GigaDevice Semiconductor Inc.

GD32G553
Arm® Cortex®-M33 32-bit MCU

用户手册

1.3 版本

（2026 年 1 月）

目录

目录 2
图索引 27
表索引 38
1. 系统及存储器架构 43
1.1. Arm® Cortex®-M33 处理器 43
1.2. 系统架构 44
1.3. 存储器映射 47
1.3.1. 片上 SRAM 存储器 52
1.3.2. 片上 FLASH 存储器概述 54
1.4. 引导配置 54
1.5. 重映射配置 55
1.6. 系统配置控制器（SYSCFG） 55
1.7. SYSCFG 寄存器 56
1.7.1. 配置寄存器 0(SYSCFG_CFG0) 56
1.7.2. 配置寄存器 1(SYSCFG_CFG1) 57
1.7.3. EXTI 源选择寄存器 0(SYSCFG_EXTISS0) 58
1.7.4. EXTI 源选择寄存器 1(SYSCFG_EXTISS1) 60
1.7.5. EXTI 源选择寄存器 2(SYSCFG_EXTISS2) 61
1.7.6. EXTI 源选择寄存器 3(SYSCFG_EXTISS3) 62
1.7.7. 配置寄存器 2(SYSCFG_CFG2) 64
1.7.8. 系统状态寄存器 (SYSCFG_STAT) 65
1.7.9. 配置寄存器 3(SYSCFG_CFG3) 67
1.7.10. 配置寄存器 4(SYSCFG_CFG4) 68
1.7.11. 配置寄存器 5(SYSCFG_CFG5) 69
1.7.12. TCMSRAM 控制与状态寄存器(SYSCFG_TCMSRAMCS) 70
1.7.13. TCMSRAM 密钥寄存器(SYSCFG_TCMSRAMKEY) 71
1.7.14. TCMSRAM 写保护寄存器(SYSCFG_TCMSRAMWP) 71
1.7.15. I/O 补偿控制寄存器(SYSCFG_CPSCTL) 72
1.7.16. TIMER 输入源选择寄存器 0 (SYSCFG_TIMERCISEL0) 72
1.7.17. TIMER 输入源选择寄存器 1 (SYSCFG_TIMERCISEL1) 74
1.7.18. TIMER 输入源选择寄存器 2 (SYSCFG_TIMERCISEL2) 75
1.7.19. TIMER 输入源选择寄存器 3 (SYSCFG_TIMERCISEL3) 77
1.7.20. TIMER 输入源选择寄存器 4 (SYSCFG_TIMERCISEL4) 79

1.7.21. TIMERx 配置寄存器 0（SYSCFG_TIMERxCFG0, x=0，1，2，3，4，7，19） 80
1.7.22. TIMERx 配置寄存器 1（SYSCFG_TIMERxCFG1, x=0，1，2，3，4，7，19） 83
1.7.23. TIMERx 配置寄存器 2（SYSCFG_TIMERxCFG2, x=0，1，2，3，4，7，19） 86
1.7.24. TIMERx 配置寄存器 0（SYSCFG_TIMERxCFG0, x=14） 87
1.7.25. TIMERx 配置寄存器 1（SYSCFG_TIMERxCFG1, x=14） 89
1.7.26. TIMERx 配置寄存器 2（SYSCFG_TIMERxCFG2, x=14） 91
1.8. 设备电子签名 92
1.8.1. 存储容量信息 92
1.8.2. 设备唯一 ID(96 位/位域) 93
2. 闪存控制器（FMC） 95
2.1. 简介 95
2.2. 主要特性 95
2.3. 功能说明 96
2.3.1. 闪存结构 96
2.3.2. 错误检查与纠正（ECC） 97
2.3.3. 读操作 98
2.3.4. 双 bank 边读边写特性 100
2.3.5. FMC_CTL/FMC_OBCTL 寄存器解锁 100
2.3.6. 页擦除 101
2.3.7. 整片擦除 102
2.3.8. 主存储闪存块编程 104
2.3.9. OTP 编程 105
2.3.10. 选项字节 106
2.3.11. 仅执行区域（DCRP） 112
2.3.12. 页擦除 / 编程保护（WP） 113
2.3.13. 安全保护（SPC） 114
2.3.14. 安全用户区域（SCR） 114
2.3.15. 禁用核心调试访问 115
2.3.16. 强制从闪存启动 115
2.3.17. FMC 中断 115
2.4. FMC 寄存器 116
2.4.1. 等待状态寄存器(FMC_WS) 116
2.4.2. 系统运行时闪存的运行模式解锁寄存器(FMC_RUNKEY) 117
2.4.3. 解锁寄存器(FMC_KEY) 118
2.4.4. 选项字节操作解锁寄存器(FMC_OBKEY) 118
2.4.5. 状态寄存器(FMC_STAT) 119
2.4.6. 控制寄存器(FMC_CTL) 120
2.4.7. ECC 控制与状态寄存器(FMC_ECCCS) 123

2.4.8. 选项字节控制寄存器(FMC_OBCTL) 124
2.4.9. DCRP0 起始地址(FMC_DCRP_SADDR0) 127
2.4.10. DCRP0 结束地址(FMC_DCRP_EADDR0) 127
2.4.11. Bank0 擦除/编程保护区域 0 寄存器(FMC_BK0WP0) 128
2.4.12. Bank0 擦除/编程保护区域 1 寄存器(FMC_BK0WP1) 129
2.4.13. DCRP1 起始地址(FMC_DCRP_SADDR1) 129
2.4.14. DCRP1 结束地址(FMC_DCRP_EADDR1) 130
2.4.15. Bank1 擦除/编程保护区域 0 寄存器(FMC_BK1WP0) 130
2.4.16. Bank1 擦除/编程保护区域 1 寄存器(FMC_BK1WP1) 131
2.4.17. Bank0 安全用户区域寄存器(FMC_BK0SCR) 132
2.4.18. Bank1 安全用户区域寄存器(FMC_BK1SCR) 132
2.4.19. 产品 ID 寄存器(FMC_PID) 133

3. 电源管理单元（PMU） 134

3.1. 简介 134

3.2. 主要特征 134

3.3. 功能说明 135

3.3.1. 备份域 136
3.3.2. V_{DD} / V_{DDA} 电源域 139
3.3.3. 1.1V 电源域 142
3.3.4. 省电模式 144

3.4. PMU 寄存器 146

3.4.1. 控制寄存器 0（PMU_CTL0） 146
3.4.2. 电源控制和状态寄存器（PMU_CS） 148
3.4.3. 控制寄存器 1（PMU_CTL1） 150
3.4.4. 控制寄存器 2（PMU_CTL2） 152
3.4.5. 控制寄存器 3（PMU_CTL3） 152

4. 复位和时钟单元（RCU） 154

4.1. 复位控制单元（RCTL） 154

4.1.1. 简介 154
4.1.2. 功能描述 154

4.2. 时钟控制单元（CCTL） 155

4.2.1. 简介 155
4.2.2. 主要特性 158
4.2.3. 功能描述 158

4.3. RCU 寄存器 162

4.3.1. 控制寄存器（RCU_CTL） 162
4.3.2. PLL 寄存器（RCU_PLL） 164

4.3.3. 时钟配置寄存器 0（RCU_CFG0） 166
4.3.4. 时钟中断寄存器（RCU_INT） 168
4.3.5. AHB1 复位寄存器（RCU_AHB1RST） 171
4.3.6. AHB2 复位寄存器（RCU_AHB2RST） 172
4.3.7. AHB3 复位寄存器（RCU_AHB3RST） 174
4.3.8. APB1 复位寄存器（RCU_APB1RST） 175
4.3.9. APB2 复位寄存器（RCU_APB2RST） 178
4.3.10. APB3 复位寄存器（RCU_APB3RST） 180
4.3.11. AHB1 使能寄存器（RCU_AHB1EN） 182
4.3.12. AHB2 使能寄存器（RCU_AHB2EN） 183
4.3.13. AHB3 使能寄存器（RCU_AHB3EN） 185
4.3.14. APB1 使能寄存器（RCU_APB1EN） 185
4.3.15. APB2 使能寄存器（RCU_APB2EN） 188
4.3.16. APB3 使能寄存器（RCU_APB3EN） 190
4.3.17. AHB1 睡眠和深度睡眠使能寄存器（RCU_AHB1SPDPEN） 192
4.3.18. AHB2 睡眠和深度睡眠使能寄存器（RCU_AHB2SPDPEN） 194
4.3.19. AHB3 睡眠和深度睡眠使能寄存器（RCU_AHB3SPDPEN） 195
4.3.20. APB1 睡眠和深度睡眠使能寄存器（RCU_APB1SPDPEN） 196
4.3.21. APB2 睡眠和深度睡眠使能寄存器（RCU_APB2SPDPEN） 199
4.3.22. APB3 睡眠和深度睡眠使能寄存器（RCU_APB3SPDPEN） 201
4.3.23. 备份域控制寄存器（RCU_BDCTL） 203
4.3.24. 复位源/时钟寄存器（RCU_RSTSCK） 205
4.3.25. 时钟配置寄存器 1（RCU_CFG1） 207
4.3.26. 时钟配置寄存器 2（RCU_CFG2） 209

5. 中断/事件控制器（EXTI） 212

5.1. 简介 212

5.2. 主要特性 212

5.3. 功能说明 212

5.4. 外部中断及事件（EXTI）框图 217

5.5. 外部中断及事件功能概述 217

5.6. EXTI 寄存器 220

5.6.1. 中断使能寄存器 0（EXTI_INTEN0） 220
5.6.2. 事件使能寄存器 0（EXTI_EVEN0） 220
5.6.3. 上升沿触发使能寄存器 0（EXTI_RTEN0） 221
5.6.4. 下降沿触发使能寄存器 0（EXTI_FTEN0） 221
5.6.5. 软件中断事件寄存器 0（EXTI_SWIEV0） 221
5.6.6. 挂起寄存器 0（EXTI_PD0） 222
5.6.7. 中断使能寄存器 1（EXTI_INTEN1） 222

5.6.8. 事件使能寄存器 1（EXTI_EVEN1） 223
5.6.9. 上升沿触发使能寄存器 1（EXTI_RTEN1） 223
5.6.10. 下降沿触发使能寄存器 1（EXTI_FTEN1） 224
5.6.11. 软件中断事件寄存器 1（EXTI_SWIEV1） 224
5.6.12. 挂起寄存器 1（EXTI_PD1） 225
6. 触发选择控制器（TRIGSEL） 226
6.1. 简介 226
6.2. 主要特征 226
6.3. 功能说明 226
6.4. 内部连接 227
6.5. TRIGSEL 寄存器 240
6.5.1. EXTOUT 触发选择寄存器 0（TRIGSEL_EXTOUT_0） 240
6.5.2. EXTOUT 触发选择寄存器 1（TRIGSEL_EXTOUT_1） 240
6.5.3. EXTOUT 触发选择寄存器 2（TRIGSEL_EXTOUT_2） 241
6.5.4. EXTOUT3 触发选择寄存器 3（TRIGSEL_EXTOUT_3） 242
6.5.5. ADC0 触发选择寄存器（TRIGSEL_ADC0） 242
6.5.6. ADC1 触发选择寄存器（TRGSEL_ADC1） 243
6.5.7. ADC2 触发选择寄存器（TRIGSEL_ADC2） 244
6.5.8. ADC3 触发选择寄存器（TRIGSEL_ADC3） 244
6.5.9. TIMER0_BRKIN 触发选择寄存器（TRIGSEL_TIMER0BRKIN） 245
6.5.10. TIMER7_BRKIN 触发选择寄存器（TRIGSEL_TIMER7BRKIN） 246
6.5.11. TIMER14_BRKIN 触发选择寄存器（TRIGSEL_TIMER14BRKIN） 247
6.5.12. TIMER15_BRKIN 触发选择寄存器（TRIGSEL_TIMER15BRKIN） 247
6.5.13. TIMER16_BRKIN 触发寄存器（TRIGSEL_TIMER16BRKIN） 248
6.5.14. TIMER19_BRKIN 触发选择寄存器（TRIGSEL_TIMER19BRKIN） 249
6.5.15. CAN0 触发选择寄存器（TRIGSEL_CAN0） 249
6.5.16. CAN1 触发选择寄存器（TRIGSEL_CAN1） 250
6.5.17. CAN2 触发选择寄存器（TRIGSEL_CAN2） 250
6.5.18. TIMER0_ETI 触发选择寄存器（TRIGSEL_TIMER0ETI） 251
6.5.19. TIMER1_ETI 触发选择寄存器（TRIGSEL_TIMER1ETI） 252
6.5.20. TIMER2_ETI 触发选择寄存器（TRIGSEL_TIMER2ETI） 252
6.5.21. TIMER3_ETI 触发选择寄存器（TRIGSEL_TIMER3ETI） 253
6.5.22. TIMER4_ETI 触发选择寄存器（TRIGSEL_TIMER4ETI） 253
6.5.23. TIMER7_ETI 触发选择寄存器（TRIGSEL_TIMER7ETI） 254
6.5.24. TIMER19_ETI 触发选择寄存器（TRIGSEL_TIMER19ETI） 255
6.5.25. HPDF_ITRG 触发选择控制寄存器（TRIGSEL_HPDF） 255
6.5.26. TIMER0_ITI14 触发选择寄存器（TRIGSEL_TIMER0ITI14） 256
6.5.27. TIMER1_ITI14 触发选择寄存器（TRIGSEL_TIMER1ITI14） 257

6.5.28. TIMER2_ITI14 触发选择寄存器（TRIGSEL_TIMER2ITI14） 257
6.5.29. TIMER3_ITI14 触发选择寄存器（TRIGSEL_TIMER3ITI14） 258
6.5.30. TIMER4_ITI14 触发选择寄存器（TRIGSEL_TIMER4ITI14） 258
6.5.31. TIMER7_ITI14 触发选择寄存器（TRIGSEL_TIMER7ITI14） 259
6.5.32. TIMER14_ITI14 触发选择寄存器（TRIGSEL_TIMER14ITI14） 260
6.5.33. TIMER19_ITI14 触发选择寄存器（TRIGSEL_TIMER19ITI14） 260
6.5.34. DAC0 触发选择寄存器（TRIGSEL_DAC0） 261
6.5.35. DAC1 触发选择寄存器（TRIGSEL_DAC1） 262
6.5.36. DAC2 触发选择寄存器（TRIGSEL_DAC2） 263
6.5.37. DAC3 触发选择寄存器（TRIGSEL_DAC3） 263
6.5.38. DAC0 触发选择扩展寄存器（TRIGSEL_EXTDAC0） 264
6.5.39. DAC1 触发选择扩展寄存器（TRIGSEL_EXTDAC1） 265
6.5.40. DAC2 触发选择扩展寄存器（TRIGSEL_EXTDAC2） 266
6.5.41. DAC3 触发选择扩展寄存器（TRIGSEL_EXTDAC3） 266
6.5.42. CLA 触发选择寄存器 0（TRIGSEL_CLA_0） 267
6.5.43. CLA 触发选择寄存器 1（TRIGSEL_CLA_1） 268
6.5.44. CLA 触发选择寄存器 2（TRIGSEL_CLA_2） 269
6.5.45. CLA 触发选择寄存器 3（TRIGSEL_CLA_3） 270
6.5.46. CLA 触发选择寄存器 4（TRIGSEL_CLA_4） 270

7. 通用和备用输入/输出接口（GPIO 和 AFIO） 272

7.1. 简介 272

7.2. 主要特性 272

7.3. 功能说明 272

7.3.1. GPIO 引脚配置 274
7.3.2. 外部中断及事件 274
7.3.3. 备用功能（AF） 274
7.3.4. 附加功能 275
7.3.5. 输入配置 275
7.3.6. 输出配置 275
7.3.7. 模拟配置 276
7.3.8. 备用功能（AF）配置 276
7.3.9. GPIO 锁定功能 277
7.3.10. GPIO 单周期输出翻转功能 277
7.3.11. I/O 补偿单元 278
7.3.12. 输入滤波 278

7.4. GPIO 寄存器 281

7.4.1. 端口控制寄存器（GPIOx_CTL, x=A…G） 281
7.4.2. 端口输出模式寄存器（GPIOx_OMODE, x=A…G） 283

7.4.3. 端口输出速度寄存器（GPIOx_OSPD, x=A…G） 285
7.4.4. 端口上拉/下拉寄存器（GPIOx_PUD, x=A…G） 286
7.4.5. 端口输入状态寄存器（GPIOx_ISTAT, x=A…G） 288
7.4.6. 端口输出控制寄存器（GPIOx_OCTL, x=A…G） 289
7.4.7. 端口位操作寄存器（GPIOx_BOP, x=A…G） 290
7.4.8. 端口配置锁定寄存器（GPIOx_LOCK, x=A…G） 290
7.4.9. 备用功能选择寄存器 0（GPIOx_AFSEL0, x=A…G） 291
7.4.10. 备用功能选择寄存器 1（GPIOx_AFSEL1, x=A…G） 292
7.4.11. 位清除寄存器（GPIOx_BC, x=A…G） 293
7.4.12. 端口位翻转寄存器（GPIOx_TG, x=A…G） 294
7.4.13. 输入滤波寄存器（GPIOx_IFL, x=A…G） 294
7.4.14. 输入滤波类型寄存器（GPIOx_IFTP, x=A…G） 295

8. 直接存储器访问控制器（DMA） 298

8.1. 简介 298

8.2. 主要特征 298

8.3. 结构框图 299

8.4. 功能说明 299

8.4.1. DMA 操作 299
8.4.2. 外设握手 301
8.4.3. 仲裁 301
8.4.4. 地址生成 301
8.4.5. 循环模式 302
8.4.6. 存储器到存储器模式 302
8.4.7. 通道配置 302
8.4.8. 中断 302
8.4.9. DMA 请求映射 303

8.5. DMA 寄存器 304

8.5.1. 中断标志位寄存器（DMA_INTF） 304
8.5.2. 中断标志位清除寄存器（DMA_INTC） 305
8.5.3. 通道 x 控制寄存器（DMA_CHxCTL） 305
8.5.4. 通道 x 计数寄存器（DMA_CHxCNT） 307
8.5.5. 通道 x 外设基地址寄存器（DMA_CHxPADDR） 308
8.5.6. 通道 x 存储器基地址寄存器（DMA_CHxMADDR） 308

9. DMA 请求多路复用器（DMAMUX） 310

9.1. 简介 310

9.2. 主要特征 310

9.3. 结构框图 311
9.4. 功能说明 311
9.4.1. DMAMUX 信号 312
9.4.2. DMAMUX 请求路由器 312
9.4.3. DMAMUX 请求生成器 315
9.4.4. 通道配置 315
9.4.5. 中断 316
9.4.6. DMAMUX 映射 316
9.5. DMAMUX 寄存器 322
9.5.1. 请求路由通道 x 配置寄存器（DMAMUX_RM_CHxCFG） 322
9.5.2. 请求路由通道中断标志位寄存器（DMAMUX_RM_INTF） 323
9.5.3. 请求路由通道中断标志位清除寄存器（DMAMUX_RM_INTC） 323
9.5.4. 请求生成通道 x 配置寄存器（DMAMUX_RG_CHxCFG） 324
9.5.5. 请求生成通道中断标志位寄存器（DMAMUX_RG_INTF） 325
9.5.6. 请求生成通道中断标志位清除寄存器（DMAMUX_RG_INTC） 325
10. 循环冗余校验管理单元（CRC） 327
10.1. 简介 327
10.2. 主要特征 327
10.3. 功能说明 328
10.4. CRC 寄存器 329
10.4.1. 数据寄存器（CRC_DATA） 329
10.4.2. 独立数据寄存器（CRC_FDATA） 329
10.4.3. 控制寄存器（CRC_CTL） 330
10.4.4. 初值寄存器（CRC_IDATA） 331
10.4.5. 多项式寄存器（CRC_POLY） 331
11. 可配置逻辑阵列（CLA） 332
11.1. 简介 332
11.2. 主要特性 332
11.3. 模块框图 332
11.4. 功能描述 333
11.4.1. CLA 输入信号选择器 333
11.4.2. 逻辑控制单元（LCU）控制 334
11.4.3. CLA 输出 335
11.4.4. 中断 335
11.5. CLA 寄存器 337

11.5.1. 全局控制寄存器 (CLA_GCTL) 337
11.5.2. 中断使能寄存器(CLA_INTE) 337
11.5.3. 中断标志寄存器 (CLA_INTF) 339
11.5.4. 状态寄存器 (CLA_STAT) 340
11.5.5. 信号选择寄存器 (CLAx_SIGS)(x=0..3) 341
11.5.6. LCU 控制寄存器 (CLAx_LCUCTL)(x=0..3) 342
11.5.7. 控制寄存器 (CLAx_CTL)(x=0..3) 342
12. 真随机数生成器（TRNG） 344
12.1. 简介 344
12.2. 主要特征 344
12.3. 限制 344
12.4. 功能说明 345
12.4.1. 线性反馈移位寄存器（LFSR） 345
12.4.2. 后处理 346
12.4.3. 训练单元 346
12.4.4. 输出 FIFO 346
12.4.5. 健康测试 347
12.4.6. NIST 模式状态 347
12.4.7. 操作流程 348
12.4.8. 错误标志 348
12.4.9. 低功耗配置 349
12.5. TRNG 寄存器 350
12.5.1. 控制寄存器（TRNG_CTL） 350
12.5.2. 状态寄存器（TRNG_STAT） 352
12.5.3. 数据寄存器（TRNG_DATA） 353
12.5.4. 健康测试配置寄存器（TRNG_HTCFG） 353
13. 加密处理器（CAU） 355
13.1. 简介 355
13.2. 主要特征 355
13.3. CAU 数据类型和初始化向量 356
13.3.1. 数据类型 356
13.3.2. 初始化向量 357
13.4. 加密处理器流程 357
13.4.1. DES / TDES 加密处理流程 358
13.4.2. AES 加密处理流程 362
13.5. 操作模式 370

13.6. CAU DMA 接口 371
13.7. CAU 中断 371
13.8. CAU 挂起模式 372
13.9. CAU 寄存器 374
13.9.1. 控制寄存器（CAU_CTL） 374
13.9.2. 状态寄存器 0（CAU_STAT0） 376
13.9.3. 数据输入寄存器（CAU_DI） 377
13.9.4. 数据输出寄存器（CAU_DO） 377
13.9.5. DMA 使能寄存器（CAU_DMAEN） 378
13.9.6. 中断使能寄存器（CAU_INTEN） 378
13.9.7. 状态寄存器 1（CAU_STAT1） 379
13.9.8. 中断标志寄存器（CAU_INTF） 379
13.9.9. 密钥寄存器（CAU_KEY0..3（H/L）） 380
13.9.10. 初始化向量寄存器（CAU_IV0..1（H/L）） 383
13.9.11. GCM 或 CCM 模式上下文交换寄存器 x（CAU_GCMCCMCTXSx） （x=0..7） 384
13.9.12. GCM 模式上下文交换寄存器 x（CAU_GCMCTXSx） （x=0..7） 385
14. 三角函数加速器（TMU） 386
14.1. 简介 386
14.2. 主要特征 386
14.3. 结构框图 386
14.4. 功能描述 387
14.4.1. 数据格式和配置 387
14.4.2. 浮点数据格式 388
14.4.3. 模式配置 389
14.4.4. TMU 运算挂起 400
14.4.5. 零开销 401
14.4.6. 中断和 DMA 401
14.5. TMU 寄存器 401
14.5.1. 控制和状态寄存器（TMU_CS） 401
14.5.2. 输入数据寄存器（TMU_IDATA） 404
14.5.3. 输出数据寄存器（TMU_ODATA） 405
15. 快速傅里叶变换（FFT） 406
15.1. 简介 406
15.2. 主要特性 406
15.3. 功能描述 406

15.4. 数据格式 407
15.5. 基-2 FFT 407
15.6. 蝶形运算单元 408
15.7. IFFT 模式 409
15.8. FFT SRAM 409
15.9. FFT 循环模式 409
15.10. 操作指南 409
15.11. FFT 中断 411
15.12. FFT 寄存器 412
15.12.1. 控制和状态寄存器（FFT_CSR） 412
15.12.2. 实部基地址寄存器（FFT_RESADDR） 414
15.12.3. 虚部基地址寄存器（FFT_IMSADDR） 414
15.12.4. 窗函数基地址寄存器（FFT_WSADDR） 415
15.12.5. 输出基地址寄存器（FFT_OSADDR） 415
15.12.6. 循环长度寄存器（FFT_LOOPLEN） 416
16. 调试（DBG） 417
16.1. 简介 417
16.2. JTAG/SW 功能描述 417
16.2.1. 切换 JTAG / SW 接口 417
16.2.2. 引脚分配 417
16.2.3. JTAG 418
16.2.4. 调试复位 420
16.2.5. JEDEC-106 ID code 420
16.3. 调试保持功能描述 420
16.3.1. 低功耗模式调试支持 420
16.3.2. TIMER, I2C, WWDGT, FWDGT, RTC, CAN, LPTIMER 和 HRTIMER 外设调试支持 421
16.4. DBG 寄存器 422
16.4.1. ID 寄存器（DBG_ID） 422
16.4.2. 控制寄存器 0（DBG_CTL0） 422
16.4.3. 控制寄存器 1（DBG_CTL1） 423
16.4.4. 控制寄存器 2（DBG_CTL2） 425
17. 模数转换器（ADC） 428
17.1. 简介 428
17.2. 主要特征 428

17.3. 引脚和内部信号 429

17.4. 功能描述 430

17.4.1. 前置校准功能 430
17.4.2. 双时钟域架构 431
17.4.3. ADCON 使能 431
17.4.4. 单端和差分输入通道 431
17.4.5. 常规序列 432
17.4.6. 运行模式 433
17.4.7. 转换结果阈值监测功能 435
17.4.8. 数据存储模式 436
17.4.9. 采样时间配置 437
17.4.10. 外部触发配置 437
17.4.11. DMA 请求 437
17.4.12. 溢出检测 437
17.4.13. ADC 内部通道 438
17.4.14. 电池电压检测电路 439
17.4.15. 使用 HPDF 管理转换结果 439
17.4.16. 可编程分辨率（DRES） 439
17.4.17. 片上硬件过采样 440
17.4.18. 增益模式 441
17.4.19. ADC 转换信号 441

17.5. ADC 同步模式 441

17.5.1. 独立模式 443
17.5.2. 常规并行模式 443
17.5.3. 常规跟随模式 443
17.5.4. 在 ADC 同步模式中使用 DMA 444

17.6. 中断 445

17.7. ADC 寄存器 446

17.7.1. 状态寄存器（ADC_STAT） 446
17.7.2. 控制寄存器 0（ADC_CTL0） 447
17.7.3. 控制寄存器 1（ADC_CTL1） 449
17.7.4. 看门狗高阈值寄存器 0（ADC_WDHT0） 452
17.7.5. 看门狗低阈值寄存器 0（ADC_WDLT0） 453
17.7.6. 常规序列寄存器 0（ADC_RSQ0） 453
17.7.7. 常规序列寄存器 1（ADC_RSQ1） 454
17.7.8. 常规序列寄存器 2（ADC_RSQ2） 454
17.7.9. 常规序列寄存器 3（ADC_RSQ3） 455
17.7.10. 常规序列寄存器 4（ADC_RSQ4） 456
17.7.11. 常规序列寄存器 5（ADC_RSQ5） 456

17.7.12. 常规序列寄存器 6（ADC_RSQ6） 457
17.7.13. 常规序列寄存器 7（ADC_RSQ7） 457
17.7.14. 常规序列寄存器 8（ADC_RSQ8） 458
17.7.15. 常规数据寄存器（ADC_RDATA） 459
17.7.16. 过采样控制寄存器（ADC_OVSAMPCTL） 459
17.7.17. 看门狗 1 通道选择寄存器（ADC_WD1SR） 461
17.7.18. 看门狗 2 通道选择寄存器（ADC_WD2SR） 461
17.7.19. 看门狗高阈值寄存器 1（ADC_WDHT1） 462
17.7.20. 看门狗低阈值寄存器 1（ADC_WDLT1） 462
17.7.21. 看门狗高阈值寄存器 2（ADC_WDHT2） 463
17.7.22. 看门狗低阈值寄存器 2（ADC_WDLT2） 463
17.7.23. 差分模式控制寄存器（ADC_DIFCTL） 464
17.7.24. 增益配置寄存器（ADC_GAINCFG） 464
17.7.25. 摘要状态寄存器（ADC_SSTAT） 465
17.7.26. 同步控制寄存器（ADC_SYNCCTL） 466
17.7.27. 同步常规数据寄存器（ADC_SYNCDATA） 468

18. 数模转换器（DAC） 470

18.1. 简介 470

18.2. 主要特性 470

18.3. 功能描述 472

18.3.1. DAC 使能 472
18.3.2. DAC 输出缓冲 472
18.3.3. DAC 数据配置 472
18.3.4. DAC 触发 473
18.3.5. DAC 转换 473
18.3.6. DAC 噪声波 473
18.3.7. DAC 锯齿波 474
18.3.8. DAC 输出电压 476
18.3.9. DMA 请求 476
18.3.10. DAC 并发转换 476
18.3.11. DAC 复位保持模式 477
18.3.12. DAC 输出缓冲区校准 477
18.3.13. DAC 模式 478
18.3.14. DAC 低功耗模式 480

18.4. DAC 寄存器 481

18.4.1. DACx 控制寄存器 (DAC_CTL0) 481
18.4.2. DACx 软件触发寄存器 (DAC_SWT) 484
18.4.3. DACx_OUT0 12 位右对齐数据保持寄存器 (DAC_OUT0_R12DH) 485

18.4.4. DACx_OUT0 12 位左对齐数据保持寄存器 (DAC_OUT0_L12DH) 485
18.4.5. DACx_OUT0 8 位右对齐数据保持寄存器 (DAC_OUT0_R8DH) 486
18.4.6. DACx_OUT1 12 位右对齐数据保持寄存器 (DAC_OUT1_R12DH) 486
18.4.7. DACx_OUT1 12 位左对齐数据保持寄存器 (DAC_OUT1_L12DH) 487
18.4.8. DACx_OUT1 8 位右对齐数据保持寄存器 (DAC_OUT1_R8DH) 487
18.4.9. DACx 并发模式 12 位右对齐数据保持寄存器 (DACC_R12DH) 488
18.4.10. DACx 并发模式 12 位左对齐数据保持寄存器 (DACC_L12DH) 488
18.4.11. DACx 并发模式 8 位右对齐数据保持寄存器 (DACC_R8DH) 489
18.4.12. DACx_OUT0 数据输出寄存器 (DAC_OUT0_DO) 489
18.4.13. DACx_OUT1 数据输出寄存器 (DAC_OUT1_DO) 490
18.4.14. DACx 状态寄存器 0 (DAC_STAT0) 490
18.4.15. DACx 校准寄存器 (DAC_CALR) 491
18.4.16. DACx 模式寄存器 (DAC_MDCR) 492
18.4.17. DACx 采样保持模式采样时间寄存器 0 (DAC_SKSTR0) 493
18.4.18. DACx 采样保持模式采样时间寄存器 1 (DAC_SKSTR1) 494
18.4.19. DACx 采样保持模式保持时间寄存器 (DAC_SKKTR) 494
18.4.20. DACx 采样保持模式刷新时间寄存器 (DAC_SKRTR) 495
18.4.21. DACx_OUT0 锯齿波寄存器(DAC_OUT0_SAW) 495
18.4.22. DACx_OUT1 锯齿波寄存器(DAC_OUT1_SAW) 496
18.4.23. DACx 锯齿波模式寄存器 (DAC_SAWMDR) 496

19. 比较器（CMP） 498

19.1. 简介 498

19.2. 主要特征 498

19.3. 功能描述 498

19.3.1. 比较器时钟 499
19.3.2. 比较器的 I/O 配置 499
19.3.3. 比较器迟滞 500
19.3.4. 比较器寄存器写保护 501
19.3.5. 比较器输出消隐 501
19.3.6. 电压定标器功能 502
19.3.7. 比较器中断 502
19.3.8. 复位保持功能 502

19.4. 比较器寄存器 503

19.4.1. 状态寄存器（CMP_STAT） 503
19.4.2. 中断标志位清除寄存器（CMP_IFC） 505
19.4.3. CMP0 控制状态寄存器（CMP0_CS） 506
19.4.4. CMP1 控制状态寄存器（CMP1_CS） 508
19.4.5. CMP2 控制状态寄存器（CMP2_CS） 511

19.4.6. CMP3 控制状态寄存器（CMP3_CS） 513
19.4.7. CMP4 控制状态寄存器（CMP4_CS） 515
19.4.8. CMP5 控制状态寄存器（CMP5_CS） 517
19.4.9. CMP6 控制状态寄存器（CMP6_CS） 520
19.4.10. CMP7 控制状态寄存器（CMP7_CS） 522
20. VREF 525
20.1. 简介 525
20.2. 主要特性 525
20.3. 功能说明 525
20.3.1. VREF 校准 526
20.4. VREF 寄存器 527
20.4.1. 控制状态寄存器（VREF_CS） 527
20.4.2. 校准寄存器（VREF_CALIB） 528
21. 看门狗定时器（WDGT） 529
21.1. 独立看门狗定时器（FWDGT） 529
21.1.1. 简介 529
21.1.2. 主要特征 529
21.1.3. 功能说明 529
21.1.4. FWDGT 寄存器 532
21.2. 窗口看门狗定时器（WWDGT） 536
21.2.1. 简介 536
21.2.2. 主要特征 536
21.2.3. 功能说明 536
21.2.4. WWDGT 寄存器 538
22. 实时时钟（RTC） 540
22.1. 简介 540
22.2. 主要特征 540
22.3. 功能描述 541
22.3.1. 结构框图 541
22.3.2. 时钟源和预分频 542
22.3.3. 影子寄存器 542
22.3.4. 位域可屏蔽可配置的闹钟 542
22.3.5. 可配置周期的自动唤醒定时器 543
22.3.6. RTC 初始化和配置 543
22.3.7. 读取日历 544
22.3.8. RTC 复位 546

22.3.9. RTC 移位功能 546
22.3.10. RTC 参考时钟检测 547
22.3.11. RTC 数字平滑校准 547
22.3.12. 时间戳功能 549
22.3.13. 侵入检测 549
22.3.14. 校准时钟输出 551
22.3.15. 闹钟输出 551
22.3.16. RTC 引脚配置 551
22.3.17. RTC 省电模式管理 552
22.3.18. RTC 中断 552
22.4. RTC 寄存器 554
22.4.1. 时间寄存器（RTC_TIME） 554
22.4.2. 日期寄存器（RTC_DATE） 554
22.4.3. 控制寄存器（RTC_CTL） 555
22.4.4. 状态寄存器（RTC_STAT） 558
22.4.5. 预分频寄存器（RTC_PSC） 560
22.4.6. 唤醒定时器寄存器（RTC_WUT） 561
22.4.7. 闹钟 0 时间日期寄存器（RTC_ALRM0TD） 561
22.4.8. 闹钟 1 时间日期寄存器（RTC_ALRM1TD） 563
22.4.9. 写保护钥匙寄存器（RTC_WPK） 564
22.4.10. 亚秒寄存器（RTC_SS） 564
22.4.11. 移位控制寄存器（RTC_SHIFTCTL） 565
22.4.12. 时间戳时间寄存器（RTC_TTS） 565
22.4.13. 时间戳日期寄存器（RTC_DTS） 566
22.4.14. 时间戳亚秒寄存器（RTC_SSTS） 567
22.4.15. 高精度频率补偿寄存器（RTC_HRFC） 567
22.4.16. 侵入寄存器（RTC_TAMP） 568
22.4.17. 闹钟 0 亚秒寄存器（RTC_ALRM0SS） 572
22.4.18. 闹钟 1 亚秒寄存器（RTC_ALRM1SS） 573
22.4.19. 配置寄存器（RTC_CFG） 574
22.4.20. 备份寄存器（RTC_BKPx）（x=0..31） 574
23. 定时器（TIMER） 576
23.1. 高级定时器（TIMERx, x = 0, 7, 19） 578
23.1.1. 简介 578
23.1.2. 主要特征 578
23.1.3. 结构框图 578
23.1.4. 功能描述 579
23.1.5. TIMERx 寄存器（x = 0, 7, 19） 632
23.2. 通用定时器 L0（TIMERx, x = 1, 2 ,3, 4） 706

23.2.1. 简介 706
23.2.2. 主要特征 706
23.2.3. 结构框图 707
23.2.4. 功能描述 707
23.2.5. TIMERx 寄存器（x = 1 / 2 / 3 / 4） 732
23.3. 通用定时器 L3（TIMERx, x = 14） 772
23.3.1. 简介 772
23.3.2. 主要特征 772
23.3.3. 结构框图 773
23.3.4. 功能描述 773
23.3.5. TIMERx 寄存器(x = 14) 801
23.4. 通用定时器 L4（TIMERx, x = 15, 16） 839
23.4.1. 简介 839
23.4.2. 主要特征 839
23.4.3. 结构框图 840
23.4.4. 功能描述 840
23.4.5. TIMERx 寄存器（x = 15 / 16） 857
23.5. 基本定时器（TIMERx, x = 5, 6） 887
23.5.1. 简介 887
23.5.2. 主要特征 887
23.5.3. 结构框图 887
23.5.4. 功能描述 887
23.5.5. TIMERx 寄存器（x = 5 / 6） 893
24. 低功耗定时器（LPTIMER） 899
24.1. 简介 899
24.2. 主要特征 899
24.3. 结构框图 900
24.4. 功能描述 900
24.4.1. 时钟源配置 900
24.4.2. LPTIMER 使能 902
24.4.3. 预分频器 902
24.4.4. 输入滤波 903
24.4.5. 外部输入高电平计数器 903
24.4.6. 计数器启动 904
24.4.7. 外部触发映射 904
24.4.8. 计数器运行模式 905
24.4.9. 计数器复位 907

24.4.10. 输出模式 907
24.4.11. 超时模式 908
24.4.12. 译码器模式 909
24.4.13. 寄存器更新操作 914
24.4.14. 低功耗模式 915
24.4.15. 中断 915
24.4.16. LPTIMER 调试模式 916
24.5. LPTIMER 寄存器 917
24.5.1. 中断标志寄存器（LPTIMER_INTF） 917
24.5.2. 中断标志清除寄存器（LPTIMER_INTC） 918
24.5.3. 中断使能寄存器（LPTIMER_INTEN） 920
24.5.4. 控制寄存器 0（LPTIMER_CTL0） 922
24.5.5. 控制寄存器 1（LPTIMER_CTL1） 926
24.5.6. 比较寄存器（LPTIMER_CMPV） 927
24.5.7. 计数器自动重载寄存器（LPTIMER_CAR） 928
24.5.8. 计数器寄存器（LPTIMER_CNT） 928
24.5.9. 外部输入映射寄存器（LPTIMER_EIRMP） 929
24.5.10. 输入高电平计数最大值寄存器（LPTIMER_INHLCMV） 929
25. 高分辨率定时器（HRTIMER） 931
25.1. 简介 931
25.2. 主要特征 931
25.3. 结构框图 931
25.4. 功能说明 932
25.4.1. Master_TIMER 单元 932
25.4.2. Slave_TIMERx(x=0..7)单元 940
25.4.3. DLL 校准 980
25.4.4. 突发模式 981
25.4.5. 同步输入/输出 985
25.4.6. 外部事件 986
25.4.7. 故障输入 988
25.4.8. ADC 触发 993
25.4.9. DAC 触发 996
25.4.10. 双通道触发模式 997
25.4.11. 中断 999
25.4.12. DMA 请求 1000
25.4.13. DMA 模式 1001
25.4.14. Debug 模式 1002
25.5. HRTIMER 寄存器 1003

25.5.1. Master_TIMER 寄存器 1003
25.5.2. Slave_TIMERx 寄存器(x=0..7) 1015
25.5.3. 通用寄存器 1127
26. 红外接口（IFRP） 1203
26.1. 简介 1203
26.2. 主要特性 1203
26.3. 功能描述 1203
27. 通用同步异步收发器（USART） 1205
27.1. 简介 1205
27.2. 主要特征 1205
27.3. 功能说明 1207
27.3.1. USART 帧格式 1207
27.3.2. 波特率发生 1208
27.3.3. USART 发送器 1209
27.3.4. USART 接收器 1210
27.3.5. DMA 方式访问数据缓冲区 1211
27.3.6. 硬件流控制 1213
27.3.7. 多处理器通信 1214
27.3.8. LIN 模式 1215
27.3.9. 同步通信模式 1216
27.3.10. 串行红外（IrDA SIR）编解码功能模块 1217
27.3.11. 半双工通信模式 1218
27.3.12. 智能卡（ISO7816-3）模式 1218
27.3.13. ModBus 通信 1220
27.3.14. 接收 FIFO 1220
27.3.15. 发送 FIFO 1221
27.3.16. 从 Deepsleep 模式唤醒 1221
27.3.17. USART 中断 1222
27.4. USART 寄存器 1224
27.4.1. USART 控制寄存器 0（USART_CTL0） 1224
27.4.2. USART 控制寄存器 1（USART_CTL1） 1227
27.4.3. USART 控制寄存器 2（USART_CTL2） 1230
27.4.4. USART 波特率寄存器（USART_BAUD） 1233
27.4.5. USART 保护时间和预分频器寄存器（USART_GP） 1234
27.4.6. USART 接收超时寄存器（USART_RT） 1235
27.4.7. USART 请求寄存器（USART_CMD） 1235
27.4.8. USART 状态寄存器（USART_STAT） 1236

27.4.9. USART 中断标志清除寄存器（USART_INTC） 1241
27.4.10. USART 数据接收寄存器（USART_RDATA） 1242
27.4.11. USART 数据发送寄存器（USART_TDATA） 1243
27.4.12. USART 兼容性控制寄存器（USART_CHC） 1243
27.4.13. USART FIFO 控制和状态寄存器（USART_FCS） 1244
28. 内部集成电路总线接口（I2C） 1248
28.1. 简介 1248
28.2. 主要特征 1248
28.3. 功能说明 1248
28.3.1. 时钟要求 1249
28.3.2. I2C 通讯流程 1250
28.3.3. 噪声滤波器 1252
28.3.4. I2C 时序配置 1252
28.3.5. I2C 复位 1254
28.3.6. 数据传输 1255
28.3.7. I2C 从机模式 1257
28.3.8. I2C 主机模式 1262
28.3.9. SMBus 支持 1267
28.3.10. SMBus 模式 1270
28.3.11. 从省电模式唤醒 1271
28.3.12. DMA 模式下数据传输 1271
28.3.13. I2C 错误和中断 1272
28.3.14. I2C 调试模式 1272
28.4. I2C 寄存器 1273
28.4.1. 控制寄存器 0（I2C_CTL0） 1273
28.4.2. 控制寄存器 1（I2C_CTL1） 1275
28.4.3. 从机地址寄存器 0（I2C_SADDR0） 1277
28.4.4. 从机地址寄存器 1（I2C_SADDR1） 1278
28.4.5. 时序寄存器（I2C_TIMING） 1279
28.4.6. 超时寄存器（I2C_TIMEOUT） 1280
28.4.7. 状态寄存器（I2C_STAT） 1281
28.4.8. 状态清除寄存器（I2C_STATC） 1284
28.4.9. PEC 寄存器（I2C_PEC） 1285
28.4.10. 接收数据寄存器（I2C_RDATA） 1285
28.4.11. 发送数据寄存器（I2C_TDATA） 1286
28.4.12. 控制寄存器 2（I2C_CTL2） 1286
29. 控制器局域网络（CAN） 1288
29.1. 简介 1288

29.2. 主要特征 1288

29.3. 功能说明 1289

29.3.1. 邮箱描述符 1290
29.3.2. 接收 FIFO 描述符 1295
29.3.3. 通信模式 1300
29.3.4. 省电模式 1302
29.3.5. 数据发送 1303
29.3.6. 数据接收 1306
29.3.7. 在虚拟联网模式下的数据接收 1313
29.3.8. CAN FD 操作 1316
29.3.9. 错误和状态 1318
29.3.10. 通信参数 1321
29.3.11. 中断 1323

29.4. 典型的 CAN 配置流程示例 1324

29.5. CAN 寄存器 1326

29.5.1. 控制寄存器 0（CAN_CTL0） 1326
29.5.2. 控制寄存器 1（CAN_CTL1） 1329
29.5.3. 计数器寄存器（CAN_TIMER） 1330
29.5.4. 接收邮箱公有过滤寄存器（CAN_RMPUBF） 1331
29.5.5. 错误寄存器 0（CAN_ERR0） 1331
29.5.6. 错误寄存器 1（CAN_ERR1） 1332
29.5.7. 中断使能寄存器（CAN_INTEN） 1335
29.5.8. 状态寄存器（CAN_STAT） 1336
29.5.9. 控制寄存器 2（CAN_CTL2） 1337
29.5.10. 常规帧 CRC 寄存器（CAN_CRCC） 1339
29.5.11. 接收 FIFO 共有过滤寄存器（CAN_RFIFOPUBF） 1340
29.5.12. 接收 FIFO 标识符过滤元素匹配序号寄存器（CAN_RFIFOIFMN） 1340
29.5.13. 位时间寄存器（CAN_BT） 1341
29.5.14. 接收 FIFO/邮箱私有过滤 x 寄存器（CAN_RFIFOMPFx）（x=0..31） 1342
29.5.15. 虚拟联网模式控制寄存器 0（CAN_PN_CTL0） 1342
29.5.16. 虚拟联网模式超时寄存器（CAN_PN_TO） 1344
29.5.17. 虚拟联网模式状态寄存器（CAN_PN_STAT） 1344
29.5.18. 虚拟联网模式期望标识符 0 寄存器（CAN_PN_EID0） 1345
29.5.19. 虚拟联网模式期望 DLC 寄存器（CAN_PN_EDLC） 1346
29.5.20. 虚拟联网模式期望数据低字 0 寄存器（CAN_PN_EDL0） 1346
29.5.21. 虚拟联网模式期望数据低字 1 寄存器（CAN_PN_EDL1） 1347
29.5.22. 虚拟联网模式标识符过滤器 / 期望标识符 1 寄存器（CAN_PN_IFEID1） 1348
29.5.23. 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） 1348
29.5.24. 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） 1349

29.5.25. 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） 1350
29.5.26. 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） 1350
29.5.27. 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） 1351
29.5.28. 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） 1351
29.5.29. FD 控制寄存器（CAN_FDCTL） 1352
29.5.30. FD 位时间寄存器（CAN_FDBT） 1353
29.5.31. 常规帧和 FD 帧 CRC 寄存器（CAN_CRCCFD） 1354

30. 串行外设接口（SPI） 1356

30.1. 简介 1356

30.2. 主要特性 1356

30.2.1. SPI 主要特性 1356

30.3. SPI 功能说明 1357

30.3.1. SPI 结构框图 1357
30.3.2. SPI 信号线描述 1357
30.3.3. SPI 时序和数据帧格式 1358
30.3.4. 独立发送和接收缓冲区 1359
30.3.5. NSS 功能 1361
30.3.6. SPI 运行模式 1362
30.3.7. DMA 功能 1371
30.3.8. CRC 功能 1371
30.3.9. SPI 中断 1372

30.4. SPI 寄存器 1374

30.4.1. 控制寄存器 0（SPI_CTL0） 1374
30.4.2. 控制寄存器 1（SPI_CTL1） 1376
30.4.3. 状态寄存器（SPI_STAT） 1378
30.4.4. 数据寄存器（SPI_DATA） 1379
30.4.5. CRC 多项式寄存器（SPI_CRCPOLY） 1380
30.4.6. 接收 CRC 寄存器（SPI_RCRC） 1380
30.4.7. 发送 CRC 寄存器（SPI_TCRC） 1381
30.4.8. SPI0 四线 SPI 控制寄存器（SPI_QCTL） 1382

31. 四线 SPI 接口（QSPI） 1383

31.1. 简介 1383

31.2. 主要特征 1383

31.3. 功能描述 1383

31.3.1. QSPI 结构框图 1383
31.3.2. QSPI 命令格式 1384
31.3.3. QSPI 信号线的模式 1385

31.3.4. QSPI DDR 模式 1386
31.3.5. DQS 信号 1387
31.3.6. CSN 和 SCK 的行为 1387
31.4. 操作模式 1388
31.4.1. 普通模式 1388
31.4.2. 读轮询模式 1389
31.4.3. 内存映射模式 1390
31.5. QSPI 配置 1390
31.5.1. Flash 配置 1390
31.5.2. IP 配置 1390
31.6. 只发送一次指令 1391
31.7. 错误和中断 1391
31.8. QSPI 寄存器 1392
31.8.1. 控制寄存器（QSPI_CTL） 1392
31.8.2. 设备配置寄存器（QSPI_DCFG） 1394
31.8.3. 状态寄存器（QSPI_STAT） 1396
31.8.4. 状态清除寄存器（QSPI_STATC） 1397
31.8.5. 数据长度寄存器（QSPI_DTLEN） 1398
31.8.6. 传输配置寄存器（QSPI_TCFG） 1398
31.8.7. 地址寄存器（QSPI_ADDR） 1401
31.8.8. 交替字节寄存器（QSPI_ALTE） 1401
31.8.9. 数据寄存器（QSPI_DATA） 1402
31.8.10. 状态屏蔽寄存器（QSPI_STATMK） 1402
31.8.11. 状态匹配寄存器（QSPI_STATMATCH） 1403
31.8.12. 间隔寄存器（QSPI_INTERVAL） 1403
31.8.13. 超时寄存器（QSPI_TMOUT） 1404
31.8.14. FIFO 刷新寄存器（QSPI_FLUSH） 1404
32. 时钟相位延迟模块（CPDM） 1405
32.1. 简介 1405
32.2. 主要特征 1405
32.3. 功能说明 1405
32.3.1. 概述 1406
32.3.2. 操作流程 1407
32.4. CPDM 寄存器 1409
32.4.1. 控制寄存器（CPDM_CTL） 1409
32.4.2. 配置寄存器（CPDM_CFG） 1409

33. 外部存储器控制器（EXMC）.....1411
33.1. 简介.....1411
33.2. 主要特性.....1411
33.3. 功能说明.....1411
33.3.1. 结构框图.....1411
33.3.2. EXMC 访问基本规范.....1412
33.3.3. 外部设备地址映射.....1413
33.3.4. NOR / PSRAM 控制器.....1414
33.4. EXMC 寄存器.....1438
33.4.1. NOR / PSRAM 控制器寄存器.....1438
34. 高性能数字滤波器（HPDF）.....1445
34.1. 简介.....1445
34.2. 主要特性.....1445
34.3. 功能描述.....1446
34.3.1. HPDF 结构框图.....1446
34.3.2. HPDF 开关控制.....1447
34.3.3. HPDF 时钟.....1447
34.3.4. 复用串行数据通道.....1448
34.3.5. 并行数据输入.....1455
34.3.6. 规则组转换.....1457
34.3.7. 注入组转换.....1458
34.3.8. 数字滤波器.....1459
34.3.9. 积分器.....1460
34.3.10. 阈值监视器.....1460
34.3.11. 故障监视器.....1462
34.3.12. 极值监视器.....1463
34.3.13. 数据单元.....1463
34.3.14. HPDF 中断.....1464
34.4. HPDF 寄存器.....1466
34.4.1. HPDF 通道 x 寄存器（x = 0...7）.....1466
34.4.2. HPDF 滤波器 y 寄存器（y = 0...3）.....1472
35. FAC 滤波算法加速器（FAC）.....1486
35.1. 简介.....1486
35.2. 主要特征.....1486
35.3. 功能描述.....1486

35.3.1. 基本描述 1486
35.3.2. 本地存储和缓冲区 1487
35.3.3. 输入缓冲区 1488
35.3.4. 输出缓冲区 1490
35.3.5. 初始化函数 1492
35.3.6. 滤波器函数 1493
35.3.7. 定点数据格式 1494
35.3.8. 浮点数据格式 1495
35.3.9. FIR 滤波器 1495
35.3.10. IIR 滤波器 1496
35.4. FAC 寄存器 1498
35.4.1. FAC X0 缓冲区配置寄存器（FAC_X0BCFG） 1498
35.4.2. FAC X1 缓冲区配置寄存器（FAC_X1BCFG） 1498
35.4.3. FAC Y 缓冲区配置寄存器（FAC_YBCFG） 1499
35.4.4. FAC 参数配置寄存器（FAC_PARACFG） 1500
35.4.5. FAC 控制寄存器（FAC_CTL） 1501
35.4.6. FAC 状态寄存器（FAC_STAT） 1502
35.4.7. FAC 写数据寄存器（FAC_WDATA） 1503
35.4.8. FAC 读数据寄存器（FAC_RDATA） 1504
36. 附录 1505
36.1. 寄存器表中使用的缩写列表 1505
36.2. 术语表 1505
36.3. 可用外设 1506
37. 版本历史 1507

图索引

图 1-1. Cortex®-M33 处理器结构框图 44
图 1-2. GD32G553 系列器件的系统架构示意图 46
图 1-3. ECC 解码器示意图 52
图 2-1. 页擦除操作流程 102
图 2-2. 整片擦除操作流程 103
图 2-3. 双字编程操作流程 105
图 3-1. 电源域概览 135
图 3-2. 1.1V 电源域概览 136
图 3-3. 备用域电压阈值的波形 138
图 3-4. 温度阈值监测 139
图 3-5. 上电 / 掉电复位波形图 140
图 3-6. BOR 波形图 140
图 3-7. LVD 阈值波形图 141
图 3-8. VAVD 阈值监测波形图 142
图 3-9. VOVD 波形 143
图 3-10. VUVD 波形 143
图 4-1. 系统复位电路 155
图 4-2. 时钟树 156
图 4-3. HXTAL 时钟源 158
图 4-4. 旁路模式下 HXTAL 时钟源 159
图 5-1. EXTI 框图 217
图 6-1. TRIGSEL 主要组成示例 227
图 7-1. 标准 I/O 端口位的基本结构 273
图 7-2. 浮空/上拉/下拉输入配置 275
图 7-3. 输出配置 276
图 7-4. 模拟高阻配置 276
图 7-5. 备用功能配置 277
图 7-6. 使用采样窗口进行过滤 278
图 7-7. 输入滤波时钟周期 280
图 8-1. DMA 结构框图 299
图 8-2. 握手机制 301
图 8-3. DMA 中断逻辑图 303
图 9-1. DMAMUX 结构框图 311
图 9-2. 同步模式 313
图 9-3. 通道事件输出 314
图 10-1. CRC 计算单元框图 327
图 11-1. CLA 模块框图 332

图 11-2. CLA 中断逻辑 336
图 12-1. TRNG 模块框图 345
图 13-1. DATAM 不交换/半字交换 356
图 13-2. DATATM 字节交换/位交换 357
图 13-3. CAU 框图 358
图 13-4. DES / TDES ECB 加密 359
图 13-5. DES / TDES ECB 解密 360
图 13-6. DES / TDES CBC 加密 361
图 13-7. DES / TDES CBC 解密 362
图 13-8. AES ECB 加密 363
图 13-9. AES ECB 解密 364
图 13-10. AES CBC 加密 365
图 13-11. AES CBC 解密 366
图 13-12. 计数器块结构 366
图 13-13. AES CTR 加密/解密 367
图 14-1. TMU 模块结构框图 386
图 15-1. FFT 模块框图 406
图 15-2. 8 点 DIF 的 FFT 流程图 408
图 15-3. 蝶形运算框图 408
图 16-1. JTAG 模块框图 418
图 17-1. ADC 模块框图 430
图 17-2. 单次运行模式 433
图 17-3. 连续运行模式 433
图 17-4. 扫描转换模式，且连续转换模式失能 434
图 17-5. 扫描转换模式，连续转换模式使能 435
图 17-6. 间断运行模式 435
图 17-7. 12 位数据存储模式 436
图 17-8. 6 位数据存储模式 436
图 17-9. HFDF 与 ADC 模块握手信号示意图 439
图 17-10. 12 位 ADC 右移 5 位和取整的数例 440
图 17-11. 常规序列连续转换模式下的 ADC 转换信号 441
图 17-12. ADC 同步框图 442
图 17-13. 基于 16 个通道的常规并行模式 443
图 17-14. 一个采用连续转换模式通道上的常规跟随模式 444
图 18-1. DAC 结构框图 471
图 18-2. DAC LFSR 算法 474
图 18-3. DAC 三角噪声模式生成的波形 474
图 18-4. DAC 锯齿波（SAWDIRx = 1） 475
图 18-5. DAC 锯齿波（SAWDIRx = 0） 476
图 18-6. DAC 采样保持 479

图 19-1.比较器框图 499
图 19-2. 比较器迟滞 501
图 19-3. 比较器的输出消隐 502
图 20-1. VREF 连接 525
图 21-1. 独立看门狗定时器框图 530
图 21-2. 窗口看门狗定时器框图 536
图 21-3. 窗口看门狗定时器时序图 537
图 22-1. RTC 结构框图 541
图 23-1. 高级定时器结构框图 579
图 23-2. 内部时钟分频为 1 时正常模式下的控制电路 580
图 23-3. 当预分频器的参数从 1 变到 2 时，计数器的时序图 581
图 23-4. 向上计数时序图，PSC = 0/2 582
图 23-5. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 582
图 23-6. 向下计数时序图，PSC = 0/2 583
图 23-7. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值 584
图 23-8. 中央计数模式计数器时序图 585
图 23-9. 中央计数模式下计数器重复时序图 586
图 23-10. 在向上计数模式下计数器重复时序图 586
图 23-11. 在向下计数模式下计数器重复时序图 587
图 23-12. 通道 0 输入捕获逻辑 588
图 23-13. 多模式通道 0 输入捕获逻辑 588
图 23-14. 输出比较逻辑（当 MCHxMSEL = 2’b00 时，x = 0,1,2,3） 589
图 23-15. 输出比较逻辑（当 MCHxMSEL = 2’b11 时，x = 0,1,2,3） 590
图 23-16. 三种输出比较模式 591
图 23-17. EAPWM 时序图 592
图 23-18. CAPWM 时序图 592
图 23-19. 微调模式：数据格式和寄存器位域 593
图 23-20. PWM 微调模式原理 594
图 23-21 通道 x 输出 PWM（CHxVAL < CHxCOMVAL_ADD） 597
图 23-22 通道 x 输出 PWM（CHxVAL = CHxCOMVAL_ADD） 597
图 23-23. 通道 x 输出 PWM（CHxVAL > CHxCOMVAL_ADD） 598
图 23-24. 通道 x 输出 PWM（CHxVAL 或 CHxCOMVAL_ADD > CARL） 598
图 23-25. 通道 x 输出 PWM 占空比随着 CHxCOMVAL_ADD 值而改变 599
图 23-26. 复合 PWM 模式下四通道输出 599
图 23-27. 边沿对齐模式下 CHx_O 输出脉冲（CHxOMPSEL ≠ 2’b00） 600
图 23-28. 中央对齐模式下 CHx_O 输出脉冲（CHxOMPSEL ≠ 2’b00） 600
图 23-29. 带死区时间的互补输出 603
图 23-30.不同死区时间的互补输出（DTDIFEN=1） 604
图 23-31. BREAK0 中止功能逻辑图 605
图 23-32. BREAK1 中止功能逻辑图 605

图 23-33. 通道响应 BREAK0 中止输入（高电平有效）时，输出信号的行为（IOS=1） 606
图 23-34. BREAK0 和 BREAK1 中止输入有效时通道输出信号的行为 607
图 23-35. BREAK0 的 BRKINx（x = 0…2）引脚锁存功能逻辑图 608
图 23-36. 译码器接口模式下计数器运行例子 610
图 23-37. CI0FE0 极性反相的译码器接口模式下的例子 610
图 23-38. 正交译码器信号断线检测框图 611
图 23-39. 译码器模式 0/1 计数器运行实例（CH1P=0） 612
图 23-40. 译码器模式 2 / 3 计数器运行实例（CH0P / CH1P=0） 612
图 23-41. 三种类型的索引信号 614
图 23-42. 索引信号和 A 相同时的计数器情况（INDP[1:0]=2'b11） 615
图 23-43. 索引信号和计数器复位事件的关系 615
图 23-44. 计数器复位事件和 FINDRST 位 616
图 23-45. 向上计数模式时的索引错误检测 617
图 23-46. 霍尔传感器用在 BLDC 电机控制中 618
图 23-47. 两个定时器之间的霍尔传感器时序图 619
图 23-48. 复位模式 620
图 23-49. 暂停模式下的控制电路 621
图 23-50. 事件模式 621
图 23-51. 单脉冲模式，TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60 622
图 23-52. 可延时单脉冲模式（TIMERx_CHxCV = 0x00，TIMERx_CAR = 0x60） 623
图 23-53. 可编程脉冲输出电路 623
图 23-54. 可编程脉冲输出电路波形 624
图 23-55. CH2_O 和 CH3_O 同时输出脉冲 624
图 23-56. 用定时器 2 的使能信号触发定时器 0 625
图 23-57. 用定时器 2 的更新事件来触发定时器 0 626
图 23-58. 用定时器 2 的使能来选通定时器 0 626
图 23-59. 用定时器 2 的 O0CPRE 信号选通定时器 1 627
图 23-60. 用定时器 2 的 CI0 输入来触发定时器 0 和定时器 2 628
图 23-61. 配置相移方法的框图 629
图 23-62. 3 个定时器的相移框图 629
图 23-63. 中央对齐计数模式下计数器复位后的计数方向 630
图 23-64. 通用定时器 L0 结构框图 707
图 23-65. 内部时钟分频为 1 时正常模式下的控制电路 708
图 23-66. 当预分频器的参数从 1 变到 2 时，计数器的时序图 709
图 23-67. 向上计数时序图，PSC=0 / 2 710
图 23-68. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 710
图 23-69. 向下计数时序图，PSC=0 / 2 711
图 23-70. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值 712
图 23-71. 中央计数模式计数器时序图 713
图 23-72. 输入捕获逻辑 714

图 23-73. 输出比较逻辑（x = 0, 1, 2, 3）.....715
图 23-74. 三种输出比较模式.....716
图 23-75. EAPWM 时序图.....717
图 23-76. CAPWM 时序图.....717
图 23-77. 微调模式：数据格式和寄存器位域.....718
图 23-78. PWM 微调模式原理（16 位）.....719
图 23-79 通道 x 输出 PWM（CHxVAL < CHxCOMVAL_ADD）.....722
图 23-80. 通道 x 输出 PWM（CHxVAL = CHxCOMVAL_ADD）.....722
图 23-81. 通道 x 输出 PWM（CHxVAL > CHxCOMVAL_ADD）.....723
图 23-82. 通道 x 输出 PWM（CHxVAL 或 CHxCOMVAL_ADD > CARL）.....723
图 23-83. 通道 x 输出 PWM 占空比随着 CHxCOMVAL_ADD 值而改变.....724
图 23-84. 复合 PWM 模式下四通道输出.....724
图 23-85. 边沿对齐模式下 CHx_O 输出脉冲（CHxOMPSEL ≠ 2'b00）.....725
图 23-86. 中央对齐模式下 CHx_O 输出脉冲（CHxOMPSEL ≠ 2'b00）.....725
图 23-87. 复位模式.....728
图 23-88. 暂停模式.....728
图 23-89. 事件模式.....728
图 23-90. 单脉冲模式，TIMERx_CHxCV = 0x04，TIMERx_CAR = 0x60.....729
图 23-91. 可延时单脉冲模式（TIMERx_CHxCV = 0x00，TIMERx_CAR = 0x60）.....730
图 23-92. 通用定时器 L3 结构框图.....773
图 23-93. 内部时钟分频为 1 时正常模式下的控制电路.....774
图 23-94. 当预分频器的参数从 1 变到 2 时，计数器的时序图.....775
图 23-95. 向上计数时序图，PSC=0 / 2.....776
图 23-96. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值.....776
图 23-97. 在向上计数模式下计数器重复时序图.....777
图 23-98. 通道 0 输入捕获逻辑.....778
图 23-99. 多模式通道 0 输入捕获逻辑.....778
图 23-100. 输出比较逻辑（当 MCHxMSEL = 2'b00 时，x = 0）.....779
图 23-101. 输出比较逻辑（当 MCHxMSEL = 2'b11 时，x = 0）.....780
图 23-102. 输出比较逻辑（x = 1）.....780
图 23-103. 三种输出比较模式.....782
图 23-104. PWM 时序图.....783
图 23-105. 微调模式：数据格式和寄存器位域.....783
图 23-106. PWM 微调模式原理.....784
图 23-107 通道 x 输出 PWM（CHxVAL < CHxCOMVAL_ADD）.....786
图 23-108. 通道 x 输出 PWM（CHxVAL = CHxCOMVAL_ADD）.....787
图 23-109. 通道 x 输出 PWM（CHxVAL > CHxCOMVAL_ADD）.....787
图 23-110. 通道 x 输出 PWM（CHxVAL 或 CHxCOMVAL_ADD > CARL）.....788
图 23-111. 通道 x 输出 PWM 占空比随着 CHxCOMVAL_ADD 值而改变.....788
图 23-112. 边沿对齐模式下 CHx_O 输出脉冲（CHxOMPSEL = 2'b01）.....789

图 23-113. 带死区时间的互补输出 792
图 23-114.不同死区时间的互补输出（DTDIFEN=1） 792
图 23-115. BREAK0 中止功能逻辑图 793
图 23-116. 通道响应 BREAK0 中止输入（高电平有效）时，输出信号的行为（IOS=1） 794
图 23-117. BREAK0 的 BRKIN0 引脚锁存功能逻辑图 795
图 23-118. 复位模式下的控制电路 796
图 23-119. 暂停模式下的控制电路 797
图 23-120. 事件模式下的控制电路 797
图 23-121. 单脉冲模式，TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60 798
图 23-122. 可延时单脉冲模式（TIMERx_CHxCV = 0x00，TIMERx_CAR = 0x60） 799
图 23-123. 通用定时器 L4 结构框图 840
图 23-124. 内部时钟分频为 1 时正常模式下的控制电路 841
图 23-125. 当预分频器的参数从 1 变到 2 时，计数器的时序图 841
图 23-126. 向上计数时序图，PSC = 0 / 2 842
图 23-127. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 843
图 23-128. 在向上计数模式下计数器重复时序图 844
图 23-129. 通道 0 输入捕获逻辑 845
图 23-130. 多模式通道 0 输入捕获逻辑 845
图 23-131. 输出比较逻辑（当 MCHxMSEL = 2’b00 时，x = 0） 846
图 23-132. 输出比较逻辑（当 MCHxMSEL = 2’b11 时，x = 0） 846
图 23-133. 三种输出比较模式 848
图 23-134. PWM 时序图 849
图 23-135. 带死区时间的互补输出 852
图 23-136.不同死区时间的互补输出（DTDIFEN=1） 852
图 23-137. BREAK0 中止功能逻辑图 853
图 23-138. 通道响应 BREAK0 中止输入（高电平有效）时，输出信号的行为（IOS=1） 854
图 23-139. BREAK0 的 BRKIN0 引脚锁存功能逻辑图 855
图 23-140. 单脉冲模式，TIMERx_CHxCV = 0x04，TIMERx_CAR=0x60 856
图 23-141. 基本定时器结构框图 887
图 23-142. 内部时钟分频为 1 时正常模式下的控制电路 888
图 23-143. 当预分频器的参数从 1 变到 2 时，计数器的时序图 888
图 23-144. 向上计数时序图，PSC = 0 / 2 889
图 23-145. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 890
图 23-146. 微调模式：数据格式和寄存器位域 890
图 23-147. PWM 微调模式原理 891
图 24-1. LPTIMER 结构框图 900
图 24-2. LPTIMER 时钟源选择 901
图 24-3. 内部时钟模式 1（CKSSEL = 0，CNTMEN = 1，PSC[2:0] = 000） 902
图 24-4. 输入滤波时序图（ECKFLT=2’b01） 903
图 24-5. 外部输入高电平计数器 904

图 24-6. LPTIMER 输出（SMST = 1，16 位）.....906
图 24-7. LPTIMER 输出（OMSEL = 1，16 位）.....906
图 24-8. LPTIMER 输出（CTNMST = 1，16 位）.....907
图 24-9. LPTIMER_O 输出（OPSEL=0/1，16 位）.....908
图 24-10. LPTIMER 超时模式（16 位）.....909
图 24-11. 计数器运行在译码器模式 0（上升沿计数）.....911
图 24-12. 计数器运行在译码器模式 0（下降沿计数）.....911
图 24-13. 计数器运行在译码器模式 1（同相）.....912
图 24-14. 计数器运行在译码器模式 1（同相，IN1EIF）.....912
图 24-15. 计数器运行在译码器模式 1（同相，IN0EIF）.....913
图 24-16. 计数器运行在译码器模式 1（同相，INRFOEIF）.....913
图 24-17. 计数器运行在译码器模式 1（同相，INHLOEIF）.....914
图 25-1. HRTIMER 结构框图.....932
图 25-2. Master_TIMER 结构图.....933
图 25-3. 分频为 16 时计数器时钟.....934
图 25-4. 单脉冲模式下计数器的动作.....935
图 25-5. 连续模式下计数器的动作.....935
图 25-6. 连续模式下重复计数器的动作.....936
图 25-7. 单脉冲模式下，CNTRSTM = 0 时重复计数器的动作.....936
图 25-8. 单脉冲模式下，CNTRSTM = 1 时重复计数器的动作.....937
图 25-9. 当预分频系数为 64 时，复位事件重新同步.....937
图 25-10. Slave_TIMERx 结构框图.....941
图 25-11. EXEV0 和 EXEV1 触发捕获 0.....945
图 25-12. STxCAR=0x8, STxCMP1V=0x02 时，比较 1 寄存器的动作.....945
图 25-13. 比较延迟模式框图.....947
图 25-14. 比较 1 延迟模式 0.....948
图 25-15. 比较 1 延迟模式 1.....949
图 25-16.比较延迟模式（SHWEN=0）.....950
图 25-17.可变频率半波模式.....951
图 25-18. 当 IMPUD2V = 1 和 IMPUD2V = 0 时输出的 PWM 波形.....952
图 25-19. 通道输出结构图.....952
图 25-20. O0PRE 波形：CMP0 事件置位，CMP1 事件复位.....953
图 25-21. 中央对齐模式下 OyPRE.....955
图 25-22. 中央对齐模式下重复计数器值 CREP[7:0]和 ROVM[1:0].....956
图 25-23. 每个 t_{HRTMER_CK} 周期的仲裁机制过程.....957
图 25-24. 仲裁机制示例.....959
图 25-25. 脉冲宽度为 1 个 t_{HRTMER_CK} 周期.....959
图 25-26.脉冲宽度为 2 个 t_{HRTMER_CK} 周期.....960
图 25-27. 高分辨率 OxPRE 波形.....960
图 25-28. CNTCKDIV[2:0] = 3'b110 时的 OxPRE 波形.....961

图 25-29. 常规模式下的 C0OPRE 波形 962
图 25-30. 具有死区时间的 C0OPRE 和 C1OPRE 互补输出波形 963
图 25-31. 脉宽小于死区时间时的互补波形 964
图 25-32. 均衡模式的结构框图 964
图 25-33. 均衡模式下的 C0OPRE 和 C1OPRE 波形 966
图 25-34. 延迟空闲模式，ISO0 = 0 和 CHOP = 0 968
图 25-35. 延迟空闲模式，ISO0 = 1 和 CHOP = 0 968
图 25-36. 延迟空闲模式，ISO0 = 0 和 CHOP = 1 969
图 25-37. 延迟空闲模式，ISO0 = 1 和 CHOP = 1 969
图 25-38. 均衡空闲模式，ISO0 = 0 和 ISO1 = 0 970
图 25-39. CHyP=0 或 CHyP=1 时的 STxCHy_O 波形 972
图 25-40. 载波信号结构框图 973
图 25-41. 载波模式使能时的 HRTIMER 输出 974
图 25-42. 消隐模式和窗口模式 976
图 25-43. 外部事件 X 计数器 979
图 25-44. 当 EXEVXCEN 为 1 和 EXEVXCNTTHR[5:0]为 2 时外部事件 X 计数器 980
图 25-45. 突发模式时序图 982
图 25-46. 突发模式的常规进入 983
图 25-47. 突发模式的延迟进入 984
图 25-48. 模拟突发模式示例 985
图 25-49. 外部事件 y（y=0..4）处理过程框图 987
图 25-50. 外部事件 y（y=5..9）处理过程框图 987
图 25-51. 故障输入结构图 989
图 25-52. 当 FLTxRST 为 1，FLTRECCTL 为 1 和 FLTxCNT[3:0] = 0x03 时故障计数器 991
图 25-53. 当 FLTxRST 为 1 和 FLTAR 为 1 时故障自动恢复 992
图 25-54. ADC 触发源选择图 994
图 25-55. 向上计数模式下 ADC 触发分频 995
图 25-56. 中央对齐计数模式下 ADC 触发分频 996
图 25-57. DAC 触发源选择图 997
图 25-58. 当 TRIG0M= 0 和 TRIG1M = 0 时 DAC 触发 998
图 25-59. 当 TRIG0M = 1 和 TRIG1M = 1 时 DAC 触发 999
图 25-60. DMA 模式运行流程图 1002
图 26-1. IFRP 输出时序图 1 1203
图 26-2. IFRP 输出时序图 2 1204
图 26-3. IFRP 输出时序图 3 1204
图 27-1. USART 模块内部框图 1207
图 27-2. USART 字符帧（8 数据位和 1 停止位） 1208
图 27-3. USART 发送步骤 1210
图 27-4. 过采样方式接收一个数据位（OSB = 0） 1211
图 27-5. 采用 DMA 方式实现 USART 数据发送配置步骤 1212

图 27-6. 采用 DMA 方式实现 USART 数据接收配置步骤 1213
图 27-7. 两个 USART 之间的硬件流控制 1213
图 27-8. 硬件流控制 1214
图 27-9. 空闲状态下检测断开帧 1216
图 27-10. 数据传输过程中检测断开帧 1216
图 27-11. 同步模式下的 USART 示例 1216
图 27-12. 8-bit 格式的 USART 同步通信波形（CLEN = 1） 1217
图 27-13. IrDA SIR ENDEC 模块 1217
图 27-14. IrDA 数据调制 1218
图 27-15. ISO7816-3 数据帧格式 1219
图 27-16. USART 接收 FIFO 结构 1221
图 27-17. USART 发送 FIFO 结构 1221
图 27-18. USART 中断映射框图 1223
图 28-1. I2C 模块框图 1249
图 28-2. 数据有效性 1250
图 28-3. 开始和停止信号 1251
图 28-4. 10 位地址的 I2C 通讯流程（主机发送） 1251
图 28-5. 7 位地址的 I2C 通讯流程（主机发送） 1251
图 28-6. 7 位地址的 I2C 通讯流程（主机接收） 1252
图 28-7. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R = 0） 1252
图 28-8. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R = 1） 1252
图 28-9. 数据保持时间 1253
图 28-10. 数据建立时间 1253
图 28-11. 数据发送 1255
图 28-12. 数据接收 1256
图 28-13. I2C 从机初始化 1259
图 28-14. I2C 从机发送编程模型（SS=0） 1260
图 28-15. I2C 从机发送编程模型（SS=1） 1261
图 28-16. I2C 从机接收编程模型 1262
图 28-17. I2C 主机初始化 1263
图 28-18. I2C 主机发送编程模型（N<=255） 1264
图 28-19. I2C 主机发送编程模型（N>255） 1265
图 28-20. I2C 主机接收编程模型（N<=255） 1266
图 28-21. I2C 主机接收编程模型（N>255） 1267
图 28-22. SMBus 主机发送器和从机接收器通信流程 1270
图 28-23. SMBus 主机接收器和从机发送器通信流程 1271
图 29-1. CAN 模块结构框图 1289
图 29-2. 传输延迟 1318
图 29-3. CAN 位时间 1321
图 30-1. SPI 结构框图 1357

图 30-2. SPI0 常规模式下的时序图 1359
图 30-3. SPI0 数据帧右对齐示意图 1359
图 30-4. SPI 四线模式下的 SPI 时序图（CKPL=1, CKPH=1, LF=0） 1359
图 30-5. 发送/接收缓冲区 1360
图 30-6. 典型的全双工模式连接 1363
图 30-7. 典型的单工模式连接（主机：接收，从机：发送） 1364
图 30-8. 典型的单工模式连接（主机：只发送，从机：接收） 1364
图 30-9. 典型的双向线连接 1364
图 30-10. 主机 TI 模式在不连续发送时的时序图 1366
图 30-11. 主机 TI 模式在连续发送时的时序图 1366
图 30-12. 从机 TI 模式时序图 1367
图 30-13. NSS 脉冲模式时序图（主机连续发送） 1368
图 30-14. SPI 四线模式写操作时序图 1369
图 30-15. SPI 四线模式读操作时序图 1370
图 31-1. QSPI 结构框图 1384
图 31-2. QSPI 命令格式 1385
图 31-3. QSPI DDR 模式 1387
图 31-4. 使用 DQS 信号 DDR 模式读 1387
图 31-5. CSN 和 SCK 的行为（CSNCKM = 0） 1388
图 32-1. CPDM 模块结构框图 1406
图 32-2. CPDM 延迟线长度配置流程图 1407
图 32-3. CPDM 输出时钟相位配置流程图 1408
图 33-1. 系统架构 1412
图 33-2. EXMC Bank 划分 1413
图 33-3. Bank0 地址映射 1413
图 33-4. 模式 1 读时序 1419
图 33-5. 模式 1 写时序 1419
图 33-6. 模式 A 读时序 1420
图 33-7. 模式 A 写时序 1421
图 33-8. 模式 2 / B 读时序 1423
图 33-9. 模式 2 写时序 1423
图 33-10. 模式 B 写时序 1423
图 33-11. 模式 C 读时序 1425
图 33-12. 模式 C 写时序 1425
图 33-13. 模式 D 读时序 1427
图 33-14. 模式 D 写时序 1427
图 33-15. 复用模式读时序 1429
图 33-16. 复用模式写时序 1429
图 33-17. 异步等待有效时的读时序 1431
图 33-18. 异步等待有效时的写时序 1432

图 33-19. 同步复用突发传输读时序 1434
图 33-20. 同步复用突发传输写时序 1435
图 34-1. HPDF 结构框图 1446
图 34-2. SPI 数据传输时序图 1449
图 34-3. 曼切斯特数据传输时序图 1450
图 34-4. 曼切斯特同步时序图 1451
图 34-5. 时钟丢失检测时序图 1453
图 34-6. 通道引脚重定向 1454
图 34-7. HPDF 模块外部输入数据处理流程 1463
图 34-8. HPDF 中断逻辑图 1465
图 35-1. FAC 结构框图 1487
图 35-2. 输入缓冲区 1488
图 35-3. 循环输入缓冲区 1489
图 35-4. 循环输入缓冲区区域 1490
图 35-5. 循环输出缓冲区 1491
图 35-6. 循环输出缓冲区区域 1492
图 35-7. FIR 滤波器结构图 1493
图 35-8. IIR 滤波器结构图 1494

表索引

表 1-1. 总线互联矩阵 44
表 1-2. GD32G553 系列器件的存储器映射表 47
表 1-3.引导模式 54
表 2-1. 512KB 双 bank 闪存基地址和构成 96
表 2-2. 512KB 单 bank 闪存基地址和构成 97
表 2-3. WSCNT 与 AHB 时钟频率对应关系（LDO= 1.1V） 99
表 2-4. 选项字节 106
表 2-5. 出厂值 110
表 2-6. WP 保护 113
表 2-7. FMC 中断请求 115
表 3-1. 节电模式总结 144
表 4-1. 时钟输出的时钟源选择 161
表 4-2. 低速时钟输出的时钟源选择 161
表 5-1. Cortex®-M33 中的 NVIC 异常类型 212
表 5-2. 中断向量表 213
表 5-3. EXTI 触发源 218
表 6-1. 触发输入位域选择 227
表 6-2. TRIGSEL 输入输出映射关系 233
表 7-1. GPIO 配置表 273
表 8-1. DMA 传输操作 300
表 8-2. 中断事件 303
表 9-1. DMAMUX 信号 312
表 9-2. 中断事件 316
表 9-3. DMAMUX 请求路由输入信号映射 316
表 9-4. 触发输入信号映射 320
表 9-5. 同步输入信号映射 321
表 11-1. CLAxSIGS0 输入选择 333
表 11-2. CLAxSIGS1 输入选择 333
表 11-3. LCU 控制 334
表 12-1. ALGO 配置 346
表 14-1. 输入数据配置 387
表 14-2. 输出数据配置 388
表 14-3. IEEE754 32 位单精度浮点格式 388
表 14-4. TMU 模式配置 389
表 14-5. 模式 0 描述，IFLTEN 位和 OFLTEN 位均置位时 389
表 14-6. 模式 0 描述，IFLTEN 位或 OFLTEN 位复位时 390
表 14-7. 模式 1 描述，IFLTEN 位和 OFLTEN 位均置位时 391

表 14-8. 模式 1 描述，IFLTEN 位或 OFLTEN 位复位时 391
表 14-9. 模式 2 描述，IFLTEN 位和 OFLTEN 位均置位时 392
表 14-10. 模式 2 描述，IFLTEN 位或 OFLTEN 位复位时 392
表 14-11. 模式 3 描述，IFLTEN 位和 OFLTEN 位均置位时 393
表 14-12. 模式 3 描述，IFLTEN 位或 OFLTEN 位复位时 393
表 14-13. 模式 4 描述，IFLTEN 位和 OFLTEN 位均置位时 394
表 14-14. 模式 4 描述，IFLTEN 位或 OFLTEN 位复位时 395
表 14-15. 模式 5 描述，IFLTEN 位和 OFLTEN 位均置位时 395
表 14-16. 模式 5 描述，IFLTEN 位或 OFLTEN 位复位时 396
表 14-17. 模式 6 描述，IFLTEN 位和 OFLTEN 位均置位时 396
表 14-18. 模式 6 描述，IFLTEN 位或 OFLTEN 位复位时 397
表 14-19. 模式 7 描述，IFLTEN 位和 OFLTEN 位均置位时 397
表 14-20. 模式 7 描述，IFLTEN 位或 OFLTEN 位复位时 398
表 14-21. 模式 8 描述，IFLTEN 位和 OFLTEN 位均置位时 398
表 14-22. 模式 8 描述，IFLTEN 位或 OFLTEN 位复位时 398
表 14-23. 模式 8 推荐的缩放因子 399
表 14-24. 模式 9 描述，IFLTEN 位和 OFLTEN 位均置位时 399
表 14-25. 模式 9 描述，IFLTEN 位或 OFLTEN 位复位时 400
表 14-26. 模式 9 推荐的缩放因子 400
表 15-1. IEEE32 位单精度浮点格式 407
表 15-2. 8 点输入位倒序操作 408
表 16-1. 引脚分配 418
表 16-2. OTP JTAG 字节 419
表 17-1. ADC 内部输入信号 429
表 17-2. ADC 输入引脚定义 429
表 17-3. ADC 差分通道引脚匹配 431
表 17-4. 常规序列外部触发源 437
表 17-5. ADC 不同分辨率对应的 t_{CONV} 时间 440
表 17-6. ADC 同步模式表 442
表 18-1. DAC 引脚 471
表 18-2. DAC 触发与输出 471
表 18-3. DAC 数据格式（12-bit 数据） 472
表 18-4. DAC 外部触发 473
表 18-5. DAC 锯齿波复位触发信号 475
表 18-6. DAC 锯齿波步进触发信号 475
表 18-7. 采样和刷新时间计算公式 479
表 19-1. CMP 的输入和输出总结 499
表 20-1. VREF 模式 526
表 21-1. 独立看门狗定时器在 32kHz (IRC32K)时的最小/最大超时周期 530
表 21-2. 在 216MHz (f_{PCLK1})时的最大/最小超时值 537

表 22-1. RTC 引脚配置和功能 551
表 22-2. 省电模式管理 552
表 22-3. RTC 中断控制 552
表 23-1. 定时器（TIMERx）分为 5 种类型 576
表 23-2. 高级定时器通道介绍 579
表 23-3. 边沿对齐模式中 CHxVAL 和 CARL 位域的变化 594
表 23-4. 中央对齐模式中 CHxVAL 和 CARL 位域的变化 595
表 23-5 复合 PWM 脉冲宽度 595
表 23-6. 由参数控制的互补输出表（MCHxMSEL = 2'b11） 602
表 23-7. BREAK0 和 BREAK1 输入信号时，TIMER 互补通道输出情况（break 输入高电平有效） 606
表 23-8. 中止功能锁存/释放条件 608
表 23-9. 不同译码器模式下的计数方向 609
表 23-10. 译码器模式 1 的计数情况 611
表 23-11. 译码器模式 2 / 3 下的计数方向 613
表 23-12. 从模式例子列表 619
表 23-13. 边沿对齐模式中 CHxVAL 和 CARL 位域的变化 719
表 23-14. 中央对齐模式中 CHxVAL 和 CARL 位域的变化 720
表 23-15. 复合 PWM 脉冲宽度 721
表 23-16. 从模式列表和举例（通用定时器 L0） 727
表 23-17. 边沿对齐模式中 CHxVAL 和 CARL 位域的变化 784
表 23-18. 复合 PWM 脉冲宽度 785
表 23-19. 由参数控制的互补输出表（MCHxMSEL=2'b11） 790
表 23-20. 中止功能锁存/释放条件 795
表 23-21. 从模式例子列表 795
表 23-22. 由参数控制的互补输出表（MCHxMSEL = 2'b11） 850
表 23-23. 中止功能锁存/释放条件 855
表 23-24. 边沿对齐模式中 CHxVAL 和 CARL 位域的变化 892
表 24-1. 预分频器的分频系数 902
表 24-2. 外部触发映射 904
表 24-3. 计数方向与译码器信号之间的关系 910
表 24-4. LPTIMER 工作在低功耗模式 915
表 24-5. LPTIMER 中断事件 915
表 25-1. 自动重载寄存器和比较 y（y = 0..3）寄存器的限值 933
表 25-2. $f_{HRTIMER_CK}$ = 216MHz 的分辨率 934
表 25-3. 交错模式选择 938
表 25-4. 交错模式的比较值 938
表 25-5. Master_TIMER 影子寄存器和更新事件 939
表 25-6. 计数器和捕获 y(y=0,1)寄存器限值 942
表 25-7. 交错模式选择 946
表 25-8. 两种交错模式比较值 946

表 25-9. Slave_TIMER 内部连接事件 954
表 25-10. 消隐模式和窗口模式在向上计数模式和中央对齐模式 956
表 25-11. 交叉开关和空闲控制同时运行 966
表 25-12. 进入和退出空闲状态的请求 967
表 25-13. 突发模式控制的空闲状态时的输出 971
表 25-14. 输出级状态编程（x=0..7, y=0,1） 972
表 25-15. Slave_TIMERx 影子寄存器和通用寄存器和更新事件 975
表 25-16. STxUPINy（y=0..2）和芯片内部信号 975
表 25-17. 消隐模式下的滤波信号映射 977
表 25-18. 窗口模式的滤波信号映射 979
表 25-19. 突发模式的芯片内部信号 981
表 25-20. 外部事件映射 988
表 25-21. 故障通道映射 989
表 25-22. 故障通道消隐 990
表 25-23. 故障通道复位源 991
表 25-24. 中断映射 999
表 25-25. DMA 请求映射 1000
表 27-1. USART 重要引脚描述 1207
表 27-2. 停止位配置 1208
表 27-3. USART 中断请求 1222
表 28-1. I2C 总线术语说明（参考飞利浦 I2C 规范） 1249
表 28-2. 数据建立时间和数据保持时间 1254
表 28-3. 可关闭通信模式 1256
表 28-4. I2C 错误标志 1272
表 28-5. I2C 中断事件 1272
表 29-1. 64 字节数据的邮箱描述符 1290
表 29-2. DLC 表示的数据字节长度 1292
表 29-3. 接收邮箱 CODE 1292
表 29-4. 发送邮箱 CODE 1293
表 29-5. 邮箱数目 1295
表 29-6. 接收 FIFO 描述符 1295
表 29-7. 当本地优先级禁用时的邮箱仲裁值（32 位） 1305
表 29-8. 当本地优先级使能时的邮箱仲裁值（35 位） 1305
表 29-9. 接收邮箱匹配 1311
表 29-10. Rx FIFO 匹配 1312
表 29-11. 中断事件 1323
表 29-12. Rx FIFO 标识符过滤表元素数目 1338
表 30-1. SPI 信号描述 1357
表 30-2. SPI 四线信号描述 1358
表 30-3. 从机模式 NSS 功能 1361

表 30-4. 主机模式 NSS 功能 1362
表 30-5. SPI 运行模式 1362
表 30-6. SPI 中断请求 1373
表 31-1. QSPI 信号线描述 1383
表 31-2. QSPI 命令描述 1385
表 31-3. QSPI 信号线模式 1386
表 31-4. QSPI 使用引脚与支持的最高通信时钟对应关系 1386
表 31-5. AHB 写访问方式与 FIFO 增加的字节数的关系 1389
表 31-6. TERR 和 AHB 错误条件 1391
表 31-7. QSPI 中断事件 1391
表 33-1. NOR Flash 接口信号描述 1414
表 33-2. PSRAM 非复用接口信号描述 1415
表 33-3. 支持的 16 位传输 1416
表 33-4. 支持的 8 位传输 1416
表 33-5. NOR / PSRAM 控制时序参数 1417
表 33-6. EXMC 时序模型 1418
表 33-7. 模式 1 相关寄存器配置 1419
表 33-8. 模式 A 相关寄存器配置 1421
表 33-9. 模式 2 / B 相关寄存器配置 1424
表 33-10. 模式 C 相关寄存器配置 1426
表 33-11. 模式 D 相关寄存器配置 1428
表 33-12. 复用模式相关寄存器配置 1430
表 33-13. 同步复用模式读时序配置 1434
表 33-14. 同步复用模式写时序配置 1435
表 34-1. HPDF 引脚定义 1446
表 34-2. HPDF 断路连接 1447
表 34-3. SPI 接口时钟配置 1448
表 34-4. 并行数据封装模式 1456
表 34-5. 注入组的触发信号 1458
表 34-6. Sinc^x 滤波的最大输出分辨率与过采样滤的关系 1460
表 34-7. 积分器的最大输出分辨率与 IOR、SFOR、SFO 之间的关系 1460
表 34-8. 阈值监视器工作模式特点 1461
表 34-9. 最大输出速率 1463
表 34-10. HPDF 中断事件 1464
表 35-1. IEEE 32 位单精度浮点格式 1495
表 36-1. 寄存器功能位访问属性 1505
表 36-2. 术语 1505
表 37-1. 版本历史 1507

1. 系统及存储器架构

GD32G553系列器件是基于Arm® Cortex®-M33处理器的32位通用微控制器。Arm® Cortex®-M33处理器包括数据总线和系统总线这两条AHB总线。Cortex®-M33处理器的所有存储访问，根据不同的目的和目标存储空间，都会在这两条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4 GB的存储空间，充分保证了系统的灵活性和可扩展性。

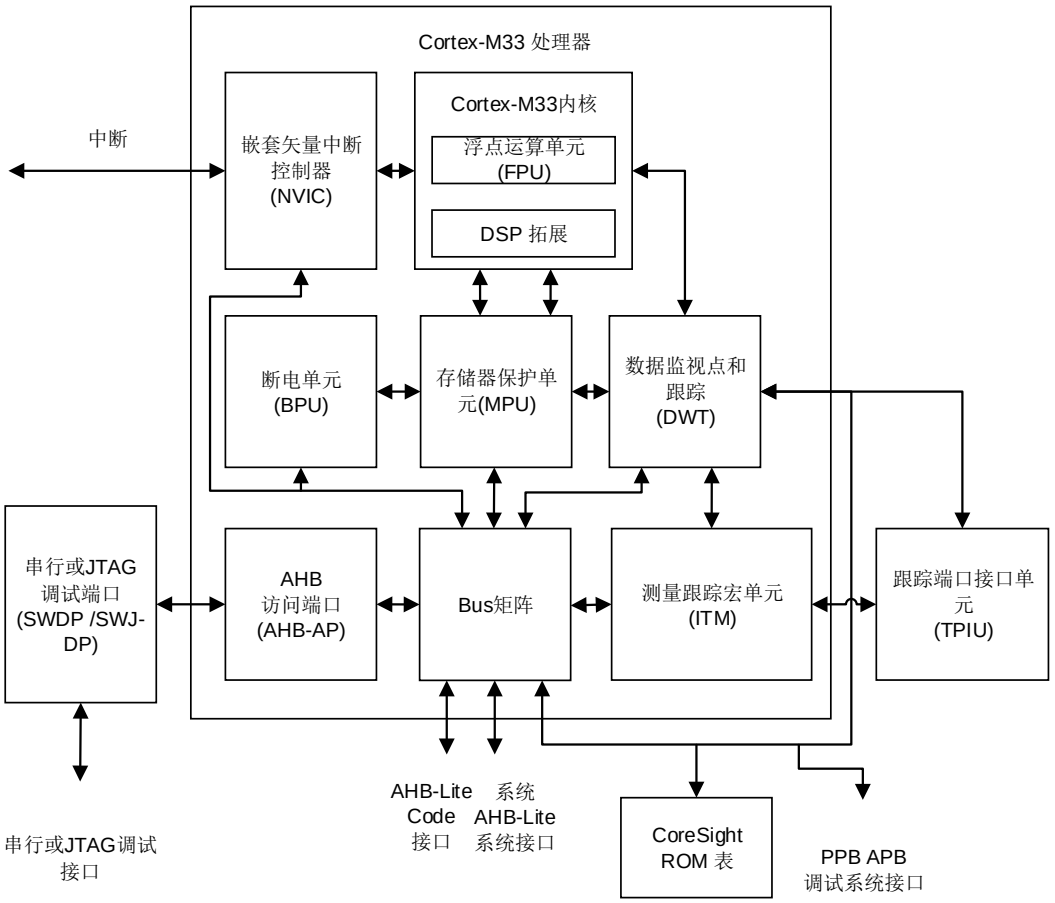
1.1. Arm® Cortex®-M33 处理器

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

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

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

图 1-1. Cortex®-M33 处理器结构框图



1.2. 系统架构

GD32G553 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、一个 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. 总线互联矩阵*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，“0”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. 总线互联矩阵

	CBUS	SBUS	DMA0	DMA1	FFT
FMC	1	0	1	1	1
SRAM0	1	1	1	1	1
SRAM1	0	1	1	1	1
TCMSRAM	1	1	1	1	1
AHB1	0	1	1	1	1

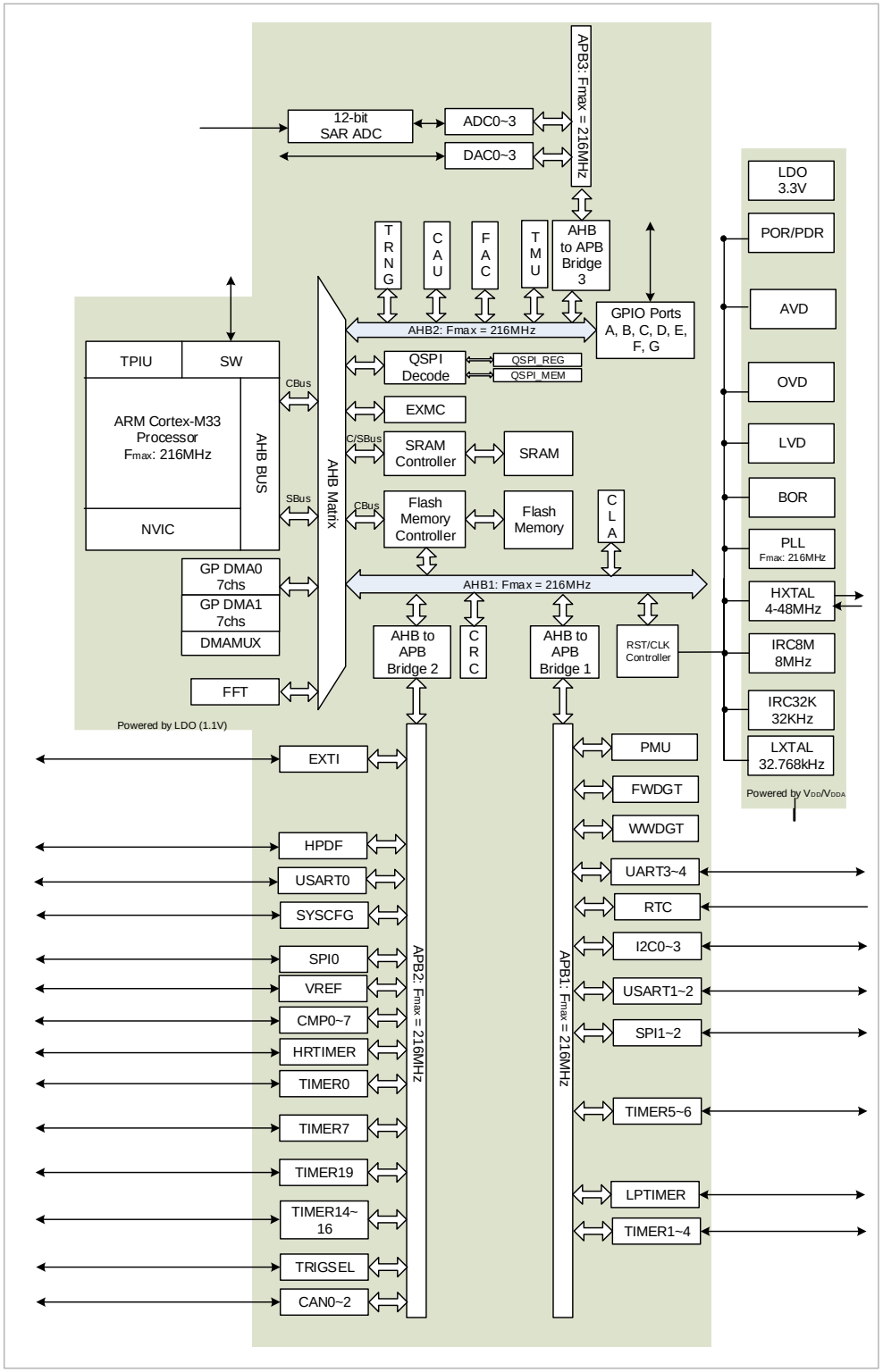
	CBUS	SBUS	DMA0	DMA1	FFT
AHB2	0	1	1	1	1
EXMC	1	1	1	1	1
QSPI	1	1	1	1	1

如*表 1-1. 总线互联矩阵*所示，AHB 互联矩阵连接了一些个主机，分别为：CBUS、SBUS、DMA0、DMA1 和 FFT。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。DMA 总线连接 DMA 的 AHB 主接口到总线矩阵。

AHB 互联矩阵也连接了一些从机，分别为：FMC、SRAM0、SRAM1、TCMSRAM、EXMC、AHB1、AHB2 和 QSPI。FMC 是闪存存控制器的总线接口。TCMSRAM 是紧耦合存储器 SRAM。SRAM0 和 SRAM1 是片上静态随机存取存储器。EXMC 是外部存储器控制器。AHB1 是连接所有 AHB1 从机的 AHB 总线，AHB2 是连接所有 AHB2 从机的 AHB 总线，而 APB1 和 APB2 是连接所有 APB 从机的两条 APB 总线。AHB1 外设包括 APB1 外设和 APB2 外设，AHB2 外设包括 APB3 外设和通过 EXMC 或 QSPI 的外部存储器。

GD32G553 系列器件的系统架构如*图 1-2. GD32G553 系列器件的系统架构示意图*所示。

图 1-2. GD32G553 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. GD32G553 系列器件的存储器映射表*显示了 GD32G553 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. GD32G553 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
External RAM		0xD000 0000 - 0xDFFF FFFF	保留
		0xC000 0000 - 0xCFFF FFFF	保留
		0xA000 1400 - 0xBFFF FFFF	保留
		0xA000 1000 - 0xA000 13FF	QSPI- REG
		0xA000 0400 - 0xA000 0FFF	EXMC - SWREG
		0xA000 0000 - 0xA000 03FF	
		0x9000 0000 - 0x9FFF FFFF	QSPI- MEM
		0x8000 0000 - 0x8FFF FFFF	保留
		0x7000 0000 - 0x7FFF FFFF	保留
		0x6000 0000 - 0x6FFF FFFF	EXMC - NOR/PSRAM/SRAM
Peripheral	AHB1	0x5001 0000 - 0x5FFF FFFF	保留
		0x5000 4000 - 0x5000 FFFF	保留
		0x5000 3000 - 0x5000 3FFF	保留
		0x5000 2000 - 0x5000 2FFF	保留
	APB3	0x5000 1C00 - 0x5000 1FFF	DAC3
		0x5000 1800 - 0x5000 1BFF	DAC2
		0x5000 1400 - 0x5000 17FF	DAC1
		0x5000 1000 - 0x5000 13FF	DAC0
		0x5000 0C00 - 0x5000 0FFF	ADC3
		0x5000 0800 - 0x5000 0BFF	ADC2
		0x5000 0400 - 0x5000 07FF	ADC1
		0x5000 0000 - 0x5000 03FF	ADC0
	AHB2	0x4802 5000 - 0x4FFF FFFF	保留
		0x4802 4800 - 0x4802 4FFF	FAC
		0x4802 4400 - 0x4802 47FF	TMU

预先定义的地址空间	总线	地址	外设
		0x4802 4000 - 0x4802 43FF	保留
		0x4802 3000 - 0x4802 3FFF	保留
		0x4802 2C00 - 0x4802 2FFF	保留
		0x4802 2800 - 0x4802 2BFF	CPDM
		0x4802 2400 - 0x4802 27FF	保留
		0x4802 1C00 - 0x4802 23FF	保留
		0x4802 1800 - 0x4802 1BFF	TRNG
		0x4802 1400 - 0x4802 17FF	保留
		0x4802 1000 - 0x4802 13FF	CAU
		0x4802 0400 - 0x4802 0FFF	保留
		0x4800 1C00 - 0x4802 03FF	保留
		0x4800 1800 - 0x4800 1BFF	GPIOG
		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 8400 - 0x47FF FFFF	保留
		0x4003 8000 - 0x4003 83FF	CLA
		0x4002 7800 - 0x4003 7FFF	保留
		0x4002 5000 - 0x4002 77FF	FFT
		0x4002 3400 - 0x4002 4FFF	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 D000 - 0x4001 FFFF	保留
		0x4001 C000 - 0x4001 CFFF	CAN2
		0x4001 B000 - 0x4001 BFFF	CAN1

预先定义的地址空间	总线	地址	外设
		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 7800 - 0x4001 7BFF	VREF
		0x4001 7400 - 0x4001 77FF	保留
		0x4001 7000 - 0x4001 73FF	HPDF
		0x4001 6800 - 0x4001 6FFF	保留
		0x4001 5800 - 0x4001 67FF	HRTIMER
		0x4001 5400 - 0x4001 57FF	保留
		0x4001 5000 - 0x4001 53FF	TIMER19
		0x4001 4C00 - 0x4001 4FFF	保留
		0x4001 4800 - 0x4001 4BFF	TIMER16
		0x4001 4400 - 0x4001 47FF	TIMER15
		0x4001 4000 - 0x4001 43FF	TIMER14
		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	保留
		0x4001 2400 - 0x4001 27FF	保留
		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
	APB1	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

预先定义的地址空间	总线	地址	外设
		0x4000 C800 - 0x4000 CBFF	保留
		0x4000 C400 - 0x4000 C7FF	保留
		0x4000 C000 - 0x4000 C3FF	I2C2
		0x4000 9800 - 0x4000 BFFF	保留
		0x4000 9400 - 0x4000 97FF	LPTIMER
		0x4000 8400 - 0x4000 93FF	保留
		0x4000 8000 - 0x4000 83FF	保留
		0x4000 7C00 - 0x4000 7FFF	保留
		0x4000 7800 - 0x4000 7BFF	保留
		0x4000 7400 - 0x4000 77FF	保留
		0x4000 7000 - 0x4000 73FF	PMU
		0x4000 6C00 - 0x4000 6FFF	保留
		0x4000 6800 - 0x4000 6BFF	保留
		0x4000 6400 - 0x4000 67FF	保留
		0x4000 6000 - 0x4000 63FF	保留
		0x4000 5C00 - 0x4000 5FFF	I2C3
		0x4000 5800 - 0x4000 5BFF	I2C1
		0x4000 5400 - 0x4000 57FF	I2C0
		0x4000 5000 - 0x4000 53FF	UART4
		0x4000 4C00 - 0x4000 4FFF	UART3
		0x4000 4800 - 0x4000 4BFF	USART2
		0x4000 4400 - 0x4000 47FF	USART1
		0x4000 4000 - 0x4000 43FF	保留
		0x4000 3C00 - 0x4000 3FFF	SPI2
		0x4000 3800 - 0x4000 3BFF	SPI1
		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	TIMER4
		0x4000 0800 - 0x4000 0BFF	TIMER3

预先定义的地址空间	总线	地址	外设
		0x4000 0400 - 0x4000 07FF	TIMER2
		0x4000 0000 - 0x4000 03FF	TIMER1
SRAM		0x2002 0000 - 0x3FFF FFFF	保留
		0x2001 C000 - 0x2001 FFFF	TCMSRAM aliased(32KB)
		0x2001 8000 - 0x2001 BFFF	
		0x2001 4000 - 0x2001 7FFF	SRAM1(16KB)
		0x2000 D000 - 0x2001 3FFF	SRAM0(80KB)
		0x2000 C000 - 0x2000 CFFF	
		0x2000 8000 - 0x2000 BFFF	
		0x2000 5000 - 0x2000 7FFF	
		0x2000 2000 - 0x2000 4FFF	
		0x2000 1000 - 0x2000 1FFF	
		0x2000 0000 - 0x2000 0FFF	
Code		0x1FFF FC10 - 0x1FFF FFFF	保留
		0x1FFF FC00 - 0x1FFF FC0F	保留
		0x1FFF F830 - 0x1FFF BFFF	保留
		0x1FFF F800 - 0x1FFF F82F	Option Bytes
		0x1FFF C000 - 0x1FFF F7FF	保留
		0x1FFF 8000 - 0x1FFF BFFF	System memory 1
		0x1FFF 7830 - 0x1FFF 7FFF	保留
		0x1FFF 7800 - 0x1FFF 782F	Option Bytes
		0x1FFF 7000 - 0x1FFF 77FF	OTP
		0x1FFF 3400 - 0x1FFF 6FFF	保留
		0x1FFF 0000 - 0x1FFF 33FF	System memory 0
		0x1000 8000 - 0x1FFE FFFF	保留
		0x1000 0000 - 0x1000 7FFF	TCMSRAM
		0x0A00 8000 - 0x0FFF FFFF	保留
		0x0A00 6000 - 0x0A00 7FFF	保留
		0x0A00 4000 - 0x0A00 5FFF	保留
		0x0A00 0000 - 0x0A00 3FFF	保留
		0x08C0 1000 - 0x09FF FFFF	保留
		0x08C0 0000 - 0x08C0 0FFF	保留
		0x0881 0000 - 0x08BF FFFF	保留
		0x0880 0000 - 0x0880 FFFF	保留
		0x0808 0000 - 0x0871 FFFF	保留
		0x0806 0000 - 0x0807 FFFF	Main Flash memory
		0x0802 0000 - 0x0805 FFFF	

预先定义的地址空间	总线	地址	外设
		0x0801 0000 - 0x0801 FFFF	
		0x0800 0000 - 0x0800 FFFF	
		0x0006 0000 - 0x07FF FFFF	保留
		0x0002 0000 - 0x0007 FFFF	Aliased to Flash or system memory
		0x0001 0000 - 0x0001 FFFF	
		0x0000 0000 - 0x0000 FFFF	

1.3.1. 片上 SRAM 存储器

GD32G553 系列微控制器含有高达 80KB 的 SRAM0，16KB 的 SRAM1 和 32KB 的 TCMSRAM。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 SRAM，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。

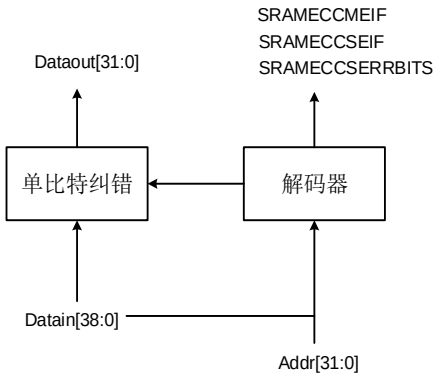
ECC 模块由编码器和解码器两部分构成：

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*图 1-3. ECC 解码器示意图*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

当检测到在 SRAM0 中发生了单比特可纠错的事件时，EEIC 模块有如下配置：

（1） SYSCFG_STAT 寄存器中的 SRAM0ECCSEIF 位置位，软件写 1 可以清除。

（2） SYSCFG_CFG3 寄存器记录发生单比特可纠错 ECC 事件的地址。

当检测到在 SRAM1 中发生了单比特可纠错的事件时，EEIC 模块有如下配置：

（1） SYSCFG_STAT 寄存器中的 SRAM1ECCSEIF 位置位，软件写 1 可以清除。

（2） SYSCFG_CFG4 寄存器记录发生单比特可纠错 ECC 事件的地址。

当检测到在 TCMSRAM 中发生了单比特可纠错的事件时，EEIC 模块有如下配置：

（1） SYSCFG_STAT 寄存器中的 TCMSRAMECCSEIF 位置位，软件写 1 可以清除。

（2） SYSCFG_CFG5 寄存器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

当检测到在 SRAM0 中发生了多比特（两比特）不可纠错事件时，EEIC 模块有如下配置：

（1） SYSCFG_STAT 寄存器中的 SRAM0ECCMEIF 位置位，软件写 1 可以清除。

（2） SYSCFG_CFG3 寄存器记录发生多比特（两比特）不可纠错 ECC 事件的地址。

当检测到在 SRAM1 中发生了多比特（两比特）不可纠错的事件时，EEIC 模块有如下配置：

（1） SYSCFG_STAT 寄存器中的 SRAM1ECCMEIF 位置位，软件写 1 可以清除。

（2） SYSCFG_CFG4 寄存器记录发生多比特（两比特）不可纠错 ECC 事件的地址。

当检测到在 TCMSRAM 中发生了多比特（两比特）不可纠错的事件时，EEIC 模块有如下配置：

（1） SYSCFG_STAT 寄存器中的 TCMSRAMECCMEIF 位置位，软件写 1 可以清除。

（2） SYSCFG_CFG5 寄存器记录发生多比特（两比特）不可纠错 ECC 事件的地址。

TCMSRAM 写保护

TCMSRAM 以 1KB 的页粒度被写保护。TCMSRAM 页的写保护功能可以通过设置 SYSCFG TCMSRAM 写保护寄存器(SYSCFG_TCMSRAMWP)中的 PxWPEN(x=0..31)位来实现。写保护功能只能通过系统复位才能解除/清除。当 TCMSRAM 设置写保护时，对其进行操作会产生 hardfault。

TCMSRAM 擦除

可以通过软件设置 SYSCFG_TCMSRAMCS 寄存器中 TCMSRAMERS 位来请求 TCMSRAM 的擦除操作，而 TCMSRAMBSYF 位则反映了 TCMSRAM 的擦除操作是否正在进行。

配置用户选项字节中的选项位中的 TCMSRAM_ERS 位，可以在系统复位时擦除 TCMSRAM。

1.3.2. 片上 FLASH 存储器概述

GD32G553 系列微控制器可以提供高密度片上 FLASH 存储器，按以下分类进行组织：

- 高达512KB主FLASH存储器；
- 器件配置的选项字节。

更多详细说明请参考*闪存控制器（FMC）*章节。

1.4. 引导配置

GD32G553 系列微控制器提供了三种引导源，可以通过 BOOT0 引脚和用户选项字节中的引导配置位（nBOOT1、nBOOT0、nSWBT0）来进行选择，详细说明见*表 1-3.引导模式*。

BOOT0 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，通过设置引导模式配置。一旦这个引脚电平被采样，它可以被释放并用于其他用途。

表 1-3.引导模式

引导源选择	启动模式配置				
	BOOTLK	nBOOT1 位	BOOT0 位	nSWBT0 位	nBOOT0 位
主 Flash 存储器	1	X	X	X	X
主 Flash 存储器	0	X	0	1	X
主 Flash 存储器	0	X	X	0	1
系统存储器	0	1	1	1	X
系统存储器	0	1	X	0	0
片上 SRAM0	0	0	1	1	X
片上 SRAM0	0	0	X	0	0

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。

根据所选引导源，主 Flash 存储器（从 0x0800 0000 开始的存储空间）或者系统存储器（从 0x1FFF 0000 开始的存储空间）会被映射到引导存储空间，即从 0x0000 0000 开始的地址空间。如果片上 SRAM（开始于 0x2000 0000 的存储空间）被选为引导源，用户必须在应用程序初始化代码中通过修改 NVIC 异常向量表和偏移地址将向量表重置到 SRAM 中。

嵌入式引导加载程序位于系统内存中，用于对闪存进行重新编程。引导加载程序可以通过某些接口激活，请参阅 datasheet 的引导模式章节。

1.5. 重映射配置

当选择相应的引导模式配置，软件可以通过系统配置寄存器 0 (SYSCFG_CFG0)中的 BOOT_MODE 位来配置内存重映射功能。存储器包括主 Flash 存储器、系统存储器、内嵌 SRAM、EXMC BANK0 和 QSPI 存储器。

Flash 存储器扇区重映射功能由系统配置寄存器 0 (SYSCFG_CFG0)中的 FMC_SWP 位配置。

1.6. 系统配置控制器（SYSCFG）

系统配置控制器的主要功能如下：

- 重新映射存储器区域
- 配置FPU中断
- 配置I2C Fm+和I/O模拟开关的电压增强器
- 配置TCMSRAM写保护和软件擦除
- 管理定时器break输入锁定
- 管理外部中断线与GPIO的连接

1.7. SYSCFG 寄存器

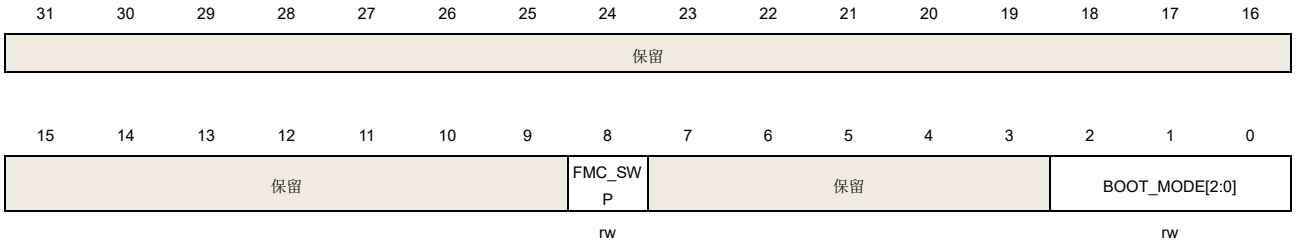
SYSCFG 基地址：0x4001 0000

1.7.1. 配置寄存器 0(SYSCFG_CFG0)

地址偏移：0x00

复位值：0x0000 010X (根据 BOOT0 引脚和用户选项字节中的引导配置位（nBOOT1、nBOOT0、nSWBT0），X 表示 BOOT_MODE[1:0]，可能为任意值)

该寄存器只能按字（32 位）访问



位/位域	名称	描述
31:9	保留	必须保持复位值。
8	FMC_SWP	FMC存储器映射切换。 这些位控制主FLASH存储器的Bank0和Bank1的地址映射切换功能。 0：主FLASH存储器的Bank1被映射到地址0x0800 0000，主FLASH存储器的Bank0被映射到地址0x0804 0000 1：主FLASH存储器的Bank0映射到地址0x0800 0000，主FLASH存储器的Bank1映射到地址0x0804 0000 **注意**：以上地址取决于具体芯片系列的扇区大小。
7:3	保留	必须保持复位值。
2:0	BOOT_MODE[2:0]	引导模式 这些位选择映射到0x0000 0000的设备存储器 。 000：主FLASH存储器被映射到地址0x0000 0000 001：引导装载代码所在系统存储器被映射到地址0x0000 0000。 010：EXMC存储器地址被映射到地址0x0000 0000。 011：片上SRAM的SRAM0被映射到地址0x0000 0000。 101：QSPI存储器地址被映射到地址0x0000 0000 其他：保留。

1.7.2. 配置寄存器 1(SYSCFG_CFG1)

地址偏移：0x04

复位值：0x7C00 0000

该寄存器只能按字（32 位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IXIE	IDIE	OVFIE	UFIE	DZIE	IOPIE	保留		I2C3FMP EN	I2C2FMP EN	I2C1FMP EN	I2C0FMP EN	PB9FMP EN	PB8FMP EN	PB7FMP EN	PB6FMP EN
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															

位/位域	名称	描述
31	IXIE	不精确中断使能位 0：不精确中断禁能 1：不精确中断使能
30	IDIE	输入异常中断使能位 0：输入异常中断禁能 1：输入异常中断使能
29	OVFIE	溢出中断使能位 0：溢出中断禁能 1：溢出中断使能
28	UFIE	下溢中断使能位 0：下溢中断禁能 1：下溢中断使能
27	DZIE	除0中断使能位 0：除0中断禁能 1：除 0 中断使能
26	IOPIE	无效操作中断使能位 0：无效操作中断禁能 1：无效操作中断使能
25:24	保留	必须保持复位值。
23	I2C3FMPEN	I2C3 Fm+模式使能 该位控制 I2C3 的 Fm+功能。 0：禁能 Fm+模式

		1：使能 Fm+模式
22	I2C2FMPEN	I2C2 Fm+模式使能 该位控制 I2C2 的 Fm+功能。 0：禁能 Fm+模式 1：使能 Fm+模式
21	I2C1FMPEN	I2C1 Fm+模式使能 该位控制 I2C1 的 Fm+功能。 0：禁能 Fm+模式 1：使能 Fm+模式
20	I2C0FMPEN	I2C0 Fm+模式使能 该位控制 I2C0 的 Fm+功能。 0：禁能 Fm+模式 1：使能 Fm+模式
19	PB9FMPEN	PB9 引脚 Fm+模式使能 该位控制 I2C 的 Fm+功能，同时该引脚的速度控制被忽略 0：禁能 Fm+模式 1：使能 Fm+模式
18	PB8FMPEN	PB8 引脚 Fm+模式使能 该位控制 I2C 的 Fm+功能，同时该引脚的速度控制被忽略 0：禁能 Fm+模式 1：使能 Fm+模式
17	PB7FMPEN	PB7 引脚 Fm+模式使能 该位控制 I2C 的 Fm+功能，同时该引脚的速度控制被忽略 0：禁能 Fm+模式 1：使能 Fm+模式
16	PB6FMPEN	PB6 引脚 Fm+模式使能 该位控制 I2C 的 Fm+功能，同时该引脚的速度控制被忽略 0：禁能 Fm+模式 1：使能 Fm+模式
15:0	保留	必须保持复位值。

1.7.3. 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			

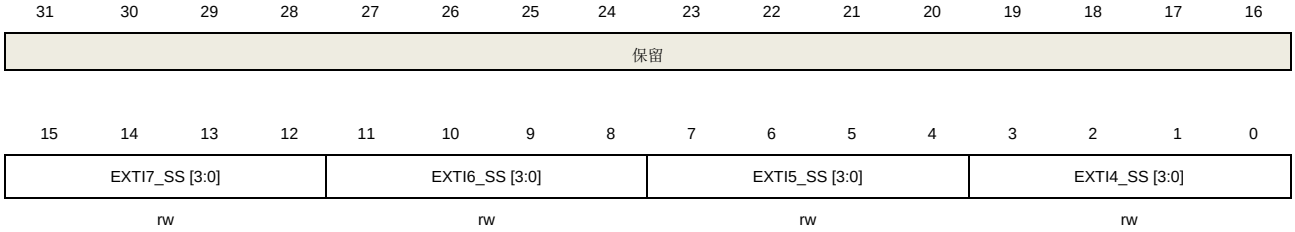
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI3_SS[3:0]	EXTI 3源选择 0000：PA3引脚 0001：PB3引脚 0010：PC3引脚 0011：PD3引脚 0100：PE3引脚 0101：PF3引脚 0110：PG3引脚 其他配置保留
11:8	EXTI2_SS[3:0]	EXTI 2源选择 0000：PA2引脚 0001：PB2引脚 0010：PC2引脚 0011：PD2引脚 0100：PE2引脚 0101：PF2引脚 0110：PG2引脚 其他配置保留
7:4	EXTI1_SS[3:0]	EXTI 1源选择 0000：PA1引脚 0001：PB1引脚 0010：PC1引脚 0011：PD1引脚 0100：PE1引脚 0101：PF1引脚 0110：PG1引脚 其他配置保留
3:0	EXTI0_SS[3:0]	EXTI 0 源选择

0000：PA0引脚
0001：PB0引脚
0010：PC0引脚
0011：PD0引脚
0100：PE0引脚
0101：PF0引脚
0110：PG0引脚
其他配置保留

1.7.4. 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



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI7_SS[3:0]	EXTI 7源选择 0000：PA7引脚 0001：PB7引脚 0010：PC7引脚 0011：PD7引脚 0100：PE7引脚 0101：PF7引脚 0110：PG7引脚 其他配置保留
11:8	EXTI6_SS[3:0]	EXTI 6源选择 0000：PA6引脚 0001：PB6引脚 0010：PC6引脚 0011：PD6引脚 0100：PE6引脚

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

1.7.5. 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 0000：PA11引脚

		0001：PB11引脚 0010：PC11引脚 0011：PD11引脚 0100：PE11引脚 0101：PF11引脚 其他配置保留
11:8	EXTI10_SS[3:0]	EXTI 10源选择 0000：PA10引脚 0001：PB10引脚 0010：PC10引脚 0011：PD10引脚 0100：PE10引脚 0101：PF10引脚 0110：PG10引脚 其他配置保留
7:4	EXTI9_SS[3:0]	EXTI 9源选择 0000：PA9引脚 0001：PB9引脚 0010：PC9引脚 0011：PD9引脚 0100：PE9引脚 0101：PF9引脚 0110：PG9引脚 其他配置保留
3:0	EXTI8_SS[3:0]	EXTI 8源选择 0000：PA8引脚 0001：PB8引脚 0010：PC8引脚 0011：PD8引脚 0100：PE8引脚 0101：PF8引脚 0110：PG8引脚 其他配置保留

1.7.6. 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			

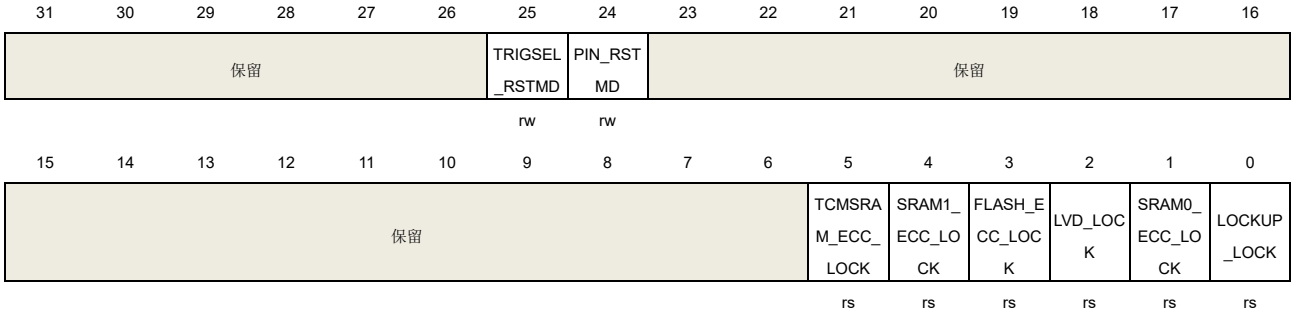
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI15_SS[3:0]	EXTI 15源选择 0000：PA15引脚 0001：PB15引脚 0010：PC15引脚 0011：PD15引脚 0100：PE15引脚 0101：PF15引脚 其他配置保留
11:8	EXTI14_SS[3:0]	EXTI 14源选择 0000：PA14引脚 0001：PB14引脚 0010：PC14引脚 0011：PD14引脚 0100：PE14引脚 0101：PF14引脚 其他配置保留
7:4	EXTI13_SS[3:0]	EXTI 13源选择 0000：PA13引脚 0001：PB13引脚 0010：PC13引脚 0011：PD13引脚 0100：PE13引脚 0101：PF13引脚 其他配置保留
3:0	EXTI12_SS[3:0]	EXTI 12源选择 0000：PA12引脚 0001：PB12引脚 0010：PC12引脚

0011：PD12引脚
0100：PE12引脚
0101：PF12引脚
其他配置保留

1.7.7. 配置寄存器 2(SYSCFG_CFG2)

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字（32 位）访问



位/位域	名称	描述
31:26	保留	必须保持复位值。
25	TRIGSEL_RSTMD	Trigsel复位模式 此位只能通过电源复位来复位 0：正常模式。当任何复位事件发生时，CLA触发器的触发输入源选择将复位 1：复位持续模式。CLA触发器的触发输入源选择将在除POR事件之外的任何复位事件中保持状态。
24	PIN_RSTMD	引脚复位模式 该位控制 GPIO 引脚配置（GPIOx_CTL，GPIOx_OMODE，GPIOx_OSPD, GPIOx_PUD，GPIOx_OCTL）在任何复位事件发生时是否复位。除了 POR 之外的任何复位事件都将保留该位。在发生 POR 事件后，该位被复位为 0。 0：GPIO 引脚配置在任何复位事件发生后复位 1：GPIO引脚配置将在除POR事件外任何复位事件中都保持状态
23:6	保留	必须保持复位值。
5	TCMSRAM_ECC_LOCK	TCMSRAM ECC锁定使能 该位由软件设置，仅在系统复位被清除。 该位用于使能和锁定TCMSRAM ECC错误连接到TIMER0/7/14/15/16/19的break输入。

		0：TCMSRAM ECC错误从TIMER0/7/14/15/16/19的break输入端和HRTIMER0的系统异常输入端断开。 1：TCMSRAM ECC与TIMER0/7/14/15/16/19的break输入端和HRTIMER0的系统异常输入端连接。
4	SRAM1_ECC_LOCK	SRAM1 ECC锁定使能 该位由软件设置，仅在系统复位被清除。 该位用于使能和锁定SRAM1 ECC错误连接到TIMER0/7/14/15/16/19的break输入。 0：SRAM1 ECC错误从TIMER0/7/14/15/16/19的break输入端和HRTIMER0的系统异常输入端断开。 1：SRAM1 ECC错误与TIMER0/7/14/15/16/19的break输入端和HRTIMER0的系统异常输入端连接。
3	FLASH_ECC_LOCK	Flash ECC 锁定使能 该位由软件设置，仅在系统复位被清除。 0：ECC错误从TIMER0/7/14/15/16/19的break输入端和HRTIMER0的系统异常输入端断开。 1：ECC错误与TIMER0/7/14/15/16/19的break输入端和HRTIMER0的系统异常输入端连接。
2	LVD_LOCK	LVD 锁定使能 0：LVD 中断从 TIMER0/7/14/15/16/19 的 break 输入端和 HRTIMER0 的系统异常输入端断开。 1：LVD 中断与 TIMER0/7/14/15/16/19 的 break 输入端和 HRTIMER0 的系统异常输入端连接。
1	SRAM0_ECC_LOCK	SRAM0 ECC 锁定使能 该位由软件设置，仅在系统复位被清除。 0：SRAM0 ECC 错误从 TIMER0/7/14/15/16/19 的 break 输入端和 HRTIMER0 的系统异常输入端断开。 1：SRAM0 ECC 错误与 TIMER0/7/14/15/16/19 的 break 输入端和 HRTIMER0 的系统异常输入端连接。
0	LOCKUP_LOCK	Cortex®-M33 lockup (Hardfault)输出锁定使能 该位由软件设置，仅在系统复位被清除。 0：Cortex®-M33 LOCKUP 输出从 TIMER0/7/14/15/16/19 的 break 输入端和 HRTIMER0 的系统异常输入端断开。 1：Cortex®-M33 LOCKUP 输出与 TIMER0/7/14/15/16/19 的 break 输入端和 HRTIMER0 的系统异常输入端链接。

1.7.8. 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							TCMSRAMECCSEIF	TCMSRAMECCMEIF	SRAM1ECCSEIF	SRAM1ECCMEIF	NMIPINIF	CKMNMIIF	FLASHECCIF	SRAM0ECCSEIF	SRAM0ECCMEIF
							rc_w1	rc_w1	rc_w1	rc_w1	r	rc_w1	rc_w1	rc_w1	rc_w1

位/位域	名称	描述
31:9	保留	必须保持复位值。
8	TCMSRAMECCSEIF	TCMSRAM 单比特可纠错事件中断标志 软件写 1 清零。 0：没有检测到 TCMSRAM ECC 错误事件。 1：检测到 TCMSRAM ECC 错误事件。
7	TCMSRAMECCMEIF	TCMSRAM 多比特（两比特）不可纠错事件中断标志 软件写 1 清零。 0：没有检测到 TCMSRAM ECC 多比特（两比特）不可纠错事件。 1：检测到 TCMSRAM ECC 多比特（两比特）不可纠错事件。 **注意：**在 TCMSRAMECCMEIE 位置位时，TCMSRAM 多比特（两比特）不可纠错事件将产生 NMI 中断。
6	SRAM1ECCSEIF	SRAM1 单比特可纠错事件中断标志 软件写 1 清零。 0：没有检测到 SRAM1 ECC 错误事件。 1：检测到 SRAM1 ECC 错误事件。
5	SRAM1ECCMEIF	SRAM1 多比特（两比特）不可纠错事件中断标志 软件写 1 清零。 0：没有检测到 SRAM1 ECC 多比特（两比特）不可纠错事件。 1：检测到 SRAM1 ECC 多比特（两比特）不可纠错事件。 **注意：**在 SRAM1ECCMEIE 位置位时，SRAM1 多比特（两比特）不可纠错事件将产生 NMI 中断。
4	NMIPINIF	来自 NMI 引脚的中断标志 0：无输入。 1：来自 NMI 引脚的中断输入。
3	CKNMIIF	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 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	SRAM0ECCSEIF	SRAM0 单比特可纠错事件中断标志 软件写 1 清零。 0：没有检测到 SRAM0 ECC 错误事件。 1：检测到 SRAM0 ECC 错误事件。
0	SRAM0ECCMEIF	SRAM0 多比特（两比特）纠正错误事件中断标志 软件写 1 清零。 0：没有检测到 SRAM0 ECC 多比特（两比特）不可纠错事件。 1：检测到 SRAM0 ECC 多比特（两比特）不可纠错事件。 **注意：**在 SRAM0ECCMEIE 位置位时，SRAM0 多比特（两比特）不可纠错事件将产生 NMI 中断。

1.7.9. 配置寄存器 3(SYSCFG_CFG3)

地址偏移：0x20
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
SRAM0ECCEADDR[14:0]															保留
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRAM0ECCSERRBITS[5:0]						保留					NMIPINIE	CKMNMIIE	FLASHECCIE	SRAM0ECCSEIE	SRAM0ECCMEIE
r											rw	rw	rw	rw	rw

位/位域	名称	描述
31:17	SRAM0ECCEADDR[14:0]	记录上一次发生 SRAM0 ECC 事件的 SRAM0 故障系统地址（SRAM0ECCEADDR = SRAM0 address[16:0] >> 2）
16	保留	必须保持复位值。
15:10	SRAM0ECCSERRBITS[5:0]	发生 ECC 单比特可纠错事件的比特 0：无错误

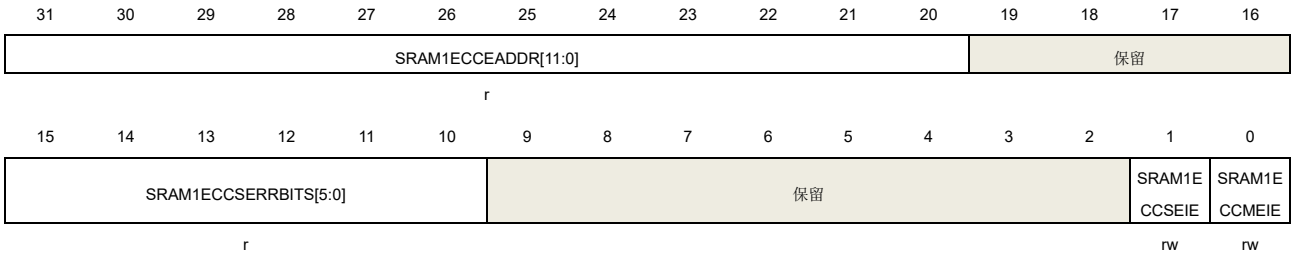
		1：比特 0 … 32: 比特 31
9:5	保留	必须保持复位值。
4	NMIPINIE	NMI 引脚中断使能 0：禁能 NMI 引脚中断。 1：使能 NMI 引脚中断。
3	CKNMIIE	HXTAL 时钟检测故障 NMI 中断使能 0：禁能 HXTAL 时钟检测故障 NMI 中断。 1：使能 HXTAL 时钟检测故障 NMI 中断。
2	FLASHECCIE	Flash ECC NMI 中断使能 0：禁能 Flash ECC NMI 中断。 1：使能 Flash ECC NMI 中断。
1	SRAM0ECCSEIE	SRAM0 单比特纠错事件标志连接到 NVIC 这个位可以用来启用 SRAM0 单比特纠错事件标志连接到 NVIC。 0：禁能 SRAM0 单比特可纠错中断。 1：使能 SRAM0 单比特可纠错中断。
0	SRAM0ECCMEIE	SRAM0 多比特（两比特）不可纠错中断志连接到 NMI 这个位可以用来启用 SRAM0 多比特纠错事件标志连接到 NMI。 0：禁能 SRAM0 多比特（两比特）不可纠错中断。 1：使能 SRAM0 多比特（两比特）不可纠错中断。

1.7.10. 配置寄存器 4(SYSCFG_CFG4)

地址偏移：0x24

复位值：0xXXX0 XX03

该寄存器只能按字（32 位）访问



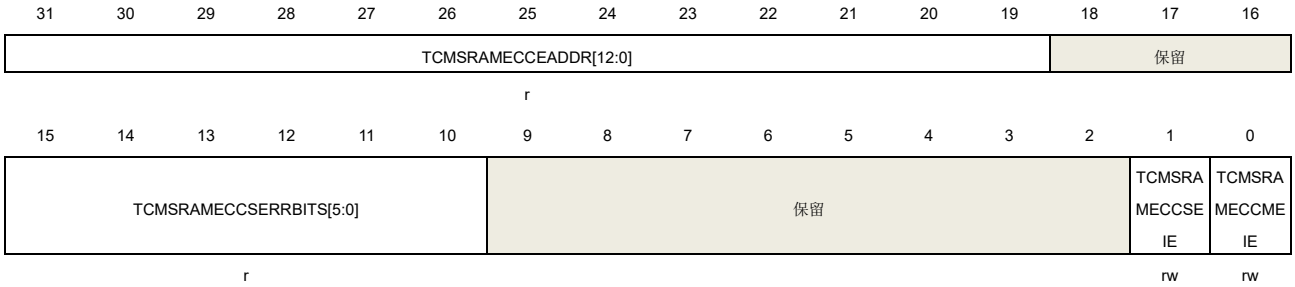
位/位域	名称	描述
31:20	SRAM1ECCEADDR[1	记录上一次发生 SRAM1 ECC 事件的 SRAM1 故障系统地址（SRAM1ECCEADDR =

	1:0]	SRAM1 address[13:0] >> 2）
19:16	保留	必须保持复位值。
15:10	SRAM1ECCSERRBITS[5:0]	发生 ECC 单比特可纠错事件的比特 0：无错误 1：比特 0 … 32：比特 31
9:2	保留	单比特可纠错中断。
1	SRAM1ECCSEIE	SRAM1 单比特纠错事件标志连接到 NVIC 这个位可以用来启用 SRAM1 单比特纠错事件标志连接到 NVIC。 0：禁能 SRAM1 单比特可纠错中断。 1：使能 SRAM1 单比特可纠错中断。
0	SRAM1ECCMEIE	SRAM1 多比特（两比特）不可纠错中断志连接到 NMI 这个位可以用来启用 SRAM1 多比特纠错事件标志连接到 NMI。 0：禁能 SRAM1 多比特（两比特）不可纠错 NMI 中断。 1：使能 SRAM1 多比特（两比特）不可纠错 NMI 中断。

1.7.11. 配置寄存器 5(SYSCFG_CFG5)

地址偏移：0x28
复位值：0xXXXX XX03

该寄存器只能按字（32 位）访问



位/位域	名称	描述
31:19	TCMSRAMECCEADDR[12:0]	记录上一次发生 TCMSRAM ECC 事件的 TCMSRAM 故障系统地址（TCMSRAMECCEADDR = TCM SRAM address[14:0] >> 2）
18:16	保留	必须保持复位值。
15:10	TCMSRAMECCSERRBITS[5:0]	发生 TCMSRAM ECC 单比特可纠错事件的比特 0：无错误

		1：比特 0 … 32：比特 31
9:2	保留	必须保持复位值。
1	TCMSRAMECCSEIE	TCMSRAM 单比特纠错事件标志连接到 NVIC 这个位可以用来启用 TCMSRAM 单比特纠错事件标志连接到 NVIC。 0：禁能 TCMSRAM 单比特可纠错中断。 1：使能 TCMSRAM 单比特可纠错中断。
0	TCMSRAMECCMEIE	TCMSRAM 多比特（两比特）不可纠错中断志连接到 NMI 这个位可以用来启用 TCMSRAM 多比特纠错事件标志连接到 NMI。 0：禁能 TCMSRAM 多比特（两比特）不可纠错 NMI 中断。 1：使能 TCMSRAM 多比特（两比特）不可纠错 NMI 中断。

1.7.12. TCMSRAM 控制与状态寄存器(SYSCFG_TCMSRAMCS)

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														TCMSRAMBSY	TCMSRAMERS
														r	rw

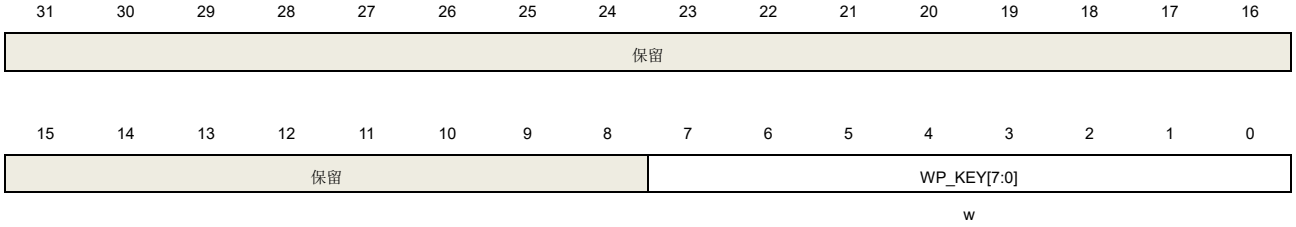
位/位域	名称	描述
31:2	保留	必须保持复位值。
1	TCMSRAMBSYF	TCMSRAM 擦除操作忙标志 0：无 TCMSRAM 擦除操作。 1：TCMSRAM 正在擦除中。
0	TCMSRAMERS	TCMSRAM 擦除 TCMSRAM 硬件擦除操作将在该位置 1 时开始，并在 TCMSRAM 擦除操作结束时自动清除。在擦除 TCMSRAM 之前须先解锁 TCMSRAM。

1.7.13. TCMSRAM 密钥寄存器(SYSCFG_TCMSRAMKEY)

地址偏移：0x34

复位值：0x0000 0000

该寄存器只能按字（32 位）访问



位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	WP_KEY[7:0]	TCMSRAMERS 写保护密钥 按照以下步骤解锁 SYSCFG_TCMSRAMCS 寄存器中 TCMSRAMERS 位的写保护。 步骤 1、将“0xCA”写入 WP_KEY[7:0]。 步骤 2、将“0x53”写入 WP_KEY[7:0]。 **注意：**错误的密钥将重新激活 TCMSRAMERS 写保护。

1.7.14. TCMSRAM 写保护寄存器(SYSCFG_TCMSRAMWP)

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
P31WPEN	P30WPEN	P29WPEN	P28WPEN	P27WPEN	P26WPEN	P25WPEN	P24WPEN	P23WPEN	P22WPEN	P21WPEN	P20WPEN	P19WPEN	P18WPEN	P17WPEN	P16WPEN
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15WPEN	P14WPEN	P13WPEN	P12WPEN	P11WPEN	P10WPEN	P9WPEN	P8WPEN	P7WPEN	P6WPEN	P5WPEN	P4WPEN	P3WPEN	P2WPEN	P1WPEN	P0WPEN
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

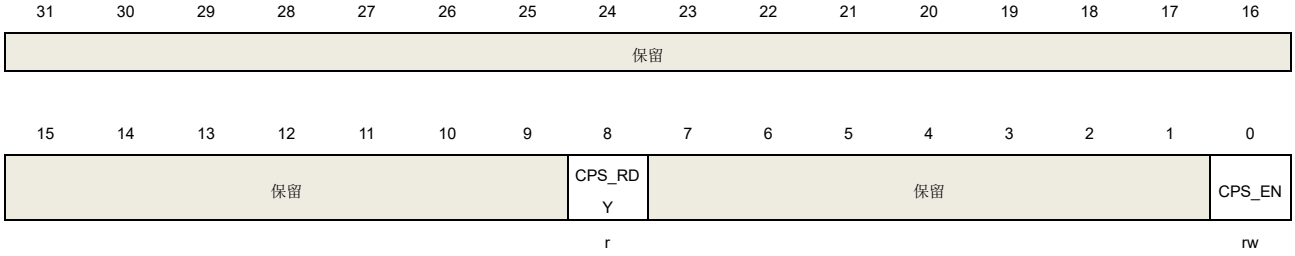
位/位域	名称	描述
31:0	PxWPEN(x=0..31)	TCMSRAM 第 x 页写保护使能 这些位由软件设置，并仅在系统复位时清除 0：禁能 1：使能

1.7.15. I/O 补偿控制寄存器(SYSCFG_CPSCTL)

地址偏移：0x3C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问



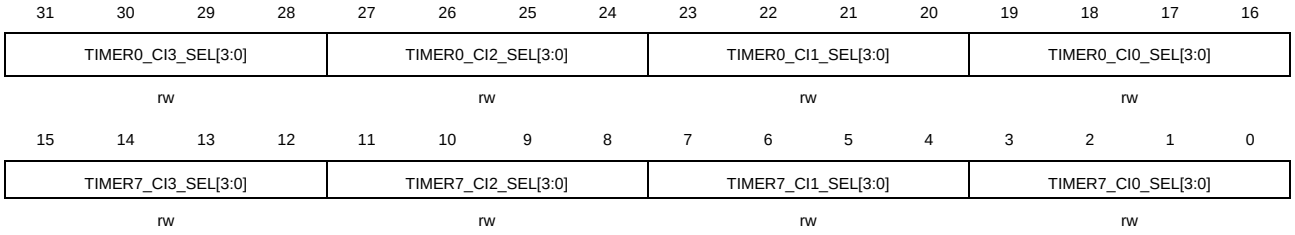
位/位域	名称	描述
31:9	保留	必须保持复位值。
8	CPS_RDY	I/O 补偿单元是否准备好 该位只读 0：I/O 补偿单元未准备好 1：I/O 补偿单元准备好
7:1	保留	必须保持复位值。
0	CPS_EN	I/O 补偿单元使能 该位使能 I/O 补偿单元。 0：禁能 I/O 补偿单元 1：使能 I/O 补偿单元

1.7.16. TIMER 输入源选择寄存器 0 (SYSCFG_TIMERCISEL0)

地址偏移：0x44

复位值：0x0000 0000

该寄存器只能按字（32 位）访问



位/位域	名称	描述

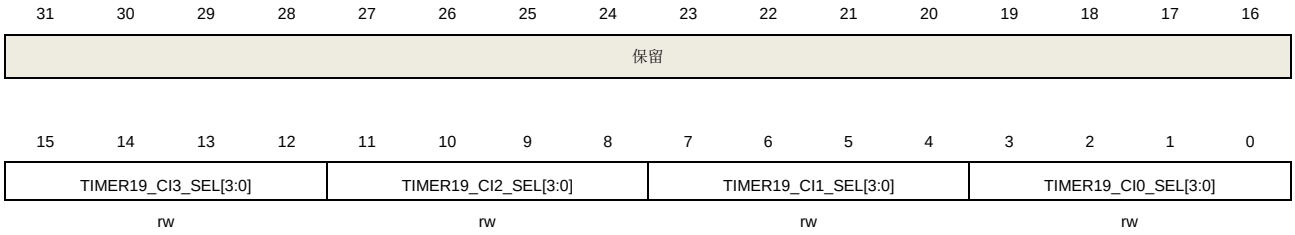
31:28	TIMER0_CI3_SEL[3:0]	选择 TIMER0_CI3 的输入源 0000：TIMER0_CH3 0001：CLA3OUT 其他：保留
27:24	TIMER0_CI2_SEL[3:0]	选择 TIMER0_CI2 的输入源 0000：TIMER0_CH2 0001：CLA2OUT 其他：保留
23:20	TIMER0_CI1_SEL[3:0]	选择 TIMER0_CI1 的输入源 0000：TIMER0_CH1 0001：CLA1OUT 其他：保留
19:16	TIMER0_CI0_SEL[3:0]	选择 TIMER0_CI0 的输入源 0000：TIMER0_CH0 0001：CMP0 输出 0010：CMP1 输出 0011：CMP2 输出 0100：CMP3 输出 0101：CLA0OUT 其他：保留
15:12	TIMER7_CI3_SEL[3:0]	选择 TIMER7_CI3 的输入源 0000：TIMER7_CH3 0001：CLA3OUT 其他：保留
11:8	TIMER7_CI2_SEL[3:0]	选择 TIMER7_CI2 的输入源 0000：TIMER7_CH2 0001：CLA2OUT 其他：保留
7:4	TIMER7_CI1_SEL[3:0]	选择 TIMER7_CI1 的输入源 0000：TIMER7_CH1 0001：CLA1OUT 其他：保留
3:0	TIMER7_CI0_SEL[3:0]	选择 TIMER7_CI0 的输入源 0000：TIMER7_CH0 0001：CMP0 输出 0010：CMP1 输出 0011：CMP2 输出

0100：CMP3 输出
0101：CLA0OUT
其他：保留

1.7.17. TIMER 输入源选择寄存器 1 (SYSCFG_TIMERCISEL1)

地址偏移：0x48
复位值：0x0000 0000

该寄存器只能按字（32 位）访问



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	TIMER19_CI3_SEL[3:0]	选择 TIMER19_CI3 的输入源 0000：TIMER19_CH3 0001：CLA3OUT 其他：保留
11:8	TIMER19_CI2_SEL[3:0]	选择 TIMER19_CI2 的输入源 0000：TIMER19_CH2 0001：CLA2OUT 其他：保留
7:4	TIMER19_CI1_SEL[3:0]	选择 TIMER19_CI1 的输入源 0000：TIMER19_CH1 0001：CLA1OUT 其他：保留
3:0	TIMER19_CI0_SEL[3:0]	选择 TIMER19_CI0 的输入源 0000：TIMER19_CH0 0001：CMP0 输出 0010：CMP1 输出 0011：CMP2 输出 0100：CMP3 输出 0101：CLA0OUT

其他：保留

1.7.18. TIMER 输入源选择寄存器 2 (SYSCFG_TIMERCISEL2)

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
TIMER1_CI3_SEL[3:0]				TIMER1_CI2_SEL[3:0]				TIMER1_CI1_SEL[3:0]				TIMER1_CI0_SEL[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER2_CI3_SEL[3:0]				TIMER2_CI2_SEL[3:0]				TIMER2_CI1_SEL[3:0]				TIMER2_CI0_SEL[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:28	TIMER1_CI3_SEL[3:0]	选择 TIMER1_CI3 的输入源 0000：TIMER1_CH3 0001：CMP0 输出 0010：CMP1 输出 0011：CLA3OUT 其他：保留
27:24	TIMER1_CI2_SEL[3:0]	选择 TIMER1_CI2 的输入源 0000：TIMER1_CH2 0001：CMP3 输出 0010：CLA2OUT 其他：保留
23:20	TIMER1_CI1_SEL[3:0]	选择 TIMER1_CI1 的输入源 0000：TIMER1_CH1 0001：CMP0 输出 0010：CMP1 输出 0011：CMP2 输出 0100：CMP3 输出 0101：CMP5 输出 0110：CLA1OUT 其他：保留
19:16	TIMER1_CI0_SEL[3:0]	选择 TIMER1_CI0 的输入源 0000：TIMER1_CH0 0001：CMP0 输出

0010：CMP1 输出
0011：CMP2 输出
0100：CMP3 输出
0101：CMP4 输出
0110：CLA0OUT
其他：保留

15:12 TIMER2_CI3_SEL[3:0] 选择 TIMER2_CI3 的输入源
0000：TIMER2_CH3
0001：CLA3OUT
其他：保留

11:8 TIMER2_CI2_SEL[3:0] 选择 TIMER2_CI2 的输入源
0000：TIMER2_CH2
0001：CMP2 输出
0010：CLA2OUT
其他：保留

7:4 TIMER2_CI1_SEL[3:0] 选择 TIMER2_CI1 的输入源
0000：TIMER2_CH1
0001：CMP0 输出
0010：CMP1 输出
0011：CMP2 输出
0100：CMP3 输出
0101：CMP4 输出
0110：CMP5 输出
0111：CMP6 输出
1000：CLA1OUT
其他：保留

3:0 TIMER2_CI0_SEL[3:0] 选择 TIMER2_CI0 的输入源
0000：TIMER2_CH0
0001：CMP0 输出
0010：CMP1 输出
0011：CMP2 输出
0100：CMP3 输出
0101：CMP4 输出
0110：CMP5 输出
0111：CMP6 输出
1000：CLA0OUT
其他：保留

1.7.19. TIMER 输入源选择寄存器 3 (SYSCFG_TIMERCISEL3)

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
TIMER3_CI3_SEL[3:0]				TIMER3_CI2_SEL[3:0]				TIMER3_CI1_SEL[3:0]				TIMER3_CI0_SEL[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER4_CI3_SEL[3:0]				TIMER4_CI2_SEL[3:0]				TIMER4_CI1_SEL[3:0]				TIMER4_CI0_SEL[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:28	TIMER3_CI3_SEL[3:0]	选择 TIMER3_CI3 的输入源 0000：TIMER3_CH3 0001：CMP5 输出 0010：CLA3OUT 其他：保留
27:24	TIMER3_CI2_SEL[3:0]	选择 TIMER3_CI2 的输入源 0000：TIMER3_CH2 0001：CMP4 输出 0010：CLA2OUT 其他：保留
23:20	TIMER3_CI1_SEL[3:0]	选择 TIMER3_CI1 的输入源 0000：TIMER3_CH1 0001：CMP0 输出 0010：CMP1 输出 0011：CMP2 输出 0100：CMP3 输出 0101：CMP4 输出 0110：CMP5 输出 0111：CMP6 输出 1000：CLA1OUT 其他：保留
19:16	TIMER3_CI0_SEL[3:0]	选择 TIMER3_CI0 的输入源 0000：TIMER3_CH0 0001：CMP0 输出 0010：CMP1 输出

0011：CMP2 输出
0100：CMP3 输出
0101：CMP4 输出
0110：CMP5 输出
0111：CMP6 输出
1000：CLA0OUT
其他：保留

15:12 TIMER4_CI3_SEL[3:0] 选择 TIMER4_CI3 的输入源
0000：TIMER4_CH3
0001：CLA3OUT
其他：保留

11:8 TIMER4_CI2_SEL[3:0] 选择 TIMER4_CI2 的输入源
0000：TIMER4_CH2
0001：CLA2OUT
其他：保留

7:4 TIMER4_CI1_SEL[3:0] 选择 TIMER4_CI1 的输入源
0000：TIMER4_CH1
0001：CMP0 输出
0010：CMP1 输出
0011：CMP2 输出
0100：CMP3 输出
0101：CMP4 输出
0110：CMP5 输出
0111：CMP6 输出
1000：CLA1OUT
其他：保留

3:0 TIMER4_CI0_SEL[3:0] 选择 TIMER4_CI0 的输入源
0000：TIMER4_CH0
0001：IRC32K
0010：LXTAL
0011：RTC 唤醒
0100：CMP0 输出
0101：CMP1 输出
0110：CMP2 输出
0111：CMP3 输出
1000：CMP4 输出
1001：CMP5 输出
1010：CMP6 输出

1011：CLA0OUT

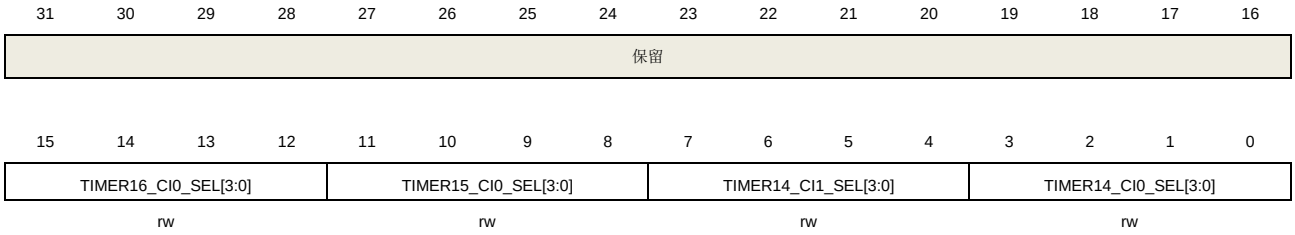
其他：保留

1.7.20. TIMER 输入源选择寄存器 4 (SYSCFG_TIMERCISEL4)

地址偏移：0x54

复位值：0x0000 0000

该寄存器只能按字（32 位）访问



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	TIMER16_CI0_SEL[3:0]	选择 TIMER16_CI0 的输入源 0000：TIMER16_CH0 0001：CMP4 输出 0010：CKOUT 0011：HXTAL/32 0100：RTC 时钟 0101：LXTAL 0110：IRC32K 0111：CLA0OUT 1000：CLA1OUT 1001：CLA2OUT 1010：CLA3OUT 1011：HXTAL 其他：保留
11:8	TIMER15_CI0_SEL[3:0]	选择 TIMER15_CI0 的输入源 0000：TIMER15_CH0 0001：CMP5 输出 0010：CKOUT 0011：HXTAL/32 0100：RTC 时钟 0101：LXTAL

0110：IRC32K
0111：CLA0OUT
1000：CLA1OUT
1001：CLA2OUT
1010：CLA3OUT
1011：HXTAL
其他：保留

7:4	TIMER14_CI1_SEL[3:0]	选择 TIMER14_CI1 的输入源 0000：TIMER14_CH1 0001：CMP1 输出 0010：CMP2 输出 0011：CMP5 输出 0100：CMP6 输出 其他：保留
3:0	TIMER14_CI0_SEL[3:0]	选择 TIMER14_CI0 的输入源 0000：TIMER14_CH0 0001：LXTAL 0010：CMP0 输出 0011：CMP1 输出 0100：CMP4 输出 0101：CMP6 输出 0110：CLA0OUT 0111：CLA1OUT 1000：CLA2OUT 1001：CLA3OUT 其他：保留

1.7.21. TIMERx 配置寄存器 0（SYSCFG_TIMERxCFG0, x=0，1，2，3，4，7，19）

地址偏移：0x100适用于TIMER0
地址偏移：0x10C适用于TIMER1
地址偏移：0x118适用于TIMER2
地址偏移：0x124适用于TIMER3
地址偏移：0x130适用于TIMER4
地址偏移：0x13C适用于TIMER7
地址偏移：0x154适用于TIMER19
复位值：0x0000 0000

TSCFG0[4:0]，TSCFG1[4:0]..TSCFG14[4:0]之间相互互斥，不能同时配置。

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	TSCFG5[4:0]					TSCFG4[4:0]					TSCFG3[4:0]				
	rw					rw					rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TSCFG2[4:0]					TSCFG1[4:0]					TSCFG0[4:0]				
	rw					rw					rw				

位/位域	名称	描述
31	保留	必须保持复位值。
30:26	TSCFG5[4:0]	事件模式配置 计数器在触发输入的上升沿启动。 00000：事件模式禁能 00001：内部触发输入 0（ITI0） 00010：内部触发输入 1（ITI1） 00011：内部触发输入 2（ITI2） 00100：内部触发输入 3（ITI3） 00101：CI0 的边沿标志位（CI0F_ED） 00110：滤波后的通道 0 输入（CI0FE0） 00111：滤波后的通道 1 输入（CI1FE1） 01000：滤波后的外部触发输入（ETIFP） 01001：内部触发输入 4（ITI4） 01010：内部触发输入 5（ITI5） 01011：内部触发输入 6（ITI6） 01100：内部触发输入 7（ITI7） 01101：内部触发输入 8（ITI8） 01110：内部触发输入 9（ITI9） 01111：内部触发输入 10（ITI10） 10000：保留 10001：保留 10010：保留 10011：内部触发输入 14（ITI14） 其他值：保留
25:21	TSCFG4[4:0]	暂停模式配置 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，且 TSCFG4[4:0]不为零，则计数器停止。 00000：暂停模式禁能 00001：内部触发输入 0（ITI0） 00010：内部触发输入 1（ITI1）

00011：内部触发输入 2（ITI2）
00100：内部触发输入 3（ITI3）
00101：保留
00110：滤波后的通道 0 输入（CI0FE0）
00111：滤波后的通道 1 输入（CI1FE1）
01000：滤波后的外部触发输入（ETIFP）
01001：内部触发输入 4（ITI4）
01010：内部触发输入 5（ITI5）
01011：内部触发输入 6（ITI6）
01100：内部触发输入 7（ITI7）
01101：内部触发输入 8（ITI8）
01110：内部触发输入 9（ITI9）
01111：内部触发输入 9（ITI10）
10000：保留
10001：保留
10010：保留
10011：内部触发输入 14（ITI14）
其他：保留

20:16	TSCFG3[4:0]	复位模式配置 当 TSCFG3[4:0]不为零，选中的触发输入的上升沿重新初始化计数器，并且更新影子寄存器。 00000：复位模式禁能 00001：内部触发输入 0（ITI0） 00010：内部触发输入 1（ITI1） 00011：内部触发输入 2（ITI2） 00100：内部触发输入 3（ITI3） 00101：CI0 的边沿标志位（CI0F_ED） 00110：滤波后的通道 0 输入（CI0FE0） 00111：滤波后的通道 1 输入（CI1FE1） 01000：滤波后的外部触发输入（ETIFP） 01001：内部触发输入 4（ITI4） 01010：内部触发输入 5（ITI5） 01011：内部触发输入 6（ITI6） 01100：内部触发输入 7（ITI7） 01101：内部触发输入 8（ITI8） 01110：内部触发输入 9（ITI9） 01111：内部触发输入 10（ITI10） 10000：保留 10001：保留 10010：保留

		10011：内部触发输入 14（ITI14） 其他：保留
15	保留	必须保持复位值。
14:10	TSCFG2[4:0]	正交译码器模式 2 配置 00000：编码器模式 2 禁能 其他：根据另一个信号的输入电平，计数器在 CI0FE0 和 CI1FE1 的边沿开始计数。
9:5	TSCFG1[4:0]	正交译码器模式 1 配置 00000：编码器模式 1 禁能 其他：根据 CI0FE0 的电平，计数器在 CI1FE1 的边沿开始计数。
4:0	TSCFG0[4:0]	正交译码器模式 0 配置 00000：正交译码器模式 0 禁能 其他：根据 CI1FE1 的电平，计数器在 CI0FE0 的边沿开始计数。

1.7.22. TIMERx 配置寄存器 1（SYSCFG_TIMERxCFG1, x=0，1，2，3，4，7，19）

地址偏移：TIMER0为0x104
地址偏移：TIMER1为0x110
地址偏移：TIMER2为0x11C
地址偏移：TIMER3为0x128
地址偏移：TIMER4为0x134
地址偏移：TIMER7为0x140
地址偏移：TIMER19为0x158
复位值：0x0000 0000

TSCFG0[4:0]，TSCFG1[4:0]..TSCFG14[4:0]之间相互互斥，不能同时配置。

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	TSCFG11[4:0]					TSCFG10[4:0]					TSCFG9[4:0]				
	rw					rw					rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TSCFG8[4:0]					TSCFG7[4:0]					TSCFG6[4:0]				
	rw					rw					rw				

