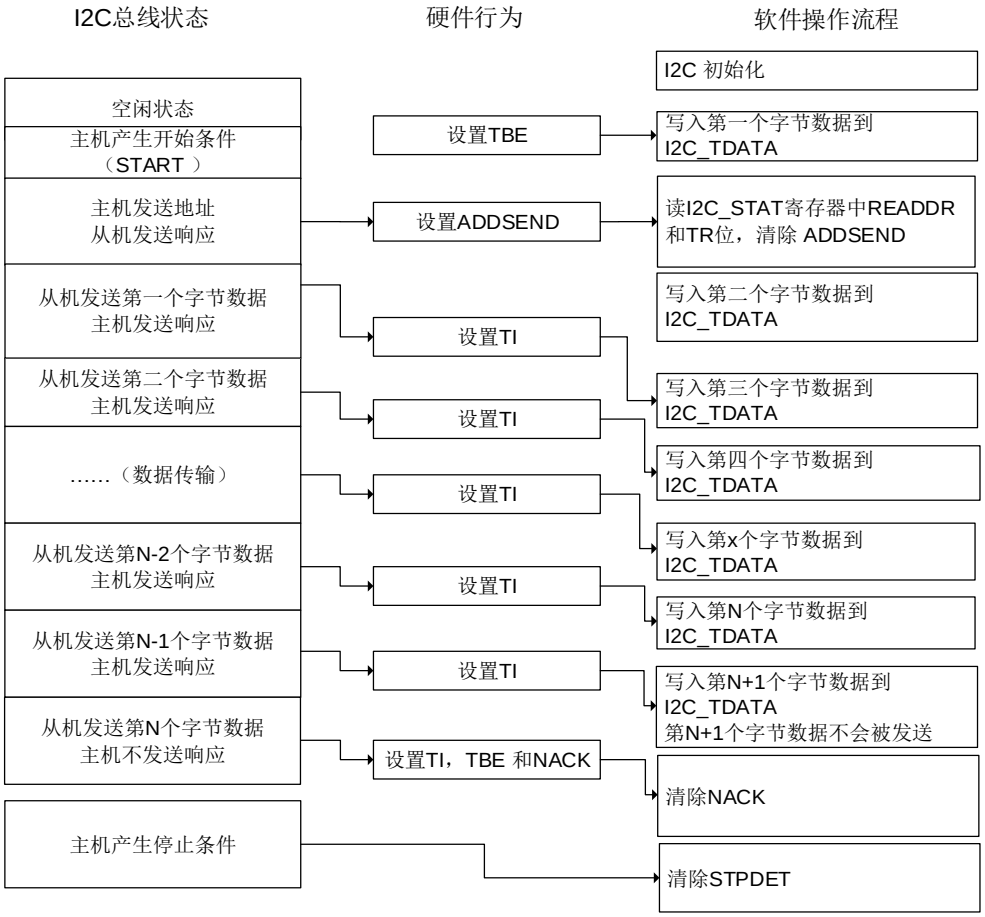


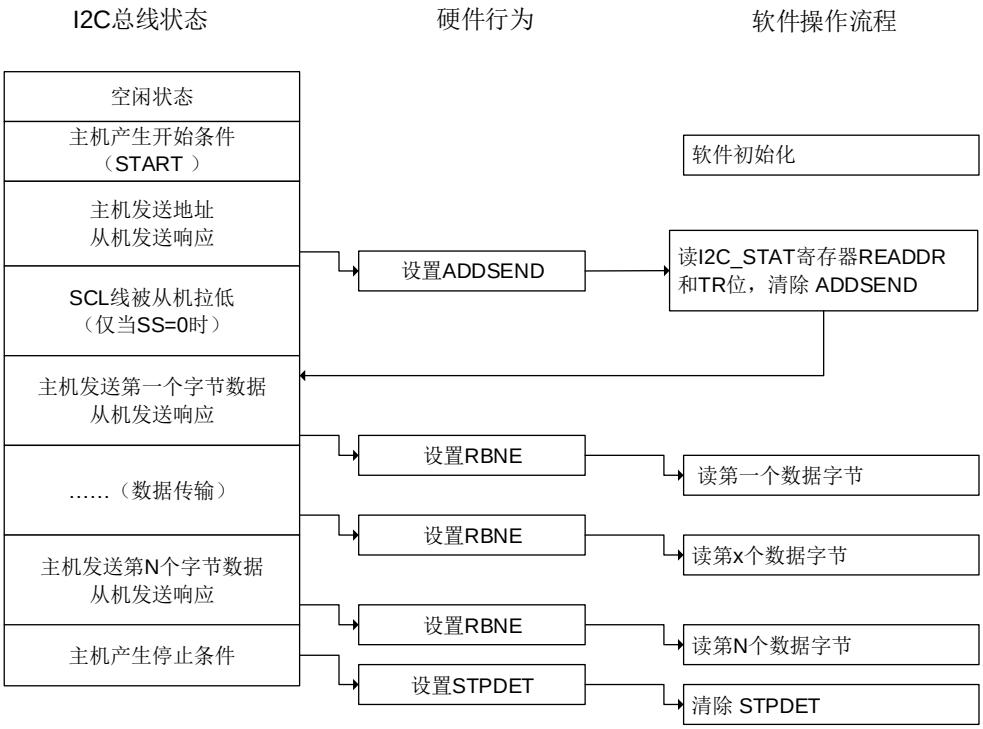
图 20-15. I2C 从机发送编程模型（SS=1）



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 20-16. I2C 从机接收编程模型



20.3.8. 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$DNF[3:0]*t_{I2CCLK}$。

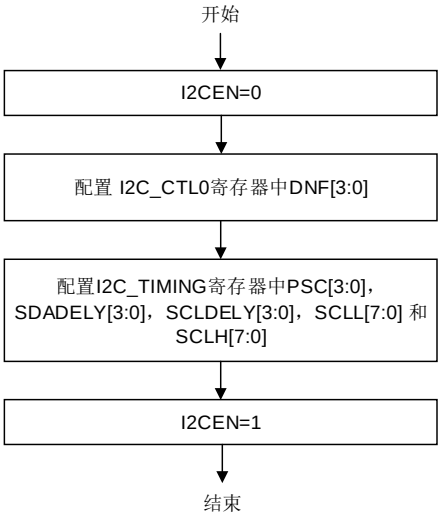
在主机模式下，必须配置 I2C_CTL1 寄存器中 ADD10EN，SADDRESS[9:0]以及 TRDIR 位。当在主机接收模式下使用 10 位寻址时，必须配置 HEAD10R 来选择是执行完整的地址寻址序列，还是只发送地址头。待传输的字节数在 I2C_CTL1 寄存器 BYTENUM[7:0]配置。如果待传输的字节数大于或者等于 255，必须将 BYTENUM[7:0]配置为 0xFF。然后主机发送 START 信号。以上提到的所有位必须在 START 位置 1 之前配置。START 信号发送完成之后，待 I2C_STAT 寄存器 I2CBSY 位为 0 时，发送从机地址。当仲裁丢失时，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。

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 20-17. 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_CTL0 寄存器 BYTENUM[7:0]。如果发送字节数大于 255，必须通过将 I2C_CTL0 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 20-18. I2C 主机发送编程模型（N<=255）

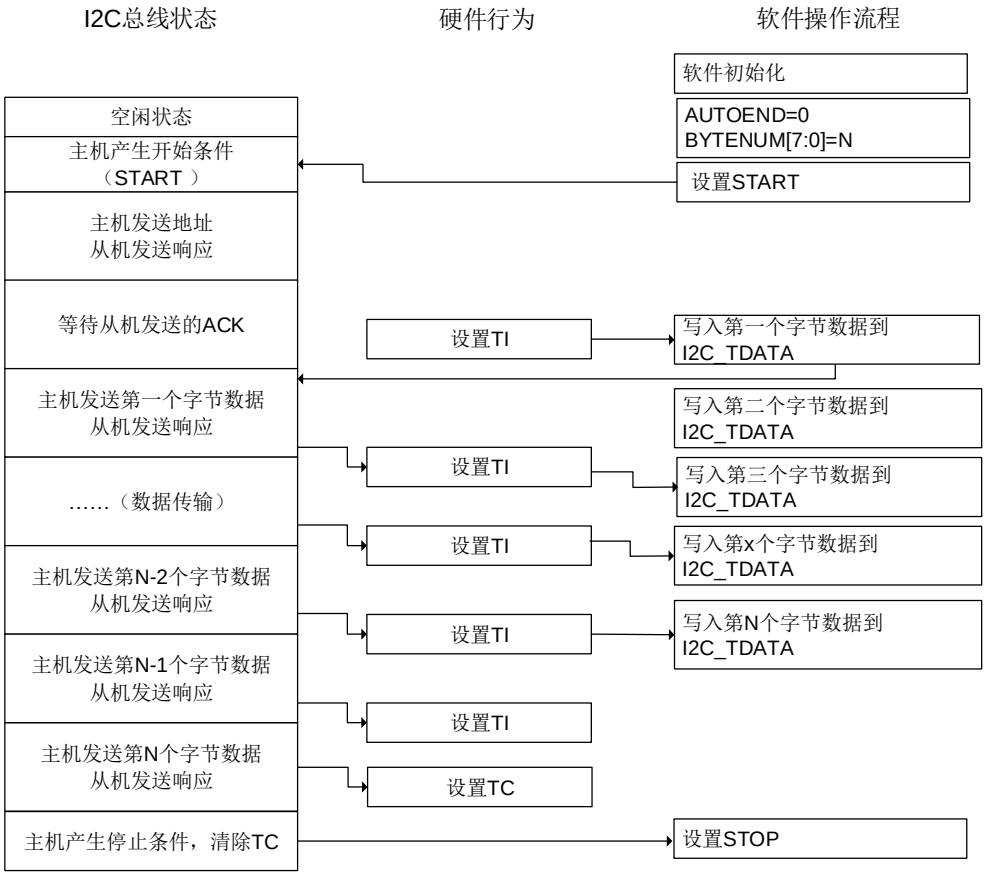
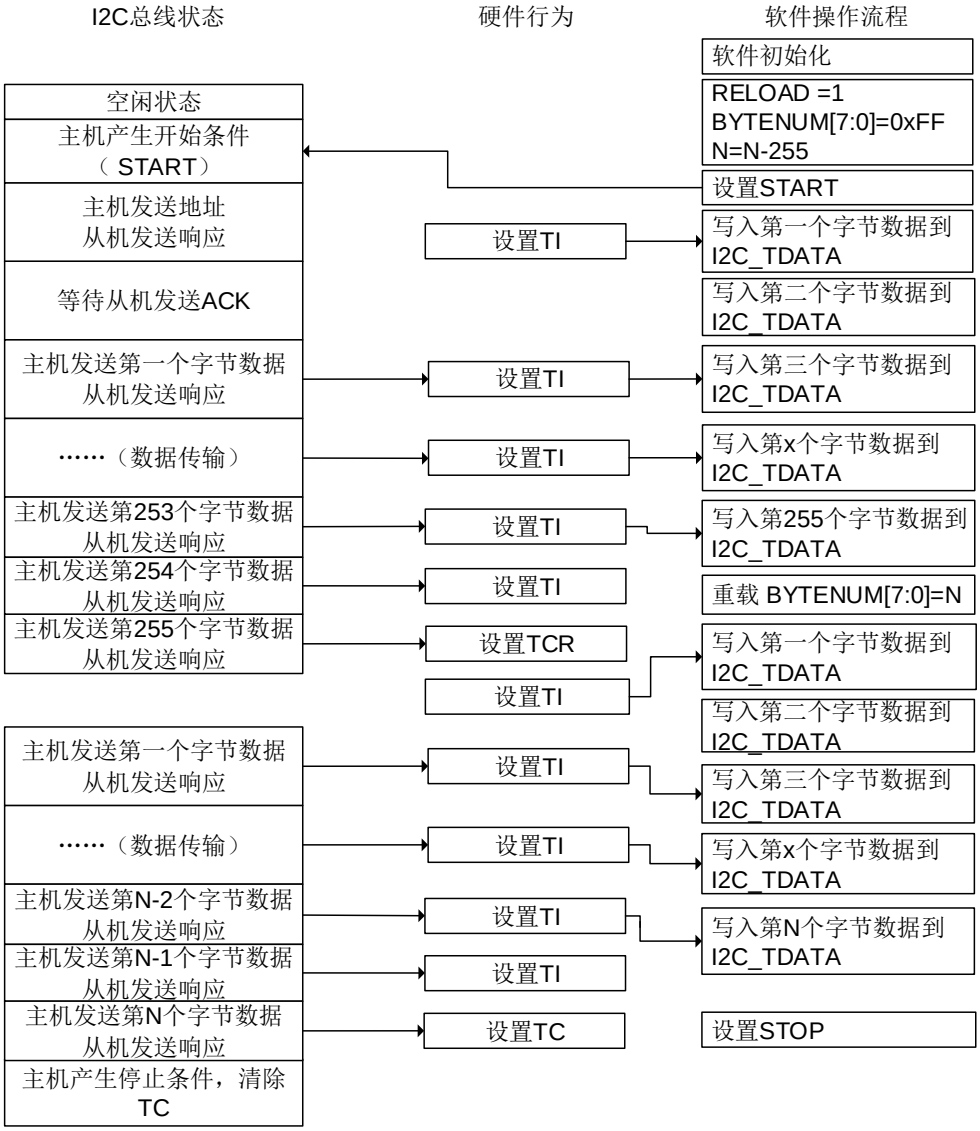


图 20-19. I2C 主机发送编程模型（N>255）



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_CTL0 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 20-20. I2C 主机接收编程模型（N<=255）

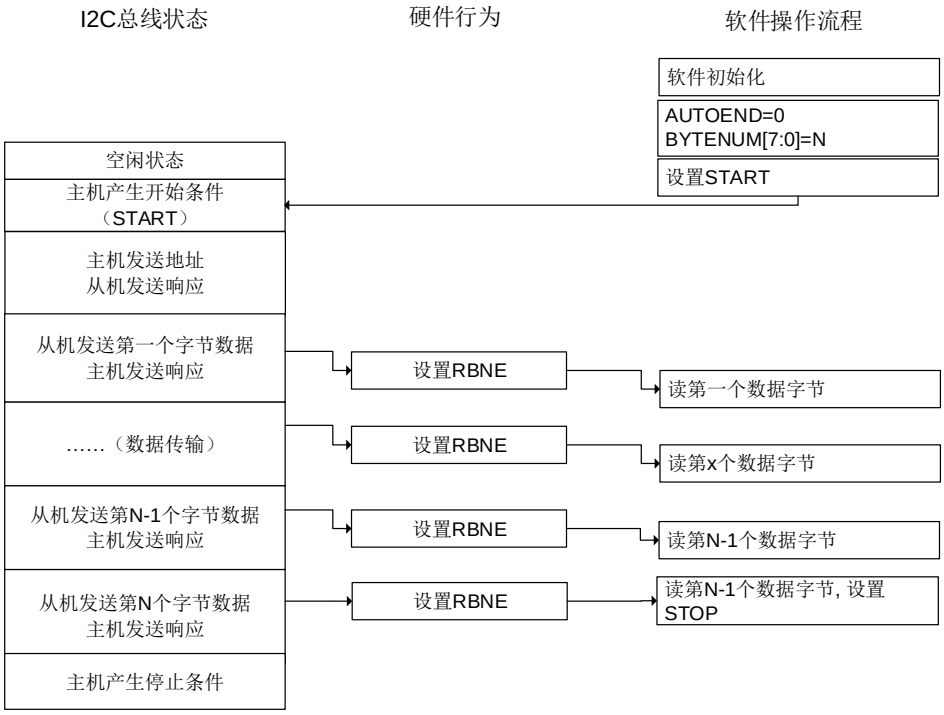


图 20-21. I2C 主机接收编程模型（N>255）

20.3.9. 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）。

20.3.10. SMBus 模式

SMBus 主机发送器和从机接收器

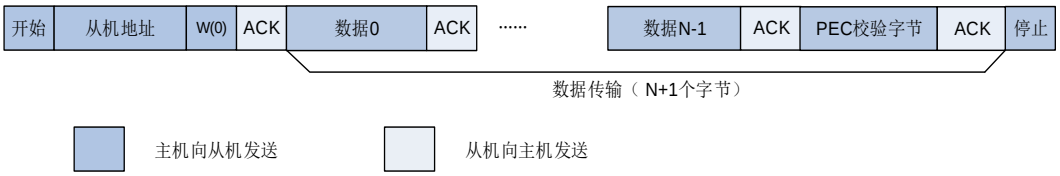
当 SMBus 主机发送 PEC 时，必须在 START 位置 1 前，将 PECTRANS 位置 1 并在

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 20-22. 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

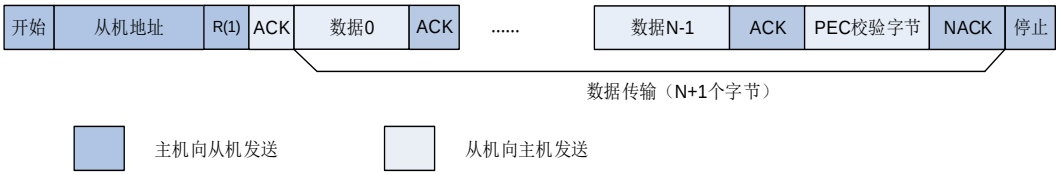
如果 SMBus 主机需要在数据传输完成后接收 PEC 字节，可以使能自动结束模式。在 START 信号发送之前，必须将 PECTRANS 位置 1，且配置好从机地址。在接收 BYTENUM-1 数据之后，接收的下一个字节将自动与 I2C_PEC 寄存器中的数据进行比较。在停止信号发送之前，接收 PEC 字节之后会给出 NACK 响应。

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：PECTRANS 位在 RELOAD 置 1 之后不能被修改。

图 20-23. SMBus 主机接收器和从机发送器通信流程



20.3.11. 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。

20.3.12. I2C 错误和中断

I2C 错误标志如*表 20-4. I2C 错误标志*所示。

表 20-4. 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 20-5. I2C 中断事件*所示。

表 20-5. 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
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	

中断事件	事件标志	使能控制位
PEC 错误	PECERR	
超时错误	TIMEOUT	
SMBus 报警	SMBALT	

20.3.13. 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。

20.4. I2C 寄存器

I2C0 基地址：0x4000 5400

I2C1 基地址：0x4000 5800

20.4.1. 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	保留	SS	SBCTL
								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	保留	必须保持复位值。
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
13:12	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。

20.4.2. 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，这些位不能被修改。
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。
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。
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。
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。

20.4.3. 从机地址寄存器 0（I2C_SADDR0）

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字（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
ADDRESSEN	保留				ADDFORMAT	ADDRESS[1:0]		ADDRESS[6:0]							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[1:0]	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[6:0]	7 位地址或者 10 位地址的第 7-1 位 **注意：**当 ADDRESSEN =1 时，该位不能被改写。
0	ADDRESS0	10 位地址的第 0 位 **注意：**当 ADDRESSEN =1 时，该位不能被改写。

20.4.4. 从机地址寄存器 1（I2C_SADDR1）

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字（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
ADDRESS2EN	保留				ADDMSK2[2:0]			ADDRESS2[6:0]							保留
rw					rw			rw							

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:0]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[6:0]	I2C 从机的第二个地址 **注意：**当 ADDRESS2EN =1 时，该位不能被改写。
0	保留	必须保持复位值。

20.4.5. 时序寄存器（I2C_TIMING）

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
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。 $t_{SDADELY}$=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。

20.4.6. 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。

20.4.7. 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。

20.4.8. 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。

20.4.9. 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。

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								PECV[7:0]							
											r				

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。

20.4.10. 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。

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[7:0]							
											r				

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

20.4.11. 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。

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 [7:0]							
											rw				

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

20.4.12. 控制寄存器 2（I2C_CTL2）

地址偏移：0x90

复位值：0x0000 FE00

该寄存器只能按字（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
ADDM[6:0]							保留								
			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 的位则忽略（此时接收到的地址在该位可以为 0 或 1）。
8:0	保留	必须保持复位值。

21. 串行外设接口/片上音频接口（SPI/I2S）

21.1. 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。

21.2. 主要特性

21.2.1. SPI 主要特性

- 具有全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）。

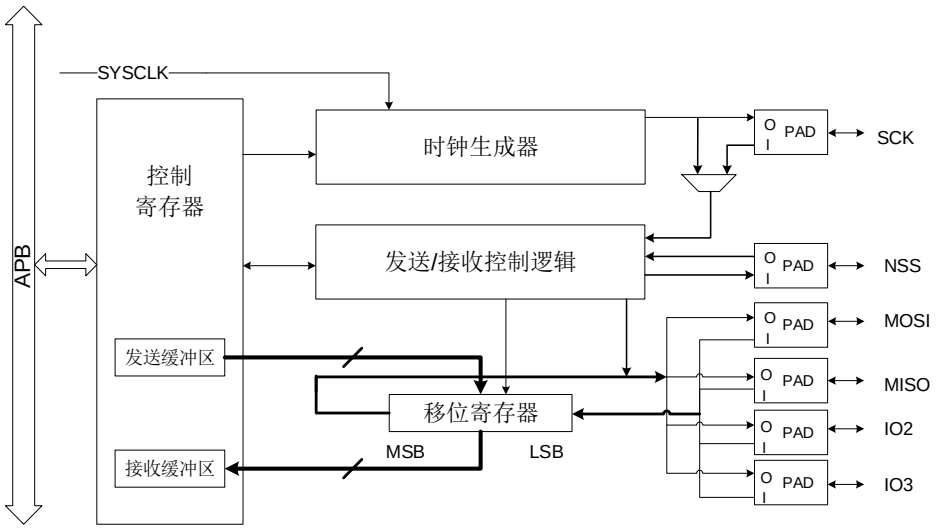
21.2.2. I2S 主要特性

- 具有发送和接收功能的主从操作；
- 支持四种I2S音频标准：飞利浦标准，MSB对齐标准，LSB对齐标准和PCM标准；
- 数据长度可以为16位，24位和32位；
- 通道长度为16位或32位；
- 16位缓冲区用于发送和接收；
- 通过I2S时钟分频器，可以得到8 kHz到192 kHz的音频采样频率；
- 可编程空闲状态时钟极性；
- 可以输出主时钟（MCK）；
- 发送和接收支持DMA功能。

21.3. SPI 功能说明

21.3.1. SPI 结构框图

图 21-1. SPI 结构框图



21.3.2. SPI 信号线描述

常规配置（非 SPI 四线模式）

表 21-1. 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默认配置为非四线模式，当SPI_QCTL中的QMOD位置1时，配置为SPI四线模式（只适用于SPI0）。SPI四线模式只能工作在主机模式。

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 21-2. 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

21.3.3. 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 21-2. 常规模式下的 SPI 时序图

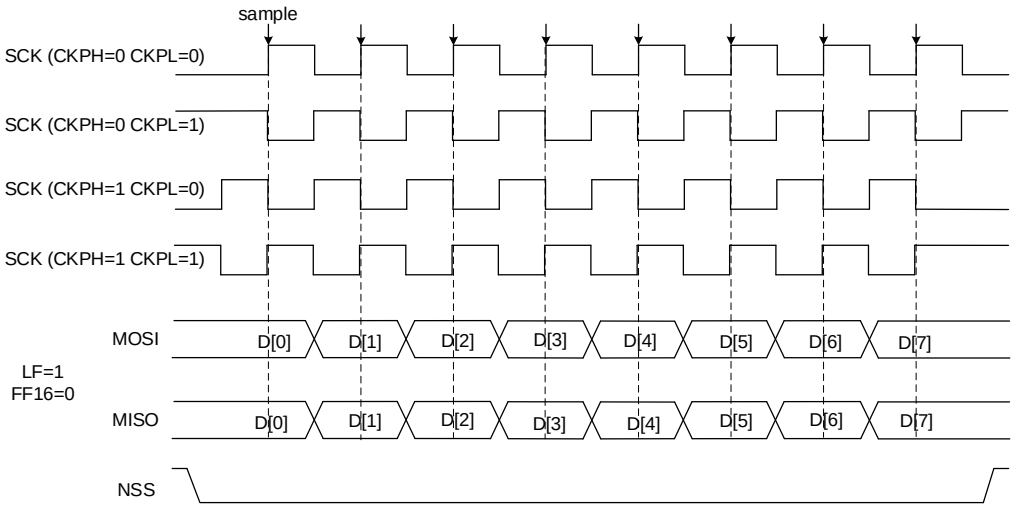
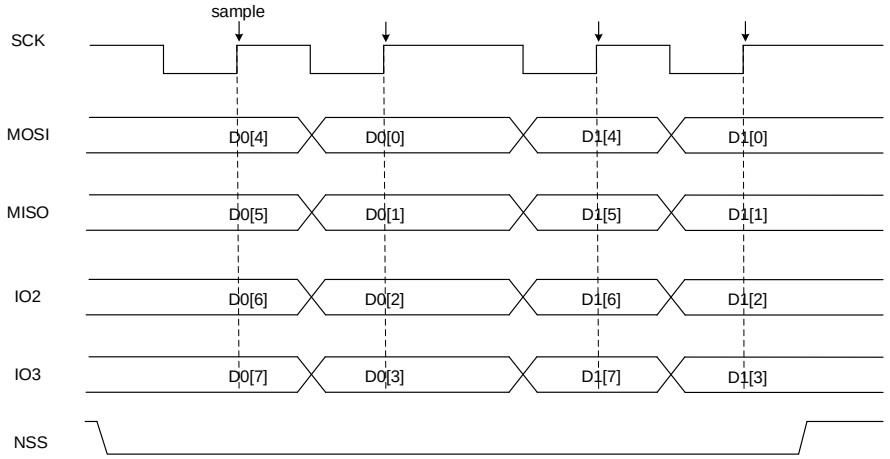


图 21-3. SPI 四线模式下的 SPI 时序图(CKPL=1, CKPH=1, LF=0)



在常规模式中，通过SPI_CTL0中的FF16位配置数据长度，当FF16=1时，数据长度为16位，否则为8位。在SPI四线模式下，数据帧长度固定为8位。

通过设置SPI_CTL0中的LF位可以配置数据顺序，当LF=1时，SPI先发送LSB位，当LF=0时，则先发送MSB位。在TI模式中，数据顺序固定为先发MSB位。

21.3.4. 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 21-3. 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 21-4. 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

21.3.5. SPI 运行模式

表 21-5. 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 RO = 0	MOSI：接收 MISO：不使用

模式	描述	寄存器配置	数据引脚用法
		BDEN = 1 BDOEN = 0	
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 21-4. 典型的全双工模式连接

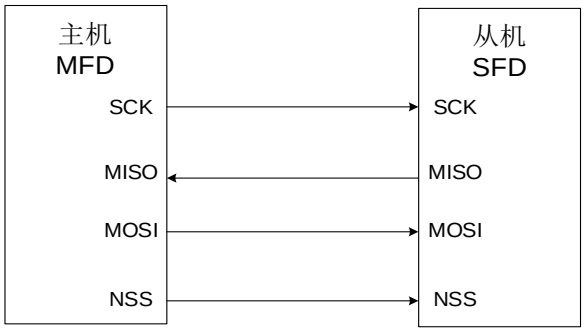


图 21-5. 典型的单工模式连接（主机：接收，从机：发送）

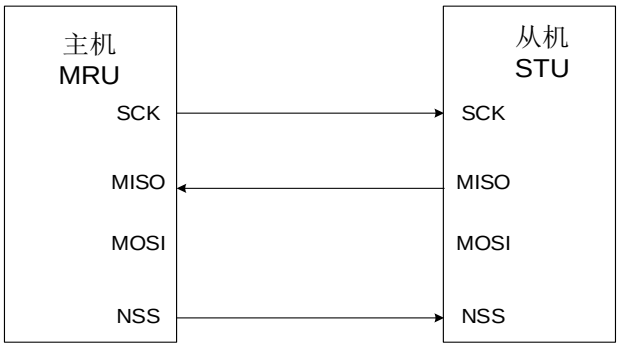


图 21-6. 典型的单工模式连接（主机：只发送，从机：接收）

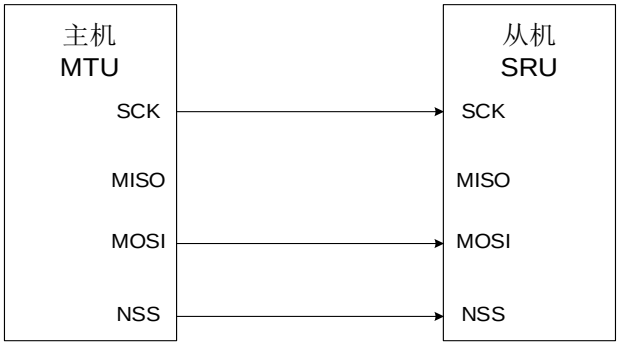
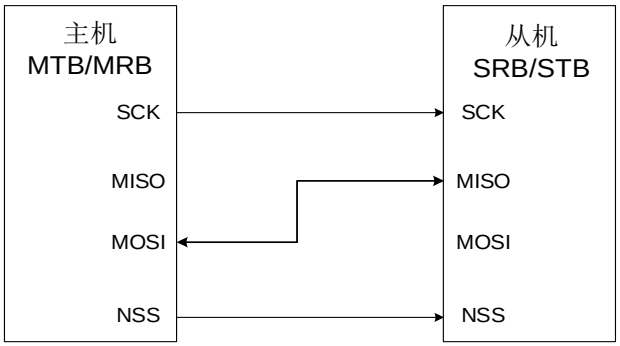


图 21-7. 典型的双向线连接



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SPI_CTL1中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. 根据***表21-5. SPI运行模式***，配置MSTMOD位、RO位、BDEN位和BDOEN位。
8. 如果工作在SPI四线模式，需要将SPI_QCTL中的QMOD位置1，如果不是，则忽略此步骤。
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 21-8. 主机 TI 模式在不连续发送时的时序图

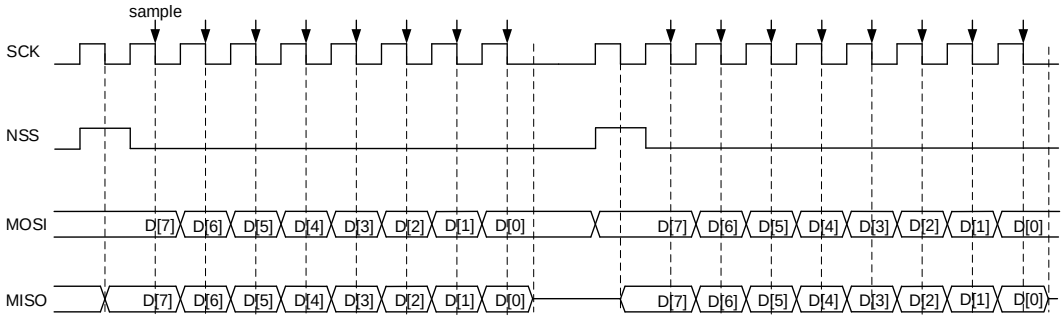
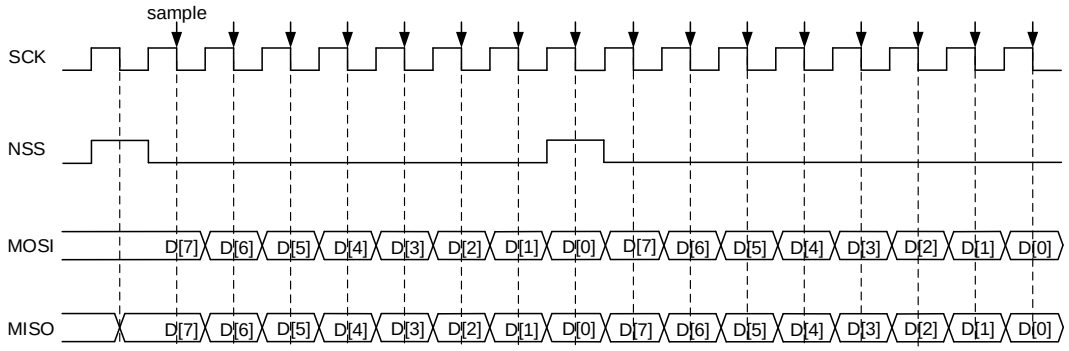
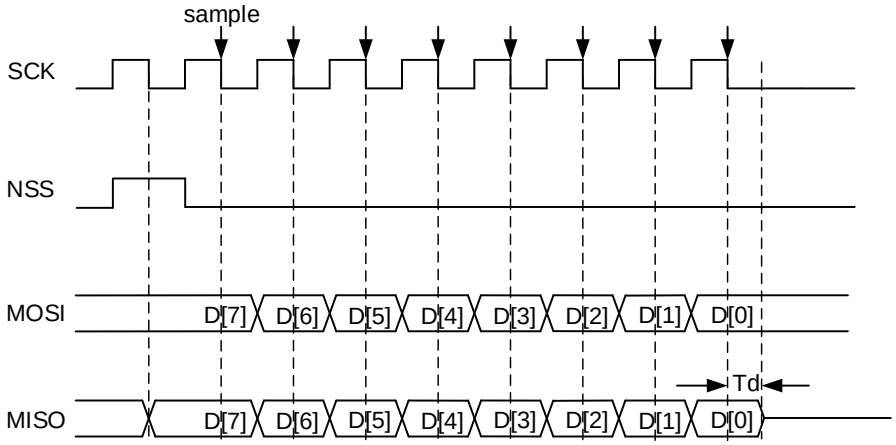


图 21-9. 主机 TI 模式在连续发送时的时序图



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 21-10. 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。

$$T_d=\frac{T_{bit}}{2}+5*T_{pclk} \tag{21-1}$$

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

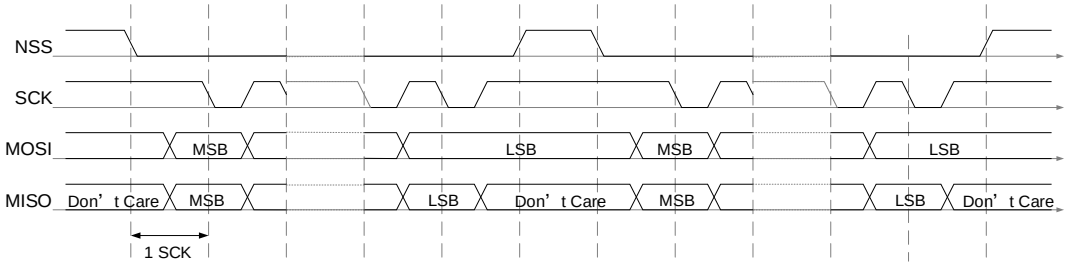
配置SPI_CTL1寄存器中的NSSP位使能该功能，为了确保使用该功能实现，需满足以下几个条件：配置设备为主机模式，使用普通SPI协议的数据帧格式，同时在第一个时钟跳变沿采样数据。

总之：NSSP = 1， MSTMOD = 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 21-11. NSS 脉冲模式时序图（主机连续发送）



SPI 四线模式操作流程

SPI四线模式用于控制四线SPI flash外设。

要配置成SPI四线模式，首先要确认TBE位置1，且TRANS位清零，然后将SPI_QCTL寄存器中的QMOD位置1。在SPI四线模式，SPI_CTL0寄存器中BDEN位、BDOEN位、CRCEN位、CRCNT位、FF16位、RO位和LF位保持清零，且MSTMOD位置1，以保证SPI工作于主机模式。SPIEN位、PSC位、CKPL位和CKPH位根据需要进行配置。

SPI四线模式有两种运行模式：四线写模式和四线读模式，通过SPI_QCTL寄存器中的QRD位进行配置。

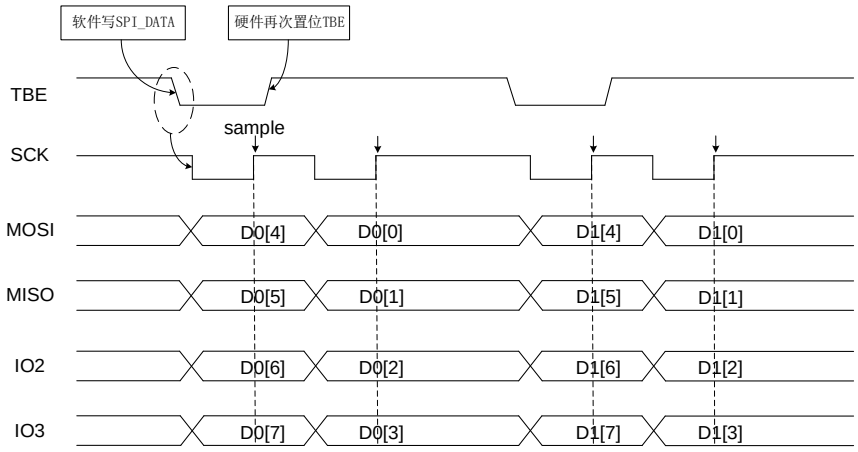
四线写模式

当SPI_QCTL寄存器中的QMOD位置1且QRD位清零时，SPI工作在四路写模式。在四路写模式中，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。

四路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 21-12. SPI 四线模式四线写操作时序图



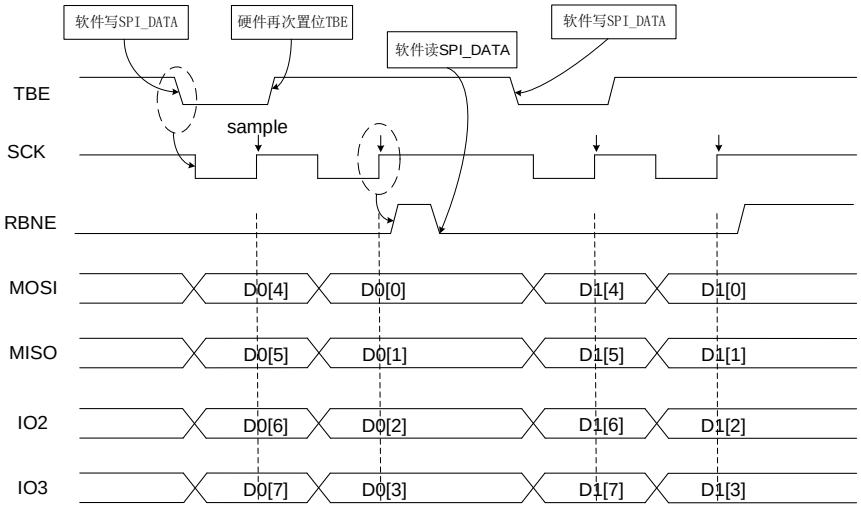
四线读模式

当SPI_QCTL寄存器中的QMOD位和QRD位都置1时，SPI工作在四路读模式。在四路读模式中，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。

四路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 21-13. SPI 四路模式四路读操作时序图



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四路模式

在禁用SPI四路模式和关闭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。

21.3.6. 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，SPI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。

21.3.7. 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。

21.3.8. SPI 中断

状态标志位

- **发送缓冲区空标志位（TBE）**

当发送缓冲区为空时，TBE置位。软件可以通过写SPI_DATA寄存器将下一个待发送数据写入发送缓冲区。

- **接收缓冲区非空标志位（RBNE）**

当接收缓冲区非空时，RBNE置位，表示此时接收到一个数据，并已存入到接收缓冲区中，软件可以通过读SPI_DATA寄存器来读取此数据。

- **SPI通信进行中标志位（TRANS）**

TRANS位是用来指示当前传输是否正在进行或结束的状态标志位，它由内部硬件置位和清除，无法通过软件控制。该标志位不会产生任何中断。

错误标志

- **配置错误标志（CONFERR）**

在主机模式中，CONFERR位是一个错误标志位。在硬件NSS模式中，如果NSSDRV没有使能，当NSS被拉低时，CONFERR位被置1。在软件NSS模式中，当SWNSS位为0时，CONFERR位置1。当CONFERR位置1时，SPIEN位和MSTMOD位由硬件清除，SPI关闭，设备强制进入从机模式。

在CONFERR位清零之前，SPIEN位和MSTMOD位保持写保护，从机的CONFERR位不能置1。在多主机配置中，设备可以在CONFERR位置1时进入从机模式，这意味着发生了系统控制的多主冲突。

- **接收过载错误（RXORERR）**

在RBNE位为1时，如果再有数据被接收，RXORERR位将会置1。这说明，上一帧数据还未被读出而新的数据已经接收了。接收缓冲区的内容不会被新接收的数据覆盖，所以新接收的数据丢失。

■ **帧错误（FERR）**

在TI从机模式下，从机也要监视NSS信号，如果检测到错误的NSS信号，将会置位FERR标志位。例如，NSS信号在一个字节的中间位发生翻转。

■ **CRC错误（CRCERR）**

当CRCEN位置1时，SPI_RCRC寄存器中接收到的数据的CRC计算值将会和紧随着最后一帧数据接收到的CRC值进行比较，当两者不同时，CRCERR位将会置1。

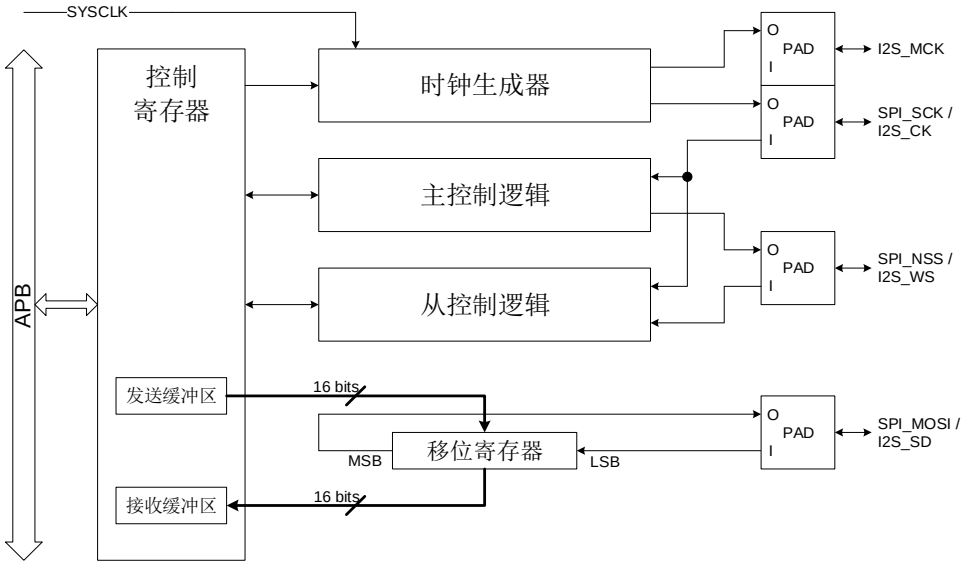
表 21-6. SPI 中断请求

中断事件	描述	清除方式	中断使能位
TBE	发送缓冲区空	写SPI_DATA寄存器	TBEIE
RBNE	接收缓冲区非空	读SPI_DATA寄存器	RBNEIE
CONFERR	配置错误	读或写 SPI_STAT 寄存器，然后写 SPI_CTL0 寄存器	ERRIE
RXORERR	接收过载错误	读SPI_DATA寄存器，然后读 SPI_STAT寄存器	
CRCERR	CRC错误	写0到CRCERR位	
FERR	TI模式帧错误	写0到FERR位	

21.4. I2S 功能描述

21.4.1. I2S 结构框图

图 21-14. 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。

21.4.2. I2S 信号线描述

I2S接口有4个引脚，分别是I2S_CK、I2S_WS、I2S_SD和I2S_MCK。I2S_CK是串行时钟信号，与SPI_SCK共享引脚。I2S_WS是数据帧控制信号，与SPI_NSS共享引脚。I2S_SD是串行数据信号，与SPI_MOSI共享引脚。I2S_MCK是主时钟信号，它提供了一个256倍于Fs的时钟频率，其中Fs是音频采样率。

21.4.3. 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 21-15. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=0, CKPL=0）

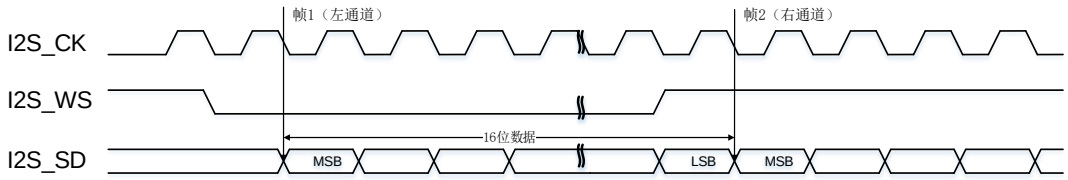
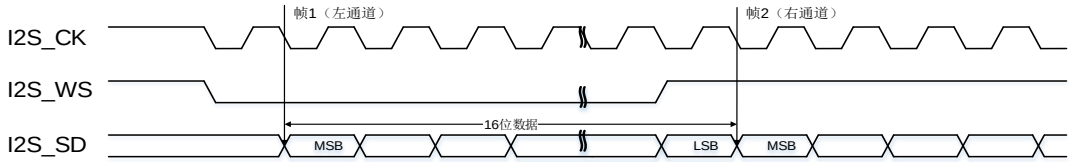


图 21-16. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=0, CKPL=1）



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 21-17. I2S 飞利浦标准时序图（DTLEN=10, CHLEN=1, CKPL=0）

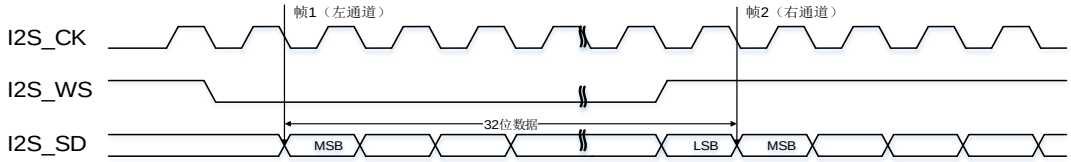
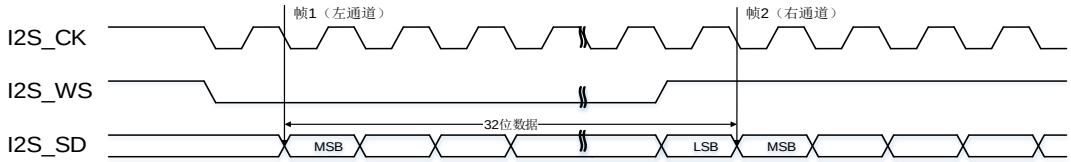


图 21-18. I2S 飞利浦标准时序图（DTLEN=10, CHLEN=1, CKPL=1）



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 21-19. I2S 飞利浦标准时序图（DTLEN=01, CHLEN=1, CKPL=0）

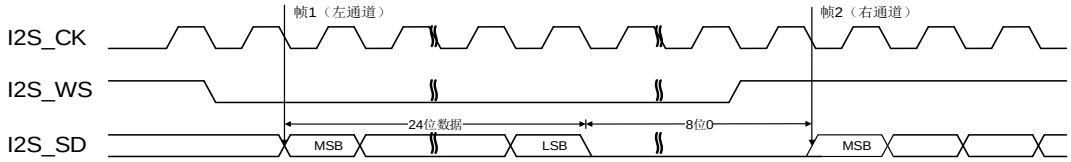
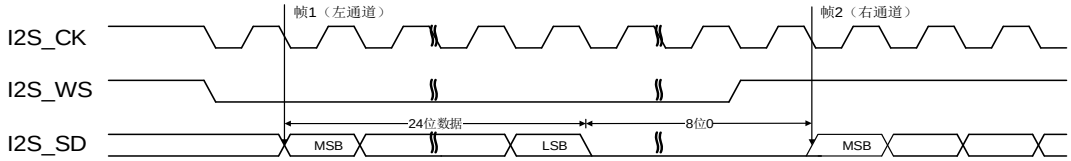


图 21-20. I2S 飞利浦标准时序图（DTLEN=01, CHLEN=1, CKPL=1）



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 21-21. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=1, CKPL=0）

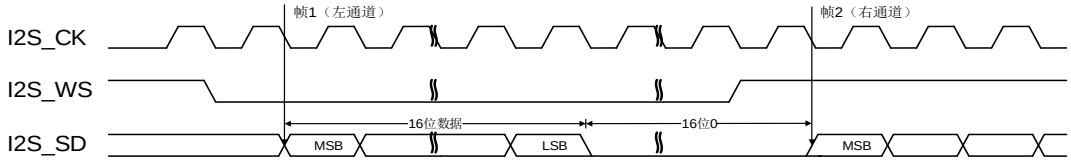
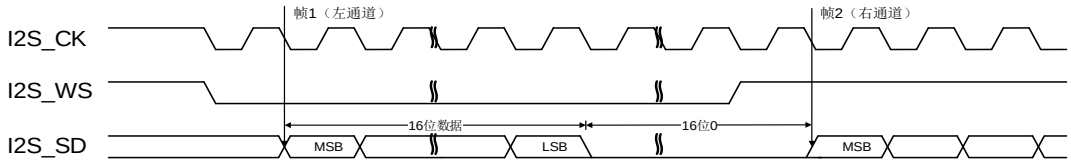


图 21-22. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=1, CKPL=1）



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 21-23. MSB 对齐标准时序图（DTLEN=00, CHLEN=0, CKPL=0）

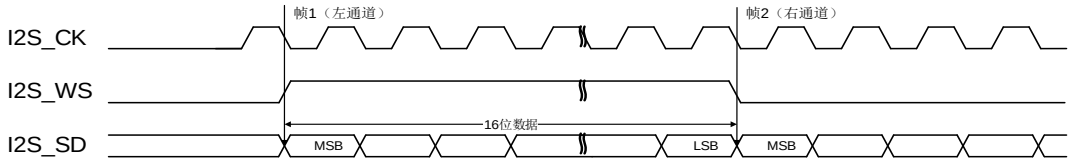


图 21-24. MSB 对齐标准时序图（DTLEN=00, CHLEN=0, CKPL=1）

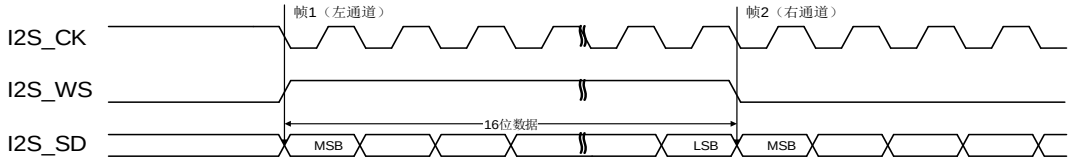


图 21-25. MSB 对齐标准时序图（DTLEN=10, CHLEN=1, CKPL=0）

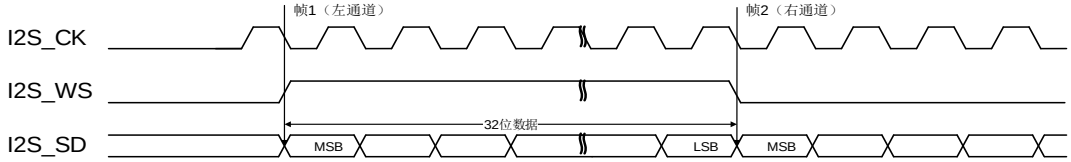


图 21-26. MSB 对齐标准时序图（DTLEN=10, CHLEN=1, CKPL=1）

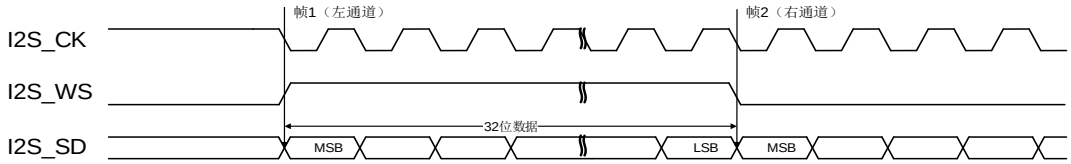


图 21-27. MSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=0）

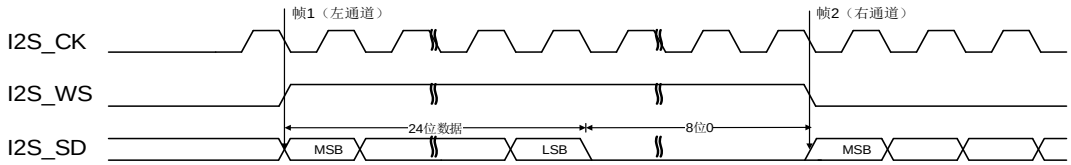


图 21-28. MSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=1）

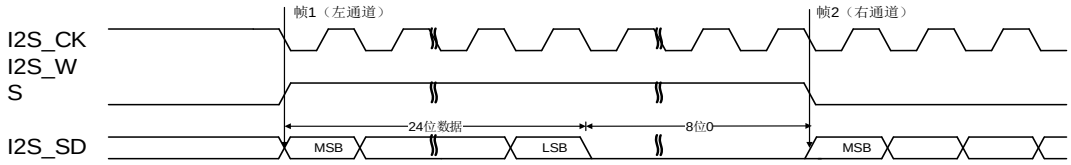


图 21-29. MSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=0）

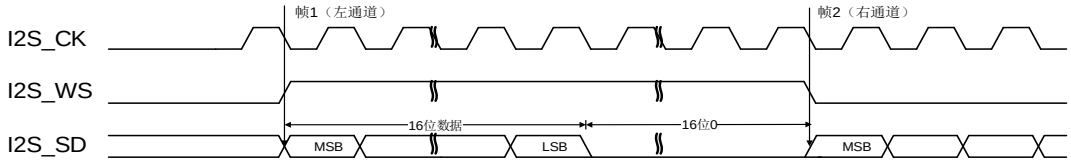
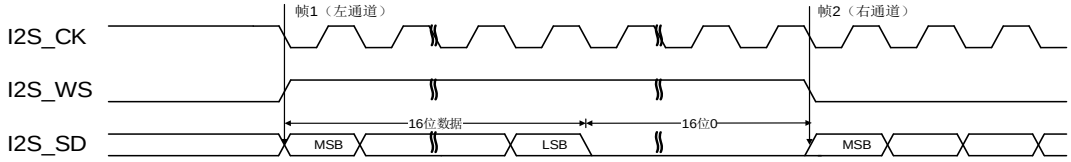


图 21-30. MSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=1）



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 21-31. LSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=0）

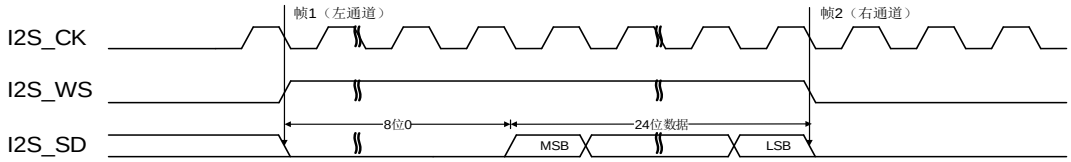
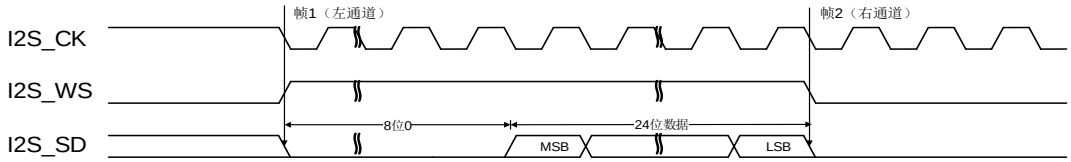


图 21-32. LSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=1）



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 21-33. LSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=0）

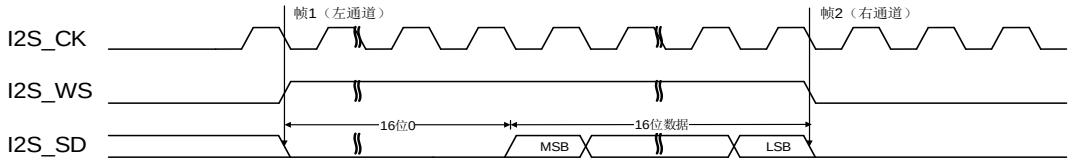
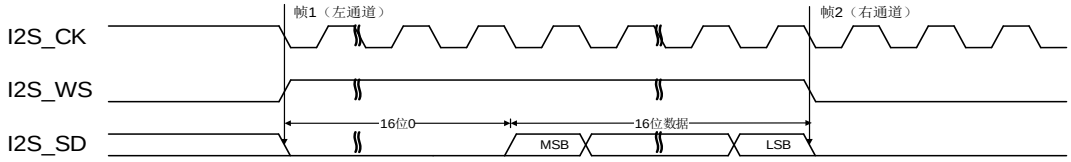


图 21-34. LSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=1）



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 21-35. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=0）

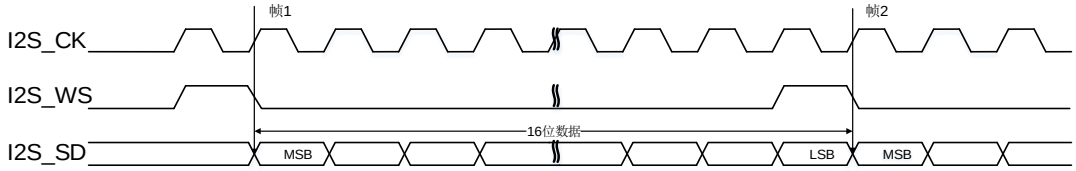


图 21-36. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=1）

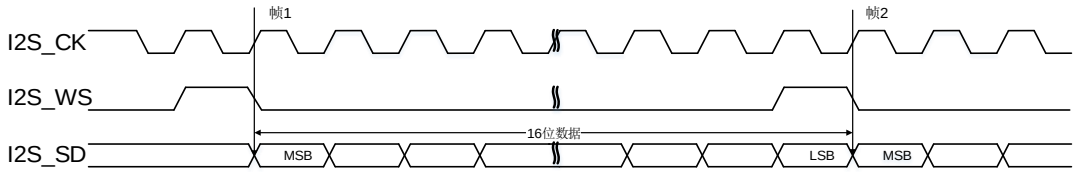


图 21-37. PCM 标准短帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=0）

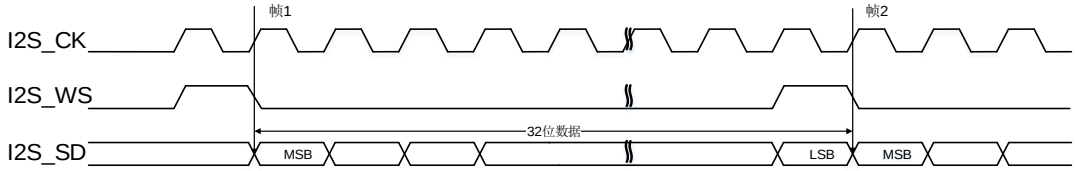


图 21-38. PCM 标准短帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=1）

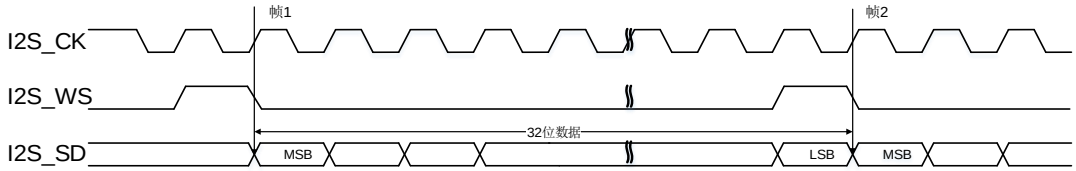


图 21-39. PCM 标准短帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=0）

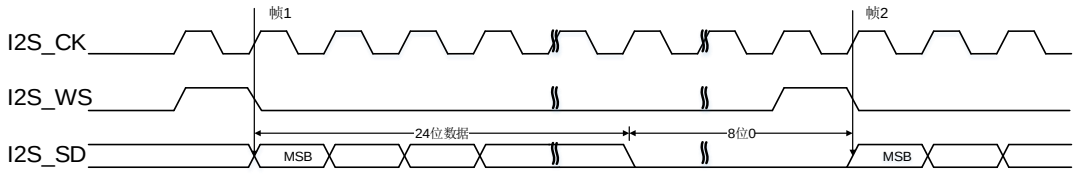


图 21-40. PCM 标准短帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=1）

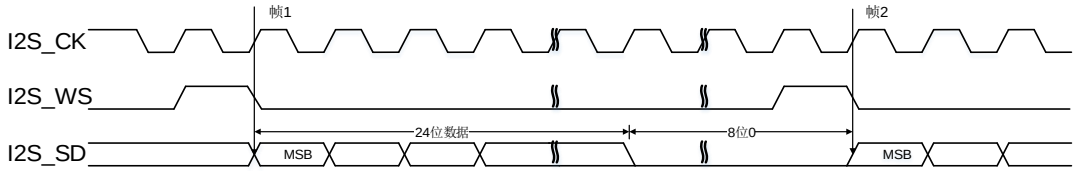


图 21-41. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=0）

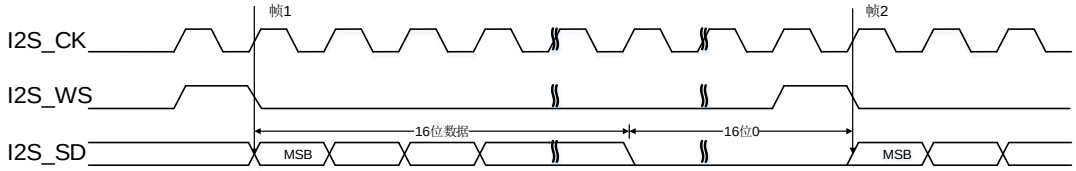
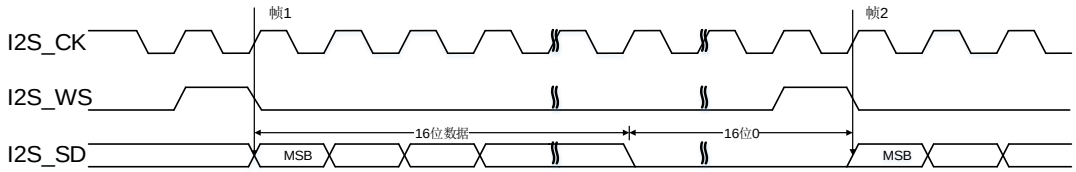