位/位域	名称	描述
31	保留	必须保持复位值。
30:26	TSCFG11[4:0]	译码器模式 2 配置 00000：禁能

		其他值：计数器将在 CI0FE0 和 CI1FE1 信号的上升沿和下降沿计数，计数方向由 CH0P 位和 CH1P 位决定。
25:21	TSCFG10[4:0]	译码器模式 1 配置 00000：禁能 其他值：CI0FE0 作为计数方向信号，CI1FE1 信号作为计数脉冲。计数器在 CI1FE1 的边沿开始计数，该边沿由 CH1P 设置。
20:16	TSCFG9[4:0]	译码器模式 0 配置 00000：禁能 其他值：CI0FE0 作为计数方向信号，CI1FE1 信号作为计数脉冲。计数器在 CI1FE1 的上升沿和下降沿开始计数。
15	保留	必须保持复位值。
14:10	TSCFG8[4:0]	暂停+复位模式配置 当触发输入的上升沿或下降沿(由 TIMERx_SMCFG 寄存器中的 PRMRPSEL 位配置)到来时，计数器将复位。当触发输入高时计数器计数，当触发输入低时计数器停止。在这种模式下，计数器的开始和停止可以被控制。 00000：暂停+复位模式禁能 00001：内部触发输入 0（ITI0） 00010：内部触发输入 1（ITI1） 00011：内部触发输入 2（ITI2） 00100：内部触发输入 3（ITI3） 00101：CI0 的边沿标志位（CI0F_ED） 00110：滤波后的通道 0 输入（CI0FE0） 00111：滤波后的通道 1 输入（CI1FE1） 01000：滤波后的外部触发输入（ETIFP） 01001：内部触发输入 4（ITI4） 01010：内部触发输入 5（ITI5） 01011：内部触发输入 6（ITI6） 01100：内部触发输入 7（ITI7） 01101：内部触发输入 8（ITI8） 01110：内部触发输入 9（ITI9） 01111：内部触发输入 10（ITI10） 10000：保留 10001：保留 10010：保留 10011：内部触发输入 14（ITI14） 其他值：保留
9:5	TSCFG7[4:0]	复位+事件模式配置 选中的触发输入的上升沿重新初始化计数器，并且更新影子寄存器。

00000：复位+事件模式禁能
00001：内部触发输入 0（ITI0）
00010：内部触发输入 1（ITI1）
00011：内部触发输入 2（ITI2）
00100：内部触发输入 3（ITI3）
00101：CI0 的边沿标志位（CI0F_ED）
00110：滤波后的通道 0 输入（CI0FE0）
00111：滤波后的通道 1 输入（CI1FE1）
01000：滤波后的外部触发输入（ETIFP）
01001：内部触发输入 4（ITI4）
01010：内部触发输入 5（ITI5）
01011：内部触发输入 6（ITI6）
01100：内部触发输入 7（ITI7）
01101：内部触发输入 8（ITI8）
01110：内部触发输入 9（ITI9）
01111：内部触发输入 10（ITI10）
10000：保留
10001：保留）
10010：保留
10011：内部触发输入 14（ITI14）
其他：保留

4:0	TSCFG6[4:0]	外部时钟模式 0 配置 在 TSCFG6[4:0]非零时，选中的触发输入的上升沿驱动计数器。 00000：外部时钟模式 0 禁能 00001：内部触发输入 0（ITI0） 00010：内部触发输入 1（ITI1） 00011：内部触发输入 2（ITI2） 00100：内部触发输入 3（ITI3） 00101：CI0 的边沿标志位（CI0F_ED） 00110：滤波后的通道 0 输入（CI0FE0） 00111：滤波后的通道 1 输入（CI1FE1） 01000：滤波后的外部触发输入（ETIFP） 01001：内部触发输入 4（ITI4） 01010：内部触发输入 5（ITI5） 01011：内部触发输入 5（ITI6） 01100：内部触发输入 7（ITI7） 01101：内部触发输入 8（ITI8） 01110：内部触发输入 9（ITI9） 01111：内部触发输入 10（ITI10） 10000：保留

10001：保留
10010：保留
10011：内部触发输入 14（ITI14）
其他：保留

1.7.23. TIMERx 配置寄存器 2（SYSCFG_TIMERxCFG2, x=0，1，2，3，4，7，19）

地址偏移：TIMER0为0x108
地址偏移：TIMER1为0x114
地址偏移：TIMER2为0x120
地址偏移：TIMER3为0x12C
地址偏移：TIMER4为0x138
地址偏移：TIMER7为0x144
地址偏移：TIMER19为0x15C
复位值：0x0000 0000

TSCFG0[4:0]，TSCFG1[4:0]..TSCFG14[4:0]之间相互互斥，不能同时配置。

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留											TSCFG15[4:0]				
													rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TSCFG14[4:0]					TSCFG13[4:0]					TSCFG12[4:0]				

位/位域	名称	描述
31:21	保留	必须保持复位值。
20:16	TSCFG15[4:0]	内部触发输入源配置 00000：保留 00001：内部触发输入 0（ITI0） 00010：内部触发输入 1（ITI1） 00011：内部触发输入 2（ITI2） 00100：内部触发输入 3（ITI3） 00101：CI0 的边沿标志位（CI0F_ED） 00110：保留 00111：保留 01000：保留 01001：内部触发输入 4（ITI4） 01010：内部触发输入 5（ITI5） 01011：内部触发输入 6（ITI6）

		01100：内部触发输入 7（ITI7） 01101：内部触发输入 8（ITI8） 01110：内部触发输入 9（ITI9） 01111：内部触发输入 10（ITI10） 10000：保留 10001：保留 10010：保留 10011：内部触发输入 14（ITI14） 其他：保留
15	保留	必须保持复位值。
14:10	TSCFG14[4:0]	正交译码器模式 4 配置 00000：禁能 其他：根据 CI0FE1 的电平，计数器仅在 CI1FE1 的边沿开始计数。
9:5	TSCFG13[4:0]	正交译码器模式 3 配置 00000：禁能 其他值：根据 CI0FE0 的电平，计数器仅在 CI1FE0 的边沿开始计数。
4:0	TSCFG12[4:0]	译码器模式3配置 00000：禁用 其他：计数器计数CI0FE0和CI1FE1信号的上升沿或下降沿。当CHxP=0时，计数器将在高电平或下降沿或CIxFEx信号上开始计数；当CHxP=1时，计数器将在低电平或上升沿或CIxFEx信号上开始计数。

1.7.24. TIMERx 配置寄存器 0（SYSCFG_TIMERxCFG0, x=14）

地址偏移：TIMER14为0x148
复位值：0x0000 0000

TSCFG0[4:0]，TSCFG1[4:0]..TSCFG7[4:0]之间相互互斥，不能同时配置。

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	TSCFG5[4:0]					TSCFG4[4:0]					TSCFG3[4:0]				
	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	保留	必须保持复位值。
30:26	TSCFG5[4:0]	事件模式配置 计数器在触发输入的上升沿启动。 00000：事件模式禁能 00001：内部触发输入 0（ITI0） 00010：内部触发输入 1（ITI1） 00011：内部触发输入 2（ITI2） 00100：内部触发输入 3（ITI3） 00101：CI0 的边沿标志位（CI0F_ED） 00110：滤波后的通道 0 输入（CI0FE0） 00111：滤波后的通道 1 输入（CI1FE1） 01000：保留 01001：内部触发输入 4（ITI4） 01010：内部触发输入 5（ITI5） 01011：内部触发输入 6（ITI6） 01100：内部触发输入 7（ITI7） 01101：内部触发输入 8（ITI8） 01110：内部触发输入 9（ITI9） 01111：内部触发输入 10（ITI10） 10000：保留 10001：保留 10010：保留 10011：内部触发输入 14（ITI14） 其他值：保留
25:21	TSCFG4[4:0]	暂停模式配置 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，并且 TSCFG4[4:0]不为零，则计数器停止。 00000：暂停模式禁能 00001：内部触发输入 0（ITI0） 00010：内部触发输入 1（ITI1） 00011：内部触发输入 2（ITI2） 00100：内部触发输入 3（ITI3） 00101：保留 00110：滤波后的通道 0 输入（CI0FE0） 00111：滤波后的通道 1 输入（CI1FE1） 01000：保留 01001：内部触发输入 4（ITI4） 01010：内部触发输入 5（ITI5） 01011：内部触发输入 6（ITI6）

		01100：内部触发输入 7（ITI7） 01101：内部触发输入 8（ITI8） 01110：内部触发输入 9（ITI9） 01111：内部触发输入 9（ITI10） 10000：保留 10001：保留） 10010：保留 10011：内部触发输入 14（ITI14） 其他：保留
20:16	TSCFG3[4:0]	复位模式配置 选中的触发输入的上升沿重新初始化计数器，并且更新影子寄存器。 00000：复位+模式禁能 00001：内部触发输入 0（ITI0） 00010：内部触发输入 1（ITI1） 00011：内部触发输入 2（ITI2） 00100：内部触发输入 3（ITI3） 00101：CI0 的边沿标志位（CI0F_ED） 00110：滤波后的通道 0 输入（CI0FE0） 00111：滤波后的通道 1 输入（CI1FE1） 01000：保留 01001：内部触发输入 4（ITI4） 01010：内部触发输入 5（ITI5） 01011：内部触发输入 6（ITI6） 01100：内部触发输入 7（ITI7） 01101：内部触发输入 8（ITI8） 01110：内部触发输入 9（ITI9） 01111：内部触发输入 10（ITI10） 10000：保留 10001：保留） 10010：保留 10011：内部触发输入 14（ITI14） 其他：保留
15:0	保留	必须保持复位值。

1.7.25. TIMERx 配置寄存器 1（SYSCFG_TIMERxCFG1, x=14）

地址偏移：TIMER14为0x14C
复位值：0x0000 0000

TSCFG3[4:0]，TSCFG4[4:0]..TSCFG7[4:0]之间相互互斥，不能同时配置。

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留											TSCFG15[4:0]				
											rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TSCFG14[4:0]					TSCFG13[4:0]					TSCFG12[4:0]				

位/位域	名称	描述
31:10	保留	必须保持复位值。
9:5	TSCFG7[4:0]	复位+事件模式配置 当 TSCFG7[4:0]不为零时，选中的触发输入的上升沿重新初始化计数器，并且更新影子寄存器。 00000：复位+事件模式禁能 00001：内部触发输入 0（ITI0） 00010：内部触发输入 1（ITI1） 00011：内部触发输入 2（ITI2） 00100：内部触发输入 3（ITI3） 00101：CI0 的边沿标志位（CI0F_ED） 00110：滤波后的通道 0 输入（CI0FE0） 00111：滤波后的通道 1 输入（CI1FE1） 01000：保留 01001：内部触发输入 4（ITI4） 01010：内部触发输入 5（ITI5） 01011：内部触发输入 6（ITI6） 01100：内部触发输入 7（ITI7） 01101：内部触发输入 8（ITI8） 01110：内部触发输入 9（ITI9） 01111：内部触发输入 10（ITI10） 10000：保留 10001：保留） 10010：保留 10011：内部触发输入 14（ITI14） 其他：保留
4:0	TSCFG6[4:0]	外部时钟模式 0 配置 在 TSCFG6[4:0]非零时，选中的触发输入的上升沿驱动计数器。 00000：外部时钟模式 0 禁能 00001：内部触发输入 0（ITI0） 00010：内部触发输入 1（ITI1）

00011：内部触发输入 2（ITI2）
00100：内部触发输入 3（ITI3）
00101：CI0 的边沿标志位（CI0F_ED）
00110：滤波后的通道 0 输入（CI0FE0）
00111：滤波后的通道 1 输入（CI1FE1）
01000：保留
01001：内部触发输入 4（ITI4）
01010：内部触发输入 5（ITI5）
01011：内部触发输入 5（ITI6）
01100：内部触发输入 7（ITI7）
01101：内部触发输入 8（ITI8）
01110：内部触发输入 9（ITI9）
01111：内部触发输入 10（ITI10）
10000：保留
10001：保留）
10010：保留
10011：内部触发输入 14（ITI14）
其他：保留

1.7.26. TIMERx 配置寄存器 2（SYSCFG_TIMERxCFG2, x=14）

地址偏移：TIMER14为0x150
复位值：0x0000 0000

TSCFG3[4:0]，TSCFG4[4:0]..TSCFG7[4:0]之间相互互斥，不能同时配置。

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留											TSCFG15[4:0]				
											rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TSCFG14[4:0]					TSCFG13[4:0]					TSCFG12[4:0]				

位/位域	名称	描述
31:21	保留	必须保持复位值。
20:16	TSCFG15[4:0]	内部触发输入源配置 00000：保留 00001：内部触发输入 0（ITI0） 00010：内部触发输入 1（ITI1）

00011：内部触发输入 2（ITI2）
00100：内部触发输入 3（ITI3）
00101：CI0 的边沿标志位（CI0F_ED）
00110：保留
00111：保留
01000：保留
01001：内部触发输入 4（ITI4）
01010：内部触发输入 5（ITI5）
01011：内部触发输入 6（ITI6）
01100：内部触发输入 7（ITI7）
01101：内部触发输入 8（ITI8）
01110：内部触发输入 9（ITI9）
01111：内部触发输入 10（ITI10）
10000：保留
10001：保留
10010：保留
10011：内部触发输入 14（ITI14）
其他：保留

15:0	保留	必须保持复位值。

1.8. 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。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。

1.8.1. 存储容量信息

基地址：0x1FFF B3E0

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：0x0008表示8Kbytes。
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：0x0020表示32Kbytes。

1.8.2. 设备唯一 ID(96 位/位域)

基地址：0x1FFF B3E8

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

基地址：0x1FFF B3EC

该值是原厂设定的，不能由用户修改。

该寄存器只能按字（32 位）访问

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

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

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

基地址：0x1FFF B3F0

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]	设备唯一设备 ID。

2. 闪存控制器（FMC）

2.1. 简介

闪存控制器（FMC），提供了片上闪存需要的所有功能。在闪存的前512K字节空间内，CPU执行指令需要少量等待时间。FMC也提供了页擦除，整片擦除，以及编程操作。

2.2. 主要特性

- 高达512KB字节的片上闪存可用于存储指令或数据，高达2KB一次性编程块。
  - bank0：256KB。
  - bank1：256KB。
  - OTP：2KB。
  - Bootloader：2x13KB。
  - 单bank模式（DBS=0）：128位读取位宽，闪存页大小2KB。
  - 双bank模式（DBS=1）：64位读取位宽，闪存页大小1KB。
- ECC支持单个位错误纠正和双位错误检测。
- 在闪存空间内，CPU执行指令和读取数据需要0~7个等待时间。
- 预取缓冲区以加速读操作。
- CBUS指令缓存区为2KB，由64条缓存线组成，每条缓存线为4x64位或2x128位。
- CBUS数据缓存区为512B，由16条缓存线组成，每条缓存线为4x64位或2x128位。
- 双bank架构支持边读边写(RWW)。
- 支持64位双字编程，页擦除和整片擦除操作。
- 2K字节OTP块（一次性编程），用于存储用户数据。
- 大小为2x48字节的选项字节可根据用户需求配置。
- 当系统电源上下电复位时，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4块擦写保护区域（当DBS=1时每个bank有2块擦写保护区域，当DBS=0时整个片上闪存有4块擦写保护区域），可阻止意外擦写操作。
- 具有 2 块仅执行的专用代码读保护区域（当 DBS=1 时每个 bank 有 1 块擦写保护区域，当 DBS=0 时整个片上闪存有 2 块仅执行区域）。
- 具有仅执行一次的安全用户区域（当DBS=1时每个bank有1块安全用户区域，当DBS=0时整个片上闪存只有1块安全用户区域）.
- 支持低功耗模式。

2.3. 功能说明

2.3.1. 闪存结构

主存储闪存高达 512KB，双 bank 结构下由 2x256 页组成，每页 1KB，还包含 2x13KB 的用于引导装载程序的信息块。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. 512KB 双 bank 闪存基地址和构成* 所示。

表 2-1. 512KB 双 bank 闪存基地址和构成

闪存块		名称	地址范围	大小（字节）
主存储闪存块	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
	Bank1	第 0 页	0x0804 0000 - 0x0804 03FF	1KB
		第 1 页	0x0804 0400 - 0x0804 07FF	1KB
		第 2 页	0x0804 0800 - 0x0804 0BFF	1KB
		. . .	. . .	. . .
		第 255 页	0x0807 FC00 - 0x0807 FFFF	1KB
信息块		Bank0 引导装载程序	0x1FFF 0000 - 0x1FFF 33FF	13KB
		Bank1 引导装载程序	0x1FFF 8000 - 0x1FFF B3FF	13KB
选项字节块		选项字节 0	0x1FFF 7800~0x1FFF 782F	48B
		选项字节 1	0x1FFF F800~0x1FFF F82F	48B
一次性编程块		Bank0 一次性编程块	0x1FFF 7000~0x1FFF 77FF	2KB
JTAG 一次编程块		Bank1 一次性编程块	0x1FFF F000~0x1FFF F00F	16B

注意：对于 256KB 系列产品，闪存页只含第 0 页到第 127 页。对于 128KB 系列产品，闪存页只含第 0 页到第 63 页。BANK0 和 BANK1 地址是连续的。

主存储闪存高达 512KB，单 bank 结构下由 256 页组成，每页 2KB。基地址和大小如*表 2-2. 512KB 单 bank 闪存基地址和构成* 所示。

表 2-2. 512KB 单 bank 闪存基地址和构成

闪存块	名称	地址范围	大小（字节）
主存储闪存块	第 0 页	0x0800 0000 - 0x0800 07FF	2KB
	第 1 页	0x0800 0800 - 0x0800 0FFF	2KB
	第 2 页	0x0800 1000 - 0x0800 17FF	2KB
	. . .	. . .	. . .
	第 254 页	0x0807F000 - 0x0807 F7FF	2KB
	第 255 页	0x0807 F800 - 0x0807 FFFF	2KB
信息块	Bank0 引导装载程序	0x1FFF 0000 - 0x1FFF 33FF	13KB
	Bank1 引导装载程序	0x1FFF 8000 - 0x1FFF B3FF	13KB
选项字节块	选项字节 0	0x1FFF 7800~0x1FFF 782F	48B
	选项字节 1	0x1FFF F800~0x1FFF F82F	48B
一次性编程块	Bank0 一次性编程块	0x1FFF 7000~0x1FFF 77FF	2KB
JTAG 一次编程块	Bank1 一次性编程块	0x1FFF F000~0x1FFF F00F	16B

注意：1. 对于 256KB 系列产品，每个 bank 闪存页只含第 0 页到第 127 页。对于 128KB 系列产品，每个 bank 闪存页只含第 0 页到第 63 页。

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. 2KB（256 双字）OTP（一次性编程）数据区域供用户使用，OTP 区域仅由 bank0 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。

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。

双 bank 模式（DBS=1）

当检测到一个错误并纠正时，FMC_ECCCS 寄存器中的 ECCCOR0 位将被置位。如果 FMC_ECCCS 寄存器中的 ECCCORIE 位被置位，则会产生中断。ECC 错误位所处的 bank 号保存在 FMC_ECCCS 寄存器的 BK_ECC 中，错误位所在的地址保存在 ECCADDR[18:0]中。

当检测到两个错误时，FMC_ECCCS 寄存器中的 ECCDET0 位会被置位，并产生一个 NMI。错误

位所在的地址保存在 FMC_ECCCS 寄存器的 ECCADDR[18:0]中。

注意：当 bank 交换后，ECC 地址按照交换后的地址报错，例如访问地址 0x08000000，即实际访问 bank1，如果 ECC 报错，ECCADDR 存的地址是 0x08040000。

单 bank 模式（DBS=0）

如果 DBS 复位，必须首先检查 SYS_ECC 标志位。SYS_ECC 为 1，BK_ECC 表示引导加载程序的 ECC 错误发生在哪个 bank。

当在 LSB（bits63：0）检测到一个错误并纠正时，FMC_ECCCS 寄存器中的 ECCCOR0 位将被置位。当在 MSB（bits127：64）检测到一个错误并纠正时，FMC_ECCCS 寄存器中的 ECCCOR1 位将被置位。如果 FMC_ECCCS 寄存器中的 ECCCORIE 位被置位，则会产生中断。错误位所在的地址保存在 ECCADDR[18:0]中。

当在 LSB（bits63：0）检测到两个错误时，FMC_ECCCS 寄存器中的 ECCDET0 位将被置位并产生一个 NMI。当在 MSB（bits127：64）检测到两个错误时，FMC_ECCCS 寄存器中的 ECCDET1 位将被置位并产生一个 NMI。错误位所在的地址保存在 FMC_ECCCS 寄存器的 ECCADDR[18:0]中。

注意：1. Flash 内存中的数据保存为 72 位字，每双字（64 位）增加了 8 位纠错码，但增加的 8 位是由硬件自动计算的，用户不能访问。

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

3. 当出现新的 ECC 错误时，仅当清除 ECCCOR0/ECCCOR1 和 ECCDET0/ECCDET1 后，才会更新 ECCADDR 和 BK_ECC。当 bank 切换后，ECCADDR 表示的映射地址也随之变化。

4. 当清除 ECCDET0/ECCDET1 时，SYSCFG_STAT 寄存器中的 FLASHECCIF 位会被同步清除，清除 FLASHECCIF 位会同步清除 ECCDET0/ECCDET1。

5. 预取 buffer 或 cache 读取数据或指令也可能生成 ECC 错误，即使数据或指令未被 CPU 使用。

6. 当编程原始数据为全 F 时，ECC 计算会 bypass 本次操作。不需要擦除，再次编程其他值可正常写入。

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 CBUS 总线。

增加等待状态

读取闪存时，需要根据 AHB 时钟频率正确配置 FMC_WS 寄存器中的 WSCNT 位。WSCNT 与 AHB 时钟频率的关系如 *表 2-3. WSCNT 与 AHB 时钟频率对应关系（LDO= 1.1V）*所示。

表 2-3. WSCNT 与 AHB 时钟频率对应关系（LDO= 1.1V）

AHB时钟频率	WSCNT配置
<= 10MHz	0（添加0个等待状态）
<= 20MHz	1（添加1个等待状态）
<= 50MHz	2（添加2个等待状态）
<= 70MHz	3（添加3个等待状态）
<= 90MHz	4（添加4个等待状态）
<= 120MHz	5（添加5个等待状态）
<= 150MHz	6（添加6个等待状态）
<= 170MHz	7（添加7个等待状态）
<= 216MHz	7（添加7个等待状态）

如果发生系统复位，AHB 时钟频率为 8MHz，此时 WSCNT 置为 0。当 AHB 时钟频率大于等于 170MHz 时，WSCNT 均为 7（添加 7 个等待状态）。

注意：

1. 如果希望增加 AHB 时钟频率。首先，参考*表 2-3. 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 2-3. 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：170MHz 时需添加 7 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 位或 128 位数据。因为 CPU 每次读操作只需要 32 位或 16 位数据。因此在顺序代码下，CPU 所需数据可以从当前缓存区获取而不必重复从闪存中获取。

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。

指令缓存区

置位 FMC_WS 寄存器中 ICEN 位来使能指令缓存区。指令缓存区仅在 CBUS 取数据时使用。指

令缓存区为 2KB，由 64 条缓存线组成，每条缓存线为 2 x 128 或 4 x 64 位。

当指令存在于 CBUS 指令缓存区时，CPU 从 CBUS 指令缓存区读取指令无延迟。当指令不存在于 CBUS 指令缓存区并且也不存在于当前缓存区/预取缓存区时，CBUS 指令缓存区从闪存中读取指令并复制到 CBUS 指令缓存区。当所有指令缓存线被填充，LRU（最近最少使用）策略被用于转移指令缓存线中的代码。

数据缓存区

置位 FMC_WS 寄存器中 DCEN 位来使能 CBUS 数据缓存区。CBUS 数据缓存区仅在 CPU 通过 CBUS 取数据（不是通过 DMA）时使用，此时选项字节不可缓存。CBUS 数据缓存区为 512B，由 16 条缓存线组成，每条缓存线为 2 x 128 或 4 x 64 位。

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

注意：如果在开启数据缓存的情况下修改 SYSCFG_CFG0 寄存器中的 FMC_SWP 位，可能导致 CPU 在第一个周期读取 flash 数据错误，这与访问 flash 序列有关。建议延迟读取 flash 两个 AHB 周期，以避免这种情况。关闭数据缓存，修改 FMC_SWP 位读取 flash 数据正常。

2.3.4. 双 bank 边读边写特性

该闪存具有基于 bank0（256KB）和 bank1（最大 256KB）的双 bank 架构。该结构支持 RWW（边读边写）特性，即当一个 bank 上有读操作操作时，另一个 bank 无需等待其操作完成，就可以进行编程操作。

2.3.5. FMC_CTL/FMC_OBCTL 寄存器解锁

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

FMC_OBCTL 寄存器、FMC_CTL 中的 OBRLD 位和 OBSTART 位在 FMC_CTL 被解锁后仍然处于被保护状态。解锁过程为两次写操作，向 FMC_OBKEY 寄存器先后写入 0x08192A3B 和 0x4C5D6E7F，然后硬件将 FMC_OBCTL 寄存器中的 OBLK 位清零。软件可以将 FMC_OBCTL 的 OBLK 位置 1 来锁定 FMC_OBCTL 寄存器、FMC_CTL 中的 OBRLD 位和 OBSTART 位。

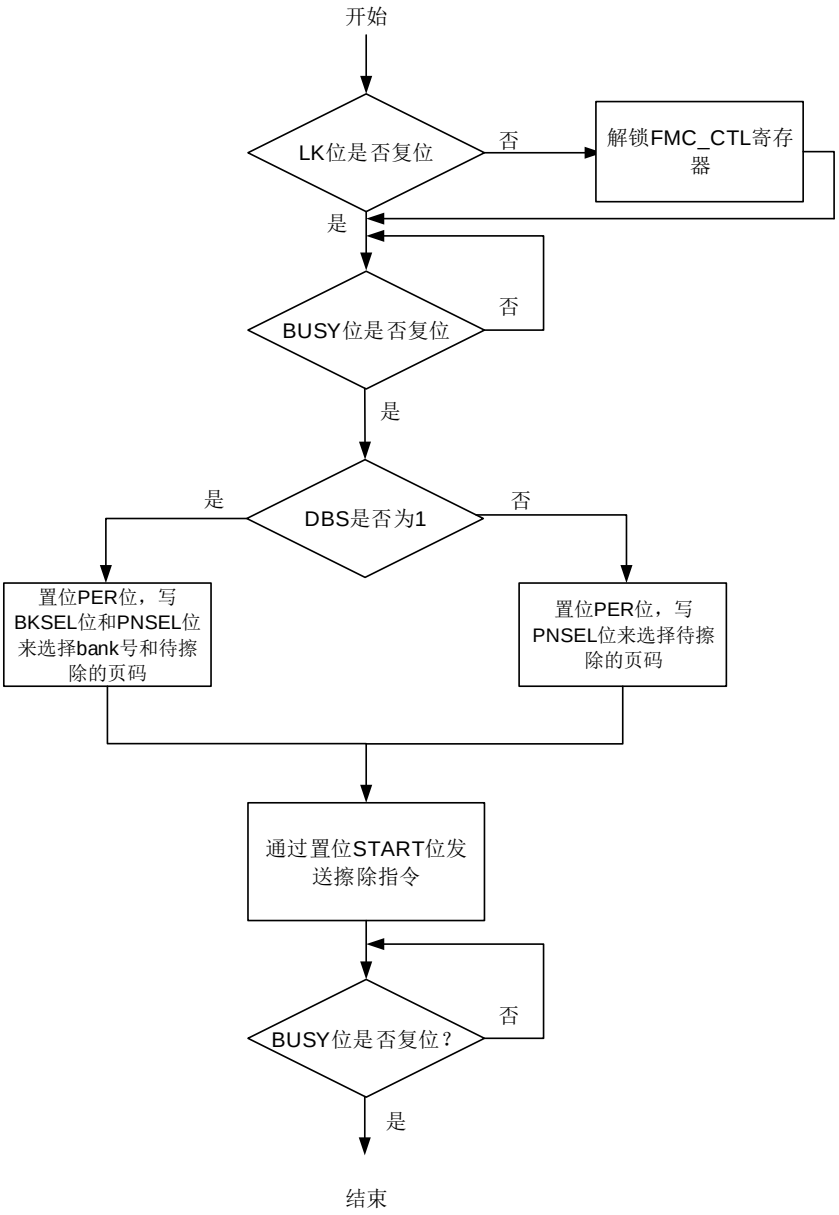
2.3.6. 页擦除

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_CTL 寄存器不处于锁定状态。
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0。
- 在双bank情况下（DBS位置位），应先置PER位。然后在控制寄存器（FMC_CTL）中选择要擦除的页面（PNSEL）和页面所在的bank（BKSEL）。在单bank情况下（DBS位复位），设置PER位并选择要擦除的页面（PNSEL），控制寄存器（FMC_CTL）中的BKSEL位必须保持清除状态。
- 通过将FMC_CTL寄存器的START位置1来发送页擦除命令到FMC。
- 等待擦除指令执行完毕，FMC_STAT寄存器的BUSY位清0。
- 如果需要，使用CBUS读并验证该页是否擦除成功。

当页擦除操作成功执行且使能操作结束中断 (ENDIE = 1)时，FMC_STAT 寄存器中的 ENDF 将被置位。如果 FMC_CTL 寄存器中的 ENDIE 位被置位，FMC 将触发中断。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。此时如果 FMC_CTL 寄存器中的 ERRIE 位置位，则 FMC 将触发闪存操作错误中断。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

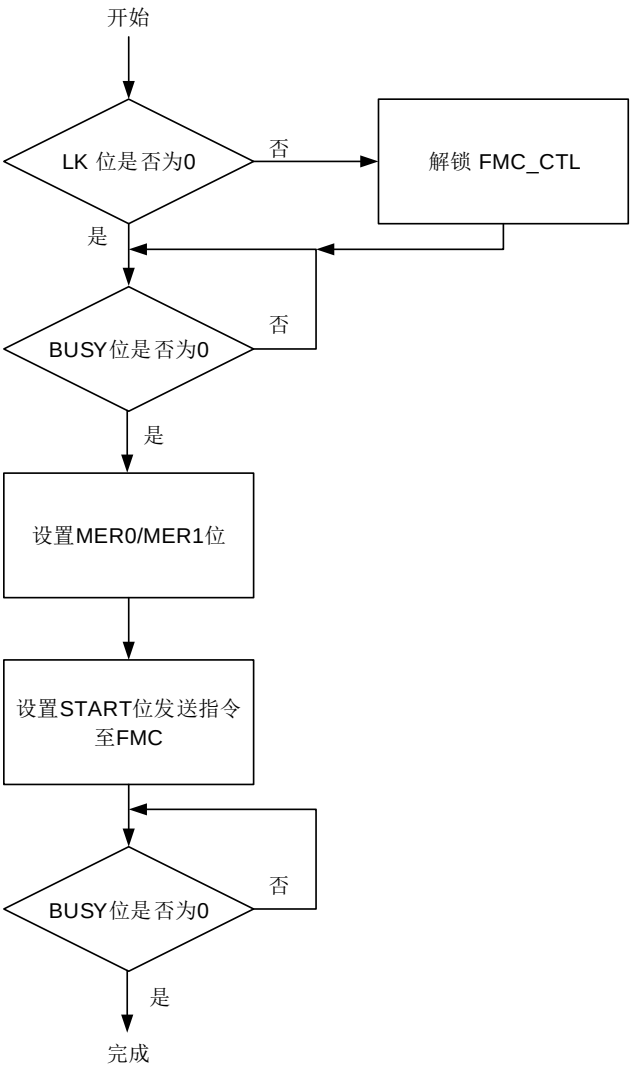
FMC 提供了整片擦除功能可以初始化主存储闪存块的内容。这种擦除可以通过将 MER0 位设置为 1 来仅擦除 bank0，或者通过将 MER1 位设置为 1 来仅擦除 Bank1，或者通过将 MER0 和 MER1 位都设置为 1 来擦除整个闪存。整片擦除操作，寄存器设置具体步骤如下：

- 确保FMC_CTL寄存器不处于锁定状态。
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0。

- 如果只擦除bank0，则在FMC_CTL寄存器中置位MER0位。如果只擦除bank1，则在FMC_CTL寄存器中置位MER1位。如果要擦除整个闪存，在FMC_CTL寄存器中置位MER0和MER1位。
- 通过将FMC_CTL寄存器的START位置1来发送整片擦除命令到FMC。
- 等待擦除指令执行完毕，FMC_STAT寄存器的BUSY位清0。
- 如果需要，使用CBUS读并验证是否擦除成功。

当整片擦除成功执行且使能操作结束中断 (ENDIE = 1)，FMC_STAT 寄存器的 ENDF 位置位。若 FMC_CTL 寄存器的 ENDIE 位被置 1，FMC 将触发一个中断。由于所有的闪存数据都将被复位为 0xFFFF FFFF，可以通过运行在 SRAM 中的程序或使用调试工具直接访问 FMC 寄存器来实现整片擦除操作。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



2.3.8. 主存储闪存块编程

FMC 提供了一个通过 CBUS 修改主存储闪存内容的 64 位双字编程功能（2 x 32 位+ 8 位 ECC）。下面的步骤显示了编程操作的寄存器设置顺序。

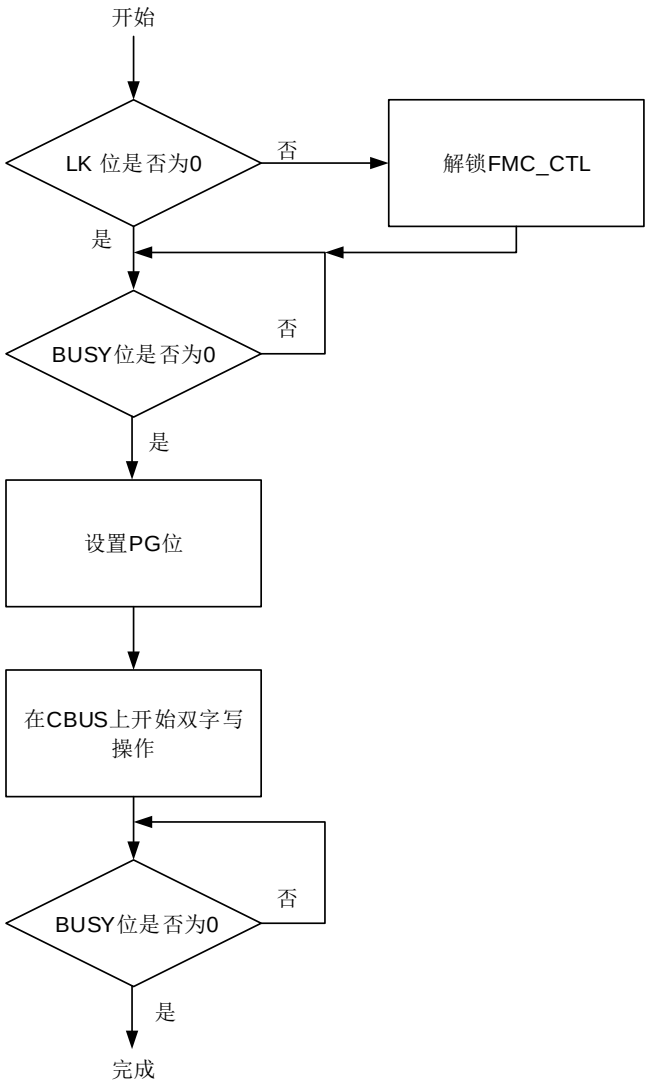
- 确保FMC_CTL寄存器不处于锁定状态。
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0。
- 置位FMC_CTL寄存器的PG位。
- CBUS写数据到目的绝对地址（0x08XX XXXX）。

CBUS 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_STAT寄存器的BUSY位清0。
- 如果需要，使用CBUS读并验证是否编程成功。

当主存储块编程执行成功且使能操作结束中断 (ENDIE = 1)时，FMC_STAT 寄存器中的 ENDF 置位，如果 FMC_CTL 寄存器中的 ENDIE 位置位，FMC 将触发中断。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_STAT 寄存器的 PGERR 位也将被置 1。另外，在擦写保护页上的编程操作会被忽略，同时 FMC_STAT 中的 WPERR 位将置位。在这些情况下，如果 FMC_CTL 寄存器中的 ERRIE 位被置位，则 FMC 将产生一个闪存操作错误中断。软件可以检查 FMC_STAT 寄存器中的 PGERR、PGSERR、PGMERR、PGAERR 和 RPERR 位，以检测出现哪种错误。主存储块双字编程操作流程如*图 2-3. 双字编程操作流程*所示。

图 2-3. 双字编程操作流程



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. OTP 编程

OTP 编程方法与主储存闪存编程相同。OTP 块只能被编程一次并且不能被擦除。

注意：必须确保在 OTP 编程操作时不会发生任何意外中断，例如系统复位或掉电。如果发生意外

中断，闪存中的数据有很小可能性会出错。

2.3.10. 选项字节

选项字节说明

每次系统上电/下电复位或在 FMC_CTL 寄存器中 OBRLD 位置 1 后，闪存的选项字节寄存器重新加载到相应选项字节块中，选项字节生效。选项补码字节与选项字节相反。当选项字节重新加载时，如果选项字节补码与选项字节不匹配，则 FMC_STAT 寄存器中的 OBERR 位将被置 1。选项字节详情见下表*表 2-4. 选项字节*。

表 2-4. 选项字节

地址	名称	说明
0x1fff 7800	SPC	选项字节安全保护值 0xAA：未保护状态 除 0xAA 和 0xCC 外的任何值：保护级别低 0xCC：保护级别高
0x1fff 7801	OB_USER[7:0]	[7]：FMC_SWP FMC 存储器映射切换。 0：主 FLASH 存储器的 Bank0 被映射到地址 0x0804 0000，主 FLASH 存储器的 Bank1 被映射到地址 0x0800 0000 1：主 FLASH 存储器的 Bank0 映射到地址 0x0800 0000，主 FLASH 存储器的 Bank1 映射到地址 0x0804 0000 [6]：保留 [5]：nRST_STDBY 0：进入待机模式时产生复位 1：进入待机模式时不产生复位 [4]：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 [3:2]：保留 [1:0]：BOR_TH（BOR 复位阈值） 00：BOR 功能关闭。 01：BOR 阈值 1。阈值约 2.2 V。 10：BOR 阈值 2。阈值约 2.5 V。 11：BOR 阈值 3。阈值约 2.8 V。
0x1fff 7802	OB_USER[15:8]	[7]：nBOOT1 0：BOOT1 为 1 1：BOOT1 为 0 它与 BOOT0 引脚共同决定 boot 模式

地址	名称	说明
		[6]：DBS 0：单 bank 模式，128 位读取位宽 1：双 bank 模式，64 位读取位宽 仅当 DCRP0/1 失能时该位可写。 [5]：保留 [4]：BB 0：当配置从主存储块启动时，从 bank0 启动（出厂值） 1：从 bank1 启动，若 bank1 无启动程序，则从 bank0 启动 [3]：保留 [2]：FWDGSPD_STDBY 0：在系统待机模式下独立看门狗暂停 1：在系统待机模式下独立看门狗运行 [1]：FWDGSPD_DPSLP 0：在系统深度睡眠模式下独立看门狗暂停 1：在系统深度睡眠模式下独立看门狗运行 [0]：nFWDG_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 7803	OB_USER[23:16]	[7:6]：保留 [5:4]：NRST_MDSEL 00：NRST 引脚配置为复位输入/输出模式 01：NRST 引脚上的低电平可以复位系统，内部复位不能驱动 NRST 引脚 10：NRST 引脚功能与普通 GPIO 相同，只有内部复位 11：NRST 引脚配置为复位输入/输出模式 [3]：nBOOT0 0：BOOT0 为 1 1：BOOT0 为 0 [2]：nSWBT0 0：BOOT0 取决于选项位 nBOOT0 1：BOOT0 取决于 PB8/BOOT0 引脚 [1]：TCMSRAM_ERS 0：系统复位后，TCM SRAM 被擦除 1：当系统复位时，TCM SRAM 不会被擦除 [0]：SRAM_ECCEN 0：SRAM 与 TCM SRAM ECC 使能 1：SRAM 与 TCM SRAM ECC 失能
0x1fff 7804	SPC_N	SPC 补码字节
0x1fff 7805	OB_USER_N[7:0]	OB_USER 补码字节 7 到 0 位

地址	名称	说明
0x1fff 7806	OB_USER_N[15:8]	OB_USER 补码字节 15 到 8 位
0x1fff 7807	OB_USER_N[23:16]	OB_USER 补码字节 23 到 16 位
0x1fff 7808	DCRP_SADDR0[7:0]	Bank0 的 DCRP 区域起始地址
0x1fff 7809	DCRP_SADDR0[14:8]	[7]：保留 [6:0]：Bank0 的 DCRP 区域起始地址
0x1fff 780c	DCRP_SADDR0_N[7:0]	DCRP_SADDR0 补码字节 7 到 0 位
0x1fff 780d	DCRP_SADDR0_N[14:8]	[7]：保留 [6:0]：DCRP_SADDR0 补码字节 14 到 8 位
0x1fff 7810	DCRP_EADDR0[7:0]	Bank0 的 DCRP 区域结束地址
0x1fff 7811	DCRP_EADDR0[14:8]	[7]：保留 [6:0]：Bank0 的 DCRP 区域结束地址
0x1fff 7813	DCRP0_EREN	[7]：DCRP0_EREN 0：当SPC等级从1降低到0时，DCRP不会被擦除 1：当 SPC 等级从 1 降低到 0 时，DCRP 会被擦除 [6:0]：保留
0x1fff 7814	DCRP_EADDR0_N[7:0]	DCRP_EADDR0 补码字节 7 到 0 位
0x1fff 7815	DCRP_EADDR0_N[14:8]	[7]：保留 [6:0]：DCRP_EADDR0 补码字节 14 到 8 位
0x1fff 7817	DCRP0_EREN_N	DCRP0_EREN 补码字节
0x1fff 7818	BK0WP0_SADDR[7:0]	DBS=1 BK0WP0_SADDR[7:0]包含在 bank0 的第 1 个 WP 区域的首页 DBS=0 BK0WP0_SADDR[7:0]包含在主存储闪存的第 1 个 WP 区域的首页
0x1fff 781a	BK0WP0_EADDR[7:0]	DBS=1 BK0WP0_EADDR[7:0]包含在 bank0 的第 1 个 WP 区域的尾页 DBS=0 BK0WP0_EADDR[7:0]包含在主存储闪存的第 1 个 WP 区域的尾页
0x1fff 781c	BK0WP0_SADDR_N[7:0]	BK0WP0_SADDR 补码字节 7 到 0 位
0x1fff 781e	BK0WP0_EADDR_N[7:0]	BK0WP0_EADDR 补码字节 7 到 0 位
0x1fff 7820	BK0WP1_SADDR[7:0]	DBS=1 BK0WP1_SADDR[7:0]包含在 bank0 的第 2 个 WP 区域的首页 DBS=0 BK0WP1_SADDR[7:0]包含在主存储闪存的第 2 个 WP 区域的首页
0x1fff 7822	BK0WP1_EADDR[7:0]	DBS=1

地址	名称	说明
		BK0WP1_EADDR[7:0]包含在 bank0 的第 2 个 WP 区域的尾页 DBS=0 BK0WP1_EADDR[7:0]包含在主存储闪存的第 2 个 WP 区域的尾页
0x1fff 7824	BK0WP1_SADDR_N [7:0]	BK0WP1_SADDR 补码字节 7 到 0 位
0x1fff 7826	BK0WP1_EADDR_N [7:0]	BK0WP1_EADDR 补码字节 7 到 0 位
0x1fff 7828	SCR_PAGE_CNT0[7:0]	[7:0]：配置 bank0 安全用户区域的页数
0x1fff 7829	SCR_PAGE_CNT0[8]	[0]：SCR_PAGE_CNT0[8]
0x1fff 782a	BOOTLK	[0]：该位置1后强制从用户闪存区启动 0：支持闪存，ram和系统启动 1：只能从主闪存启动
0x1fff 782c	SCR_PAGE_CNT0_N [7:0]	SCR_PAGE_CNT0 补码字节 7 到 0 位
0x1fff 782d	SCR_PAGE_CNT0_N [8]	SCR_PAGE_CNT0 补码第 8 位
0x1fff 782e	BOOTLK_N	BOOTLK 补码字节比特 0 位
0x1fff f808	DCRP_SADDR1[7:0]	bank1 的 DCRP 区域起始地址
0x1fff f809	DCRP_SADDR1[14:8]	[7]：保留 [6:0]：bank1 的 DCRP 区域起始地址
0x1fff f80c	DCRP_SADDR1_N[7:0]	DCRP_SADDR1 补码字节 7 到 0 位
0x1fff f80d	DCRP_SADDR1_N [14:8]	[7]：保留 [6:0]：DCRP_SADDR1 补码字节 14 到 8 位
0x1fff f810	DCRP_EADDR1[7:0]	bank1 的 DCRP 区域结束地址
0x1fff f811	DCRP_EADDR1[14:8]	[7]：保留 [6:0]：bank1 的 DCRP 区域结束地址
0x1fff f814	DCRP_EADDR1_N[7:0]	DCRP_EADDR1 补码字节 7 到 0 位
0x1fff f815	DCRP_EADDR1_N [14:8]	[7]：保留 [6:0]：DCRP_EADDR1 补码字节 14 到 8 位
0x1fff f818	BK1WP0_SADDR[7:0]	DBS=1 BK1WP0_SADDR[7:0]包含在 bank1 的第 1 个 WP 区域的首页 DBS=0 BK1WP0_SADDR[7:0]包含在主存储闪存的第 1 个 WP 区域的首页
0x1fff f81a	BK1WP0_EADDR[7:0]	DBS=1 BK1WP0_EADDR[7:0]包含在 bank1 的第 1 个 WP 区域的尾页 DBS=0 BK1WP0_EADDR[7:0]包含在主存储闪存的第 3 个 WP 区域的

地址	名称	说明
		尾页
0x1fff f81c	BK1WP0_SADDR_N [7:0]	BK1WP0_SADDR 补码字节 7 到 0 位
0x1fff f81e	BK1WP0_EADDR_N [7:0]	BK1WP0_EADDR 补码字节 7 到 0 位
0x1fff f820	BK1WP1_SADDR[7:0]	DBS=1 BK1WP1_SADDR[7:0]包含在 bank1 的第 2 个 WP 区域的首页 DBS=0 BK1WP1_SADDR[7:0]包含在主存储闪存的第 4 个 WP 区域的首页
0x1fff f822	BK1WP1_EADDR[7:0]	DBS=1 BK1WP1_EADDR[7:0]]包含在 bank1 的第 2 个 WP 区域的尾页 DBS=0 BK1WP1_EADDR[7:0]包含在主存储闪存的第 4 个 WP 区域的尾页
0x1fff f824	BK1WP1_SADDR_N [7:0]	BK1WP1_SADDR 补码字节 7 到 0 位
0x1fff f826	BK1WP1_EADDR_N [7:0]	BK1WP1_EADDR 补码字节 7 到 0 位
0x1fff f828	SCR_PAGE_CNT1[7:0]	[7:0]：配置在 bank1 的安全用户区域的页数
0x1fff f829	SCR_PAGE_CNT1[8]	[0]：SCR_PAGE_CNT1[8]
0x1fff f82c	SCR_PAGE_CNT1_N [7:0]	SCR_PAGE_CNT1 补码字节 7 到 0 位
0x1fff f82d	SCR_PAGE_CNT1_N [8]	SCR_PAGE_CNT1 补码第 8 位

表 2-5. 出厂值

地址	出厂值
0x1fff 7800	0x0010 0355 FFEF FCAA
0x1fff 7808	0x0000 0000 FFFF FFFF
0x1fff 7810	0xFF00 FFFF 00FF 0000
0x1fff 7818	0x01FF 0000 FE00 FFFF
0x1fff 7820	0x01FF 0000 FE00 FFFF
0x1fff 7828	0x00FF 03FF FF00 FC00
…	…
0x1FFFF800	0x0000 0000 FFFF FFFF
0x1FFFF808	0x0000 0000 FFFF FFFF
0x1FFFF810	0xFF00 FFFF 00FF 0000

地址	出厂值
0x1FFFF818	0x01FF 0000 FE00 FFFF
0x1FFFF820	0x01FF 0000 FE00 FFFF
0x1FFFF828	0x00FF 01FF FF00 FE00

选项字节编程

FMC 提供了选项字节编程功能，可用来修改选项字节内容。编程操作过程如下：

- 若FMC_CTL寄存器处于锁定状态则解锁FCM_CTL寄存器。
- 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。
- 将正确的序列写入FMC_OBKEY寄存器来解锁FMC_CTL寄存器中的OBLK位。
- 等待FMC_CTL寄存器的OBLK位清0。
- 在选项字节寄存器中写选项字节值到对应的寄存器地址。
- 置位FMC_CTL寄存器中的OBSTART位来发送选项字节编程命令。
- 检查FMC_STAT寄存器中BUSY位的值，若BUSY位为1，则需等待该操作结束。
- 启动系统上电/下电复位(或从待机模式退出)或设置FMC_CTL寄存器中的OBRLD位来加载到选项字节。

注意：

1. 一旦执行了一个选项字节的修改，首先两个 bank 的用户选项字节将自动擦除。当操作执行成功时，FMC_STAT 寄存器中的 ENDF 被置位，如果 FMC_CTL 寄存器中的 ENDIE 位被置位，FMC 将触发中断。

2. Bits 63:32 是 Bits 31:0 的补码。每个选项字节位在相应的补码中都有它的补位。当选项字节加载时，ECC 先检查选项字节。只有当选项字节与其补码匹配时，才可以将选项字节写入相应的寄存器。

3. 如果选项字节与它的补码不匹配，则 FMC_STAT 寄存器中的 OBERR 位将置位。所有的 OB_USER 字节都强制为 0xFF，除了 BOR_TH 是 0b000，WP 页面的状态为“无保护”，DCRP 的状态是“所有区域受保护”。

切换单/双 bank 模式

强烈建议从 SRAM 中执行代码来执行从一种闪存模式切换到另一种闪存模式时。闪存中的数据也必须重新编程，因为在更改存储模式后闪存中的数据会被损坏。切换步骤如下所示。

- 如果指令缓存、数据缓存和预取被启用，清除 FMC_WS 寄存器中的 ICEN、DCEN 和 PFEN 位。
- 置位 FMC_WS 寄存器中的 DCRST 位和 ICRST 位来刷新指令和数据缓存。
- 清除所有 WP 页面，再 DBS 位置 1 或清 0。置位 FMC_CTL 寄存器中的 OBSTART 位来发送选项字节编程命令。然后设置 OBRLD 位来加载选项字节。

- 在重新加载选项字节后擦除整个闪存。重新编程代码并设置所需的 WP 页面。然后根据需要在 FMC_WS 寄存器中设置 ICEN、DCEN 和 PFEN 位。

2.3.11. 仅执行区域（DCRP）

在主闪存块中，FMC 可以定义仅可执行区域，只允许来自系统的指令事务，而不允许数据访问。它允许在 DBS 位置位时为每个 bank 指定一个 DCRP 区域，或者在 DBS 位清 0 时指定两个不同的 DCRP 区域。

注意：当使用仅可执行区域功能时，用户需要相应地使用仅执行选项去编译其原生代码。

DCRP 区域由起始地址偏移量和结束地址偏移量来定义。如果选择单 bank 模式，DCRP 区域最小为 32 字节。DCRP 区域 x（x= 0,1）如下定义：

- 从 bank 基地址 + [FMC_DCRP_SADDRx x 16]（包含）到 bank 基地址 + [(FMC_DCRP_EADDRx + 1) x 16]（不包含）。

DCRP 区域由起始地址偏移量和结束地址偏移量定义。如果选择双 bank 模式，最小 DCRP 区域为 16 字节。DCRP 区域 x（x= 0,1）如下定义：

- 从 bank 基地址 + [FMC_DCRP_SADDRx x 8]（包含）到 bank 基地址 + [(FMC_DCRP_EADDRx + 1) x 8]（不包含）。

例如，当选择双 bank 模式（DBS = 1）时，为了通过 DCRP 从地址 0x0802 6B40（包含）保护到地址 0x0803 2004（包含），选项字节应如下设置：

- FMC_DCRP_SADDR0 = 0x4D68。
- FMC_DCRP_EADDR0 = 0x6400。

如果将两个 bank 交换，则应对 bank1 应用保护，选项字节应如下设置：

- FMC_DCRP_SADDR1 = 0x4D68.
- FMC_DCRP_EADDR1 = 0x6400.

在此区域执行代码时，将忽略调试事件。只有 CPU 可以访问 DCRP 区域，只使用指令读取任务。在所有其他情况下，访问 DCRP 区域都是非法的。例如，读操作将返回 0 并置位 FMC_STAT 寄存器中的 RPERR 标志，而写操作将被忽略并置位 FMC_STAT 寄存器中的 WPERR 标志。

DCRP 区域受擦除保护，无法擦除该区域内的页。如果设置了有效的 DCRP 区域，则除非执行 SPC 保护等级低到无保护状态的降级擦除，否则无法执行整片擦除。

只有 CPU 可以修改 DCRP 区域定义位和 DCRP_EREN 位。如果 DCRP 区域有效，在 SPC 等级低到无保护降级期间，DCRP_EREN 置 0，则 DCRP 区域不被擦除，否则该区域将被擦除。

注意：失效 DCRP 区域的唯一方法：当 DCRP_EREN 置 1 时，将 SPC 等级从低降到无保护。直接修改 DCRP 选项字节来减小 DCRP 区域不起作用，但是增大 DCRP 区域起作用。

2.3.12. 页擦除 / 编程保护（WP）

FMC 的扇区擦除/编程保护功能可以阻止对闪存的意外操作。当 FMC 对被保护扇区进行扇区擦除或编程操作时，操作本身无效且 FMC_STAT 寄存器的 WPERR 位将被置 1。WP 区域由起始地址偏移量和结束地址偏移量定义。页面保护功能可以通过配置 WP 地址寄存器分别启用：FMC_BK0WPx（x = 0,1）和 FMC_BK1WPx （x = 0,1）。

如果选择单 bank 模式（DBS = 0），则可以在 bank 中定义 4 个 WP 区域，粒度为 2Kbytes。WP 区域定义如下：

- 从 bank 基地址 + [(BKxWPy_SADDR [7:0]) x 0x800]（包含）到 bank 基地址 + [(BKxWPy_EADDR [7:0] + 1) x 0x800]（不包含）。

如果选择双 bank 模式（DBS = 1），则可以在每个 bank 中定义 2 个 WP 区域，粒度为 1Kbytes。WP 区域定义如下：

- 从 bank 基地址 + [(BKxWPy_SADDR [7:0]) x 0x400]（包含）到 bank 基地址 + [(BKxWPy_EADDR [7:0] + 1) x 0x400]（不包含）。

例如，当选择双 bank 模式(DBS = 1)时，要保护从 0x0802 2800（包含）到 0x0803 07FF（包含）的地址，则选项字节应当如下设置：

- BK0WP0_SADDR [7:0] = 0x8A.
- BK0WP0_EADDR [7:0] = 0xC1.

如果将两个 bank 交换，则应当如下设置：

- BK1WP0_SADDR [7:0] = 0x8A.
- BK1WP0_SADDR [7:0] = 0xC1.

擦除/编程保护页既不能擦除也不能编程。因此，如果一个页面受到擦除/编程保护，则不能执行整片擦除操作。

如果设置的 SPC 级别为“高”，则不能修改 WP 区域，否则可以不受限制地修改 WP 区域。

注意：DCRP 或安全用户区域受到擦除/编程保护。

表 2-6. WP 保护

WP寄存器值（x, y = 0,1）	WP区域
BKxWPy_SADDR = BKxWPy_EADDR	页BKxWPy_SADDR被WP保护
BKxWPy_SADDR > BKxWPy_EADDR	无WP区域
BKxWPy_SADDR < BKxWPy_EADDR	从BKxWPy_SADDR到BKxWPy_EADDR的页被WP保护

2.3.13. 安全保护（SPC）

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 SPC 字节及其补字节被设置为 0xAA55，系统复位以后，闪存将处于非安全保护状态。主存储块和选项字节可以被所有操作模式访问。

低等级保护：当设置 SPC 字节值为任何除 0xAA 或 0xCC 外的值，系统复位以后，低安全保护状态生效。注意，如果在调试器保持连接到 JTAG/SWD 设备时执行了 SPC 修改，则应该进行电源重置，而不是系统重置。在低等级保护下，主 flash 只能通过用户代码访问。在调试模式下，从 SRAM 或引导加载程序模式启动，禁止对主 flash 进行所有操作。如果在调试模式、从 SRAM 启动或引导加载程序模式下对主 flash 进行读操作、擦写操作，则会产生总线错误。但这些模式下都可以对选项字节进行操作，从而可以通过该方式失能安全保护功能。通过将 SPC 字节及其补码值设置为 0xAA55，返回到无保护级别，然后自动触发一次整片擦除操作。如果 TCMSRAM 的某些页受保护，整个 TCMSRAM 将被擦除。注意如果配置了低级别保护且没有定义 DCRP 区域，则必须设置 DCRP_EREN 位。

高等级保护：将 SPC 字节及其补字节设置为 0xCC33 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，从 SRAM 中启动，或者从 boot loader 启动都被禁止。主存储闪存块可由用户代码的所有操作进行访问。SPC 字节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。

2.3.14. 安全用户区域（SCR）

在主闪存块中，FMC 可以定义安全用户区域，这些安全用户区域只能在引导时执行一次，除非发生复位，否则不会再次执行。

安全用户区域可以将安全代码与应用程序非安全代码隔离开来。安全用户区域可用于保护自定义安全引导库、固件更新代码或第三方安全库。当使能了安全区域（FMC_CTL 寄存器中的 SCR0（或 SCR1）置位）时，读安全用户区域操作将置 RDERR 位，擦写安全用户区域操作将置 WPERR 位。

安全用户区域的大小由 FMC_BK0SCR(或 FMC_BK1SCR)寄存器的 SCR_PAGE_CNT0 [8:0] (或 SCR_PAGE_CNT1 [8:0]位定义。只能在 SPC 级为无保护下修改。安全用户区域的内容在从 SPC 低保护等级更改为 SPC 无保护等级时被擦除，即使它与 DCRP 页面重叠。

安全用户区域定义如下：

- 如果选择单 bank 模式（DBS = 0），则从 bank 基地址到 bank 基地址+ [(SCR_PAGE_CNT0[8:0]) x 0x800]（不包含）。
- 如果选择双 bank 模式（DBS = 0），则从 bank0 基地址（包含）到 bank0 基地址（包含）+ [(SCR_PAGE_CNT0[8:0]) x 0x400]（不包含）和从 bank1 基地地址（包含）到 bank1 基地地

址+ [(SCR_PAGE_CNT1[8:0]) x 0x400]（不包含）。

2.3.15. 禁用核心调试访问

在安全用户区域执行敏感代码或访问敏感数据时，可以暂时禁用对核心的调试访问。当 SPC 级别没有保护或低级别保护时，调试器可以通过软件清除在 FMC_WS 寄存器中的 DBGEN 位来禁用核心调试访问。

2.3.16. 强制从闪存启动

FMC_BK0SCR 寄存器中的 BOOTLK 位可以配置强制系统从主闪存引导启动。该位只能在以下情况下复位：

1. SPC 为无保护等级。
2. SPC 为低保护级别，发出无保护等级修改请求后执行了全片擦除。

2.3.17. FMC 中断

FMC 中断事件和标志如*表 2-7. FMC 中断请求*所示。

表 2-7. FMC 中断请求

标志	描述	清除条件	中断使能位
ENDF	操作结束	写1到FMC_STAT寄存器中对应的位	ENDIE
OPRERR	操作失败错误		ERRIE
RPERR	读保护错误		RPERRIE
ECCCOR0/ ECCCOR1	ECC修正	写1到FMC_ECCCS寄存器对应的位	ECCCORIE

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：0x0004 0600

该寄存器只能按字(32 位)访问。

位/位域	名称	描述
31:19	保留	必须保持复位值
18	DBGEN	该位用于软件启用/禁用调试 0：禁用调试。 1：使能调试。
17:15	保留	必须保持复位值
14	SLP_MDSEL	在睡眠模式下闪存的运行模式 该位用于决定系统进入睡眠模式时，闪存进入掉电模式还是空闲模式 0:当系统进入睡眠模式时，闪存仍为空闲模式。 1:当系统进入睡眠模式时，闪存进入掉电模式。
13	RUN_MDSEL	在运行模式下闪存的运行模式 当系统进入运行模式时，用于确定闪存进入掉电模式还是空闲模式。只有当代码在RAM中运行时，闪存才能置于掉电模式。当置位RUN_MDSEL时，闪存无法访问。这个位被FMC_RUNKEY写保护。 0：当系统在运行模式下，闪存仍为空闲模式。 1：当系统在运行模式下，闪存进入掉电模式。
12	DCRST	复位数据缓存区。该位仅可在DCEN位置0时可写。 0：无效果。 1：复位数据缓存区。

11	ICRST	复位指令缓存区。该位仅可在ICEN位置0时可写。 0：无效果。 1：复位指令缓存区。
10	DCEN	数据缓存区使能位 0：失能数据缓存区。 1：使能数据缓存区。
9	ICEN	指令缓存区使能位 0：失能指令缓存区。 1：使能指令缓存区。
8	PFEN	预取功能使能位 0：失能预取功能。 1：使能预取功能。
7:4	保留	必须保持复位值
3:0	WSCNT[3:0]	等待状态计数寄存器 软件置 1 和清 0。 0000：不增加等待状态。 0001：增加 1 个等待状态。 0010：增加 2 个等待状态。 0011：增加 3 个等待状态。 0100：增加 4 个等待状态。 … 1111：增加 15 个等待状态。

2.4.2. 系统运行时闪存的运行模式解锁寄存器(FMC_RUNKEY)

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
RUN_KEY[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RUN_KEY[15:0]															
w															

位/位域	名称	描述
31:0	RUN_KEY[31:0]	RUN_MDSEL 解锁寄存器

这些位仅能被软件写。

写解锁值到 RUN_KEY[31:0]来解锁在 FMC_WS 寄存器中的 RUN_MDSEL 位。

RUN_KEY1：0x04152637

RUN_KEY2：0xFAFBFCFD

2.4.3. 解锁寄存器(FMC_KEY)

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_CTL 解锁寄存器 这些位仅能被软件写。 写解锁值到 KEY[31:0]可以解锁 FMC_CTL 寄存器。

2.4.4. 选项字节操作解锁寄存器(FMC_OBKEY)

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
OBKEY[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OBKEY[15:0]															
w															

位/位域	名称	描述
31:0	OBKEY[31:0]	FMC_CTL 选项字节操作解锁寄存器 这些位仅能被软件写 写解锁值到 OBKEY[31:0]解锁 FMC_CTL 寄存器的选项字节命令。

OBKEY1：0x0819 2A3B

OBKEY2：0x4C5D 6E7F

2.4.5. 状态寄存器(FMC_STAT)

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															BUSY
															r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OBERR	RPERR	保留						PGSERR	PGMERR	PGAERR	WPERR	PGERR	保留	OPRERR	ENDF
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
31:17	保留	必须保持复位值
16	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。
15	OBERR	选项字节读取错误标志位 当选项字节与其补充字节不匹配时，该位由硬件置位，并且该选项字节强制设置为 0xFF。
14	RPERR	读保护错误标志位 当通过 CBUS 访问的地址被 DCRP 或 SCR 保护时，该位由硬件置 1。这个位可以通过写入 1 来清除。 0：未发生读保护错误。 1：发生读保护错误。
13:8	保留	必须保持复位值
7	PGSERR	编程顺序错误标志位 当编程操作没有将PG位先置1就编程时，该位由硬件置位。由于先前的编程错误导致PGERR，PGMERR，PGAERR或WPERR置1时，该位也置1。 该位软件写 1 清 0。 0：未发生编程顺序错误。 1：发生编程顺序错误。
6	PGMERR	程序大小与错误标志位不匹配。当编程大小为半字/字访问时，该位由硬件置位。唯一正确的编程大小是双字。

		该位软件写1清0。
5	PGAERR	编程对齐错误标志位 当CBUS写入数据未64位对齐时，即要编程的数据不能包含在同一64位闪存行中，该位由硬件置位。 该位软件写1清0。
4	WPERR	擦除/编程保护错误标志位 该位由硬件置位，软件写 1 清 0。 0：未发生擦除/编程保护错误。 1：发生擦除/编程保护错误。
3	PGERR	编程错误标志位 当要编程的闪存双字地址的数据不是 0xFFFF FFFF FFFF FFFF 时，该位由硬件置 1，软件写 1 清 0。
2	保留	必须保持复位值
1	OPRERR	操作失败错误标志位 如果闪存编程或擦除操作完成不成功，并且使能了错误中断(ERRIE = 1)，则该位由硬件置1，软件写1清0。
0	ENDF	操作结束标志位 当操作执行成功且使能操作结束中断 (ENDIE = 1)，此位被硬件置 1。软件写 1 清 0。

2.4.6. 控制寄存器(FMC_CTL)

地址偏移：0x14
复位值：0xC000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	OBLK	SCR1	SCR0	OBRLD	RPERRIE	ERRIE	ENDIE	保留						OBSTART	START
rs	rs	rs	rs	rc_w1	rw	rw	rw							rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MER1	保留		BKSEL	保留	PNSEL[7:0]								MER0	PER	PG
rw			rw		rw								rw	rw	rw

位/位域	名称	描述
31	LK	FMC_CTL 寄存器锁定标志位 当正确的序列写入 FMC_KEY 寄存器，此位由硬件清 0。此位可以由软件置 1。
30	OBLK	FMC_OBCTL 锁定标志位

		如果这个位被置1，FMC_OBCTL寄存器中关于用户选项字节的所有位以及选项字节页都将被锁定。当正确的序列被写入FMC_OBKEY寄存器时，该位被硬件清除。该位可通过软件设置。此位可以由软件置1。
29	SCR1	Bank1安全用户区域使能位 该位置1时锁定bank1的安全用户区域。当退出安全用户区域时可以由软件设置，该位只能写一次。在DBS=0的情况下，该位不起作用。 0：失能bank1的安全用户区域。 1：使能bank1的安全用户区域。
28	SCR0	Bank0安全用户区域使能位 该位置1时锁定bank0的安全用户区域。当退出安全用户区域时可以由软件设置，该位只能写一次。 0：失能bank0的安全用户区域。 1：使能bank0的安全用户区域。
27	OBRLD	选项字节重加载位 此位可以由软件置 1。 0：完成选项字节重加载。 1：强制选项字节重加载。 **注意：**1. 如果当 OBLK 置位时，该位无法写入。2. OBSTART 置位时，该位无法写入，OBSTART 和 OBRLD 两个同时写，两位都不生效。
26	RPERRIE	读保护错误中断使能位 当 LK 设置为 0 时，该位才能被软件置 1 和清 0。 0：失能读保护错误中断。 1：使能读保护错误中断。
25	ERRIE	操作失败错误中断使能位 当 LK 设置为 0 时，该位才能被软件置 1 和清 0。 0：失能操作失败错误中断。 1：使能操作失败错误中断。
24	ENDIE	操作结束中断使能位 当 LK 设置为 0 时，该位才能被软件置 1 和清 0。 0：失能操作结束中断。 1：使能操作结束中断。
23:18	保留	必须保持复位值
17	OBSTART	发送选项字节更改命令位 该位由软件置 1，只有当 OBLK 设置为 0 时，才向 FMC 发送选项字节更改命令。当 BUSY 位被清除时，该位被硬件清除。

16	START	向 FMC 发送擦除命令位 当 LK 设置为 0 时，该位才能被软件置 1 和清 0。用于向 FMC 发送擦除命令。当 BUSY 位被清除时，该位被硬件清除。 当MER0、MER1和PER位均清0，如果START位被置1，可能会出现不可预知的问题，而不会产生任何错误标志。应当禁止这种操作。
15	MER1	Bank1批量擦除选择位 当 LK 设置为 0 时，该位才能被软件置 1 和清 0。 0：无作用。 1：选择 bank1 批量擦除。
14:13	保留	必须保持复位值
12	BKSEL	选择bank号来进行页擦除 DBS=1 0：选择bank0进行页擦除。 1：选择bank1进行页擦除。 DBS=0 保留，必须保持复位值。
11	保留	必须保持复位值
10:3	PNSEL[7:0]	擦除页码选择位 这些位用于选择要擦除的页码： 00000000：第0页。 00000001：第1页。 ... 11111111：第 255 页。
2	MER0	Bank0批量擦除选择位 当 LK 设置为 0 时，该位才能被软件置 1 和清 0。 0：无作用。 1：选择 bank0 批量擦除。
1	PER	页擦除命令位 当 LK 设置为 0 时，该位才能被软件置 1 和清 0。 0：无作用。 1：使能页擦除命令。
0	PG	主存储块编程命令位 软件置 1 和清 0 0：失能主存储块编程命令。 1：使能主存储块编程命令。

2.4.7. ECC 控制与状态寄存器(FMC_ECCCS)

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
ECCDET0	ECCCOR0	ECCDET1	ECCCOR1	保留			ECCCORIE	保留	SYS_ECC	BK_ECC	保留		ECCADDR[18:16]		
rc_w1	rc_w1	rc_w1	rc_w1				rw		r	r				r	

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

位/位域	名称	描述
31	ECCDET0	检测到双位错误标志位 DBS=0： 当在LSB（bits63：0）检测到两个ECC错误时此位置位。软件写1清0。 DBS=1： 当检测到两个ECC错误时此位置位。软件写1清0。 0：未检测到 ECC 双位错误。 1：检测到ECC双位错误。
30	ECCCOR0	检测并纠正单个位错误标志 DBS=0： 当在LSB（bits63：0）检测并纠正单个ECC错误时此位置位。软件写1清0。 DBS=1： 当检测并纠正单个ECC错误时此位置位。软件写1清0。 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	ECCDET1	检测到双位错误标志位 DBS=0： 当在MSB（bits127：64）检测到两个ECC错误时此位置位。软件写1清0。 DBS=1： 保留，必须保持复位值。 0：未检测到 ECC 双位错误。 1：检测到 ECC 双位错误。
28	ECCCOR1	检测并纠正单个位错误标志 DBS=0： 当在MSB（bits127：64）检测并纠正单个ECC错误时此位置位。软件写1清0。

		DBS=1： 保留，必须保持复位值。 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。
27:25	保留	必须保持复位值
24	ECCCORIE	纠正单个位中断使能 0：失能纠正单个位中断。 1：使能纠正单个位中断。 当ECCCOR位置1时，此位使能时生成一个中断。
23	保留	必须保持复位值
22	SYS_ECC	当检测到系统引导装载程序 ECC 纠错或双位 ECC 错误时，此位置 1。 0：未检测到系统引导装载程序中 ECC 纠错或双位 ECC 错误。 1：检测到系统引导装载程序中 ECC 纠错或双 ECC 错误。
21	BK_ECC	ECC错误bank号 DBS=1： 表示ECC纠错或双ECC纠错发生在哪个bank。 0：Bank 0 1：Bank 1 DBS=0： 如果SYS_ECC置1，表明系统引导装载程序ECC纠错或双ECC纠错发生在哪个bank。如果SYS_ECC为0，保留，必须保持复位值。
20:19	保留	必须保持复位值
18:0	ECCADDR[18:0]	ECC错误地址 DBS=0 表示在闪存中发生了ECC纠错或双位ECC错误的地址。 DBS=1 表示在某一bank中发生了ECC纠错或双位ECC错误的地址。

2.4.8. 选项字节控制寄存器(FMC_OBCTL)

地址偏移：0x20
复位值：0xXXXX XXXX

该寄存器只能按字(32 位)访问。

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

保留	NRST_MDSEL[1:0]	nBOOT0	nSWBT0	TCMSRAM_ERS	SRAM_ECCEN	nBOOT1	DBS	保留	BB	保留	FWDGSPD_STDBY	FWDGSPD_DPSLP	nFWDG_HW
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

FMC_SWP	保留	nRST_STDBY	nRST_DPSLP	保留	BOR_TH[1:0]	SPC[7:0]
rw		rw	rw		rw	rw

位/位域	名称	描述
31:30	保留	必须保持复位值
29:28	NRST_MDSEL[1:0]	NRST引脚模式选择位. 00：NRST引脚配置为复位输入/输出模式。 01：NRST引脚上的低电平可以复位系统，内部复位不能驱动NRST引脚。 10：NRST引脚功能与普通GPIO相同，只有内部复位。 11：NRST引脚配置为复位输入/输出模式。
27	nBOOT0	BOOT0选择位 0：BOOT0为1。 1：BOOT0 为 0。
26	nSWBT0	软件BOOT0失能位 0：BOOT0取决于选项字节位nBOOT0。 1：BOOT0 取决于 PB8/BOOT0 引脚。
25	TCMSRAM_ERS	当系统复位时TCM SRAM擦除失能位 0：系统复位时擦除TCM SRAM。 1：系统复位时不擦除 TCM SRAM。
24	SRAM_ECCEN	SRAM与TCM SRAM ECC失能位 0：使能SRAM与TCM SRAM ECC。 1：失能SRAM与TCM SRAM ECC。
23	nBOOT1	Boot1选择位 0：BOOT1为1。 1：BOOT1为0。 与BOOT0共同决定启动模式。
22	DBS	单双bank模式选择位 0：选择单bank模式，128位读取位宽。 1：选择双bank模式，64位读取位宽。 该位只能在禁用DCRP时写入。

21	保留	必须保持复位值
20	BB	Boot 启动 bank 选择位 0：当配置为从主闪存启动时，从 bank0 启动。 1：当配置为从主内存启动时，从bank1启动，若bank1为空则从bank0启动。(或者当bank0，bank1均空且SPC等级不为高时从bootloader启动)。 **注意**：在该位置1前必须先将FMC_SWP位置1。
19	保留	必须保持复位值
18	FWDGSPD_STDBY	当系统处于待机模式时独立看门狗的运行状态 0：当系统处于待机模式时独立看门狗暂停运行。 1：当系统处于待机模式时独立看门狗继续运行。
17	FWDGSPD_DPSLP	当系统处于深度睡眠模式时独立看门狗的运行状态 0：当系统处于深度睡眠模式时独立看门狗暂停运行。 1：当系统处于深度睡眠模式时独立看门狗继续运行。
16	nFWDG_HW	软件独立看门狗配置位 0：硬件独立看门狗。 1：软件独立看门狗。
15	FMC_SWP	FMC存储器映射切换。 这些位控制主FLASH存储器的Bank0和Bank1的地址映射切换功能。 0：主FLASH存储器的Bank0被映射到地址0x0804 0000，主FLASH存储器的Bank1被映射到地址0x0800 0000 1：主FLASH存储器的Bank0映射到地址0x0800 0000，主FLASH存储器的Bank1映射到地址0x0804 0000 **注意**：以上地址取决于具体芯片系列的扇区大小。使能BB位后该位不能置0。
14	保留	必须保持复位值
13	nRST_STDBY	选项字节待机复位选择位 0：当系统进入待机模式时复位。 1：当系统进入待机模式时不复位。
12	nRST_DPSLP	选项字节深度睡眠复位选择位 0：当系统进入深度睡眠模式时复位。 1：当系统进入深度睡眠模式时不复位。
11:10	保留	必须保持复位值
9:8	BOR_TH[1:0]	选项字节BOR阈值 00：BOR功能关闭。 01：BOR阈值1。阈值约2.2 V。 10：BOR阈值2。阈值约2.5 V。

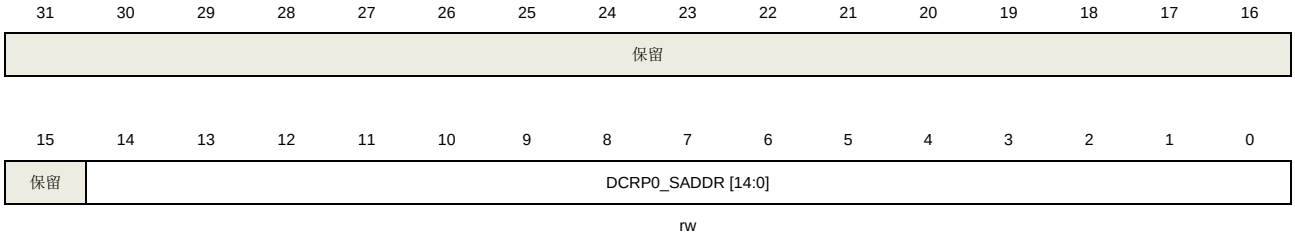
11：BOR阈值3。阈值约2.8 V。

7:0	SPC[7:0]	选项字节安全保护等级 0xAA：无安全保护。 0xCC：高安全保护。 除 0xAA 或 0xCC 之外任何值：低安全保护。

2.4.9. DCRP0 起始地址(FMC_DCRP_SADDR0)

地址偏移：0x24
复位值：0xFFFF XXXX

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:15	保留	必须保持复位值
14:0	DCRP0_SADDR [14:0]	Bank0 仅执行区域起始地址 DBS=1： DCRP0_SADDR包含bank0的仅执行区域起始地址。 DBS=0： DCRP0_SADDR包含整个闪存的第1个仅执行区域起始地址。

2.4.10. DCRP0 结束地址(FMC_DCRP_EADDR0)

地址偏移：0x28
复位值：0xX0FF XXXX

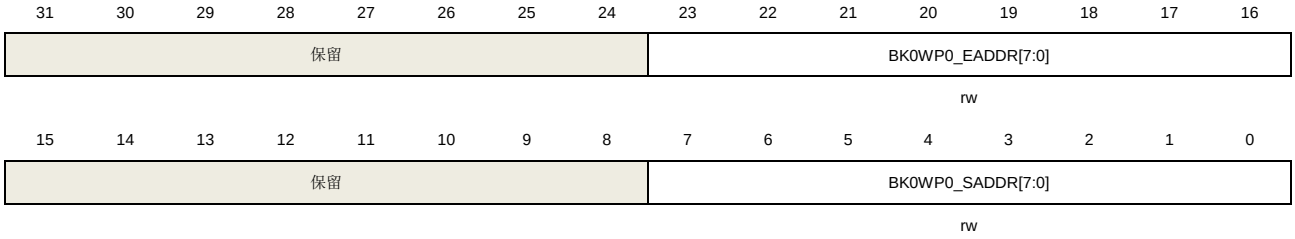
该寄存器只能按字(32 位)访问。

位/位域	名称	描述
31	DCRP_EREN	DCRP 区域配置擦除使能位。 该位只能置 1。此位仅能在当 SPC 等级从低保护等级降到无保护等级时清除。 0：当SPC等级从低保护等级降到无保护等级时，DCRP区域不擦除。 1：当 SPC 等级从低保护等级降到无保护等级时，DCRP 区域擦除。
30:15	保留	必须保持复位值
14:0	DCRP0_EADDR [14:0]	Bank0 仅执行区域结束地址 DBS=1： DCRP0_EADDR包含bank0的仅执行区域结束地址。 DBS=0： DCRP0_EADDR 包含整个闪存的第 1 个仅执行区域结束地址。

2.4.11. Bank0 擦除/编程保护区域 0 寄存器(FMC_BK0WP0)

地址偏移：0x2C
复位值：0xFEXX FEXX

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:24	保留	必须保持复位值
23:16	BK0WP0_EADDR[7:0]	第1个编程/保护区域尾页 DBS=1： BK0WP0_EADDR[7:0]包含在bank0的第1个WP区域的尾页。 DBS=0： BK0WP0_EADDR[7:0]包含在主存储闪存的第 1 个 WP 区域的尾页。
15:8	保留	必须保持复位值
7:0	BK0WP0_SADDR[7:0]	第1个编程/保护区域首页 DBS=1： BK0WP0_SADDR[7:0]包含在bank0的第1个WP区域的首页。 DBS=0：

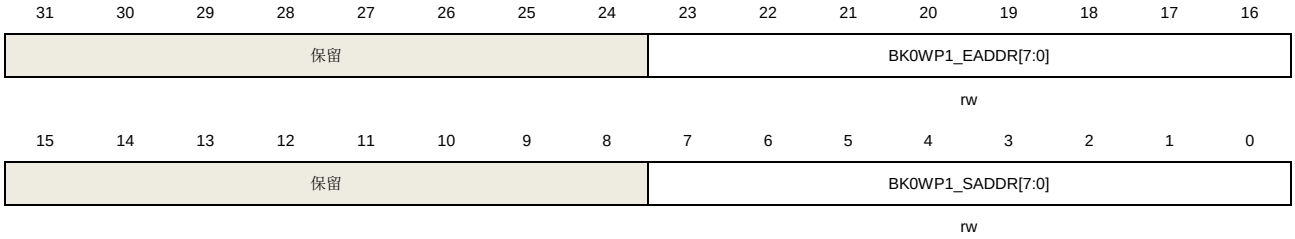
BK0WP0_SADDR[7:0]包含在主存储闪存的第1个WP区域的首页。

2.4.12. Bank0 擦除/编程保护区域 1 寄存器(FMC_BK0WP1)

地址偏移：0x30

复位值：0xFEXX FEXX

该寄存器只能按字(32 位)访问。



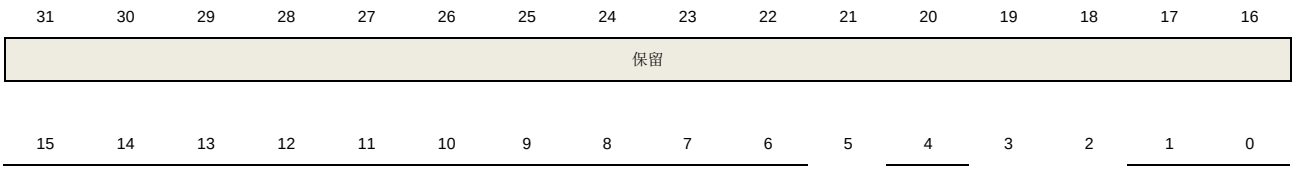
位/位域	名称	描述
31:24	保留	必须保持复位值
23:16	BK0WP1_EADDR[7:0]	第2个编程/保护区域尾页 DBS=1： BK0WP1_EADDR[7:0]包含在bank0的第2个WP区域的尾页。 DBS=0： BK0WP1_EADDR[7:0]包含在主存储闪存的第 2 个 WP 区域的尾页。
15:8	保留	必须保持复位值
7:0	BK0WP1_SADDR[7:0]	第2个编程/保护区域首页 DBS=1： BK0WP1_SADDR[7:0]包含在bank0的第2个WP区域的首页。 DBS=0： BK0WP1_SADDR[7:0]包含在主存储闪存的第2个WP区域的首页。

2.4.13. DCRP1 起始地址(FMC_DCRP_SADDR1)

地址偏移：0x44

复位值：0xFFFF XXXX

该寄存器只能按字(32 位)访问。



保留	DCRP1_SADDR [14:0]

rw

位/位域	名称	描述
31:15	保留	必须保持复位值
14:0	DCRP1_SADDR [14:0]	Bank1 仅执行区域起始地址 DBS=1： DCRP1_SADDR包含bank1的仅执行区域起始地址。 DBS=0： DCRP1_SADDR包含整个闪存的第2个仅执行区域起始地址。

2.4.14. DCRP1 结束地址(FMC_DCRP_EADDR1)

地址偏移：0x48
复位值：0x00FF 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	DCRP1_EADDR [14:0]														

rw

位/位域	名称	描述
31:15	保留	必须保持复位值
14:0	DCRP1_EADDR [14:0]	Bank1 仅执行区域结束地址 DBS=1： DCRP1_EADDR包含bank1的仅执行区域结束地址。 DBS=0： DCRP1_EADDR 包含整个闪存的第 2 个仅执行区域结束地址。

2.4.15. Bank1 擦除/编程保护区域 0 寄存器(FMC_BK1WP0)

地址偏移：0x4C
复位值：0xFEXX FEXX

该寄存器只能按字(32 位)访问。

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

保留	BK1WP0_EADDR[7:0]
	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								BK1WP0_SADDR[7:0]							
								rw							

位/位域	名称	描述
31:24	保留	必须保持复位值
23:16	BK1WP0_EADDR[7:0]	第3个编程/保护区域尾页 DBS=1： BK1WP0_EADDR[7:0]包含在bank1的第1个WP区域的尾页。 DBS=0： BK1WP0_EADDR[7:0]包含在主存储闪存的第 3 个 WP 区域的尾页。
15:8	保留	必须保持复位值
7:0	BK1WP0_SADDR[7:0]	第3个编程/保护区域首页 DBS=1： BK1WP0_SADDR[7:0]包含在bank1的第1个WP区域的首页。 DBS=0： BK1WP0_SADDR[7:0]包含在主存储闪存的第3个WP区域的首页。

2.4.16. Bank1 擦除/编程保护区域 1 寄存器(FMC_BK1WP1)

地址偏移：0x50

复位值：0xFEXX FEXX

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								BK1WP1_EADDR[7:0]							
								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								BK1WP1_SADDR[7:0]							
								rw							

位/位域	名称	描述
31:24	保留	必须保持复位值
23:16	BK1WP1_EADDR[7:0]	第4个编程/保护区域尾页 DBS=1： BK1WP1_EADDR[7:0]包含在bank1的第2个WP区域的尾页。

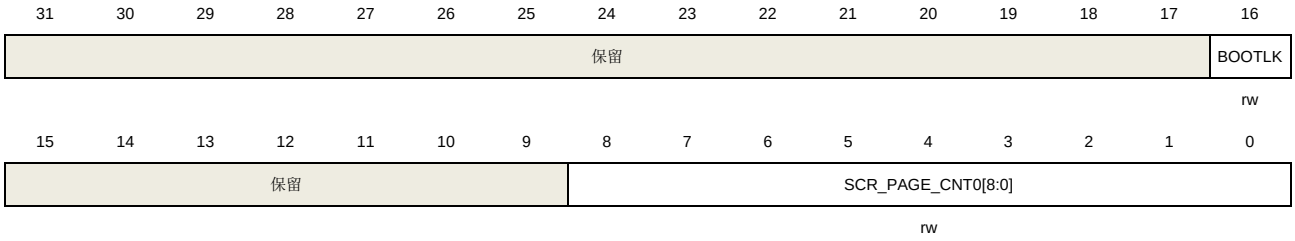
		DBS=0： BK1WP1_EADDR[7:0]包含在主存储闪存的第 4 个 WP 区域的尾页。
15:8	保留	必须保持复位值
7:0	BK1WP1_SADDR[7:0]	第4个编程/保护区域首页 DBS=1： BK1WP1_SADDR[7:0]包含在bank1的第2个WP区域的首页。 DBS=0： BK1WP1_SADDR[7:0]包含在主存储闪存的第4个WP区域的首页。

2.4.17. Bank0 安全用户区域寄存器(FMC_BK0SCR)

地址偏移：0x70

复位值：0xFF0X FXXX

该寄存器只能按字(32 位)访问。



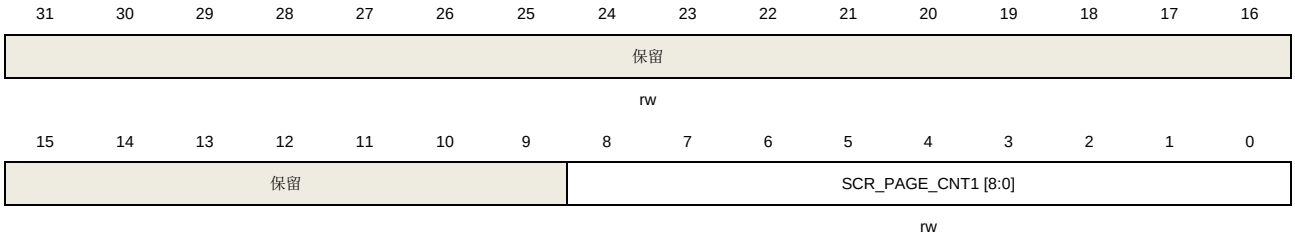
位/位域	名称	描述
31:17	保留	必须保持复位值
16	BOOTLK	该位置1时强制从用户闪存启动 0：支持闪存，RAM和系统启动。 1：只能从主闪存启动。
15:9	保留	必须保持复位值
8:0	SCR_PAGE_CNT0[8:0]	配置bank0安全用户区域的页数 安全用户区域从bank0基地址开始。内存大小为SCR_PAGE_CNT0乘以页面大小。 该区域仅当SPC等级为无保护时可以修改。

2.4.18. Bank1 安全用户区域寄存器(FMC_BK1SCR)

地址偏移：0x74

复位值：0xFF00 FXXX

该寄存器只能按字(32 位)访问。

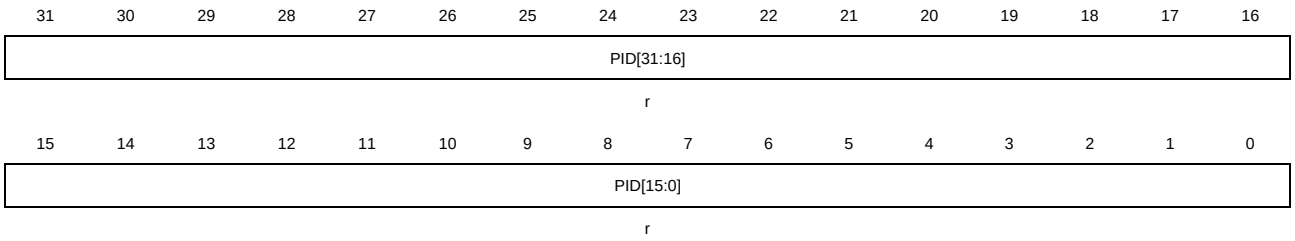


位/位域	名称	描述
31:9	保留	必须保持复位值
8:0	SCR_PAGE_CNT1 [8:0]	配置bank1安全用户区域的页数 安全用户区域bank1基地址开始。内存大小为SCR_PAGE_CNT1乘以页面大小。 该区域仅当 SPC 等级为无保护时可以修改。

2.4.19. 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



位/位域	名称	描述
31:0	PID[31:0]	产品 ID 寄存器 该寄存器为只读 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 GD32G553 系列产品比较注重的问题之一。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，GD32G553 系列设备有三个电源域，包括 V_{DD} / V_{DDA} 域，V_{CORE} 域和备份域。V_{DD} / V_{DDA} 域由电源直接供电。在 V_{DD} / V_{DDA} 域嵌入的 LDO 用来为 V_{CORE} 域供电。在备份域中有一个电源切换器，当 V_{DD} 电源关闭时，电源切换器可以将备份域的电源切换到 VBAT 引脚，此时备份域由 VBAT 引脚（电池）供电。

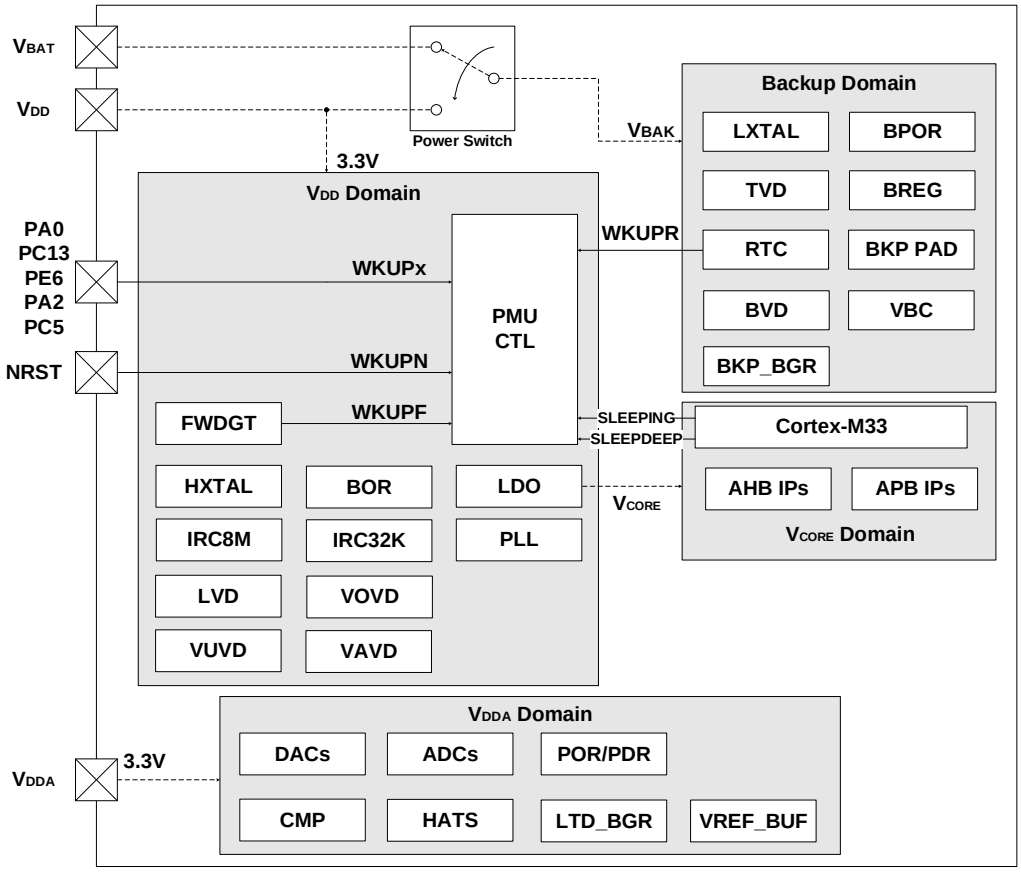
3.2. 主要特征

- 三个电源域：备份域、V_{DD} / V_{DDA}域和V_{CORE}电源域。
- 三种省电模式：睡眠模式，深度睡眠模式，和待机模式。
- 内部电压调节器（LDO）为核心电源域V_{CORE}提供V_{CORE}电源。
- 提供低电压检测器（LVD），当电压低于所设定的阈值时能发出中断或事件。
- 当V_{DD}供电关闭时，由V_{BAT}（电池）为备份域供电。
- 供电监控：POR / PDR监控、BOR监控、LVD监控、VOVD监控、VAVD监控、VUVD监控、V_{BAK}阈值监测、温度阈值监测。

3.3. 功能说明

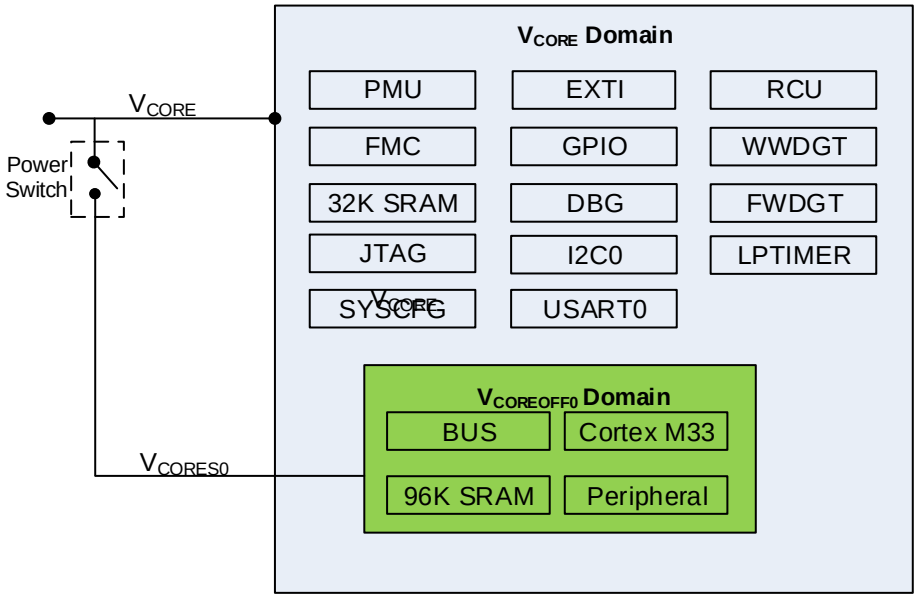
*图 3-1. 电源域概览*提供了 PMU 及相关电源域的内部结构框图。

图 3-1. 电源域概览



LVD: 低电压检测器
LDO: 电压调节器
BPOR:备份域上电复位
VOVD: V_{CORE} 过电压检测器
POR: 上电复位
PDR: 掉电复位
VAVD:模拟电压检测器
VUVD: V_{CORE} 欠电压检测器
BVD: V_{BAK} 电压检测器
BREG:备份寄存器
TVD: 温度电压检测器
BOR: 欠压复位
VBC: V_{BAK} 电池充电
BKP_BGR: 备份域带隙基准
HATS: 高精度温度传感器
LTD_BGR: 低温漂带隙基准
VREF_BUF: 电压缓冲器

图 3-2. V_{CORE} 电源域概览



如图*图 3-2. VCORE 电源域概览*所示，V_{CORE} 域包含两个部分：

1. $V_{COREOFF0}$ 域包含：总线，Cortex®-M33 CPU，剩余 96K SRAM（除前 32K），外设；
2. 深度睡眠保持供电模块：FMC，PMU，RCU，EXTI，GPIO，前 32K SRAM，DBG，FWDGT，JTAG，I2C0，LPTIMER，SYSCFG，USART0。

电源切换器 S0 用于省电模式的供电控制。

3.3.1. 备份域

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

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

RTC的时钟源可以是低速内部32KHz RC振荡器（IRC32K）或低速外部晶体振荡器（LXTAL），或由高速外部晶体振荡器（HXTAL）时钟32分频。当V_{DD}被关闭时，RTC只能选择LXTAL作为时钟源。在通过WFI / WFE指令进入省电模式之前，Cortex®-M33能够通过RTC寄存器预期的唤醒时间并启用唤醒功能或者根据EXTI，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）*来描述。

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

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

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

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

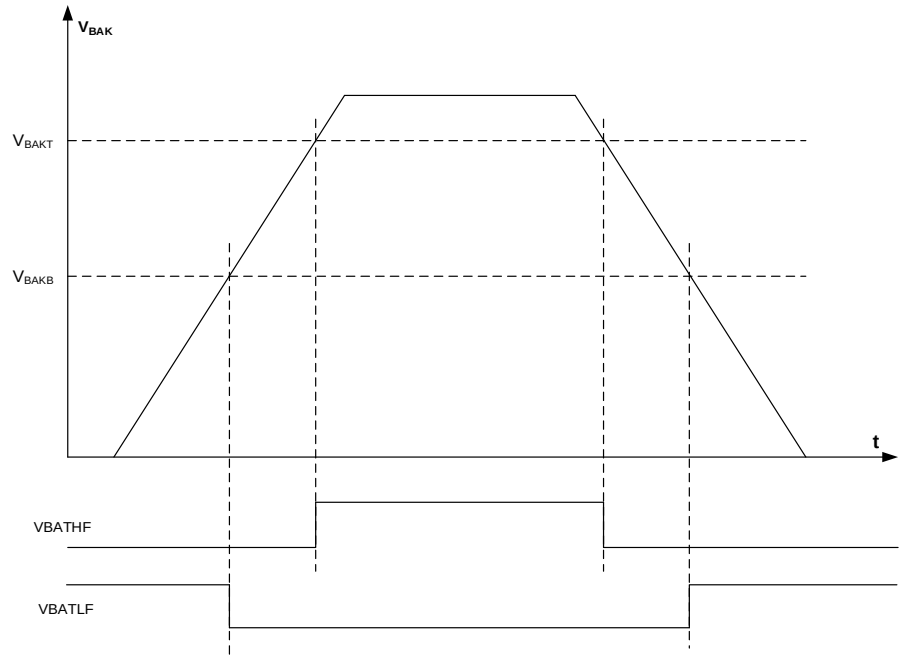
注意：由于 PC13 至 PC15 引脚是通过电源切换器供电的，电源切换器仅可通过小电流，因此当 PC13 至 PC15 的 GPIO 口在输出模式时，其工作的速度不能超过 2MHz（最大负载为 30pF）。

V_{DD} 可以通过一个内部电阻给外部电池充电。通过配置 PMU_CTL2 寄存器中 VCRSEL 位，可以选择内部电阻 5K 欧姆或 1.5K 欧姆用于外部 V_{BAT} 电池充电。将 PMU_CTL2 寄存器中 VCEN 位置 1 可以使能 V_{BAT} 电池充电。在 BKP only 模式，V_{BAT} 电池充电不可用。

备份域电压阈值监测

芯片内部有一个内部电源开关，可以选择备份域的电压源为 V_{BAT} 或 V_{DD}。当 VBTMEN 位置位时，备份域（V_{BAK}）的电源电压可以通过上限电压和下限电压（V_{BAKT} 和 V_{BAKB}）进行监控，如果 V_{BAK} 超过 V_{BAKT}，则标志位 VBATHF 将设置，如果 V_{BAK} 低于 V_{BAKB}，则标志位 VBATLF 将设置。*图 3-3. 备用域电压阈值的波形*，显示了备用域电压阈值监测。

图 3-3. 备用域电压阈值的波形

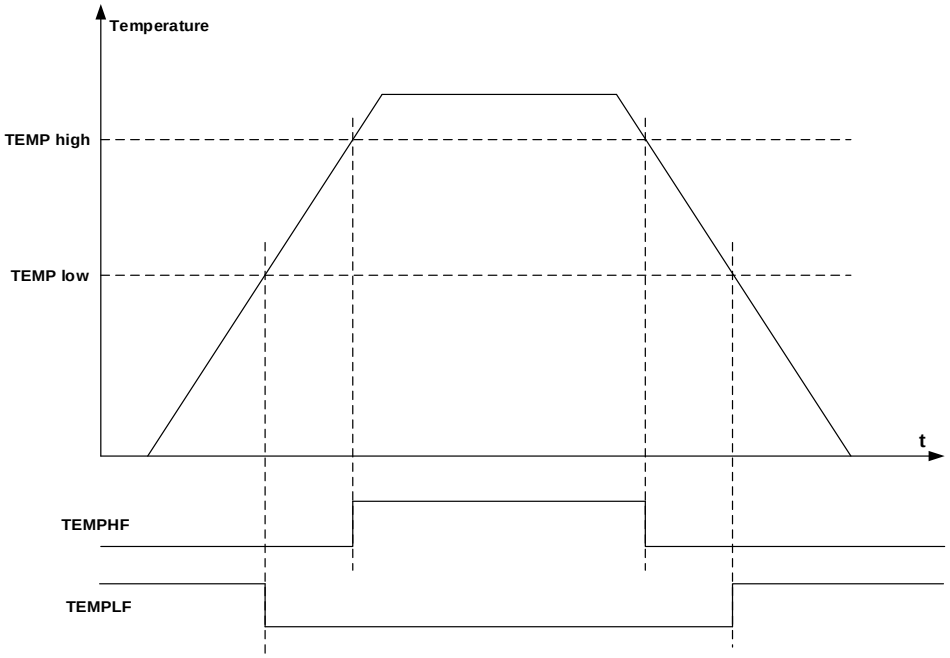


温度电压阈值监测

和备份域电压阈值监测类似，通过与温度高、低两个阈值水平比较可以来监测结温。PMU_CTL1 寄存器中 TEMPHF 和 TEMPLF 标志指示设备温度是否高于或低于阈值。可以通过 PMU_CTL1 寄存器中的 VBTMEN 位使能 / 关闭温度电压阈值监测。使能后，温度阈值监测将增加功耗。温度阈值监测可以用来触发执行温度控制任务的相关的程序。只有 PMU_CTL1 寄存器中的 BKPVSEN 位置位，温度阈值监测才有效。

TEMPH 和 TEMPL 唤醒中断可用于 RTC 侵入信号

图 3-4. 温度阈值监测



3.3.2. V_{DD} / V_{DDA} 电源域

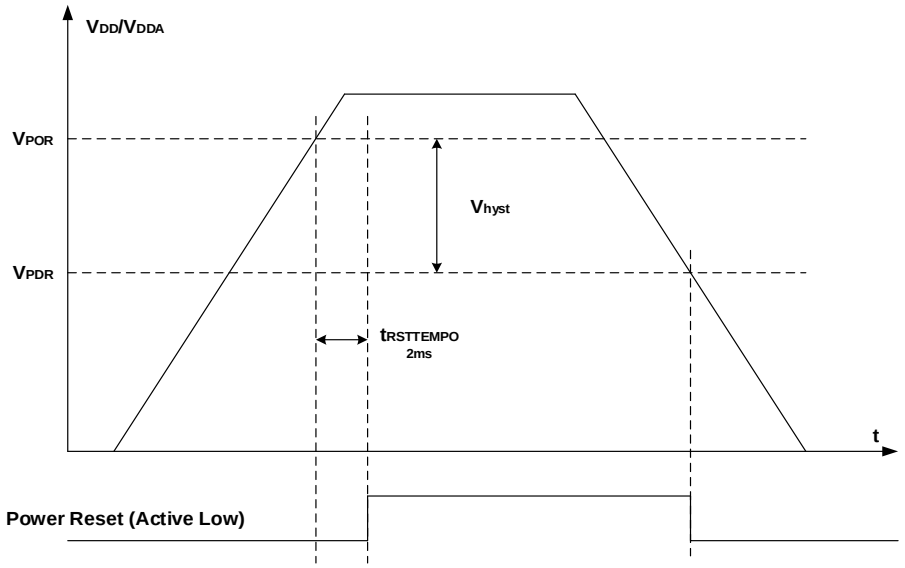
V_{DD} / V_{DDA} 域包括 V_{DD} 域和 V_{DDA} 域两部分。V_{DD} 域包括 HXTAL（高速外部晶体振荡器）、IRC8M（内部 8MHz RC 振荡器时钟）、IRC32K（内部 32KHz RC 振荡器时钟）、LDO（电压调节器）、BOR（欠压复位）、FWDGT（独立看门狗定时器）、PLL（锁相环）、LVD（低电压检测器）、VOVD（V_{CORE} 过压检测器）、VAVD（V_{DDA} 电压检测器）、VUVD（V_{CORE} 低压检测器）和除 PC13、PC14 和 PC15 之外的所有 PAD 等。V_{DDA} 域包括 ADC / DAC（AD / DA 转换器）、POR / PDR（上电 / 掉电复位）、CMP（比较器），HATS（高精度温度传感器）、LTD_BGR（低温漂带隙基准）、VREF_BUF（电压缓冲器）等。

V_{DD} 域

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

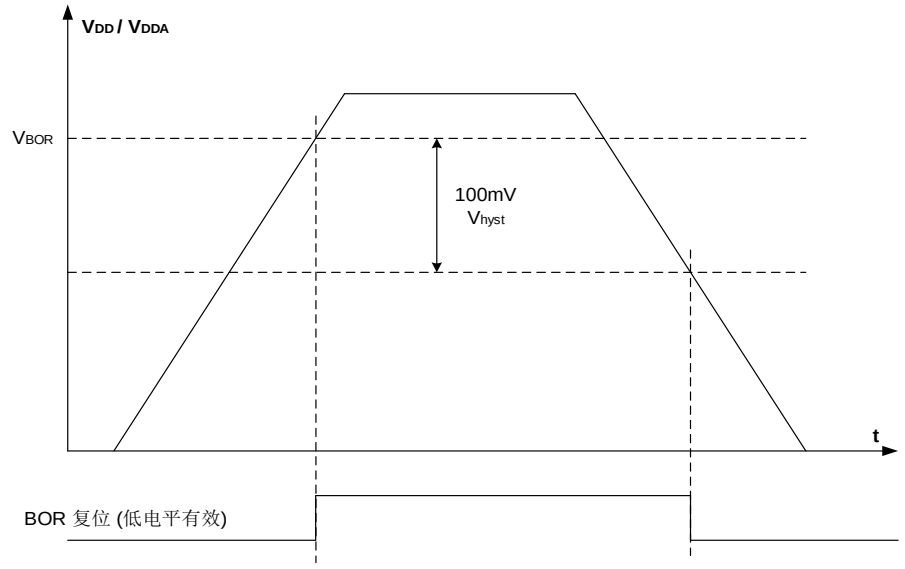
POR / PDR（上电 / 掉电复位）电路检测 V_{DD} / V_{DDA} 并在电压低于特定阈值时产生电源复位信号复位除备份域之外的整个芯片。*图 3-5. 上电 / 掉电复位波形图*显示了供电电压和电源复位信号之间的关系。V_{POR} 表示上电复位的阈值电压， V_{PDR} 表示掉电复位的阈值电压。迟滞电压 V_{hyst} 值参考 datasheet。

图 3-5. 上电 / 掉电复位波形图



BOR 电路检测 V_{DD} / V_{DDA} 并在 BOR_TH 不为 0b11，同时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。POR / PDR（上电/掉电复位）电路处于检测状态，无论选项字节的 BOR_TH 是否为 0b11。*图 3-6. BOR 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 BOR 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} 值参考 datasheet。

图 3-6. BOR 波形图

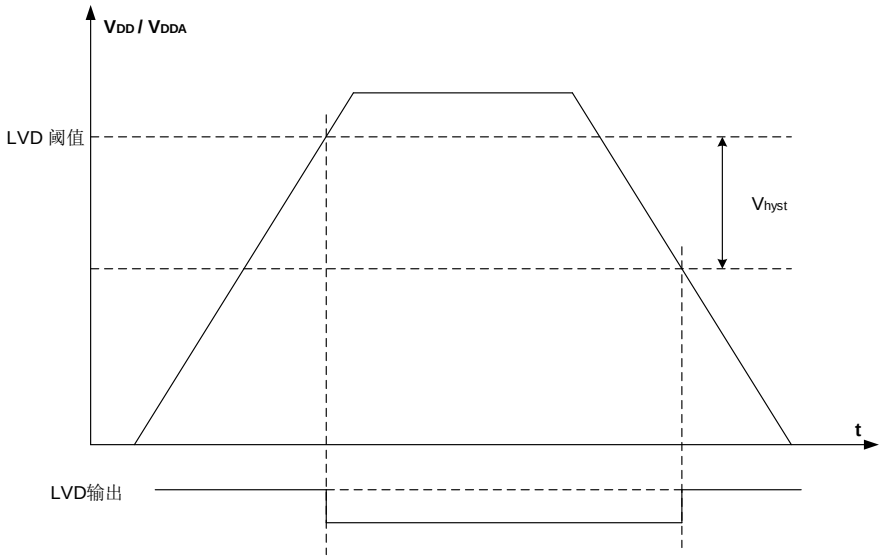


V_{DDA} 域

LVD 的功能是检测 V_{DD} / 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 0（PMU_CTL0）中的 LVDT[2:0]位进行配置。LVD 通过 LVDEN 置位使能，位于电源控制状态寄存

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 3-7. LVD 阈值波形图*显示了 V_{DD} / V_{DDA} 供电电压和 LVD 输出信号的关系。（LVD 中断信号依赖于 EXTI 第 16 线的上升或下降沿配置）。迟滞电压 V_{hyst} 值参考 datasheet。

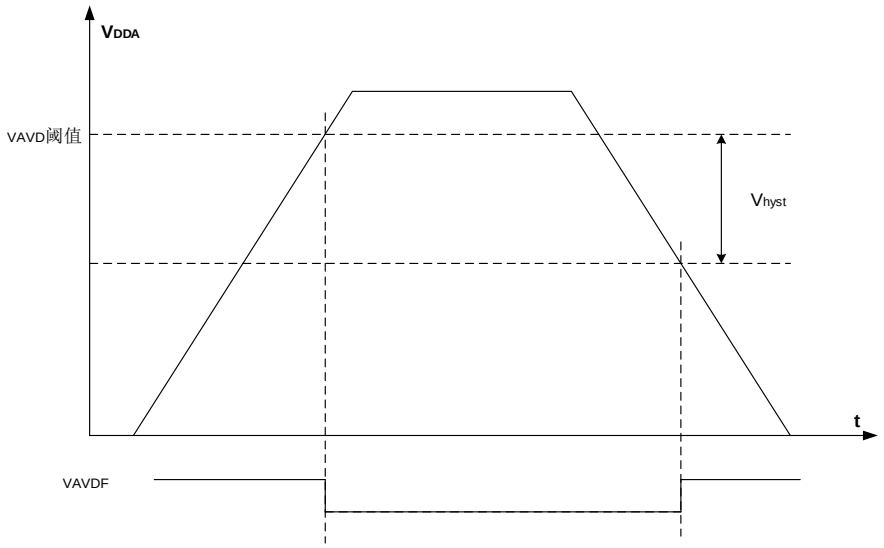
图 3-7. LVD 阈值波形图



V_{DDA} 模拟电压检测器（VAVD）用于检测 V_{DDA} 电源电压是否低于电源控制寄存器（PMU_CTL0）中 VAVDVC[1:0]位域选择的编程阈值。通过置位 VAVDEN 位能够使能 VAVD，PMU_CS 寄存器中的 VAVDF 位指示 V_{DDA} 高于或低于指定的 VAVD 阈值，如果 VAVDF 置位能够产生对应的事件，这个事件在内部连接到 EXTI 第 16 线。如果通过 EXTI 寄存器使能，可以产生一个中断。

*图 3-8. VAVD 阈值监测波形图*显示了 VAVD 门限与 VAVDF 之间的关系。迟滞电压 V_{hyst} 值参考 datasheet。

图 3-8. VAVD 阈值监测波形图



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，VDDA 通过外部滤波电路连接至 VDD，相应的 VSSA 通过特定电路连接至 VSS。当 V_{DDA} 和 V_{DD} 不同时，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 VREFP。根据不同的封装，VREFP 可被连接至 VDDA 引脚，或者外部参考电压，外部参考电压的范围请参考*表 17-2. ADC 输入引脚定义*和*表 18-1. DAC 引脚*。

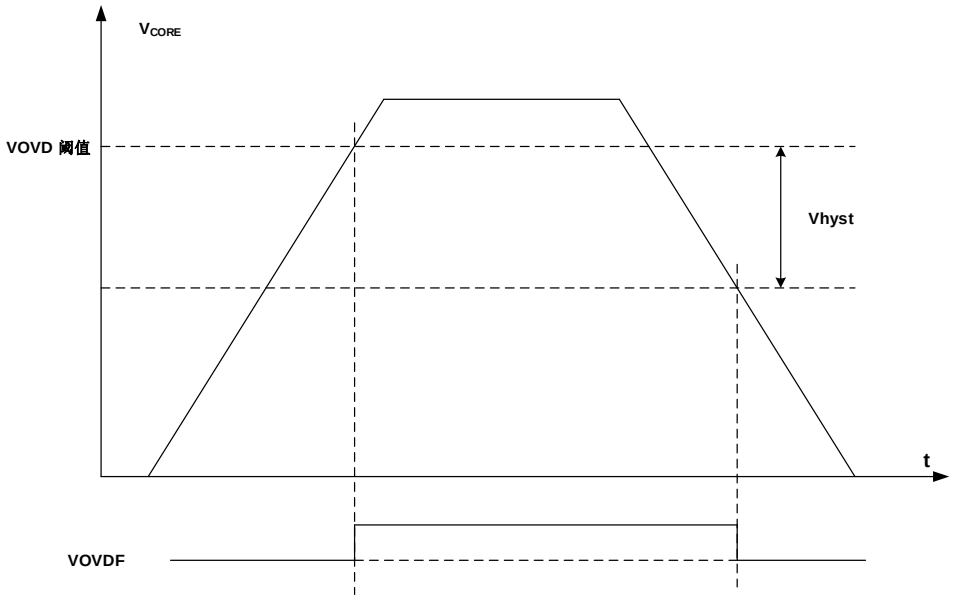
3.3.3. V_{CORE} 电源域

主要功能包括 Cortex®-M33 内核逻辑、AHB / APB 外设、备份域和 V_{DD} / V_{DDA} 域的 APB 接口等。当 V_{CORE} 电压上电后，POR 将在 V_{CORE} 域中产生一个复位序列，复位完成后，如果要进入指定的省电模式，须先配置相关的控制位，之后一旦执行 WFI 或 WFE 指令，设备便进入该省电模式。可通过 PMU_CTL0 寄存器中的 LDOVS[4:0]配置该电源域的电压。

V_{CORE} 电源域电源监测

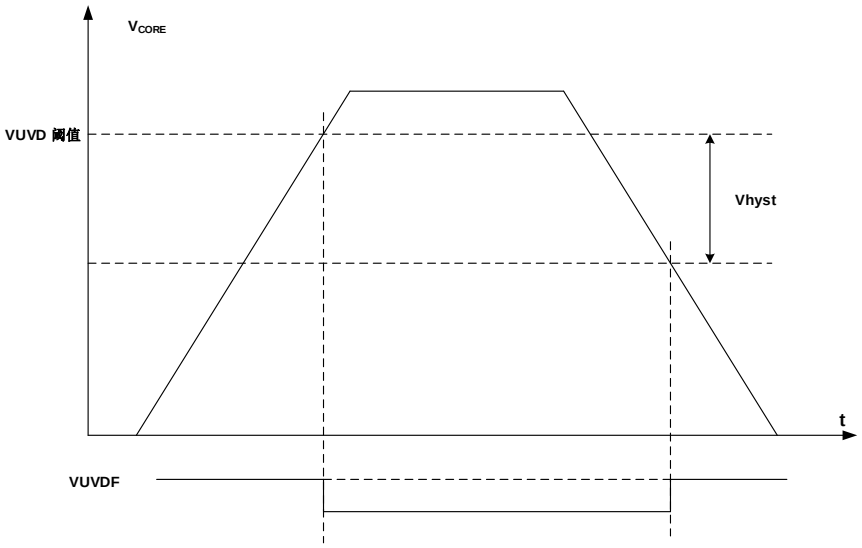
芯片内部有一个 V_{CORE} 电源域电压监测器，当 VOVDEN 为 0b1，将使能 V_{CORE} 电源域电压检测器。一旦 V_{CORE} 电源域超过电源控制寄存器（PMU_CTL0）中 VOVDVC[1:0]位选择的编程阈值，在模拟两个触发器同步后，VOVDF0 将立即被置位。该事件在内部连接到 EXTI16，如果通过 EXTI 使能，则可以产生中断。通过配置 PMU_CTL3 寄存器中的 VOVDO_DNF[7:0]位，可以使用数字滤波后的 VOVDF1。允许抑制峰值的可编程长度为 1024* T_{pclk} 的 1 到 255。VOVDF1 中断与 IRQ63 内部连接。滞电压 V_{hyst} 值参考 datasheet。

图 3-9. VOVD 波形



芯片内部有一个 V_{CORE} 电源域电压监测器，当 VUVDEN 为 0b1，将使能 V_{CORE} 电源域电压检测器。一旦 V_{CORE} 电源域低于电源控制寄存器（PMU_CTL0）中 VUVDVC[1:0]位选择的编程阈值，在模拟两个触发器同步后，VUVDF0 将立即被置位。该事件在内部连接到 EXTI16，如果通过 EXTI 使能，则可以产生中断。通过配置 PMU_CTL3 寄存器中的 VUVDO_DNF[7:0]位，可以使用数字滤波后的 VUVDF1。允许抑制峰值的可编程长度为 1024* T_{pclk} 的 1 到 255。VUVDF1 中断与 IRQ63 内部连接。滞电压 V_{hyst} 值参考 datasheet。

图 3-10. VUVD 波形



3.3.4. 省电模式

系统复位或电源复位后，GD32G553 MCU 处于全功能状态且电源域全部处于供电状态。实现较低的功耗的方法有三种：减慢系统时钟（HCLK，PCLK1，PCLK2 和 PCLK3），或者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。

表3-1. 节电模式总结

模式	睡眠	深度睡眠	待机
描述	仅关闭 CPU 时钟	1. 关闭 V_{CORE} 电源域的所有时钟 2. 关闭 IRC8M 、HXTAL 和 PLL	1. 关闭 V_{CORE} 电源域的所有时钟 2. 关闭 IRC8M 、HXTAL 和 PLL
LDO 状态	开启	开启或者低功耗模式	关闭
配置	SLEEPDEEP = 0	SLEEPDEEP = 1, STBMOD = 0	SLEEPDEEP = 1 STBMOD = 1, WURST=1
进入指令	执行 WFI 或 WFE	执行 WFI 或 WFE	执行 WFI 或 WFE
唤醒	若通过 WFI 进入，则任何中断均可唤醒； 若通过 WFE 进入，则任何事件（当 SEVONPEND=1 时的中断）均可唤醒	若通过 WFI 进入，来自 EXTI 的任何中断均可唤醒； 若通过 WFE 进入，来自 EXTI 任何事件（当 SEVONPEND=1 时的中断）可唤醒	1. NRST 引脚 2. WKUP 引脚 3. FWDGT 复位 4. RTC 5. LCKMD
唤醒延时	无	IRC8M 唤醒时间，如果 LDO 在低功耗模式，增加 LDO 唤醒时间	上电序列

睡眠模式

睡眠模式与 Cortex®-M33 的 SLEEPING 模式相对应。在睡眠模式下，仅关闭 Cortex®-M33 的时钟。如需进入睡眠模式，只要清除 Cortex®-M33 系统控制寄存器中的 SLEEPDEEP 位，并执行一条 WFI 或 WFE 指令即可。如果睡眠模式是通过执行 WFI 指令进入的，任何中断都可以唤醒系统。如果睡眠模式是通过执行 WFE 指令进入的，任何唤醒事件都可以唤醒系统（如果 SEVONPEND 为 1，任何中断都可以唤醒系统，请参考 Cortex®-M33 技术手册）。由于无需在进入或退出中断上消耗时间，该模式所需的唤醒时间最短。

根据 Cortex®-M33 中 SCR（系统控制寄存器）的 SLEEPONEXIT 位，有两种睡眠进入机制可选：

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

MCU立即进入睡眠模式。

深度睡眠模式

深度睡眠模式与 Cortex®-M33 的 SLEEPDEEP 模式相对应。在深度睡眠模式下，V_{CORE} 域中的所有时钟全部关闭，IRC8M、HXTAL 及 PLL 也全部被禁用。SRAM 和寄存器的内容被保留。根据 PMU_CTL0 寄存器中的 LDOLP 位，LDO 可以正常工作，也可以在低功耗模式下工作。进入深度睡眠模式之前，先将 Cortex®-M33 系统控制寄存器的 SLEEPDEEP 位置 1，再清除 PMU_CTL0 寄存器的 STBMOD 位，然后执行 WFI 或 WFE 指令即可进入深度睡眠模式。如果睡眠模式是通过执行 WFI 指令进入的，任何来自 EXTI 的中断可以将系统从深度睡眠模式中唤醒。如果睡眠模式是通过执行 WFE 指令进入的，任何来自 EXTI 的事件可以将系统从深度睡眠模式中唤醒（如果 SEVONPEND 为 1，任何来自 EXTI 的中断都可以唤醒系统，请参考 Cortex®-M33 技术手册）。当退出深度睡眠模式时，IRC8M 被选择为系统时钟。注意，如果 LDO 在低功耗模式下工作，则会产生额外的唤醒延迟。

注意：为了顺利进入深度睡眠模式，必须重置所有 EXTI 挂起状态（EXTI_PD 寄存器）和相关的外设标志。如果没有，程序将跳过深度睡眠模式的进入过程，继续执行下面程序。

待机模式

待机模式是基于 Cortex®-M33 的 SLEEPDEEP 模式实现的。在待机模式下，整个 V_{CORE} 域全部停止供电，LDO 关闭，同时包括 IRC4M、HXTAL 和 PLL 也会被关闭。进入待机模式前，先将 Cortex®-M33 系统控制寄存器的 SLEEPDEEP 位置 1，再将 PMU_CTL0 寄存器的 STBMOD 位域置位，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。待机模式有五个唤醒源，包括来自 NRST 引脚的外部复位，RTC 闹钟，FWDGT 复位，LXTAL 时钟失败检测，WKUP 引脚的上升沿。待机模式可以达到最低的功耗，但唤醒时间最长。另外，一旦进入待机模式，SRAM 和 V_{CORE} 电源域寄存器的内容都会丢失。退出待机模式时，会发生上电复位，复位之后 Cortex®-M33 将从 0x00000000 地址开始执行指令代码。

BKP 模式

当 VDD 引脚被外部电源开关切断时，进入 BKP 模式。当 VBAT 供电时，BKP 域包括 RTC / LXTAL / BKP POR 是开启的。在这种模式下，用户可以使用 RTC，当 VDD 引脚供电时，退出 BKP 模式。

3.4. PMU 寄存器

PMU 基地址：0x4000 7000

3.4.1. 控制寄存器 0（PMU_CTL0）

地址偏移：0x00
复位值：0x0002 6000

该寄存器可以按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			VUVDVC		VOVDVC		VUVDEN	VOVDEN	VAVDVC		VAVDEN	保留		DSLPVS	
			rw		rw		rw	rw	rw		rw			rw	rw

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

位/位域	名称	描述
31:29	保留	必须保持复位值。
28:27	VUVDVC[1:0]	V_{CORE}低压检测器电压等级配置位 这些位由软件置位和清除。 00：配置V_{CORE}低压检测器电压等级为0.95V 01：配置V_{CORE}低压检测器电压等级为0.85V 10：配置V_{CORE}低压检测器电压等级为0.75V 11：配置V_{CORE}低压检测器电压等级为0.65V
26:25	VOVDVC[1:0]	V_{CORE}过压检测器电压等级配置位 这些位由软件置位和清除。 00：配置V_{CORE}过压检测器电压等级为1.25V 01：配置V_{CORE}过压检测器电压等级为1.35V 10：配置V_{CORE}过压检测器电压等级为1.45V 11：配置V_{CORE}过压检测器电压等级为1.55V
24	VUVDEN	V_{CORE}低压检测器使能位 该位由软件置位和清除。 0：失能V_{CORE}低压检测器 1：使能 V_{CORE} 低压检测器
23	VOVDEN	V_{CORE}过压检测器使能位 该位由软件置位和清除。 0：失能V_{CORE}过压检测器

		1：使能V_{CORE}过压检测器
22:21	VAVDVC[1:0]	V_{DDA}模拟电压检测器电压等级配置位 这些位由软件置位和清除。 00：配置V_{DDA}模拟电压检测器电压等级为1.8V 01：配置V_{DDA}模拟电压检测器电压等级为2.2V 10：配置V_{DDA}模拟电压检测器电压等级为2.6V 11：配置 V_{DDA} 模拟电压检测器电压等级为 2.9V
20	VAVDEN	V_{DDA}模拟电压检测器使能位 该位由软件置位和清除。 0：失能V_{DDA}模拟电压检测器 1：使能 V_{DDA} 模拟电压检测器
19:18	保留	必须保持复位值。
17:16	DSLPVS	深度睡眠模式电压选择 这些位控制系统睡眠模式时V_{CORE}电压以获得功耗和性能之间的最佳权衡。 00：0.8V 01：0.9V 10：1.0V（默认） 11：1.1V
15:11	LDOVS[4:0]	LDO 输出电压选择 当主锁相环关闭时，这些位由软件设置。当主锁相环使能时，LDOVS 位选择的 LDO 输出电压生效。如果主锁相环闭合，LDO 输出电压低模式选择(该位的值不改变)。 00000~01011：保留 01100：配置 VCORE 电压为 1.1V 01101：保留 01110：配置 VCORE 电压为 1.15V 01111：保留 10000：保留 10001~11111：保留 **注意：**该位只能配置为 1.1V 或 1.15V，禁止配置为保留。
10:9	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

		000：2.15V 001：2.3V 010：2.45V 011：2.6V 100：2.75V 101：2.9V 110：3.0V 111：PA10 外部输入模拟电压 LDO_IN（与 1.2V 进行比较）
4	LVDEN	低电压检测器使能 0：关闭低电压检测器 1：开启低电压检测器
3	STBRST	待机标志复位 0：无影响 1：复位待机标志 读该位，始终返回 0。
2	WURST	唤醒标志复位 0：无影响 1：复位唤醒标志 读该位，始终返回 0。
1	STBMOD	待机模式选择 0：当 Cortex®-M33 进入 SLEEPDEEP 模式时，MCU 进入 Deep-sleep 模式。 1：当 Cortex®-M33 进入 SLEEPDEEP 模式时，MCU 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。

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

该寄存器可以按字（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			WUPEN4	WUPEN3	WUPEN2	WUPEN1	WUPEN0	VUVDF1	VOVDF1	VUVDF0	VOVDF0	VAVDF	LVDF	STBF	WUF
			rw	rw	rw	rw	rw	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:13	保留	必须保持复位值。
12	WUPEN4	WKUP 引脚 4（PC5）唤醒使能 0：关闭 WKUP 引脚 4 唤醒功能。 1：开启 WKUP 引脚 4 唤醒功能。 如果 WUPEN4 在进入省电模式之前置 1，WKUP 引脚 4 的上升沿会将系统从省电模式唤醒。由于 WKUP 引脚 4 为高电平有效，WKUP 引脚 4 内部被配置为输入下拉模式。当在输入已经为高的时候置位该控制位，将会触发一个唤醒事件。
11	WUPEN3	WKUP 引脚 3（PA2）唤醒使能 0：关闭 WKUP 引脚 3 唤醒功能。 1：开启 WKUP 引脚 3 唤醒功能。 如果 WUPEN3 在进入省电模式之前置 1，WKUP 引脚 3 的上升沿会将系统从省电模式唤醒。由于 WKUP 引脚 3 为高电平有效，WKUP 引脚 3 内部被配置为输入下拉模式。当在输入已经为高的时候置位该控制位，将会触发一个唤醒事件。
10	WUPEN2	WKUP 引脚 2（PE6）唤醒使能 0：关闭 WKUP 引脚 2 唤醒功能。 1：开启 WKUP 引脚 2 唤醒功能。 如果 WUPEN2 在进入省电模式之前置 1，WKUP 引脚 2 的上升沿会将系统从省电模式唤醒。由于 WKUP 引脚 2 为高电平有效，WKUP 引脚 2 内部被配置为输入下拉模式。当在输入已经为高的时候置位该控制位，将会触发一个唤醒事件。
9	WUPEN1	WKUP 引脚 1（PC13）唤醒使能 0：关闭 WKUP 引脚 1 唤醒功能。 1：开启 WKUP 引脚 1 唤醒功能。 如果 WUPEN1 在进入省电模式之前置 1，WKUP 引脚 1 的上升沿会将系统从省电模式唤醒。由于 WKUP 引脚 1 为高电平有效，WKUP 引脚 1 内部被配置为输入下拉模式。当在输入已经为高的时候置位该控制位，将会触发一个唤醒事件。
8	WUPEN0	WKUP 引脚 0（PA0）唤醒使能 0：关闭 WKUP 引脚 0 唤醒功能。 1：开启 WKUP 引脚 0 唤醒功能。 如果 WUPEN0 在进入省电模式之前置 1，WKUP 引脚 0 的上升沿会将系统从省电模式唤醒。由于 WKUP 引脚 0 为高电平有效，WKUP 引脚 0 内部被配置为输入下拉模式。当在输入已经为高的时候置位该控制位，将会触发一个唤醒事件。
7	VUVDF1	数字滤波后V_{CORE}低电压检测器标志 位由硬件设置和清除，仅当VUVDEN使能时有效 0：V_{CORE}大于VUVD阈值。 1：V_{CORE} 小于等于 VUVD 阈值。
6	VOVDF1	数字滤波后V_{CORE}过电压检测器标志

位由硬件设置和清除，仅当VOVDEN使能时有效
0：V_{CORE}小于VOVD阈值。
1：V_{CORE} 大于等于 VOVD 阈值。

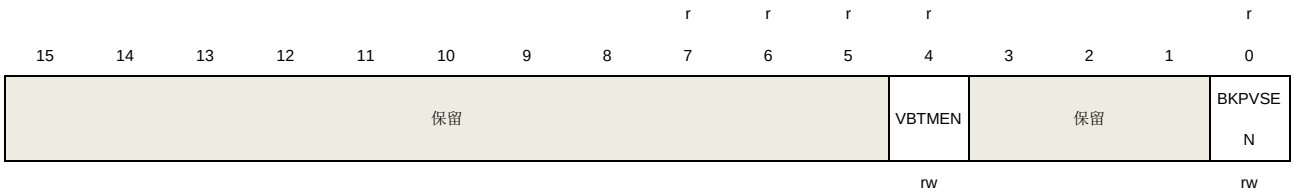
5	VUVDF0	数字滤波后V_{CORE}低电压检测器标志 位由硬件设置和清除，仅当VUVDEN使能时有效 0：V_{CORE}大于VUVD阈值。 1：V_{CORE} 小于等于 VUVD 阈值。
4	VOVDF0	数字滤波后V_{CORE}过电压检测器标志 位由硬件设置和清除，仅当VOVDEN使能时有效 0：V_{CORE}小于VOVD阈值。 1：V_{CORE} 大于等于 VOVD 阈值。
3	VAVDF	V_{DDA}模拟电压检测器输出标志 位由硬件设置和清除，仅当VAVDEN使能时有效 0：V_{DDA}大于等于由VAVDVC位配置的VAVD阈值。 1：V_{DDA} 小于由 VAVDVC 位配置的 VAVD 阈值。
2	LVDF	低电压状态标志 0：低电压事件没出现（V_{DD} 高于设定的 LVD 阈值）。 1：低电压事件出现（V_{DD} 等于或低于 LVD 阈值）。 **注意：**LVD 功能在待机模式被禁用。
1	STBF	待机标志 0：设备没进入过待机模式。 1：设备曾进入过待机模式。 该位只能由 POR / PDR 或通过置位 PMU_CTL0 寄存器的 STBRST 位来清零。
0	WUF	唤醒标志 0：没有收到唤醒事件。 1：唤醒事件由 WKUP 引脚或 RTC 事件包括 RTC 闹钟事件，时间戳事件，侵入事件和自动唤醒事件触发。 该位只能由 POR / PDR 或通过设置 PMU_CTL0 寄存器的 WURST 位来清零。

3.4.3. 控制寄存器 1（PMU_CTL1）

地址偏移：0x08
复位值：0x0000 0000

该寄存器可以按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								TEMPHF	TEMPLF	VBATHF	VBATLF	保留			BKPVSRF



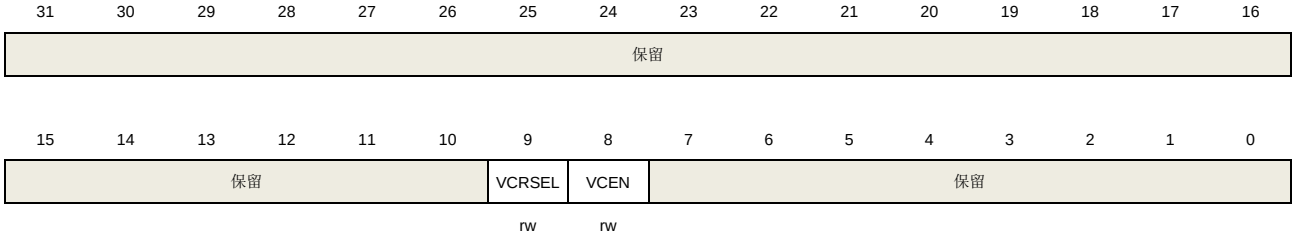
位/位域	名称	描述
31:24	保留	必须保持复位值。
23	TEMPHF	温度监测高阈值标志位 0：温度低于温度监测的高阈值 1：温度等于或高于温度监测的高阈值
22	TEMPLF	温度监测低阈值标志位 0：温度低于温度监测的低阈值 1：温度等于或高于温度监测的低阈值
21	VBATHF	V_{BAT}监测高阈值标志位 0：V_{BAT}电压低于V_{BAT}监测的高阈值 1：V_{BAT} 电压等于或高于 V_{BAT} 监测的高阈值
20	VBATLF	V_{BAT}监测低阈值标志位 0：V_{BAT}电压低于V_{BAT}监测的低阈值 1：V_{BAT} 电压等于或高于 V_{BAT} 监测的低阈值
19:17	保留	必须保持复位值。
16	BKPVSRF	备份域电压稳压器就绪标志位 该位由硬件置位用于指示备份域电压稳压器是否就绪。 0：备份域电压稳压器未就绪 1：备份域电压稳压器已就绪
15:5	保留	必须保持复位值。
4	VBTMEN	V_{BAT} 和温度监测器使能位 当该位置位将使能V_{BAT}供电电压监测和温度监测。 0：失能V_{BAT}和温度监测器 1：使能V_{BAT}和温度监测器
3:1	保留	必须保持复位值。
0	BKPVSEN	BGR 电路使能位 0：失能BGR参考电路 1：使能BGR参考电路

3.4.4. 控制寄存器 2（PMU_CTL2）

地址偏移：0x0C

复位值：0x0000 0000（仅上电复位后复位）

该寄存器可以按字（32位）访问。



位/位域	名称	描述
31:10	保留	必须保持复位值。
9	VCRSEL	V_{BAT} 电池充电电阻的选择 0：5 k 欧姆电阻用于 V_{BAT} 电池充电。 1：1.5 k 欧姆电阻用于 V_{BAT} 电池充电。
8	VCEN	V_{BAT} 电池充电使能 0：禁能 V_{BAT} 电池充电。 1：使能 V_{BAT} 电池充电。
7:0	保留	必须保持复位值。

3.4.5. 控制寄存器 3（PMU_CTL3）

地址偏移：0x18

复位值：0x0000 0000

该寄存器可以按字（32位）访问。

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	VUVDO_DNF	VUVD 模拟输出数字噪声滤波器

		这些位用于配置 VUVD 模拟输出上的数字噪声滤波器，数字滤波器将滤波峰值的长度高达 VUVDO_DNF[7:0] * 1024 * T_{PCLK} 0：关闭数字滤波器 1：开启数字滤波器，滤波峰值长度高达 1024 * T_{PCLK} … 255：开启数字滤波器，滤波峰值长度高达 255 * 1024 * T_{PCLK}
7:0	VOVDO_DNF	VOVD 模拟输出数字噪声滤波器 这些位用于配置 VOVD 模拟输出上的数字噪声滤波器，数字滤波器将滤波峰值的长度高达 VOVDO_DNF[7:0] * 1024 * T_{PCLK} 0：关闭数字滤波器 1：开启数字滤波器，滤波峰值长度高达1024 * T_{PCLK} … 255：开启数字滤波器，滤波峰值长度高达255 * 1024 * T_{PCLK}

4. 复位和时钟单元（RCU）

4.1. 复位控制单元（RCTL）

4.1.1. 简介

GD32G553复位控制包括三种控制方式：电源复位、系统复位和备份域复位。电源复位又称为冷复位，其复位除了备份域的所有系统。系统复位将复位除了SW-DP控制器和备份域之外的其余部分，包括处理器内核和外设IP。备份域复位将复位备份区域。复位能够被外部信号、内部事件和复位发生器触发。后续章节将介绍关于这些复位的详细信息。

4.1.2. 功能描述

电源复位

当发生以下任一事件时，产生电源复位：上电/掉电复位（POR/PDR复位），从待机模式中返回后由内部复位发生器产生。电源复位复位所有的寄存器除了备份域。电源复位为低电平有效，当内部LDO电源基准准备好提供V_{CORE}电压时，电源复位电平将变为无效。复位入口向量被固定在存储器映射的地址0x0000_0004。

系统复位

当发生以下任一事件时，产生一个系统复位：

- 上电复位（POWER_RSTn）
- 外部引脚复位（NRST）
- 窗口看门狗计数终止（WWDGT_RSTn）
- 独立看门狗计数终止（FWDGT_RSTn）
- Cortex®-M33的中断应用和复位控制寄存器中的SYSRESETREQ位置‘1’（SW_RSTn）
- 选项字节重载复位（OBL_RSTn）
- 用户选择字节寄存器nRST_STDBY设置为0，并且进入待机模式时将产生复位（OB_STDBY_RSTn）
- 用户选择字节寄存器nRST_DPSLP设置为0，并且进入深度睡眠模式时（OB_DPSLP_RSTn）

注意：NRST引脚通过配置选项字节NRST_MDSEL[1:0]配置成以下三种模式下：

1. 输入/输出模式（默认模式）：在这种模式下，NRST pin的GPIO功能不可用。复位信号可以从NRST引脚传输到MCU，导致MCU复位，复位脉冲信号可以通过NRST引脚反应出来，最小复位脉冲持续时间为20us。

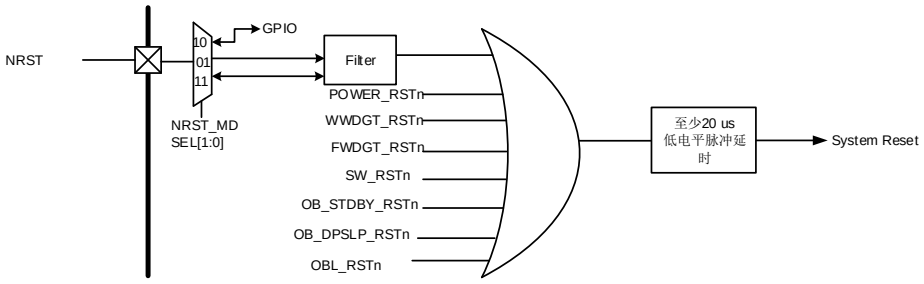
2. 输入模式：在这种模式下，NRST引脚的GPIO功能不可用，复位信号可以从NRST引脚传输到MCU，导致MCU复位，但在NRST引脚上不可见该MCU的内部复位。

3. GPIO模式：NRST引脚只能作为标准GPIO使用，复位功能不可用，复位信号仅在MCU内部，不反映在NRST引脚上。

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

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

图4-1. 系统复位电路



备份域复位

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

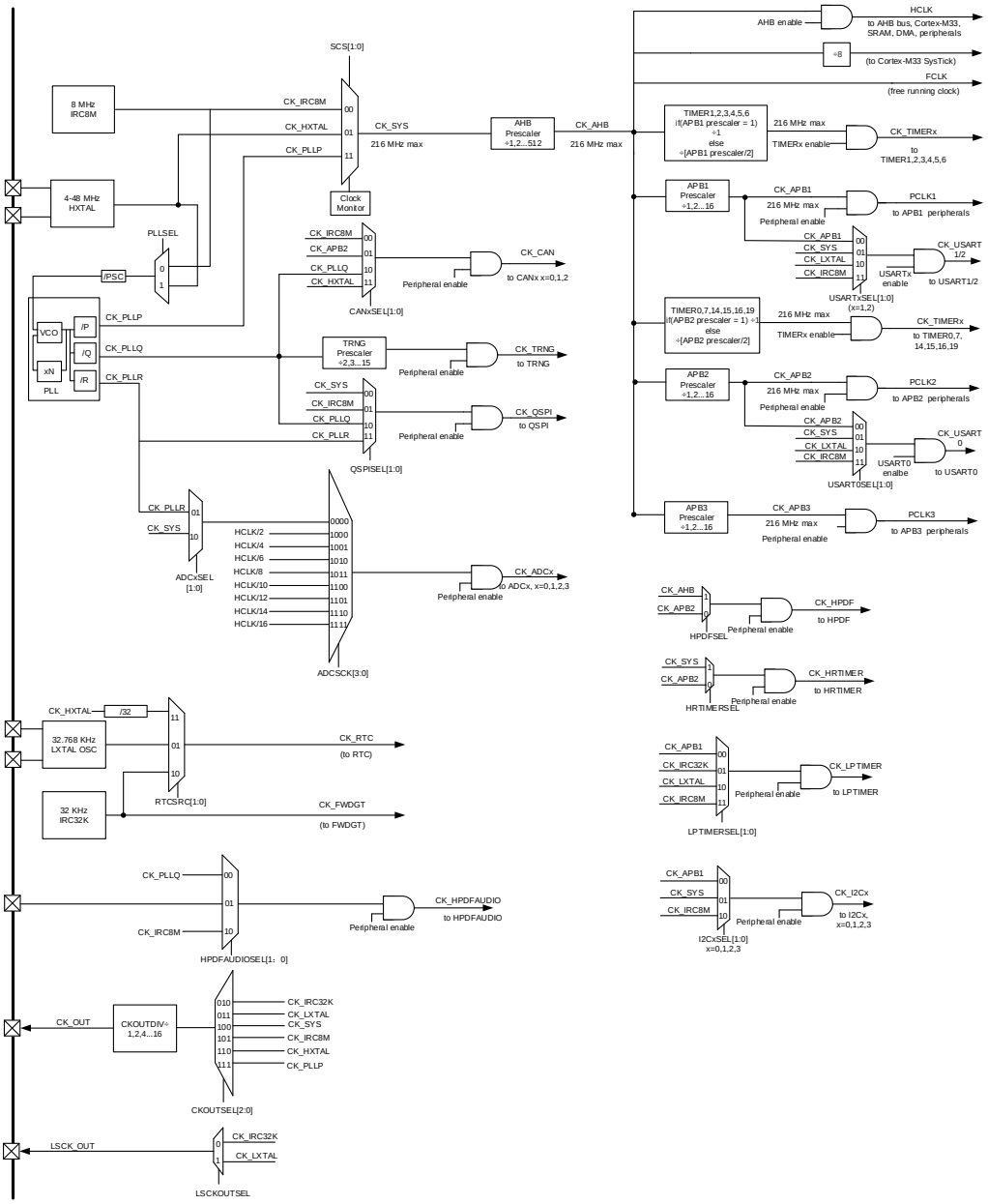
4.2. 时钟控制单元（CCTL）

4.2.1. 简介

时钟控制单元提供了一系列频率的时钟功能，包括一个内部8M RC振荡器时钟（IRC8M）、一个外部高速晶体振荡器时钟（HXTAL）、一个内部32K RC振荡器时钟（IRC32K）、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、一个LXTAL时钟监视器、时钟预分频器、时钟多路复用器和时钟门控电路。

AHB、APB和Cortex®-M33时钟都源自系统时钟（CK_SYS），系统时钟的时钟源可以选择IRC8M、HXTAL或PLL。系统时钟的最大运行时钟频率可以达到216MHz。

图4-2. 时钟树



预分频器可以配置 AHB、APB3、APB2 和 APB1 域的时钟频率。AHB、APB3、APB2、APB1 域的最高时钟频率分别为 216MHz / 216MHz / 216MHz / 216MHz。RCU 通过 AHB 时钟（HCLK）8 分频后作为 Cortex®系统定时器（SysTick）的外部时钟。通过对 SysTick 控制和状态寄存器的设置，可选择上述时钟或 AHB（HCLK）时钟作为 SysTick 时钟。

ADCx（x = 0，1，2，3）的时钟由CKPLLR或者CK_SYS或者HCKL经2、4、6、8、10、12、14、16分频获得，通过配置RCU_CFG2寄存器中的ADCxSEL（x=0，1，2）位和ADC_SYNCCTL寄存

器的ADCSCK位来选择。

CANx（x=0，1，2）的时钟由IRC8M或者HXTAL或者PLLQ或者APB2提供，通过配置RCU_CFG1寄存器中的CANxSEL（x=0，1，2）位来选择。

HPDF_AUDIO的时钟由CK_PLLQ或者CK_IRC8M或者外部HPDF_CKIN引脚提供。通过配置RCU_CFG1寄存器中HPDFAUDIOSEL位来选择。

TRNG的时钟由CK_PLLQ时钟的2到15分频提供。通过配置RCU_CFG2寄存器中TRNGPSC位来选择。

QSPI的时钟由IRC8M或者PLLQ或者PLLR或者系统时钟提供。通过配置RCU_CFG2寄存器的QSPISEL位来选择。

USART0的时钟由IRC8M或者LXTAL时钟或者系统时钟或者APB2时钟提供。通过配置RCU_CFG1寄存器的USART0SEL位来选择。USARTx（x = 1，2）的时钟由IRC8M或者HXTAL或者系统时钟或者APB1时钟提供。通过配置RCU_CFG1寄存器的USARTxSEL（x = 1，2）位来选择。

HPDF的时钟由AHB或者APB2时钟提供。通过配置RCU_CFG1寄存器的HPDFSEL位来选择。

HRTIMER的时钟由CK_APB2或者CK_SYS时钟提供。通过配置RCU_CFG2寄存器的HRTIMERSEL位来选择。

如果用户不需要使用HRTIMER高分辨率模式，可以保持RCU_CFG2寄存器中的HRTIMERSEL位清零，在这种情况下，HRTIMER_MTCTL0寄存器中的CNTCKDIV[2:0]值必须大于或等于5（预分频比大于或等于32）。

如果用户需要使用HRTIMER高分辨率模式，必须在系统时钟源选择为PLL时，通过配置RCU_CFG2寄存器中的HRTIMERSEL位为1，选择CK_SYS为时钟源，此时HRTIMER_MTCTL0寄存器中的CNTCKDIV[2:0]值可以配置为任何可选值。

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

LPTIMER的时钟由CK_APB1或者CK_IRC32K或者CK_LXTAL或者IRC8M时钟提供。通过配置RCU_CFG2寄存器的LPTIMERSEL位来选择。

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

通过配置RCU_BDCTL寄存器的RTCSRC位，RTC时钟可以选择由LXTAL时钟、IRC32K时钟或HXTAL时钟的32分频提供。RTC时钟选择HXTAL时钟的32分频做为时钟源后，当V_{CORE}电源域掉电时，时钟将停止。RTC时钟选择IRC32K时钟做为时钟源后，当V_{DD}掉电时，时钟将停止。RTC时钟选择LXTAL时钟做为时钟源后，当V_{DD}和V_{BAT}都掉电时，时钟将停止。

当FWDGT启动时，FWDGT时钟被强制选择由IRC32K时钟做为时钟源。

4.2.2. 主要特性

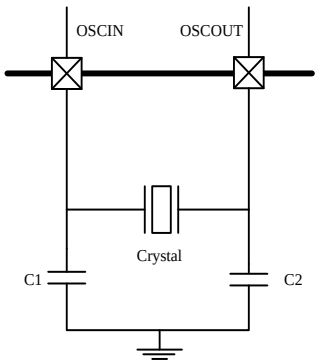
- 4到48MHz外部高速晶体振荡器（HXTAL）；
- 内部8MHz RC振荡器（IRC8M）；
- 32768 Hz外部低速晶体振荡器（LXTAL）；
- 内部32KHz RC振荡器（IRC32K）；
- PLL时钟源可选HXTAL、IRC8M；
- HXTAL时钟监视器;
- LXTAL时钟监视器。

4.2.3. 功能描述

外部高速晶体振荡时钟（HXTAL）

4到48M的外部高速晶体振荡器可为系统时钟提供更为精确时钟源。带有特定频率的晶体必须靠近两个HXTAL的引脚连接。和晶体连接的外部电阻和电容必须根据所选择的振荡器来调整。

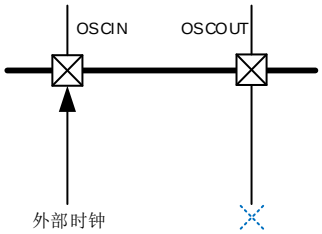
图4-3. HXTAL时钟源



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

将控制寄存器RCU_CTL的HXTALBPS和HXTALEN位置‘1’可以设置外部时钟旁路模式。旁路输入时，信号接至OSCIN，OSCOUT保持悬空状态，如*图4-4. 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4-4. 旁路模式下HXTAL时钟源



内部 8M RC 振荡器时钟（IRC8M）

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

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

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

锁相环（PLL）

内部有一个锁相环，PLL，可以提供16 ~ 216 MHz时钟的输出，基本参考频率2 ~ 40 MHz的2 ~ 31倍。

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

通过配置PLLPEN、PLLQEN、PLLREN来使能所需的PLL输出。PLLP时钟可用于产生系统时钟(不超过216MHz)，PLLQ时钟可以提供给TRNG / QSPI / CAN外设。PLLR时钟可以提供给到ADC外设。每个PLL输出时钟(PLLPEN,PLLPEN,PLLREN)的使能可以在不停止对应的PLL的情况下修改。如果CK_PLLP被用作系统时钟,那么PLLPEN不能失能。

当进入Deepsleep/Standby模式或者HXTAL监视器检测到时钟阻塞时（HXTAL做为锁相环的输入时钟），PLL将被关闭。

外部低速晶体振荡器时钟（LXTAL）

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

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

内部 32K RC 振荡器时钟（IRC32K）

IRC32K内部RC振荡器时钟担当一个低功耗时钟源的角色，不需要外部器件，它的时钟频率大约32kHz，为独立看门狗和实时时钟电路提供时钟。IRC32K RC振荡器可以通过设置复位源/时钟寄存器RCU_RSTSCK中的IRC32KEN位被启动和关闭。复位源/时钟寄存器RCU_RSTSCK中的IRC32KSTB位用来指示IRC32K时钟是否已稳定。如果复位源/时钟寄存器RCU_RSTSCK中的相应中断使能位IRC32KSTBIE被置‘1’，在IRC32K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时钟默认做为CK_SYS的时钟源，改变配置寄存器0（RCU_CFG0）中的系统时钟变换位SCS可以切换系统时钟源为HXTAL或CK_PLL。当SCS的值被改变，系统时钟将使用原来的时钟源继续运行直到转换的目标时钟源稳定。当一个时钟源被直接或通过PLL间接作为系统时钟时，它将不能被停止。

HXTAL 时钟监视器（CKM）

设置控制寄存器RCU_CTL中的HXTAL时钟监视使能位CKMEN，HXTAL可以使能时钟监视功能。该功能必须在HXTAL启动延迟完毕后使能，在HXTAL停止后禁止。一旦监测到HXTAL故障，HXTAL将自动被禁止，中断寄存器RCU_INT中的HXTAL时钟阻塞中断标志位CKMIF将被置‘1’，产生HXTAL故障事件。这个故障引发的中断和Cortex®-M33的不可屏蔽中断NMI相连。如果HXTAL被选作系统，PLL或是RTC的时钟源，HXTAL故障将促使选择IRC8M为系统时钟源，PLL将被自动禁止，RTC的时钟源需要重新配置。

LXTAL 时钟监视器（LCKM）

设置时钟控制寄存器 RCU_CTL 中的 LXTAL 时钟监视使能位 LCKMEN，LXTAL 可以使能时钟监视功能。该功能必须在 LXTAL 启动延迟完毕和 IRC32K 使能后使能。
当 LCKMEN 启用时，一个 4 位加一计数器将在 IRC32K 域工作。如果 LXTAL 时钟卡在 0/1 错误或时钟减慢约 20KHz，计数器将溢出。将发现 LXTAL 时钟故障。

时钟输出功能

时钟输出功能输出从32KHz到216MHz的时钟。通过设置时钟配置寄存器0（RCU_CFG0）中的CK_OUT时钟源选择位域CKOUTSEL能够选择不同的时钟信号。相应的GPIO引脚应该被配置成备用功能I/O（AFIO）模式来输出选择的时钟信号。

表4-1. 时钟输出的时钟源选择

时钟输出的时钟源选择位域	时钟源
000	NO CLK
010	CK_IRC32K
011	CK_LXTAL
100	CK_SYS
101	CK_IRC8M
110	CK_HXTAL
111	CK_PLLP

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。

通过设置 RCU_BDCTL 寄存器的 LSCKOUSEL 位，CK_LXTAL 和 CK_IRC32K 时钟可以通过 LSCK_OUT 引脚输出，即使在深度睡眠模式和待机模式。

表4-2. 低速时钟输出的时钟源选择

时钟输出的时钟源选择位域	时钟源
0	CK_IRC32K
1	LXTAL

深度睡眠模式时钟控制

当MCU处于深度睡眠模式时，USART0 / 1 / 2外设时钟由LXTA提供且LXTAL时钟使能时，则USART0 / 1 / 2外设可以唤醒MCU。

如果USART0 / 1 / 2时钟选择IRC8M处于深度睡眠模式时，则它们能够打开IRC8M时钟或关闭IRC8M时钟，从而使USART0 / 1 / 2从深度睡眠模式唤醒。

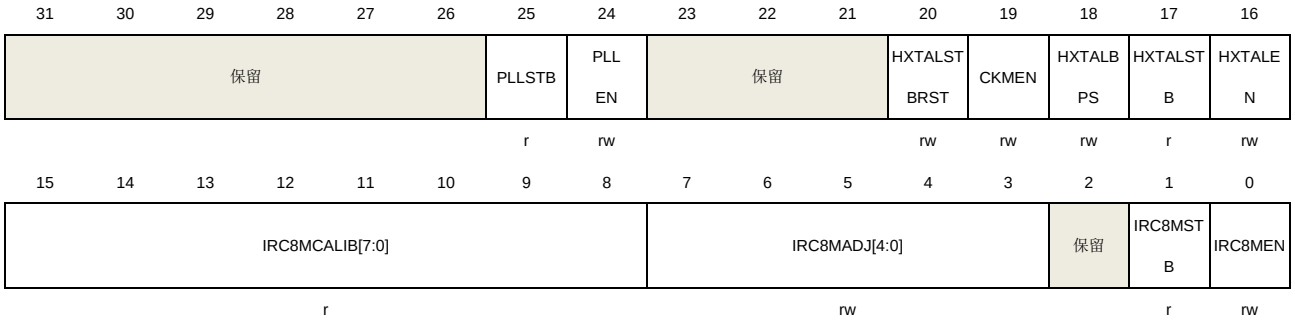
4.3. RCU 寄存器

RCU 基地址：0x4002 1000

4.3.1. 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。



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 来表示 PLL 输出时钟是否稳定待用 0：PLL 未稳定 1：PLL 已稳定
24	PLLEN	PLL 使能 软件置位或复位，当 PLL 时钟做为系统时钟时该位不能被复位。当进入深度睡眠或待机模式时由硬件复位 0：PLL 被关闭 1：PLL 被打开
23:21	保留	必须保持复位值。
20	HXTALSTBRST	HXTAL 时钟稳定标志位复位 由软件置位或复位 0：HXTAL 时钟稳定标志位不复位 1：HXTAL 时钟稳定标志位复位
19	CKMEN	HXTAL 时钟监视器使能 0：禁止高速 4 ~ 48 MHz 晶体振荡器（HXTAL）时钟监视器

		1：使能高速 4 ~ 48 MHz 晶体振荡器（HXTAL）时钟监视器 当硬件检测到 HXTAL 时钟被阻塞在低或高状态时，内部硬件自动切换系统时钟到 IRC8M 时钟。恢复原来系统时钟的方式有以下几种：外部复位，上电复位，软件清 CKMIF 位。 **注意：**使能 HXTAL 时钟监视器以后，硬件无视控制位 IRC8MEN 的状态，自动使能 IRC8M 时钟。
18	HXTALBPS	高速晶体振荡器（HXTAL）时钟旁路模式使能 只有在 HXTALEN 位为 0 时 HXTALBPS 位才可写 0：禁止 HXTAL 旁路模式 1：使能 HXTAL 旁路模式 HXTAL 输出时钟等于输入时钟
17	HXTALSTB	高速晶体振荡器（HXTAL）时钟稳定标志位 硬件置‘1’来指示 HXTAL 振荡器时钟是否稳定待用 0：HXTAL 振荡器未稳定 1：HXTAL 振荡器已稳定
16	HXTALEN	高速晶体振荡器（HXTAL）使能 软件置位或复位，如果 HXTAL 时钟作为系统时钟或者当 PLL 时钟做为系统时钟时，其做为 PLL 的输入时钟，该位不能被复位。进入深度睡眠或待机模式时硬件自动复位 0：高速 4 ~ 48 MHz 晶体振荡器被关闭 1：高速 4 ~ 48 MHz 晶体振荡器被打开
15:8	IRC8MCALIB[7:0]	内部 8MHz RC 振荡器校准值寄存器 上电时自动加载这些位
7:3	IRC8MADJ[4:0]	内部 8MHz RC 振荡器时钟调整值 这些位由软件置位，最终调整值为 IRC8MADJ[4:0]位域的当前值加上 IRC8MCALIB[7:0]位域的值。最终调整值应该调整 IRC8M 到 8MHz ± 1%
2	保留	必须保持复位值。
1	IRC8MSTB	IRC8M 内部 8MHz RC 振荡器稳定标志位 硬件置‘1’来指示 IRC8M 振荡器时钟是否稳定待用 0：IRC8M 振荡器未稳定 1：IRC8M 振荡器已稳定
0	IRC8MEN	内部 8MHz RC 振荡器使能 软件置位或复位，如果 IRC8M 时钟做为系统时钟时，该位不能被复位。当从深度睡眠或待机模式返回，或当 CKMEN 置位同时用作系统时钟的 HXTAL 振荡器发生故障时，该位由硬件置 1 来启动 IRC8M 振荡器。 0：内部 8MHz RC 振荡器被关闭 1：内部 8MHz RC 振荡器被打开

4.3.2. PLL 寄存器（RCU_PLL）

地址偏移：0x04

复位值：0x0000 0400

配置PLL时钟可参考下列公式：

CK_PLLVCOSRC = CK_PLLSRC / PLLPSC

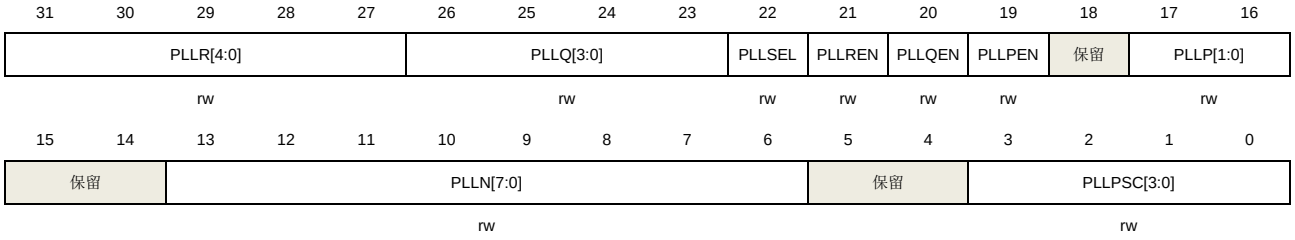
CK_PLLVCO = CK_PLLVCOSRC × PLLN

CK_PLLP = CK_PLLVCO / PLLP

CK_PLLQ = CK_PLLVCO / PLLQ

CK_PLLR = CK_PLLVCO / PLLR

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域	名称	描述
31:27	PLLR[4:0]	PLLR 输出频率的分频系数（PLL VCO 时钟做为输入） 当 PLL 被关闭时由软件置位或清零。这些位域用做将 PLL VCO 时钟（CK_PLLVCO）分频生成 PLLQ 输出时钟（CK_PLLQ）。RCU_PLL 寄存器的 PLLN 位域对 CK_PLLVCO 时钟进行了描述。 00000：保留 00001：保留 00010：CK_PLLQ = CK_PLLVCO / 2 00011：CK_PLLQ = CK_PLLVCO / 3 00100：CK_PLLQ = CK_PLLVCO / 4 … 11111：CK_PLLQ = CK_PLLVCO / 31
26:23	PLLQ[3:0]	PLLQ 输出频率的分频系数（PLL VCO 时钟做为输入） 当 PLL 被关闭时由软件置位或清零。这些位域用做将 PLL VCO 时钟（CK_PLLVCO）分频生成 PLLQ 输出时钟（CK_PLLQ）。CK_PLLQ 时钟可以被用作 TRNG（48MHz）模块的时钟源。RCU_PLL 寄存器的 PLLN 位域对 CK_PLLVCO 时钟进行了描述。 0000：保留 0001：保留 0010：CK_PLLQ = CK_PLLVCO / 2 0011：CK_PLLQ = CK_PLLVCO / 3 0100：CK_PLLQ = CK_PLLVCO / 4

		… 1111：CK_PLLQ = CK_PLLVCO / 15
22	PLLSEL	PLL时钟源选择 由软件置位或复位，控制PLL时钟源 0：IRC8M时钟被选择为PLL时钟的时钟源 1：HXTAL时钟被选择为PLL时钟的时钟源
21	PLLREN	PLLR 分频器输出使能 由软件置位或复位。只有在PLLEN位为0时PLLREN位才可写。 0：禁止CK_PLLR输出 1：使能CK_PLLR输出
20	PLLQEN	PLLQ 分频器输出使能 由软件置位或复位。只有在PLLEN位为0时PLLQEN位才可写。 0：禁止CK_PLLQ输出 1：使能CK_PLLQ输出
19	PLLPEN	PLLP 分频器输出使能 由软件置位或复位。只有在PLLEN位为0时PLLPEN位才可写。 0：禁止CK_PLLP输出 1：使能CK_PLLP输出
18	保留	必须保持复位值。
17:16	PLLP[1:0]	PLLP 输出频率分频系数（PLL VCO 时钟做为输入） 当 PLL 被关闭时由软件置位或清零。这些位域用做将 PLL VCO 时钟（CK_PLLVCO）分频生成 PLLP 输出时钟（CK_PLLP）。CK_PLLP 时钟可以被用作系统时钟（不超过 216MHz）。RCU_PLL 寄存器的 PLLN 位域对 CK_PLLVCO 时钟进行了描述。 00：CK_PLLP = CK_PLLVCO / 2 01：CK_PLLP = CK_PLLVCO / 4 10：CK_PLLP = CK_PLLVCO / 6 11：CK_PLLP = CK_PLLVCO / 8
15:14	保留	必须保持复位值。
13:6	PLLN[8:0]	PLL VCO 时钟倍频因子 当 PLL 被关闭时由软件置位或清零（仅支持全字/半字写操作）。这些位域用做将 PLL VCO 源时钟（CK_PLLVCOSRC）倍频生成 PLL VCO 输出时钟（CK_PLLVCO）。RCU_PLL 寄存器的 PLLPSC 位域对 CK_PLLVCOSRC 时钟进行了描述。 注意：CK_PLLVCO 时钟频率范围必须在 96MHz 到 480MHz 之间 PLLN 的值必须满足：8≤PLLN≤180 000000000：保留 000000001：保留

		… 000111111：保留 0001000: PLLN = 8 0001001: PLLN = 9 0001010: PLLN = 10 … 10110100: PLLN = 180
5:4	保留	必须保持复位值。
3:0	PLLPSC[5:0]	PLL VCO 源时钟分频器 当 PLL 被关闭时由软件置位或清零。这些位域用做将 PLL 源时钟（CK_PLLSRC）分频生成 PLL VCO 源时钟（CK_PLLVCOSRC）。RCU_PLL 寄存器的 PLLSEL 位对 CK_PLLSRC 时钟进行了描述。 VCO 源时钟频率范围必须在 2MHz 到 16MHz 之间 0000: CK_PLLSRC 0001: CK_PLLSRC / 2 0010: CK_PLLSRC / 3 0011: CK_PLLSRC / 4 … 1111: CK_PLLSRC / 16

4.3.3. 时钟配置寄存器 0（RCU_CFG0）

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		APB3PSC[2:0]			CKOUTDIV[2:0]			CKOUTSEL[2:0]			保留				
		rw			rw			rw							

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

位/位域	名称	描述
31:30	保留	必须保持复位值。
29:27	APB3PSC[2:0]	APB3 预分频选择 由软件置位或清零，控制 APB3 时钟分频因子 0xx：CK_APB3 = CK_AHB 100：CK_APB3 = CK_AHB / 2

		101：CK_APB3 = CK_AHB / 4 110：CK_APB3 = CK_AHB / 8 111：CK_APB3 = CK_AHB / 16
26:24	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 23:21 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
23:21	CKOUTSEL[2:0]	CKOUT 时钟源选择 由软件置位或清零 000：无时钟输出 001：保留 010：选择 CK_IRC32K 时钟 011：选择低速晶体振荡器时钟 （LXTAL） 100：选择系统时钟 CK_SYS 101：选择内部 8M RC 振荡器时钟 110：选择高速晶体振荡器时钟 （HXTAL） 111：选择 CK_PLLP 时钟
20:16	保留	必须保持复位值。
15:13	APB2PSC[2:0]	APB2 预分频选择 由软件置位或清零，控制 APB2 时钟分频因子 0xx：CK_APB2 = CK_AHB 100：CK_APB2 = CK_AHB / 2 101：CK_APB2 = CK_AHB / 4 110：CK_APB2 = CK_AHB / 8 111：CK_APB2 = CK_AHB / 16
12:10	APB1PSC[2:0]	APB1 预分频选择 由软件置位或清零，控制 APB1 时钟分频因子. 0xx：CK_APB1 = CK_AHB 100：CK_APB1 = CK_AHB / 2 101：CK_APB1 = CK_AHB / 4 110：CK_APB1 = CK_AHB / 8 111：CK_APB1 = CK_AHB / 16
9:8	保留	必须保持复位值。
7:4	AHBPSC[3:0]	AHB 预分频选择

由软件置位或清零，控制 AHB 时钟分频因子.
0xxx：CK_AHB = CK_SYS
1000：CK_AHB = CK_SYS / 2
1001：CK_AHB = CK_SYS / 4
1010：CK_AHB = CK_SYS / 8
1011：CK_AHB = CK_SYS / 16
1100：CK_AHB = CK_SYS / 64
1101：CK_AHB = CK_SYS / 128
1110：CK_AHB = CK_SYS / 256
1111：CK_AHB = CK_SYS / 512

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

4.3.4. 时钟中断寄存器（RCU_INT）

地址偏移：0x0C
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			LCKMIC	LCKMIF	保留			CKMIC	保留		PLL STBIC	HXTAL STBIC	IRC8M STBIC	LXTAL STBIC	IRC32K STBIC
			w	r				w			w	w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			PLL STBIE	HXTAL STBIE	IRC8M STBIE	LXTAL STBIE	IRC32K STBIE	CKMIF	保留		PLL STBIF	HXTAL STBIF	IRC8M STBIF	LXTAL STBIF	IRC32K STBIF

rw rw rw rw rw r r r r r r

位/位域	名称	描述
31:29	保留	必须保持复位值。
28	LCKMIC	LXTAL 时钟阻塞中断清零 软件写 1 复位 LCKMIF 标志位 0：不复位 LCKMIF 标志位 1：复位 LCKMIF 标志位
27	LCKMIF	LXTAL 时钟阻塞中断标志位 当 LXTAL 时钟被阻塞时由硬件置位 软件置位 LCKMIC 位时清除该位 0：时钟正常运行 1：LXTAL 时钟阻塞
26:24	保留	必须保持复位值。
23	CKMIC	HXTAL 时钟阻塞中断清零 软件写 1 复位 CKMIF 标志位. 0：不复位 CKMIF 标志位 1：复位 CKMIF 标志位
22:21	保留	必须保持复位值。
20	PLLSTBIC	PLL 时钟稳定中断清零 软件写 1 复位 PLLSTBIF 标志位 0：不复位 PLLSTBIF 标志位 1：复位 PLLSTBIF 标志位
19	HXTALSTBIC	HXTAL 时钟稳定中断清零 软件写 1 复位 HXTALSTBIF 标志位 0：不复位 HXTALSTBIF 标志位 1：复位 HXTALSTBIF 标志位
18	IRC8MSTBIC	IRC8M 时钟稳定中断清零 软件写 1 复位 IRC8MSTBIF 标志位 0：不复位 IRC8MSTBIF 标志位 1：复位 IRC8MSTBIF 标志位
17	LXTALSTBIC	LXTAL 时钟稳定中断清零 软件写 1 复位 LXTALSTBIF 标志位 0：不复位 LXTALSTBIF 标志位 1：复位 LXTALSTBIF 标志位

16	IRC32KSTBIC	IRC32K 时钟稳定中断清零 软件写 1 复位 IRC32KSTBIF 标志位 0：不复位 IRC32KSTBIF 标志位 1：复位 IRC32KSTBIF 标志位
15:13	保留	必须保持复位值。
12	PLLSTBIE	PLL 时钟稳定中断使能 软件置位和复位来使能/禁止 PLL 时钟稳定中断 0：禁止 PLL 时钟稳定中断 1：使能 PLL 时钟稳定中断
11	HXTALSTBIE	HXTAL 时钟稳定中断使能 软件置位和复位来使能/禁止 HXTAL 时钟稳定中断 0：禁止 HXTAL 时钟稳定中断 1：使能 HXTAL 时钟稳定中断
10	IRC8MSTBIE	IRC8M 时钟稳定中断使能 软件置位和复位来使能/禁止 IRC8M 时钟稳定中断 0：禁止 IRC8M 时钟稳定中断 1：使能 IRC8M 时钟稳定中断
9	LXTALSTBIE	LXTAL 时钟稳定中断使能 软件置位和复位来使能/禁止 LXTAL 时钟稳定中断 0：禁止 LXTAL 时钟稳定中断 1：使能 LXTAL 时钟稳定中断
8	IRC32KSTBIE	IRC32K 时钟稳定中断使能 软件置位和复位来使能/禁止 IRC32K 时钟稳定中断 0：禁止 IRC32K 时钟稳定中断 1：使能 IRC32K 时钟稳定中断
7	CKMIF	HXTAL 时钟阻塞中断标志位 当 HXTAL 时钟被阻塞时由硬件置位. 软件置位 CKMIC 位时清除该位 0：时钟正常运行 1：HXTAL 时钟阻塞
6:5	保留	必须保持复位值。
4	PLLSTBIF	PLL 时钟稳定中断标志位 当 PLL 时钟稳定且 PLLSTBIE 位被置 1 时由硬件置 1 软件置位 PLLSTBIC 位时清除该位 0：无 PLL 时钟稳定中断产生

1：产生 PLL 时钟稳定中断

3	HXTALSTBIF	HXTAL 时钟稳定中断标志位 当高速 4~32 MHz 晶体振荡器时钟稳定且 HXTALSTBIE 位被置 1 时由硬件置 1 软件置位 HXTALSTBIC 位时清除该位 0：无 HXTAL 时钟稳定中断产生 1：产生 HXTAL 时钟稳定中断
2	IRC8MSTBIF	IRC8M 时钟稳定中断标志位 当内部 8MHz RC 振荡器时钟稳定且 IRC8MSTBIE 位被置 1 时由硬件置 1 软件置位 IRC8MSTBIC 位时清除该位 0：无 IRC8M 时钟稳定中断产生 1：产生 IRC8M 时钟稳定中断
1	LXTALSTBIF	LXTAL 时钟稳定中断标志位 当低速晶体振荡器时钟稳定且 LXTALSTBIE 位被置 1 时由硬件置 1 软件置位 LXTALSTBIC 位时清除该位 0：无 LXTAL 时钟稳定中断产生 1：产生 LXTAL 时钟稳定中断
0	IRC32KSTBIF	IRC32K 时钟稳定中断标志位 当内部 32kHz RC 振荡器时钟稳定且 IRC32KSTBIE 位被置 1 时由硬件置 1 软件置位 IRC32KSTBIC 位时清除该位 0：无 IRC32K 时钟稳定中断产生 1：产生 IRC32K 时钟稳定中断

4.3.5. AHB1 复位寄存器（RCU_AHB1RST）

地址偏移：0x010

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
FFTRST	保留							DMAMUXRST	DMA1RST	DMA0RST	保留				
rw								rw	rw	rw					

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

位/位域	名称	描述
31	FFTRST	FFT 复位

		由软件置位或复位 0：无作用 1：复位 FFT
30:24	保留	必须保持复位值。
23	DMAMUXRST	DMAMUX 复位 由软件置位或复位 0：无作用 1：复位 DMAMUX
22	DMA1RST	DMA1 复位 由软件置位或复位 0：无作用 1：复位 DMA1
21	DMA0RST	DMA0 复位 由软件置位或复位 0：无作用 1：复位 DMA0
20:16	保留	必须保持复位值。
15	CLARST	CLA 复位 由软件置位或复位 0：无作用 1：复位 CLA
14:13	保留	必须保持复位值。
12	CRCRST	CRC 复位 由软件置位或复位 0：无作用 1：复位 CRC
11:0	保留	必须保持复位值。

4.3.6. AHB2 复位寄存器（RCU_AHB2RST）

地址偏移：0x014
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								PGRST	PFRST	PERST	PDRST	PCRST	PBRST	PARST	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								TMURST	TRNGRST	保留		CAURST	保留	FACRST	保留
								rw	rw			rw		rw	

位/位域	名称	描述
31:24	保留	必须保持复位值。
23	PGRST	GPIO 端口 G 复位 由软件置位或复位 0：无作用 1：复位 GPIO 端口 G
22	PFRST	GPIO 端口 F 复位 由软件置位或复位 0：无作用 1：复位 GPIO 端口 F
21	PERST	GPIO 端口 E 复位 由软件置位或复位 0：无作用 1：复位 GPIO 端口 E
20	PDRST	GPIO 端口 D 复位 由软件置位或复位 0：无作用 1：复位 GPIO 端口 D
19	PCRST	GPIO 端口 C 复位 由软件置位或复位 0：无作用 1：复位 GPIO 端口 C
18	PBRST	GPIO 端口 B 复位 由软件置位或复位 0：无作用 1：复位 GPIO 端口 B
17	PARST	GPIO 端口 A 复位 由软件置位或复位 0：无作用 1：复位 GPIO 端口 A

16:8	保留	必须保持复位值。
7	TMURST	TMU 复位 由软件置位或复位 0：无作用 1：复位 TMU
6	TRNGRST	TRNG 复位 由软件置位或复位 0：无作用 1：复位 TRNG
5:4	保留	必须保持复位值。
3	CAURST	CAU 复位 由软件置位或复位 0：无作用 1：复位 CAU
2	保留	必须保持复位值。
1	FACRST	FAC 复位 由软件置位或复位 0：无作用 1：复位 FAC
0	保留	必须保持复位值。

4.3.7. AHB3 复位寄存器（RCU_AHB3RST）

地址偏移：0x018

复位值：0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

位/位域	名称	描述

31:2	保留	必须保持复位值。
1	QSPIRST	QSPI 复位 由软件置位或复位 0：无作用 1：复位 QSPI
0	EXMCRST	EXMC 复位 由软件置位或复位 0：无作用 1：复位 EXMC

4.3.8. APB1 复位寄存器（RCU_APB1RST）

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
保留			PMURST	保留			I2C3RST	I2C2RST	I2C1RST	I2C0RST	UART4RST	UART3RST	USART2RST	USART1RST	保留
			rw				rw	rw	rw	rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI2RST	SPI1RST	保留		WWDGTRST	保留	LPTIMERRST	保留			TIMER6RST	TIMER5RST	TIMER4RST	TIMER3RST	TIMER2RST	TIMER1RST
rw	rw			rw		rw				rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:29	保留	必须保持复位值。
28	PMURST	PMU 复位 由软件置位或复位 0：无作用 1：复位 PMU
27:25	保留	必须保持复位值。
24	I2C3RST	I2C3 复位 由软件置位或复位 0：无作用 1：复位 I2C3
23	I2C2RST	I2C2 复位

由软件置位或复位
0：无作用
1：复位 I2C2

22 I2C1RST I2C1 复位
由软件置位或复位
0：无作用
1：复位 I2C1

21 I2C0RST I2C0 复位
由软件置位或复位
0：无作用
1：复位 I2C0

20 UART4RST UART4 复位
由软件置位或复位
0：无作用
1：复位 UART4

19 UART3RST UART3 复位
由软件置位或复位
0：无作用
1：复位 UART3

18 USART2RST USART2 复位
由软件置位或复位
0：无作用
1：复位 USART2

17 USART1RST USART1 复位
由软件置位或复位
0：无作用
1：复位 USART1

16 保留 必须保持复位值。

15 SPI2RST SPI2 复位
由软件置位或复位
0：无作用
1：复位 SPI2

14 SPI1RST SPI1 复位
由软件置位或复位
0：无作用

		1：复位 SPI1
13:12	保留	必须保持复位值。
11	WWDGTRST	WWDGT 复位 由软件置位或复位 0：无作用 1：复位 WWDGT
10	保留	必须保持复位值。
9	LPTIMERRST	LPTIMER 复位 由软件置位或复位 0：无作用 1：复位 LPTIMER
8:6	保留	必须保持复位值。
5	TIMER6RST	TIMER6 复位 由软件置位或复位 0：无作用 1：复位 TIMER6
4	TIMER5RST	TIMER5 复位 由软件置位或复位 0：无作用 1：复位 TIMER5
3	TIMER4RST	TIMER4 复位 由软件置位或复位 0：无作用 1：复位 TIMER4
2	TIMER3RST	TIMER3 复位 由软件置位或复位 0：无作用 1：复位 TIMER3
1	TIMER2RST	TIMER2 复位 由软件置位或复位 0：无作用 1：复位 TIMER2
0	TIMER1RST	TIMER1 复位 由软件置位或复位 0：无作用

1：复位 TIMER1

4.3.9. APB2 复位寄存器（RCU_APB2RST）

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
TRIGSEL RST	保留	HRTIME RRST	保留									HPDFRST	TIMER16 RST	TIMER15 RST	TIMER14 RST
rw		rw										rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER19 RST	SYSCFG RST	保留	SPI0RST	保留	CAN2RST	CAN1RST	CAN0RST	保留			USART0 RST	CMPRST	VREFRST	TIMER7RST	TIMER0RST
rw	rw		rw		rw	rw	rw				rw	rw	rw	rw	rw

位/位域	名称	描述
31	TRIGSELRST	TRIGSEL 复位 由软件置位或复位 0：无作用 1：复位 TRIGSEL
30	保留	必须保持复位值。
29	HRTIMEREN	HRTIMER 复位 由软件置位或复位 0：无作用 1：复位 HRTIMER
28:20	保留	必须保持复位值。
19	HPDFRST	HPDF 复位 由软件置位或复位 0：无作用 1：复位 HPDF
18	TIMER16RST	TIMER16 复位 由软件置位或复位 0：无作用 1：复位 TIMER16
17	TIMER15RST	TIMER15 复位

		由软件置位或复位 0：无作用 1：复位 TIMER15
16	TIMER14RST	TIMER14 复位 由软件置位或复位 0：无作用 1：复位 TIMER14
15	TIMER19RST	TIMER19 复位 由软件置位或复位 0：无作用 1：复位 TIMER19
14	SYSCFGRST	SYSCFG 复位 由软件置位或复位 0：无作用 1：复位 SYSCFG
13	保留	必须保持复位值。
12	SPI0RST	SPI0 复位 由软件置位或复位 0：无作用 1：复位 SPI0
11	保留	必须保持复位值。
10	CAN2RST	CAN2 复位 由软件置位或复位 0：无作用 1：复位 CAN2
9	CAN1RST	CAN1 复位 由软件置位或复位 0：无作用 1：复位 CAN1
8	CAN0RST	CAN0 复位 由软件置位或复位 0：无作用 1：复位 CAN0
7:5	保留	必须保持复位值。
4	USART0RST	USART0 复位

由软件置位或复位
0：无作用
1：复位 USART0

3	CMPRST	CMP 复位 由软件置位或复位 0：无作用 1：复位 CMP
2	VREFRST	VREFR 复位 由软件置位或复位 0：无作用 1：复位 VREFR
1	TIMER7RST	TIMER7 复位 由软件置位或复位 0：无作用 1：复位 TIMER7
0	TIMER0RST	TIMER0 复位 由软件置位或复位 0：无作用 1：复位 TIMER0

4.3.10. APB3 复位寄存器（RCU_APB3RST）

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											DAC3RST	DAC2RST	DAC1RST	DAC0RST	DACHOLDRST
											rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				ADC3RST	ADC2RST	ADC1RST	ADC0RST	保留							
				rw	rw	rw	rw								

位/位域	名称	描述
31:21	保留	必须保持复位值。
20	DAC3RST	DAC3 复位

		由软件置位或复位 0：无作用 1：复位 DAC3
19	DAC2RST	DAC2 复位 由软件置位或复位 0：无作用 1：复位 DAC2
18	DAC1RST	DAC1 复位 由软件置位或复位 0：无作用 1：复位 DAC1
17	DAC0RST	DAC0 复位 由软件置位或复位 0：无作用 1：复位 DAC0
16	DACHOLDRST	DAC 保持时钟复位 由软件置位或复位，DAC 保持时钟源为 IRC32K 0：无作用 1：复位 DAC 保持时钟
15:12	保留	必须保持复位值。
11	ADC3RST	ADC3 复位 由软件置位或复位 0：无作用 1：复位 ADC3
10	ADC2RST	ADC2 复位 由软件置位或复位 0：无作用 1：复位 ADC2
9	ADC1RST	ADC1 复位 由软件置位或复位 0：无作用 1：复位 ADC1
8	ADC0RST	ADC0 复位 由软件置位或复位 0：无作用

1：复位 ADC0

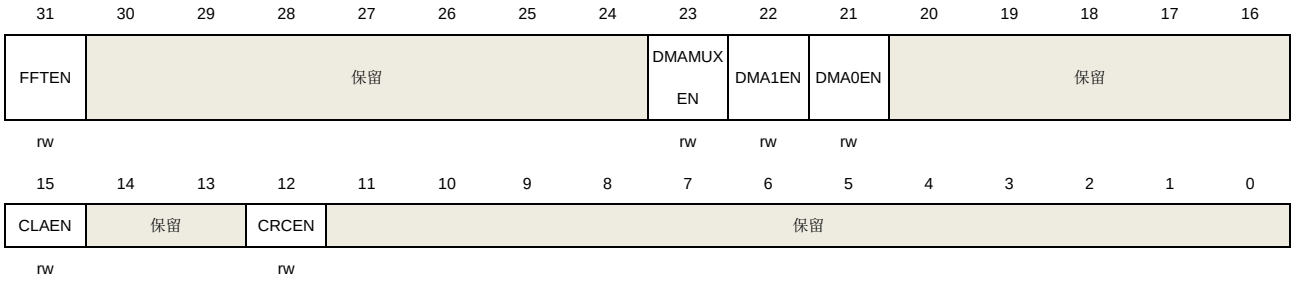
7:0	保留	必须保持复位值。

4.3.11. AHB1 使能寄存器（RCU_AHB1EN）

地址偏移：0x030

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域	名称	描述
31	FFTEN	FFT 时钟使能 由软件置位或复位 0：关闭 FFT 时钟 1：开启 FFT 时钟
30:24	保留	必须保持复位值。
23	DMAMUXEN	DMAMUX 时钟使能 由软件置位或复位 0：关闭 DMAMUX 时钟 1：开启 DMAMUX 时钟
22	DMA1EN	DMA1 时钟使能 由软件置位或复位 0：关闭 DMA1 时钟 1：开启 DMA1 时钟
21	DMA0EN	DMA0 时钟使能 由软件置位或复位 0：关闭 DMA0 时钟 1：开启 DMA0 时钟
20:16	保留	必须保持复位值。
15	CLAEN	CLA 时钟使能

由软件置位或复位
0：关闭 CLA 时钟
1：开启 CLA 时钟

14:13	保留	必须保持复位值。
12	CRCEN	CRC 时钟使能 由软件置位或复位 0：关闭 CRC 时钟 1：开启 CRC 时钟
11:0	保留	必须保持复位值。

4.3.12. AHB2 使能寄存器（RCU_AHB2EN）

地址偏移：0x034
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								PGEN	PFEN	PEEN	PDEN	PCEN	PBEN	PAEN	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								TMUEN	TRNGEN	保留		CAUEN	保留	FACEN	保留
								rw	rw			rw		rw	

位/位域	名称	描述
31:24	保留	必须保持复位值。
23	PGEN	GPIO 端口 G 时钟使能 由软件置位或复位 0：关闭 GPIO 端口 G 时钟 1：开启 GPIO 端口 G 时钟
22	PFEN	GPIO 端口 F 时钟使能 由软件置位或复位 0：关闭 GPIO 端口 F 时钟 1：开启 GPIO 端口 F 时钟
21	PEEN	GPIO 端口 E 时钟使能 由软件置位或复位 0：关闭 GPIO 端口 E 时钟 1：开启 GPIO 端口 E 时钟

20	PDEN	GPIO 端口 D 时钟使能 由软件置位或复位 0：关闭 GPIO 端口 D 时钟 1：开启 GPIO 端口 D 时钟
19	PCEN	GPIO 端口 C 时钟使能 由软件置位或复位 0：关闭 GPIO 端口 C 时钟 1：开启 GPIO 端口 C 时钟
18	PBEN	GPIO 端口 B 时钟使能 由软件置位或复位 0：关闭 GPIO 端口 B 时钟 1：开启 GPIO 端口 B 时钟
17	PAEN	GPIO 端口 A 时钟使能 由软件置位或复位 0：关闭 GPIO 端口 A 时钟 1：开启 GPIO 端口 A 时钟
16:8	保留	必须保持复位值。
7	TMUEN	TMU 时钟使能 由软件置位或复位 0：关闭 TMU 时钟 1：开启 TMU 时钟
6	TRNGEN	TRNG 时钟使能 由软件置位或复位 0：关闭 TRNG 时钟 1：开启 TRNG 时钟
5:4	保留	必须保持复位值。
3	CAUEN	CAU 时钟使能 由软件置位或复位 0：关闭 CAU 时钟 1：开启 CAU 时钟
2	保留	必须保持复位值。
1	FACEN	FAC 时钟使能 由软件置位或复位 0：关闭 FAC 时钟 1：开启 FAC 时钟

0	保留	必须保持复位值。

4.3.13. AHB3 使能寄存器（RCU_AHB3EN）

地址偏移：0x038
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														QSPIEN	EXMCEN
														rw	rw

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	QSPIEN	QSPI 时钟使能 由软件置位或复位 0：关闭 QSPI 时钟 1：开启 QSPI 时钟
0	EXMCEN	EXMC 时钟使能 由软件置位或复位 0：关闭 EXMC 时钟 1：开启 EXMC 时钟

4.3.14. APB1 使能寄存器（RCU_APB1EN）

地址偏移：0x40
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			PMUEN	保留			I2C3EN	I2C2EN	I2C1EN	I2C0EN	UART4EN	UART3EN	USART2EN	USART1EN	保留
			rw				rw	rw	rw	rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI2EN	SPI1EN	保留		WWDGTEN	保留	LPTIMEREN	保留			TIMER6EN	TIMER5EN	TIMER4EN	TIMER3EN	TIMER2EN	TIMER1EN

rw rw rw rw rw rw rw rw rw rw

位/位域	名称	描述
31:29	保留	必须保持复位值。
28	PMUEN	PMU 时钟使能 由软件置位或复位 0：关闭 PMU 时钟 1：开启 PMU 时钟
27:25	保留	必须保持复位值。
24	I2C3EN	I2C3 时钟使能 由软件置位或复位 0：关闭 I2C3 时钟 1：开启 I2C3 时钟
23	I2C2EN	I2C2 时钟使能 由软件置位或复位 0：关闭 I2C2 时钟 1：开启 I2C2 时钟
22	I2C1EN	I2C1 时钟使能 由软件置位或复位 0：关闭 I2C1 时钟 1：开启 I2C1 时钟
21	I2C0EN	I2C0 时钟使能 由软件置位或复位 0：关闭 I2C0 时钟 1：开启 I2C0 时钟
20	UART4EN	UART4 时钟使能 由软件置位或复位 0：关闭 UART4 时钟 1：开启 UART4 时钟
19	UART3EN	UART3 时钟使能 由软件置位或复位 0：关闭 UART3 时钟 1：开启 UART3 时钟
18	USART2EN	USART2 时钟使能 由软件置位或复位 0：关闭 USART2 时钟

		1：开启 USART2 时钟
17	USART1EN	USART1 时钟使能 由软件置位或复位 0：关闭 USART1 时钟 1：开启 USART1 时钟
16	保留	必须保持复位值。
15	SPI2EN	SPI2 时钟使能 由软件置位或复位 0：关闭 SPI2 时钟 1：开启 SPI2 时钟
14	SPI1EN	SPI1 时钟使能 由软件置位或复位 0：关闭 SPI1 时钟 1：开启 SPI1 时钟
13:12	保留	必须保持复位值。
11	WWDGTEN	WWDGT 时钟使能 由软件置位或复位 0：关闭 WWDGT 时钟 1：开启 WWDGT 时钟
10	保留	必须保持复位值。
9	LPTIMEREN	LPTIMER 时钟使能 由软件置位或复位 0：关闭 LPTIMER 时钟 1：开启 LPTIMER 时钟
8:6	保留	必须保持复位值。
5	TIMER6EN	TIMER6 时钟使能 由软件置位或复位 0：关闭 TIMER6 时钟 1：开启 TIMER6 时钟
4	TIMER5EN	TIMER5 时钟使能 由软件置位或复位 0：关闭 TIMER5 时钟 1：开启 TIMER5 时钟
3	TIMER4EN	TIMER4 时钟使能

由软件置位或复位
0：关闭 TIMER4 时钟
1：开启 TIMER4 时钟

2 TIMER3EN TIMER3 时钟使能
由软件置位或复位
0：关闭 TIMER3 时钟
1：开启 TIMER3 时钟

1 TIMER2EN TIMER2 时钟使能
由软件置位或复位
0：关闭 TIMER2 时钟
1：开启 TIMER2 时钟

0 TIMER1EN TIMER1 时钟使能
由软件置位或复位
0：关闭 TIMER1 时钟
1：开启 TIMER1 时钟

4.3.15. APB2 使能寄存器（RCU_APB2EN）

地址偏移：0x44
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
TRIGSEL EN	保留	HRTIME REN	保留									HPDFEN	TIMER16 EN	TIMER15 EN	TIMER14 EN
rw		rw										rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER19 EN	SYSCFG EN	保留	SPI0EN	保留	CAN2EN	CAN1EN	CAN0EN	保留			USART0 EN	CMPEN	VREFEN	TIMER7E N	TIMER0E N
rw	rw		rw		rw	rw	rw				rw	rw	rw	rw	rw

位/位域	名称	描述
31	TRIGSELEN	TRIGSEL 时钟使能 由软件置位或复位 0：关闭 TRIGSEL 时钟 1：开启 TRIGSEL 时钟
30	保留	必须保持复位值。
29	HRTIMEREN	HRTIMER 时钟使能

		由软件置位或复位 0：关闭 HRTIMER 时钟 1：开启 HRTIMER 时钟
28:20	保留	必须保持复位值。
19	HPDFEN	HPDF 时钟使能 由软件置位或复位 0：关闭 HPDF 时钟 1：开启 HPDF 时钟
18	TIMER16EN	TIMER16 时钟使能 由软件置位或复位 0：关闭 TIMER16 时钟 1：开启 TIMER16 时钟
17	TIMER15EN	TIMER14 时钟使能 由软件置位或复位 0：关闭 TIMER14 时钟 1：开启 TIMER14 时钟
16	TIMER14EN	TIMER14 时钟使能 由软件置位或复位 0：关闭 TIMER14 时钟 1：开启 TIMER14 时钟
15	TIMER19EN	TIMER19 时钟使能 由软件置位或复位 0：关闭 TIMER19 时钟 1：开启 TIMER19 时钟
14	SYSCFGEN	SYSCFG 时钟使能 由软件置位或复位 0：关闭 SYSCFG 时钟 1：开启 SYSCFG 时钟
13	保留	必须保持复位值。
12	SPI0EN	SPI0 时钟使能 由软件置位或复位 0：关闭 SPI0 时钟 1：开启 SPI0 时钟
11	保留	必须保持复位值。
10	CAN2EN	CAN2 时钟使能

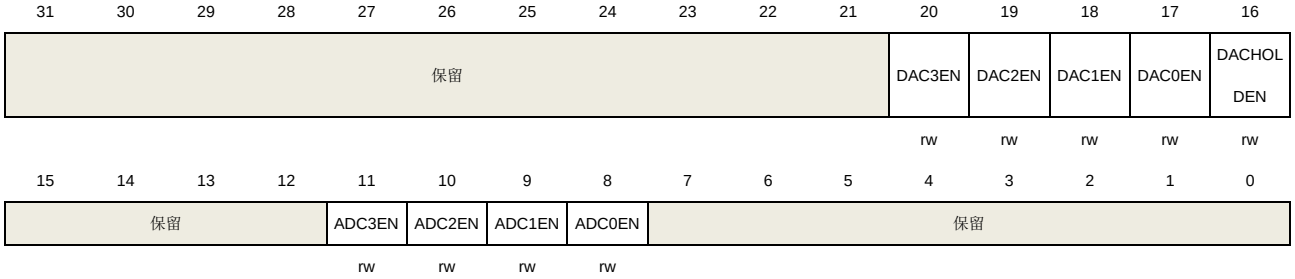
		由软件置位或复位 0：关闭 CAN2 时钟 1：开启 CAN2 时钟
9	CAN1EN	CAN1 时钟使能 由软件置位或复位 0：关闭 CAN1 时钟 1：开启 CAN1 时钟
8	CAN0EN	CAN0 时钟使能 由软件置位或复位 0：关闭 CAN0 时钟 1：开启 CAN0 时钟
7:5	保留	必须保持复位值。
4	USART0EN	USART0 时钟使能 由软件置位或复位 0：关闭 USART0 时钟 1：开启 USART0 时钟
3	CMPEN	CMP 时钟使能 由软件置位或复位 0：关闭 CMP 时钟 1：开启 CMP 时钟
2	VREFEN	VREF 时钟使能 由软件置位或复位 0：关闭 VREF 时钟 1：开启 VREF 时钟
1	TIMER7EN	TIMER7 时钟使能 由软件置位或复位 0：关闭 TIMER7 时钟 1：开启 TIMER7 时钟
0	TIMER0EN	TIMER0 时钟使能 由软件置位或复位 0：关闭 TIMER0 时钟 1：开启 TIMER0 时钟

4.3.16. APB3 使能寄存器（RCU_APB3EN）

地址偏移：0x48

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域	名称	描述
31:21	保留	必须保持复位值。
20	DAC3EN	DAC3 时钟使能 由软件置位或复位 0：关闭 DAC3 时钟 1：开启 DAC3 时钟
19	DAC2EN	DAC2 时钟使能 由软件置位或复位 0：关闭 DAC2 时钟 1：开启 DAC2 时钟
18	DAC1EN	DAC1 时钟使能 由软件置位或复位 0：关闭 DAC1 时钟 1：开启 DAC1 时钟
17	DAC0EN	DAC0 时钟使能 由软件置位或复位 0：关闭 DAC0 时钟 1：开启 DAC0 时钟
16	DACHOLDEN	DAC 保持时钟使能 由软件置位或复位，DAC 保持时钟源为 IRC32K。 0：关闭 DAC 保持时钟 1：开启 DAC 保持时钟
15:12	保留	必须保持复位值。
11	ADC3EN	ADC3 时钟使能 由软件置位或复位 0：关闭 ADC3 时钟

		1：开启 ADC3 时钟
10	ADC2EN	ADC2 时钟使能 由软件置位或复位 0：关闭 ADC2 时钟 1：开启 ADC2 时钟
9	ADC1EN	ADC1 时钟使能 由软件置位或复位 0：关闭 ADC1 时钟 1：开启 ADC1 时钟
8	ADC0EN	ADC0 时钟使能 由软件置位或复位 0：关闭 ADC0 时钟 1：开启 ADC0 时钟
7:0	保留	必须保持复位值。

4.3.17. AHB1 睡眠和深度睡眠使能寄存器（RCU_AHB1SPDPEN）

地址偏移：0x050

复位值：0x80EB 9100

注意：Bit8不能被清零。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FFTSPDPEN	保留							DMAMUXSPDPEN	DMA1SPDPEN	DMA0SPDPEN	保留	TCMSRAMSPDPEN	保留	SRAM1SPDPEN	SRAM0SPDPEN
rw								rw	rw	rw		rw		rw	rw

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

位/位域	名称	描述
31	FFTSPDPEN	在睡眠和深度睡眠模式下 FFT 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 FFT 时钟 1：在睡眠和深度睡眠模式下开启 FFT 时钟

30:24	保留	必须保持复位值。
23	DMAMUXSPDPEN	在睡眠和深度睡眠模式下 DMAMUX 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 DMAMUX 时钟 1：在睡眠和深度睡眠模式下开启 DMAMUX 时钟
22	DMA1SPDPEN	在睡眠和深度睡眠模式下 DMA1 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 DMA1 时钟 1：在睡眠和深度睡眠模式下开启 DMA1 时钟
21	DMA0SPDPEN	在睡眠和深度睡眠模式下 DMA0 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 DMA0 时钟 1：在睡眠和深度睡眠模式下开启 DMA0 时钟
20	保留	必须保持复位值。
19	TCMSRAMSPDPEN	在睡眠和深度睡眠模式下 TCMSRAM 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 TCMSRAM 时钟 1：在睡眠和深度睡眠模式下开启 TCMSRAM 时钟
18	保留	必须保持复位值。
17	SRAM1SPDPEN	在睡眠和深度睡眠模式下 SRAM1 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 SRAM1 时钟 1：在睡眠和深度睡眠模式下开启 SRAM1 时钟
16	SRAM0SPDPEN	在睡眠和深度睡眠模式下 SRAM0 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 SRAM0 时钟 1：在睡眠和深度睡眠模式下开启 SRAM0 时钟
15	CLASPDPEN	在睡眠和深度睡眠模式下 CLA 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 CLA 时钟 1：在睡眠和深度睡眠模式下开启 CLA 时钟
14:13	保留	必须保持复位值。
12	CRCSPDPEN	在睡眠和深度睡眠模式下 CRC 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 CRC 时钟

1：在睡眠和深度睡眠模式下开启 CRC 时钟

11:0	保留	必须保持复位值。

4.3.18. AHB2 睡眠和深度睡眠使能寄存器（RCU_AHB2SPDPEN）

地址偏移：0x054

复位值：0x00FE 00CA

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								PGSPDPEN	PFSPDPEN	PESPDPEN	PDSPDPEN	PCSPDPEN	PBSPDPEN	PASPDPEN	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								TMUSPDPEN	TRNGSPDPEN	保留		CAUSPDPEN	保留	FACSPDPEN	保留
								rw	rw			rw		rw	

位/位域	名称	描述
31:24	保留	必须保持复位值。
23	PGSPDPEN	在睡眠和深度睡眠模式下 GPIO 端口 G 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 GPIO 端口 G 时钟 1：在睡眠和深度睡眠模式下开启 GPIO 端口 G 时钟
22	PFSPDPEN	在睡眠和深度睡眠模式下 GPIO 端口 F 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 GPIO 端口 F 时钟 1：在睡眠和深度睡眠模式下开启 GPIO 端口 F 时钟
21	PESPDPEN	在睡眠和深度睡眠模式下 GPIO 端口 E 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 GPIO 端口 E 时钟 1：在睡眠和深度睡眠模式下开启 GPIO 端口 E 时钟
20	PDSPDPEN	在睡眠和深度睡眠模式下 GPIO 端口 D 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 GPIO 端口 D 时钟 1：在睡眠和深度睡眠模式下开启 GPIO 端口 D 时钟
19	PCSPDPEN	在睡眠和深度睡眠模式下 GPIO 端口 C 时钟使能 由软件置位或复位

		0：在睡眠和深度睡眠模式下关闭 GPIO 端口 C 时钟 1：在睡眠和深度睡眠模式下开启 GPIO 端口 C 时钟
18	PBSPDPEN	在睡眠和深度睡眠模式下 GPIO 端口 B 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 GPIO 端口 B 时钟 1：在睡眠和深度睡眠模式下开启 GPIO 端口 B 时钟
17	PASPDPEN	在睡眠和深度睡眠模式下 GPIO 端口 A 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 GPIO 端口 A 时钟 1：在睡眠和深度睡眠模式下开启 GPIO 端口 A 时钟
16:8	保留	必须保持复位值。
7	TMUSPDPEN	在睡眠和深度睡眠模式下 TMU 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 TMU 时钟 1：在睡眠和深度睡眠模式下开启 TMU 时钟
6	TRNGSPDPEN	在睡眠和深度睡眠模式下 TRNG 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 TRNG 时钟 1：在睡眠和深度睡眠模式下开启 TRNG 时钟
5:4	保留	必须保持复位值。
3	CAUSPDPEN	在睡眠和深度睡眠模式下 CAU 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 CAU 时钟 1：在睡眠和深度睡眠模式下开启 CAU 时钟
2	保留	必须保持复位值。
1	FACSPDPEN	在睡眠和深度睡眠模式下 FAC 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 FAC 时钟 1：在睡眠和深度睡眠模式下开启 FAC 时钟
0	保留	必须保持复位值。

4.3.19. AHB3 睡眠和深度睡眠使能寄存器（RCU_AHB3SPDPEN）

地址偏移：0x058

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														QSPISPDPEN	EXMCSPDPEN
														rw	rw

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	QSPISPDPEN	在睡眠和深度睡眠模式下 QSPI 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 QSPI 时钟 1：在睡眠和深度睡眠模式下开启 QSPI 时钟
0	EXMCSPDPEN	在睡眠和深度睡眠模式下 EXMC 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 EXMC 时钟 1：在睡眠和深度睡眠模式下开启 EXMC 时钟

4.3.20. APB1 睡眠和深度睡眠使能寄存器（RCU_APB1SPDPEN）

地址偏移：0x60

复位值：0x11FE CA3F

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			PMUSPDPEN	保留			I2C3SPDPEN	I2C2SPDPEN	I2C1SPDPEN	I2C0SPDPEN	UART4SPDPEN	UART3SPDPEN	USART2SPDPEN	USART1SPDPEN	保留
			rw				rw	rw	rw	rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI2SPDPEN	SPI1SPDPEN	保留		WWDGTSPDPEN	保留	LPTIMERSPDPEN	保留			TIMER6SPDPEN	TIMER5SPDPEN	TIMER4SPDPEN	TIMER3SPDPEN	TIMER2SPDPEN	TIMER1SPDPEN
rw	rw			rw		rw				rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:29	保留	必须保持复位值。
28	PMUSPDPEN	在睡眠和深度睡眠模式下 PMU 时钟使能 由软件置位或复位

		0：在睡眠和深度睡眠模式下关闭 PMU 时钟 1：在睡眠和深度睡眠模式下开启 PMU 时钟
27:25	保留	必须保持复位值。
24	I2C3SPDPEN	在睡眠和深度睡眠模式下 I2C3 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 I2C3 时钟 1：在睡眠和深度睡眠模式下开启 I2C3 时钟
23	I2C2SPDPEN	在睡眠和深度睡眠模式下 I2C2 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 I2C2 时钟 1：在睡眠和深度睡眠模式下开启 I2C2 时钟
22	I2C1SPDPEN	在睡眠和深度睡眠模式下 I2C1 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 I2C1 时钟 1：在睡眠和深度睡眠模式下开启 I2C1 时钟
21	I2C0SPDPEN	在睡眠和深度睡眠模式下 I2C0 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 I2C0 时钟 1：在睡眠和深度睡眠模式下开启 I2C0 时钟
20	UART4SPDPEN	在睡眠和深度睡眠模式下 UART4 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 UART4 时钟 1：在睡眠和深度睡眠模式下开启 UART4 时钟
19	UART3SPDPEN	在睡眠和深度睡眠模式下 UART3 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 UART3 时钟 1：在睡眠和深度睡眠模式下开启 UART3 时钟
18	USART2SPDPEN	在睡眠和深度睡眠模式下 USART2 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 USART2 时钟 1：在睡眠和深度睡眠模式下开启 USART2 时钟
17	USART1SPDPEN	在睡眠和深度睡眠模式下 USART1 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 USART1 时钟 1：在睡眠和深度睡眠模式下开启 USART1 时钟

16	保留	必须保持复位值。
15	SPI2SPDPEN	在睡眠和深度睡眠模式下 SPI2 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 SPI2 时钟 1：在睡眠和深度睡眠模式下开启 SPI2 时钟
14	SPI1SPDPEN	在睡眠和深度睡眠模式下 SPI1 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 SPI1 时钟 1：在睡眠和深度睡眠模式下开启 SPI1 时钟
13:12	保留	必须保持复位值。
11	WWDGTSPDPEN	在睡眠和深度睡眠模式下 WWDGT 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 WWDGT 时钟 1：在睡眠和深度睡眠模式下开启 WWDGT 时钟
10	保留	必须保持复位值。
9	LPTIMERSPDPEN	在睡眠和深度睡眠模式下 LPTIMER 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 LPTIMER 时钟 1：在睡眠和深度睡眠模式下开启 LPTIMER 时钟
8:6	保留	必须保持复位值。
5	TIMER6SPDPEN	在睡眠和深度睡眠模式下 TIMER6 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 TIMER6 时钟 1：在睡眠和深度睡眠模式下开启 TIMER6 时钟
4	TIMER5SPDPEN	在睡眠和深度睡眠模式下 TIMER5 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 TIMER5 时钟 1：在睡眠和深度睡眠模式下开启 TIMER5 时钟
3	TIMER4SPDPEN	在睡眠和深度睡眠模式下 TIMER4 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 TIMER4 时钟 1：在睡眠和深度睡眠模式下开启 TIMER4 时钟
2	TIMER3SPDPEN	在睡眠和深度睡眠模式下 TIMER3 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 TIMER3 时钟

1：在睡眠和深度睡眠模式下开启 TIMER3 时钟

1	TIMER2SPDPEN	在睡眠和深度睡眠模式下 TIMER2 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 TIMER2 时钟 1：在睡眠和深度睡眠模式下开启 TIMER2 时钟
0	TIMER1SPDPEN	在睡眠和深度睡眠模式下 TIMER1 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 TIMER1 时钟 1：在睡眠和深度睡眠模式下开启 TIMER1 时钟

4.3.21. APB2 睡眠和深度睡眠使能寄存器（RCU_APB2SPDPEN）

地址偏移：0x64

复位值：0xA00F D71F

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
TRIGSEL SPDPEN	保留	HRTIME RSPDPE N	保留									HPDFSP DPEN	TIMER16 SPDPEN	TIMER15 SPDPEN	TIMER14 SPDPEN
rw		rw										rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER19 SPDPEN	SYSCFG SPDPEN	保留	SPI0SPD PEN	保留	CAN2SP DPEN	CAN1SP DPEN	CAN0SP DPEN	保留			USART0 SPDPEN	CMPSPD PEN	VREFSP DPEN	TIMER7S PDPEN	TIMER0S PDPEN
rw	rw		rw		rw	rw	rw				rw	rw	rw	rw	rw

位/位域	名称	描述
31	TRIGSELSPDPEN	在睡眠和深度睡眠模式下 TRIGSEL 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 TRIGSEL 时钟 1：在睡眠和深度睡眠模式下开启 TRIGSEL 时钟
30	保留	必须保持复位值。
29	HRTIMERSPDPEN	在睡眠和深度睡眠模式下 HRTIMER 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 HRTIMER 时钟 1：在睡眠和深度睡眠模式下开启 HRTIMER 时钟
28:20	保留	必须保持复位值。

19	HPDFSPDPEN	在睡眠和深度睡眠模式下 HPDF 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 HPDF 时钟 1：在睡眠和深度睡眠模式下开启 HPDF 时钟
18	TIMER16SPDPEN	在睡眠和深度睡眠模式下 TIMER16 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 TIMER16 时钟 1：在睡眠和深度睡眠模式下开启 TIMER16 时钟
17	TIMER15SPDPEN	在睡眠和深度睡眠模式下 TIMER15 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 TIMER15 时钟 1：在睡眠和深度睡眠模式下开启 TIMER15 时钟
16	TIMER14SPDPEN	在睡眠和深度睡眠模式下 TIMER14 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 TIMER14 时钟 1：在睡眠和深度睡眠模式下开启 TIMER14 时钟
15	TIMER19SPDPEN	在睡眠和深度睡眠模式下 TIMER19 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 TIMER19 时钟 1：在睡眠和深度睡眠模式下开启 TIMER19 时钟
14	SYSCFGSPDPEN	在睡眠和深度睡眠模式下 SYSCFG 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 SYSCFG 时钟 1：在睡眠和深度睡眠模式下开启 SYSCFG 时钟
13	保留	必须保持复位值。
12	SPI0SPDPEN	在睡眠和深度睡眠模式下 SPI0 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 SPI0 时钟 1：在睡眠和深度睡眠模式下开启 SPI0 时钟
11	保留	必须保持复位值。
10	CAN2SPDPEN	在睡眠和深度睡眠模式下 CAN2 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 CAN2 时钟 1：在睡眠和深度睡眠模式下开启 CAN2 时钟
9	CAN1SPDPEN	在睡眠和深度睡眠模式下 CAN1 时钟使能

		由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 CAN1 时钟 1：在睡眠和深度睡眠模式下开启 CAN1 时钟
8	CAN0SPDPEN	在睡眠和深度睡眠模式下 CAN0 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 CAN0 时钟 1：在睡眠和深度睡眠模式下开启 CAN0 时钟
7:5	保留	必须保持复位值。
4	USART0SPDPEN	在睡眠和深度睡眠模式下 USART0 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 USART0 时钟 1：在睡眠和深度睡眠模式下开启 USART0 时钟
3	CMPSPDPEN	在睡眠和深度睡眠模式下 CMP 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 CMP 时钟 1：在睡眠和深度睡眠模式下开启 CMP 时钟
2	VREFSPDPEN	在睡眠和深度睡眠模式下 VREF 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 VREF 时钟 1：在睡眠和深度睡眠模式下开启 VREF 时钟
1	TIMER7SPDPEN	在睡眠和深度睡眠模式下 TIMER7 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 TIMER7 时钟 1：在睡眠和深度睡眠模式下开启 TIMER7 时钟
0	TIMER0SPDPEN	在睡眠和深度睡眠模式下 TIMER0 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 TIMER0 时钟 1：在睡眠和深度睡眠模式下开启 TIMER0 时钟

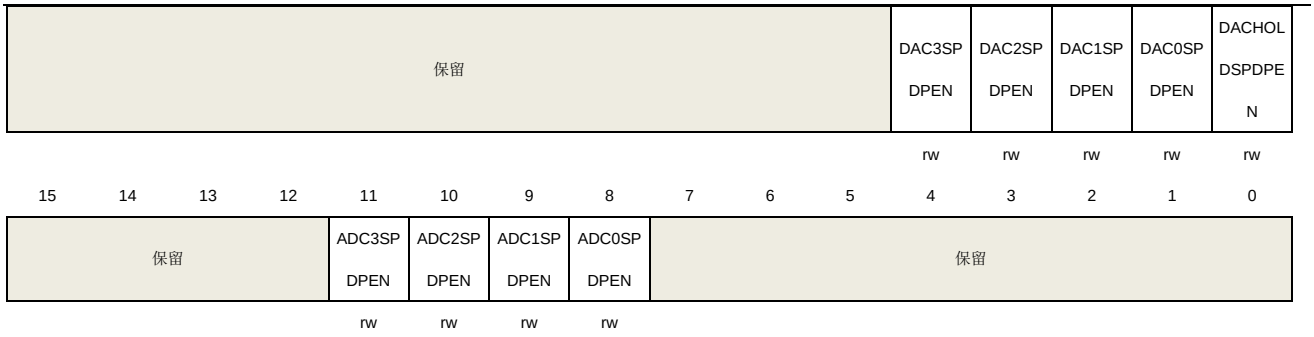
4.3.22. APB3 睡眠和深度睡眠使能寄存器（RCU_APB3SPDPEN）

地址偏移：0x68

复位值：0x001F 0F00

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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



位/位域	名称	描述
31:21	保留	必须保持复位值。
20	DAC3SPDPEN	在睡眠和深度睡眠模式下 DAC3 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 DAC3 时钟 1：在睡眠和深度睡眠模式下开启 DAC3 时钟
19	DAC2SPDPEN	在睡眠和深度睡眠模式下 DAC2 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 DAC2 时钟 1：在睡眠和深度睡眠模式下开启 DAC2 时钟
18	DAC1SPDPEN	在睡眠和深度睡眠模式下 DAC1 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 DAC1 时钟 1：在睡眠和深度睡眠模式下开启 DAC1 时钟
17	DAC0SPDPEN	在睡眠和深度睡眠模式下 DAC0 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 DAC0 时钟 1：在睡眠和深度睡眠模式下开启 DAC0 时钟
16	DACHOLDSPDPEN	在睡眠和深度睡眠模式下 DAC 保持时钟使能 由软件置位或复位，DAC 保持时钟源为 IRC32K 0：在睡眠模式下关闭 DAC 保持时钟 1：在睡眠模式下开启 DAC 保持时钟
15:12	保留	必须保持复位值。
11	ADC3SPDPEN	在睡眠和深度睡眠模式下 ADC3 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 ADC3 时钟 1：在睡眠和深度睡眠模式下开启 ADC3 时钟

10	ADC2SPDPEN	在睡眠和深度睡眠模式下 ADC2 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 ADC2 时钟 1：在睡眠和深度睡眠模式下开启 ADC2 时钟
9	ADC1SPDPEN	在睡眠和深度睡眠模式下 ADC1 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 ADC1 时钟 1：在睡眠和深度睡眠模式下开启 ADC1 时钟
8	ADC0SPDPEN	在睡眠和深度睡眠模式下 ADC0 时钟使能 由软件置位或复位 0：在睡眠和深度睡眠模式下关闭 ADC0 时钟 1：在睡眠和深度睡眠模式下开启 ADC0 时钟
7:0	保留	必须保持复位值。

4.3.23. 备份域控制寄存器（RCU_BDCTL）

地址偏移：0x70

复位值：0x0000 0018，只能由备份域复位进行复位

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

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

位/位域	名称	描述
31:26	保留	必须保持复位值。
25	LSCKOUTSEL	低速时钟输出选择 0：IRC32K 1：LXTAL

24	LSCKOUTEN	低速时钟输出使能 0：低速时钟输出失能 1：低速时钟输出使能
23:17	保留	必须保持复位值。
16	BKPRST	备份域复位 由软件置位或复位 0：无作用 1：复位备份域
15	RTCEN	RTC 时钟使能 由软件置位或复位 0：关闭 RTC 时钟 1：开启 RTC 时钟
14:10	保留	必须保持复位值。
9:8	RTCSRC[1:0]	RTC 时钟源选择 由软件置位或清零来控制 RTC 的时钟源。一旦 RTC 的时钟源选择后，除了将备份域复位否则时钟源不能被改变。 00：没有时钟 01：选择 CK_LXTAL 时钟作为 RTC 的时钟源 10：选择 CK_IRC32K 时钟作为 RTC 的时钟源 11：选择 CK_HXTAL / 32 时钟作为 RTC 的时钟源
7	LXTALSTBRST	低速晶体振荡器稳定标志位复位 0：低速晶体振荡器稳定标志位不复位 1：低速晶体振荡器稳定标志位复位
6	LCKMD	LXTAL 时钟故障检测 由硬件置位，当外部 32 kHz 振荡器（LXTAL）上的时钟安全系统检测到故障。当 LCKMEN 或 LXTALEN 关闭时，该位清零。 0：LXTAL（32 kHz 振荡器）上未检测到故障 1：在 LXTAL（32 kHz 振荡器）上检测到故障
5	LCKMEN	LXTAL 时钟监视器使能 0：禁止 LXTAL 时 1：使能 LXTAL 时钟监视器 通过软件设置，启用 LXTAL（32 kHz 振荡器）上的时钟安全系统。LXTALEN 必须在 LXTAL 已启用（LXTALEN 位已启用）和就绪（LXTALSTB 标志由硬件设置)。 **注意**：一旦该位被置位，该位可以通过备份域复位清除或者在检测到 LXTAL 时钟故障后（LCKMD = 1）通过复位 LCKMEN 清除。钟监视器

4:3	LXTALDRI[1:0]	LXTAL 驱动能力 由软件置位或复位。当备份域复位时将复位该值 00：禁止 01：低驱动能力 10：中驱动能力 11：高驱动能力（复位后的缺省值） **注意：**LXTALDRI 位在旁路模式下无效
2	LXTALBPS	LXTAL 旁路模式使能 由软件置位或复位 0：禁止 LXTAL 旁路模式 1：使能 LXTAL 旁路模式
1	LXTALSTB	低速晶体振荡器稳定标志位 硬件置‘1’来指示 LXTAL 振荡器时钟是否稳定待用 0：LXTAL 未稳定 1：LXTAL 已稳定
0	LXTALEN	LXTAL 时钟使能 由软件置位或复位 0：关闭 LXTAL 时钟 1：使能 LXTAL 时钟

4.3.24. 复位源/时钟寄存器（RCU_RSTSCK）

地址偏移：0x74

复位值：0x0E00 0000，所有复位标志位仅在电源复位时被清零，RSTFC / IRC32KEN在系统复位时被清零。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LP RSTF	WWDGT RSTF	FWDGT RSTF	SW RSTF	POR RSTF	EP RSTF	BORRSTF	RSTFC	OBLRST	保留						
r	r	r	r	r	r	r	rw	r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														IRC32K STB	IRC32KEN
														r	rw

位/位域	名称	描述
31	LPRSTF	低功耗复位标志位 深度睡眠/待机复位发生时由硬件置位

向 RSTFC 位写 1 来清除该位
0：无低功耗管理复位发生
1：发生低功耗管理复位

30 WWDGTRSTF 窗口看门狗定时器复位标志位
窗口看门狗定时器复位发生时由硬件置 1
向 RSTFC 位写 1 来清除该位
0：无窗口看门狗复位发生
1：发生窗口看门狗复位

29 FWDGTRSTF 独立看门狗定时器复位标志位
独立看门狗复位发生时由硬件置 1
向 RSTFC 位写 1 来清除该位
0：无独立看门狗定时器复位发生
1：发生独立看门狗定时器复位

28 SWRSTF 软件复位标志位
软件复位发生时由硬件置 1
向 RSTFC 位写 1 来清除该位
0：无软件复位发生
1：发生软件复位

27 PORRSTF 电源复位标志位
电源复位发生时由硬件置 1
向 RSTFC 位写 1 来清除该位
0：无电源复位发生
1：发生电源复位

26 EPRSTF 外部引脚复位标志位
外部引脚复位发生时由硬件置 1
向 RSTFC 位写 1 来清除该位
0：无外部引脚复位发生
1：发生外部引脚复位

25 BORRSTF 欠压复位复位标志位
欠压复位复位发生时由硬件置 1
向 RSTFC 位写 1 来清除该位
0：无欠压复位复位发生
1：发生欠压复位复位

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：清除所有复位标志位
23	OBLRSTF	选项字节重载复位标志位 选项字节重载复位发生时由硬件置 1 向 RSTFC 位写 1 来清除该位 0：无选项字节重载复位复位发生 1：发生选项字节重载复位
22:2	保留	必须保持复位值。
1	IRC32KSTB	IRC32K 时钟稳定标志位 该位由硬件置 1 指示 IRC32K 输出时钟是否稳定待用 0：IRC32K 时钟未稳定 1：IRC32K 已稳定
0	IRC32KEN	IRC32K 使能 由软件置位和复位 0：关闭 IRC32K 时钟 1：开启 IRC32K 时钟

4.3.25. 时钟配置寄存器 1（RCU_CFG1）

地址偏移：0x8C

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

位/位域	名称	描述
31	HPDFSEL	HPDF 时钟源选择 由软件置位或复位，控制 HPDF 时钟源 0：CK_APB2 被选择为 HPDF 时钟的时钟源 1：CK_AHB 被选择为 HPDF 时钟的时钟源
30:29	HPDFAUDIOSEL[1:0]	HPDF AUDIO 时钟源预选择 由软件置位或复位，控制 HPDF AUDIO 时钟源 00：CK_PLLQ 被选择为 HPDF AUDIO 时钟的时钟源

		01：外部 HPDF_CKIN 引脚被选择为 HPDF AUDIO 时时钟的时钟源 10：IRC8M 被选择为 HPDF AUDIO 时时钟的时钟源 11：保留
28:22	保留	必须保持复位值。
21:20	USART2SEL[1:0]	USART2 时钟源预选择 由软件置位或复位，控制 USART2 时钟源 00：CK_APB1 被选择为 USART2 时钟的时钟源 01：CK_SYS 被选择为 USART2 时时钟的时钟源 10：CK_LXTAL 被选择为 USART2 时时钟的时钟源 11：CK_IRC8M 被选择为 USART2 时时钟的时钟源
19:18	USART1SEL[1:0]	USART1 时钟源预选择 由软件置位或复位，控制 USART1 时钟源 00：CK_APB1 被选择为 USART1 时钟的时钟源 01：CK_SYS 被选择为 USART1 时时钟的时钟源 10：CK_LXTAL 被选择为 USART1 时时钟的时钟源 11：CK_IRC8M 被选择为 USART1 时时钟的时钟源
17:14	保留	必须保持复位值。
13:12	CAN2SEL[1:0]	CAN2 时钟源预选择 由软件置位或复位，控制 CAN2 时钟源 00：CK_IRC8M 被选择为 CAN2 时钟的时钟源 01：CK_APB2 被选择为 CAN2 时时钟的时钟源 10：CK_PLLQ 被选择为 CAN2 时时钟的时钟源 11：CK_HXTAL 被选择为 CAN2 时时钟的时钟源
11:10	CAN1SEL[1:0]	CAN1 时钟源预选择 由软件置位或复位，控制 CAN1 时钟源 00：CK_IRC8M 被选择为 CAN1 时钟的时钟源 01：CK_APB2 被选择为 CAN1 时时钟的时钟源 10：CK_PLLQ 被选择为 CAN1 时时钟的时钟源 11：CK_HXTAL 被选择为 CAN1 时时钟的时钟源
9:8	CAN0SEL[1:0]	CAN0 时钟源预选择 由软件置位或复位，控制 CAN0 时钟源 00：CK_IRC8M 被选择为 CAN0 时钟的时钟源 01：CK_APB2 被选择为 CAN0 时时钟的时钟源 10：CK_PLLQ 被选择为 CAN0 时时钟的时钟源 11：CK_HXTAL 被选择为 CAN0 时时钟的时钟源
7:2	保留	必须保持复位值。

1:0	USART0SEL[1:0]	USART0 时钟源预选择 由软件置位或复位，控制 USART0 时钟源 00：CK_APB2 被选择为 USART0 时钟的时钟源 01：CK_SYS 被选择为 USART0 时时钟的时钟源 10：CK_LXTAL 被选择为 USART0 时时钟的时钟源 11：CK_IRC8M 被选择为 USART0 时时钟的时钟源

4.3.26. 时钟配置寄存器 2（RCU_CFG2）

地址偏移：0x90

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		ADC3SEL[1:0]		ADC0_1_2SEL[1:0]		保留						HRTIMERSEL	保留	QSPISEL[1:0]	
		rw		rw								rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRNGPSC[3:0]				保留	LPTIMERSEL[1:0]		保留	I2C3SEL[1:0]		I2C2SEL[1:0]		I2C1SEL[1:0]		I2C0SEL[1:0]	
rw					rw			rw		rw		rw		rw	

位/位域	名称	描述
31:30	保留	必须保持复位值。
29:28	ADC3SEL[1:0]	ADC3 时钟源预选择 由软件置位或复位，控制 ADC3 时钟源 00：保留 01：CK_PLLR 被选择为 ADC3 时时钟的时钟源 10：CK_SYS 被选择为 ADC3 时时钟的时钟源 11：保留
27:26	ADC0_1_2SEL[1:0]	ADC0/1/2 时钟源预选择 由软件置位或复位，控制 ADC0/1/2 时钟源 00：保留 01：CK_PLLR 被选择为 ADC0/1/2 时时钟的时钟源 10：CK_SYS 被选择为 ADC0/1/2 时时钟的时钟源 11：保留
25:20	保留	必须保持复位值。
19	HRTIMERSEL	HRTIMER 时钟源预选择 由软件置位或复位，控制 HRTIMER 时钟源

		0：CK_APB2 被选择为 HRTIMER 时时钟的时钟源 1：CK_SYS 被选择为 HRTIMER 时时钟的时钟源
18	保留	必须保持复位值。
17:16	QSPISEL[1:0]	QSPI 时钟源预选择 由软件置位或复位，控制 QSPI 时钟源 00：CK_SYS 被选择为 QSPI 时钟的时钟源 01：CK_IRC8M 被选择为 QSPI 时时钟的时钟源 10：CK_PLLQ 被选择为 QSPI 时时钟的时钟源 11：CK_PLLR 被选择为 QSPI 时时钟的时钟源
15:12	TRNGPSC[3:0]	TRNG 分频选择 由软件置位或复位，控制 TRNG 时钟分频 0000：保留 0001：保留 0010：CK_PLLQ / 2 0011：CK_PLLQ / 3 0100：CK_PLLQ / 4 … 1111：CK_PLLQ / 15
11	保留	必须保持复位值。
10:9	LPTIMERSEL[1:0]	LPTIMER 时钟源预选择 由软件置位或复位，控制 LPTIMER 时钟源 00：CK_APB1 被选择为 LPTIMER 时钟的时钟源 01：CK_IRC32K 被选择为 LPTIMER 时时钟的时钟源 10：CK_LXTAL 被选择为 LPTIMER 时时钟的时钟源 11：CK_IRC8M 被选择为 LPTIMER 时时钟的时钟源
8	保留	必须保持复位值。
7:6	I2C3SEL[1:0]	I2C3 时钟源预选择 由软件置位或复位，控制 I2C3 时钟源 00：CK_APB1 被选择为 I2C3 时钟的时钟源 01：CK_SYS 被选择为 I2C3 时时钟的时钟源 10：CK_IRC8M 被选择为 I2C3 时时钟的时钟源 11：保留
5:4	I2C2SEL[1:0]	I2C2 时钟源预选择 由软件置位或复位，控制 I2C2 时钟源 00：CK_APB1 被选择为 I2C2 时钟的时钟源 01：CK_SYS 被选择为 I2C2 时时钟的时钟源

		10：CK_IRC8M 被选择为 I2C2 时时钟的时钟源 11：保留
3:2	I2C1SEL[1:0]	I2C1 时钟源预选择 由软件置位或复位，控制 I2C1 时钟源 00：CK_APB1 被选择为 I2C1 时钟的时钟源 01：CK_SYS 被选择为 I2C1 时时钟的时钟源 10：CK_IRC8M 被选择为 I2C1 时时钟的时钟源 11：保留
1:0	I2C0SEL[1:0]	I2C0 时钟源预选择 由软件置位或复位，控制 I2C0 时钟源 00：CK_APB1 被选择为 I2C0 时钟的时钟源 01：CK_SYS 被选择为 I2C0 时时钟的时钟源 10：CK_IRC8M 被选择为 I2C0 时时钟的时钟源 11：保留

5. 中断/事件控制器（EXTI）

5.1. 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 39 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。

5.2. 主要特性

- Cortex®-M33系统异常；
- 多达137种可屏蔽的外设中断；
- 4位中断优先级配置位—16个中断优先等级；
- 高效的中断处理；
- 支持异常抢占和咬尾中断；
- 将系统从省电模式唤醒；
- EXTI中有多达39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。

5.3. 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。下表列出了Cortex®-M33中的NVIC异常类型。

表5-1. 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

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

SysTick校准值设为25000，SysTick时钟频率配置为CK_SYS*0.125。此时若CK_SYS时钟被配置为200MHz，则SysTick中断会1ms响应一次。

表5-2. 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 16 的 LVD / VAVD / VOVD / VUVD 中断	0x0000_0044
IRQ 2	18	连接到 EXTI 线 18 的 RTC 侵入和时间戳中断 LXTAL 时钟阻塞中断	0x0000_0048
IRQ 3	19	连接到 EXTI 线 19 的 RTC 唤醒中断	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

中断编号	向量编号	外设中断描述	向量地址
IRQ 17	33	DMA0 通道 6 全局中断	0x0000_0084
IRQ 18	34	ADC0 和 ADC1 中断	0x0000_0088
IRQ 19-22	35-38	保留	0x0000_008C- 0x0000_0098
IRQ 23	39	EXTI 线[9:5]中断	0x0000_009C
IRQ 24	40	TIMER0 中止中断	0x0000_00A0
IRQ 25	41	TIMER0 更新中断	0x0000_00A4
IRQ 26	42	TIMER0 触发，换相和索引中断	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	TIMER2 全局中断	0x0000_00B4
IRQ 30	46	TIMER3 全局中断	0x0000_00B8
IRQ 31	47	连接到 EXTI 线 31 的 I2C0 事件和唤醒中断	0x0000_00BC
IRQ 32	48	I2C0 错误中断	0x0000_00C0
IRQ 33	49	连接到 EXTI 线 32 的 I2C1 事件和唤醒中断	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	连接到 EXTI 线 28 的 USART0 全局和唤醒中断	0x0000_00D4
IRQ 38	54	连接到 EXTI 线 29 的 USART1 全局和唤醒中断	0x0000_00D8
IRQ 39	55	连接到 EXTI 线 30 的 USART2 全局和唤醒中断	0x0000_00DC
IRQ 40	56	EXTI 线[15:10]中断	0x0000_00E0
IRQ 41	57	连接到 EXTI 线 17 的 RTC 闹钟中断	0x0000_00E4
IRQ 42	58	保留	0x0000_00E8
IRQ 43	59	TIMER7 中止，传输和索引错误中断	0x0000_00EC
IRQ 44	60	TIMER7 更新中断	0x0000_00F0
IRQ 45	61	TIMER7 触发，换相和索引中断	0x0000_00F4
IRQ 46	62	TIMER7 捕获比较中断	0x0000_00F8
IRQ 47	63	ADC2 全局中断	0x0000_00FC
IRQ 48	64	SYSCFG 全局中断	0x0000_0100
IRQ 49	65	连接到 EXTI 线 35 的 LPTIMER 全局和唤醒中断	0x0000_0104
IRQ 50	66	TIMER4 全局中断	0x0000_0108
IRQ 51	67	SPI2 全局中断	0x0000_010C
IRQ 52	68	UART3 全局中断	0x0000_0110
IRQ 53	69	UART4 全局中断	0x0000_0114
IRQ 54	70	TIMER5 全局中断 DAC2，DAC0 下溢错误中断	0x0000_0118
IRQ 55	71	TIMER6 全局中断	0x0000_011C

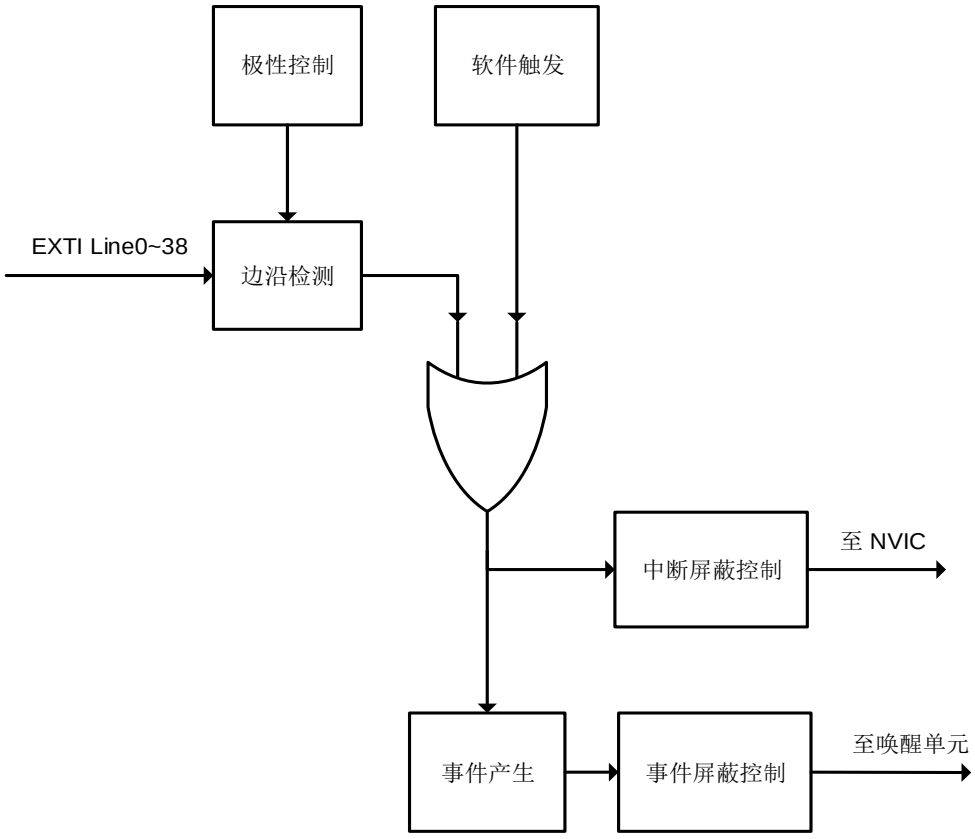
中断编号	向量编号	外设中断描述	向量地址
		DAC3，DAC1 下溢错误中断	
IRQ 56	72	DMA1 通道 0 全局中断	0x0000_0120
IRQ 57	73	DMA1 通道 1 全局中断	0x0000_0124
IRQ 58	74	DMA1 通道 2 全局中断	0x0000_0128
IRQ 59	75	DMA1 通道 3 全局中断	0x0000_012C
IRQ 60	76	DMA1 通道 4 全局中断	0x0000_0130
IRQ 61	77	ADC3 全局中断	0x0000_0134
IRQ 62	78	保留	0x0000_0138
IRQ 63	79	VUVD1，VOVD1 中断	0x0000_013C
IRQ 64	80	连接到 EXTI 线 20 / 21 / 22 / 23 的 CMP0 / CMP1 / CMP2 / CMP3 中断	0x0000_0140
IRQ 65	81	连接到 EXTI 线 24 / 36 / 37 / 38 的 CMP4 / CMP5 / CMP6 / CMP7 中断	0x0000_0144
IRQ 66	82	CMP 全局中断	0x0000_0148
IRQ 67	83	HRTIMER 中断 0	0x0000_014C
IRQ 68	84	HRTIMER 中断 1	0x0000_0150
IRQ 69	85	HRTIMER 中断 2	0x0000_0154
IRQ 70	86	HRTIMER 中断 3	0x0000_0158
IRQ 71	87	HRTIMER 中断 4	0x0000_015C
IRQ 72	88	HRTIMER 中断 5	0x0000_0160
IRQ 73	89	HRTIMER 中断 6	0x0000_0164
IRQ 74	90	HRTIMER 中断 7	0x0000_0168
IRQ 75	91	HRTIMER 中断 8	0x0000_016C
IRQ 76	92	HRTIMER 中断 9	0x0000_0170
IRQ 77	93	TIMER19 中止，传输和索引错误中断	0x0000_0174
IRQ 78	94	TIMER19 更新中断	0x0000_0178
IRQ 79	95	TIMER19 触发，换相和索引中断	0x0000_017C
IRQ 80	96	TIMER19 捕获比较中断	0x0000_0180
IRQ 81	97	FPU 全局中断	0x0000_0184
IRQ 82	98	连接到 EXTI 线 33 的 I2C2 事件和唤醒中断	0x0000_0188
IRQ 83	99	I2C2 错误中断	0x0000_018C
IRQ 84	100	保留	0x0000_0190
IRQ 85	101	CAU 全局中断	0x0000_0194
IRQ 86-89	102-105	保留	0x0000_0198-0x0000_01A4
IRQ 90	106	TRNG 全局中断	0x0000_01A8
IRQ 91	107	保留	0x0000_01AC
IRQ 92	108	连接到 EXTI 线 34 的 I2C3 事件和唤醒中断	0x0000_01B0

中断编号	向量编号	外设中断描述	向量地址
IRQ 93	109	I2C3 错误中断	0x0000_01B4
IRQ 94	110	DMAMUX 上溢中断	0x0000_01B8
IRQ 95	111	QSPI 全局中断	0x0000_01BC
IRQ 96	112	FFT 全局中断	0x0000_01C0
IRQ 97	113	DMA1 通道 5 全局中断	0x0000_01C4
IRQ 98	114	DMA1 通道 6 全局中断	0x0000_01C8
IRQ 99	115	CLA 全局中断	0x0000_01CC
IRQ 100	116	TMU 全局中断	0x0000_01D0
IRQ 101	117	FAC 全局中断	0x0000_01D4
IRQ 102	118	HPDF 全局中断 0	0x0000_01D8
IRQ 103	119	HPDF 全局中断 1	0x0000_01DC
IRQ 104	120	HPDF 全局中断 2	0x0000_01E0
IRQ 105	121	HPDF 全局中断 3	0x0000_01E4
IRQ 106	122	TIMER14 全局中断	0x0000_01E8
IRQ 107	123	TIMER15 全局中断	0x0000_01EC
IRQ 108	124	TIMER16 全局中断	0x0000_01F0
IRQ 109	125	连接到 EXTI 线 25 的 CAN 唤醒中断	0x0000_01F4
IRQ 110	126	CAN0 消息缓冲区中断	0x0000_01F8
IRQ 111	127	CAN0 总线关闭/总线关闭完成中断	0x0000_01FC
IRQ 112	128	CAN0 错误中断	0x0000_0200
IRQ 113	129	CAN0 快速传输错误中断	0x0000_0204
IRQ 114	130	CAN0 发送警告中断	0x0000_0208
IRQ 115	131	CAN0 接收警告中断	0x0000_020C
IRQ 116	132	连接到 EXTI 线的 CAN1 唤醒中断	0x0000_0210
IRQ 117	133	CAN1 消息缓冲区中断	0x0000_0214
IRQ 118	134	CAN1 总线关闭/总线关闭完成中断	0x0000_0218
IRQ 119	135	CAN1 错误中断	0x0000_021C
IRQ 120	136	CAN1 快速传输错误中断	0x0000_0220
IRQ 121	137	CAN1 发送警告中断	0x0000_0224
IRQ 122	138	CAN1 接收警告中断	0x0000_0228
IRQ 123	139	连接到 EXTI 线 27 的 CAN2 唤醒中断	0x0000_022C
IRQ 124	140	CAN2 消息缓冲区中断	0x0000_0230
IRQ 125	141	CAN2 总线关闭/总线关闭完成中断	0x0000_0234
IRQ 126	142	CAN2 错误中断	0x0000_0238
IRQ 127	143	CAN2 快速传输错误中断	0x0000_023C
IRQ 128	144	CAN2 发送警告中断	0x0000_0240
IRQ 129	145	CAN2 接收警告中断	0x0000_0244
IRQ 130	146	TIMER0 DEC 中断	0x0000_0248

中断编号	向量编号	外设中断描述	向量地址
IRQ 131	147	TIMER1 DEC 中断	0x0000_024C
IRQ 132	148	TIMER2 DEC 中断	0x0000_0250
IRQ 133	149	TIMER3 DEC 中断	0x0000_0254
IRQ 134	150	TIMER4 DEC 中断	0x0000_0258
IRQ 135	151	TIMER7 DEC 中断	0x0000_025C
IRQ 136	152	TIMER19 DEC 中断	0x0000_0260

5.4. 外部中断及事件（EXTI）框图

图5-1. EXTI框图



5.5. 外部中断及事件功能概述

EXTI 包含多达 39 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 23 根线，（包括 LVD，RTC 闹钟，RTC 侵入和时间戳，LXTAL 时钟阻塞，RTC 唤醒，CMP，CAN，USART 唤醒，I2C 唤醒，LPTIM 唤醒，OVD）。通过配置 SYSCFG 模块的 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 ***SYSCFG 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。

表5-3. EXTI触发源

EXTI 线编号	触发源
0	PA0 / PB0 / PC0 / PD0 / PE0 / PF0 / PG0
1	PA1 / PB1 / PC1 / PD1 / PE1 / PF1 / PG1
2	PA2 / PB2 / PC2 / PD2 / PE2 / PF2 / PG2
3	PA3 / PB3 / PC3 / PD3 / PE3 / PF3 / PG3
4	PA4 / PB4 / PC4 / PD4 / PE4 / PF4 / PG4
5	PA5 / PB5 / PC5 / PD5 / PE5 / PF5 / PG5
6	PA6 / PB6 / PC6 / PD6 / PE6 / PF6 / PG6
7	PA7 / PB7 / PC7 / PD7 / PE7 / PF7 / PG7
8	PA8 / PB8 / PC8 / PD8 / PE8 / PF8 / PG8
9	PA9 / PB9 / PC9 / PD9 / PE9 / PF9 / PG9
10	PA10 / PB10 / PC10 / PD10 / PE10 / PF10 / PG10
11	PA11 / PB11 / PC11 / PD11 / PE11 / PF11
12	PA12 / PB12 / PC12 / PD12 / PE12 / PF12
13	PA13 / PB13 / PC13 / PD13 / PE13 / PF13
14	PA14 / PB14 / PC14 / PD14 / PE14 / PF14
15	PA15 / PB15 / PC15 / PD15 / PE15 / PF15
16	LVD，VAVD，VOVD 和 VUVD
17	RTC 闹钟
18	RTC 侵入和时间戳事件，LXTAL 时钟阻塞
19	RTC 唤醒
20	CMP0 输出
21	CMP1 输出
22	CMP2 输出
23	CMP3 输出
24	CMP4 输出
25	CAN0 唤醒
26	CAN1 唤醒

EXTI 线编号	触发源
27	CAN2 唤醒
28	USART0 唤醒
29	USART1 唤醒
30	USART2 唤醒
31	I2C0 唤醒
32	I2C1 唤醒
33	I2C2 唤醒
34	I2C3 唤醒
35	LPTIM 唤醒
36	CMP5 输出
37	CMP6 输出
38	CMP7 输出

5.6. EXTI 寄存器

EXTI 基地址: 0x4001 0400

5.6.1. 中断使能寄存器 0（EXTI_INTEN0）

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
INTEN31	INTEN30	INTEN29	INTEN28	INTEN27	INTEN26	INTEN25	INTEN24	INTEN23	INTEN22	INTEN21	INTEN20	INTEN19	INTEN18	INTEN17	INTEN16
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
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
31:0	INTENx	中断使能位 x（x = 0…31） 0：第 x 线中断被禁用 1：第 x 线中断被使能

5.6.2. 事件使能寄存器 0（EXTI_EVEN0）

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
EVEN31	EVEN30	EVEN29	EVEN28	EVEN27	EVEN26	EVEN25	EVEN24	EVEN23	EVEN22	EVEN21	EVEN20	EVEN19	EVEN18	EVEN17	EVEN16
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
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
31:0	EVENx	事件使能位 x（x = 0…31） 0：第 x 线事件被禁用 1：第 x 线事件被使能

5.6.3. 上升沿触发使能寄存器 0（EXTI_RTEN0）

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
RTEN31	RTEN30	RTEN29	RTEN28	RTEN27	RTEN26	RTEN25	RTEN24	RTEN23	RTEN22	RTEN21	RTEN20	RTEN19	RTEN18	RTEN17	RTEN16
rw	rw	rw	rw	w	rw	rw	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
31:0	RTENx	上升沿触发使能位x（x = 0…31） 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）

5.6.4. 下降沿触发使能寄存器 0（EXTI_FTEN0）

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
FTEN31	FTEN30	FTEN29	FTEN28	FTEN27	FTEN26	FTEN25	FTEN24	FTEN23	FTEN12	FTEN21	FTEN20	FTEN19	FTEN18	FTEN17	FTEN16
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
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
31:0	FTENx	下降沿触发使能位 x（x = 0…31） 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）

5.6.5. 软件中断事件寄存器 0（EXTI_SWIEV0）

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
SWIEV31	SWIEV30	SWIEV29	SWIEV28	SWIEV27	SWIEV26	SWIEV25	SWIEV24	SWIEV23	SWIEV22	SWIEV21	SWIEV20	SWIEV19	SWIEV18	SWIEV17	SWIEV16
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
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
31:0	SWIEVx	中断/事件软件触发位 x（x = 0…31） 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

5.6.6. 挂起寄存器 0（EXTI_PD0）

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
PD31	PD30	PD29	PD28	PD27	PD26	PD25	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	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
31:0	PDx	中断挂起状态位 x（x = 0…31） 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。

5.6.7. 中断使能寄存器 1（EXTI_INTEN1）

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									INTEN38	INTEN37	INTEN36	INTEN35	INTEN34	INTEN33	INTEN32
									rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:6	保留	必须保持复位值。
5:0	INTENx	中断使能位 x（x = 32…38） 0：第 x 线中断被禁用 1：第 x 线中断被使能

5.6.8. 事件使能寄存器 1（EXTI_EVEN1）

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									EVEN38	EVEN37	EVEN36	EVEN35	EVEN34	EVEN33	EVEN32
									rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:6	保留	必须保持复位值。
5:0	EVENx	事件使能位 x（x = 32…38） 0：第 x 线事件被禁用 1：第 x 线事件被使能

5.6.9. 上升沿触发使能寄存器 1（EXTI_RTEN1）

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	RTEN38	RTEN37	RTEN36	RTEN35	RTEN34	RTEN33	RTEN32
	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:6	保留	必须保持复位值。
5:0	RTENx	上升沿触发使能位x（x = 32…38） 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）

5.6.10. 下降沿触发使能寄存器 1（EXTI_FTEN1）

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									FTEN38	FTEN37	FTEN36	FTEN35	FTEN34	FTEN33	FTEN32
									rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:6	保留	必须保持复位值。
5:0	FTENx	下降沿触发使能位 x（x = 32…38） 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）

5.6.11. 软件中断事件寄存器 1（EXTI_SWIEV1）

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									SWIEV38	SWIEV37	SWIEV36	SWIEV35	SWIEV34	SWIEV33	SWIEV32

rw	rw	rw	rw	rw	rw	rw	

位/位域	名称	描述
31:6	保留	必须保持复位值。
5:0	SWIEVx	中断/事件软件触发位 x（x = 32…38） 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

5.6.12. 挂起寄存器 1（EXTI_PD1）

地址偏移：0x2C

复位值：0x0000 00XX 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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									PD38	PD37	PD36	PD35	PD34	PD33	PD32
									rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位/位域	名称	描述
31:6	保留	必须保持复位值。
5:0	PDx	中断挂起状态位 x（x = 32…38） 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。

6. 触发选择控制器（TRIGSEL）

6.1. 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，有多达 243 个触发输入信号可选。每个外设对应独立的触发选择控制器，配置相应的触发选择寄存器，可以为外设的指定触发输入选择不同的触发信号。

6.2. 主要特征

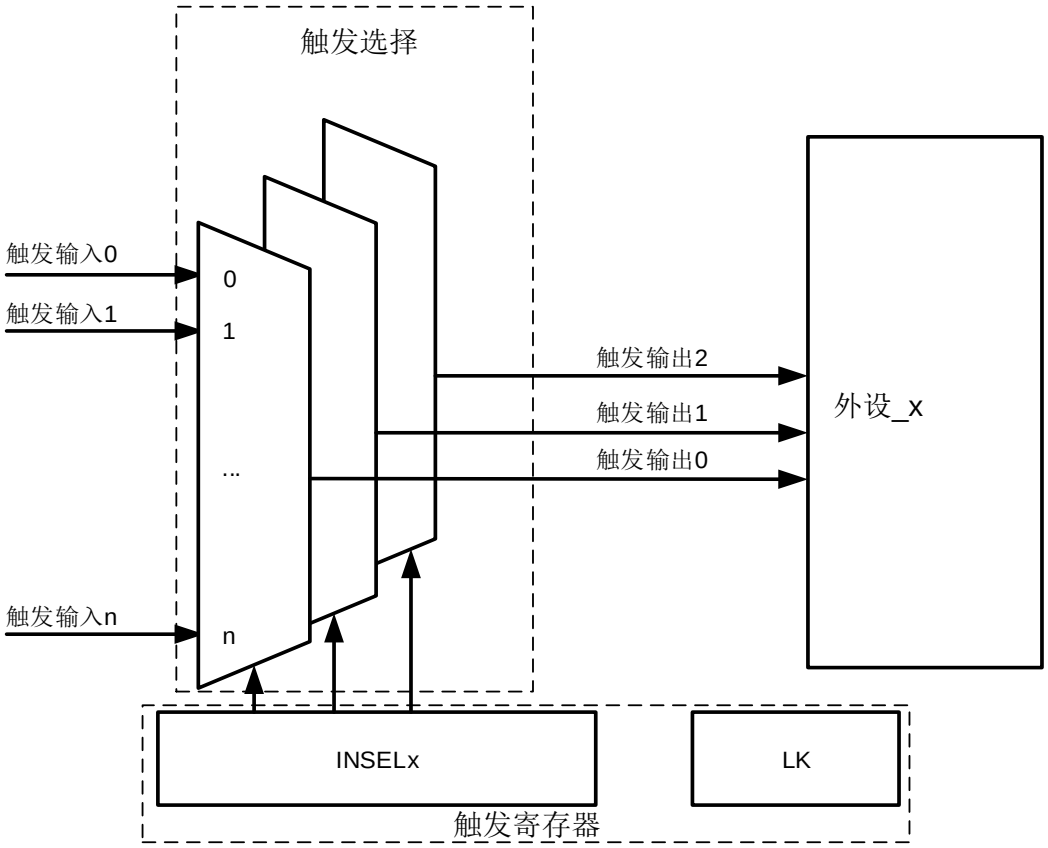
- 支持不同的可选输入触发；
- 每个外设都有专用的触发信号选择寄存器；
- 触发选择控制器的输入信号来源于外部输入或外设输出；
- 触发选择控制器的输出信号可输出到外部输出或者到外设输入。

6.3. 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3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6-1. TRIGSEL主要组成示例*显示了TRIGSEL的主要组成结构。

图 6-1. TRIGSEL 主要组成示例



6.4. 内部连接

TRIGSEL 允许软件方式为外设选择触发输入。*表 6-1. 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 6-1. 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

位域名称	位域值	触发输入选择
	0x09	TRIGSEL_IN7
	0x0a	TRIGSEL_IN8
	0x0b	TRIGSEL_IN9
	0x0c	TRIGSEL_IN10
	0x0d	TRIGSEL_IN11
	0x0e	TRIGSEL_IN12
	0x0f	TRIGSEL_IN13
	0x10	TIMER0_TRGO0
	0x11	TIMER0_TRGO1
	0x12	TIMER0_CH0
	0x13	TIMER0_CH1
	0x14	TIMER0_CH2
	0x15	TIMER0_CH3
	0x16	TIMER0_MCH0
	0x17	TIMER0_MCH1
	0x18	TIMER0_MCH2
	0x19	TIMER0_MCH3
	0x1a-0x1f	保留
	0x20	TIMER0_ETI
	0x21	TIMER1_TRGO0
	0x22	TIMER1_CH0
	0x23	TIMER1_CH1
	0x24	TIMER1_CH2
	0x25	TIMER1_CH3
	0x26	TIMER1_ETI
	0x27	TIMER2_TRGO0
	0x28	TIMER2_CH0
	0x29	TIMER2_CH1
	0x2a	TIMER2_CH2
	0x2b	TIMER2_CH3
	0x2c	TIMER2_ETI
	0x2d	TIMER3_TRGO0
	0x2e	TIMER3_CH0
	0x2f	TIMER3_CH1
	0x30	TIMER3_CH2
	0x31	TIMER3_CH3
	0x32	TIMER3_ETI

位域名称	位域值	触发输入选择
	0x33	TIMER4_TRGO0
	0x34	TIMER4_CH0
	0x35	TIMER4_CH1
	0x36	TIMER4_CH2
	0x37	TIMER4_CH3
	0x38	TIMER4_ETI
	0x39	TIMER5_TRGO0
	0x3a	TIMER6_TRGO0
	0x3b	TIMER7_TRGO0
	0x3c	TIMER7_TRGO1
	0x3d	TIMER7_CH0
	0x3e	TIMER7_CH1
	0x3f	TIMER7_CH2
	0x40	TIMER7_CH3
	0x41	TIMER7_MCH0
	0x42	TIMER7_MCH1
	0x43	TIMER7_MCH2
	0x44	TIMER7_MCH3
	0x45-0x4a	保留
	0x4b	TIMER7_ETI
	0x4c	TIMER14_TRGO0
	0x4d	TIMER14_CH0
	0x4e	TIMER14_CH1
	0x4f	TIMER14_MCH0
	0x50-0x52	保留
	0x53	TIMER15_CH0
	0x54	TIMER15_MCH0
	0x55-0x57	保留
	0x58	TIMER16_CH0
	0x59	TIMER16_MCH0
	0x5a-0x5b	保留
	0x5c	TIMER19_TRGO0
	0x5d	TIMER19_TRGO1
	0x5e	TIMER19_CH0
	0x5f	TIMER19_CH1
	0x60	TIMER19_CH2
	0x61	TIMER19_CH3

位域名称	位域值	触发输入选择
	0x62	TIMER19_MCH0
	0x63	TIMER19_MCH1
	0x64	TIMER19_MCH2
	0x65	TIMER19_MCH3
	0x66-0x6b	保留
	0x6c	TIMER19_ETI
	0x6d	TIMER0_BKIN0
	0x6e	TIMER0_BKIN1
	0x6f	TIMER0_BKIN2
	0x70	TIMER7_BKIN0
	0x71	TIMER7_BKIN1
	0x72	TIMER7_BKIN2
	0x73	TIMER14_BKIN0
	0x74	TIMER15_BKIN0
	0x75	TIMER16_BKIN0
	0x76	TIMER19_BKIN0
	0x77	TIMER19_BKIN1
	0x78	TIMER19_BKIN2
	0x79	LPTIMER_OUT
	0x7a	LPTIMER_ETI
	0x7b	HRTIMER_SCOUT
	0x7c	HRTIMER_SCIN
	0x7d	HRTIMER_ADCTRIG0
	0x7e	HRTIMER_ADCTRIG1
	0x7f	HRTIMER_ADCTRIG2
	0x80	HRTIMER_ADCTRIG3
	0x81	HRTIMER_ADCTRIG4
	0x82	HRTIMER_ADCTRIG5
	0x83	HRTIMER_ADCTRIG6
	0x84	HRTIMER_ADCTRIG7
	0x85	HRTIMER_ADCTRIG8
	0x86	HRTIMER_ADCTRIG9
	0x87	HRTIMER_DACTRIG0
	0x88	HRTIMER_DACTRIG1
	0x89	HRTIMER_DACTRIG2
	0x8a	HRTIMER_ST0_TRIG1
	0x8b	HRTIMER_ST1_TRIG1

位域名称	位域值	触发输入选择
	0x8c	HRTIMER_ST2_TRIG1
	0x8d	HRTIMER_ST3_TRIG1
	0x8e	HRTIMER_ST4_TRIG1
	0x8f	HRTIMER_ST5_TRIG1
	0x90	HRTIMER_ST6_TRIG1
	0x91	HRTIMER_ST7_TRIG1
	0x92~0x95	保留
	0x96	HRTIMER_SYS_FLT
	0x97	ADC0_WD0_OUT
	0x98	ADC0_WD1_OUT
	0x99	ADC0_WD2_OUT
	0x9a	ADC1_WD0_OUT
	0x9b	ADC1_WD1_OUT
	0x9c	ADC1_WD2_OUT
	0x9d	ADC2_WD0_OUT
	0x9e	ADC2_WD1_OUT
	0x9f	ADC2_WD2_OUT
	0xa0	ADC3_WD0_OUT
	0xa1	ADC3_WD1_OUT
	0xa2	ADC3_WD2_OUT
	0xa3	HXTAL_DIV32_TRIG
	0xa4	IRC32K_TRIG
	0xa5	LXTAL_TRIG
	0xa6	CKOUT_TRIG
	0xa7	EXTI2_TRIG
	0xa8	EXTI3_TRIG
	0xa9	EXTI9_TRIG
	0xaa	EXTI10_TRIG
	0xab	EXTI11_TRIG
	0xac	EXTI15_TRIG
	0xad	RTC_WAKEUP
	0xae	RTC_ALARM0
	0xaf	RTC_ALARM1
	0xb0	RTC_TAMP0
	0xb1	RTC_TAMP1
	0xb2	RTC_TAMP2
	0xb3	CMP0_OUT

位域名称	位域值	触发输入选择
	0xb4	CMP1_OUT
	0xb5	CMP2_OUT
	0xb6	CMP3_OUT
	0xb7	CMP4_OUT
	0xb8	CMP5_OUT
	0xb9	CMP6_OUT
	0xba	CMP7_OUT
	0xbb	CLA0_OUT
	0xbc	CLA1_OUT
	0xbd	CLA2_OUT
	0xbe	CLA3_OUT
	0xbf	HRTIMER_ST0_TRIG0
	0xc0	HRTIMER_ST1_TRIG0
	0xc1	HRTIMER_ST2_TRIG0
	0xc2	HRTIMER_ST3_TRIG0
	0xc3	HRTIMER_ST4_TRIG0
	0xc4	HRTIMER_ST5_TRIG0
	0xc5	HRTIMER_ST6_TRIG0
	0xc6	HRTIMER_ST7_TRIG0
	0xc7~0xde	保留
	0xdf	HRTIMER_ST0_CH0
	0xe0	HRTIMER_ST0_CH1
	0xe1	HRTIMER_ST1_CH0
	0xe2	HRTIMER_ST1_CH1
	0xe3	HRTIMER_ST2_CH0
	0xe4	HRTIMER_ST2_CH1
	0xe5	HRTIMER_ST3_CH0
	0xe6	HRTIMER_ST3_CH1
	0xe7	HRTIMER_ST4_CH0
	0xe8	HRTIMER_ST4_CH1
	0xe9	HRTIMER_ST5_CH0
	0xea	HRTIMER_ST5_CH1
	0xeb	HRTIMER_ST6_CH0
	0xec	HRTIMER_ST6_CH1
	0xed	HRTIMER_ST7_CH0
	0xee	HRTIMER_ST7_CH1
	0xef	ADC0_CONV

位域名称	位域值	触发输入选择
	0xf0	ADC1_CONV
	0xf1	ADC2_CONV
	0xf2	ADC3_CONV
	0xf3	LCKM_OUT
	0xf4~0xff	保留

如*表 6-2. 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[7:0]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 3 路输出，这些输出连接到对应的外设。

表 6-2. 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
0 1 TRIGSEL_IN0 TRIGSEL_IN1 TRIGSEL_IN2 TRIGSEL_IN3 TRIGSEL_IN4 TRIGSEL_IN5 TRIGSEL_IN6 TRIGSEL_IN7 TRIGSEL_IN8 TRIGSEL_IN9 TRIGSEL_IN10 TRIGSEL_IN11 TRIGSEL_IN12 TRIGSEL_IN13 TIMER0_TRGO0 TIMER0_TRGO1 TIMER0_CH0 TIMER0_CH1 TIMER0_CH2 TIMER0_CH3 TIMER0_MCH0 TIMER0_MCH1	INSELx[7:0]	TRIGSEL_EXTOUT_0	Output0 Output1	TRIGSEL_OUT0 TRIGSEL_OUT1
		TRIGSEL_EXTOUT_1	Output0 Output1	TRIGSEL_OUT2 TRIGSEL_OUT3
		TRIGSEL_EXTOUT_2	Output0 Output1	TRIGSEL_OUT4 TRIGSEL_OUT5
		TRIGSEL_EXTOUT_3	Output0 Output1	TRIGSEL_OUT6 TRIGSEL_OUT7
		TRIGSEL_ADC0	Output0	ADC0_ROUTRG
		TRIGSEL_ADC1	Output0	ADC1_ROUTRG

触发源	触发选择	TRIGSEL 寄存器	TRIGSEL 输出	外设
TIMER0_MCH2 TIMER0_MCH3 TIMER0_ETI TIMER1_TRGO0 TIMER1_CH0 TIMER1_CH1 TIMER1_CH2 TIMER1_CH3 TIMER1_ETI TIMER2_TRGO0 TIMER2_CH0 TIMER2_CH1 TIMER2_CH2 TIMER2_CH3 TIMER2_ETI TIMER3_TRGO0 TIMER3_CH0 TIMER3_CH1 TIMER3_CH2 TIMER3_CH3 TIMER3_ETI TIMER4_TRGO0 TIMER4_CH0 TIMER4_CH1 TIMER4_CH2 TIMER4_CH3 TIMER4_ETI TIMER5_TRGO0 TIMER6_TRGO0 TIMER7_TRGO0 TIMER7_TRGO1 TIMER7_CH0 TIMER7_CH1 TIMER7_CH2 TIMER7_CH3 TIMER7_MCH0		TRIGSEL_ADC2	Output0	ADC2_ROUTRG
		TRIGSEL_ADC3	Output0	ADC3_ROUTRG
		TRIGSEL_TIMER0BRKIN	Output0 Output1 Output2	TIMER0_BRKIN0 TIMER0_BRKIN1 TIMER0_BRKIN2
		TRIGSEL_TIMER7BRKIN	Output0 Output1 Output2	TIMER7_BRKIN0 TIMER7_BRKIN1 TIMER7_BRKIN2
		TRIGSEL_TIMER14BRKIN	Output0	TIMER14_BRKIN0
		TRIGSEL_TIMER15BRKIN	Output0	TIMER15_BRKIN0
		TRIGSEL_TIMER16BRKIN	Output0	TIMER16_BRKIN0
		TRIGSEL_TIMER19BRKIN	Output0 Output1 Output2	TIMER19_BRKIN0 TIMER19_BRKIN1 TIMER19_BRKIN2
		TRIGSEL_CAN0	Output0	CAN0_EX_TIME_TICK

触发源	触发选择	TRIGSEL 寄存器	TRIGSEL 输出	外设
TIMER7_MCH1 TIMER7_MCH2 TIMER7_MCH3 TIMER7_ETI TIMER14_TRGO0 TIMER14_CH0 TIMER14_CH1 TIMER14_MCH0 TIMER15_CH0 TIMER15_MCH0 TIMER16_CH0 TIMER16_MCH0 TIMER19_TRGO0 TIMER19_TRGO1 TIMER19_CH0 TIMER19_CH1 TIMER19_CH2 TIMER19_CH3 TIMER19_MCH0 TIMER19_MCH1 TIMER19_MCH2 TIMER19_MCH3 TIMER19_ETI TIMER0_BKIN0 TIMER0_BKIN1 TIMER0_BKIN2 TIMER7_BKIN0 TIMER7_BKIN1 TIMER7_BKIN2 TIMER14_BKIN0 TIMER15_BKIN0 TIMER16_BKIN0 TIMER19_BKIN0 TIMER19_BKIN1 TIMER19_BKIN2 LPTIMER_OUT		TRIGSEL_CAN1	Output0	CAN1_EX_TIME_TICK
		TRIGSEL_CAN2	Output0	CAN2_EX_TIME_TICK
		TRIGSEL_TIMER0_ETI	Output0	TIMER0_ETI
		TRIGSEL_TIMER1_ETI	Output0	TIMER1_ETI
		TRIGSEL_TIMER2_ETI	Output0	TIMER2_ETI
		TRIGSEL_TIMER3_ETI	Output0	TIMER3_ETI
		TRIGSEL_TIMER4_ETI	Output0	TIMER4_ETI
		TRIGSEL_TIMER7_ETI	Output0	TIMER7_ETI
		TRIGSEL_TIMER19_ETI	Output0	TIMER19_ETI

触发源	触发选择	TRIGSEL 寄存器	TRIGSEL 输出	外设
LPTIMER_ETI HRTIMER_SCOUT HRTIMER_SCIN HRTIMER_ADCTRIG0 HRTIMER_ADCTRIG1 HRTIMER_ADCTRIG2 HRTIMER_ADCTRIG3 HRTIMER_ADCTRIG4 HRTIMER_ADCTRIG5 HRTIMER_ADCTRIG6 HRTIMER_ADCTRIG7 HRTIMER_ADCTRIG8 HRTIMER_ADCTRIG9 HRTIMER_DACTRIG0 HRTIMER_DACTRIG1 HRTIMER_DACTRIG2 HRTIMER_ST0_TRIG1 HRTIMER_ST1_TRIG1 HRTIMER_ST2_TRIG1 HRTIMER_ST3_TRIG1 HRTIMER_ST4_TRIG1 HRTIMER_ST5_TRIG1 HRTIMER_ST6_TRIG1 HRTIMER_ST7_TRIG1 HRTIMER_SYS_FLT ADC0_WD0_OUT ADC0_WD1_OUT ADC0_WD2_OUT ADC1_WD0_OUT ADC1_WD1_OUT ADC1_WD2_OUT ADC2_WD0_OUT ADC2_WD1_OUT ADC2_WD2_OUT ADC3_WD0_OUT ADC3_WD1_OUT ADC3_WD2_OUT		TRIGSEL_HPDF	Output0	HPDF_ITRG
		TRIGSEL_TIMER0ITI14	Output0	TIMER0_ITI14
		TRIGSEL_TIMER1ITI14	Output0	TIMER1_ITI14
		TRIGSEL_TIMER2ITI14	Output0	TIMER2_ITI14
		TRIGSEL_TIMER3ITI14	Output0	TIMER3_ITI14
		TRIGSEL_TIMER4ITI14	Output0	TIMER4_ITI14
		TRIGSEL_TIMER7ITI14	Output0	TIMER7_ITI14
		TRIGSEL_TIMER14ITI14	Output0	TIMER14_ITI14
		TRIGSEL_TIMER19ITI14	Output0	TIMER19_ITI14

触发源	触发选择	TRIGSEL 寄存器	TRIGSEL 输出	外设
HXTAL_DIV32_TRIG IRC32K_TRIG LXTAL_TRIG CKOUT_TRIG EXTI2_TRIG EXTI3_TRIG EXTI9_TRIG EXTI10_TRIG EXTI11_TRIG EXTI15_TRIG RTC_WAKEUP RTC_ALARM0 RTC_ALARM1 RTC_TAMP0 RTC_TAMP1 RTC_TAMP2 CMP0_OUT CMP1_OUT CMP2_OUT CMP3_OUT CMP4_OUT CMP5_OUT CMP6_OUT CMP7_OUT CLA0_OUT CLA1_OUT CLA2_OUT CLA3_OUT HRTIMER_ST0_TRIG0 HRTIMER_ST1_TRIG0 HRTIMER_ST2_TRIG0 HRTIMER_ST3_TRIG0 HRTIMER_ST4_TRIG0 HRTIMER_ST5_TRIG0 HRTIMER_ST6_TRIG0 HRTIMER_ST7_TRIG0 HRTIMER_ST0_CH0		TRIGSEL_DAC0	Output0 Output1	DAC0_EXTRIG DAC0_EXTRIG
		TRIGSEL_DAC1	Output0 Output1	DAC1_EXTRIG DAC1_EXTRIG
		TRIGSEL_DAC2	Output0 Output1	DAC2_EXTRIG DAC2_EXTRIG
		TRIGSEL_DAC3	Output0 Output1	DAC3_EXTRIG DAC3_EXTRIG
		TRIGSEL_EXTDAC0	Output0 Output1	DAC0_ST_EXTRIG DAC0_ST_EXTRIG
		TRIGSEL_EXTDAC1	Output0 Output1	DAC1_ST_EXTRIG DAC1_ST_EXTRIG
		TRIGSEL_EXTDAC2	Output0 Output1	DAC2_ST_EXTRIG DAC2_ST_EXTRIG
		TRIGSEL_EXTDAC3	Output0 Output1	DAC3_ST_EXTRIG DAC3_ST_EXTRIG
		TRIGSEL_CLA_0	Output0 Output1 Output2	TRIGSEL_CLA_IN0 TRIGSEL_CLA_IN1 TRIGSEL_CLA_IN2

触发源	触发选择	TRIGSEL 寄存器	TRIGSEL 输出	外设
HRTIMER_ST0_CH1 HRTIMER_ST1_CH0 HRTIMER_ST1_CH1 HRTIMER_ST2_CH0 HRTIMER_ST2_CH1 HRTIMER_ST3_CH0 HRTIMER_ST3_CH1 HRTIMER_ST4_CH0 HRTIMER_ST4_CH1 HRTIMER_ST5_CH0 HRTIMER_ST5_CH1 HRTIMER_ST6_CH0 HRTIMER_ST6_CH1 HRTIMER_ST7_CH0 HRTIMER_ST7_CH1 ADC0_CONV ADC1_CONV ADC2_CONV ADC3_CONV LCKM_OUT		TRIGSEL_CLA_1	Output0 Output1 Output2	TRIGSEL_CLA_IN3 TRIGSEL_CLA_IN4 TRIGSEL_CLA_IN5
		TRIGSEL_CLA_2	Output0 Output1 Output2	TRIGSEL_CLA_IN6 TRIGSEL_CLA_IN7 TRIGSEL_CLA_IN8
		TRIGSEL_CLA_3	Output0 Output1	TRIGSEL_CLA_IN9 TRIGSEL_CLA_IN10
		TRIGSEL_CLA_4	Output0	TRIGSEL_CLA_IN11

注意：除了 TIMERx_BRKINy，TIMERx_ETI，TIMERx_ITI14，CLA_IN_x，LCKM_OUT 和 ADCx_CONV 外，其他所有的输出都可以选择所有输入作为触发源。

1. TIMERx_BRKINy只能选择TIMERx_BRKINy作为触发。
2. TIMERx_BRKINy只能被TIMERy_BRKINx和EXTOUT选择为触发。
3. TIMERx_ETI不能选择TIMERx_BRKINy和CKOUT_TRIG作为触发。
4. TIMERx_ITI14不能选择HRTIMER和他们自己的信号作为触发。
5. CLA_IN_x（x = 0,..,5）只能选择HRTIMER_STx_CHy，TIMERx_CHy和CMPx_OUT作为触发。
6. HRTIMER_STx_CHy只能被CLA_IN_x（x = 0,..,5）和EXTOUT选择为触发。
7. CLA_IN_x（x = 6,..,10）只能选择TIMERx_TRGOy作为触发。
8. CLA_IN_11只能选择ADCx_CONV作为触发。
9. LCKM_OUT只能被TIMERx_ETI和EXTOUT选择为触发。
10. ADCx_CONV只能被CLA_IN_11和EXTOUT选择为触发。

触发输入选择 INSELx[7:0]位域值配置为 0x00 时，TRIGSEL 触发输入选择为低电平；配置为 0x01 时，触发输入选择为高电平。

当选择了非法触发输入时，其输出将被强制选择为 0。

6.5. TRIGSEL 寄存器

TRIGSEL 基地址：0x4001 8400

6.5.1. EXTOUT 触发选择寄存器 0（TRIGSEL_EXTOUT_0）

地址偏移：0x00
复位值：0x0000 0000

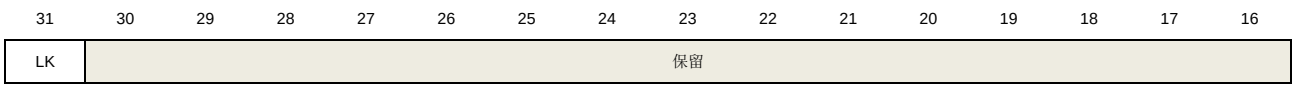
该寄存器只能按字（32 位）访问。

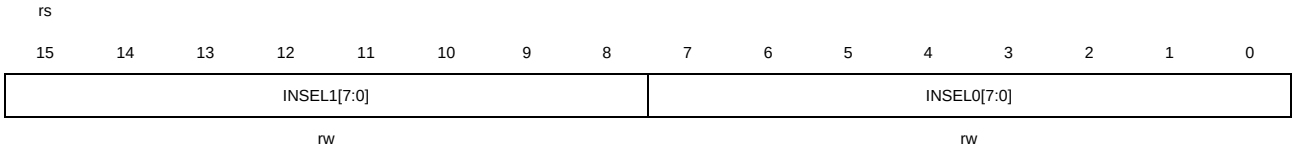
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_EXTOUT_0将不能被修改。 0：TRIGSEL_EXTOUT_0 寄存器可写可读 1：TRIGSEL_EXTOUT_0 寄存器只读
30:16	保留	必须保持复位值
15:8	INSEL1[7:0]	触发输出 1 的输入源选择 这些位用来选择连接到输出 1 的触发输入信号，输出 1 作为 TRIGSEL_OUT1（外部输出 1）的信号源。关于具体配置请参考***表 6-1. 触发输入位域选择***。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 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 TRIGSEL_OUT0（外部输出 0）的信号源。关于具体配置请参考***表 6-1. 触发输入位域选择***。

6.5.2. EXTOUT 触发选择寄存器 1（TRIGSEL_EXTOUT_1）

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。





位/位域	名称	描述
31	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_EXTOUT_1将不能被修改。 0：TRIGSEL_EXTOUT_1 寄存器可写可读 1：TRIGSEL_EXTOUT_1 寄存器只读
30:16	保留	必须保持复位值
15:8	INSEL1[7:0]	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 TRIGSEL_OUT3（外部输出 3）的信号源。关于具体配置请参考***表 6-1. 触发输入位域选择***。
7:0	INSEL0[7:0]	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 TRIGSEL_OUT2（外部输出 2）的信号源。关于具体配置请参考***表 6-1. 触发输入位域选择***。

6.5.3. EXTOUT 触发选择寄存器 2（TRIGSEL_EXTOUT_2）

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

位/位域	名称	描述
31	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_EXTOUT_2将不能被修改。 0：TRIGSEL_EXTOUT_2 寄存器可写可读 1：TRIGSEL_EXTOUT_2 寄存器只读

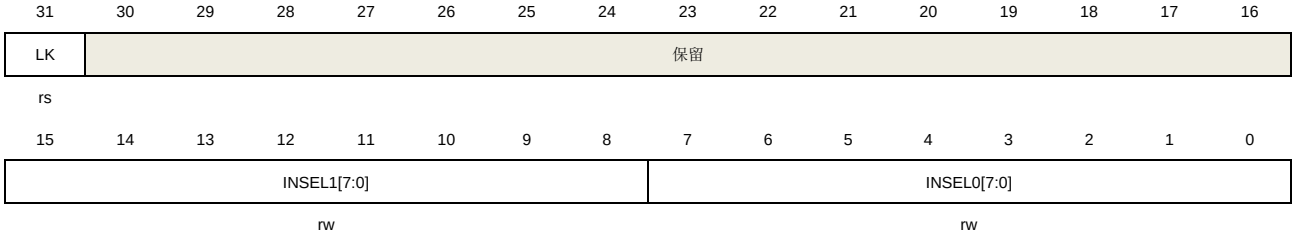
30:16	保留	必须保持复位值
15:8	INSEL1[7:0]	触发输出 1 的输入源选择 这些位用来选择连接到输出 1 的触发输入信号，输出 1 作为 TRIGSEL_OUT5（外部输出 5）的信号源。关于具体配置请参考*表 6-1. 触发输入位域选择*。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TRIGSEL_OUT4（外部输出 4）的信号源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.4. EXTOUT3 触发选择寄存器 3（TRIGSEL_EXTOUT_3）

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



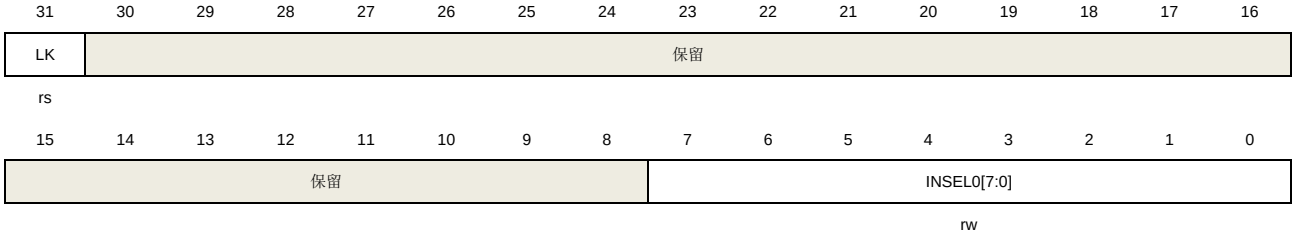
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_EXTOUT_3将不能被修改。 0：TRIGSEL_EXTOUT_3 寄存器可写可读 1：TRIGSEL_EXTOUT_3 寄存器只读
30:16	保留	必须保持复位值
15:8	INSEL1[7:0]	触发输出 1 的输入源选择 这些位用来选择连接到输出 1 的触发输入信号，输出 1 作为 TRIGSEL_OUT7（外部输出 7）的信号源。关于具体配置请参考*表 6-1. 触发输入位域选择*。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TRIGSEL_OUT6（外部输出 6）的信号源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.5. ADC0 触发选择寄存器（TRIGSEL_ADC0）

地址偏移：0x10

复位值：0x0000 1012

该寄存器只能按字（32 位）访问。



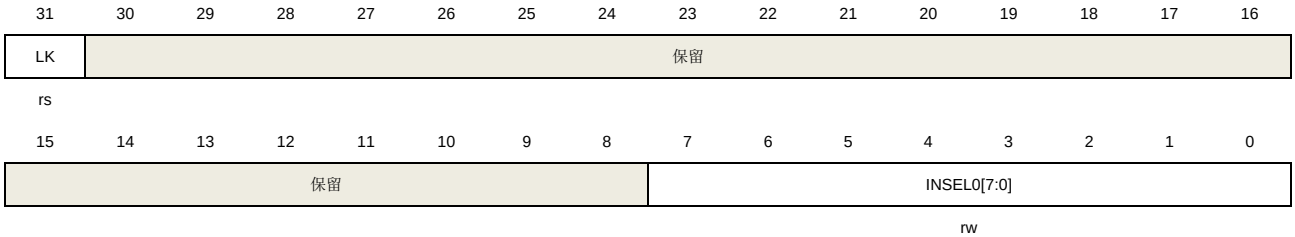
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_ADC0将不能被修改。 0：TRIGSEL_ADC0 寄存器可写可读 1：TRIGSEL_ADC0 寄存器只读
30:16	保留	必须保持复位值
15:8	保留	必须保持复位值
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 ADC0_ROUTRG（ADC0 常规序列）的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.6. ADC1 触发选择寄存器（TRGSEL_ADC1）

地址偏移：0x14

复位值：0x0000 1012

该寄存器只能按字（32 位）访问。



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 寄存器可写可读 1：TRIGSEL_ADC1 寄存器只读
30:16	保留	必须保持复位值
15:8	保留	必须保持复位值
7:0	INSEL0[7:0]	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 ADC1_ROUTRG（ADC1 常规序列）的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.7. ADC2 触发选择寄存器（TRIGSEL_ADC2）

地址偏移：0x18

复位值：0x0000 1028

该寄存器只能按字（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[7:0]							
								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_ADC2将不能被修改。 0：TRIGSEL_ADC2 寄存器可写可读 1：TRIGSEL_ADC2 寄存器只读
30:16	保留	必须保持复位值
15:8	保留	必须保持复位值
7:0	INSEL0[7:0]	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 ADC2_ROUTRG（ADC2 常规序列）的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.8. ADC3 触发选择寄存器（TRIGSEL_ADC3）

地址偏移：0x1C

复位值：0x0000 1028

该寄存器只能按字（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[7:0]							
								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_ADC3将不能被修改。 0：TRIGSEL_ADC3 寄存器可写可读 1：TRIGSEL_ADC3 寄存器只读
30:16	保留	必须保持复位值
15:8	保留	必须保持复位值
7:0	INSEL0[7:0]	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 ADC3_ROUTRG（ADC3 常规序列）的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

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

地址偏移：0x2C

复位值：0x006F 616D

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	保留							INSEL2[7:0]							
rs								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INSEL1[7:0]								INSEL0[7:0]							
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_TIMER0BRIKIN将不能被修改。 0：TRIGSEL_TIMER0BRIKIN 寄存器可写可读

		1：TRIGSEL_TIMER0BRIKIN 寄存器只读
30:24	保留	必须保持复位值
23:16	INSEL2[7:0]	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 TIMER0BRKIN2 的触发源。关于具体配置请参考***表 6-1. 触发输入位域选择***。
15:8	INSEL1[7:0]	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 TIMER0BRKIN1 的触发源。关于具体配置请参考***表 6-1. 触发输入位域选择***。
7:0	INSEL0[7:0]	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 TIMER0BRKIN0 的触发源。关于具体配置请参考***表 6-1. 触发输入位域选择***。

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

地址偏移：0x30

复位值：0x0072 7170

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	保留							INSEL2[7:0]							
rs								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INSEL1[7:0]								INSEL0[7:0]							
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_TIMER7BRIKIN将不能被修改。 0：TRIGSEL_TIMER7BRIKIN 寄存器可写可读 1：TRIGSEL_TIMER7BRIKIN 寄存器只读
30:24	保留	必须保持复位值
23:16	INSEL2[7:0]	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 TIMER7BRKIN2 的触发源。关于具体配置请参考***表 6-1. 触发输入位域选择***。
15:8	INSEL1[7:0]	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 TIMER7BRKIN1 的触发

源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

7:0	INSEL0[7:0]	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 TIMER7BRKIN0 的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.11. TIMER14_BRKIN 触发选择寄存器（TRIGSEL_TIMER14BRKIN）

地址偏移：0x34

复位值：0x0000 0073

该寄存器只能按字（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[7:0]							
								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_TIMER14BRIKIN 将不能被修改。 0：TRIGSEL_TIMER14BRIKIN 寄存器可写可读 1：TRIGSEL_TIMER14BRIKIN 寄存器只读
30:8	保留	必须保持复位值
7:0	INSEL0[7:0]	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 TIMER14BRKIN0 的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.12. TIMER15_BRKIN 触发选择寄存器（TRIGSEL_TIMER15BRKIN）

地址偏移：0x38

复位值：0x0000 0074

该寄存器只能按字（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[7:0]
	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_TIMER15BRIKIN 将不能被修改。 0：TRIGSEL_TIMER15BRIKIN 寄存器可写可读 1：TRIGSEL_TIMER15BRIKIN 寄存器只读
30:8	保留	必须保持复位值
7:0	INSEL0[7:0]	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 TIMER15BRKIN0 的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.13. TIMER16_BRKIN 触发寄存器（TRIGSEL_TIMER16BRKIN）

地址偏移：0x3C
复位值：0x0000 0075

该寄存器只能按字（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[7:0]							
								rw							

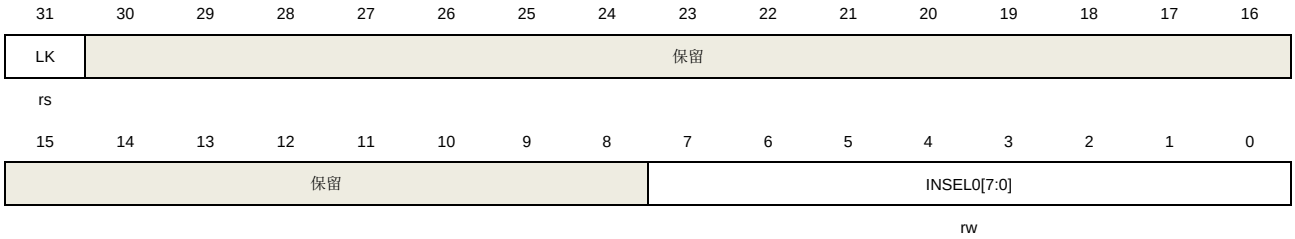
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER16BRIKIN 将不能被修改。 0：TRIGSEL_TIMER16BRIKIN 寄存器可写可读 1：TRIGSEL_TIMER16BRIKIN 寄存器只读
30:8	保留	必须保持复位值
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER16BRKIN0 的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

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

地址偏移：0x40

复位值：0x0078 7776

该寄存器只能按字（32 位）访问。



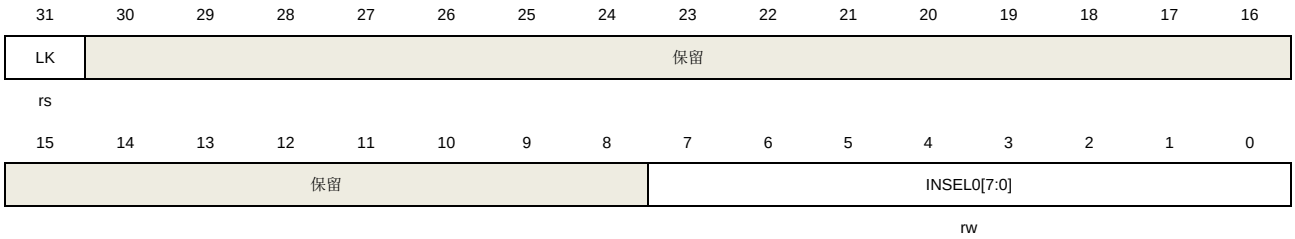
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER19BRIKIN 将不能被修改。 0：TRIGSEL_TIMER19BRIKIN 寄存器可写可读 1：TRIGSEL_TIMER19BRIKIN 寄存器只读
30:8	保留	必须保持复位值
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER19BRKIN0 的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.15. CAN0 触发选择寄存器（TRIGSEL_CAN0）

地址偏移：0x44

复位值：0x0000 0039

该寄存器只能按字（32 位）访问。



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 寄存器可写可读 1：TRIGSEL_CAN0 寄存器只读
30:8	保留	必须保持复位值
7:0	INSEL0[7:0]	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 CAN0_EX_TIME_TICK 的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.16. CAN1 触发选择寄存器（TRIGSEL_CAN1）

地址偏移：0x48
复位值：0x0000 0039

该寄存器只能按字（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[7:0]							
								rw							

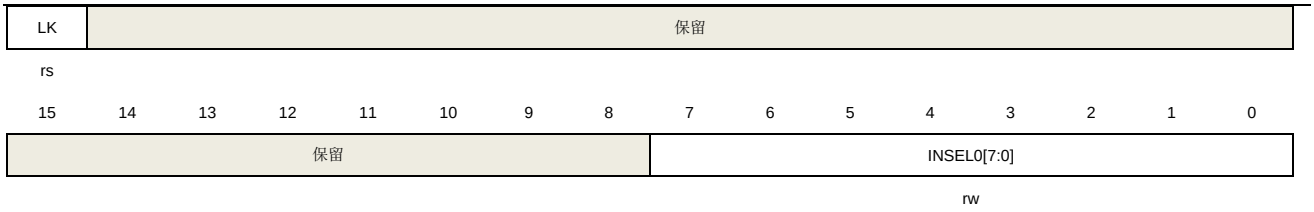
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_CAN1将不能被修改。 0：TRIGSEL_CAN1 寄存器可写可读 1：TRIGSEL_CAN1 寄存器只读
30:8	保留	必须保持复位值
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 CAN1_EX_TIME_TICK 的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.17. CAN2 触发选择寄存器（TRIGSEL_CAN2）

地址偏移：0x4C
复位值：0x0000 0039

该寄存器只能按字（32 位）访问。

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



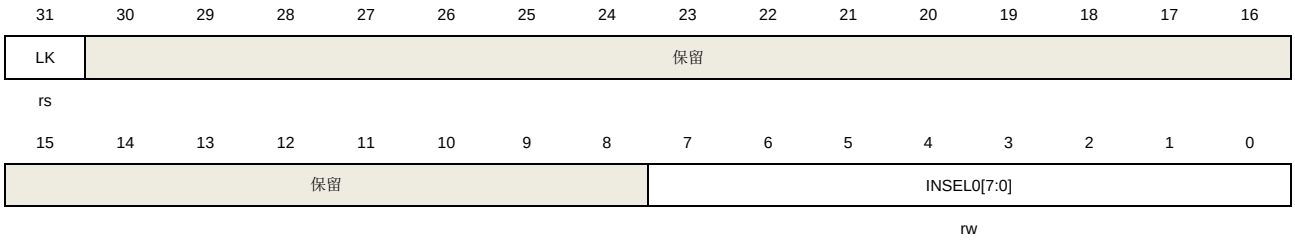
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_CAN2将不能被修改。 0：TRIGSEL_CAN2 寄存器可写可读 1：TRIGSEL_CAN2 寄存器只读
30:8	保留	必须保持复位值
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 CAN2_EX_TIME_TICK 的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.18. TIMER0_ETI 触发选择寄存器（TRIGSEL_TIMER0ETI）

地址偏移：0x50

复位值：0x0000 0020

该寄存器只能按字（32 位）访问。



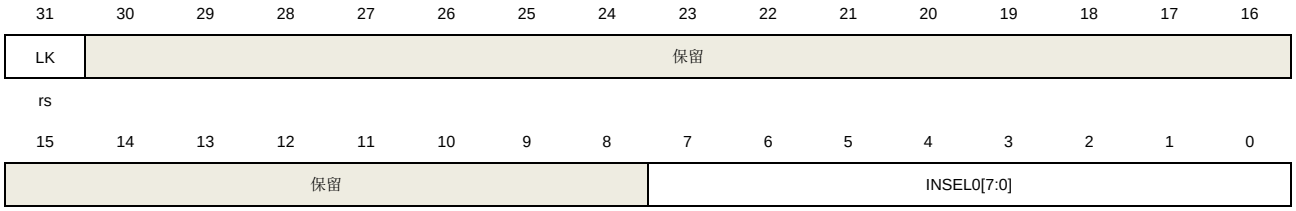
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER0ETI将不能被修改。 0：TRIGSEL_TIMER0ETI 寄存器可写可读 1：TRIGSEL_TIMER0ETI 寄存器只读
30:8	保留	必须保持复位值
7:0	INSEL0[7:0]	触发输出 0 的输入源选择

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_ETI 的触发源。
关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.19. TIMER1_ETI 触发选择寄存器（TRIGSEL_TIMER1ETI）

地址偏移：0x54
复位值：0x0000 0026

该寄存器只能按字（32 位）访问。

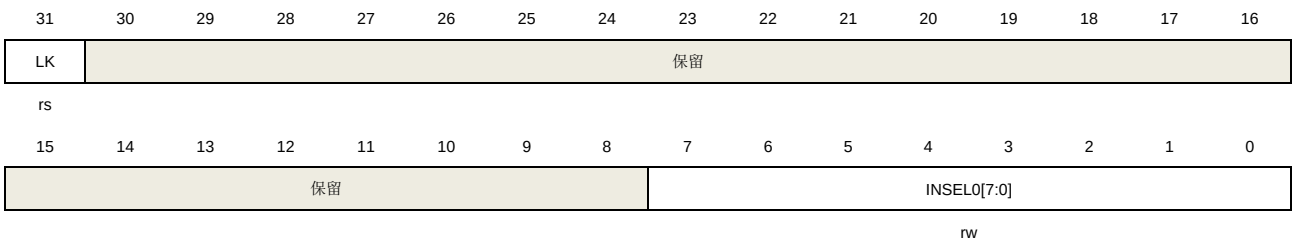


位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER1ETI将不能被修改。 0：TRIGSEL_TIMER1ETI 寄存器可写可读 1：TRIGSEL_TIMER1ETI 寄存器只读
30:8	保留	必须保持复位值
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER1_ETI 的触发源。 关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.20. TIMER2_ETI 触发选择寄存器（TRIGSEL_TIMER2ETI）

地址偏移：0x58
复位值：0x0000 002C

该寄存器只能按字（32 位）访问。



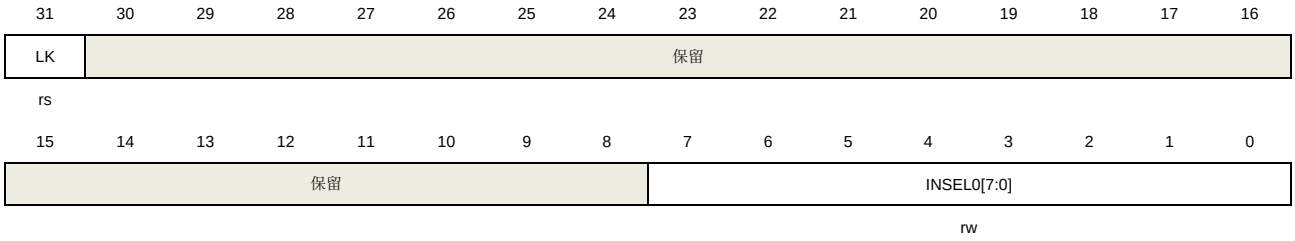
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER2ETI将不能被修改。 0：TRIGSEL_TIMER2ETI 寄存器可写可读 1：TRIGSEL_TIMER2ETI 寄存器只读
30:8	保留	必须保持复位值
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER2_ETI 的触发源。 关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.21. TIMER3_ETI 触发选择寄存器（TRIGSEL_TIMER3ETI）

地址偏移：0x5C

复位值：0x0000 0032

该寄存器只能按字（32 位）访问。



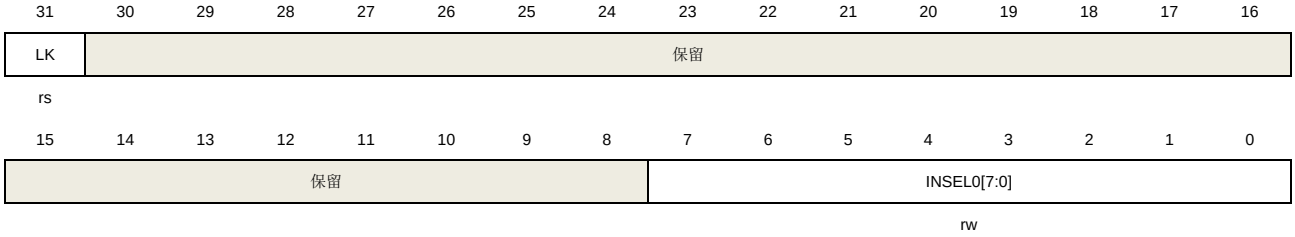
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER3ETI将不能被修改。 0：TRIGSEL_TIMER3ETI 寄存器可写可读 1：TRIGSEL_TIMER3ETI 寄存器只读
30:8	保留	必须保持复位值
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER3_ETI 的触发源。 关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.22. TIMER4_ETI 触发选择寄存器（TRIGSEL_TIMER4ETI）

地址偏移：0x60

复位值：0x0000 0038

该寄存器只能按字（32 位）访问。



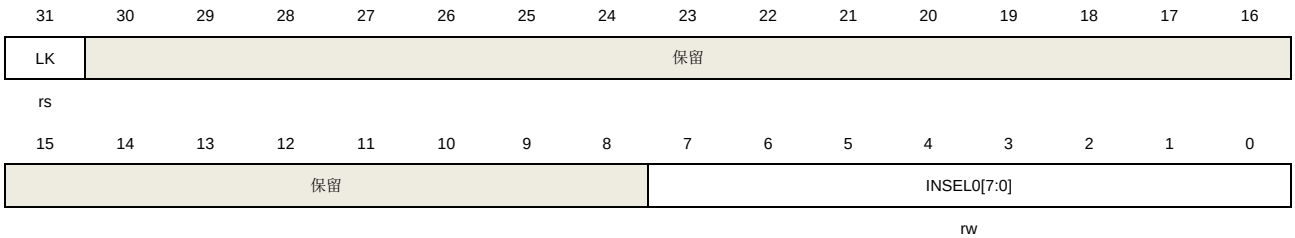
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER4ETI将不能被修改。 0：TRIGSEL_TIMER4ETI 寄存器可写可读 1：TRIGSEL_TIMER4ETI 寄存器只读
30:8	保留	必须保持复位值
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER4_ETI 的触发源。 关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.23. TIMER7_ETI 触发选择寄存器（TRIGSEL_TIMER7ETI）

地址偏移：0x64

复位值：0x0000 004B

该寄存器只能按字（32 位）访问。



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_TIMER7ETI将不能被修改。 0：TRIGSEL_TIMER7ETI 寄存器可写可读

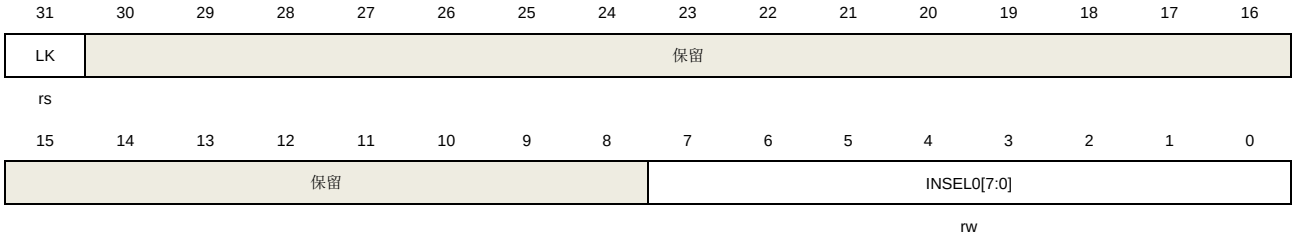
		1：TRIGSEL_TIMER7ETI 寄存器只读
30:8	保留	必须保持复位值
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER7_ETI 的触发源。 关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.24. TIMER19_ETI 触发选择寄存器（TRIGSEL_TIMER19ETI）

地址偏移：0x68

复位值：0x0000 006C

该寄存器只能按字（32 位）访问。



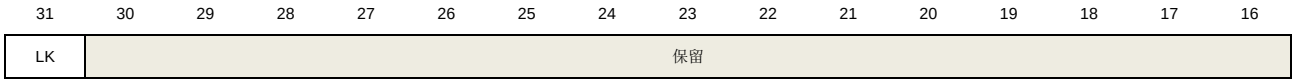
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER19ETI将不能被修改。 0：TRIGSEL_TIMER19ETI 寄存器可写可读 1：TRIGSEL_TIMER0ETI 寄存器只读
30:8	保留	必须保持复位值
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER19_ETI 的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.25. HPDF_ITRG 触发选择控制寄存器（TRIGSEL_HPDF）

地址偏移：0x6C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



rs

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								INSEL0[7:0]							
								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_HPDF将不能被修改。 0：TRIGSEL_HPDF 寄存器可写可读 1：TRIGSEL_HPDF 寄存器只读
30:8	保留	必须保持复位值
7:0	INSEL0[7:0]	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 HPDF_ITRG 的触发源。 关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.26. TIMER0_ITI14 触发选择寄存器（TRIGSEL_TIMER0ITI14）

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
LK	保留														
rs															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								INSEL0[7:0]							
								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_TIMER0ITI14将不能被修改。 0：TRIGSEL_TIMER0ITI14 寄存器可写可读 1：TRIGSEL_TIMER0ITI14 寄存器只读
30:8	保留	必须保持复位值
7:0	INSEL0[7:0]	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_ITI14 的触发

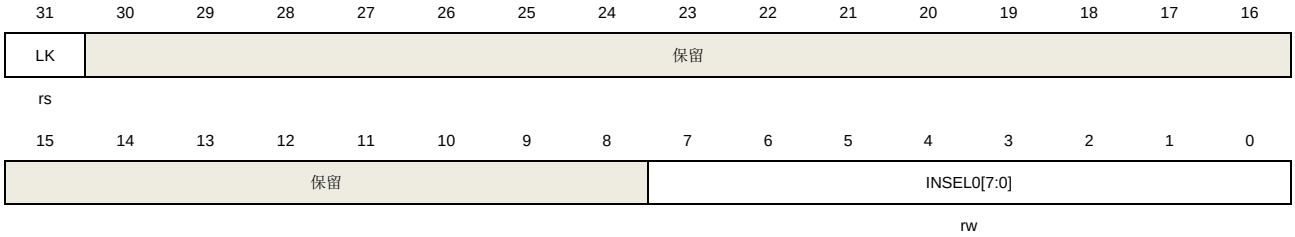
源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.27. TIMER1_ITI14 触发选择寄存器（TRIGSEL_TIMER1ITI14）

地址偏移：0x74

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



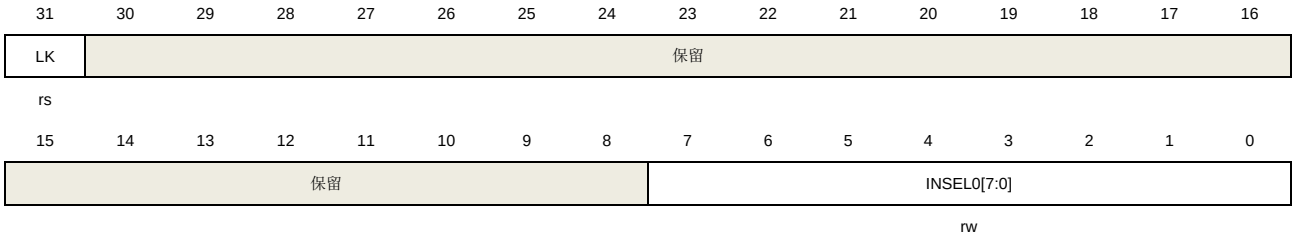
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER1ITI14将不能被修改。 0：TRIGSEL_TIMER1ITI14 寄存器可写可读 1：TRIGSEL_TIMER1ITI14 寄存器只读
30:8	保留	必须保持复位值
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER1_ITI14 的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.28. TIMER2_ITI14 触发选择寄存器（TRIGSEL_TIMER2ITI14）

地址偏移：0x78

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31	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_TIMER2ITI14将不能被修改。 0：TRIGSEL_TIMER2ITI14 寄存器可写可读 1：TRIGSEL_TIMER2ITI14 寄存器只读
30:8	保留	必须保持复位值
7:0	INSEL0[7:0]	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 TIMER2_ITI14 的触发源。关于具体配置请参考***表 6-1. 触发输入位域选择***。

6.5.29. TIMER3_ITI14 触发选择寄存器（TRIGSEL_TIMER3ITI14）

地址偏移：0x7C

复位值：0x0000 0000

该寄存器只能按字（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[7:0]							
								rw							

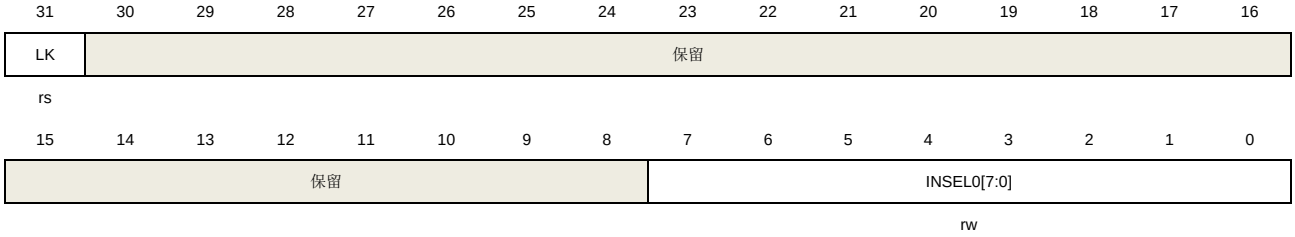
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER3ITI14将不能被修改。 0：TRIGSEL_TIMER3ITI14 寄存器可写可读 1：TRIGSEL_TIMER3ITI14 寄存器只读
30:8	保留	必须保持复位值
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER3_ITI14 的触发源。关于具体配置请参考***表 6-1. 触发输入位域选择***。

6.5.30. TIMER4_ITI14 触发选择寄存器（TRIGSEL_TIMER4ITI14）

地址偏移：0x80

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



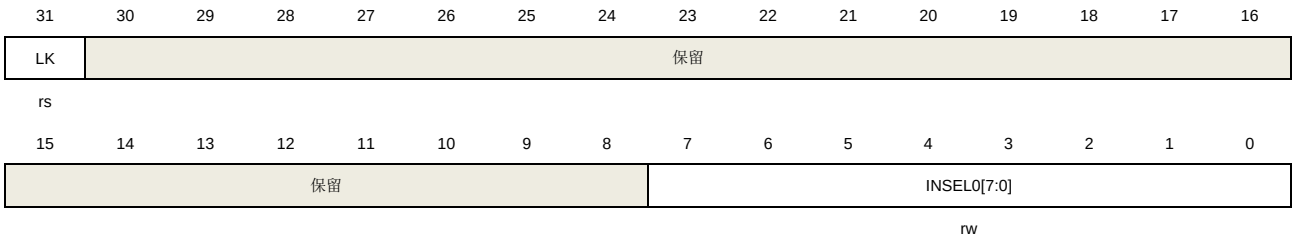
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER4ITI14将不能被修改。 0：TRIGSEL_TIMER4ITI14 寄存器可写可读 1：TRIGSEL_TIMER4ITI14 寄存器只读
30:8	保留	必须保持复位值
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER4_ITI14 的触发源。关于具体配置请参考***表 6-1. 触发输入位域选择***。

6.5.31. TIMER7_ITI14 触发选择寄存器（TRIGSEL_TIMER7ITI14）

地址偏移：0x84

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31	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_TIMER7ITI14将不能被修改。 0：TRIGSEL_TIMER7ITI14 寄存器可写可读

		1：TRIGSEL_TIMER7ITI14 寄存器只读
30:8	保留	必须保持复位值
7:0	INSEL0[7:0]	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_ITI14 的触发源。关于具体配置请参考***表 6-1. 触发输入位域选择***。

6.5.32. TIMER14_ITI14 触发选择寄存器（TRIGSEL_TIMER14ITI14）

地址偏移：0x88

复位值：0x0000 0000

该寄存器只能按字（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[7:0]							
								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_TIMER14ITI14将不能被修改。 0：TRIGSEL_TIMER14ITI14 寄存器可写可读 1：TRIGSEL_TIMER14ITI14 寄存器只读
30:8	保留	必须保持复位值
7:0	INSEL0[7:0]	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 TIMER14_ITI14 的触发源。关于具体配置请参考***表 6-1. 触发输入位域选择***。

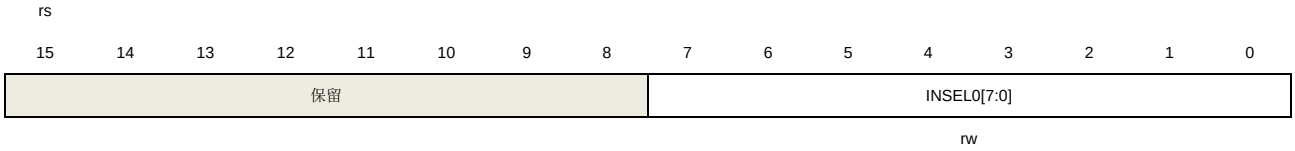
6.5.33. TIMER19_ITI14 触发选择寄存器（TRIGSEL_TIMER19ITI14）

地址偏移：0x8C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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



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_TIMER19ITI14将不能被修改。 0：TRIGSEL_TIMER19ITI14 寄存器可写可读 1：TRIGSEL_TIMER19ITI14 寄存器只读
30:8	保留	必须保持复位值
7:0	INSEL0[7:0]	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_ITI14 的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.34. DAC0 触发选择寄存器（TRIGSEL_DAC0）

地址偏移：0x90

复位值：0x0000 3B3B

该寄存器只能按字（32 位）访问。

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_DAC0将不能被修改。 0：TRIGSEL_DAC0 寄存器可写可读 1：TRIGSEL_DAC0 寄存器只读
30:16	保留	必须保持复位值
15:8	INSEL1[7:0]	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 DAC0_OUT1_EXTRIG

（DAC0_OUT1 外部触发）的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

7:0	INSEL0[7:0]	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 DAC0_OUT0_EXTRIG（DAC0_OUT0 外部触发）的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.35. DAC1 触发选择寄存器（TRIGSEL_DAC1）

地址偏移：0x94

复位值：0x0000 3B3B

该寄存器只能按字（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
INSEL1[7:0]								INSEL0[7:0]							
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_DAC1将不能被修改。 0：TRIGSEL_DAC1 寄存器可写可读 1：TRIGSEL_DAC1 寄存器只读
30:16	保留	必须保持复位值
15:8	INSEL1[7:0]	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 DAC1_OUT1_EXTRIG（DAC1_OUT1 外部触发）的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。
7:0	INSEL0[7:0]	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 DAC1_OUT0_EXTRIG（DAC1_OUT0 外部触发）的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.36. DAC2 触发选择寄存器（TRIGSEL_DAC2）

地址偏移：0x98

复位值：0x0000 1010

该寄存器只能按字（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
INSEL1[7:0]								INSEL0[7:0]							
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_DAC2将不能被修改。 0：TRIGSEL_DAC2 寄存器可写可读 1：TRIGSEL_DAC2 寄存器只读
30:16	保留	必须保持复位值
15:8	INSEL1[7:0]	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 DAC2_OUT1_EXTRIG（DAC2_OUT1 外部触发）的触发源。关于具体配置请参考***表 6-1. 触发输入位域选择***。
7:0	INSEL0[7:0]	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 DAC2_OUT0_EXTRIG（DAC2_OUT0 外部触发）的触发源。关于具体配置请参考***表 6-1. 触发输入位域选择***。

6.5.37. DAC3 触发选择寄存器（TRIGSEL_DAC3）

地址偏移：0x9C

复位值：0x0000 3B3B

该寄存器只能按字（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

INSEL1[7:0]	INSEL0[7:0]
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_DAC3将不能被修改。 0：TRIGSEL_DAC3 寄存器可写可读 1：TRIGSEL_DAC3 寄存器只读
30:16	保留	必须保持复位值
15:8	INSEL1[7:0]	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 DAC3_OUT1_EXTRIG（DAC3_OUT1 外部触发）的触发源。关于具体配置请参考***表 6-1. 触发输入位域选择***。
7:0	INSEL0[7:0]	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 DAC3_OUT0_EXTRIG（DAC3_OUT0 外部触发）的触发源。关于具体配置请参考***表 6-1. 触发输入位域选择***。

6.5.38. DAC0 触发选择扩展寄存器（TRIGSEL_EXTDAC0）

地址偏移：0xA0
复位值：0x0000 3B3B

该寄存器只能按字（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
INSEL1[7:0]								INSEL0[7:0]							
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_EXTDAC0将不能被修改。 0：TRIGSEL_EXTDAC0 寄存器可写可读 1：TRIGSEL_EXTDAC0 寄存器只读

30:16	保留	必须保持复位值
15:8	INSEL1[7:0]	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 DAC0_OUT1_ST_EXTRIG（DAC0_OUT1 锯齿递增/递减外部触发）的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。
7:0	INSEL0[7:0]	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 DAC0_OUT0_ST_EXTRIG（DAC0_OUT0 锯齿递增/递减外部触发）的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.39. DAC1 触发选择扩展寄存器（TRIGSEL_EXTDAC1）

地址偏移：0xA4

复位值：0x0000 3B3B

该寄存器只能按字（32 位）访问。

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_EXTDAC1将不能被修改。 0：TRIGSEL_EXTDAC1 寄存器可写可读 1：TRIGSEL_EXTDAC1 寄存器只读
30:16	保留	必须保持复位值
15:8	INSEL1[7:0]	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 DAC1_OUT1_ST_EXTRIG（DAC1_OUT1 锯齿递增/递减外部触发）的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。
7:0	INSEL0[7:0]	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 DAC1_OUT0_ST_EXTRIG（DAC1_OUT0 锯齿递增/递减外部触发）的触发源。关

于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.40. DAC2 触发选择扩展寄存器（TRIGSEL_EXTDAC2）

地址偏移：0xA8

复位值：0x0000 1010

该寄存器只能按字（32 位）访问。

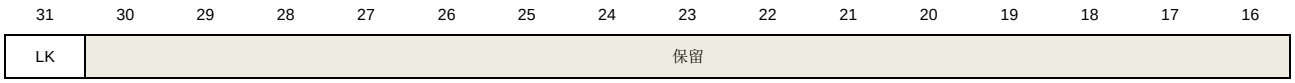
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_EXTDAC2将不能被修改。 0：TRIGSEL_EXTDAC2 寄存器可写可读 1：TRIGSEL_EXTDAC2 寄存器只读
30:16	保留	必须保持复位值
15:8	INSEL1[7:0]	触发输出 1 的输入源选择 这些位用来选择连接到输出 1 的触发输入信号，输出 1 作为 DAC2_OUT1_ST_EXTRIG（DAC2_OUT1 锯齿递增/递减外部触发）的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。
7:0	INSEL0[7:0]	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 DAC2_OUT0_ST_EXTRIG（DAC2_OUT0 锯齿递增/递减外部触发）的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.41. DAC3 触发选择扩展寄存器（TRIGSEL_EXTDAC3）

地址偏移：0xAC

复位值：0x0000 3B3B

该寄存器只能按字（32 位）访问。



rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INSEL1[7:0]								INSEL0[7:0]							
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_EXTDAC3将不能被修改。 0：TRIGSEL_EXTDAC3 寄存器可写可读 1：TRIGSEL_EXTDAC3 寄存器只读
30:16	保留	必须保持复位值
15:8	INSEL1[7:0]	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 DAC3_OUT1_ST_EXTRIG（DAC3_OUT1 锯齿递增/递减外部触发）的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。
7:0	INSEL0[7:0]	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 DAC0_OUT3_ST_EXTRIG（DAC3_OUT0 锯齿递增/递减外部触发）的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.42. CLA 触发选择寄存器 0（TRIGSEL_CLA_0）

地址偏移：0xB0

复位值：0x00C1 C0BF

该寄存器的复位取决于 SYSCFG_CFG2 中的 TRIGSEL_RSTMD 位。若 TRIGSEL_RSTMD = 0，该寄存器将在系统复位后复位。若 TRIGSEL_RSTMD = 1，该寄存器只能在 POR 复位后复位。

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
LK	保留							INSEL2[7:0]							
rs								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INSEL1[7:0]								INSEL0[7:0]							
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，通过系统复位或POR复位（取决于TRIGSEL_RSTMD的值）清除。该位置位后，TRIGSEL_CLA_0将不能被修改。 0：TRIGSEL_CLA_0 寄存器可写可读 1：TRIGSEL_CLA_0 寄存器只读
30:24	保留	必须保持复位值
23:16	INSEL2[7:0]	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 TRIGSEL_CLA_IN2 的触发源。关于具体配置请参考***表 6-1. 触发输入位域选择***。
15:8	INSEL1[7:0]	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 TRIGSEL_CLA_IN1 的触发源。关于具体配置请参考***表 6-1. 触发输入位域选择***。
7:0	INSEL0[7:0]	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 TRIGSEL_CLA_IN0 的触发源。关于具体配置请参考***表 6-1. 触发输入位域选择***。

6.5.43. CLA 触发选择寄存器 1（TRIGSEL_CLA_1）

地址偏移：0xB4

复位值：0x00C4 C3C2

该寄存器的复位取决于 SYSCFG_CFG2 中的 TRIGSEL_RSTMD 位。若 TRIGSEL_RSTMD = 0，该寄存器将在系统复位后复位。若 TRIGSEL_RSTMD = 1，该寄存器只能在 POR 复位后复位。

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
LK	保留							INSEL2[7:0]							
rs								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INSEL1[7:0]								INSEL0[7:0]							
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，通过系统复位或POR复位（取决于TRIGSEL_RSTMD的值）清除。该位置位后，TRIGSEL_CLA_1将不能被修改。 0：TRIGSEL_CLA_1 寄存器可写可读 1：TRIGSEL_CLA_1 寄存器只读
30:24	保留	必须保持复位值

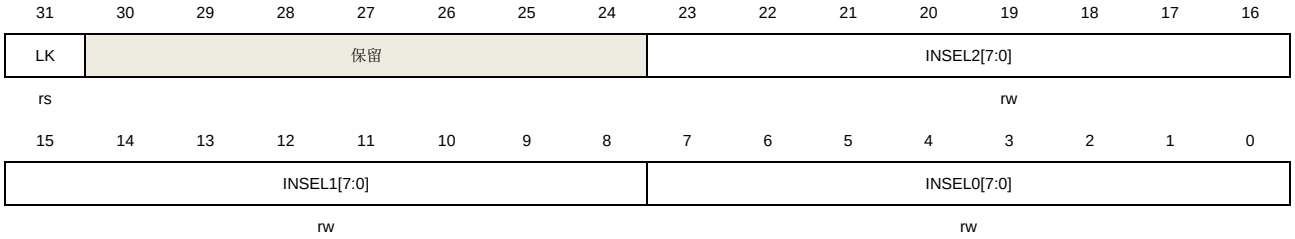
23:16	INSEL2[7:0]	触发输出 2 的输入源选择 这些位用来选择连接到输出 2 的触发输入信号，输出 2 作为 TRIGSEL_CLA_IN5 的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。
15:8	INSEL1[7:0]	触发输出 1 的输入源选择 这些位用来选择连接到输出 1 的触发输入信号，输出 1 作为 TRIGSEL_CLA_IN4 的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TRIGSEL_CLA_IN3 的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.44. CLA 触发选择寄存器 2（TRIGSEL_CLA_2）

地址偏移：0xB8

复位值：0x0027 2110

该寄存器只能按字（32 位）访问。



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_CLA_2将不能被修改。 0：TRIGSEL_CLA_2 寄存器可写可读 1：TRIGSEL_CLA_2 寄存器只读
30:24	保留	必须保持复位值
23:16	INSEL2[7:0]	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 TRIGSEL_CLA_IN8 的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。
15:8	INSEL1[7:0]	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 TRIGSEL_CLA_IN7 的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。
7:0	INSEL0[7:0]	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 TRIGSEL_CLA_IN6 的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.45. CLA 触发选择寄存器 3（TRIGSEL_CLA_3）

地址偏移：0xBC

复位值：0x0000 3A39

该寄存器只能按字（32 位）访问。

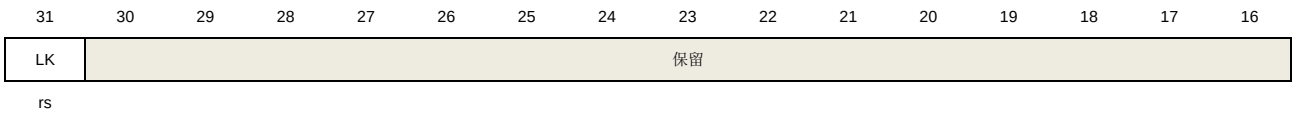
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_CLA_3将不能被修改。 0：TRIGSEL_CLA_3 寄存器可写可读 1：TRIGSEL_CLA_3 寄存器只读
30:16	保留	必须保持复位值
15:8	INSEL1[7:0]	触发输出 1 的输入源选择 这些位用来选择连接到输出 1 的触发输入信号，输出 1 作为 TRIGSEL_CLA_IN10 的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TRIGSEL_CLA_IN9 的触发源。关于具体配置请参考*表 6-1. 触发输入位域选择*。

6.5.46. CLA 触发选择寄存器 4（TRIGSEL_CLA_4）

地址偏移：0xC0

复位值：0x0000 00EF

该寄存器只能按字（32 位）访问。



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								INSEL0[7:0]							
								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_CLA_4将不能被修改。 0：TRIGSEL_CLA_4 寄存器可写可读 1：TRIGSEL_CLA_4 寄存器只读
30:8	保留	必须保持复位值
7:0	INSEL0[7:0]	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 TRIGSEL_CLA_IN11 的触发源。关于具体配置请参考***表 6-1. 触发输入位域选择***。

7. 通用和备用输入/输出接口（GPIO 和 AFIO）

7.1. 简介

最多可支持 107 个通用 I/O 引脚（GPIO），分别为 PA0～PA15，PB0～PB15，PC0～PC15，PD0～PD15，PE0～PE15，PF0～PF15，PG0～PG10。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。

7.2. 主要特性

- 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；
- 输入信号可以去除不必要的噪声；
- GPIO 具有系统复位时的状态保持功能（不包括电源复位）。