图 21-42. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=1）



长帧同步模式的各种配置情况时序图如下所示。

图 21-43. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=0）

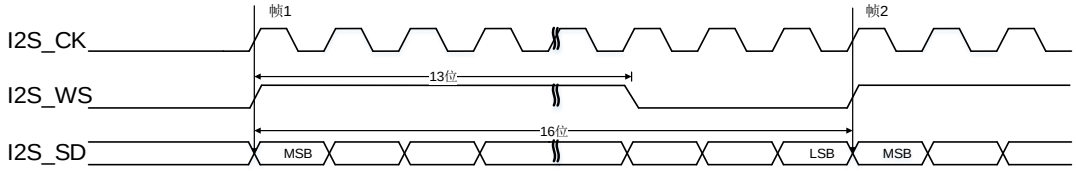


图 21-44. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=1）

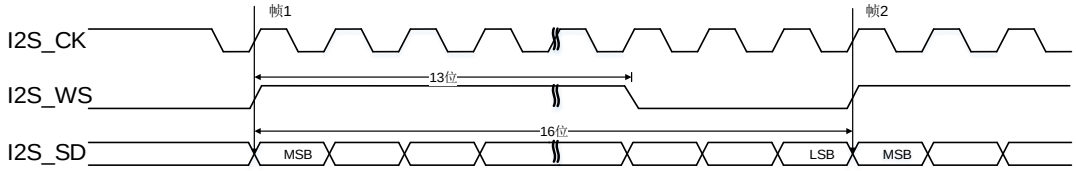


图 21-45. PCM 标准长帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=0）

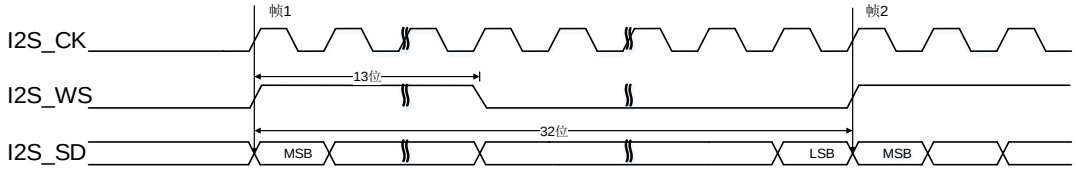


图 21-46. PCM 标准长帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=1）

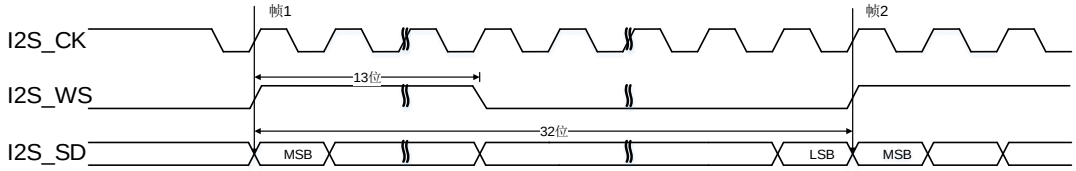


图 21-47. PCM 标准长帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=0）

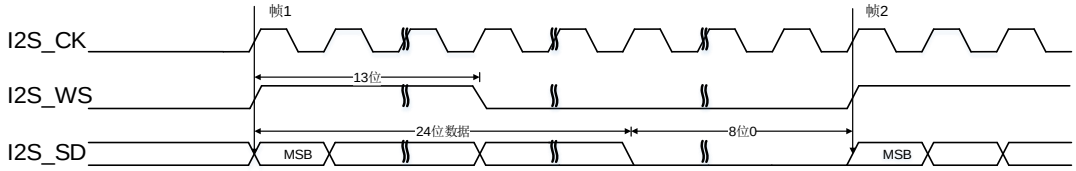


图 21-48. PCM 标准长帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=1）

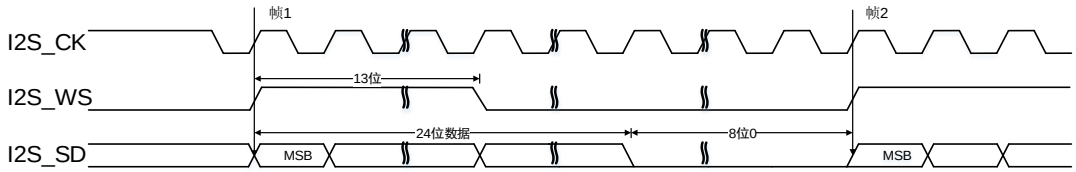


图 21-49. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=0）

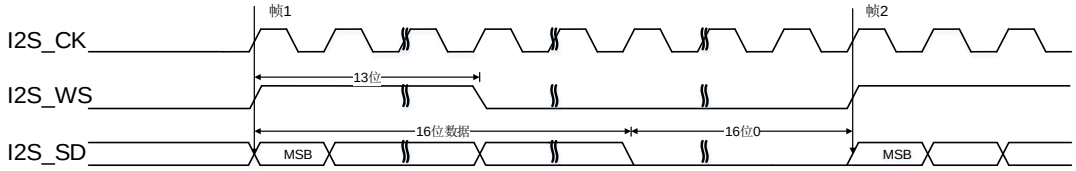
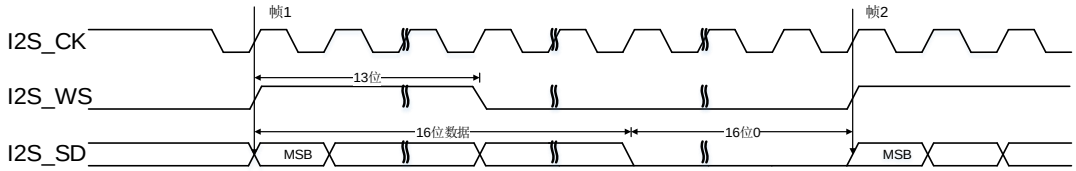
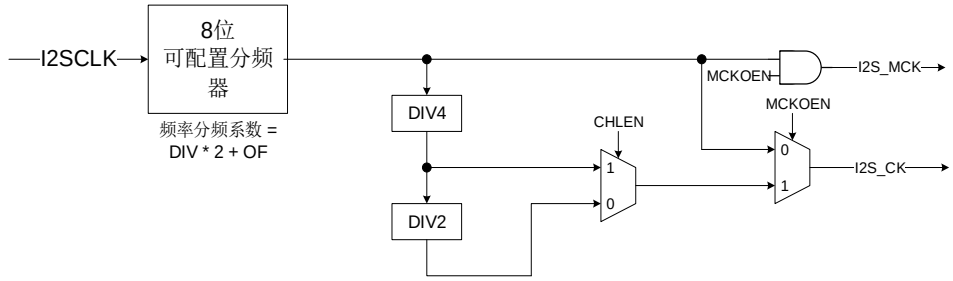


图 21-50. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=1）



21.4.4. I2S 时钟

图 21-51. I2S 时钟生成结构框图



I2S 时钟生成器框图如*图21-51. I2S 时钟生成结构框图*所示。I2S 接口时钟是通过 SPI_I2SPSC 寄存器的 DIV 位，OF 位和 MCKOEN 位以及 SPI_I2SCTL 寄存器的 CHLEN 位来配置的。时钟源是系统时钟（CK_SYS）。I2S 比特率可以通过*表 21-7. I2S 比特率计算公式*所示的公式计算。

表 21-7. 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 21-8. 音频采样频率计算公式*所列的公式进行配置。

表 21-8. 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))

21.4.5. 运行

运行模式

运行模式是通过 SPI_I2SCTL 寄存器的 I2SOPMOD 位来选择的。共有四种运行模式可供选择：主机发送模式，主机接收模式，从机发送模式和从机接收模式。各种运行模式下 I2S 接口信号的方向如*表 21-9. 各种运行模式下 I2S 接口信号的方向*所示。

表 21-9. 各种运行模式下 I2S 接口信号的方向

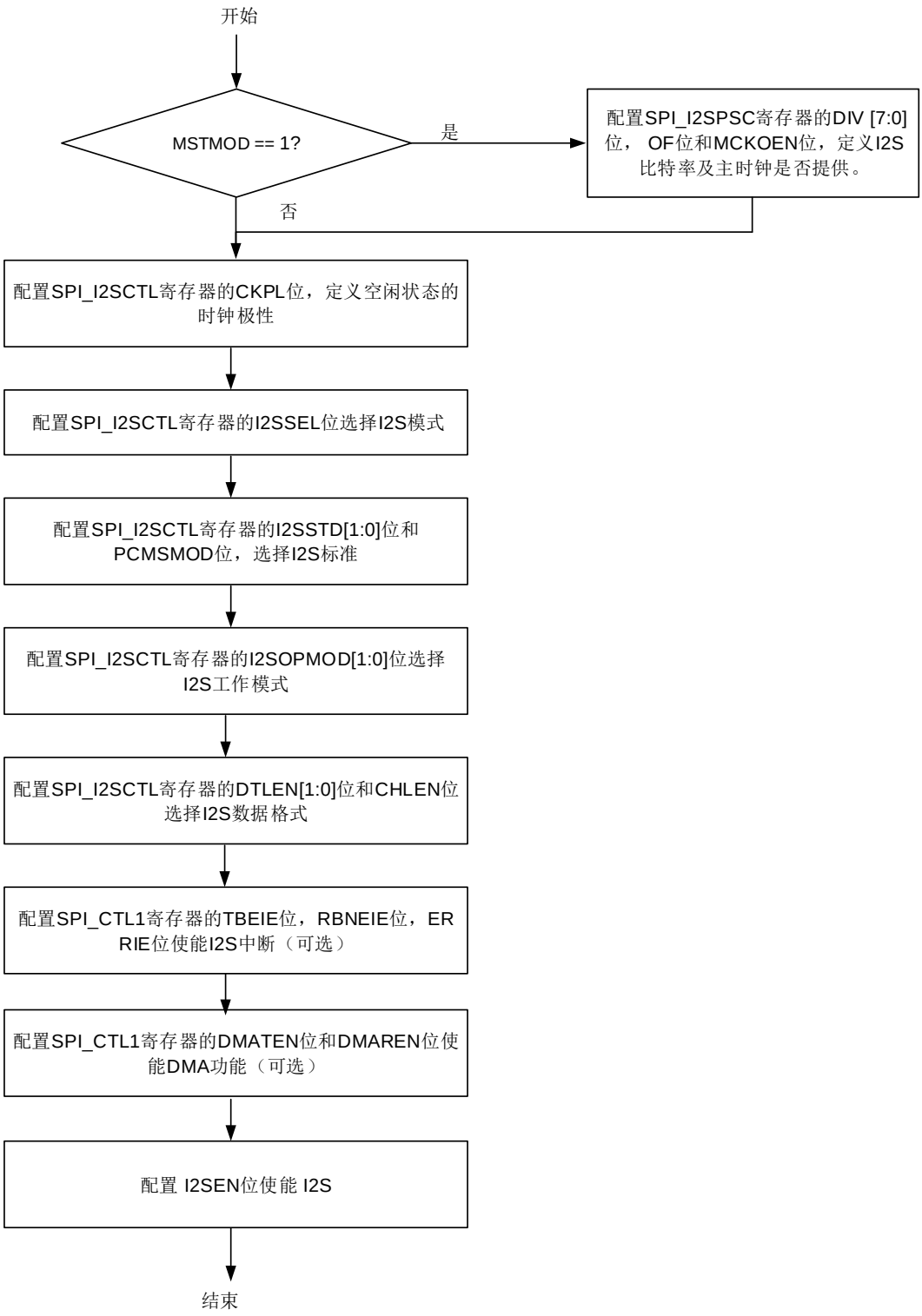
运行模式	I2S_MCK	I2S_CK	I2S_WS	I2S_SD
主机发送	输出或 NU(1)	输出	输出	输出
主机接收	输出或 NU(1)	输出	输出	输入
从机发送	输出或 NU(1)	输入	输入	输出
从机接收	输出或 NU(1)	输入	输入	输入

1. NU表示该引脚没有被I2S使用，可以用于其他功能。

I2S 初始化流程

I2S初始化过程如*图21-52. I2S初始化流程*所示。

图 21-52. I2S 初始化流程



I2S 主机发送流程

TBE标志位被用来控制发送流程。如前文所述，TBE标志位表示发送缓冲区空，此时，如果SPI_CTL1寄存器的TBEIE位为1，将产生中断。首先，发送缓冲区为空（TBE为1），且移位寄

存器中没有发送序列。当16位数据被写入SPI_DATA寄存器时（TBE变为0），数据立即从发送缓冲区装载到移位寄存器中（TBE变为1）。此时，发送序列开始。

数据是并行地装载到16位移位寄存器中的，然后串行地从I2S_SD引脚发出（高位先发）。下一个数据应该在TBE为1时写入SPI_DATA寄存器。数据写入SPI_DATA寄存器之后，TBE变为0。当前发送序列结束时，发送缓冲区的数据会自动装载到移位寄存器中，然后TBE标志变回1。为了保证连续的音频数据发送，下一个将要发送的数据必须在当前发送序列结束之前写入SPI_DATA寄存器。

对于除PCM标准外的所有标准，I2SCH标志用来区别当前传输数据所属的通道。I2SCH标志在每次TBE标志由0变1的时候更新。刚开始I2SCH标志为0，表示左通道的数据应该被写入SPI_DATA寄存器。

为了关闭I2S，I2SEN位必须在TBE标志为1且TRANS标志为0之后清零。

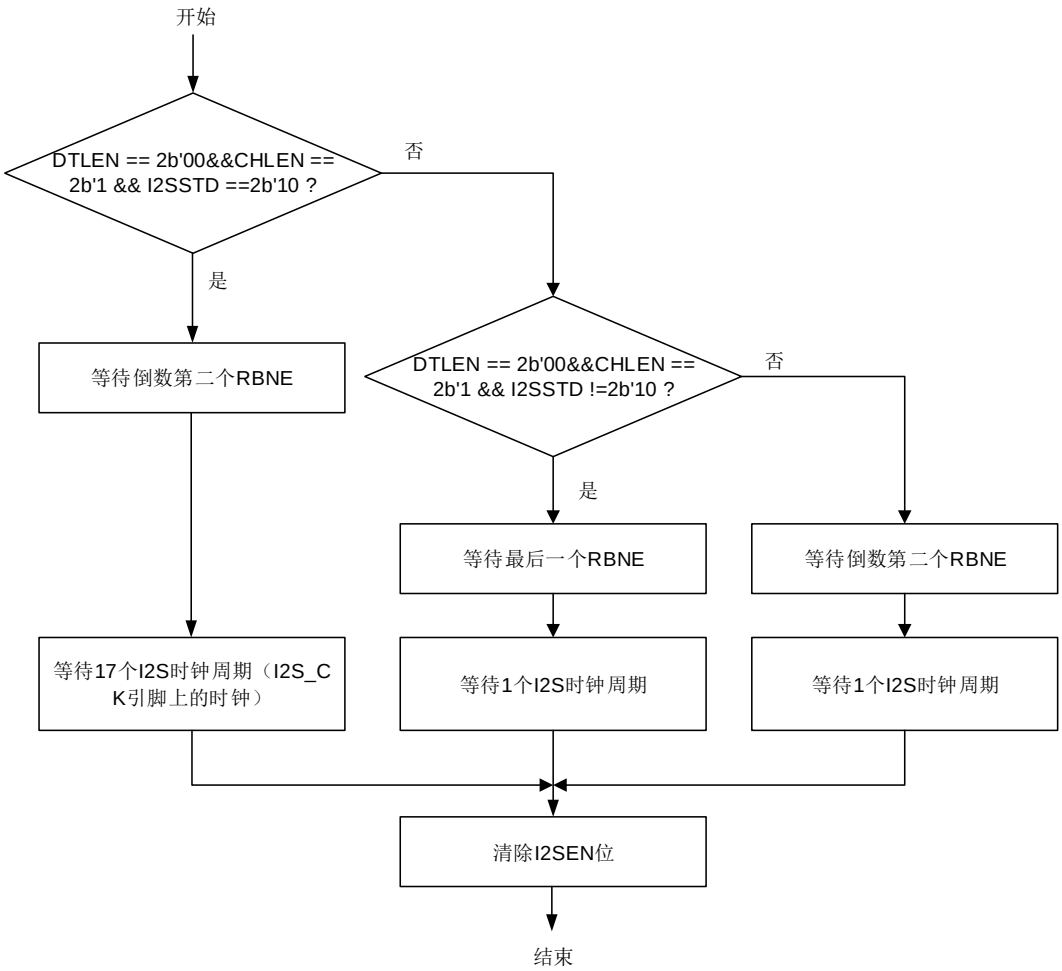
I2S 主机接收流程

RBNE标志被用来控制接收序列。如前文所述，RBNE标志表示接收缓冲区非空，如果SPI_CTL1寄存器的RBNEIE位为1，将产生中断。当SPI_I2SCTL寄存器的I2SEN位被置1时，接收流程立即开始。首先，接收缓冲区为空（RBNE为0）。当一个接收流程结束时，接收到的数据将从移位寄存器装载到接收缓冲区（RBNE变为1）。当RBNE为1时，用户应该将数据从SPI_DATA寄存器中读走。读操作完成后，RBNE变为0。必须在下一次接收结束之前读走SPI_DATA寄存器中的数据，否则将发生接收过载错误。此时RXORERR标志位会被置1，如果SPI_CTL1寄存器的ERRIE位为1，将会产生中断。这种情况下，必须先关闭I2S再打开I2S，然后再恢复通讯。

对于除PCM之外的所有标准来说，I2SCH标志用来区分当前传输数据所属的通道。I2SCH标志在每次RBNE标志由0变1时更新。

为了关闭I2S，不同的音频标准，数据长度和通道长度采用不同的操作步骤。每种情况的操作如*图21-53. I2S主机接收禁能流程*所示。

图 21-53. I2S 主机接收禁能流程



I2S 从机发送流程

从机发送流程和主机发送流程相似，不同之处如下：

在从机模式下，从机需要在外部主机开始通讯之前使能。当外部主机开始发送时钟信号且I2S_WS信号请求传输数据时，发送流程开始。数据需要在外部主机发起通讯之前写入SPI_DATA寄存器。为了确保音频数据的连续传输，必须在当前发送序列结束之前将下一个待发送的数据写入SPI_DATA寄存器，否则会产生发送欠载错误。此时TXURERR标志会置1，如果SPI_CTL1寄存器的ERRIE位为1，将会产生中断。这种情况下，必须先关闭I2S再打开I2S来恢复通讯。从机模式下，I2SCH标志是根据外部主机发送的I2S_WS信号而变化的。

为关闭I2S，必须在TBE标志变为1且TRANS标志变为0之后，才能清除I2SEN位。

I2S 从机接收流程

从机接收流程与主机接收流程类似。不同之处如下。

在从机模式下，从机需要在外部主机开始通讯之前使能。当外部主机开始发送时钟信号且I2S_WS信号指示数据开始时，接收流程开始。从机模式下，I2SCH标志是根据外部主机发送

的I2S_WS信号而变化的。

为了关闭I2S，必须在收到最后一个RBNE之后立即清除I2SEN位。

21.4.6. 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。

21.4.7. 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。

- **发送缓冲区空标志（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 21-10. I2S 中断*总结了 I2S 中断事件和相应的使能位。

表 21-10. 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	

21.5. SPI/I2S 寄存器

SPI0基地址：0x4001 3000

SPI1/I2S1基地址：0x4000 3800

21.5.1. 控制寄存器 0（SPI_CTL0）

地址偏移：0x00
复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

该寄存器在I2S模式下没有意义。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BDEN	BDOEN	CRCEN	CRCNT	FF16	RO	SWNSSEN	SWNSS	LF	SPIEN	PSC[2:0]			MSTMOD	CKPL	CKPH
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	BDEN	双向数据模式使能 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：CRC计算禁止 1：CRC计算使能
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（TCRC） 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：SPI设备禁止 1：SPI设备使能
5:3	PSC[2:0]	主时钟预分频选择 000：PCLK/2　100：PCLK/32 001：PCLK/4　101：PCLK/64 010：PCLK/8　110：PCLK/128 011：PCLK/16　111：PCLK/256 当使用SPI0时，PCLK=PCLK2，当使用SPI1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时钟跳变沿采集第一个数据

21.5.2. 控制寄存器 1（SPI_CTL1）

地址偏移：0x04

复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

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：TBE 中断禁止 1：TBE 中断使能。当 TBE 置位时，产生中断。
6	RBNEIE	接收缓冲区非空中断使能 0：RBNE 中断禁止. 1：RBNE 中断使能。当 RBNE 置位时，产生中断。
5	ERRIE	错误中断使能 0：错误中断禁止 1：错误中断使能。当 CRCERR 位，CONFERR 位，RXORERR 位或者 TXURERR 位置 1 时，产生中断。
4	TMOD	SPI TI 模式使能 0：SPI TI 模式禁止 1：SPI TI 模式使能
3	NSSP	SPI NSS 脉冲模式使能 0：SPI NSS 脉冲模式禁止 1：SPI NSS 脉冲模式使能
2	NSSDRV	NSS 输出使能 0：NSS 输出禁止 1：NSS 输出使能。 当 SPI 使能时，如果 NSS 引脚配置为输出模式，NSS 引脚在主模式时被拉低。如果 NSS 引脚配置为输入模式，NSS 引脚在主模式时被拉高，此时该位无效。
1	DMATEN	发送缓冲区 DMA 使能 0：发送缓冲区 DMA 禁止 1：发送缓冲区 DMA 使能。当 SPI_STAT 中的 TBE 置位时，将会在相应的 DMA 通道上产生一个 DMA 请求。

0	DMAREN	接收缓冲区 DMA 使能 0：接收缓冲区 DMA 禁止 1：接收缓冲区 DMA 使能。当 SPI_STAT 中的 RBNE 置位时，将会在相应的 DMA 通道上产生一个 DMA 请求。

21.5.3. 状态寄存器（SPI_STAT）

地址偏移：0x08

复位值：0x0000 0002

该寄存器可以按半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							FERR	TRANS	RXORERR	CONFERR	CRCERR	TXURERR	I2SCH	TBE	RBNE
							rc_w0	r	r	r	rc_w0	r	r	r	r

位/位域	名称	描述
31:9	保留	必须保持复位值。
8	FERR	帧错误 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：先读或写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模式下该位无用，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

21.5.4. 数据寄存器（SPI_DATA）

地址偏移：0x0C
复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

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
SPI_DATA[15:0]															
rw															

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。

21.5.5. CRC 多项式寄存器（SPI_CRCPOLY）

地址偏移：0x10
复位值：0x0000 0007

该寄存器可以按半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CRCPOLY[15:0]	CRC多项式寄存器值 该值包含了CRC多项式，用于CRC计算，默认值为0007h。

21.5.6. 接收 CRC 寄存器（SPI_RCRC）

地址偏移：0x14
复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	RCRC[15:0]	接收CRC寄存器值 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。

21.5.7. 发送 CRC 寄存器（SPI_TCRC）

地址偏移：0x18

复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	TCRC[15:0]	发送CRC寄存器值 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。

21.5.8. I2S 控制寄存器（SPI_I2SCTL）

地址偏移：0x1C

复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				I2SSEL	I2SEN	I2SOPMOD[1:0]		PCMSMOD	保留	I2SSTD[1:0]		CKPL	DTLEN[1:0]		CHLEN
				rw	rw	rw		rw		rw		rw	rw		rw

位/位域	名称	描述

31:12	保留	必须保持复位值。
11	I2SSEL	I2S模式选择 0：SPI模式 1：I2S模式 当SPI或I2S关闭时配置该位。
10	I2SEN	I2S使能 0：I2S禁止 1：I2S使能 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。

21.5.9. I2S 时钟预分频寄存器（SPI_I2SPSC）

地址偏移：0x20

复位值：0x0000 0002

该寄存器可以按半字（16位）或字（32位）访问。

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						MCKOEN	OF	DIV[7:0]							
						rw	rw	rw							

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：I2S_MCK输出禁止 1：I2S_MCK输出使能 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。

21.5.10. SPI0 四路 SPI 控制寄存器（SPI_QCTL）

地址偏移：0x80

复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

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	四路SPI模式读选择 0：SPI四路模式写操作 1：SPI四路模式读操作 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	四路SPI模式使能 0：SPI工作在单路模式 1：SPI工作在四路模式 该位仅能在SPI未通信时配置（TRANS位清零）。 该位仅适用于SPI0。

22. 比较器（CMP）

22.1. 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。

比较器可通过模拟信号将 MCU 从低功耗模式中唤醒，在一定的条件下，可将模拟信号作为 TIMER 的触发源，结合 DAC 和 TIMER 的 PWM 输出，可以实现电流控制。

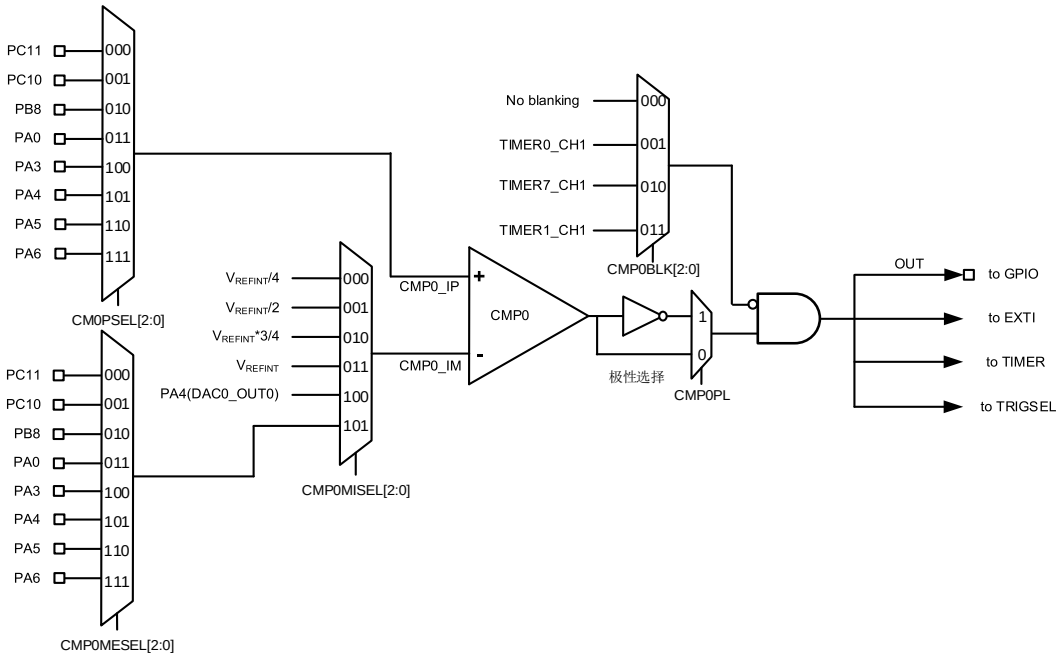
22.2. 主要特征

- 轨对轨比较器；
- 迟滞可配置；
- 速度、功耗可配置；
- 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；
- 输出到 EXTI；
- 输出到 NVIC；
- 输出到 TRIGSEL。

22.3. 功能描述

比较器的框图展示如下：

图 22-1. 比较器框图



注意：V_{REFINT} 是 1.2V。

22.3.1. 比较器时钟

比较器与 APB 总线连接，时钟与 PCLK 同步。它拥有独立的复位和时钟使能位。

22.3.2. 比较器的 I / O 配置

在被选为比较器输入端之前，相应管脚必须配置为模拟模式。

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

比较器的输出可同时实现内部和外部输出。

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；

为了在深度睡眠模式下工作，比较器端口的极性选择和输出重定向不会因为 PCLK 关闭。

*表 22-1 CMP 的输入和输出总结*详细描述了 CMP 的输入和输出。

表 22-1 CMP 的输入和输出总结

	CMP0
CMP 同相输入连接到 I / O	PA0 PA3 PA4 PA5 PA6

	CMP0
	PB8 PC10 PC11
CMP 反相输入连接到 I / O	PA0 PA3 PA4 PA5 PA6 PB8 PC10 PC11
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
CMP 输出连接到 I / O	PB9 PF2
CMP 输出连接到 EXTI	•
CMP 输出连接到 TRIGSEL	•
CMP 输出连接到内部信号	TIMER0_CH0 TIMER7_CH0

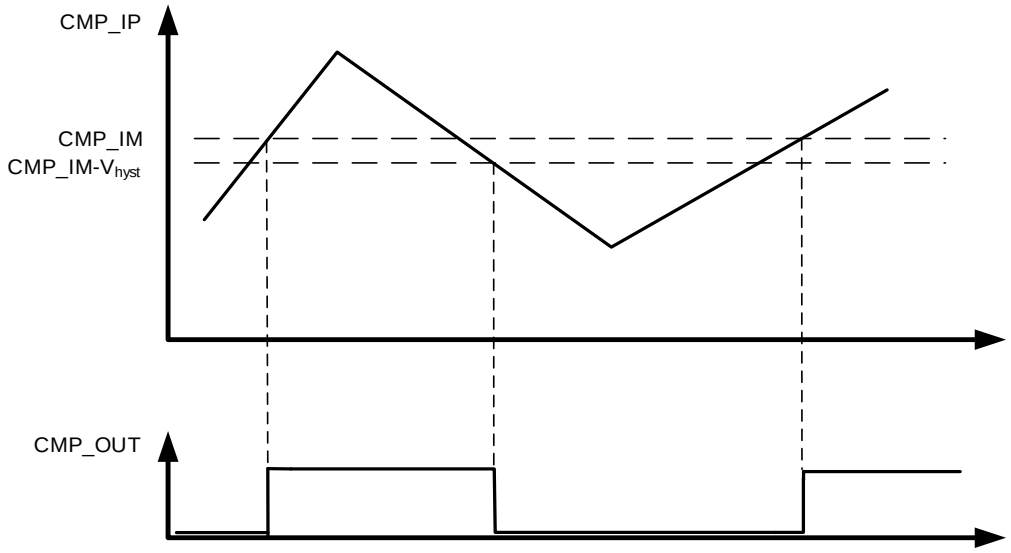
22.3.3. 比较器供电模式

对于给定的程序，在比较器功耗和传输迟滞之间存在着权衡，可通过寄存器 CMPx_CS 的位 CMxPM[1:0]的配置进行调整。当 CMPxM[1:0]为 2’b00 时，比较器以运行速度最快和功耗最大模式工作，但当 CMPxM[1:0]为 2’b11 时，比较器以运行速度最慢和功耗最小的模式工作。