7.3. 功能说明

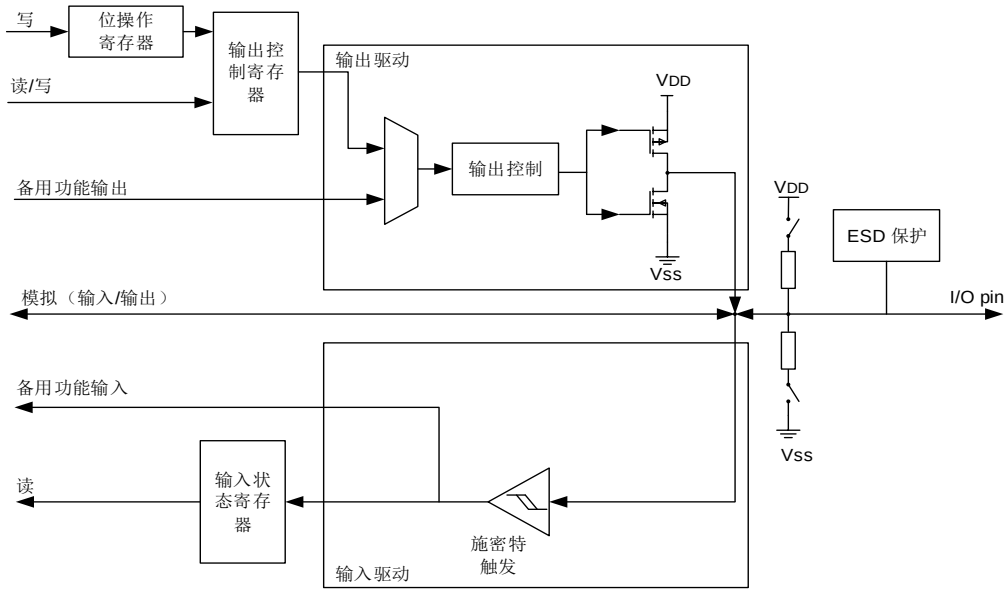
每个通用I/O端口都可以通过32位控制寄存器（GPIOx_CTL）配置为GPIO输入，GPIO输出，AF功能或模拟模式。引脚AFIO输入/输出是通过AFIO功能使能来选择。当端口配置为输出（GPIO输出或AFIO输出）时，可以通过GPIO输出模式寄存器（GPIOx_OMODE）配置为推挽或开漏模式。输出端口的最大速度可以通过GPIO输出速度寄存器（GPIOx_OSPD）配置。每个端口可以通过GPIO上/下拉寄存器（GPIOx_PUD）配置为浮空（无上拉或下拉），上拉或下拉功能。

表 7-1. GPIO 配置表

PAD TYPE			CTLy	OMy	PUDy
GPIO 输入	X	悬空	00	X	00
		上拉			01
		下拉			10
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7-1. 标准I/O端口位的基本结构*为标准I/O端口位的基本结构图。

图 7-1. 标准 I/O 端口位的基本结构



7.3.1. 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，JTAG/串行线调试为输入 PU/PD 模式：

PA15：JTDI为PU上拉模式
PA14：JTCK/SWCLK为PD下拉模式
PA13：JTMS/SWDIO为PU上拉模式
PB4：NJTRST为PU上拉模式
PB3：JTDO为输出模式

在重置期间，PB0/BOOT0至少在选项字节加载阶段结束之前处于输入模式。PB8可以作为引导引脚（BOOT0）或作为GPIO。根据用户选项字节中的nSWBOOT0位，它从输入模式切换到模拟输入模式：

- 如果 nSWBOOT0 = 1，则在选项字节加载阶段之后；
- 如果 nSWBOOT0 = 0，则复位后。

注意：建议将PB8设置为模拟模式以外的其他模式，以限制引脚未连接时的功耗。

使用PG10作为GPIO：PG10可以作为复位引脚（NRST）或作为GPIO。根据用户选项字节中的NRST MODE位，它切换到这些模式：

- 输入/输出复位:默认上电复位或选项字节加载后复位 NRST MODE = 3；
- 仅复位输入:选项字节加载后 NRST MODE = 1；
- PG10 模式：选项字节加载后 NRST mode = 2。

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。

7.3.2. 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。

7.3.3. 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。

7.3.4. 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。

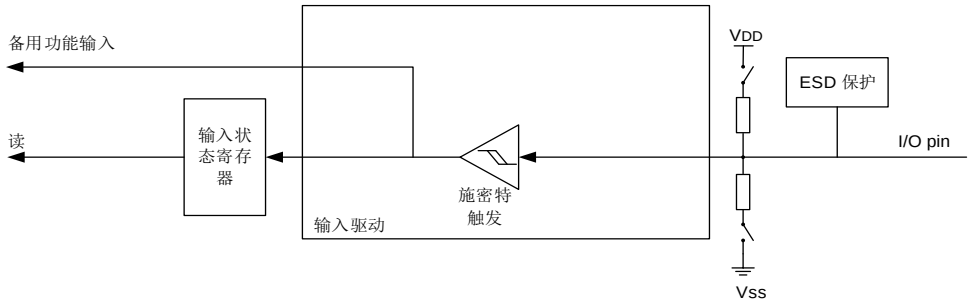
7.3.5. 输入配置

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7-2. 浮空/上拉/下拉输入配置*是I/O引脚的输入配置。

图 7-2. 浮空/上拉/下拉输入配置



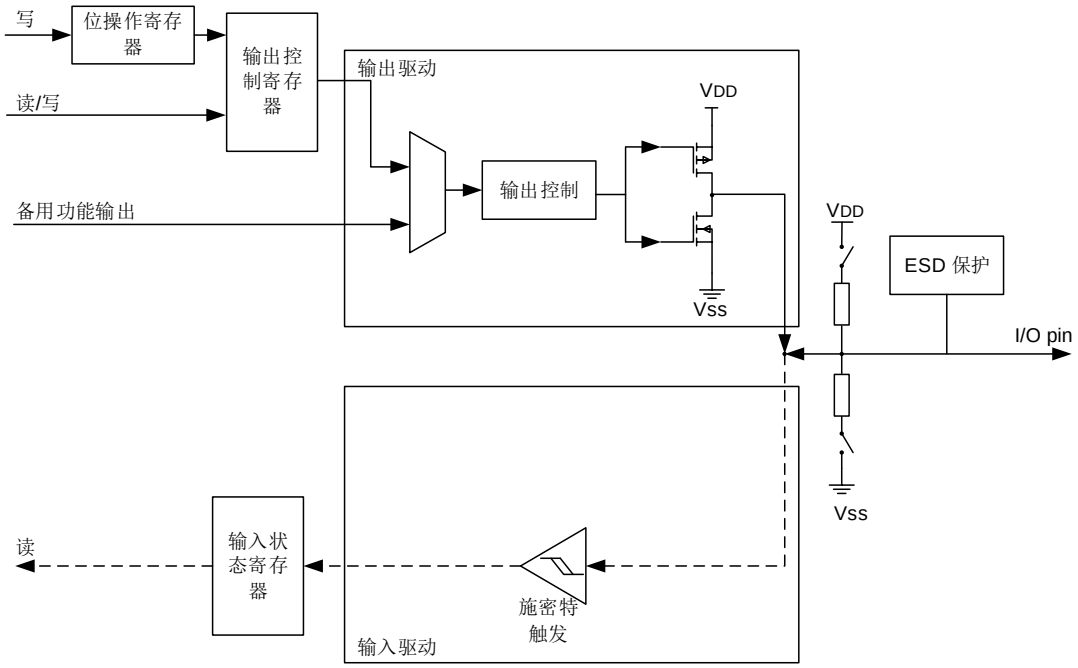
7.3.6. 输出配置

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；
- 使能输出缓冲；
- 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7-3. 输出配置*是 I/O 端口的输出配置。

图 7-3. 输出配置



7.3.7. 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7-4. 模拟高阻配置*是I/O端口的模拟高阻配置。

图 7-4. 模拟高阻配置

7.3.8. 备用功能（AF）配置

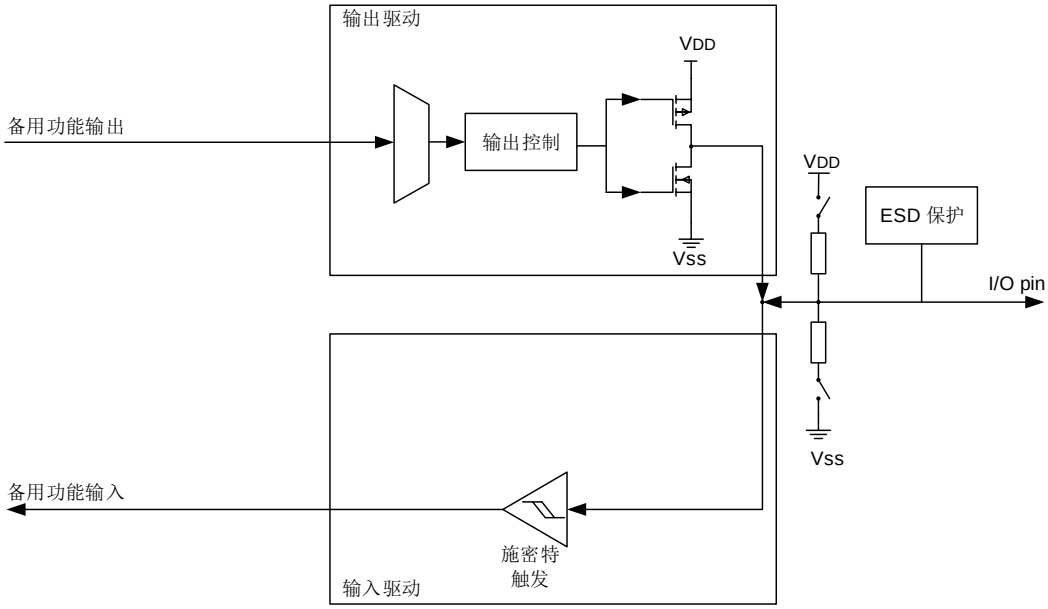
为了适应不同的器件封装，GPIO端口支持软件配置将一些备用功能应用到其他引脚上。

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7-5. 备用功能配置*是I/O端口备用功能配置图。

图 7-5. 备用功能配置



7.3.9. 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。

7.3.10. 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。

7.3.11. I/O 补偿单元

补偿单元用于控制I/O换向的转换速率（上升时间/下降时间），以降低电源上的I/O噪声。

该单位分为两个区域。第一个区域提供当前PVT的最佳补偿码，当设置SYSCFG_CPSCTL的CPS_RDY位时，可以读取存储在该区域的补偿码。第二个区域控制I/O转换速率。用户通过软件编程选择要应用的补偿码。

I/O补偿单元具有2个电压范围：1.62至2.0V和2.7至3.6V。

7.3.12. 输入滤波

通过配置GPIOx_IFTP寄存器，可以为每个GPIO引脚选择输入滤波的类型。对于配置为通用数字输入引脚（GPIO）的情况下，过滤可以指定为仅同步到CK_AHB或通过采样窗口来进行。对于配置为外设输入的管脚，除了同步到CK_AHB或通过采样窗口限定外，输入还可以是异步的。

异步输入

此模式用于不需要输入同步或外围设备本身执行同步的外围设备。如果引脚用作通用数字输入引脚（GPIO）时，异步选项无效，如果配置了异步输入，则输入滤波默认为同步到CK_AHB。

注意：当外设本身执行同步时使用输入同步可能会导致意外的结果。在这种情况下，用户应该确保GPIO配置为异步。

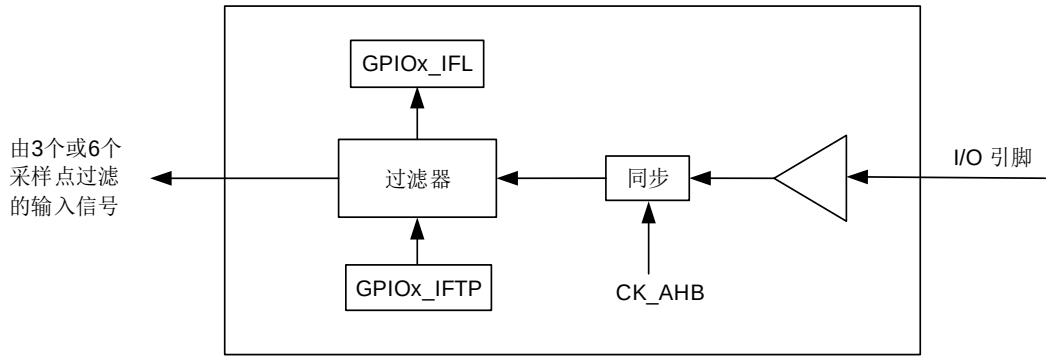
仅同步到 CK_AHB

在这种模式下，输入信号仅与CK_AHB同步。由于输入信号是异步的，它可能需要一个CK_AHB的延迟周期，以更改MCU的输入。此后信号不会进行进一步的过滤。

使用采样窗口进行过滤

在这种模式下，信号首先与CK_AHB进行同步，然后在允许改变输入之前通过指定的周期数进行过滤。用户需要为此类型的过滤指定了两个参数：采样周期和采样次数。

图7-6. 使用采样窗口进行过滤



采样周期

为了过滤，对输入信号以一个固定的周期进行采样。采样周期由用户指定，并确定采样之间的持续时间，或相对于CK_AHB的信号采样频率。

采样周期由GPIOx_IFL寄存器中的过滤采样频率（FLPRDx）位指定。采样周期可配置为8个输入信号组。例如，GPIO0到GPIO7使用GPIOx_IFL寄存器的FLPRD0设置，GPIO8到GPIO15使用GPIOx_IFL的FLPRD1。

如果GPIOx_IFL寄存器的FLPRD0=0，则采样频率为f_{CK_AHB}。例如，如果f_{CK_AHB}=100MHz，信号将以100兆赫兹或每10ns采样一次。

如果GPIOx_IFL寄存器的FLPRD0=0xFF(255)，则采样频率为f_{CK_AHB}×1÷(2×FLPRDx)。例如，如果f_{CK_AHB}=100MHz，然后信号将以100MHz×1÷(2×255)或每5.1μs采样一次。

采样次数

信号的采样次数为3个样本或6个样本，在输入滤波类型寄存器中（GPIOx_IFTP）有详细的描述。当三个或六个连续周期相同时，输入的变化将传递给MCU。

总采样窗口宽度

采样窗口是对输入信号进行采样所消耗的时间，如*图7-7. 输入滤波时钟周期*所示。通过使用采样周期和采样次数的进行计算，可以确定窗口的总宽度。

为了使输入滤波能够检测到输入的变化，信号电平必须在采样窗口宽度或更长时间内保持稳定。

采样窗口的数目始终比采样数少一个。对于三个采样的窗口，采样窗口宽度为两个采样周期宽，同样，对于六个采样的窗口，采样窗口宽度为五个采样周期宽。

注意：外部信号变化与采样周期和CK_AHB都是异步的。由于外部信号的异步特性，输入应在大于采样窗口宽度的时间内保持稳定，以确保在逻辑上检测到信号的变化。额外所需的时间可以达到额外的采样周期加上T_{CK_AHB}。

采样窗口示例

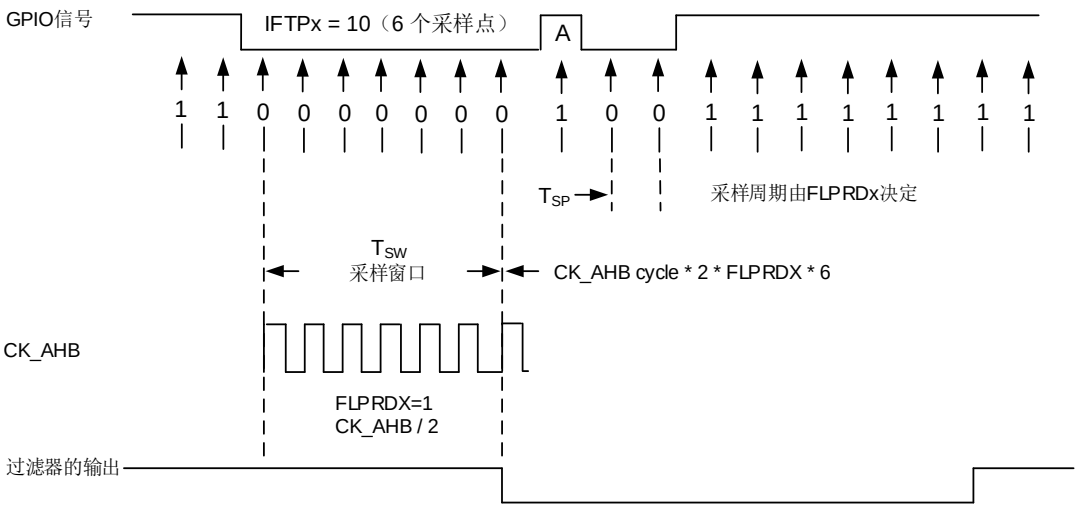
如*图7-7. 输入滤波时钟周期*所示的例子，输入滤波配置如下：

- GPIOx_IFTP寄存器的IFTP0[1:0]=10，表示有6个采样点；
- GPIOx_IFL寄存器的FLPRD0=1，采样周期T_{SP}=2×FLPRD0×T_{CK_AHB}=2×T_{CK_AHB}；

配置结果如下：

- 采样宽度为：T_{SW}=6×T_{SP}=6×2×FLPRD0×T_{CK_AHB}=6×2×T_{CK_AHB}；
- 如果T_{CK_AHB}=10ns，则采样窗口的持续时间为：
  T_{SP}=2×T_{CK_AHB}=2×10ns=20ns
  T_{SW}=6×2×T_{CK_AHB}=6×20ns=120ns
- 为了说明相对于采样周期和系统时钟的输入异步性质，可能需要一个额外的采样周期和CK_AHB周期来检测输入信号的变化。
  T_{SW}+T_{CK_AHB}=120+10=130ns
- 在*图7-7. 输入滤波时钟周期*，干扰（A）比总采样窗口短，所以将会被过滤。

图7-7. 输入滤波时钟周期



7.4. 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

GPIOG基地址：0x4800 1800

7.4.1. 端口控制寄存器（GPIOx_CTL, x=A…G）

地址偏移：0x00

复位值：端口 A 0xABFF FFFF；端口 B 0xFFFF FEBF；其他端口 0xFFFF FFFF

该寄存器可以按字（32 位）访问。

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
CTL15[1:0]	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 01：GPIO输出模式 10：备用功能模式 11：模拟模式（输入和输出）（复位值）

7.4.2. 端口输出模式寄存器（GPIOx_OMODE, x=A…G）

地址偏移：0x04

复位值：0x0000 0000

该寄存器可以按字（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
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

7.4.3. 端口输出速度寄存器（GPIOx_OSPD, x=A…G）

地址偏移：0x08

复位值：端口 A 0x0C00 0000；端口 B 0x0000 00C0；其他端口 0x0000 0000

该寄存器可以按字（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。 00：输出最大速度12M（复位值） 01：输出最大速度60M 10：输出最大速度85M 11：输出最大速度100M/220M

7.4.4. 端口上拉/下拉寄存器（GPIOx_PUD, x=A…G）

地址偏移：0x0C

复位值：端口 A 0x6400 0000；端口 B 0x0000 0100；其他端口 0x0000 0000

该寄存器可以按字（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

7.4.5. 端口输入状态寄存器（GPIOx_ISTAT, x=A…G）

地址偏移：0x10

复位值：0x0000 XXXX

该寄存器可以按字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISTAT15	ISTAT14	ISTAT13	ISTAT12	ISTAT11	ISTAT10	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） 这些位由硬件置位和清除。 0：引脚输入信号为低电平 1：引脚输入信号为高电平

7.4.6. 端口输出控制寄存器（GPIOx_OCTL, x=A…G）

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

7.4.7. 端口位操作寄存器（GPIOx_BOP, x=A…G）

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

7.4.8. 端口配置锁定寄存器（GPIOx_LOCK, x=A…G）

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

7.4.9. 备用功能选择寄存器 0（GPIOx_AFSEL0, x=A…G）

地址偏移：0x20

复位值：0x0000 0000

该寄存器可以按字（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

7.4.10. 备用功能选择寄存器 1（GPIOx_AFSEL1, x=A…G）

地址偏移：0x24

复位值：0x0000 0000

该寄存器可以按字（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

7.4.11. 位清除寄存器（GPIOx_BC, x=A…G）

地址偏移：0x28

复位值：0x0000 0000

该寄存器可以按字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CRy	端口清除位y（y=0..15） 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

7.4.12. 端口位翻转寄存器（GPIOx_TG, x=A…G）

地址偏移：0x2C

复位值：0x0000 0000

该寄存器可以按字（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
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

7.4.13. 输入滤波寄存器（GPIOx_IFL, x=A…G）

地址偏移：0x30

复位值：0x0000 0000

该寄存器可以按字（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
FLPRD1								FLPRD0							
rw								rw							

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	FLPRD1	GPIO8 到 GPIO15 的过滤采样频率： 00：FLPRDx = CK_AHB 01：FLPRDx = CK_AHB / 2 02：FLPRDx = CK_AHB / 4 .... FF：FLPRDx = CK_AHB / 510
7:0	FLPRD0	GPIO0 到 GPIO7 的过滤采样频率： 00：FLPRDx = CK_AHB 01：FLPRDx = CK_AHB / 2 02：FLPRDx = CK_AHB / 4 .... FF：FLPRDx = CK_AHB / 510

7.4.14. 输入滤波类型寄存器（GPIOx_IFTP, x=A…G）

地址偏移：0x34

复位值：0xFFFF FFFF

该寄存器可以按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IFTP15[1:0]		IFTP14[1:0]		IFTP13[1:0]		IFTP12[1:0]		IFTP11[1:0]		IFTP10[1:0]		IFTP9[1:0]		IFTP8[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
IFTP7[1:0]		IFTP6[1:0]		IFTP5[1:0]		IFTP4[1:0]		IFTP3[1:0]		IFTP2[1:0]		IFTP1[1:0]		IFTP0[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	

位/位域	名称	描述

31:30	IFTP15[1:0]	Pin 15 的过滤输入类型 该位由软件置位和清除。 参照 IFTP0[1:0]的描述
29:28	IFTP14[1:0]	Pin 14 的过滤输入类型 该位由软件置位和清除。 参照 IFTP0[1:0]的描述
27:26	IFTP13[1:0]	Pin 13 的过滤输入类型 该位由软件置位和清除。 参照 IFTP0[1:0]的描述
25:24	IFTP12[1:0]	Pin 12 的过滤输入类型 该位由软件置位和清除。 参照 IFTP0[1:0]的描述
23:22	IFTP11[1:0]	Pin 11 的过滤输入类型 该位由软件置位和清除。 参照 IFTP0[1:0]的描述
21:20	IFTP10[1:0]	Pin 10 的过滤输入类型 该位由软件置位和清除。 参照 IFTP0[1:0]的描述
19:18	IFTP9[1:0]	Pin 9 的过滤输入类型 该位由软件置位和清除。 参照 IFTP0[1:0]的描述
17:16	IFTP8[1:0]	Pin 8 的过滤输入类型 该位由软件置位和清除。 参照 IFTP0[1:0]的描述
15:14	IFTP7[1:0]	Pin 7 的过滤输入类型 该位由软件置位和清除。 参照 IFTP0[1:0]的描述
13:12	IFTP6[1:0]	Pin 6 的过滤输入类型 该位由软件置位和清除。 参照 IFTP0[1:0]的描述
11:10	IFTP5[1:0]	Pin 5 的过滤输入类型 该位由软件置位和清除。 参照 IFTP0[1:0]的描述
9:8	IFTP4[1:0]	Pin 4 的过滤输入类型 该位由软件置位和清除。

		参照 IFTP0[1:0]的描述
7:6	IFTP3[1:0]	Pin 3 的过滤输入类型 该位由软件置位和清除。 参照 IFTP0[1:0]的描述
5:4	IFTP2[1:0]	Pin 2 的过滤输入类型 该位由软件置位和清除。 参照 IFTP0[1:0]的描述
3:2	IFTP1[1:0]	Pin 1 的过滤输入类型 该位由软件置位和清除。 参照 IFTP0[1:0]的描述
1:0	IFTP0[1:0]	Pin 0 的过滤输入类型 该位由软件置位和清除。 00：同步 01：过滤（3 个采样点） 10：过滤（6 个采样点） 11：异步（非同步和过滤模式）

8. 直接存储器访问控制器（DMA）

8.1. 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 14 个通道（DMA0 有 7 个通道，DMA1 有 7 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。

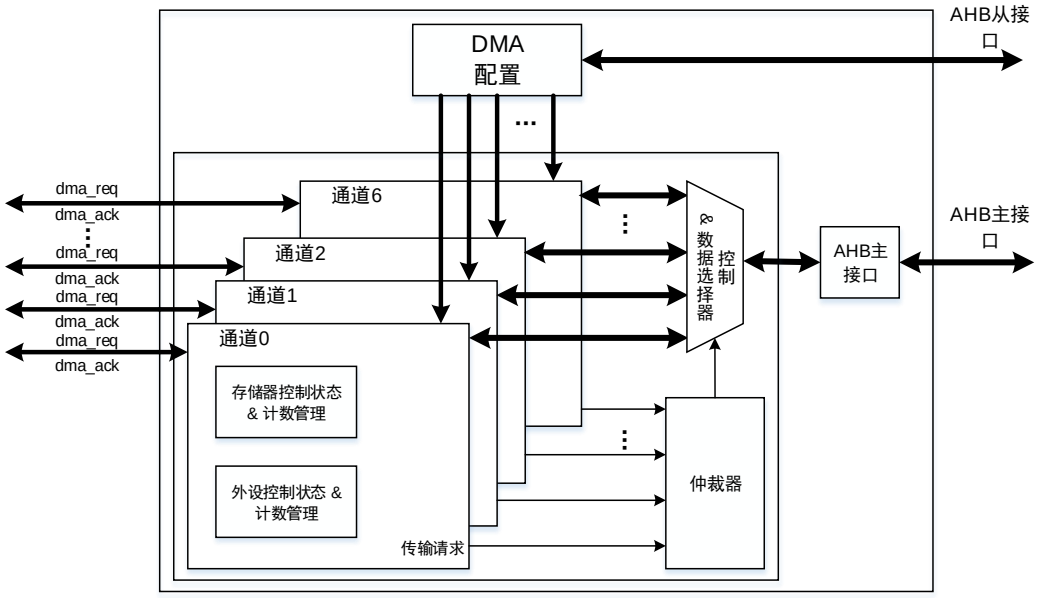
DMA 控制器和 Cortex®-M33 内核共享系统总线。当 DMA 和 CPU 访问同样的地址空间时，DMA 访问可能会阻挡 CPU 访问系统总线几个总线周期。总线矩阵中实现了循环仲裁算法来分配 DMA 与 CPU 的访问权，它可以确保 CPU 得到至少一半的系统总线带宽。

8.2. 主要特征

- 传输数据长度可编程配置，最大到 65536；
- 14 个通道（DMA0 有 7 个通道，DMA1 有 7 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（DMA 通道 0 的优先级最高，DMA 通道 6 的优先级最低）；
- 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；
- 支持中断使能和清除。

8.3. 结构框图

图 8-1. DMA 结构框图



由*图 8-1. 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。

8.4. 功能说明

8.4.1. 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 8-1. DMA 传输操作*。

表 8-1. 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。

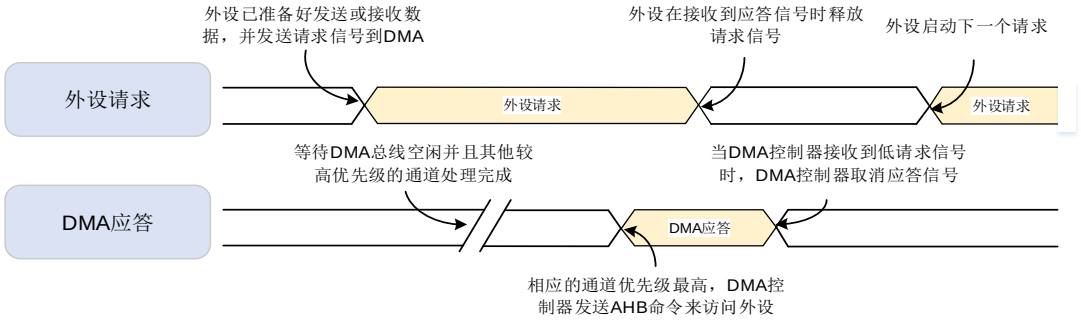
8.4.2. 外设握手

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8-2. 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 8-2. 握手机制



8.4.3. 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。

8.4.4. 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。

8.4.5. 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。

8.4.6. 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。

8.4.7. 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。

8.4.8. 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。

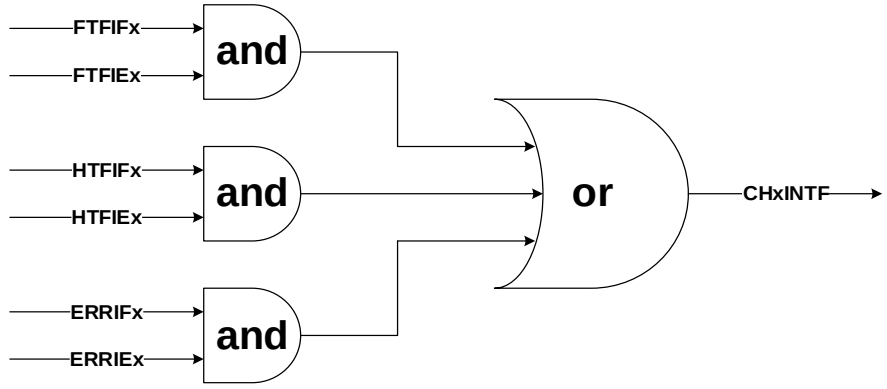
每一个中断事件在DMA_INTF寄存器中有专用的标志位，在DMA_INTC寄存器中有专用的清除位，在DMA_CHxCTL寄存器中有专用的使能位。*表8-2. 中断事件*描述了其对应关系。

表 8-2. 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8-3. 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 8-3. DMA 中断逻辑图



注意：“x”表示通道数（对应x=0…6）

8.4.9. 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 9-3. DMAMUX 请求路由输入信号映射*。

8.5. DMA 寄存器

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

8.5.1. 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
保留				ERRIF6	HTFIF6	FTFIF6	GIF6	ERRIF5	HTFIF5	FTFIF5	GIF5	ERRIF4	HTFIF4	FTFIF4	GIF4
				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIF3	HTFIF3	FTFIF3	GIF3	ERRIF2	HTFIF2	FTFIF2	GIF2	ERRIF1	HTFIF1	FTFIF1	GIF1	ERRIF0	HTFIF0	FTFIF0	GIF0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

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

8.5.2. 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
保留				ERRIFC6	HTFIFC6	FTFIFC6	GIFC6	ERRIFC5	HTFIFC5	FTFIFC5	GIFC5	ERRIFC4	HTFIFC4	FTFIFC4	GIFC4
				w	w	w	w	w	w	w	w	w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIFC3	HTFIFC3	FTFIFC3	GIFC3	ERRIFC2	HTFIC2	FTFIFC2	GIFC2	ERRIFC1	HTFIFC1	FTFIFC1	GIFC1	ERRIFC0	HTFIFC0	FTFIFC0	GIFC0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

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

8.5.3. 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

8.5.4. 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。

8.5.5. 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。

8.5.6. 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。

9. DMA 请求多路复用器（DMAMUX）

9.1. 简介

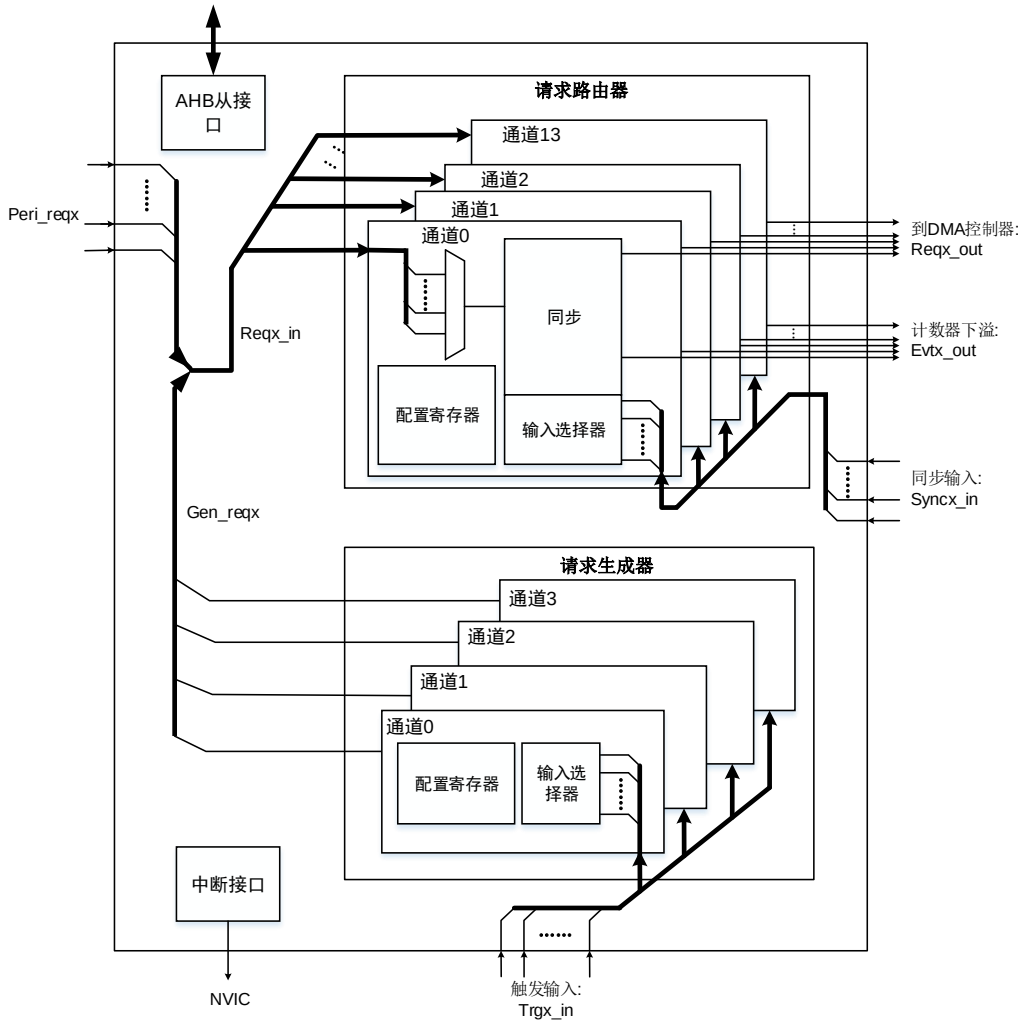
DMAMUX 是 DMA 请求的传输调度器。可编程的 DMA 请求多路复用器 DMAMUX，可在外设和 DMA 控制器之间路由 DMA 请求线路，或者 DMAMUX 也可以将可编程事件连入到输入触发信号上，作为一个 DMAMUX 请求发生器，再由 DMAMUX 请求路由器在 DMAMUX 请求生成器产生的 DMA 请求和 DMA 控制器之间路由 DMA 请求线路。每个 DMAMUX 请求路由通道选择一条唯一的 DMA 请求线路，无条件地或同步地从它的 DMAMUX 同步输入事件。DMA 请求信号会一直挂起，直到 DMA 控制器响应它，并且产生一个 DMA 确认信号，此时相应的 DMA 请求信号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

9.2. 主要特征

- 14 个可配置的 DMAMUX 请求路由输出通道；
- 4 个 DMAMUX 请求生成通道；
- 21 路触发输入信号到 DMAMUX 请求生成器；
- 21 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：
  - 125 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

9.3. 结构框图

图 9-1. DMAMUX 结构框图



9.4. 功能说明

如*图 9-1. DMAMUX 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。

9.4.1. DMAMUX 信号

表 9-1. 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

9.4.2. 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[7:0]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 9-3. 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[7:0]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[7:0]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。

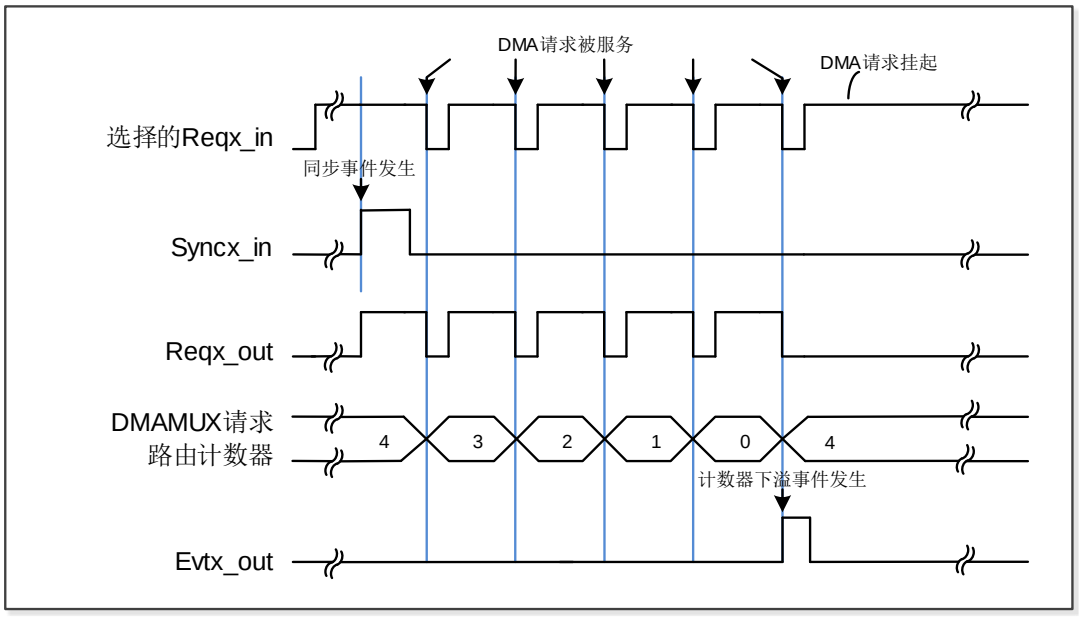
注意：只有当 DMAMUX 请求路由通道 x 的同步使能位 SYNCEN 位和通道事件输出使能位 EVGEN 位都为 0 时，才能配置其 NBR[4:0]位域。

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 9-2. 同步模式*为当 NBR[4:0]=4，SYNCEN=1，EVGEN=1，SYNCP[1:0]=0b01 时的举例。

图 9-2. 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 9-5. 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。

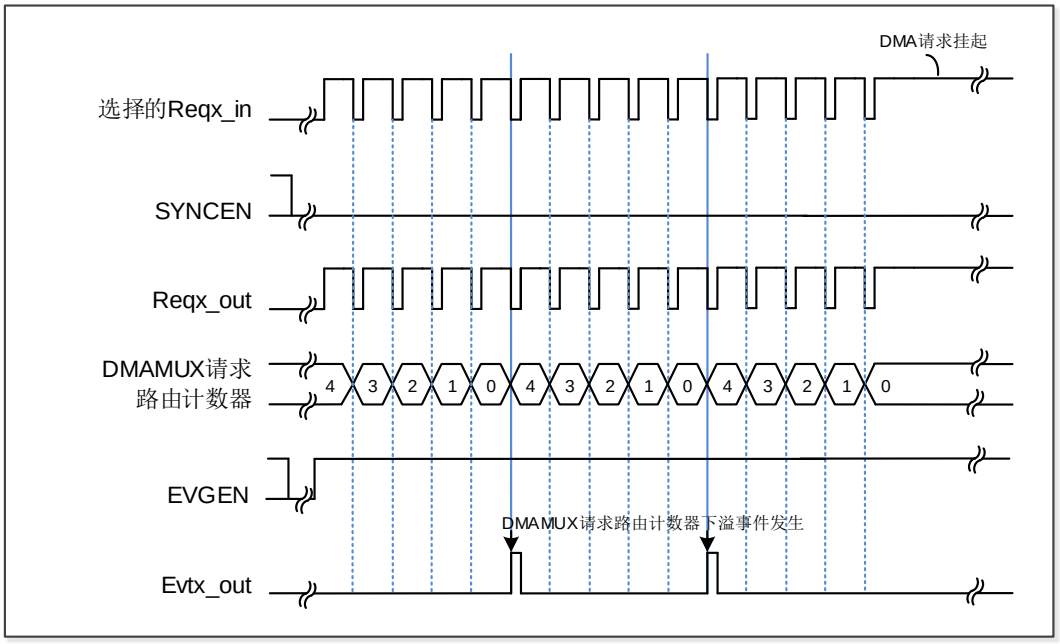
注意：如果同步输入事件发生时，DMAMUX 输入上没有挂起的 DMA 请求，则这个同步输入事件将被忽略，之后如有 DMA 请求被挂起，它将不会被连接到 DMAMUX 请求路由通道 x 的输出，直到发生下一个同步输入事件。

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 9-3. 通道事件输出*为当 NBR[4:0]=4，SYNCEN=0，EVGEN=1 时的举例。

图 9-3. 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。

9.4.3. 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 9-4. 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。

9.4.4. 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。

9.4.5. 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9-2. 中断事件*描述了其对应关系。

表 9-2. 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。

9.4.6. 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 9-3. DMAMUX 请求路由输入信号映射*，由 DMAMUX_RM_CHxCFG 寄存器的 MUXID[7:0]位域配置 DMAMUX 请求路由通道 x 的输入。

表 9-3. DMAMUX 请求路由输入信号映射

请求路由通道输入标识 MUXID[7:0]	来源
1	Gen_reqx0
2	Gen_reqx1
3	Gen_reqx2
4	Gen_reqx3

请求路由通道输入标识 MUXID[7:0]	来源
5	ADC0
6	DAC0_CH0
7	DAC0_CH1
8	TIMER5_UP
9	TIMER6_UP
10	SPI0_RX
11	SPI0_TX
12	SPI1_RX
13	SPI1_TX
14	SPI2_RX
15	SPI2_TX
16	I2C0_RX
17	I2C0_TX
18	I2C1_RX
19	I2C1_TX
20	I2C2_RX
21	I2C2_TX
22	I2C3_RX
23	I2C3_TX
24	USART0_RX
25	USART0_TX
26	USART1_RX
27	USART1_TX
28	USART2_RX
29	USART2_TX
30	UART3_RX
31	UART3_TX
32	UART4_RX
33	UART4_TX
34	ADC1
35	ADC2
36	ADC3
37	QSPI
38	DAC1_CH0
39	DAC1_CH1
40	TIMER0_CH0
41	TIMER0_CH1

请求路由通道输入标识 MUXID[7:0]	来源
42	TIMER0_CH2
43	TIMER0_CH3
44	TIMER0_CH0N
45	TIMER0_CH1N
46	TIMER0_CH2N
47	TIMER0_CH3N
48	TIMER0_ UP
49	TIMER0_TI
50	TIMER0_CO
51	TIMER7_CH0
52	TIMER7_CH1
53	TIMER7_CH2
54	TIMER7_CH3
55	TIMER7_CH0N
56	TIMER7_CH1N
57	TIMER7_CH2N
58	TIMER7_CH3N
59	TIMER7_ UP
60	TIMER7_TI
61	TIMER7_CO
62	TIMER1_CH0
63	TIMER1_CH1
64	TIMER1_CH2
65	TIMER1_CH3
66	TIMER1_ UP
67	TIMER1_TI
68	TIMER2_CH0
69	TIMER2_CH1
70	TIMER2_CH2
71	TIMER2_CH3
72	TIMER2_ UP
73	TIMER2_TI
74	TIMER3_CH0
75	TIMER3_CH1
76	TIMER3_CH2
77	TIMER3_CH3
78	TIMER3_ UP

请求路由通道输入标识 MUXID[7:0]	来源
79	TIMER3_TI
80	TIMER4_CH0
81	TIMER4_CH1
82	TIMER4_CH2
83	TIMER4_CH3
84	TIMER4_ UP
85	TIMER4_TI
86	TIMER14_CH0
87	TIMER14_CH1
88	TIMER14_CH0N
89	TIMER14_ UP
90	TIMER14_ TI
91	TIMER14_ CO
92	TIMER15_CH0
93	TIMER15_CH0N
94	TIMER15_ UP
95	TIMER16_CH0
96	TIMER16_CH0N
97	TIMER16_ UP
98	TIMER19_CH0
99	TIMER19_CH1
100	TIMER19_CH2
101	TIMER19_CH3
102	TIMER19_CH0N
103	TIMER19_CH1N
104	TIMER19_CH2N
105	TIMER19_CH3N
106	TIMER19_ UP
107	TIMER19_ TI
108	TIMER19_ CO
109	CAU_IN
110	CAU_OUT
111	HRTIMER_M
112	HRTIMER_0
113	HRTIMER_1
114	HRTIMER_2
115	HRTIMER_3

请求路由通道输入标识 MUXID[7:0]	来源
116	HRTIMER_4
117	HRTIMER_5
118	HRTIMER_6
119	HRTIMER_7
120	DAC2_CH0
121	DAC2_CH1
122	DAC3_CH0
123	DAC3_CH1
124	HPDF_FLT0
125	HPDF_FLT1
126	HPDF_FLT2
127	HPDF_FLT3
128	FAC_RD
129	FAC_WR
130	TMU_RD
131	TMU_WR
132	CAN0
133	CAN1
134	CAN2

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 9-4. 触发输入信号映射*。

表 9-4. 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

触发输入标识TID[4:0]	来源
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	LPTIMER_OUT

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 9-5. 同步输入信号映射*。

表 9-5. 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	LPTIMER_OUT

9.5. DMAMUX 寄存器

DMAMUX基地址：0x4002 0800

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

地址偏移：0x00 + 0x04 * x（x = 0...13，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	MUXID[7:0]							
						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
7:0	MUXID[7:0]	请求路由标识 选择DMAMUX请求路由通道的DMA请求输入源。

9.5.2. 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		SOIF13	SOIF12	SOIF11	SOIF10	SOIF9	SOIF8	SOIF7	SOIF6	SOIF5	SOIF4	SOIF3	SOIF2	SOIF1	SOIF0
		r	r	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..13）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。

9.5.3. 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	SOIFC13	SOIFC12	SOIFC11	SOIFC10	SOIFC9	SOIFC8	SOIFC7	SOIFC6	SOIFC5	SOIFC4	SOIFC3	SOIFC2	SOIFC1	SOIFC0
	w	w	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..13）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。

9.5.4. 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					

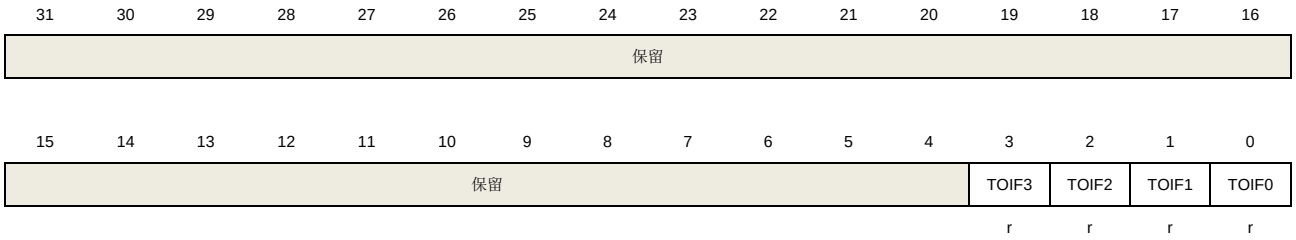
位/位域	名称	描述
31:24	保留	必须保持复位值。
23:19	NBRG[4:0]	待产生的DMA请求数量 在触发输入事件之后，待产生的DMA请求数量为NBRG[4:0] + 1。 **注意：**只有当RGEN位为0时才能写该位域。
18:17	RGTP[1:0]	DMAMUX请求生成触发输入极性 00：不检测事件 01：上升沿 10：下降沿 11：上升沿和下降沿
16	RGEN	DMAMUX请求生成通道x使能 0：禁能DMAMUX请求生成通道x 1：使能DMAMUX请求生成通道x
15:9	保留	必须保持复位值。
8	TOIE	触发溢出中断使能 0：禁能中断

		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。

9.5.5. 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。

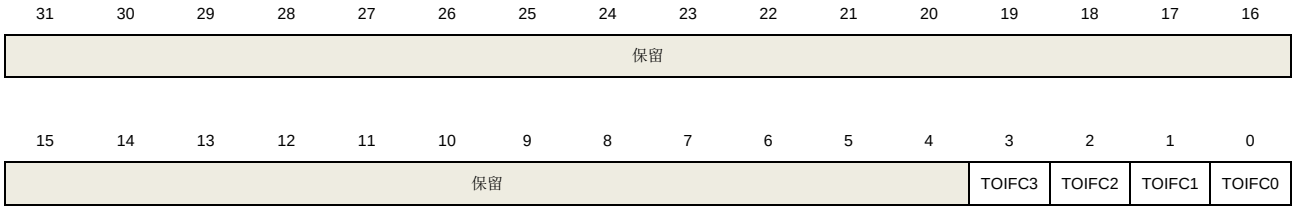


位/位域	名称	描述
31:4	保留	必须保持复位值。
3:0	TOIFx	DMAMUX请求生成通道x（x=0..3）的触发溢出标志位 如果在DMAMUX请求生成计数器（通过DMAMUX_RG_CHxCFG寄存的NBRG[4:0]位域配置）发生下溢之前，DMAMUX请求生成通道x发生了一个新的触发输入事件，则该标志位置位。 通过对DMAMUX_RG_INTC寄存器的TOIFCx位写1来清除相应通道的触发溢出标志。

9.5.6. 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。



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。

10. 循环冗余校验管理单元（CRC）

10.1. 简介

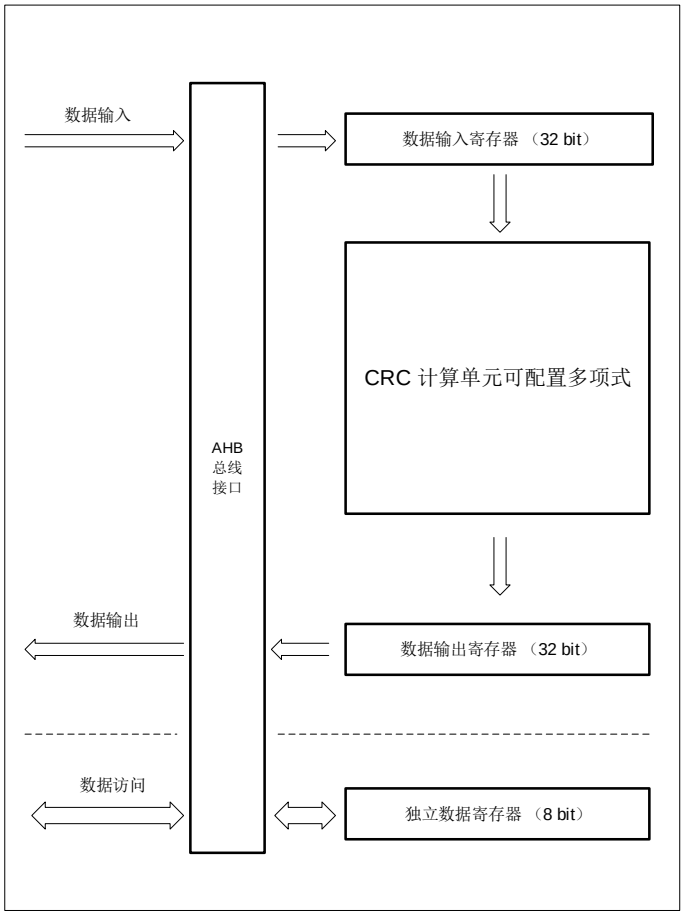
循环冗余校验码是一种用在数字网络和存储设备上的差错校验码，可以校验原始数据的偶然误差。

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

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0x3456CDEF为例：

  1）按字节逆序：
  32位数据被分成四组，组内完成颠倒。逆序后的数据为：0x2C6AB3F7

  2）按半字逆序：
  32位数据被分成两组，组内完成颠倒。逆序后的数据为：0x6A2CF7B3

  3）按字逆序：
  32位数据被分成一组，组内完成颠倒，逆序后的数据为：0xF7B36A2C

  对于输出数据来说，逆序形式为按字逆序。

  例如：当REV_O=1，计算结果0x3344CCDD将被逆序成0xBB3322CC。

- 用户可配置的初始计算数据。

  当RST位置位或对CRC_IDATA寄存器进行写操作时，CRC_DATA寄存器将自动初始化为CRC_IDATA寄存器中的值。

- 用户配置多项式。

  通过配置PS[1:0]，用户可以选择有效多项式和输出数据位宽。如果多项式少于32位，那么输入和输出数据的高位无效。当PS[1:0]或多项式改变后，需要复位CRC。

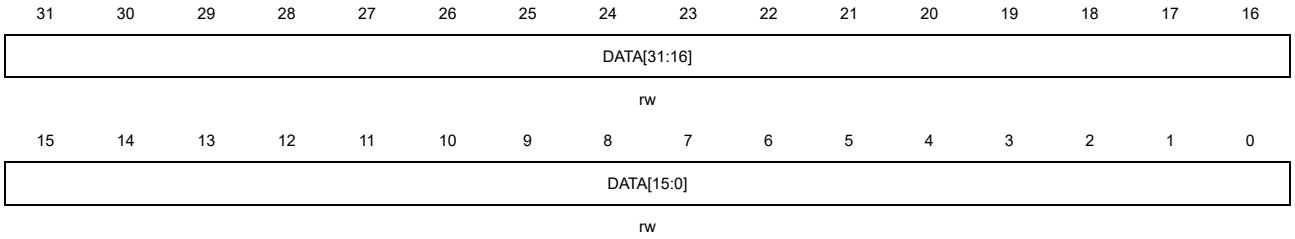
10.4. CRC 寄存器

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。

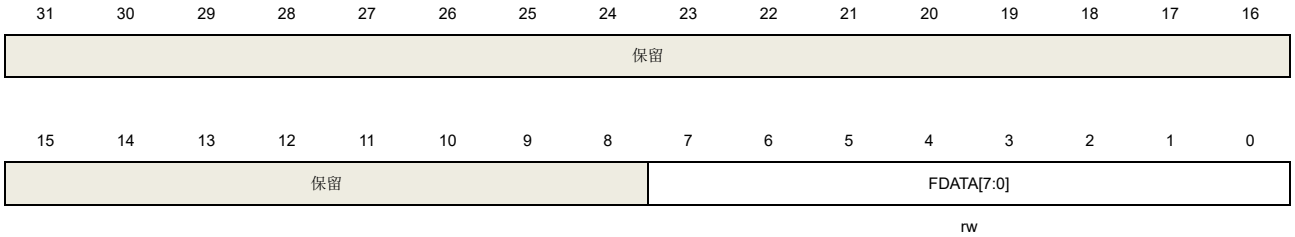


位/位域	名称	描述
31:0	DATA[31:0]	CRC 计算结果位 软件可读可写。 该寄存器用于接收待计算的新数据，直接将其写入即可。刚写入的数据不能被读出来因为读取该寄存器得到的是上次 CRC 计算的结果。

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。



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。

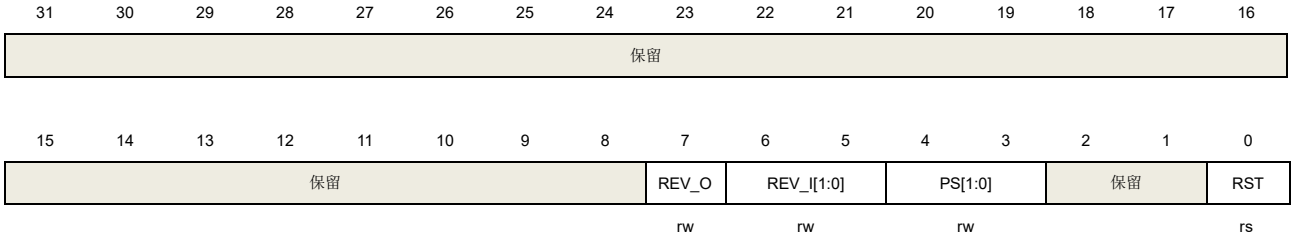
这些位与 CRC 计算无关。该字节能被任何其他外设用于其他任何目的。该字节不受 CRC_CTL 寄存器的影响。

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。



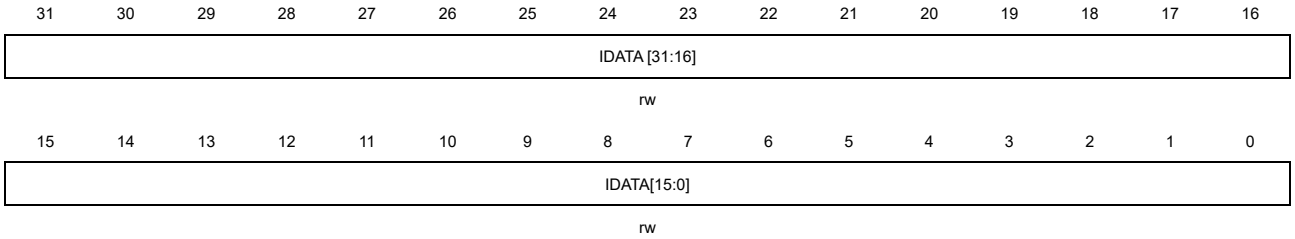
位/位域	名称	描述
31:8	保留	必须保持复位值。
7	REV_O	按位顺序翻转输出数据功能 0：输出数据不翻转 1：输出数据按位顺序翻转
6:5	REV_I[1:0]	翻转输入数据功能 0：输入数据不翻转 1：输入数据按字节翻转 2：输入数据按半字翻转 3：输入数据按字翻转
4:3	PS[1:0]	多项式长度 0：32 位 1：16（POLY[15:0]用于计数）位 2：8（POLY[7:0]用于计数）位 3：7（POLY[6:0]用于计数）位
2:1	保留	必须保持复位值。
0	RST	软件可读写 该位用来复位 CRC_DATA 寄存器。 置位时，CRC_DATA 寄存器的值将自动初始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



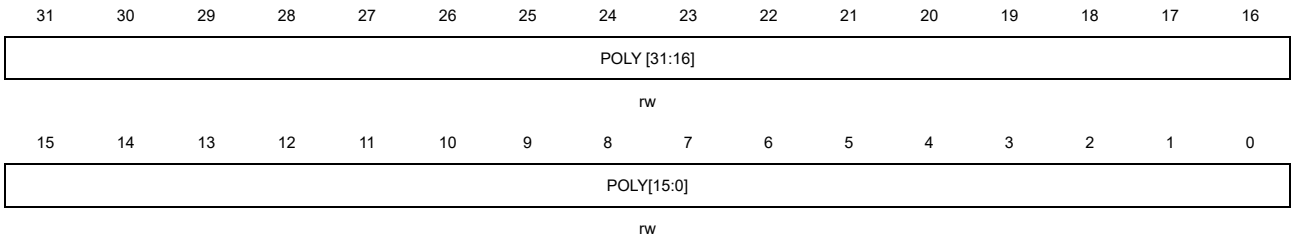
位/位域	名称	描述
31:0	IDATA[31:0]	配置 CRC 初值 CRC_CTL 寄存器的 RST 位置位后，CRC_DATA 寄存器的值将被更新为此寄存器的值。

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



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. 可配置逻辑阵列（CLA）

11.1. 简介

可配置逻辑阵列为外部引脚、CMP、ADC 和定时器提供 256 个可编程数字逻辑操作，而无需 CPU 干预。本模块实现了四个独立的 CLA 单元。每个 CLA 单元支持 GPIO 引脚的可配置异步和同步输出。

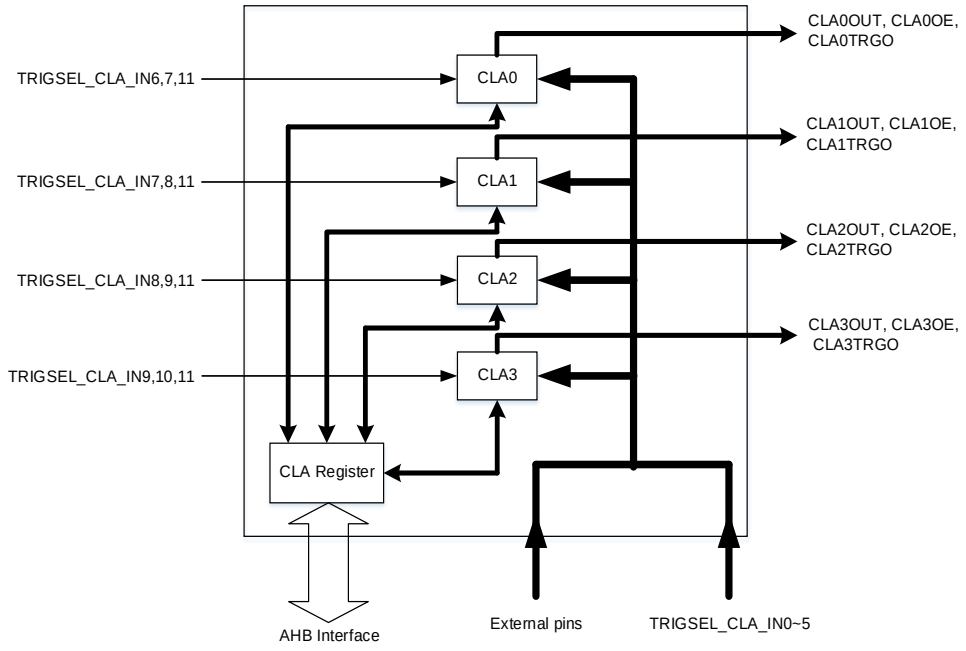
11.2. 主要特性

- 四个独立的 CLA 单元，每个 CLA 单元具有两个信号选择器，支持 16 个输入信号，包括外部引脚、定时器通道、CMP、ADC 和 CLA 异步输出；
- 在每个 CLA 单元中实现逻辑配置单元（LCU），提供 256 个可编程数字逻辑功能；
- 可编程异步和同步输出；
- 可以将 CLA 输出配置为与外部引脚和定时器同步；
- 四个 CLA 单元可以组合并支持复杂的逻辑操作。

11.3. 模块框图

CLA 接口的内部结构如所示。

图 11-1. CLA 模块框图



11.4. 功能描述

此模块中包含四个相同的CLA单元。每个CLA单元实现了两个信号选择器。此外，每个CLA还包括一个LCU。

11.4.1. CLA 输入信号选择器

每个 CLA 单元包括两个信号选择器：SIGS0 和 SIGS1。每个信号选择器的输入可以为：外部引脚、定时器通道(TIMERx_CHx)、定时器触发信号(TIMERx_TRGO)、ADC 转换信号(ADC_CONV)和 CLA 单元异步输出(CLAx_ASYNC_OUT)。当另一个 CLA 单元的异步输出作为 SIGS 的输入时，可以实现一个复杂的组合逻辑运算。当选择 TIMERx_TRGO 作为 SIGS 的输入时，只有第一个 HCLK 周期有效，其余的 TIMERx_TRGO 高被视为逻辑低。

*表 11-1. CLAxSIGS0 输入选择*和*表 11-2. CLAxSIGS1 输入选择*显示了 CLAxSIGS0 和 CLAxSIGS1 的输入选择。TRIGSEL_CLA_IN0~TRIGSEL_CLA_IN11 信号来自如*图 11-1. CLA 模块框图*所示的 TRIGSEL 模块。

表 11-1. CLAxSIGS0 输入选择

SIGS0[3:0]	CLA0SIGS0	CLA1SIGS0	CLA2SIGS0	CLA3SIGS0
0000	CLA0_ASYNC_OUT	CLA0_ASYNC_OUT	CLA0_ASYNC_OUT	CLA0_ASYNC_OUT
0001	CLA1_ASYNC_OUT	CLA1_ASYNC_OUT	CLA1_ASYNC_OUT	CLA1_ASYNC_OUT
0010	CLA2_ASYNC_OUT	CLA2_ASYNC_OUT	CLA2_ASYNC_OUT	CLA2_ASYNC_OUT
0011	CLA3_ASYNC_OUT	CLA3_ASYNC_OUT	CLA3_ASYNC_OUT	CLA3_ASYNC_OUT
0100	TRIGSEL_CLA_IN7	TRIGSEL_CLA_IN8	TRIGSEL_CLA_IN9	TRIGSEL_CLA_IN10
0101	TRIGSEL_CLA_IN0	TRIGSEL_CLA_IN0	TRIGSEL_CLA_IN1	TRIGSEL_CLA_IN2
0110	TRIGSEL_CLA_IN1	TRIGSEL_CLA_IN3	TRIGSEL_CLA_IN3	TRIGSEL_CLA_IN4
0111	TRIGSEL_CLA_IN2	TRIGSEL_CLA_IN4	TRIGSEL_CLA_IN5	TRIGSEL_CLA_IN5
1000	CLAIN0	CLAIN4	CLAIN0	CLAIN2
1001	CLAIN2	CLAIN5	CLAIN1	CLAIN3
1010	CLAIN4	CLAIN8	CLAIN8	CLAIN6
1011	CLAIN6	CLAIN10	CLAIN9	CLAIN7
1100	CLAIN8	CLAIN12	CLAIN14	CLAIN10
1101	CLAIN10	CLAIN13	CLAIN15	CLAIN11
1110	CLAIN12	CLAIN16	CLAIN16	CLAIN18
1111	CLAIN14	CLAIN18	CLAIN17	CLAIN19

表 11-2. CLAxSIGS1 输入选择

SIGS1[3:0]	CLA0SIGS1	CLA1SIGS1	CLA2SIGS1	CLA3SIGS1
0000	CLA0_ASYNC_OUT	CLA0_ASYNC_OUT	CLA0_ASYNC_OUT	CLA0_ASYNC_OUT
0001	CLA1_ASYNC_OUT	CLA1_ASYNC_OUT	CLA1_ASYNC_OUT	CLA1_ASYNC_OUT

0010	CLA2_ASYNC_OUT	CLA2_ASYNC_OUT	CLA2_ASYNC_OUT	CLA2_ASYNC_OUT
0011	CLA3_ASYNC_OUT	CLA3_ASYNC_OUT	CLA3_ASYNC_OUT	CLA3_ASYNC_OUT
0100	TRIGSEL_CLA_IN11	TRIGSEL_CLA_IN11	TRIGSEL_CLA_IN11	TRIGSEL_CLA_IN11
0101	TRIGSEL_CLA_IN3	TRIGSEL_CLA_IN1	TRIGSEL_CLA_IN0	TRIGSEL_CLA_IN0
0110	TRIGSEL_CLA_IN4	TRIGSEL_CLA_IN2	TRIGSEL_CLA_IN2	TRIGSEL_CLA_IN1
0111	TRIGSEL_CLA_IN5	TRIGSEL_CLA_IN5	TRIGSEL_CLA_IN4	TRIGSEL_CLA_IN3
1000	CLAIN1	CLAIN6	CLAIN2	CLAIN0
1001	CLAIN3	CLAIN7	CLAIN3	CLAIN1
1010	CLAIN5	CLAIN9	CLAIN10	CLAIN4
1011	CLAIN7	CLAIN11	CLAIN11	CLAIN5
1100	CLAIN9	CLAIN14	CLAIN12	CLAIN8
1101	CLAIN11	CLAIN15	CLAIN13	CLAIN9
1110	CLAIN13	CLAIN17	CLAIN18	CLAIN16
1111	CLAIN15	CLAIN19	CLAIN19	CLAIN17

11.4.2. 逻辑控制单元（LCU）控制

每个 CLA 单元都有一个 LCU，LCU 引入了 256 种逻辑组合功能，通过 CLAx_LCUCTL 寄存器的 LCU [7:0]位控制。LCU 有三个输入源，当 CLA 单元被禁用时，它们对 LCU 输出没有影响。

- Input0: SIGS0输出
- Input1: SIGS1输出
- Input2: CLA结果来自前一个CLA单元（CLA[x-1]）的LCU结果

如果 CLA 单元被禁用，则 LCU 输入强制为“0”。

LCU [7:0]的 bit7~0 控制 input0(IN0)、input1(IN1)、input2(IN2)的哪个逻辑功能可以影响输出，如*表 11-3. LCU 控制*所示。

表 11-3. LCU 控制

LCU[7:0]	input 0	input 1	input 2
bit 0	0	0	0
bit 1	0	0	1
bit 2	0	1	0
bit 3	0	1	1
bit 4	1	0	0
bit 5	1	0	1
bit 6	1	1	0
bit 7	1	1	1

例如，当{IN0, IN1, IN2} == 3’b000 时，(IN0^IN1^IN2)的结果为 1’b0，所以 LCU[7:0]的 bit 0 为 0；当{IN0, IN1, IN2} == 3’b001 时，(IN0^IN1^IN2)的结果为 1’b1，所以 LCU[7:0]的 bit 1 为 1；当{IN0,

IN1, IN2} == 3'b010 时，(IN0^IN1^IN2)的结果为 1'b1，所以 LCU[7:0]的 bit 2 为 1；当{IN0, IN1, IN2} == 3'b011 时，(IN0^IN1^IN2)的结果为 1'b0，所以 LCU[7:0]的 bit 3 为 0；当{IN0, IN1, IN2} == 3'b100 时，(IN0^IN1^IN2)的结果为 1'b1，所以 LCU[7:0]的 bit 4 为 1；当{IN0, IN1, IN2} == 3'b101 时，(IN0^IN1^IN2)的结果为 1'b0，所以 LCU[7:0]的 bit 5 为 0；当{IN0, IN1, IN2} == 3'b110 时，(IN0^IN1^IN2)的结果为 1'b0，所以 LCU[7:0]的 bit 6 为 0；当{IN0, IN1, IN2} == 3'b111 时，(IN0^IN1^IN2)的结果为 1'b1，所以 LCU[7:0]的 bit 7 为 1。因此，如果要实现逻辑函数(IN0^IN1^IN2)，那么 CLAx_LCUCTL 的 LCU[7:0]应该被配置为 8'b10010110。

11.4.3. CLA 输出

每个 CLA 单元都有一个 GPIO 引脚作为其输出，该输出可以是 LCU 的结果，也可以是经过触发器后的 LCU 结果。这可以通过 CLAx_CTL 寄存器的 OSEL 位选择。

触发器的时钟源和时钟极性可以分别通过 CLAx_CTL 寄存器的 CSEL[1:0]位和 CPOL 位来选择。有四个时钟可以作为触发器的时钟源，它们分别是：

- CLA[x-1]的结果：前一个 CLA 单元的 LCU 结果（CLA3 的 LCU 结果发送给 CLA0）
- SIGS0：多路选择器 SIGS0 的输出
- HCLK
- TIMER_TRGO：TRIGSEL_CLA_IN6 对应 CLA0，TRIGSEL_CLA_IN7 对应 CLA1，TRIGSEL_CLA_IN8 对应 CLA2，TRIGSEL_CLA_IN9 对应 CLA3

给寄存器 CLAx_CTL 的位 FFRST 写 1 可以复位触发器的输出。如果触发器的输出作为 CLA 的输出，建议在使能 CLA 单元之前置位 FFRST。

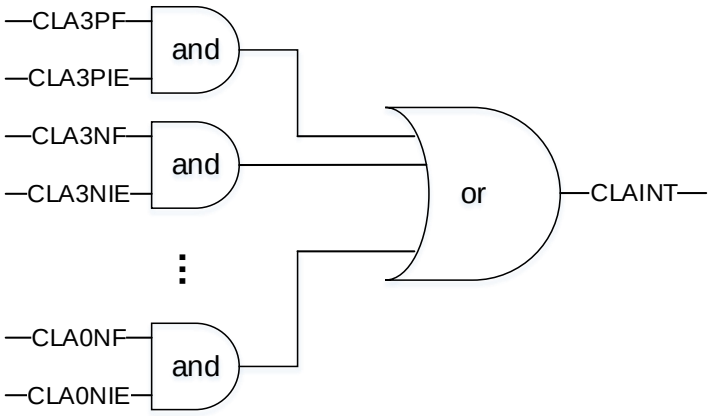
如果 OEN 位复位，那么 CLAxOE 也复位，对应的 GPIO 引脚输出失能。

每个 CLA 单元都有一个 HCLK 同步输出信号 CLAx_TRGO，该信号作为触发信号被发送到发送到 TRIGSEL，TRIGSEL 允许软件选择输出到 ADC、DAC 和 SYSCFG。

11.4.4. 中断

CLA_INTF 寄存器中有八个中断标志位，它们分别是 CLA3PF、CLA3NF、CLA2PF、CLA2NF CLA1PF、CLA1NF、CLA0PF、CLA0NF。对于每一个标志位，在 CLA_GCTL 寄存器中都有一个对应的中断使能位。CLA 中断逻辑如*图 11-2. CLA 中断逻辑*所示，当检测到标志位置位并且对应的中断使能位使能，就会产生中断。

图 11-2. CLA 中断逻辑



11.5. CLA 寄存器

CLA基地址：0x4003 8000

11.5.1. 全局控制寄存器 (CLA_GCTL)

地址偏移：0x00
复位值：0x0000 0000（必须是power reset）

该寄存器只能按字(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												CLA3EN	CLA2EN	CLA1EN	CLA0EN
												rw	rw	rw	rw

位/位域	名称	描述
31:4	保留	必须保持复位值
3	CLA3EN	CLA3单元使能 该位由软件置位和清除 0：禁能CAL3单元 1：使能CLA3单元
2	CLA2EN	CLA2单元使能 该位由软件置位和清除 0：禁能CAL2单元 1：使能CLA2单元
1	CLA1EN	CLA1单元使能 该位由软件置位和清除 0：禁能CAL1单元 1：使能CLA1单元
0	CLA0EN	CLA0单元使能 该位由软件置位和清除 0：禁能CAL0单元 1：使能CLA0单元

11.5.2. 中断使能寄存器(CLA_INTE)

地址偏移：0x04

复位值：0x0000 0000（必须是power reset）

该寄存器只能按字(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								CLA3PIE	CLA3NIE	CLA2PIE	CLA2NIE	CLA1PIE	CLA1NIE	CLA0PIE	CLA0NIE
								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	CLA3PIE	CLA3单元上升沿中断使能 该位由软件置位和清除 0: 禁能CLA3单元上升沿中断 1: 使能CLA3单元上升沿中断。当CLA3PF位置位时，会产生中断。
6	CLA3NIE	CLA3单元下降沿中断使能 该位由软件置位和清除 0: 禁能CLA3单元下降沿中断 1: 使能 CLA3 单元下降沿中断。当 CLA3NF 位置位时，会产生中断。
5	CLA2PIE	CLA2单元上升沿中断使能 该位由软件置位和清除 0: 禁能CLA2单元上升沿中断 1: 使能 CLA2 单元上升沿中断。当 CLA2PF 位置位时，会产生中断。
4	CLA2NIE	CLA2单元下降沿中断使能 该位由软件置位和清除 0: 禁能CLA2单元下降沿中断 1: 使能 CLA2 单元下降沿中断。当 CLA2NF 位置位时，会产生中断。
3	CLA1PIE	CLA1单元上升沿中断使能 该位由软件置位和清除 0: 禁能CLA1单元上升沿中断 1: 使能 CLA1 单元上升沿中断。当 CLA1PF 位置位时，会产生中断。
2	CLA1NIE	CLA1单元下降沿中断使能 该位由软件置位和清除 0: 禁能CLA1单元下降沿中断 1: 使能 CLA1 单元下降沿中断。当 CLA1NF 位置位时，会产生中断。
1	CLA0PIE	CLA0单元上升沿中断使能 该位由软件置位和清除

0: 禁能CLA0单元上升沿中断
1: 使能 CLA0 单元上升沿中断。当 CLA0PF 位置位时，会产生中断。

0	CLA0NIE	CLA0单元下降沿中断使能 该位由软件置位和清除 0: 禁能CLA0单元下降沿中断 1: 使能 CLA0 单元下降沿中断。当 CLA0NF 位置位时，会产生中断。

11.5.3. 中断标志寄存器 (CLA_INTF)

地址偏移：0x08
复位值：0x0000 0000（必须是power reset）

该寄存器只能按字(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								CLA3PF	CLA3NF	CLA2PF	CLA2NF	CLA1PF	CLA1NF	CLA0PF	CLA0NF
								rs_w0	rs_w0	rs_w0	rs_w0	rs_w0	rs_w0	rw_w0	rs_w0

位/位域	名称	描述
31:8	保留	必须保持复位值
7	CLA3PF	CLA3单元上升沿标志 该位由硬件置位，软件清除 0: 没有检测到CLA3的上升沿输出 1: 检测到CLA3的上升沿输出
6	CLA3NF	CLA3单元下降沿标志 该位由硬件置位，软件清除 0: 没有检测到CLA3的下降沿输出 1: 检测到CLA3的下降沿输出
5	CLA2PF	CLA2单元上升沿标志 该位由硬件置位，软件清除 0: 没有检测到CLA2的上升沿输出 1: 检测到CLA2的上升沿输出
4	CLA2NF	CLA2单元下降沿标志 该位由硬件置位，软件清除 0: 没有检测到CLA2的下降沿输出

1: 检测到CLA2的下降沿输出

3	CLA1PF	CLA1单元上升沿标志 该位由硬件置位，软件清除 0: 没有检测到CLA1的上升沿输出 1: 检测到CLA1的上升沿输出
2	CLA1NF	CLA1单元下降沿标志 该位由硬件置位，软件清除 0: 没有检测到CLA1的下降沿输出 1: 检测到CLA1的下降沿输出
1	CLA0PF	CLA0单元上升沿标志 该位由硬件置位，软件清除 0: 没有检测到CLA0的上升沿输出 1: 检测到CLA0的上升沿输出
0	CLA0NF	CLA0单元下降沿标志 该位由硬件置位，软件清除 0: 没有检测到CLA0的下降沿输出 1: 检测到CLA0的下降沿输出

11.5.4. 状态寄存器 (CLA_STAT)

地址偏移：0x0C
复位值：0x0000 0000（必须是power reset）

该寄存器只能按字(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												CLA3OUT	CLA2OUT	CLA1OUT	CLA0OUT
												r	r	r	r

位/位域	名称	描述
31:4	保留	必须保持复位值
3	CLA3OUT	CLA3单元输出状态 该位由硬件置位和清除 0: CLA3单元的当前逻辑电平为低 1: CLA3单元的当前逻辑电平为高

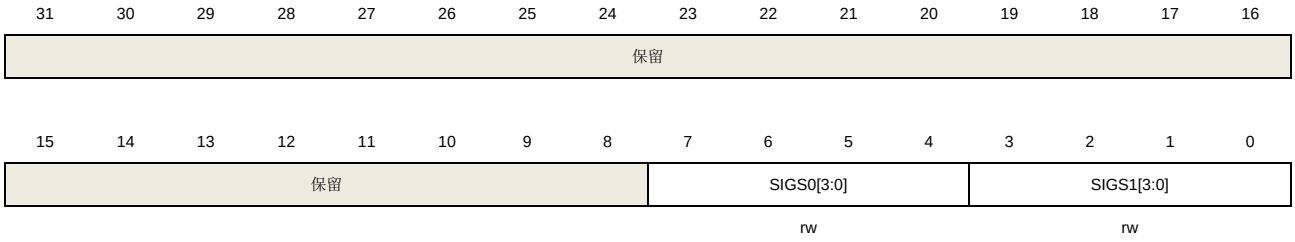
2	CLA2OUT	CLA2单元输出状态 该位由硬件置位和清除 0: CLA2单元的当前逻辑电平为低 1: CLA2单元的当前逻辑电平为高
1	CLA1OUT	CLA1单元输出状态 该位由硬件置位和清除 0: CLA1单元的当前逻辑电平为低 1: CLA1单元的当前逻辑电平为高
0	CLA0OUT	CLA0单元输出状态 该位由硬件置位和清除 0: CLA0单元的当前逻辑电平为低 1: CLA0单元的当前逻辑电平为高

11.5.5. 信号选择寄存器 (CLAx_SIGS)(x=0..3)

地址偏移：0x10 + 0x0C * x

复位值：0x0000 0000（必须是power reset）

该寄存器只能按字(32 位)访问。



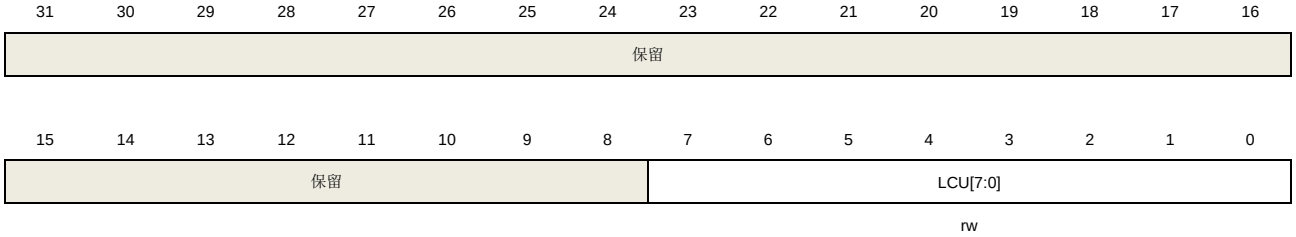
位/位域	名称	描述
31:8	保留	必须保持复位值
7:4	SIGS0[3:0]	信号选择器0输入选择 该位由软件置位和清除 这些位选择信号选择器0的输入。 请参考***表11-1. CLAxSIGS0输入选择***
3:0	SIGS1[3:0]	信号选择器1输入选择 该位由软件置位和清除 这些位选择信号选择器1的输入。 请参考***表11-2. CLAxSIGS1输入选择***

11.5.6. LCU 控制寄存器 (CLAx_LCUCTL)(x=0..3)

地址偏移：0x14 + 0x0C * x

复位值：0x0000 0000（必须是power reset）

该寄存器只能按字(32 位)访问。



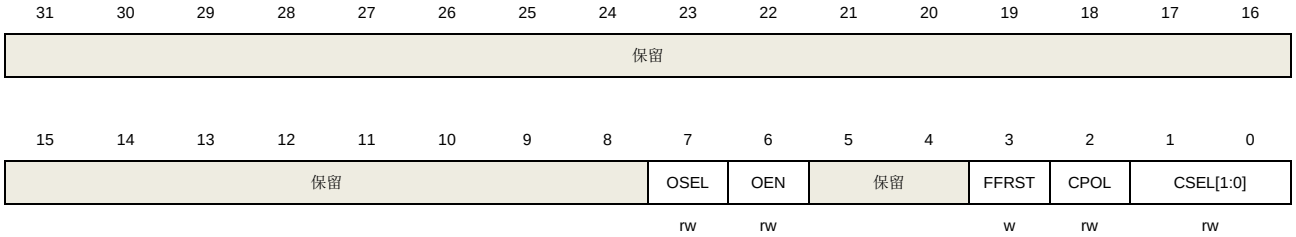
位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	LCU[7:0]	LCU控制 由软件置位和清除 这些位控制input0、input1、input2的哪个逻辑函数会对输出产生影响。 例如： IN1 | IN2: LCU = 8'b11101110 IN0 & (IN1 ^ IN2): LCU = 8'b01100000

11.5.7. 控制寄存器 (CLAx_CTL)(x=0..3)

地址偏移：0x18 + 0x0C * x

复位值：0x0000 0000（必须是power reset）

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:8	保留	必须保持复位值
7	OSEL	输出选择 由软件置位和清除 0: 触发器的输出作为CLAx的输出

		1: LCU的结果作为CLAx的输出
6	OEN	输出使能 由软件置位和清除 0: CLAx输出禁能 1: CLAx输出使能
5:4	保留	必须保持复位值
3	FFRST	触发器输出复位 0: 没有作用 1: 复位触发器输出异步
2	CPOL	触发器时钟极性选择 由软件置位和清除 当CLAx使能时(CLAxEN = 1)禁止修改该位域 0: 时钟上升沿有效 1: 时钟下降沿有效
1:0	CSEL[1:0]	触发器时钟源选择 由软件置位和清除 当CLAx使能时(CLAxEN = 1)禁止修改该位域 00: CLA[x-1] LCU的结果 01: SIGS0输出 10: HCLK 11: TIMER_TRGO

12. 真随机数生成器（TRNG）

12.1. 简介

真随机数发生器模块(TRNG)能够通过连续模拟噪声生成一个 32 位的随机数值，该模块遵循 NIST SP800-90B 标准。

12.2. 主要特征

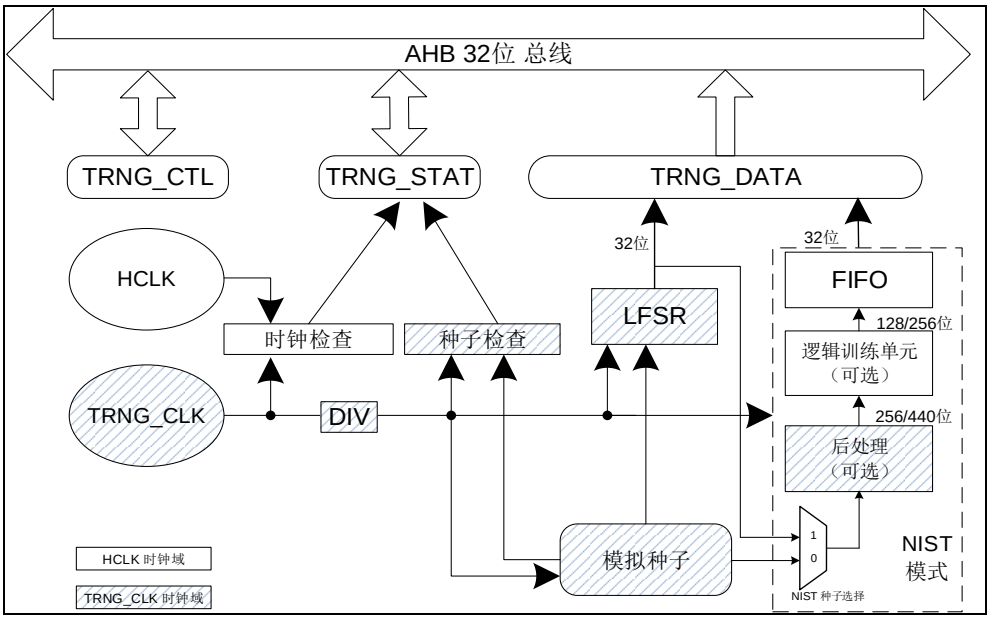
- LFSR模式和NIST模式两种可选模式产生随机数；
- LFSR模式下两个连续随机数的间隔大约为40个TRNG_CLK时钟周期；
- LFSR模式下每次可以产生32位随机数；
- TRNG模块中NIST模式遵循NIST SP800-90B标准；
- 支持NIST SP800-90B标准推荐的健康测试；
- NIST模式下每次可产生32位 * 4或32位 * 8的随机数；
- TRNG模块具有健康测试功能，并有相关的错误标志；
- 关闭TRNG模块可以一定程度上降低功耗；
- 通过模拟信号噪声生成128位随机数种子用于产生真随机数。

12.3. 限制

HCLK 时钟频率应大于从内部时钟分频的 TRNG_CLK 时钟两倍。

12.4. 功能说明

图 12-1. TRNG 模块框图



真随机数（TRNG）模块有两种工作模式，分别是 NIST 模式和 LFSR 模式。

在 NIST 模式中，模拟随机数种子由 4 路噪声源信号产生。噪声源信号会先进行异或运算得到 1 位模拟种子，这些种子随即被送入到逻辑训练单元，最终输出 128 位或 256 位数据。在逻辑训练单元中利用遵循 NIST SP800-90B 标准的哈希运算以增加随机数的熵值。当 TRNG_STAT 寄存器中的准备标志位（DRDY）置位后，可以通过连续 4 次或 8 次读取 TRNG_DATA 寄存器值将生成的随机数全部读出。

在 LFSR 模式中，模拟随机数种子会被加入到一个线性反馈移位寄存器中（LFSR）并产生一个 32 位随机数，随后当移位寄存器再次被填满数据后该 32 位随机数会被传送至 TRNG_DATA 寄存器中。

该模拟种子由几个环形振荡器的输出生成。LFSR 由可配置的 TRNG_CLK 时钟（*复位和时钟单元（RCU）*相关章节）驱动，因此随机数质量仅与 TRNG_CLK 时钟有关，与 HCLK 频率无关。

在 NIST 模式中，被送入逻辑训练单元的种子源可以通过设置 TRNG_CTL 寄存器的 SEEDSEL 位进行选择。除了模拟种子之外，LFSR 产生的值也可以作为 NIST 模式的种子源。

12.4.1. 线性反馈移位寄存器（LFSR）

线性反馈移位寄存器是一种具有组合逻辑的顺序移位寄存器，可使其伪随机循环二进制值序列，这种操作可以增加随机数的熵值。TRNG 每次产生一个 32 位随机数。

12.4.2. 后处理

当该功能打开后，会将从噪声源中采集到的数据一半保持原样，一半进行取反操作。当生成的随机数已经准备好后，即使没有被读出，后处理单元也会开始新一轮随机数种子采集，该操作可以提高 TRNG 模块的工作效率。

12.4.3. 训练单元

TRNG 模块中的训练单元内部会进行一系列的哈希运算，用以增加输出的固定长度比特流。符合 NIST SP800-90B 的目标输出是满熵的。

通过配置 TRNG_CTL 寄存器中的 ALGO[1:0]和 INIT 位对哈希运算进行配置，随机数的输入比特流宽度可设置为 256 比特或 440 比特，由 TRNG_CTL 寄存器中的 INMOD 位配置，同时，输出的比特流宽度可设置为 128 比特或 256 比特，由 TRNG_CTL 寄存器中的 OUTMOD 位配置。

12.4.4. 输出 FIFO

在 NIST 模式中，数据输出 FIFO 的宽度是 4 个字（4 * 32bit）或 8 个字（8 * 32bit），由 TRNG_CTL 寄存器中的 OUTMOD 位配置。输出的随机数被临时存储在 FIFO 中，当通过 TRNG_DATA 寄存器将 FIFO 中的数据全部读出时，新一轮的训练过程会自动开始，然后训练出的新的 128 位或 256 位数据会被放入到 FIFO 中等待下一次的读取。

当一个随机数已准备好并可以通过 TRNG_DATA 寄存器读出时，TRNG_STAT 寄存器中的 DRDY 位会被置 1，在 NIST 模式下，该位会在输出 FIFO 非空前一直保持置位直到通过连续 4 次或 8 次读取 TRNG_DATA 寄存器将随机数完全读出为止。在 LFSR 模式下，该位在随机数从 TRNG_DATA 寄存器中被读出前保持置位。

在 NIST 模式中产生一次随机数需要花费大约 10+输入的种子数量（由 TRNG_CTL 寄存器中的 INMOD 位决定）个 TRNG_CLK 再加 70 个 HCLK。在 LFSR 模式中产生一个 32 比特的随机数则需要约 40 个 TRNG_CLK。

TRNG_CTL 寄存器中的 ALGO 位域的不同配置对应训练单元中不同的哈希算法，选择不同的算法后 TRNG 模块会产生不同长度的有效结果，并将有效位放在 FIFO 中，所以首先被读出的数据均为有效数据。用户需要根据自己的实际情况选择相应的算法。详细请参考*表 12-1. ALGO 配置*。

表 12-1. ALGO 配置

ALGO	00	01	10	11
算法	SHA1	MD5	SHA224	SHA256
有效长度	160	128	224	256

12.4.5. 健康测试

健康测试组件可以保证 TRNG 模块的稳定运行，并可以快速检测到错误的发生。

TRNG 模块中的健康测试组件同样遵循 NIST SP800-90B 标准，相关的推荐阈值描述参考 TRNG_HTCFG 寄存器中的描述。

1. 启动阶段健康测试：该测试会在芯片复位后且第一次启动 TRNG 模块前执行一次
- 自适应比例测试：TRNG 模块会验证输出的噪声源信号第一个比特位在长度为 1024 的比特流中重复出现的次数不超过阈值设定，该阈值的设定参考 TRNG_HTCFG 寄存器中 APTTH 位域的描述，默认值为 691 次。若超出阈值将会产生一次错误并且相应的错误标志会被置位；
- 重复计数测试：如果噪声源信号连续输出了超过设定的阈值数量的比特位（0 或 1）后，将会产生一次错误并且相应的错误标志会被置位，该阈值的设定参考 TRNG_HTCFG 寄存器中 REPTH 位域的描述，默认值为 40 次；
- 替换测试：TRNG 模块可以将一定数量的输入到训练单元的噪声源信号替换成已经过测试的信号输入，并将该信号通过训练单元输出的结果与该信号在不同算法下生成的最佳输出结果进行对比。

2. 持续健康测试：该测试在 TRNG 模块运行期间持续对噪声源输出进行测试
- 自适应比例测试：参考启动阶段健康测试中的自适应比例测试描述；
- 重复计数测试：参考启动阶段健康测试中的重复计数测试描述。

3. GD 自定义健康测试：该测试由 GD 自定义实施
- 转移计数测试：如果噪声源信号连续出现了 32 次两位模板信号（01 或 10）时，将产生一次错误且相应的错误标志位会被置位；
- 时钟检测：如果 TRNG 模块的时钟分频前低于 AHB 时钟的 16 分之一，将产生一次错误且相应的错误标志位会被置位。

4. 重启测试
- 支持重新启动熵源和重新运行启动阶段健康测试。

注意：

- 在 NIST 模式下，当有错误出现时，TRNG_STAT 寄存器中的 ERRSTA 位将会置位，若使能错误中断，则每个错误可以产生一次错误中断；
- 当替换测试使能时，产生的随机数仅用于验证训练单元的功能性。测试完成后，产生的随机数应当丢弃，不能作为真随机数使用。

12.4.6. NIST 模式状态

NIST 模式下各阶段状态转换如下所列：

1. TRNG 的初始状态为空闲状态；
2. 通过置位 TRNG_CTL 寄存器中的 RNGEN 位使能 TRNG 后，TRNG 进入热身状态，该状态

表明模拟信号发生器开始初始化；

3. 在经历了 16 个未经分频的 TRNG_CLK 时钟周期后，状态切换为启动状态，并且开始执行健康测试，该测试将耗费 1024 个分频后的 TRNG_CLK 时钟周期；
4. 然后状态切换至样本生成状态并开始生成随机数样本。当输出 FIFO 为空时 TRNG 模块会生成新的随机数。

注意：

- 模块运行期间 TRNG_CTL 寄存器的 CLKDIV[3:0]位域不能够被修改；
- 出于减少第一个样本生成时间的考虑，在切换至启动状态也会产生一次随机数，即使 FIPS PUB 140-2 标准建议丢弃此次样本；
- 如果 TRNG_CTL 寄存器中的 RTEN 位置 1，第一次输出的随机数则为替换测试中的输出，该随机数需要被丢弃。

12.4.7. 操作流程

以下步骤为 TRNG 模块的推荐操作流程：

1. 将 TRNG_CTL 寄存器中的 CONDRST 位置 1；
2. 在 TRNG_CTL 寄存器中写入需要的配置，例如模块功耗、时钟分频系数、工作模式、输入 / 输出位宽、算法等；
3. 将 TRNGEN 位置位；
4. 将 CONDRST 位清零使配置生效；
5. 检测 TRGN_STAT 寄存器，如果 SEIF、CEIF、ERRSTA、SECS、CECS 均为 0 并且 DRDY=1 那么 TRNG_DATA 寄存器中的随机数可以被读取。

当 TRNG_CTL 寄存器中的 IE 位置位时，发生如下情况则会产生一次中断：

- 成功生成随机数，且 TRNG_STAT 寄存器中的 DRDY 位置位；
- 产生一次种子错误，且 TRNG_STAT 寄存器中的 SEIF 和 ERRSTA 位置位；
- 产生一次时钟错误，且 TRNG_STAT 寄存器中的 CEIF 和 ERRSTA 位置位。

按照 FIPS PUB 140-2 标准的要求，数据寄存器中的第一个随机数需要保留而不是被使用。每一个新生成的随机数应当与之前的随机数相比较。只有当该随机数与前一个随机数不相等时，该数据才可被使用。

12.4.8. 错误标志

（1）时钟错误

当 TRNG_CLK 时钟频率低于 HCLK 频率的 1 / 16 时，CECS 和 CEIF 位将被置 1。此时，软件应当检查 TRNG_CLK 和 HCLK 时钟频率配置并清除 CEIF 位。时钟错误对上一个产生的随机数没有影响。

（2）种子错误

当模拟种子的值在 64 个 TRNG_CLK 时钟周期内不发生变化或连续不断的翻转，SECS 和 SEIF 位将被置位。在这种情况下，数据寄存器中的随机数值不应当被使用。此时软件需要通过将 CONDRST 位写 1 后再写 0 来执行一次 TRNG 模块软复位，并且在完成操作后清除 SEIF 位，之后等待 TRNG 模块清除 TRNG_STAT 寄存器中的 SECS 位。

12.4.9. 低功耗配置

如果使用者需要考虑功耗问题，除了将 CLKDIV[3:0]位域配置为尽可能大的分频系数外，也可以在 TRNG_STAT 寄存器中的 DRDY 位置位后关闭 TRNG，模块会仍旧保持关闭前的状态，并且已经生成的随机数依旧可以通过 TRNG_DATA 寄存器读取。当需要一个新的随机数时，重新启动 TRNG 模块即可。

12.5. TRNG 寄存器

TRNG 基地址：0x4802 1800

12.5.1. 控制寄存器（TRNG_CTL）

地址偏移：0x00
复位值：0x0300 0410

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTLLK	CONDRST	保留				NR[1:0]		保留				CLKDIV[3:0]			
rs	rw					rw						rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INMOD	OUTMOD	ALGO[1:0]		保留	CONDEN	PPEN	INIT	RTEN	SEEDSEL	CED	MODSEL	IE	TRNGEN	保留	
rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw		

位/位域	名称	描述
31	CTLLK	TRNG_CTL 寄存器锁定位 0：允许写位域[29:4] 1：锁定位域[29:4]，且对该位域写操作会被忽略
30	CONDRST	复位逻辑训练单元 先置 1 后再清 0 以复位逻辑训练单元。需要注意的是，TRNG_HTCFG 寄存器和 TRNG_CTL 寄存器的位域[29:4]只能在该位为 1 时被改写。
29:26	保留	必须保持复位值。
25:24	NR[1:0]	TRNG 模块功耗模式，复位值：2b’ 11 00：极低 01：低 10：中 11：高
23:20	保留	必须保持复位值。
19:16	CLKDIV[3:0]	TRNG 时钟分频系数 0000：TRNG 时钟 2^0 分频 0001：TRNG 时钟 2^1 分频 ……

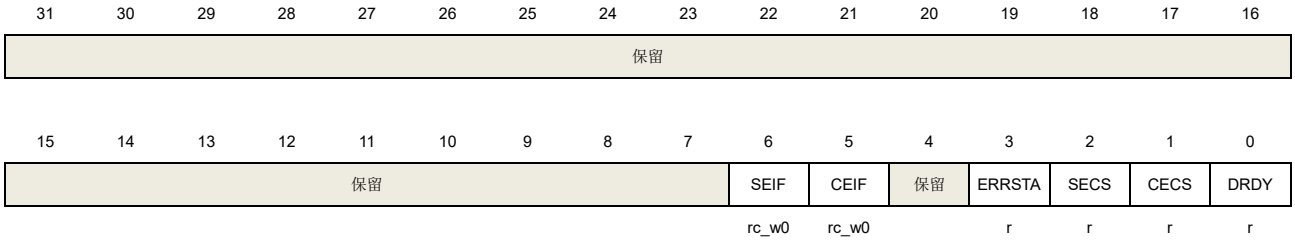
		1111：TRNG 时钟 2^{15} 分频
15	INMOD	随机数种子输入模式选择 0：向训练单元输入 256 比特 1：向训练单元输入 440 比特
14	OUTMOD	随机数输出模式选择 0：从训练单元输出 128 比特 1：从训练单元输出 256 比特
13:12	ALGO[1:0]	训练单元算法选择 00：sha1 算法 01：md5 算法 10：sha224 算法 11：sha256 算法
11	保留	必须保持复位值。
10	CONDEN	启动训练单元 0：失能训练单元 1：使能训练单元
9	PPEN	启动后处理功能 0：失能后处理功能 1：使能后处理功能
8	INIT	使能训练单元时初始化哈希算法 0：不初始化哈希算法 1：初始化哈希算法
7	RTEN	替换测试使能位 0：失能替换测试 1：使能替换测试
6	SEEDSEL	NIST 种子模式选择 0：选择模拟噪声 1：选择 LFSR 输出
5	CED	时钟错误检测 0：失能时钟错误检测 1：使能时钟错误检测
4	MODSEL	TRNG 模式选择 0：LFSR 模式 1：NIST 模式

3	IE	中断使能位，当 DRDY，SEIF，CEIF 或 ERRSTA 位被置位时该位控制生成一个中断。 0：禁止 TRNG 中断 1：使能 TRNG 中断
2	TRNGEN	TRNG 使能位 0：禁止 TRNG 模块（降低功耗） 1：使能 TRNG 模块
1:0	保留	必须保持复位值。

12.5.2. 状态寄存器（TRNG_STAT）

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:7	保留	必须保持复位值。
6	SEIF	种子错误中断标志位 如果超过 64 个连续位具有相同值或超过 32 组连续交替的 0 和 1 被检测到则此位将置 1。 0：未检测到错误 1：检测到种子错误。写 0 将清除该位
5	CEIF	时钟错误中断标志位 如果 TRNG_CLK 时钟频率低于 HCLK 频率的 1/16 时该位被置位。 0：未检测到错误 1：检测到时钟错误，写 0 将清除该位。
4	保留	必须保持复位值。
3	ERRSTA	NIST 模式错误标志，该位可以被 CONDRST 清零 0：NIST 模式中未出现错误 1：NIST 模式中出现错误

2	SECS	种子错误当前状态 0：当前未检测到种子错误。如果 SEIF=1 和 SECS=0，说明之前已经检测到种子错误但现在已恢复正常。 1：当前检测到种子错误。如果超过 64 个连续位具有相同值或超过 32 组连续交替的 0 和 1 被检测到时，该位置 1。
1	CECS	时钟错误当前状态 0：当前未检测到时钟错误。如果 CEIF=1 和 CECS=0，则意味着之前已检测到时钟错误但现在已恢复正常。 1：当前检测到时钟错误。此时 TRNG_CLK 时钟频率低于 1/16 HCLK 频率。
0	DRDY	随机数准备状态位 读 TRNG_DATA 寄存器会清零该位，当一个新的随机数产生时被置位。 0：TRNG 数据寄存器的内容无效 1：TRNG 数据寄存器的内容有效

12.5.3. 数据寄存器（TRNG_DATA）

地址偏移：0x08
复位值：0x0000 0000

在读此寄存器之前，软件必须确保 DRDY 位已置 1。

该寄存器只能按字（32 位）访问。

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

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

位/位域	名称	描述
31:0	TRNDATA[31:0]	32 位随机数据

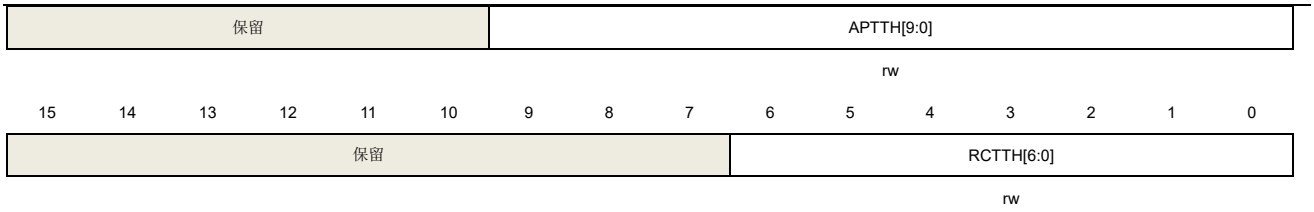
12.5.4. 健康测试配置寄存器（TRNG_HTCFG）

地址偏移：0x10
复位值：0x02B3 0028

在读此寄存器之前，软件必须确保 DRDY 位已置 1。

该寄存器只能按字（32 位）访问。

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



位/位域	名称	描述
31:26	保留	必须保持复位值。
25:16	APTTH[9:0]	自适应比例测试阈值，默认值 691
15:7	保留	必须保持复位值。
6:0	RCTTH[6:0]	重复计数测试阈值，默认值 40

13. 加密处理器（CAU）

13.1. 简介

加密处理单元支持处理DES，三重DES或AES（128，192或256）算法，对数据进行加密或解密。加密处理器完全兼容下列标准：

- 联邦信息处理标准出版物“FIPS PUB 46-3，1999年10月25日”规定的数据加密标准（DES）和三重DES（TDES）。它遵循美国国家标准协会（ANSI）X9.52标准；
- 联邦信息处理标准出版物（FIPS PUB 197，2001年11月26日）规定的高级加密标准（AES）。

CAU处理器可在多种模式下使用DES / 三重DES / 多种长度密钥的AES算法执行数据加密和解密。

CAU外设为32位AHB外设，它支持对输入FIFO和输出FIFO的DMA传输。

13.2. 主要特征

- 支持DES，三重DES和AES加密解密算法；
- 支持DES，三重DES和AES下的多种模式，包括电子密码本（ECB）、加密分组链接（CBC）模式、计数器模式（CTR）、伽罗瓦 / 计数器模式（GCM）、伽罗瓦消息验证码模式（GMAC）、加密分组链接-消息验证码模式（CCM）、密码反馈模式（CFB）和输出反馈模式（OFB）；
- 输入与输出FIFO支持DMA传输。

DES/三重DES

- 支持电子密码本（ECB）或加密分组链接（CBC）模式；
- 支持在CBC模式下使用2×32位初始化向量（IV）；
- 输入FIFO和输出FIFO可存储8x32位数据；
- 对于输入/输出FIFO的数据支持半字、字节、位交换或不交换；
- 数据可通过DMA或CPU中断进行传输，也可以不通过两者进行传输。

AES

- 支持支持电子密码本（ECB）、加密分组链接（CBC）模式、计数器模式（CTR）、伽罗瓦/计数器模式（GCM）、伽罗瓦消息验证码模式（GMAC）、加密分组链接-消息验证码模式（CCM）、密码反馈模式（CFB）和输出反馈模式（OFB）；
- 支持128位、192位或256位密钥；
- 支持在CBC、CTR、GCM、GMAC、CCM、CFB和OFB模式下使用4×32位初始化向量(IV)；
- 输入和输出FIFO各8字深；
- 对于输入/输出FIFO的数据支持半字、字节、位交换或不交换；
- 数据可通过DMA或CPU中断进行传输，也可以不通过两者进行传输。

13.3. CAU 数据类型和初始化向量

13.3.1. 数据类型

CAU 处理器一次输入 32 位（字）数据，DES 每 64 位对数据流进行处理，AES 每 128 位对数据流进行处理。对于每个数据块，在其进入 CAU 处理器之前，可对这些数据执行位、字节、半字交换或不交换操作（取决于要加密的数据类型）。在 CAU 数据写入 OUT FIFO 之前，需要对其执行同样的交换操作。注意由于系统存储器结构采用小端模式，无论使用何种数据类型，最低有效数据均占用最低地址位置。

*图 13-1. DATAM 不交换/半字交换*和*图 13-2. DATATM 字节交换/位交换*介绍了 128 位 AES 块在不同数据类型下的数据交换。（对于 DES，数据块大小为 2 个 32 位字，请参考图中前两个字的数据交换）

图 13-1. DATAM 不交换/半字交换

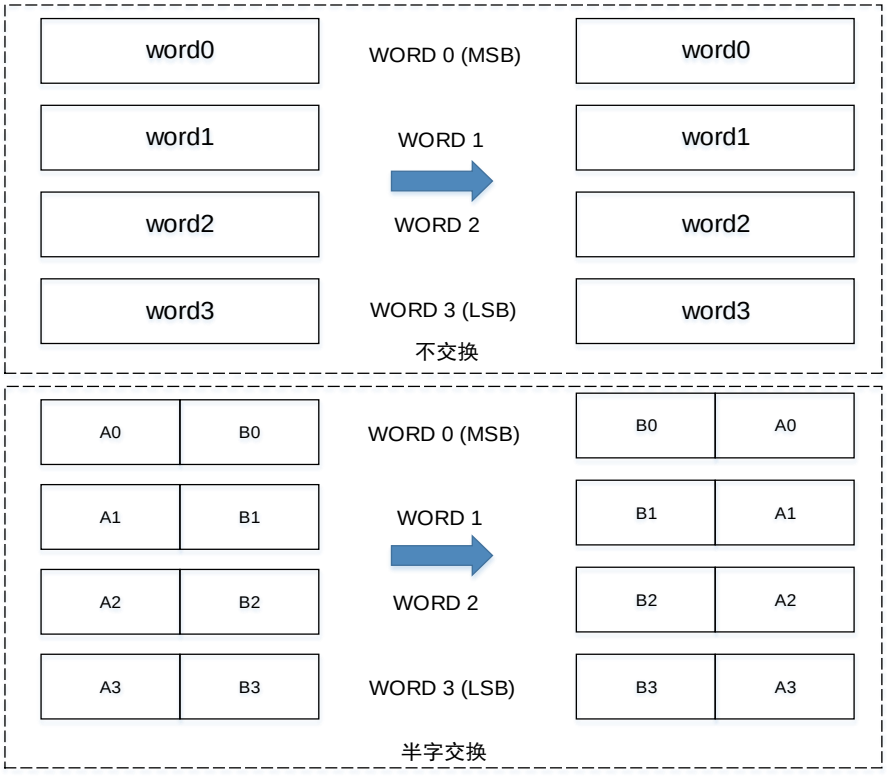
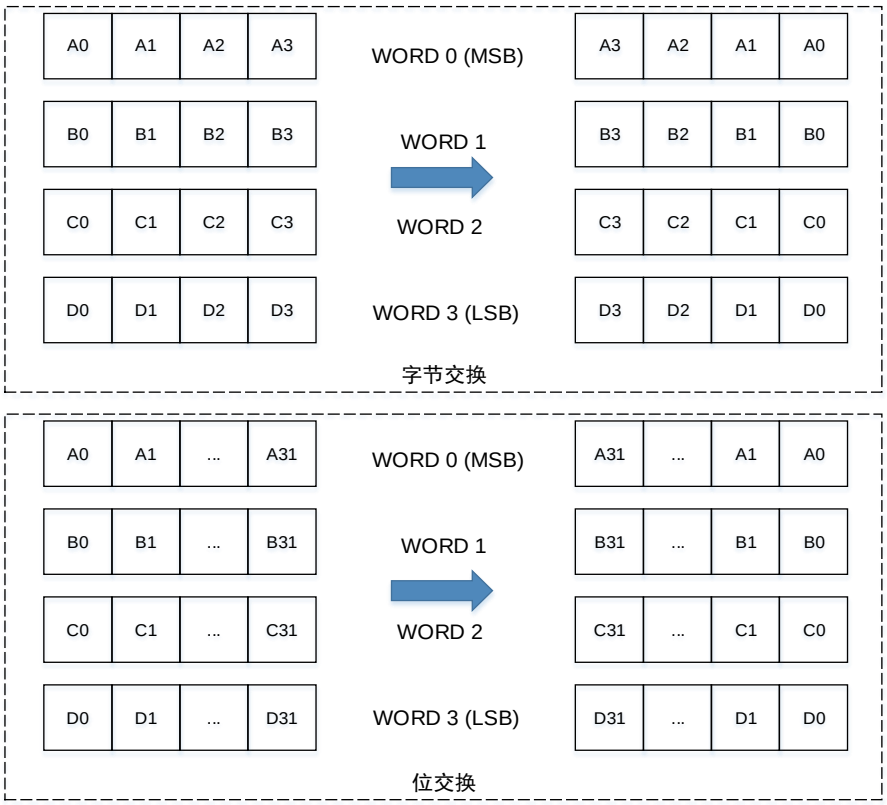


图 13-2. DATATM 字节交换/位交换



13.3.2. 初始化向量

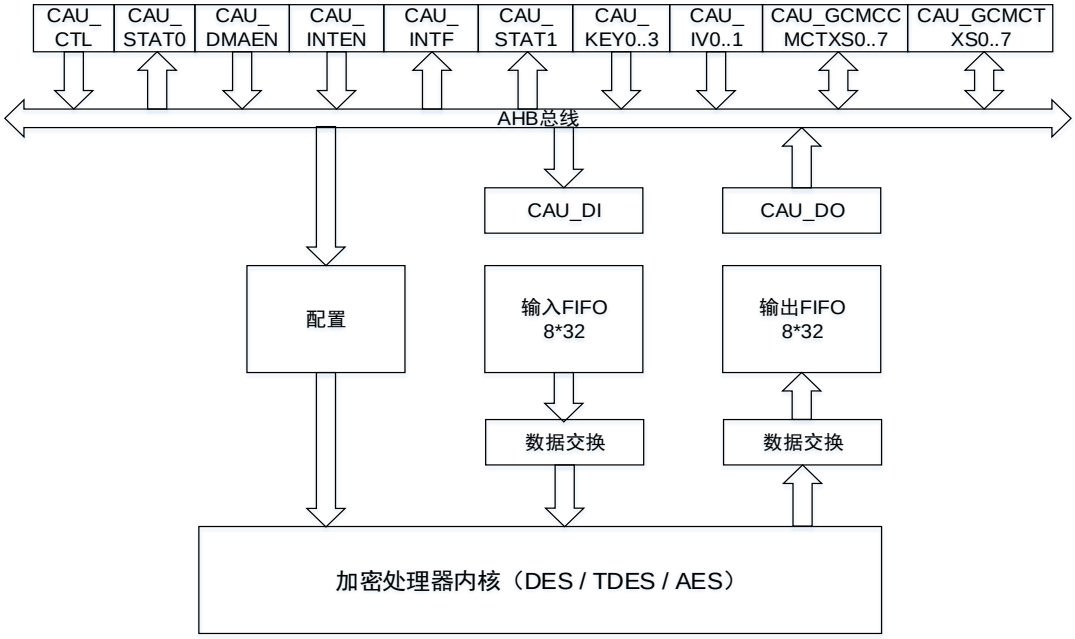
初始化向量用于在 CBC、CTR、GCM、GMAC、CCM、CFB 和 OFB 模式下与数据块进行异或。初始化向量与明文或密码数据无关，而且它们不受 DATAM 值的影响。注意初始化向量寄存器 CAU_IV0..1（H / L）只有在 BUSY 位（CAU_STAT0 寄存器位 4）为 0 时才能被修改，否则写操作都是无效的。

13.4. 加密处理器流程

加密处理器关于 DES 和 AES 加密处理的实现具体请参考章节 ***DES / TDES 加密处理流程***和 ***AES 加密处理流程***。

图 13-3. CAU 框图为加密处理器的模块框图。

图 13-3. CAU 框图



13.4.1. DES / TDES 加密处理流程

DES / 三重 DES 加密处理器由 DES 算法（DEA），密钥（DES 算法使用 1 个密钥，TDES 算法使用 3 个密钥），以及在 CBC 模式下使用的初始化向量组成。

DES / TDES 密钥

DES 模式密钥为[KEY1]，TDES 模式密钥为[KEY3 KEY2 KEY1]。当配置使用 TDES 算法，支持以下三种密钥选项：

1. 三个相同密钥

三个密钥 KEY3、KEY2 和 KEY1 是相同的，即 KEY3=KEY2=KEY1。该选项详见 FIPS PUB 46-3-1999（以及 ANSI X9.52-1998）。这种模式下实际上与 DES 是等同的。

2. 两个独立密钥

这个选项中，KEY2 与 KEY1 不同，KEY3 与 KEY1 相同，即 KEY1 与 KEY2 独立，而 KEY3=KEY1。该选项详见 FIPS PUB 46-3-1999（以及 ANSI X9.52-1998）。

3. 三个独立密钥

这个选项中，KEY1，KEY2 与 KEY3 都是独立的。详见 FIPS PUB 46-3 -1999（以及 ANSI X9.52-1998）。

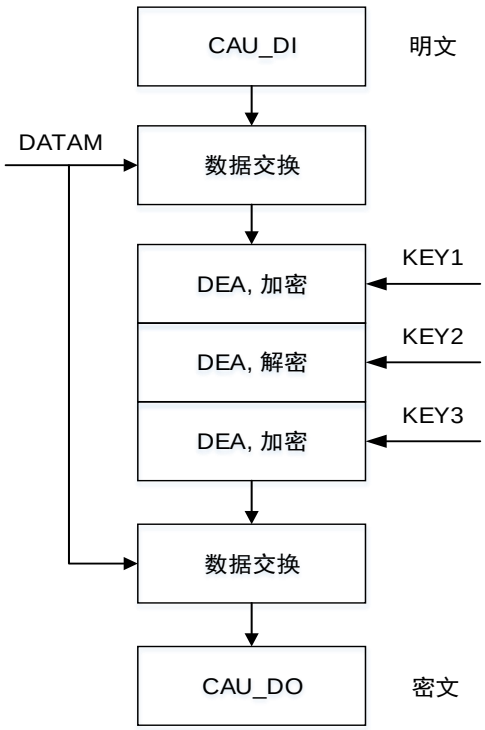
FIPS PUB 46-3（以及 ANSI X9.52-1998）对 DES/TDES 中密钥的使用进行了详尽的解释，在本

手册中不进行赘述。

DES / TDES 电子密码本（ECB）加密

64 位输入明文数据首先经过根据数据类型值进行数据交换后作为输入数据块。若配置使用的是 TDES 算法，则输入数据块通过 DEA 使用 KEY1 进行加密处理。处理结果输出直接反馈到 DEA，使用 KEY2 进行解密处理。之后处理结果输出直接反馈到到最后的 DEA，使用 KEY3 进行加密处理。上述的处理过程的输出需要再次根据数据类型值进行数据交换，生成一个 64 位密文输出数据块。若配置使用的是 DES 算法，在通过 DEA 使用 KEY1 进行加密处理后的结果直接根据数据类型值进行数据交换，生成一个 64 位密文输出数据块。DES / TDES 电子密码本加密流程图见*图 13-4. DES / TDES ECB 加密*。

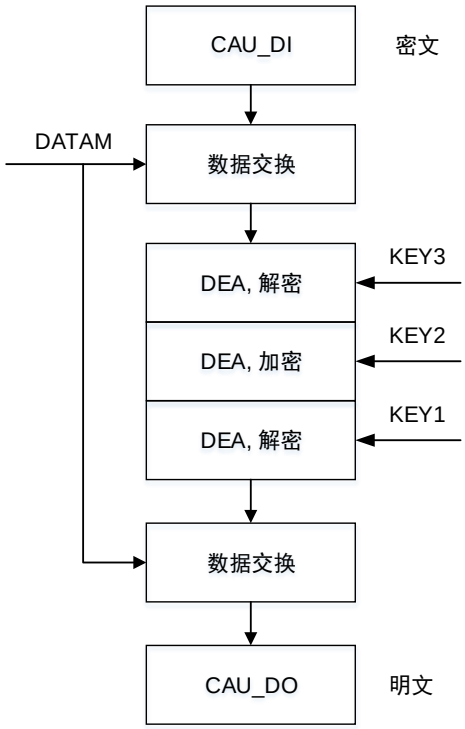
图 13-4. DES / TDES ECB 加密



DES / TDES 电子密码本（ECB）解密

根据数据类型进行数据交换后，首先得到 64 位的输入密文。若配置使用的是 TDES 算法，将在 DEA 中读取输入数据块并使用 KEY3 进行解密处理。处理结果输出直接反馈到下一个 DEA，使用 KEY2 进行加密处理。之后处理结果输出直接反馈到到最后的 DEA，使用 KEY1 进行解密处理。上述的处理过程的输出需要再次根据数据类型进行数据交换，生成一个 64 位明文输出数据块。若配置使用的是 DES 算法，在通过 DEA 使用 KEY1 进行解密处理后的结果直接根据数据类型值进行数据交换，生成一个 64 位明文输出数据块。DES / TDES 电子密码本解密流程图见*图 13-5. DES / TDES ECB 解密*。

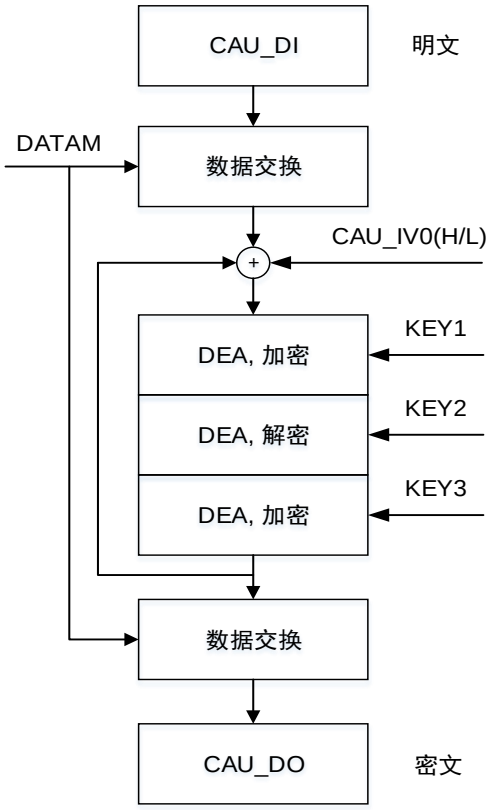
图 13-5. DES / TDES ECB 解密



DES / TDES 加密分组链接（CBC）加密

CBC 模式下 DEA 块的输入包括两部分：根据数据类型进行数据交换后的输入明文数据块，以及初始化向量。若配置使用的是 TDES 算法，第一个数据交换后的输入明文数据块与 64 位初始化向量 CAU_IV0..1 进行异或运算，结果在 DEA 中读取并使用 KEY1 进行加密处理。处理结果输出直接反馈到下一个 DEA，使用 KEY2 进行解密处理。之后处理结果输出直接反馈到到最后的 DEA，使用 KEY3 进行加密处理。上述的处理过程的输出作为下一个初始化向量，并与下一个明文数据块进行异或运算，进行下一轮的加密处理。重复上述的操作，直到完成最后一个明文数据块的加密处理。注意如果明文消息中的数据块数不是整数，则应按指定的方式对最后的不完整数据块进行加密处理。最后，对上述处理结果的输出需要再次根据数据类型进行数据交换，生成密文输出数据块。若配置使用的是 DES 算法，则在上述的步骤操作中忽略第二次和第三次 DEA 的运算处理。DES / TDES 加密分组链接加密流程图见*图 13-6. DES / TDES CBC 加密*。

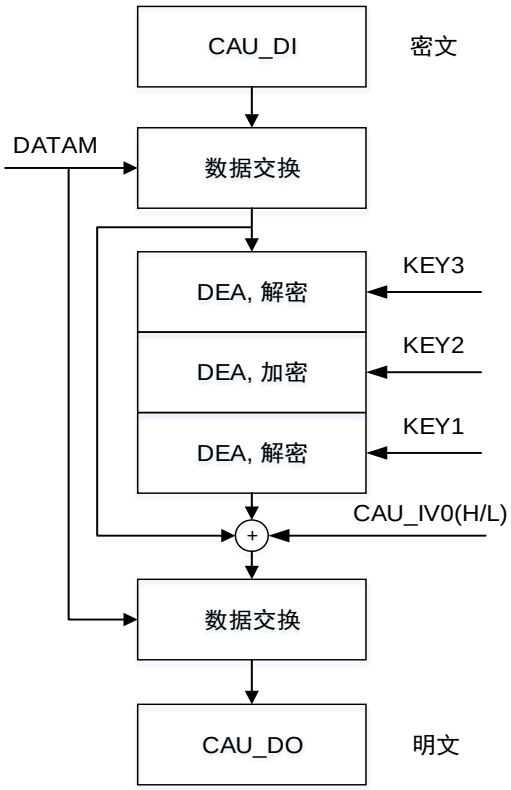
图 13-6. DES / TDES CBC 加密



DES / TDES 密码块链接（CBC）解密

使用 DES / TDES CBC 模式解密，若配置使用的是 TDES 算法，第一个数据交换后的输入密文数据块，通过 DEA 读取并使用 KEY3 进行解密处理。处理结果输出直接反馈到下一个 DEA，使用 KEY2 进行加密处理。之后处理结果输出直接反馈到到最后的 DEA，使用 KEY1 进行解密处理。上述的处理过程的输出再与 64 位初始化向量 CAU_IV0..1 进行异或运算。之后，第一个输入密文数据块作为下一个初始化向量，并与后续的 DEA 解密处理后的输出结果进行异或运算。重复上述的操作，直到完成最后一个密文数据块的解密处理。注意如果密文消息中的数据块数不是整数，则应按指定的方式对最后的不完整数据块进行解密处理。最后，对上述处理结果的输出需要再次根据数据类型进行数据交换，生成明文输出数据块。若配置使用的是 DES 算法，则在上述的步骤操作中忽略第二次和第三次 DEA 的运算处理。DES/TDES 加密分组链接解密流程图见*图 13-7. DES / TDES CBC 解密*。

图 13-7. DES / TDES CBC 解密



13.4.2. AES 加密处理流程

AES 加密处理器由 AES 算法（AEA），多个密钥，以及初始化向量或随机数三部分组成。

AES 支持三种长度的密钥：128、192 和 256 位密钥，根据操作模式的不同使用不同数目的初始化向量或随机数。

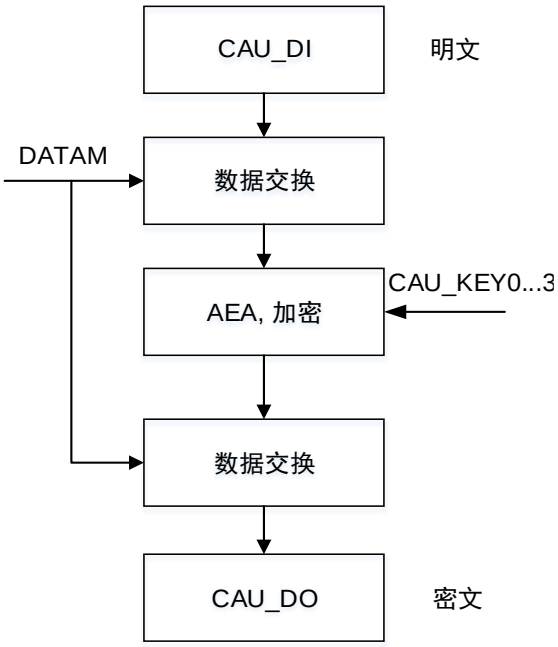
使用 128 位密钥时 AES 密钥为[KEY3 KEY2]，使用 192 位密钥时 AES 密钥为[KEY3 KEY2 KEY1]，使用 256 位密钥时 AES 密钥为[KEY3 KEY2 KEY1 KEY0]。

FIPS PUB 197（2001 年 11 月 26 日）中对 AES 中使用的密钥进行了详细的解释，本手册不再进行赘述。

AES 电子密码本（ECB）加密

根据数据类型进行数据交换后，首先得到 128 位输入明文数据块。输入数据块通过 AEA 使用 128 位，或 192 位，或 256 位密钥进行加密处理。处理结果再根据数据类型进行数据交换，生成一个 128 位密文输出数据块，并存储在输出 FIFO 中。AES 电子密码本加密流程图见*图 13-8. AES ECB 加密*。

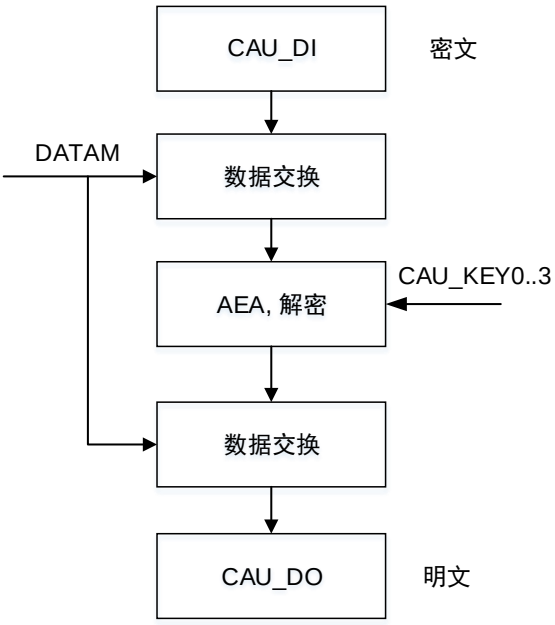
图 13-8. AES ECB 加密



AES 电子密码本（ECB）解密

首先需要准备密钥，以用于解密，密钥准备过程的输入密钥与加密处理中的密钥相同。从上述操作中获得的最后一个密钥将作为解密处理用的第一个密钥。密钥准备完成后，首先根据数据类型进行数据交换得到 128 位输入密文数据块。输入数据块在 AEA 中读取并使用上面准备的密钥进行解密处理。处理结果输出再根据数据类型值进行数据交换，生成一个 128 位明文输出数据块。AES 电子密码本解密流程图见*图 13-9. AES ECB 解密*。

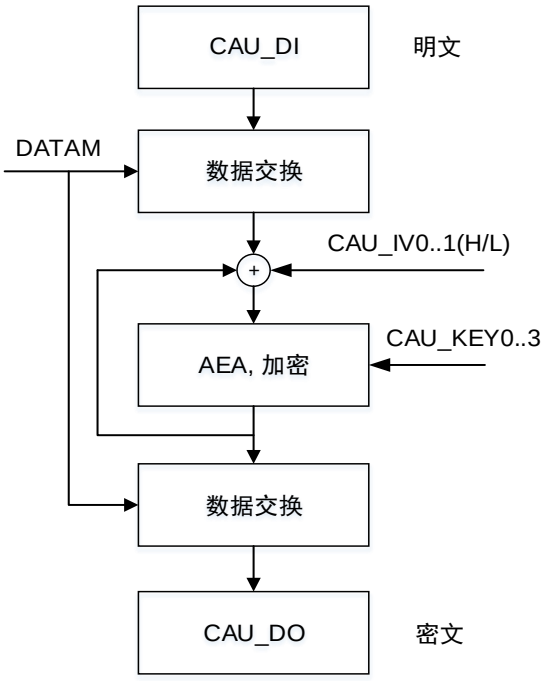
图 13-9. AES ECB 解密



AES 加密分组链接（CBC）加密

CBC 模式下 AEA 块的输入包括两部分：根据数据类型进行数据交换后的输入明文数据块，以及初始化向量。数据交换后的输入明文数据块与 128 位初始化向量 CAU_IV0..1 进行异或运算，结果再通过 AEA 使用 128 位，或 192 位，或 256 位密钥进行加密处理。处理结果作为下一个初始化向量，并与下一个输入明文数据块进行异或运算，进行下一轮加密处理。重复上述的操作，直到完成最后一个明文数据块的加密处理。注意如果明文消息中的数据块数不是整数，则应按指定的方式对最后的不完整数据块进行加密处理。最后，对上述处理结果的输出需要再次根据数据类型值进行数据交换，生成密文输出数据块。AES 加密分组链接加密流程图见*图 13-10. AES CBC 加密*。

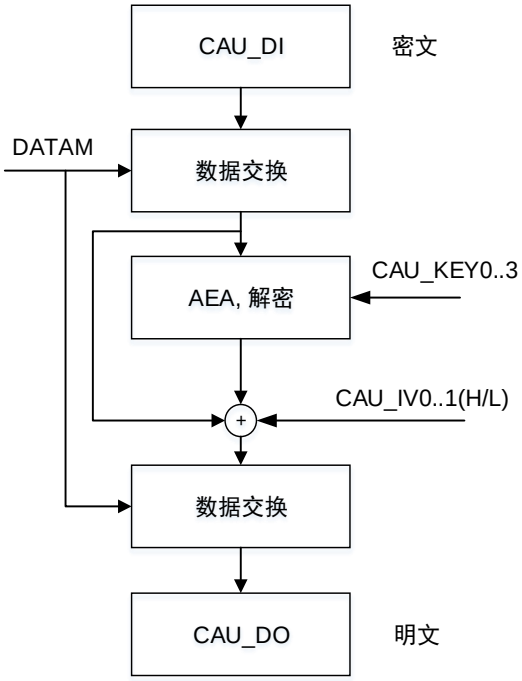
图 13-10. AES CBC 加密



AES 加密分组链接（CBC）解密

与 AES 电子密码本（ECB）模式解密类似，首先需要准备密钥以用于解密，密钥准备过程的输入密钥与加密处理中的密钥相同。从上述操作中获得的最后一个密钥将作为解密处理用的第一个密钥。密钥准备完成后，首先根据数据类型进行数据交换，得到 128 位输入密文数据块，输入数据块在 AEA 中读取并使用准备的密钥进行解密处理。之后，第一个输入密文数据块作为下一个初始化向量，并与下一个 AEA 解密处理结果进行异或运算（第一次的初始化向量为输入 CAU_IV0..1 的初值）。重复上述的操作，直到完成最后一个密文数据块的解密处理。注意如果密文消息中的数据块数不是整数，则应按指定的方式对最后的不完整数据块进行解密处理。最后，对上述处理结果的输出需要再次根据数据类型进行数据交换，生成明文输出数据块。AES 加密分组链接解密流程图见*图 13-11. AES CBC 解密*。

图 13-11. AES CBC 解密

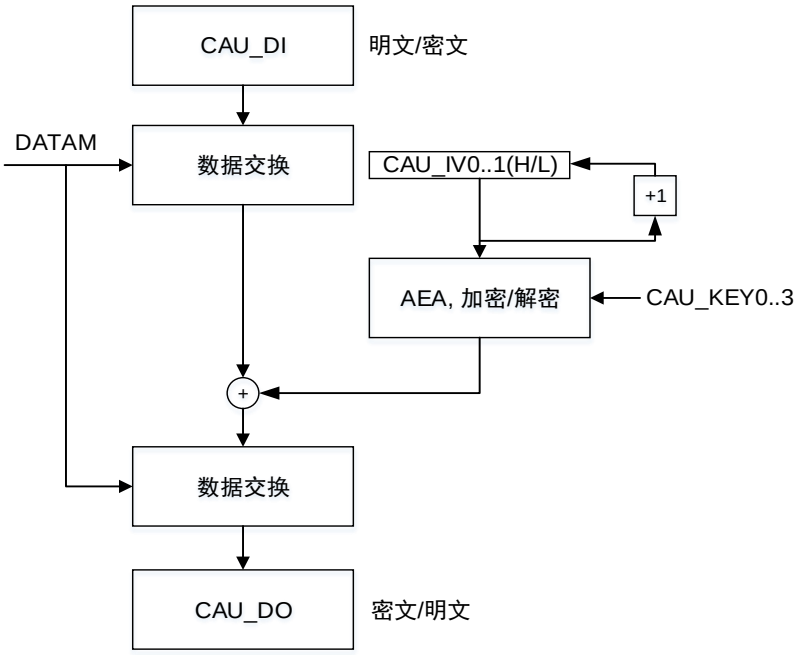


AES 计数器（CTR）模式

在计数器模式下，随机数与计数器的组合会作为 AEA 计算单元的输入来进行运算，运算结果会与输入的明文或密文进行异或，来求得最终加密或者解密的结果。由于加密和解密处理的计数器值是由相同的初始值进行递增的，因此加密和解密处理用的密钥序列是相同的。解密处理的操作与加密操作的流程完全相同。128 位初始化向量的低 32 位表示为计数器值，这意味着其余 96 位在操作过程中保持不变，并且计数器的初始值应当设置为 1。随机数是一个 32 位一次性值，应当更新到每个通信块。64 位的初始化向量应确保每个给定值只用于一个给定密钥。计数器块框图结构见*图 13-12. 计数器块结构*，AES 计数器加密/解密流程图见*图 13-13. AES CTR 加密/解密*。

图 13-12. 计数器块结构

图 13-13. AES CTR 加密/解密



AES-GCM 模式

AES 伽罗瓦/计数器模式(GCM)可用于加密或验证消息，来获得密文和标签。该算法基于 AES 计数器模式，保证了机密性。利用固定的有限域乘法运算来生成标签。

在该模式中，执行加密/解密需要四个步骤：

1. GCM准备阶段

内部计算和保存哈希密钥以在后续使用。

(a) 将CAUEN清零，禁能CAU；
(b) 配置ALGM[3:0]位域为‘1000’；
(c) 配置GCM_CCMPH[1:0]位域为‘00’；
(d) 配置密钥寄存器CAU_KEY0..3(H/L)和初始化向量寄存器CAU_IV0..1(H/L)；
(e) 置位CAUEN位，使能CAU；
(f) 等待CAUEN位被硬件清零，然后再置位CAUEN，使能CAU，进行下个步骤。

2. GCM AAD（附加身份验证数据）阶段

AAD 阶段必须在 GCM 初始化阶段之后进行，并在加密解密阶段之前。在这个阶段，数据仅进行了验证，而没有被保密。

(g) 配置GCM_CCMPH[1:0]位域为‘01’；
(h) 将AAD数据写入CAU_DI寄存器，并使用CAU_STAT0寄存器的INF和IEM标志来判断输入FIFO是否能接收数据。AAD大小必须为128位的倍数。也可使用DMA来写入AAD数据。

(i) 重复步骤(h)直到所有AAD数据都写入，并等待CAU_STAT0寄存器的BUSY位清零。

3. GCM加密解密阶段

加密解密阶段必须在 GCM AAD 阶段之后进行。在这个阶段，对消息进行了验证，并加密或解密。

(j) 配置GCM_CCMPH[1:0]位为‘10’；
(k) 配置CAUDIR位来选择算法方向；
(l) 将有效负载消息写入CAU_DI寄存器，并使用CAU_STAT0寄存器的INF和IEM标志来判断输入FIFO是否能接收数据。使用CAU_STAT0寄存器的ONE和OFU标志判断输出FIFO是否为空，如果不为空，就读取CAU_DO寄存器。也可使用DMA来写入有效负载消息。
(m) 重复步骤(l)直到所有的有效负载块都完成计算。

4. GCM标签阶段

在这个阶段，将生成最后的验证标签。

(n) 配置GCM_CCMPH[1:0]位为‘11’；
(o) 将最后的数据块（由64位AAD大小和64位有效负载消息大小组成）写入CAU_DI寄存器；
(p) 在完成写4次CAU_DI寄存器之后，等待CAU_STAT0寄存器的ONE标志置位，然后读取CAU_DO寄存器4次，这个输出数据就是最后生成的验证标签；
(q) 禁能CAU。

注意：解密时，必须在开始阶段时准备好密钥。

AES-GMAC 模式

AES 伽罗瓦消息验证码（GMAC）模式支持提供对消息的完整性验证。这个模式处理流程可视为 AES-GCM 模式流程除去加密解密阶段。

AES-CCM 模式

AES 结合了类似于 AES-GCM 的密码机模式，支持消息的保密，以及完整性验证。AES-CCM 模式基于 AES-CTR 模式来确保了消息的保密性，使用 AES-CBC 模式来生成 128 位标签。

CCM 标准（RFC 3610 Counter with CBC-MAC (CCM) 标准，2003 年 9 月发布）为首个验证块（在该标准中称为 B0）定义了特定的编码规则，具体来说，首个块包括标志、随机数以及以字节计的有效负载大小。CCM 标准为加密/解密指定了另外的格式，称为 A 或者计数器。计数器在有效负载阶段递增计数，在生成标签阶段计数器低 32 位有效位初始化为’1’（在 CCM 标准中称为 A0 数据包）。

注意：B0 数据包的格式化操作需要在软件中处理完成。

在该模式中，执行加密/解密需要四个步骤：

1. CCM准备阶段

准备阶段，将 B0 数据包（首个块）写入 CAU_DI 寄存器。在这个阶段，CAU_DO 寄存器不包含任何输出数据。

(a) 清零CAUEN位，禁能CAU；
(b) 配置ALGM[3:0]位域为‘1001’；
(c) 配置GCM_CCMPH[1:0]位域为‘00’；
(d) 配置密钥寄存器CAU_KEY0..3(H/L)和初始化向量寄存器CAU_IV0..1(H/L)；
(e) 置位CAUEN位，使能CAU；
(f) 将B0数据包写入CAU_DI寄存器；
(g) 等待CAUEN位被硬件清零，然后再置位CAUEN，使能CAU，进行下个步骤。

2. CCM AAD（附件身份验证数据）阶段

AAD 阶段必须在 CCM 准备阶段之后进行，并在加密解密阶段之前。在这个阶段，CAU_DO 寄存器不包含任何输出数据。

如果没有附加的验证数据，可以跳过这个阶段。

(h) 配置GCM_CCMPH[1:0]位域为‘01’；
(i) 将AAD数据写入到CAU_DI寄存器，并使用CAU_STAT0寄存器的INF和IEM标志来判断输入FIFO是否能接收数据。AAD大小必须为128位的倍数。也可以使用DMA来写入AAD数据。
(j) 重复步骤(i)直到所有的AAD数据都写入，并等待CAU_STAT0寄存器的BUSY位清零。

3. CCM加密解密阶段

加密解密阶段必须在 CCM AAD 阶段之后进行。在这个阶段，对消息进行了验证，并加密或解密。

与 GCM 类似，CCM 链接模式可用于仅由经过验证的原文数据(即只有 AAD，没有有效负载)组成的消息。需要注意的是，这种使用 CCM 的方式不称为 CMAC(它与 GCM/GMAC 不同)。

(k) 配置GCM_CCMPH[1:0]位为‘10’；
(l) 配置CAUDIR位来选择算法方向；
(m) 将有效负载消息写入CAU_DI寄存器，并使用CAU_STAT0寄存器的INF和IEM标志来判断输入FIFO是否能接收数据。使用CAU_STAT0寄存器的ONE和OFU标志判断输出FIFO是否为空，如果不为空，就读取CAU_DO寄存器。也可使用DMA来写入有效负载消息；
(n) 重复步骤(m)直到所有的有效负载块都完成计算。

4. CCM标签阶段

在这个阶段，将生成最后的验证标签。

(o) 配置GCM_CCMPH[1:0]位为‘11’；
(p) 将128位A0数据包写入到CAU_DI寄存器，分为4次的写操作；
(q) 等待CAU_STAT0寄存器的ONE标志置位，然后读取CAU_DO寄存器4次，这个输出数据就是最后生成的验证标签；
(r) 禁能CAU。

AES-CFB 模式

密码反馈(CFB)模式是保密模式，其特征在于将连续密文段反馈到前向密码的输入块中，以生成与明文异或的输出块，从而产生密文，反之解密过程与加密的过程类似。

AES-OFB 模式

输出反馈(OFB)模式是保密模式，其特征在于在 IV 上对前向密码进行迭代，以生成与明文异或以产生密文的输出块序列，反之解密过程与加密的过程类似。

13.5. 操作模式

加密

1. 将CAU_CTL寄存器的CAUEN位清零，以禁用CAU；
2. 将PMU_CTL1寄存器中的CORE1WAKE置位以使能CAU的电源域，再打开CAU的外设时钟；
3. 若选择了AES算法，则对CAU_CTL寄存器的KEYM位进行选择设置，配置密钥的长度；
4. 根据算法配置CAU_KEY0..3(H/L)寄存器；
5. 设置CAU_CTL寄存器的DATAM位，配置数据交换类型；
6. 设置CAU_CTL寄存器的ALGM[3:0]位，配置算法（DES/TDES/AES）和模式（ECB/CBC/CTR/GCM/GMAC/CCM/CFB/OFB）；
7. 设置CAU_CTL寄存器的CAUDIR位为0，配置为加密操作；
8. 设置CAU_IV0..1(H/L)寄存器，配置初始化向量；
9. 在CAUEN位为0时，设置CAU_CTL寄存器的FFLUSH位，配置刷新输入FIFO和输出FIFO；
10. 设置CAU_CTL寄存器的CAUEN位为1，使能CAU；
11. 当CAU_STAT0寄存器的INF位为1时，向CAU_DI寄存器写数据块。数据可以通过DMA传输或者CPU中断传输，也可不通过两者进行传输；
12. 等待CAU_STAT0寄存器的ONE位为1时，读CAU_DO寄存器。输出数据可以通过DMA传输或者CPU中断传输，也可不通过两者进行传输；
13. 重复步骤10和步骤11，直到所有的数据块都完成加密。

解密

1. 将CAU_CTL寄存器的CAUEN位清零，以禁用CAU；
2. 将PMU_CTL1寄存器中的CORE1WAKE置位以使能CAU的电源域，再打开CAU的外设时钟；
3. 若选择了AES算法，则对CAU_CTL寄存器的KEYM位进行选择设置，配置密钥的长度；
4. 根据算法配置CAU_KEY0..3(H/L)寄存器；
5. 设置CAU_CTL寄存器的DATAM位，配置数据交换类型；
6. 设置CAU_CTL寄存器的ALGM[3:0]位为“0111”，配置准备密钥用于解密；
7. 设置CAU_CTL寄存器的CAUEN位为1，使能CAU；
8. 等待BUSY位和CAUEN位为0，确保解密用的密钥已准备好；

9. 设置CAU_CTL寄存器的ALGM[3:0]位，配置算法（DES/TDES/AES）和模式（ECB/CBC/CTR/GCM/GMAC/CCM/CFB/OFB）；
10. 设置CAU_CTL寄存器的CAUDIR位为1，配置为解密操作；
11. 设置CAU_IV0..1(H/L)寄存器，配置初始化向量；
12. 在CAUEN位为0时，设置CAU_CTL寄存器的FFLUSH位，配置刷新输入FIFO和输出FIFO；
13. 设置CAU_CTL寄存器的CAUEN位为1，使能CAU；
14. 当CAU_STAT0寄存器的INF位为1时，向CAU_DI寄存器写数据块。数据可以通过DMA传输或者CPU中断传输，也可不通过两者进行传输；
15. 等待CAU_STAT0寄存器的ONE位为1时，读CAU_DO寄存器。输出数据可以通过DMA传输或者CPU中断传输，也可不通过两者进行传输；
16. 重复步骤13和步骤14，直到所有的数据块都完成解密。

数据填充

对于 GCM 加密和 CCM 解密，CAU 模块支持非 128 比特整数倍的数据块处理。当最后一个数据块不满 128 比特时，使用‘0’对其剩余位进行填充，然后在 CAU_CTL 寄存器的 NBPILB 位域中配置用于填充的字节数，AES 会自动去除相应填充数量的填充块后进行加密。需要注意的是，只有在倒数第二个数据块加密完成后，才可以对 NBPILB 位域进行配置。

13.6. CAU DMA 接口

DMA 可用于 CAU 模块的数据块传输。DMA 的传输操作由 CAU_DMAEN 寄存器来控制。DMAIEN 位用于输入数据的 DMA 请求传输使能，由 DMA 将一个字数据写入 CAU_DI 寄存器。DMAOEN 位用于输出数据的 DMA 请求传输使能，获得 CAU 输出的一个字。

DMA 输出数据的传输请求优先级高于输入数据的传输请求，因此输出 FIFO 空的事件可能会早于输入 FIFO 满的事件。

13.7. CAU 中断

CAU 有两个中断状态寄存器，CAU_STAT1 和 CAU_INTF 寄存器。CAU 中的中断用于指示输入和输出 FIFO 的状态。

可以通过配置 CAU_INTEN 寄存器来使能或禁用输入或输出 FIFO 中断。将寄存器中相应位置 1 可以使能相应中断。

输入 FIFO 中断

当输入 FIFO 中的数据少于 4 个字时产生输入 FIFO 中断，ISTA 位置位。此时如果 IINTEN 位为 1，使能了输入 FIFO 中断，则 IINTF 位将置位。注意当 CAUEN 位为 0 时，ISTA 位和 IINTF 位将保持为 0。

输出 FIFO 中断

当输出 FIFO 中存在一个或多个字数据时产生输出 FIFO 中断，OSTA 位置位。此时如果 OINTEN 位为 1 从而使能了输出 FIFO 中断，则 OINTF 位将置位。注意与输入 FIFO 中断不同的是，当 CAUEN 位为 0 时，不会影响到 OSTA 位与 OINTF 位的状态。

13.8. CAU 挂起模式

当 CAU 中待处理的新的数据块优先级高于正在处理的数据块，则正在处理的数据块可能被挂起。按照下列的步骤来完成被挂起数据块的加密/解密处理。

当使用 DMA 进行数据传输：

1. 停止当前输入数据传输。将CAU_DMAEN寄存器的DMAIEN位清零；
2. 若为DES或AES算法，则需等待直到输入和输出FIFO均为空，如果检查到输入FIFO不为空即IEM位为0，则写入一个字的数据，再检查IEM位，直到IEM位为1，则停止写入数据，再等待BUSY位为0，以确保下一个数据块不会被上一个数据块影响。若为TDES算法，则与AES算法相似，但不需要等待输入FIFO为空；
3. 将CAU_DMAEN寄存器中的DMAOEN位清零，停止输出数据传输。并将CAU_CTL寄存器中的CAUEN位清零，禁用CAU；
4. 保存当前配置，包括密钥长度，数据类型，算法模式，算法方向，GCM CCM阶段，以及密钥值。若为CBC / CTR / GCM / GMAC / CCM / CFB / OFB模式，则还需要保存初始化向量。若为GCM / GMAC / CCM模式，则还需要保存上下文交换寄存器CAU_GCMCCMCTXSx (x=0..7)和CAU_GCMCTXSx (x=0..7)；
5. 配置并处理新数据块；
6. 恢复之前的处理环境。将CAU重新用存储的参数进行配置，并准备好密钥和初始化向量，还需恢复CAU_GCMCCMCTXSx (x=0..7)和CAU_GCMCTXSx (x=0..7)寄存器。再将CAU_CTL寄存器的CAUEN位置位以使能CAU。

当使用 CPU 来传输数据到 CAU_DI 和 CAU_DO：

1. 当使用CPU来进行数据传输，则需要等待第四次读CAU_DO寄存器，并在写CAU_DI之前，以确保一个数据块处理结束的时候再挂起消息处理；
2. 将CAU_CTL寄存器的CAUEN位清零，禁用CAU；
3. 保存当前配置，包括密钥长度，数据类型，算法模式，算法方向，GCM CCM阶段，以及密钥值。若为CBC/CTR/GCM/GMAC/CCM/CFB/OFB模式，则还需要保存初始化向量。若为GCM/GMAC/CCM模式，则还需要保存上下文交换寄存器CAU_GCMCCMCTXSx (x=0..7)和CAU_GCMCTXSx (x=0..7)；
4. 配置并处理新数据块；
5. 恢复之前的处理环境。将CAU重新用存储的参数进行配置，并准备好密钥和初始化向量，还需恢复CAU_GCMCCMCTXSx (x=0..7)和CAU_GCMCTXSx (x=0..7)寄存器。再将CAU_CTL

寄存器的CAUEN位置位以使能CAU。

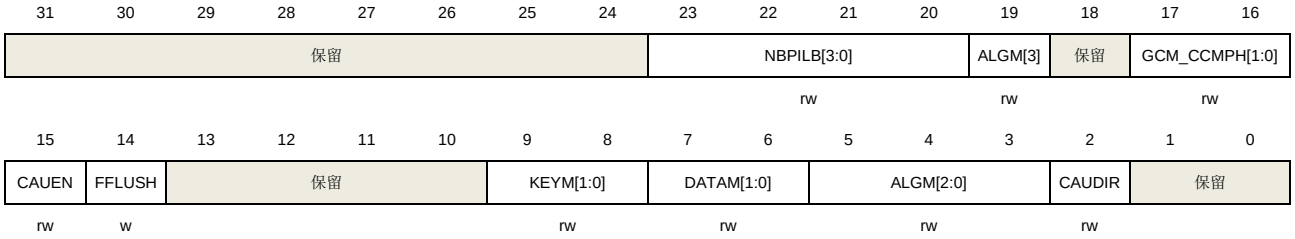
13.9. CAU 寄存器

CAU 基地址：0x4802 1000

13.9.1. 控制寄存器（CAU_CTL）

偏移地址：0x00
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:24	保留	必须保持复位值。
23:20	NBPILB[3:0]	最后一个非 128 比特整数倍数据块的填充字节数 0000：所有数据有效（无填充） 0001：一个填充字节 … 1111：15 个填充字节
19	ALGM[3]	加密/解密算法模式位 3
18	保留	必须保持复位值。
17:16	GCM_CCMPH[1:0]	GCM CCM 阶段 00：准备阶段 01：AAD 阶段 10：加密解密阶段 11：标签阶段
15	CAUEN	加密处理器使能 0：加密处理器禁用 1：加密处理器使能 **注意**：当准备密钥（ALGM=0111b）完成后，CAUEN 位将硬件自动清零。
14	FFLUSH	FIFO 刷新 0：不产生影响

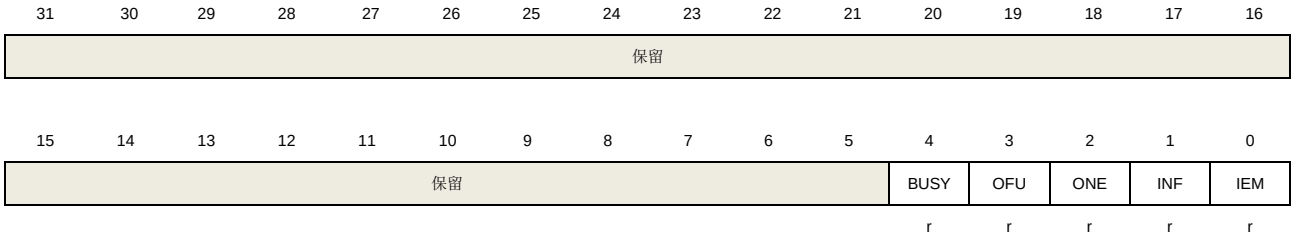
		1：当 CAUEN=1 时，刷新输入和输出 FIFO 读取该位时，始终返回 0
13:10	保留	必须保持复位值
9:8	KEYM[1:0]	AES 密钥长度配置，必须在 BUSY=0 时才可配置 00：128 位密钥长度 01：192 位密钥长度 10：256 位密钥长度 11：保留
7:6	DATAM[1:0]	数据交换模式配置，必须在 BUSY=0 时才可配置 00：不交换 01：半字交换 10：字节交换 11：位交换
5:3	ALGM[2:0]	加密/解密算法模式位 0 到位 2 该位域和位 19 必须在 BUSY=0 时才可配置。 0000：TDES-ECB（三重 DES 电子密码本），使用 CAU_KEY1, 2, 3. 不使用初始化向量（CAU_IV0..1） 0001：TDES-CBC（三重 DES 加密分组链接），使用 CAU_KEY1, 2, 3. 使用初始化向量（CAU_IV0）与数据块进行异或 0010：DES-ECB（DES 电子密码本），仅使用 CAU_KEY1 不使用初始化向量（CAU_IV0..1） 0011：DES-CBC（DES 加密分组链接），仅使用 CAU_KEY1 使用初始化向量（CAU_IV0）与数据块进行异或 0100：AES-ECB（AES 电子密码本），使用 CAU_KEY0, 1, 2, 3. 不使用初始化向量（CAU_IV0..1） 0101：AES-CBC（AES 加密分组链接），使用 CAU_KEY0, 1, 2, 3. 使用初始化向量（CAU_IV0..1）与数据块进行异或 0110：AES-CTR（AES 计数器模式），使用 CAU_KEY0, 1, 2, 3. 使用初始化向量（CAU_IV0..1）与数据块进行异或 该模式下，加密与解密处理相同，忽略 CAUDIR 位 0111：AES 解密密钥准备模式。输入密钥必须与加密处理中用的密钥相同。BUSY 位将保持置位直到完成密钥的准备，随后 CAUEN 位会清零。 1000：AES-GCM（伽罗瓦/计数器模式），该模式算法同样适用于GMAC算法。 1001：AES-CCM（加密分组链接-消息验证码模式）。 1010：AES-CFB（密码反馈模式） 1011：AES-OFB（输出反馈模式）
2	CAUDIR	CAU 算法方向，必须在 BUSY=0 时才可配置

		0：加密 1：解密
1:0	保留	必须保持复位值。

13.9.2. 状态寄存器 0（CAU_STAT0）

偏移地址：0x04
复位值：0x0000 0003

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:5	保留	必须保持复位值。
4	BUSY	忙碌标志位 0：CAU 内核空闲，这是由于 - CAUEN=0 从而 CAU 内核被禁用，或这处理已完成 - 正在等待输入数据或输出 FIFO 有足够的自由空间来处理数据块 1：CAU 内核忙碌，正在处理数据块或准备密钥
3	OFU	输出 FIFO 满 0：输出 FIFO 未满 1：输出 FIFO 满
2	ONE	输出 FIFO 非空 0：输出 FIFO 为空 1：输出 FIFO 非空
1	INF	输入 FIFO 未满 0：输入 FIFO 满 1：输入 FIFO 未满
0	IEM	输入 FIFO 空 0：输入 FIFO 非空 1：输入 FIFO 空

13.9.3. 数据输入寄存器（CAU_DI）

偏移地址：0x08
复位值：0x0000 0000

数据输入寄存器用于传输明文或密文数据块到输入 FIFO 中进行处理。首先写入 FIFO 的是数据块的 MSB，最后才是 LSB。当 CAUEN 位为 0，并且输入 FIFO 非空时，读取该寄存器时返回 FIFO 中的首个字。当 CAUEN 位为 1 时，读取该寄存器返回一个不确定的值。一旦执行了读操作，则必须要刷新 FIFO 以处理新数据块。

该寄存器只能按字（32 位）访问。

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

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

位/位域	名称	描述
31:0	DI[31:0]	数据输入 写这些位，数据会写入输入 FIFO。当 CAUEN 位为 0 时，读这些位将返回输入 FIFO 中的值，否则将返回不确定的值。

13.9.4. 数据输出寄存器（CAU_DO）

偏移地址：0x0C
复位值：0x0000 0000

数据输出寄存器是只读寄存器，用于接收来自输出 FIFO 的明文或密文处理结果。与 CAU_DI 类似，读取时首先读取的是数据块的 MSB，最后才是 LSB。

该寄存器只能按字（32 位）访问。

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

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

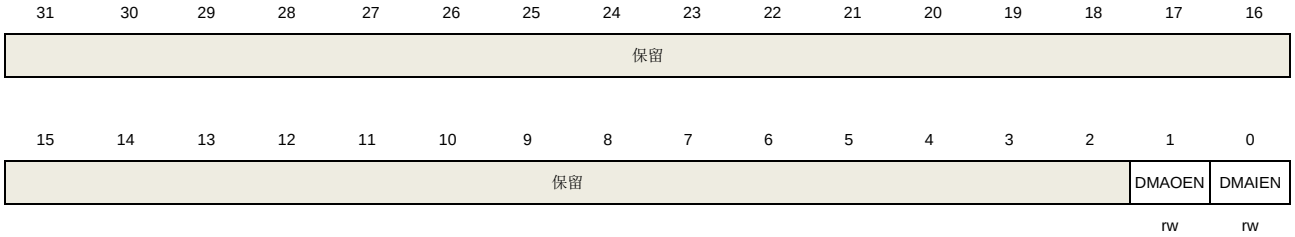
位/位域	名称	描述
31:0	DO[31:0]	数据输出

这些位为只读，读这些位将返回输出 FIFO 中的值。

13.9.5. DMA 使能寄存器（CAU_DMAEN）

偏移地址：0x10
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

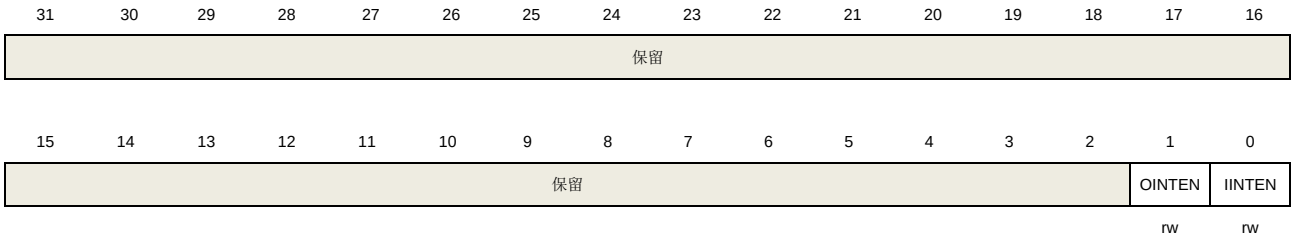


位/位域	名称	描述
31:2	保留	必须保持复位值。
1	DMAOEN	DMA 输出使能 0：禁用用于输出 FIFO 数据传输的 DMA 1：使能用于输出 FIFO 数据传输的 DMA
0	DMAIEN	DMA 输入使能 0：禁用用于输入 FIFO 数据传输的 DMA 1：使能用于输入 FIFO 数据传输的 DMA

13.9.6. 中断使能寄存器（CAU_INTEN）

偏移地址：0x14
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



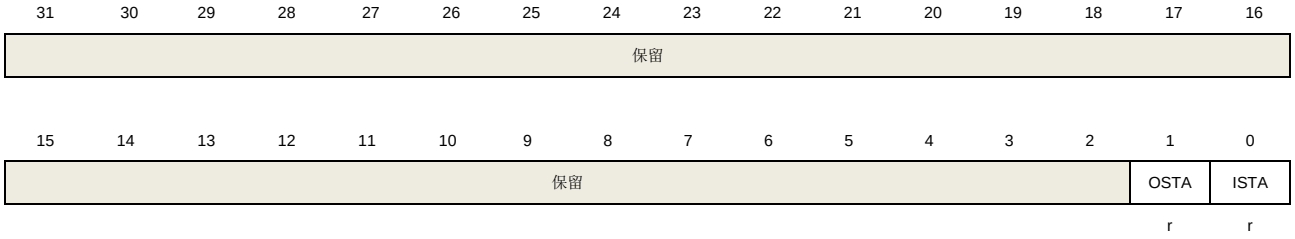
位/位域	名称	描述

31:2	保留	必须保持复位值。
1	OINTEN	输出 FIFO 中断使能 0：禁用输出 FIFO 中断 1：使能输出 FIFO 中断
0	IINTEN	输入 FIFO 中断使能 0：禁用输入 FIFO 中断 1：使能输入 FIFO 中断

13.9.7. 状态寄存器 1（CAU_STAT1）

偏移地址：0x18
复位值：0x0000 0001

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:2	保留	必须保持复位值。
1	OSTA	输出 FIFO 状态 0：输出 FIFO 状态未挂起 1：输出 FIFO 状态挂起
0	ISTA	输入 FIFO 状态 0：输入 FIFO 状态未挂起 1：输入 FIFO 状态挂起

13.9.8. 中断标志寄存器（CAU_INTF）

偏移地址：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														OINTF	IINTF
														r	r

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	OINTF	输出 FIFO 中断标志 0：输出 FIFO 中断状态未挂起 1：输出 FIFO 中断状态挂起
0	IINTF	输入 FIFO 中断标志 0：输入 FIFO 中断状态未挂起 1：当 CAUEN 位为 1 时输入 FIFO 中断状态挂起

13.9.9. 密钥寄存器（CAU_KEY0..3（H/L））

偏移地址：0x20~0x3C
复位值：0x0000 0000

该寄存器只能按字（32 位）访问，必须在 BUSY 位为 0 时写这些寄存器。

在 DES 模式下，仅使用 CAU_KEY1。

在 TDES 模式下，使用 CAU_KEY1，CAU_KEY2 和 CAU_KEY3。

在 AES-128 模式下，KEY2H[31:0]和 KEY2L[31:0]分别对应于 AES_KEY[0:63]的高 32 位与低 32 位，而 KEY3H[31:0]和 KEY3L[31:0]分别对应于 AES_KEY[64:127]的高 32 位与低 32 位。

在 AES-192 模式下，KEY1H[31:0]和 KEY1L[31:0]分别对应于 AES_KEY[0:63]的高 32 位与低 32 位，KEY2H[31:0]和 KEY2L[31:0]分别对应于 AES_KEY[64:127]的高 32 位与低 32 位，KEY3H[31:0]和 KEY3L[31:0]分别对应于 AES_KEY[128:191]的高 32 位与低 32 位。

在 AES-256 模式下，KEY0H[31:0]和 KEY0L[31:0]分别对应于 AES_KEY[0:63]的高 32 位与低 32 位，KEY1H[31:0]和 KEY1L[31:0]分别对应于 AES_KEY[64:127]的高 32 位与低 32 位，KEY2H[31:0]和 KEY2L[31:0]分别对应于 AES_KEY[128:191]的高 32 位与低 32 位，KEY3H[31:0]和 KEY3L[31:0]分别对应于 AES_KEY[192:255]的高 32 位与低 32 位。

CAU_KEY0H

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY0H[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY0H[15:0]															
w															

CAU_KEY0L

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY0L[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY0L[15:0]															
w															

CAU_KEY1H

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY1H[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY1H[15:0]															
w															

CAU_KEY1L

偏移地址：0x2C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY1L[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY1L[15:0]															
w															

CAU_KEY2H

偏移地址：0x30

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY2H[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY2H[15:0]															
w															

CAU_KEY2L

偏移地址：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY2L[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY2L[15:0]															
w															

CAU_KEY3H

偏移地址：0x38

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY3H[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY3H[15:0]															
w															

CAU_KEY3L

偏移地址：0x3C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY3L[31:16]															
w															

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

KEY3L[15:0]

w

位/位域	名称	描述
31:0	KEY0...3(H/L)	用于 DES 或 TDES 或 AES 的密钥

13.9.10. 初始化向量寄存器（CAU_IV0..1（H/L））

偏移地址：0x40~0x4C
复位值：0x0000 0000

该寄存器只能按字（32 位）访问，必须在 BUSY 位为 0 时写这些寄存器。

在 DES/TDES 模式下，IV0H 和 IV0L 分别对应于初始化向量的高 32 位和低 32 位。

在 AES 模式下，IV0H 和 IV1H 分别对应于 128 位初始化向量的最高 32 位和最低 32 位。

CAU_IV0H

偏移地址：0x40
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV0H[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV0H[15:0]															
rw															

CAU_IV0L

偏移地址：0x44
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV0L[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV0L[15:0]															
rw															

CAU_IV1H

偏移地址：0x48

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV1H[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV1H[15:0]															
rw															

CAU_IV1L

偏移地址：0x4C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV1L[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV1L[15:0]															
rw															

位/位域	名称	描述
31:0	IV0...1(H/L)	用于 DES 或 TDES 或 AES 的初始化向量

13.9.11. GCM 或 CCM 模式上下文交换寄存器 x（CAU_GCMCCMCTXSx） （x=0..7）

偏移地址：0x50 ~ 0x6C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

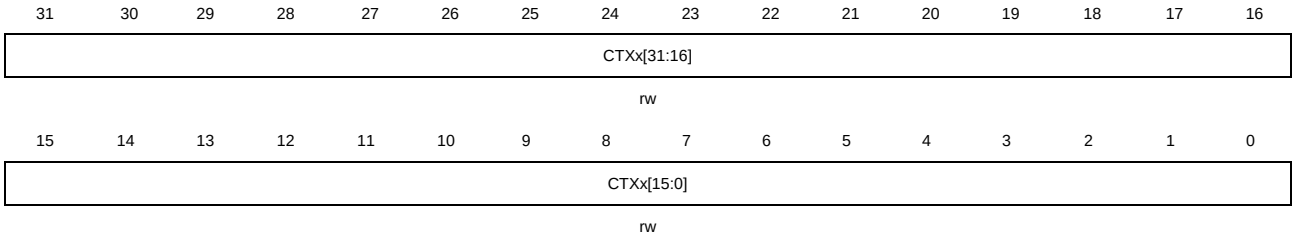
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTXx[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTXx[15:0]															
rw															

位/位域	名称	描述
31:0	CTXx[31:0]	CAU处理器的内部状态信息。当有一个更高优先级的任务需要处理时，读取并保存这些寄存器的数据，恢复的时候将保存的数据写回到这些寄存器从而恢复前面被挂起的任务。 **注意：**这些寄存器只能在GCM，GMAC，或CCM模式下使用。

13.9.12. GCM 模式上下文交换寄存器 x（CAU_GCMCTXSx） （x=0..7）

偏移地址：0x70 ~ 0x8C
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:0	CTXx[31:0]	CAU处理器的内部状态信息。当有一个更高优先级的任务需要处理时，读取并保存这些寄存器的数据，恢复的时候将保存的数据写回到这些寄存器从而恢复前面被挂起的任务。 **注意：**这些寄存器只能在GCM或GMAC模式下使用。

14. 三角函数加速器（TMU）

14.1. 简介

三角函数加速器（TMU）是一个完全可配置的单元，可执行常见的三角运算和算术运算操作。TMU 可以减轻 CPU 的负担，通常应用于电机控制，信号处理和很多其他应用场景。

TMU 可以计算 10 种函数，输入和输出数据支持 q1.31、q1.15 定点格式和 IEEE754 32 位单精度浮点格式。

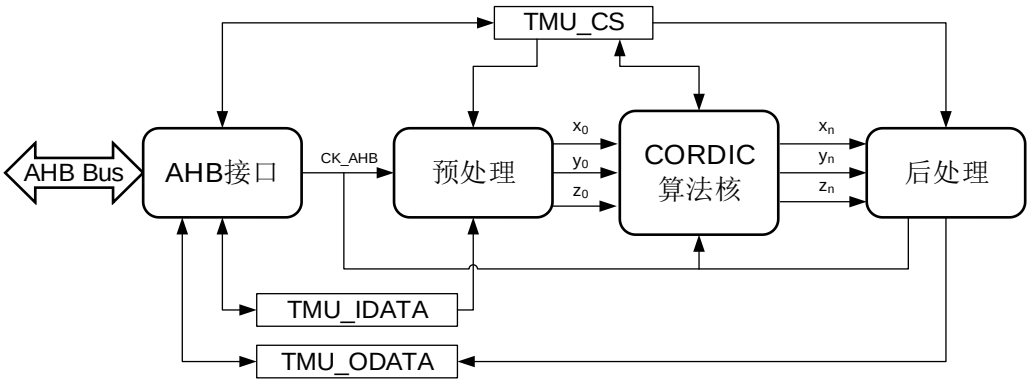
14.2. 主要特征

- 10 种函数；
- 中断和 DMA 请求；
- 可配置的数据格式：q1.31、q1.15 定点格式和 IEEE754 32 位单精度浮点格式；
- 可编程的计算精度；
- CORDIC 算法核：支持两种圆周系统和双曲线系统，支持旋转模式和向量模式。

14.3. 结构框图

*图 14-1. TMU 模块结构框图*TMU 模块内部结构细节。

图 14-1. TMU 模块结构框图



预处理模块将输入数据寄存器（TMU_IDATA）中的数据进行转换，得到 CORDIC 算法核需要的初始数据(x_0,y_0,z_0)。输入数据寄存器中的数据格式为 q1.31、q1.15 定点格式或 IEEE754 32 位单精度浮点格式。

CORDIC 算法核心模块根据初始数据(x_0,y_0,z_0)，经过迭代和运算，得到(x_n,y_n,z_n)。TMU 算法核心

模块支持圆周系统和双曲线系统，每种系统支持旋转模式和向量模式。

后处理模块对(x_n,y_n,z_n)进行数据转换和缩放等处理，并将处理后的数据写入输出数据寄存器（TMU_ODATA）。输出数据寄存器中的数据格式为 q1.31、q1.15 定点格式或 IEEE754 32 位单精度浮点格式。

14.4. 功能描述

14.4.1. 数据格式和配置

当 IFLTEN / OFLTEN 位复位时，TMU 模块的输入/输出数据是定点有符号整型格式（q1.31、q1.15 格式）。当 IFLTEN / OFLTEN 位置位时，TMU 模块的输入/输出数据是 IEEE754 32 位单精度浮点格式。

q1.31 格式中，第 31 位是符号位，0~30 位是小数位，表达数值范围是[-1,1-2^{-31}]，对应[0x80000000,0x7FFFFFFF]。

q1.15 格式中，第 15 位是符号位，0~14 位是小数位，表达数值范围是[-1,1-2^{-15}]，对应[0x8000,0x7FFF]。

IEEE754 32 位单精度浮点格式的数据范围如*表 14-3. IEEE754 32 位单精度浮点格式*所示。

当输入数据格式为定点时（IFLTEN=0），可以通过 TMU_CS 寄存器中的 IWIDTH 位配置输入数据的定点格式。当输入数据格式为浮点时（IFLTEN=1），IWIDTH 位的配置无效。每个模式需要的输入数据的个数有所不同（例如，模式 0 需要两个输入数据，模式 5 只需要一个），可以通过 TMU_CS 寄存器的 INUM 位配置输入数据的数量。详细配置参考*表 14-1. 输入数据配置*。

注意：当输入数据配置为 q1.15 格式，如果所配模式需要两个输入数据，只需要写一次 TMU_IDATA 寄存器，第一个输入数据在低半字，第二个输入数据在高半字。如果所配模式只需要一个输入数据，则只使用低半字，高半字的不使用。

表 14-1. 输入数据配置

IWIDTH 位	INUM 位	IFLTEN 位	数据格式	写 TMU_IDATA 寄存器
0	0	0	q1.31 定点	写一次
0	1	0	q1.31 定点	连续写两次
1	0	0	q1.15 定点	写一次
1	1	0	q1.15 定点	不可用
X	0	1	IEEE754 32 位单精度浮点	写一次
X	1	1		连续写两次

当输出数据格式为定点时（OFLTEN=0），可以通过 TMU_CS 寄存器的 OWIDTH 位配置输出数据的定点格式。当输出数据格式为浮点时（OFLTEN=1），OWIDTH 位的配置无效。每个模式的输出数据的个数有所不同（例如，模式 0 有两个输出数据，模式 8 只有一个），可以通过 TMU_CS 寄

存器的 ONUM 位配置输出数据的数量。详细配置参考*表 14-2. 输出数据配置*。

注意：当输出数据配置为 q1.15 格式，如果所配模式有两个输出数据，只需要读一次 TMU_ODATA 寄存器。第一个输出数据在低半字，第二个输出数据在高半字。如果所配模式只有一个输出，则只使用低半字，不使用高半字。

表 14-2. 输出数据配置

OWIDTH 位	ONUM 位	OFLTEN 位	数据格式	读 TMU_ODATA 寄存器
0	0	0	q1.31 定点	读一次
0	1	0	q1.31 定点	连续读两次
1	0	0	q1.15 定点	读一次
1	1	0	q1.15 定点	不可用
X	0	1	IEEE754 32 位单精度浮点	读一次
X	1	1		连续读两次

14.4.2. 浮点数据格式

表 14-3. IEEE754 32 位单精度浮点格式

S [31]	E [30:23]	M [22:0]	值（V）
0	0	0	零（V = 0）
1	0	0	负零（V = -0）
0 + ve 1 - ve	0	非零	非规格化($V=(-1)^s*2^{(-126)}*(0.M)$)
0 + ve 1 - ve	1 to 254	0 to 0x7FFFFF	正常范围($V=(-1)^s*2^{(E-127)}*(1.M)$)
0	254	0x7FFFFF	正最大值（V = +Max）
1	254	0x7FFFFF	负最大值（V = -Max）
0	最大值 255	0	正无穷（V = + Infinity）
1	最大值 255	0	负无穷（V = - Infinity）
x	最大值 255	非零	非数（V = NaN）

TMU 对于 IEEE 浮点数据格式中不同的数据处理如下：

负零：如果运算结果是 0，TMU 将会输出正零（S=0, E=0, M=0），不会产生负零。所有的 TMU 运算都将负零当做零处理。

非规格化数：输入的非规格化操作数（E=0, M!=0）被视为 0（E=0, M=0）。

非数（NaN）：输入的 NaN 操作数（E=max, M!=0）被视为无穷（E=max, M=0）。

溢出：当 IFLTEN 和 OFLTEN 位同时被置位时，如果输入数据或者运算中的数据按照既定浮点格式显得过大会产生上溢，在这种情景下，TMU_CS 寄存器的 OVRF 标志位置位。

14.4.3. 模式配置

TMU_CS 寄存器的 MODE[3:0]位域用来配置 CORDIC 算法核模块的运行模式。不同的模式使用不同的系统（圆周系统或者双曲线系统）和不同的模式（旋转模式或者向量模式）。详细信息参考*表 14-4. TMU 模式配置*。

表 14-4. TMU 模式配置

模式	第一个输入数据	第二个输入数据	第一个输出数据	第二个输出数据	使用的系统和模式
模式 0	θ	m	m*cos(θ)	m*sin(θ)	圆周系统，旋转模式
模式 1	θ	m	m*sin(θ)	m*cos(θ)	圆周系统，旋转模式
模式 2	x	y	atan2 (y,x)	$\sqrt{x^2+y^2}$	圆周系统，向量模式
模式 3	x	y	$\sqrt{x^2+y^2}$	atan2 (y,x)	圆周系统，向量模式
模式 4	x	无	$\tan^{-1}$ (x)	无	圆周系统，向量模式
模式 5	x	无	cosh (x)	sinh (x)	双曲线系统，旋转模式
模式 6	x	无	sinh (x)	cosh (x)	双曲线系统，旋转模式
模式 7	x	无	$\tanh^{-1}$ (x)	无	双曲线系统，向量模式
模式 8	x	无	ln (x)	无	双曲线系统，向量模式
模式 9	x	无	$\sqrt{x}$	无	双曲线系统，向量模式

尽管 TMU 算法仅能够直接计算少量的函数，但更多的函数可以通过间接的方法来获得。比如，$e^x = \sinh(x) + \cosh(x)$。

模式 0：m*cos(θ)

该模式用来计算余弦函数，有两个输入数据和两个输出数据。

当 IFLTEN 位和 OFLTEN 位均置位时，详细信息参考*表 14-5. 模式 0 描述，IFLTEN 位和 OFLTEN 位均置位时*。

表 14-5. 模式 0 描述，IFLTEN 位和 OFLTEN 位均置位时

参数	范围	描述
第一个输入数据	$\frac{\theta}{\pi}$ ∈(-2^24,2^24)	角度值θ单位是弧度（rad），超出范围时溢出。
第二个输入数据	m∈(-∞,+∞)	超出浮点表示范围时溢出
第一个输出数据	m* cos(θ) ∈(-∞,+∞)	--
第二个输出数据	m* sin(θ) ∈(-∞,+∞)	--
缩放因子 FACTOR[2:0]	不可用	保持复位值 3'b000

当 IFLTEN 位或 OFLTEN 位复位时，详细信息参考*表 14-6. 模式 0 描述，IFLTEN 位或 OFLTEN*

位复位时。

表 14-6. 模式 0 描述，IFLTEN 位或 OFLTEN 位复位时

参数	范围	描述
第一个输入数据	$\frac{\theta}{\pi} \in [-1,1)$	角度值θ单位是弧度（rad），范围 θ∈[-π,π)。软件用θ除以π后，转换为[-1,1)范围内，再按照 q1.31、q1.15 或者 float 格式写入 TMU_IDATA 寄存器
第二个输入数据	m ∈ [0,1)	当0 ≤ m < 1时，直接按照 q1.31、q1.15 或者 float 格式写入 TMU_IDATA 寄存器。当m ≥ 1时，需要先通过软件缩小m到[0,1)范围内，再写入到寄存器中。
第一个输出数据	m*cos(θ),范围[-1,1)	如果之前软件缩小过m，需要对该输出数据进行相应比例的放大，以获得真实结果。
第二个输出数据	m*sin(θ),范围[-1,1)	
缩放因子 FACTOR[2:0]	不可用	保持复位值 3'b000

注意：当模长m > 1时，软件缩放的比例是自行选择的。

例如，计算$100*\cos\left(\frac{\pi}{2}\right)$，当 IFLTEN 位和 OFLTEN 位均复位，IWIDTH 位和 OWIDTH 位均置位时，输入输出数据均为 q1.15 格式。可以按照以下步骤进行：

1. 软件处理角度$\frac{\pi}{2}$。$\frac{\frac{\pi}{2}}{\pi}=\frac{1}{2}$，q1.15 格式下数据为 0x4000。
2. 软件处理模长m，将其缩小 128 倍。$\frac{100}{128}$=0.78125，q1.15 格式下数据为 0x6400。
3. 往寄存器 TMU_IDATA 写第一个输入数据：角度值 0x4000。
4. 往寄存器 TMU_IDATA 写第二个输入数据：模长 0x6400。
5. 等待 ENDF 标志置 1，读 TMU_ODATA 获取第一个输出数据$y_1=\frac{100}{128}*\cos\left(\frac{\pi}{2}\right)$，再读一次 TMU_ODATA 获取第二个输出数据$y_2=\frac{100}{128}*\sin\left(\frac{\pi}{2}\right)$。
6. 结果处理。由于之前对模长m缩小了 128 倍，结果需要再乘以 128，则$100*\cos\left(\frac{\pi}{2}\right)=128*y_1$。

本例（计算$100*\cos\left(\frac{\pi}{2}\right)$）中对模长m和结果处理使用了 128 倍缩放，当然也可以使用其他缩放倍数，比如 101。

模式 1：m*sin(θ)

该模式用来计算正弦函数，有两个输入数据和两个输出数据。

当 IFLTEN 位和 OFLTEN 位均置位时，详细信息参考*表 14-7. 模式 1 描述，IFLTEN 位和 OFLTEN*

位均置位时。

表 14-7. 模式 1 描述，IFLTEN 位和 OFLTEN 位均置位时

参数	范围	描述
第一个输入数据	$\frac{\theta}{\pi}$∈(-2^24,2^24)	角度值θ单位是弧度（rad），超出范围时溢出。
第二个输入数据	m∈(-∞,+∞)	超出浮点表示范围时溢出
第一个输出数据	m*sin(θ) ∈(-∞,+∞)	--
第二个输出数据	m*cos(θ) ∈(-∞,+∞)	--
缩放因子 FACTOR[2:0]	不可用	保持复位值 3'b000

当 IFLTEN 位或 OFLTEN 位复位时，详细信息参考*表 14-8. 模式 1 描述，IFLTEN 位或 OFLTEN 位复位时*。

表 14-8. 模式 1 描述，IFLTEN 位或 OFLTEN 位复位时

参数	范围	描述
第一个输入数据	$\frac{\theta}{\pi}$ ∈ [-1,1)	角度值θ单位是弧度（rad），范围 θ∈[-π,π)。软件用θ除以π后，转换为[-1,1)范围内，再按照 q1.31、q1.15 或者 float 格式写入 TMU_IDATA 寄存器
第二个输入数据	m ∈ [0,1)	当0 ≤ m < 1时，直接按照 q1.31、q1.15 或者 float 格式写入 TMU_IDATA 寄存器。当m ≥ 1时，需要先通过软件缩小m到[0,1)范围内，再写入到寄存器中。
第一个输出数据	m*sin(θ),范围[-1,1)	如果之前软件缩小过m，需要对该输出数据进行相应比例的放大，以获得真实结果。
第二个输出数据	m*cos(θ),范围[-1,1)	
缩放因子 FACTOR[2:0]	不可用	保持复位值 3'b000

注意：当模长m > 1时，软件缩放的比例是自行选择的。

例如，计算$100*\sin\left(\frac{\pi}{2}\right)$，当 IFLTEN 位和 OFLTEN 位均复位，IWIDTH 位和 OWIDTH 位均置位时，输入输出数据均为 q1.15 格式。可以按照以下步骤进行：

1. 软件处理角度$\frac{\pi}{2}$。$\frac{\frac{\pi}{2}}{\pi}=\frac{1}{2}$，q1.15 格式下数据为 0x4000。
2. 软件处理模长m，将其缩小 128 倍。$\frac{100}{128}$=0.78125，q1.15 格式下数据为 0x6400。
3. 往寄存器 TMU_IDATA 写第一个输入数据：角度值 0x4000。
4. 往寄存器 TMU_IDATA 写第二个输入数据：模长 0x6400。
5. 等待 ENDF 标志置 1，读 TMU_ODATA 获取第一个输出数据$y_1=\frac{100}{128}*\sin\left(\frac{\pi}{2}\right)$，再读一次 TMU_ODATA 获取第二个输出数据$y_2=\frac{100}{128}*\cos\left(\frac{\pi}{2}\right)$。

6. 结果处理。由于之前对模长m缩小了 128 倍，结果需要再乘以 128，则$100*\sin\left(\frac{\pi}{2}\right)=128*y_1$。

本例（计算$100*\sin\left(\frac{\pi}{2}\right)$）中对模长m和结果处理使用了 128 倍缩放，当然也可以使用其他缩放倍数，比如 101。

模式 2: phase= atan2 (y,x)

该模式用来计算atan2(y,x)函数，有两个输入数据和两个输出数据。

当 IFLTEN 位和 OFLTEN 位均置位时，详细信息参考*表 14-9. 模式 2 描述，IFLTEN 位和 OFLTEN 位均置位时*。

表 14-9. 模式 2 描述，IFLTEN 位和 OFLTEN 位均置位时

参数	范围	描述
第一个输入数据	2^-24<|y/x|<2^24	超出范围时溢出。
第二个输入数据		
第一个输出数据	角度 θ∈[-1,1)	坐标位置对应的角度，[-1,1)对应[-π,π)。该输出数据乘以π得到真实角度值。
第二个输出数据	模长 m∈[0,+∞)	$m=\sqrt{x^2+y^2}$，超出浮点表示范围时溢出
缩放因子 FACTOR[2:0]	不可用	保持复位值 3'b000

当 IFLTEN 位或 OFLTEN 位复位时，详细信息参考*表 14-10. 模式 2 描述，IFLTEN 位或 OFLTEN 位复位时*。

表 14-10. 模式 2 描述，IFLTEN 位或 OFLTEN 位复位时

参数	范围	描述
第一个输入数据	x∈[-1,1)	笛卡尔坐标系中横坐标值。如果x≥1或者x<-1，则需要进行软件缩放。
第二个输入数据	y∈[-1,1)	笛卡尔坐标系中纵坐标。如果y≥1或者x<-1，则需要进行软件缩放。
第一个输出数据	角度 θ∈[-1,1)	坐标位置对应的角度，[-1,1)对应[-π,π)。该输出数据乘以π得到真实角度值。
第二个输出数据	模长 m∈[0,1)	$m=\sqrt{x^2+y^2}$。如果之前对x和y进行了缩放，该模长需要进行等比例放大。
缩放因子 FACTOR[2:0]	不可用	保持复位值 3'b000

注意：当 IFLTEN 位或 OFLTEN 位复位时，如果x 或 y超出[-1,1)的范围，或者$\sqrt{x^2+y^2}\geq 1$，需要同时对x和y进行同比例缩放，避免出现超出数据范围的情况。不能只缩放一个，这样可以保证缩放前后坐标对应的角度不变。

例如，计算θ=atan2(5,80)，当 IFLTEN 位和 OFLTEN 位均复位，IWIDTH 位和 OWIDTH 位均置位时，输入输出数据均为 q1.15 格式。可以按照以下步骤进行：

1. 软件缩放。输入数据(5,80)除以 128，结果为 (0.0390625,0.625)，q1.15 格式下数据为 (0x0500,0x5000)。
2. 往寄存器 TMU_IDATA 写入第一个输入数据0x0500。
3. 往寄存器 TMU_IDATA 写入第二个输入数据0x5000。
4. 等待 ENDF 标志置 1，读 TMU_ODATA 获取第一个输出数据θ，再读一次 TMU_ODATA 获取第二个输出数据m。
5. 结果处理。第一个输出数据角度θ乘以π，得到真实弧度。由于之前输入数据缩小了 128 倍，读出的第二个输出数据模长m需要再乘以 128 才是真实模长。

本例（计算θ=atan2(5,80)）中对输入和模长使用了 128 倍缩放，当然也可以使用其他缩放倍数，比如 81。

模式 3: modulus=$\sqrt{x^2+y^2}$

该模式用来计算$\sqrt{x^2+y^2}$函数，有两个输入数据和两个输出数据。

当 IFLTEN 位和 OFLTEN 位均置位时，详细信息参考*表 14-11. 模式 3 描述，IFLTEN 位和 OFLTEN 位均置位时*。

表 14-11. 模式 3 描述，IFLTEN 位和 OFLTEN 位均置位时

参数	范围	描述
第一个输入数据	2^-24<|y/x|<2^24	超出范围时溢出
第二个输入数据		
第一个输出数据	模长 m∈[0,+∞)	角度值θ单位是弧度（rad）
第二个输出数据	角度 θ∈[-1,1)	坐标位置对应的角度，[-1,1)对应[-π,π)。该输出数据乘以π得到真实角度值。
缩放因子 FACTOR[2:0]	不可用	保持复位值 3'b000

当 IFLTEN 位或 OFLTEN 位复位时，详细信息参考*表 14-12. 模式 3 描述，IFLTEN 位或 OFLTEN 位复位时*。

表 14-12. 模式 3 描述，IFLTEN 位或 OFLTEN 位复位时

参数	范围	描述
第一个输入数据	x∈[-1,1)	笛卡尔坐标系中横坐标值。如果x≥1或者x≤-1，则需要进行软件缩放。
第二个输入数据	y∈[-1,1)	笛卡尔坐标系中纵坐标。如果y≥1或者x≤-1，则需要进行软件缩放。
第一个输出数据	模长 m∈[0,1)	模长，m=$\sqrt{x^2+y^2}$。如果之前对x和y进行了缩放，该模长需要进行等比例放大。

参数	范围	描述
第二个输出数据	角度 θ∈[-1,1)	坐标位置对应的角度，[-1,1)对应[-π,π)。该输出数据乘以π得到真实角度值。
缩放因子 FACTOR[2:0]	不可用	保持复位值 3'b000

注意：当 IFLTEN 位或 OFLTEN 位复位时，如果x 或 y超出[-1,1)的范围，或者$\sqrt{x^2+y^2}\geq 1$，需要同时对x和y进行同比例缩放，避免出现超出数据范围的情况。不能只缩放一个，这样可以保证缩放前后坐标对应的角度不变。

例如，计算$\sqrt{5^2+80^2}$，当 IFLTEN 位和 OFLTEN 位均复位，IWIDTH 位和 OWIDTH 位均置位时，输入输出数据均为 q1.15 格式。可以按照以下步骤进行：

1. 软件缩放。输入数据(5,80)除以 128，结果为(0.0390625,0.625)，q1.15 格式下数据为(0x0500,0x5000)。
2. 往寄存器 TMU_IDATA 写入第一个输入数据0x0500。
3. 往寄存器 TMU_IDATA 写入第二个输入数据0x5000，TMU 启动计算。
4. 等待 ENDF 标志置 1，读 TMU_ODATA 获取第一个输出数据m，再读一次 TMU_ODATA 获取第二个输出数据θ。
5. 软件结果处理。由于之前输入数据缩小了 128 倍，读出的第一个输出数据模长m需要再乘以 128 才是真实模长。第二个输出数据角度θ乘以π，得到真实弧度。

本例（计算$\sqrt{5^2+80^2}$）中对输入和模长使用了 128 倍缩放，当然也可以使用其他缩放倍数，比如 81。

模式 4: $\tan^{-1}(x)$

该模式用来计算$\tan^{-1}(x)$函数，有一个输入数据和一个输出数据。

当 IFLTEN 位和 OFLTEN 位均置位时，详细信息参考*表 14-13. 模式 4 描述，IFLTEN 位和 OFLTEN 位均置位时*。

表 14-13. 模式 4 描述，IFLTEN 位和 OFLTEN 位均置位时

参数	范围	描述
输入数据	x∈(-2^7,2^7)	超出范围时溢出
输出数据	$\theta\in(-\frac{1}{2},\frac{1}{2})$	坐标位置对应的角度，[-1,1)对应[-π,π)。该输出数据乘以π得到真实角度值。
缩放因子 FACTOR[2:0]	不可用	保持复位值 3'b000

当 IFLTEN 位或 OFLTEN 位复位时，详细信息参考*表 14-14. 模式 4 描述，IFLTEN 位或 OFLTEN*

位复位时。

表 14-14. 模式 4 描述，IFLTEN 位或 OFLTEN 位复位时

参数	范围	描述
输入数据	$\frac{x}{2^f} \in [-1,1)$	当 x∈[-1,1]时，不需要缩放处理，f=0。如果x超出[-1,1]范围，需要软件缩放，缩放后要保证 $-1 \le x*2^{-f} < 1$，f为缩放因子，把缩放后的数据 $\frac{x}{2^f}$以 q1.15、q1.31 或者浮点格式写入 TMU_IDATA 寄存器。
输出数据	$\frac{\theta}{2^f} \in (-\frac{1}{2},\frac{1}{2})$	[-1,1)对应[-π,π)。该输出数据乘以π和2^f得到真实角度值。
缩放因子 FACTOR[2:0]	f ∈ [0,7]	FACTOR[2:0]配置为 f

注意：当 IFLTEN 位或 OFLTEN 位复位时，缩放因子 FACTOR[2:0]需要配置。

例如，计算$\tan^{-1}(100)$，当 IFLTEN 位和 OFLTEN 位均复位，IWIDTH 位和 OWIDTH 位均置位时，输入输出数据均为 q1.15 格式。可以按照以下步骤进行：

1. 软件缩放。输入数据100除以 128（缩放因子f=7=3'b111），得0.78125，q1.15 格式下数据为0x6400。
2. 缩放因子f=7=3'b111写入 TMU_CS 寄存器的 FACTOR[2:0]位域。
3. 往寄存器 TMU_IDATA 写入输入数据0x6400，TMU 开始计算。
4. 等待 ENDF 标志置 1，读 TMU_ODATA 获取输出数据$\frac{\theta}{2^7}$。
5. 结果处理。输出数据$\frac{\theta}{2^7}$需要乘以π和2^7得到真实弧度。

模式 5: cosh (x)

该模式用来计算cosh (x)函数，有一个输入数据和两个输出数据。

当 IFLTEN 位和 OFLTEN 位均置位时，详细信息参考*表 14-15. 模式 5 描述，IFLTEN 位和 OFLTEN 位均置位时*。

表 14-15. 模式 5 描述，IFLTEN 位和 OFLTEN 位均置位时

参数	范围	描述
输入数据	x∈[-1.118,1.118]	超出范围时溢出
第一个输出数据	cosh (x) ∈[1,1.692]	--
第二个输出数据	sinh (x) ∈[-1.366,1.366]	--
缩放因子 FACTOR[2:0]	不可用	保持复位值 3'b000

当 IFLTEN 位或 OFLTEN 位复位时，详细信息参考*表 14-16. 模式 5 描述，IFLTEN 位或 OFLTEN*

位复位时。

表 14-16. 模式 5 描述，IFLTEN 位或 OFLTEN 位复位时

参数	范围	描述
输入数据	$\frac{x}{2} \in [-0.559,0.559]$	x∈[-1.118,1.118]，需要软件将x除以 2，然后以 q1.15、q1.31 或浮点格式写入 TMU_IDATA 寄存器。
第一个输出数据	$\frac{\cosh(x)}{2} \in [0.5,0.846)$	该输出数据乘以 2 可以得到双曲余弦cosh (x)的值。
第二个输出数据	$\frac{\sinh(x)}{2} \in [-0.683,0.683]$	该输出数据乘以 2 可以得到双曲正弦sinh (x)的值
缩放因子 FACTOR[2:0]	f=1	FACTOR[2:0]配置为 3'b001

注意：当 IFLTEN 位或 OFLTEN 位复位时，缩放因子 FACTOR[2:0]只能配置为 3'b001。

例如，计算cosh (1.0)，当 IFLTEN 位和 OFLTEN 位均复位，IWIDTH 位和 OWIDTH 位均置位时，输入输出数据均为 q1.15 格式。可以按照以下步骤进行：

1. 软件缩放。输入数据 1.0 除以 2（f=3'b001），结果为 0.5，q1.15 格式下数据为0x4000。
2. 将缩放因子f=3'b001写入 TMU_CS 寄存器的 FACTOR[2:0]位域。
3. 往寄存器 TMU_IDATA 写入输入数据0x4000, TMU 开始计算。
4. 等待 ENDF 标志置 1，读 TMU_ODATA 获取第一个输出数据$y_1=\frac{\cosh(1.0)}{2}$，再读一次获取第二个输出数据$y_1=\frac{\sinh(1.0)}{2}$。
5. 结果处理。两个输出数据都乘以 2，得到双曲余弦cosh (x)和双曲正弦sinh (x)。

模式 6: sinh (x)

该模式用来计算sinh (x)函数。有一个输入数据和两个输出数据。

当 IFLTEN 位和 OFLTEN 位均置位时，详细信息参考*表 14-18. 模式 6 描述，IFLTEN 位或 OFLTEN 位复位时*。

表 14-17. 模式 6 描述，IFLTEN 位和 OFLTEN 位均置位时

参数	范围	描述
输入数据	x∈[-1.118,1.118]	超出范围时溢出
第一个输出数据	sinh (x) ∈[-1.366,1.366]	--
第二个输出数据	cosh (x) ∈[1,1.692]	--
缩放因子 FACTOR[2:0]	不可用	保持复位值 3'b000

当 IFLTEN 位或 OFLTEN 位复位时，详细信息参考*表 14-18. 模式 6 描述，IFLTEN 位或 OFLTEN*

位复位时。

表 14-18. 模式 6 描述，IFLTEN 位或 OFLTEN 位复位时

参数	范围	描述
输入数据	$\frac{x}{2} \in [-0.559,0.559]$	x∈[-1.118,1.118]，需要软件将 x 除以 2，然后以 q1.15、q1.31 或浮点格式写入 TMU_IDATA 寄存器。
第一个输出数据	$\frac{\sinh(x)}{2} \in [-0.683,0.683]$	输出数据乘以 2 得到双曲正弦$\sinh(x)$。
第二个输出数据	$\frac{\cosh(x)}{2} \in [0.5,0.846)$	输出数据乘以 2 得到双曲余弦$\cosh(x)$
缩放因子 FACTOR[2:0]	f=1	FACTOR[2:0]配置为 3'b001

注意：当 IFLTEN 位或 OFLTEN 位复位时，缩放因子 FACTOR[2:0]只能配置为 3'b001。

例如，计算$\sinh(1.0)$，当 IFLTEN 位和 OFLTEN 位均复位，IWIDTH 位和 OWIDTH 位均置位时，输入输出数据均为 q1.15 格式。可以按照以下步骤进行：

1. 软件缩放。输入数据 1.0 除以 2（f=3'b001），结果为 0.5，q1.15 格式下数据为0x4000。
2. 将缩放因子f=3'b001写入 TMU_CS 寄存器的 FACTOR[2:0]位域。
3. 往寄存器 TMU_IDATA 写入输入数据0x4000, TMU 开始计算。
4. 等待 ENDF 标志置 1，读 TMU_ODATA 获取第一个输出数据$y_1=\frac{\sinh(1.0)}{2}$，再读一次获取第二个输出数据$y_2=\frac{\cosh(1.0)}{2}$。这两个数据都是 q1.15 格式。
5. 结果处理。两个输出数据都乘以 2，得到双曲正弦$\sinh(x)$和双曲余弦$\cosh(x)$。

模式 7: $\tanh^{-1}(x)$

该模式用来计算$\tanh^{-1}(x)$函数，有一个输入数据和一个输出数据。

当 IFLTEN 位和 OFLTEN 位均置位时，详细信息参考*表 14-19. 模式 7 描述，IFLTEN 位和 OFLTEN 位均置位时*。

表 14-19. 模式 7 描述，IFLTEN 位和 OFLTEN 位均置位时

参数	范围	描述
输入数据	x∈[-0.806,0.806]	超出范围时溢出
输出数据	$\tanh^{-1}(x) \in [-1.118,1.118]$	--
缩放因子 FACTOR[2:0]	不可用	保持复位值 3'b000

当 IFLTEN 位或 OFLTEN 位复位时，详细信息参考*表 14-20. 模式 7 描述，IFLTEN 位或 OFLTEN*

位复位时。

表 14-20. 模式 7 描述，IFLTEN 位或 OFLTEN 位复位时

参数	范围	描述
输入数据	$\frac{x}{2} \in [-0.403,0.403]$	x∈[-0.806,0.806]，软件将 x 除以 2，然后以 q1.15、q1.31 或浮点格式写入 TMU_IDATA 寄存器。
输出数据	$\frac{\tanh^{-1}(x)}{2} \in [-0.559,0.559]$	输出数据乘以 2 得到反双曲正切$\tanh^{-1}(x)$。
缩放因子 FACTOR[2:0]	f=1	FACTOR[2:0]配置为 3'b001

注意：当 IFLTEN 位或 OFLTEN 位复位时，缩放因子 FACTOR[2:0]只能配置为 3'b001。

例如，计算$\tanh^{-1}(0.5)$，当 IFLTEN 位和 OFLTEN 位均复位，IWIDTH 位和 OWIDTH 位均置位时，输入输出数据均为 q1.15 格式。可以按照以下步骤进行：

1. 软件缩放。输入数据 0.5 除以 2（f=3'b001），结果为 0.25，q1.15 格式下数据为0x2000。
2. 将缩放因子f=3'b001写入 TMU_CS 寄存器的 FACTOR[2:0]位域。
3. 往寄存器 TMU_IDATA 写入输入数据0x2000，TMU 开始计算。
4. 等待 ENDF 标志置 1，读 TMU_ODATA 获取输出数据 y=$\frac{\tanh^{-1}(0.5)}{2}$。
5. 结果处理。输出数据乘以 2，得到反双曲正切$\tanh^{-1}(0.5)$。

模式 8：ln (x)

该模式用来计算ln (x)函数，有一个输入数据和一个输出数据。

当 IFLTEN 位和 OFLTEN 位均置位时，详细信息参考*表 14-21. 模式 8 描述，IFLTEN 位和 OFLTEN 位均置位时*。

表 14-21. 模式 8 描述，IFLTEN 位和 OFLTEN 位均置位时

参数	范围	描述
输入数据	x∈[0.107,9.35]	超出范围时溢出
输出数据	ln (x) ∈(-2.235,2.235)	--
缩放因子 FACTOR[2:0]	不可用	保持复位值 3'b000

当 IFLTEN 位或 OFLTEN 位复位时，详细信息参考*表 14-22. 模式 8 描述，IFLTEN 位或 OFLTEN 位复位时*。

表 14-22. 模式 8 描述，IFLTEN 位或 OFLTEN 位复位时

参数	范围	描述
输入数据	$\frac{x}{2^f} \in [0.0535,0.875]$	x∈[0.107,9.35]，需要软件缩放，保证$\frac{x}{2^f}<(1-\frac{1}{2^f})$其

参数	范围	描述
		中f为缩放因子，然后将$\frac{x}{2^f}$以 q1.15、q1.31 或者浮点格式写入 TMU_IDATA 寄存器。
输出数据	$\frac{\ln(x)}{2^{(f+1)}} \in [-0.558,0.137]$	输出数据乘以$2^{(f+1)}$得到自然对数ln (x)。
缩放因子 FACTOR[2:0]	f∈[1,4]	FACTOR[2:0]配置为 f

注意：当 IFLTEN 位或 OFLTEN 位复位时，缩放因子 FACTOR[2:0]需要配置。

例如，计算ln (8)，当 IFLTEN 位和 OFLTEN 位均复位，IWIDTH 位和 OWIDTH 位均置位时，输入输出数据均为 q1.15 格式。可以按照以下步骤进行：

1. 软件缩放。输入数据 8 除以 16（缩放因子f=3'b100），结果为0.5，q1.15 格式下数据为0x4000。
2. 将缩放因子f=3'b100写入 TMU_CS 寄存器的 FACTOR[2:0]位域。
3. 往寄存器 TMU_IDATA 写入输入数据0x4000，TMU 开始计算。
4. 等待 ENDF 标志置 1，读 TMU_ODATA 获取输出数据 y=$\frac{\ln(x)}{2^{(4+1)}}$。
5. 结果处理。输出数据乘以$2^{(4+1)}$，得到自然对数ln (x)。

为保证计算精度，对于不同输入推荐使用*表 14-23. 模式 8 推荐的缩放因子*中的缩放因子。

表 14-23. 模式 8 推荐的缩放因子

输入 x 范围	缩放因子 FACTOR[2:0]	输入数据范围
0.107 ≤ x < 1	3'b001	[0.0535,0.5)
1 ≤ x < 3	3'b010	[0.25,0.75)
3 ≤ x < 7	3'b011	[0.375,0.875)
7 ≤ x < 9.35	3'b100	[0.4375,0.584)

模式 9: $\sqrt{x}$

该模式用来计算$\sqrt{x}$函数，有一个输入数据和一个输出数据。

当 IFLTEN 位和 OFLTEN 位均置位时，详细信息参考*表 14-24. 模式 9 描述，IFLTEN 位和 OFLTEN 位均置位时*。

表 14-24. 模式 9 描述，IFLTEN 位和 OFLTEN 位均置位时

参数	范围	描述
输入数据	x∈[0,+∞)	超出范围时溢出
输出数据	$\sqrt{x} \in [0,+\infty)$	--
缩放因子 FACTOR[2:0]	不可用	保持复位值 3'b000

当 IFLTEN 位或 OFLTEN 位复位时，详细信息参考*表 14-25. 模式 9 描述，IFLTEN 位或 OFLTEN*

位复位时。

表 14-25. 模式 9 描述，IFLTEN 位或 OFLTEN 位复位时

参数	范围	描述
输入数据	$\frac{x}{2^f} \in [0.027,0.875]$	$x \in [0.027,2.34]$。软件进行缩放处理，保证$\frac{x}{2^f} < (1-\frac{1}{2^{f+2}})$，其中 f 为缩放因子。然后将$\frac{x}{2^f}$以 q1.15 或者 q1.31 写入 TMU_IDATA 寄存器。
输出数据	$\frac{\sqrt{x}}{2^f} \in [0.164,0.866]$	输出数据乘以2^f得到$\sqrt{x}$。
缩放因子 FACTOR[2:0]	$f \in [0,2]$	FACTOR[2:0]配置为 f

注意：当 IFLTEN 位或 OFLTEN 位复位时，缩放因子 FACTOR[2:0]需要配置。

例如，计算$\sqrt{2}$，当 IFLTEN 位和 OFLTEN 位均复位，IWIDTH 位和 OWIDTH 位均置位时，输入输出数据均为 q1.15 格式。可以按照以下步骤进行：

1. 软件缩放。输入数据 2 除以 4（缩放因子f=3'b010），结果为0.5，q1.15 格式下数据为0x4000。
2. 将缩放因子f=3'b010写入 TMU_CS 寄存器的 FACTOR[2:0]位域。
3. 往寄存器 TMU_IDATA 写入输入数据0x4000，TMU 开始计算。
4. 等待 ENDF 标志置 1，读 TMU_ODATA 获取输出数据$y_1=\frac{\sqrt{2}}{2^2}$。
5. 结果处理。输出数据乘以2^2，得到$\sqrt{2}$。

为保证计算精度，对于不同输入推荐使用*表 14-26. 模式 9 推荐的缩放因子*中的缩放因子。

表 14-26. 模式 9 推荐的缩放因子

输入x范围	缩放因子 FACTOR[2:0]	输入数据范围
0.027<x<0.75	3’b000	[0.027,0.75)
0.75≤x<1.75	3’b001	[0.375,0.875)
1.75≤x<2.341	3’b010	[0.4375,0.585)

14.4.4. TMU 运算挂起

如果当前正在执行 TMU 运算，可以写 TMU_CS 和 TMU_IDATA 寄存器，写入的内容将被挂起。当 TMU 运算完成（结果被读取，ENDF 标志清零）时，如果挂起的输入数据数量符合配置（定义在被挂起的 TMU_CS 中），TMU 模块将按照挂起的配置和数据开始新一次 TMU 运算。

例如，如果配置的 TMU 模式需要两个 32-bit 输入数据（IWIDTH=0,INUM=1），当往 TMU_IDATA 寄存器写入两个 32-bit 的数据后，TMU 启动一次运算。如果第二个输入参数在下一次 TMU 运算中不改变，此时可以修改 INUM=0。当前一次 TMU 运算结束后，往 TMU_IDATA 寄存器写入一个输入数据，TMU 启动运算，只要 TMU 模式没有改变，第二个参数仍使用之前的数值。

注意：复位后，第二个数值为+1（0x7FFFFFF）。

如果当前已经存在挂起的 TMU 数据，再往 TMU_CS 和 TMU_IDATA 寄存器写入新的数据，则新数据覆盖原来数据，新的 TMU 数据被挂起，原数据的挂起失效。

14.4.5. 零开销

当一个 TMU 运算开始后，可以直接读取输出数据寄存器，在结果返回之前总线会自动插入等待周期。可以按照以下步骤进行：

1. 根据需要 TMU_CS 寄存器。
2. 往 TMU_IDATA 写入需要的参数，启动一个 TMU 运算。
3. 根据需要配置下一次 TMU 模式，并往 TMU_IDATA 写入下一次需要的数据。
4. 读取 TMU_ODATA。总线自动插入等待周期。当读取 TMU_ODATA 操作完成后，在第 3 步骤中配置得 TMU 操作会自动启动。
5. 返回第 3 步。

14.4.6. 中断和 DMA

当 ENDF 标志位置 1 时，如果 TMU_CS 寄存器中的 RIE 为 1，则产生中断请求。ENDF 标志清 0 后，中断请求也清除。

当 OVRF 标志设置为 1 时，如果 TMU_CS 寄存器中的 OVRIE 位为 1，则产生中断请求。当 OVRF 标志被清 0 后，中断请求也被清除。

如果 TMU_CS 寄存器中的 WDEN 为 1 并且此时没有 TMU 挂起，则产生 DMA 请求，DMA 请求的数量却决于 TMU_CS 寄存器中的 INUM 位。如果 TMU_CS 寄存器中的 INUM=0，产生一次 DMA 传输请求。如果 TMU_CS 寄存器中的 INUM=1，产生两次 DMA 传输请求。

当 ENDF 标志位置 1 时，如果 TMU_CS 寄存器中的 RDEN 为 1，则产生 DMA 请求，DMA 请求的数量 TMU_CS 寄存器中的 ONUM 位。如果 TMU_CS 寄存器中的 ONUM=0，产生一次 DMA 传输请求。如果 TMU_CS 寄存器中的 ONUM=1，产生两次 DMA 传输请求。

14.5. TMU 寄存器

TMU 基地址：0x4802 4400

14.5.1. 控制和状态寄存器（TMU_CS）

地址偏移：0x00
复位值：0x0000 0050

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ENDF	OVRF	OVRIE	保留						IWIDTH	OWIDTH	INUM	ONUM	WDEN	RDEN	RIE
r	rc_w1	rw							rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IFLTEN	OFLTEN	保留			FACTOR[2:0]			ITRTNUM[3:0]				MODE[3:0]			
rw	rw				rw			rw				rw			

位/位域	名称	描述
31	ENDF	TMU运算结束标志 0：TMU当前无运算或者正在进行运算 1：TMU运算结束，结果已经写入TMU_ODATA寄存器 当TMU运算结束并且结果已经写入TMU_ODATA寄存器时，该位硬件置1。 读TMU_ODATA寄存器（ONUM+1）次，该位硬件清0。 **注意：**当该位为1时，新的TMU运算不会启动。
30	OVRF	溢出标志 该位由硬件置 1，软件写 1 清 0。 0：浮点数据无溢出 1：浮点数据溢出
29	OVRIE	溢出中断使能 0：溢出中断禁能 1：溢出中断使能
28:23	保留	必须保持复位值。
22	IWIDTH	输入数据位宽 0：32-bit 1：16-bit 该位决定了输入数据的定点格式。 如果配置为32-bit，则写入TMU_IDATA寄存器的数据为q1.31定点格式。 如果配置为16-bit，则写入TMU_IDATA寄存器的数据为q1.15定点格式。第一个数据写入TMU_IDATA的低半字，第二个数据写入TMU_IDATA的高半字。只需要一个输入数据（INUM=0）时，TMU_IDATA的高半字不使用。 **注意：**该位域仅在输入数据为定点格式（IFLTEN=0）时生效。
21	OWIDTH	输出数据位宽 0：32-bit 1：16-bit 该位决定了输出数据的定点格式。 如果配置为32-bit，则TMU_ODATA寄存器包含的输出数据为q1.31定点格式。 如果配置为16-bit，则TMU_ODATA寄存器包含的输出数据为q1.15定点格式。第一个输出数据在TMU_ODATA的低半字，第二个输出数据在TMU_IDATA的高半字。

		注意：该位域仅在输出数据为定点格式（OFLTEN=0）时生效。
20	INUM	写TMU_IDATA寄存器的次数 0：一次 32-bit 写操作。写入 TMU_IDATA 寄存器一次，启动一次 TMU 运算。 1：两次 32-bit 写操作。连续写入 TMU_IDATA 寄存器两次，启动一次 TMU 运算。 **注意**：当输入数据格式为q1.15时（IWIDTH=1，IFLTEN=0），只需要一个32-bit写操作。
19	ONUM	读TMU_ODATA寄存器的次数 0：一次 32-bit 读操作。当 TMU 运算结束，只有一个 32-bit 运算结果传输进 TMU_ODATA 寄存器。读一次 TMU_ODATA 寄存器将清除 ENDF 标志。 1：两次 32-bit 读操作。当 TMU 运算结束，有两个 32-bit 运算结果传输进 TMU_ODATA 寄存器。读两次 TMU_ODATA 寄存器将清除 ENDF 标志。 **注意**：当输出数据格式为q1.15时（OWIDTH=1，OFLTEN=0），只需要一个32-bit读操作。
18	WDEN	DMA 写请求使能 0：禁能 1：使能。当无 TMU 运算挂起时，产生 DMA 写请求。
17	RDEN	DMA 读请求使能 0：禁能 1：使能。当ENDF置1时，产生DMA读请求。
16	RIE	读中断使能 0：禁能 1：使能。当ENDF置1时，产生读中断请求。
15	IFLTEN	输入数据浮点格式使能 0：输入数据为定点有符号整形格式 q1.15 或 q1.31 1：输入数据为 32 位单精度浮点格式
14	OFLTEN	输出数据浮点格式使能 0：输出数据为定点有符号整形格式 q1.15 或 q1.31 1：输出数据为 32 位单精度浮点格式
13:11	保留	必须保持复位值。
10:8	FACTOR[2:0]	缩放因子 该位域定义了缩放因子：$2^{FACTOR[2:0]}$。 000：2^0 001：2^1 010：2^2 … 110：2^6

		111：2^7 当实际输入参数超过规定的输入数据范围[-1,1)，实际输入参数需要除以 $2^{FACTOR[2:0]}$，并且输出数据需要乘以 $2^{FACTOR[2:0]}$以得到实际输出结果，细节如下： TMU_IDATA = 实际输入参数/$2^{FACTOR[2:0]}$ 实际输出结果 = TMU_ODATA*$2^{FACTOR[2:0]}$. **注意：**对模式 8 和模式 9，该位域针对不同输入参数推荐了一些配置。对于模式 0、模式 1、模式 2 和模式 3，建议该位域配置为 3'b000。对模式 5、模式 6 和模式 7，该位域建议配置为 3'b001。
7:4	ITRTNUM[3:0]	迭代次数 该位域定义了CORDIC的迭代次数为：ITRTNUM[3:0]*4. 0000：保留 0001：4次迭代 0010：8次迭代 … 0110：24次迭代 0111~1111：保留 **注意：**迭代次数越高，精度越高。
3:0	MODE[3:0]	TMU模式 0000：模式0，m* cos(θ) 0001：模式1，m*sin(θ) 0010：模式2，phase= atan2 (y,x) 0011：模式3，modulus=$\sqrt{x^2+y^2}$ 0100：模式4，$\tan^{-1}(x)$ 0101：模式5，cosh (x) 0110：模式6，sinh (x) 0111：模式7，$\tanh^{-1}(x)$ 1000：模式8，ln (x) 1001：模式9，$\sqrt{x}$ 1010~1111：保留 **注意：** x、θ：第一个输入数据 y、m：第二个输入数据

14.5.2. 输入数据寄存器（TMU_IDATA）

地址偏移：0x04
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
IDATA[31:16]															
w															

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

位/位域	名称	描述
31:0	IDATA[31:0]	输入数据 输入数据写入该寄存器。细节参考*表 14-1. 输入数据配置*。 **注意：** 1. 当无TMU运算正在进行并且需要的输入参数已经写入该寄存器，将启动一次新的TMU运算。 2. 当TMU正在进行一次运算时，再写入的数据将会被挂起，直到当前的TMU运算结束并且输出数据被读取。在数据挂起期间，如果写入新的数据，则新数据覆盖之前被挂起的数据。

14.5.3. 输出数据寄存器（TMU_ODATA）

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
ODATA[31:16]															
r															

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

位/位域	名称	描述
31:0	ODATA[31:0]	输出数据 当TMU运算结束，结果传输进该寄存器。细节参考*表14-2. 输出数据配置*。 **注意：** 1. 当ENDF位置1时，读取该寄存器可以获得TMU运算结果。 2. 当符合配置的读操作完成，ENDF位被硬件自动清0。

15. 快速傅里叶变换（FFT）

15.1. 简介

快速傅里叶变换（FFT）是离散傅里叶变换（DFT）的高效计算方法。该模块可以进行 FFT 运算，减轻了 CPU 负担。与软件实现相比，该模块可以加速了 FFT 的计算时间。该模块支持 6 个可配置的 FFT 点数，最多 1024，输入和输出数据应为 IEEE-754 单精度浮点复数。

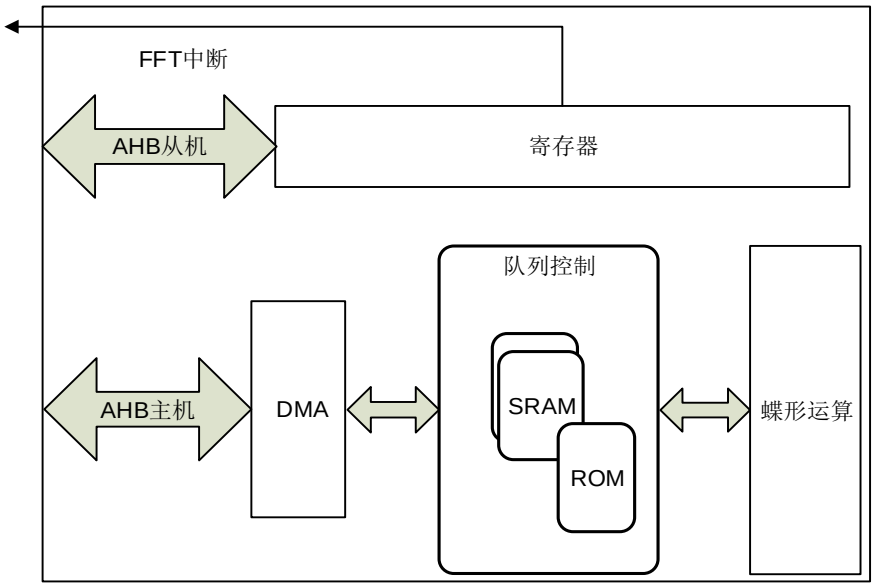
15.2. 主要特性

- 支持1024/512/256/128/64/32点FFT；
- 支持IFFT模式；
- IEEE-754单精度浮点复数输入和输出数据；
- DMA主机加载和存储数据；
- 支持在内存中配置的窗函数功能；
- 支持输入下采样。

15.3. 功能描述

*图15-1. FFT模块框图*提供了FFT模块的内部配置细节。

图 15-1. FFT 模块框图



FFT 模块有 DMA 主机、队列控制子模块、蝶形运算子模块和寄存器组成。该模块是使用的基-2 FFT 算法。先从存储中加载 IEEE754 单精度浮点格式的输入数据（包含实部和虚部）和窗函数数据，经过窗函数运算和输入位反运算之后，将结果写入内部 SRAM。旋转因子已经存储在 ROM 中。

蝶形运算子模块在 FFT 计算中被重复使用，并计算 N / 2 * log_2N 次。每次计算是相同的地址，也就是说，每次计算的输出数据被写回之前存储输入数据的相同地址。

所有蝶形计算迭代完成后，DMA 主机将输出数据从内部 SRAM 传输到内存。

15.4. 数据格式

运算数据和计算结果格式如*表 15-1. IEEE32 位单精度浮点格式*所示。这些数据必须符合 IEEE32 位单精度浮点格式。

表 15-1. IEEE32 位单精度浮点格式

符号位 S[31]	阶数位 E[30:23]	尾数位 M[22:0]	值(V)
0	0	0	零（V = 0）
1	0	0	负零（V = -0）
0+ve 1-ve	0	非零	非规格数（$V=(-1)^s*2^{(-126)}*(0.M)$）
0+ve 1-ve	1 到 254	0 到 0x7FFFFF	正常范围（$V=(-1)^{□}*2^{(E-127)}*(1.M)$）
0	254	0x7FFFFF	V = +Max
1	254	0x7FFFFF	V = -Max
0	max=255	0	正无穷 （V = +Infinity）
1	max=255	0	负无穷 （V = -Infinity）
x	max=255	非零	非数字（V = NaN）

本模块的各种 IEEE 浮点数字格式的处理如下：

非规格化数字：非规格数（E=0，M！=0）输入被视为零（E=O，M=0）。

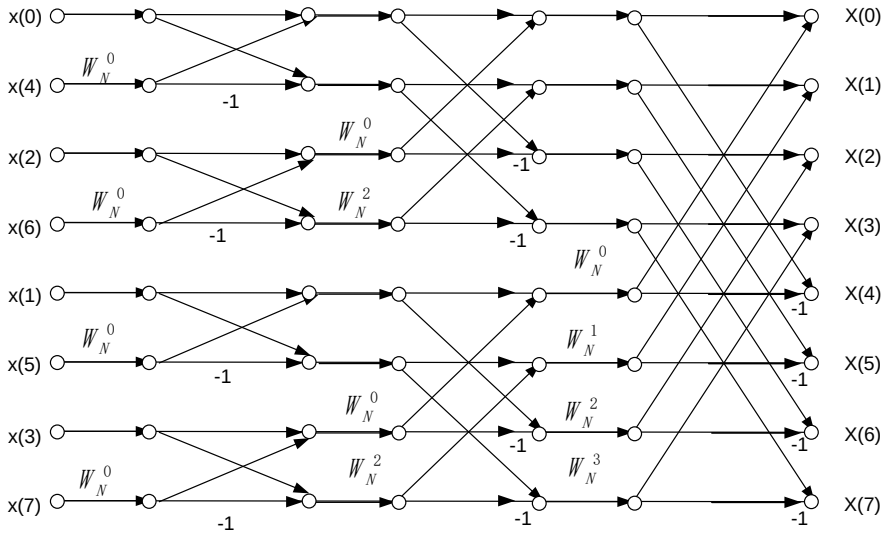
溢出：当操作生成的值太大而无法以给定的浮点格式表示时，会发生溢出。

非数字（NaN）：NaN 操作数（E=max，M！=0）输入被视为无限（E=max，M=0）。

15.5. 基-2 FFT

FFT 模块使用 DIT（时间抽取）基-2 算法，该算法依赖于将 N 点变换递归分解为两个（N / 2）点变换。为了便于说明，8 点时间抽取算法如*图 15-2. 8 点 DIF 的 FFT 流程图*所示。

图 15-2. 8 点 DIF 的 FFT 流程图



8 点输入序列相对于输出以比特倒序出现，如*表 15-2. 8 点输入位倒序操作*所示。

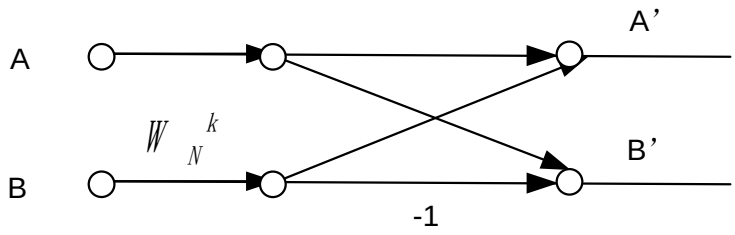
表 15-2. 8 点输入位倒序操作

索引值	二进制索引值	二进制位反索引值	位反索引值
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

15.6. 蝶形运算单元

蝶形运算单元式 FFT 计算的基本元素，其过程如*图 15-3. 蝶形运算框图*所示。

图 15-3. 蝶形运算框图



$$A'=A+BW=Re(A)+Re(BW)+j[Im(A)+Im(BW)]$$

$$B'=A-BW=Re(A)-Re(BW)+j[Im(A)-Im(BW)]$$

其中$W=W_N^k=e^{-j\frac{2\pi}{N}k}$。

15.7. IFFT 模式

IFFT 转换频域矢量信号到时域矢量信号，FFT 与 IFFT 在是线上的主要区别有：

- 旋转因子共轭
- 结果除以N，其中N为FFT的点数

15.8. FFT SRAM

当 FFT_CSR 寄存器的 FFTEN 位为 0 时，可以通过 AHB 访问 FFT 的 8KB SRAM 空间（0x40025800~0x40027FFF）。

注意:

- 当 FFT_CSR 寄存器的 FFTEN 位为 1 时，通过 AHB 访问 FFT SRAM 会发生总线错误。
- 如果 FFT 模块未使能，该 FFT SRAM 可以用作其他用途。

15.9. FFT 循环模式

FFT 支持循环模式。每次 FFTEN 置位，会启动一次运算，当 FFT 运算完 INDEX[15:0]索引加 1。

当每次 FFT 运算完，实部和虚部地址会增加，指向下一组输入数据。也就是说，每次 FFT 运算完成，FFT_LOOPLEN 寄存器的 INDEX[15:0]加 1。实部的起始地址是 FFT_RESADDR + {INDEX[15:0], 2'b00}，虚部的起始地址是 FFT_IMSADDR + {INDEX[15:0], 2'b00}。当 INDEX[15:0]增加到 LENGTH[15:0]，会自动清零。

注意：对应地址的输入数据要准备好。

15.10. 操作指南

该部分介绍 FFT 的推荐操作指南。

1. 如果需要，配置FFT_IMSADDR寄存器来设置FFT虚部的基地址，即虚部数据不为零。
2. 配置FFT_RESADDR寄存器来设置FFT实部基地址。

3. 如果需要，配置WINEN位和FFT_WSADDR寄存器来设置FFT窗函数的基地址。
4. 配置FFT_OSADDR寄存器来设置FFT输出结果的基地址。
5. 如果需要，配置FFT_LOOPLEN寄存器。
6. 如果需要，配置IFFTMODE。
7. 配置FFT_CSR寄存器中的NUMPT[2:0]位域来设置FFT点的数量。
8. 配置FFTEN位。
9. 等待FFTEN清零，或者CCF位置1。

15.11. FFT 中断

以下任一个标志发生都可以产生中断：

- FFT计算完成标志；
- 传输访问错误中断标志模拟看门狗事件。

这些标志都被映射到同一个中断向量 IRQ96。

15.12. FFT 寄存器

FFT 基地址： 0x4002 5000

15.12.1. 控制和状态寄存器（FFT_CSR）

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
DMABSY	CCF	CCIE	TAEIF	TAEIE	保留										
r	rc_w1	rw	rc_w1	rw											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	IMSEL[1:0]		DOWNSAMP[3:0]				WINEN	保留			IFFTMODE	NUMPT[2:0]			FFTEN
	rw		rw				rw				rw	rw			rw

位/位域	名称	描述
31	DMABSY	DMA忙 该位表示DMA是否在忙。 该位只读。 0：DMA不忙 1：DMA在忙
30	CCF	FFT计算完成标志 该位写1清0。 0：FFT正在进行计算 1：FFT计算完成
29	CCIE	使能FFT计算完成中断 0：禁能计算完成中断 1：使能计算完成中断
28	TAEIF	传输访问错误中断标志 该位写1清0。 0：未检测到传输访问错误 1：检测到传输访问错误
27	TAEIE	使能传输访问错误中断 0：禁能传输访问错误中断 1：使能传输访问错误中断

26:15	保留	必须保持复位值。
14:13	IMSEL[1:0]	虚部输入源选择 00：虚部输入来源于 FFT_IMSADDR 01：虚部输入是 0 10：虚部输入是与 FFT_IMSADDR 中的虚部数据符号相反的值
12:9	DOWNSAMP[3:0]	输入数据下采样选择 0000：下采样 1 0001：下采样 2 0010：下采样 3 0011：下采样 4 0100：下采样 5 0101：下采样 6 0110：下采样 7 0111：下采样 8 1000：下采样 9 1001：下采样 10 1010：下采样 11 1011：下采样 12 1100：下采样 13 1101：下采样 14 1110：下采样 15 1111：下采样 16 当 FFT_CSR 寄存器的 FFTEN 位为 1 时，该位不可写。
8	WINEN	使能窗函数 0：无窗函数 1：窗函数使能。窗函数来源于 FFT_WSADDR。 当 FFT_CSR 寄存器的 FFTEN 位为 1 时，该位不可写。
7:5	保留	必须保持复位值。
4	IFFTMODE	使能 IFFT 模式 0：FFT 模式 1：IFFT 模式（FFT 逆变换） 当 FFT_CSR 寄存器的 FFTEN 位为 1 时，该位不可写。
3:1	NUMPT[2:0]	FFT点数 000：32 001：64 010：128 011：256

		100：512 101：1024 其他：保留 当FFT_CSR寄存器的FFTEN位为1时，该位不可写。
0	FFTEN	使能FFT 当FFT计算结束时，该位被自动清0。软件不要清零该位。 0：禁能FFT 1：使能 FFT

15.12.2. 实部基地址寄存器（FFT_RESADDR）

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
RESADDR[31:16]															
rw															

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

位/位域	名称	描述
31:0	RESADDR[31:0]	FFT 实部基地址 该地址必须按照32位格式对齐。 当 FFT_CSR 寄存器的 FFTEN 位为 1 时，该位不可写。

15.12.3. 虚部基地址寄存器（FFT_IMSADDR）

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
IMSADDR[31:16]															
rw															

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

位/位域	名称	描述
31:0	IMSADDR[31:0]	FFT 虚部基地址 该地址必须按照32位格式对齐。 当 FFT_CSR 寄存器的 FFTEN 位为 1 时，该位不可写。

15.12.4. 窗函数基地址寄存器（FFT_WSADDR）

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
WSADDR[31:16]															
rw															

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

位/位域	名称	描述
31:0	WSADDR[31:0]	FFT 窗函数基地址 该地址必须按照32位格式对齐。 当 FFT_CSR 寄存器的 FFTEN 位为 1 时，该位不可写。

15.12.5. 输出基地址寄存器（FFT_OSADDR）

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
OSADDR[31:16]															
rw															

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

位/位域	名称	描述
31:0	OSADDR[31:0]	FFT 输出结果基地址 该地址必须按照32位格式对齐。 当 FFT_CSR 寄存器的 FFTEN 位为 1 时，该位不可写。

15.12.6. 循环长度寄存器（FFT_LOOPLEN）

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
INDEX[15:0]															
rw															

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

位/位域	名称	描述
31:16	INDEX[15:0]	DMA循环缓冲区索引 索引值不能超过LENGTH[15:0]，其范围在0~ LENGTH[15:0]。 每次结束时加1，当增加到LENGTH[15:0]时归零。 实际的实部DMA起始地址 = FFT_RESADDR + { INDEX, 2'b00}。 实际的虚部DMA起始址 = FFT_IMSADDR + {INDEX, 2'b00}。
15:0	LENGTH[15:0]	FFT输入数据的DMA循环缓冲区长度 当FFT_CSR寄存器的FFTEN位为1时，该位不可写。

16. 调试（DBG）

16.1. 简介

GD32G553 系列产品提供了各种各样的调试，跟踪和测试功能。这些功能通过 Arm® CoreSight™ 组件的标准配置和链状连接的 TAP 控制器来实现的。调试和跟踪功能集成在 ARM Cortex®-M33 内核中。调试系统支持串行（SW）调试和跟踪功能，也支持 JTAG 调试。调试和跟踪功能请参考下列文档：

- Cortex®-M33技术参考手册；
- ARM调试接口v5结构规范。

调试系统帮助调试者在低功耗模式下调试。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、RTC、I2C、RTC、CAN、LPTIMER 和 HRTIMER。

16.2. JTAG/SW 功能描述

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

16.2.1. 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。

16.2.2. 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）。

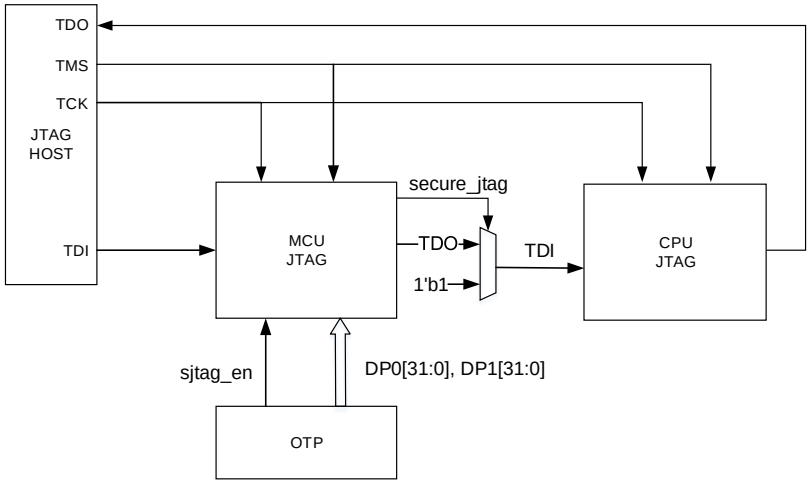
表 16-1. 引脚分配

引脚	调试接口
PA15	JTDI
PA14	JTCK/SWCLK
PA13	JTMS/SWDIO
PB3	JTDO
PB4	NJTRST

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

16.2.3. JTAG

图 16-1. JTAG 模块框图



JTAG 链状结构

Cortex®-M33 内核的 JTAG TAP（CPU JTAG）和 MCU JTAG TAP 串行连接。MCU JTAG 的 IR（指令寄存器）是 5 位，而 Cortex®-M33 内核的 JTAG 的 IR（指令寄存器）是 4 位。所以当 JTAG 进行 IR 移位输入时，首先移位 5 位 BYPASS 指令给 MCU JTAG，然后移位 4 位标准指令给 Cortex®-M33 JTAG。当进行数据移位时，数据链只需要额外添加一位，因为 MCU JTAG 已处在 BYPASS 模式。

MCU JTAG ID 代码是 0x790007A3。

安全 JTAG

1. 安全 JTAG 只支持 JTAG，不支持 SW

2. OTP 配置

OTP 相关位：SWEN, NDBG[1:0], DPx[31:0](x=0,1)

表 16-2. OTP JTAG 字节

地址	名称	属性	描述
0x1fff f000	DP0[31:0]	w	安全 JTAG 密码 0 出厂值：0xFFFFFFFF
0x1fff f004	DP1[31:0]	w	安全 JTAG 密码 1 出厂值：0xFFFFFFFF
0x1fff f008	OTP_USER[31:0]	rw	[31:16]：LK 写秘钥值 0x3CC3 上锁 OTP JTAG 字节，其它值解锁 出厂值：0xFFFF [15:3]：保留 [2:1]：NDBG 0x：无调试 10：安全 JTAG 11：无安全 JTAG（出厂值） [0] ：SWEN 0：SW 失能 1：SW 使能（出厂值）
0x1fff f00c	-	-	保留

注意：OTP JTAG bytes 保持出厂值，全 FFFF，设置 LK[31:16] 秘钥值 0x3cc3 之后，保护 OTP JTAG bytes 的值不被修改。如果 OTP JTAG bytes 的值已经修改过，设置 LK[31:16] 为 0x3cc3 的值将没有作用。

3. 安全 JTAG 的使用

a) 配置 OTP 为安全 JTAG：首先配置 JTAG 安全密码 DPx[31:0]（x = 0，1），再配置 NDBG[1:0] = 2b’10。

b) 电源复位：电源复位后，JTAG 处于安全状态，secure_jtag 为 1，此时无法通过 JTAG 操作 CPU。

c) 安全 JTAG 解除：JTAG 主机依次将以下两个密码写入 MCU JTAG 以解除安全模式。此时 secure_jtag 为 0，可通过 JTAG 对 CPU 进行操作。

IR：写入 5’b10101，DR：写入 DP0[31:0]。

IR：写入 5′b10110，DR：写入 DP2[31:0]。

注意：

1 如果密码输入错误，则需电源复位。

2 发生任何错误的输入序列后，若想重新解密都需要电源复位。

d) 读取写入值和 JTAG 状态。

IR：写入 5′b11000，DR：可读出 IR 为 5′b10101 写入的值，检查写入值是否正确。

IR：写入 5′b11001，DR：可读出 IR 为 5′b10110 写入的值，检查写入值是否正确。

IR：写入 5′b11010，DR：可读出{30‘b0, wrong_seq, secure_jtag}。secure_jtag 表示 JTAG 状态，其中：

1：无法通过 JTAG 操作 CPU；

0：可以通过 JTAG 操作 CPU。

wrong_seq 表示解密过程错误标志，其中：

1：解密过程发生错误；

0：解密过程未发生错误。

16.2.4. 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，ITM）。NJTRST 能复位 JTAG TAP 控制器。所以，可以在系统复位下实现调试功能。例如：复位后停止，用户在系统复位后配置相应停止位，系统复位释放后处理器会立即停止。

16.2.5. 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。

16.3. 调试保持功能描述

16.3.1. 低功耗模式调试支持

当 DBG 控制寄存器 0（DBG_CTL0）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 0（DBG_CTL0）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，退出深度睡眠时，PLL 关闭，

系统时钟切换到 IRC8M。

当 DBG 控制寄存器 0（DBG_CTL0）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。

16.3.2. TIMER, I2C, WWDGT, FWDGT, RTC, CAN, LPTIMER 和 HRTIMER 外设调试支持

当内核停止，并且 DBG 控制寄存器 x（DBG_CTLx，x=0，1，2）中的相应位置 1。对于不同外设，有不同动作：

对于 TIMER/LPTIMER/HRTIMER 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 RTC 外设，计数器停止并进行调试；

对于 CAN 外设，接收寄存器停止计数并进行调试。

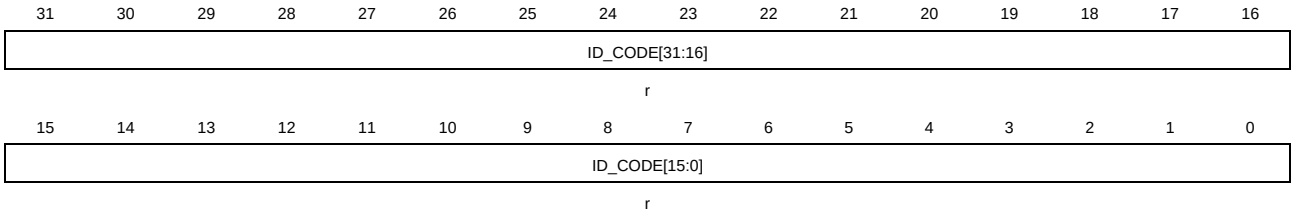
16.4. DBG 寄存器

DBG 基地址：0xE004 4000

16.4.1. ID 寄存器（DBG_ID）

地址偏移：0x00
复位值：0xXXXX XXXX
只读寄存器

该寄存器只能按字（32 位）访问。



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

16.4.2. 控制寄存器 0（DBG_CTL0）

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。

位/位域	名称	描述
31:6	保留	必须保持复位值。
5	TRACE_IOEN	跟踪引脚分配使能 该位由软件置位和复位。

		0：跟踪引脚分配禁用 1：跟踪引脚分配使能。
4:3	保留	必须保持复位值。
2	STB_HOLD	待机模式保持位 该位由软件置位和复位 0：无影响 1：在待机模式下，所有工作的时钟继续运行，支持待机模式下调试。
1	DSLP_HOLD	深度睡眠模式保持 该位由软件置位和复位 0：无影响 1：在深度睡眠模式下，所有工作的时钟继续运行，支持深度睡眠模式下调试。
0	SLP_HOLD	睡眠模式保持位 该位由软件置位和复位 0：无影响 1：在睡眠模式下，所有工作时钟继续运行，支持睡眠模式下调试。

16.4.3. 控制寄存器 1（DBG_CTL1）

地址偏移：0x08
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
LPTIMER_HOLD	保留						I2C3_HOLD	I2C2_HOLD	I2C1_HOLD	I2C0_HOLD	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			FWDGT_HOLD	WWDGT_HOLD	RTC_HOLD	保留				TIMER6_HOLD	TIMER5_HOLD	TIMER4_HOLD	TIMER3_HOLD	TIMER2_HOLD	TIMER1_HOLD
			rw	rw	rw					rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	LPTIMER_HOLD	LPTIMER 保持位 该位由软件置位和复位 0：无影响 1：当内核停止时保持 LPTIMER 计数器不变，用于调试。
30:25	保留	必须保持复位值。

24	I2C3_HOLD	I2C3 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 I2C3 的 SMBUS 状态不变，用于调试。
23	I2C2_HOLD	I2C2 保持寄存器 该位由软件置位和复位。 0：无影响 1：当内核停止时保持 I2C2 的 SMBUS 状态不变，用于调试。
22	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。
21	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。
20:13	保留	必须保持复位值。
12	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。
11	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。
10	RTC_HOLD	RTC 保持位 该位由软件置位和复位 0：无影响 1：当内核停止时保持 RTC 计数器不变，用于调试。
9:6	保留	必须保持复位值。
5	TIMER6_HOLD	TIMER6 保持位 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。
4	TIMER5_HOLD	TIMER5 保持位

		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。
3	TIMER4_HOLD	TIMER4 保持位 该位由软件置位和复位 0：无影响 1：当内核停止时保持定时器 4 计数器不变，用于调试。
2	TIMER3_HOLD	TIMER3 保持位 该位由软件置位和复位 0：无影响 1：当内核停止时保持定时器 3 计数器不变，用于调试。
1	TIMER2_HOLD	TIMER2 保持位 该位由软件置位和复位 0：无影响 1：当内核停止时保持定时器 2 计数器不变，用于调试。
0	TIMER1_HOLD	TIMER1 保持位 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。

16.4.4. 控制寄存器 2（DBG_CTL2）

地址偏移：0x0C

复位值：0x0000 0000，仅上电复位

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留					HRTIMER_HOLD	保留					TIMER19_HOLD	保留	TIMER16_HOLD	TIMER15_HOLD	TIMER14_HOLD
					rw						rw		rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		TIMER7_HOLD	保留	TIMER0_HOLD	保留								CAN2_HOLD	CAN1_HOLD	CAN0_HOLD
		rw		rw									rw	rw	rw

位/位域	名称	描述
31:27	保留	必须保持复位值。
26	HRTIMER_HOLD	HRTIMER 保持位

		该位由软件置位和复位 0：无影响 1：当内核停止时保持 HRTIMER 计数器不变，用于调试。
25:21	保留	必须保持复位值。
20	TIMER19_HOLD	TIMER19 保持位 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。
19	保留	必须保持复位值。
18	TIMER16_HOLD	TIMER16 保持位 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 16 计数器不变，用于调试。
17	TIMER15_HOLD	TIMER15 保持位 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 15 计数器不变，用于调试。
16	TIMER14_HOLD	TIMER14 保持位 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 14 计数器不变，用于调试。
15:14	保留	必须保持复位值。
13	TIMER7_HOLD	TIMER7 保持位 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。
12	保留	必须保持复位值。
11	TIMER0_HOLD	TIMER0 保持位 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。
10:3	保留	必须保持复位值。
2	CAN2_HOLD	CAN2 保持位 该位由软件置位和复位 0：无影响

		1：当内核停止时 CAN2 接收寄存器停止接收数据。
1	CAN1_HOLD	CAN1 保持位 该位由软件置位和复位 0：无影响 1：当内核停止时 CAN1 接收寄存器停止接收数据。
0	CAN0_HOLD	CAN0 保持位 该位由软件置位和复位 0：无影响 1：当内核停止时 CAN0 接收寄存器停止接收数据。

17. 模数转换器（ADC）

17.1. 简介

MCU 片上集成了 12 位逐次逼近式模数转换器模块（ADC），ADC0 有 14 个外部通道，5 个内部通道（内部温度传感通道（V_{SENSE}）、电池电压（V_{BAT}）通道、DAC0_OUT0 通道、DAC0_OUT1 通道和参考电压输入通道（V_{REFINT}）），ADC1 有 16 个外部通道，3 个内部通道（DAC1_OUT0 通道、DAC1_OUT1 通道和参考电压输入通道（V_{REFINT}）），ADC2 有 15 个外部通道，5 个内部通道（参考电压输入通道（V_{REFINT}）、DAC2_OUT0 通道、DAC2_OUT1 通道、高精度温度传感器通道（V_{SENSE2}）和电池电压（V_{BAT}）通道），ADC3 有 18 个外部通道，3 个内部通道（DAC3_OUT0 通道、DAC3_OUT1 通道和参考电压输入通道（V_{REFINT}））。ADC 采样通道均支持多种运行模式，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。

17.2. 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；
  - ADC采样率：12位分辨率为5.3 MSPs，10位分辨率为6.15 MSPs，8位分辨率为7.27 MSPs，6位分辨率为8.89 MSPs。分辨率越低，转换越快；
  - 前置校准时间：902个ADC时钟周期；
  - 可编程采样时间；
  - 数据存储模式：最高有效位对齐和最低有效位对齐；
  - DMA请求。
- 模拟输入通道：
  - ADC0有14个外部模拟输入通道，ADC1有16个外部模拟输入通道，ADC2有15个外部模拟输入通道，ADC3有18个外部模拟输入通道；
  - 1个内部温度传感通道（V_{SENSE}）；
  - 1个内部参考电压输入通道（V_{REFINT}）；
  - 1个外部监测电池V_{BAT}供电引脚输入通道；
  - 1个内部高精度温度传感器通道（V_{SENSE2}）；
  - 与DAC内部通道连接。
- 转换开始的发起：
  - 软件；
  - TRIGSEL触发。
- 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）。
- 转换结果阈值监测器功能：模拟看门狗。
- 常规序列转换结束、模拟看门狗事件和溢出事件都可以产生中断。
- 过采样：
  - 32位的数据寄存器；
  - 可调整的过采样率，从2x到1024x；
  - 11位的可编程数据移位。
- 通道输入范围：$V_{REFN} \leq V_{IN} \leq V_{REFP}$；
- 数据可以路由到HPDF进行后期处理。

17.3. 引脚和内部信号

*图 17-1. ADC 模块框图*给出了 ADC 框图。*表 17-1. ADC 内部输入信号*给出了 ADC 内部信号。*表 17-2. ADC 输入引脚定义*给出了 ADC 引脚说明。

表 17-1. ADC 内部输入信号

内部信号名称	说明
V_{SENSE}	内部温度传感器输出电压
V_{SENSE2}	内部高精度温度传感器输出电压
V_{REFINT}	内部参考输出电压
V_{BAT}	外部电池电压

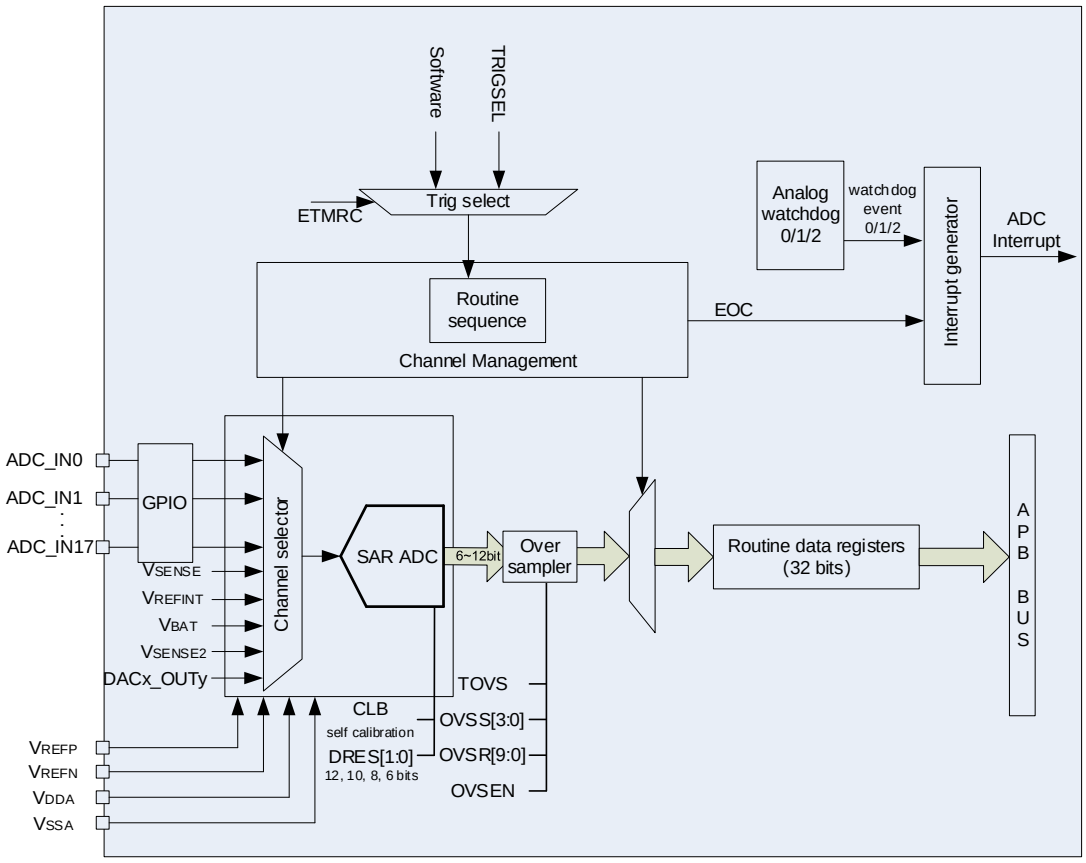
表 17-2. ADC 输入引脚定义

名称	注释
V_{DDA}	模拟电源输入等于V_{DD}
V_{SSA}	模拟地，等于V_{SS}
V_{REFP}	ADC正参考电压
V_{REFN}	ADC负参考电压
ADCx_IN[17:0]	多达18路外部通道

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

17.4. 功能描述

图 17-1. ADC 模块框图



17.4.1. 前置校准功能

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

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

内部的模拟校准通过设置 ADC_CTL1 寄存器的 RSTCLB 位来重置。

软件校准过程：

1. 确保ADCON=1；
2. 延迟14个CK_ADC以等待ADC稳定；
3. 设置RSTCLB （可选的）；

4. 设置CLB=1；
5. 等待直到CLB=0。

17.4.2. 双时钟域架构

时钟控制器提供的 CK_ADC 时钟与 AHB 时钟同步。在此模式下，ADC_SYNCCTL 寄存器中的 ADCSCK[3:0]不能设置为 0000。分割因子可以是 2、4、6、8、10、12、14、16，最大频率为 80 MHz。

CK_ADC 也可以由 CK_PLLR 或 CK_SYS 提供，后者可以是异步的，独立于 AHB 时钟。在此模式下，ADC_SYNCCTL 中的 ADCSCK[3:0]应设置为 0000。可通过 ADC_SYNCCTL 的 ADCCK[3:0]配置分割因子。

RCU 控制器具有专用于 ADC 时钟的可编程预分频器。

注意：ADC1 和 ADC2 时钟共享 ADC0 时钟，当使用 ADC1 和 ADC2 时，必须打开 ADC0 时钟，且只能通过 ADC0 进行时钟分频。

17.4.3. ADCON 使能

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

17.4.4. 单端和差分输入通道

通过配置 ADC_DIFCTL 寄存器中的 DIFCTL[21:0]位域，可以配置 ADC 通道为单端输入模式或差分输入模式。只有在 ADC 禁能（ADCON = 0）的情况下才能进行该配置。

单端输入模式下，通道 n 要转换的模拟电压是外部电压 V_{INn}（正输入）和 V_{REFN}（负输入）之间的差。差分输入模式下，通道 n 要转换的模拟电压是外部电压 V_{INn}（正输入）和通道 m 外部电压 V_{INm}（负输入）之间的差。差分通道引脚分配如*表 17-3. ADC 差分通道引脚匹配*。

表 17-3. ADC 差分通道引脚匹配

差分通道n编号	ADC0		ADC1		ADC2		ADC3	
	V_{INn}引脚	V_{INm}引脚	V_{INn}引脚	V_{INm}引脚	V_{INn}引脚	V_{INm}引脚	V_{INn}引脚	V_{INm}引脚
0	PA0	PA1	PA0	PA1	PB1	PE9	PE14	PE15
1	PA1	PA2	PA1	PA6	PE9	PE13	PE15	PB12
2	PA2	PA3	PA6	PA7	PE13	PE7	PB12	PB14
3	PA3	PB14	PA7	PC4	PE7	PB13	PB14	PB15
4	PB14	PC0	PC4	PC0	PB13	PE8	PB15	PE8
5	PC0	PC1	PC0	PC1	PE8	PD10	PE8	PD10

6	PC1	PC2	PC1	PC2	PD10	PD11	PD10	PD11
7	PC2	PC3	PC2	PC3	PD11	PD12	PD11	PD12
8	PC3	PF0	PC3	PF1	PD12	PD13	PD12	PD13
9	PF0	PB12	PF1	PC5	PD13	PD14	PD13	PD14
10	PB12	PB1	PC5	PB2	PD14	PB0	PD14	PD8
11	PB1	PB0	PB2	PA5	PE11	PE10	PD8	PD9
12	PB0	PB1	PA5	PB11	PE10	PE11	PD9	PE10
13	PB11	PB0	PB11	PB15	PE12	PE11	PE10	PE11
14	PB12	PB11	PA5	PA4	PB1	PB0	PE11	PE12
15	PB11	PB12	PA4	PA5	PB0	PB1	PE12	PE14
16	PC1	PC0	PC1	PC0	PE10	PE9	PA8	PA9
17	PC2	PC1	PC2	PC1	PE9	PE10	PA9	PA8
18	PC3	PC2	PC3	PC2	PD14	PD13	PB15	PD8
19	PA1	PA0	PA1	PA0	PD13	PD12	PD8	PB15
20	PA2	PA1	PC4	PA7	PD12	PD11	PD9	PD8
21	PA3	PA2	PA7	PA6	PD11	PD10	PE15	PE14

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

以右对齐，12 位分辨率为例，

1) 当V_{INn}为V_{REFP}，V_{INm}为V_{REFN}时，通道n的转换结果为0x0FFF；
2) 当V_{INn}为V_{REFN}，V_{INm}为V_{REFP}时，通道n的转换结果为0x0000；
3) 当V_{INn}为$V_{REFP}/2$，V_{INm}为$V_{REFP}/2$时，通道n的转换结果为0x07FF。

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

$$V_{INn}-V_{INm} = V_{REFP}*(2*D_{out}/4095-1) \quad (17\text{-}1)$$

17.4.5. 常规序列

通道管理电路可以将采样通道组织成一个序列：常规序列。常规序列支持最多 22 个通道，每个通道称为常规通道。

ADC_RSQ0~ADC_RSQ8 寄存器规定了常规序列的通道选择。ADC_RSQ0 寄存器的 RL[3:0]位规定了整个常规序列的长度。

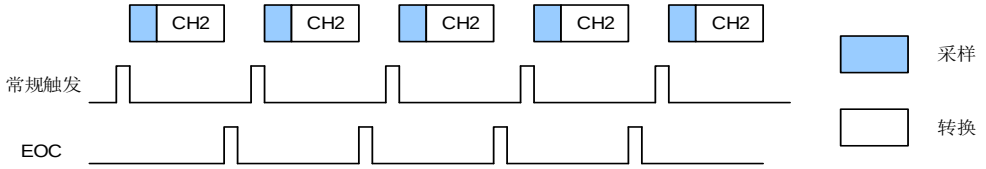
注意：尽管 ADC 支持 22 个通道，但常规序列一次最多转换 16 个通道。

17.4.6. 运行模式

单次运行模式

单次转换模式下，ADC_RSQ8 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 TRIGSEL 触发发生，ADC 就会采样和转换一个通道。

图 17-2. 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。

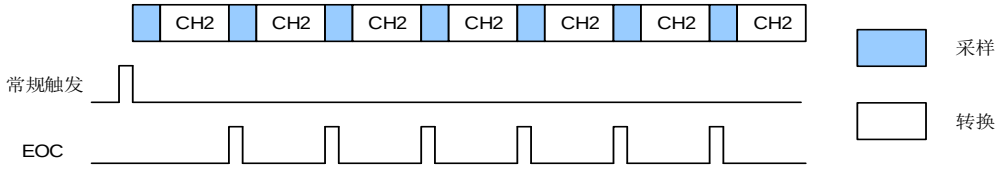
常规序列单次转换模式的软件流程：

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_RSQx寄存器；
4. 如果有需要，可以配置ADC_CTL1寄存器的ETMRC[1:0]位；
5. 设置SWRCST位，或者为常规序列产生一个TRIGSEL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

对 ADC_CTL1 寄存器的 CTN 位置 1 可以使能连续运行模式。在此模式下，ADC 执行由 RSQ0 规定的转换通道。当 ADCON 位被置 1，一旦相应软件触发或者 TRIGSEL 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 17-3. 连续运行模式



常规序列连续运行模式的软件流程：

1. 设置ADC_CTL1寄存器的CTN位为1；
2. 根据模拟通道编号配置RSQ0；
3. 配置ADC_RSQx寄存器；

4. 如果有需要，配置ADC_CTL1寄存器的ETMRC[1:0]位；
5. 设置SWRCST位，或者给常规序列产生一个TRIGSEL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。

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

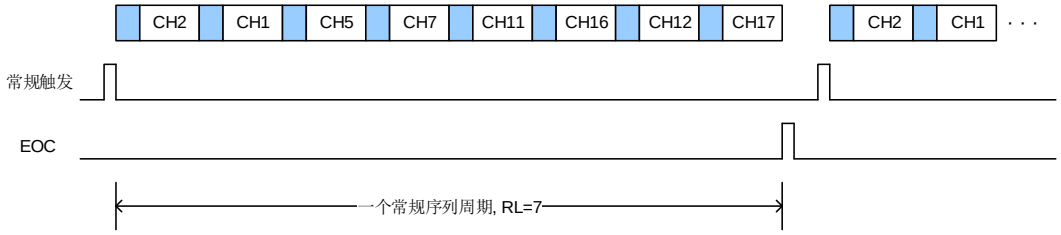
1. 设置ADC_CTL1寄存器的CTN位为1；
2. 根据模拟通道编号配置RSQ0；
3. 配置ADC_RSQx寄存器；
4. 如果有需要，配置ADC_CTL1寄存器的ETMRC[1:0]位；
5. 准备DMA模块，用于传输来自ADC_RDATA的数据；
6. 设置SWRCST位，或者给常规序列产生一个TRIGSEL触发。

扫描运行模式

扫描转换模式可以通过将 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_RSQ1~ADC_RSQ8 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 TRIGSEL 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 被置 1，将产生中断。当常规序列工作在扫描模式下时，ADC_CTL1 寄存器的 DMA 位必须设置为 1。

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

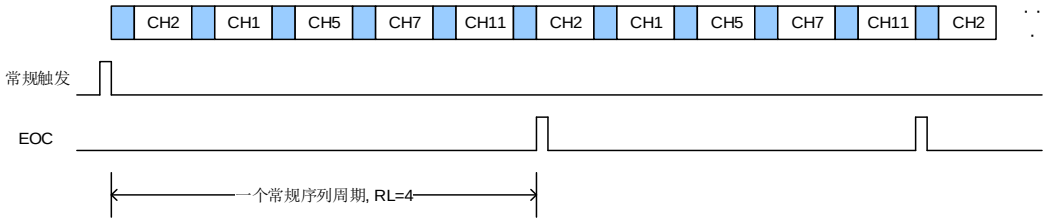
图 17-4. 扫描转换模式，且连续转换模式失能



常规序列扫描转换模式的软件流程：

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 寄存器；
3. 如果有需要，配置 ADC_CTL1 寄存器中的 ETMRC[1:0]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 TRIGSEL 触发；
6. 等待 EOC 标志位置 1；
7. 写 0 清除 EOC 标志位。

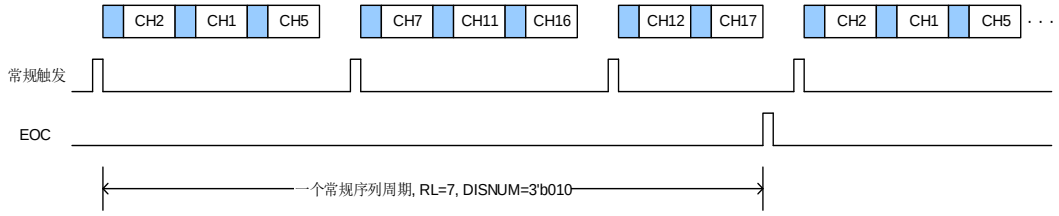
图 17-5. 扫描转换模式，连续转换模式使能



间断模式

ADC_CTL0 寄存器的 DISRC 位置 1 时，常规序列使能间断运行模式。该模式下可以执行一次 n 个通道的短序列转换（n 不超过 8），该序列是 ADC_RSQ0~ADC_RSQ8 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]位配置。当相应的软件触发或 TRIGSEL 触发发生，ADC 就会采样和转换在 ADC_RSQ0~ADC_RSQ8 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 位将被置 1。如果 EOCIE 位被置 1 将产生一个中断。

图 17-6. 间断运行模式



常规序列断模式的软件流程：

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 寄存器；
4. 如果有需要，配置 ADC_CTL1 寄存器中的 ETMR[1:0]位；
5. 准备 DMA 模块，用于传输来自 ADC_RDATA 的数据（参考 DMA 模块）；
6. 设置 SWRCST 位，或者给常规序列产生一个 TRIGSE 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。

17.4.7. 转换结果阈值监测功能

模拟看门狗 0

配置 ADC_CTL0 寄存器的 RWD0EN 位为 1，可使能常规序列的模拟看门狗功能 0。

如果 ADC 的模拟转换电压低于低阈值或高于高阈值时，ADC_STAT 状态寄存器的 WDE0 位将置 1。若 WDE0IE 位置 1，将产生中断。ADC_WDHT0 和 ADC_WDLT0 寄存器用来设定高低阈值。内部数据的比较在对齐之前完成，因此阀值与 ADC_CTL1 寄存器的 DAL 位确定的对齐方式无关。ADC_CTL0 寄存器的 RWD0EN，WD0SC 和 WD0CHSEL[4:0]位可以用来选择模拟看门狗 0 监控单一通道或者多个通道。

模拟看门狗 1/2

模拟看门狗 1/2 更加的灵活，可以进行单个或多个通道的看门狗功能配置。

通过配置 ADC_WD1SR 寄存器中的 AWD1CS[21:0]位域中的相应位，可以使能相应通道的模拟看门狗 1 功能，同理，可以配置看门狗 2 功能。模拟看门狗 1/2 的高/低阈值可在 ADC_WDLT1, ADC_WDHT1, ADC_WDLT2 和 ADC_WDHT2 寄存器中进行配置。

注：如果 OVSEN=1，模拟看门狗 0/1/2 可以将转换的模拟电压（过采样后）与低阈值或高阈值进行比较。如果 OVSEN=0，模拟看门狗 0/1/2 可以将转换的模拟电压（过采样前）与低阈值或高阈值进行比较。

17.4.8. 数据存储模式

ADC_CTL1 寄存器的 DAL 位确定转换后数据存储的对齐方式。

图 17-7. 12 位数据存储模式

常规通道数据

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

DAL=0

常规通道数据

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

DAL=1

6 位分辨率的数据存储模式不同于 12 位/10 位/8 位分辨率数据存储模式，如*图 17-8. 6 位数据存储模式*。

图 17-8. 6 位数据存储模式

常规通道数据

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

DAL=0

常规通道数据

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

DAL=1

注意：ADC_OVSAMPCTL 寄存器中的 OVSEN 置位时，ADC_CTL1 寄存器中的 DAL 位值将被忽略，ADC 仅支持 LSB 对齐。

17.4.9. 采样时间配置

ADC 使用若干个 CK_ADC 周期对输入电压采样，采样周期数目可以通过 ADC_RSQ0~ADC_RSQ8 寄存器的 RSMPn[9:0]位更改。每个序列可以用不同的时间采样。在 12 位分辨率的情况下，总转换时间=采样时间+12.5 个 CK_ADC 周期。

例如：

CK_ADC = 40MHz ，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 0.375us。

17.4.10. 外部触发配置

TRIGSEL 或者 SWRCST 的上升沿可以触发常规序列的转换。常规序列的触发源由 ADC_CTL1 寄存器中的 ETMRC[1:0]位控制。

表 17-4. 常规序列外部触发源

ETMRC[1:0]	触发源	触发类型
01, 10, 11	TRIGSEL	来自TRIGSEL的信号
00	SWRCST	软件触发

17.4.11. DMA 请求

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

17.4.12. 溢出检测

当 DMA 使能的时候，将 ADC_CTL1 寄存器的 EOCM 位置 1 可以使能溢出检测。如果一个常规转换在上一个常规转换数据读出之前已经完成，则会产生一个溢出事件，相应的 ADC_STAT 状态寄存器的 ROVF 标志位会置位。如果 ADC_CTL0 寄存器的 ROVFIE 置位，溢出中断产生。

为了使得 ADC 从 ROVF 溢出状态中恢复过来，建议对 DMA 模块重新进行初始化。内部状态机复位，以保证常规转换数据正确的传输。ADC 转换将会停止，直到 ROVF 位被清零。

ADC 从 ROVF 状态恢复的软件流程如下：

1. 将 ADC_CTL1 寄存器的 DMA 位清 0；
2. 将 ADC_CTL1 寄存器的 ADCON 位清 0；
3. 将 DMA_CHxCTL 寄存器的 CHEN 位清 0，用于重新初始化 DMA 模块；
4. 将 ADC_STAT 寄存器的 ROVF 位清 0；
5. 将 DMA_CHxCTL 寄存器的 CHEN 位置 1；

6. 将 ADC_CTL1 寄存器的 DMA 位置 1；
7. 将 ADC_CTL1 的 ADCON 位置 1；
8. 等待 T（setup）；
9. 通过软件或触发开始 ADC 转换。

17.4.13. ADC 内部通道

将 ADC_CTL1 寄存器的 TSVEN1 位置 1 可以使能温度传感器通道（ADC0_IN14），将 ADC_CTL1 寄存器的 TSVEN2 位置 1 可以使能高精度温度传感器通道（ADC2_IN18）。将 ADC_CTL1 寄存器的 INREFEN 位置 1 可以使能内部电压参考通道（ADC0_IN18 / ADC1_IN18 / ADC2_IN15 / ADC3_IN20）。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 文档）。温度传感器不用时，复位 TSVEN1 和 TSVEN2，可以将其置于掉电模式。

温度传感器（只针对普通温度传感器）的输出电压随温度会发生线性变化，由于芯片生产过程的多样化，温度变化曲线的偏差在芯片间会有不同（最多相差 45°C）。内部温度传感器更适用于检测温度的变化，而不是用于测量绝对温度。如果需要测量精确的温度，应该使用一个外置的温度传感器来校准这个偏移错误。

内部电压参考（V_{REFINT}）提供了一个稳定的（带隙基准）电压输出给 ADC 和比较器。V_{REFINT} 内部连接到 ADC0_IN18、ADC1_IN18、ADC2_IN15、ADC3_IN20 输入通道。

使用温度传感器：

1．配置温度传感器通道（ADC0_IN14）的转换序列和采样时间为 t_{s_temp} us；
2．置位 ADC_CTL1 寄存器中的 TSVEN1 位，使能温度传感器；
3．置位 ADC_CTL1 寄存器的 ADCON 位，或者由外部触发 ADC 转换；
4．从 ADC 数据寄存器中读取并计算温度传感器数据 $V_{temperature}$，并由下面公式计算出实际温度：

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

V_{25}：内部温度传感器在 25° C 下的电压，典型值及出厂校准值地址请参考 datasheet（参考 Temperature sensor characteristics 章节）。

Avg_Slope：温度与内部温度传感器电压曲线的均值斜率，典型值请参考 datasheet（参考 Temperature sensor characteristics 章节）。

使用高精度温度传感器：

1．配置 ADC 时钟（不超过 5MHz）；
2．配置温度传感器通道（ADC2_IN18）的转换序列和采样时间为 t_{s_temp} us；
3．置位 ADC_CTL1 寄存器中的 TSVEN2 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．从 ADC 数据寄存器中读取并计算温度传感器数据 $V_{temperature}$，并由下面公式计算出实际温度：

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

V_{25}：内部温度传感器在 25° C 下的电压，典型值及出厂校准值地址请参考 datasheet（参考 High-precision temperature sensor characteristics 章节）。

Avg_Slope：温度与内部温度传感器电压曲线的均值斜率，典型值请参考 datasheet（参考 High-precision temperature sensor characteristics 章节）。

注意：

1) 当高精度温度传感器使能，至少需要等待 3 个 ADC 采样周期，前三个转换数据应当被舍弃；
2) 可以通过过采样和软件平均提高高精度温度传感器准确度。

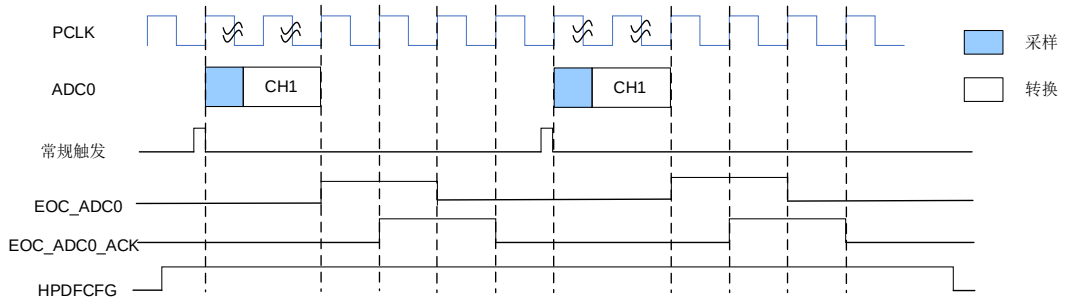
17.4.14. 电池电压检测电路

V_{BAT} 通道可用于测量 V_{BAT} 引脚上的备用电池电压。当 ADC_CTL1 寄存器中的 VBATEN 位置位时，V_{BAT} 通道（ADC0_IN15 / ADC2_IN19）被启用，集成在 V_{BAT} 引脚上的 3 分压桥也被自动启用。由于 V_{BAT} 可能高于 V_{DDA}，此桥用于确保 ADC 正确运行。它将 V_{BAT}/3 连接到 ADC0_IN15 / ADC2_IN19 通道中。因此，转换后的数字值为 V_{BAT}/3。为了防止不必要的电池能耗，建议仅在需要时启用桥接器。

17.4.15. 使用 HPDF 管理转换结果

高性能数字滤波器（HPDF）可用于管理 ADC 转换结果。在这种情况下，HPDFCFG 位必须置 1，DMA 位 SYNCDMA[1:0]位域和 SYNCDDM 位必须清除为 0。如果 DMA 和 HPDF 并行工作，只有 DMA 生效。ADC 将常规数据寄存器数据的 16 个最低有效位传输到 HPDF，一旦传输完成，HPDF 将重置 EOC 标志。如*图 17-9. HFDF 与 ADC 模块握手信号示意图*所示。

图 17-9. HFDF 与 ADC 模块握手信号示意图



17.4.16. 可编程分辨率（DRES）

对寄存器 ADC_CTL0 中的 DRES[1:0]位进行编程即可配置分辨率为 6、8、10 及 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转换时间。如*表 17-5. ADC 不同分辨率对应的 tCONV 时间*所示，较低的分辨率能够减少逐次逼近步骤所需的转换时间 t_{ADC}。

表 17-5. ADC 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}=40MHz	t_{SMPL}(min) (ADC clock cycles)	t_{ADC} (ADC clock cycles)	t_{ADC}(ns) at f_{ADC}=40MHz
12	12.5	312.5ns	2.5	15	375 ns
10	10.5	262.5 ns	2.5	13	325 ns
8	8.5	212.5ns	2.5	11	275 ns
6	6.5	162.5 ns	2.5	9	225 ns

17.4.17. 片上硬件过采样

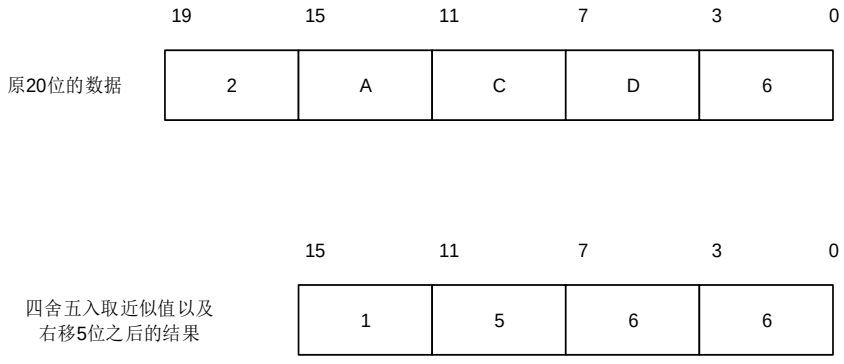
片上硬件过采样单元执行数据预处理以减轻 CPU 负担。它能够处理多个转换，并将多个转换的结果取平均，增加数据宽度，最高可达 32 位。其结果值根据如下公式计算得出，其中 N 和 M 的值可以被调整，过采样单元可以通过设置 ADC_OVSAMPCTL 寄存器的 OVSEN 位来使能，它是以降低数据输出率为代价，换取较高的数据分辨率。$D_{out}(n)$是指 ADC 输出的第 n 个数字信号：

$$Result=\frac{1}{M}*\sum_{n=0}^{N-1}D_{out}(n) \quad (17\text{-}2)$$

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[9:0]位定义，它的取值范围为 2x 到 1024x。除法系数 M 定义一个多达 11 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 22 位（1024*12 位）的值。首先进行右移，最终值传入对应的数据寄存器中。

图 17-10. 12 位 ADC 右移 5 位和取整的数例



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：

$$N \times t_{ADC} = N \times (t_{SMPL} + t_{CONV}) \quad (17\text{-}3)$$

17.4.18. 增益模式

当 GAINEN 位在 ADC_CTL1 寄存器中置位时，对所有 ADC 转换的数据进行增益校准。每次转换后，使用下面的公式计算数据:

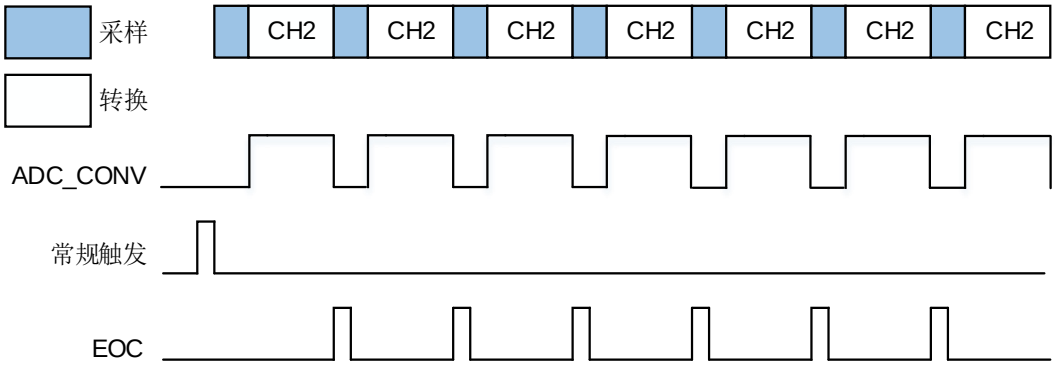
$$DATA(\text{new ADC result}) = DATA(\text{original ADC result}) \times (GAIN/4096) \quad (17\text{-}4)$$

可编程 GAIN 位域范围为 0 ~ 16383，实际增益因子(GAIN / 4096)为 0 ~ 3.999756。

17.4.19. ADC 转换信号

ADC转换信号ADC_CONV在ADC通道转换期间保持高电平状态，而在其他时间段都为低电平。ADC_CONV信号可内部通过TRIGSEL连接到CLAx模块，作为CLAx多路选择器的输入。

图 17-11. 常规序列连续转换模式下的 ADC 转换信号



17.5. ADC 同步模式

在具有多个 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 和 ADC2 的转换。根据 ADC_SYNCCTL 寄存器的 SYNCM[4:0]位来选择两个或三个 ADC 按并行模式还是交替模式进行转换。

在 ADC 同步模式中，当转换配置成外部事件触发时，ADC1 和 ADC2 的外部触发必须失能。常规通道的转换结果存储在 ADC 同步常规数据寄存器(ADC_SYNCDATA)中。

ADC 同步模式如*表 17-6. ADC 同步模式表*所示。

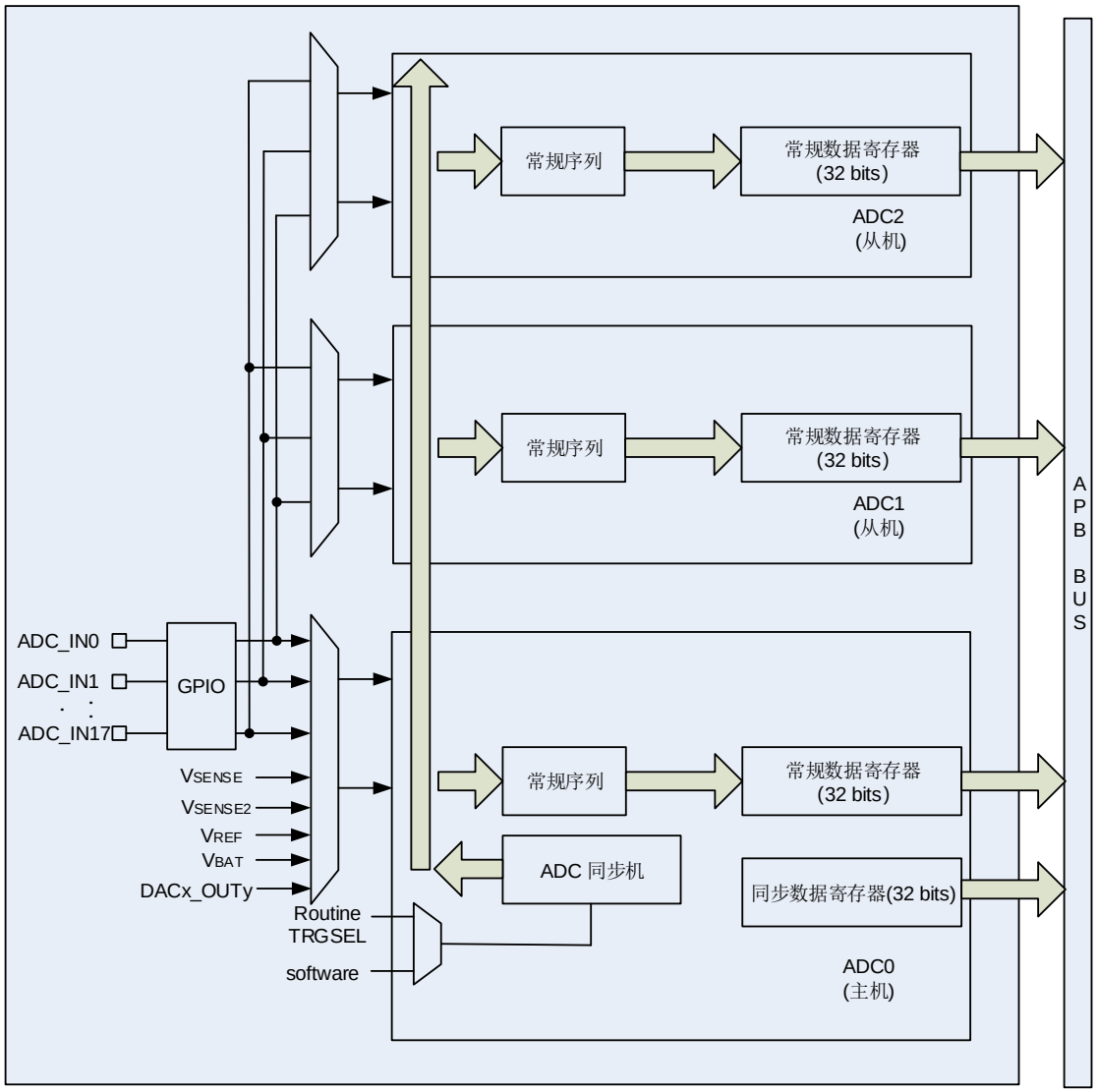
表 17-6. ADC 同步模式表

SYNCM[4: 0]	mode
00000	独立模式。所有的ADC都独立工作。
00110	ADC0和ADC1工作在常规并行模式。ADC2 独立工作。
00111	ADC0和ADC1工作在常规跟随模式。ADC2 独立工作。
10110	ADC0、ADC1和ADC2工作在常规并行模式。
10111	ADC0、ADC1和ADC2工作在常规跟随模式。

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 17-12. ADC 同步框图*所示。

图 17-12. ADC 同步框图



17.5.1. 独立模式

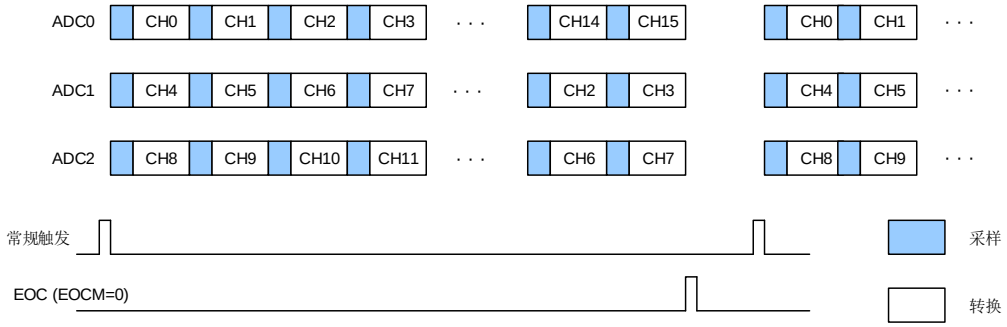
在这种模式下，ADC 同步是忽略的，每个 ADC 都独立工作。

17.5.2. 常规并行模式

设置 ADC_SYNCCTL 寄存器的 SYNCM[4:0]位为 00110 或 10110，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 ETMRC[1:0]位进行配置。

根据 ADC_CTL1 寄存器中的 EOCM 位的设置，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 17-13. 基于 16 个通道的常规并行模式*所示。

图 17-13. 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。
3. 如果 SYNCM=5’b 00110，ADC2 工作在独立模式。

17.5.3. 常规跟随模式

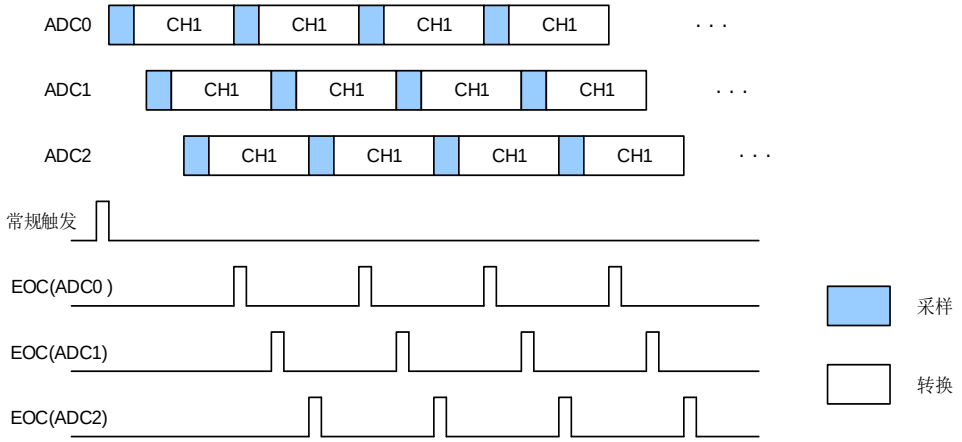
设置 ADC_SYNCCTL 寄存器的 SYNCM[4:0]位为 5’b 00111 或 5’b 10111，使能常规跟随模式。在常规跟随模式中，根据选择的外部触发，ADC0 开始转换常规序列。外部触发选择由 ADC0 的 ADC_CTL1 寄存器 ETMRC[1:0]位进行配置。经过一定的延迟之后，ADC1 开始转换常规序列，再经过另一个延迟之后，ADC2 开始转换常规序列。以上描述中提到的常规序列只能包含一个通道。

在两个连续采样阶段之间的延迟时间，由 ADC_SYNCCTL 寄存器的 SYNCDLY[3:0]位进行配置。如果 SYNCDLY[3:0]位配置的延迟时间比采样时间还短，为了避免在一个给定时间，多个 ADC 对同一个通道进行采样，会将(采样时间 + 2) CK_ADC 周期作为实际的延迟时间。

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。根据 ADC_CTL1 寄存器的 EOCM 的配置，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 17-14. 一个采用连续转换模式通道上的常规跟随模式*所示。

图 17-14. 一个采用连续转换模式通道上的常规跟随模式



注意：

1. 确保在没有任何一个 ADC 在进行转换的时候才触发 ADC（当有某些转换还没完成时，不触发 ADC0）；
2. 如果 SYNCM=5’b 00111，ADC2 工作在独立模式。

17.5.4. 在 ADC 同步模式中使用 DMA

在 ADC 同步模式中，常规序列通道转换的数据存储在 ADC 同步常规数据寄存器(ADC_SYNCDATA)中，DMA 可以用来传输 ADC_SYNCDATA 寄存器的数据。有以下两种 DMA 工作模式，可以和各种 ADC 同步模式很好地配合使用。

ADC 同步 DMA 模式 0

在 ADC 同步 DMA 模式 0 中，DMA 传输的位宽为 32。一次 DMA 请求传输一个数据，这个数据轮流的从各 ADC 的常规转换结果中取出。对于每次 DMA 请求，DMA 通道的源地址固定为 ADC_SYNCDATA 寄存器，而这个寄存器的内容会变成 DMA 要被传输的数值。当 ADC0 和 ADC1 工作在同步模式时，DMA 的传输序列为：ADC0_RDATA[31:0] -> ADC1_RDATA[31:0] -> ADC0_RDATA[31:0] -> ADC1_RDATA[31:0]。当所有的 ADC 都工作在同步模式时，DMA 的传输序列为：ADC0_RDATA[31:0] -> ADC1_RDATA[31:0] -> ADC2_RDATA[31:0] -> ADC0_RDATA[31:0] -> ADC1_RDATA[31:0] -> ADC2_RDATA[31:0]。

ADC 同步 DMA 模式 0 适用于：

- ADC0 和 ADC1 工作在常规并行模式(SYNCM=00110)；
- 所有的 ADC 工作在常规并行模式(SYNCM=5’b 10110)。

ADC 同步 DMA 模式 1

在 ADC 同步 DMA 模式 1 中，DMA 传输的位宽为 32。一次 DMA 请求传输两个数据，这些数据轮流的从各 ADC 的常规通道转换结果中取出。对于每次 DMA 请求，DMA 通道的源地址固定为 ADC_SYNCDATA 寄存器，而这个寄存器的内容会变成 DMA 要被传输的数值。当 ADC0 和 ADC1 工作在同步模式时，DMA 的数据每次都为：{ADC1_RDATA[15:0], ADC0_RDATA[15:0]}。当所有的 ADC 都工作在同步模式时，DMA 的传输序列为：{ADC1_RDATA[15:0],ADC0_RDATA[15:0]} -> {ADC0_RDATA[15:0],ADC2_RDATA[15:0]} -> {ADC2_RDATA[15:0],ADC1_RDATA[15:0]} -> {ADC1_RDATA[15:0],ADC0_RDATA[15:0]}。

ADC 同步 DMA 模式 1 适用于：

- ADC0 和 ADC1 工作在常规并行模式(SYNCM=5’b 00110)；
- ADC0 和 ADC1 工作在常规跟随模式(SYNCM=5’b 00111)；
- 所有的 ADC 工作在常规跟随模式(SYNCM=5’b 10111)。

17.6. 中断

以下任一个事件发生都可以产生中断：

- 常规序列转换结束；
- 模拟看门狗事件；
- 溢出事件。

ADC0 和 ADC1 都被映射到同一个中断向量 IRQ18，ADC2 被映射到同一个中断向量 IRQ47，ADC3 都被映射到同一个中断向量 IRQ61。

17.7. ADC 寄存器

ADC0 基地址：0x5000 0000

ADC1 基地址:：0x5000 0400

ADC2 基地址:：0x5000 0800

ADC3 基地址:：0x5000 0C00

17.7.1. 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
WDE2	WDE1	保留													
rc_w0	rc_w0														

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

位/位域	名称	描述
31	WDE2	模拟看门狗 2 事件标志 0：没有模拟看门狗 2 事件 1：产生模拟看门狗 2 事件 转换电压超过 ADC_WDLT2 和 ADC_WDHT2 寄存器设定的阈值时由硬件置 1，软件写 0 清除。
30	WDE1	模拟看门狗 1 事件标志 0：没有模拟看门狗 1 事件 1：产生模拟看门狗 1 事件 转换电压超过 ADC_WDLT1 和 ADC_WDHT1 寄存器设定的阈值时由硬件置 1，软件写 0 清除。
29:6	保留	必须保持复位值。
5	ROVF	常规数据寄存器溢出 0：常规数据寄存器没有溢出 1：常规数据寄存器溢出 在单次或多次模式中，当常规数据寄存器溢出时，该位由硬件置位。只有在 DMA 使能或者转换结束模式被置 1（EOCM=1）时，这个标志位才会置位。如果出现

		ROVF 置位，则最后的常规数据会被丢失。 软件写‘0”清除。
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	序列转换结束标志 0：序列转换没有结束 1：序列转换结束 常规序列转换结束时硬件置位。 软件写 0 或读 ADC_RDATA 寄存器清除。
0	WDE0	模拟看门狗 0 事件标志 0： 没有模拟看门狗 0 事件 1： 产生模拟看门狗 0 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。

17.7.2. 控制寄存器 0（ADC_CTL0）

地址偏移：0x04

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WDE2IE	WDE1IE	保留			ROVFIE	DRES[1:0]		RWD0EN	保留						
rw	rw				rw	rw		rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISNUM[2:0]			保留	DISRC	保留	WD0SC	SM	保留	WDE0IE	EOCIE	WD0CHSEL[4:0]				
rw				rw		rw	rw		rw	rw	rw				

位/位域	名称	描述
31	WDE2IE	WDE2 中断使能 0：WDE2 中断禁止 1：WDE2 中断使能
30	WDE1IE	WDE1 中断使能 0：WDE1 中断禁止

		1：WDE1 中断使能
29:27	保留	必须保持复位值。
26	ROVFIE	ROVF 中断使能 0：ROVF 中断失能 1：ROVF 中断使能
25:24	DRES[1:0]	ADC 数据分辨率 00:12 位 01:10 位 10:8 位 11:6 位
23	RWD0EN	常规序列看门狗 0 使能 0：常规序列看门狗 0 禁止 1：常规序列看门狗 0 使能
22:16	保留	必须保持复位值。
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：间断运行模式禁止 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：扫描运行模式禁止 1：扫描运行模式使能
7	保留	必须保持复位值。
6	WDE0IE	WDE0 中断使能 0：中断禁止 1：中断使能
5	EOCIE	EOC 中断使能 0：中断禁止

		1：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 10010:ADC 通道 18 10000:ADC 通道 19 10001:ADC 通道 20 10010:ADC 通道 21 其他值保留。 **注意：**ADC0 模拟输入通道 14、通道 15、通道 16、通道 17 和通道 18 内部连接至温度传感器、电池、DAC0_OUT0、DAC0_OUT1 和 V_{REFINT} 输入。 ADC1 模拟输入通道 16、通道 17 和通道 18 内部连接至 DAC1_OUT0、DAC1_OUT1 和 V_{REFINT} 输入。 ADC2 模拟输入通道 15、通道 16、通道 17、通道 18 和通道 19 内部连接至 V_{REFINT} 输入、DAC2_OUT0、DAC2_OUT1、高精度温度传感器和电池。 ADC3 模拟输入通道 18、通道 19 和通道 20 内部连接至 DAC3_OUT0 和 DAC3_OUT1 和 V_{REFINT} 输入。

17.7.3. 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
TSVEN2	SWRCST	ETMRC[1:0]		CALMOD	保留	VBATEN	INREFEN	TSVEN1	保留						GAINEN
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			HPDFCFG	DAL	EOCM	DDM	DMA	保留	CALNUM[2:0]			RSTCLB	CLB	CTN	ADCON
			rw	rw	rw	rw	rw		rw			rw	rw	rw	rw

位/位域	名称	描述
31	TSVEN2	在 ADC2 中，该位可由软件置位或清零。 ADC2 通道 18（高精度温度传感器）使能。 0：高精度温度传感器通道失能 1：高精度温度传感器通道使能
30	SWRCST	常规序列软件启动转换 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，立刻由硬件清零。
29:28	ETMRC[1:0]	常规序列外部触发模式 00：常规序列外部触发失能 01：常规序列外部触发上升沿使能 10：常规序列外部触发下降沿使能 11：常规序列外部触发双边沿使能
27	CALMOD	ADC 校准模式 0：偏移、不匹配模式 1：偏移模式
26	保留	必须保持复位值。
25	VBATEN	在 ADC0 和 ADC2 中，该位可由软件置位或清零。 使能 ADC0 的通道 15（外部电池电压的 1/3） 使能 ADC2 的通道 19（外部电池电压的 1/3） 0：外部电池电压的 1/3 失能 1：外部电池电压的 1/3 使能
24	INREFEN	在 ADC0、ADC1、ADC2 和 ADC3 中，该位可由软件置位或清零。 使能 ADC0 的通道 18（内部参考电压） 使能 ADC1 的通道 18（内部参考电压） 使能 ADC2 的通道 15（内部参考电压） 使能 ADC3 的通道 20（内部参考电压） 0：内部参考电压失能 1：内部参考电压使能

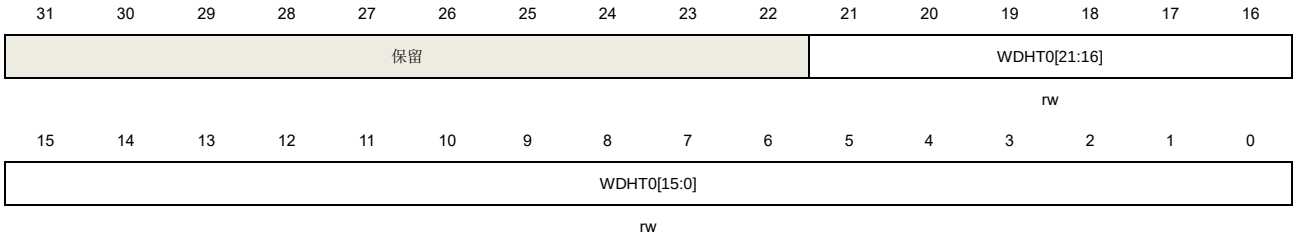
23	TSVEN1	在 ADC0 中，该位可由软件置位或清零。 ADC0 通道 14（温度传感器）使能。 0：温度传感器通道失能 1：温度传感器通道使能
22:17	保留	必须保持复位值。
16	GAINEN	增益模式使能 该位由软件置位或清零。 0：关闭增益模式 1：启用增益模式，对所有通道生效 **注意：**软件允许写这个位必须确保没有正在进行的 ADC 转换。
15:13	保留	必须保持复位值。
12	HPDFCFG	HPDF模式配置 该位由软件置位或清零，使能或使能HPDF模式。仅在DMA=0时有效。 0：HPDF模式失能 1：HPDF 模式使能
11	DAL	数据对齐 0：最低有效位对齐 1：最高有效位对齐
10	EOCM	转换结束模式 0：只有在常规转换序列转换结束时，才将 EOC 置 1。如果不设置 DMA=1，则溢出检测失能。 1：在每个常规通道转换结束时，将 EOC 置 1。溢出检测自动使能。
9	DDM	DMA 失能模式 该位用于在单次 ADC 模式下配置 DMA 失能。 0：DMA 机制在 DMA 控制器的传输结束信号之后失能。 1：当 DMA=1，在每个常规通道转换结束时 DMA 机制产生一个 DMA 请求。
8	DMA	DMA 请求使能 0：DMA 请求禁止 1：DMA 请求使能
7	保留	必须保持复位值。
6:4	CALNUM[2:0]	校准次数 这些位定义了 ADC 的校准次数。 000：1 次 001：2 次 010：4 次

		011：8 次 100：16 次 其它：保留。
3	RSTCLB	校准复位 在校准寄存器初始化后该位可以软件置位和硬件清零。 0：校准寄存器初始化结束. 1：校准寄存器初始化开始
2	CLB	ADC 校准 0：校准结束 1：校准开始
1	CTN	连续模式 0：禁止连续运行模式 1：使能连续运行模式
0	ADCON	开启 ADC。该位从 0 变成 1 将唤醒 ADC。为了省电，当该位为 0 时，模拟子模块将会进入掉电模式。 0：失能 ADC，并进入掉电模式 1：使能 ADC

17.7.4. 看门狗高阈值寄存器 0（ADC_WDHT0）

地址偏移：0x1C
复位值：0x003F FFFF

该寄存器只能按字（32 位）访问。

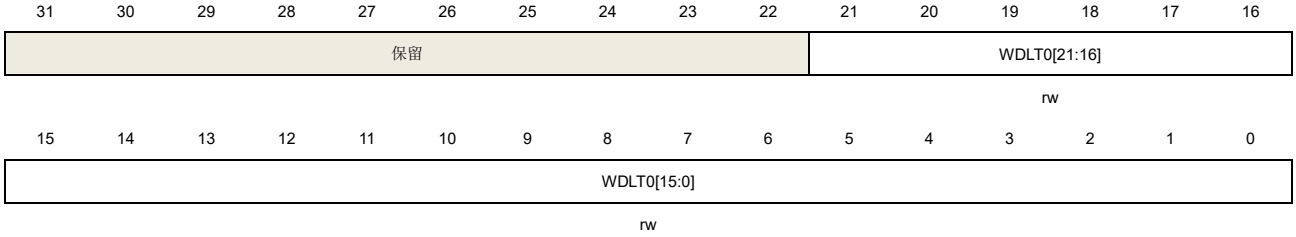


位/位域	名称	描述
31:22	保留	必须保持复位值。
21:0	WDHT0[21:0]	模拟看门狗 0 高侧阈值。 这些位定义了模拟看门狗 0 的高侧阈值。

17.7.5. 看门狗低阈值寄存器 0（ADC_WDLT0）

地址偏移：0x20
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

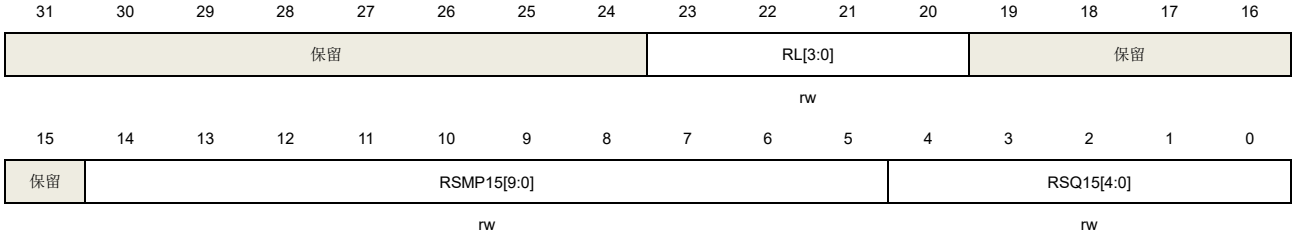


位/位域	名称	描述
31:22	保留	必须保持复位值。
21:0	WDLT0[21:0]	模拟看门狗 0 低侧阈值 这些位定义了模拟看门狗 0 的低侧阈值。

17.7.6. 常规序列寄存器 0（ADC_RSQ0）

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:24	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	保留	必须保持复位值。
14:5	RSMP15[9:0]	常规通道采样时间 参考 RSMP0[9:0]描述。

4:0	RSQ15[4:0]	参考 RSQ0[4:0]描述

17.7.7. 常规序列寄存器 1（ADC_RSQ1）

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	RSMP14[9:0]										RSQ14[4:0]				
	rw										rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	RSMP13[9:0]										RSQ13[4:0]				
	rw										rw				

位/位域	名称	描述
31	保留	必须保持复位值。
30:21	RSMP14[9:0]	常规通道采样时间 参考 RSMP0[9:0]描述。
20:16	RSQ14[4:0]	参考 RSQ0[4:0]描述
15	保留	必须保持复位值。
14:5	RSMP13[9:0]	常规通道采样时间 参考 RSMP0[9:0]描述。
4:0	RSQ13[4:0]	参考 RSQ0[4:0]描述

17.7.8. 常规序列寄存器 2（ADC_RSQ2）

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	RSMP12[9:0]										RSQ12[4:0]				
	rw										rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	RSMP11[9:0]										RSQ11[4:0]				
	rw										rw				

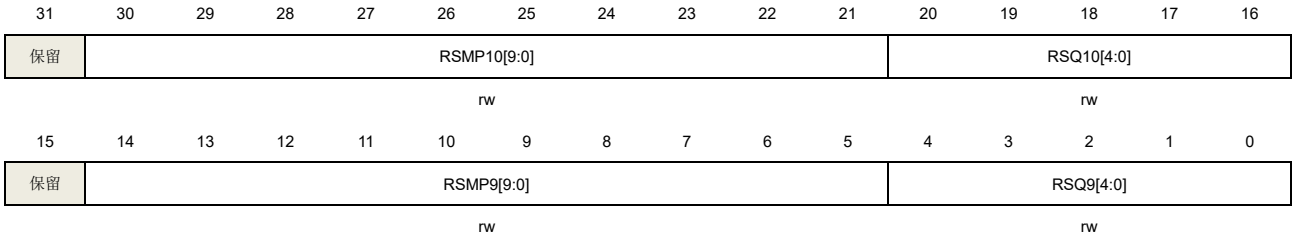
位/位域	名称	描述
31	保留	必须保持复位值。
30:21	RSMP12[9:0]	常规通道采样时间 参考 RSMP0[9:0]描述。
20:16	RSQ12[4:0]	参考 RSQ0[4:0]描述
15	保留	必须保持复位值。
14:5	RSMP11[9:0]	常规通道采样时间 参考 RSMP0[9:0]描述。
4:0	RSQ11[4:0]	参考 RSQ0[4:0]描述

17.7.9. 常规序列寄存器 3（ADC_RSQ3）

地址偏移：0x30

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31	保留	必须保持复位值。
30:21	RSMP10[9:0]	常规通道采样时间 参考 RSMP0[9:0]描述。
20:16	RSQ10[4:0]	参考 RSQ0[4:0]描述
15	保留	必须保持复位值。
14:5	RSMP9[9:0]	常规通道采样时间 参考 RSMP0[9:0]描述。
4:0	RSQ9[4:0]	参考 RSQ0[4:0]描述

17.7.10. 常规序列寄存器 4（ADC_RSQ4）

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	RSMP8[9:0]										RSQ8[4:0]				
	rw										rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	RSMP7[9:0]										RSQ7[4:0]				
	rw										rw				

位/位域	名称	描述
31	保留	必须保持复位值。
30:21	RSMP8[9:0]	常规通道采样时间 参考 RSMP0[9:0]描述。
20:16	RSQ8[4:0]	参考 RSQ0[4:0]描述
15	保留	必须保持复位值。
14:5	RSMP7[9:0]	常规通道采样时间 参考 RSMP0[9:0]描述。
4:0	RSQ7[4:0]	参考 RSQ0[4:0]描述

17.7.11. 常规序列寄存器 5（ADC_RSQ5）

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	RSMP6[9:0]										RSQ6[4:0]				
	rw										rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	RSMP5[9:0]										RSQ5[4:0]				
	rw										rw				

位/位域	名称	描述
31	保留	必须保持复位值。

30:21	RSMP6[9:0]	常规通道采样时间 参考 RSMP0[9:0]描述。
20:16	RSQ6[4:0]	参考 RSQ0[4:0]描述
15	保留	必须保持复位值。
14:5	RSMP5[9:0]	常规通道采样时间 参考 RSMP0[9:0]描述。
4:0	RSQ5[4:0]	参考 RSQ0[4:0]描述

17.7.12. 常规序列寄存器 6（ADC_RSQ6）

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	RSMP4[9:0]										RSQ4[4:0]				
	rw										rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	RSMP3[9:0]										RSQ3[4:0]				
	rw										rw				

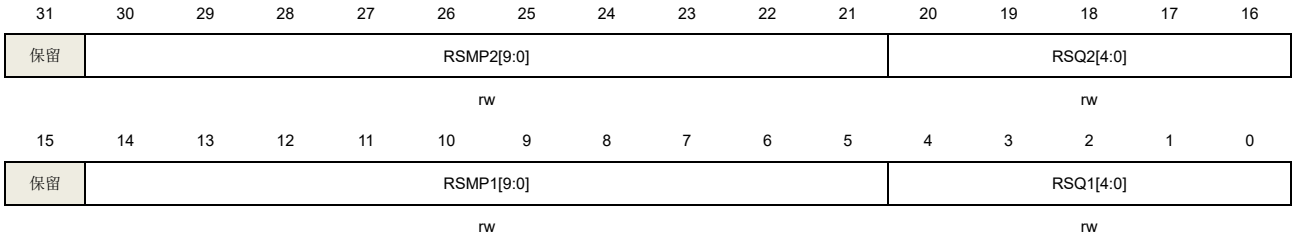
位/位域	名称	描述
31	保留	必须保持复位值。
30:21	RSMP4[9:0]	常规通道采样时间 参考 RSMP0[9:0]描述。
20:16	RSQ4[4:0]	参考 RSQ0[4:0]描述
15	保留	必须保持复位值。
14:5	RSMP3[9:0]	常规通道采样时间 参考 RSMP0[9:0]描述。
4:0	RSQ3[4:0]	参考 RSQ0[4:0]描述

17.7.13. 常规序列寄存器 7（ADC_RSQ7）

地址偏移：0x40

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



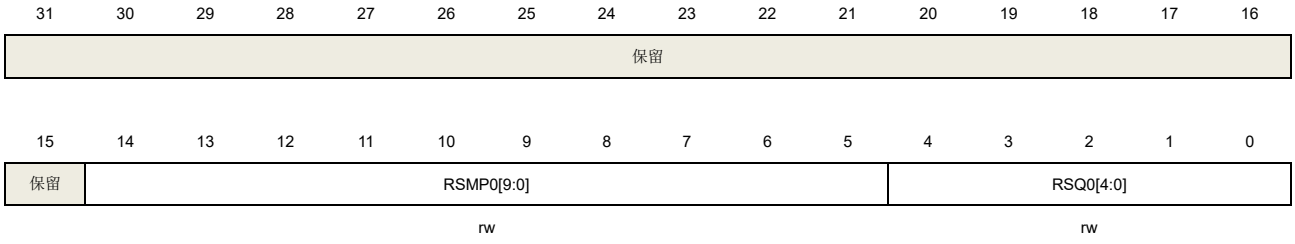
位/位域	名称	描述
31	保留	必须保持复位值。
30:21	RSMP2[9:0]	常规通道采样时间 参考 RSMP0[9:0]描述。
20:16	RSQ2[4:0]	参考 RSQ0[4:0]描述
15	保留	必须保持复位值。
14:5	RSMP1[9:0]	常规通道采样时间 参考 RSMP0[9:0]描述。
4:0	RSQ1[4:0]	参考 RSQ0[4:0]描述

17.7.14. 常规序列寄存器 8（ADC_RSQ8）

地址偏移：0x44

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:15	保留	必须保持复位值。
14:5	RSMP0[9:0]	常规通道采样时间 ADC 采样时间为 RSMP0[9:0] + 2.5 周期。例如，RSMP0[9:0] = 100 时，采样时间为 102.5 周期。 RSMP0[9:0]最大为 638，其余位保留。

注意：如果 RSMPx[9:0]>638，则 ADC 的采样行为将无法保证。

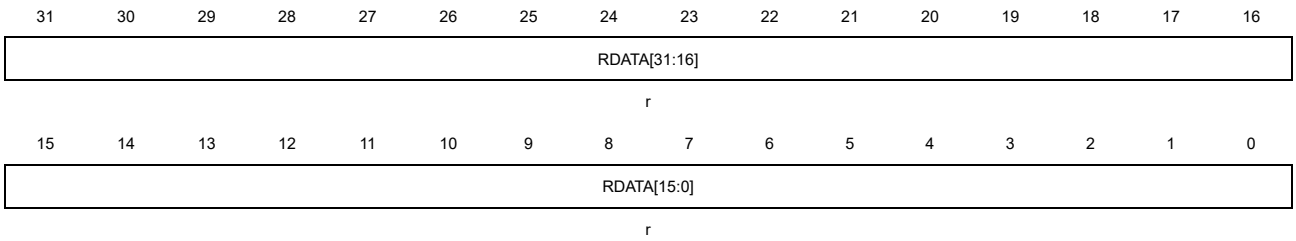
4:0	RSQ0[4:0]	通道编号（0..21）写入这些位来选择常规通道的第 n 个转换的通道。

17.7.15. 常规数据寄存器（ADC_RDATA）

地址偏移：0x64

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



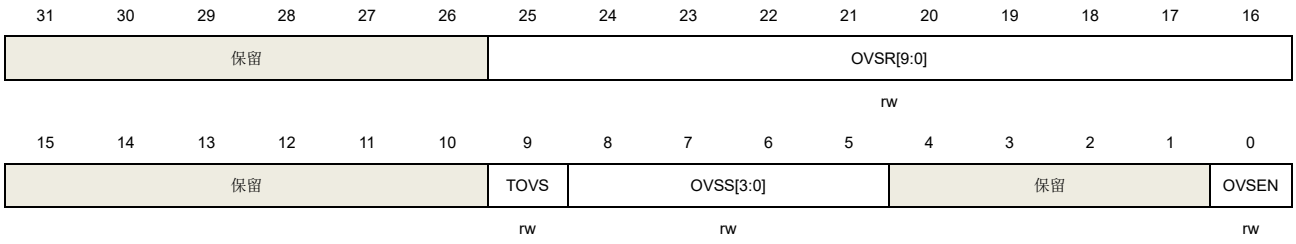
位/位域	名称	描述
31:0	RDATA[31:0]	常规通道数据 这些位包含了常规通道的转换结果，只读。

17.7.16. 过采样控制寄存器（ADC_OVSAMPCTL）

地址偏移：0x80

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



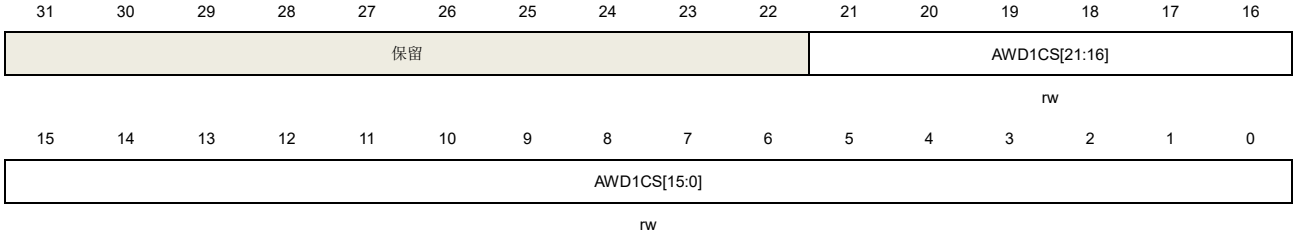
位/位域	名称	描述
31:26	保留	必须保持复位值。
25:16	OVSR[9:0]	过采样率 这些位定义了过采样率的大小 10’d0：1x（无过采样） 10’d1：2x

		10’d2：3x …… 10’d1023：1024x **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）。
15:10	保留	必须保持复位值。
9	TOVS	过采样触发 该位通过软件置位和清除。 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[9:0]）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：移 1 位 0010：移 2 位 0011：移 3 位 0100：移 4 位 0101：移 5 位 0110：移 6 位 0111：移 7 位 1000：移 8 位 1001：移 9 位 1010：移 10 位 1011：移 11 位 其余位保留 **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）。
4:1	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：过采样失能 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）。

17.7.17. 看门狗 1 通道选择寄存器（ADC_WD1SR）

地址偏移：0xA0
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

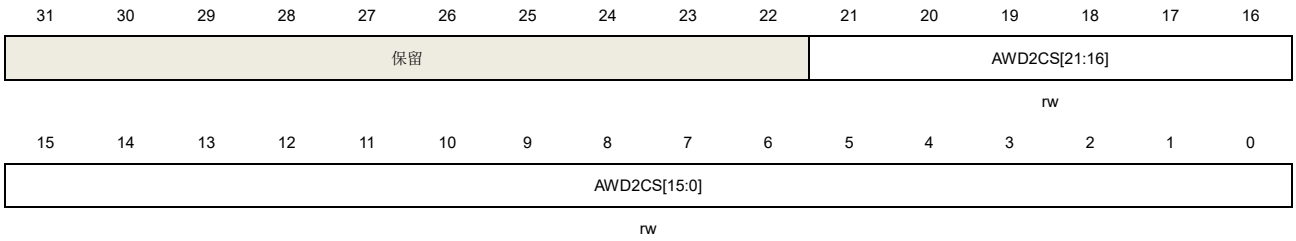


位/位域	名称	描述
31:22	保留	必须保持复位值。
21:0	AWD1CS[21:0]	模拟看门狗1通道选择 这些位由软件置位和复位，它们使能并选择要由模拟看门狗1保护的输入通道。 AWD1CS[n] = 0：ADC模拟输入通道n不由模拟看门狗1保护。 AWD1CS[n] = 1：ADC模拟输入通道n由模拟看门狗1保护。 当AWD1CH[21:0] = 000..0，模拟看门狗1禁能。 **注意：** 1）通过AWD1CS位域配置的模拟看门狗1功能的通道，必须是ADC_RSQn寄存器中配置的通道； 2）只有在ADC禁能（ADCON=0）时，才能软件写这些位。

17.7.18. 看门狗 2 通道选择寄存器（ADC_WD2SR）

地址偏移：0xA4
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:22	保留	必须保持复位值。

21:0	AWD2CS[21:0]	模拟看门狗2通道选择 这些位由软件置位和复位，它们使能并选择要由模拟看门狗2保护的输入通道。 AWD2CS[n] = 0：ADC模拟输入通道n不由模拟看门狗2保护。 AWD2CS[n] = 1：ADC模拟输入通道n由模拟看门狗2保护。 当AWD2CH[21:0] = 000..0，模拟看门狗2禁能。 **注意：** 1) 通过AWD2CS位域配置模拟看门狗2功能的通道，必须是ADC_RSQn寄存器中配置的通道； 2) 只有在ADC禁能（ADCON=0）时，才能软件写这些位。

17.7.19. 看门狗高阈值寄存器 1（ADC_WDHT1）

地址偏移：0xA8

复位值：0x003F FFFF

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留										WDHT1[21:16]					
										rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDHT1[15:0]															
rw															

位/位域	名称	描述
31:22	保留	必须保持复位值。
21:0	WDHT1[21:0]	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。

17.7.20. 看门狗低阈值寄存器 1（ADC_WDLT1）

地址偏移：0xAC

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留										WDLT1[21:16]					
										rw					

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

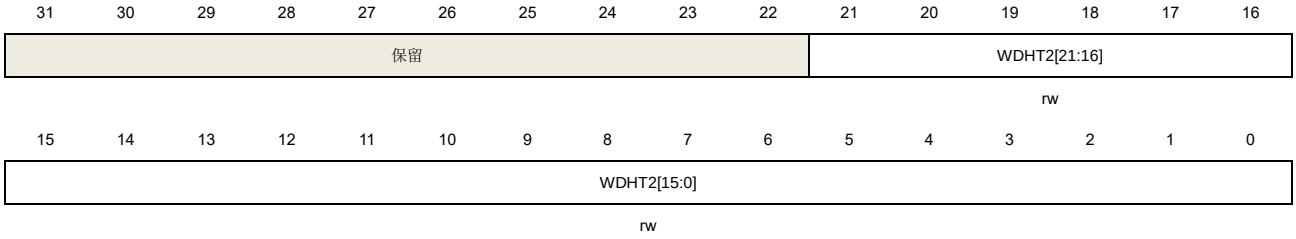
位/位域	名称	描述
31:22	保留	必须保持复位值。
21:0	WDLT1[21:0]	模拟看门狗 1 低侧阈值 这些位定义了模拟看门狗 0 的低侧阈值。

17.7.21. 看门狗高阈值寄存器 2（ADC_WDHT2）

地址偏移：0xB0

复位值：0x003F FFFF

该寄存器只能按字（32 位）访问。



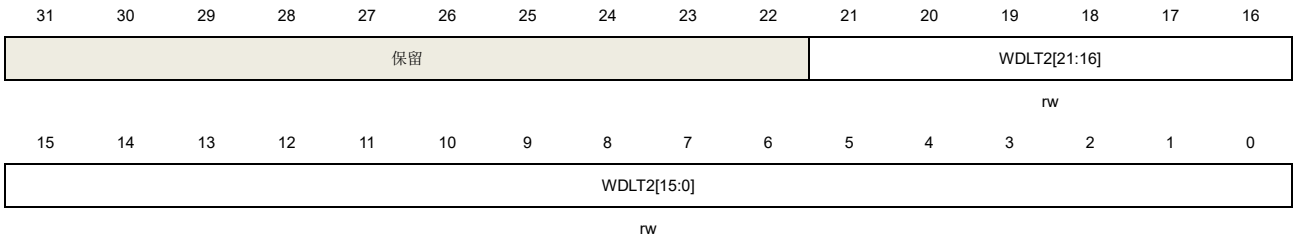
位/位域	名称	描述
31:22	保留	必须保持复位值。
21:0	WDHT2[21:0]	模拟看门狗 2 高侧阈值。 这些位定义了模拟看门狗 2 的高侧阈值。

17.7.22. 看门狗低阈值寄存器 2（ADC_WDLT2）

地址偏移：0xB4

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:22	保留	必须保持复位值。

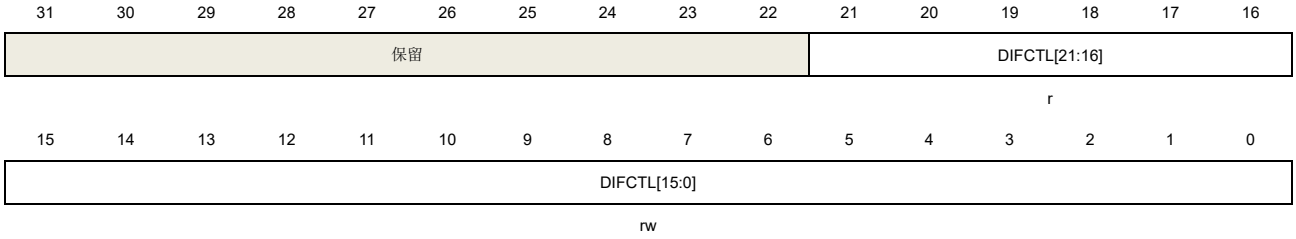
21:0	WDLT2[21:0]	模拟看门狗 2 低侧阈值 这些位定义了模拟看门狗 2 的低侧阈值。

17.7.23. 差分模式控制寄存器（ADC_DIFCTL）

地址偏移：0xB8

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



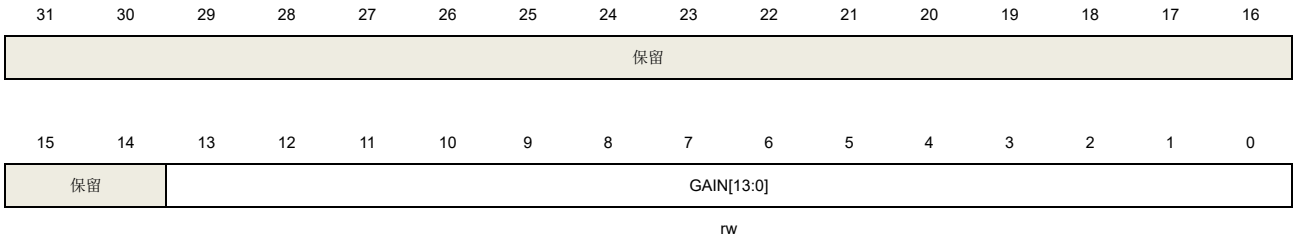
位/位域	名称	描述
31:22	保留	必须保持复位值。
21:0	DIFCTL[21:0]	差分模式通道21..0。 这些位用于配置通道用于单端输入模式还是差分模式。 DIFCTL[n] = 0：ADC模拟输入通道n配置为单端模式。 DIFCTL[n] = 1：ADC模拟输入通道n配置为差分模式。 **注意：**只有在ADC禁能（ADCON=0）时，才能软件写这些位。

17.7.24. 增益配置寄存器（ADC_GAINCFG）

地址偏移：0xC0

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	说明
31:14	保留	必须保持复位值。

13:0	GAIN[13:0]	增益系数 该位域可以通过软件设置或清除。 增益可编程范围为 0～16383，实际增益系数计算（GAIN / 4096）为 0～3.999756。 00 0000 0000 0000：增益为 0，（GAIN / 4096）取值为 0。 … 01 0000 0000 0000：增益为 4096，（GAIN / 4096）取值为 1。 … 10 0000 0000 0000：增益为 8192，（GAIN / 4096）取值为 2。 … 11 0000 0000 0000：增益为 12288，（GAIN / 4096）取值为 3。 … 11 1111 1111 1111：增益因子 3.999756 **注意：**ADC_CTL1 寄存器的 GAINEN 位必须设置为 1，增益系数生效。

17.7.25. 摘要状态寄存器（ADC_SSTAT）

地址偏移：0x300
复位值：0x0000 0000

该寄存器是只读的，提供了 4 个 ADC 状态的摘要。这个寄存器在 ADC1、ADC2 和 ADC3 中不可用。

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADC3_ROVF	ADC3_STRC	保留		ADC3_EOC	ADC3_WDE2	ADC3_WDE1	ADC3_WDE0	ADC2_ROVF	ADC2_STRC	保留		ADC2_EOC	ADC2_WDE2	ADC2_WDE1	ADC2_WDE0
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
ADC1_ROVF	ADC1_STRC	保留		ADC1_EOC	ADC1_WDE2	ADC1_WDE1	ADC1_WDE0	ADC0_ROVF	ADC0_STRC	保留		ADC0_EOC	ADC0_WDE2	ADC0_WDE1	ADC0_WDE0
r	r			r	r	r	r	r	r			r	r	r	r

位/位域	名称	描述
31	ADC3_ROVF	该位是 ADC3 的 ROVF 的镜像
30	ADC3_STRC	该位是 ADC3 的 STRC 的镜像
29:28	保留	必须保持复位值。
27	ADC3_EOC	该位是 ADC3 的 EOC 的镜像
26	ADC3_WDE2	该位是 ADC3 的 WDE2 的镜像