22.3.4. 比较器迟滞

为了避免噪声信号所引起的假输出，电路设计了可编程的迟滞功能，通过配置控制状态寄存器来控制迟滞电压值。该功能可以在无需要时关闭。

图 22-2. 比较器迟滞



22.3.5. 比较器寄存器写保护

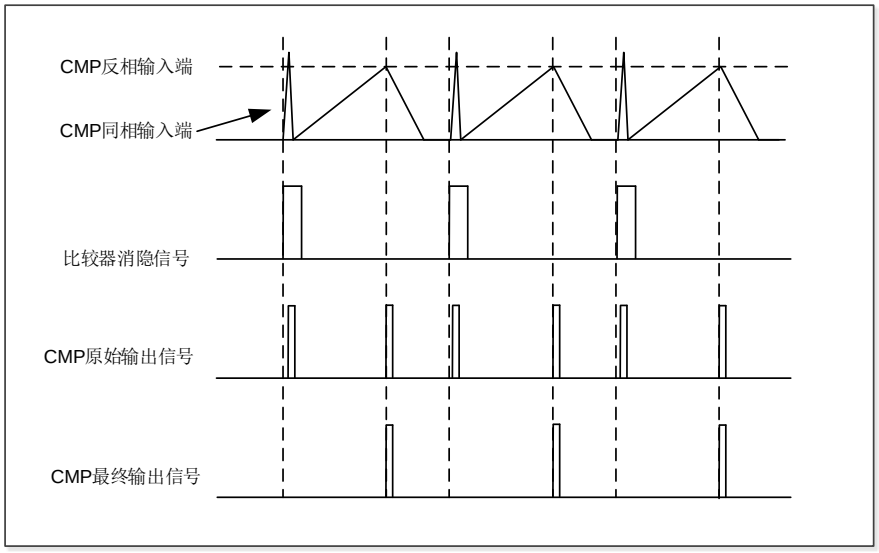
比较器的控制状态寄存器（CMPx_CS）可通过设置 CMPxLK 位为 1 来进行写保护，CMPx_CS 寄存器，包含 CMPxLK 位，就会变为只读位，只有在 MCU 复位时才可以复位。

22.3.6. 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 22-3. 比较器的输出消隐*显示了比较器的输出消隐功能。

图 22-3. 比较器的输出消隐



22.3.7. 电压定标器功能

电压定标器功能可为 CMP 输入提供可选择的 1 / 4、1 / 2、3 / 4 参考电压。它由位于 CMP 控制状态寄存器中的 CMPxSEN 位和 CMPxBEN 位控制，CMPxSEN 位和 CMPxBEN 位分别用于使能 V_{REFINT} 电压输出和分压电路，以产生所选择的电压。

22.3.8. 比较器中断

CMP 输出连接到 EXTI，EXTI 线对每个 CMP 都是独占的。通过这个功能，可以产生中断或者事件，用于退出省电模式。

22.4. CMP 寄存器

CMP 基地址：0x4001 7C00

22.4.1. CMP 控制状态寄存器（CMPx_CS）

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
CMP0LK	CMP0O	保留						CMP0SEN	CMP0BEN	保留	CMP0BLK[2:0]			CMP0HST[1:0]	
rwo	r							rw	rw		rw			rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP0PL	CMP0OSEL[3:0]		CMP0PSEL[2:0]			CMP0MISEL[2:0]			CMP0MESEL[2:0]			CMP0M[1:0]		保留	CMP0EN
rw	rw		rw			rw			rw			rw			rw

位/位域	名称	描述
31	CMP0LK	CMP0 写保护 该位可将 CMP0 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：CMPx_CS[31:0]是可读可写位 1：CMPx_CS[31:0]是只读位
30	CMP0O	CMP0 输出 该位反映 CMP0 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
29:24	保留	必须保持复位值。
23	CMP0SEN	电压标量使能位 该位可通过软件置位和清除，可使能 VREFINT 分频器的输出，被视为反相输入端。 0：除能带隙标量 1：使能带隙标量
22	CMP0BEN	标量桥接使能位 0：除能标量电阻桥接功能 1：使能标量电阻桥接功能
21	保留	必须保持复位值。
20:18	CMP0BLK[2:0]	CMP0 输出消隐源 该位域用于选择哪个定时器输出控制比较器的输出消隐。 000：无消隐 001：选择 TIMER0_CH1 为消隐源 010：选择 TIMER7_CH1 为消隐源

		011：选择 TIMER1_CH1 为消隐源 100~111：保留
17:16	CMP0HST[1:0]	CMP0 迟滞 该位域用于控制迟滞水平。 00：无迟滞 01：低迟滞 10：中迟滞 11：高迟滞
15	CMP0PL	CMP0 输出极性 该位用于控制输出极性。 0：输出是正相的 1：输出是反相的
14:13	CMP0OSEL[1:0]	CMP0 输出选择 该位域用于控制 CMP 输出选择。 00：无选择 01：定时器 0 通道 0 输入捕获 10：定时器 7 通道 0 输入捕获 11：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。
12:10	CMP0PSEL[2:0]	CMP0_IP 输入选择 该位域用于选择 CMP0 的输入端 CMP0_IP 的输入源。 000：PC11 001：PC10 010：PB8 011：PA0 100：PA3 101：PA4 110：PA5 111：PA6
9:7	CMP0MISEL[2:0]	CMP0_IM 内部输入选择 该位域用于选择 CMP0 的输入端 CMP0_IM 的内部输入源。 000：V_{REFINT} / 4 001：V_{REFINT} / 2 010：V_{REFINT} * 3 / 4 011：V_{REFINT} 100：PA7（DAC0_OUT0） 101：CMP_IM 外部输入选择 110~111：保留
6:4	CMP0MESEL[2:0]	CMP0_IM 外部输入选择 该位域用于选择 CMP0 的输入端 CMP0_IM 的外部输入源。

		000: PC11 001: PC10 010: PB8 011: PA0 100: PA3 101: PA4 110: PA5 111: PA6
3:2	CMP0M[1:0]	CMP0 模式 该位域用于控制 CMP 的运行模式以调整速度和功耗。 00：高速 / 全功耗 01 / 10：中速 / 中功耗 11：低速 / 低功耗
1	保留	必须保持复位值。
0	CMP0EN	CMP0 使能 0：CMP 禁能 1：CMP 使能

23. 控制器局域网络（CAN）

23.1. 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和设备之间相互通信的总线标准。CAN 网络接口支持 CAN 总线协议 2.0A/B、ISO11898-1:2015 规范和 BOSCH CAN-FD 规范。

CAN 总线控制器集成了可灵活配置的邮箱系统用于 CAN 帧的发送和接收。邮箱系统包含一组邮箱，用于存储控制数据，时间戳，消息标识符和消息数据，最大支持 32 个邮箱。可将邮箱配置为接收 FIFO，接收 FIFO 具有标识符过滤的功能，可最大支持 104 个扩展标识符的过滤，或者 208 个标准标识符的过滤，或者 416 个对标识符部分 8 位的过滤，最多有 32 个标识符过滤表元素可通过接收 FIFO/邮箱私有过滤寄存器进行配置。

23.2. 主要特征

- 支持CAN总线协议2.0A/B；
- 遵循ISO 11898-1:2015规范；
- 支持CAN FD帧，最大64字节数据，通信波特率最大为8 Mbit/s；
- 支持CAN常规帧，最大8字节数据，通信波特率最大为1 Mbit/s；
- 支持发送和接收时间戳，基于16位内部计数器；
- 支持传输延迟补偿，用于CAN FD帧的高速率数据阶段；
- 中断可配置屏蔽；
- 支持4种通信模式：正常模式，暂停模式，回环静默模式，和监听模式；
- 支持2种省电模式：CAN_Disable模式，和虚拟联网模式；
- 支持2种从虚拟联网模式唤醒的方式：唤醒匹配事件，和唤醒超时事件；
- 最大32个邮箱，此时每个邮箱都配置为8字节数据长度，可灵活配置为发送或接收邮箱；
- 支持通过一个特殊帧同步全局网络时间。
- 协议控制器的时钟（CANCLK）源可配置[1]。

发送

- 支持发送中止；
- 发送邮箱状态可查看；
- 发送帧消息的CRC校验；
- 支持发送优先级：最小邮箱号优先，或最高优先级优先。

接收

- 接收私有过滤寄存器用于每个接收邮箱或者接收FIFO；
- 接收邮箱公有过滤寄存器用于接收邮箱，接收FIFO公有过滤寄存器用于接收FIFO；
- 支持接收优先级，可配置在匹配阶段的接收邮箱和接收FIFO的优先级；
- 接收FIFO的标识符过滤功能支持最大104个扩展标识符的过滤，或者208个标准标识符的过滤，或者416个对标识符部分8位的过滤；
- 深度为6帧的接收FIFO，支持DMA功能。

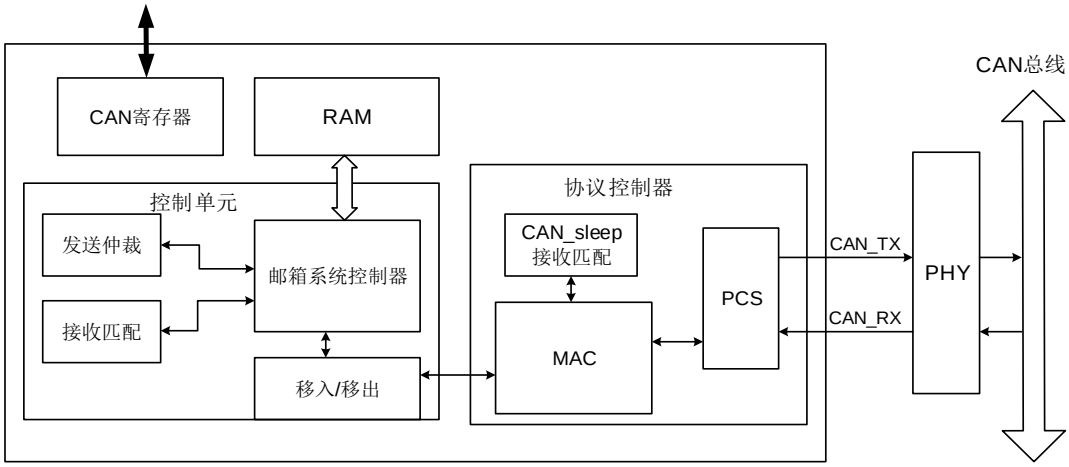
注意：

（1）CANCLK 时钟源的选择有使用限制，请查看 GD32A50x 系列的勘误手册。

23.3. 功能说明

CAN 模块结构框图如*图 23-1. CAN 模块结构框图*所示。

图 23-1. CAN 模块结构框图



CAN 模块包含三个部分：

- 协议控制器

  协议控制器管理CAN总线上的通信，包括：

  MAC（介质访问控制器）：
  - 位填充/去填充；
  - FD帧的填充位计数；
  - 添加CRC；
  - 构造MAC帧；
  - 检测ACK，发送ACK。

  PCS（物理编码子层）：
  - 位时间；
  - 同步；
  - TDC（传输延迟补偿）。

  虚拟联网接收匹配：
  - 在虚拟联网模式下进行接收匹配。
- 控制单元

  控制单元主要用于发送和接收的RAM管理，包括：

  发送仲裁：
  - 找出当前优先级最高的帧。

  接收匹配：
  - 按配置顺序将接收移位缓存（一个内部描述符）中的帧数据内容与接收邮箱或者Rx FIFO中的域进行匹配。

  邮箱系统控制器：

- 管理发送和接收的RAM分配，控制邮箱描述符CODE，控制Rx FIFO指针，完成总线对RAM的访问申请。

  消息存储在CAN模块专用的RAM区。专用RAM的基地址为模块基地址。

移入/移出：

- 在选择的邮箱描述符 / Rx FIFO描述符与发送或者接收移位缓存之间进行数据搬运。

■ CAN寄存器

CAN寄存器负责完成CAN模块与系统总线的交互。

23.3.1. 邮箱描述符

邮箱描述符如下表*表23-1. 64字节数据的邮箱描述符*所示，可用于标准帧（11位标识符）和扩展帧（29位标识符）。每个邮箱可由16字节，24字节，40字节或者72字节组成，分别包含8字节，16字节，32字节或者64字节的数据。偏移地址从0x80到0x27F的RAM区域可用作邮箱。

表 23-1. 64 字节数据的邮箱描述符

	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
MDES0	FDF	BRS	ESI	保留	CODE[3:0]				保留	SRR	IDE	RTR	DLC[3:0]				TIMESTAMP[15:0]															
MDES1	PRIO[2:0]			ID_STD[10:0]											ID_EXD[17:0]																	
MDES2	DATA_0[7:0]								DATA_1[7:0]								DATA_2[7:0]								DATA_3[7:0]							
…	…								…								…								…							
MDES17	DATA_60[7:0]								DATA_61[7:0]								DATA_62[7:0]								DATA_63[7:0]							

MDES0：邮箱描述符字 0

地址偏移：0x80

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FDF	BRS	ESI	保留	CODE[3:0]				保留	SRR	IDE	RTR	DLC[3:0]			
rw	rw	rw		rw					rw	rw	rw	rw			

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

位/位域	名称	描述
31	FDF	FD 格式指示 该位用于区分 CAN 常规帧和 CAN FD 帧。 对于接收邮箱，不需要配置该位，该位用于存储 CAN 总线上接收到的该位值。
30	BRS	位速率切换 该位用于定义 CAN FD 帧中位速率是否切换到更高的速率。 对于接收邮箱，不需要配置该位，该位用于存储 CAN 总线上接收到的该位值。
29	ESI	错误状态指示位 该位指示发送节点是主动错误状态或者被动错误状态。在 CAN 常规帧中该位保留。 对于发送邮箱，主动错误节点发送为显性位，被动错误节点发送为隐性位。 对于接收邮箱，不需要配置该位，该位用于存储 CAN 总线上接收到的该位值。

28	保留	必须保持复位值。
27:24	CODE[3:0]	邮箱代码（CODE） 该位域可被 CPU 和 CAN 模块访问，作为邮箱发送仲裁和接收匹配流程的一部分。代码取值可参考*表 23-3. 接收邮箱 CODE* 和*表 23-4. 发送邮箱 CODE*。
23	保留	必须保持复位值。
22	SRR	替代远程请求 该位仅用于扩展帧格式。 对于发送邮箱，该位须设置为 1（隐性），如果总线发送该位为 0（显性），表示该节点发生了仲裁丢失。 对于接收邮箱，该位用于存储 CAN 总线上接收到的该位值。 0：在扩展帧中，无效的发送 1：在扩展帧中强制发送‘1’
21	IDE	标识符扩展位 该位指示该帧是标准帧还是扩展帧。 对于接收邮箱，该位用于存储CAN总线上接收到的该位值。 0：帧格式为标准帧 1：帧格式为扩展帧
20	RTR	远程传输请求 对于发送邮箱：当该位设置为 1（隐性），而总线发送该位为 0（显性），表示该节点发生了仲裁丢失；当该位设置为 0（显性），而总线发送该位为 1（隐性），表示发生了位错误；当配置的位值与发送的位值相同，表示一个成功的位传输。 对于接收邮箱，该位用于存储 CAN 总线上接收到的该位值。 0：对于发送邮箱，表示当前邮箱有一个数据帧要发送。对于接收邮箱，该位将参与匹配过程。 1：对于发送邮箱，表示当前邮箱有一个远程请求帧要发送。对于接收邮箱，表示可能接收到一个远程请求帧。 **注意：**当配置为 CAN FD 帧时，该位必须配置为 0。该位只能用于 CAN 常规帧。
19:16	DLC[3:0]	数据字节长度代码 该位域表示发送帧和接收帧的数据字节长度。 对于接收邮箱，不需要配置该位，该位域将被 CAN 总线上接收到的 DLC 域值改写。 对于发送邮箱，表示要发送的帧数据字节长度。当 RTR 位为 1 时，有一个远程请求帧要发送，没有数据场，忽略该位域。
15:0	TIMESTAMP[15:0]	时间戳 该位域是在发送帧或者接收帧的标识符域出现在 CAN 总线的时刻，抓取到的内部计数器的值。

表 23-2. DLC 表示的数据字节长度

DLC	数据字节大小
i（0≤i≤8）	i（0≤i≤8）
9	12
10	16

DLC	数据字节大小
11	20
12	24
13	32
14	48
15	64

表 23-3. 接收邮箱 CODE

CODE	含义	接收后的 CODE	完成服务(1)	RRFRMS(2)	描述
0b0000	INACTIVE	-	-	-	邮箱不参与匹配过程。
0b0100	EMPTY	FULL	-	-	当成功接收了一个帧后（在移入过程之后），CODE域自动更新为FULL。
0b0010	FULL	FULL	是	-	保持为FULL。如果新的一帧在该邮箱完成服务之后移入该邮箱，则邮箱代码保持为FULL。
		OVERRUN	否		如果邮箱代码已经为FULL，而在该邮箱完成服务之前又有新的一帧移入该邮箱，则邮箱代码自动更新为OVERRUN。
0b0110	OVERRUN	FULL	是	-	如果邮箱代码为OVERRUN，在邮箱完成服务之后有新的一帧移入了该邮箱，则邮箱代码更新为FULL。
		OVERRUN	否		如果邮箱代码为OVERRUN，而有新的一帧必须移入，则该邮箱将再次被覆盖，邮箱代码保持为OVERRUN。
0b1010	RANSWER(3)	TANSWER(0x1110)	-	0	邮箱代码为RANSWER的邮箱用于远程请求帧接收的识别。在接收远程请求帧之后，如果CAN_CTL2寄存器的RRFRMS位为0，则该邮箱将自动设置发送一个响应帧，邮箱代码自动修改为TANSWER。
		-		1	邮箱在接收匹配和发送仲裁过程中被忽略。
CODE[0] = 1	BUSY(4)	FULL	-	-	表示邮箱正在更新。
		OVERRUN			

1. 完成服务：邮箱被CPU读取过，并且通过读取CAN_TIMER寄存器或者读取其他邮箱的方式解锁了该邮箱。
2. 远程请求帧存储位，参考*控制寄存器2（CAN_CTL2*。
3. 邮箱代码为0b1010的不可被中止。
4. 对于接收邮箱，如果CODE[0]位置位，则对应的邮箱将不参与接收匹配过程。

表 23-4. 发送邮箱 CODE

CODE	含义	发送后的 CODE	RTR	描述
0b1000	INACTIVE	-	-	邮箱不参与发送仲裁过程。
0b1001	ABORT	-	-	邮箱不参与发送仲裁过程。
0b1100	DATA	INACTIVE	0	发送数据帧。在发送之后，该邮箱自动更新为INACTIVE状态。

CODE	含义	发送后的CODE	RTR	描述
	REMOTE	EMPTY	1	发送远程请求帧。在发送之后，该邮箱自动变为相同标识符的接收空邮箱。
0b1110	TANSWER	RANSWER	-	当接收到一个匹配的远程请求帧，控制单元会自动改写邮箱的CODE到一个中间态CODE，TANSWER。在发送远程应答帧之后，邮箱将自动恢复到RANSWER状态。对邮箱手动设置TANSWER会有相同的效果。根据RTR位的值，远程应答帧可以是一个数据帧或者新的一个远程请求帧。

MDES1：邮箱描述符字 1

地址偏移：0x84

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRIO[2:0]			ID_STD[10:0]											ID_EXD[17:16]	
rw			rw											rw	

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

位/位域	名称	描述
31:29	PRIO[2:0]	本地优先级 该位域只有当 CAN_CTL0 寄存器的 LAPRIOEN 位为 1 时才适用。 该位域仅用于发送邮箱，但在发送消息时不发送这些位，它们附加到标识符之前，共同用作发送优先级的判断。
28:18	ID_STD[10:0]	标准帧的标识符 对于标准帧，这 11 个有效意义位用作发送接收帧的标识符。邮箱描述符字 1 的低 18 位忽略不用。
17:0	ID_EXD[17:0]	扩展帧的标识符 对于扩展帧，ID_STD[10:0]和这 18 位共同用作发送接收帧的标识符。

MDESx：邮箱描述符字 x（x = 2..17）

地址偏移：0x80 + 0x04 * x (x = 2..17)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA_i[7:0]								DATA_i+1[7:0]							
rw								rw							

15	14	13	12	11	rw	9	8	7	6	5	4	3	2	1	0
DATA_i+2[7:0]								DATA_i+3[7:0]							
rw								rw							

位/位域	名称	描述
31:24	DATA_i[7:0]	数据字节 i (i = 4*x - 8)

		参考 DATA_i+3[7:0]描述。
23:16	DATA_i+1[7:0]	数据字节 i+1 (i = 4*x - 8) 参考 DATA_i+3[7:0]描述。
15:8	DATA_i+2[7:0]	数据字节 i+2 (i = 4*x - 8) 参考 DATA_i+3[7:0]描述。
7:0	DATA_i+3[7:0]	数据字节 i+3 (i = 4*x - 8) 一个数据帧最大包含 64 个字节数据，主要由邮箱的 DLC 值决定。 对于接收帧，该位域用于存储 CAN 总线上接收到的数据。

邮箱编号

当接收FIFO禁能时，专用的RAM空间只被邮箱占用，因此邮箱编号与邮箱描述符编号相同，邮箱描述符编号按整个邮箱描述符为单位递增，每个邮箱描述符的数据可以是8字节，16字节，32字节或者64字节。

当接收FIFO使能时（CAN FD模式处于禁能状态，因此数据是8字节长度），专用的RAM空间同时被邮箱和FIFO占用，因此统一以8字节数据的邮箱描述符为单位进行描述符计数编号，那么邮箱编号就是邮箱描述符所占用的计数编号。

CAN FD 模式下邮箱数目

当CAN FD模式使能时，由CAN_FDCTL寄存器的MDSZ[1:0]位域来配置邮箱的数目，决定512字节的RAM空间的分配。

表 23-5. 邮箱数目

MDSZ[1:0]	数据字节大小	邮箱数目
0b00	8	32
0b01	16	21
0b10	32	12
0b11	64	7

23.3.2. 接收 FIFO 描述符

接收FIFO描述符如*表23-6. 接收FIFO描述符*所示。

当CAN_CTL0寄存器的RFEN位为1时，按8字节的数据载荷计数，通常被邮箱编号0-5占用的RAM空间被用于接收FIFO。FDES0 – FDES3包含最早接收到的还未被CPU读取的消息。偏移地址从0x90到0xDF的RAM空间保留给FIFO内部使用。

当CAN_CTL0寄存器的RFEN位为1时，按8字节的数据载荷计数，通常被邮箱编号6-31占用的RAM空间被用作标识符过滤器表（可配置为8到104个过滤元素），用于FIFO接收匹配过程。

复位时默认标识符过滤表包含8个过滤元素，从FDES4到FDES11。

表 23-6. 接收 FIFO 描述符

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

FDES0	IDFMN[8:0]		SRR	IDE	RTR	DLC[3:0]	TIMESTAMP[15:0]
FDES1	保留	ID_STD[10:0]				ID_EXD[17:0]	
FDES2	DATA_0[7:0]	DATA_1[7:0]		DATA_2[7:0]		DATA_3[7:0]	
FDES3	DATA_4[7:0]	DATA_5[7:0]		DATA_6[7:0]		DATA_7[7:0]	
0x90 - 0xDC	保留						
FDES4	标识符过滤元素 0						
…	…						
FDES107	标识符过滤元素 103						

FDES0：接收 FIFO 描述符字 0

地址偏移：0x80

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDFMN[8:0]									SRR	IDE	RTR	DLC[3:0]			
r									r	r	r	r			

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

位/位域	名称	描述
31:23	IDFMN[8:0]	标识符过滤元素匹配序号 该位域表示在接收 FIFO 输出中的消息是与哪个标识符过滤元素相匹配。
22	SRR	替代远程请求 该位仅用于扩展帧格式。 对于接收，该位将存储 CAN 总线上接收到的该位值。
21	IDE	标识符扩展位 该位指示了该帧是标准帧还是扩展帧。 0：帧格式为标准帧 1：帧格式为扩展帧
20	RTR	远程传输请求 0：接收数据帧 1：接收远程帧
19:16	DLC[3:0]	数据字节长度代码 该位域表示接收帧的数据字节长度。 对于接收邮箱，不需要配置该位，该位域将被 CAN 总线上接收到的 DLC 域值改写。
15:0	TIMESTAMP[15:0]	时间戳 该位域是在接收帧的标识符域出现在 CAN 总线的时刻，抓取到的内部计数器的值。

FDES1：接收 FIFO 描述符字 1

地址偏移：0x84

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			ID_STD[10:0]											ID_EXD[17:16]	
								r						r	

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

位/位域	名称	描述
31:29	保留	必须保持复位值。
28:18	ID_STD[10:0]	标准帧的标识符 对于标准帧，这 11 个有效意义位用作接收帧的标识符。邮箱描述符字 1 的低 18 位忽略不用。
17:0	ID_EXD[17:0]	扩展帧的标识符 对于扩展帧，ID_STD[10:0]和这 18 位共同用作接收帧的标识符。

FDES2：接收 FIFO 描述符字 2

地址偏移：0x88

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA_0[7:0]								DATA_1[7:0]							
r								r							

15	14	13	12	11	rw	9	8	7	6	5	4	3	2	1	0
DATA_2[7:0]								DATA_3[7:0]							
r								r							

位/位域	名称	描述
31:24	DATA_0[7:0]	数据字节 0 参考 DATA_3[7:0]描述。
23:16	DATA_1[7:0]	数据字节 1 参考 DATA_3[7:0]描述。
15:8	DATA_2[7:0]	数据字节 2 参考 DATA_3[7:0]描述。
7:0	DATA_3[7:0]	数据字节 3 一个数据帧最大包含 8 个字节数据，主要由邮箱的 DLC 值决定。 接收 FIFO 不支持接收 FD 帧。

FDES3：接收 FIFO 描述符字 3

地址偏移：0x8C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA_4[7:0]								DATA_5[7:0]							
r								r							

15	14	13	12	11	rw	9	8	7	6	5	4	3	2	1	0
DATA_6[7:0]								DATA_7[7:0]							
r								r							

位/位域	名称	描述
31:24	DATA_4[7:0]	数据字节 4 参考 DATA_7[7:0]描述。
23:16	DATA_5[7:0]	数据字节 5 参考 DATA_7[7:0]描述。
15:8	DATA_6[7:0]	数据字节 6 参考 DATA_7[7:0]描述。
7:0	DATA_7[7:0]	数据字节 7 一个数据帧最大包含 8 个字节数据，主要由邮箱的 DLC 值决定。 接收 FIFO 不支持接收 FD 帧。

FDESx：接收 FIFO 描述符字 x（x = 4..107）

地址偏移：0xE0 + 4 * (x - 4)

该描述符字给出了标识符过滤元素的3种不同格式，可以通过CAN_CTL0寄存器的FS[1:0]位域来配置。

注意：所有的标识符过滤元素只能同时使用同一种格式，不同的格式不能混合在一个标识符过滤器表中使用。

格式A模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RTR_A	IDE_A	保留	ID_STD_A[10:0]											保留	
			ID_EXD_A[28:16]												
rw	rw		rw												

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

位/位域	名称	描述
31	RTR_A	格式 A 远程帧 该位指示了是否接收匹配的远程帧到 FIFO。 0：远程帧都被拒绝，只存储数据帧 1：数据帧都被拒绝，只存储远程帧
30	IDE_A	格式 A 标识符扩展位

		该位指示了是否接收匹配的扩展帧到 FIFO。 0：扩展帧都被拒绝，只存储标准帧 1：标准帧都被拒绝，只存储扩展帧
29	保留	必须保持复位值。
28:0	ID_STD_A[10:0]/ ID_EXD_A[28:0]	格式 A 标识符 该位域指示一个用于接收 FIFO 匹配过程的完整的标识符（标准格式或者扩展格式）。 如果 IDE_A 为 0，则 18 到 28 位用作标准格式标识符，其余位保留；否则，所有位用作扩展格式标识符。

格式B模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RTR_B0	IDE_B0	ID_STD_B_0[10:0]											保留		
		ID_EXD_B_0[13:0]													
rw	rw	rw													

15	14	13	12	11	rw	9	8	7	6	5	4	3	2	1	0
RTR_B1	IDE_B1	ID_STD_B_1[10:0]											保留		
		ID_EXD_B_1[13:0]													
rw	rw	rw													

位/位域	名称	描述
31	RTR_B0	格式 B 远程帧 0 该位指示了是否接收匹配的远程帧到 FIFO。 0：远程帧都被拒绝，只存储数据帧 1：数据帧都被拒绝，只存储远程帧
30	IDE_B0	格式 B 标识符扩展位 0 该位指示了是否接收匹配的扩展帧到 FIFO。 0：扩展帧都被拒绝，只存储标准帧 1：标准帧都被拒绝，只存储扩展帧
29:16	ID_STD_B_0[10:0]/ ID_EXD_B_0[13:0]	格式 B 标识符 0 该位域指示一个用于接收 FIFO 匹配过程的完整的标准格式标识符或者扩展格式标识符其中 14 位。 如果 IDE_B0 位为 0，则 19 到 29 位用作标准格式标识符，其余位保留；否则，这些位都用作扩展格式标识符其中 14 位，其与接收到的标识符的最高有效 14 位进行比较。
15	RTR_B1	格式 B 远程帧 1 参考 RTR_B0 描述。
14	IDE_B1	格式 B 标识符扩展位 1 参考 IDE_B0 描述。
13:0	ID_STD_B_1[10:0]/ ID_EXD_B_1[13:0]	格式 B 标识符 1 参考 ID_STD_B_0[10:0]/ ID_EXD_B_0[13:0]描述。

格式C模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID_C_0[7:0]								ID_C_1[7:0]							
rw								rw							

15	14	13	12	11	rw	9	8	7	6	5	4	3	2	1	0
ID_C_2[7:0]								ID_C_3[7:0]							
rw								rw							