25	ADC3_WDE1	该位是 ADC3 的 WDE1 的镜像
24	ADC3_WDE0	该位是 ADC3 的 WDE0 的镜像
23	ADC2_ROVF	该位是 ADC2 的 ROVF 的镜像
22:21	保留	必须保持复位值。
20	ADC2_EOIC	该位是 ADC2 的 EOIC 的镜像
19	ADC2_EOC	该位是 ADC2 的 EOC 的镜像
18	ADC2_WDE2	该位是 ADC2 的 WDE2 的镜像
17	ADC2_WDE1	该位是 ADC2 的 WDE1 的镜像
16	ADC2_WDE0	该位是 ADC2 的 WDE0 的镜像
15	ADC1_ROVF	该位是 ADC1 的 ROVF 的镜像
14	ADC1_STRC	该位是 ADC1 的 STRC 的镜像
13:12	保留	必须保持复位值。
11	ADC1_EOC	该位是 ADC1 的 EOC 的镜像
10	ADC1_WDE2	该位是 ADC1 的 WDE2 的镜像
9	ADC1_WDE1	该位是 ADC1 的 WDE1 的镜像
8	ADC1_WDE0	该位是 ADC1 的 WDE0 的镜像
7	ADC0_ROVF	该位是 ADC0 的 ROVF 的镜像
6	ADC0_STRC	该位是 ADC0 的 STRC 的镜像
5:4	保留	必须保持复位值。
3	ADC0_EOC	该位是 ADC0 的 EOC 的镜像
2	ADC0_WDE2	该位是 ADC0 的 WDE2 的镜像
1	ADC0_WDE1	该位是 ADC0 的 WDE1 的镜像
0	ADC0_WDE0	该位是 ADC0 的 WDE0 的镜像

17.7.26. 同步控制寄存器（ADC_SYNCCTL）

地址偏移：0x304
复位值：0x0000 0000

这个寄存器在 ADC1 和 ADC2 中不可用。

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								ADCCK[3:0]				ADCSCK[3:0]			
								rw				rw			

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

位/位域	名称	描述
31:24	保留	必须保持复位值。
23:20	ADCCK[3:0]	ADC 时钟分频 这些位配置所有 ADC 的时钟，可以通过软件设置异步时钟频率分频。 0000：ADC 时钟 1 分频 0001：ADC 时钟 2 分频 0010：ADC 时钟 4 分频 0011：ADC 时钟 6 分频 0100：ADC 时钟 8 分频 0101：ADC 时钟 10 分频 0110：ADC 时钟 12 分频 0111：ADC 时钟 16 分频 1000：ADC 时钟 32 分频 1001：ADC 时钟 64 分频 1010：ADC 时钟 128 分频 1011：ADC 时钟 256 分频 其它值保留。
19:16	ADCSCK[3:0]	ADC 同步时钟配置 这些位配置所有 ADC 的时钟，可以通过软件设置 ADC 同步时钟模式。 0000：CLK_ADC（异步时钟模式） 1000：HCLK 2 分频（同步时钟模式） 1001：HCLK 4 分频（同步时钟模式） 1010：HCLK 6 分频（同步时钟模式） 1011：HCLK 8 分频（同步时钟模式） 1100：HCLK 10 分频（同步时钟模式） 1101：HCLK 12 分频（同步时钟模式） 1110：HCLK 14 分频（同步时钟模式） 1111：HCLK 16 分频（同步时钟模式） 其它值保留。
15:14	SYNCDMA[1:0]	ADC 同步 DMA 模式选择

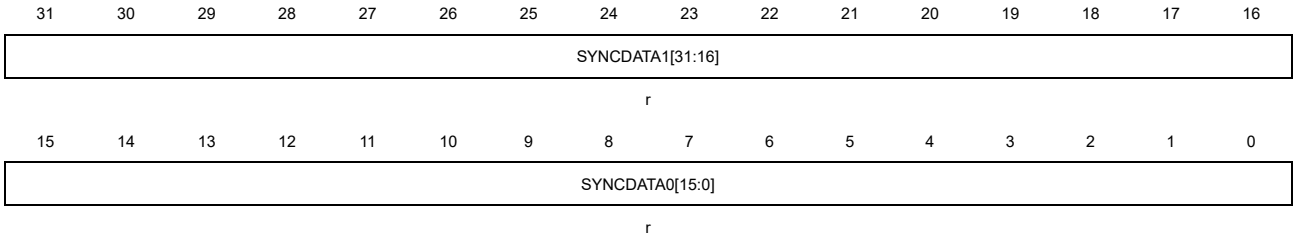
		00：ADC 同步 DMA 失能: 01：ADC 同步 DMA 模式 0 10：ADC 同步 DMA 模式 1 11：保留
13	SYNCDDM	ADC 同步 DMA 使能模式 该位配置 ADC 同步模式时 DMA 失能模式 0：当检测到来自 DMA 控制器的传输结束信号后，DMA 机制失能 1：当 SYNCDMA 不为 00 时，根据 SYNCDMA 位来产生 DMA 请求。
12	保留	必须保持复位值。
11:8	SYNCDLY[3:0]	ADC 同步延迟 在 ADC 同步模式中，这些位用于配置两个采样阶段之间的延迟为（5+SYNCDLY）ADC 时钟周期。
7:4	保留	必须保持复位值。
3:0	SYNCM[3:0]	ADC 同步模式 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 00000：ADC 同步模式失能。所有的 ADC 都独立工作。 00110：ADC0 和 ADC1 工作在常规并行模式。 00111：ADC0 和 ADC1 工作在常规跟随模式。 10110：ADC0、ADC1 和 ADC2 工作在常规并行模式。 10111：ADC0、ADC1 和 ADC2 工作在常规跟随模式。 其他值保留。

17.7.27. 同步常规数据寄存器（ADC_SYNCDATA）

地址偏移：0x308
复位值：0x0000 0000

这个寄存器在 ADC1、ADC2 和 ADC3 中不可用。

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:16	SYNCDATA1[31:16]	常规数据 1 在 ADC 同步模式中。
15:0	SYNCDATA0[31:0]	常规数据 0 在 ADC 同步模式中。

18. 数模转换器（DAC）

18.1. 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。通过校准可提高输出缓冲区的输出精度，采样保持模式可降低 DAC 的功耗。

采样保持模式可以降低 DAC 的功耗。

每个 DAC 的两个单元可以独立或并发工作。

18.2. 主要特性

DAC 的主要特征如下：

- 8 位或 12 位分辨率；
- 数据左对齐或右对齐；
- DMA 功能与欠载检测；
- 同步更新转换；
- 外部事件触发转换；
- 可配置的内部缓冲区；
- 输入参考电压 V_{REFP}；
- 输出缓冲区校准；
- 低功耗的采样保持功能；
- 噪声波生成（LSFR 噪声模式和三角噪声模式）；
- 锯齿波生成；
- DACx 双单元并发模式；
- 复位保持功能。

*图 18-1. DAC 结构框图*为 DAC 的结构框图，*表 18-1. DAC 引脚*给出了引脚描述。

图 18-1. DAC 结构框图

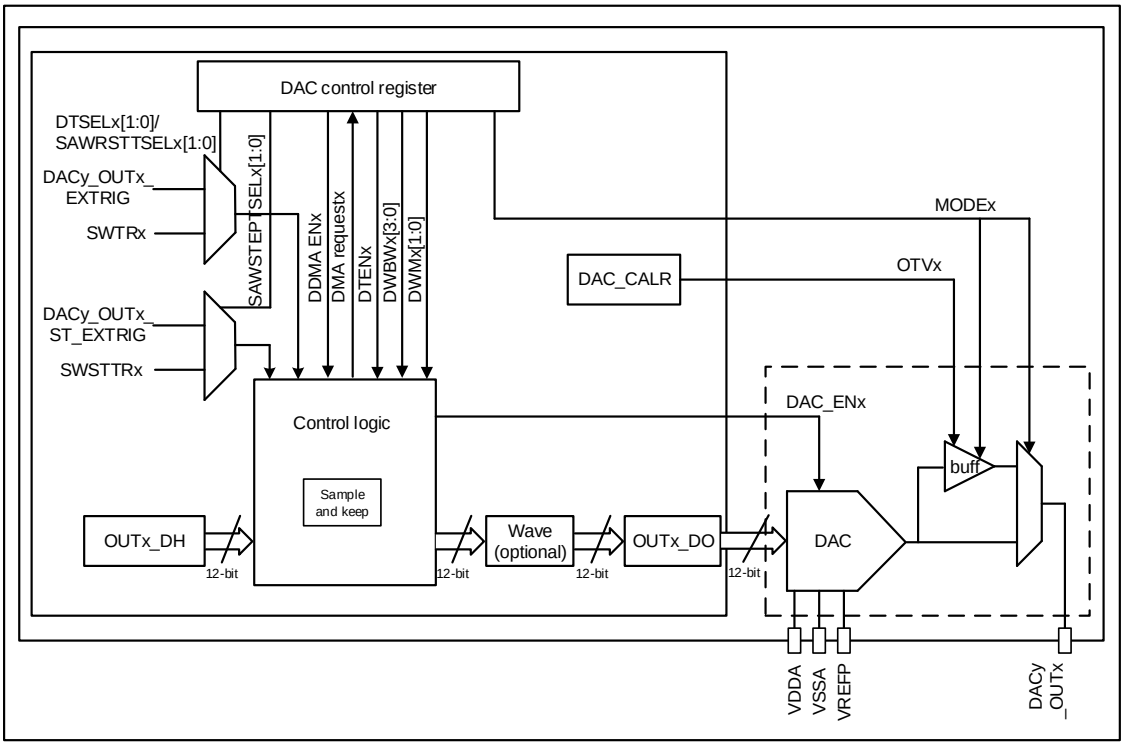


表 18-1. DAC 引脚

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 18-2. DAC 触发与输出

	DAC0		DAC1		DAC2		DAC3	
单元	单元 0	单元 1	单元 0	单元 1	单元 0	单元 1	单元 0	单元 1
DAC 输出 I/O	PA4	PA5	PA6	PA7	/	/	/	/
DAC 输出 BUFFER 功能	•	•	•	•	/	/	/	/
TRIGSEL 触发功能	•		•		•		•	
软件触发功能	•		•		•		•	
最大采样率	1MSPS				15MSPS			

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

18.3. 功能描述

18.3.1. 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。

18.3.2. 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_MDCR 寄存器的 MODEx 位来开启或关闭缓冲区。

注意：DAC2 与 DAC3 无输出缓冲功能。

18.3.3. 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。

写入 DAC_OUTx_R12DH, DAC_OUTx_L12DH or DAC_OUTx_R8DH 寄存器的数据默认以无符号格式处理，也可以通过设置 DAC_MDCR 寄存器的 DHFMTx 位为 1，配置成有符号格式（二进制补码，如 Q1.15, Q1.11 or Q1.7 格式）。

无论有符号或者无符号数据格式，DAC 输出电压范围都是 0 到 V_{REFP}。并且传输到 DAC_OUTx_DO 寄存器的数据将被转换成无符号格式，如下表所示。

表 18-3. DAC 数据格式（12-bit 数据）

DHFMTx 位	写入 DAC_OUTx_DH 寄存器的数据	十进制值	传输到 DAC_OUTx_DO 寄存器的数据	输出电压
0	0x000	0	0x000	0
0	0xFFF	4095	0xFFF	V_{REFP}
1	0x7FF	2047	0xFFF	V_{REFP}
1	0x000	0	0x800	$\frac{V_{REFP}}{2}$
1	0xFFF	-1	0x7FF	$\frac{V_{REFP}}{2} - \frac{V_{REFP}}{4096}$

DHFMTx 位	写入 DAC_OUTx_DH 寄存器的数据	十进制值	传输到 DAC_OUTx_DO 寄存器的数据	输出电压
1	0x800	-2048	0x000	0

18.3.4. 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 18-4. DAC 外部触发*所示。

表 18-4. DAC 外部触发

DTSELx[1:0]	触发源	触发类型
2b'00	来自 TRIGSEL 的外部触发 DAC_OUTx_EXTRIG	硬件触发
2b'01	DAC_SWT 寄存器的 SWTRx	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。

18.3.5. 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）。

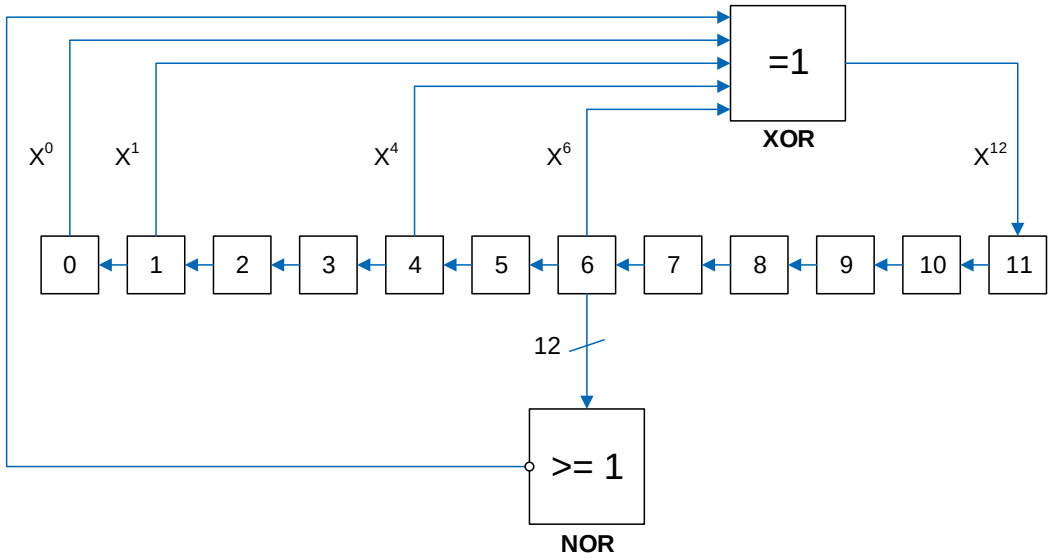
当 DAC 保持数据（OUTx_DH）加载到 DAC_OUTx_DO 寄存器时，经过 $t_{SETTLING}$ 时间之后，模拟输出变得有效，$t_{SETTLING}$ 的值与电源电压和模拟输出负载有关。

18.3.6. 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。

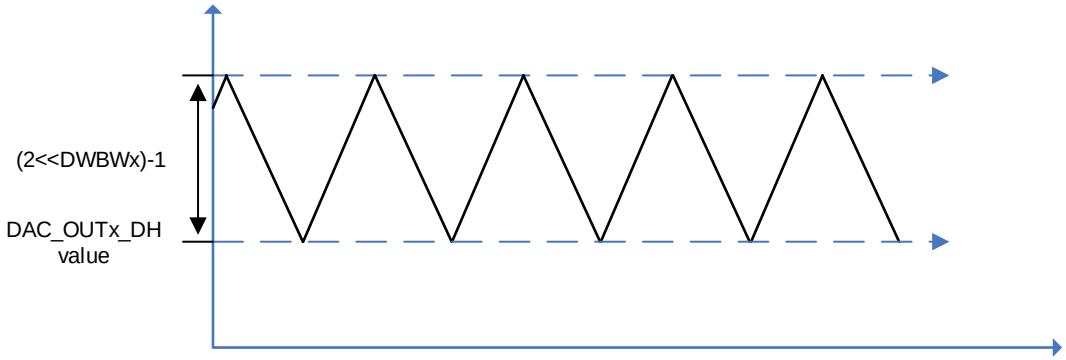
LFSR 噪声模式：在 DAC 控制逻辑中有一个线性反馈移位寄存器（LFSR）。在此模式下，LFSR 的值与 OUTx_DH 值相加后，被写入到 DAC 数据输出寄存器（DAC_OUTx_DO）。当配置的 DAC 噪声波位宽小于 12 时，LFSR 的值等于 LFSR 寄存器最低的 DWBWx 位，高位被屏蔽。

图 18-2. 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 18-3. DAC 三角噪声模式生成的波形



18.3.7. DAC 锯齿波

为了产生锯齿波，模块实现了一个带复位触发和递增/递减触发的 16 位锯齿波计数器。

计数器初始值由 DAC_OUTx_SAW 寄存器的 SAWINITx[11:0]位指定，寄存器的高 12 到 15 位被设为 4b'0000。

计数器的步进方向由 DAC_OUTx_SAW 寄存器的 SAWDIRx 位定义，此位决定计数器向上或者向下计数。

计数器的步进值由 DAC_OUTx_SAW 寄存器的 SAWSTEPx[15:0]位定义。锯齿波计数器的高 12 位将被传输到 DAC_OUTx_DO 寄存器用以转换输出电压。

锯齿波计数器在每个步进（复位）触发信号上升沿递增/递减（复位）。递增/递减（复位）触发信号分别由 DAC_SAWMDR 寄存器的 SAWRSTTSELx 位和 SAWSTEPTSELx 位选择。

表 18-5. DAC 锯齿波复位触发信号

SAWRSTTSELx[1:0]	触发源	触发类型
2b'00	来自 TRIGSEL 的外部触发 DAC_OUTx_EXTRIG	硬件触发
2b'01	DAC_SWT 寄存器的 SWTRx	软件触发
2b'10	保留	保留
2b'11		

表 18-6. DAC 锯齿波步进触发信号

SAWSTEPTSELx[1:0]	触发源	触发类型
2b'00	来自 TRIGSEL 的外部触发 DAC_OUTx_ST_EXTRIG	硬件触发
2b'01	DAC_SWT 寄存器的 SWSTTRx	软件触发
2b'10	保留	保留
2b'11		

图 18-4. DAC 锯齿波（SAWDIRx = 1）

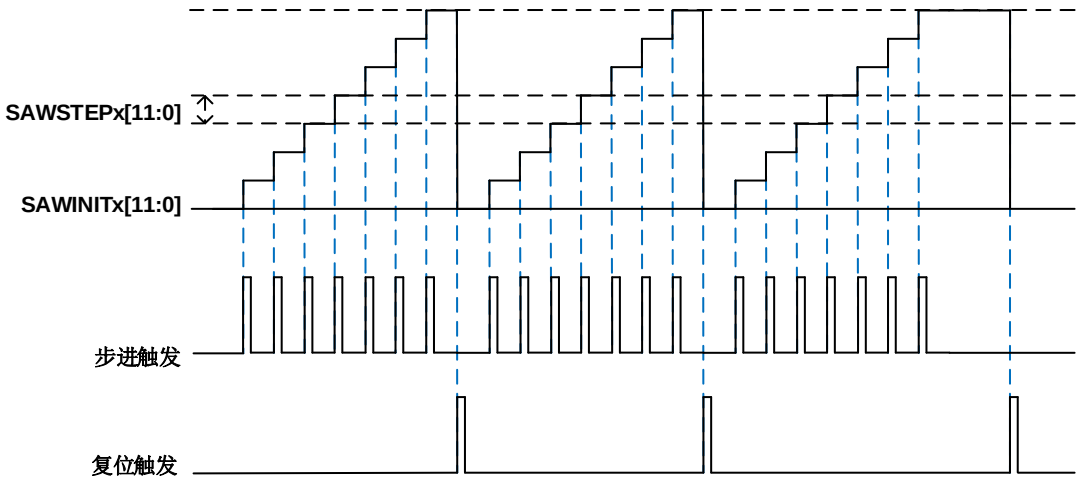
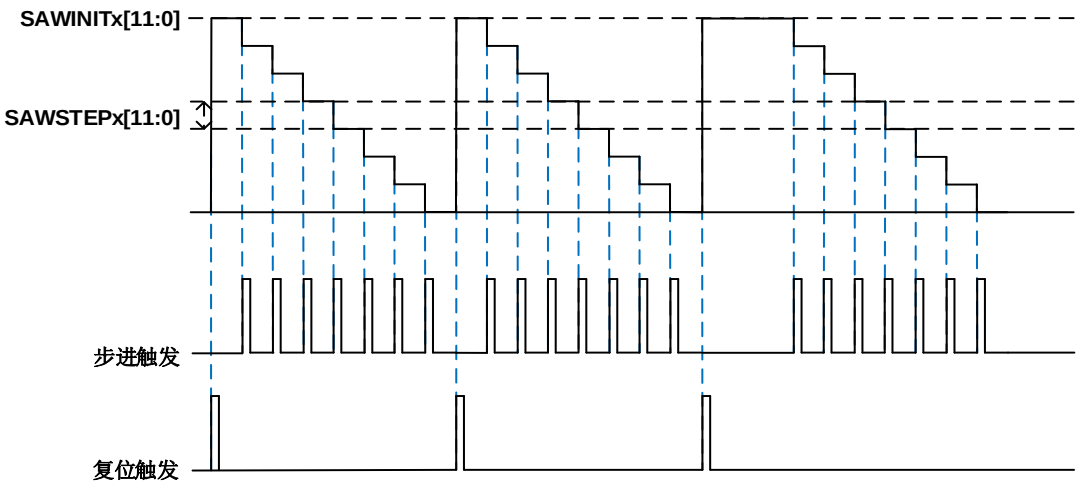


图 18-5. DAC 锯齿波（SAWDIRx = 0）



18.3.8. DAC 输出电压

DAC 引脚上的模拟输出电压取决于下面的等式：

$$V_{DAC_OUT}=V_{REFP}*OUTx_DO/4096 \quad (18\text{-}1)$$

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}。

18.3.9. 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。

18.3.10. DAC 并发转换

当 DAC 的两个单元同时工作时，为了在特定应用中最大限度利用总线带宽，DAC 的两个单元可以被配置为并发模式。在并发模式中，DAC 的 OUTx_DH 和 DAC_OUTx_DO 寄存器值将同时被更新。

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

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

当使能了 DMA 功能时，DAC 任一单元的 DDMAENx 位置 1 即可。

噪声模式和噪声位宽可以根据使用情况配置为相同或不同。

18.3.11. DAC 复位保持模式

通过设置 DAC_CTL0 寄存器的 DRSTMDx 位使能 DAC 的复位保持模式，在除了上电复位之外的其他复位信号到来时，DAC 单元的输出将会保持。

当 DRSTMDx 位置 1 时，寄存器 DAC_CTL0(DRSTMDx, DTENx, DBOFFx and DENx), OUTx_DH, DAC_OUTx_DO 只能被上电复位清零。

注意：当 DAC 使用内部参考电压（VREF_EN = 1）时，需要使能 VREF 的复位保持功能来实现此模式。

18.3.12. DAC 输出缓冲区校准

当 DAC 使用缓冲区时，输出电压可能会发生偏移，因此需要对输出电压进行补偿。

DAC 校准函数为：

$$V_{out}=(D/2^{N-1})*G*V_{REFP}+V_{of} \quad (18\text{-}2)$$

式中，N 是 DAC 的有效位数，D 是 DAC 的数字输入，G 是增益，V_{REFP}是 DAC 的参考电压，V_{of}是偏移电压，对于理想 DAC，G是 1，V_{of}是 0。

当缓冲区启用时，校准将生效，在校准过程中：

- 缓冲区与外部引脚和片上外设断开连接并进入三态。
- 缓冲区将用作比较器来检测中间码值 0x800，并通过内部电桥与$V_{REFP}/2$进行对比，DAC_STAT0 寄存器的 CALFx 位会根据比较结果置 1 或清零。

有两种校准方法可用：

- 出厂校准(始终使能)
  - DAC 缓冲区偏移在工厂进行校准，当 DAC 复位时，自动加载 DAC_CALR 寄存器 OTV0[4:0]的默认值。
- 用户校准
  - 如果用户工作条件与工厂校准条件不同，特别是 VDDA, V_{REFP} 和温度发生改变时，用户可在应用过程中通过软件进行校准。

用户校准过程为：

- DAC_CTL0 寄存器中 DENx 位写 0 以禁能 DAC 输出。
- DAC_CTL0 寄存器中 CALENx 位置 1 使能 DAC 校准。
- 执行校准算法
  - 从 0x00000b 开始写入 OTVx[4:0]。
  - 等待T_{cal}时间。

- 检查 DAC_STAT0 寄存器中的 CALFx 位。
- 当 CALFx 位置 1，证明正确的校准值已找到，否则码值加 1 直至找到正确的校准值。

通过使用逐次逼近或二分法技术，可以更快地计算 OTVx[4:0]的值。

注意：校准过程后，CALENx 应写入 0，然后将 DENx 写入 1，从而在正常模式下使用 DAC。禁止同时将 DENx 和 CALENx 设置为 1。

18.3.13. DAC 模式

DAC 两个单元可以配置为普通模式和采样保持模式。DAC 输出可连接到外部引脚或片上外设。

普通模式

通过设置 DAC_MDCR 寄存器的 MODEx[2]位为 0，DAC 工作在普通模式。

采样保持模式

通过设置 DAC_MDCR 寄存器的 MODEx[2]位为 1，DAC 工作在采样保持模式。DAC 内核在触发转换后对数据进行转换，并将转换后的电压在电容上保持。当不转换时，DAC 内核在两次采样之间保持关闭状态。并且 DAC 输出为三态，因此可以降低整体功耗。在此模式下，DAC 内核和所有相应的逻辑以及寄存器均由 IRC32K 驱动。因此，DAC 可在深度睡眠模式下使用。

采样保持模式可分为三个阶段：

采样阶段

采样保持原件被充电到所需电压，充电时间取决于电容值，采样时间由 DAC_SKSTRx 寄存器中的 TSAMPx[9:0]位配置。当对 TSAMPx[9:0]位进行写操作时，DAC_SATA0 寄存器中的 BWTx 位会自动置 1，用以指示 AHB 时钟和 IRC32K 时钟正在同步。当写操作完成时，BWTx 位由硬件自动清零，用户可以再次对 TSAMPx[9:0]位执行写操作。在 DAC 正常输出过程中，可通过软件更改 TSAMPx[9:0]。

保持阶段：

在保持阶段，DAC 内核在保持阶段处于关闭状态，从而降低系统功耗，保持时间由 DAC_SKKTR 寄存器中的 TKEEPx[9:0]位配置，该模式下 DAC 输出为三态。

刷新阶段：

在刷新阶段，DAC 内核再次打开，将下降的电压充电至目标值。刷新时间由 DAC_SKRTR 寄存器中的 TREFx[7:0]位配置。

当新的 OUTx_DH 更新时（DTENx=1 时触发或 DTENx=0 时更新），操作阶段将进入采样阶段，

同时 DAC 内核将新数据转换为所需电压，在采样保持模式下，两个连续的数据更新操作之间需要 3 个以上的 IRC32K 时钟周期才能同步。

注意:

如果在采样保持模式下选择锯齿波生成（DWMx=2'b11），SAWRSTTSELx and DTSELx (x= 0,1) 配置需要保持一致。

时间计算

上述三个阶段的时间计算均基于IRC32K时钟周期。为了配置足够的采样和刷新时间，参考下面的公式：

表 18-7. 采样和刷新时间计算公式

缓冲状态	t_{sample} (1) (2)	$t_{refresh}$ (1) (2) (3)	t_{keep} (3)
开启	$t_{wakeup} + R_{BON}*C_{SK}*ln(2^{N+1})$	$t_{wakeup} + R_{BON}*C_{SK}*ln(2*N_{LSB})$	$(V_{REFP}/2^N)*N_{LSB}*C_{SK}/I_{leak}$
关闭	$t_{wakeup} + R_{BOFF}*C_{SK}*ln(2^{N+1})$	$t_{wakeup} + R_{BOFF}*C_{SK}*ln(2*N_{LSB})$	$(V_{REFP}/2^N)*N_{LSB}*C_{SK}/I_{leak}$

注意:

(1) 上述公式，t_{wakeup}是DAC从关闭状态唤醒到输出达到最终设置值的时间，充电时间计算于电容电压重新充电到设定输出1/2 LSB误差以内的时间。N是DAC分辨率，12位或8位。

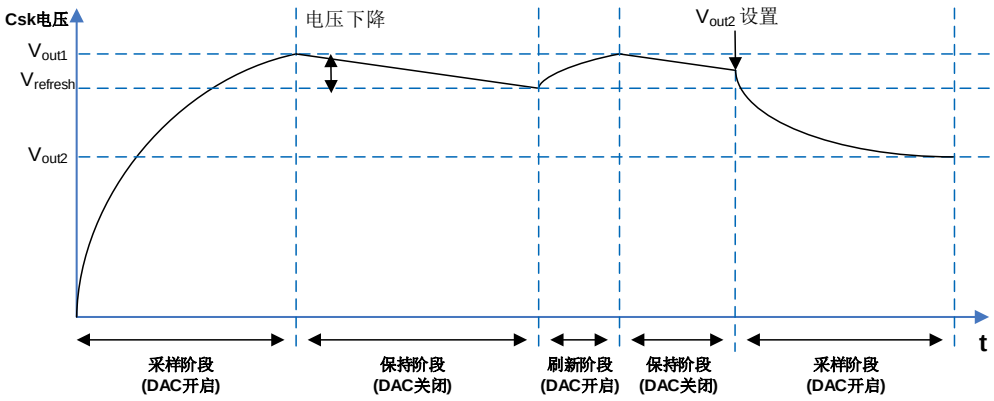
(2) R_{BON}/R_{BOFF}是输出缓冲打开或者关闭的输出阻抗，C_{SK}是采样保持电容(内部或外部)。当DAC_MDCR寄存器的MODEx[2:0]位是3'b111时，内部电容用于保持DAC输出到片上外设的电压。

(3) 保持时间取决于容许的电压下降值，保持阶段采样保持电容会因输出漏电流放电导致电压下降。N_{LSB}代表电压下降的位数，I_{leak}是输出漏电流。

(4) R_{BON}，R_{BOFF}，C_{SK}以及t_{wakeup}的值请参考器件数据手册。

采样保持阶段框图如下所示。

图 18-6. DAC 采样保持



18.3.14. DAC 低功耗模式

睡眠模式

在睡眠模式中，DAC 可正常工作，并且可以与 DMA 一起使用。

深度睡眠模式

在深度睡眠模式中，若在进入深度睡眠模式前，采样保持功能使能，DAC 可保持静态输出，否则 DAC 停止工作。

待机模式

在待机模式中，DAC 停止工作，退出待机模式并重新初始化 DAC，DAC 可再次工作。

18.4. DAC 寄存器

DAC0 基地址：0x5000 1000

DAC1 基地址：0x5000 1400

DAC2 基地址：0x5000 1800

DAC3 基地址：0x5000 1C00

18.4.1. 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
DRSTMD1	CALEN1	DDUDRIE1	DDMAEN1	DWBW1[3:0]				DWM1[1:0]		保留		DTSEL1[1:0]		DTEN1	DEN1
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
DRSTMD0	CALEN0	DDUDRIE0	DDMAEN0	DWBW0[3:0]				DWM0[1:0]		保留		DTSEL0[1:0]		DTEN0	DEN0
rw	rw	rw	rw	rw				rw				rw		rw	rw

位/位域	名称	描述
31	DRSTMD1	DACx_OUT1 复位模式。 该位只会被上电复位所复位。 0：普通模式：所有的的复位信号都会复位 DACx_OUT1 相关的寄存器。 1：复位保持模式：在除了上电复位之外的其他复位信号到来时，DACx_OUT1 的输出将会保持。
30	CLAEN1	DACx_OUT1校准使能 0：DACx_OUT1 DMA校准模式禁能 1：DACx_OUT1 DMA 校准模式使能 只有 DEN1=0 时，才可对 CALEN1 写 1。
29	DDUDRIE1	DACx_OUT1 DMA欠载中断使能 0：DACx_OUT1 DMA欠载中断禁能 1：DACx_OUT1 DMA 欠载中断使能
28	DDMAEN1	DACx_OUT1 DMA 使能 0：DACx_OUT1 DMA 模式禁能

		1：DACx_OUT1 DMA 模式使能
27:24	DWBW1[3:0]	DACx_OUT1 噪声波位宽 这些位指定了 DACx_OUT1 的噪声波信号的位宽。LFSR 噪声模式下，这些位表示不屏蔽 LFSR 的位[n-1, 0]；三角噪声模式下，这些位表示三角波幅值为(2<<(n-1))-1。其中，n 为噪声波位宽。 0000：波形信号的位宽为 1 0001：波形信号的位宽为 2 0010：波形信号的位宽为 3 0011：波形信号的位宽为 4 0100：波形信号的位宽为 5 0101：波形信号的位宽为 6 0110：波形信号的位宽为 7 0111：波形信号的位宽为 8 1000：波形信号的位宽为 9 1001：波形信号的位宽为 10 1010：波形信号的位宽为 11 ≥1011：波形信号的位宽为 12
23:22	DWM1[1:0]	DACx_OUT1 噪声波模式 这些位指定了在 DACx_OUT1 外部触发使能(DTEN1=1)的情况下，DACx_OUT1 的噪声波模式的选择。 00：波形生成禁能 01：LFSR 噪声模式 10：三角波噪声模式 11：锯齿波噪声模式
21:20	保留	必须保持复位值。
19:18	DTSEL1[1:0]	DACx_OUT1 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：来自 TRIGSEL 的外部触发 DACx_OUT1_EXTRIG 01：软件触发 其他值：保留
17	DTEN1	DACx_OUT1 触发使能 0：DACx_OUT1 触发禁能 1：DACx_OUT1 触发使能
16	DEN1	DACx_OUT1 使能 0：DACx_OUT1 禁能 1：DACx_OUT1 使能
15	DRSTMD0	DACx_OUT0 复位模式。

		该位只会被上电复位所复位。 0：普通模式：所有的的复位信号都会复位 DACx_OUT0 相关的寄存器。 1：复位保持模式：在除了上电复位之外的其他复位信号到来时，DACx_OUT0 的输出将会保持。
14	CLAEN0	DACx_OUT0 校准使能 0：DACx_OUT0 DMA校准模式禁能 1：DACx_OUT0 DMA 校准模式使能 只有 DEN0=0 时，才可对 CALEN0 写 1。
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 10：三角波噪声模式 11：锯齿波噪声模式

5:4	保留	必须保持复位值。
3:2	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：来自 TRIGSEL 的外部触发 DACx_OUT0_EXTRIG 01：软件触发 其他值：保留
1	DTEN0	DACx_OUT0 触发使能 0：DACx_OUT0 触发禁能 1：DACx_OUT0 触发使能
0	DEN0	DACx_OUT0 使能 0：DACx_OUT0 禁能 1：DACx_OUT0 使能

18.4.2. 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														SWSTTR1	SWSTTR0
														w	w

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

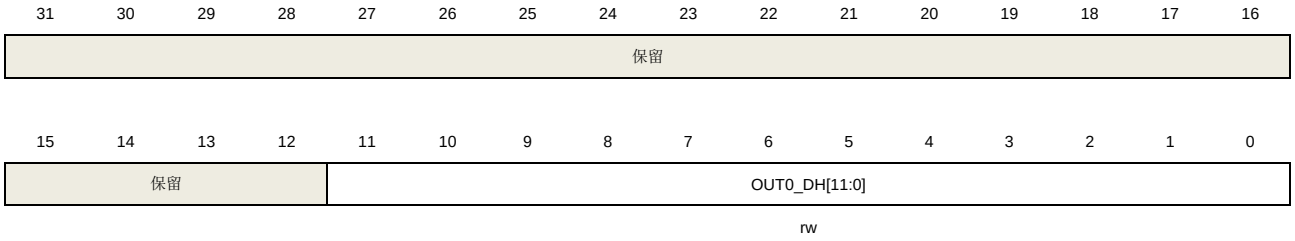
位/位域	名称	描述
31:18	保留	必须保持复位值
17	SWSTTR1	DACx_OUT1 锯齿波计数器递增/递减软件触发. 当DAC触发选择软件触发时，此位写1产生触发信号（锯齿波生成）。由硬件自动清零。 0: 软件触发禁能 1: 软件触发使能
16	SWSTTR0	DACx_OUT0 锯齿波计数器递增/递减软件触发. 当DAC触发选择软件触发时，此位写1产生触发信号（锯齿波生成）。由硬件自动清零。 0: 软件触发禁能

		1: 软件触发使能
15:2	保留	必须保持复位值
1	SWTR1	DACx_OUT1 转换或者锯齿波复位软件触发，由硬件清除。 0：软件触发禁能 1：软件触发使能
0	SWTR0	DACx_OUT0 转换或者锯齿波复位软件触发，由硬件清除。 0：软件触发禁能 1：软件触发使能

18.4.3. 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。

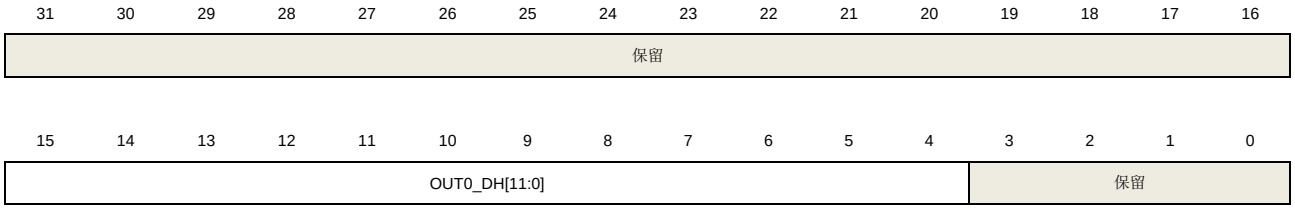


位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	OUT0_DH[11:0]	DACx_OUT0 12 位右对齐数据 这些位指定了将由 DACx_OUT0 转换的数据。

18.4.4. 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。



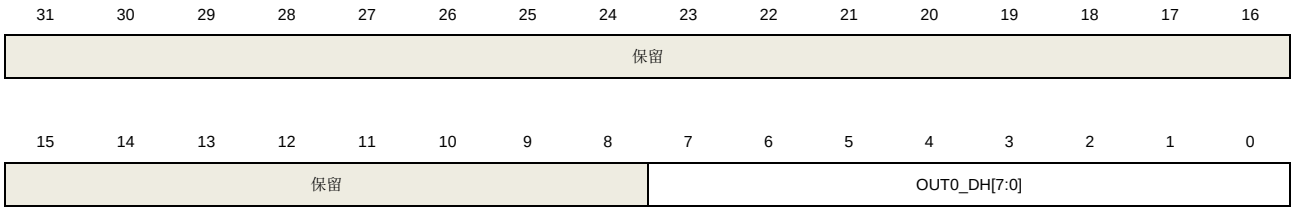
rw

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

18.4.5. 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。



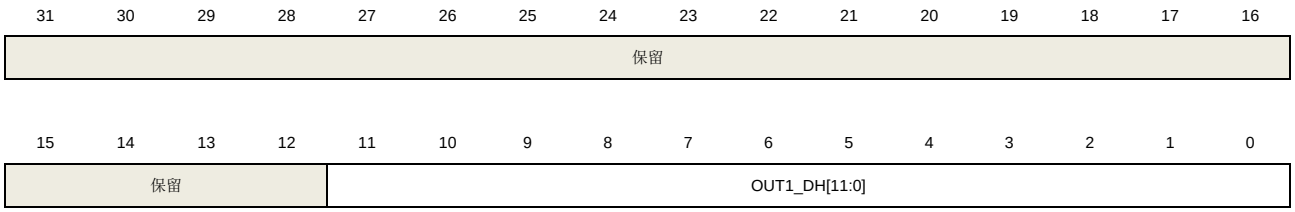
rw

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。

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

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



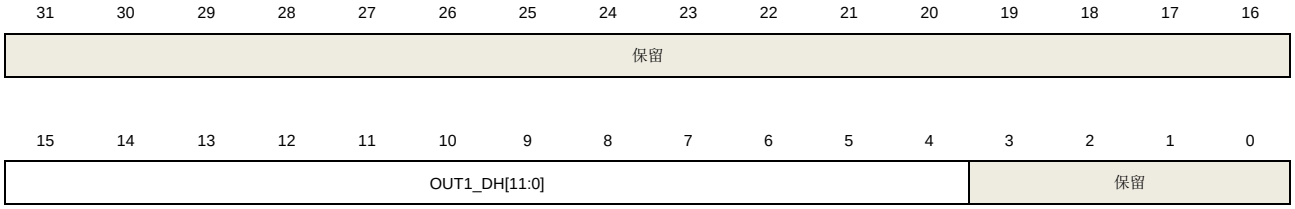
rw

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

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

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

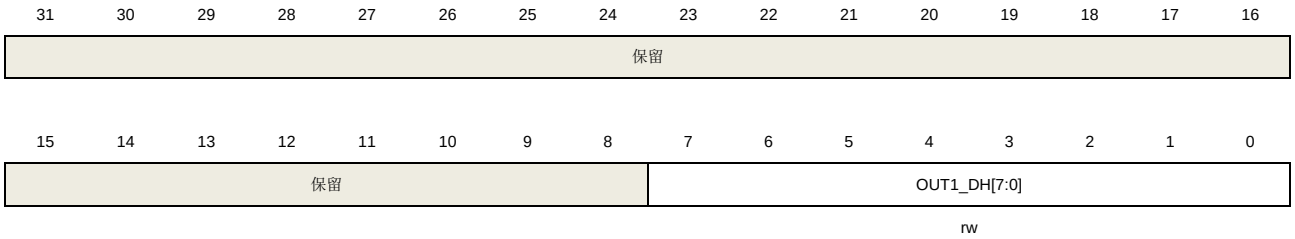


位/位域	名称	描述
31:16	保留	必须保持复位值
15:4	OUT1_DH[11:0]	DACx_OUT1 12 位左对齐数据 这些位指定了将由 DACx_OUT1 转换的数据。
3:0	保留	必须保持复位值

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

地址偏移：0x1C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

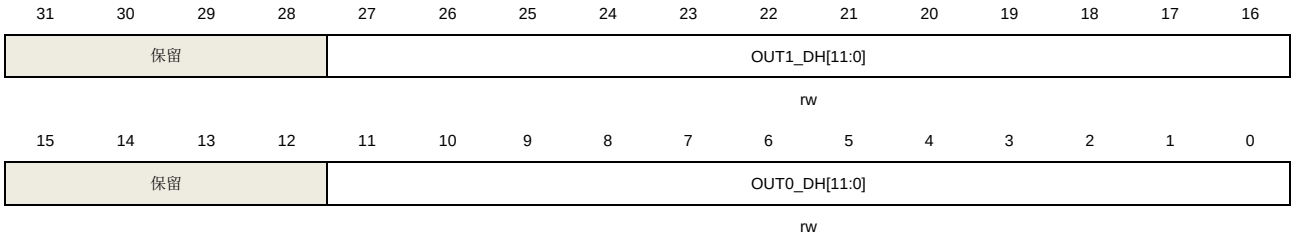


位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	OUT1_DH[7:0]	DACx_OUT1 8 位右对齐数据 这些位指定了将由 DACx_OUT1 转换的数据的 8 位最高有效位。

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

地址偏移：0x20
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

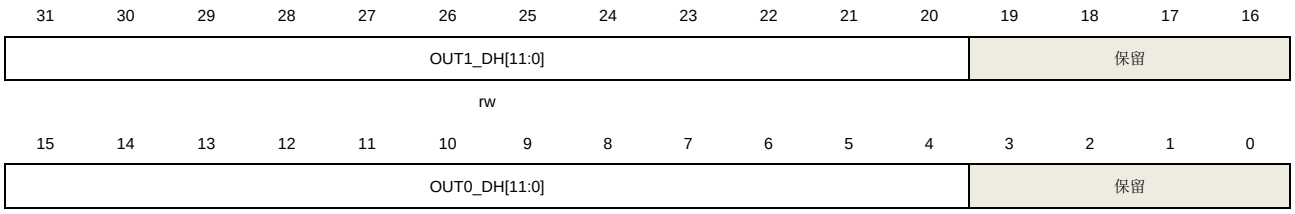


位/位域	名称	描述
31:28	保留	必须保持复位值
27:16	OUT1_DH[11:0]	DACx_OUT1 12 位右对齐数据 这些位指定了将由 DACx_OUT1 转换的数据。
15:12	保留	必须保持复位值
11:0	OUT0_DH[11:0]	DACx_OUT0 12 位右对齐数据 这些位指定了将由 DACx_OUT0 转换的数据。

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

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



rw

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

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

地址偏移：0x28
复位值：0x0000 0000

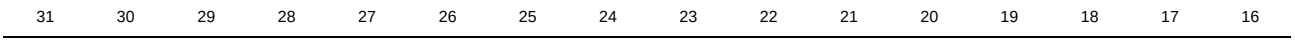
该寄存器只能按字(32 位)访问。

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

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

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



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。

18.4.13. DACx_OUT1 数据输出寄存器 (DAC_OUT1_DO)

地址偏移：0x30
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	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				OUT1_DO[11:0]											
				r											

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

18.4.14. DACx 状态寄存器 0 (DAC_STAT0)

地址偏移：0x34
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BWT1	CALF1	DDUDR1	保留												
r	r	rc_w1													

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

BWT0	CALF0	DDUDR0	保留
r	r	rc_w1	

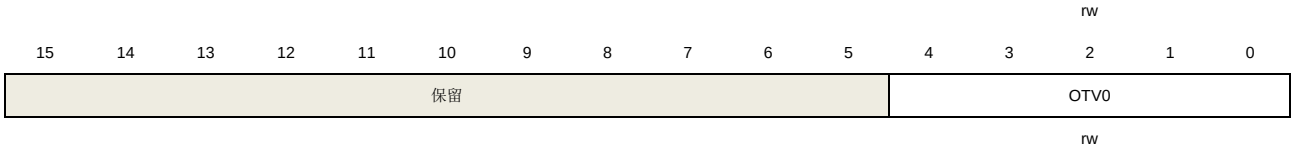
位/位域	名称	描述
31	BWT1	DACx_OUT1 TSAMP1[9:0]写忙标志。 当使能采样保持模式后，该位由系统设置，当 TSAMP1[9:0]正在执行写操作时，该位置 1；当完成写操作后，硬件清零。 0：TSAMP1[9:0]没有进行写操作 1：TSAMP1[9:0]正在进行行写操作
30	CALF1	DACx_OUT1 校准偏移标志，该位由硬件置 1 和清零。 0：校准值低于偏移校正值。 1：校准值等于或大于偏移校正值
29	DDUDR1	DACx_OUT1 DMA 欠载标志位，硬件置位，软件写 1 清零。 0：没有欠载发生 1：发生欠载（DAC 触发产生速度快于 DMA 传输速度）
28:16	保留	必须保持复位值。
15	BWT0	DACx_OUT0 TSAMP0[9:0]写忙标志。 当使能采样保持模式后，该位由系统设置，当 TSAMP0[9:0]正在执行写操作时，该位置 1；当完成写操作后，硬件清零。 0：TSAMP0[9:0]没有进行写操作 1：TSAMP0[9:0]正在进行行写操作
14	CALF0	DACx_OUT0 校准偏移标志，该位由硬件置 1 和清零。 0：校准值低于偏移校正值 1：校准值等于或大于偏移校正值
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。

18.4.15. DACx 校准寄存器 (DAC_CALR)

地址偏移：0x38
复位值：0x00XX 00XX

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留											OTV1				



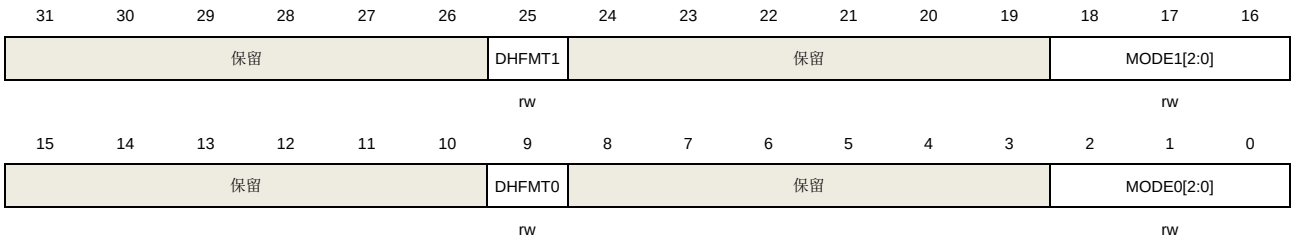
位/位域	名称	描述
31:21	保留	必须保持复位值。
20:16	OTV1[4:0]	DACx_OUT1 偏移校准值。
15:5	保留	必须保持复位值。
4:0	OTV0[4:0]	DACx_OUT0 偏移校准值。

18.4.16. DACx 模式寄存器 (DAC_MDCR)

地址偏移：0x3C

复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:26	保留	必须保持复位值。
25	DHFMT1	DACx_OUT1 写入数据保持寄存器数据格式 0: 写入的数据以无符号格式处理 1: 写入的数据以有符号（二进制补码）格式处理
24:19	保留	必须保持复位值。
18:16	MODE1[2:0]	DACx_OUT1 模式 当 DAC_CTL0 寄存器中 DEN1=0 和 CALEN1=0 时才可对这些位进行写操作。当 DAC_CTL0 寄存器中 DEN1=1 或 CALEN1=1，写操作无效。 -普通模式下 DACx_OUT1 000:DACx_OUT1 连接到外部引脚，缓冲区启用。 001: DACx_OUT1 连接到外部引脚和片上外设，缓冲区启用。 010:DACx_OUT1 连接到外部引脚，缓冲区禁用。

		011: DACx_OUT1 连接到片上外设，缓冲区禁用。 -采样保持模式下 DACx_OUT1 100:DACx_OUT1 连接到外部引脚，缓冲区启用。 101: DACx_OUT1 连接到外部引脚和片上外设，缓冲区启用。 110: DACx_OUT1 连接到外部引脚和片上外设，缓冲区禁用。 111: DACx_OUT1 连接到片上外设，缓冲区禁用。 注意：对于无输出缓冲的 DACx(x=2,3)，只有 MODE1[2]位是可用的，用于选择正常模式或者采样保持模式。
15:10	保留	必须保持复位值。
9	DHFMT0	DACx_OUT0 写入数据保持寄存器数据格式 0: 写入的数据以无符号格式处理 1: 写入的数据以有符号（二进制补码）格式处理
8:3	保留	必须保持复位值。
2:0	MODE0[2:0]	DACx_OUT0 模式 当 DAC_CTL0 寄存器中 DEN0=0 和 CALEN0=0 可进行写操作。当 DAC_CTL0 寄存器中 DEN0=1 或 CALEN0=1，写操作被忽略）。 -普通模式下 DAxC_OUT0 000:DACx_OUT0 连接到外部引脚，缓冲区启用。 001: DACx_OUT0 连接到外部引脚和片上外设，缓冲区启用。 010:DACx_OUT0 连接到外部引脚，缓冲区禁用。 011: DACx_OUT0 连接到片上外设，缓冲区禁用。 -采样保持模式下 DACx_OUT0 100:DACx_OUT0 连接到外部引脚，缓冲区启用。 101: DACx_OUT0 连接到外部引脚和片上外设，缓冲区启用。 110: DACx_OUT0 连接到外部引脚和片上外设，缓冲区禁用。 111: DACx_OUT0 连接到片上外设，缓冲区禁用。 注意：对于无输出缓冲的 DACx(x=2,3)，只有 MODE0[2]位是可用的，用于选择正常模式或者采样保持模式。

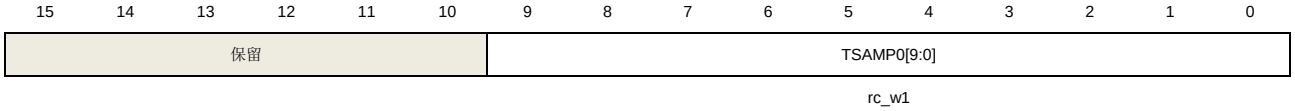
18.4.17. DACx 采样保持模式采样时间寄存器 0 (DAC_SKSTR0)

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															

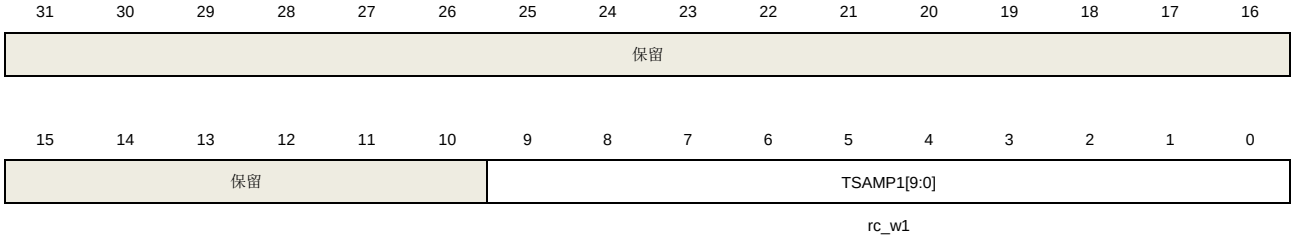


位/位域	名称	描述
31:10	保留	必须保持复位值。
9:0	TSAMP0[9:0]	DACx_OUT0 采样时间。

18.4.18. DACx 采样保持模式采样时间寄存器 1 (DAC_SKSTR1)

地址偏移：0x44
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:10	保留	必须保持复位值。
9:0	TSAMP1[9:0]	DACx_OUT1 采样时间。

18.4.19. DACx 采样保持模式保持时间寄存器 (DAC_SKKTR)

地址偏移：0x48
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

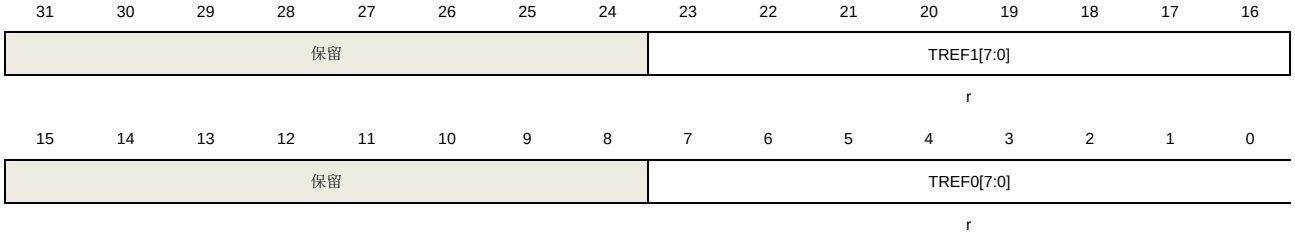
位/位域	名称	描述

31:26	保留	必须保持复位值。
25:16	TKEEP1[9:0]	DACx_OUT1 保持时间(仅在采样保持模式有效)。
15:10	保留	必须保持复位值。
9:0	TKEEP0[9:0]	DACx_OUT0 保持时间(仅在采样保持模式有效)。

18.4.20. DACx 采样保持模式刷新时间寄存器 (DAC_SKRTR)

地址偏移：0x4C
复位值：0x0001 0001

该寄存器只能按字(32 位)访问。

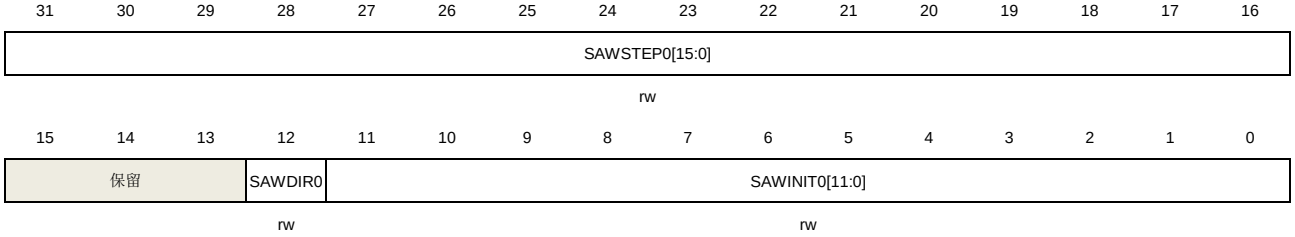


位/位域	名称	描述
31:24	保留	必须保持复位值。
23:16	TREF1[7:0]	DACx_OUT1 刷新时间(仅在采样保持模式有效)。
15:8	保留	必须保持复位值。
7:0	TREF0[7:0]	DACx_OUT0 刷新时间(仅在采样保持模式有效)。

18.4.21. DACx_OUT0 锯齿波寄存器(DAC_OUT0_SAW)

地址偏移：0x58
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



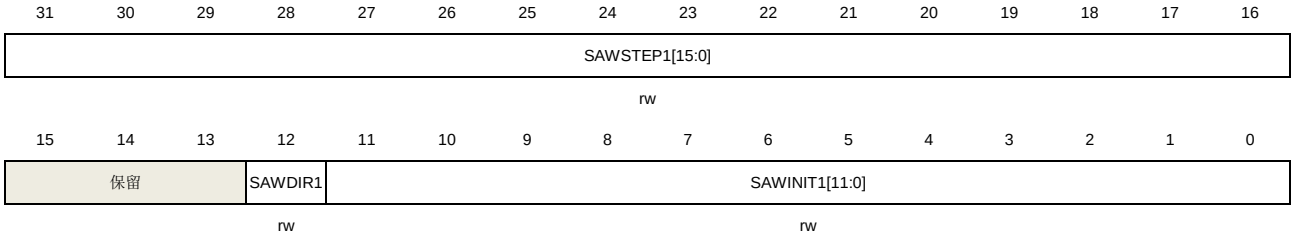
位/位域	名称	描述
31:16	SAWSTEP0[15:0]	DACx_OUT0 锯齿波步长值。
15:13	保留	必须保持复位值。
12	SAWDIR0	DACx_OUT0锯齿波步进方向。 软件写此位选择锯齿波步进方向。 0: 向下递减 1: 向上递增
11:0	SAWINIT0[11:0]	DACx_OUT0 锯齿波初始值。

18.4.22. DACx_OUT1 锯齿波寄存器(DAC_OUT1_SAW)

地址偏移：0x5C

复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:16	SAWSTEP1[15:0]	DACx_OUT1 锯齿波步长值。
15:13	保留	必须保持复位值。
12	SAWDIR1	DACx_OUT1锯齿波步进方向。 软件写此位选择锯齿波步进方向。 0: 向下递减 1: 向上递增
11:0	SAWINIT1[11:0]	DACx_OUT1 锯齿波初始值

18.4.23. DACx 锯齿波模式寄存器 (DAC_SAWMDR)

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	SAWSTEPTSEL1[1:0]	保留	SAWRSTTSEL1[1:0]
	rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						SAWSTEPTSEL0[1:0]		保留						SAWRSTTSEL0[1:0]	
						rw								rw	

位/位域	名称	描述
31:26	保留	必须保持复位值。
25:24	STEPTSEL1[1:0]	DACx_OUT1 锯齿波递增/递减触发选择。 00: 来自 TRIGSEL 的外部触发 DACx_OUT1_ST_EXTRIG 01: 软件触发 其他值：保留
23:18	保留	必须保持复位值。
17:16	RSTTSEL1[1:0]	DACx_OUT1 锯齿波复位触发选择。 00: 来自 TRIGSEL 的外部触发 DACx_OUT1_EXTRIG 01: 软件触发 其他值：保留
15:10	保留	必须保持复位值。
9:8	STEPTSEL0[1:0]	DACx_OUT0 锯齿波递增/递减触发选择。 00: 来自 TRIGSEL 的外部触发 DACx_OUT0_ST_EXTRIG 01: 软件触发 其他值：保留
7:2	保留	必须保持复位值。
1:0	RSTTSEL0[1:0]	DACx_OUT0 锯齿波复位触发选择。 00: 来自 TRIGSEL 的外部触发 DACx_OUT0_EXTRIG 01: 软件触发 其他值：保留

19. 比较器（CMP）

19.1. 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。

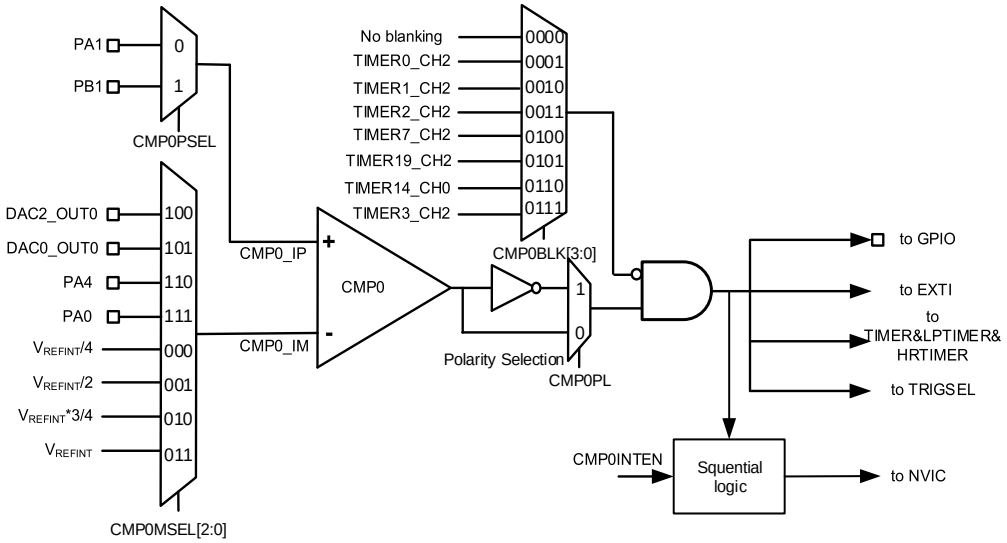
19.2. 主要特征

- 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；
- 系统复位时输出保持。

19.3. 功能描述

比较器的框图展示如下：

图 19-1.比较器框图



注意：

1) V_{REFINT} 是 1.2V。
2) CMPx(x=1,2..7)和 CMP0 结构相似，CMPxPSEL、CMPxBLK、CMPxMSEL 等参考*表 19-1. CMP 的输入和输出总结*和寄存器。

19.3.1. 比较器时钟

比较器与 APB 总线连接，时钟与 PCLK 同步。

19.3.2. 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。

比较器的输出可同时实现内部和外部输出。

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

比较器输出内部连接到定时器，他们的连接关系如下：

- CMP 输出连接到定时器中止功能（通过 TRIGSEL）。

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 19-1. CMP 的输入和输出总结*详细描述了 CMP 的输入和输出。.

表 19-1. CMP 的输入和输出总结

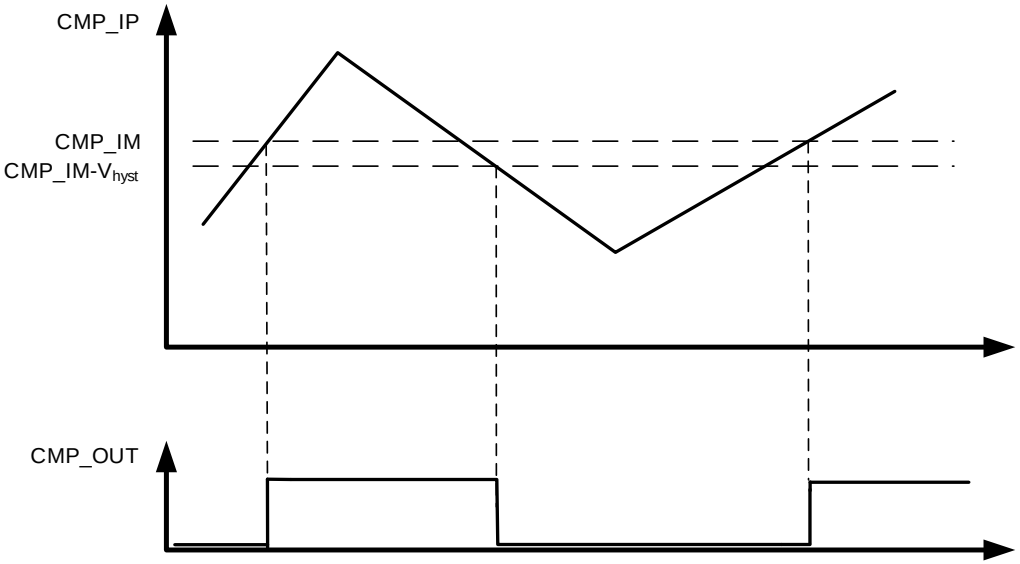
	CMP0	CMP1	CMP2	CMP3	CMP4	CMP5	CMP6	CMP7
CMP 同相	PA1	PA7	PA0	PB0	PB13	PB11	PB14	PC2

	CMP0	CMP1	CMP2	CMP3	CMP4	CMP5	CMP6	CMP7
输入连接到 I/O	PB1	PA3	PC1	PE7	PD12	PD11	PD14	PE9
CMP 反相输入连接到 I/O	PA4 PA0	PA5 PA2	PF1 PC0	PE8 PB2	PB10 PD13	PD10 PB15	PD15 PB12	PD8 PD9
CMP 反相输入连接到内部信号	$V_{REFINT}/4$, $V_{REFINT}/2$, $V_{REFINT}*3/4$, V_{REFINT}, DAC2_OUT0 DAC0_OUT0	$V_{REFINT}/4$, $V_{REFINT}/2$, $V_{REFINT}*3/4$, V_{REFINT}, DAC2_OUT1 DAC0_OUT1	$V_{REFINT}/4$, $V_{REFINT}/2$, $V_{REFINT}*3/4$, V_{REFINT}, DAC2_OUT0 DAC0_OUT0	$V_{REFINT}/4$, $V_{REFINT}/2$, $V_{REFINT}*3/4$, V_{REFINT}, DAC2_OUT1 DAC0_OUT0	$V_{REFINT}/4$, $V_{REFINT}/2$, $V_{REFINT}*3/4$, V_{REFINT}, DAC3_OUT0 DAC0_OUT1	$V_{REFINT}/4$, $V_{REFINT}/2$, $V_{REFINT}*3/4$, V_{REFINT}, DAC3_OUT1 DAC1_OUT0	$V_{REFINT}/4$, $V_{REFINT}/2$, $V_{REFINT}*3/4$, V_{REFINT}, DAC3_OUT0 DAC1_OUT0	$V_{REFINT}/4$, $V_{REFINT}/2$, $V_{REFINT}*3/4$, V_{REFINT}, DAC3_OUT1 DAC1_OUT1
CMP 输出连接到 I/O	PA0 PA6 PA11 PB8 PF4	PA2 PA7 PA12 PB9	PB7 PB15 PC2	PB1 PB6 PB14	PA9 PC7	PC6 PA10	PC8 PA8	PA13 PA14
CMP 输出连接到 EXTI	•							
CMP 输出连接到 TRIGSEL	•							
CMP 输出连接到 NVIC	•							
CMP 输出连接到内部信号	TIMER0, TIMER1, TIMER2, TIMER3, TIMER4, TIMER7, TIMER19, LPTIMER, HRTIMER							
CMP 输出连接到 break 信号	BREAK0(TIMER0, TIMER7, TIMER14, TIMER15, TIMER16, TIMER19)							
	BREAK1(TIMER0, TIMER7, TIMER19)							

19.3.3. 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 19-2. 比较器迟滞



19.3.4. 比较器寄存器写保护

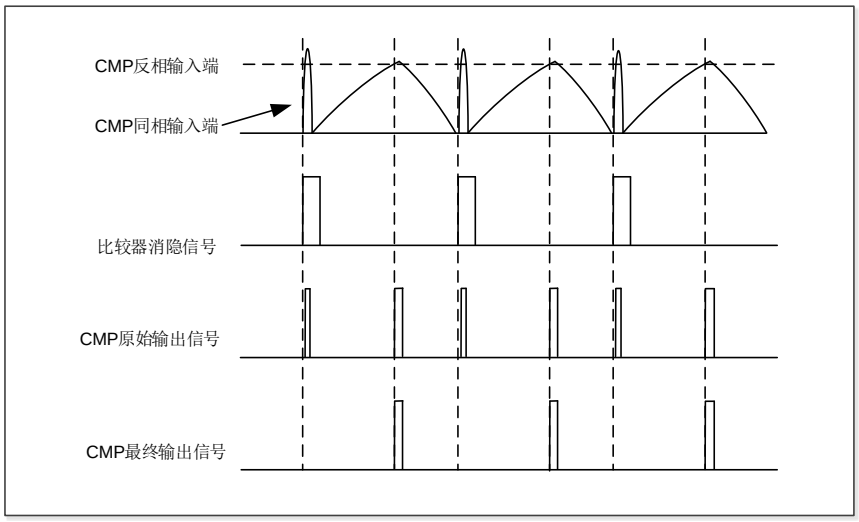
比较器的控制状态寄存器（CMPx_CS）可通过设置 CMPxLK 位为 1 来进行写保护，CMPx_CS 寄存器，包含 CMPxLK 位，就会变为只读位，只有在 MCU 复位时才可以复位。

19.3.5. 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 19-3. 比较器的输出消隐*显示了比较器的输出消隐功能。

图 19-3. 比较器的输出消隐



19.3.6. 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 CMPx 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。

19.3.7. 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。

CMP 还可以输出到 NVIC 产生中断。它是一个序列逻辑信号，因此需要 PCLK。

19.3.8. 复位保持功能

复位保持功能可以通过在 CMPx_CS 寄存器中设置 CMPxRSTMD 位来启用。除了上电复位外，输出通道将在所有复位过程中保持。

19.4. 比较器寄存器

CMP 基地址：0x4001 7C00

19.4.1. 状态寄存器（CMP_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								CMP7IF	CMP6IF	CMP5IF	CMP4IF	CMP3IF	CMP2IF	CMP1IF	CMP0IF
								r	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								CMP7O	CMP6O	CMP5O	CMP4O	CMP3O	CMP2O	CMP1O	CMP0O
								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	CMP7IF	CMP7中断标志位 0：无CMP7中断发生 1：CMP7中断发生 当CMP7输出置位时，该位由硬件置1。 软件写1至CMP_IFC寄存器CMP7IC位清0。
22	CMP6IF	CMP6中断标志位 0：无CMP6中断发生 1：CMP6中断发生 当CMP6输出置位时，该位由硬件置1。 软件写1至CMP_IFC寄存器CMP6IC位清0。
21	CMP5IF	CMP5中断标志位 0：无CMP5中断发生 1：CMP5中断发生 当CMP5输出置位时，该位由硬件置1。 软件写1至CMP_IFC寄存器CMP5IC位清0。
20	CMP4IF	CMP4中断标志位 0：无CMP4中断发生 1：CMP4中断发生 当CMP4输出置位时，该位由硬件置1。

		软件写1至CMP_IFC寄存器CMP4IC位清0。
19	CMP3IF	CMP3中断标志位 0：无CMP3中断发生 1：CMP3中断发生 当CMP3输出置位时，该位由硬件置1。 软件写1至CMP_IFC寄存器CMP3IC位清0。
18	CMP2IF	CMP2中断标志位 0：无CMP2中断发生 1：CMP2中断发生 当CMP2输出置位时，该位由硬件置1。 软件写1至CMP_IFC寄存器CMP2IC位清0。
17	CMP1IF	CMP1中断标志位 0：无CMP1中断发生 1：CMP1中断发生 当CMP1输出置位时，该位由硬件置1。 软件写1至CMP_IFC寄存器CMP1IC位清0。
16	CMP0IF	CMP0中断标志位 0：无CMP0中断发生 1：CMP0中断发生 当CMP0输出置位时，该位由硬件置1。 软件写1至CMP_IFC寄存器CMP0IC位清0。
15:8	保留	必须保持复位值。
7	CMP7O	CMP7 输出 该位反映 CMP7 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
6	CMP6O	CMP6 输出 该位反映 CMP6 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
5	CMP5O	CMP5 输出 该位反映 CMP5 输出状态，是只读位 0：同相输入端低于反相输入端，输出为低电平 1：同相输入端高于反相输入端，输出为高电平
4	CMP4O	CMP4 输出 该位反映 CMP4 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

3	CMP3O	CMP3 输出 该位反映 CMP3 输出状态，是只读位 0：同相输入端低于反相输入端，输出为低电平 1：同相输入端高于反相输入端，输出为高电平
2	CMP2O	CMP2 输出 该位反映 CMP2 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
1	CMP1O	CMP1 输出 该位反映 CMP1 输出状态，是只读位 0：同相输入端低于反相输入端，输出为低电平 1：同相输入端高于反相输入端，输出为高电平
0	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

19.4.2. 中断标志位清除寄存器（CMP_IFC）

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								CMP7IC	CMP6IC	CMP5IC	CMP4IC	CMP3IC	CMP2IC	CMP1IC	CMP0IC
								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															

位/位域	名称	描述
31:24	保留	必须保持复位值。
23	CMP7IC	CMP7中断标志位清除 0：无清除中断标志位发生 1：清除中断标志位

22	CMP6IC	CMP6中断标志位清除 0：无清除中断标志位发生 1：清除中断标志位
21	CMP5IC	CMP5中断标志位清除 0：无清除中断标志位发生 1：清除中断标志位
20	CMP4IC	CMP4中断标志位清除 0：无清除中断标志位发生 1：清除中断标志位
19	CMP3IC	CMP3中断标志位清除 0：无清除中断标志位发生 1：清除中断标志位
18	CMP2IC	CMP2中断标志位清除 0：无清除中断标志位发生 1：清除中断标志位
17	CMP1IC	CMP1中断标志位清除 0：无清除中断标志位发生 1：清除中断标志位
16	CMP0IC	CMP0中断标志位清除 0：无清除中断标志位发生 1：清除中断标志位
15:0	保留	必须保持复位值。

19.4.3. CMP0 控制状态寄存器（CMP0_CS）

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
CMP0LK	保留			CMP0BLK[3:0]				保留			CMP0PSEL	保留	CMP0MSEL[2:0]		
rw				rw							rw		rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					CMP0HST[2:0]			CMP0RSTMD	CMP0INTEN	保留		CMP0PL	CMP0SEN	CMP0BEN	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。该位只可通过软件置位一次，通过系统复位清除。 0：CMP0_CS 是可读可写位 1：CMP0_CS 是只读位
30:28	保留	必须保持复位值。
27:24	CMP0BLK[3:0]	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 0 的输出消隐。 0000：无消隐 0001：选择 TIMER0_CH2 输出比较信号为消隐源 0010：选择 TIMER1_CH2 输出比较信号为消隐源 0011：选择 TIMER2_CH2 输出比较信号为消隐源 0100：选择 TIMER7_CH2 输出比较信号为消隐源 0101：选择 TIMER19_CH2 输出比较信号为消隐源 0110：选择 TIMER14_CH0 输出比较信号为消隐源 0111：选择 TIMER3_CH2 输出比较信号为消隐源 其它：保留。
23:21	保留	必须保持复位值。
20	CMP0PSEL	CMP0_IP 输入选择 该位用于选择 CMP0 的 CMP0_IP 输入源。 0：PA1 1：PB1
19	保留	必须保持复位值。
18:16	CMP0MSEL[2:0]	CMP0_IM 输入选择 该位域用于选择 CMP0 的 CMP0_IM 输入源。 000：V_{REFINT} / 4 001：V_{REFINT} / 2 010：V_{REFINT} * 3 / 4 011：V_{REFINT} 100：DAC2_OUT0 101：DAC0_OUT0 110：PA4 111：PA0
15:11	保留	必须保持复位值。

10:8	CMP0HST[2:0]	CMP0 迟滞 该位域用于控制迟滞水平 000：无迟滞 001：10mv 010：20mv 011：30mv 100：40mv 101：50mv 110：60mv 111：70mv
7	CMP0RSTMD	CMP0 复位后模式，该位只能通过上电复位 0：正常模式。所有的复位将重置 CMP0 及其相关的寄存器。 1：复位保持模式。除上电复位外，CMP0 输出将在所有复位过程中保持。
6	CMP0INTEN	CMP0中断使能 0：禁能 1：使能
5:4	保留	必须保持复位值。
3	CMP0PL	CMP0 输出极性 该位用于控制 CMP0 输出极性 0：输出是正相的 1：输出是反相的
2	CMP0SEN	电压定标器使能位 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 CMP 反相输入端。 0：如果 CMP1_CS 的 CMP1SEN 位也复位，则禁用 V_{REFINT} 电压定标器 1：启用电压定标器
1	CMP0BEN	定标器使能位 0：如果 CMP1_CS 的 CMP1BEN 位也复位，则禁用定标器电阻桥 1：启用定标器电阻桥
0	CMP0EN	CMP0 使能 0：CMP0 禁能 1：CMP0 使能

19.4.4. CMP1 控制状态寄存器（CMP1_CS）

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
CMP1LK	保留			CMP1BLK[3:0]				保留			CMP1PSEL	保留	CMP1MSEL[2:0]		
rw				rw							rw		rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					CMP1HST[2:0]			CMP1RSTMD	CMP1INTEN	保留		CMP1PL	CMP1SEN	CMP1BEN	CMP1EN
					rw			rw	rw			rw	rw	rw	rw

位/位域	名称	描述
31	CMP1LK	CMP1 写保护 该位可将 CMP1 的各控制位设为只读。该位只可通过软件置位一次，通过系统复位清除。 0：CMP1_CS 是可读可写位 1：CMP1_CS 是只读位
30:28	保留	必须保持复位值。
27:24	CMP1BLK[3:0]	CMP1 输出消隐源 该位域用于选择哪个定时器输出控制比较器 0 的输出消隐。 0000：无消隐 0001：选择 TIMER0_CH2 输出比较信号为消隐源 0010：选择 TIMER1_CH2 输出比较信号为消隐源 0011：选择 TIMER2_CH2 输出比较信号为消隐源 0100：选择 TIMER7_CH2 输出比较信号为消隐源 0101：选择 TIMER19_CH2 输出比较信号为消隐源 0110：选择 TIMER14_CH0 输出比较信号为消隐源 0111：选择 TIMER3_CH2 输出比较信号为消隐源 其它：保留。
23:21	保留	必须保持复位值。
20	CMP1PSEL	CMP1_IP 输入选择 该位用于选择 CMP1 的 CMP1_IP 输入源。 0：PA7 1：PA3
19	保留	必须保持复位值。
18:16	CMP1MSEL[2:0]	CMP1_IM 输入选择 该位域用于选择 CMP1 的 CMP1_IM 输入源。 000：V_{REFINT} / 4

		001：V_{REFINT} / 2 010：V_{REFINT} * 3 / 4 011：V_{REFINT} 100：DAC2_OUT1 101：DAC0_OUT1 110：PA5 111：PA2
15:11	保留	必须保持复位值。
10:8	CMP1HST[2:0]	CMP1 迟滞 该位域用于控制迟滞水平 000：无迟滞 001：10mv 010：20mv 011：30mv 100：40mv 101：50mv 110：60mv 111：70mv
7	CMP1RSTMD	CMP1 复位后模式，该位只能通过上电复位 0：正常模式。所有的复位将重置 CMP1 及其相关的寄存器。 1：复位保持模式。除上电复位外，CMP1 输出将在所有复位过程中保持。
6	CMP1INTEN	CMP1中断使能 0：禁能 1：使能
5:4	保留	必须保持复位值。
3	CMP1PL	CMP1 输出极性 该位用于控制 CMP1 输出极性 0：输出是正相的 1：输出是反相的
2	CMP1SEN	电压定标器使能位 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 CMP 反相输入端。 0：如果 CMP0_CS 的 CMP0SEN 位也复位，则禁用 V_{REFINT} 电压定标器 1：启用电压定标器
1	CMP1BEN	定标器使能位 0：如果 CMP0_CS 的 CMP0BEN 位也复位，则禁用定标器电阻桥

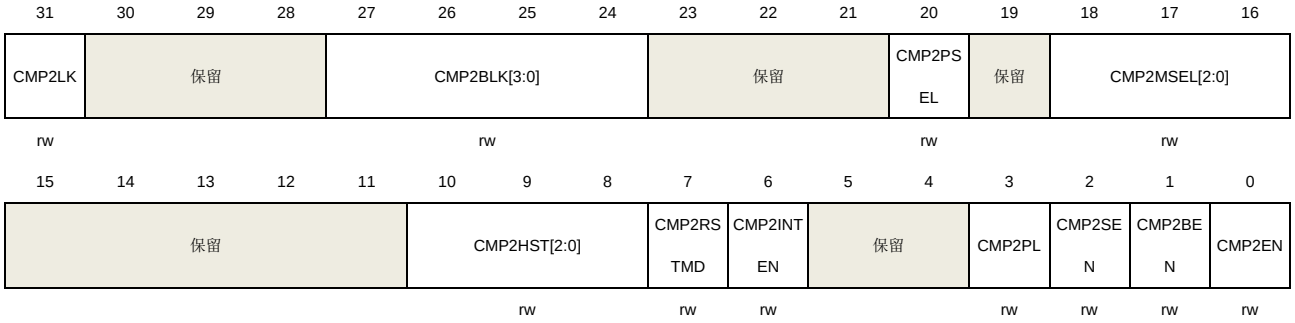
		1：启用定标器电阻桥
0	CMP1EN	CMP1 使能 0：CMP1 禁能 1：CMP1 使能

19.4.5. CMP2 控制状态寄存器（CMP2_CS）

地址偏移：0x10

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31	CMP2LK	CMP2 写保护 该位可将 CMP2 的各控制位设为只读。该位只可通过软件置位一次，通过系统复位清除。 0：CMP2_CS 是可读可写位 1：CMP2_CS 是只读位
30:28	保留	必须保持复位值。
27:24	CMP2BLK[3:0]	CMP2 输出消隐源 该位域用于选择哪个定时器输出控制比较器 0 的输出消隐。 0000：无消隐 0001：选择 TIMER0_CH2 输出比较信号为消隐源 0010：选择 TIMER2_CH2 输出比较信号为消隐源 0011：选择 TIMER1_CH3 输出比较信号为消隐源 0100：选择 TIMER7_CH2 输出比较信号为消隐源 0101：选择 TIMER19_CH2 输出比较信号为消隐源 0110：选择 TIMER14_CH0 输出比较信号为消隐源 0111：选择 TIMER3_CH2 输出比较信号为消隐源 其它：保留。

23:21	保留	必须保持复位值。
20	CMP2PSEL	CMP2_IP 输入选择 该位用于选择 CMP2 的 CMP2_IP 输入源。 0：PA0 1：PC1
19	保留	必须保持复位值。
18:16	CMP2MSEL[2:0]	CMP2_IM 输入选择 该位域用于选择 CMP2 的 CMP2_IM 输入源。 000：V_{REFINT} / 4 001：V_{REFINT} / 2 010：V_{REFINT} * 3 / 4 011：V_{REFINT} 100：DAC2_OUT0 101：DAC0_OUT0 110：PF1 111：PC0
15:11	保留	必须保持复位值。
10:8	CMP2HST[2:0]	CMP2 迟滞 该位域用于控制迟滞水平 000：无迟滞 001：10mv 010：20mv 011：30mv 100：40mv 101：50mv 110：60mv 111：70mv
7	CMP2RSTMD	CMP2 复位后模式，该位只能通过上电复位 0：正常模式。所有的复位将重置 CMP2 及其相关的寄存器。 1：复位保持模式。除上电复位外，CMP2 输出将在所有复位过程中保持。
6	CMP2INTEN	CMP2中断使能 0：禁能 1：使能
5:4	保留	必须保持复位值。
3	CMP2PL	CMP2 输出极性 该位用于控制 CMP2 输出极性

0：输出是正相的
1：输出是反相的

2	CMP2SEN	电压定标器使能位 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 CMP 反相输入端。 0：如果 CMP3_CS 的 CMP3SEN 位也复位，则禁用 V_{REFINT} 电压定标器 1：启用电压定标器
1	CMP2BEN	定标器使能位 0：如果 CMP3_CS 的 CMP3BEN 位也复位，则禁用定标器电阻桥 1：启用定标器电阻桥
0	CMP2EN	CMP2 使能 0：CMP2 禁能 1：CMP2 使能

19.4.6. CMP3 控制状态寄存器（CMP3_CS）

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
CMP3LK	保留			CMP3BLK[3:0]				保留			CMP3PSEL	保留	CMP3MSEL[2:0]		
rw				rw							rw		rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					CMP3HST[2:0]			CMP3RSTMD	CMP3INTEN	保留		CMP3PL	CMP3SEN	CMP3BEN	CMP3EN
					rw			rw	rw			rw	rw	rw	rw

位/位域	名称	描述
31	CMP3LK	CMP3 写保护 该位可将 CMP3 的各控制位设为只读。该位只可通过软件置位一次，通过系统复位清除。 0：CMP3_CS 是可读可写位 1：CMP3_CS 是只读位
30:28	保留	必须保持复位值。
27:24	CMP3BLK[3:0]	CMP3 输出消隐源 该位域用于选择哪个定时器输出控制比较器 0 的输出消隐。

		0000：无消隐 0001：选择 TIMER2_CH3 输出比较信号为消隐源 0010：选择 TIMER7_CH2 输出比较信号为消隐源 0011：选择 TIMER14_CH1 输出比较信号为消隐源 0100：选择 TIMER0_CH2 输出比较信号为消隐源 0101：选择 TIMER19_CH2 输出比较信号为消隐源 0110：选择 TIMER14_CH0 输出比较信号为消隐源 0111：选择 TIMER3_CH2 输出比较信号为消隐源 其它：保留。
23:21	保留	必须保持复位值。
20	CMP3PSEL	CMP3_IP 输入选择 该位用于选择 CMP3 的 CMP3_IP 输入源。 0：PB0 1：PE7
19	保留	必须保持复位值。
18:16	CMP3MSEL[2:0]	CMP3_IM 输入选择 该位域用于选择 CMP3 的 CMP3_IM 输入源。 000：V_{REFINT} / 4 001：V_{REFINT} / 2 010：V_{REFINT} * 3 / 4 011：V_{REFINT} 100：DAC2_OUT1 101：DAC0_OUT1 110：PE8 111：PB2
15:11	保留	必须保持复位值。
10:8	CMP3HST[2:0]	CMP3 迟滞 该位域用于控制迟滞水平 000：无迟滞 001：10mv 010：20mv 011：30mv 100：40mv 101：50mv 110：60mv 111：70mv
7	CMP3RSTMD	CMP3 复位后模式，该位只能通过上电复位

0：正常模式。所有的复位将重置 CMP3 及其相关的寄存器。
1：复位保持模式。除上电复位外，CMP3 输出将在所有复位过程中保持。

6	CMP3INTEN	CMP3中断使能 0：禁能 1：使能
5:4	保留	必须保持复位值。
3	CMP3PL	CMP3 输出极性 该位用于控制 CMP3 输出极性 0：输出是正相的 1：输出是反相的
2	CMP3SEN	电压定标器使能位 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 CMP 反相输入端。 0：如果 CMP2_CS 的 CMP2SEN 位也复位，则禁用 V_{REFINT} 电压定标器 1：启用电压定标器
1	CMP3BEN	定标器使能位 0：如果 CMP2_CS 的 CMP2BEN 位也复位，则禁用定标器电阻桥 1：启用定标器电阻桥
0	CMP3EN	CMP3 使能 0：CMP3 禁能 1：CMP3 使能

19.4.7. CMP4 控制状态寄存器（CMP4_CS）

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
CMP4LK	保留			CMP4BLK[3:0]				保留			CMP4PSEL	保留	CMP4MSEL[2:0]		
rw				rw							rw		rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					CMP4HST[2:0]			CMP4RSTMD	CMP4INTEN	保留		CMP4PL	CMP4SEN	CMP4BEN	CMP4EN
					rw			rw	rw			rw	rw	rw	rw

位/位域	名称	描述
31	CMP4LK	CMP4 写保护 该位可将 CMP4 的各控制位设为只读。该位只可通过软件置位一次，通过系统复位清除。 0：CMP4_CS 是可读可写位 1：CMP4_CS 是只读位
30:28	保留	必须保持复位值。
27:24	CMP4BLK[3:0]	CMP4 输出消隐源 该位域用于选择哪个定时器输出控制比较器 0 的输出消隐。 0000：无消隐 0001：选择 TIMER1_CH2 输出比较信号为消隐源 0010：选择 TIMER7_CH2 输出比较信号为消隐源 0011：选择 TIMER2_CH2 输出比较信号为消隐源 0100：选择 TIMER0_CH2 输出比较信号为消隐源 0101：选择 TIMER19_CH2 输出比较信号为消隐源 0110：选择 TIMER14_CH0 输出比较信号为消隐源 0111：选择 TIMER3_CH2 输出比较信号为消隐源 其它：保留。
23:21	保留	必须保持复位值。
20	CMP4PSEL	CMP4_IP 输入选择 该位用于选择 CMP4 的 CMP4_IP 输入源。 0：PB13 1：PD12
19	保留	必须保持复位值。
18:16	CMP4MSEL[2:0]	CMP4_IM 输入选择 该位域用于选择 CMP4 的 CMP4_IM 输入源。 000：V_{REFINT} / 4 001：V_{REFINT} / 2 010：V_{REFINT} * 3 / 4 011：V_{REFINT} 100：DAC3_OUT0 101：DAC0_OUT1 110：PB10 111：PD13
15:11	保留	必须保持复位值。
10:8	CMP4HST[2:0]	CMP4 迟滞

		该位域用于控制迟滞水平 000：无迟滞 001：10mv 010：20mv 011：30mv 100：40mv 101：50mv 110：60mv 111：70mv
7	CMP4RSTMD	CMP4 复位后模式，该位只能通过上电复位 0：正常模式。所有的复位将重置 CMP4 及其相关的寄存器。 1：复位保持模式。除上电复位外，CMP4 输出将在所有复位过程中保持。
6	CMP4INTEN	CMP4中断使能 0：禁能 1：使能
5:4	保留	必须保持复位值。
3	CMP4PL	CMP4 输出极性 该位用于控制 CMP4 输出极性 0：输出是正相的 1：输出是反相的
2	CMP4SEN	电压定标器使能位 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 CMP 反相输入端。 0：如果 CMP5_CS 的 CMP5SEN 位也复位，则禁用 V_{REFINT} 电压定标器 1：启用电压定标器
1	CMP4BEN	定标器使能位 0：如果 CMP5_CS 的 CMP5BEN 位也复位，则禁用定标器电阻桥 1：启用定标器电阻桥
0	CMP4EN	CMP4 使能 0：CMP4 禁能 1：CMP4 使能

19.4.8. CMP5 控制状态寄存器（CMP5_CS）

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
CMP5LK	保留				CMP5BLK[3:0]			保留			CMP5PSEL	保留	CMP5MSEL[2:0]		
rw					rw						rw		rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					CMP5HST[2:0]			CMP5RSTMD	CMP5INTEN	保留		CMP5PL	CMP5SEN	CMP5BEN	CMP5EN
					rw			rw	rw			rw	rw	rw	rw

位/位域	名称	描述
31	CMP5LK	CMP5 写保护 该位可将 CMP5 的各控制位设为只读。该位只可通过软件置位一次，通过系统复位清除。 0：CMP5_CS 是可读可写位 1：CMP5_CS 是只读位
30:28	保留	必须保持复位值。
27:24	CMP5BLK[3:0]	CMP5 输出消隐源 该位域用于选择哪个定时器输出控制比较器 0 的输出消隐。 0000：无消隐 0001：选择 TIMER7_CH2 输出比较信号为消隐源 0010：选择 TIMER1_CH3 输出比较信号为消隐源 0011：选择 TIMER14_CH1 输出比较信号为消隐源 0100：选择 TIMER0_CH2 输出比较信号为消隐源 0101：选择 TIMER19_CH2 输出比较信号为消隐源 0110：选择 TIMER14_CH0 输出比较信号为消隐源 0111：选择 TIMER3_CH2 输出比较信号为消隐源 其它：保留。
23:21	保留	必须保持复位值。
20	CMP5PSEL	CMP5_IP 输入选择 该位用于选择 CMP5 的 CMP5_IP 输入源。 0：PB11 1：PD11
19	保留	必须保持复位值。
18:16	CMP5MSEL[2:0]	CMP5_IM 输入选择 该位域用于选择 CMP5 的 CMP5_IM 输入源。 000：V_{REFINT} / 4

		001：V_{REFINT} / 2 010：V_{REFINT} * 3 / 4 011：V_{REFINT} 100：DAC3_OUT1 101：DAC1_OUT0 110：PD10 111：PB15
15:11	保留	必须保持复位值。
10:8	CMP5HST[2:0]	CMP5 迟滞 该位域用于控制迟滞水平 000：无迟滞 001：10mv 010：20mv 011：30mv 100：40mv 101：50mv 110：60mv 111：70mv
7	CMP5RSTMD	CMP5 复位后模式，该位只能通过上电复位 0：正常模式。所有的复位将重置 CMP5 及其相关的寄存器。 1：复位保持模式。除上电复位外，CMP5 输出将在所有复位过程中保持。
6	CMP5INTEN	CMP5中断使能 0：禁能 1：使能
5:4	保留	必须保持复位值。
3	CMP5PL	CMP5 输出极性 该位用于控制 CMP5 输出极性 0：输出是正相的 1：输出是反相的
2	CMP5SEN	电压定标器使能位 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 CMP 反相输入端。 0：如果 CMP4_CS 的 CMP4SEN 位也复位，则禁用 V_{REFINT} 电压定标器 1：启用电压定标器
1	CMP5BEN	定标器使能位 0：如果 CMP4_CS 的 CMP4BEN 位也复位，则禁用定标器电阻桥

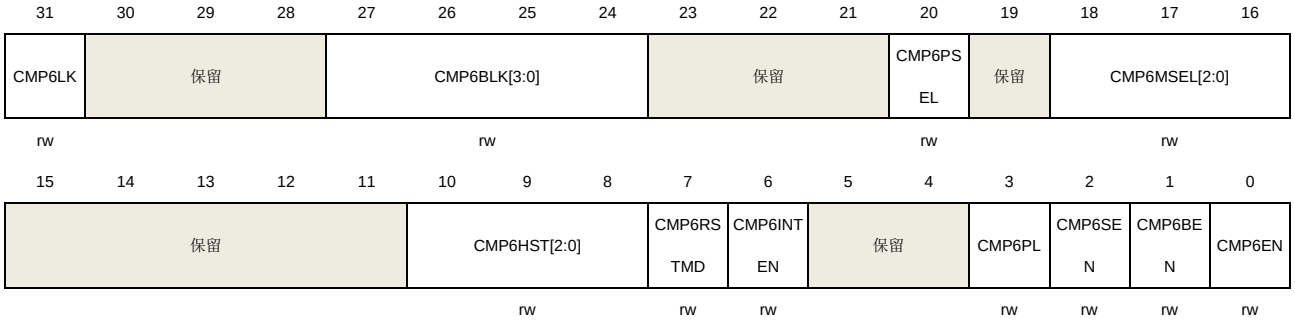
		1：启用定标器电阻桥
0	CMP5EN	CMP5 使能 0：CMP5 禁能 1：CMP5 使能

19.4.9. CMP6 控制状态寄存器（CMP6_CS）

地址偏移：0x20

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31	CMP6LK	CMP6 写保护 该位可将 CMP6 的各控制位设为只读。该位只可通过软件置位一次，通过系统复位清除。 0：CMP6_CS 是可读可写位 1：CMP6_CS 是只读位
30:28	保留	必须保持复位值。
27:24	CMP6BLK[3:0]	CMP6 输出消隐源 该位域用于选择哪个定时器输出控制比较器 0 的输出消隐。 0000：无消隐 0001：选择 TIMER0_CH2 输出比较信号为消隐源 0010：选择 TIMER7_CH2 输出比较信号为消隐源 0011：选择 TIMER2_CH2 输出比较信号为消隐源 0100：选择 TIMER14_CH1 输出比较信号为消隐源 0101：选择 TIMER19_CH2 输出比较信号为消隐源 0110：选择 TIMER14_CH0 输出比较信号为消隐源 0111：选择 TIMER3_CH2 输出比较信号为消隐源 其它：保留。

23:21	保留	必须保持复位值。
20	CMP6PSEL	CMP6_IP 输入选择 该位用于选择 CMP6 的 CMP6_IP 输入源。 0：PB14 1：PD14
19	保留	必须保持复位值。
18:16	CMP6MSEL[2:0]	CMP6_IM 输入选择 该位域用于选择 CMP6 的 CMP6_IM 输入源。 000：V_{REFINT} / 4 001：V_{REFINT} / 2 010：V_{REFINT} * 3 / 4 011：V_{REFINT} 100：DAC3_OUT0 101：DAC1_OUT0 110：PD15 111：PB12
15:11	保留	必须保持复位值。
10:8	CMP6HST[2:0]	CMP6 迟滞 该位域用于控制迟滞水平 000：无迟滞 001：10mv 010：20mv 011：30mv 100：40mv 101：50mv 110：60mv 111：70mv
7	CMP6RSTMD	CMP6 复位后模式，该位只能通过上电复位 0：正常模式。所有的复位将重置 CMP6 及其相关的寄存器。 1：复位保持模式。除上电复位外，CMP6 输出将在所有复位过程中保持。
6	CMP6INTEN	CMP6中断使能 0：禁能 1：使能
5:4	保留	必须保持复位值。
3	CMP6PL	CMP6 输出极性 该位用于控制 CMP6 输出极性

0：输出是正相的
1：输出是反相的

2	CMP6SEN	电压定标器使能位 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 CMP 反相输入端。 0：如果 CMP7_CS 的 CMP7SEN 位也复位，则禁用 V_{REFINT} 电压定标器 1：启用电压定标器
1	CMP6BEN	定标器使能位 0：如果 CMP7_CS 的 CMP7BEN 位也复位，则禁用定标器电阻桥 1：启用定标器电阻桥
0	CMP6EN	CMP6 使能 0：CMP6 禁能 1：CMP6 使能

19.4.10. CMP7 控制状态寄存器（CMP7_CS）

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
CMP7LK	保留			CMP7BLK[3:0]				保留			CMP7PSEL	保留	CMP7MSEL[2:0]		
rw				rw							rw		rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					CMP7HST[2:0]			CMP7RSTMD	CMP7INTEN	保留		CMP7PL	CMP7SEN	CMP7BEN	CMP7EN
					rw			rw	rw			rw	rw	rw	rw

位/位域	名称	描述
31	CMP7LK	CMP7 写保护 该位可将 CMP7 的各控制位设为只读。该位只可通过软件置位一次，通过系统复位清除。 0：CMP7_CS 是可读可写位 1：CMP7_CS 是只读位
30:28	保留	必须保持复位值。
27:24	CMP7BLK[3:0]	CMP7 输出消隐源 该位域用于选择哪个定时器输出控制比较器 0 的输出消隐。

		0000：无消隐 0001：选择 TIMER1_CH2 输出比较信号为消隐源 0010：选择 TIMER7_CH2 输出比较信号为消隐源 0011：选择 TIMER2_CH2 输出比较信号为消隐源 0100：选择 TIMER0_CH2 输出比较信号为消隐源 0101：选择 TIMER19_CH2 输出比较信号为消隐源 0110：选择 TIMER14_CH0 输出比较信号为消隐源 0111：选择 TIMER3_CH2 输出比较信号为消隐源 其它：保留。
23:21	保留	必须保持复位值。
20	CMP7PSEL	CMP7_IP 输入选择 该位用于选择 CMP7 的 CMP7_IP 输入源。 0：PC2 1：PE9
19	保留	必须保持复位值。
18:16	CMP7MSEL[2:0]	CMP7_IM 输入选择 该位域用于选择 CMP7 的 CMP7_IM 输入源。 000：V_{REFINT} / 4 001：V_{REFINT} / 2 010：V_{REFINT} * 3 / 4 011：V_{REFINT} 100：DAC3_OUT1 101：DAC1_OUT1 110：PD8 111：PD9
15:11	保留	必须保持复位值。
10:8	CMP7HST[2:0]	CMP7 迟滞 该位域用于控制迟滞水平 000：无迟滞 001：10mv 010：20mv 011：30mv 100：40mv 101：50mv 110：60mv 111：70mv
7	CMP7RSTMD	CMP7 复位后模式，该位只能通过上电复位

		0：正常模式。所有的复位将重置 CMP7 及其相关的寄存器。 1：复位保持模式。除上电复位外，CMP7 输出将在所有复位过程中保持。
6	CMP7INTEN	CMP7中断使能 0：禁能 1：使能
5:4	保留	必须保持复位值。
3	CMP7PL	CMP7 输出极性 该位用于控制 CMP7 输出极性 0：输出是正相的 1：输出是反相的
2	CMP7SEN	电压定标器使能位 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 CMP 反相输入端。 0：如果 CMP6_CS 的 CMP6SEN 位也复位，则禁用 V_{REFINT} 电压定标器 1：启用电压定标器
1	CMP7BEN	定标器使能位 0：如果 CMP6_CS 的 CMP6BEN 位也复位，则禁用定标器电阻桥 1：启用定标器电阻桥
0	CMP7EN	CMP7 使能 0：CMP7 禁能 1：CMP7 使能

20. VREF

20.1. 简介

MCU 有一个精准的内部参考电路，用于为 ADC/DAC 提供基准电压，或供连接到 VREFP 引脚的片外电路使用。

20.2. 主要特性

内部参考电压特性描述如下：

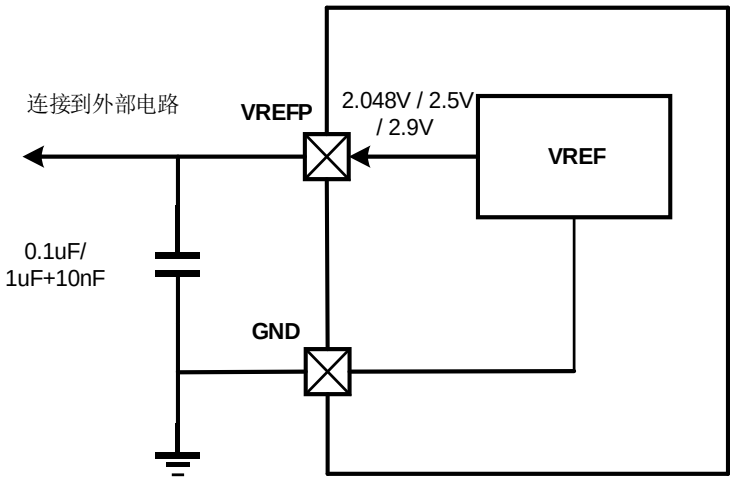
- 电压稳定，产品经过校准；
- 连接 VREFP 引脚可供片外电路使用；
- 2.048V、2.5V 或者 2.9V 可配置的参考电压输出。

20.3. 功能说明

通过将 VREF_CS 寄存器中的 VREFEN 位置 1 使能 VREF 模块，配置 VREFS[1:0]位可以输出 2.048V、2.5V、或者 2.9V 的参考电压。当 VREF 被使能时，复位 HIPM 位，可将内部参考电压输出连接到 VREFP 引脚上。当 VREF 失能时，置位 HIPM 位，可将片外参考电压注入到 VREFP 引脚作为 ADC/DAC 的参考电压。如果没有 VREFP 引脚（请参阅数据手册），则 VREFP 被内部连接到 VDDA，且 VREFEN 位必须保持为 0。

当使用精准的内部参考电压时，建议连接一个 0.1uF（或 1uF 和 10nF 并联）的旁路电容，并接地。

图 20-1. VREF 连接



如下*表 20-1. VREF 模式*所示，根据 VREF_CS 寄存器中 VREFEN 和 HIPM 位的配置，内部参考电压单元可以被配置成四种不同的模式。

表 20-1. VREF 模式

VREFEN	HIPM	模式
0	0	VREF 失能， VREFP 引脚下拉到 VSSA。
0	1	外部参考电压模式（默认）： VREF 失能， VREFP 引脚是输入模式。
1	0	内部参考电压模式： VREF 使能， VREFP 引脚连接到 VREF 输出。
1	1	保持模式： VREF 失能， VREFP 引脚浮空，通过外部电容保持电压。 失能 VREFRDY 位检测，VREFRDY 位保持最后一个状态。

当 VREF_CS 寄存器中 VREFEN 位置 1 且 HIPM 位复位时，即 VREF 工作在内部参考电压模式时，用户必须等待一段时间直到 VREFRDY 位被置位，表明 VREF 输出已经达到了要求的数值。

20.3.1. VREF 校准

内部参考电压在生产测试的过程中已经经过了校准，每个电压档位（2.048V，2.5V 和 2.9V，由 VREFS[1:0]位域控制）都有相应的校准值，保存在 Flash 内存中。复位时，或者所选的参考电压改变时，VREF_CALIB 寄存器中的 VREFCAL[5:0]位域将被自动初始化为相应的校准值。

写入 VREFCAL[5:0]位域可以调整 VREF 的输出电压。写入这些位后，如果所选的参考电压改变，VREF 校准值将不会被自动初始化直到 MCU 复位。

20.4. VREF 寄存器

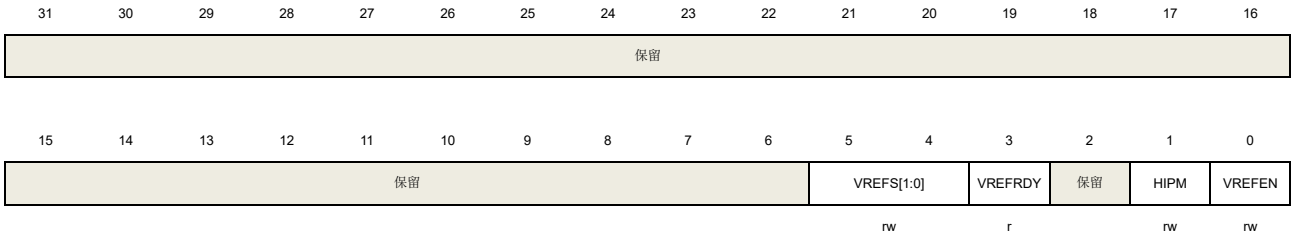
VREF 基地址：0x4001 7800

当 DACx（x=0,1,2,3）中任一个的 DRSTMDy（y=0,1）位置 1，VREF 寄存器将在除 POR 以外的所有复位事件中保持。

20.4.1. 控制状态寄存器（VREF_CS）

地址偏移：0x00
复位值：0x0000 0002

该寄存器可以按半字（16 位）或字（32 位）访问。



位/位域	名称	描述
31:6	保留	必须保持复位值。
5:4	VREFS[1:0]	参考电压选择 这些位定义了VREF产生的参考电压的数值。 00：参考电压在2.048V左右 01：参考电压在2.5V左右 10：参考电压在2.9V左右 11：保留 此位只有在 VREF 失能（VREFEN 位为 0）的时候可以被更改。
3	VREFRDY	VREF就绪 0：VREF输出未就绪 1：VREF 输出就绪
2	保留	必须保持复位值。
1	HIPM	高阻抗模式 0：VREFP 引脚内部连接到 VREF 输出 1：VREFP 引脚为高阻抗模式
0	VREFEN	VREF使能 0：VREF失能

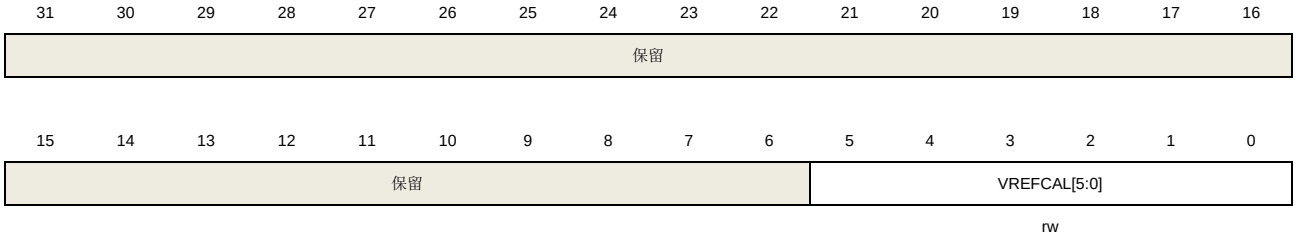
1：VREF 使能

20.4.2. 校准寄存器（VREF_CALIB）

地址偏移：0x04

复位值：0x0000 00XX

该寄存器可以按半字（16 位）或字（32 位）访问。



位/位域	名称	描述
31:6	保留	必须保持复位值。
5:0	VREFCAL	VREF校准值 复位时，或者所选的参考电压改变时（由VREFS[1:0]位域控制）这些位将被自动初始化为在生产测试期间存储在Flash中的校准值。写入这些位也可调节内部VREF电压。写入这些位后，如果所选的参考电压改变，VREF校准值将不会被自动初始化知道MCU复位。 **注意：**如果用户执行校准程序，则校准值必须从0x00递增到0x3F。

21. 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了预设的门限时，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。

21.1. 独立看门狗定时器（FWDGT）

21.1.1. 简介

独立看门狗定时器（FWDGT）有独立时钟源（IRC32K）。因此，即使主时钟失效了，它仍然能保持工作状态，这非常适合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，刷新计数器，独立看门狗会产生一个复位。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。

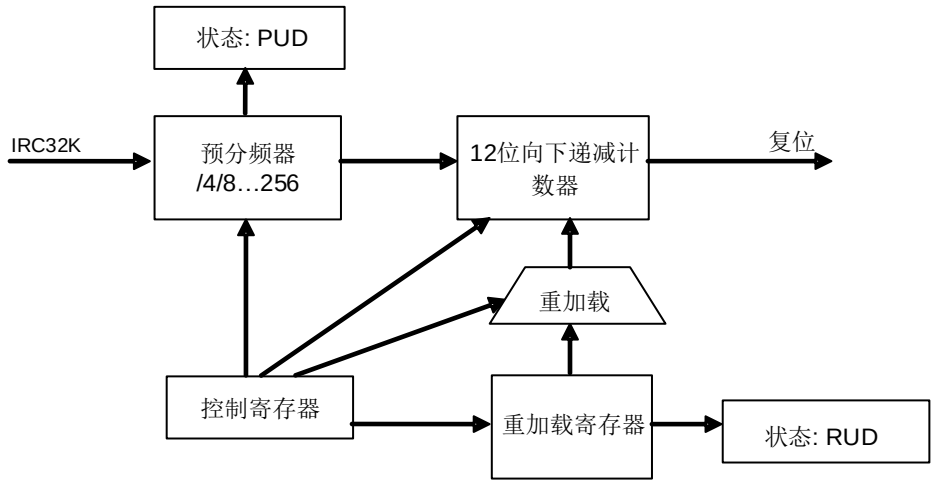
21.1.2. 主要特征

- 独立运行的12位向下计数器。
- 如果看门狗定时器被使能，有以下两种情况下会产生复位：
  - 当计数器到0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。
- 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，可以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。
- 通过配置FWDGSPD_STDBY或FWDGSPD_DPSLP，在待机模式或深度睡眠模式中，FWDGT可以停止工作或唤醒控制器继续工作。

21.1.3. 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21-1. 独立看门狗定时器框图*的独立看门狗定时器的功能模块。

图 21-1. 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，产生一次复位。

在任何时候向控制寄存器（FWDGT_CTL）中写0xAAAA都可以重装载计数器，重装载值来源于FWDGT_RLD寄存器。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FWDGT_WND寄存器中设置适当的值即可。如果重加载操作执行的同时，看门狗定时器计数器的值大于窗口寄存器（FWDGT_WND）中存储的值，也会引起系统复位。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。为了避免复位，软件应该在计数器达到0x000之前重装载计数器。

FWDGT_PSC寄存器和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到控制寄存器（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。当预分频寄存器（FWDGT_PSC）或者重装载寄存器（FWDGT_RLD）更新时，FWDGT_STAT寄存器的状态位会被置1。

如果在MCU调试模块中的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位被置1，独立看门狗定时器将在调试模式下停止工作。

表 21-1. 独立看门狗定时器在 32kHz (IRC32K)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	0.125	512
1/8	001	0.25	1024

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
1/16	010	0.5	2048
1/32	011	1.0	4096
1/64	100	2.0	8192
1/128	101	4.0	16384
1/256	110或111	8.0	32768

通过IRC32K校准可以使独立看门狗定时器超时更加精确。

注意：

1. 当执行完喂狗reload操作之后，如需要立即进入deepsleep/standby模式时，必须通过软件设置，在reload命令及deepsleep/standby模式命令中间插入（3个以上）IRC32K时钟间隔。
2. 两次连续喂狗之间需插入7个及以上IRC32K时钟间隔。

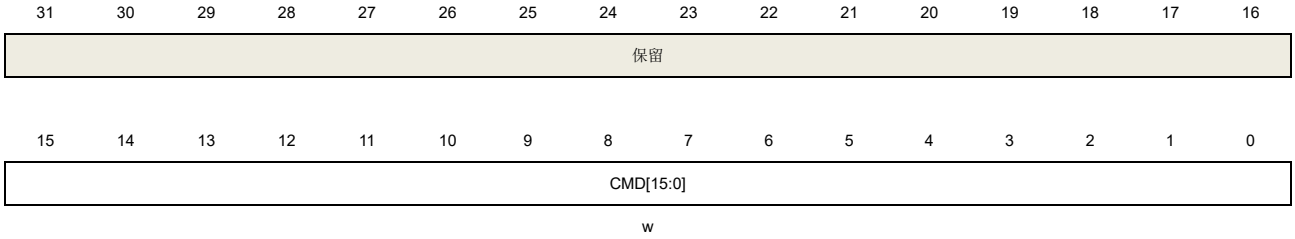
21.1.4. FWDGT 寄存器

FWDGT 基地址：0x4000 3000

控制寄存器（FWDGT_CTL）

地址偏移：0x00
复位值：0x0000 0000

该寄存器可以按半字（16 位）或字（32 位）访问。

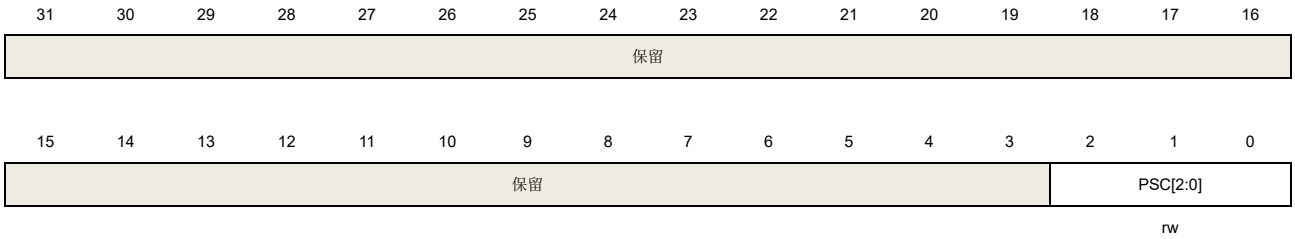


位/位域	名称	说明
31:16	保留	必须保持复位值。
15:0	CMD[15:0]	只可写，写入不同的值来产生不同的功能 0x5555：关闭FWDGT_PSC和FWDGT_RLD的写保护 0xCCCC：开启独立看门狗定时器计数器。计数减到0时产生中断 0xAAAA：重装载计数器

预分频寄存器（FWDGT_PSC）

地址偏移：0x04
复位值：0x0000 0000

该寄存器可以按半字（16 位）或字（32 位）访问。



位/位域	名称	说明
31:3	保留	必须保持复位值。
2:0	PSC[2:0]	独立看门狗定时器计时预分频选择。写这些位之前要通过向FWDGT_CTL寄存器写0x5555去除写保护。在改写这个寄存器的过程中，FWDGT_STAT寄存器的PUD位被

置1，此时读取此寄存器的值都是无效的。

000：1/4

001：1/8

010：1/16

011：1/32

100：1/64

101：1/128

110：1/256

111：1/256

如果应用需要使用几个预分频系数，改变预分频值之前必须等到PUD位被清0。更新了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。

重装载寄存器（FWDGT_RLD）

地址偏移：0x08

复位值：0x0000 0FFF

该寄存器可以按半字（16 位）或字（32 位）访问。

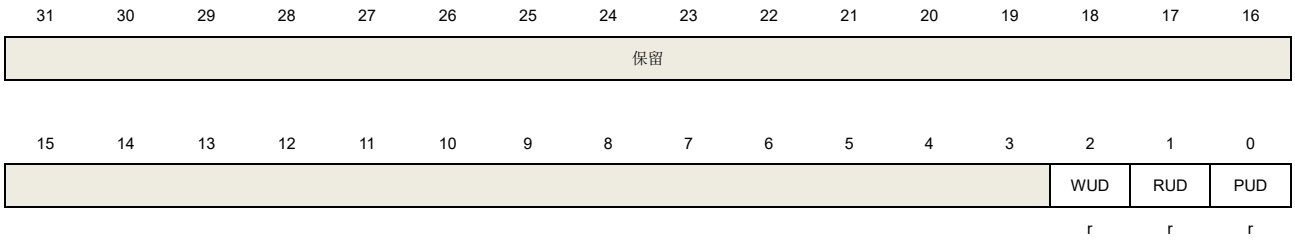
位/位域	名称	说明
31:12	保留	必须保持复位值。
11:0	RLD[11:0]	独立看门狗定时器计数器重装载值，向FWDGT_CTL寄存器写入0xAAAA的时候，这个值会被更新到看门狗定时器计数器中。 这些位有写保护功能。在写这些位之前需向FWDGT_CTL寄存器中写0x5555。在改写这个寄存器的过程中，FWDGT_STAT寄存器的RUD位被置1，从此寄存器中读取的任何值都是无效的。 如果应用需要使用几个重装载值，改变重加载值之前必须等到RUD位被清0。更新了重加载寄存器的值后，在代码持续执行之前不必等待RUD值被清零。

状态寄存器（FWDGT_STAT）

地址偏移：0x0C

复位值：0x0000 0000

该寄存器可以按半字（16 位）或字（32 位）访问。



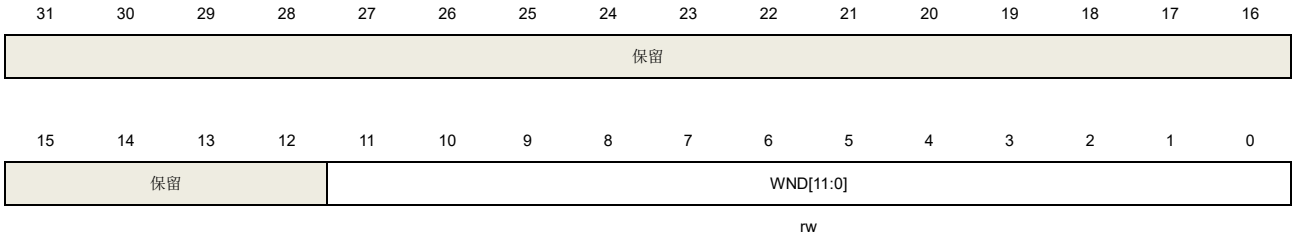
位/位域	名称	说明
31:3	保留	必须保持复位值。
2	WUD	独立看门狗定时器计数器窗口值更新 FWDGT_WND寄存器写操作时，该位被置1，此时读取FWDGT_WND寄存器的任何值都是无效的。
1	RUD	独立看门狗定时器计数器重装载值更新 FWDGT_RLD寄存器写操作时，该位被置1，此时读取FWDGT_RLD寄存器的任何值都是无效的。在FWDGT_RLD寄存器更新后，该位由硬件清零。
0	PUD	独立看门狗定时器预分频值更新 FWDGT_PSC寄存器写操作时，该位被置1，此时读取FWDGT_PSC寄存器的任何值都是无效的。在FWDGT_PSC寄存器更新后，该位由硬件清零。

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。



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WND[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。

21.2. 窗口看门狗定时器（WWDGT）

21.2.1. 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，都会产生一个提前唤醒标志，如果使能中断也将会产生中断。

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。

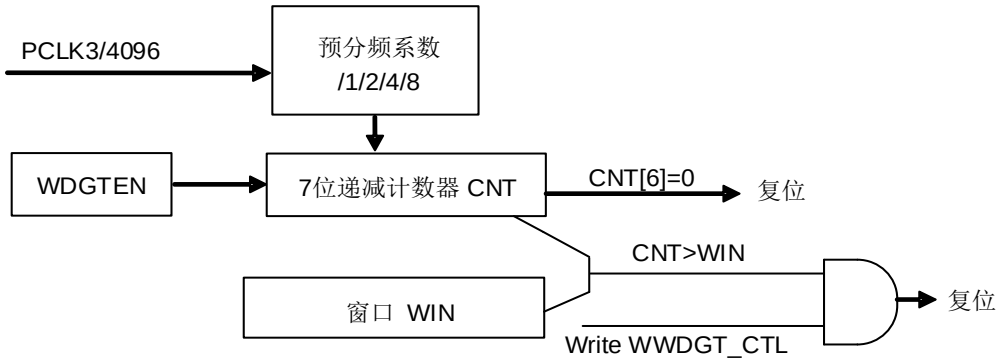
21.2.2. 主要特征

- 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）：如果看门狗定时器打开，支持中断，当计数值达到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。

21.2.3. 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的时候产生复位（CNT[6]位被清0），或者，在计数值达到窗口寄存器值之前，更新计数器也会产生复位。

图 21-2. 窗口看门狗定时器框图



上电复位之后看门狗定时器总是关闭的。软件可以向WWDGT_CTL的WDGTEN写1开启看门狗定

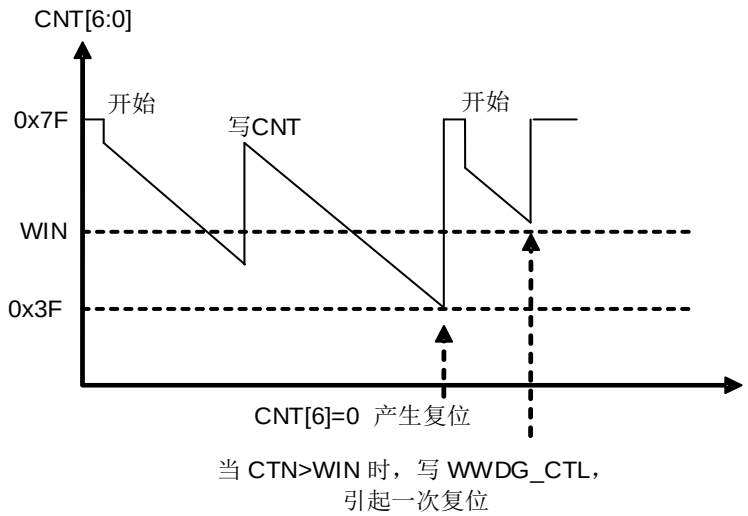
时器。窗口看门狗定时器打开后，计数器始终递减计数，计数器配置的值应该大于0x3F，也就是说CNT[6:0]位应该被置1。CNT[6:0]决定了两次重装载之间的最大间隔时间。计数器的递减速度取决于APB1时钟和预分频器(WWDGT_CFG寄存器的PSC[1:0]位)。

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 21-3. 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：

$$t_{WWDGT}=t_{PCLK1} \times 4096 \times 2^{PSC} \times (CNT[6:0]+1) \quad (ms) \qquad (21\text{-}1)$$

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21-2. 在216MHz (fPCLK1)时的最大/最小超时值*。

表 21-2. 在 216MHz (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	18.96 μs	1.21 ms

1/2	01	37.93 μs	2.43 ms
1/4	10	75.85 μs	4.85 ms
1/8	11	151.70μs	9.71 ms

如果DBG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，窗口看门狗定时器会随着内核停止工作而停止计数。

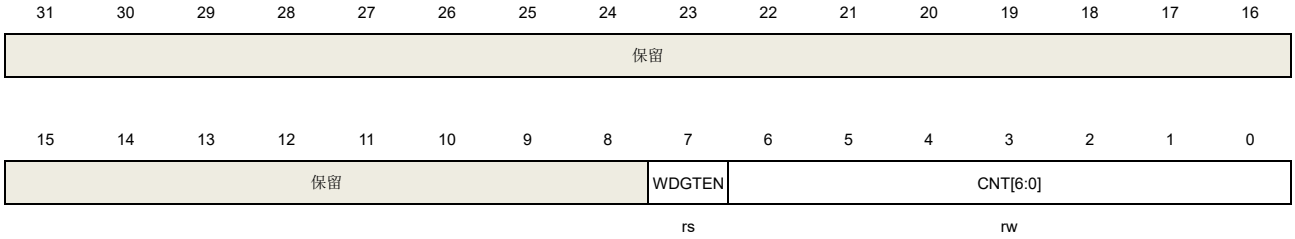
21.2.4. WWDGT 寄存器

WWDGT 基地址：0x40002C00

控制寄存器（WWDGT_CTL）

地址偏移：0x00
复位值：0x0000 007F

该寄存器可以按半字（16 位）或字（32 位）访问。

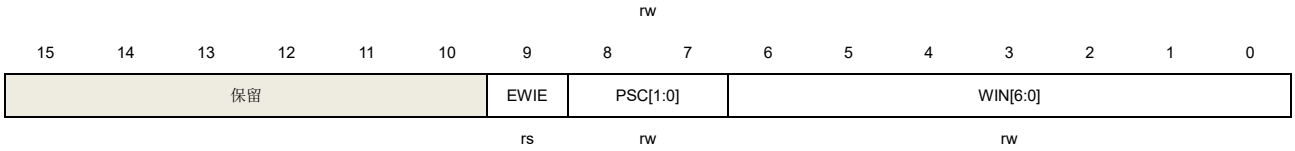


位/位域	名称	说明
31:8	保留	必须保持复位值。
7	WDGTEN	开启窗口看门狗定时器，硬件复位的时候清0，写0无效。 0：关闭窗口看门狗定时器 1：开启窗口看门狗定时器
6:0	CNT[6:0]	看门狗定时器计数器的值。当计数值从0x40降到0x3F时，产生看门狗定时器复位。当计数器值高于窗口值的时候，写计数器可以产生看门狗定时器复位。

配置寄存器（WWDGT_CFG）

地址偏移：0x04
复位值：0x0000 007F

该寄存器可以按半字（16 位）或字（32 位）访问。



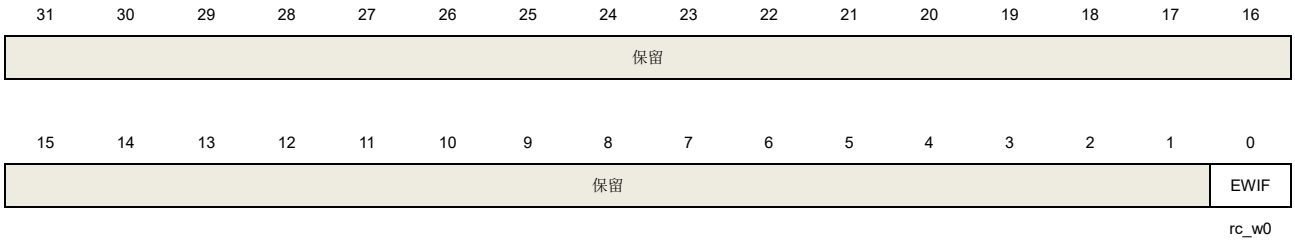
位/位域	名称	说明
31:10	保留	必须保持复位值。
9	EWIE	提前唤醒中断使能。如果该位被置1，计数值达到0x40时触发中断，能触发中断。该位由硬件复位清0，或通过RCU模块的WWDGT软件复位来清0。写0没有任何作用。
8:7	PSC[1:0]	预分频器，看门狗定时器的时间基准。 00：(PCLK1 / 4096) / 1 01：(PCLK1 / 4096) / 2 10：(PCLK1 / 4096) / 4 11：(PCLK1 / 4096) / 8
6:0	WIN[6:0]	窗口值，当看门狗定时器计数器的值大于窗口值时，写看门狗定时器计数器（WWDGT_CTL的CNT位）会产生复位。

状态寄存器（WWDGT_STAT）

地址偏移：0x08

复位值：0x0000 0000

该寄存器可以按半字（16 位）或字（32 位）访问。



位/位域	名称	说明
31:1	保留	必须保持复位值。
0	EWIF	提前唤醒中断标志位。当计数值达到0x40，更新计数器，即使中断没有被使能（WWDGT_CFG中的EWIE位为0）该位也会被硬件置1。这个bit可以通过写0清零，写1无效。

22. 实时时钟（RTC）

22.1. 简介

RTC 模块提供了一个包含日期（年/月/日）和时间（时/分/秒/亚秒）的日历功能。除亚秒用二进制码显示外，时间和日期都以 BCD 码的形式显示。RTC 可以进行夏令时补偿。RTC 可以工作在省电模式下，并通过软件配置来智能唤醒。RTC 支持外接更高精度的低频时钟，用以达到更高的日历精度。

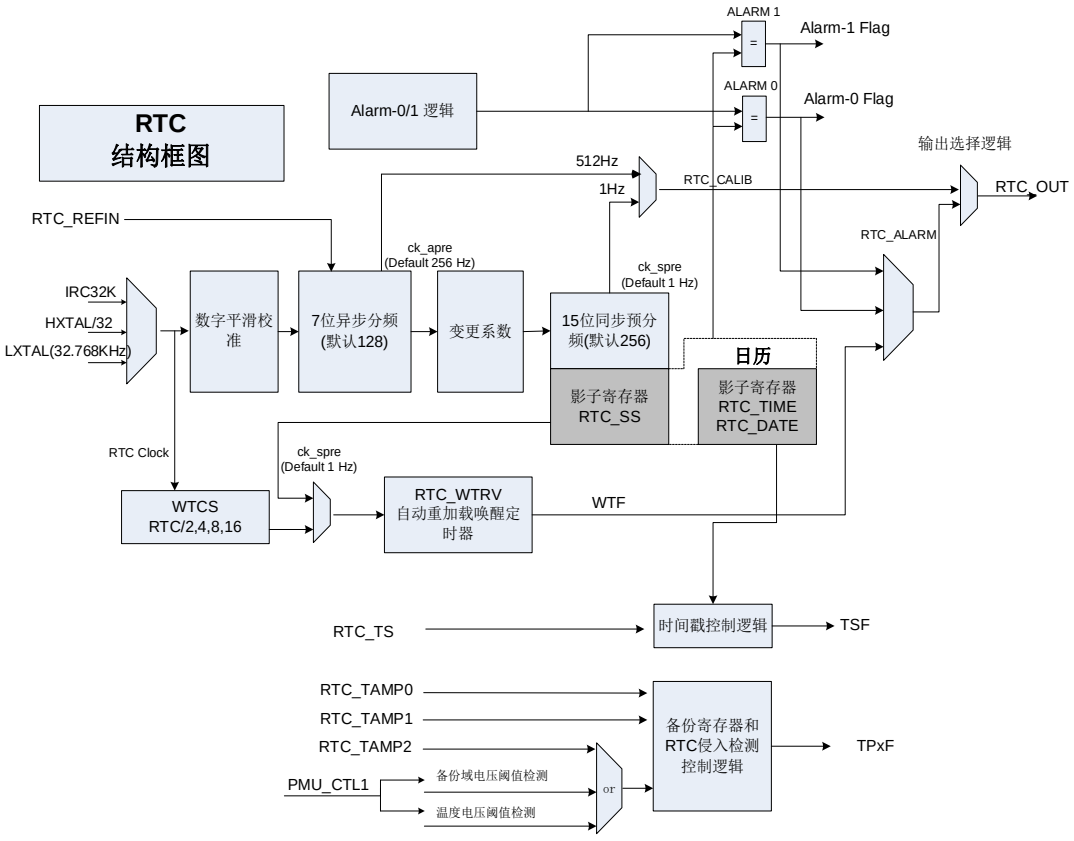
22.2. 主要特征

- 支持日历功能，可支持年、月、日（月的天数）、日（星期几）、时、分、秒和亚秒。
- 通过软件设置来实现夏令时补偿。
- 参考时钟检测功能：通过外接更高精度的低频率时钟源（50Hz或60Hz）来提高日历精度。
- 数字校准功能：通过调整最小时间单位（最大可调精度0.95ppm）来进行日历校准。
- 通过移位功能进行亚秒级调整。
- 记录事件时间的时间戳功能。
- 两个模式可配置的独立的侵入检测（RTC_TAMP0、RTC_TAMP1和RTC_TAMP2）。
- 可编程的日历和两个位域可屏蔽的闹钟。
- 可屏蔽的中断源：
  - 闹钟 0 和闹钟 1；
  - 时间戳检测；
  - 侵入检测；
  - 自动唤醒
- 32个32位（共128字节）通用备份寄存器，能够在省电模式下保存数据。当有外部事件侵入时，备份寄存器将会复位。

22.3. 功能描述

22.3.1. 结构框图

图 22-1. RTC 结构框图



RTC 单元包括：

- 两个闹钟事件/中断，支持两个来自I/O引脚的闹钟事件和中断。
- 侵入事件/中断。
- 当检测到侵入事件时将擦除备份寄存器。
- 当电源切换到V_{BAT}时将产生一个时间戳事件。
- 32位备份寄存器，数量增加到32个。
- 可选的RTC输出功能：
  - 512Hz（默认预分频值）：RTC_OUT(PC13 或 PB2)；
  - 1Hz（默认预分频值）：RTC_OUT(PC13 或 PB2)；
  - 闹钟事件（极性可配置）：RTC_OUT(PC13 或 PB2)；
  - 自动唤醒事件（极性可配置）：RTC_OUT(PC13 或 PB2)。
- 可选的RTC输入功能：
  - 时间戳事件检测：RTC_TS(PC13)；

- 侵入事件检测 0：RTC_TAMP0(PC13)(支持 VDD 和 VBAT 下工作)；
- 侵入事件检测 1：RTC_TAMP1(PA0)(仅支持 VDD 下工作)；
- 侵入事件检测 2：RTC_TAMP2(PE6)(仅支持 VDD 下工作)；
- 参考时钟输入：RTC_REFIN(PB15 或 PA1)；
- 侵入事件检测 2 不仅支持 I/O 引脚触发，当 PMU_CTL1 中的 VBTMEN 位置位后，备份域电压阈值监测和温度电压阈值监测也能触发侵入事件检测 2。

22.3.2. 时钟源和预分频

RTC 单元有三个可选的独立时钟源：LXTAL、IRC32K 和 HXTAL 的 32（由 RCU_CFG 寄存器配置）分频后的时钟。

在 RTC 单元，有两个预分频器用来实现日历功能和其他功能。一个分频器是 7 位异步预分频器，另一个是 15 位同步预分频器。异步分频器主要用来降低功率消耗。如果两个分频器都被使用，建议异步分频器的值尽可能大。

两个预分频器的频率计算公式如下：

$$f_{ck_apre}=\frac{f_{rtcclk}}{FACTOR_A+1} \tag{22-1}$$

$$f_{ck_spre}=\frac{f_{ck_apre}}{FACTOR_S+1}=\frac{f_{rtcclk}}{(FACTOR_A+1)*(FACTOR_S+1)} \tag{22-2}$$

ck_apre 用于为 RTC_SS 亚秒寄存器自减计数器提供时钟，该寄存器值为二进制，表示到达下一秒时间，该寄存器自减到 0 时，自动加载 FACTOR_S 的值。ck_spre 用于为日历寄存器提供时钟，每个时钟增加一秒。

22.3.3. 影子寄存器

当APB总线访问RTC日历寄存器RTC_DATE、RTC_TIME和RTC_SS时，BPSHAD位决定是访问影子寄存器还是真实日历寄存器。默认情况下BPSHAD为0，APB总线访问影子日历寄存器。每两个RTC时钟，影子日历寄存器值会更新为真实日历寄存器的值，与此同时RSYNF位也会再次置位。在Deep-sleep和Standby模式下，影子寄存器不会更新。退出这两种模式时，软件必须清除RSYNF位。如果想要在BPSHAD=0的情况下读日历寄存器的值，须等待RSYNF置1（最大的等待时间是2个RTC时钟周期）。

注意：在BPSHAD=0下，读日历寄存器（RTC_SS，RTC_TIME，RTC_DATE）的APB时钟的频率（f_{apb}）必须至少是RTC时钟频率（f_{rtcclk}）的七倍。

系统复位将复位影子寄存器。

22.3.4. 位域可屏蔽可配置的闹钟

RTC闹钟功能被划分为多个位域并且每一个位域有一个该域的可屏蔽位。

RTC闹钟功能的使能由RTC_CTL寄存器中的ALRMxEN（x=0,1）位控制。当ALRMxEN=1（x=0,1）并且闹钟所有位域的值与对应的日历时间值匹配，ALRMxF（x=0,1）标志位将会置位。

注意：当秒字段未被屏蔽时(RTC_ALRMxTD 寄存器的 MSKS=0)，为确保正常运行，RTC_PSC 寄存器的同步预分频系数（FACTOR_S）应大于等于 3。

如果一个位域被屏蔽，这个位域被认为在逻辑上匹配的。如果所有的位域被屏蔽，在 ALRMxEN 位被置位 3 个 RTC 时钟周期后，ALRMxF 位将置位。

22.3.5. 可配置周期的自动唤醒定时器

RTC具有一个16位的自动递减计数器用来周期性产生唤醒标志

该功能通过WTEN置1来使能，并且可以工作在省电模式。

自动递减计数器有两种可选的时钟来控制：

1. RTC 时钟的 2/4/8/16 分频：
   如果 RTC 时钟为 LXTAL(32.768 KHz)，则唤醒中断周期在 122us 和 32s 之间，分辨率低至 61us。

2. 内部时钟 ck_spre：
   如果 ck_spre 为 1Hz，则唤醒中断周期在 1s 到 36h 之间，分辨率低至 1s。
   - WTCS[2:1] = 0b10，唤醒中断周期在 1s 到 18 h
   - WTCS[2:1] = 0b11，唤醒中断周期在 18h 到 36 h

该功能使能后，计数器自动递减。当计数器到0时，WTF标志位置1，唤醒计数器自动重载RTC_WUT的值。

当WTF置1后，必须软件清除该标志。

如果WTIE被置位，计数器到0时，会产生唤醒中断，从而使系统退出省电模式。系统复位对该功能没有影响。

WTF标志可以从RTC_ALARM通道输出到RTC_OUT。

22.3.6. RTC 初始化和配置

RTC 寄存器写保护

在默认情况下，PMU_CTL寄存器的BKPWEN位被清0。所以写RTC寄存器前需要软件提前设置BKPWEN位。

上电复位后，大多数RTC寄存器是被写保护的。写入这些寄存器的第一步是解锁这些保护。

通过下面的步骤，可以解锁这些保护：

1. 写‘0xCA’到RTC_WPK寄存器；

2. 写 ‘0x53’到RTC_WPK寄存器。

写一个错误的值到RTC_WPK会使写保护再次生效。

备份域复位后，一些RTC寄存器被写保护：RTC_TIME，RTC_DATE，RTC_CTL，RTC_STAT，RTC_PSC，RTC_WUT，RTC_ALRM0TD, RTC_ALRM1TD，RTC_HRFC，RTC_SHIFTCTL，RTC_ALRM0SS, RTC_ALRM1SS，RTC_CFG。

日历初始化和配置

通过以下步骤可以设置日历和预分频器的值：

1. 设置 INITM 位为 1 进入初始化模式。等待 INITF 位被置 1。
2. 在 RTC_PSC 寄存器中，设置同步和异步预分频器的分频系数。
3. 在影子寄存器（RTC_TIME 和 RTC_DATE）中写初始的日历值，并且通过设置 RTC_CTL 寄存器的 CS 位来配置时间的格式（12 或 24 小时制）。
4. 清除 INITM 位退出初始化模式。

大约4个RTC时钟周期后，真正的日历寄存器将从影子寄存器载入时间和日期的设定值，同时日历计数器将要重新开始运行。

注意：初始化以后如果要读取日历寄存器（BPSHAD=0），软件应该确保RSYNF位已经置1。

YCM标志表明日历是否完成初始化，该标志会硬件检查日历的年份值。

夏令时

通过S1H，A1H和DSM位配置，RTC模块可以支持夏令时补偿调节功能。

当日历正在运行时，S1H和A1H能使日历减去或加上1小时。S1H和A1H功能可以重复设置，可以软件配置DSM位来记录这个调节操作。设置S1H或A1H位后，减或加1小时将在下一秒钟到来时生效。

闹钟功能操作步骤

为了避免意外的闹钟标记置位和亚稳态，闹钟功能的操作应遵循如下流程：

1. 清除寄存器 RTC_CTL 的 ALRMxEN（x=0,1）位，禁用闹钟；
2. 设置 Alarm 寄存器（RTC_ALRMxTD/RTC_ALRMxSS（x=0,1））；
3. 设置寄存器 RTC_CTL 的 ALRMxEN（x=0,1）位，使能闹钟功能。

22.3.7. 读取日历

当 BPSHAD=0 时，读日历寄存器

当BPSHAD=0，从影子寄存器读日历的值。由于同步机制的存在，正常读取日历需要满足一个基

本要求：APB总线时钟频率必须大于或等于RTC时钟频率的7倍。在任何情况下APB总线时钟的频率都不能低于RTC的时钟频率。

当APB总线时钟频率低于7倍RTC时钟频率时，日历的读取应该遵守以下流程：

1. 读取两次日历时间和日期寄存器；
2. 如果两次的值相等，那么这个值就是正确的；
3. 如果这两次的值不相等，应该再读一次；
4. 第三次的值可以认为是正确的。

RSYNF每2个RTC时钟周期被置位一次。在这时，影子日历寄存器会更新为真实的日历时间和日期。

为了确保这3个值（RTC_SS，RTC_TIME，RTC_DATE）为同一时间，硬件上采取了如下一致性机制：

1. 读RTC_SS锁定RTC_TIME和RTC_DATE的更新；
2. 读RTC_TIME锁定RTC_DATE的更新；
3. 读 RTC_DATE 解锁 RTC_TIME 和 RTC_DATE 的更新。

如果想在一个很短的时间间隔内（少于2个RTCCLK）读取日历，应先清除RSYNF位并等待其置位后再读取。

下面几种情况，软件须等待RSYNF置位后才能读日历寄存器（RTC_SS，RTC_TIME，RTC_DATE）：

1. 系统复位之后；
2. 日历初始化之后；
3. 一次移位操作之后。

特别是从低功耗模式唤醒后，软件必须清除RSYNF位并等待RSYNF再次置位后才能读取日历寄存器。

当 BPSHAD=1 时，读日历寄存器

当BPSHAD=1，RSYNF位会被硬件清0，读日历寄存器不需考虑RSYNF位。当前真实的日历寄存器值会被直接读取。如此配置的好处是当从低功耗模式(Deep-sleep/Standby模式)唤醒后，软件可以立即获取当前日历寄存器的值而无需加入任何等待延迟(此延迟最大为2个RTC时钟周期)。

由于没有RSYNF位周期性的置位，如果两次读日历寄存器之间出现ck_apre时钟边沿，不同寄存器（RTC_SS/RTC_TIME/RTC_DATE）的值可能并非同一时刻。

另外，如果日历寄存器的值正在发生变化的时刻被APB总线读取，那么有可能APB总线读取的值是不准确的。

为了确保日历值的正确性和一致性，读取时软件须如下操作：连续读取所有日历寄存器的值两次，如果上两次的值是一样的，那么这个值就是一致的且准确的。

22.3.8. RTC 复位

在RTC单元，有两个复位源可用：系统复位和备份域复位。

当系统复位有效时，日历影子寄存器和RTC_STAT寄存器的某些位将要复位到默认值。

备份域复位将会影响下面的寄存器，但系统复位不会对它们产生影响：

- RTC 真实的日历寄存器；
- RTC 控制寄存器（RTC_CTL）；
- RTC 预分频寄存器（RTC_PSC）；
- RTC 唤醒时间寄存器（RTC_WUT）
- RTC 高精度频率补偿寄存器（RTC_HRFC）；
- RTC 移位控制寄存器（RTC_SHIFTCTL）；
- RTC 时间戳寄存器（RTC_SSTS/RTC_TTS/RTC_DTS）；
- RTC 侵入寄存器（RTC_TAMP）；
- RTC 备份寄存器（RTC_BKPx，RTC_CFG）；
- RTC 闹钟寄存器（RTC_ALRMxSS/RTC_ALRMxTD（x=0，1））。

当系统复位或者进入省电模式的时候，RTC单元将会继续运行。但是如果备份域复位，RTC将会停止计数并且所有的寄存器会复位。

22.3.9. RTC 移位功能

当用户有一个高精度的远程时钟而且RTC1Hz时钟（ck_spre）和远程时钟只有一个亚秒级的偏差，RTC单元提供一个称作移位的功能去消除这个偏差来提高秒钟的精确性。

以二进制格式显示亚秒值，RTC运行时该值是递减计数。因此通过增加RTC_SHIFTCTL寄存器的SFS[14:0]的值到RTC_SS同步预分频器计数器值SSC[15:0])或通过增加SFS[14:0]的值到同步预分频器计数器SSC[15:0]并且同时置位A1S位，能分别延迟或提前下一秒到达的时间。

RTC_SS的最大值取决于RTC_PSC寄存器的FACTOR_S的值。FACTOR_S越大，调整的精度也就越高。

因为1Hz的时钟（ck_spre）由FACTOR_A和FACTOR_S共同产生，越高的FACTOR_S值就意味着越低的FACTOR_A值，同时越低的FACTOR_A意味着越高的功耗。

注意：在使用移位功能之前，软件必须检查 RTC_SS 中 SSC 的第 15 位(SSC[15])并确保该位为 0。写 RTC_SHIFTCTL 寄存器之后，RTC_STAT 寄存器的 SOPF 位将会再次置位。当同步移位操作完成时，SOPF 位被硬件清 0。系统复位不影响 SOPF 位。当 REFEN=0 时，移位操作才能正确的工作。如果 REFEN=1，软件禁止写入 RTC_SHIFTCTL。

22.3.10. RTC 参考时钟检测

RTC参考时钟是另外一种提高RTC秒级精度的方法。为了使能这项功能，需要有一个相对于LXTAL有更高精度的外部参考时钟源（50Hz或60Hz）。

使能这项功能之后(REFEN=1)，每一个秒更新的时钟(1Hz)边沿将与最近的RTC_REFIN参考时钟沿进行对比。在大多数情况下，这两个时钟沿是对齐的。但当两个时钟沿由于LXTAL准确度的原因没有对齐的时候，RTC参考时钟的检测功能会偏移1Hz时钟沿一点相位，使得下一个1Hz时钟沿和参考时钟沿对齐。

当REFEN=1，每一秒前后都会有一个进行检测的时间窗，处于不同的检测状态，时间窗时长也不同。当检测状态处于检测第一个参考时钟边沿时，使用7个ck_apre时长的时间窗，当检测状态处于边沿对齐操作时，使用3个ck_apre时长的时间窗。

无论使用哪一种时间窗，当参考时钟在时间窗中被检测到的时候，同步预分频计数器会被强制重载。当两个时钟（ck_spre和参考时钟）边沿是对齐的，这个重载操作对1Hz日历更新没有任何影响。但是当两个时钟边沿没有对齐时，这个重载操作将会移动ck_spre时钟边沿，以使得ck_spre（1Hz）时钟边沿和参考时钟边沿对齐。

当参考检测功能正在运行中但外部参考时钟消失（在3个ck_apre时长时间窗内没有发现参考时钟边沿），日历也能通过LXTAL继续自动更新。如果这个参考时钟重新恢复，参考时钟检测功能会先用7个ck_apre时长时间窗口去检测参考时钟，然后用3个ck_apre时长时间窗口去调节ck_spre（1Hz）时钟边沿。

注意：使能参考时钟检测功能之前(REFEN=1)，软件必须配置 FACTOR_A 为 0x7F，FACTOR_S 为 0xFF。

待机模式下，参考时钟检测功能不可用。

22.3.11. RTC 数字平滑校准

RTC平滑校准是一种用于校准RTC频率的方法，该方法通过调整校准周期内的RTC时钟脉冲个数的方式来实现校准。

完成一次这种校准相当于在一次校准周期内，RTC时钟的脉冲个数增加或者减少了一定的数目。这种校准的分辨率大约为0.954ppm，范围为从-487.1ppm到+488.5ppm。

校准周期的时间可以配置到 $2^{20}/2^{19}/2^{18}$ RTC 时钟周期，如果 RTC 的输入频率是 32.768KHz，这些校准周期时间分别代表 32/16/8 秒。

高精度频率补偿寄存器(RTC_HRFC)指定了在校准周期内要屏蔽的RTC时钟数目，CMSK[8：0]位能屏蔽0到511个RTC时钟，这样RTC的频率最多降低487.1PPM。

为了提高RTC频率可以设置FREQI位。如果FREQI位被置位，将会有512个额外的RTC时钟周期增加到校准周期(32/16/8 秒)时间期间，这意味着每$2^{11}/2^{10}/2^{9}$ RTC时钟插入一个RTC时钟周期。

因此使用FREQI可以使RTC频率增加488.5ppm。

同时使用CMSK和FREQI，每个周期时间可以调整-511到+512个RTC时钟周期。这意味着在0.954ppm分辨率的情况下，调整范围为从-487.1ppm到+488.5ppm。

当数字平滑校准功能正在运行时，按如下公式计算输出校准频率：

$$f_{cal}=f_{rtcclk} \times(1+\frac{FREQI\times512-CMSK}{2^{N}+CMSK-FREQI\times512}) \quad (22\text{-}3)$$

注意： N=20/19 /18 (32/16/8 秒)校准时间周期。

当 FACTOR_A < 3 时校准：

当异步预分频器值(FACTOR_A)被设置小于3时，若要使用校准功能，软件不能将FREQI位设置为1。当FACTOR_A<3，FREQI位设置将会被忽略。

当FACTOR_A小于3时，FACTOR_S值应小于标称值。假设RTC时钟频率是正常的32.768KHz，对应的FACTOR_S应该按下面所示设置：

FACTOR_A = 2：FACTOR_S减少2(8189)
FACTOR_A = 1：FACTOR_S减少4(16379)
FACTOR_A = 0：FACTOR_S减少8(32759)
当FACTOR_A小于3，CMSK为0x100，校准频率公式如下：

$$f_{cal}=f_{rtcclk} \times(1+\frac{256-CMSK}{2^{N}+CMSK-256}) \quad (22\text{-}4)$$

注意： N=20/19 /18 (32/16/8 秒)校准时间周期。

验证 RTC 校准

提供1Hz校准时钟的输出用于协助软件测量并验证RTC的精度。

在有限的测量周期内测量RTC的频率，最高可能发生2个RTCCLK的测量误差。

为了消除这一测量误差，测量周期应该和校准周期一致。

- 校准周期设为32秒(默认配置)
  用准确的32秒周期去测量1Hz校准输出的准确性能保证这个测量误差在0.477ppm（在32秒周期内0.5个RTCCLK）之内。
- 校准周期设为16秒（通过设置CWND16位）
  使用此配置，CMSK[0]被硬件置0。
  用准确的16秒周期去测量1Hz校准输出的准确性能保证这个测量误差在0.954ppm（在16秒周期内0.5个RTCCLK）之内。
- 校准周期设为8秒（通过设置CWND8位）
  使用此配置，CMSK[1：0]被硬件置0。
  用准确的8秒周期去测量1Hz校准输出的准确性能保证这个测量误差在1.907ppm(在8秒周期内0.5

个RTCCLK)之内。

运行中重校准

当INITF位是0，用下面的步骤，软件可以更新RTC_HRFC：

1) 等待SCPF位置0；
2) 写一个新的值到RTC_HRFC寄存器；
3) 3 个 ck_apre 时钟周期之后，新的校准设置开始生效。

22.3.12. 时间戳功能

时间戳功能由RTC_TS管脚输入，通过配置TSEN位来使能，也可以通过ITSEN使能。

当RTC_TS管脚检测到时间戳事件发生时，会将日历的值保存在时间戳寄存器中（RTC_DTS/RTC_TTS/RTC_SSTS），同时时间戳标志（TSF）也将由硬件置1。如果时间戳中断使能被启用（TSIE），时间戳事件会产生一个中断。

当检测到内部时间戳事件发生时，会将日历的值保存在时间戳寄存器中（RTC_DTS/RTC_TTS/RTC_SSTS），同时时间戳标志（TSF）和内部时间戳标志（ITSF）也将由硬件置1。如果时间戳中断使能被启用（TSIE），时间戳事件会产生一个中断。当切换到V_{BAT}供电时会产生内部时间戳事件。

时间戳寄存器只会在时间戳事件第一次发生的时刻（TSF=0）记录日历时间，而当TSF=1时，时间戳事件的值不会被记录。

RTC模块提供了一个可选的功能特性，来增加时间戳事件的触发源：设置TPTS=1，使得侵入检测功能的侵入事件同时也作为时间戳事件的输入源。

注意：因为同步机制的原因，当时间戳事件发生时，TSF 会延迟 2 个 ck_apre 周期置位。

22.3.13. 侵入检测

RTC_TAMPx管脚可以作为侵入事件检测功能输入管脚，检测模式有两种可供用户选择：边沿检测模式或者是带可配置滤波功能的电平检测模式。

入侵检测的配置可以用于以下目的：

- 默认配置会擦除RTC备份寄存器
- 可以从深度睡眠模式和待机模式唤醒并产生中断

RTC 备份寄存器（RTC_BKPx）

RTC备份寄存器处于VDD备份域中，即使VDD电源被切断，该区域的寄存器的电源还可由VBAT提供。从待机模式唤醒或系统复位操作都不会影响这些寄存器。

只有当被检测到有侵入事件和备份域复位时，这些寄存器复位。

初始化侵入检测功能

TPxEN位可以独立使能对应于不同管脚上的RTC侵入检测功能。使能TPxEN位启动侵入检测功能之前，需要设置好侵入检测的配置。

TPxF标志会在引脚上出现侵入事件后置位，并存在以下延迟：

- 当FLT不为0x0时（带可配置滤波的电平检测），延迟为3个ck_apre 周期；
- 当TPTS=1时（入侵事件的时间戳），延迟为3个ck_apre 周期；
- 当FLT=0x0（边沿检测）且TPTS=0时，无延迟；

在此期间，只要TPxF置1，就无法检测到同一引脚上出现的新入侵。

侵入事件源的时间戳

使能TPTS位，能让侵入检测功能被用作时间戳功能。如果这位被设置为1，当检测到侵入事件时，TSF也将会被置位，如同使能了时间戳功能。当检测到侵入事件时，无论TPTS位的值如何，TPxF位将置位。

侵入事件检测为边沿检测模式

当FLT位为0x0时，侵入检测被设置成边沿检测模式，TPxEG位决定检测沿是上升沿还是下降沿。当侵入检测配置为边沿检测模式时，侵入检测输入管脚上的上拉电阻将会被禁用。

由于检测侵入事件会复位备份寄存器（RTC_BKPx），因此对备份寄存器写操作时应该确保侵入事件导致的复位和写操作不会同时发生。避免这种情形的推荐方法是先关闭侵入检测功能，在完成写操作后再重新启动该功能。

注意：Tamper 上的侵入检测功能即使 V_{DD} 电源被关掉也依然可以运行。

侵入事件检测为带可配置滤波功能的电平检测模式

当FLT位没有被设置成0x0时，侵入检测被设置成电平检测模式，FLT位决定有效电平需连续采样的次数（2，4或者8）。

当DISPU被设置成0(默认值)，内部的上拉电阻将会在每一次采样前预充电侵入管脚，这样侵入事件的输入管脚上就允许连接更大的电容。预充电的时间可以通过PRCH位来配置。越大的电容，所需的充电时间越长。

电平检测模式下每次采样之间的时间间隔是可配置的。通过调整采样频率(FREQ)，软件能在功耗和检测延迟之间取得一个平衡。

22.3.14. 校准时钟输出

如果COEN位设置为1，RTC_OUT会输出参考校准时钟。

当COS位设置为0（默认值）并且异步预分频器（FACTOR_A）设为0x7F时，RTC_CALIB的频率是f_{rtcclk}/64。因此若RTCCLK的频率为32.768KHz，RTC_CALIB对应的输出为512Hz。因为下降沿存在轻微的抖动，因此推荐使用RTC_CALIB输出的上升沿。

当COS位设置为1时，RTC_CALIB的频率计算公式为：

$$f_{rtc_calib} = \frac{f_{rtcclk}}{(FACTOR_A+1)\times(FACTOR_S+1)} \quad (22\text{-}5)$$

若RTCCLK为32.768KHz，如果预分频器是默认值，那么RTC_CALIB对应的输出是1Hz。

22.3.15. 闹钟输出

当OS控制位不为0x00时，RTC_ALARM复用输出功能被启用。这个功能将直接输出RTC_STAT寄存器的闹钟标志ALRMxF或者自动唤醒标志WTF。

RTC_CTL寄存器中的OPOL位可以配置ALRMxF标志或者WTF标志输出时候的极性，因此RTC_ALARM的输出电平有可能与相应的位值相反。

22.3.16. RTC 引脚配置

RTC_OUT，RTC_TS和RTC_TAMP0都使用同一个PC13引脚。无论PC13的GPIO是什么配置，PC13的功能由RTC控制。PC13的RTC功能可以用于所以低功耗模式和Vbat模式

PC13的输出优先级如表*表22-1. RTC引脚配置和功能*

表 22-1. RTC 引脚配置和功能

功能配置和引脚功能	OS[1:0]（输出选择）	COEN（校准输出）	TP0EN（侵入检测0使能）	TSEN（时间戳使能）	ALRMOUTTYPE（闹钟输出类型）
闹钟开漏输出	01 或 10 或 11	-	-	-	0
闹钟推挽输出	01 或 10 或 11	-	-	-	1
校准推挽输出	00	1	-	-	-
TAMP0浮空输入	00	0	1	0	-
时间戳和TAMP0浮空输入	00	0	1	1	-
时间戳浮空输入	00	0	0	1	-
标准GPIO	00	0	0	0	-

PC13可用于以下用途:

- RTC_ALARM输出：这个输出可以是RTC Alarm 0, RTC Alarm 1或RTC Wakeup取决于RTC_CTL寄存器中的OS[1:0]位域。
- RTC_CALIB输出：该特性通过在RTC_CTL寄存器中设置COEN[23]来启用
- RTC_TAMP0：篡改事件检测
- RTC_TS：时间戳事件检测

RTC_CFG中的ALRMOUTTYPE用于选择RTC_ALARM输出是推挽模式还是开漏模式。

由于RTC_CFG[31]中的OUT2EN位，可以在PB2或PC13引脚上输出RTC_OUT。该输出在VBAT / Standby模式下不可用。

22.3.17. RTC 省电模式管理

表 22-2. 省电模式管理

模式	模式下能否工作	退出该模式的方法
睡眠模式	是	RTC中断
深度睡眠	当时钟源是LXTAL或IRC32K时可以工作	RTC闹钟/侵入事件/时间戳事件/唤醒
待机模式	当时钟源是LXTAL或IRC32K时可以工作	RTC闹钟/侵入事件/时间戳事件/唤醒

22.3.18. RTC 中断

所有的RTC中断都被连接到EXTI控制器。

如果想使用RTC闹钟/侵入事件/时间戳中断/自动唤醒中断，应按下面步骤操作：

1. 设置并使能对应的 EXTI 中连接到 RTC 闹钟/侵入事件/时间戳/自动唤醒的中断线，然后配置该线为上升沿触发模式；
2. 配置并使能 RTC 闹钟/侵入事件/时间戳/自动唤醒的全局中断；
3. 配置并使能 RTC 闹钟/侵入事件/时间戳功能。

表 22-3. RTC 中断控制

中断	事件标志	控制位	退出睡眠模式	退出深度睡眠模	退出待机睡眠模
闹钟0	ALRM0F	ALRM0IE	Y	Y(1)	Y(1)
闹钟1	ALRM1F	ALRM1IE	Y	Y(1)	Y(1)
唤醒	WTF	WTIE	Y	Y(1)	Y(1)
时间戳	TSF	TSIE	Y	Y(1)	Y(1)

中断	事件标志	控制位	退出睡眠模式	退出深度睡眠模	退出待机睡眠模
侵入0	TP0F	TPIE	Y	Y(1)	Y(1)
侵入1	TP1F	TPIE	Y	Y(1)	N(1)
侵入2	TP2F	TPIE	Y	Y(1)	N(1)

(1). 仅当RTC时钟源为LXTAL或IRC32K时，才可以从深度睡眠和待机模式唤醒。

22.4. RTC 寄存器

RTC基地址：0x4000 2800

22.4.1. 时间寄存器（RTC_TIME）

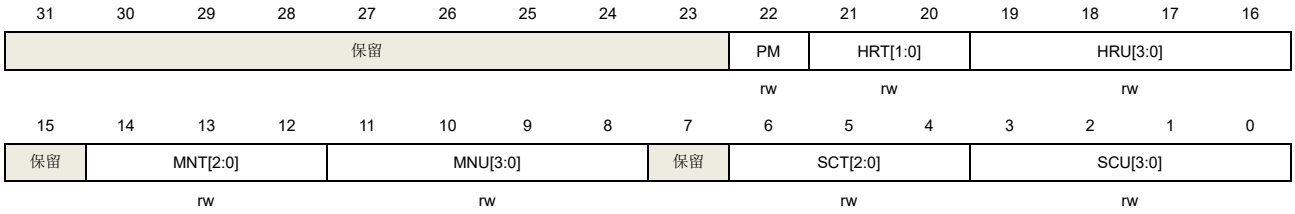
偏移地址：0x00

系统复位值：当BPSHAD = 0，0x0000 0000

当BPSHAD = 1，无影响

写保护寄存器，仅在初始化状态可以进行写操作

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:23	保留	必须保持复位值。
22	PM	AM/PM 标志 0：AM 或 24 小时制 1：PM
21:20	HRT[1:0]	小时十位值，以 BCD 码形式存储
19:16	HRU[3:0]	小时个位值，以 BCD 码形式存储
15	保留	必须保持复位值。
14:12	MNT[2:0]	分钟十位值，以 BCD 码形式存储
11:8	MNU[3:0]	分钟个位值，以 BCD 码形式存储
7	保留	必须保持复位值。
6:4	SCT[2:0]	秒钟十位值，以 BCD 码形式存储
3:0	SCU[3:0]	秒钟个位值，以 BCD 码形式存储

22.4.2. 日期寄存器（RTC_DATE）

偏移地址：0x04

系统复位值：当 BPSHAD = 0，0x0000 2101
当 BPSHAD = 1，无影响
写保护寄存器，仅在初始化状态可以进行写操作。

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								YRT[3:0]				YRU[3:0]			
								rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DOW[2:0]			MONT	MONU[2:0]				保留		DAYT		DAYU			
rw			rw	rw						rw		rw			