位/位域	名称	描述
31:24	ID_C_0[7:0]	格式 C 标识符 0 该位域指示一个用于接收 FIFO 匹配过程的标准格式标识符其中 8 位，或者扩展格式标识符其中 8 位。 在标准格式帧和扩展格式帧中，这 8 位都是与其接收到的标识符的最高有效 8 位进行比较。
23:16	ID_C_1[7:0]	格式 C 标识符 1 参考 ID_C_0[7:0]描述。
15:8	ID_C_2[7:0]	格式 C 标识符 2 参考 ID_C_0[7:0]描述。
7:0	ID_C_3[7:0]	格式 C 标识符 3 参考 ID_C_0[7:0]描述。

23.3.3. 通信模式

CAN接口有四种通信模式：

- 正常模式
- 暂停模式
- 回环静默模式
- 监听模式

正常模式

在正常模式，消息的接收、发送，以及错误都正常处理，所有的CAN协议功能都使能。

暂停模式

为了进入暂停模式，需要将CAN_CTL0寄存器的INAMOD位置位，然后置位CAN_CTL0寄存器的HALT位或者设置芯片进入Debug模式。

当CAN模块发出进入暂停模式的请求后，在INAS位置位前执行了如下几个步骤：

1. 等待总线上连续11位隐性位。
2. 等待当前发送或者接收流程完成，也就是所有内部活动比如仲裁、匹配、移入和移出都完成。挂起的移入流程不影响暂停模式的进入。

3. Tx发送管脚驱动为‘1’（隐性电平）。
4. 停止预分频器。
5. 使能CAN_ERR0寄存器的写访问，该寄存器在其他模式中为只读。
6. 置位CAN_CTL0寄存器中的NRDY和INAS位。

当进入了暂停模式时，CAN_CTL0寄存器的INAS位由CAN模块置位。

在暂停模式中，不能发送接收消息，CAN模块预分频器停止工作，所有寄存器都可访问。

为了退出暂停模式，有以下两种方式：

- 清除CAN_CTL0寄存器的INAMOD位。
- 清除CAN_CTL0寄存器的HALT位，或者芯片退出Debug模式。

当CAN模块发出退出暂停模式的请求后，则在CAN预分频器恢复工作之后，CAN_CTL0寄存器的INAS位被清零。退出暂停模式后，CAN模块通过等待11个连续隐性位尝试恢复与CAN总线的同步。

注意：在暂停模式时，发出进入CAN_Disable模式的请求会导致CAN_CTL0寄存器的INAS位清零并且CAN_CTL0寄存器的LPS位置位。

回环静默模式

为了进入该模式，置位CAN_CTL1寄存器的LSCMOD位。在该模式下，所有发送的消息将内部输回到接收管脚，并且将忽略ACK场中的ACK间隙发送位，以确保能接收到自己发送的消息，同时发送和接收中断都能正常产生。

回环静默模式由于模块自检。Rx接收管脚被忽略，Tx管脚保持为隐性电平。

监听模式

为了进入该模式，置位CAN_CTL1寄存器的MMOD位。

在监听模式下，CAN_ERR1寄存器的ERRSI[1:0]位域由CAN模块设为0b01来指示模块此时工作在被动错误状态。在该模式下，所有的错误计数器都被冻结。

在该模式下，发送被禁止，只有被其他CAN节点应答了的消息才能被接收，如果CAN模块检测到一个没有被应答的消息，则位显性错误标志将置位，同时不改变CAN_ERR0寄存器中的RECNT[7:0]和REFCNT[7:0]位域。

23.3.4. 省电模式

CAN接口有两种省电模式：

- CAN_Disable模式
- 虚拟联网模式

在这两种省电模式下，专用的RAM以及处于SRAM的寄存器都不能访问。

CAN_Disable 模式

通过配置CAN_CTL0寄存器的CANDIS位来使能或失能CAN模块。

为了减少电源能耗，当置位CANDIS位来禁能CAN模块时，CAN模块将延迟一段时间后进入CAN_Disable模式，此时CAN_CTL0寄存器的LPS位和NRDY位均置位。

当CAN模块失能时，协议控制器和控制单元的时钟都关闭，寄存器除了CAN_RMPUBF，CAN_RFIFOPUBF，CAN_RFIFOIFMN和CAN_RFIFOMPFx（x = 0..31）都仍可访问，同时专用的RAM也不可访问。

在CAN模块使能后，仍然需要延迟一段时间等待CAN_CTL0寄存器的LPS位清零，以通知协议控制器，CAN模块将请求协议控制器和控制单元恢复时钟。

虚拟联网模式

虚拟联网模式用于在省电模式下接收唤醒帧。该模式可与芯片深度睡眠模式一起使用。

为了进入虚拟联网模式，设置CAN_CTL0寄存器的PNMOD位和PNEN位为1，如需要，可以设置MCU进入深度睡眠模式。

在发出虚拟联网模式请求后，执行如下几个步骤：

1. 等待总线处于空闲状态，或者等待帧间隔的第三个位并检查为隐性位。
2. 置位CAN_CTL0寄存器的PNS位。
3. 请求关闭控制单元的时钟，保持协议控制器时钟运行。

在虚拟联网模式下，控制单元时钟被关闭，而协议控制器保持运行（如果MCU也进入了深度睡眠模式，则需要事先将CAN协议控制器的时钟源配置为HXTAL或者IRC8M，否则CAN协议控制器将丢失时钟而无法继续运行），从而可以继续接收并过滤消息。在该模式下不进行匹配、仲裁、移入和移出流程。

为了退出虚拟联网模式，可按以下方式：

- 当检测到一个唤醒事件，发生了唤醒中断。清除CAN_CTL0寄存器的PNMOD位和PNEN位。
- 清除CAN_CTL0寄存器的PNMOD位和PNEN位。

当CAN模块发出退出虚拟联网模式请求后，CAN模块将等待总线处于空闲状态或者等待帧间隔的第三个位到来时清零CAN_CTL0寄存器的PNS位，恢复到正常模式，CAN模块将重新与CAN总线同步。

23.3.5. 数据发送

对于发送，应用了仲裁机制来决定发送邮箱的优先级是根据消息标识符（PRIO域也可配置参与到发送仲裁中）还是邮箱编号。

CAN FD模式下的邮箱数目由CAN_FDCTL寄存器的MDSZ[1:0]位域来决定，参考*表23-5. 邮箱数目*。

发送流程

为了发送一个CAN帧，需要按如下步骤准备一个发送邮箱：

1. 检查相应邮箱在CAN_STAT寄存器的状态MSx位是否置位，并清除位。
2. 如果邮箱是激活状态（不论是发送邮箱还是接收邮箱），则按*发送邮箱失活*或*接收邮箱失活*来失活该邮箱。当执行了发送邮箱失活操作，则按后续的步骤继续操作。如果执行了接收邮箱失活操作，跳到步骤6。如果邮箱是失活状态（不论是发送邮箱还是接收邮箱），则跳到步骤6。
3. 轮询CAN_STAT寄存器，等待相应MSx位置位，或者置位CAN_INTEN寄存器中相应的MIEx位使能相应的中断，通过中断请求处理。
4. 读CODE域来获取邮箱状态（中止的，或者已发送）。
5. 清零CAN_STAT寄存器的相应标志位MSx位。
6. 写邮箱MDES1字的标识符域（当CAN_CTL0寄存器的LAPRIOEN位置位时，也包括邮箱的PRIO域）。
7. 写邮箱MDESx（x = 2..17）字的载荷数据字节。
8. 配置邮箱MDES0字的IDE，RTR，FDF，BRS，ESI和DLC域。
9. 设置邮箱CODE域为0b1100，来激活邮箱发送帧。当邮箱被激活后，它将会参与仲裁过程，并根据其优先级最终发送到总线上。当邮箱的数据字节数目小于邮箱的DLC域值，CAN会附加一些常数字节0xCC以匹配期望的DLC值。

在一次成功的帧发送之后，CODE域将自动更新，并且TIMESTAMP域也将自动更新为内部计数器的值；CRC寄存器（CAN_CRCC寄存器和CAN_CRCCFD寄存器）将自动更新，CAN_STAT寄存器中相应的MSx位将置位，如果CAN_INTEN寄存器中相应的MIEx中断使能位置位了，则将产生一个中断。

仲裁过程

如果有多个发送邮箱处于挂起状态，则仲裁机制将会从最小邮箱编号到最大邮箱编号的方向进行搜索，找到最大优先级的邮箱进行发送。仲裁算法由CAN_CTL1寄存器的MTO位来控制选择。

当满足下列情况中任意一种，则开始一次仲裁过程：

- CAN总线上的CRC场：CRC场第一个位后，延迟ASD[4:0]（在CAN_CTL2寄存器中）个CAN位。
- CAN总线上错误界定符或者过载界定符。
- CAN总线从离线状态恢复：在TECNT[7:0]计数器计到124之后，延迟ASD[4:0]（在CAN_CTL2寄存器中）个CAN位。从离线状态恢复需要128次连续的11位隐性位，而这是由CAN_ERR0寄存器的TECNT[7:0]计数器来计数的。
- 退出暂停模式，或者退出省电模式（CAN_Disable模式和虚拟联网模式）。
- 重写仲裁获胜（暂时获胜或者最终获胜）邮箱MDES0字。
- 重写搜索过的（仲裁正在进行中）邮箱MDES0字：如果搜索完毕之后没有找到获胜邮箱，则仲裁将马上重新开始；否则，仲裁过程结束。
- 写任意邮箱的MDES0字：如果没有仲裁正在进行，并且没有仲裁获胜邮箱存在，同时CAN总线不在数据帧/远程帧的SOF-DATA / SOF-Control或者错误帧/过载帧的错误标志/过载标志，则开始仲裁过程。
- CAN节点进入总线集成状态（参考*总线集成状态*）：进入该状态后，延迟ASD[4:0]（在

CAN_CTL2寄存器中）个CAN位。

当满足下列情况中任意一种，则停止仲裁过程：

- 所有邮箱都被搜索过。
- 当CAN_CTL1寄存器MTO位置位，最小邮箱编号优先时，找到了一个激活的发送邮箱。
- CAN总线上错误标志或者过载标志。
- CAN总线上下一帧的SOF。
- 当发出进入暂停模式，CAN_Disable模式或者虚拟联网模式请求。

最小邮箱编号优先

如果CAN_CTL1寄存器的MTO位置位，则最小邮箱编号优先发送，此时CAN_CTL0寄存器的LAPRIOEN位不起作用。

最高优先级优先

如果CAN_CTL1寄存器的MTO位清零，则最高优先级的邮箱优先发送。最高优先级的发送邮箱在所有发送邮箱中具有最小的仲裁值（参考*表23-7. 当本地优先级禁用时的邮箱仲裁值（32位）*和*表23-8. 当本地优先级使能时的邮箱仲裁值（35位）*）。如果有超过一个邮箱具有相等的仲裁值，则更小邮箱编号的邮箱为仲裁获胜者。

当CAN_CTL0寄存器的LAPRIOEN位清零，本地优先级禁用时，参与到仲裁过程的位都将最终被发送到CAN总线上，如*表23-7. 当本地优先级禁用时的邮箱仲裁值（32位）*所示。

当CAN_CTL0寄存器的LAPRIOEN位置位，本地优先级使能时，则邮箱PRIO域将参与到内部仲裁过程。如*表23-8. 当本地优先级使能时的邮箱仲裁值（35位）*所示，邮箱PRIO域为仲裁值的最高有效位部分，因此具有低PRIO域值的邮箱比高PRIO域值的邮箱具有更高的优先级，忽略剩余的仲裁值，但PRIO域不会最终发送到CAN总线上。

表 23-7. 当本地优先级禁用时的邮箱仲裁值（32 位）

IDE	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
0	ID_STD[10:0]	RTR	IDE	保留	
1	ID_EXD[28:18]	SRR	IDE	ID_EXD[17:0]	RTR

表 23-8. 当本地优先级使能时的邮箱仲裁值（35 位）

IDE	34 33 32	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
0	PRIO[2:0]	ID_STD[10:0]	RTR	IDE	保留	
1	PRIO[2:0]	ID_EXD[28:18]	SRR	IDE	ID_EXD[17:0]	RTR

仲裁启动延迟

仲裁启动延迟由CAN_CTL2寄存器的ASD[4:0]位域来配置，用于优化当仲裁过程结束的太早，

可能导致仲裁获胜的发送邮箱被CPU重写，从而导致仲裁过程被重启，因而不能及时地发送出去的过程。

移出

移出过程是在找到仲裁获胜邮箱后，将获胜发送邮箱中的内容拷贝到发送移位缓存（一个内部邮箱描述符）的过程。发送移位缓存中的消息将按照CAN协议规则进行发送。

当移出过程完成后，即使CAN_CTL0寄存器的MST位置位，对相应发送邮箱的MDES0字的写操作都将被阻塞。当符合下述中的一种情形时，将恢复对相应发送邮箱的MDES0字的写操作：

- 在邮箱发送完毕，并且CAN_STAT寄存器中相应的标志位MSx被清零。
- CAN节点进入暂停模式或者离线状态。
- CAN节点在总线仲裁中失利，或者在发送过程中发生了一个错误。

当符合下述中的一种情形时，将启动移出过程：

- CAN总线上帧间隔的第一个位。
- 处于总线空闲状态。
- 处于等待总线空闲状态。

在移出过程中，CPU在总线空闲状态可优先访问相应的内存，移出操作对相应的内存具有较低的访问权限。

中止

为了请求发送中止，推荐的操作为首先置位CAN_CTL0寄存器的MST位，然后对邮箱的CODE域写ABORT（0b1001）。

如果邮箱不是仲裁获胜邮箱，或者邮箱是仲裁获胜邮箱，但还未完成移出过程，则对该邮箱MDES0字写ABORT（0b1001）的操作可以成功，CAN_STAT寄存器中对应的MSx位将置位。

如果邮箱是仲裁获胜邮箱，且移出过程已经完成，或者邮箱正在发送，则对邮箱MDES0字写ABORT（0b1001）的操作将被阻塞。在这种情况下，发送中止请求会被保存并保持挂起，直到帧被成功发送出去或者发送失败：

- 帧被成功发送，邮箱未被中止：如果帧最终发送成功，则挂起的中止请求会自动清除，CAN_STAT寄存器中对应的MSx位将置位，如果CAN_INTEN寄存器的MIEx位置位，则会触发一个中断。
- 发送失败，邮箱被中止：如果帧最终发送失败，则挂起的中止请求会收到应答信号，对邮箱的写操作将会恢复，邮箱的MDES0字被改写为ABORT，CAN_STAT寄存器中对应的MSx位将置位，如果CAN_INTEN寄存器的MIEx位置位，则会触发一个中断。
  当符合下述中的一种情形时，帧发送失败：
  - 总线仲裁失利。
  - 发送过程中发生一个错误。
  - 进入离线状态。
  - 总线有一个过载帧。

发送邮箱失活

发送邮箱失活的操作：

- 对发送邮箱的MDES0字CODE域写ABORT。这是推荐的邮箱失活操作，不会造成不可知的发送。
  该操作必须首先确保CAN_CTL0寄存器的MST位置位。

23.3.6. 数据接收

对于CAN常规帧，支持通过FIFO和邮箱来接收帧。

对于CAN FD帧，仅支持通过邮箱来接收帧。

邮箱接收

对于邮箱接收，只有当帧的标识符与邮箱标识符域中配置的ID（或者当使用了过滤寄存器时，是一组邮箱ID）相匹配时，才会将帧接收存储到邮箱中。

为了将CAN帧接收到邮箱中去，必须按如下步骤准备一个接收邮箱：

1. 如果邮箱是激活状态（不论是发送邮箱还是接收邮箱），则按*发送邮箱失活*或*接收邮箱失活*来失活该邮箱。当执行了发送邮箱失活操作，则按后续的步骤继续操作。如果执行了接收邮箱失活操作，跳到步骤4。如果邮箱是失活状态（不论是发送邮箱还是接收邮箱），则跳到步骤4。
2. 轮询CAN_STAT寄存器，等待相应MSx位置位，或者置位CAN_INTEN寄存器中相应的MIEx位使能相应的中断，通过中断请求处理。
3. 读回CODE域来确保邮箱状态是已中止，还是已发送。
4. 清零CAN_STAT寄存器的相应标志位MSx位。
5. 写邮箱MDES1字的标识符域，如果需要，配置MDES0字的IDE，RTR域。
6. 设置邮箱MDES0字CODE域为EMPTY（0b0100）来激活邮箱。

在一次成功的接收之后，邮箱描述符所有位（DATA，ID，TIMESTAMP，SRR，IDE，RTR，FDF，BRS，ESI，DLC，CODE）都存储为总线上接收到的相应位或者进行了自动更新，CAN_STAT寄存器的相应标志位MSx位置位，如果CAN_INTEN寄存器中相应的MIEx中断使能位置位了，则将产生一个中断。TIMESTAMP域将自动更新为帧的标识符域第二位时刻的内部计数器的值。

为了服务（读）接收邮箱，推荐的操作步骤如下所示：

1. 轮询CAN_STAT寄存器，等待相应MSx位置位，或者置位CAN_INTEN寄存器中相应的MIEx位使能相应的中断，通过中断请求处理。
2. 读邮箱MDES0字，轮询CODE域BUSY位，等待其清零。当BUSY位为0时，读邮箱操作将会锁定邮箱，而使邮箱不会被改写。
3. 读邮箱内容。
4. 清零CAN_STAT寄存器的相应标志位MSx位。
5. 读取CAN_TIMER寄存器来解锁邮箱。

邮箱锁定

锁定机制仅适用于接收邮箱：对于CODE域为接收FULL或者接收OVERRUN的邮箱，CPU对邮箱MDES0字的读操作将会锁定该邮箱，从而阻止新的一个匹配报文对邮箱内容进行改写。

通过读CAN_TIMER寄存器(全局解锁操作)或者对其他邮箱MDES0字的读操作可以解锁邮箱。当邮箱被解锁后，如果有未处理的报文，则将开始一个移入过程（在暂停模式下具有相同解锁功能，而当CAN_CTL0寄存器的LPS位置位时解锁邮箱，将要等到LPS位清零才会开始一个移入过程）。

如果邮箱没有及时地解锁，而又接收到一个新的匹配报文，则新的报文将会覆盖接收移位缓存，并且邮箱CODE不会有报文丢失的提示，也没有相应的错误状态的记录。

注意：邮箱失活（对邮箱CODE写接收INACTIVE或者发送ABORT）相比于邮箱锁定具有更高的优先级。

接收邮箱失活

失活接收邮箱的方式：

- 对接收邮箱MDES0字CODE域写INACTIVE（接收INACTIVE或者发送INACTIVE）。但这个操作可能会导致一个新的匹配报文的丢失且没有相应提示。

注意：接收邮箱失活操作将会自动解锁该邮箱。接收FIFO没有相应的锁定写保护机制。

Rx FIFO 接收

Rx FIFO深度为6帧。当CAN_CTL0寄存器的RFEN位置位时，使能Rx FIFO用于帧接收。Rx FIFO只能用于接收，且不能在CAN FD模式使能的时候使用。Rx FIFO描述符参考*表23-6. 接收FIFO描述符*。CAN过滤系统提供了对一组标识符的过滤功能，有效地降低中断服务的负担。Rx FIFO过滤器的数目可通过CAN_CTL2寄存器的RFFN[3:0]位域来配置，最大支持32个过滤器，对应的过滤器相关参数可通过CAN_RFIFOMPFx（x = 0..31）寄存器（如果CAN_CTL0寄存器的RPFQEN位置位），或者CAN_RFIFOPUBF和CAN_RFIFOMPFx（x = 0..31）寄存器（如果CAN_CTL0寄存器的RPFQEN位清零）来配置。

Rx FIFO有未读消息时：如果CAN_STAT寄存器的MS5_RFNE位置位，则可通过FDES0-FDES3字来读取接收到的消息。当CAN_STAT寄存器的MS5_RFNE位置位，意味着Rx FIFO中至少有一个可读的消息。如果CAN_INTEN寄存器相应的中断使能位MIEx置位，则将产生一个中断；如果CAN_CTL0寄存器的DMAEN位置位，MS5_RFNE位将会产生一个DMA传输请求，而不会产生Rx FIFO中断。

- 通过CPU方式服务（读取）Rx FIFO，推荐按如下步骤操作：

  1. 轮询CAN_STAT寄存器，直到MS5_RFNE标志置位，或者置位CAN_INTEN寄存器中MIE5位使能中断，通过中断请求处理。
  2. 读取Rx FIFO的FDES0-FDES3字，并按需要来读取CAN_RFIFOIFMN寄存器。
  3. 清除CAN_STAT寄存器的MS5_RFNE标志位。如果Rx FIFO中包含多个消息，则对MS5_RFNE标志位的清除操作会将Rx FIFO的FDES0-FDES3字更新为下一个消息，而CAN_RFIFOIFMN寄存器也在同时更新，MS5_RFNE标志位仍然保持置位，如果使能了中断，则会又产生一个中断，重复步骤2-3来读取接收到的消息。

- 通过DMA方式服务（读取）Rx FIFO，推荐按如下步骤操作：

  1. 配置DMA控制器并使能相应通道用于Rx FIFO消息接收。
  2. 通过CPU方式服务（读取）Rx FIFO，直到CAN_STAT寄存器的MS5_RFNE标志被清零，以避免在DMA使能后有额外的DMA请求产生。
  3. 使能CAN_CTL0寄存器的DMAEN位来使能DMA请求。
  4. 等待DMA请求。当CAN_STAT寄存器的MS5_RFNE标志位置位时将产生一个DMA请求。
  5. 在接收到DMA请求后，DMA控制器将会自动读取Rx FIFO的FDES0-FDES3字。必须读取FDES3字才能清除CAN_STAT寄存器的MS5_RFNE标志位，如果Rx FIFO中包含多个消息，读FDES3字的操作会使Rx FIFO的FDES0-FDES3字更新为下一个消息，而CAN_RFIFOIFMN寄存器（需要在读FDES3字之前读取）也会同时更新，MS5_RFNE标志位仍然保持置位，并再次产生一个DMA请求。重复步骤4-5。

DMA 模式

当CAN_CTL0寄存器的RFEN位和DMAEN位都置位时，可使用DMA模式来处理Rx FIFO接收。当使能了DMA模式时，就不能再使用CPU方式来读取Rx FIFO。

当使能了DMA模式时，如果Rx FIFO中有未读消息，DMA控制器将会自动读取Rx FIFO的FDES0-FDES3字来读取接收的消息。在这种模式下，CAN_STAT寄存器中的Rx FIFO警告标志位MS6_RFW和Rx FIFO溢出标志位MS7_RFO都用作保留位。

在通过清零CAN_CTL0寄存器的DMAEN位来禁能DMA模式之前，必须执行一个清FIFO内容的操作（当CAN_CTL0寄存器的RFEN位置位时，在暂停模式下对CAN_STAT寄存器的MS0位写1）。清FIFO的操作将会清除CAN_STAT寄存器的MS5_RFNE位，并取消DMA请求。

清 FIFO

当Rx FIFO使能（CAN_CTL0寄存器的RFEN位置位）后，通过在暂停模式下对CAN_STAT寄存器的MS0位写1来清除Rx FIFO的内容，但Rx FIFO的标志位不会被清除（DMA模式下除外）。因此在清FIFO操作之前，需要通过读取Rx FIFO直到将CAN_STAT寄存器的MS5_RFNE标志位清零。

标志

Rx FIFO非空

当CAN_STAT寄存器的MS5_RFNE位置位时，表示Rx FIFO中至少有一个可读消息。

Rx FIFO警告

当CAN_STAT寄存器的MS6_RFW位置位时，表示Rx FIFO又接收到了一条消息，未读消息从4个增加到了5个，FIFO即将满了。

Rx FIFO溢出

当CAN_STAT寄存器的MS7_RFO位置位时，表示Rx FIFO又接收到了一条消息，然后由于FIFO已满，因而有一个消息丢失了。

匹配过程

匹配过程是通过搜索查找与CAN总线上帧标识符相匹配的接收邮箱或接收FIFO（如果使能了FIFO）来完成，IDE域和RTR域也参与匹配过程。

当完成DLC字段的接收，则开始匹配过程。

搜索过程

- 如果使能了Rx FIFO，则CAN_CTL2寄存器的RFO位控制了搜索顺序。
  - 如果RFO位置位，则匹配过程从接收邮箱开始搜索，然后再搜索Rx FIFO。接收邮箱从邮箱编号低到高的方向进行搜索。
    首先，搜索匹配的可用于接收的邮箱。如果RPFQEN位为0，则第一个匹配到的邮箱就是获胜者，无论其是否为空或非空状态。如果RPFQEN位为1，则第一个匹配到的空邮箱就是匹配获胜者。两种情况下均不再搜索Rx FIFO。
    然后，如果RPFQEN位为1时，没有匹配到空邮箱，但找到了一个匹配的非空邮箱，则还要搜索Rx FIFO来确定匹配获胜者：如果找到了匹配的Rx FIFO并且FIFO未满，则Rx FIFO就是匹配获胜者；否则，最后一个找到的匹配的可用于接收的非空邮箱就是匹配获胜者（会导致邮箱CODE码OVERRUN）。
    最后，如果没有找到匹配的接收邮箱（即没有匹配的可用于接收的空邮箱，也没有匹配的可用于接收的非空邮箱），则搜索Rx FIFO。在这种情况下，如果Rx FIFO是匹配的但是FIFO满了，将会导致Rx FIFO溢出；如果Rx FIFO不匹配（不管FIFO是否是满的），则消息不会被接收进来。
  - 如果RFO位清零，则匹配过程从Rx FIFO开始搜索，然后再搜索接收邮箱。
    如果Rx FIFO是匹配的且FIFO未满，则Rx FIFO就是匹配获胜者。
    如果Rx FIFO不匹配或者FIFO满了，则还要搜索接收邮箱。邮箱的匹配受到RPFQEN位的影响，如果RPFQEN位为0，则第一个匹配到的邮箱就是匹配获胜者，无论其是否为空或非空状态。如果RPFQEN位为1，则第一个匹配到的空邮箱就是匹配获胜者，如果没有搜索到空邮箱，则最后一个非空状态的匹配邮箱为获胜者。
- 如果禁能了Rx FIFO，则匹配过程只搜索接收邮箱，参考前述的邮箱匹配描述。

可用于接收的空邮箱有以下两种情形：

- 对于数据帧的接收，或者当CAN_CTL2寄存器的RRFRMS位为1时的远程帧接收，可用于接收的空邮箱为：邮箱CODE域为EMPTY；邮箱CODE域为FULL或者OVERRUN，同时已经服务（读）过并解锁的。
- 对于当CAN_CTL2寄存器的RRFRMS位为0时的远程帧接收，可用于接收的空邮箱为：邮箱CODE域为RANSWER。

接收邮箱的搜索匹配条件

对接收邮箱的搜索匹配条件，参考*表23-9. 接收邮箱匹配*:

- 如果接收移位缓存中是一个数据帧（即RTR域为0），则将搜索CODE为EMPTY，FULL或者OVERRUN的接收邮箱：
  - 如果CAN_CTL2寄存器的IDERTR_RMF位为0，表示需要匹配IDE域，不用匹配RTR域（忽略相关过滤寄存器的位30和位31）。ID域需要使用相关过滤寄存器的位0到位

28过滤数据配置来进行过滤匹配。

- 如果CAN_CTL2寄存器的IDERTR_RMF位为1，表示IDE，RTR和ID域都需要分别使用相关过滤寄存器的位30，位31和位0到位28过滤数据配置来进行过滤匹配。

■ 如果接收移位缓存中是一个远程帧（即RTR域为1）：

- 如果CAN_CTL2寄存器中的RRFRMS位为0，表示将要查找CODE为RANSWER的接收邮箱，并且IDE，和ID域都需要分别使用相关过滤寄存器的位30，和位0到位28过滤数据配置来进行过滤匹配。
- 如果CAN_CTL2寄存器中的RRFRMS位为1，则搜索匹配过程与数据帧相同，将搜索CODE为EMPTY，FULL或者OVERRUN的接收邮箱：

  如果CAN_CTL2寄存器的IDERTR_RMF位为0，表示需要匹配IDE域，不用匹配RTR域（忽略相关过滤寄存器的位30和位31）。ID域需要使用相关过滤寄存器的位0到位28过滤数据配置来进行过滤匹配。

  如果CAN_CTL2寄存器的IDERTR_RMF位为1，表示IDE，RTR和ID域都需要分别使用相关过滤寄存器的位30，位31和位0到位28过滤数据配置来进行过滤匹配。

表 23-9. 接收邮箱匹配

接收到的位	配置位		邮箱描述符中用于匹配的域			
RTR	**IDERTR_RMF（在CAN_CTL2寄存器）**	**RRFRMS（在CAN_CTL2寄存器）**	**IDE**	**RTR**	**ID**	**CODE**
0	0	-	匹配(1)	从不(2)	过滤匹配(3)	EMPTY / FULL / OVERRUN
	1		过滤匹配			EMPTY / FULL / OVERRUN
1	-	0	匹配	从不	匹配	RANSWER
	0	1	匹配	从不	过滤匹配	EMPTY / FULL / OVERRUN
	1		过滤匹配			EMPTY / FULL / OVERRUN

1. 匹配：邮箱描述符中的域始终需要与接收到的位进行匹配比较，忽略相关过滤寄存器中的过滤数据配置。
2. 从不：邮箱描述符中的域始终不与接收到的位进行匹配比较，忽略相关过滤寄存器中的过滤数据配置。
3. 过滤匹配：邮箱描述符中的域需要使用相关过滤寄存器中的过滤数据配置，与接收到的位进行匹配比较。

Rx FIFO 的搜索匹配条件

对Rx FIFO的搜索匹配条件，参考*表23-10. Rx FIFO匹配*:

■ 如果CAN_CTL0寄存器的FS[1:0]位域值为0或者1，表示标识符过滤元素格式采用格式A或者格式B，并且IDE，RTR和ID域都需要使用相关过滤寄存器的位0到位31过滤数据配置来进行过滤匹配。

■ 如果CAN_CTL0寄存器的FS[1:0]位域值为2，表示标识符过滤元素格式采用格式C，并且IDE，RTR域不进行匹配比较（FIFO描述符中没有这些位域），ID域需要使用相关过滤寄

存器的位0到位31过滤数据配置来进行过滤匹配。

- 如果CAN_CTL0寄存器的FS[1:0]位域值为3，表示标识符过滤元素格式采用格式D，不接受所有帧。

表 23-10. Rx FIFO 匹配

配置位	Rx FIFO描述符中用于匹配的域		
FS[1:0]（在CAN_CTL0寄存器）	IDE	RTR	ID
0	过滤匹配		
1	过滤匹配		
2	从不		过滤匹配
3	不匹配的(1)		