位/位域	名称	描述
31:24	保留	必须保持复位值。
23:20	YRT[3:0]	年份十位值，以 BCD 码形式存储
19:16	YRU[3:0]	年份个位值，以 BCD 码形式存储
15:13	DOW[2:0]	星期 0x0：保留 0x1：星期一 … 0x7：星期日
12	MONT	月份十位值，以 BCD 码形式存储
11:8	MONU[2:0]	月份个位值，以 BCD 码形式存储
7:6	保留	必须保持复位值。
5:4	DAYT[1:0]	日期十位值，以 BCD 码形式存储
3:0	DAYU[3:0]	日期个位值，以 BCD 码形式存储

22.4.3. 控制寄存器（RTC_CTL）

偏移地址：0x08
系统复位：无影响
备份域复位值：0x0000 0000
写保护寄存器

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						TAMPCLK	ITSEN	COEN	OS[1:0]		OPOL	COS	DSM	S1H	A1H
						rw	rw	rw	rw		rw	rw	rw	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIE	WTIE	ALRM1IE	ALRM0IE	TSEN	WTEN	ALRM1EN	ALRM0EN	保留	CS	BPSHAD	REFEN	TSEG	WTCS[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw		

位/位域	名称	描述
31:26	保留	必须保持复位值。
25	TAMPCLK	边沿 Tamper 需要 RTC 时钟 0：需要 RTC 时钟 1：不需要 RTC 时钟
24	ITSEN	内部时间戳事件使能 0：关闭内部时间戳事件 1：使能内部时间戳事件
23	COEN	校准输出使能 0：关闭校准输出 1：使能校准输出
22:21	OS[1:0]	输出选择 该位用来选择输出的标志源。 0x00：禁用 RTC_ALARM 输出 0x01：启用闹钟 0 标志输出 0x10：启用闹钟 1 标志输出 0x11：启用唤醒标志输出
20	OPOL	输出极性 该位用来反转 RTC_ALARM 输出。 0：禁用反转 RTC_ALARM 输出 1：启用反转 RTC_ALARM 输出
19	COS	校准输出选择 仅当 COEN=1 并且预分频器是默认值时有效。 0：校准输出是 512Hz 1：校准输出是 1Hz
18	DSM	夏令时屏蔽位 该位可以通过软件灵活使用。常用来记录夏令时调整。
17	S1H	减 1 小时(冬季时间变化) 当前时间非零的情况下，将当前时间减去一个小时。 0：没有影响 1：在下一个秒改变时，将减少一个小时

16	A1H	增加 1 小时(夏季时间变化) 将当前时间增加一个小时。 0：没有影响 1：在下一个秒改变时，将增加一个小时
15	TSIE	时间戳中断使能 0：禁用时间戳中断 1：启用时间戳中断
14	WTIE	自动唤醒定时器中断使能 0：禁用自动唤醒定时器中断 1：启用自动唤醒定时器中断
13	ALRM1IE	RTC 闹钟 1 中断使能 0：禁用闹钟中断 1：启用闹钟中断
12	ALRM0IE	RTC 闹钟 0 中断使能 0：禁用闹钟中断 1：启用闹钟中断
11	TSEN	时间戳功能使能 0：禁用时间戳功能 1：启用时间戳功能
10	WTEN	自动唤醒定时器功能使能 0：禁用自动唤醒定时器功能 1：启用自动唤醒定时器功能
9	ALRM1EN	闹钟 1 功能使能 0：禁用闹钟功能 1：启用闹钟功能
8	ALRM0EN	闹钟 0 功能使能 0：禁用闹钟功能 1：启用闹钟功能
7	保留	必须保持复位值。
6	CS	时间格式 0：24 小时制 1：12 小时制 注意：仅能在初始化状态进行写入
5	BPSHAD	禁止影子寄存器 0：读取的日历的值来自影子日历寄存器

		1：读取的日历的值来自真正日历寄存器 **注意：**如果 APB 时钟的频率小于 RTCCLK 频率的 7 倍，该位必须设为 1
4	REFEN	参考时钟检测功能使能 0：禁用参考时钟检测功能 1：启用参考时钟检测功能 **注意：**仅能在初始化状态进行写入并且 FACTOR_S 必须为 0x00FF
3	TSEG	时间戳事件有效检测边沿 0：上升沿是时间戳事件有效检测沿 1：下降沿是时间戳事件有效检测沿
2:0	WTCS[2:0]	自动唤醒定时器时钟选择 0x0：RTC 时钟的 16 分频 0x1：RTC 时钟的 8 分频 0x2：RTC 时钟的 4 分频 0x3：RTC 时钟的 2 分频 0x4，0x5：ck_spre (默认 1Hz)时钟 0x6，0x7：ck_spre (默认 1Hz)时钟并且将唤醒计数器值增加2^{16}

22.4.4. 状态寄存器（RTC_STAT）

偏移地址：0x0C
系统复位：仅INITM，INITF和RSYNF位被置0，其他位无影响。
备份域复位值：0x0000 0007
写保护寄存器

该寄存器只能按字(32位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TP2F	TP1F	TP0F	TSOVRF	TSF	WTF	ALRM1F	ALRM0F	INITM	INITF	RSYNF	YCM	SOPF	WTWF	ALRM1WF	ALRM0WF
rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rw	r	rc_w0	r	r	r	r	r

位/位域	名称	描述
31:18	保留	必须保持复位值。
17	ITSF	内部时间戳标志 当检测到内部时间戳事件时，该位硬件置 1。可以通过向该位软件写 0 来清除，并且和 TSF 位一起清零。
16	SCPF	平滑校准挂起标志 在未进入初始化模式时向 RTC_HRFC 进行软件写操作，该位被硬件置 1。当平滑

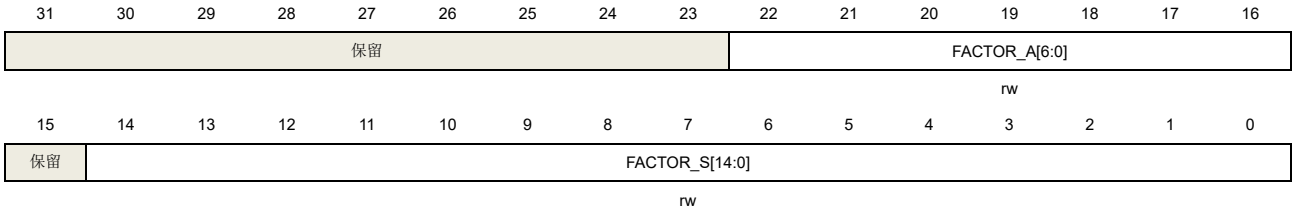
		校准设置开始执行后，该位被硬件清零 0。
15	TP2F	RTC_TAMP2 事件标志 当在 tamper2 输入管脚检测到侵入事件时，该位硬件置 1。可以通过向该位软件写 0 来清除。
14	TP1F	RTC_TAMP1 事件标志 当在 tamper1 输入管脚检测到侵入事件时，该位硬件置 1。可以通过向该位软件写 0 来清除。
13	TP0F	RTC_TAMP0 事件标志 当在 tamper0 输入管脚检测到侵入事件时，该位硬件置 1。可以通过向该位软件写 0 来清除。
12	TSOVRF	时间戳事件溢出标志 如果 TSF 位已经置位，当再次检测到时间戳事件时，该位会通过硬件置 1。 可以通过向该位软件写 0 来清除。
11	TSF	时间戳事件标志 当检测到一个时间戳事件时，该位会通过硬件置 1。可以通过向该位软件写 0 来清除。
10	WTF	唤醒定时器标志 当唤醒定时器减到 0 时，该位会通过硬件置 1。可以通过向该位软件写 0 来清除。 该标志需要在 WTF 位再次置 1 之前的 1.5 个 RTC 时钟周期前完成软件清除该位。
9	ALRM1F	Alarm1 发生标志 当现在的时间/日期与闹钟 1 设置的时间/日期匹配的时候，该位会通过硬件置 1。可以通过向该位软件写 0 来清除。
8	ALRM0F	Alarm0 发生标志 当现在的时间/日期与闹钟 0 设置的时间/日期匹配的时候，该位会通过硬件置 1。可以通过向该位软件写 0 来清除。
7	INITM	进入初始化模式 0：自由运行模式 1：进入初始化模式设置时间/日期和预分频，计数器将停止运行
6	INITF	初始化状态标志 该位被硬件置 1，初始化状态时可以设置日历寄存器和预分频器。 0：日历寄存器和预分频器的值不能改变 1：日历寄存器和预分频器的值可以改变
5	RSYNF	寄存器同步标志 每 2 个 RTCCLK 将会由硬件置 1 一次，同时会复制当前日历时间/日期到影子日历寄存器。初始化模式（INITM），移位操作挂起标志（SOPF）或者禁止影子寄存器

		模式（BPSHAD = 1）会清除该位。该位也可以通过软件写 0 清除。 0：影子寄存器未同步 1：影子寄存器已同步
4	YCM	年份配置标志 当日历寄存器的年份值不为 0 时硬件置 1 0：日历尚未初始化 1：日历已经初始化
3	SOPF	移位功能操作挂起标志 0：移位操作没有挂起 1：移位操作挂起
2	WTWF	唤醒定时器可写标志 0：不允许更新唤醒定时器 1：允许更新唤醒定时器
1	ALRM1WF	Alarm1 配置可写标志 硬件置位和清零。ALRM1EN=0 时，标记 alarm 是否可写。 0：不允许修改 Alarm 寄存器设置 1：允许修改 Alarm 寄存器设置
0	ALRM0WF	Alarm0 配置可写标志 硬件置位和清零。ALRM0EN=0 时，标记 alarm 是否可写。 0：不允许修改 Alarm 寄存器设置 1：允许修改 Alarm 寄存器设置

22.4.5. 预分频寄存器（RTC_PSC）

偏移地址：0x10
系统复位：无影响
备份域复位值：0x007F 00FF
写保护寄存器，仅在初始化状态可以进行写操作。

该寄存器只能按字(32位)访问。



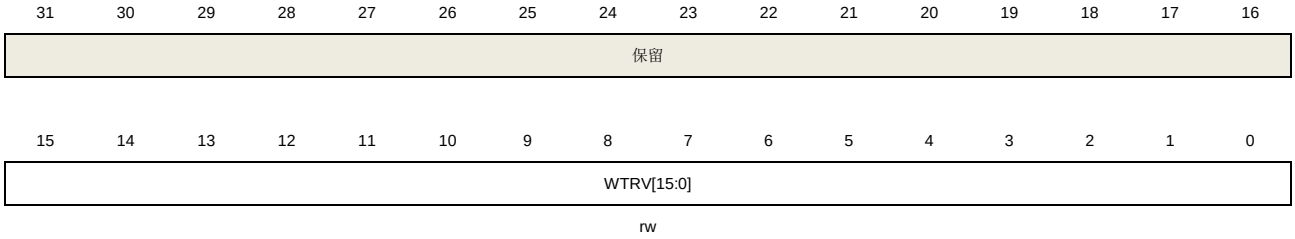
位/位域	名称	描述

31:23	保留	必须保持复位值。
22:16	FACTOR_A[6:0]	异步预分频系数 ck_apre 频率 = RTCCLK 频率/(FACTOR_A+1)
15	保留	必须保持复位值。
14:0	FACTOR_S[14:0]	同步预分频系数 ck_spre 频率 = ck_apre 频率/(FACTOR_S+1)

22.4.6. 唤醒定时器寄存器（RTC_WUT）

偏移地址：0x14
系统复位：无影响
备份域复位值：0x0000 FFFF
写保护寄存器

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	WTRV[15:0]	自动唤醒定时器重载值 当 WTEN 置 1 时，每隔 (WTRV[15:0]+1) 个 ck_wut 周期，WTF 置 1 一次 。 ck_wut 通过 WTCS[2:0] 位选择. 注意： 禁止在 WTCS[2:0]=0b 011 时配置 WTRV=0x0000。 该寄存器仅在 WTWF=1 时才能写操作

22.4.7. 闹钟 0 时间日期寄存器（RTC_ALRM0TD）

偏移地址：0x1C
系统复位：无影响
备份域复位值：0x0000 0000
写保护寄存器，仅在初始化状态可以进行写操作。

该寄存器只能按字(32位)访问。

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

MSKD	DOWS	DAYT[1:0]		DAYU[3:0]				MSKH	PM	HRT[1:0]		HRU[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
MSKM	MNT[2:0]			MNU[3:0]				MSKS	SCT[2:0]			SCU[3:0]			
rw	rw			rw				rw	rw			rw			

位/位域	名称	描述
31	MSKD	闹钟日期位域屏蔽位 0：不屏蔽日期/天位域 1：屏蔽日期/天位域
30	DOWS	星期选择 0：此时 DAYU[3:0]代表日期个位值 1：此时 DAYU[3:0]代表星期几，此时 DAYT[1:0]无意义
29:28	DAYT[1:0]	日期十位值，以 BCD 码格式存储
27:24	DAYU[3:0]	日期个位值或星期天数，以 BCD 码格式存储
23	MSKH	闹钟小时位域屏蔽位 0：不屏蔽小时位域 1：屏蔽小时位域
22	PM	AM/PM 标志 0：AM 或 24 小时制 1：PM
21:20	HRT[1:0]	小时十位值，以 BCD 码形式存储
19:16	HRU[3:0]	小时个位值，以 BCD 码形式存储
15	MSKM	闹钟分钟位域屏蔽位 0：不屏蔽分钟位域 1：屏蔽分钟位域
14:12	MNT[2:0]	分钟十位值，以 BCD 码形式存储
11:8	MNU[3:0]	分钟个位值，以 BCD 码形式存储
7	MSKS	闹钟秒位域屏蔽位 0：不屏蔽秒位域 1：屏蔽秒位域
6:4	SCT[2:0]	秒钟十位值，以 BCD 码形式存储
3:0	SCU[3:0]	秒钟个位值，以 BCD 码形式存储

22.4.8. 闹钟 1 时间日期寄存器（RTC_ALRM1TD）

偏移地址：0x20
系统复位：无影响
备份域复位值：0x0000 0000
写保护寄存器，仅在初始化状态可以进行写操作。

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSKD	DOWS	DAYT[1:0]		DAYU[3:0]				MSKH	PM	HRT[1:0]		HRU[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
MSKM	MNT[2:0]			MNU[3:0]				MSKS	SCT[2:0]			SCU[3:0]			
rw	rw			rw				rw	rw			rw			

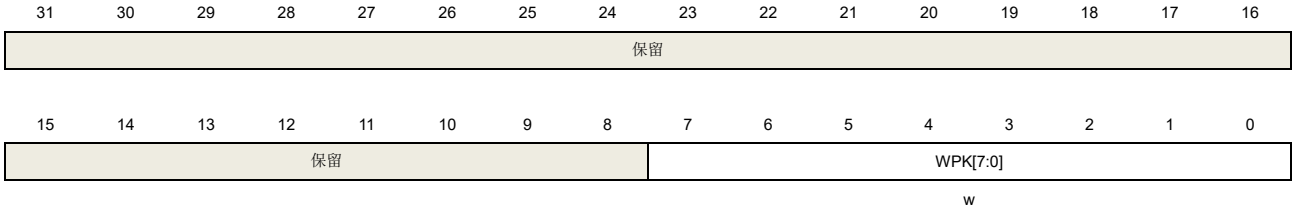
位/位域	名称	描述
31	MSKD	闹钟日期位域屏蔽位 0：不屏蔽日期/天位域 1：屏蔽日期/天位域
30	DOWS	星期选择 0：此时 DAYU[3:0] 代表日期个位值 1：此时 DAYU[3:0] 代表星期几，此时 DAYT[1:0]无意义
29:28	DAYT[1:0]	日期十位值，以 BCD 码格式存储
27:24	DAYU[3:0]	日期个位值或星期天数，以 BCD 码格式存储
23	MSKH	闹钟小时位域屏蔽位 0：不屏蔽小时位域 1：屏蔽小时位域
22	PM	AM/PM 标志 0：AM 或 24 小时制 1：PM
21:20	HRT[1:0]	小时十位值，以 BCD 码形式存储
19:16	HRU[3:0]	小时个位值，以 BCD 码形式存储
15	MSKM	闹钟分钟位域屏蔽位 0：不屏蔽分钟位域 1：屏蔽分钟位域
14:12	MNT[2:0]	分钟十位值，以 BCD 码形式存储

11:8	MNU[3:0]	分钟个位值，以 BCD 码形式存储
7	MSKS	闹钟秒位域屏蔽位 0：不屏蔽秒位域 1：屏蔽秒位域
6:4	SCT[2:0]	秒钟十位值，以 BCD 码形式存储
3:0	SCU[3:0]	秒钟个位值，以 BCD 码形式存储

22.4.9. 写保护钥匙寄存器（RTC_WPK）

偏移地址：0x24
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

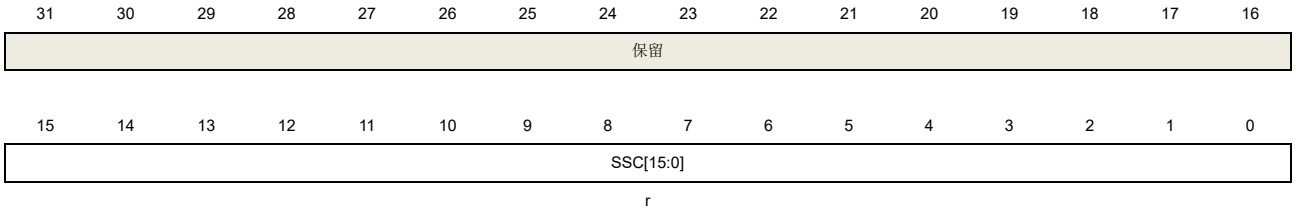


位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	WPK[7:0]	写保护的解锁值

22.4.10. 亚秒寄存器（RTC_SS）

偏移地址：0x28
系统复位值：当BPSHAD = 0，0x0000 0000。
当BPSHAD = 1，无影响。

该寄存器只能按字(32位)访问。



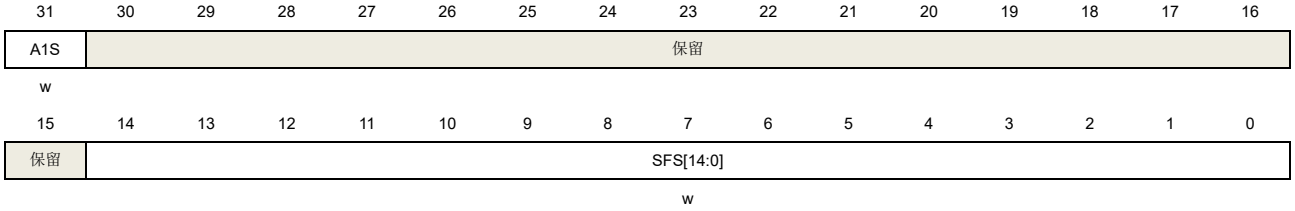
位/位域	名称	描述
31:16	保留	必须保持复位值。

15:0	SSC[15:0]	亚秒值 该位值是同步预分频计数器的值。秒的小数部分由下面公式给出： 秒的小数部分 = (FACTOR_S - SSC)/(FACTOR_S + 1)

22.4.11. 移位控制寄存器（RTC_SHIFTCTL）

偏移地址：0x2C
系统复位：无影响
备份域复位值：0x0000 0000
写保护寄存器，仅当SOPF=0，该寄存器可写。

该寄存器只能按字(32位)访问。



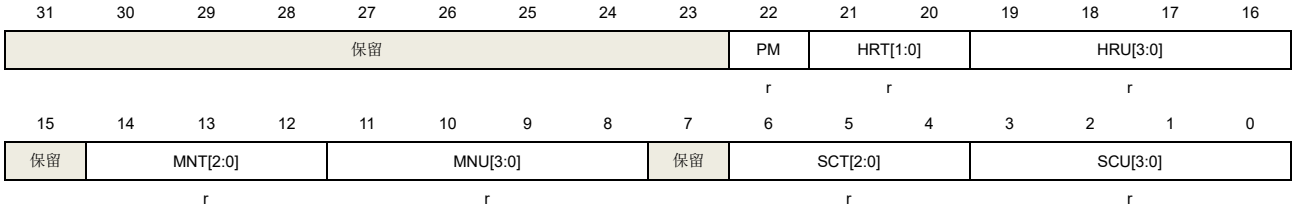
位/位域	名称	描述
31	A1S	增加一秒 0：无影响 1：增加一秒到时钟/日历 该位与 SFS 位一起使用，增加小于一秒到当前时间。
30:15	保留	必须保持复位值。
14:0	SFS[14:0]	减去小于一秒的一段时间 这位的值将增加到同步预分频计数器 当仅用 SFS 时，由于同步预分频器是一个递减计数器，所以时钟将会延迟。 延迟(秒) = SFS/(FACTOR_S + 1) 当 A1S 和 SFS 一起使用时，时钟将会提前 提前(秒) = (1 - (SFS/(FACTOR_S + 1)))

注意：写入此寄存器会导致 RSYNF 位被清 0。

22.4.12. 时间戳时间寄存器（RTC_TTS）

偏移地址：0x30
备份域复位值：0x0000 0000
系统复位：无影响
当TSF被置1，该位用来记录日历时间。
清除TSF位也会清除此寄存器。

该寄存器只能按字(32位)访问。

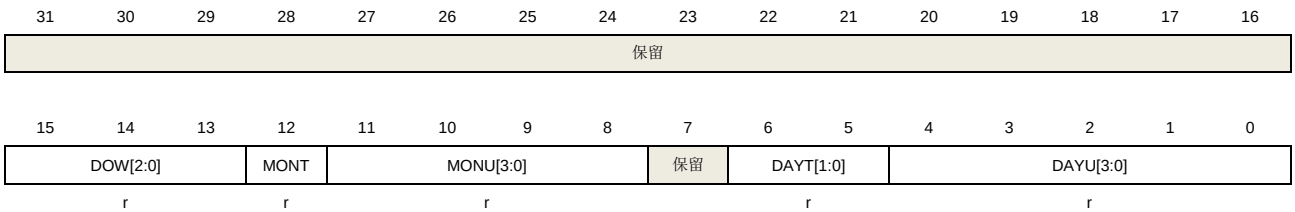


位/位域	名称	描述
31:23	保留	必须保持复位值。
22	PM	AM/PM 标记 0：AM 或 24 小时制 1：PM
21:20	HRT[1:0]	小时十位值，以 BCD 码形式存储
19:16	HRU[3:0]	小时个位值，以 BCD 码形式存储
15	保留	必须保持复位值。
14:12	MNT[2:0]	分钟十位值，以 BCD 码形式存储
11:8	MNU[3:0]	分钟个位值，以 BCD 码形式存储
7	保留	必须保持复位值。
6:4	SCT[2:0]	秒钟十位值，以 BCD 码形式存储
3:0	SCU[3:0]	秒钟个位值，以 BCD 码形式存储

22.4.13. 时间戳日期寄存器（RTC_DTS）

偏移地址：0x34
备份域复位值：0x0000 0000
系统复位：无影响
当TSF被置1，该位用来记录日历日期。
清除TSF位也会清除此寄存器。

该寄存器只能按字(32位)访问。

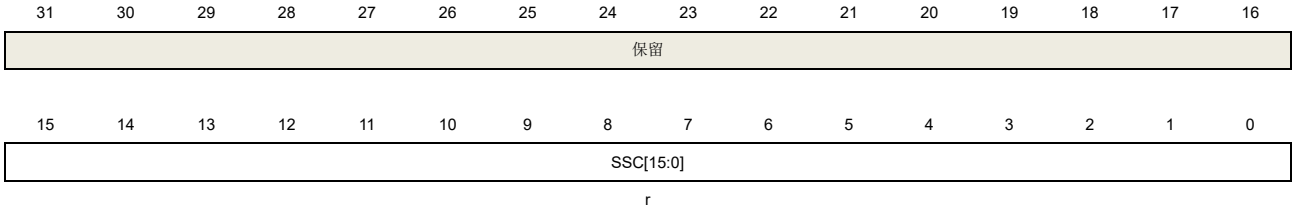


位/位域	名称	描述
31:16	保留	必须保持复位值。
15:13	DOW[2:0]	星期数
12	MONT	月份十位值，以 BCD 码形式存储
11:8	MONU[3:0]	月份个位值，以 BCD 码形式存储
7	保留	必须保持复位值。
6:5	DAYT[1:0]	日期十位值，以 BCD 码形式存储
4:0	DAYU[3:0]	日期个位值，以 BCD 码形式存储

22.4.14. 时间戳亚秒寄存器（RTC_SSTS）

偏移地址：0x38
备份域复位：0x0000 0000
系统复位：无影响
当TSF被置1，该位用来记录日历时间。
清除TSF位也会清除此寄存器。

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	SSC[15:0]	亚秒值 TSF 置 1 时记录当时的同步预分频计数器的值。

22.4.15. 高精度频率补偿寄存器（RTC_HRFC）

偏移地址：0x3C
备份域复位：0x0000 0000
系统复位：无影响
写保护寄存器。

该寄存器只能按字(32位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FREQI	CWND8	CWND16	保留				CMSK[8:0]								
rw	rw	rw					rw								

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	FREQI	RTC 频率增加 488.5ppm 0：无影响 1：每 2^{11} 个脉冲增加一个 RTCCLK 脉冲 该位需与 CMSK 位一起使用。如果输入时钟频率是 32.768KHz，在 32s 校准窗期间，增加的 RTCCLK 脉冲数是(512 * FREQI) - CMSK
14	CWND8	采用 8 秒校准周期 0：无影响 1：采用 8 秒校准周期 注意：当 CWND8=1，CMSK[1：0]被锁定在“00”。
13	CWND16	采用 16 秒校准周期 0：无影响 1：采用 16 秒校准周期 注意：当 CWND16=1，CMSK[0] 被锁定在“0”.
12:9	保留	必须保持复位值。
8:0	CMSK[8:0]	校准周期 RTCCLK 脉冲屏蔽数 在 2^{20} 个 RTCCLK 脉冲之内屏蔽的脉冲数 此项功能可以以 0.9537 ppm 的分辨率来降低日历频率

22.4.16. 侵入寄存器（RTC_TAMP）

偏移地址：0x40
备份域复位：0x0000 0000
系统复位：无影响

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		TP2IE	TP1IE	TP0IE	保留	TP2MASK	TP1MASK	TP0MASK	保留	TP2 NOERASE	TP1 NOERASE	TP0 NOERASE	保留		TP2_DISPIN
		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
DISPU	PRCH[1:0]		FLT[1:0]		FREQ[2:0]s			TPTS	TP2EG	TP2EN	TP1EG	TP1EN	TPIE	TP0EG	TP0EN
rw	rw		rw		rw			rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:30	保留	Must be kept at reset value.
29	TP2IE	Tamper 2 中断使能 0: 禁用侵入 2 中断 1: 启用侵入 2 中断
28	TP1IE	Tamper 1 中断使能 0: 禁用 Tamper 1 中断 1: 启用 Tamper 1 中断
27	TP0IE	侵入 0 中断使能 0: 禁用 Tamper 0 中断 1: 启用 Tamper 0 中断
26	保留	必须保持复位值
25	TP2MASK	Tamper 2 掩码标志 0: Tamper 2事件产生一个触发事件，TP2F必须被软件清除，以允许下一个侵入事件检测 1: Tamper 2 事件产生一个触发事件，TP2F 被硬件屏蔽并在内部清除。备份寄存器不会被清除 **Note:** 当 TP2MASK 被置位时，必须禁用 Tamper 2 中断
24	TP1MASK	Tamper1 掩码标志 0: Tamper1事件产生一个触发事件，TP2F必须被软件清除，以允许下一个侵入事件检测 1: Tamper1 事件产生一个触发事件，TP2F 被硬件屏蔽并在内部清除。备份寄存器不会被清除 **Note:** 当 TP2MASK 被置位时，必须禁用 Tamper 1 中断
23	TP0MASK	Tamper0 掩码标志 0: Tamper0事件产生一个触发事件，TP2F必须被软件清除，以允许下一个侵入事件检测 1: Tamper0 事件产生一个触发事件，TP2F 被硬件屏蔽并在内部清除。备份寄存器不会被清除 **Note:** 当 TP2MASK 被置位时，必须禁用 Tamper 0 中断
22	保留	必须保持复位值

21	TP2 NOERASE	Tamper2 不擦除 0: Tamper2 擦除备份寄存器 1: Tamper2 不擦除备份寄存器
20	TP1 NOERASE	Tamper1 不擦除 0: Tamper1 擦除备份寄存器 1: Tamper1 不擦除备份寄存器
19	TP0 NOERASE	Tamper0 不擦除 0: Tamper0 擦除备份寄存器 1: Tamper0 不擦除备份寄存器
18:17	保留	必须保持复位值
16	TP2_DISPIN	Tamper2 选择 0: 引脚触发 Tamper2 1: 引脚不触发 Tamper2
15	DISPU	RTC_TAMPx 上拉禁用位 0：使能内部 RTC_TAMPx 引脚上的上拉电阻并在采样前进行预充电 1：禁用预充电功能
14:13	PRCH[1:0]	RTC_TAMPx 的预充电时间 该位设置决定了每次采样前的预充电时间 0x0：1 个 RTC 时钟 0x1：2 个 RTC 时钟 0x2：4 个 RTC 时钟 0x3：8 个 RTC 时钟
12:11	FLT[1:0]	RTC_TAMPx 过滤器计数设置 该位决定了侵入事件检测模式和在电平检测模式下连续采样的次数。 0x0：用边沿模式检测侵入事件，预充电功能被自动禁用。 0x1：用电平模式检测侵入事件。连续采样到 2 个有效电平时认为发生侵入事件 0x2：用电平模式检测侵入事件。连续采样到 4 个有效电平时认为发生侵入事件 0x3：用电平模式检测侵入事件。连续采样到 8 个有效电平时认为发生侵入事件
10:8	FREQ[2:0]	侵入事件电平模式检测的采样频率 0x0：每次采样间隔 32768 个 RTCCLK(若 RTCCLK=32.768KHz，频率为 1Hz) 0x1：每次采样间隔 16384 个 RTCCLK(若 RTCCLK=32.768KHz，频率为 2Hz) 0x2：每次采样间隔 8192 个 RTCCLK(若 RTCCLK=32.768KHz，频率为 4Hz) 0x3：每次采样间隔 4096 个 RTCCLK(若 RTCCLK=32.768KHz，频率为 8Hz) 0x4：每次采样间隔 2048 个 RTCCLK(若 RTCCLK=32.768KHz，频率为 16Hz) 0x5：每次采样间隔 1024 个 RTCCLK(若 RTCCLK=32.768KHz，频率为 32Hz) 0x6：每次采样间隔 512 个 RTCCLK(若 RTCCLK=32.768KHz，频率为 64Hz)

		0x7：每次采样间隔 256 个 RTCCLK(若 RTCCLK=32.768KHz，频率为 128Hz)
7	TPTS	侵入事件时触发时间戳 0：无影响 1：当检测到侵入事件时，即使 TSEN=0，TSF 也会被置位
6	TP2EG	Tamper2 输入管脚的侵入事件检测触发沿 如果侵入检测处于边沿模式(FLT =0)： 0：上升沿触发一个侵入检测事件 1：下降沿触发一个侵入检测事件 如果侵入检测处于电平模式(FLT !=0)： 0：低电平触发一个侵入检测事件 1：高电平触发一个侵入检测事件
5	TP2EN	Tamper2 检测使能位 0：禁用 Tamper2 检测功能 1：启用 Tamper2 检测功能
4	TP1EG	Tamper1 输入管脚的侵入事件检测触发沿 如果侵入检测处于边沿模式(FLT =0)： 0：上升沿触发一个侵入检测事件 1：下降沿触发一个侵入检测事件 如果侵入检测处于电平模式(FLT !=0)： 0：低电平触发一个侵入检测事件 1：高电平触发一个侵入检测事件
3	TP1EN	Tamper1 检测使能位 0：禁用 Tamper1 检测功能 1：启用 Tamper1 检测功能
2	TPIE	侵入检测中断使能 0：禁用 Tamper 中断 1：启用 Tamper 中断
1	TP0EG	Tamper0 输入管脚的侵入事件检测触发沿 如果侵入检测处于边沿模式(FLT =0)： 0：上升沿触发一个侵入检测事件 1：下降沿触发一个侵入检测事件 如果侵入检测处于电平模式(FLT !=0)： 0：低电平触发一个侵入检测事件 1：高电平触发一个侵入检测事件
0	TP0EN	Tamper0 检测使能位 0：禁用 Tamper0 检测功能

1：启用 Tamper0 检测功能

注意：强烈建议在改变侵入检测配置之前，应该复位 TPxEN 位。

22.4.17. 闹钟 0 亚秒寄存器（RTC_ALRM0SS）

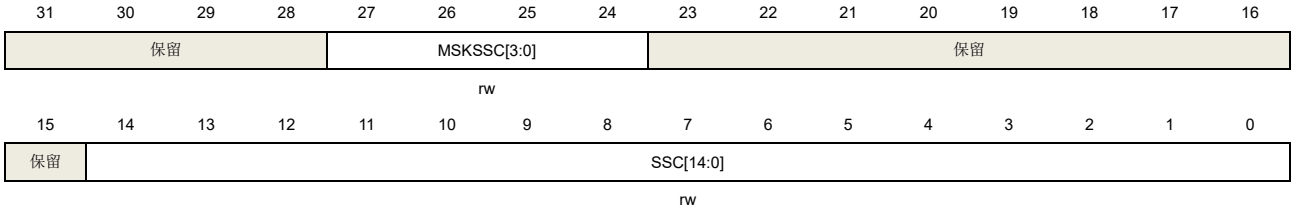
偏移地址： 0x44

备份域复位：0x0000 0000

系统复位：无影响

写保护寄存器，仅当ALRM0EN=0或INITM=1，可以进行写操作。

该寄存器只能按字(32位)访问。



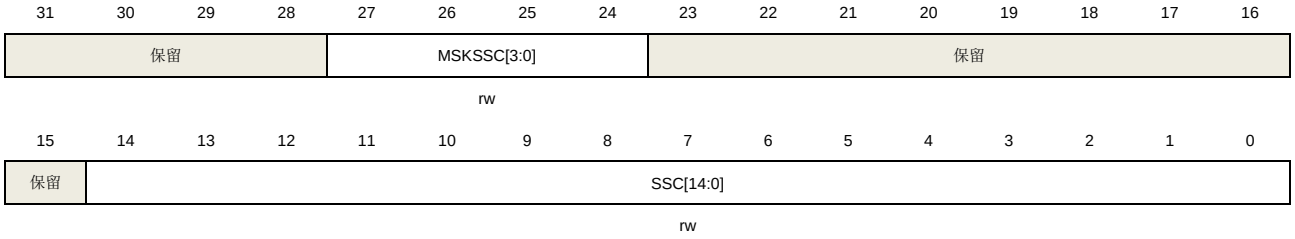
位/位域	名称	描述
31:28	保留	必须保持复位值。
27:24	MSKSSC[3:0]	亚秒位域的屏蔽控制位 0x0：屏蔽闹钟亚秒设置。当所有其他的闹钟位域匹配的时候，闹钟将会在每一秒钟到达的时刻置 1。 0x1：SSC[0]位用于时间匹配，其他位被忽略。 0x2：SSC[1：0]位用于时间匹配，其他位被忽略。 0x3：SSC[2：0]位用于时间匹配，其他位被忽略。 0x4：SSC[3：0]位用于时间匹配，其他位被忽略。 0x5：SSC[4：0]位用于时间匹配，其他位被忽略。 0x6：SSC[5：0]位用于时间匹配，其他位被忽略。 0x7：SSC[6：0]位用于时间匹配，其他位被忽略。 0x8：SSC[7：0]位用于时间匹配，其他位被忽略。 0x9：SSC[8：0]位用于时间匹配，其他位被忽略。 0xA：SSC[9：0]位用于时间匹配，其他位被忽略。 0xB：SSC[10：0]位用于时间匹配，其他位被忽略。 0xC：SSC[11：0]位用于时间匹配，其他位被忽略。 0xD：SSC[12：0]位用于时间匹配，其他位被忽略。 0xE：SSC[13：0]位用于时间匹配，其他位被忽略。 0xF：SSC[14：0]位用于时间匹配，其他位被忽略。 注意：同步预分频计数器的第 15 位(RTC_SS 寄存器中的 SSC[15])从不被匹配。
23:15	保留	必须保持复位值。

14:0	SSC[14:0]	闹钟亚秒值 该值为闹钟亚秒值，用于与同步预分频计数器匹配。 匹配位数由 MSKSSC 位控制。

22.4.18. 闹钟 1 亚秒寄存器（RTC_ALRM1SS）

偏移地址：0x48
备份域复位：0x0000 0000
系统复位：无影响
写保护寄存器，仅当 ALRM1EN=0 或 INITM=1，可以进行写操作。

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:28	保留	必须保持复位值。
27:24	MSKSSC[3:0]	亚秒位域的屏蔽控制位 0x0：屏蔽闹钟亚秒设置。当所有其他的闹钟位域匹配的时候，闹钟将会在每一秒钟到达的时刻置 1。 0x1：SSC[0]位用于时间匹配，其他位被忽略。 0x2：SSC[1：0] 位用于时间匹配，其他位被忽略。 0x3：SSC[2：0] 位用于时间匹配，其他位被忽略。 0x4：SSC[3：0] 位用于时间匹配，其他位被忽略。 0x5：SSC[4：0] 位用于时间匹配，其他位被忽略。 0x6：SSC[5：0] 位用于时间匹配，其他位被忽略。 0x7：SSC[6：0] 位用于时间匹配，其他位被忽略。 0x8：SSC[7：0] 位用于时间匹配，其他位被忽略。 0x9：SSC[8：0] 位用于时间匹配，其他位被忽略。 0xA：SSC[9：0] 位用于时间匹配，其他位被忽略。 0xB：SSC[10：0] 位用于时间匹配，其他位被忽略。 0xC：SSC[11：0] 位用于时间匹配，其他位被忽略。 0xD：SSC[12：0] 位用于时间匹配，其他位被忽略。 0xE：SSC[13：0] 位用于时间匹配，其他位被忽略。 0xF：SSC[14：0] 位用于时间匹配，其他位被忽略。

		注意：同步预分频计数器的第 15 位(RTC_SS 寄存器中的 SSC[15])从不被匹配。
23:15	保留	必须保持复位值。
14:0	SSC[14:0]	闹钟亚秒值 该值为闹钟亚秒值，用于与同步预分频计数器匹配。 匹配位数由 MSKSSC 位控制。

22.4.19. 配置寄存器（RTC_CFG）

偏移地址：0x4C
备份域复位：0x0000 0000
系统复位：无影响

该寄存器只能按字(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														OUT2EN	ALRMOUTTYPE
														rw	rw

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	OUT2EN	RTC_OUT 引脚选择 0：RTC_OUT 输出到 PC13 1：RTC_OUT 输出到 PB2
0	ALRMOUTTYPE	RTC_ALARM 输出类型 0：开漏输出 1：推挽输出

22.4.20. 备份寄存器（RTC_BKPx）（x=0..31）

偏移地址： 0x50 到 0xCC
备份域复位：0x0000 0000
系统复位：无影响

该寄存器只能按字(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]	数据 软件可读写寄存器。由于此寄存器可由 V_{BAT} 供电，因此寄存器值在省电模式下依然保持有效。当侵入检测标志位 TPxF 置 1，这些寄存器会被复位。当 FMC 读保护功能禁用时，这些寄存器会被复位。

23. 定时器（TIMER）

表 23-1. 定时器（TIMERx）分为 5 种类型

定时器		定时器 0/7/19	定时器 1/2/3/4	定时器 14	定时器 15/16	定时器 5/6
类型		高级	通用 L0	通用 L3	通用 L4	基本
预分频器		16 位	16 位	16 位	16 位	16 位
计数器		16 位	16 位（TIMER2/3） 32 位（TIMER1/4）	16 位	16 位	16 位 （TIMER5/6）
计数模式		向上，向下，中央对齐	向上，向下，中央对齐	只有向上	只有向上	只有向上
可重复性		●	×	●	●	×
捕获/比较通道数		8	4	3	2	0
复合 PWM 模式		●	●	●	×	×
输出匹配脉冲选择		●	●	●	×	×
互补和死区时间		●	×	●	●	×
中止功能	BREAK0	●	×	●	●	×
	BREAK1	●	×	×	×	×
锁存中止功能		●	×	●	●	×
单脉冲		●	●	●	●	●
可延时的单脉冲		●	●	●	×	×
正交译码器		●	●	×	×	×
译码器		●	●	×	×	×
主-从管理		●	●	●	×	×
内部连接		● (1)	● (2)	● (3)	×	TRGO TO DAC
同步和初始值、计数方向刷新		●	×	×	×	×
DMA		●	●	●	●	● (4)
Debug 模式		●	●	●	●	●

	TIMERx	ITI0	ITI1	ITI2	ITI3	ITI4	ITI5	ITI6	ITI7	ITI8	ITI9	ITI10	ITI11	ITI12	ITI13	ITI14
(1)	TIMER0	-	TIMER1_	TIMER2_	TIMER3_	TIMER4	TIMER7_	TIMER14	TIMER15	TIMER16	TIMER19	HRTIME	-	-	-	from

	TIMERx	ITI0	ITI1	ITI2	ITI3	ITI4	ITI5	ITI6	ITI7	ITI8	ITI9	ITI10	ITI11	ITI12	ITI13	ITI14
			TRGO0	TRGO0	TRGO0	_TRGO0	TRGO0	_TRGO0	_CH0	_CH0	_TRGO0	R_SCOUT（具体可参考离分辨率定时器（HRTIMER）同步输入/输出章节）				TRIGSEL
	TIMER7	TIMER0_TRGO0	TIMER1_TRGO0	TIMER2_TRGO0	TIMER3_TRGO0	TIMER4_TRGO0	-	TIMER14_TRGO0	TIMER15_CH0	TIMER16_CH0	TIMER19_TRGO0		-	-	-	
	TIMER19	TIMER0_TRGO0	TIMER1_TRGO0	TIMER2_TRGO0	TIMER3_TRGO0	TIMER4_TRGO0	TIMER7_TRGO0	TIMER14_TRGO0	TIMER15_CH0	TIMER16_CH0	-		-	-	-	
(2)	TIMER1	TIMER0_TRGO0	-	TIMER2_TRGO0	TIMER3_TRGO0	TIMER4_TRGO0	TIMER7_TRGO0	TIMER14_TRGO0	TIMER15_CH0	TIMER16_CH0	TIMER19_TRGO0			-	-	
(2)	TIMER2	TIMER0_TRGO0	TIMER1_TRGO0	-	TIMER3_TRGO0	TIMER4_TRGO0	TIMER7_TRGO0	TIMER14_TRGO0	TIMER15_CH0	TIMER16_CH0	TIMER19_TRGO0		-			
(2)	TIMER3	TIMER0_TRGO0	TIMER1_TRGO0	TIMER2_TRGO0	-	TIMER4_TRGO0	TIMER7_TRGO0	TIMER14_TRGO0	TIMER15_CH0	TIMER16_CH0	TIMER19_TRGO0		-			
(2)	TIMER4	TIMER0_TRGO0	TIMER1_TRGO0	TIMER2_TRGO0	TIMER3_TRGO0	-	TIMER7_TRGO0	TIMER14_TRGO0	TIMER15_CH0	TIMER16_CH0	TIMER19_TRGO0		-			
(3)	TIMER14	TIMER0_TRGO0	TIMER1_TRGO0	TIMER2_TRGO0	TIMER3_TRGO0	TIMER4_TRGO0	TIMER7_TRGO0	-	TIMER15_CH0	TIMER16_CH0	TIMER19_TRGO0		-	-	-	
(4)	只有更新事件可以产生DMA请求。但是定时器5/6中没有DMA配置DMAS。															
(5)	TIMER5 / 6的TRGO和DAC连接由TRIGSEL模块配置。															

23.1. 高级定时器（TIMERx, x = 0, 7, 19）

23.1.1. 简介

高级定时器（TIMER0 / 7 /19）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。

23.1.2. 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：BREAK0和BREAK1；
- 中断输出和DMA请求：更新事件，触发事件，比较/捕获事件和中止事件；
- 多个定时器的菊链使得一个定时器可以同时启动多个定时器；
- 定时器的同步允许被选择的定时器在同一个时钟周期开始计数；
- 定时器主-从管理。

23.1.3. 结构框图

*图 23-1. 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 23-2. 高级定时器通道介绍*介绍了通道输入和输出情况。

图 23-1. 高级定时器结构框图

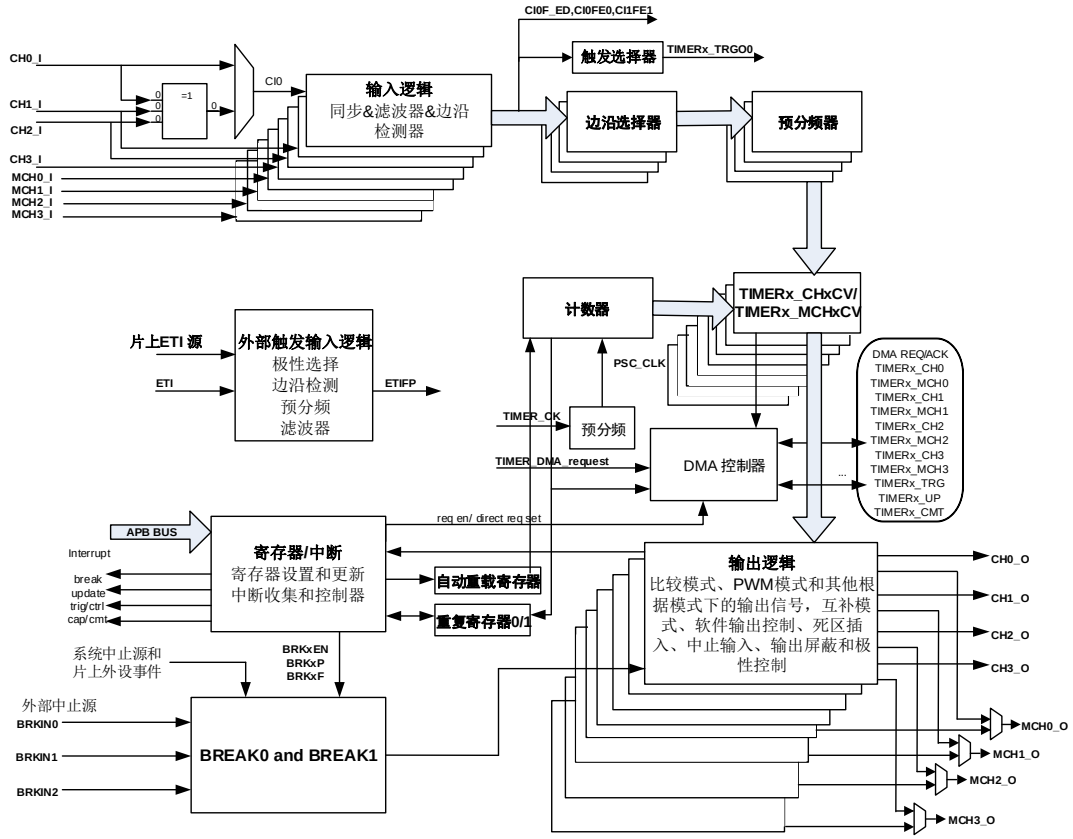


表 23-2. 高级定时器通道介绍

通道名称 （x = 0…3）	MCHxMSEL[1:0]=00 独立模式	MCHxMSEL[1:0]=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	只有 CHx 可用于输入，CHx 和 MCHx 输出互补
MCHx （多模式通道 x）		

23.1.4. 功能描述

时钟源选择

高级定时器可以由内部时钟源 CK_TIMER 或者由 SYSCFG_TIMERxCFG(x = 0, 7, 19)寄存器中的 TSCFGy[4:0] (y = 0…15)位域控制的复用时钟源驱动。

- 当SYSCFG_TIMERxCFG(x = 0, 7, 19)寄存器中的TSCFGy[4:0] = 5’b00000(y = 0…15)时，定时器选择内部时钟源（连接到RCU模块的CK_TIMER）

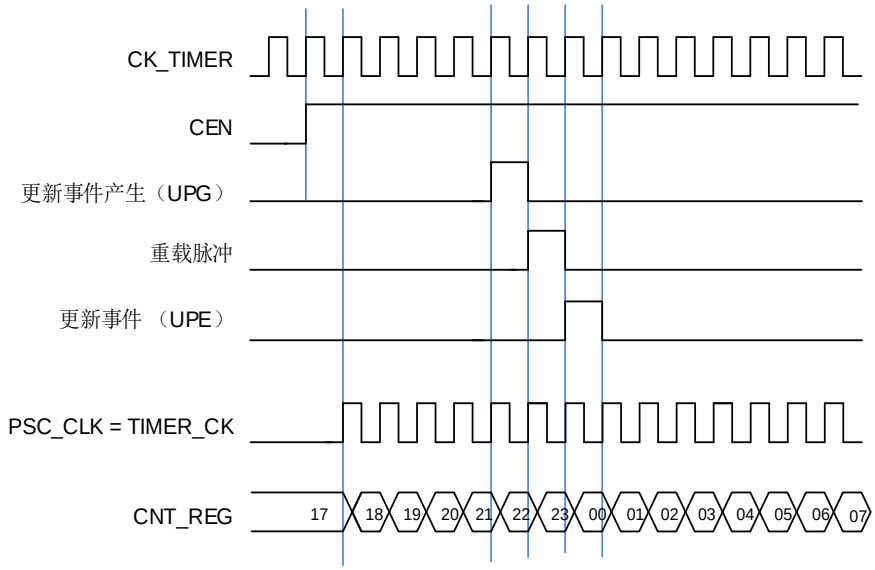
如果 SYSCFG_TIMERxCFG(x = 0, 7, 19)寄存器中的 TSCFGy[4:0] = 5’b00000(y = 0…15)，默认

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 SYSCFG_TIMERxCFG(x = 0, 7, 19)寄存器中的 TSCFGy[4:0] (y = 0...2, 6, 8, 9)位域设置为非零值，预分频器被其他时钟源驱动，具体在下文说明。当 TSCFGy[4:0] (y = 3, 4, 5, 7)被设置为非零值时，计数器预分频器时钟源由内部时钟 TIMER_CK 驱动。

图 23-2. 内部时钟分频为 1 时正常模式下的控制电路



- TSCFG6[4:0] != 5'b00000（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 CI0 / 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 TSCFG6[4:0]为 0x5 ~ 0x7 来选择。

计数器预分频器也可以在内部触发信号 ITI0 ~ 10 / ITI14 的上升沿计数。这种模式可以通过设置 TSCFG6[4:0]为 0x1 ~ 0x4，0x9 ~ 0xF 和 0x13 来选择。

- 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

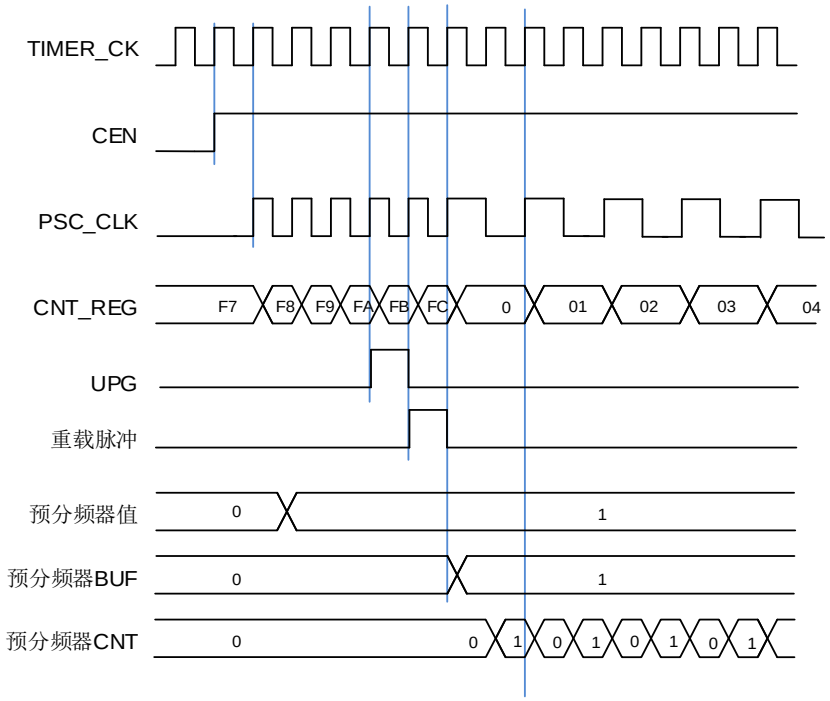
计数器预分频器可以在外部引脚 ETI 的每个上升沿或下降沿计数。这种模式可以通过设置 TIMERx_SMCFG 寄存器中的 SMC1 位为 1 来选择。另一种选择 ETI 信号作为时钟源方式是，设置 TSCFG6[4:0]为 0x8。注意 ETI 信号是通过数字滤波器采样 ETI 引脚得到的。如果选择 ETI 信号为时钟源，触发控制器包括边沿监测电路将在每个 ETI 信号上升沿产生一个时钟脉冲来为计数器预分频器提供时钟。

注意：ETI 信号可以从外部 ETI 引脚输入，也可由片上外设提供，具体情况可以参考 ***TIMER0_ETI 触发选择寄存器(TRIGSEL_TIMER0ETI)***、***TIMER7_ETI 触发选择寄存器(TRIGSEL_TIMER7ETI)*** 和 ***TIMER19_ETI 触发选择寄存器（TRIGSEL_TIMER19ETI）*** 模块。

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 23-3. 当预分频器的参数从 1 变到 2 时，计数器的时序图



向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从 0 开始向上连续计数到自动加载值（定义在 TIMERx_CAR 寄存器中），一旦计数器计数到自动加载值，会重新从 0 开始向上计数。如果设置了重复计数器，在（TIMERx_CREP0 / 1 + 1）次上溢后产生更新事件，否则在每次上溢时都会产生更新事件。在向上计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 0。

当通过 TIMERx_SWEVG 寄存器的 UPG 位置 1 来设置更新事件时，计数值会被清 0，并产生更新事件。

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

当发生更新事件时，所有的寄存器（重复计数器，自动重载寄存器，预分频寄存器）都将被更新。

图 23-4. 向上计数时序图，PSC = 0/2 和 *图 23-5. 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 23-4. 向上计数时序图，PSC = 0/2

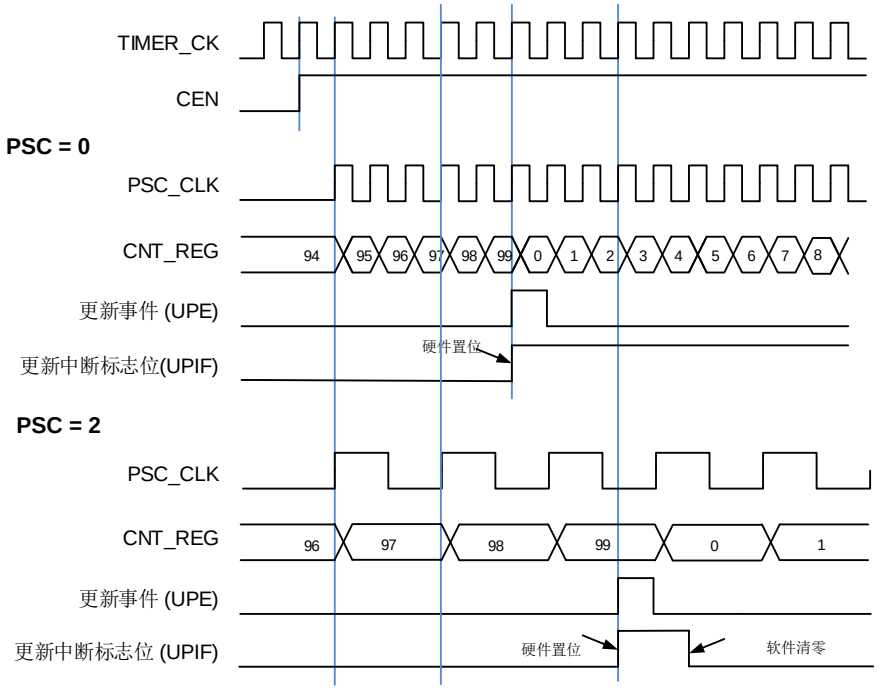
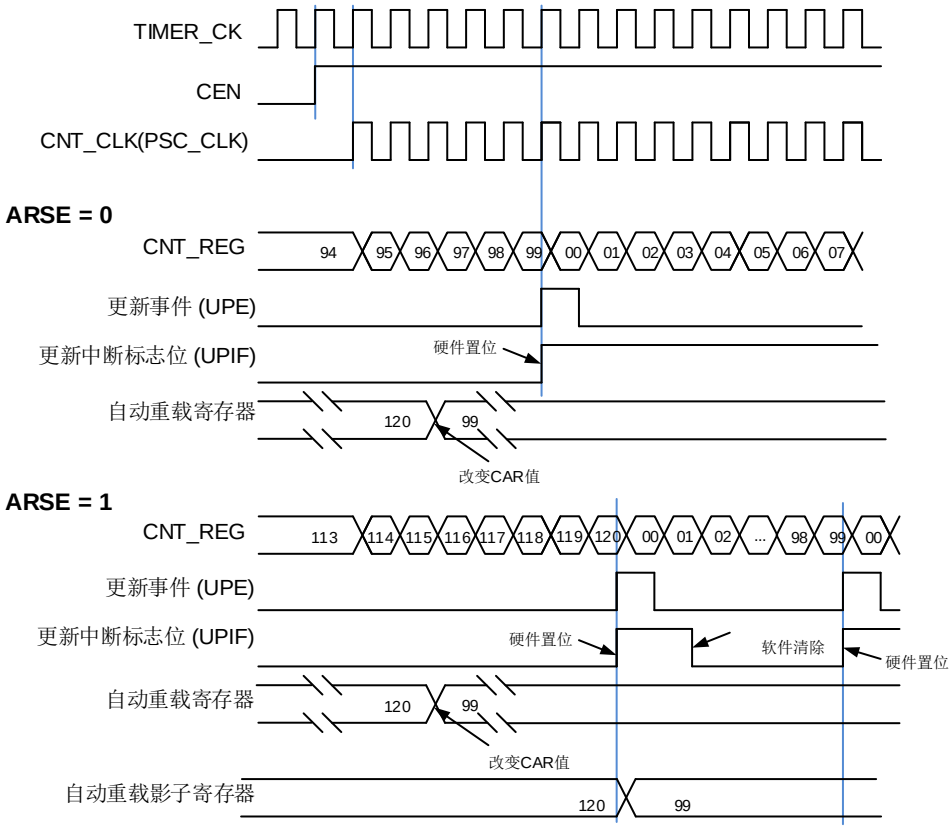


图 23-5. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



向下计数模式

在这种模式，计数器的计数方向是向下计数。计数器从自动加载值（定义在 TIMERx_CAR 寄存器中）向下连续计数到 0。一旦计数器计数到 0，计数器会重新从自动加载值开始计数。如果设置了重复计数器，在（TIMERx_CREP0 / 1 + 1）次下溢后产生更新事件，否则在每次下溢时都会产生更新事件。在向下计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 1。

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

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

当发生更新事件时，所有的寄存器（重复计数器，自动重载寄存器，预分频寄存器）都将被更新。

图 23-6. 向下计数时序图，PSC = 0/2 和 *图 23-7. 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 23-6. 向下计数时序图，PSC = 0/2

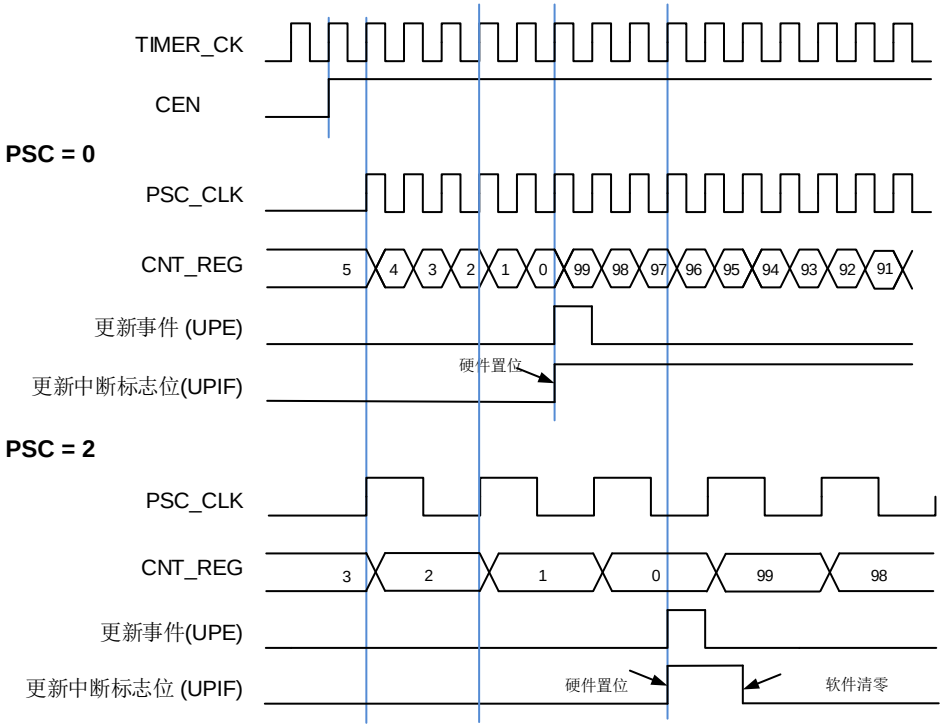
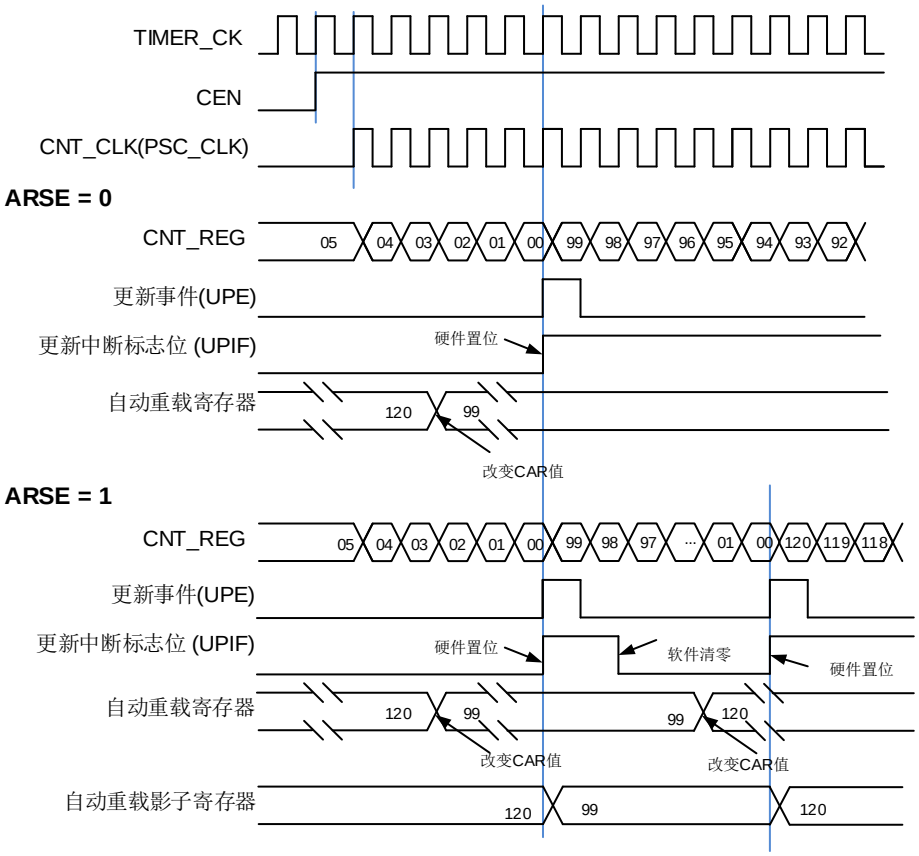


图 23-7. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值



中央对齐模式

在中央对齐模式下，计数器交替的从 0 开始向上计数到自动加载值，然后再向下计数到 0。向上计数模式中，定时器模块在计数器计数到（TIMERx_CAR - 1）产生一个上溢事件；向下计数模式中，定时器模块在计数器计数到 1 时产生一个下溢事件。在中央计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 只读，表明了的计数方向。计数方向被硬件自动更新。

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。

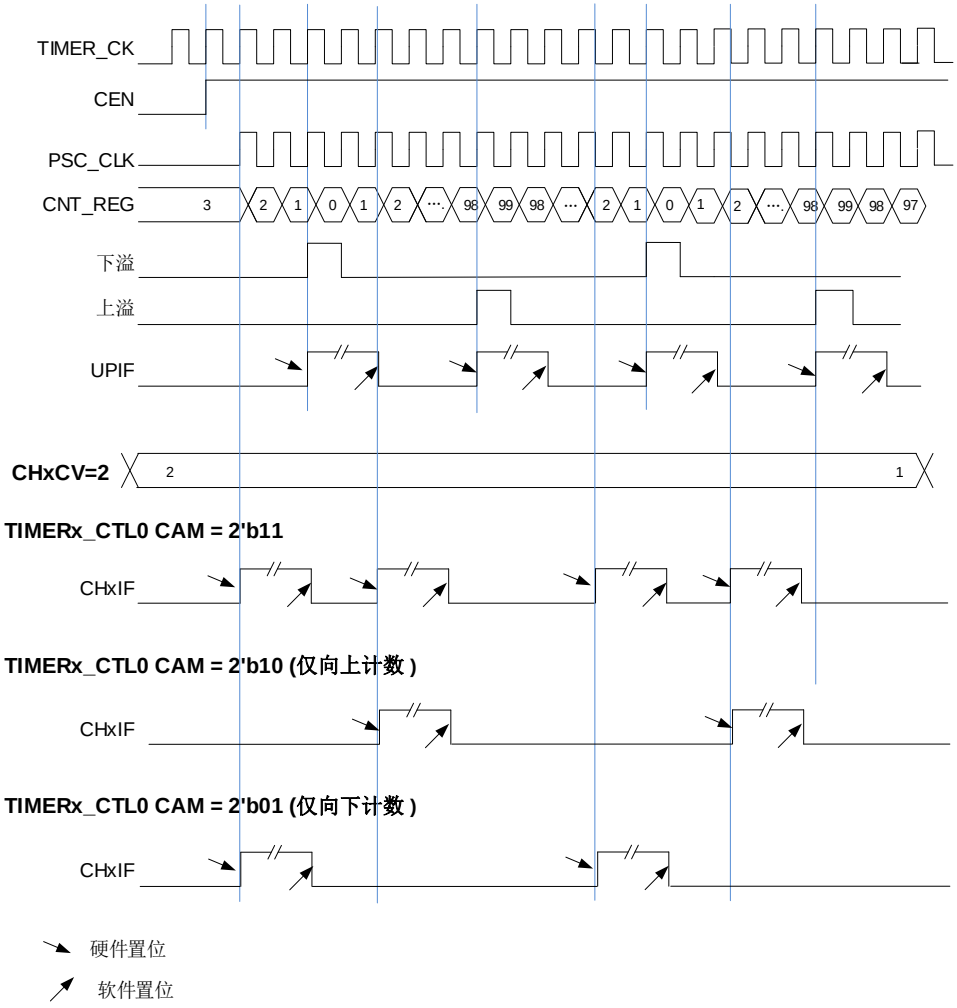
上溢或者下溢时，TIMERx_INTF 寄存器中的 UPIF 位都会被置 1，然而 CHxIF 位置 1 与 TIMERx_CTL0 寄存器中 CAM 的值有关。具体细节参考*图 23-8. 中央计数模式计数器时序图*。

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，所有的寄存器（重复计数器，自动重载寄存器，预分频寄存器）都将被更新。

*图23-8. 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 23-8. 中央计数模式计数器时序图



重复计数器

高级定时器有两个重复寄存器TIMERx_CREP0 / 1，可通过配置TIMERx_CFG寄存器中的CREPSEL位来选择。其中TIMERx_CREP0寄存器中的CREP0[7:0]是8位的，TIMERx_CREP1寄存器中的CREP1[31:0]是32位，用户可根据需求选择使用。

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_CREP0 / 1 寄存器的 CREP0 / 1 的值。向上计数模式下，重复计数器在每次计数器上溢时递减；向下计数模式下，重复计数器在每次计数器下溢时递减；在中央对齐模式下，重复计数器在计数器上溢和下溢时递减。

将 TIMERx_SWEVG 寄存器的 UPG 位置 1 可以重载 TIMERx_CREP0 / 1 寄存器中 CREP0 / 1 的值并产生一个更新事件。

新写入的 CREP0 / 1 值将在下一次更新事件到来时生效。当 CREP0 / 1 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 CREP0 / 1 值何时生效。如果在写入奇数到 CREP0 / 1 寄存器后由软件生成更新事件（UPG 位置 1），则在下溢时产生更新事件。如果在写入奇数到 CREP0 / 1 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 23-9. 中央计数模式下计数器重复时序图

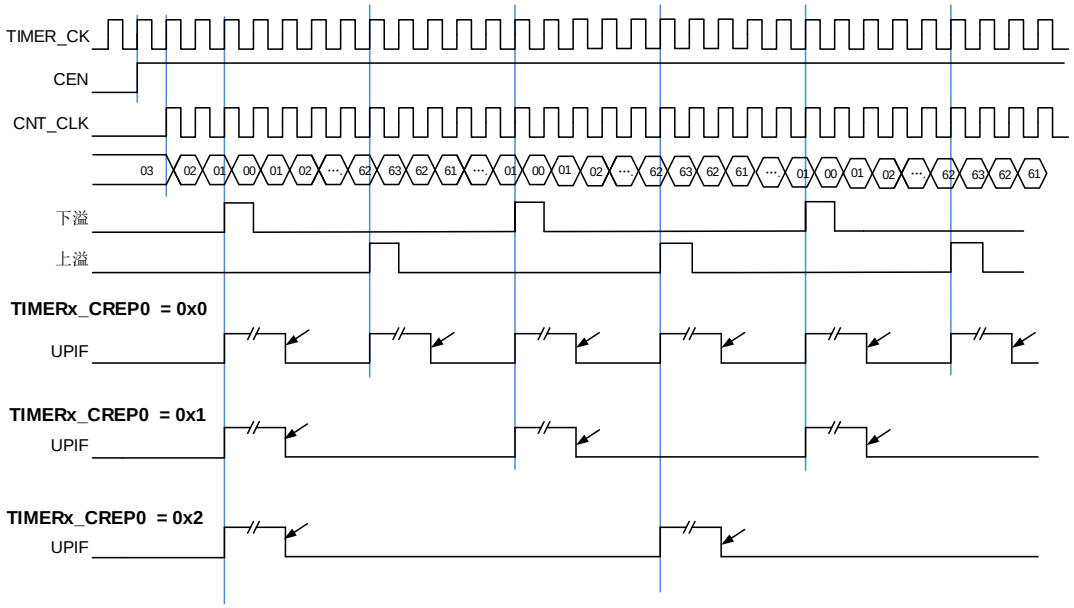


图 23-10. 在向上计数模式下计数器重复时序图

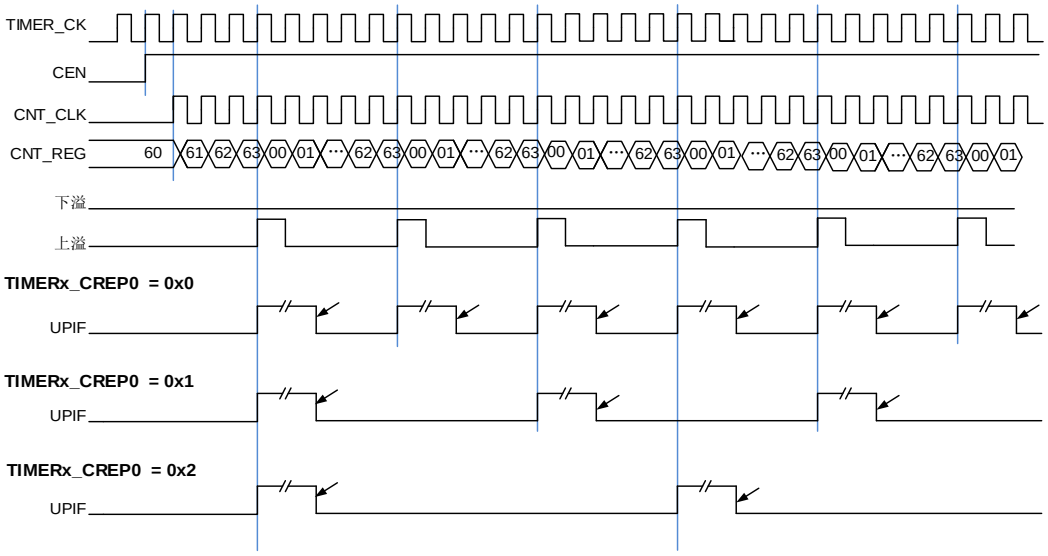
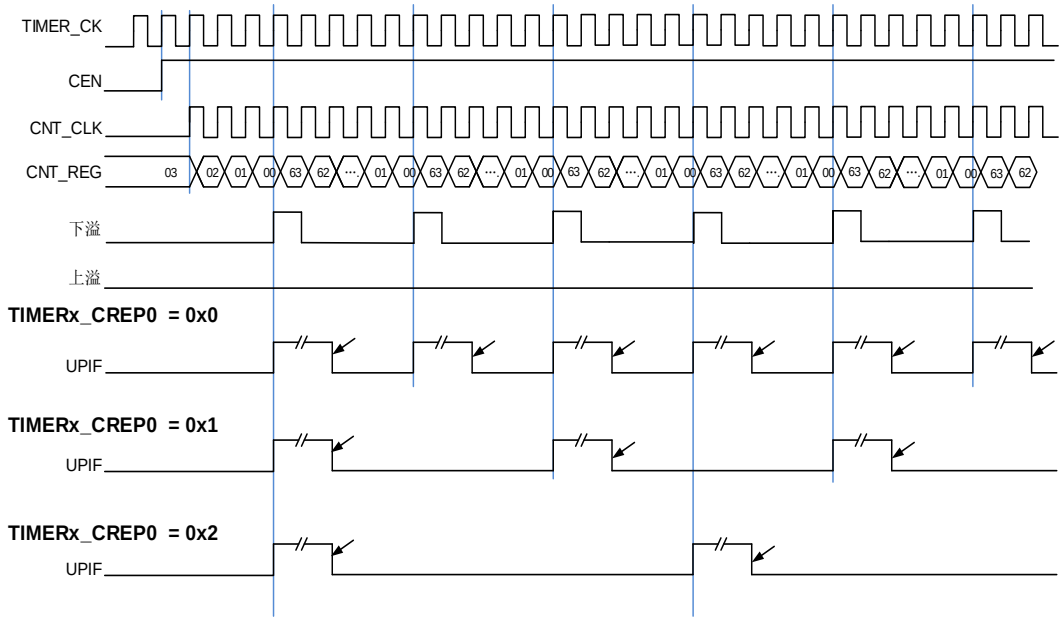


图 23-11. 在向下计数模式下计数器重复时序图



捕获 / 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 可输出独立和互补。

- 输入捕获模式

当 MCHxMSEL = 2'b00（独立模式）时，通道 x 和多模式通道 x 才可以独立进行输入捕获。

捕获模式允许通道测量一个波形时序，频率，周期，占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，TIMERx_CHxCV / 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 23-12. 通道 0 输入捕获逻辑

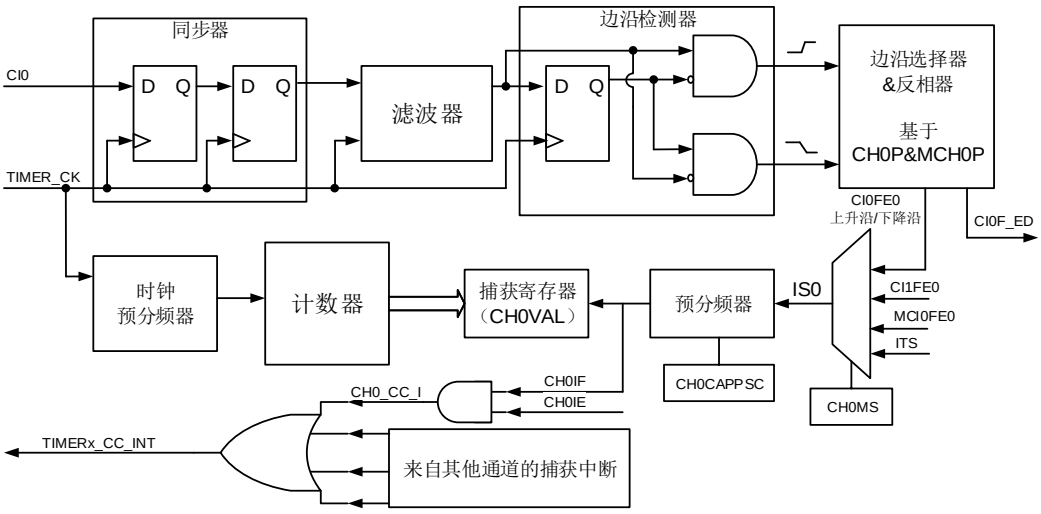
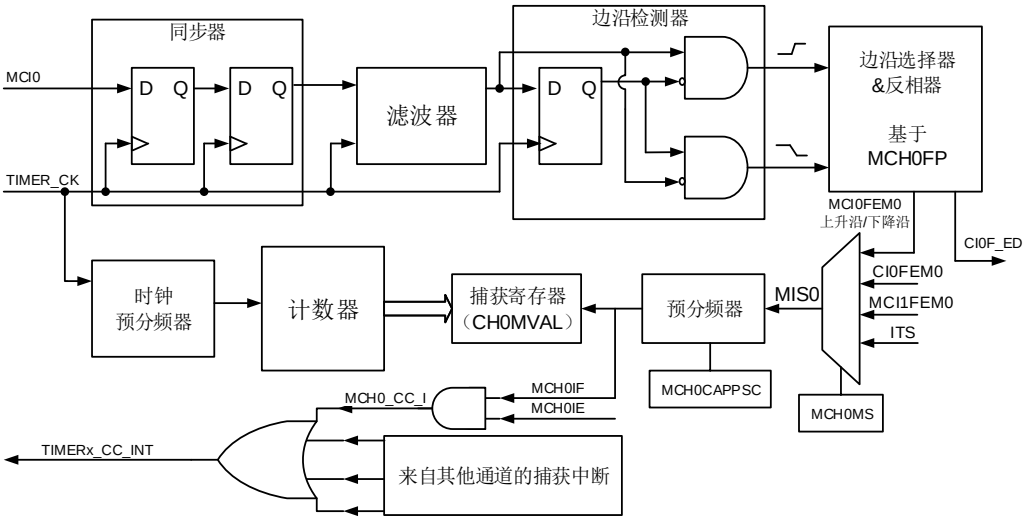


图 23-13. 多模式通道 0 输入捕获逻辑



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。

输入捕获模式也可用来测量 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 23-14. 输出比较逻辑（当 MCHxMSEL = 2'b00 时，x = 0,1,2,3）*和*图 23-15. 输出比较逻辑（当 MCHxMSEL = 2'b11 时，x = 0,1,2,3）*给出了通道的输出比较逻辑。

图 23-14. 输出比较逻辑（当 MCHxMSEL = 2'b00 时，x = 0,1,2,3）

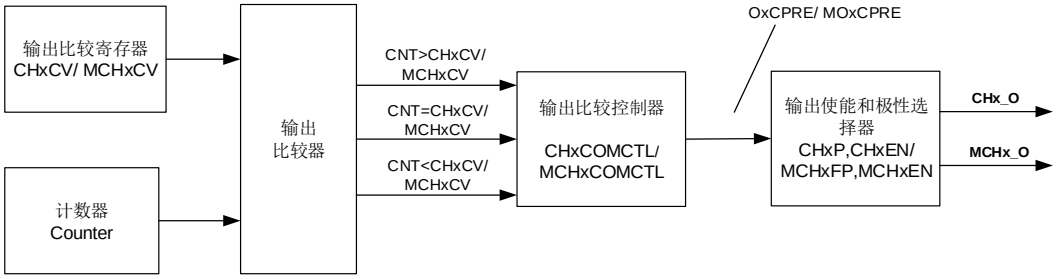
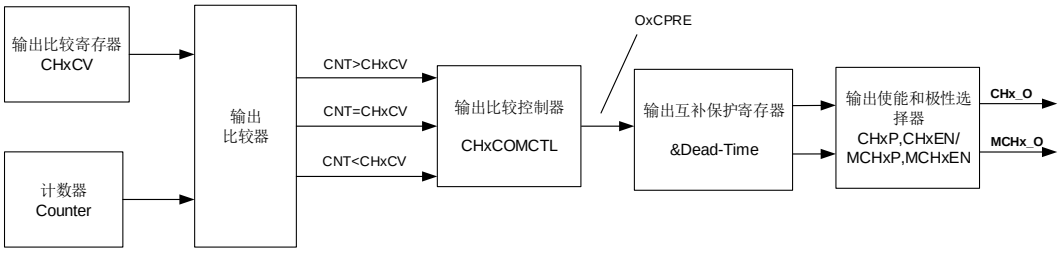


图 23-15. 输出比较逻辑（当 MCHxMSEL = 2'b11 时，x = 0,1,2,3）



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和TIMERx_MCHCTL2寄存器）。请参考*图23-14. 输出比较逻辑（当MCHxMSEL = 2'b00时，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23-15. 输出比较逻辑（当MCHxMSEL = 2'b11时，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_CCHP0寄存器中的相关位（ROS、IOS、POE和DTCFG等位）有关。详情请见*互补输出*。

在输出比较模式，TIMERx 可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的 TIMERx_CHxCV / 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。

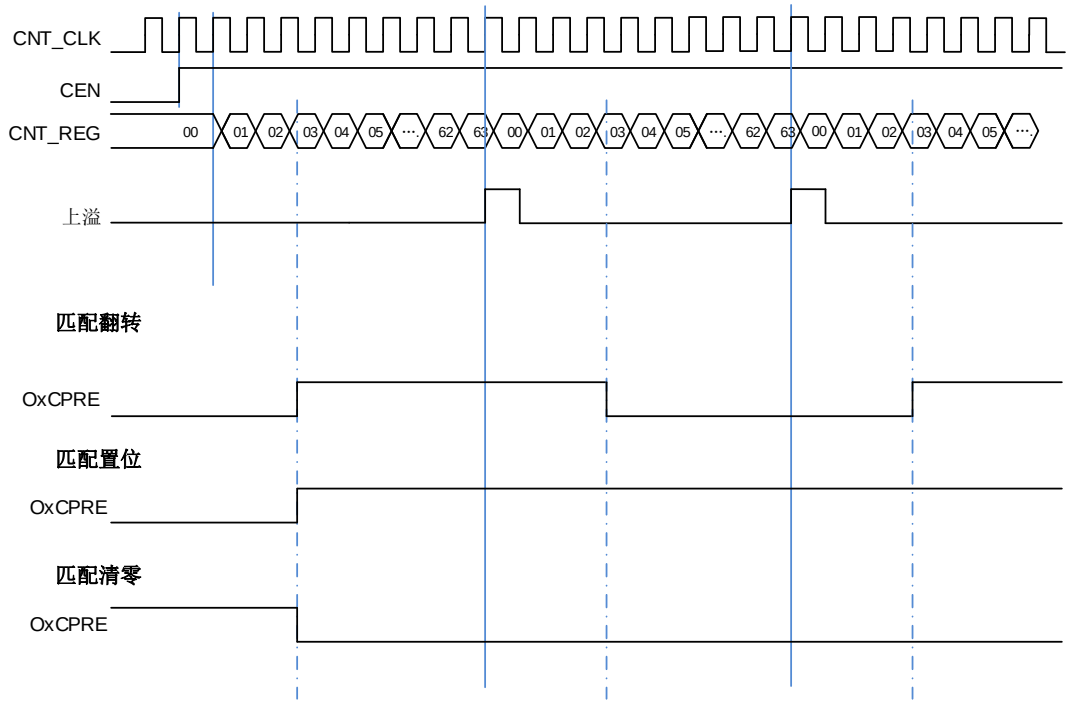
第三步：通过CHxIE / MCHxIE / CHxDEN / MCHxDEN位配置中断 / DMA请求使能。

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23-16. 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 23-16. 三种输出比较模式



PWM 模式

在 PWM 输出模式下（PWM 模式 0 是配置 CHxCOMCTL/ MCHxCOMCTL 为 4'b0110，PWM 模式 1 是配置 CHxCOMCTL/ MCHxCOMCTL 为 4'b0111），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23-17. 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23-18. 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 =4'b0110），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=4'b0111），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 23-17. EAPWM 时序图

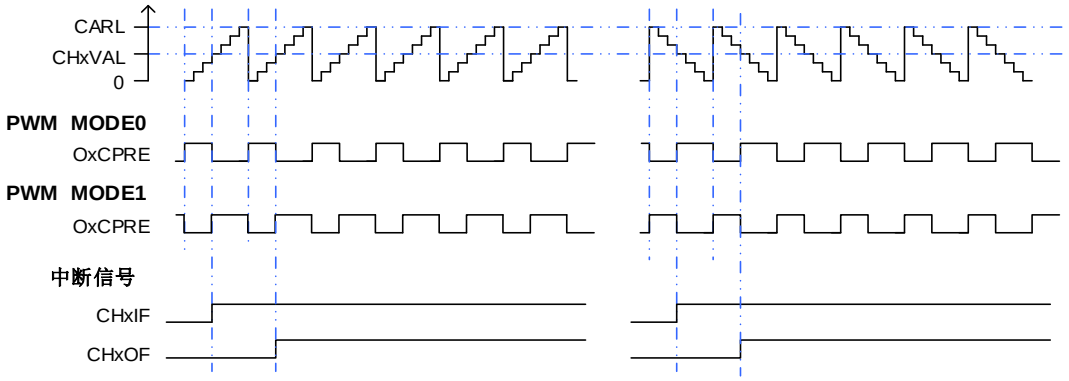
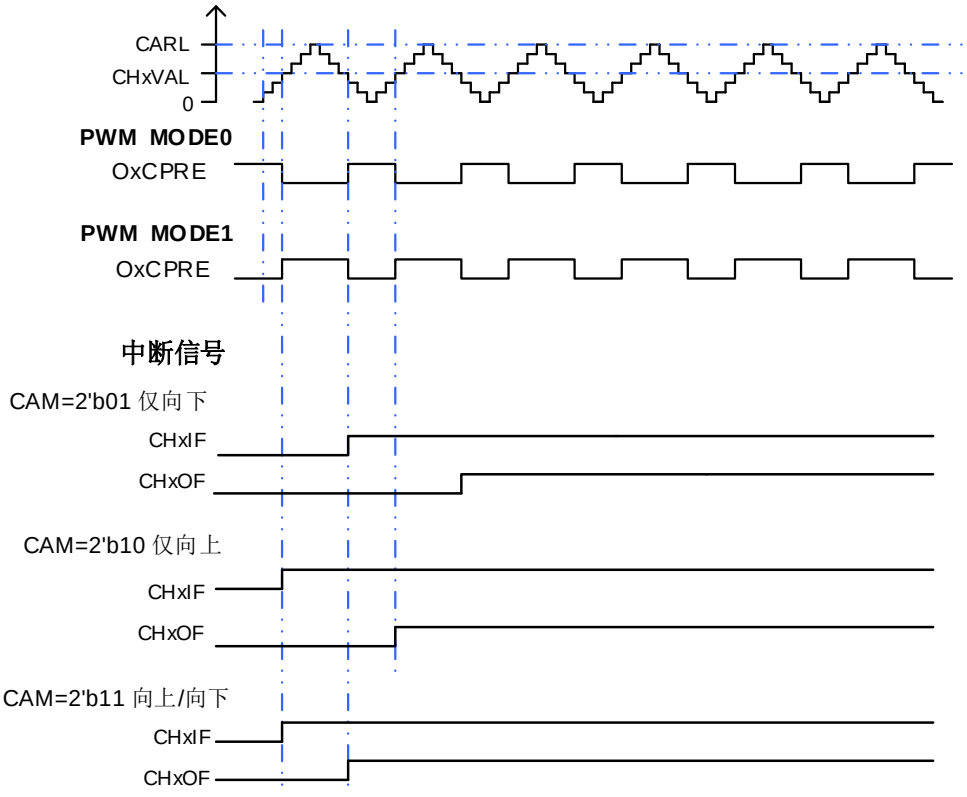


图 23-18. CAPWM 时序图

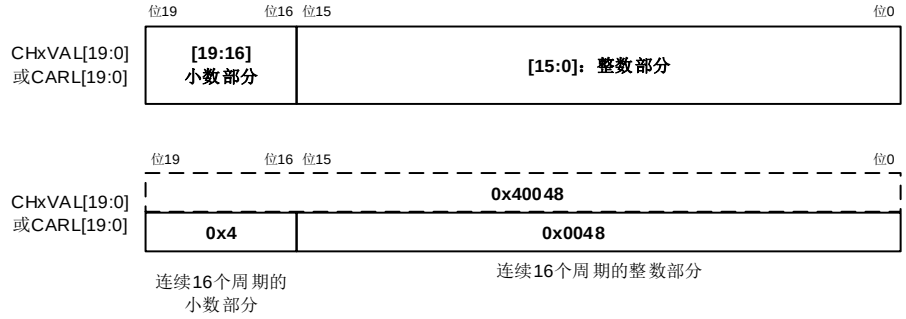


微调模式

通过配置TIMERx_CTL0寄存器中的ADMEN位为1，可以使能微调模式。该模式可以提高输出PWM波的有效分辨率，通过TIMERx_CHxCV寄存器中的CHxVAL[19:0]位域可以提高占空比分辨率，通过TIMERx_CAR寄存器中的CARL[19:0]位域可以提高PWM频率的分辨率。

当微调模式使能时，CHxVAL位域和CARL位域的低16位[15:0]用于整数部分，高4位[19:16]用于微调的小数部分。通过预定义的方式，在连续16个周期内对CHxVAL值或CARL值进行微调（每次调整不超过一个TIMER时钟周期），可增加16倍的分辨率。

图 23-19. 微调模式：数据格式和寄存器位域



根据ADMEN位的配置（置位或清零），CHxVAL位域和CARL位域将自动更新。当需要对ADMEN位进行清零时，需要遵循以下步骤：

1. CEN位和ARSE位必须清零；
2. CARL[19:16]位域必须清零；
3. ADMEN位必须清零；
4. CHxIF位必须清零；
5. 可以将CEN位置1。

以下公式可以计算PWM分辨率：

$$Resolution = f_{PSC_CLK} \ / \ f_{pwm} \quad (23\text{-}1)$$

由式*(23-1)*可得，微调模式禁能时（ADMEN=0），PWM的最小频率f_{pwm}:

$$(f_{pwm})_{min} = f_{PSC_CLK} \ / \ 65536 \quad (23\text{-}2)$$

微调模式使能时（ADMEN=1），

$$(f_{pwm})_{min} = f_{PSC_CLK} \ / \ (65535 + 15 \ / \ 16) \quad (23\text{-}3)$$

当微调模式使能时，CHxVAL[19:0]位域和CARL[19:0]位域的最大值为0xFFFFE（整数部分为0xFFFE，小数部分为0xF）。

在连续16个周期内，占空比和周期的变化情况，具体如*图23-20. PWM微调模式原理*和*表23-3. 边沿对齐模式中CHxVAL和CARL位域的变化*所示。

图 23-20. PWM 微调模式原理

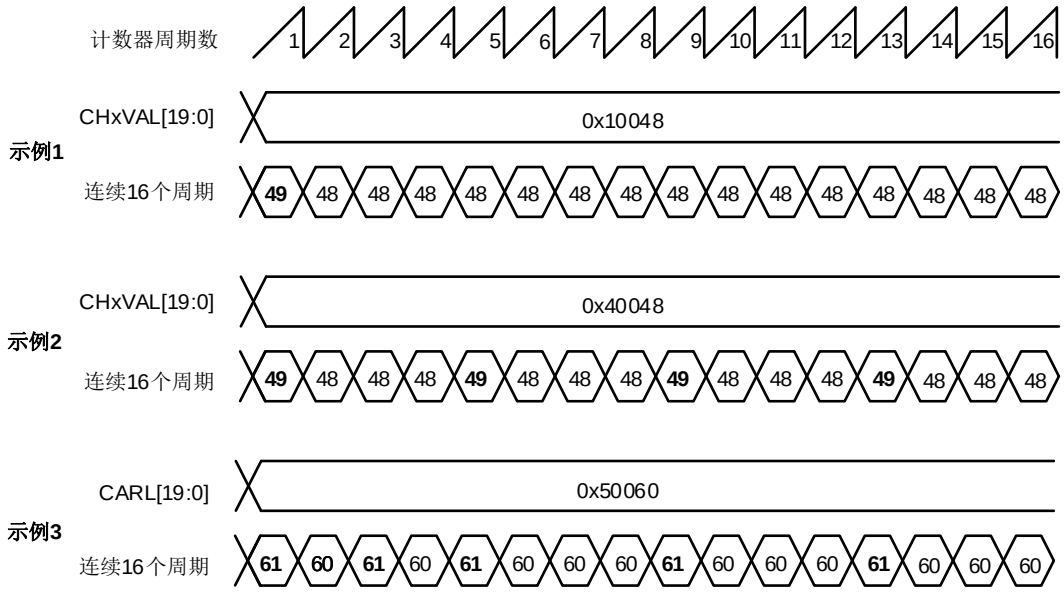


表 23-3. 边沿对齐模式中 CHxVAL 和 CARL 位域的变化

CHxVAL[19:16] / CARL[19:16]	周期															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0010	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-
0011	+1	-	-	-	+1	-	-	-	+1	-	-	-	-	-	-	-
0100	+1	-	-	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0101	+1	-	+1	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0110	+1	-	+1	-	+1	-	-	-	+1	-	+1	-	+1	-	-	-
0111	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	-	-
1000	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1001	+1	+1	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1010	+1	+1	+1	-	+1	-	+1	-	+1	+1	+1	-	+1	-	+1	-
1011	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	-	+1	-
1100	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1101	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1110	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	+1	+1	+1	+1	-
1111	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	-

PWM微调模式也适用于中央对齐模式，具体请见*表23-4. 中央对齐模式中CHxVAL和CARL位域*

的变化，微调模式应用在8个连续的PWM周期。

表 23-4. 中央对齐模式中 CHxVAL 和 CARL 位域的变化

CHxVAL [19:16] / CARL [19:16]	周期															
	1		2		3		4		5		6		7		8	
	向上计数	向下计数	向上计数	向下计数	向上计数	向下计数	向上计数	向下计数	向上计数	向下计数	向上计数	向下计数	向上计数	向下计数	向上计数	向下计数
0000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0010	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-
0011	+1	-	-	-	+1	-	-	-	+1	-	-	-	-	-	-	-
0100	+1	-	-	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0101	+1	-	+1	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0110	+1	-	+1	-	+1	-	-	-	+1	-	+1	-	+1	-	-	-
0111	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	-	-
1000	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1001	+1	+1	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1010	+1	+1	+1	-	+1	-	+1	-	+1	+1	+1	-	+1	-	+1	-
1011	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	-	+1	-
1100	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1101	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1110	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	+1	+1	+1	+1	-
1111	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	-

复合 PWM 模式

在复合 PWM 模式中（CHxCPWMEN = 1’b1，CHxMS[2:0] = 3’b000 和 CHxCOMCTL = 4’b0110、4’b0111），通道 x（x = 0…3）上的 PWM 输出信号由 CHxVAL 和 CHxCOMVAL_ADD 位确定。

如果 CHxCOMCTL = 4’b0110（PWM 模式 0）且 DIR = 1’b0（向上计数模式），或者 CHxCOMCTL = 4’b0111（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 =4’b0111（PWM 模式 1）且 DIR = 1’b0（向上计数模式），或者 CHxCOMCTL =4’b0110（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 23-5 复合 PWM 脉冲宽度*计算。

表 23-5 复合 PWM 脉冲宽度

条件	模式	PWM 脉冲宽度
CHxVAL < CHxCOMVAL_ADD	PWM 模式 0	（CARL + 0x0001） +

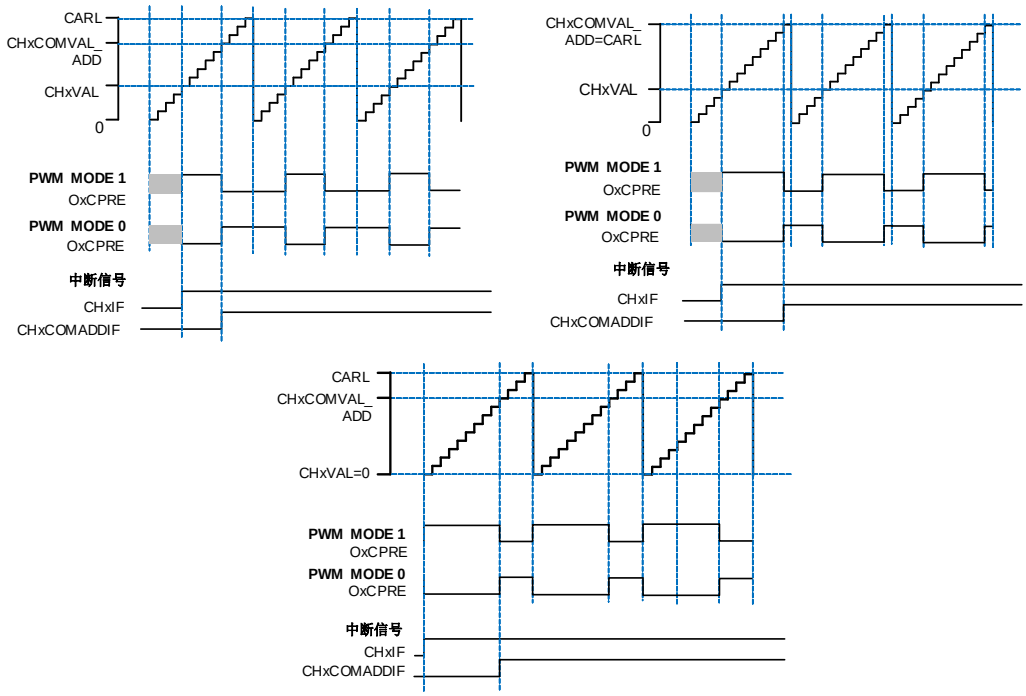
条件	模式	PWM 脉冲宽度
≤ CARL		（CHxVAL – CHxCOMVAL_ADD）
	PWM 模式 1	（CHxCOMVAL_ADD – CHxVAL）
CHxCOMVAL_ADD < CHxVAL ≤ CARL	PWM 模式 0	（CHxVAL - CHxCOMVAL_ADD）
	PWM 模式 1	（CARL + 0x0001） + （CHxCOMVAL_ADD – CHxVAL）
（CHxVAL = CHxCOMVAL_ADD ≤ CARL）或（CHxVAL > CARL > CHxCOMVAL_ADD）	PWM 模式 0（向上计数）或 PWM 模式 1（向下计数）	100%
	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：

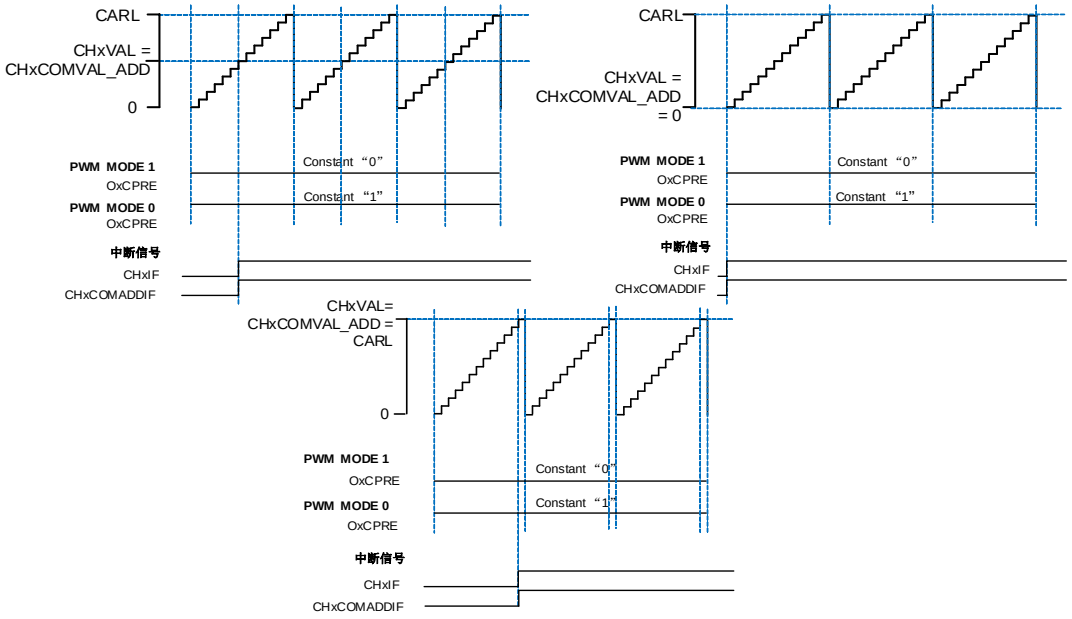
- CHxVAL < CHxCOMVAL_ADD，CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 23-21 通道 x 输出 PWM（CHxVAL < CHxCOMVAL_ADD）



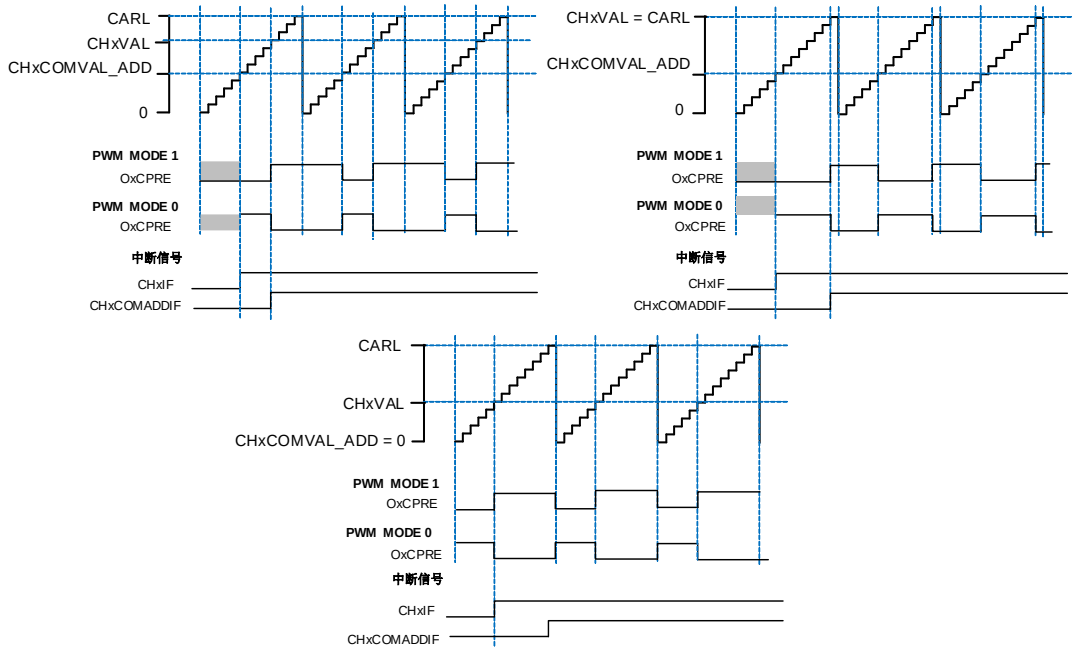
- CHxVAL = CHxCOMVAL_ADD，CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 23-22 通道 x 输出 PWM（CHxVAL = CHxCOMVAL_ADD）



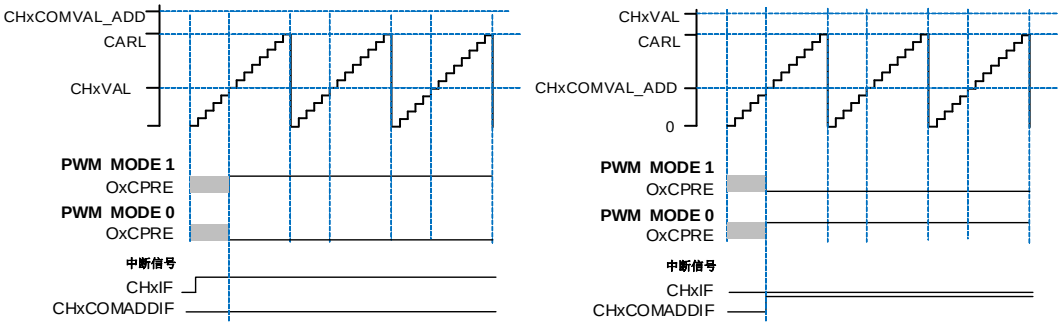
- CHxVAL > CHxCOMVAL_ADD, CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 23-23. 通道 x 输出 PWM（CHxVAL > CHxCOMVAL_ADD）



- CHxVAL或CHxCOMVAL_ADD值大于CARL。

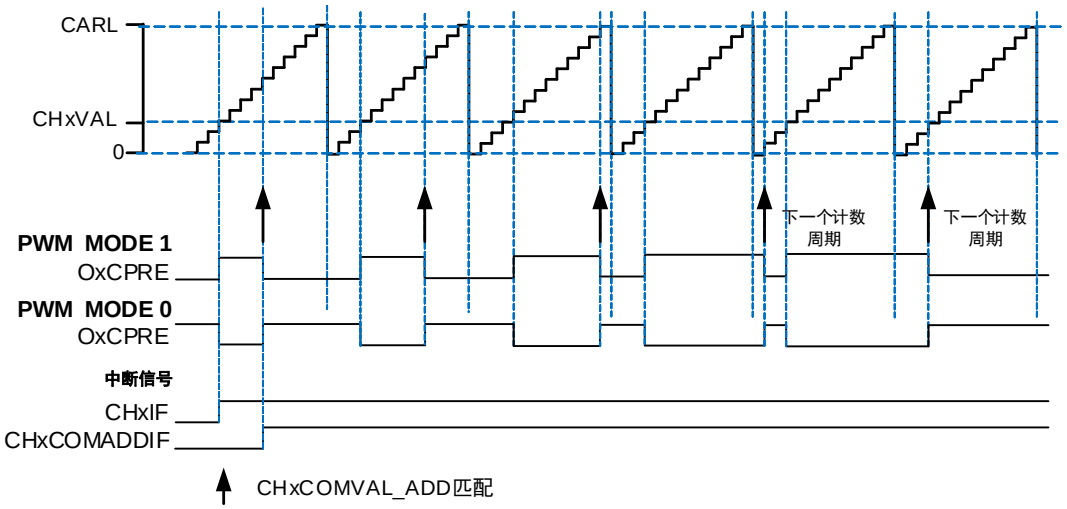
图 23-24. 通道 x 输出 PWM（CHxVAL 或 CHxCOMVAL_ADD > CARL）



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23-25. 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。

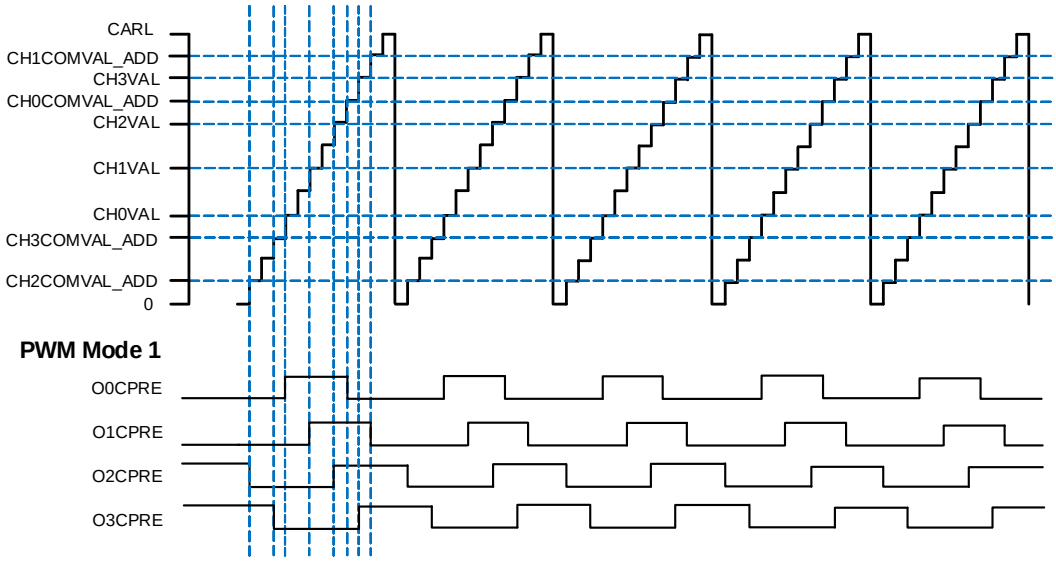
在某些情况下，CHxCOMVAL_ADD的匹配事件可以发生在下一个计数周期（CHxCOMVAL_ADD值在计数器到达CHxVAL值之后被写入，且CHxCOMVAL_ADD值小于或者等于CHxVAL值）。

图 23-25. 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 23-26. 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[3:0]（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[3:0]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 23-27. 边沿对齐模式下 CHx_O 输出脉冲（CHxOMPSEL ≠ 2’b00）

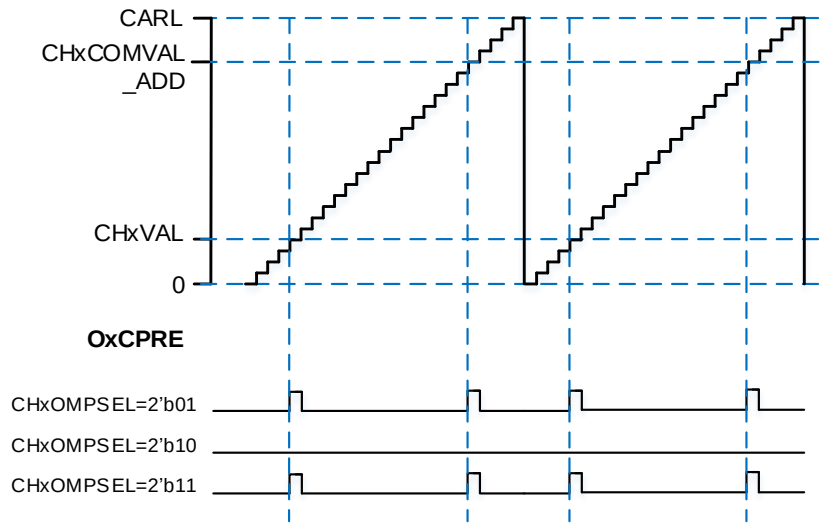
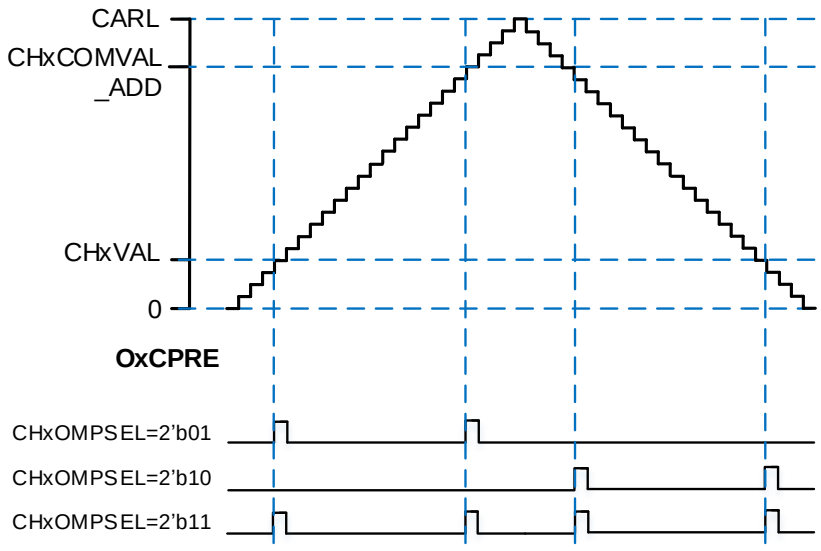


图 23-28. 中央对齐模式下 CHx_O 输出脉冲（CHxOMPSEL ≠ 2’b00）



通道输出准备信号

如*图 23-14. 输出比较逻辑（当 MCHxMSEL = 2’b00 时，x = 0,1,2,3）*和*图 23-15. 输出比较逻辑（当 MCHxMSEL = 2’b11 时，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 或多模式通道 x 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 ETIFP 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。

清除通道输出准备信号

当CHxCOMCEN位或MCHxCOMCEN位（在TIMERx_CHCTLy / TIMERx_MCHCTLy寄存器中）置 1 时，OxCPRE 和 MOxCPRE 信号可以由 OCPRE_CLR_INT 信号清除。该功能用于CHxCOMCTL[3:0]位域或MCHxCOMCTL[3:0]位域（4'b0100和4'b0101除外）中配置的比较输出模式。

可以通过TIMERx_SMCFG寄存器中的OCRC位来选择OCPRE_CLR_INT的信号源。

OCRC 位清 0 时，OCPRE_CLR_INT 连接到 OCPRE_CLR 输入。OxCPRE /MOxCPRE 信号被 OCPRE_CLR_INT 信号的高电平清除，直到下一个更新事件发生时才会恢复输出。在 TIMERx_AFCTL1 寄存器的 OCRINSEL[2:0]位域中选择 OCPRE_CLR 的输入。

OCRC 位置 1 时，OCPRE_CLR_INT 连接到 ETIF。由 TIMERx_SMCFG 寄存器中的 ETP 位配置 OCPRE_CLR_INT 的输入极性。此时，ETPSC[1:0]位域必须设置为 2'b00。

互补输出

CHx_O 和 MCHx_O 的输出具有两种情况：

- MCHxMSEL = 2’b00：MCHx_O输出独立于CHx_O输出。
- MCHxMSEL = 2’b11：MCHx_O 输出与 CHx_O 输出互补，且 MCHx_O 的输出不由 MCHxCOMCTL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_CCHP0 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*独立的死区时间插入和中止功能*）。输出极性由 TIMERx_CHCTL2 寄存器中的 CHxP 和 MCHxP 位来决定。

当 CHx_O 和 MCHx_O 的输出互补时，有三种输出情况：输出使能、输出关闭状态和输出禁能，具体情况可参考*表 23-6. 由参数控制的互补输出表（MCHxMSEL = 2'b11）*。

表 23-6. 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="3">0</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O / MCHx_O = LOW
CHx_O / MCHx_O 输出禁能(1)</td></tr>
<tr><td>1</td><td colspan="2" rowspan="3">CHx_O/MCHx_O输出关闭状态(2)：
通道先输出无效电平：CHx_O = CHxP，MCHx_O = CHxNP）；如果死区产生时钟未失效，在死区时间之后：
CHx_O = ISOx，MCHx_O = ISOxN (3)</td></tr>
<tr><td rowspan="2">1</td><td>0</td></tr>
<tr><td>1</td><td>1</td></tr>
<tr><td>1</td><td colspan="3">CHx_O/MCHx_O输出关闭状态：
通道先输出无效电平：CHx_O = CHxP，MCHx_O = CHxNP）；如果死区产生时钟未失效，在死区时间之后：
CHx_O = ISOx，MCHx_O = ISOxN</td></tr>
<tr><td rowspan="8">1</td><td rowspan="4">0</td><td rowspan="6">0/1</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O/MCHx_O = LOW
CHx_O/MCHx_O输出禁能</td></tr>
<tr><td>1</td><td>CHx_O = LOW
CHx_O输出禁能</td><td>MCHx_O=OxCPRE⊕(2)
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输出禁能</td></tr>
<tr><td>1</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O输出使能</td><td>MCHx_O=(!OxCPRE)(3)⊕
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输出关闭状态</td><td>MCHx_O = MCHxP
MCHx_O输出关闭状态</td></tr>
<tr><td>1</td><td>CHx_O = CHxP
CHx_O输出关闭状态</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输出关闭状态</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 / MCHx_O 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 / MCHx_O 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。

死区时间插入

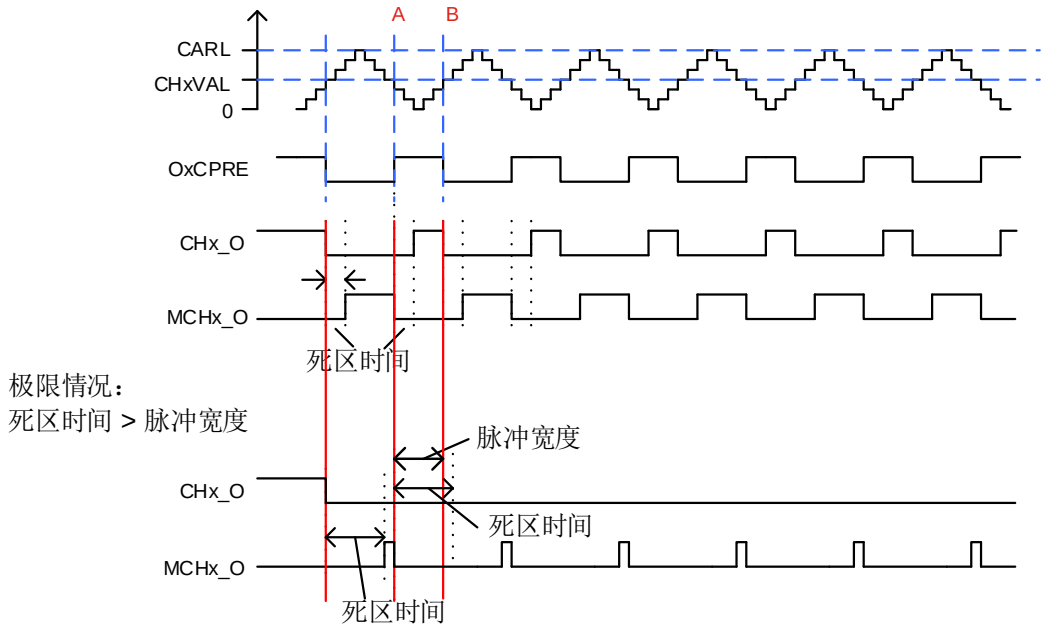
设置 MCHxMSEL = 2'b11，CHxEN 和 MCHxEN 为 1'b1，同时设置 POEN=1，就可以使能死区插入功能。DTCFG 位域定义了死区时间，死区时间对所有通道有效。死区时间设置的细节请参考*互补通道保护寄存器 0（TIMERx_CCHP0）*。

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 23-29. 带死区时间的互补输出*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23-29. 带死区时间的互补输出*所示。

图 23-29. 带死区时间的互补输出



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。

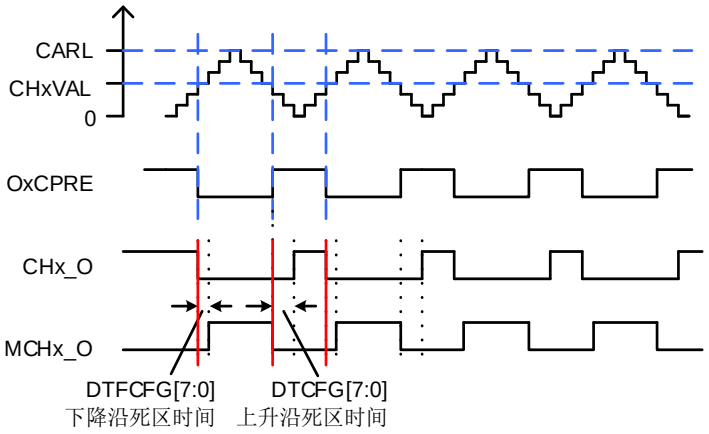
不同的死区时间插入

当DTDIFEN位（在TIMERx_CCHP1寄存器中）设置为1时，CHx_O和MCHx_O信号可以输出不同的死区时间，具体如*图23-30.不同死区时间的互补输出（DTDIFEN=1）*所示。

通道输出准备信号OxCPRE上升沿的死区时间由TIMERx_CCHP0寄存器或TIMERx_FCCHPy寄存器中的DTCFG[7:0]位域配置。OxCPRE信号的下降沿的死区时间由TIMERx_CCHP1寄存器或TIMERx_FCCHPy寄存器中的DTFCFG[7:0]位域配置。

可以在CHx_O和MCHx_O信号输出时修改死区时间。当TIMERx_CCHP1寄存器中的DTMODEN位置1时，可以使能该功能。DTCFG[7:0]位域和DTFCFG[7:0]位域的新值将会在下一次更新事件发生时生效。

图 23-30.不同死区时间的互补输出（DTDIFEN=1）



中止功能

当 MCHxMSEL = 2’b11（MCHx_O 的输出不使用 MCHxCOMCTL 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。

高级定时器有两种中止功能：BREAK0 和 BREAK1。可以通过将 TIMERx_CCHP0 寄存器中的 BRK0EN/ BRK1EN 位置 1 来使能中止功能。中止输入极性由 TIMERx_CCHP0 寄存器中的 BRK0P/BRK1P 位配置，电平有效。

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_CCHP0 寄存器的 POEN、IOS 和 ROS 位，TIMERx_CTL1 寄存器的 ISOx 和 ISOxN 位。

中止事件是所有源逻辑或运算的结果。中止功能可以处理三种类型的事件源：

- 外部信号源：来自BRKINx（x = 0…2）输入；
- 系统源：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、SRAM奇偶校验错误或FLASH ECC错误事件；
- 片上外设源：比较器输出、HPDF的看门狗输出。

中止事件也可以由软件置位TIMERx_SWEVG寄存器中的BRK0G/ BRK1G位产生。

两种中止功能逻辑如*图 23-31. BREAK0 中止功能逻辑图*和*图 23-32. BREAK1 中止功能逻辑图*所示，其中 BRKINx（x = 0…2）可以从 TRIGSEL 模块选择 GPIO 引脚，具体可参考 *TIMER0_BRKIN 触发选择寄存器（TRIGSEL_TIMER0BRKIN）*、*TIMER7_BRKIN 触发选择寄存器（TRIGSEL_TIMER7BRKIN）*和 *TIMER19_BRKIN 触发选择寄存器（TRIGSEL_TIMER19BRKIN）*。

图 23-31. BREAK0 中止功能逻辑图

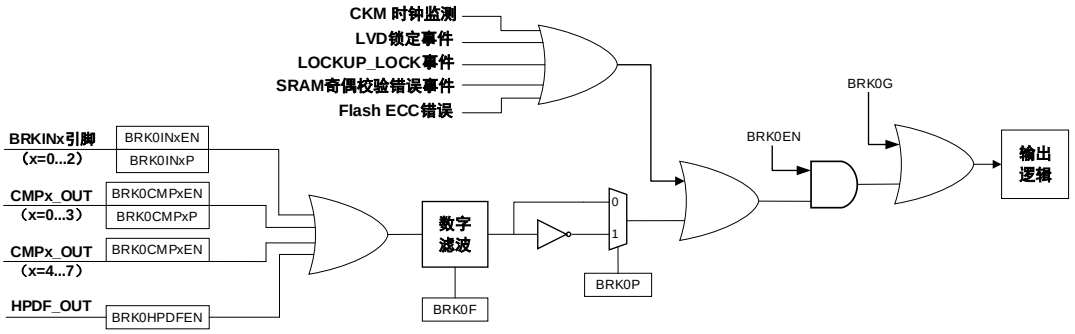
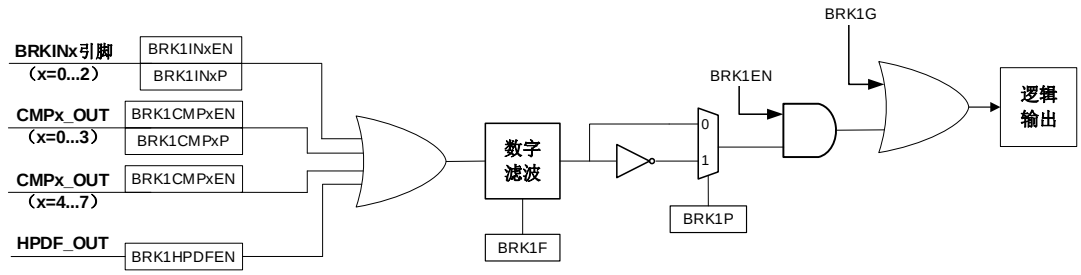


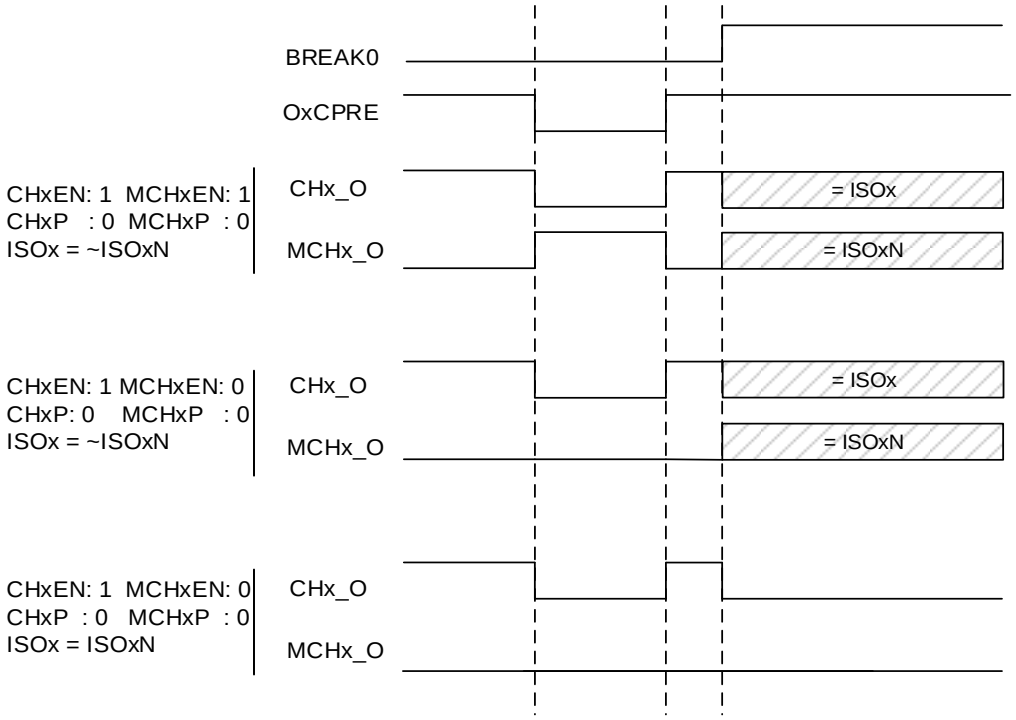
图 23-32. BREAK1 中止功能逻辑图



BREAK0可用于处理系统源、片上外设和外部输入信号源的故障，当发生BREAK0中止事件时，输出强制为无效电平，或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；BREAK1只用于处理片上外设和外部输入信号源的故障，当发生BREAK1中止事件时，输出强制为无效电平。

当 MCHxMSEL = 2’b11 且发生 BREAK0 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。当 IOS=1 时，通道输出情况如*图 23-33. 通道响应 BREAK0 中止输入（高电平有效）时，输出信号的行为（IOS=1）*所示，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。

图 23-33. 通道响应 BREAK0 中止输入（高电平有效）时，输出信号的行为（IOS=1）



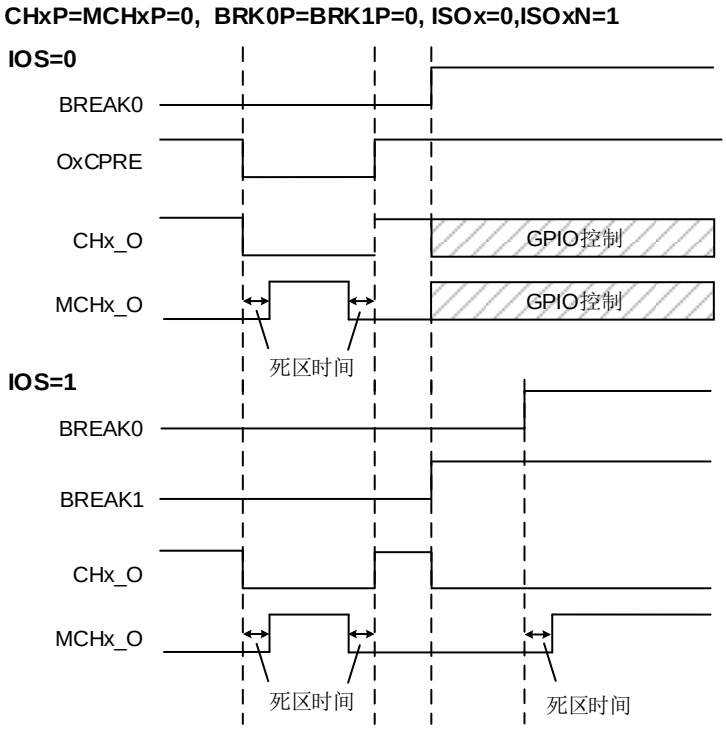
BREAK0的优先级高于BREAK1。只有在IOS=1和ROS=1时，才能使用BREAK1功能。

表 23-7. BREAK0 和 BREAK1 输入信号时，TIMER 互补通道输出情况（break 输入高电平有效）

BREAK0 输入	BREAK1 输入	输出状态	
		CHx_O	MCHx_O
高电平	高电平	IOS=1：CHx_O 输出无效，然后在一个死区时间之后输出空闲电平（由 IOSx 位确定）。 IOS=0：CHx_O 输出禁能（无效）	IOS=1：MCHx_O 输出无效，然后在一个死区时间之后输出空闲电平（由 IOSxN 位确定）。 IOS=0：MCHx_O 输出禁能（无效）
	低电平		
低电平	高电平	CHx_O 输出禁能（无效）	MCHx_O 输出禁能（无效）

发生中止事件时，TIMERx_INTF 寄存器的 BRK0IF/BRK1IF 位被置 1。如果 BRKIE=1，中断产生。

图 23-34. BREAK0 和 BREAK1 中止输入有效时通道输出信号的行为



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 BRKENCHx（x = 0…3）位，可实现对每对通道的中止功能进行独立控制。当 BRKENCHx（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。

锁存中止功能

高级定时器的中止输入引脚BRKINx（x = 0…2）具有锁存中止功能，可通过设置TIMERx_CCHP0寄存器中的BRK0LK/ BRK1LK位为1，将相应的BRKINx（x = 0…2）配置为锁存中止功能。

当使能了锁存中止功能时，需要将 BRKINx（x = 0…2）引脚设置为开漏模式，且低电平有效（BRK0P/BRK1P=0，BRK0INxP/ BRK1INxP =0）。任何中止源请求发生时，都可以将相应的BRKINx（x = 0…2）引脚强制为低电平。若 BRKINx（x = 0…2）引脚设置为高电平有效（BRK0P/BRK1P=1，BRK0INxP/ BRK1INxP =1），则锁存中止功能被禁止。

当中止功能使能（将TIMERx_CCHP0寄存器中的BRK0EN=1或BRK1EN=1）时，通过软件将TIMERx_SWEVG寄存器中的BRK0G/ BRK1G位置1也可以将BRKINx（x = 0…2）引脚强制为低电平。

当中止功能未使能（将TIMERx_CCHP0寄存器中的BRK0EN/ BRK1EN位为0）时，通过软件将BRK0G/ BRK1G位置1，对BRKINx（x = 0…2）引脚无影响。但BRK0F/BRK1F标志位会置位，通道输出为安全状态。

将 TIMERx_CCHP0 寄存器中的 BRK0REL/ BRK1REL 位置 1，可以释放 BRKINx（x = 0…2）引脚，当中止输入源无效时，BRK0REL/ BRK1REL 位由硬件清零，BRKINx（x = 0…2）引脚将恢复锁存中止功能。

在下面两种情况下，不能释放中止输入引脚 BRKINx（x = 0…2）：

- 中止输入源有效：虽然BRK0REL/ BRK1REL位置1，释放了BRKINx（x = 0…2）引脚，但由于中止源仍然存在，故中止事件仍然有效；
- POEN=1：通道输出使能时，即使BRK0REL/ BRK1REL位置1，也不能释放BRKINx（x = 0…2）引脚。

表 23-8. 中止功能锁存/释放条件

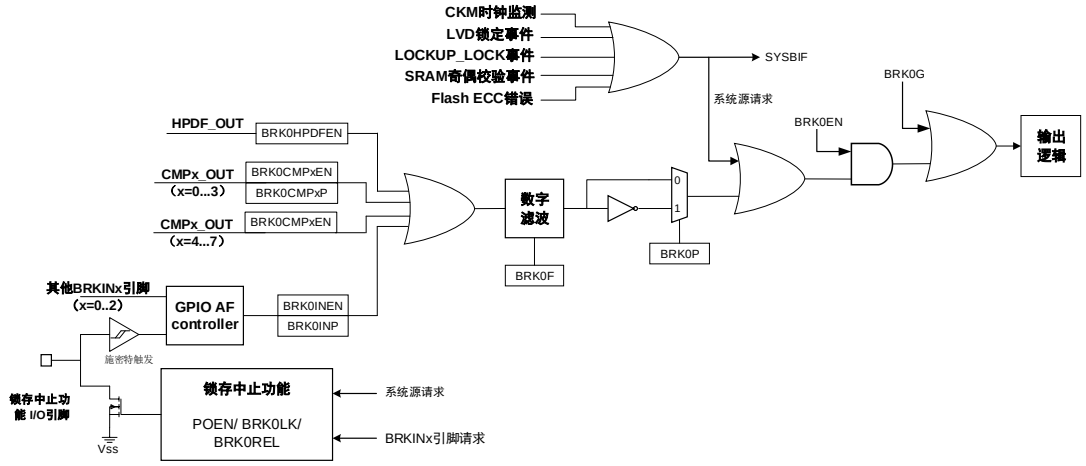
POEN	BRK0LK/ BRK1LK	BRK0REL/ BRK1REL	中止输入引脚状态
0	1	0	锁存
	1	1	释放

BREAK0/BREAK1 输入引脚 BRKINx（x = 0…2）的锁存中止功能默认是使能的（BRK0REL=0 和 BRK1REL=0），当 BREAK0/BREAK1 中止事件发生时，可以通过下面的方法来重新配置锁存中止功能：

- BRK0REL=1或BRK1REL=1，释放BRKINx（x = 0…2）引脚；
- 软件等待系统中止源无效，可通过软件清除SYSBIF标志位；
- 软件轮询BRK0REL和BRK1REL位，直到BRK0REL=0和BRK1REL=0（硬件实现）。

上述过程完成后，BREAK0/BREAK1 锁存中止功能重新使能，此时，可通过软件将 POEN 置 1 来恢复 PWM 输出。

图 23-35. BREAK0 的 BRKINx（x = 0…2）引脚锁存功能逻辑图



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_CCHP0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CI0和CI1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位被硬件自动改变。

输入源可以是只有CI0，可以只有CI1，或者可以同时有CI0和CI1，通过设置TSCFGy[4:0](y = 0, 1, 2, 13, 14) != 5'b00000来选择使用哪种模式。计数器计数方向改变的机制如*表23-9. 不同译码器模式下的计数方向*所示。其中，CI0FE0、CI1FE1是经过滤波和极性选择后的CI0、CI1信号。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 23-9. 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 TSCFG0[4:0]!= 5'b00000	CI1FE1=1	向下	向上	-	-
	CI1FE1=0	向上	向下	-	-
正交译码器模式1 TSCFG1[4:0]!= 5'b00000	CI0FE0=1	-	-	向上	向下
	CI0FE0=0	-	-	向下	向上
正交译码器模式2 TSCFG2[4:0]!= 5'b00000	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
正交译码器模式3 TSCFG13[4:0]!= 5'b00000	CI0FE0=1	-	-	向下	向上
	CI0FE0=0	-	-	-	-
正交译码器模式4 TSCFG14[4:0]!= 5'b00000	CI1FE1=1	向上	向下	-	-
	CI1FE1=0	-	-	-	-

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 23-36. 译码器接口模式下计数器运行例子

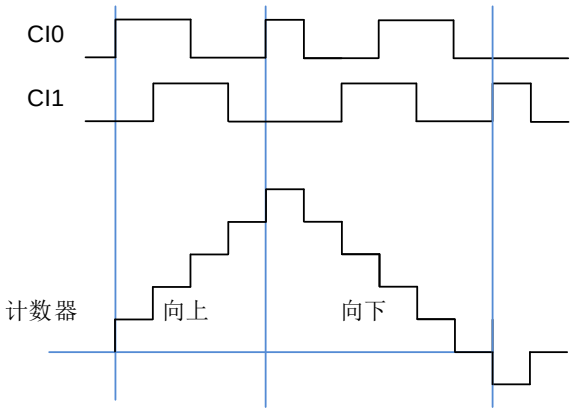
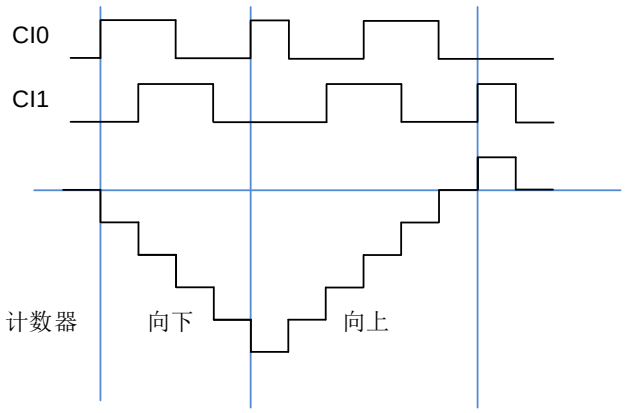


图 23-37. CI0FE0 极性反相的译码器接口模式下的例子



当正交译码器模式下计数器计数方向发生变化时，TIMERx_CTL0寄存器中的DIR位发生改变，同时TIMERx_INTF寄存器中的DIRTRANIF标志位置1。若TIMERx_DMAINTEN寄存器中的DIRTRANIE位置1，则产生相应的中断。

正交译码器信号检测

支持两种正交译码器信号检测：信号跳变检测和断线检测。

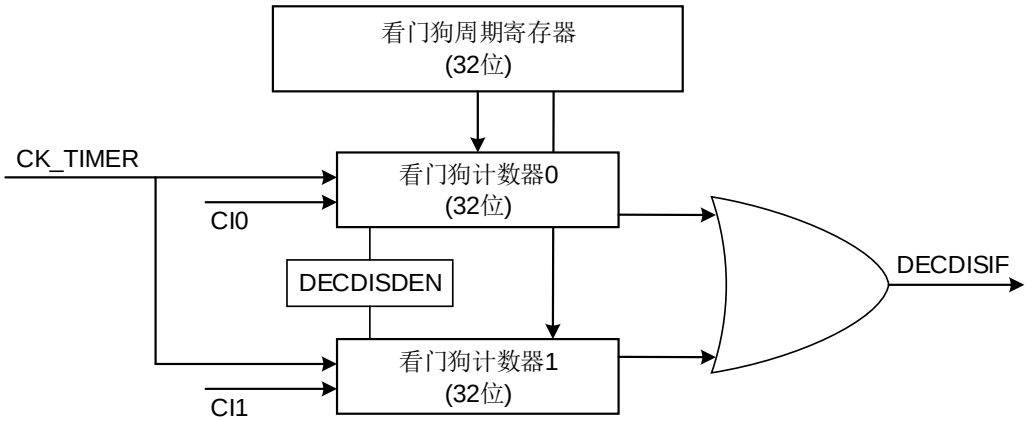
正交译码器信号跳变检测功能可用于检测两个正交译码器输入信号CI0、CI1的电平跳变沿（上升沿或下降沿）是否同时发生，可通过将TIMERx_CTL2寄存器中的DECJDEN位置1来使能。当DECJDEN=1时，若两个正交信号CI0和CI1的电平跳变同时发生，则中断标志位DECJIF置位。若DECJIE=1，则相应的中断产生。

正交译码器信号断线检测功能可用于检测正交译码器输入信号CI0、CI1是否正常，可通过将TIMERx_CTL2寄存器中的DECDISDEN位置1来使能。正交译码器信号检测模块包括2个32位的看

门狗计数器和1个周期寄存器，具体如*图23-38. 正交译码器信号断线检测框图*所示，CI0FE0、CI1FE1信号分别用于复位2个看门狗计数器。

当DECDISDEN=1时，2个看门狗计数器同时开始向上计数，若看门狗计数器计数到看门狗周期值（该值由TIMERx_WDGPER寄存器中的WDGPER[31:0]位域确定），则看门狗计数器计数超时，中断标志位DECDISIF置位。若DECDISIE=1，则相应的中断产生。

图 23-38. 正交译码器信号断线检测框图



译码器

译码器功能有4种模式：译码器模式0~3，通过设置TSCFGy[4:0](y = 9, 10, 11, 12) != 5'b00000来选择。这4种计数模式下的输入源有两个：CI0和CI1，其中，CI0FE0、CI1FE1是经过滤波和极性选择后的CI0、CI1信号。

使用译码器模式0/1时，CI0FE0作为计数方向信号，CI1FE1作为计数脉冲。

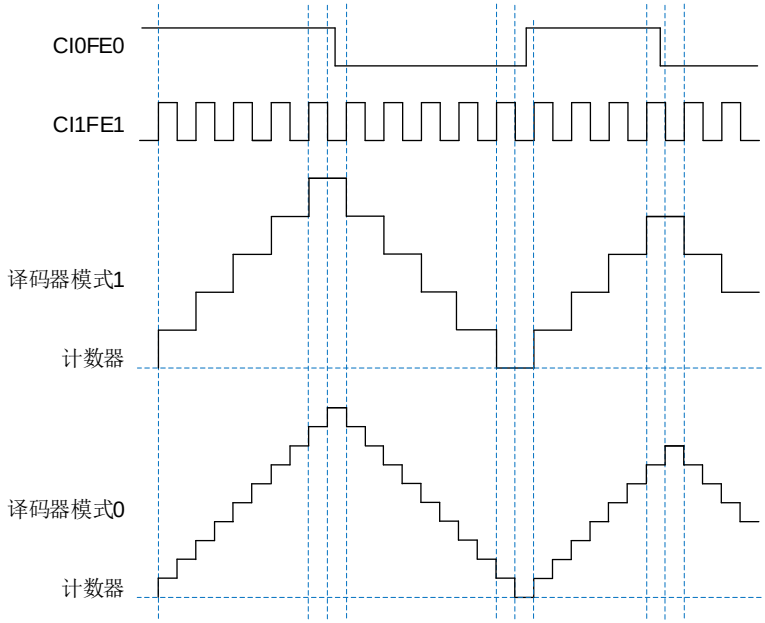
其中，CH0P用于计数方向选择：当CH0P=0时，CI0FE0为高电平时向上计数，CI0FE0为低电平时向下计数；当CH0P=1时，CI0FE0为高电平时向下计数，CI0FE0为低电平时向上计数。

CH1P用于选择CI1FE1信号的计数边沿：译码器模式0时，计数器在CI1FE1信号的上升沿和下降沿进行计数；译码器模式1时，当CH1P=0时，在CI1FE1信号的上升沿计数，当CH1P=1时，在CI1FE1信号的下降沿计数。更多译码器模式1的细节如*表23-10. 译码器模式1的计数情况*和*图23-39. 译码器模式0/1计数器运行实例（CH1P=0）*所示。

表 23-10. 译码器模式 1 的计数情况

CH1P	电平	计数器计数情况
0	CI0FE0为高电平	计数器在CI1FE1信号的上升沿向上计数
	CI0FE0为低电平	计数器在CI1FE1信号的上升沿向下计数
1	CI0FE0为高电平	计数器在CI1FE1信号的下降沿向上计数
	CI0FE0为低电平	计数器在CI1FE1信号的下降沿向下计数

图 23-39. 译码器模式 0/1 计数器运行实例（CH1P=0）



译码器模式2/3由CI0FE0、CI1FE1信号各自相互作用产生计数值，DIR位被硬件自动改变。

译码器模式2时，计数器在CI0FE0、CI1FE1信号的上升沿和下降沿进行计数，计数方向由CH0P和CH1P确定；译码器模式3，计数器在CI0FE0、CI1FE1信号的上升沿或下降沿进行计数，当CHxP=0时，信号为高电平时计数，或在信号的下降沿计数；当CHxP=1时，信号为低电平时计数，或在信号上升沿计数。具体情况请见*图23-40. 译码器模式2 / 3计数器运行实例（CH0P / CH1P=0）*和*表23-11. 译码器模式2 / 3下的计数方向*。

图 23-40. 译码器模式 2 / 3 计数器运行实例（CH0P / CH1P=0）

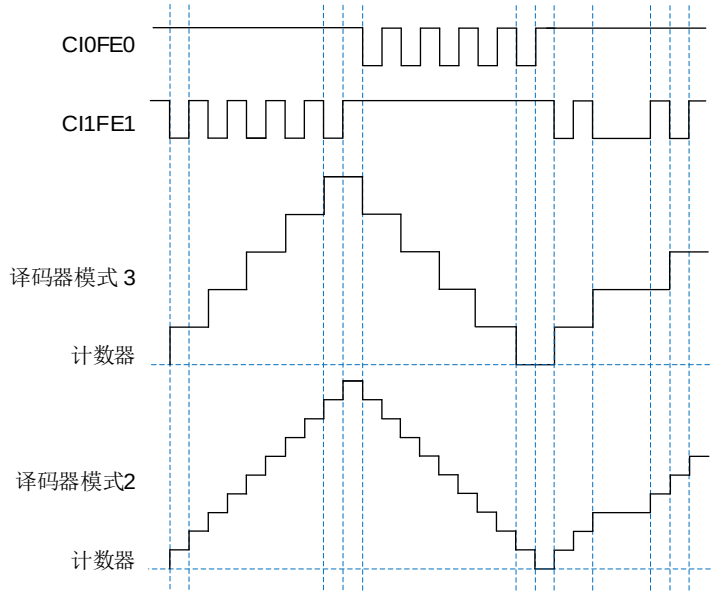


表 23-11. 译码器模式 2 / 3 下的计数方向

计数模式	极性	电平	CI0FE0		CI1FE1	
			上升	下降	上升	下降
译码器模式2 TSCFG11[4:0]!= 5'b00000	CHxP=0 (x = 0, 1)	CI1FE1=1	向下	向下	x	x
		CI1FE1=0	-	-	x	x
		CI0FE0=1	x	x	向上	向上
		CI0FE0=0	x	x	-	-
	CHxP=1 (x = 0, 1)	CI1FE1=1	-	-	x	x
		CI1FE1=0	向下	向下	x	x
		CI0FE0=1	x	x	-	-
		CI0FE0=0	x	x	向上	向上
译码器模式3 TSCFG12[4:0]!= 5'b00000	CHxP=0 (x = 0, 1)	CI1FE1=1	-	向下	x	x
		CI1FE1=0	-	-	x	x
		CI0FE0=1	x	x	-	向上
		CI0FE0=0	x	x	-	-
	CHxP=1 (x = 0, 1)	CI1FE1=1	-	-	x	x
		CI1FE1=0	向下	-	x	x
		CI0FE0=1	x	x	-	-
		CI0FE0=0	x	x	向上	-

当译码器模式下计数器计数方向发生变化时，TIMERx_CTL0寄存器中的DIR位改变，同时TIMERx_INTF寄存器中的DIRTRANIF标志位置1。若TIMERx_DMAINTEN寄存器中的DIRTRANIE位置1，则产生相应的中断。

正交译码器和译码器的时钟输出

定时器可以通过TRGO0输出译码器时钟输出信号。该功能仅用于正交译码器模式0~4和译码器模式0~3，通过将TIMERx_CTL1寄存器中MMC0[2:0]位域设置为4'b1000来使能。

索引输入功能

正交译码模式的索引信号

译码器常用的输出信号由三种：A相脉冲、B相脉冲和1个表示参考位置的索引脉冲信号。其中，索引脉冲信号可以将TIMER的计数器复位。当使用该功能时，索引脉冲信号必须连接到TIMER的ETI引脚，并可以进行滤波处理。

该功能仅用于正交译码器模式0~4和译码器模式0~3，通过配置TIMERx_DECCTL寄存器中的INDRSTEN位为1来使能。

TIMER的索引输入功能支持以下3种正交译码器的索引脉冲信号（无需进行额外配置）：

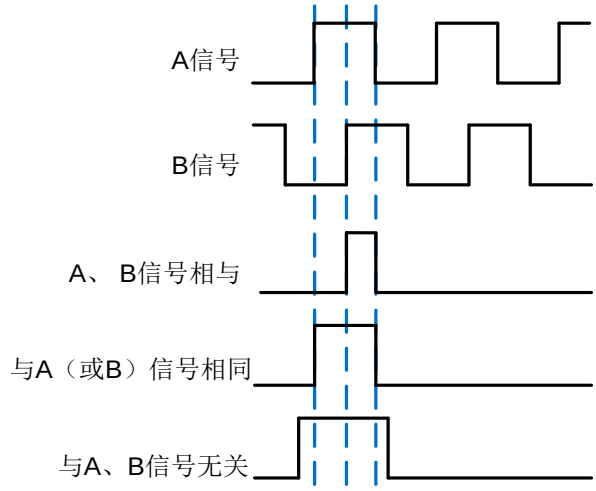
1) A、B信号相与：索引信号为A/B信号脉冲周期的四分之一宽度，与A和B信号的边沿对齐;

2) 与A（或B）信号相同：索引信号为A/B信号脉冲周期的二分之一宽度，与A（或B）信号的两个边沿对齐

3) 与A、B信号无关：索引信号超过一个脉冲周期，不与A、B信号的任何边沿对齐。

根据索引信号的不同，电路对索引信号抖动的容忍度也是不一样的，在第3)种类型时，索引信号需要小于2个脉冲周期，否则，计数器将被复位多次。

图 23-41. 三种类型的索引信号



配置TIMERx_DECCTL寄存器中的INDP[1:0]位域，来选择A、B相信号与索引信号的关系。

根据计数器计数模式的不同，索引输入检测事件发生时的现象也不同：

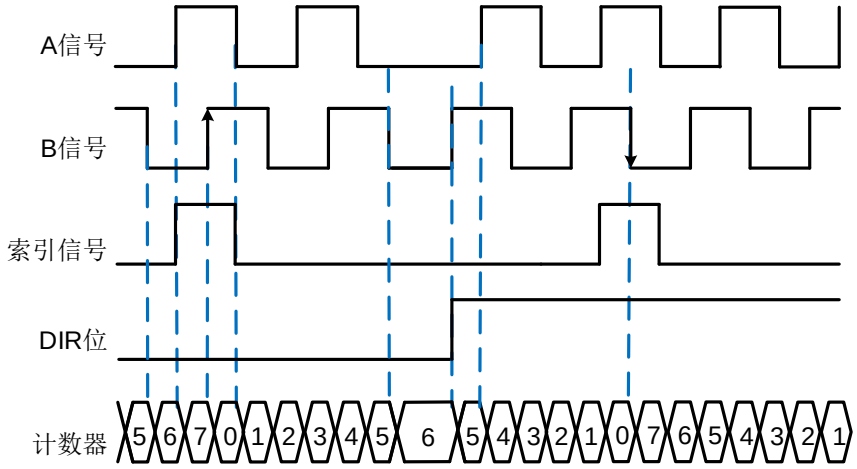
- 向上计数时，计数器复位（DIR位为0）；
- 向下计数时，计数器设置为TIMERx_CAR寄存器的值（DIR位为1）。

这样可以保证无论是向上计数模式还是向下计数模式，索引脉冲信号总是在同一机械角度位置产生。

INDP[1:0]=2’b11时，*图23-42. 索引信号和A相同时的计数器情况（INDP[1:0]=2'b11）*给出了计数器的计数情况。根据计数器计数方向的不同，发生索引输入复位事件时，计数器复位情况不同。图中箭头指向表明索引事件产生的变化。

- 向上计数：当编码器AB信号状态变为“11”时，计数器值清零复位；
- 向下计数：当编码器AB信号状态离开“11”时，计数器值设置为TIMERx_CAR寄存器值。

图 23-42. 索引信号和 A 相同时的计数器情况（INDP[1:0]=2'b11）



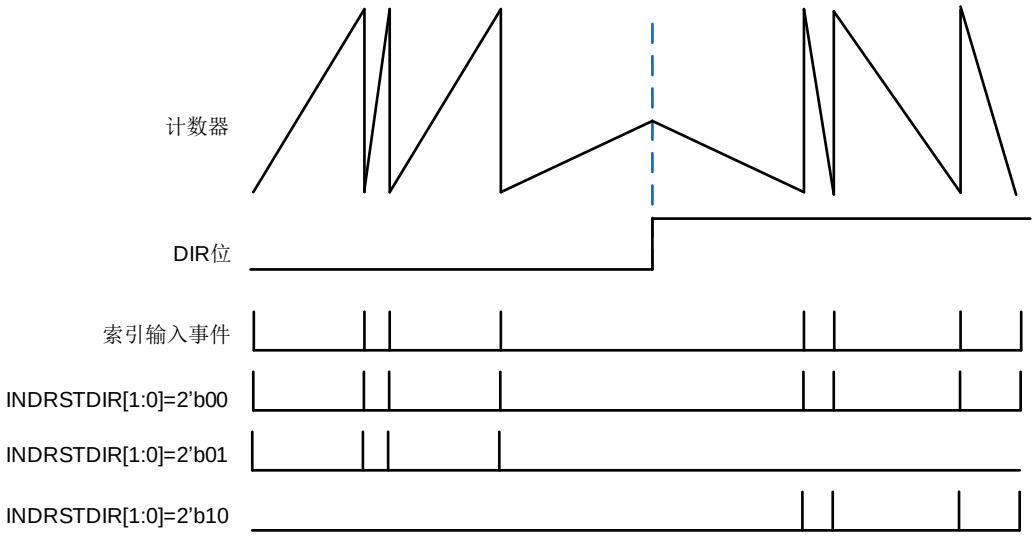
当索引信号事件产生时，TIMERx_INTF寄存器中的INDIF位置1，若TIMERx_DMAINTEN寄存器中的INDIE位置1，则产生相应的中断。

索引信号方向选择

通过配置TIMERx_DECCTL寄存器中的INDRSTDIR[1:0]位域，可以选择使索引复位事件有效的计数方向。*图23-43. 索引信号和计数器复位事件的关系*给出了INDRSTDIR[1:0]位域为不同值时，计数器的复位情况。

注意：当使用译码器模式0/1时，INDRSTDIR[1:0]位域必须为2’b00。

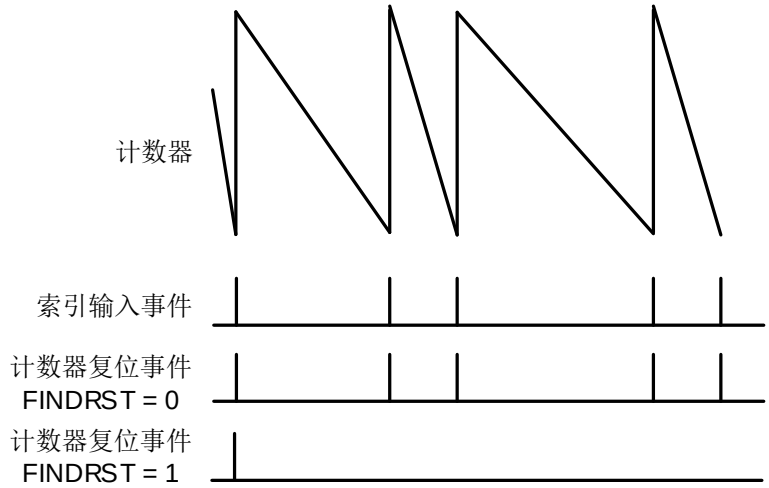
图 23-43. 索引信号和计数器复位事件的关系



第一个索引信号复位计数器

将TIMERx_DECCTL寄存器中的FINDRST位置1，可以使能第一个索引信号复位计数器的功能，具体如*图23-44. 计数器复位事件和FINDRST位*所示。当FINDRST=1时，仅第一个索引脉冲信号复位计数器，后续的脉冲信号都无效。

图 23-44. 计数器复位事件和 FINDRST 位



译码器模式的索引输入功能

在译码器模式0~3时，通过配置TIMERx_DECCTL寄存器中的INDP[0]位，可以确定索引信号复位事件的生效时间：

- INDP[0]=0，在计数方向选择信号为低电平时，索引信号复位计数器功能有效；
- INDP[0]=1，在计数方向选择信号为高电平时，索引信号复位计数器功能有效。

注意：在译码器模式0~3时，INDP[1]位不使用。

当索引信号事件产生时，TIMERx_INTF寄存器中的INDIF标志位置1。若TIMERx_DMAINTEN寄存器中的INDIE位置1，则产生相应的中断。

索引错误检测

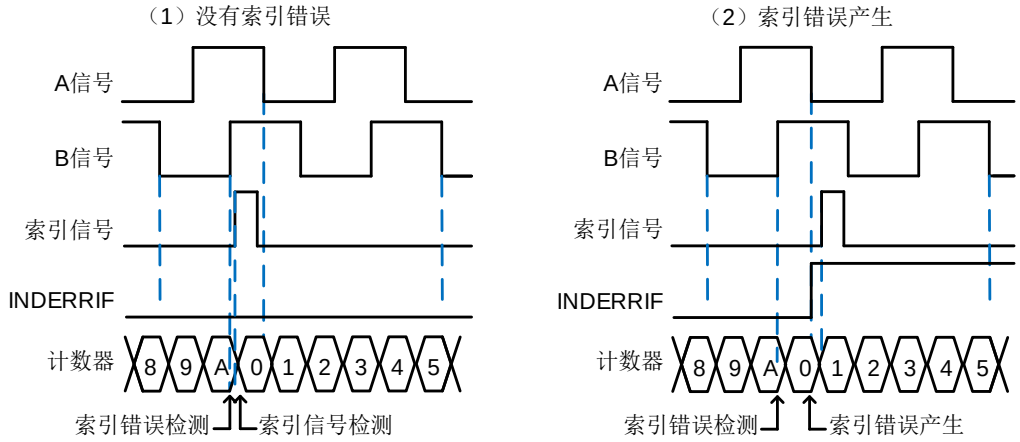
当计数器从TIMERx_CAR寄存器值增计数到0（向上计数时）或者从0减计数到TIMERx_CAR寄存器值（向下计数时）的期间，没有检测到任何索引脉冲信号，则会产生索引错误。

在向上计数时，索引错误会延迟到计数值从0变为1的时候产生，具体如*图23-45. 向上计数模式时的索引错误检测*所示，在左图的波形中，计数值从0变为1的过程中，没有索引信号，故索引错误产生；在右图的波形中，计数值从0变为1的过程中，检测到索引信号，故无索引错误产生。

在向下计数时，索引错误会提前到计数值从1变为0的时候产生。

当索引信号错误产生时，TIMERx_INTF寄存器中的INDERRIF标志位置1。若TIMERx_DMAINTEN寄存器中的INDERRIE位置1，则产生相应的中断。

图 23-45. 向上计数模式时的索引错误检测



正交译码器模式和译码器模式在线修改

正交译码模式之间和译码模式之间可以进行模式的在线修改，通过配置TIMERx_SMCFG寄存器中的DECMODEN位为1可以使能该功能。

在正交译码模式中，只能从正交译码器模式2切换到模式0/1，或从正交译码器模式0/1切换到模式3/4。在译码器中，只能从译码器模式0切换到1，或从译码器模式2切换到3。

配置TIMERx_SMCFG寄存器中的DECMODS位可以选择模式修改的更新源：DECMODS=0，译码器模式在TIMER的更新事件发生时进行更新；DECMODS=1，译码器模式在TIMER的索引事件发生时进行更新。

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 23-46. 霍尔传感器用在 BLDC 电机控制中*是定时器和电机的连接示意图。众所周知，我们要两个定时器。TIMER_in 定时器（可以是高级定时器或者通用 L0 定时器）接收来自电机霍尔传感器的三路信号，这三路信号是电机转子的位置信号。

三个霍尔传感器与 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 23-46. 霍尔传感器用在 BLDC 电机控制中