1. 不匹配的：拒绝接收所有的帧。

移入

移入过程是在找到匹配的接收邮箱或者Rx FIFO之后，将接收移位缓存（一个内部描述符）中的内容拷贝到接收邮箱或者Rx FIFO的过程。

当找到匹配的接收邮箱或者Rx FIFO时，将挂起一个移入操作。当符合下述所有条件时，将开始移入操作：

- 接收移位缓存中的帧有找到匹配的获胜邮箱或Rx FIFO。
- CAN总线处于：
  - 帧间隔第二个位。
  - 过载帧的第一个位。
- 目标邮箱未被锁定。

如果目标邮箱有一个挂起的移入操作，而邮箱在暂停模式下解锁了，则开始移入操作；如果邮箱在CAN_CTL0寄存器LPS位为1时解锁了，挂起的移入操作将等到LPS位清0时才会开始。

当接收邮箱上正在进行一个移位过程，目标邮箱的的BUSY位（CODE[0]）将置位用于指示当前邮箱正在更新。

接收邮箱上的移入操作可以被取消，而Rx FIFO上的移入操作无法被取消。当符合下述中的一种情形时，接收邮箱的移入操作将被取消：

- 当CAN总线在到达接收移位缓存中存储的帧之后的帧间隔第一个位之后，目标邮箱被失活了，并且已经完成匹配过程。
- 接收移位缓存中存储了一帧CAN节点自己发送的帧，而CAN_CTL0寄存器中SRDIS位为1，禁能了自接收功能。
- 发生了一个CAN协议错误。

当完成了移入操作，接收邮箱描述符或者Rx FIFO描述符（如果使能了Rx FIFO）将更新为接收到的帧，如果是移入到Rx FIFO，则CAN_RFIFOIFMN寄存器也会更新，如果是移入到接收邮箱，则接收邮箱描述符的CODE域也会更新。

过滤数据配置

当禁能Rx FIFO时：

- 如果CAN_CTL0寄存器的RPFQEN位为0，则使用CAN_RMPUBF寄存器来配置所有接收邮箱的过滤数据配置。
- 如果CAN_CTL0寄存器的RPFQEN位为1，则使用CAN_RFIFOMPFx（x = 0..31）寄存器来分别配置接收邮箱的过滤数据配置。

当使能Rx FIFO时：

- 如果CAN_CTL0寄存器的RPFQEN位为0，则使用CAN_RMPUBF寄存器来配置所有接收邮箱的过滤数据配置，使用CAN_RFIFOPUBF和CAN_RFIFOMPFx（x = 0..31）寄存器来配置所有Rx FIFO标识符过滤表元素，并且所有这些寄存器的值的配置必须相同。
- 如果CAN_CTL0寄存器的RPFQEN位为1，则使用CAN_RFIFOMPFx（x=0..31）寄存器来配置由CAN_CTL2寄存器RFFN[3:0]位域设置的Rx FIFO标识符过滤表元素以及接收邮箱（由于接收邮箱描述符和Rx FIFO描述符不能同时占用同一个区域的RAM，因此用一组寄存器进行独立控制过滤数据的配置），由CAN_RFIFOPUBF寄存器来配置剩余所有的Rx FIFO标识符过滤表元素。

自接收

当CAN_CTL0寄存器的SRDIS位置位时，自接收功能被禁止，从而不接收所有由本节点发送的帧，即使已经找到了相匹配的接收邮箱或者Rx FIFO，并且不会有任何的标志或者中断产生。当SRDIS位清零时，允许将本节点发送的帧接收到相匹配的描述符中去。

23.3.7. 在虚拟联网模式下的数据接收

当设置CAN_CTL0寄存器的PNEN位和PNMOD位为1时，使能虚拟联网模式，CAN模块可以在MCU睡眠模式下接收帧。一个唤醒事件可以将CAN模块从虚拟联网模式唤醒。

有四组寄存器用于匹配的消息存储：CAN_PN_RWMxCS，CAN_PN_RWMxI，CAN_PN_RWMxD0和CAN_PN_RWMxD1寄存器，组号x从0到3。因此最多可以存储4帧消息（当CAN_PN_CTL0寄存器的NMM[7:0]位域值大于等于4时），并且只存储最新的消息。组号x表示消息到达的顺序。如果NMM[7:0]位域值小于4，则只存储NMM[7:0]个消息，存放在组号0到NMM[7:0]减1的寄存器组中。

如果要存储的消息的数据长度小于8个字节，则在接收到的DATA域后填充若干常数0字节到CAN_PN_RWMxD0和CAN_PN_RWMxD1（x = 0..3）寄存器中。对于匹配的唤醒帧不存储时间戳值。

注意：当处于虚拟联网模式时将忽略CAN FD格式的消息帧。

唤醒中断

有两种类型的唤醒中断事件，包括匹配唤醒事件，和超时唤醒事件。每个中断事件在CAN_PN_STAT寄存器中都有专门的标志位，在CAN_PN_CTL0寄存器组中有专门的使能位。它们的关系如*表23-11. 中断事件*所示。

当任意一种唤醒中断被使能，并且发生了相应的事件，则会产生一个唤醒中断。

超时唤醒事件

当CAN达到了超时事件，则发生一个超时唤醒事件。超时时间由CAN_PN_TO的WTO[15:0]位域来配置。

注意：即使到达了超时时间，在CPU真正唤醒之前CAN模块仍然不会停止消息的接收过滤。

匹配唤醒事件

当CAN在超时时间之内接收到了一个或一组匹配的唤醒帧，则发生一个匹配唤醒事件。CAN_PN_STAT 寄存器的MMCNT[7:0]位域指示了从进入虚拟联网模式开始到CPU被唤醒的时间内所接收到的所有匹配帧的数目。

注意：即使CAN接收到了一个或一组匹配的唤醒帧，在CPU真正唤醒之前超时计数器不会停止计数。

帧匹配

参与唤醒匹配过程的帧域有IDE，RTR，ID，DLC和DATA域。

- 如果CAN_PN_CTL0寄存器的FFT[1:0]位域配置为0，则当接收到一个帧除了DATA，DLC域之外的其他域（即IDE，RTR和ID域）都匹配时，发生一个匹配唤醒事件。
- 如果CAN_PN_CTL0寄存器的FFT[1:0]位域配置为1，则当接收到一个帧所有域（即IDE，RTR，ID，DLC和DATA域）都匹配时，发生一个匹配唤醒事件。
- 如果CAN_PN_CTL0寄存器的FFT[1:0]位域配置为2，则当接收到一组帧（帧数量由CAN_PN_CTL0寄存器的NMM[7:0]位域来配置）除了DATA，DLC域之外的其他域（即IDE，RTR和ID域）都匹配时，发生一个匹配唤醒事件。
- 如果CAN_PN_CTL0寄存器的FFT[1:0]位域配置为3，则当接收到一组帧（帧数量由CAN_PN_CTL0寄存器的NMM[7:0]位域来配置）所有域（即IDE，RTR，ID，DLC和DATA域）都匹配时，发生一个匹配唤醒事件。

IDE 域匹配

一个匹配的IDE域是使用CAN_PN_IFEID1寄存器中的过滤数据配置时，接收的帧IDE域与CAN_PN_EID0寄存器中配置的期望IDE域一致。

RTR 域匹配

一个匹配的RTR域是使用CAN_PN_IFEID1寄存器中的过滤数据配置时，接收的帧RTR域与CAN_PN_EID0寄存器中配置的期望RTR域一致。

ID 域匹配

- 当CAN_PN_CTL0寄存器的IDFT[1:0]位域配置为0，则一个匹配的ID域是使用CAN_PN_IFEID1寄存器中的过滤数据配置时，接收的帧ID域与CAN_PN_EID0寄存器中配置的期望ID域一致。
- 当CAN_PN_CTL0寄存器的IDFT[1:0]位域配置为1，则一个匹配的ID域是接收的帧ID域大于等于CAN_PN_EID0寄存器中配置的期望ID域。没有使用CAN_PN_IFEID1寄存器的

IDFD_EHT[28:0]位域。

- 当CAN_PN_CTL0寄存器的IDFT[1:0]位域配置为2，则一个匹配的ID域是接收的帧ID域小于等于CAN_PN_EID0寄存器中配置的期望ID域。没有使用CAN_PN_IFEID1寄存器的IDFD_EHT[28:0]位域。
- 当CAN_PN_CTL0寄存器的IDFT[1:0]位域配置为3，则一个匹配的ID域是接收的帧ID域大于等于CAN_PN_EID0寄存器中配置的期望ID域，并且小于等于CAN_PN_IFEID1寄存器中配置的期望ID域。

DLC域匹配

- 一个匹配的DLC域是接收的帧DLC域大于等于CAN_PN_EDLC寄存器中DLCELT[3:0]位域配置的期望DLC域下限值，并且小于等于CAN_PN_EDLC寄存器中DLCEHT[3:0]位域配置的期望DLC域上限值。

DATA域匹配

- 当CAN_PN_CTL0寄存器的DATAFT[1:0]位域配置为0，则一个匹配的DATA域是使用CAN_PN_DF0EDH0寄存器和CAN_PN_DF1EDH1寄存器中的过滤数据配置时，接收的帧DATA域与CAN_PN_EDLx（x = 0,1）寄存器中配置的期望DATA域一致。
- 当CAN_PN_CTL0寄存器的DATAFT[1:0]位域配置为1，则一个匹配的DATA域是接收的帧DATA域大于等于CAN_PN_EDLx（x = 0,1）寄存器中配置的期望DATA域。CAN_PN_DF0EDH0寄存器和CAN_PN_DF1EDH1寄存器保留不使用。
- 当CAN_PN_CTL0寄存器的DATAFT[1:0]位域配置为2，则一个匹配的DATA域是接收的帧DATA域小于等于CAN_PN_EDLx（x = 0,1）寄存器中配置的期望DATA域。CAN_PN_DF0EDH0寄存器和CAN_PN_DF1EDH1寄存器保留不使用。
- 当CAN_PN_CTL0寄存器的DATAFT[1:0]位域配置为3，则一个匹配的DATA域是接收的帧DATA域大于等于CAN_PN_EDLx（x = 0,1）寄存器中配置的期望DATA域，并且小于等于CAN_PN_DF0EDH0寄存器和CAN_PN_DF1EDH1寄存器中配置的期望DATA域。

  注意：在这种情况下，这两个8字节的期望数据寄存器都需要配置，当接收到的帧DLC域小于8个字节（DLC域已匹配），则在DATA域匹配时，是将接收的帧DATA域加上若干常数0填充字节，再与期望的DATA域进行比较。

23.3.8. CAN FD 操作

通过配置CAN_CTL2寄存器的ISO位，可以选择CAN FD功能支持ISO CAN FD（ISO11898-1规范）或非ISO CAN FD（Bosch CAN FD规范V1.0），这两种规范彼此不兼容。相比于非ISO CAN FD协议，ISO CAN FD协议引入了一个3位的计数器和一个奇偶校验位，因此错误检测能力有所提升。

CAN FD模式同时支持CAN常规帧和CAN FD帧的收发。FDF位（在常规帧中该位为保留位）用于区分当前帧是FD帧还是常规帧。当FDF位为隐性‘1’，表示是CAN FD帧；如果为显性，表示是常规帧。相比于常规帧，CAN FD帧不支持Rx FIFO，不支持Rx FIFO DMA功能，也不支持虚拟联网模式。

通过将CAN_CTL0寄存器的FDEN位置位，可以使能CAN FD模式。

CAN FD BRS

在CAN FD模式下，最多可以支持64字节数据，当BRS位为隐性时，波特率在数据阶段（从BRS位到CRC界定符的第一个采样点，或者当发生错误时到错误帧的SOF）可达到最大8 Mbit/s，详情请参考ISO11898-1或Bosch CAN FD规范V1.0。

当设置CAN_FDCTL寄存器的BRSEN位为1（在下一帧起作用），并且发送邮箱的BRS位配置为隐性位‘1’时，在CAN FD帧的数据阶段将使用更高波特率（称为数据波特率），其他位使用正常波特率来通信。波特率将在BRS位的采样点进行切换。数据波特率由CAN_FDBT寄存器来配置，正常波特率由CAN_BT寄存器来配置。

当设置CAN_FDCTL寄存器的BRSEN位为0，或者发送邮箱的BRS位配置为显性位‘0’，则在整个CAN FD帧传输期间都使用正常波特率。

注意：整个CAN FD帧的时间单元的大小应保持一致，以避免帧在通信过程中总线发生相位错误。

对于FD帧，所有节点都需要接收2位长的显性ACK应答字段作为一个有效的ACK，用以补偿与接收节点之间的相位偏移。详情请参考ISO11898-1规范。

CAN FD ESI

由发送邮箱的MDES0字的ESI域，以及CAN_ERR1寄存器的ERRSI[1:0]位域来控制ESI位（在DLC域之前的位，请参考ISO11898-1或Bosch CAN FD规范V1.0）的发送。如果MDES0字中的ESI域为0，则根据CAN_ERR1寄存器的ERRSI[1:0]位域，主动错误节点发送为显性位，被动错误节点发送为隐性位。如果MDES0字中的ESI域为1，则节点发送MSED0字中的ESI域值。

CAN FD CRC

不同帧格式使用不同的 CRC 多项式，汉明距离都为 6：

- 多项式 CRC_15 用于常规帧：0xC599
  $x^{15} + x^{14} + x^{10} + x^{8} + x^{7} + x^{4} + x^{3} + 1$
- 多项式 CRC_17 用于不超过 16 字节 DATA 域的 CAN FD 帧：0x3685B
  $x^{17} + x^{16} + x^{14} + x^{13} + x^{11} + x^{6} + x^{4} + x^{3} + x^{1} + 1$
- 多项式 CRC_21 用于超过 16 字节 DATA 域的 CAN FD 帧：0x302899
  $x^{21} + x^{20} + x^{13} + x^{11} + x^{7} + x^{4} + x^{3} + 1$

对于发送，将在帧SOF时同时使用这三种CRC多项式进行CRC计算，最终发送的CRC由帧的FDF域和DLC域来确定。在成功发送帧后，当CAN_STAT寄存器的MSx位置位时，CAN_CRCCFD寄存器将同时更新为发送消息的CRC计算结果。CAN_CRCCFD同时用于FD帧和非FD帧。CAN_CRCC寄存器只存储常规帧的CRC计算结果。

对于接收，用于CRC校验的CRC多项式由接收到的FDF域和DLC域来确定。

注意：在常规帧中，CRC界定符为单个隐性位。在FD帧中，CRC界定符可能包含一到两个隐性位。发送节点应只发送一个隐性位作为CRC界定符，但接收时应在ACK应答位前的隐性位到显性位边沿到来之前接收2个隐性位。接收节点应在第一个CRC界定符之后发送ACK位。详情

请参考ISO11898-1规范。

位填充

CAN FD帧的位填充功能不同于常规帧的位填充功能。

对于CAN FD帧的发送，将会在CRC场第一个位（忽略其他位填充条件）之前插入一个固定的填充位，另外在CRC场每4位（不包括固定的填充位）后都将插入一个固定的填充位。这些固定填充位的值都是它们前面的位的取反值。请参考ISO11898-1规范。

对于CAN FD帧的接收，将忽略这些固定的填充位。如果发现固定填充位的值与它前面位的值相同，则发生一个位填充错误。

注意：对于CAN FD帧，这些固定的填充位都将参与到CRC计算。对于常规帧，填充位不参与CRC计算。

再同步

CAN FD帧和常规帧的再同步以及硬件同步机制是相同的。在FD帧的数据阶段不执行再同步。

传输延迟补偿

当CAN FD帧的BRS域为隐性位时，发送CAN FD帧的数据阶段的位时间长度小于CAN收发器内部回路延迟的限定值，因此使用传输延迟补偿机制来避免当采样点到来时发送节点还没有收到自己发出的位，从而报位错误的情况发生。对CAN收发器内部回路延迟的测量是从发送的FDF位下降沿到接收的FDF位的下降沿，如图*图23-2. 传输延迟*所示。

传输延迟补偿机制定义了次级采样点SSP。当应用了传输延迟补偿，则发送节点应忽略在采样点检测到的位错误。当配置CAN_FDCTL寄存器的TDCEN位为1，使能了传输延迟补偿机制，则位检查将在真正接收到的位与延迟了（这个延迟的计算是基于收发器内部回路的延迟）的发送位之间进行比较。

传输延迟补偿值按下述公式进行计算：

$$t_{compensation} = t_{measure} + t_{offset} \quad (23\text{-}1)$$

其中：

$$t_{offset} = TDCO[4:0] \times t_{CANCLK} \quad (23\text{-}2)$$

$$t'_{offset} = t_{PBS1_FD} + t_{PTS_FD} + t_{SYNC_SEG} \quad (23\text{-}3)$$

$$t_{PBS1_FD} = (DPBS1[2:0] + 1) \times t_{q_FD} \quad (23\text{-}4)$$

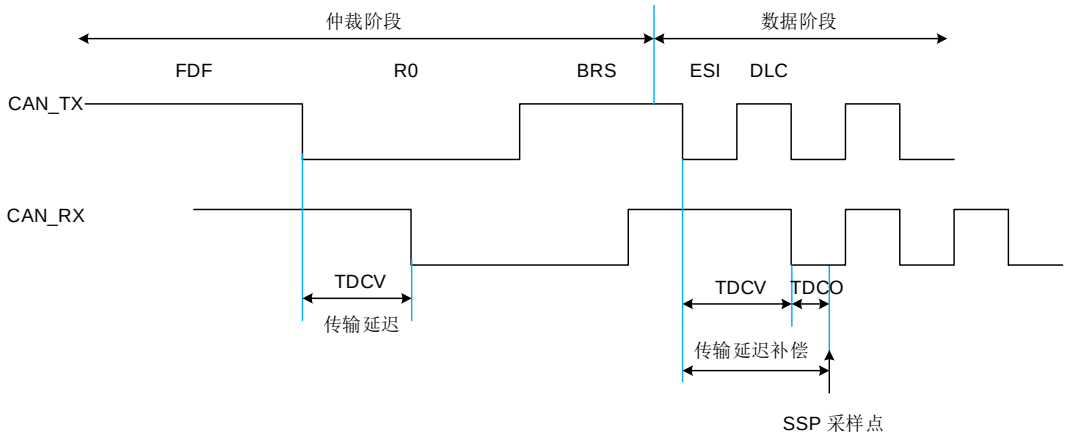
$$t_{PTS_FD} = DPTS[4:0] \times t_{q_FD} \quad (23\text{-}5)$$

$$t_{q_FD} = (DBAUDPSC[9:0] + 1) \times t_{CANCLK} \quad (23\text{-}6)$$

上述公式中$t_{measure}$是测量的传输延迟；t_{offset}是传输延迟补偿偏置，存储在CAN_FDCTL寄存器的TDCO[4:0]位域中，以t_{CANCLK}为单位存储，t_{offset}不可大于CAN数据阶段的位时间；t'_{offset}是传输延迟补偿偏置的理论值，用户可以根据t'_{offset}来设置t_{offset}。 $t_{compensation}$是传输延迟补偿值，保存在CAN_FDCTL寄存器的TDCV[5:0]位域中，以t_{CANCLK}为单位存储。

在上述公式中，DPBS1[2:0]，DPTS[4:0]，和DBAUDPSC[9:0]位域都在CAN_FDBT寄存器中配置。

图 23-2. 传输延迟



$t_{compensation}$最大值为(3 × data bit time - 2 × t_{q_FD})。如果超过这个值，就无法补偿这个传输延迟了，从而CAN_FDCTL寄存器中TDCS位将置位。传输延迟补偿应至少补偿2个数据阶段位时间长度。

23.3.9. 错误和状态

发送错误计数器（CAN_ERR0寄存器中的TECNT[7:0]位域）和接收错误计数器（CAN_ERR0寄存器中的RECNT[7:0]位域）将FD帧和非FD帧的错误都进行了统计，在错误条件触发时增加或减少相应的计数。关于TECNT[7:0]和RECNT[7:0]错误计数管理的详细信息请参考CAN协议相关章节。

对于CAN FD帧，数据阶段的发送错误计数器（CAN_ERR0寄存器的TEFCNT[7:0]位域）和数据阶段的接收错误计数器（CAN_ERR0寄存器的REFCNT[7:0]位域）只有在帧BRS域为隐性位时才起作用。这些错误计数器在离线状态停止计数并保持计数值，直到离线状态恢复为主动错误状态才重新从0开始计数。

注意：在虚拟联网模式下，接收错误计数器RECNT[7:0]和数据阶段的接收错误计数器REFCNT[7:0]都继续计数，并且保存相应的错误标志，发送错误计数器TECNT[7:0]和数据阶段的发送错误计数器TEFCNT[7:0]停止计数并保持计数值。当返回正常模式时，CAN_ERR0寄存器和CAN_ERR1寄存器将更新计数器值以及保存的错误标志位。

状态

被动错误状态

当CAN_ERR0寄存器的TECNT[7:0]或RECNT[7:0]计数值增加到大于127时，CAN_ERR1寄存器的ERRSI[1:0]位域更新为1（被动错误状态）。

主动错误状态

当节点为被动错误状态，并且当CAN_ERR0寄存器的TECNT[7:0]或RECNT[7:0]计数值其中一个已满足小于等于127的条件，而另一个也减少到小于等于127时，CAN_ERR1寄存器的

ERRSI[1:0]位域更新为0（主动错误状态）。

离线状态

如果CAN_ERR0寄存器的TECNT[7:0]计数值增加到大于255，则CAN_ERR1寄存器的ERRSI[1:0]位域更新为0b1x（离线状态），并且CAN_ERR1寄存器的BOF位将置位，如果CAN_CTL1寄存器的BOIE位置位，则将产生一个中断。随后TECNT[7:0]计数值复位为0。

离线恢复：

离线恢复要求CAN总线能检测到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）。当CAN_ERR0寄存器的TECNT[7:0]计数值达到128时，CAN_ERR1寄存器的ERRSI[1:0]位域更新为0（主动错误状态），并且CAN_ERR0寄存器的TECNT[7:0]和RECNT[7:0]计数值都复位为0。

可通过配置CAN_CTL1寄存器的ABORDIS位来控制当检测到离线恢复序列后是自动恢复还是保持在离线状态。

如果ABORDIS位为0，使能了自动离线恢复，则CAN总线在检测到离线恢复序列后将自动恢复。如果在检测到离线恢复序列后ABORDIS位才变为0，则CAN总线需要再检测到11个连续的隐性位后才恢复与总线的同步。

如果ABORDIS位为1，禁能了自动离线恢复。如果在CAN节点进入离线状态之后ABORDIS位才变为1，则在下一次CAN节点进入离线状态才禁用自动离线恢复功能。

总线集成状态

如果节点检测到了协议异常事件（当CAN_CTL0寄存器的FDEN位为0时，如果收到了一个FD帧的FDF位），或在离线恢复过程中开始协议操作，则节点进入总线集成状态。在该状态，CAN节点与总线脱离同步。当节点检测到总线空闲条件（11个连续的隐性位）时，节点退出总线集成状态。请参考CAN协议ISO11898-1规范。

协议异常的监测由CAN_CTL2寄存器的PREEN位来控制。

可通过CAN_CTL2寄存器的EFDIS位来配置边沿滤波，用于总线集成状态。当使能了边沿滤波，在硬件同步的边沿检测时需要检测到连续两个正常时间单元的显性电平。当发生了硬件同步，对总线空闲条件（11个连续的隐性位）的检测将重新开始。如果应用了边沿滤波，小于一个正常时间单元（FD帧中数据阶段的位）的总线显性电平将被忽略，以避免误触发总线空闲条件。请参考CAN协议ISO11898-1规范。

注意：建议保持EFDIS位为0来使能边沿滤波，以避免总线空闲条件被误检测。

错误

如果至少有一个错误标志位置位（CAN_ERR1寄存器中的ACKERR，BRERR，BDERR，CRCERR，FMERR和STFERR），则CAN_ERR1寄存器的ERRSF位将置位。如果CAN_CTL1寄存器的ERRSIE位为1，将产生一个错误中断。

如果至少有一个错误标志位置位（CAN_ERR1寄存器中的BRFERR，BDFERR，CRCFERR，FMFERR和STFFERR），则CAN_ERR1寄存器中的ERRFSF位将置位。如果CAN_CTL2寄存器的ERRFSIE位为1，将产生一个FD帧BRS位为隐性位时数据阶段的错误中断。

ACK错误

如果连接中只存在一个节点，则在每次发送帧的时候都会导致CAN_ERR0寄存器的TECNT[7:0]计数器值增加（由ACK错误引起，最大到128），并且发生一个ACK错误，由CAN_ERR1寄存器的ACKERR位指示。

位隐性错误

如果至少有一个位发送为‘1’，接收为‘0’，则发生了一个位隐性错误。参考CAN_ERR1寄存器的BRFERR和BRERR位。

位显性错误

如果至少有一个位发送为‘0’，接收为‘1’，则发生了一个位显性错误。参考CAN_ERR1寄存器的BDFERR和BDERR位。

CRC错误

如果计算的CRC校验值与接收帧的CRC字段值不同，则发生了一个CRC错误。请参考CAN_ERR1寄存器的CRCFERR和CRCERR位。

格式错误

如果固定格式的字段包含至少一个非法的位，则发生了一个格式错误。请参考CAN_ERR1寄存器的FMFERR和FMERR位。

填充错误

请参考CAN_ERR1寄存器的STFFERR和STFERR位。

23.3.10. 通信参数

位时间

CAN协议控制器将位时间分为三个部分：

同步段（SYNC_SEG）：期望在该段检测到有效跳变沿。该段占用1个时间单元（$1\times t_q$）。

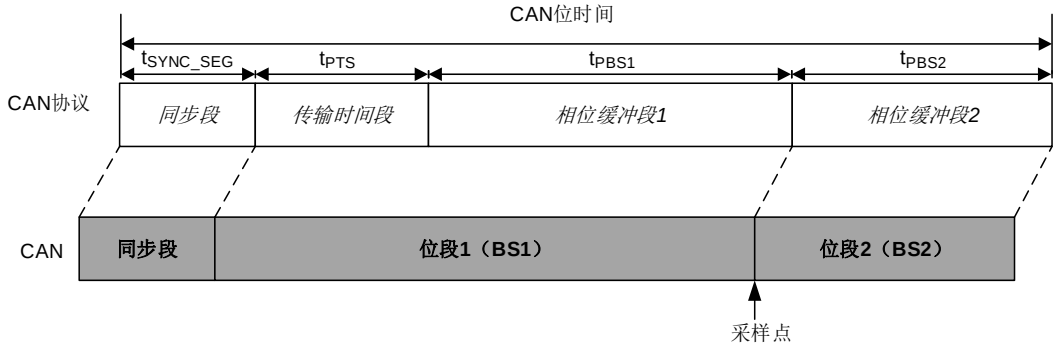
位段1（BS1）：该段包括CAN协议中的传播时间段和相位缓冲段1。该段可自动延长来补偿网络节点的频率不同引起的相位正漂移。

位段2（BS2）：该段定义了采样点。该段同样可以自动缩短来补偿相位负漂移。该段占用的时间单元不可少于2个。

注意：位时间的配置范围必须符合CAN协议规范ISO 11898-1。

位时间如图*图23-3. CAN位时间*所示。

图 23-3. CAN 位时间



再同步补偿宽度（SJW）：可延长或缩短再同步补偿宽度来补偿CAN网络节点的同步误差。通过CAN_BT寄存器的SJW[4:0]来配置正常位时间下的再同步补偿宽度，通过CAN_FDBT寄存器的DSJW[2:0]来配置数据位时间下的再同步补偿宽度。

有效跳变沿定义为在控制器发送隐性位之前，一个位时间内从显性位到隐性位的第一次转变。

如果有效跳变沿在BS1期间被检测到，而不是在SYNC_SEG期间，BS1将最多被延长SJW，因此采样点延迟。

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。

位采样

通过CAN_CTL1寄存器的BSPMOD位来定义Rx接收引脚上的采样模式。

当BSPMOD位为0，则只采样一次（即采样点）。

当BSPMOD位为1，则采样3次来决定接收的位电平，包括采样点，以及2次在采样点之前的采样。

注意：该位在CAN FD模式时不能为1。

波特率

CAN模块有两个时钟域：

- 控制单元、CAN寄存器的时钟来自APB2总线时钟。
- 协议控制器的时钟（CANCLK）由RCU_CFG2寄存器的CANxSEL[1:0]位域来配置，可配置为外部晶振时钟，或者APB2总线时钟，或者APB2总线时钟除以2，或者IRC8M内部时钟。

CAN波特率计算如下：

$$\text{BaudRate} = \frac{1}{\text{CAN Bit Time}} \tag{23-7}$$

$$\text{CAN Bit Time} = t_{SYNC_SEG} + t_{PTS} + t_{PBS1} + t_{PBS2} \tag{23-8}$$

其中：

$$t_{SYNC_SEG} = 1 \times t_q \quad (23\text{-}9)$$

$$t_{PTS} = (N_{PTS} + 1) \times t_q \text{ or } t_{PTS} = N_{DPTS} \times t_q \quad (23\text{-}10)$$

$$t_{PBS1} = (N_{PBS1} + 1) \times t_q \quad (23\text{-}11)$$

$$t_{PBS2} = (N_{PBS2} + 1) \times t_q \quad (23\text{-}12)$$

$$t_q = (N_{BAUDPSC} + 1) \times t_{CANCLK} \quad (23\text{-}13)$$

在公式中，对于正常波特率：

N_{PTS}，N_{PBS1}，N_{PBS2}和$N_{BAUDPSC}$分别由CAN_BT寄存器的PTS[5:0]，PBS1[4:0]，PBS2[4:0]和BAUDPSC[9:0]位域来配置。

对于数据波特率：

N_{DPTS}，N_{PBS1}，N_{PBS2}和$N_{BAUDPSC}$分别由CAN_FDBT寄存器的DPTS[4:0]，DPBS1[2:0]，DPBS2[2:0]和DBAUDPSC[9:0]位域来配置。

注意：接收匹配过程与发送仲裁过程中，CAN需要在一个CAN帧期间完成对所有邮箱的搜寻。为了满足这部分的搜寻时间要求，CAN位时间计算按CAN Bit Time = NUM × t_{CANCLK}，需要满足Min_{NUM}=16的要求。

时间戳

CAN硬件支持一个16位的内部计数器（计数值可通过CAN_TIMER寄存器来读写）用于生成时间戳。在一次成功的发送或者接收之后，将在CAN总线的SOF场抓取内部计数器的值，并写入到MDES0或者FDES0字的TIMESTAMP位域中。

在暂停模式下或者当 CAN_CTL0 寄存器的 LPS 位为 1 时，内部计数器停止计数。

内部计数器时钟源

如果CAN_CTL2寄存器的ITSRC位为1，则选择TRIGSEL的输出CANx_EX_TIME_TICK作为内部计数器的递增条件。其中CANx_EX_TIME_TICK和APB2总线时钟属于同一时钟域，为了保证内部计数器能够有效递增，CANx_EX_TIME_TICK信号源的脉冲宽度需要大于等于APB2总线时钟周期。

如果 CAN_CTL2 寄存器的 ITSRC 位为 0，则选择 CAN 波特率作为内部计数器的递增条件，即每发送或接收一个位，计数值加 1。当总线上没有消息时，则计数器按前一次配置的 CAN 波特率进行计数。

时间同步

如果 CAN_CTL1 寄存器的 TSYNC 位为 1，当第一个邮箱描述符成功接收到了任意报文时，则将内部计数器值复位来完成网络时间的同步。

23.3.11. 中断

CAN中断事件与标志如下表*表23-11. 中断事件*所示。

表 23-11. 中断事件

<table>
<tr><th colspan="2" rowspan="2">中断事件</th><th colspan="2">标志</th><th colspan="4">使能控制</th></tr>
<tr><th>位</th><th>寄存器</th><th>使能位</th><th>控制位</th><th>使能寄存器</th><th>控制寄存器</th></tr>
<tr><td colspan="2">离线</td><td colspan="2">BOF</td><td rowspan="13">CAN_ERR1</td><td colspan="2">BOIE</td><td colspan="2">CAN_CTL1</td></tr>
<tr><td colspan="2">离线恢复</td><td colspan="2">BORF</td><td colspan="2">BORIE</td><td colspan="2">CAN_CTL2</td></tr>
<tr><td rowspan="6">错误汇总</td><td>位隐性错误</td><td rowspan="6">ERRSF</td><td>BRERR</td><td colspan="2" rowspan="6">ERRSIE</td><td colspan="2" rowspan="6">CAN_CTL1</td></tr>
<tr><td>位显性错误</td><td>BDERR</td></tr>
<tr><td>ACK错误</td><td>ACKERR</td></tr>
<tr><td>CRC错误</td><td>CRCERR</td></tr>
<tr><td>格式错误</td><td>FMERR</td></tr>
<tr><td>填充错误</td><td>STFERR</td></tr>
<tr><td rowspan="5">FD帧数据位时间的错误汇总</td><td>位隐性错误</td><td rowspan="5">ERRFSF</td><td>BRFERR</td><td colspan="2" rowspan="5">ERRFSIE</td><td colspan="2" rowspan="5">CAN_CTL2</td></tr>
<tr><td>位显性错误</td><td>BDFERR</td></tr>
<tr><td>CRC错误</td><td>CRCFERR</td></tr>
<tr><td>格式错误</td><td>FMFERR</td></tr>
<tr><td>填充错误</td><td>STFFERR</td></tr>
<tr><td colspan="2">Tx错误警告</td><td colspan="2">TWERRIF</td><td>TWERRIE</td><td rowspan="2">WERREN</td><td rowspan="2">CAN_CTL1</td><td rowspan="2">CAN_CTL0</td></tr>
<tr><td colspan="2">Rx错误警告</td><td colspan="2">RWERRIF</td><td>RWERRIE</td></tr>
<tr><td colspan="2">匹配唤醒</td><td colspan="2">WMS</td><td rowspan="2">CAN_PN_STAT</td><td colspan="2">WMIE</td><td colspan="2" rowspan="2">CAN_PN_CTL0</td></tr>
<tr><td colspan="2">超时唤醒</td><td colspan="2">WTOS</td><td colspan="2">WTOIE</td></tr>
<tr><td colspan="2" rowspan="2">邮箱成功发送或接收帧</td><td colspan="2">所有位</td><td rowspan="6">CAN_STAT</td><td>所有位</td><td>RFEN = 0</td><td rowspan="4">CAN_INTEN</td><td rowspan="4">CAN_CTL0</td></tr>
<tr><td colspan="2">MSx</td><td>MIEx</td><td>RFEN = 1</td></tr>
<tr><td colspan="2">Rx FIFO非空</td><td colspan="2">MS5_RFNE</td><td>MIE5</td><td rowspan="3">RFEN = 1 & DMAEN = 0</td></tr>
<tr><td colspan="2">Rx FIFO警告</td><td colspan="2">MS6_RFW</td><td>MIE6</td></tr>
<tr><td colspan="2">Rx FIFO溢出</td><td colspan="2">MS7_RFO</td><td>MIE7</td></tr>
</table>

23.4. 典型的 CAN 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CAN模块：

- 配置CAN模块的时钟源CANCLK，并使能CAN模块时钟
  配置RCU_CFG2寄存器的CANxSEL[1:0]位来选择CAN模块的时钟源。配置RCU_APB2EN寄存器来使能CAN模块时钟。
- 配置通讯接口
  配置GPIO和AFIO模块，将相应的功能引脚映射到复用功能上。
- 进入暂停模式
  由于INAMOD位，HALT位，NRDY位和INAS位在上电复位或系统复位后默认置位，因此CAN将自动进入暂停模式，用以进行CAN寄存器的配置。
- 处理CAN_STAT寄存器中置位的标志位
  读取接收邮箱描述符或者Rx FIFO描述符的内容，清除CAN_STAT寄存器中相关标志位，然后读取CAN_TIMER寄存器来完成标志位的处理服务。如果使能了Rx FIFO，通过将CAN_STAT寄存器的MS0位置位来进行清FIFO操作。同样进行发送邮箱的置位的标志位的处理。

- 初始化邮箱描述符或者Rx FIFO描述符的物理内存空间
  通过CAN_CTL0寄存器的MSZ[4:0]位域来配置邮箱描述符或者Rx FIFO描述符的物理内存空间。
- 配置通信参数
  1）在CAN_BT寄存器中的PTS[5:0]，PBS1[4:0]，PBS2[4:0]，SJW[4:0]和BAUDPSC[9:0]位域来配置CAN的正常波特率。
  2）如果需要，可通过CAN_CTL1寄存器的BSPMOD位来配置采样模式。
  3）如果需要，可通过配置PREEN和EFDIS位用于总线集成状态。
- 配置发送相关的控制参数
  1）通过CAN_CTL1的MTO位和CAN_CTL0寄存器的LAPRIOEN位来配置仲裁优先级。
  2）如果需要，可通过CAN_CTL2寄存器的ASD[4:0]位域来配置仲裁启动延迟。
  3）通过配置CAN_CTL0寄存器的MST位来使能发送邮箱描述符的发送中止功能。
- 配置接收相关的控制参数
  1）通过CAN_CTL0寄存器的RFEN位来选择是否使用Rx FIFO，通过DMAEN位来选择是否使用Rx FIFO DMA功能。
  2）通过CAN_CTL0寄存器的RPFQEN位来配置接收私有过滤器&接收邮箱队列功能。
  3）通过CAN_CTL2寄存器的RFO，RRFRMS和IDERTR_RMF位来配置接收过滤相关参数。
  4）通过CAN_RMPUBF，CAN_RFIFOPUBF和CAN_RFIFOMPFx（x = 0..31）寄存器来进行接收邮箱和Rx FIFO过滤数据的配置。如果使能了Rx FIFO，还要通过CAN_CTL0寄存器的FS[1:0]位域来配置Rx FIFO标识符过滤表元素格式，通过CAN_CTL2寄存器的RFFN[3:0]位域来配置Rx FIFO标识符过滤表元素数目。
- 如果需要CAN FD操作
  1）通过CAN_CTL2寄存器的ISO位进行CAN FD协议的选择。
  2）通过CAN_CTL0寄存器的FDEN位来使能CAN FD模式。
  3）通过CAN_FDCTL寄存器的MDSZ[1:0]位域来配置邮箱数目。
  4）如果需要，通过CAN_FDCTL寄存器的TDCEN和TDCO[4:0]来进行CAN FD的传输延迟补偿功能配置。
  5）通过CAN_FDBT寄存器的DPTS[4:0]，DPBS1[2:0]，DPBS2[2:0]，DSJW[2:0]和DBAUDPSC[9:0]来进行CAN数据波特率的配置。
- 配置中断
  通过CAN_CTL0，CAN_CTL1，CAN_CTL2和CAN_INTEN寄存器来使能需要的中断。
- 初始化发送/接收邮箱描述符
  1）如果需要发送，初始化发送邮箱描述符。
  2）如果需要接收，初始化接收邮箱描述符，如果使能了Rx FIFO，则还需初始化Rx FIFO描述符，以及Rx FIFO标识符过滤表元素。
- 如果需要进入虚拟联网模式，置位CAN_CTL0寄存器的PNMOD位和PNEN位，并配置相关用于唤醒的寄存器。
- 退出暂停模式
  通过清除CAN_CTL0寄存器的HALT位来退出暂停模式，随后CAN节点将恢复与CAN总线的同步。

23.5. CAN 寄存器

CAN0基地址：0x4001 A000

CAN1基地址：0x4001 B000

23.5.1. 控制寄存器 0（CAN_CTL0）

地址偏移：0x00
复位值：0x5900 000F

该寄存器中除了位30，28，25，19之外的其他位都只可在暂停模式下配置，它们在其他模式下被硬件锁定。

该寄存器中除了位31，27，24，20之外的其他位都会被CAN_CTL0寄存器中的软件复位SWRST位复位。

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
CANDIS	INAMOD	RFEN	HALT	NRDY	保留	SWRST	INAS	保留		WERREN	LPS	PNEN	PNS	SRDIS	RPFQEN
rw	rw	rw	rw	r		rw	r			rw	r	rw	r	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAEN	PNMOD	LAPRIOEN	MST	FDEN	保留	FS[1:0]		保留			MSZ[4:0]				
rw	rw	rw	rw	rw		rw					rw				

位/位域	名称	描述
31	CANDIS	CAN禁能 0：使能CAN模块 1：禁能 CAN 模块
30	INAMOD	暂停模式使能 0：禁能暂停模式 1：使能暂停模式
29	RFEN	Rx FIFO使能 0：禁能Rx FIFO 1：使能 Rx FIFO
28	HALT	暂停CAN 0：无进入暂停模式请求 1：当 CAN_CTL0 寄存器的 INAMOD 位置位时进入暂停模式
27	NRDY	未准备好 该位指示了协议控制器的时钟是否被禁用。当在暂停模式下，或在CAN_Disable模式下，协议控制器的时钟被禁用，CAN模块未准备好。 0：CAN模块已准备好

		1：CAN 模块未准备好
26	保留	必须保持复位值。
25	SWRST	软件复位 当该位置位时，CAN内部状态机和CAN寄存器将被复位。 该位在软件复位完成后将由硬件自动清零。 0：无作用 1：软件复位
24	INAS	暂停模式状态 0：不处于暂停模式 1：处于暂停模式
23:22	保留	必须保持复位值。
21	WERREN	错误警告使能 当该位置位时，CAN_ERR1寄存器中的警告中断标志TWERRIF和RWERRIF位将被使能，分别用于反映CAN_ERR1寄存器中的TWERRF和RWERRF位状态切换。 0：禁能Tx和Rx错误警告 1：使能Tx和Rx错误警告
20	LPS	低功耗状态 0：不处于低功耗状态 1：处于低功耗状态
19	PNEN	虚拟联网模式使能 0：禁能虚拟联网模式 1：使能虚拟联网模式
18	PNS	虚拟联网状态 0：不处于虚拟联网状态 1：处于虚拟联网状态
17	SRDIS	自接收禁能 0：使能自接收 1：禁能自接收
16	RPFQEN	接收私有过滤使能&接收邮箱队列使能 0：禁能接收私有过滤&禁能接收邮箱队列 1：使能接收私有过滤&使能接收邮箱队列
15	DMAEN	DMA使能 0：禁能Rx FIFO的DMA功能 1：使能 Rx FIFO 的 DMA 功能
14	PNMOD	虚拟联网模式选择 0：不选择虚拟联网模式 1：选择虚拟联网模式
13	LAPRIOEN	本地仲裁优先级使能

		0：禁能本地仲裁优先级 1：使能本地仲裁优先级
12	MST	邮箱中止发送 0：禁能发送中止 1：使能发送中止
11	FDEN	CAN FD模式使能 0：禁能CAN FD模式 1：使能CAN FD模式
10	保留	必须保持复位值。
9:8	FS[1:0]	格式选择 该位域定义了Rx FIFO标识符过滤表元素的格式。 00：格式A：每个标识符过滤表元素包含一个完整标识符（标准格式和扩展格式） 01：格式B：每个标识符过滤表元素包含两个完整标准格式标识符或者两个扩展格式标识符其中14位 10：格式C：每个标识符过滤表元素包含四个标准格式标识符其中8位或者四个扩展格式标识符其中8位 11：格式 D：不接受所有帧
7:5	保留	必须保持复位值。
4:0	MSZ[4:0]	内存大小 该位域定义了帧发送和接收使用的最大内存大小。这个内存大小以4字（等于8字节数据段时的邮箱描述符大小）为单位计算，包含了邮箱和Rx FIFO占用的空间。 在配置该位域之前，必须将CAN_STAT寄存器中的所有置位标志位都处理服务。 00000：1单位 00001：2单位 … 11111：32单位

23.5.2. 控制寄存器 1（CAN_CTL1）

地址偏移：0x04
复位值：0x0000 0000

该寄存器中位 12，7，5，4，3 只可在暂停模式下配置，它们在其他模式下被硬件锁定。

该寄存器中所有位都不会被 CAN_CTL0 寄存器中的软件复位 SWRST 位复位。

该寄存器只能按字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOIE	ERRSIE	保留	LSCMOD	TWERRIE	RWERRIE	保留		BSPMOD	ABORDIS	TSYNC	MTO	MMOD	保留		

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	BOIE	离线中断使能 0：禁能离线中断 1：使能离线中断
14	ERRSIE	错误汇总中断使能 0：禁能错误汇总中断 1：使能错误汇总中断
13	保留	必须保持复位值。
12	LSCMOD	回环静默模式 0：禁能回环静默模式 1：使能回环静默模式 **注意：**在该模式下，不能置位CAN_CTL0寄存器的SRDIS位，和CAN_FDCTL寄存器的TDCEN位。
11	TWERRIE	发送错误警告中断使能 只有当CAN_CTL0寄存器的WERREN位置位时才可写该位。当CAN_CTL0寄存器的WERREN位为0时，该位读为0。 0：禁能发送错误警告中断 1：使能发送错误警告中断
10	RWERRIE	接收错误警告中断使能 只有当CAN_CTL0寄存器的WERREN位置位时才可写该位。当CAN_CTL0寄存器的WERREN位为0时，该位读为0。 0：禁能接收错误警告中断 1：使能接收错误警告中断
9:8	保留	必须保持复位值。
7	BSPMOD	位采样模式 0：接收位使用1个采样点 1：接收位使用3个采样点
6	ABORDIS	自动离线恢复不使能 0：使能自动离线恢复 1：不使能自动离线恢复
5	TSYNC	时间同步使能 0：禁能时间同步 1：使能时间同步
4	MTO	邮箱发送顺序 0：高优先级的邮箱先发送

		1：低邮箱编号的邮箱先发送
3	MMOD	监听模式 0：禁能监听模式 1：使能监听模式
2:0	保留	必须保持复位值。

23.5.3. 计数器寄存器（CAN_TIMER）

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
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]	计数器值 该位域包含用于产生时间戳的内部计数器值。

23.5.4. 接收邮箱公有过滤寄存器（CAN_RMPUBF）

地址偏移：0x10
复位值：0xXXXX XXXX

该寄存器位于 RAM。

该寄存器中所有位都只可在暂停模式下配置，它们在其他模式下被硬件锁定。

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MFD31	MFD30	MFD29	MFD28	MFD27	MFD26	MFD25	MFD24	MFD23	MFD22	MFD21	MFD20	MFD19	MFD18	MFD17	MFD16
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
MFD15	MFD14	MFD13	MFD12	MFD11	MFD10	MFD9	MFD8	MFD7	MFD6	MFD5	MFD4	MFD3	MFD2	MFD1	MFD0
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	MFDx	邮箱过滤数据 MFD31 位用于过滤邮箱描述符的 RTR 域。

MFD30 位用于过滤邮箱描述符的 IDE 域。

MFDx（x = 0..28）用于过滤邮箱描述符的 ID 域。

0：不关心该位

1：参与比较

23.5.5. 错误寄存器 0（CAN_ERR0）

地址偏移：0x1C

复位值：0x0000 0000

该寄存器的所有位都只读，除了在暂停模式之外。

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REFCNT[7:0]								TEFCNT[7:0]							
rw0								rw0							

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

位/位域	名称	描述
31:24	REFCNT[7:0]	FD 帧 BRS 位为隐性位时数据阶段的接收错误计数器 该位域在暂停模式下只可写为0。
23:16	TEFCNT[7:0]	FD 帧 BRS 位为隐性位时数据阶段的发送错误计数器 该位域在暂停模式下只可写为0。
15:8	RECNT[7:0]	CAN 协议定义的接收错误计数器
7:0	TECNT[7:0]	CAN 协议定义的发送错误计数器

23.5.6. 错误寄存器 1（CAN_ERR1）

地址偏移：0x20

复位值：0x000X 000X

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BRFERR	BDFERR	保留	CRCFERR	FMFERR	STFFERR	保留				ERROVR	ERRFSF	BORF	SYN	TWERRIF	RWERRIF
rc	rc		rc	rc	rc					rc_w1	rc_w1	rc_w1	r	rc_w1	rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRERR	BDERR	ACKERR	CRCERR	FMERR	STFERR	TWERRF	RWERRF	IDLEF	TS	ERRSI[1:0]		RS	BOF	ERRSF	保留
rc	rc	rc	rc	rc	rc	r	r	r	r	r		r	rc_w1	rc_w1	

位/位域	名称	描述
31	BRFERR	FD帧BRS位为隐性位时数据阶段的位隐性错误

		0：没有发生错误 1：至少有一个位发送为隐性位，接收为显性位
30	BDFERR	FD帧BRS位为隐性位时数据阶段的位显性错误 0：没有发生错误 1：至少有一个位发送为显性位，接收为隐性位
29	保留	必须保持复位值。
28	CRCFERR	FD帧BRS位为隐性位时数据阶段的CRC错误 0：没有发生错误 1：发生了一个 CRC 错误
27	FMFERR	FD帧BRS位为隐性位时数据阶段的格式错误 0：没有发生错误 1：发生了一个格式错误
26	STFFERR	FD帧BRS位为隐性位时数据阶段的填充错误 0：没有发生错误 1：发生了一个填充错误
25:22	保留	必须保持复位值。
21	ERROVR	错误溢出 该位表示在某一个错误标志位已经置位的情况下，又检测到了一个错误。 0：没有发生错误溢出 1：发生了错误溢出
20	ERRFSF	FD帧BRS位为隐性位时数据阶段的错误汇总标志 该位是下列位的逻辑或： CAN_ERR1[31]：位隐性错误 CAN_ERR1[30]：位显性错误 CAN_ERR1[28]：CRC错误 CAN_ERR1[27]：格式错误 CAN_ERR1[26]：填充错误
19	BORF	离线恢复标志 当检测到CAN总线上总线恢复序列时，该位置位，指示CAN节点可以从离线状态恢复。 0：没有事件发生 1：发生了离线恢复序列事件
18	SYN	同步标志 0：未与CAN总线同步 1：与 CAN 总线同步
17	TWERRIF	发生错误警告中断标志 该位在离线状态时不使用。 0：没有事件发生

		1：CAN_ERR1寄存器的TWERRF位从0变为1
16	RWERRIF	接收错误警告中断标志 该位在退出虚拟联网模式时将更新。 0：没有事件发生 1：CAN_ERR1 寄存器的 RWERRF 位从 0 变为 1
15	BRERR	所有格式帧的位隐性错误 该位在退出虚拟联网模式时将更新。 0：没有发生错误 1：至少有一个位发送为隐性位，接收为显性位
14	BDERR	所有格式帧的位显性错误 该位在退出虚拟联网模式时将更新。 0：没有发生错误 1：至少有一个位发送为显性位，接收为隐性位
13	ACKERR	ACK错误 该位在退出虚拟联网模式时将更新。 0：没有发生错误 1：发生了一个 ACK 错误
12	CRCERR	CRC错误 该位在退出虚拟联网模式时将更新。 0：没有发生错误 1：发生了一个 CRC 错误
11	FMERR	格式错误 该位在退出虚拟联网模式时将更新。 0：没有发生错误 1：发生了一个格式错误
10	STFERR	填充错误 该位在退出虚拟联网模式时将更新。 0：没有发生错误 1：发生了一个填充错误
9	TWERRF	发送错误警告标志 0：没有事件发生 1：CAN_ERR0 寄存器的 TECNT[7:0]值大于等于 96
8	RWERRF	接收错误警告标志 该位在退出虚拟联网模式时将更新。 0：没有事件发生 1：CAN_ERR0 寄存器的 RECNT[7:0]值大于等于 96
7	IDLEF	空闲标志 0：没有事件发生 1：处于总线空闲状态

6	TS	发送状态 0：CAN节点不处于发送状态 1：CAN 节点处于发送状态
5:4	ERRSI[1:0]	错误状态指示 当CAN_CTL1寄存器的MMOD位，和CAN_CTL0寄存器的SWRST位都置位时，该位会复位一个CAN位时间，然后变为监听模式时的0b01值。 00：主动错误 01：被动错误 1x：离线
3	RS	接收状态 0：CAN节点不处于接收状态 1：CAN 节点处于接收状态
2	BOF	离线标志 0：没有事件发生 1：处于离线状态
1	ERRSF	错误汇总标志 该位是下列位的逻辑或： CAN_ERR1[15]：位隐性错误 CAN_ERR1[14]：位显性错误 CAN_ERR1[13]：ACK错误 CAN_ERR1[12]：CRC错误 CAN_ERR1[11]：格式错误 CAN_ERR1[10]：填充错误
0	保留	必须保持复位值。

23.5.7. 中断使能寄存器（CAN_INTEN）

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
MIE31	MIE30	MIE29	MIE28	MIE27	MIE26	MIE25	MIE24	MIE23	MIE22	MIE21	MIE20	MIE19	MIE18	MIE17	MIE16
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
MIE15	MIE14	MIE13	MIE12	MIE11	MIE10	MIE9	MIE8	MIE7	MIE6	MIE5	MIE4	MIE3	MIE2	MIE1	MIE0
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	MIEx	消息发送和接收中断使能 当 Rx FIFO 禁能时，这些位用于邮箱编号 x（参考*邮箱编号*）的中断配置。 当 Rx FIFO 使能时，MIE5 到 MIE7 都用于 Rx FIFO 的中断配置，邮箱的中断配置

位为位 x 对应于邮箱编号 x（参考*邮箱编号*）。

0：禁能相应中断

1：使能相应中断

23.5.8. 状态寄存器（CAN_STAT）

地址偏移：0x30

复位值：0x0000 0000

当 CAN_CTL0 寄存器的 RFEN 位的配置改变时，该寄存器的位 1 到位 7 都会被清零。

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MS31	MS30	MS29	MS28	MS27	MS26	MS25	MS24	MS23	MS22	MS21	MS20	MS19	MS18	MS17	MS16
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MS15	MS14	MS13	MS12	MS11	MS10	MS9	MS8	MS7_RFO	MS6_RFW	MS5_RFNE	MS4_RES	MS3_RES	MS2_RES	MS1_RES	MS0_RFC
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
31:8	MSx	邮箱x状态 x是邮箱编号，参考*邮箱编号*。 0：邮箱描述符没有发生消息的成功发送或接收 1：邮箱描述符发生了一次消息的成功发送或接收
7	MS7_RFO	邮箱7状态 / Rx FIFO溢出 0：当Rx FIFO禁能时，邮箱7描述符没有发生消息的成功发送或接收 / 当Rx FIFO使能时，Rx FIFO没有发生溢出。 1：当Rx FIFO禁能时，邮箱7描述符发生了一次消息的成功发送或接收 / 当Rx FIFO使能时，Rx FIFO发生了溢出。
6	MS6_RFW	邮箱6状态 / Rx FIFO警告 0：当Rx FIFO禁能时，邮箱6描述符没有发生消息的成功发送或接收 / 当Rx FIFO使能时，Rx FIFO没有发生快满了的警告。 1：当Rx FIFO禁能时，邮箱6描述符发生了一次消息的成功发送或接收 / 当Rx FIFO使能时，Rx FIFO发生快满了的警告。
5	MS5_RFNE	邮箱5状态 / Rx FIFO非空 0：当Rx FIFO禁能时，邮箱5描述符没有发生消息的成功发送或接收 / 当Rx FIFO使能时，Rx FIFO为空。 1：当Rx FIFO禁能时，邮箱5描述符发生了一次消息的成功发送或接收 / 当Rx FIFO使能时，Rx FIFO非空。
4	MS4_RES	邮箱4状态 / 保留 与MS1_RES描述类似。
3	MS3_RES	邮箱3状态 / 保留

		与 MS1_RES 描述类似。
2	MS2_RES	邮箱2状态 / 保留 与MS1_RES描述类似。
1	MS1_RES	邮箱1状态 / 保留 0：当Rx FIFO禁能时，邮箱1描述符没有发生消息的成功发送或接收 / 当Rx FIFO使能时，该位保留。 1：当Rx FIFO禁能时，邮箱1描述符发生了一次消息的成功发送或接收 / 当Rx FIFO使能时，该位保留。
0	MS0_RFC	邮箱0状态 / 清Rx FIFO位 0：当Rx FIFO禁能时，邮箱0描述符没有发生消息的成功发送或接收 / 当Rx FIFO使能时，不起作用。 1：当Rx FIFO禁能时，邮箱0描述符发生了一次消息的成功发送或接收 / 当Rx FIFO使能时，清Rx FIFO，该位只允许在暂停模式下写入，参考***清FIFO***。

23.5.9. 控制寄存器 2（CAN_CTL2）

地址偏移：0x34
复位值：0x00A0 0000

该寄存器中除了位 31，30 之外的其他位都只可在暂停模式下配置，它们在其他模式下被硬件锁定。

该寄存器中所有位都不会被 CAN_CTL0 寄存器中的软件复位 SWRST 位复位。

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
ERRFSIE	BORIE	保留		RFFN[3:0]				ASD[4:0]					RFO	RRFRMS	IDERTR_RMF
rw	rw			rw				rw					rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ITSRC	PREEN	保留	ISO	EFDIS	保留										
rw	rw		rw	rw											

位/位域	名称	描述
31	ERRFSIE	FD帧BRS位为隐性位时数据阶段的错误汇总中断使能 0：禁能FD帧BRS位为隐性位时数据阶段的错误汇总中断 1：使能 FD 帧 BRS 位为隐性位时数据阶段的错误汇总中断
30	BORIE	离线恢复中断使能 0：禁能离线恢复中断 1：使能离线恢复中断
29:28	保留	必须保持复位值。
27:24	RFFN[3:0]	Rx FIFO过滤器数目

表 23-12. Rx FIFO 标识符过滤表元素数目

RFFN[3:0]	Rx FIFO标识符过滤表元素数目	Rx FIFO占用的空间	可用的邮箱
0000	8	邮箱描述符0 - 7	邮箱8 - 31
0001	16	邮箱描述符0 - 9	邮箱10 - 31
0002	24	邮箱描述符0 - 11	邮箱12 - 31
0003	32	邮箱描述符0 - 13	邮箱14 - 31
0004	40	邮箱描述符0 - 15	邮箱16 - 31
0005	48	邮箱描述符0 - 17	邮箱18 - 31
0006	56	邮箱描述符0 – 19	邮箱20 - 31
0007	64	邮箱描述符0 – 21	邮箱22 - 31
0008	72	邮箱描述符0 – 23	邮箱24 - 31
0009	80	邮箱描述符0 – 25	邮箱26 - 31
000A	88	邮箱描述符0 – 27	邮箱28 - 31
000B	96	邮箱描述符0 – 29	邮箱30 - 31
000C	104	邮箱描述符0 – 31	无
其他	104	邮箱描述符0 - 31	无

配置该位域时，需注意不要使Rx FIFO占用的内存空间超过由CAN_CTL0寄存器MSZ[4:0]位域配置的可用的内存空间大小，否则超过的部分将不起作用。

23:19 ASD[4:0]

仲裁启动延迟

该位域定义了在发送仲裁过程启动之前需要延迟多少个CAN位时间。

18 RFO

接收过滤顺序

0：先过滤比较Rx FIFO

1：先过滤比较邮箱

17 RRFRMS

远程请求帧存储

0：当找到了CODE为RANSWER的并且ID相匹配的邮箱，则产生一个远程应答帧。

1：当找到了 CODE 为 RANSWER 的并且 ID 相匹配的邮箱，则将这个远程请求帧如同数据帧一样存储起来，而不自动发送远程应答帧。

16 IDERTR_RMF

邮箱接收时IDE和RTR域的过滤类型

该位定义了接收邮箱描述符中IDE和RTR域与接收的位的匹配类型。

0：总是比较IDE域，从不比较RTR域。忽略相关过滤寄存器中的过滤数据配置。

1：过滤比较IDE和RTR域，使用相关过滤寄存器中的过滤数据配置。

15 ITSRC

内部计数器时钟源

0：CAN波特率

1：TRIGSEL输出的外部触发CANx_EX_TIME_TICK

14 PREEN

CAN规范中的协议异常检测使能

0：禁能协议异常检测

1：使能协议异常检测

13	保留	必须保持复位值。
12	ISO	ISO CAN FD 0：使用非ISO CAN FD协议 1：使用 ISO CAN FD 协议
11	EFDIS	边沿过滤禁能 0：使能边沿过滤 1：禁能边沿过滤
10:0	保留	必须保持复位值。

23.5.10. 常规帧 CRC 寄存器（CAN_CRCC）

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											ANTM[4:0]				
													r		

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

位/位域	名称	描述
31:21	保留	必须保持复位值。
20:16	ANTM[4:0]	发送 CRCTC[14:0]值的相关联的邮箱的编号 该位域包含发送了 CRC 值为 CRCTC[14:0]的邮箱的编号。
15	保留	必须保持复位值。
14:0	CRCTC[14:0]	发送的常规帧CRC计算值 该位域包含最新发送的常规帧的CRC计算值。

23.5.11. 接收 FIFO 共有过滤寄存器（CAN_RFIFOPUBF）

地址偏移：0x48
复位值：0xXXXX XXXX

该寄存器位域 RAM 区域。

该寄存器中所有位都只可在暂停模式下配置，它们在其他模式下被硬件锁定。

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FFD31	FFD30	FFD29	FFD27	FFD27	FFD26	FFD25	FFD24	FFD23	FFD22	FFD21	FFD20	FFD19	FFD18	FFD17	FFD16
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
FFD15	FFD14	FFD13	FFD12	FFD11	FFD10	FFD9	FFD8	FFD7	FFD6	FFD5	FFD4	FFD3	FFD2	FFD1	FFD0
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	FFDx	Rx FIFO 过滤数据 该位用于相应的标识符过滤表元素中各个位的过滤，过滤表元素中的保留位除外。 0：不关心该位 1：参与比较

23.5.12. 接收 FIFO 标识符过滤元素匹配序号寄存器（CAN_RFIFOIFMN）

地址偏移：0x4C
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															

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

位/位域	名称	描述
31:9	保留	必须保持复位值。
8:0	IDFMN[8:0]	标识符过滤元素匹配序号 只有当CAN_STAT寄存器的MS5_RFNE位置位时，该位域才有效。 该位域表示在接收FIFO输出中的消息是与哪个标识符过滤元素相匹配。如果有超过一个相匹配的标识符过滤元素，则该位域指示最小序号的匹配的标识符过滤元素。

23.5.13. 位时间寄存器（CAN_BT）

地址偏移：0x50
复位值：0x0100 0000

该寄存器中所有位都只可在暂停模式下配置，它们在其他模式下被硬件锁定。

该寄存器中所有位都不会被 CAN_CTL0 寄存器中的软件复位 SWRST 位复位。

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	BAUDPSC[9:0]										SJW[4:0]				
	rw										rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PTS[5:0]						PBS1[4:0]					PBS2[4:0]				

rw	rwr	rw

位/位域	名称	描述
31	保留	必须保持复位值。
30:21	BAUDPSC[9:0]	波特率分频系数 CAN波特率分配系数。 波特率分频系数= BAUDPSC[9:0] + 1
20:16	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
15:10	PTS[5:0]	传播时间段 传播时间段占用的时间单元数量 = PTS[5:0] + 1
9:5	PBS1[4:0]	相位缓冲段1 相位缓冲段 1 占用的时间单元数量 = PBS1[4:0] + 1
4:0	PBS2[4:0]	相位缓冲段2 相位缓冲段 2 占用的时间单元数量 = PBS2[4:0] + 1

23.5.14. 接收 FIFO/邮箱私有过滤 x 寄存器（CAN_RFIFOMPFx）（x=0..31）

地址偏移：0x880 + 4 * x
复位值：0xXXXX XXXX

该寄存器位于 RAM 区域。

这些寄存器中所有位都只可在暂停模式下配置，它们在其他模式下被硬件锁定。

这些寄存器中所有位都不会被 CAN_CTL0 寄存器中的软件复位 SWRST 位复位。

这些寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FMFD31	FMFD30	FMFD29	FMFD27	FMFD27	FMFD26	FMFD25	FMFD24	FMFD23	FMFD22	FMFD21	FMFD20	FMFD19	FMFD18	FMFD17	FMFD16
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
FMFD15	FMFD14	FMFD13	FMFD12	FMFD11	FMFD10	FMFD9	FMFD8	FMFD7	FMFD6	FMFD5	FMFD4	FMFD3	FMFD2	FMFD1	FMFD0
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	FMFDx	FIFO / 邮箱过滤数据 当用作邮箱过滤时，参考CAN_RMPUBF寄存器的MFDx位。 当用作 Rx FIFO 过滤时，参考 CAN_RFIFOPUBF 寄存器的 FFDx 位。 0：不关心该位 1：参与比较

23.5.15. 虚拟联网模式控制寄存器 0（CAN_PN_CTL0）

地址偏移：0xB00
复位值：0x0000 0100

该寄存器中除了位 17，16 之外的其他位都只可在暂停模式下配置，它们在其他模式下被硬件锁定。

该寄存器只能按字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NMM[7:0]								保留		DATAFT[1:0]		IDFT[1:0]		FFT[1:0]	
rw										rw		rw		rw	

位/位域	名称	描述
31:18	保留	必须保持复位值。
17	WTOIE	超时唤醒中断使能 0：禁能超时唤醒中断 1：使能超时唤醒中断
16	WMIE	匹配唤醒中断使能 0：禁能匹配唤醒中断 1：使能匹配唤醒中断
15:8	NMM[7:0]	消息匹配次数 事件计数器用于唤醒帧过滤，在检测到N次匹配事件后，会产生一个事件输出。 00000001：N = 1 00000010：N = 2 …… 11111111：N = 255
7:6	保留	必须保持复位值。
5:4	DATAFT[1:0]	在虚拟联网模式下DATA域的过滤类型 00：只有当帧的DATA域与相应的期望数据寄存器中DATA位域一致时，认为这是一个DATA匹配的帧 01：只有当帧的DATA域大于等于相应的期望数据寄存器中DATA下限值时，认为这是一个DATA匹配的帧 10：只有当帧的DATA域小于等于相应的期望数据寄存器中DATA上限值时，认为这是一个DATA匹配的帧 11：只有当帧的DATA域大于等于相应的期望数据寄存器中DATA下限值，并且小于等于相应的期望数据寄存器中DATA上限值时，认为这是一个DATA匹配的帧
3:2	IDFT[1:0]	在虚拟联网模式下ID域的过滤类型 00：只有当帧的ID域与相应的期望标识符寄存器中ID位域一致时，认为这是一个ID匹

		配的帧 01：只有当帧的ID域大于等于相应的期望标识符寄存器中ID下限值时，认为这是一个ID匹配的帧 10：只有当帧的ID域小于等于相应的期望标识符寄存器中ID上限值时，认为这是一个ID匹配的帧 11：只有当帧的ID域大于等于相应的期望标识符寄存器中ID下限值时，并且小于等于相应的期望标识符寄存器中ID上限值时，认为这是一个ID匹配的帧
1:0	FFT[1:0]	在虚拟联网模式下帧的过滤类型 00：除了DATA，DLC域之外的其他域都需要过滤比较 01：所有域都需要过滤比较 10：除了DATA，DLC域之外的其他域都需要过滤比较NMM[7:0]次数 11：所有域都需要过滤比较 NMM[7:0]次数

23.5.16. 虚拟联网模式超时寄存器（CAN_PN_TO）

地址偏移：0xB04
复位值：0x0000 0000

该寄存器中所有位都只可在暂停模式下配置，它们在其他模式下被硬件锁定。

该寄存器只能按字（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
WTO[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	WTO[15:0]	超时唤醒 该超时值按照CAN位时间的64倍进行计数。默认关闭超时唤醒。

23.5.17. 虚拟联网模式状态寄存器（CAN_PN_STAT）

地址偏移：0xB08
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														WTOS	WMS
														rc_w1	rc_w1

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

r r

位/位域	名称	描述
31:18	保留	必须保持复位值。
17	WTOS	超时唤醒标志状态 0：没有发生超时唤醒事件 1：发生了超时唤醒事件
16	WMS	匹配唤醒标志状态 0：没有发生匹配唤醒事件 1：发生了匹配唤醒事件
15:8	MMCNT[7:0]	在虚拟联网模式下的帧匹配计数 该位域指示了在虚拟联网模式下的匹配的帧的计数值。该位域在进入虚拟联网模式时由CAN模块复位，并且受软件复位的影响。
7	MMCNTS	帧匹配计数状态 当该位置位时，指示 MMCNT[7:0]值有效。 0：帧匹配计数MMCNT[7:0]正在更新 1：帧匹配计数 MMCNT[7:0]有效
6:0	保留	必须保持复位值。

23.5.18. 虚拟联网模式期望标识符 0 寄存器（CAN_PN_EID0）

地址偏移：0xB0C
复位值：0x0000 0000

该寄存器中所有位都只可在暂停模式下配置，它们在其他模式下被硬件锁定。

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	EIDE	ERTR	EID_ELT[28:16]												
	rw	rw							rw						

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

位/位域	名称	描述
31	保留	必须保持复位值。
30	EIDE	在虚拟联网模式下的期望IDE 0：标准格式 1：扩展格式
29	ERTR	在虚拟联网模式下的期望RTR 0：数据帧

		1：远程帧
28:0	EIDF_ELT[28:0]	在虚拟联网模式下的期望ID / 期望的ID下限值 当CAN_PN_CTL0寄存器的IDFT[1:0]位域为0 / 1 / 2时，该位域用作期望ID，当IDFT[1:0]位域为3时，该位域用作期望的ID下限值。 对于扩展格式帧，使用所有的29位。 对于标准格式帧，使用位18到28。

23.5.19. 虚拟联网模式期望 DLC 寄存器（CAN_PN_EDLC）

地址偏移：0xB10
复位值：0x0000 0008

该寄存器中所有位都只可在暂停模式下配置，它们在其他模式下被硬件锁定。

该寄存器只能按字（32 位）访问。

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

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

位/位域	名称	描述
31:20	保留	必须保持复位值。
19:16	DLCELT[3:0]	在虚拟联网模式下的期望DLC下限值
15:4	保留	必须保持复位值。
3:0	DLCEHT[3:0]	在虚拟联网模式下的期望DLC上限值

23.5.20. 虚拟联网模式期望数据低字 0 寄存器（CAN_PN_EDL0）

地址偏移：0xB14
复位值：0x0000 0000

该寄存器中所有位都只可在暂停模式下配置，它们在其他模式下被硬件锁定。

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DB0ELT[7:0]								DB1ELT[7:0]							
			rw								rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB2ELT[7:0]								DB3ELT[7:0]							
			rw								rw				

位/位域	名称	描述
31:24	DB0ELT[7:0]	在虚拟联网模式下的期望数据字节 0 下限值 参考 DB3ELT[7:0]描述。
23:16	DB1ELT[7:0]	在虚拟联网模式下的期望数据字节 1 下限值 参考 DB3ELT[7:0]描述。
15:8	DB2ELT[7:0]	在虚拟联网模式下的期望数据字节 2 下限值 参考 DB3ELT[7:0]描述。
7:0	DB3ELT[7:0]	在虚拟联网模式下的期望数据字节 3 下限值 当 CAN_PN_CTL0 寄存器的 DATAFT[1:0]位域为 0 / 1 / 2 时，该位域用作期望的 DATA，当 DATAFT[1:0]位域为 3 时，该位域用作期望的 DATA 下限值。

23.5.21. 虚拟联网模式期望数据低字 1 寄存器（CAN_PN_EDL1）

地址偏移：0xB18
复位值：0x0000 0000

该寄存器中所有位都只可在暂停模式下配置，它们在其他模式下被硬件锁定。

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DB4ELT[7:0]								DB5ELT[7:0]							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB6ELT[7:0]								DB7ELT[7:0]							
rw								rw							

位/位域	名称	描述
31:24	DB4ELT[7:0]	在虚拟联网模式下的期望数据字节 4 下限值 参考 DB3ELT[7:0]描述。
23:16	DB5ELT[7:0]	在虚拟联网模式下的期望数据字节 5 下限值 参考 DB3ELT[7:0]描述。
15:8	DB6ELT[7:0]	在虚拟联网模式下的期望数据字节 6 下限值 参考 DB3ELT[7:0]描述。
7:0	DB7ELT[7:0]	在虚拟联网模式下的期望数据字节 7 下限值 参考 DB3ELT[7:0]描述。

23.5.22. 虚拟联网模式标识符过滤器 / 期望标识符 1 寄存器（CAN_PN_IFEID1）

地址偏移：0x B1C
复位值：0x0000 0000

该寄存器中所有位都只可在暂停模式下配置，它们在其他模式下被硬件锁定。

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	IDEFD	RTRFD	IDFD_EHT[28:16]												
	rw	rw							rw						

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

位/位域	名称	描述
31	保留	必须保持复位值。
30	IDEFD	在虚拟联网模式下的 IDE 过滤数据 0：不关心该位 1：参与比较
29	RTRFD	在虚拟联网模式下的 RTR 过滤数据 0：不关心该位 1：参与比较
28:0	IDFD_EHT[28:0]	在虚拟联网模式下的ID过滤数据 / 期望的ID上限值 **ID过滤数据**（当CAN_PN_CTL0寄存器的IDFT[1:0]位域为0时）： 0：不关心该位 1：参与比较 **ID期望上限值**（当IDFT[1:0]位域为3时）。 **保留**（当IDFT[1:0]位域为1或者2时）。 对于扩展格式帧，使用所有29位。 对于标准格式帧，使用位18到28。

23.5.23. 虚拟联网模式数据 0 过滤器 / 期望数据高字 0 寄存器（CAN_PN_DF0EDH0）

地址偏移：0xB20
复位值：0x0000 0000

该寄存器中所有位都只可在暂停模式下配置，它们在其他模式下被硬件锁定。

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DB0FD_EHT[7:0]								DB1FD_EHT[7:0]							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB2FD_EHT[7:0]								DB3FD_EHT[7:0]							
rw								rw							

位/位域	名称	描述
31:24	DB0FD_EHT[7:0]	在虚拟联网模式下的数据字节0过滤数据 / 数据字节0期望上限值 参考DB3FD_EHT[7:0]描述。

23:16	DB1FD_EHT[7:0]	在虚拟联网模式下的数据字节1过滤数据 / 数据字节1期望上限值 参考DB3FD_EHT[7:0]描述。
15:8	DB2FD_EHT[7:0]	在虚拟联网模式下的数据字节2过滤数据 / 数据字节2期望上限值 参考DB3FD_EHT[7:0]描述。
7:0	DB3FD_EHT[7:0]	在虚拟联网模式下的数据字节3过滤数据 / 数据字节3期望上限值 **数据字节3过滤数据**（当CAN_PN_CTL0寄存器的DATAFT[1:0]位域为0时）： 0：不关心该位 1：参与比较 **数据字节3期望上限值**（当DATAFT[1:0]位域为3时）。 **保留**（当DATAFT[1:0]位域为1或者2时）。

23.5.24. 虚拟联网模式数据 1 过滤器 / 期望数据高字 1 寄存器（CAN_PN_DF1EDH1）

地址偏移：0xB24
复位值：0x0000 0000

该寄存器中所有位都只可在暂停模式下配置，它们在其他模式下被硬件锁定。

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DB4FD_HTF[7:0]								DB5FD_HTF[7:0]							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB6FD_HTF[7:0]								DB7FD_HTF[7:0]							
rw								rw							

位/位域	名称	描述
31:24	DB4FD_HTF[7:0]	在虚拟联网模式下的数据字节4过滤数据 / 数据字节4期望上限值 参考DB3FD_EHT[7:0]描述。
23:16	DB5FD_HTF[7:0]	在虚拟联网模式下的数据字节5过滤数据 / 数据字节5期望上限值 参考DB3FD_EHT[7:0]描述。
15:8	DB6FD_HTF[7:0]	在虚拟联网模式下的数据字节6过滤数据 / 数据字节6期望上限值 参考DB3FD_EHT[7:0]描述。
7:0	DB7FD_HTF[7:0]	在虚拟联网模式下的数据字节7过滤数据 / 数据字节7期望上限值 参考DB3FD_EHT[7:0]描述。

23.5.25. 虚拟联网模式接收唤醒邮箱 x 控制状态信息寄存器（CAN_PN_RWMxCS）（x=0..3）

地址偏移：0xB40 + 16 * x
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									RSRR	RIDE	RRTR	RDLC[3:0]			
									r	r	r	r			

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
31:23	保留	必须保持复位值。
22	RSRR	接收到的SRR位
21	RIDE	接收到的IDE位 0：帧为标准格式 1：帧为扩展格式
20	RRTR	接收到的RTR位 0：帧为数据帧 1：帧为远程帧
19:16	RDLC[3:0]	接收到的DLC域 该位域指示了有效的数据字节长度。
15:0	保留	必须保持复位值。

23.5.26. 虚拟联网模式接收唤醒邮箱 x 标识符寄存器（CAN_PN_RWMxI）（x=0..3）

地址偏移：0xB44 + 16 * x

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			RID[28:16]												
									r						

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

位/位域	名称	描述
31:29	保留	必须保持复位值。
28:0	RID[28:16]	接收到的ID域 对于扩展格式帧，使用这29位用于ID存储。 对于标准格式帧，使用位18到位28用于ID存储。

23.5.27. 虚拟联网模式接收唤醒邮箱 x 数据 0 寄存器（CAN_PN_RWMxD0）（x=0..3）

地址偏移：0xB48 + 16 * x
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
RDB0[7:0]								RDB1[7:0]							
r								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDB2[7:0]								RDB3[7:0]							
r								r							

位/位域	名称	描述
31:24	RDB0[7:0]	接收到的数据字节0
23:16	RDB1[7:0]	接收到的数据字节 1
15:8	RDB2[7:0]	接收到的数据字节 2
7:0	RDB3[7:0]	接收到的数据字节 3

23.5.28. 虚拟联网模式接收唤醒邮箱 x 数据 1 寄存器（CAN_PN_RWMxD1）（x=0..3）

地址偏移：0xB4C + 16 * x
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
RDB4[7:0]								RDB5[7:0]							
r								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDB6[7:0]								RDB7[7:0]							
r								r							

位/位域	名称	描述
31:24	RDB4[7:0]	接收到的数据字节 4
23:16	RDB5[7:0]	接收到的数据字节 5
15:8	RDB6[7:0]	接收到的数据字节 6
7:0	RDB7[7:0]	接收到的数据字节 7

23.5.29. FD 控制寄存器（CAN_FDCTL）

地址偏移：0xC00
复位值：0x8000 0101

该寄存器中位 17:16，15，12:8 只可在暂停模式下配置，它们在其他模式下被硬件锁定。

该寄存器不会被 CAN_CTL0 寄存器中的软件复位 SWRST 位复位。

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
BRSEN	保留													MDSZ[1:0]	
rw														rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDCEN	TDCS	保留	TDCO[4:0]					保留		TDCV[5:0]					
rw	rc_w1				rw							r			

位/位域	名称	描述
31	BRSEN	数据阶段波特率切换使能 0：不切换波特率 1：当发送邮箱中的 BRS 位为隐形位‘1’时，位速率需要在数据阶段从正常波特率切换到预先设置的数据波特率。
30:18	保留	必须保持复位值。
17:16	MDSZ[1:0]	邮箱数据大小 00：每个邮箱8个字节数据 01：每个邮箱16个字节数据 10：每个邮箱32个字节数据 11：每个邮箱 64 个字节数据
15	TDCEN	传输延迟补偿使能 **注意：**在回环静默模式下必须关闭传输延迟补偿功能。 0：禁能传输延迟补偿 1：使能传输延迟补偿
14	TDCS	传输延迟补偿状态 当该位置位时，表示传输延迟超出补偿了补偿范围，无法正确地补偿传输延迟用于位校验。 0：传输延迟在补偿范围内 1：传输延迟超出补偿范围
13	保留	必须保持复位值。
12:8	TDCO[4:0]	传输延迟补偿偏置 这些位被用于设置当FD帧BRS位为隐性位时的次级采样点（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。
7:6	保留	必须保持复位值。
5:0	TDCV[5:0]	传输延迟补偿值 该位域由硬件设置，显示当前测量的传输延迟值与传输延迟补偿偏置之和。

23.5.30. FD 位时间寄存器（CAN_FDBT）

地址偏移：0xC04
复位值：0x0000 0000

该寄存器中所有位都只可在暂停模式下配置，它们在其他模式下被硬件锁定。

该寄存器不会被 CAN_CTL0 寄存器中的软件复位 SWRST 位复位。

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		DBAUDPSC[9:0]										保留	DSJW[2:0]		
		rw											rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	DPTS[4:0]					保留		DPBS1[2:0]			保留		DPBS2[2:0]		
	rw							rw					rw		

位/位域	名称	描述
31:30	保留	必须保持复位值。
29:20	DBAUDPSC[9:0]	数据位时间的波特率分频系数 CAN数据位时间的波特率分配系数。 数据位时间的波特率分频系数= DBAUDPSC[9:0] + 1
19	保留	必须保持复位值。
18:16	DSJW[2:0]	数据位时间的再同步补偿宽度 再同步补偿占用的时间单元数量 = DSJW[2:0] + 1
15	保留	必须保持复位值。
14:10	DPTS[4:0]	数据位时间的传播时间段 传播时间段占用的时间单元数量 = DPTS[4:0]
9:8	保留	必须保持复位值。
7:5	DPBS1[2:0]	数据位时间的相位缓冲段1 相位缓冲段 1 占用的时间单元数量 = DPBS1[2:0] + 1
4:3	保留	必须保持复位值。
2:0	DPBS2[2:0]	数据位时间的相位缓冲段 2 相位缓冲段 2 占用的时间单元数量 = DPBS2[2:0] + 1

23.5.31. 常规帧和 FD 帧 CRC 寄存器（CAN_CRCCFD）

地址偏移：0xC08
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			ANTM[4:0]					保留			CRCTCI[20:16]				
					r								r		

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

位/位域	名称	描述
31:29	保留	必须保持复位值。
28:24	ANTM[4:0]	发送 CRCTCI[20:0]值的相关联的邮箱的编号 该位域包含发送常规帧或者 FD 帧时，CRC 值为 CRCTCI[20:0]的邮箱的编号。
23:21	保留	必须保持复位值。
20:0	CRCTCI[20:0]	发送的常规帧 / FD帧的CRC计算值 对于CRC_15，使用位0到位14，其他位为0，并且该位域值与CAN_CRCC寄存器中的CRCTC[14:0]值相同。 对于CRC_17，使用位0到位16，其他位为0。 对于CRC_21，使用所有的21位。

24. 附录

24.1. 寄存器表中使用的缩写列表

表 24-1. 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。

24.2. 术语表

表 24-2. 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

24.3. 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。

25. 版本历史

表 **25-1.** 版本历史

版本号	描述	日期
1.0	初始发布	2022 年 9 月 15 日
1.1	1.修改 ***1.6.9. 系统配置寄存器 3（SYSCFG_CFG3）*** 寄存器位域描述。 2.修正 ***2.4.1. 等待状态寄存器（FMC_WS）*** 复位值。 3. 修改 ***2.3.1. 闪存结构***章节中不同 flash 大小时 data flash 大小。 4. 对不同情况下 ***2.4.12. EP 计数寄存器（FMC_EPCNT）***复位值说明。 5.修改 ***7.5.4. ADC1 触发选择寄存器（TRIGSEL_ADC1）*** 描述。	2022 年 12 月 19 日
1.2	1.修改***表 15 1. DAC 引脚*** DAC 的参考电压输入范围，由 2.6V 改为 2.7V。 2.修正 ***1.4. 引导配置***描述中“系统存储器”中的起始地址。 3. 在 ***3.3.4. 省电模式***章节中注意中添加在“进入深度睡眠模式前，需置位 FMC_WS 寄存器中的 SLEEP_SLP 位”。 4. 修改 ***2.3.2. 错误检查与纠正（ECC）***章节的描述内容。 5.修改 ***11.4.1. DMA 操作***章节的描述内容。 6. 修改 ***1.6.1. 系统配置寄存器 0（SYSCFG_CFG0）*** 中 BOOT0_PF0_RMP 位域说明。	2023 年 6 月 27 日
1.3	1.修改 ***20.2 主要特征*** 第五条。 2.在 ***17.3. 功能描述***章节中添加“注意：在 48-pin 和 32-pin 封装中不支持使用 LXTAL 作为 RTC 时钟源”。 3. 在 ***23.3.8. CAN FD 操作***章节传输延迟补偿小节中，修正相关公式与解释。 4. 在 ***9.4.7. 典型应用配置***章节中增加描述，MFCOM 模拟 SPI 工作在 LSB 模式时，从 MFCOM_SBUFBYSx 中读取时接收的 16 位数据，需要把收到数据的高 8 位和低 8 位调换一下以得到实际的数据。 5.修改 ***12.2. 主要特征***第六条。 6. ADC\DAC\CMP 章节内容有改动。	2024 年 1 月 19 日
1.4	1.修改 ***18.2.5. TIMERx 寄存器（x=1）***章节 TIMERx_IRMP 寄存器 CI0_RMP 位域取值 11 时的描述，由 CKOUT0SEL 改为 CK_OUT。 2. 删除***图 3 2. 上电/掉电复位波形图***tRSTTEMPO 具体参数，应以数据手册为准。 3. 在 ***3.3.4 深度睡眠模式***章节，添加 SRAM0	2024 年 8 月 16 日

版本号	描述	日期
	（0KB~16KB）数据保持描述的说明。 4. 在 ***1.3 存储映射***章节存储器映射表添加说明，同一个块 sram 映射到不同逻辑地址。 5. 在 ***2. 闪存控制器（FMC）***章节中修改描述，由“1K 字节的缓存，组织为 2X64 位的 64 缓存线”改为“1K 字节的缓存，组织为 4X64 位的 32 缓存线”。 6. 修改 ***23.3.6. 数据接收***章节 CAN 接收的匹配过程描述。 7.修改 ***22. 比较器（CMP）*** 章节，将“比较器输出必须连接到对应的复用 I/O 口”改为“比较器输出输出可以通过 GPIO 的备用功能连接到对应的 I / O 口”。 8.在***图 22 2. 比较器迟滞***中比较器迟滞修改为单边迟滞。 9. 在 ***23.3.10. 通信参数***章节增加关于 CAN 位时间的 CANCLK 时钟数目与 CAN 邮箱数目之间的限制关系 10. 公式（14-1）中“V30-Vtemperature”改为“Vtemperature -V30”。 11. MFCOM 章节内容有改动。 12. 添加 GD32A503xxxx0E、GD32A503xxxx1E 启动方式说明在 ***1.4. 引导配置***章节。 13.修改系统框图中 DMA1 通道数。	
1.5	1. 在 ***2. 闪存控制器（FMC）***章节和***表 1 2. GD32A50x 系列器件的存储器映射表***删除 EEPROM 相关内容。 2. 修改***图 18 8. 中央对齐计数模式计数器时序图***中 TIMERx_CTL0 CAM=2’b10(downcount only)为 TIMERx_CTL0 CAM=2’b01(downcount only)。 3. 在 ***5.3.10. 复位源/时钟寄存器（RCU_RSTSCK）***章节，修改 RCU_RSTSCK 寄存器 bit19 LVDRSTF 描述，新增 note。 4. 更新***图 3 1. 电源域概览***，将 POR/PDR、BOR 所在电源域及监测域从 VDD 域修改为 VDDA 域 5. 更新 ***3.3.2. VDD / VDDA 电源域***，将 POR/PDR、BOR 描述从 VDD 域移动到 VDDA 域。 6. 在 ***2. 闪存控制器（FMC）***章节删除从 SRAM 启动的相关描述。 7. 修改 ***1.4. 引导配置章节***内容并删除从 SRAM 启动的相关描述。 8. 在 ***1.6.1. 系统配置寄存器 0（SYSCFG_CFG0）***章节修改 BOOT_MODE 位域描述。	2024 年 11 月 26 日
1.6	1. 在 ***23.5.13. 位时间寄存器（CAN_BT）***章节增加 BAUDPSC[9:0]波特率分频值的描述“波特率分频系数= BAUDPSC[9:0] + 1”。 2. 在 ***23.5.30. FD 位时间寄存器（CAN_FDBT）***章节	2025 年 2 月 17 日

版本号	描述	日期
	增加 DBAUDPSC[9:0]波特率分频值的描述“数据位时间的波特率分频系数= DBAUDPSC[9:0] + 1”。 3. 在 ***23.3.9. 错误和状态***章节总线集成状态小节增加注意“建议保持 EFDIS 位为 0 来使能边沿滤波，以避免总线空闲条件被误检测。”。 4. 在 ***23.2. 主要特征*** 章节增加 CANCLK 可配置的特征以及 CANCLK 的注意“CANCLK 时钟源的选择有使用限制，请查看 GD32A50x 系列的勘误手册。”。 5. 删除主闪存 64K 相关描述：删除***表 2 4. 64KB 闪存基地址和大小***，删除***表 2 7. 选项字节 1*** 中主闪存为 64k 时 EFALC[3:0]配置。 6. 在 ***9.4.3. 定时器*** 章节，修改“当定时器起始位被使能时，定时器将在移位时钟的第一个上升沿上进行比较，并重新加载比较寄存器”为“当定时器起始位被使能时，定时器将在移位时钟的第一个上升沿上重新加载比较寄存器。” 7. 在***表 14 2. ADC 引脚定义***中，删除“2.7 V ≤ VDDA ≤5V 2.7 V ≤ VREFP ≤ VDDA VREFN = VSSA” 。 8. 在 ***14.2. 主要特征***，删除 ADC 供电要求：2.7V 到 5.5V，一般电源电压为 5V。 9.在 ***23.3.10. 通信参数***章节，修改内部计数器时钟源小节的描述。 10. 修改 ***3. 电源管理单元（PMU）章节***描述， （1） ***3.1 简介***、***3.2 主要特征***小节：删除备份域内容、电源域个数改为两个 （2）***图 3-1. 电源域概览***：删除 SLEEPING、VBAK 信号；删除 BPOR；在 VDD 域中增加 Backup domain、RTC、BREG、LXTAL 和 BAP PAD；将 HXTAL 改到 VDDA 域；PLLs 改为 PLL; （3）***3.3.1.VDD / VDDA 电源域***小节：VDD 域描述增加备份域、LXTAL；VDDA 域增加 HXTAL、PLLs 改为 PLL; （4） 修改***电池备份域***小节内容描述：删除 VBAK、BPOR 相关内容；将 Backup domain 小节放到 VDD domain 小节中； （5）***3.3.2.1.1V 电源域***小节：删除备份域 （6）***VDDA 域***小节：将 VREF+ / VREF-改为 VREFP / VREFN。	

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 under the intellectual property laws and treaties of the People’s Republic of China and other jurisdictions worldwide. The Company reserves all rights under such laws and treaties and does not grant any license under its patents, copyrights, trademarks, or other intellectual property rights. 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.

The Company makes no warranty of any kind, express or implied, with regard to this document or any Product, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Company does not assume any liability arising out of the application or use of any Product described in this document. Any information provided in this document is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Except for customized products which has been expressly identified in the applicable agreement, the Products are designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only. The Products are not designed, intended, or authorized for use as components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, atomic energy control instruments, combustion control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or Product could cause personal injury, death, property or environmental damage ("Unintended Uses"). Customers shall take any and all actions to ensure using and selling the Products in accordance with the applicable laws and regulations. The Company is not liable, in whole or in part, and customers shall and hereby do release the Company as well as it’s suppliers and/or distributors from any claim, damage, or other liability arising from or related to all Unintended Uses of the Products. Customers shall indemnify and hold the Company as well as it’s 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 Products.

Information in this document is provided solely in connection with the Products. The Company reserves the right to make changes, corrections, modifications or improvements to this document and Products and services described herein at any time, without notice.

© 2025 GigaDevice – All rights reserved