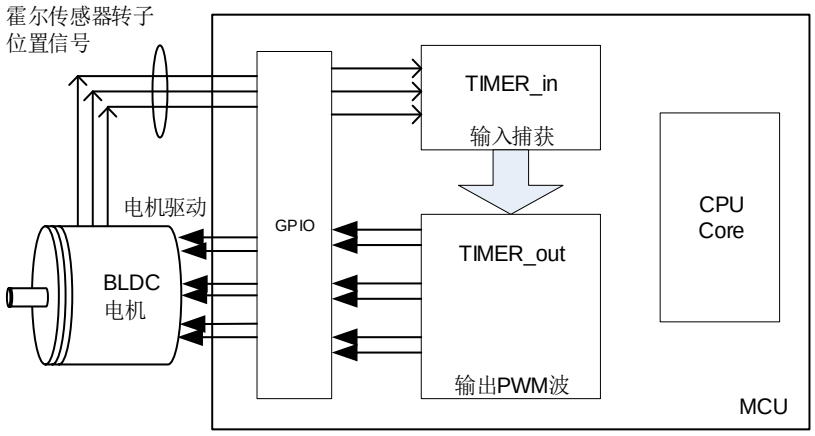
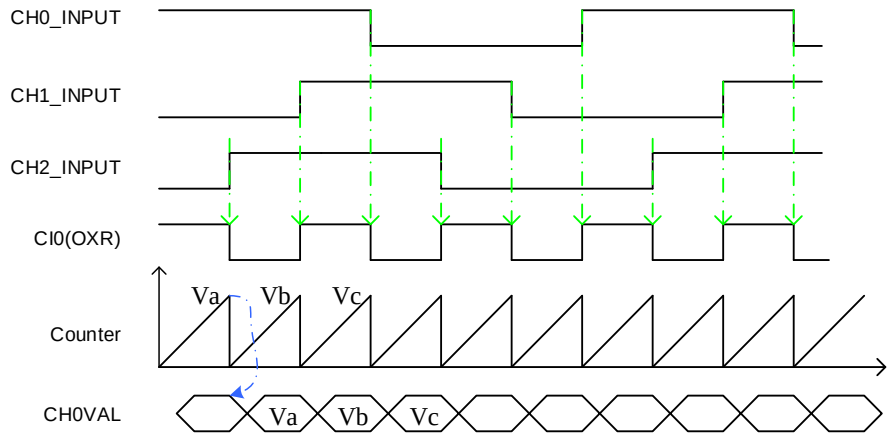
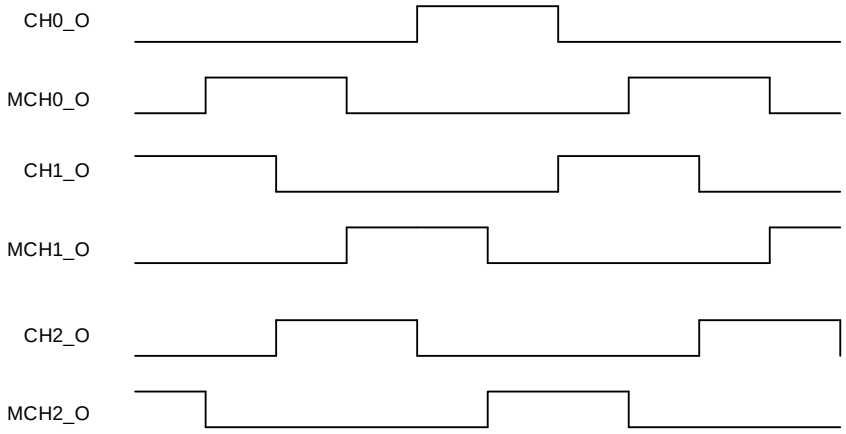


图 23-47. 两个定时器之间的霍尔传感器时序图

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)

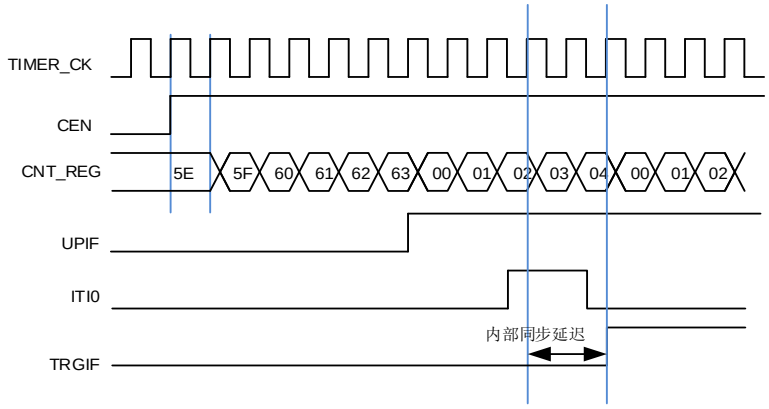


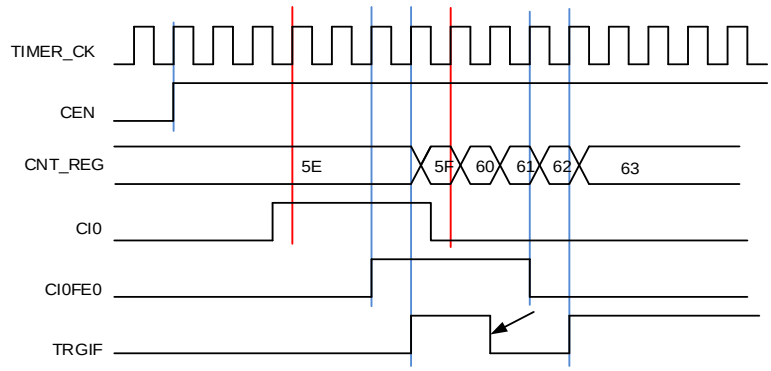
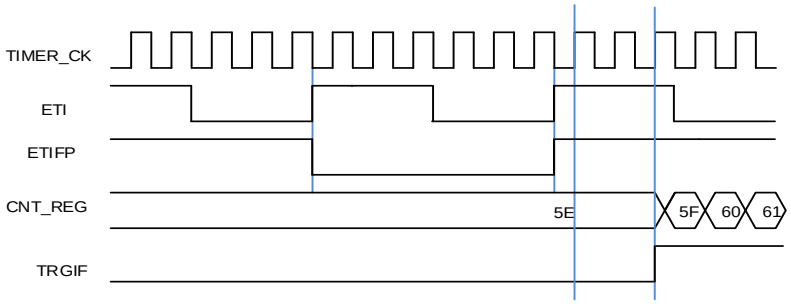
主-从管理

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等，可以通过设置 SYSCFG_TIMERxCFG(x = 0, 7, 19)寄存器中的 TSCFGy[4:0] (y = 3..8)位域来确定，具体的输入触发源可以通过 TSCFGy[4:0] (y = 3..8)位域值来选择。

表 23-12. 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	TSCFGy[4:0] y = 3：复位模式 y = 4：暂停模式 y = 5：事件模式 y = 6：外部时钟模式	TSCFGy[4:0] 00000：模式禁能 00001：ITI0 00010：ITI1 00011：ITI2	如果触发源是CIxFEx（x = 0…3）或者MCIxFEMx（x = 0…3），配置CHxP、MCHxP和MCHxFP来选择极性和	触发源ITIx，滤波和预分频不可用。 触发源CIx/MCIx，配置CHxCAPFLT/MCHxCAPFLT设置滤

<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>0
y = 7：复位 + 事件模式
y = 8：暂停 + 复位模式</td><td>00100：ITI3
00101：CI0F_ED
00110：CI0FE0
00111：CI1FE1
01000：ETIFP(1)
01001：ITI4
01010：ITI5
01011：ITI6
01100：ITI7
01101：ITI8
01110：ITI9
01111：ITI10
10000：保留
10001：保留
10010：保留
10011：ITI14</td><td>反相。
如果触发源是ETIFP（滤波后的ETI外部触发输入），配置ETP选择极性和反相。</td><td>波，分频不可用。
触发源是ETIFP，滤波和预分频不可用。</td></tr>
<tr><td rowspan="2">例1</td><td>复位模式
当触发输入上升沿，计数器清零重启</td><td>TSCFG3[4:0] = 5’b00001，选择ITI0为触发源</td><td>触发源是ITI0，极性选择不可用</td><td>触发源是ITI0，滤波和预分频不可用</td></tr>
<tr><td colspan="4">图 23-48. 复位模式
 
</td></tr>
<tr><td>例2</td><td>暂停模式
当触发输入为低的时候，计数器暂停计数</td><td>TSCFG4[4:0] = 5’b00110，选择CI0FE0为触发源</td><td>TI0S=0 （非异或）[MCH0P=0， CH0P=0] CI0FE0不反相，在上升沿捕获</td><td>在这个例子中滤波被旁路</td></tr>
</table>

<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 23-49. 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>TSCFG5[4:0]
= 5’b01000，
选择ETIFP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 23-50. 事件模式
 
</td></tr>
<tr><td>例 4</td><td colspan="4">复位 + 事件模式
当触发输入的上升沿到来时，计数器被重新初始化并开始计数。该模式仅用于可延时的单脉冲模式。</td></tr>
<tr><td>例 5</td><td colspan="4">暂停 + 复位模式
当触发输入的上升沿或下降沿（由 TIMERx_SMCFG 寄存器中的 PRMRPSEL 位配置）到来时，计数器将复位。当触发输入高时计数器计数，当触发输入低时计数器停止。在这种模式下，计数器的开始和停止可以被控制。</td></tr>
</table>

(1) ETI 信号可以从外部 ETI 引脚输入，也可由片上外设提供，具体情况可以参考 ***TIMER0 ETI 触发选择寄存器（TRIGSEL_TIMER0ETI）***、***TIMER7 ETI 触发选择寄存器（TRIGSEL_TIMER7ETI）***和***TIMER19 ETI 触发选择寄存器（TRIGSEL_TIMER19ETI）***模块。

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。

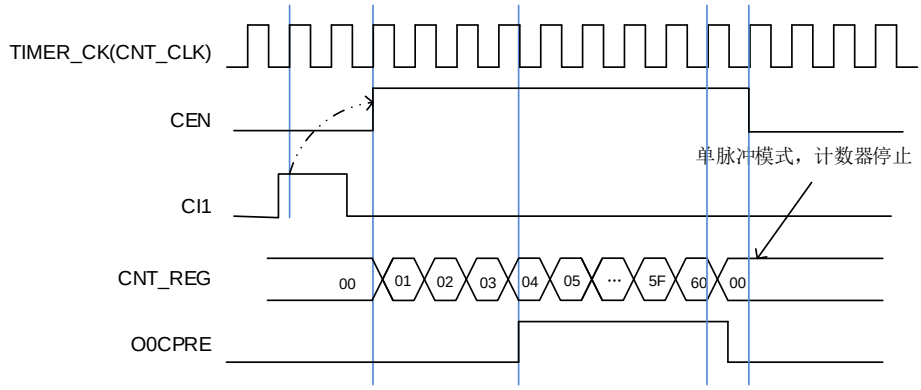
一旦设置定时器运行在单脉冲模式下，没有必要设置 TIMERx_CTL0 寄存器的定时器使能位

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。如果 CEN 值被硬件更新事件自动清 0，计数器将被再次初始化。

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 23-51. 单脉冲模式，TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60



可延时的单脉冲模式

可以通过将 TIMERx_CHCTLx / TIMERx_MCHCTLx 寄存器中的 CHxCOMCTL[3:0] / MCHxCOMCTL[3:0]位置1来使能可延时的单脉冲模式。在这个模式下，通道输出准备信号 OxCPRE / MOxCPRE的脉冲宽度由TIMERx_CAR寄存器值确定。

一旦设置定时器运行在可延时的单脉冲模式下，需进行以下配置：

- 定时器必须工作在从模式下，SYSCFG_TIMERxCFG(x = 0, 7, 19) 寄存器中的 TSCFG7[4:0] != 5'b00000，从模式选择复位 + 事件模式；
- CHxCOMCTL[3:0] / MCHxCOMCTL[3:0]位设置为 4'b1000（可延时单脉冲模式 0）或 4'b1001（可延时单脉冲模式 1）。

在可延时单脉冲模式0下，OxCPRE / MOxCPRE的输出情况类似与PWM模式0。在向上计数模式时，OxCPRE / MOxCPRE先输出有效电平，当外部触发事件发生时，立即输出无效电平，当下一次更新事件发生时，再变成有效电平；在向下计数模式时，OxCPRE / MOxCPRE先输出无效电平，当外部触发事件发生时，立即输出有效电平，当下一次更新事件发生时，再变成无效电平。

在可延时单脉冲模式1下，OxCPRE / MOxCPRE的输出情况类似与PWM模式1。在向上计数模式时，OxCPRE / MOxCPRE先输出无效电平，当外部触发事件发生时，立即输出有效电平，当下一

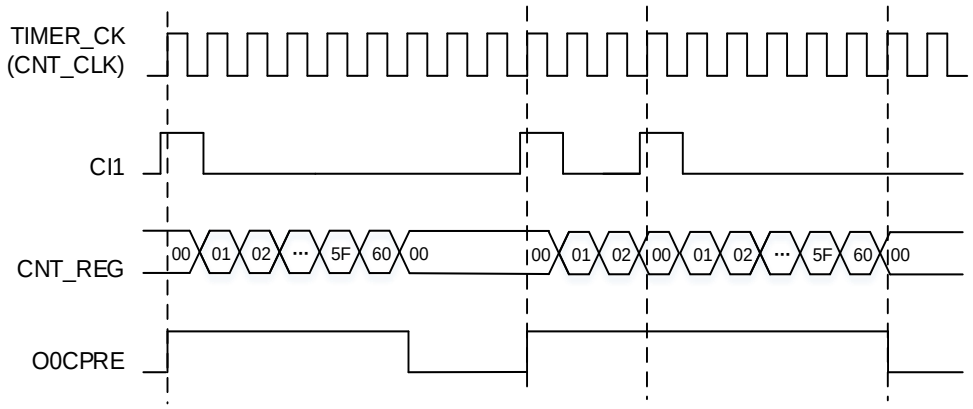
次更新事件发生时，再变成无效电平；在向下计数模式时，OxCPRE / MOxCPRE先输出有效电平，当外部触发事件发生时，立即输出无效电平，当下一次更新事件发生时，再变成有效电平。

PWM微调模式也可用于可延迟的单脉冲模式。

注意：

- 不能使用中央对齐模式，TIMERx_CTL0 寄存器中的 CAM[1:0] = 2'b00；
- 在向上计数时（TIMERx_CTL0 寄存器中的 DIR = 0），TIMERx_CHxCV / TIMERx_MCHxCV 的值设置为 0；在向下计数时，TIMERx_CHxCV / TIMERx_MCHxCV 的值应大于或等于 TIMERx_CAR 的值。

图 23-52. 可延时单脉冲模式（TIMERx_CHxCV = 0x00，TIMERx_CAR = 0x60）

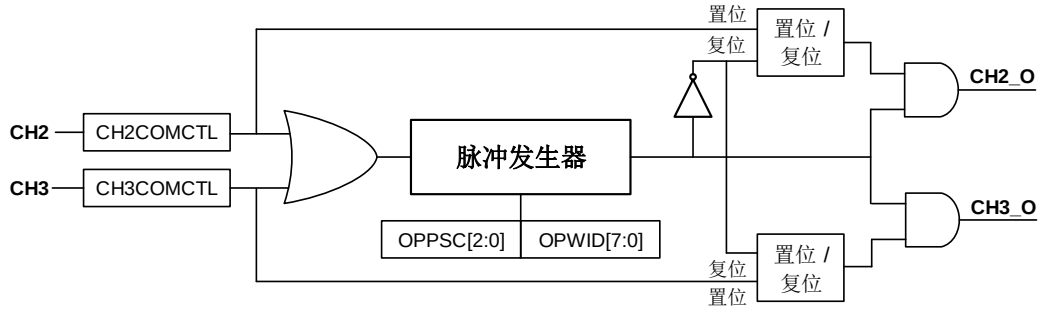


可编程的脉冲输出

配置TIMERx_CHCTL1寄存器中的CH2COMCTL / CH3COMCTL位域为4'b1010，使能可编程的脉冲输出功能。当计数器的计数值与TIMERx_CH2CV / TIMERx_CH3CV寄存器值匹配时，CH2_O / CH3_O上输出一个宽度可配置的脉冲。

输出脉冲的宽度由TIMERx_DECCTL寄存器中的OPPSC[2:0]位域和OPWID[7:0]位域确定，其中OPPSC[2:0]位域确定脉冲的时钟分频系数，OPWID[7:0]位域确定脉冲的宽度。

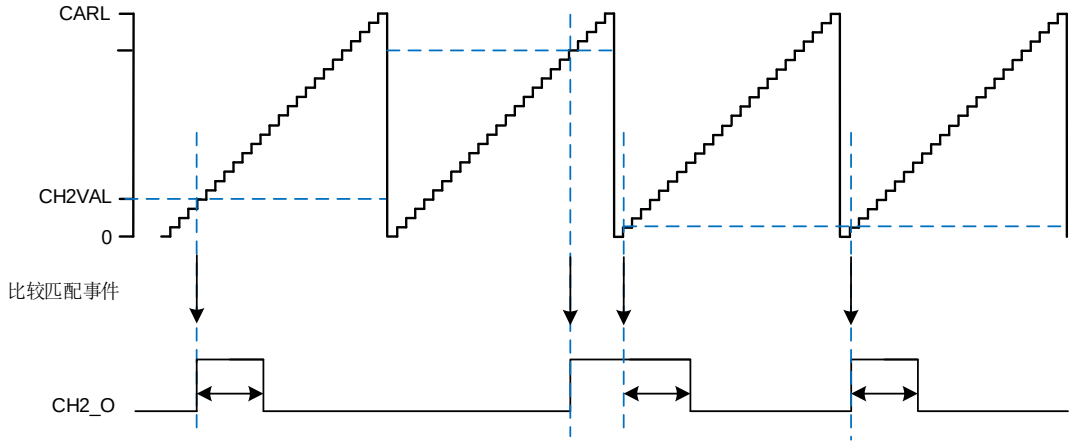
图23-53. 可编程脉冲输出电路



该模式可用于三种计数器计数方式（向上计数、向下计数和中央对齐计数）和所有的从模式。

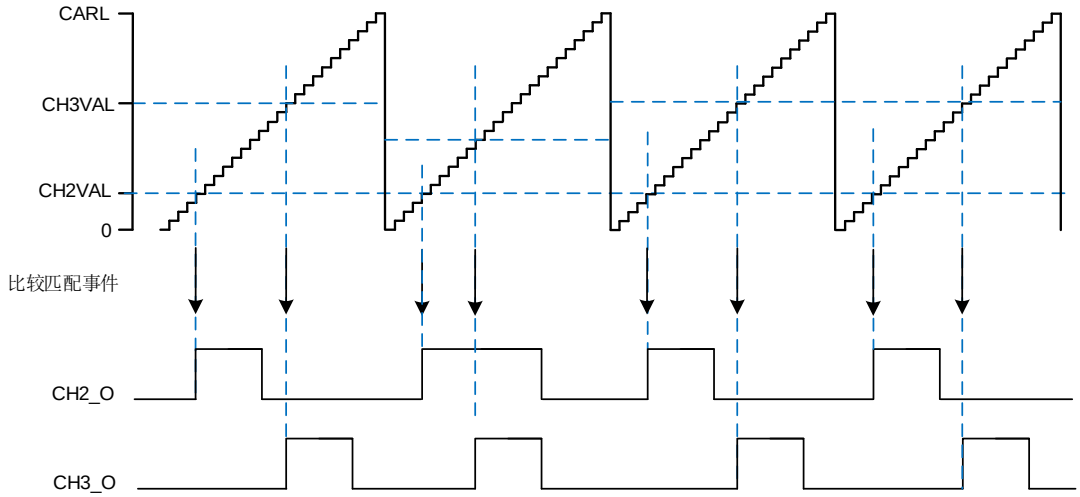
通道输出的脉冲波形是可重复触发的，当前输出脉冲还没结束时，又发生了新的通道比较匹配事件，此时输出脉冲波形将会延长。实际输出脉冲为最后一次匹配事件之前的高电平时间和1个输出脉冲的宽度。

图23-54. 可编程脉冲输出电路波形



当CH2_O和CH3_O同时使能脉冲输出功能时，若一个通道上的比较匹配事件不与另一个通道上的脉冲输出波形重叠，则两个通道各自独立输出脉冲波形。反之，第一个发生比较匹配事件的通道输出的脉冲宽度将会被延长，第二个发生比较匹配事件的通道输出脉冲宽度为配置值。具体如*图23-55. CH2_O和CH3_O同时输出脉冲*所示。

图23-55. CH2_O和CH3_O同时输出脉冲



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。互连的例子如下：

- 定时器2作为定时器0的预分频器

配置定时器 2 为定时器 0 的预分频器，步骤如下：

1. 配置定时器2为主模式，选择其更新事件（UPE）为触发输出（配置TIMER2_CTL1寄存器的MMC0 = 4'b0010）。定时器2在每次计数器溢出产生更新事件时，输出一个周期信号；
2. 配置定时器2周期（TIMER2_CAR寄存器）；
3. 配置定时器0工作在外部时钟模式0，定时器0输入触发源为定时器2（SYSCFG_TIMER0CFG1寄存器中的TRCFG6[4:0] = 5'b00011）；
4. 写1到CEN位启动定时器0（TIMER0_CTL0寄存器）；
5. 写1到CEN位启动定时器2（TIMER2_CTL0寄存器）。

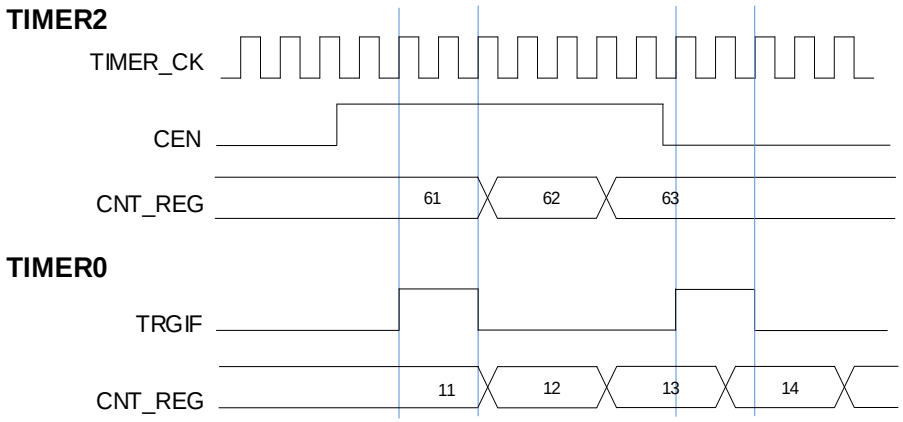
■ 用定时器2的使能/更新信号来启动定时器0

用定时器 2 的使能信号来启动定时器 0，见*图 23-56. 用定时器 2 的使能信号触发定时器 0*。在定时器 2 使能信号输出后，定时器 0 按照分频后的内部时钟从当前值开始计数。

当定时器 0 接收到触发信号，它的 CEN 位被自动置 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2为主模式，发送它的使能信号作为触发输出（配置TIMER2_CTL1寄存器的MMC0 = 4'b0001）；
2. 配置定时器0工作在事件模式，定时器0输入触发源为定时器2（SYSCFG_TIMER0CFG0寄存器中的TRCFG5[4:0] = 5'b00011）；
3. 写1到CEN来开启定时器2（TIMER2_CTL0寄存器）。

图 23-56. 用定时器 2 的使能信号触发定时器 0

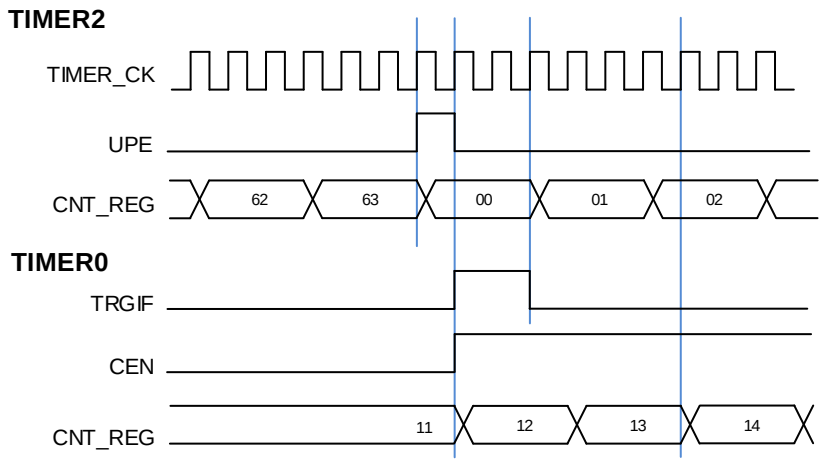


在这个例子中，我们也可以使用更新事件代替使能信号作为触发源。见*图23-57. 用定时器2的更新事件来触发定时器0*，按以下步骤进行：

1. 配置定时器2为主模式，发送它的更新事件（UPE）作为触发输出（配置TIMER2_CTL1寄存器的MMC0 = 4'b0010）；
2. 配置定时器2的周期（TIMER2_CARL寄存器）；
3. 配置定时器0工作在事件模式，定时器0输入触发源为定时器2（SYSCFG_TIMER0CFG0寄存器中的TRCFG5[4:0] = 5'b00011）

4. 写1到CEN来开启定时器2（TIMER2_CTL0寄存器）。

图 23-57. 用定时器 2 的更新事件来触发定时器 0

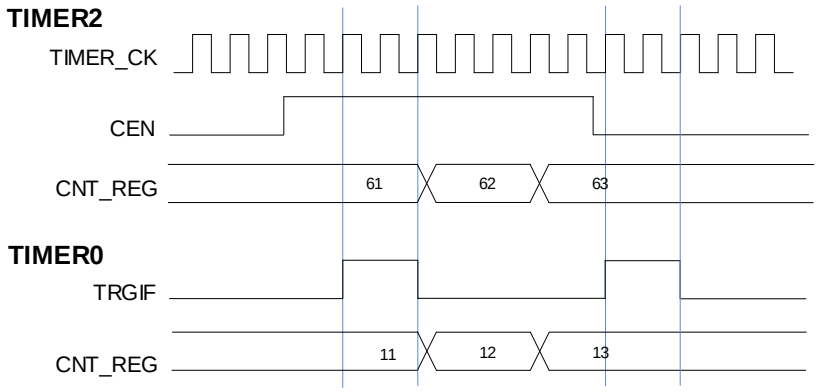


■ 使用定时器2的使能/O0CPRE参考信号来使能定时器0计数。

在这个例子中，我们使用定时器2的使能输出来控制定时器0的使能。如*图23-58. 用定时器2的使能来选通定时器0*，在定时器2被使能后，定时器0在内部分频的时钟上开始计数。两个计数器的时钟频率都是由TIMER_CK时钟3分频得来（$f_{CNT_CLK} = f_{TIMER_CK} / 3$），步骤如下：

1. 配置定时器2在主模式，配置其输出使能信号作为触发输出（配置TIMER2_CTL1寄存器的MMC0 = 4'b0001）；
2. 配置定时器0工作在暂停模式，定时器0输入触发源为定时器2（SYSCFG_TIMER0CFG0寄存器中的TRCFG4[4:0] = 5'b00011）；
3. 写1到CEN位来使能定时器0（TIMER0_CTL0寄存器）；
4. 写1到CEN位来启动定时器2（TIMER0_CTL0寄存器）；
5. 写0到CEN位来停止定时器2（TIMER0_CTL0寄存器）。

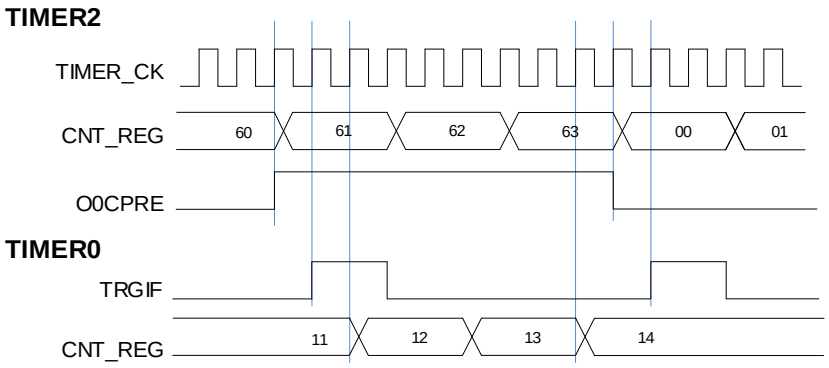
图 23-58. 用定时器 2 的使能来选通定时器 0



这个例子中，我们也可以使用定时器2的O0CPRE信号代替其使能信号输出作为触发源。步骤如下：

1. 配置定时器2在主模式下，配置O0CPRE信号为触发输出（配置TIMER2_CTL1寄存器的MMS=3’b100）；
2. 配置定时器2的O0CPRE波形（TIMER2_CH0CTL寄存器）；
3. 配置定时器0工作在暂停模式，定时器0输入触发源为定时器2（SYSCFG_TIMER0CFG0寄存器中的TRCFG4[4:0] = 5’b00011）；
4. 写1到CEN位来使能定时器0（TIMER0_CTL0寄存器）；
5. 写1到CEN位来开启定时器2（TIMER0_CTL0寄存器）。

图 23-59. 用定时器 2 的 O0CPRE 信号选通定时器 1



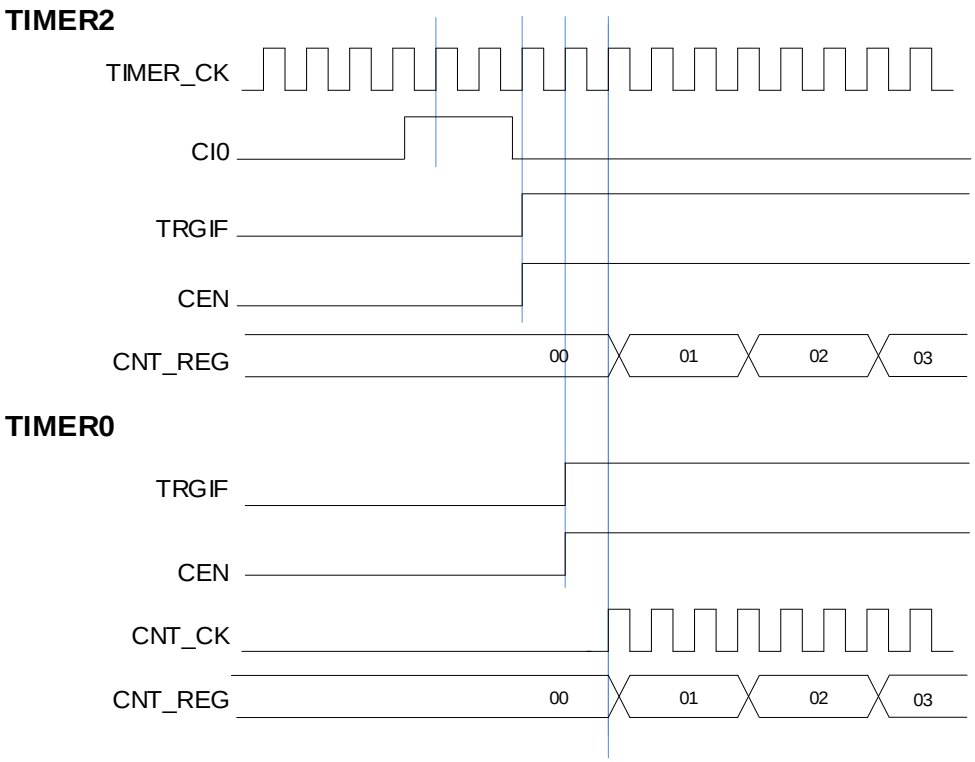
■ 使用一个外部触发来同步两个定时器

配置定时器2的使能信号触发定时器0的开启，配置定时器2的CI0输入信号上升沿来触发定时器2。为了确保两个定时器同步开启，定时器2必须配置在主/从模式。步骤如下：

1. 配置定时器2工作在事件模式，定时器2输入触发源为CI0的触发输入CI0F_ED（SYSCFG_TIMER02CFG0寄存器中的TRCFG5[4:0] = 5’b00101）；
2. 写MSM=1（TIMER2_SMCFG寄存器）来配置定时器2工作在主/从模式；
3. 配置定时器0工作在事件模式，定时器0输入触发源为定时器2（SYSCFG_TIMER0CFG0寄存器中的TRCFG5[4:0] = 5’b00011）。

当定时器2的CI0信号产生上升沿时，两个定时器的计数器在内部时钟下开始同步计数，二者的TRGIF标志位都被置1。

图 23-60. 用定时器 2 的 CI0 输入来触发定时器 0 和定时器 2



计数器同步、初始方向和值刷新

在一些定时器菊花链配置中，可以同时触发多个定时器使它们同步开始计数。由于不同的定时器使用不同的时钟总线，在长时间计数后可能会发生相移。通过触发信号定时刷新同步计数器，可以消除硬件上产生的多个定时器计数器之间的相移。此外，可以通过软件配置计数器初始值（TIMERx_CINITV寄存器中）来控制多个同步定时器之间的相移关系。

举例说明，配置3个高级定时器使用复位 + 事件模式，通过将TIMERx_CINITCTL寄存器中的CINITVEN位置1来使能计数器的初始值加载功能。如*图23-61. 配置相移方法的框图*和*图23-62. 3个定时器的相移框图*。从计数器初始值寄存器（TIMERx_CINITV）加载的3个初始计数值分别为0、20和40，分别在3个计数器中生成等距的相移（20 × TIMER_CK）。

图中红色标记的计数值9998表示在长时间的计数中，不同时钟总线上的计时器之间可能存在的相移。TIMER1运行在主模式下输出触发信号，同步刷新从定时器的计数值，从而避免相移的积累。

图 23-61. 配置相移方法的框图

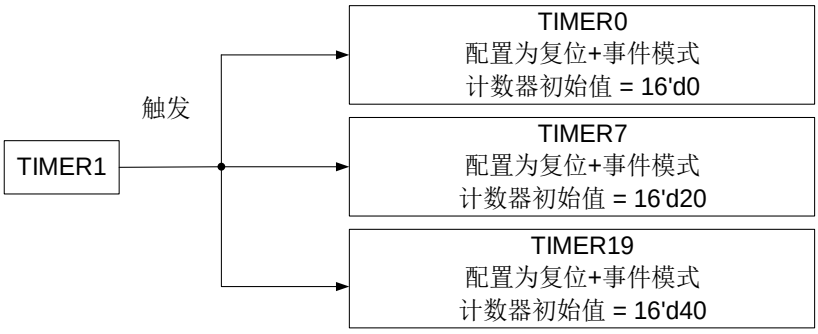
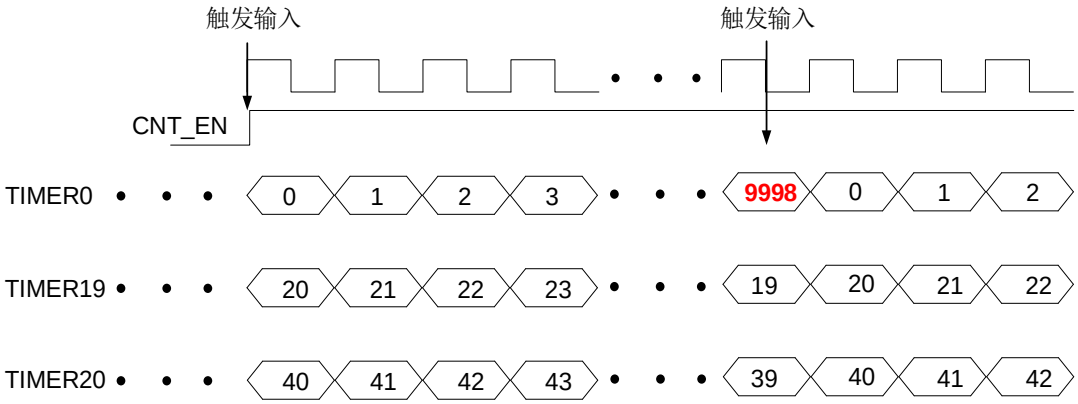


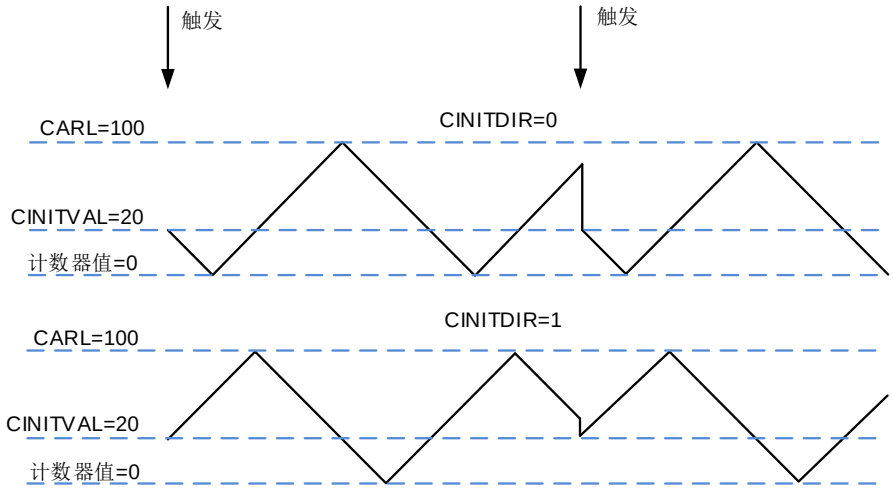
图 23-62. 3 个定时器的相移框图



当计数器使用中央对齐计数模式时，可以通过TIMERx_CINITCTL寄存器的CINITDIR位来设置复位后计数器的计数方向。只有在CINITVEN位置1时，CINITDIR位才会生效。当计数器使用向上计数和向下计数模式时，CINITDIR位无效。当计数器使用向上计数和向下计数模式时，计数器复位后的计数方向由TIMERx_CTL0寄存器的DIR位确定。

当CINITDIR位值为0时，复位计数器后的计数方向为向下计数；当CINITDIR位值为1时，复位计数器后的计数方向为向上计数。具体如*图23-63. 中央对齐计数模式下计数器复位后的计数方向*所示。

图 23-63. 中央对齐计数模式下计数器复位后的计数方向



计数器的初始计数方向和初始值也可以通过软件同步事件刷新。将TIMERx_CINITCTL寄存器中的SWSYNCG位置1，可以产生一个软件同步事件，用于刷新TIMERx计数器的初始计数方向和初始值。

当高级定时器使用主模式下来同步其他高级定时器时，在TIMERx_CTL1寄存器中MMC0[3:0]位域需要设置为4'b1001。此时，TIMERx的软件同步事件（设置SWSYNCG位为1产生）可以作为TRGO0信号输出，用于同步其他高级定时器。

定时器 DMA 模式

定时器 DMA 模式是指通过 DMA 模块配置定时器的寄存器。有两个跟定时器 DMA 模式相关的寄存器：TIMERx_DMACFG 和 TIMERx_DMATB。当然，必须要使能 DMA 请求，一些内部中断事件可以产生 DMA 请求。当中断事件发生，TIMERx 会给 DMA 发送请求。DMA 配置成 M2P 模式，PADDR 是 TIMERx_DMATB 寄存器地址，DMA 就会访问 TIMERx_DMATB 寄存器。实际上，TIMERx_DMATB 寄存器只是一个缓冲，定时器会将 TIMERx_DMATB 映射到一个内部寄存器，这个内部寄存器由 TIMERx_DMACFG 寄存器中的 DMATA 来指定。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值为 0，表示 1 次传输，定时器的发送 1 个 DMA 请求就可以完成。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值不为 1，例如其值为 3，表示 4 次传输，定时器就需要再多发 3 次 DMA 请求。在这 3 次请求下，DMA 对 TIMERx_DMATB 寄存器的访问会映射到访问定时器的 DMATA+0x4、DMATA+0x8、DMATA+0xC 寄存器。总之，发生一次 DMA 内部中断请求，定时器会连续发送（DMATC+1）次请求。

如果再来1次DMA请求事件，TIMERx将会重复上面的过程。

输出 DIR 位

DIR位可以在CH2和CH3通道上输出，配置在TIMERx_CHCTL1寄存器中的CH2COMCTL[3:0]或CH3COMCTL[3:0]位域值为4b'1011来使能该功能。

当计数器工作在中央对齐模式时，该功能可用于指示计数器的计数方向。当计数器工作在译码器模式时，此功能可用于指示外部信号的旋转方向。

UPIF 位备份功能

可以通过配置TIMERx_CTL0寄存器中的UPIFBUEN位来使能UPIF位的备份功能，UPIF和UPIFBU位之间没有延迟，两者完全同步。

使能该功能后，TIMERx_INTF寄存器中的UPIF位将会被实时备份到TIMERx_CNT寄存器中的UPIFBU位。这可以避免在读计数器和中断处理时产生冲突的情况。

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。

23.1.5. TIMERx 寄存器（x = 0, 7, 19）

TIMER0基地址：0x4001 2C00

TIMER7基地址：0x4001 3400

TIMER19基地址：0x4001 5000

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			ADMEN	UPIFBUEN	保留	CKDIV[1:0]		ARSE	CAM[1:0]		DIR	SPM	UPS	UPDIS	CEN
			rw	rw		rw		rw	rw		rw	rw	rw	rw	rw

位/位域	名称	描述
31:13	保留	必须保持复位值。
12	ADMEN	微调模式使能 0：微调模式禁能 1：微调模式使能 注意：该位只能在CEN位为0时修改。
11	UPIFBUEN	UPIF位备份使能 0：备份禁能。UPIF位没有备份到TIMERx_CNT寄存器中的UPIFBU位 1：备份使能。UPIF 位备份到 TIMERx_CNT 寄存器中的 UPIFBU 位
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）与死区时间和采样时钟（DTS）之间的分频系数，死区发生器和数字滤波器会用到 DTS 时间。 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，通道的比较中断标志置 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，通道的比较中断标志置 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，通道的比较中断标志都会置 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，CEN 硬件清零并且计数器停止计数
2	UPS	更新请求源 软件配置该位，选择更新事件源。 0：使能后，下述任一事件产生更新中断或 DMA 请求： – UPG 位被置 1 – 计数器上溢/下溢 – 从模式控制器产生的更新 1：使能后只有计数器上溢/下溢才产生更新中断或 DMA 请求
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.当以下事件之一发生时，更新事件产生，具有缓存的寄存器被装入它们的预装载值： – UPG 位被置 1 – 计数器上溢/下溢 – 从模式控制器产生一个更新事件 1：更新事件禁能。带有缓存的寄存器保持原有值，如果 UPG 位被置 1 或者从模式控制器产生一个硬件复位事件，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。触发模式可以自动地通过硬件设置 CEN 位。

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
CCUC[2:1]		保留				MMC0[3]	保留		MMC1[2:0]			保留			
rw						rw			rw						

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	MMC0[2:0]			DMAS	CCUC[0]	保留	CCSE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw		rw

位/位域	名称	描述
31:30	CCUC[2:1]	换相控制影子寄存器更新控制 请参考 CCUC [0]的描述。
29:26	保留	必须保持复位值。
25	MMC0[3]	主模式控制 0 请参考 MMC0[2:0]的描述。
24:23	保留	必须保持复位值。
22:20	MMC1[2:0]	主模式控制 1 该位域控制 TRGO1 信号的选择。 000：复位。TIMERx_SWEVG 寄存器的 UPG 位被置 1 或从模式控制器产生复位触发一次 TRGO1 脉冲，后一种情况下，TRGO1 上的信号相对实际的复位会有一个延迟。 001：使能。此模式可用于同时启动多个定时器或控制在一段时间内使能从定时器。主模式控制器选择计数器使能信号作为触发输出 TRGO1。当 CEN 控制位被置 1 或者暂停模式下触发输入为高电平时，计数器使能信号被置 1。在暂停模式下，计数器使能信号受控于触发输入，在触发输入和 TRGO1 上会有一个延迟，除非选择了主 / 从模式。 010：更新。主模式控制器选择更新事件作为 TRGO1。 011：捕获/比较脉冲。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 TRGO1 脉冲。

		100：比较。在这种模式下主模式控制器选择 O0CPRE 信号被用于作为触发输出 TRGO1。 101：比较。在这种模式下主模式控制器选择 O1CPRE 信号被用于作为触发输出 TRGO1。 110：比较。在这种模式下主模式控制器选择 O2CPRE 信号被用于作为触发输出 TRGO1。 111：比较。在这种模式下主模式控制器选择 O3CPRE 信号被用于作为触发输出 TRGO1。 **注意**：从 TIMER 或 ADC 的时钟必须在接收到主 TIMER 的 TRGO1 事件之前使能，且当接收到主 TIMER 的 TRG01 事件时，不能实时修改从 TIMER 和 ADC 时钟。
19:16	保留	必须保持复位值。
15	ISO3N	多模式通道 3 的互补通道空闲状态输出 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 的互补通道空闲状态输出 参考 ISO0N 位。
12	ISO2	通道 2 的空闲状态输出 参考 ISO0 位。
11	ISO1N	多模式通道 1 的互补通道空闲状态输出 参考 ISO0N 位。
10	ISO1	通道 1 的空闲状态输出 参考 ISO0 位。
9	ISO0N	多模式通道 0 的互补通道空闲状态输出 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_CCHP0 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_CCHP0 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	MMC0[2:0]	主模式控制 0 该位域控制 TRGO0 信号的选择，TRGO0 信号由主定时器发给从定时器用于同步功能。 0000：复位。TIMERx_SWEVG 寄存器的 UPG 位被置 1 或从模式控制器产生复位触发一次 TRGO0 脉冲，后一种情况下，TRGO0 上的信号相对实际的复位会有一个延迟。 0001：使能。此模式可用于同时启动多个定时器或控制在一段时间内使能从定时器。主模式控制器选择计数器使能信号作为触发输出 TRGO0。当 CEN 控制位被置 1 或者暂停模式下触发输入为高电平时，计数器使能信号被置 1。在暂停模式下，计数器使能信号受控于触发输入，在触发输入和 TRGO0 上会有一个延迟，除非选择了主/从模式。 0010：更新。主模式控制器选择更新事件作为 TRGO0。 0011：捕获/比较脉冲。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 TRGO0 脉冲。 0100：比较。在这种模式下主模式控制器选择 O0CPRE 信号作为触发输出 TRGO0。 0101：比较。在这种模式下主模式控制器选择 O1CPRE 信号作为触发输出 TRGO0。 0110：比较。在这种模式下主模式控制器选择 O2CPRE 信号作为触发输出 TRGO0。 0111：比较。在这种模式下主模式控制器选择 O3CPRE 信号作为触发输出 TRGO0。 1000：译码器时钟输出。在这种模式下主模式控制器选择译码器时钟信号作为触发输出TRGO0。此值仅用于正交译码器模式0~4和译码器模式0~3。 1001：在这种模式下主模式控制器选择软同步事件信号（设置SWSYNCG位为1产生）作为触发输出TRGO0。 1010~1111：保留
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 CHx/MCHx 的 DMA 请求 1：当更新事件发生，发送通道 CHx/MCHx 的 DMA 请求
2	CCUC[0]	换相控制影子寄存器更新控制 CCUC[2:1]和 CCUC[0]位域用于控制换相控制影子寄存器的更新。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根据 CCUC[2:0]位域的控制如下： 000：CMTG 位被置 1 时，更新影子寄存器 001：当 CMTG 位被置 1 或检测到 TRIGI 上升沿时，影子寄存器更新 100：当计数器上溢事件发生时，影子寄存器更新 101：当计数器下溢事件发生时，影子寄存器更新 110：当计数器上溢 / 下溢事件发生时，影子寄存器更新 其他值：保留 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
保留					PRMRPSEL	DECMODS	DECMODEN	保留							
					rw	rw	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	保留			OCRC	保留		
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	PRMRPSEL	暂停 + 复位模式复位极性选择 0：计数器在下降沿复位 1：计数器在上升沿复位
25	DECMODS	译码器模式更新源 该位用于选择译码器模式的更新源。 0：模式修改后，通过 TIMER 的更新事件更新译码器模式 1：模式修改后，通过索引事件更新译码器模式
24	DECMODEN	译码器模式在线修改使能 0：译码器模式在线修改禁能 1：译码器模式在线修改使能 该位置1时，译码器模式可以从一种模式修改为另一种模式。
23:16	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	从模式的一部分为了使能外部时钟模式 1 在外部时钟模式 1，计数器由 ETIFP 信号上的任意有效边沿驱动。 0：外部时钟模式 1 禁能 1：外部时钟模式 1 使能 复位模式，暂停模式和事件模式可以与外部时钟模式 1 同时使用，但 TSCFGy[4:0](y = 3, 4, 5)位域的值不能为 5'b01000。 如果外部时钟模式 0 和外部时钟模式 1 同时被使能，外部时钟的输入是 ETIFP。 **注意**：外部时钟模式 0 使能在 SYSCFG_TIMERxCFG1 寄存器中的 TSCFG6[4:0] 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 ETIFP 的频率。 00：预分频禁能 01：ETI 频率被 2 分频 10：ETI 频率被 4 分频 11：ETI 频率被 8 分频
11:8	ETFC[3:0]	外部触发滤波控制 数字滤波器是一个事件计数器，它记录到 N 个事件后会产生一个输出的跳变。这些位定义了对 ETI 信号采样的频率和对 ETI 数字滤波的带宽。 0000：滤波器禁能 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
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 TRGO0，定时器被连接在一起，TRGO0 用做启动事件。 0：主从模式禁能

		1：主从模式使能
6:4	保留	必须保持复位值。
3	OCRC	OxCPRE / MOxCPRE清除源选择 该位用于选择OxCPRE / MOxCPRE清除源。 0：OCPRE_CLR_INT连接OCPRE_CLR输入 1：OCPRE_CLR_INT连接到ETIF
2:0	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
CH3COMADDIE	CH2COMADDIE	CH1COMADDIE	CH0COMADDIE	MCH3DEN	MCH2DEN	MCH1DEN	MCH0DEN	MCH3IE	MCH2IE	MCH1IE	MCH0IE	INDERRIE	DIRTRANIE	DECDISIE	DECJIE
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
INDIE	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	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。
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。
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。
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。

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] = 2'b00 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] = 2'b00 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] = 2'b00 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] = 2'b00 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] = 2'b00 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] = 2'b00 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] = 2'b00 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] = 2'b00 时）。
19	INDERRIE	索引错误中断使能 0：索引错误中断禁能 1：索引错误中断使能
18	DIRTRANIE	计数方向变化中断使能 0：计数方向变化中断禁能 1：计数方向变化中断禁能
17	DECDISIE	正交译码器信号断线检测使能 0：禁能 1：使能 **注意**：该位仅用于正交译码器信号断线检测使能（DECDISDEN =1）时。
16	DECJIE	正交译码器信号跳变（两个信号同时发生跳变）中断使能 0：禁能 1：使能 **注意**：该位仅用于正交译码器信号同时跳变检测使能（DECJDEN =1）时。
15	INDIE	索引中断使能 0：索引中断禁能 1：索引中断使能
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	INDERRIF	DIRTRANIF	DECDISIF	DECJIF
rc_w0	rc_w0	rc_w0	rc_w0	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
INDIF	保留	SYSBIF	CH3OF	CH2OF	CH1OF	CH0OF	BRK1IF	BRK0IF	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	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。
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	INDERRIF	索引错误中断标志位 此标志由硬件置 1 软件清 0。 0：无索引错误中断发生 1：索引错误中断发生
18	DIRTRANIF	计数方向变化中断标志位 译码器模式下，此标志在 TIMERx_CTL0 寄存器中的 DIR 位发生变化时由硬件置 1，并由软件清 0。 0：无计数方向变化中断发生 1：计数方向变化中断发生
17	DECDISIF	正交译码器信号断线中断标志位 0：无正交译码器信号断线中断发生 1：正交译码器信号断线中断发生 **注意**：该位仅用于正交译码器信号断线检测使能（DECDISDEN =1）时。
16	DECJIF	正交译码器信号跳变（两个信号同时发生跳变）中断标志位 0：无正交译码器信号跳变中断发生 1：正交译码器信号跳变中断发生 **注意**：该位仅用于正交译码器信号同时跳变检测使能（DECJDEN =1）时。
15	INDIF	索引中断标志位 此标志由硬件置 1 软件清 0。 0：无索引中断发生 1：索引中断发生
14	保留	必须保持复位值。
13	SYSBIF	系统源中止事件中断标志位 当系统中止源有效时，该位由硬件置1，当系统源无效时，该位由软件清零。

		0：无系统中止事件中断发生 1：系统中止事件中断发生 **注意**：当该位置1时，在通道输出恢复前，该位必须由软件清零。
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	BRK1IF	BREAK1 中断标志位 一旦 BREAK1 输入有效，由硬件对该位置 1。如果 BREAK1 输入无效，则该位可由软件清 0。 0：无 BREAK1 事件产生 1：BREAK1 输入上检测到有效电平。当 TIMERx_DMAINTEN 寄存器中的 BRKIE=1 时，中断产生
7	BRK0IF	BREAK0 中断标志位 一旦 BREAK0 输入有效，由硬件对该位置 1。如果 BREAK0 输入无效，则该位可由软件清 0。 0：无 BREAK0 事件产生 1：BREAK0 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，此标志由硬件置 1。此位由软件清 0。当从模式控制器处于除暂停模式外的其它模式时，在触发输入端检测到有效边沿，产生触发事件。当从模式控制器处于暂停模式时，触发输入的任意边沿都可以产生触发事件。 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
保留							BRK1G	BRK0G	TRGG	CMTG	CH3G	CH2G	CH1G	CH0G	UPG
							w	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。
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
19:9	保留	必须保持复位值。
8	BRK1G	产生 BREAK1 事件 该位由软件置 1，用于产生一个 BREAK1 事件，由硬件自动清 0。当此位被置 1 时，POEN 位被清 0 且 BRK1IF 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 BREAK1 事件 1：产生 BREAK1 事件
7	BRK0G	产生 BREAK0 事件 该位由软件置 1，用于产生一个 BREAK0 事件，由硬件自动清 0。当此位被置 1 时，POEN 位被清 0 且 BRK0IF 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 BREAK0 事件 1：产生 BREAK0 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	CH1ADDUPS	CH0ADDUPS	保留	CH1COMCTL[3]	保留							CH0COMCTL[3]

		保留	保留	保留	保留		保留								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
CH1COM CEN	CH1COMCTL[2:0]			CH1COM SEN	CH1COM FEN	CH1MS[1:0]		CH0COM CEN	CH0COMCTL[2:0]			CH0COM SEN	CH0COM FEN	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_CCHP0 寄存器的 PROT [1:0]=11 且 CH0MS =000 时此位不能被改变。
27	CH1ADDUPS	通道1附加寄存器更新源 0：在发生更新事件时，更新TIMERx_CH1COMV_ADD寄存器 1：在发生计数器计数值匹配CH1VAL值时，更新TIMERx_CH1COMV_ADD寄存器
26	CH0ADDUPS	通道0附加寄存器更新源 0：在发生更新事件时，更新TIMERx_CH0COMV_ADD寄存器 1：在发生计数器计数值匹配 CH0VAL 值时，更新 TIMERx_CH0COMV_ADD 寄存器
25	保留	必须保持复位值。
24	CH1COMCTL[3]	通道 1 输出比较控制 参见 CH0COMCTL[2:0]描述
23:17	保留	必须保持复位值。
16	CH0COMCTL[3]	通道 0 输出比较控制

		参见 CH0COMCTL[2:0]描述
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 输出比较控制 参见 CH0COMCTL[2:0]描述
11	CH1COMSEN	通道 1 输出比较影子寄存器使能 参见 CH0COMSEN 描述
10	CH1COMFEN	通道1输出比较快速使能 参见 CH0COMFEN 描述
9:8	CH1MS[1:0]	通道 1 模式选择 CH1MS[2:0]位域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] = 2'b00 时，TIMERx_CHCTL2 寄存器的 CH1EN 位清 0；当 MCH1MSEL[1:0] = 2'b01 或 2'b11 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器输入被选中时（由 SYSCFG_TIMERxCFG2(x = 0, 7, 19)寄存器中的 TSCFG15[4:0]位域选择）。 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 输出比较控制 CH0COMCTL[3]和 CH0COMCTL[2:0]位域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] = 2'b11 时，CH0COMCTL[3]和 CH0COMCTL[2:0]位域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。 0000：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 0001：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。

0010：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。

0011：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。

0100：强制为低。强制 O0CPRE 为低电平。

0101：强制为高。强制 O0CPRE 为高电平。

0110：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。

0111：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。

1000：可延时的单脉冲模式0。O0CPRE的输出情况类似与PWM模式0。在向上计数模式时，O0CPRE先输出有效电平，当外部触发事件发生时，立即输出无效电平，当下一次更新事件发生时，再变成有效电平；在向下计数模式时，O0CPRE先输出无效电平，当外部触发事件发生时，立即输出有效电平，当下一次更新事件发生时，再变成无效电平。

1001：可延时的单脉冲模式1。O0CPRE的输出情况类似与PWM模式1。在向上计数模式时，O0CPRE先输出无效电平，当外部触发事件发生时，立即输出有效电平，当下一次更新事件发生时，再变成无效电平；在向下计数模式时，O0CPRE先输出有效电平，当外部触发事件发生时，立即输出无效电平，当下一次更新事件发生时，再变成有效电平。

1010 / 1011 / 1100 / 1101 / 1110 / 1111：保留

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***。

在 PWM 模式 0 或 PWM 模式 1 中，只有当比较结果改变了或者输出比较模式中从时基模式切换到 PWM 模式时，O0CPRE 电平才改变。

当 CH0 和 MCH0 输出互补时，该位域预装载。若 CCSE =1，则该位域只在通道换相事件发生时更新。

当 TIMERx_CCHP0 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（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_CCHP0 寄存器的 PROT [1:0]=11 且 CH0MS =000 时此位不能被改变。
2	CH0COMFEN	通道0输出比较快速使能 当该位为1时，如果通道配置为PWM模式0或者PWM模式1，会加快捕获 / 比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH0_O被设置为比较电平而与比较结果无关。 0：通道0输出比较快速禁能。当触发器的输入有一个有效沿时，激活CH0_O输出的最小延时为5个时钟周期 1：通道0输出比较快速使能。当触发器的输入有一个有效沿时，激活CH0_O输出的最小延时为3个时钟周期
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] = 2'b00 时，TIMERx_CHCTL2 寄存器的 CH1EN 位清 0；当 MCH0MSEL[1:0] = 2'b01 或 2'b11 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（由 SYSCFG_TIMERxCFG2(x = 0, 7, 19)寄存器中的 TSCFG15[4:0]位域选择）。 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

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 CI0 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	CH0CAPPSC[1:0]	通道 0 输入捕获预分频器 这 2 位定义了通道 0 输入的预分频系数。当 TIMERx_CHCTL2 寄存器中的 CH0EN =0 时，则预分频器复位。 00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获 01：每 2 个事件触发一次捕获 10：每 4 个事件触发一次捕获 11：每 8 个事件触发一次捕获
1:0	CH0MS[1:0]	通道 0 模式选择 与输出比较模式相同。

通道控制寄存器 1（TIMERx_CHCTL1）

地址偏移：0x1C

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3MS[2]	CH2MS[2]	CH3COMADDSEN	CH2COMADDSEN	CH3ADDUPS	CH2ADDUPS	保留	CH3COMCTL[3]	保留							CH2COMCTL[3]
		保留	保留	保留	保留		保留								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

CH3COMCEN	CH3COMCTL[2:0]	CH3COMSEN	CH3COMFEN	CH3MS[1:0]	CH2COMCEN	CH2COMCTL[2:0]	CH2COMSEN	CH2COMFEN	CH2MS[1:0]
CH3CAPFLT[3:0]		CH3CAPPSC[1:0]			CH2CAPFLT[3:0]		CH2CAPPSC[1:0]		
rw		rw		rw	rw		rw		rw

输出比较模式：

位/位域	名称	描述
31	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_CCHP0 寄存器的 PROT [1:0]=11 且 CH2MS =000 时此位不能被改变。
27	CH3ADDUPS	通道3附加寄存器更新源 0：在发生更新事件时，更新TIMERx_CH3COMV_ADD寄存器 1：在发生计数器计数值匹配 CH3VAL 值时，更新 TIMERx_CH3COMV_ADD 寄存器
26	CH2ADDUPS	通道2附加寄存器更新源 0：在发生更新事件时，更新TIMERx_CH2COMV_ADD寄存器 1：在发生计数器计数值匹配 CH2VAL 值时，更新 TIMERx_CH2COMV_ADD 寄存器
25	保留	必须保持复位值。
24	CH3COMCTL[3]	通道 3 输出比较控制 请参考 CH2COMCTL[2:0]描述
23:17	保留	必须保持复位值。
16	CH2COMCTL[3]	通道 2 输出比较控制 请参考 CH2COMCTL[2:0]描述
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 输出比较控制 参见 CH0COMCTL 描述。
11	CH3COMSEN	通道 3 输出比较影子寄存器使能 参见 CH0COMSEN 描述。
10	CH3COMFEN	通道3输出比较快速使能 参见 CH2COMFEN 描述
9:8	CH3MS[1:0]	通道 3 模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭（当 MCH3MSEL[1:0] = 2'b00 时，TIMERx_CHCTL2 寄存器的 CH3EN 位清 0；当 MCH3MSEL[1:0] = 2'b01 或 2'b11 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器输入被选中时（由 SYSCFG_TIMERxCFG2(x = 0, 7, 19)寄存器中的 TSCFG15[4:0]位域选择）。 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 输出比较控制 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] = 2'b11，CH2COMCTL[3]和 CH2COMCTL[2:0]位域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。 0000：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 0001：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。 0010：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。

0011：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。

0100：强制为低。强制 O2CPRE 为低电平

0101：强制为高。强制 O2CPRE 为高电平

0110：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。

0111：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。

1000：可延时的单脉冲模式0。O2CPRE的输出情况类似与PWM模式0。在向上计数模式时，O2CPRE先输出有效电平，当外部触发事件发生时，立即输出无效电平，当下一次更新事件发生时，再变成有效电平；在向下计数模式时，O2CPRE先输出无效电平，当外部触发事件发生时，立即输出有效电平，当下一次更新事件发生时，再变成无效电平。

1001：可延时的单脉冲模式1。O2CPRE的输出情况类似与PWM模式1。在向上计数模式时，O2CPRE先输出无效电平，当外部触发事件发生时，立即输出有效电平，当下一次更新事件发生时，再变成无效电平；在向下计数模式时，O2CPRE先输出有效电平，当外部触发事件发生时，立即输出无效电平，当下一次更新事件发生时，再变成有效电平。

1010：可编程的脉冲输出。当计数器计数值与输出比较寄存器TIMERx_CH2CV匹配时，CH2_O上输出可编程的脉冲。脉冲由TIMERx_DECCTL寄存器中的OPPSC[2:0]和OPWID[7:0]位域配置。

1011：输出DIR位。O2CPRE信号表示DIR位（在TIMERx_CTL0寄存器中）的值。

1100 / 1101 / 1110 / 1111：保留

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***。

在 PWM 模式 0 或 PWM 模式 1 中，只有当比较结果改变了或者输出比较模式中从时基模式切换到 PWM 模式时，O2CPRE 电平才改变。

当 CH2 和 MCH2 输出互补时，该位域预装载。若 CCSE =1，则该位域只在通道换相事件发生时更新。

当 TIMERx_CCHP0 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（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_CCHP0 寄存器的 PROT [1:0]=11 且 CH2MS =000 时此位不能被改变。
2	CH2COMFEN	通道2输出比较快速使能 当该位为1时，如果通道配置为PWM模式0或者PWM模式1，会加快捕获 / 比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH2_O被设置为比较电平而与比较结果无关。 0：通道2输出比较快速禁能。当触发器的输入有一个有效沿时，激活CH2_O输出的最小延时为5个时钟周期 1：通道2输出比较快速使能。当触发器的输入有一个有效沿时，激活CH2_O输出的最小延时为 3 个时钟周期
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] = 2'b00 时，TIMERx_CHCTL2 寄存器的 CH2EN 位清 0；当 MCH1MSEL[1:0] = 2'b01 或 2'b11 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（由 SYSCFG_TIMERxCFG2(x = 0, 7, 19)寄存器中的 TSCFG15[4:0]位域选择）。 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	CH3MS[2]	通道 3 模式选择 与输出模式相同。
30	CH2MS[2]	通道 2 模式选择 与输出模式相同。
29:16	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 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 CI2 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	CH2CAPPSC[1:0]	通道 2 输入捕获预分频器 这 2 位定义了通道 2 输入的预分频系数。当 TIMERx_CHCTL2 寄存器中的 CH2EN =0 时，则预分频器复位。 00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获 01：每 2 个事件触发一次捕获 10：每 4 个事件触发一次捕获 11：每 8 个事件触发一次捕获
1:0	CH2MS[1:0]	通道 2 模式选择 与输出比较模式相同

通道控制寄存器 2（TIMERx_CHCTL2）

地址偏移：0x20

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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] = 2'b11 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_CCHP0 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_CCHP0 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
UPIFBU	保留														
r															

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	UPIFBU	UPIF位备份 该位只读，是 TIMERx_INTF 寄存器的 UPIF 位的备份值。当 UPIFBUEN = 1 时，该位有效，若 UPIFBUEN =0，该位保留，读取该位值为零。
30:16	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。 当 PWMADMEN = 0 时，该位域表示当前计数器的值。写操作该位域可以改变计数器的值。 当 PWMADMEN = 1 时，该位域仅用于表示计数器值的整数部分，不包含小数部分。

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 PSC 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 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
CARL[19:16]				保留											
rw															

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

CARL[15:0]
rw

位/位域	名称	描述
31:28	CARL[19:16]	计数器自动重载值（16 位至 19 位） 当 PWMADMEN = 0 时，CARL[19:16]位域值 0000。 当 PWMADMEN = 1 时，CARL[19:16]位域用于表示自动重载值的小数部分。
27:16	保留	必须保持复位值。
15:0	CARL[15:0]	计数器自动重载值（0 位至 15 位） 这些位定义了计数器的自动重载值。 当 PWMADMEN = 0 时，CARL[15:0]位域用于表示计数器的自动重载值。 当 PWMADMEN = 1 时，CARL[15:0]位域用于表示自动重载值的整数部分。

重复计数寄存器 0（TIMERx_CREP0）

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								CREP0[7:0]							
								rw							

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	CREP0[7:0]	重复计数器的值 0 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）。 **注意**：当 TIMERx_CFG 寄存器中的 CREPSEL =0 时，使用该位域。

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

CH0VAL[19:16]	保留

rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															

rw

位/位域	名称	描述
31:28	CH0VAL[19:16]	通道 0 的捕获或比较值（16 位到 19 位） 当通道 0 配置为输入模式时，CH0VAL[19:16]位域值为 0000。 当通道 0 配置为输出模式时，该位域包含了即将和计数器比较的值。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。 当 PWMADMEN = 0 时，CH0VAL[19:16]位域值为 0000。 当 PWMADMEN = 1 时，CH0VAL[19:16]位域用于表示比较值的小数部分。
27:16	保留	必须保持复位值。
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值（0 位至 15 位） 当通道 0 配置为输入模式时，CH0VAL[15:0]决定了上次捕获事件的计数器值，且该位域只读。 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。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。 当 PWMADMEN = 0 时，CH0VAL[15:0]位域值表示比较值。 当 PWMADMEN = 1 时，CH0VAL[15:0]位域用于表示比较值的整数部分。

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
CH1VAL[19:16]				保留											

rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															

rw

位/位域	名称	描述
31:28	CH1VAL[19:16]	通道 1 的捕获或比较值（16 位到 19 位） 当通道 1 配置为输入模式时，CH1VAL[19:16]位域值为 0000。 当通道 1 配置为输出模式时，该位域包含了即将和计数器比较的值。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。

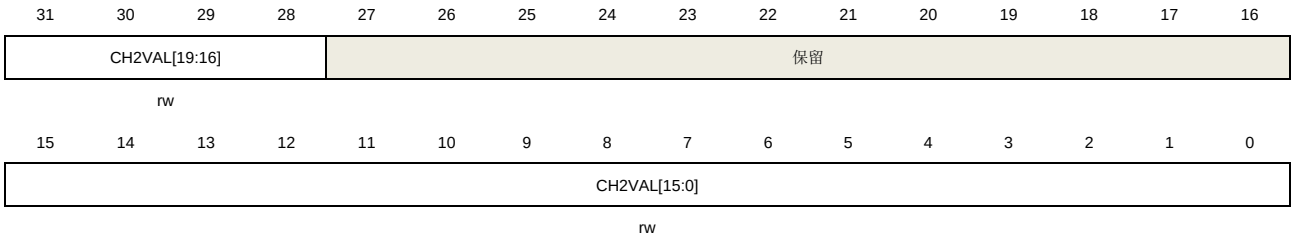
		当 PWMADMEN = 0 时，CH1VAL[19:16]位域值为 0000。 当 PWMADMEN = 1 时，CH1VAL[19:16]位域用于表示比较值的小数部分。
27:16	保留	必须保持复位值。
15:0	CH1VAL[15:0]	通道 1 的捕获或比较值（0 位至 15 位） 当通道 1 配置为输入模式时，CH1VAL[15:0]决定了上次捕获事件的计数器值，且该位域只读。 当通道 1 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。 当 PWMADMEN = 0 时，CH1VAL[15:0]位域值表示比较值。 当 PWMADMEN = 1 时，CH1VAL[15:0]位域用于表示比较值的整数部分。

通道 2 捕获 / 比较寄存器（TIMERx_CH2CV）

地址偏移：0x3C

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:28	CH2VAL[19:16]	通道 2 的捕获或比较值（16 位到 19 位） 当通道 2 配置为输入模式时，CH2VAL[19:16]位域值为 0000。 当通道 2 配置为输出模式时，该位域包含了即将和计数器比较的值。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。 当 PWMADMEN = 0 时，CH2VAL[19:16]位域值为 0000。 当 PWMADMEN = 1 时，CH2VAL[19:16]位域用于表示比较值的小数部分。
27:16	保留	必须保持复位值。
15:0	CH2VAL[15:0]	通道 2 的捕获或比较值（0 位至 15 位） 当通道 2 配置为输入模式时，CH2VAL[15:0]决定了上次捕获事件的计数器值，且该位域只读。 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。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。 当 PWMADMEN = 0 时，CH2VAL[15:0]位域值表示比较值。

当 PWMADMEN = 1 时，CH2VAL[15:0]位域用于表示比较值的整数部分。

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
CH3VAL[19:16]				保留											
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3VAL[15:0]															
rw															

位/位域	名称	描述
31:28	CH3VAL[19:16]	通道 3 的捕获或比较值（16 位到 19 位） 当通道 3 配置为输入模式时，CH3VAL[19:16]位域值为 0000。 当通道 3 配置为输出模式时，该位域包含了即将和计数器比较的值。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。 当 PWMADMEN = 0 时，CH3VAL[19:16]位域值为 0000。 当 PWMADMEN = 1 时，CH3VAL[19:16]位域用于表示比较值的小数部分。
27:16	保留	必须保持复位值。
15:0	CH3VAL[15:0]	通道 3 的捕获或比较值（0 位至 15 位） 当通道 3 配置为输入模式时，CH3VAL[15:0]决定了上次捕获事件的计数器值，且该位域只读。 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。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。 当 PWMADMEN = 0 时，CH3VAL[15:0]位域值表示比较值。 当 PWMADMEN = 1 时，CH3VAL[15:0]位域用于表示比较值的整数部分。

互补通道保护寄存器 0（TIMERx_CCHP0）

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		BRK1LK	BRK0LK	BRK1REL	BRK0REL	BRK1P	BRK1EN	BRK1F[3:0]				BRK0F[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
POEN	OAEN	BRK0P	BRK0EN	ROS	IOS	PROT[1:0]		DTCFG[7:0]							
rw	rw	rw	rw	rw	rw	rw		rw							

位/位域	名称	描述
31:30	保留	必须保持复位值。
29	BRK1LK	BREAK1输入锁存 请参考 BRK0LK 描述
28	BRK0LK	BREAK0输入锁存 0：BREAK0输入为输入模式 1：BREAK0输入为锁存模式 当BRK0LK置1时，BREAK0输入配置为开漏输出模式。 任何有效的BREAK0事件都会拉低BREAK0输入引脚电平，用于向外部设备提示有内部BREAK0事件发生。 此位只有在TIMERx_CCHP0寄存器的PROT[1:0] = 00时才可修改。 **注意**：对该位的每一次写操作，需要延时1个APB时钟才有效。
27	BRK1REL	BREAK1输入释放 请参考 BRK0REL 描述
26	BRK0REL	BREAK0输入释放 当BREAK0输入无效时，该位由硬件清零。 0：BREAK0输入锁存 1：BREAK0输入释放 当软件将该位置1时，将释放锁存输出控制（高阻态的开漏模式）。当BREAK0事件无效时，该位由硬件清零。 **注意**：对该位的每一次写操作，需要延时 1 个 APB 时钟才有效。
25	BRK1P	BREAK1输入信号极性 该位用于配置BREAK1输入信号的极性 0：BREAK1输入信号低电平有效 1：BREAK1输入信号高电平有效 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0] = 00 时才可修改。 **注意**：对该位的每一次写操作，需要延时 1 个 APB 时钟才有效。
24	BRK1EN	BREAK1输入信号使能 该位置1时，使能BREAK1输入信号。 0：BREAK1输入禁能 1：BREAK1输入使能 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0] = 00 时才可修改。 **注意**：

1） 对该位的每一次写操作，需要延时 1 个 APB 时钟才有效。
2） 该位仅用于 ROS=1 且 IOS=1 时

23:20	BRK1F[3:0]	BREAK1输入信号滤波 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 BREAK1 输入信号的采样频率和数字滤波器的长度。 0000：无滤波器，BREAK1 异步有效，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 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0] = 00 时才可修改。
19:16	BRK0F[3:0]	BREAK0输入信号滤波 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 BREAK0 输入信号的采样频率和数字滤波器的长度。 0000：无滤波器，BREAK0 异步有效，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 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0] = 00 时才可修改。
15	POEN	所有的通道输出使能 根据 OAEN 位，该位可以软件设置或者硬件自动设置。一旦中止输入有效，该位被硬件异步清 0。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
14	OAEN	自动输出使能 此位定义了 POEN 位是否可以被硬件自动置 1。 0：POEN 位不能被硬件置 1 1：如果中止输入无效，下一次更新事件发生时，POEN 位能被硬件自动置 1 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0] = 00 时才可修改。
13	BRK0P	BREAK0 输入信号极性 此位定义了 BREAK0 输入的极性。 0：BREAK0 输入低电平有效 1：BREAK0 输入高电平有效 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0] = 00 时才可修改。
12	BRK0EN	BREAK0 输入信号使能 此位置 1 使能 BREAK0 输入信号。 0：BREAK0 输入禁能 1：BREAK0 输入使能 此位只有在 TIMERx_CCHP0 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 23-6. 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_CCHP0 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 23-6. 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_CCHP0 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_CCHP0 寄存器中 BRK0EN / BRK0P / BRK1EN / BRK1P / OAEN / DTCFG 位、TIMERx_BRKCFG 寄存器中 BRKxP / BRKxEN（x = 0…3）位、TIMERx_FCCHPx（x = 0…3）寄存器中 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_CCHP0 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_CCHP0 寄存器被写入，这两位被写保护。
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'b 10x：DTvalue =（64+DTCFG [5:0]）x t_{DT}，t_{DT} = t_{DTS} * 2 DTCFG [7:5] = 3'b 110：DTvalue =（32+DTCFG [4:0]）x t_{DT}，t_{DT}= t_{DTS} * 8 DTCFG [7:5] = 3'b 111：DTvalue =（32+DTCFG [4:0]）x t_{DT}，t_{DT} = t_{DTS} * 16 此位只有在 TIMERx_CCHP0 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					MCH1COMCTL[3]	保留							MCH0COMCTL[3]

						保留								保留	
rw	rw					rw								rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCH1COMCEN	MCH1COMCTL[2:0]			MCH1COMSEN	MCH1COMFEN	MCH1MS[1:0]		MCH0COMCEN	MCH0COMCTL[2:0]			MCH0COMSEN	MCH0COMFEN	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。
29:25	保留	必须保持复位值。
24	MCH1COMCTL[3]	多模式通道 1 输出比较控制 请参考 MCH0COMCTL[2:0]描述。
23:17	保留	必须保持复位值。
16	MCH0COMCTL[3]	多模式通道 0 输出比较控制 请参考 MCH0COMCTL[2:0]描述。
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 输出比较控制 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	MCH1COMFEN	多模式通道1输出比较快速使能 参见MCH0COMSEN描述
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器输入被选中时（由 SYSCFG_TIMERxCFG2(x = 0, 7, 19)寄存器中的 TSCFG15[4:0]

位域选择）。

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 输出比较控制 当多模式通道 0 配置为输出模式，并且 MCH0MSEL[1:0] = 2’b00，MCH0COMCTL[3]和 MCH0COMCTL[2:0]位域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] = 2’b11 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。 0000：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。 0001：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。 0010：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。 0011：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。 0100：强制为低。强制 MO0CPRE 为低电平。 0101：强制为高。强制 MO0CPRE 为高电平。 0110：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。 0111：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。 1000：可延时的单脉冲模式0。MO0CPRE的输出情况类似与PWM模式0。在向上计数模式时，MO0CPRE先输出有效电平，当外部触发事件发生时，立即输出无效电平，当下一次更新事件发生时，再变成有效电平；在向下计数模式时，MO0CPRE先输出无效电平，当外部触发事件发生时，立即输出有效电平，当下一次更新事件发生时，再变成无效电平。 1001：可延时的单脉冲模式1。MO0CPRE的输出情况类似与PWM模式1。在向上计数模式时，MO0CPRE先输出无效电平，当外部触发事件发生时，立即输出有效电平，

		当下一次更新事件发生时，再变成无效电平；在向下计数模式时，MO0CPRE先输出有效电平，当外部触发事件发生时，立即输出无效电平，当下一次更新事件发生时，再变成有效电平。 1010~1111：保留 在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或者输出比较模式中从时基模式切换到 PWM 模式时，MO0CPRE 电平才改变。 当 CH0 和 MCH0 输出互补时，该位域预装载。若 CCSE = 1，则该位域只在通道换相事件发生时更新。 当 TIMERx_CCHP0 寄存器的 PROT[1:0]=11 且 MCH0MS = 000（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_CCHP0 寄存器的 PROT [1:0]=11 且 CH0MS = 00 时此位不能被改变。
2	MCH0COMFEN	多模式通道0输出比较快速使能 当该位为1时，如果通道配置为PWM模式0或者PWM模式1，会加快捕获 / 比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，MCH0_O被设置为比较电平而与比较结果无关。 0：多模式通道0输出比较快速禁能。当触发器的输入有一个有效沿时，激活MCH0_O输出的最小延时为5个时钟周期 1：多模式通道0输出比较快速使能。当触发器的输入有一个有效沿时，激活MCH0_O输出的最小延时为3个时钟周期
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（由 SYSCFG_TIMERxCFG2(x = 0, 7, 19)寄存器中的 TSCFG15[4:0]位域选择）。 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

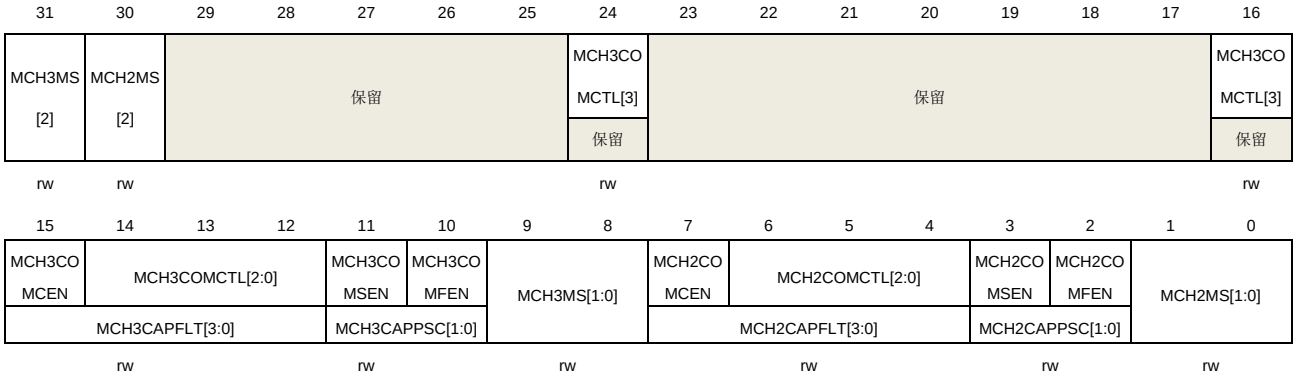
1:0	MCH0MS[1:0]	多模式通道 0 模式选择 与输出比较模式相同

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。



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。
29:25	保留	必须保持复位值。
24	MCH3COMCTL[3]	多模式通道 3 输出比较控制 请参考 MCH2COMCTL[2:0]描述。
23:17	保留	必须保持复位值。
16	MCH2COMCTL[3]	多模式通道 2 输出比较控制 请参考 MCH2COMCTL[2:0]描述。
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 输出比较控制 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	MCH3COMFEN	多模式通道3输出比较快速使能 参见 MCH2COMSEN 描述
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器输入被选中时（由 SYSCFG_TIMERxCFG2(x = 0, 7, 19)寄存器中的 TSCFG15[4:0]位域选择）。 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 输出比较控制 当多模式通道 2 配置为输出模式，并且 MCH2MSEL[1:0] = 2’b00，MCH2COMCTL[3]和 MCH2COMCTL[2:0]位域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] = 2’b11 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。 0000：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。 0001：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。 0010：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。 0011：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。 0100：强制为低。强制 MO2CPRE 为低电平。 0101：强制为高。强制 MO2CPRE 为高电平。 0110：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。 0111：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。 1000：可延时的单脉冲模式0。MO0CPRE的输出情况类似与PWM模式0。在向上计数模式时，MO0CPRE先输出有效电平，当外部触发事件发生时，立即输出无效电平，当下一次更新事件发生时，再变成有效电平；在向下计数模式时，MO0CPRE先输出无效电平，当外部触发事件发生时，立即输出有效电平，当下一次更新事件发生时，再变成无效电平。 1001：可延时的单脉冲模式1。MO0CPRE的输出情况类似与PWM模式1。在向上计数模式时，MO0CPRE先输出无效电平，当外部触发事件发生时，立即输出有效电平，当下一次更新事件发生时，再变成无效电平；在向下计数模式时，MO0CPRE先输出有效电平，当外部触发事件发生时，立即输出无效电平，当下一次更新事件发生时，再变成有效电平。 1010~1111：保留 在 PWM 模式 0 或 PWM 模式 1 中，只有当比较结果改变了或者输出比较模式中从时基模式切换到 PWM 模式时，MO2CPRE 电平才改变。 当 CH0 和 MCH0 输出互补时，该位域预装载。若 CCSE =1，则该位域只在通道换相事件发生时更新。 当 TIMERx_CCHP0 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_CCHP0 寄存器的 PROT [1:0]=11 且 MCH2MS =00 时此位不能被改变。
2	MCH2COMFEN	多模式通道2输出比较快速使能 当该位为1时，如果通道配置为PWM模式0或者PWM模式1，会加快捕获 / 比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，MCH2_O被设置为比较电平而与比较结果无关。 0：多模式通道2输出比较快速禁能。当触发器的输入有一个有效沿时，激活MCH2_O输出的最小延时为5个时钟周期 1：多模式通道 2 输出比较快速使能。当触发器的输入有一个有效沿时，激活 MCH2_O 输出的最小延时为 3 个时钟周期
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（由 SYSCFG_TIMERxCFG2(x = 0, 7, 19)寄存器中的 TSCFG15[4:0]位域选择）。

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] = 2'b00，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_CCHP0 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															

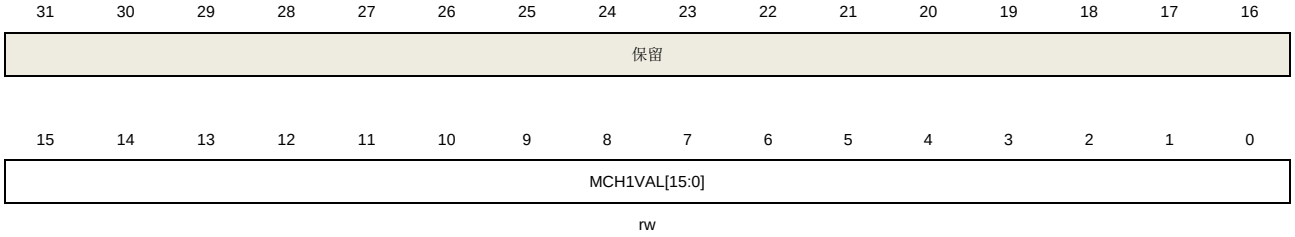
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	MCH0VAL[15:0]	多模式通道 0 的捕获 比较值 当多模式通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值，且本寄存器为只读。 当多模式通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

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。



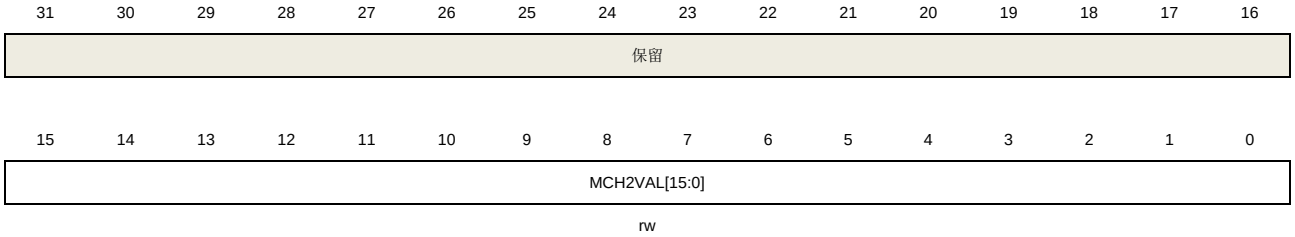
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	MCH1VAL[15:0]	多模式通道 1 的捕获或比较值 当多模式通道 1 配置为输入模式时，这些位决定了上次捕获事件的计数器值，且本寄存器为只读。 当多模式通道 1 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

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。



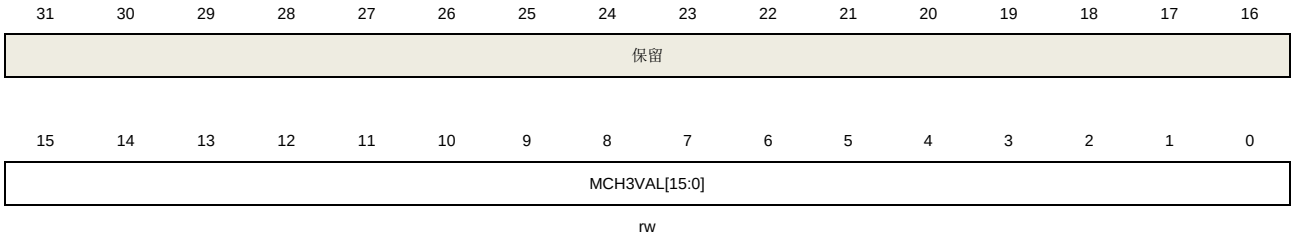
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	MCH2VAL[15:0]	多模式通道 2 的捕获或比较值 当多模式通道 2 配置为输入模式时，这些位决定了上次捕获事件的计数器值，且本寄存器为只读。 当多模式通道 2 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

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。



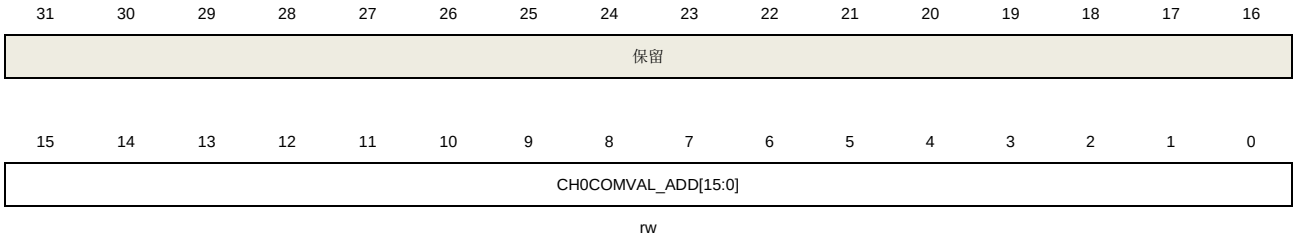
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	MCH3VAL[15:0]	多模式通道 3 的捕获或比较值 当多模式通道 3 配置为输入模式时，这些位决定了上次捕获事件的计数器值，且本寄存器为只读。 当多模式通道 3 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

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。



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
CH1COMVAL_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CH0CPWMEN = 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
CH2COMVAL_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CH0CPWMEN = 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
CH3COMVAL_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CH0CPWMEN = 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]		DECDISD EN	DECJDEN	保留	
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
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：保留 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：保留 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：保留 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：保留 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	DECDISDEN	正交译码器信号断线检测使能 0：正交译码器信号断线检测禁能 1：正交译码器信号断线检测使能
18	DECJDEN	正交译码器信号跳变（两个信号同时发生跳变沿）检测使能 0：正交译码器信号跳变（两个信号同时发生跳变沿）检测禁能 1：正交译码器信号跳变（两个信号同时发生跳变沿）检测使能
17:16	保留	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

独立互补通道保护寄存器 0（TIMERx_FCCHP0）

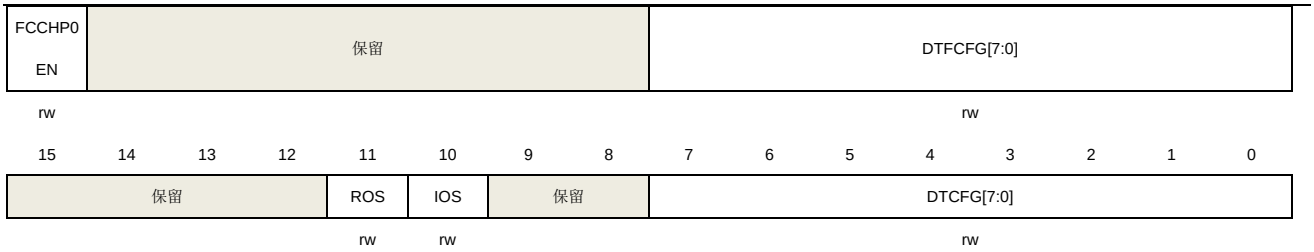
地址偏移：0x7C

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



位/位域	名称	描述
31	FCCHP0EN	独立互补通道寄存器 0 使能位 0：TIMERx_CCHP0 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_CCHP0 寄存器的 PROT [1:0]=00 时才可修改。
30:24	保留	必须保持复位值。
23:16	DTFCFG[7:0]	下降沿死区时间控制 该位域控制 OxCPRE 信号下降沿的死区时间值，该死区时间在输出转换之前插入。 DTCFG 值与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’b 10x：DTvalue =（64+DTCFG [5:0]）x t_{DT}，t_{DT} = t_{DTS} * 2 DTCFG [7:5] = 3’b 110：DTvalue =（32+DTCFG [4:0]）x t_{DT}，t_{DT}= t_{DTS}* 8 DTCFG [7:5] = 3’b 111：DTvalue =（32+DTCFG [4:0]）x t_{DT}，t_{DT} = t_{DTS} * 16 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0]=00 时才可修改。
15:12	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_CCHP0 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_CCHP0 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’b 10x：DTvalue =（64+DTCFG [5:0]）x t_{DT}，t_{DT} = t_{DTS} * 2

DTCFG [7:5] = 3’b 110：DTvalue =（32+DTCFG [4:0]）x t_{DT}，t_{DT}= t_{DTS} * 8

DTCFG [7:5] = 3’b 111：DTvalue =（32+DTCFG [4:0]）x t_{DT}，t_{DT} = t_{DTS} * 16

此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0]=00 时才可修改。

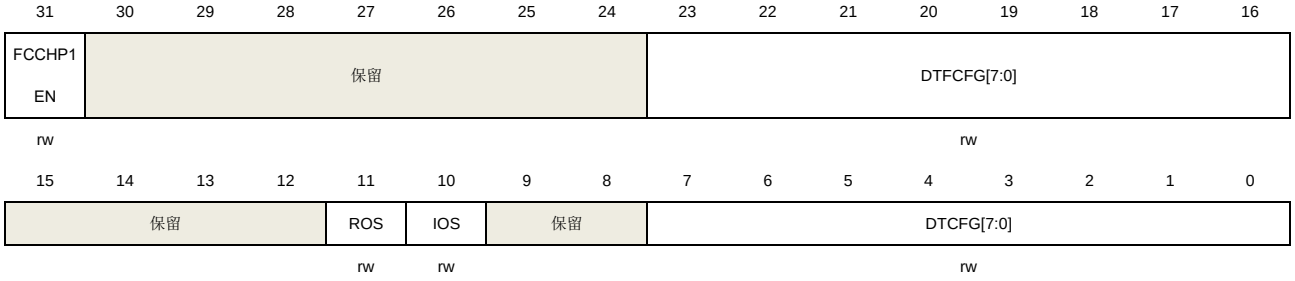
独立互补通道保护寄存器 1（TIMERx_FCCHP1）

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。



位/位域	名称	描述
31	FCCHP1EN	独立互补通道寄存器 1 使能位 0：TIMERx_CCHP0 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_CCHP0 寄存器的 PROT [1:0]=00 时才可修改。
30:24	保留	必须保持复位值。
23:16	DTFCFG[7:0]	下降沿死区时间控制 该位域控制 OxCPRE 信号下降沿的死区时间值，该死区时间在输出转换之前插入。 DTCFG 值与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’b 10x：DTvalue =（64+DTCFG [5:0]）x t_{DT}，t_{DT} = t_{DTS} * 2 DTCFG [7:5] = 3’b 110：DTvalue =（32+DTCFG [4:0]）x t_{DT}，t_{DT}= t_{DTS} * 8 DTCFG [7:5] = 3’b 111：DTvalue =（32+DTCFG [4:0]）x t_{DT}，t_{DT} = t_{DTS} * 16 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0]=00 时才可修改。
15:12	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_CCHP0 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_CCHP0 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_CCHP0 寄存器的 PROT [1:0]=00 时才可修改。

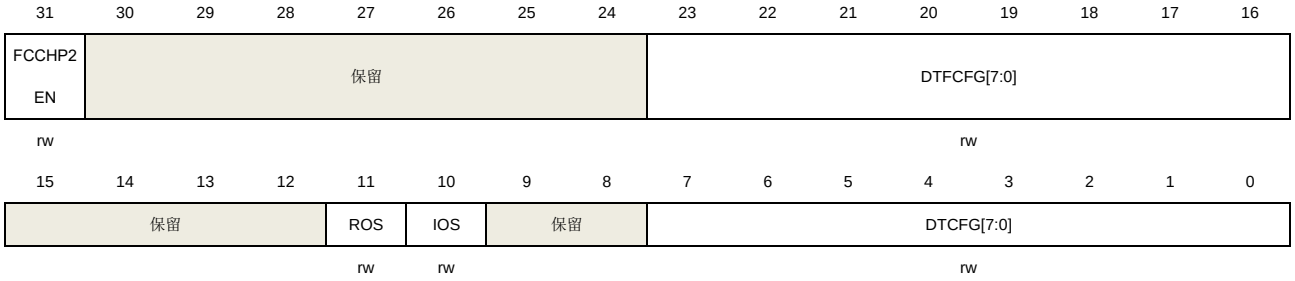
独立互补通道保护寄存器 2（TIMERx_FCCHP2）

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。



位/位域	名称	描述
31	FCCHP2EN	独立互补通道寄存器 2 使能位

		0：TIMERx_CCHP0 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_CCHP0 寄存器的 PROT [1:0]=00 时才可修改。
30:24	保留	必须保持复位值。
23:16	DTFCFG[7:0]	下降沿死区时间控制 该位域控制 OxCPRE 信号下降沿的死区时间值，该死区时间在输出转换之前插入。 DTCFG 值与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'b 10x：DTvalue =（64+DTCFG [5:0]）x t_{DT}，t_{DT} = t_{DTS} * 2 DTCFG [7:5] = 3'b 110：DTvalue =（32+DTCFG [4:0]）x t_{DT}，t_{DT}= t_{DTS} * 8 DTCFG [7:5] = 3'b 111：DTvalue =（32+DTCFG [4:0]）x t_{DT}，t_{DT} = t_{DTS} * 16 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0]=00 时才可修改。
15:12	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_CCHP0 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_CCHP0 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_CCHP0 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							DTFCFG[7:0]							
rw								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_CCHP0 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_CCHP0 寄存器的 PROT [1:0]=00 时才可修改。
30:24	保留	必须保持复位值。
23:16	DTFCFG[7:0]	下降沿死区时间控制 该位域控制 OxCPRE 信号下降沿的死区时间值，该死区时间在输出转换之前插入。 DTCFG 值与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'b 10x：DTvalue =（64+DTCFG [5:0]）x t_{DT}，t_{DT} = t_{DTS} * 2 DTCFG [7:5] = 3'b 110：DTvalue =（32+DTCFG [4:0]）x t_{DT}，t_{DT}= t_{DTS} * 8 DTCFG [7:5] = 3'b 111：DTvalue =（32+DTCFG [4:0]）x t_{DT}，t_{DT} = t_{DTS} * 16 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0]=00 时才可修改。
15:12	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_CCHP0 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_CCHP0 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_CCHP0 寄存器的 PROT [1:0]=00 时才可修改。

备用功能控制寄存器 0（TIMERx_AFCTL0）

地址偏移：0x8C
复位值：0x0000 0007

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			BRK0CMP3P	BRK0CMP2P	BRK0CMP1P	BRK0CMP0P	保留						BRK0IN2P	BRK0IN1P	BRK0IN0P
			rw	rw	rw	rw							rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRK0CMP6EN	BRK0CMP5EN	BRK0CMP4EN	BRK0CMP3EN	BRK0CMP2EN	BRK0CMP1EN	BRK0CMP0EN	BRK0HPDFEN	BRK0CMP7EN	保留				BRK0IN2EN	BRK0IN1EN	BRK0IN0EN
rw	rw	rw	rw	rw	rw	rw	rw	rw					rw	rw	rw

位/位域	名称	描述
31:29	保留	必须保持复位值。
28	BRK0CMP3P	BREAK0 CMP3输入极性 该位用于配置CMP3输入极性，具体极性是由该位和BRK0P位共同确定。 0：CMP3输入信号不反相（BRK0P = 0，输入信号低有效；BRK0P = 1，输入信号高有效） 1：CMP3输入信号反相（BRK0P = 0，输入信号高有效；BRK0P = 1，输入信号低有效） 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。

27	BRK0CMP2P	BREAK0 CMP2输入极性 该位用于配置CMP2输入极性，具体极性是由该位和BRK0P位共同确定。 0：CMP2输入信号不反相（BRK0P = 0，输入信号低有效；BRK0P = 1，输入信号高有效） 1：CMP2输入信号反相（BRK0P = 0，输入信号高有效；BRK0P = 1，输入信号低有效） 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
26	BRK0CMP1P	BREAK0 CMP1输入极性 该位用于配置CMP1输入极性，具体极性是由该位和BRK0P位共同确定。 0：CMP1输入信号不反相（BRK0P = 0，输入信号低有效；BRK0P = 1，输入信号高有效） 1：CMP1输入信号反相（BRK0P = 0，输入信号高有效；BRK0P = 1，输入信号低有效） 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
25	BRK0CMP0P	BREAK0 CMP0输入极性 0：CMP0输入信号不反相（BRK0P = 0，输入信号低有效；BRK0P = 1，输入信号高有效） 1：CMP输入信号反相（BRK0P = 0，输入信号高有效；BRK0P = 1，输入信号低有效） 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
24:19	保留	必须保持复位值。
18	BRK0IN2P	BREAK0 BRKIN2备用功能输入极性 该位用于配置BRKIN2输入极性，具体极性是由该位和BRK0P位共同确定。 0：BRKIN2输入信号不反相（BRK0P = 0，输入信号低有效；BRK0P = 1，输入信号高有效） 1：BRKIN2输入信号反相（BRK0P = 0，输入信号高有效；BRK0P = 1，输入信号低有效） 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
17	BRK0IN1P	BREAK0 BRKIN1备用功能输入极性 该位用于配置BRKIN1输入极性，具体极性是由该位和BRK0P位共同确定。 0：BRKIN1输入信号不反相（BRK0P =0，输入信号低有效；BRK0P =1，输入信号高有效） 1：BRKIN1输入信号反相（BRK0P =0，输入信号高有效；BRK0P =1，输入信号低有效） 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
16	BRK0IN0P	BREAK0 BRKIN0备用输入极性 该位用于配置BRKIN0输入极性，具体极性是由该位和BRK0P位共同确定。

		0：BRKIN0输入信号不反相（BRK0P =0，输入信号低有效；BRK0P =1，输入信号高有效） 1：BRKIN0输入信号反相（BRK0P =0，输入信号高有效；BRK0P =1，输入信号低有效） 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
15	BRK0CMP6EN	BREAK0 CMP6输入使能 0：CMP6输入禁能 1：CMP6输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
14	BRK0CMP5EN	BREAK0 CMP5输入使能 0：CMP5输入禁能 1：CMP5输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
13	BRK0CMP4EN	BREAK0 CMP4输入使能 0：CMP4输入禁能 1：CMP4输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
12	BRK0CMP3EN	BREAK0 CMP3输入使能 0：CMP3输入禁能 1：CMP3输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
11	BRK0CMP2EN	BREAK0 CMP2输入使能 0：CMP2输入禁能 1：CMP2输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
10	BRK0CMP1EN	BREAK0 CMP1输入使能 0：CMP1输入禁能 1：CMP1输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
9	BRK0CMP0EN	BREAK0 CMP0输入使能 0：CMP0输入禁能 1：CMP0输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
8	BRK0HPDFEN	BREAK0 HPDF输入使能 0：HPDF输入禁能 1：HPDF输入使能

		注意：HPDF输入对于TIMER0（HPDF_BREAK[0]）、TIMER7（HPDF_BREAK[2]）和TIMER19（HPDF_BREAK[0]）是不同的。 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
7	BRK0CMP7EN	BREAK0 CMP7输入使能 0：CMP7输入禁能 1：CMP7输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
6:3	保留	必须保持复位值。
2	BRK0IN2EN	BREAK0 BRKIN2备用输入使能 0：BRKIN2输入禁能 1：BRKIN2输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
1	BRK0IN1EN	BREAK0 BRKIN1备用输入使能 0：BRKIN1输入禁能 1：BRKIN1输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
0	BRK0IN0EN	BREAK0 BRKIN0备用输入使能 0：BRKIN0输入禁能 1：BRKIN0输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。

备用功能控制寄存器 1（TIMERx_AFCTL1）

地址偏移：0x90

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			BRK1CMP3P	BRK1CMP2P	BRK1CMP1P	BRK1CMP0P	保留			OCRINSEL[2:0]			BRK1IN2P	BRK1IN1P	BRK1IN0P
			rw	rw	rw	rw				rw			rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRK1CMP6EN	BRK1CMP5EN	BRK1CMP4EN	BRK1CMP3EN	BRK1CMP2EN	BRK1CMP1EN	BRK1CMP0EN	BRK1HPDFEN	BRK1CMP7EN	保留				BRK1IN2EN	BRK1IN1EN	BRK1IN0EN
rw	rw	rw	rw	rw	rw	rw	rw	rw					rw	rw	rw

位/位域	名称	描述

31:29	保留	必须保持复位值。
28	BRK1CMP3P	BREAK1 CMP3输入极性 该位用于配置CMP3输入极性，具体极性是由该位和BRK1P位共同确定。 0：CMP3输入信号不反相（BRK1P = 0，输入信号低有效；BRK1P = 1，输入信号高有效） 1：CMP3输入信号反相（BRK1P = 0，输入信号高有效；BRK1P = 1，输入信号低有效） 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
27	BRK1CMP2P	BREAK1 CMP2输入极性 该位用于配置CMP2输入极性，具体极性是由该位和BRK1P位共同确定。 0：CMP2输入信号不反相（BRK1P = 0，输入信号低有效；BRK1P = 1，输入信号高有效） 1：CMP2输入信号反相（BRK1P = 0，输入信号高有效；BRK1P = 1，输入信号低有效） 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
26	BRK1CMP1P	BREAK1 CMP1输入极性 该位用于配置CMP1输入极性，具体极性是由该位和BRK1P位共同确定。 0：CMP1输入信号不反相（BRK1P = 0，输入信号低有效；BRK1P = 1，输入信号高有效） 1：CMP1输入信号反相（BRK1P = 0，输入信号高有效；BRK1P = 1，输入信号低有效） 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
25	BRK1CMP0P	BREAK1 CMP0输入极性 0：CMP0输入信号不反相（BRK1P = 0，输入信号低有效；BRK1P = 1，输入信号高有效） 1：CMP输入信号反相（BRK1P = 0，输入信号高有效；BRK1P = 1，输入信号低有效）此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
24:22	保留	必须保持复位值。
21:19	OCRINSEL[2:0]	OCPRE_CLR输入选择 000：OCPRE_CLR0 001：OCPRE_CLR1 … 111：OCPRE_CLR7

OCPRE_CLR 输入选择	TIMER0 / 7 / 19
OCPRE_CLR0	CMP0_OUT
OCPRE_CLR1	CMP1_OUT
OCPRE_CLR2	CMP2_OUT

OCPRE_CLR3	CMP3_OUT
OCPRE_CLR4	CMP4_OUT
OCPRE_CLR5	CMP5_OUT
OCPRE_CLR6	CMP6_OUT
OCPRE_CLR7	CMP7_OUT

此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。

18 BRK1IN2P

BREAK1 BRKIN2备用功能输入极性

该位用于配置BRKIN2输入极性，具体极性是由该位和BRK1P位共同确定。

0：BRKIN2输入信号不反相（BRK1P = 0，输入信号低有效；BRK1P = 1，输入信号高有效）

1：BRKIN2输入信号反相（BRK1P = 0，输入信号高有效；BRK1P = 1，输入信号低有效）

此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。

17 BRK1IN1P

BREAK1 BRKIN1备用功能输入极性

该位用于配置BRKIN1输入极性，具体极性是由该位和BRK1P位共同确定。

0：BRKIN1输入信号不反相（BRK1P = 0，输入信号低有效；BRK1P = 1，输入信号高有效）

1：BRKIN1输入信号反相（BRK1P = 0，输入信号高有效；BRK1P = 1，输入信号低有效）

此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。

16 BRK1IN0P

BREAK1 BRKIN0备用输入极性

该位用于配置BRKIN0输入极性，具体极性是由该位和BRK1P位共同确定。

0：BRKIN0输入信号不反相（BRK1P = 0，输入信号低有效；BRK1P = 1，输入信号高有效）

1：BRKIN0输入信号反相（BRK1P = 0，输入信号高有效；BRK1P = 1，输入信号低有效）

此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。

15 BRK1CMP6EN

BREAK1 CMP6输入使能

0：CMP6输入禁能

1：CMP6输入使能

此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。

14 BRK1CMP5EN

BREAK1 CMP5输入使能

0：CMP5输入禁能

1：CMP5输入使能

此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。

13 BRK1CMP4EN

BREAK1 CMP4输入使能

0：CMP4输入禁能

		1：CMP4输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
12	BRK1CMP3EN	BREAK1 CMP3输入使能 0：CMP3输入禁能 1：CMP3输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
11	BRK1CMP2EN	BREAK1 CMP2输入使能 0：CMP2输入禁能 1：CMP2输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
10	BRK1CMP1EN	BREAK1 CMP1输入使能 0：CMP1输入禁能 1：CMP1输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
9	BRK1CMP0EN	BREAK1 CMP0输入使能 0：CMP0输入禁能 1：CMP0输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
8	BRK1HPDFEN	BREAK1 HPDF输入使能 0：HPDF输入禁能 1：HPDF输入使能 **注意**：HPDF输入对于TIMER0（HPDF_BREAK[1]）、TIMER7（HPDF_BREAK[3]）和TIMER19（HPDF_BREAK[1]）是不同的。 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
7:3	BRK1CMP7EN	BREAK1 CMP7输入使能 0：CMP7输入禁能 1：CMP7输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
2	BRK1IN2EN	BREAK1 BRKIN2备用输入使能 0：BRKIN2输入禁能 1：BRKIN2输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
1	BRK1IN1EN	BREAK1 BRKIN1备用输入使能 0：BRKIN1输入禁能 1：BRKIN1输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。

0	BRK1IN0EN	BREAK1 BRKIN0备用输入使能 0：BRKIN0输入禁能 1：BRKIN0输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。

看门狗计数器周期寄存器（TIMERx_WDGPER）

地址偏移：0x94

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
WDGPER[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDGPER[15:0]															
rw															

位/位域	名称	描述
31:0	WDGPER[31:0]	看门狗计数器周期值 这些位用于配置两个看门狗的计数器周期。当看门狗计数器连续计数到该值时，计数器计数超时且中断标志位DECDISIF位置位。若DECDISIE=1，则相应的中断产生。 **注意**：该寄存器位仅用于正交译码器信号断线检测功能（DECDISDEN =1）使能。

重复计数寄存器 1（TIMERx_CREP1）

地址偏移：0x98

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CREP1[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CREP1[15:0]															
rw															

位/位域	名称	描述
31:0	CREP1[31:0]	重复计数器值 1 该位域为 32 位，只读。 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）。

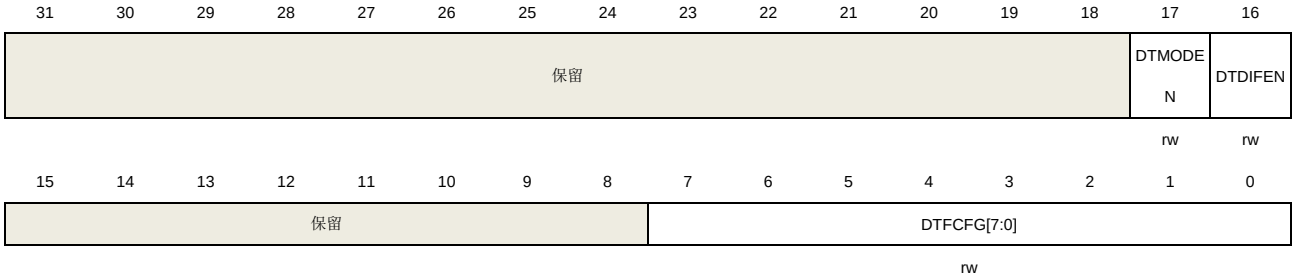
注意：当TIMERx_CFG寄存器中的CREPSEL=1时，使用该位域。

互补通道保护寄存器 1（TIMERx_CCHP1）

地址偏移：0x09C

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:18	保留	必须保持复位值。
17	DTMODEN	死区时间在线修改功能使能 0：死区时间在线修改功能禁能 1：死区时间在线修改功能使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。 **注意**：如果在CEN=1时将该位置1，则在上次更新事件之后写入的任何值（在DTCFG[7:0]位域和DTCFG[7:0]位域）都是无效的，只使用前一次的值。
16	DTDIFEN	死区时间不同配置使能 0：上升沿和下降沿的死区时间（在 TIMERx_CCHP1 或 TIMERx_FCCHPx 寄存器的 DTCFG[7:0]位域配置）相同。 1：上升沿的死区时间由 TIMERx_CCHP1 或 TIMERx_FCCHPx 寄存器的 DTCFG[7:0]位域配置；下降沿的死区时间由 TIMERx_CCHP1 或 TIMERx_FCCHPx 寄存器的 DTFCFG[7:0]位域配置。 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
15:8	保留	必须保持复位值。
7:0	DTFCFG[7:0]	下降沿死区时间控制 该位域控制 OxCPRE 信号下降沿的死区时间值，该死区时间在输出转换之前插入。 DTCFG 值与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’b 10x：DTvalue =（64+DTCFG [5:0]）x t_{DT}，t_{DT} = t_{DTS} * 2 DTCFG [7:5] = 3’b 110：DTvalue =（32+DTCFG [4:0]）x t_{DT}，t_{DT}= t_{DTS} * 8

DTCFG [7:5] = 3'b 111：DTvalue =（32+DTCFG [4:0]）x t_{DT}，t_{DT} = t_{DTS} * 16

此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0]=00 时才可修改

译码器控制寄存器（TIMERx_DECCTL）

地址偏移：0xA0

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留					OPPSC[2:0]			OPWID[7:0]							
					rw			rw							

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

位/位域	名称	描述
31:27	保留	必须保持复位值。
26:24	OPPSC[2:0]	输出脉冲预分频 该位域用于配置脉冲发生器的时钟分频系数。 t_{opclk} = ($2^{(OPPSC\ [2:0])}$) x t_{CK_TIMER}
23:16	OPWID[7:0]	输出脉冲宽度 该位域用于配置脉冲的宽度 t_{opw} = OPWID [7:0] x t_{opclk}
15:8	保留	必须保持复位值。
7:6	INDP[1:0]	索引定位 在正交译码器模式0~4中，该位域表示索引事件将在哪种类型的AB输入中复位计数器： 00：当AB相输入为00时，索引事件复位计数器 01：当AB相输入为01时，索引事件复位计数器 10：当AB相输入为10时，索引事件复位计数器 11：当AB相输入为11时，索引事件复位计数器 在译码器模式0~3中，该位域表示索引事件将在哪种类型的AB输入中复位计数器： x0：当时钟信号为低电平时，索引事件复位计数器 x1：当时钟信号为高电平时，索引事件复位计数器 **注意**：在译码器模式0~3中，INDP[1]无效。
5	FINDRST	第一个索引信号复位计数器

		0：所有的索引信号都可以复位计数器 1：只有第一个索引信号有效，可以复位计数器
4:3	保留	必须保持复位值。
2:1	INDRSTDIR[1:0]	索引信号复位计数器时的计数方向 该位域用于配置计数器的计数方向和索引 00：当计数器向上和向下计数时，索引信号复位计数器 01：仅在计数器向上计数时，索引信号复位计数器 10：仅在计数器向下计数时，索引信号复位计数器 11：保留 注意：仅在INDRSTEN = 0时，可以修改INDRSTDIR[1:0]位域。
0	INDRSTEN	索引信号复位使能 0：索引信号复位计数器禁能 1：索引信号复位计数器使能

计数器初始控制寄存器（TIMERx_CINITCTL）

地址偏移：0xA4

复位值：0x0000 0000

该寄存器只能按字（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													SWSYNCG	CINITDIR	CINITVEN
													w	rw	rw

位/位域	名称	描述
31:3	保留	必须保持复位值。
2	SWSYNCG	软件同步事件产生 该位由软件置1，硬件自动清除。读这个位的值总是返回0。 0：没有影响 1：软件同步事件产生
1	CINITDIR	计数器初始计数方向 0：同步事件发生时，计数器向上计数 1：同步事件发生时，计数器向下计数 该位表示同步事件发生后计数器的初始计数方向，而计数器的计数初始值从TIMERx_CINITV寄存器中加载。

注意：该位仅用于CAM[1:0] ≠ 00时。

0	CINITVEN	计数器初始值寄存器使能 0：计数器初始值寄存器禁能。计数器寄存器不能从计数器初始值寄存器加载计数初始值。 1：计数器初始值寄存器使能。当一个同步事件（或通过设置SWSYNCG位生成的软同步事件）发生时，计数器寄存器可以从计数器初始值寄存器中加载计数初始值。

计数器初始值寄存器（TIMERx_CINITV）

地址偏移：0xA8

复位值：0x0000 0000

该寄存器只能按字（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
CINITVAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CINITVAL[15:0]	计数器初始值 该位域用于表示计数器的初始值。 CINITVEN位为0时，计数器的寄存器不能加载在CINITVAL[15:0]位域中设置的初始值。 CINITVEN位为1时，当同步事件发生时，计数器的寄存器可以加载在CINITVAL[15:0]位域中设置的初始值。

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。 6'b000000：传输1次 6'b000001：传输2次 … 6'b111000：传输 57 次
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。 6'b0_0000：TIMERx_CTL0 6'b0_0001：TIMERx_CTL1 … 6'b111000：TIMERx_CINITV 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
DMATB[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMATB[15:0]															
rw															

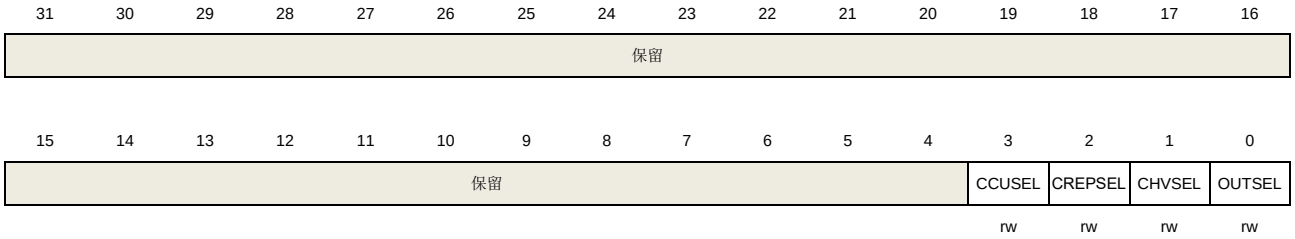
位/位域	名称	描述
31:0	DMATB [31:0]	DMA 发送缓冲 对这个寄存器的读或写，（起始地址+传输次数*4）地址范围内的寄存器会被访问 传输次数由硬件计算，范围为 0 到 DMATC。

配置寄存器（TIMERx_CFG）

地址偏移：0xFC

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:4	保留	必须保持复位值。
3	CCUSEL	换相控制影子寄存器更新选择 只有当CCUC[2:0]位域配置为100,101和110时，该位才有效。 0：当计数器产生一个上溢/下溢事件时，影子寄存器才更新 1：当重复计数器值为0，且计数器产生一个上溢/下溢事件时，影子寄存器才更新
2	CREPSEL	计数器重复寄存器选择 该位用于选择重复计数寄存器。 0：更新事件的速率由TIMERx_CREP0寄存器确定 1：更新事件的速率由 TIMERx_CREP1 寄存器确定
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

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

23.2.1. 简介

通用定时器 L0（TIMER1 / 2 / 3 / 4）是 4 通道定时器，支持输入捕获，输出比较，产生 PWM 信号控制电机和电源管理。通用定时器 L0 的计数器是 16 位或 32 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。

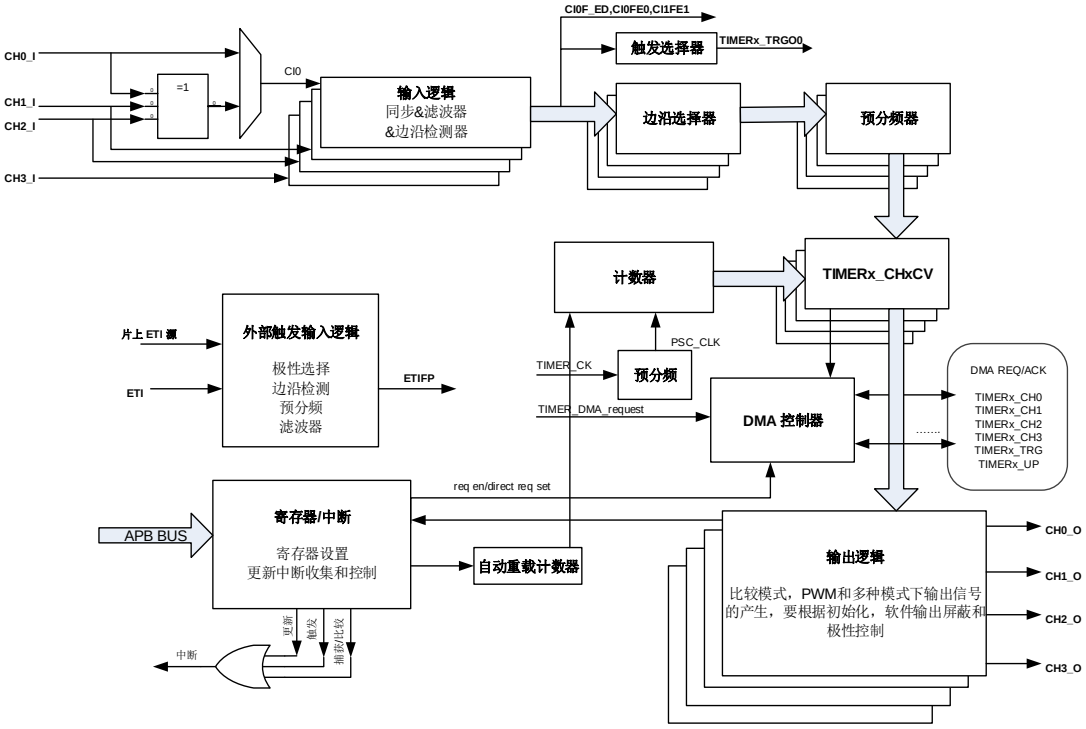
23.2.2. 主要特征

- 总通道数：4；
- 计数器宽度：16位（TIMER2 / 3）和32位（TIMER1 / 4）；
- 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。

23.2.3. 结构框图

*图 23-64. 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 23-64. 通用定时器 L0 结构框图



23.2.4. 功能描述

时钟源选择

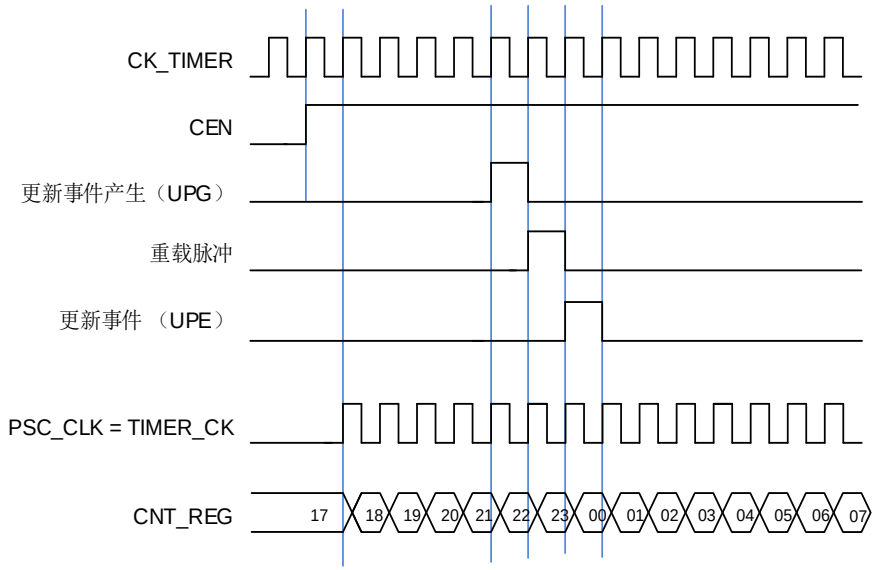
通用定时器 L0 可以由内部时钟源 CK_TIMER 或者由 SYSCFG_TIMERxCFG(x = 1..4)寄存器中的 TSCFGy[4:0] (y = 0…9,15)位域控制的复用时钟源驱动。

- 当SYSCFG_TIMERxCFG(x = 1..4)寄存器中的TSCFGy[4:0]=5'b00000(y = 0…15)时，定时器选择内部时钟源（连接到RCU模块的CK_TIMER）

如果 SYSCFG_TIMERxCFG(x = 1..4)寄存器中的 TSCFGy[4:0] =5'b00000(y = 0…15)，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 SYSCFG_TIMERxCFG(x = 1..4)寄存器中的 TSCFGy[4:0] (y = 0…2, 6, 8, 9)位域设置为非零值，预分频器被其他时钟源驱动，具体在下文说明。当 TSCFGy[4:0] (y = 3, 4, 5, 7)被设置为非零值时，计数器预分频器时钟源由内部时钟 TIMER_CK 驱动。

图 23-65. 内部时钟分频为 1 时正常模式下的控制电路



- TSCFG6[4:0] !=5'b00000（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 CI0 / 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 TSCFG6[4:0]为 0x5 ~ 0x7 来选择。

计数器预分频器也可以在内部触发信号 ITI0 ~ 10 / ITI14 的上升沿计数。这种模式可以通过设置 TSCFG6[4:0]为 0x1 ~ 0x4，0x9 ~ 0xF 和 0x13 来选择。

- 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

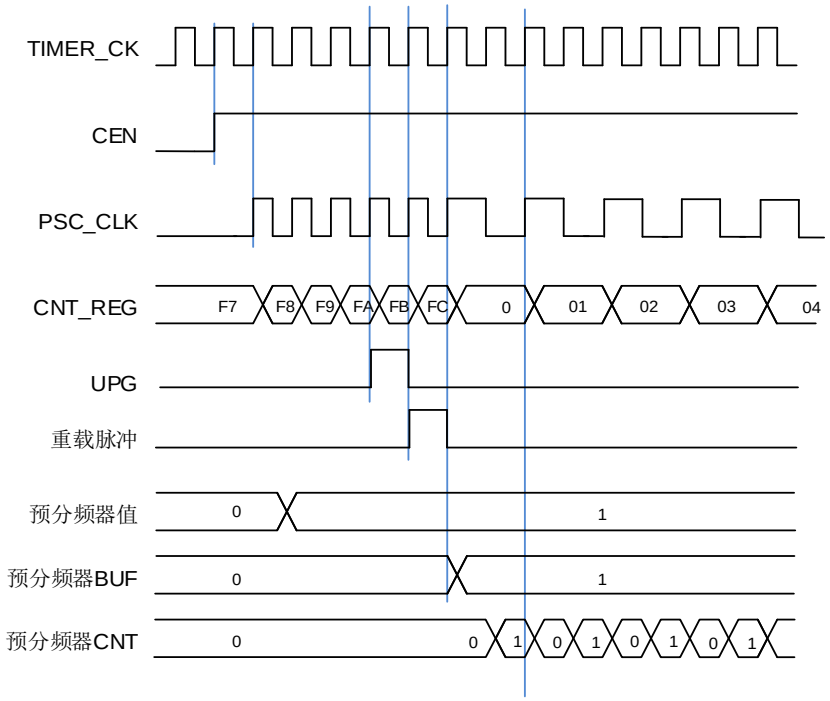
计数器预分频器可以在外部引脚 ETI 的每个上升沿或下降沿计数。这种模式可以通过设置 TIMERx_SMCFG 寄存器中的 SMC1 位为 1 来选择。另一种选择 ETI 信号作为时钟源方式是，设置 TSCFG6[4:0]为 0x08。注意 ETI 信号是通过数字滤波器采样 ETI 引脚得到的。如果选择 ETI 信号为时钟源，触发控制器包括边沿监测电路将在每个 ETI 信号上升沿产生一个时钟脉冲来为计数器预分频器提供时钟。

注意：ETI 信号可以从外部 ETI 引脚输入，也可由片上外设提供，具体情况可以参考 ***TIMER1_ETI 触发选择寄存器（TRIGSEL_TIMER1ETI）***、***TIMER2_ETI 触发选择寄存器（TRIGSEL_TIMER2ETI）***、***TIMER3_ETI 触发选择寄存器（TRIGSEL_TIMER3ETI）***和***TIMER4_ETI 触发选择寄存器（TRIGSEL_TIMER4ETI）***模块。

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 23-66. 当预分频器的参数从 1 变到 2 时，计数器的时序图



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，所有的寄存器（自动重载寄存器，预分频寄存器）都将被更新。

图 23-67. 向上计数时序图,PSC=0 / 2 和 *图 23-68. 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 23-67. 向上计数时序图，PSC=0 / 2

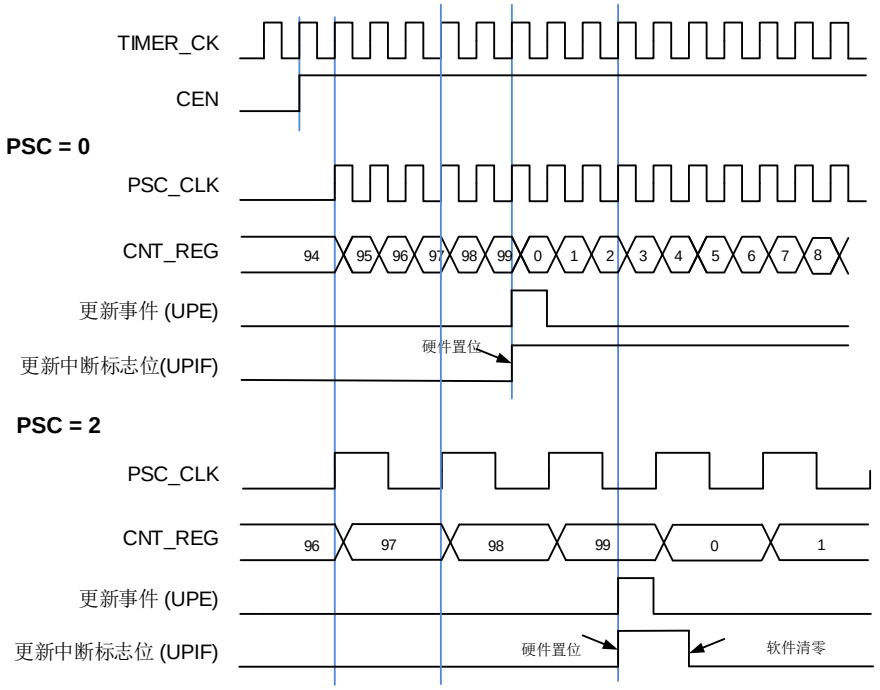
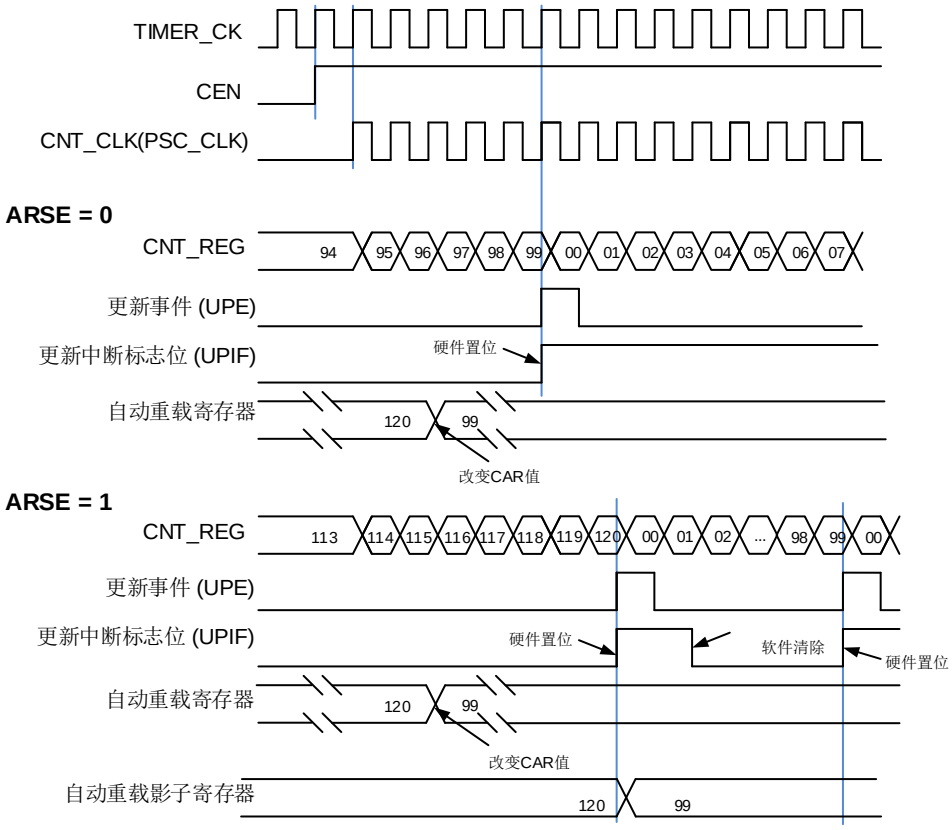


图 23-68. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



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，所有的寄存器（自动重载寄存器，预分频寄存器）都将被更新。

图 23-69. 向下计数时序图，PSC=0 / 2 和 *图 23-70. 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 23-69. 向下计数时序图，PSC=0 / 2

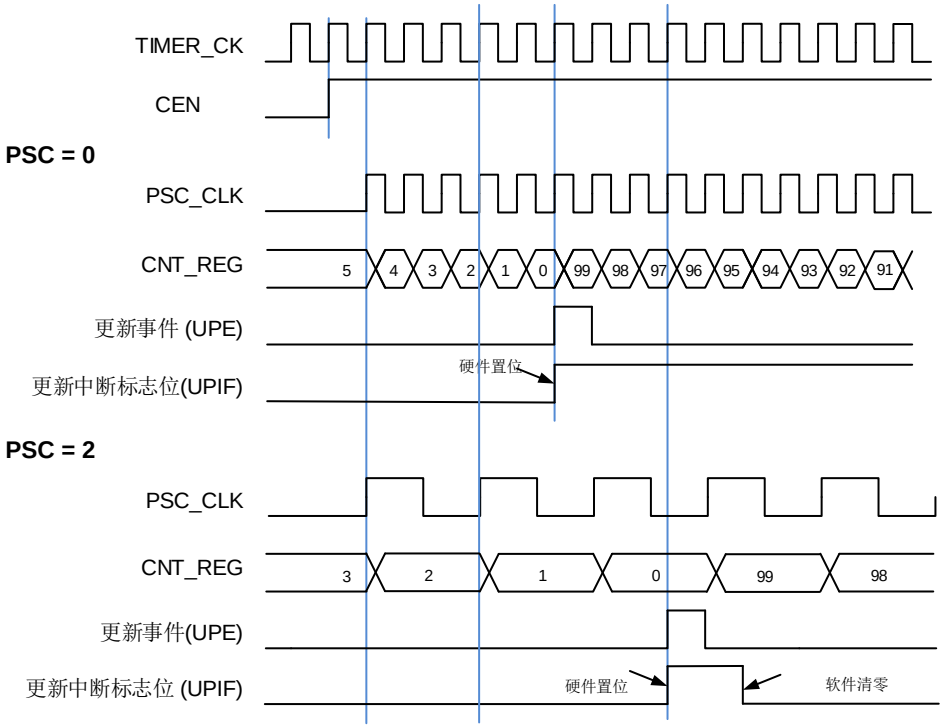
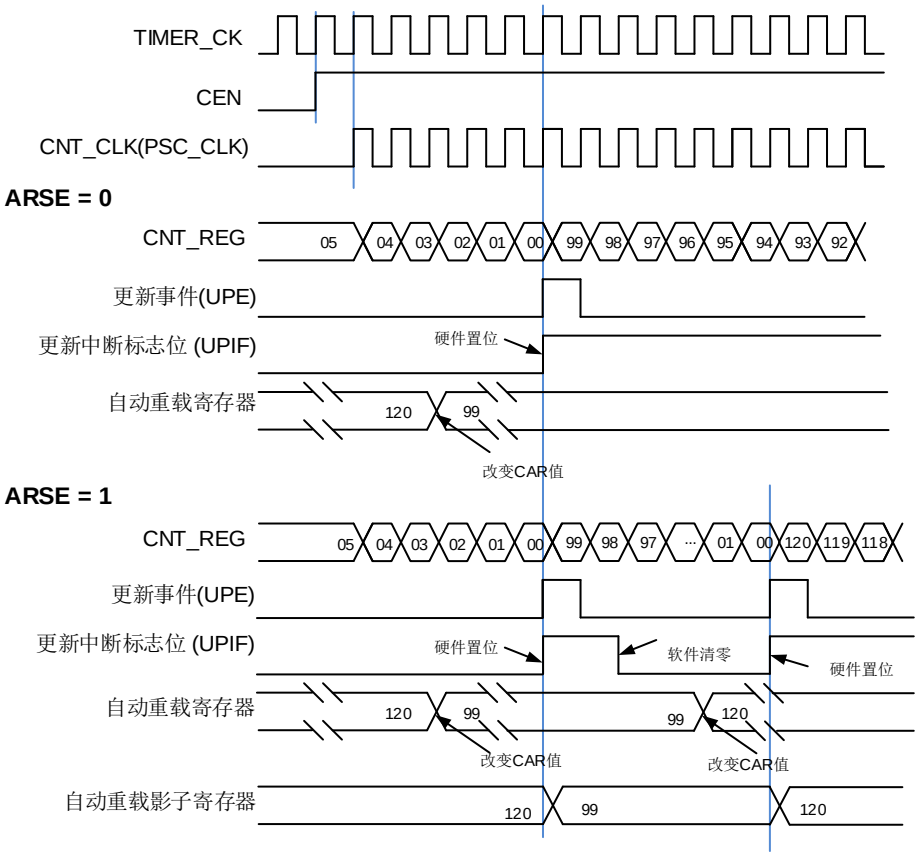


图 23-70. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值



中央对齐模式

在中央对齐模式下，计数器交替的从 0 开始向上计数到自动加载值，然后再向下计数到 0。向上计数模式中，定时器模块在计数器计数到（TIMERx_CAR-1）产生一个上溢事件；向下计数模式中，定时器模块在计数器计数到 1 时产生一个下溢事件。在中央计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 只读，表明了的计数方向。计数方向被硬件自动更新。

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。

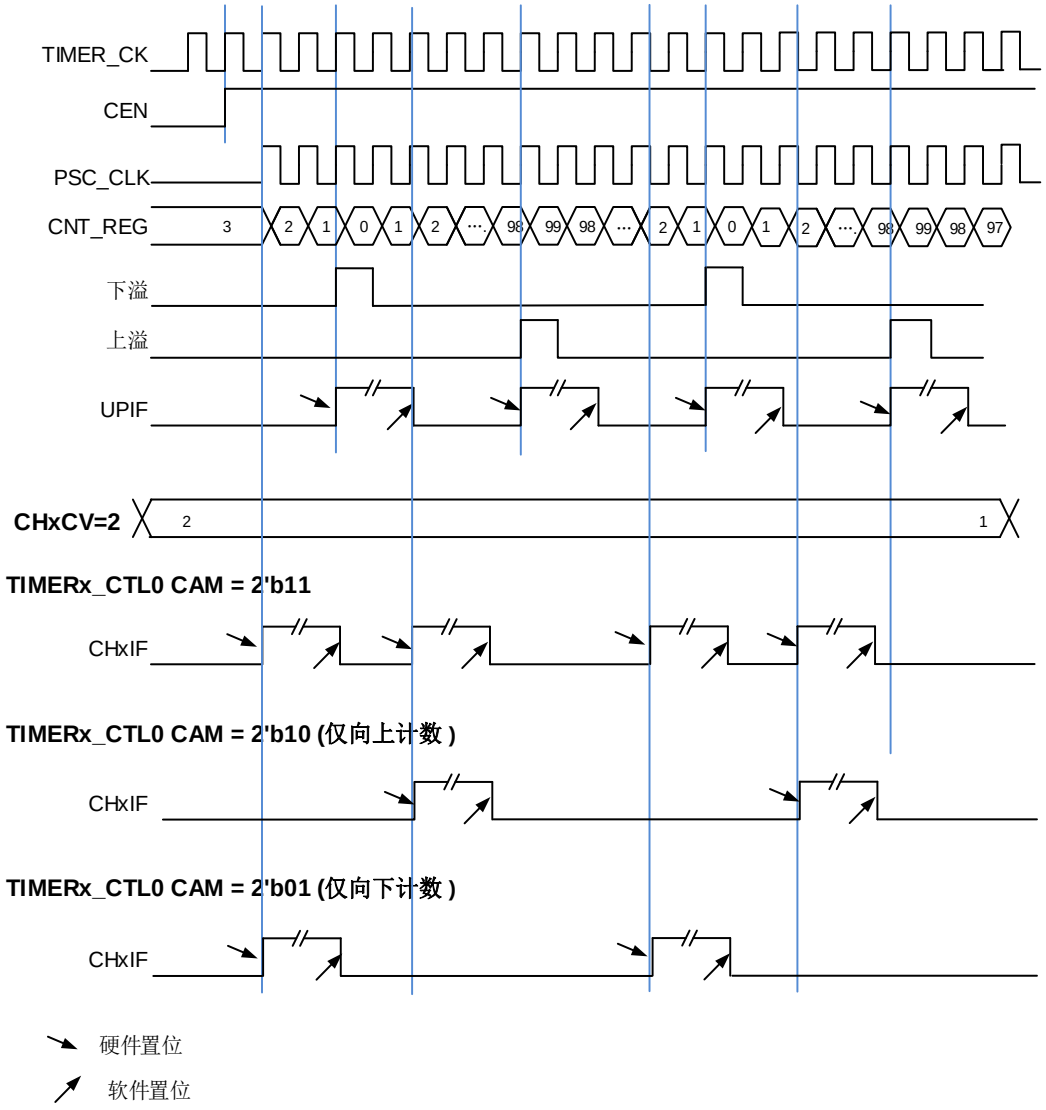
上溢或者下溢时，TIMERx_INTF 寄存器中的 UPIF 位都会被置 1，然而 CHxIF 位置 1 与 TIMERx_CTL0 寄存器中 CAM 的值有关。具体细节参考*图 23-71. 中央计数模式计数器时序图*

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，所有的寄存器（自动重载寄存器，预分频寄存器）都将被更新。

*图 23-71. 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 23-71. 中央计数模式计数器时序图



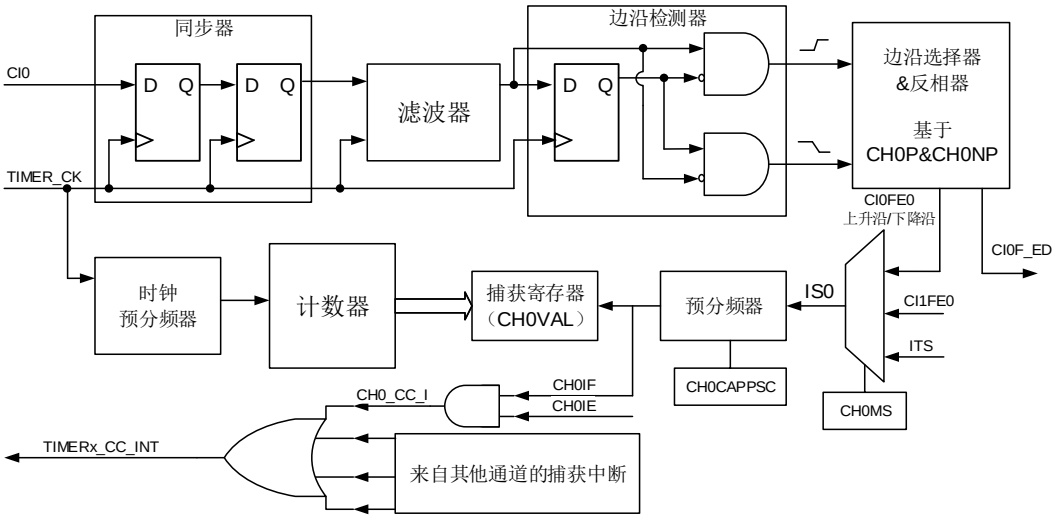
捕获 / 比较通道

通用定时器 L0 拥有四个独立的通道用于捕获输入或比较输出是否匹配。每个通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

■ 输入捕获模式

捕获模式允许通道测量一个波形时序，频率，周期，占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，TIMERx_CHxCV 寄存器会捕获计数器当前的值，同时 CHxIF 位被置 1，如果 CHxIE = 1 则产生通道中断。

图 23-72. 输入捕获逻辑



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。

输入捕获模式也可用来测量 TIMERx_CHx 引脚上信号的脉冲波宽度。例如，一个 PWM 波连接到 CI0。配置 TIMERx_CHCTL0 寄存器中 CH0MS 为 2'b01，选择通道 0 的捕获信号为 CI0 并设置上升沿捕获。配置 TIMERx_CHCTL0 寄存器中 CH1MS 为 2'b10，选择通道 1 捕获信号为 CI0 并

设置下降沿捕获。计数器配置为复位模式，在通道 0 的上升沿复位。TIMERX_CH0CV 寄存器测量 PWM 的周期值，TIMERx_CH1CV 寄存器测量 PWM 占空比值。

■ 输出比较模式

图 23-73. 输出比较逻辑（x = 0, 1, 2, 3）

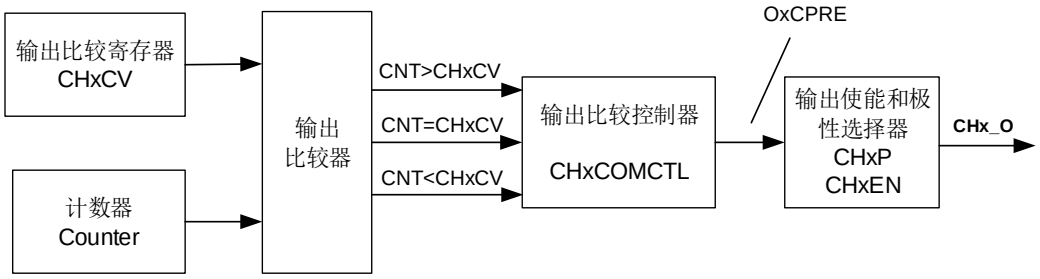


*图23-73. 输出比较逻辑（x = 0, 1, 2, 3）*给出了输出比较的逻辑电路。通道输出信号CHx_O与OxCPRE信号（详情请见*通道输出准备信号*）的关系描述如下：OxCPRE信号高电平有效，CHx_O的输出情况与OxCPRE信号，CHxP位和CHxEN位有关（具体情况请见TIMERx_CHCTL2寄存器中的描述）。例如，当设置CHxP=0（CHx_O高电平有效，与OxCPRE输出极性相同）、CHxEN=1（CHx_O输出使能）时：

若OxCPRE输出有效（高）电平，则CHx_O输出有效（高）电平；

若OxCPRE输出无效（低）电平，则CHx_O输出无效（低）电平。

在输出比较模式，TIMERx 可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的 CHxCV 寄存器与计数器的值匹配时，根据 CHxCOMCTL 的配置，这个通道的输出可以被置高电平，被置低电平或者翻转。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CHxEN使能输出。

第三步：通过CHxIE/CxCDE位配置中断/DMA请求使能。

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

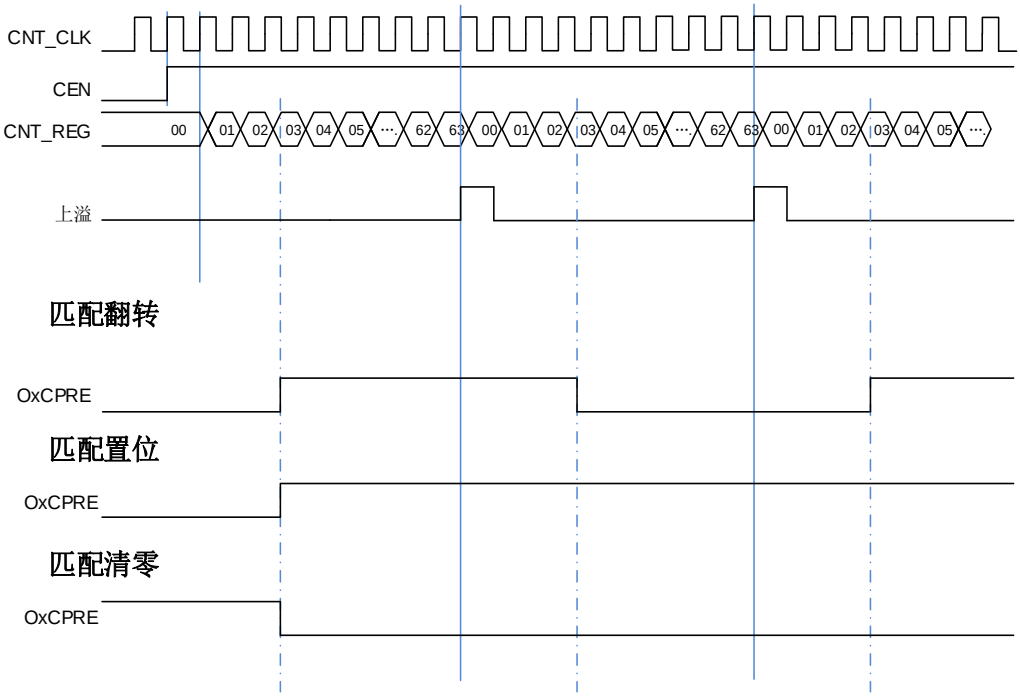
CHxVAL可以在运行时根据你所期望的波形而改变。

第五步：设置CEN位使能定时器。

*图 23-74. 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 23-74. 三种输出比较模式



PWM 模式

在 PWM 输出模式下（PWM 模式 0 是配置 CHxCOMCTL 为 4'b0110，PWM 模式 1 是配置 CHxCOMCTL 为 4'b0111），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 23-75. 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 23-76. 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 =4'b0110），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；PWM1模式下（CHxCOMCTL =4'b0111），如果TIMERx_CHxCV寄存器的值大于TIMERx_CAR寄存器的值，通道输出一直为无效电平。

图 23-75. EAPWM 时序图

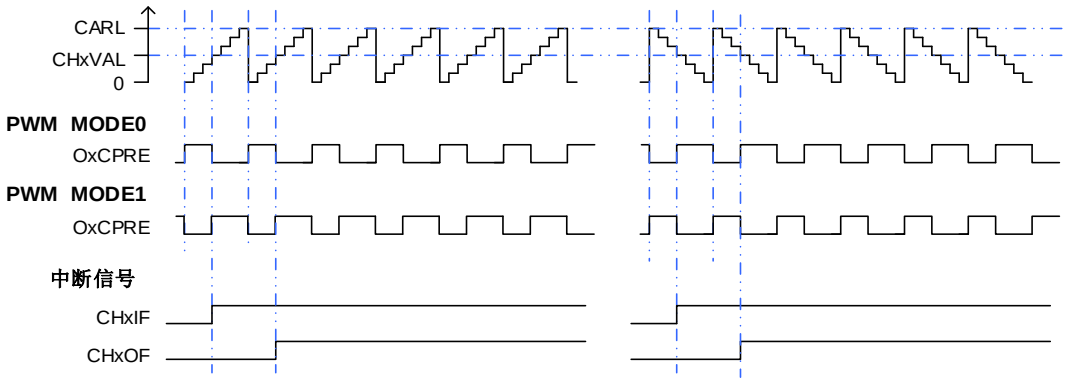
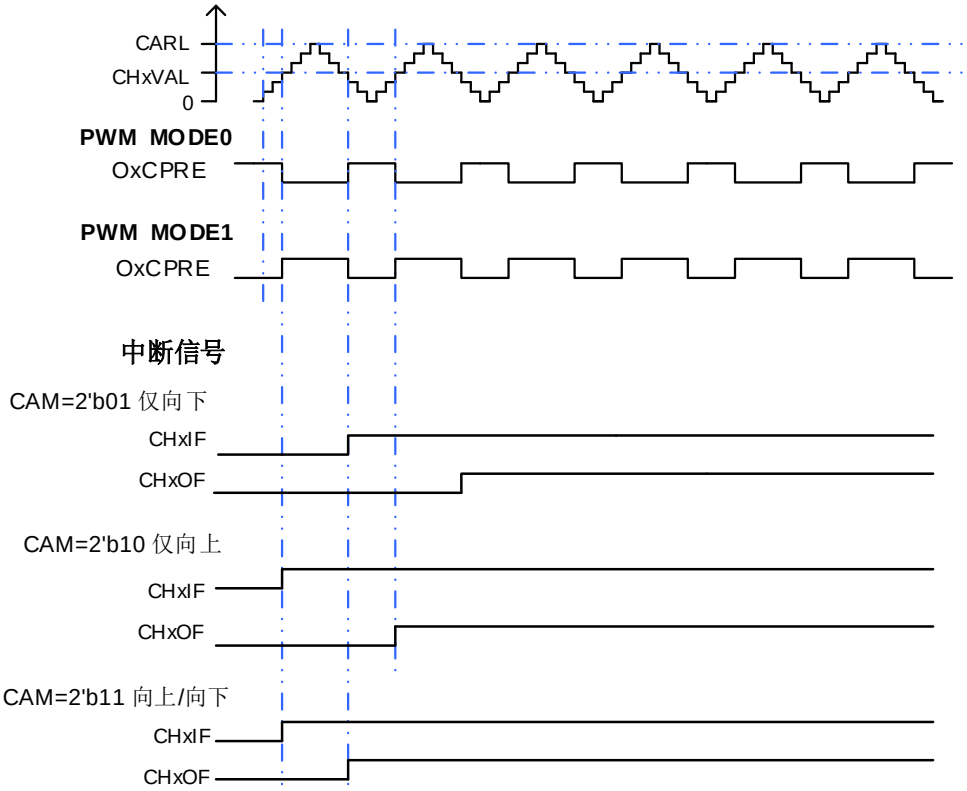


图 23-76. CAPWM 时序图



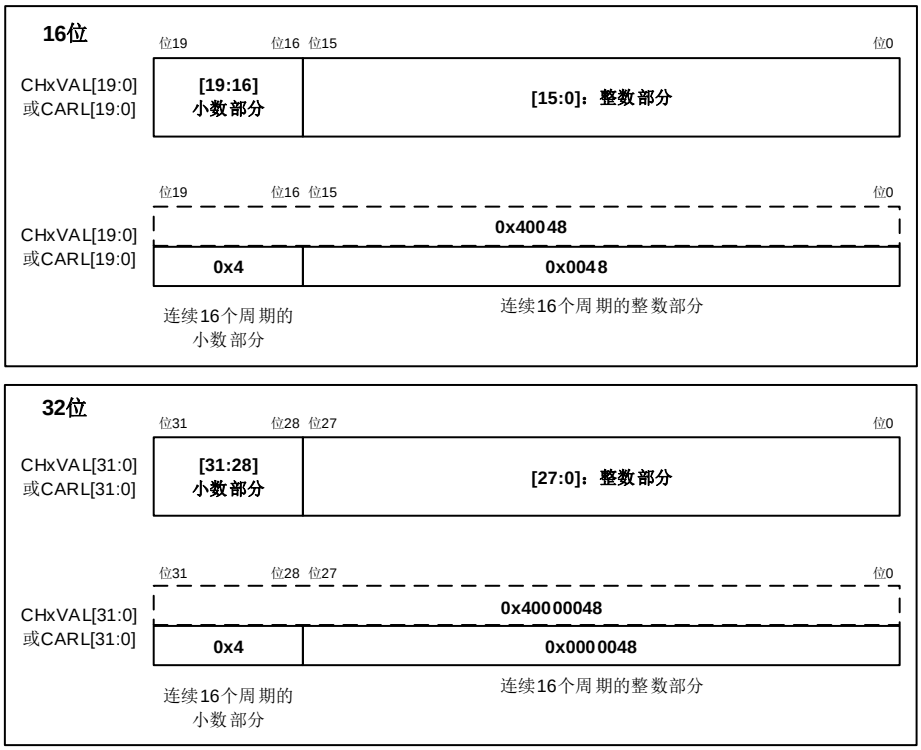
微调模式

通过配置TIMERx_CTL0寄存器中的ADMEN位为1，可以使能微调模式。该模式可以提高输出PWM波的有效分辨率，通过TIMERx_CHxCV寄存器中的CHxVAL[19:0]位域（TIMER2 / 3）或CARL[31:0]位域（TIMER1 / 4）可以提高占空比分辨率，通过TIMERx_CAR寄存器中的CHxVAL[19:0]位域（TIMER2 / 3）或CARL[31:0]位域（TIMER1 / 4）可以提高PWM频率的分辨率。

当微调模式使能时，CHxVAL位域和CARL位域的低16位[15:0]（TIMER2 / 3）或低28位[27:0]

（TIMER1 / 4）用于整数部分，高4位[19:16]（TIMER2 / 3）或[31:28]（TIMER1 / 4）用于微调的小数部分。通过预定义的方式，在连续16个周期内对CHxVAL值或CARL值进行微调（每次调整不超过一个TIMER时钟周期），可增加16倍的分辨率。

图 23-77. 微调模式：数据格式和寄存器位域



根据ADMEN位的配置（置位或清零），CHxVAL位域和CARL位域将自动更新。当需要对ADMEN位进行清零时，需要遵循以下步骤：

1. CEN位和ARSE位必须清零；
2. CARL[19:16]位域（TIMER2 / 3）或CARL[31:28]位域（TIMER1 / 4）必须清零；
3. ADMEN位必须清零；
4. CHxIF位必须清零；
5. 可以将CEN位置1。

以下公式可以计算PWM分辨率：

$$Resolution = f_{PSC_CLK} \ / \ f_{pwm} \quad (23\text{-}4)$$

由式*(23-4)*可得，微调模式禁能时（ADMEN=0），PWM的最小频率f_{pwm}:

16位（TIMER2/3），

$$(f_{pwm})_{min} = f_{PSC_CLK}/65536 \quad (23\text{-}5)$$

32位（TIMER1/4），

$$(f_{pwm})_{min} = f_{PSC_CLK} / 2^{28} \quad (23\text{-}6)$$

微调模式使能时（ADMEN=1），

16位（TIMER2 / 3），

$$(f_{pwm})_{min} = f_{PSC_CLK} / (65535 + 15/16) \quad (23\text{-}7)$$

32位（TIMER1 / 4），

$$(f_{pwm})_{min} = f_{PSC_CLK} / [(2^{28}-1) + 15/16] \quad (23\text{-}8)$$

当微调模式使能时，CHxVAL[19:0]位域和CARL[19:0]位域的最大值为0xFFFFE（整数部分为0xFFFE，小数部分为0xF），CHxVAL[31:0]位域和CARL[31:0]位域的最大值为0xFFFFFFFE（整数部分为0xFFFFFFE，小数部分为0xF）。

在连续16个周期内，占空比和周期的变化情况，具体如*图23-78. PWM微调模式原理（16位）*和*表23-13. 边沿对齐模式中CHxVAL和CARL位域的变化*所示。

图 23-78. PWM 微调模式原理（16 位）

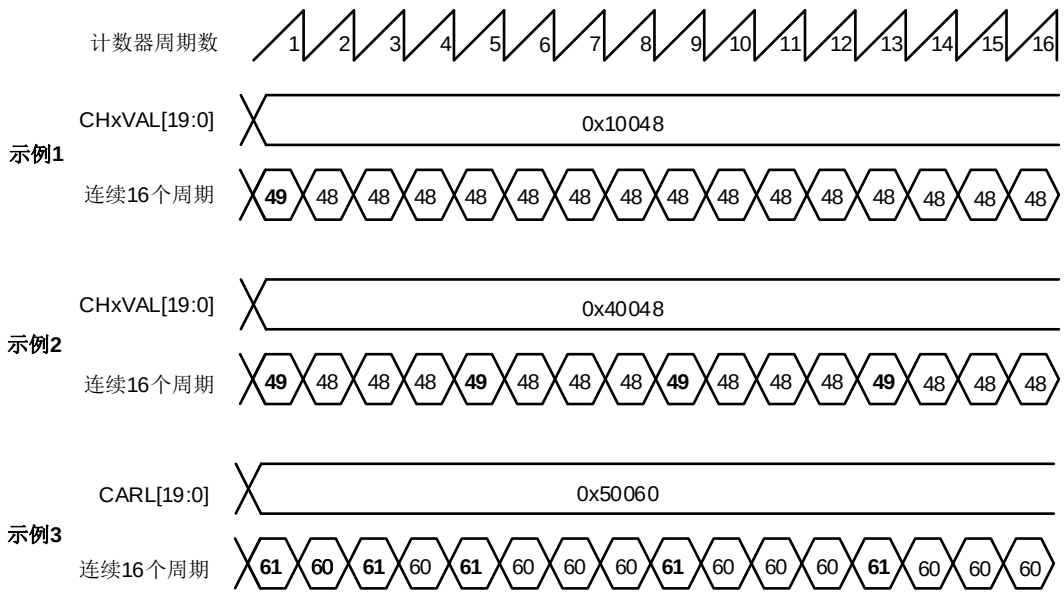


表 23-13. 边沿对齐模式中 CHxVAL 和 CARL 位域的变化

CHxVAL[19:16] / CARL[19:16]	周期															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0010	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-

CHxVAL[19:16] / CARL[19:16]	周期															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0011	+1	-	-	-	+1	-	-	-	+1	-	-	-	-	-	-	-
0100	+1	-	-	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0101	+1	-	+1	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0110	+1	-	+1	-	+1	-	-	-	+1	-	+1	-	+1	-	-	-
0111	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	-	-
1000	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1001	+1	+1	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1010	+1	+1	+1	-	+1	-	+1	-	+1	+1	+1	-	+1	-	+1	-
1011	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	-	+1	-
1100	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1101	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1110	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	+1	+1	+1	+1	-
1111	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	-

PWM微调模式也适用于中央对齐模式，具体请见*表23-14. 中央对齐模式中CHxVAL和CARL位域的变化*，微调模式应用在8个连续的PWM周期。

表 23-14. 中央对齐模式中 CHxVAL 和 CARL 位域的变化

CHxVAL [19:16] / CARL [19:16]	周期															
	1		2		3		4		5		6		7		8	
	向上计数	向下计数	向上计数	向下计数	向上计数	向下计数	向上计数	向下计数	向上计数	向下计数	向上计数	向下计数	向上计数	向下计数	向上计数	向下计数
0000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0010	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-
0011	+1	-	-	-	+1	-	-	-	+1	-	-	-	-	-	-	-
0100	+1	-	-	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0101	+1	-	+1	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0110	+1	-	+1	-	+1	-	-	-	+1	-	+1	-	+1	-	-	-
0111	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	-	-
1000	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1001	+1	+1	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1010	+1	+1	+1	-	+1	-	+1	-	+1	+1	+1	-	+1	-	+1	-
1011	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	-	+1	-
1100	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1101	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1110	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	+1	+1	+1	+1	-
1111	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	-

复合 PWM 模式

在复合 PWM 模式中（CHxCPWMEN = 1’b1，CHxMS[2:0] = 3’b000 和 CHxCOMCTL = 4’b0110、4’b0111），通道 x（x = 0…3）上的 PWM 输出信号由 CHxVAL 和 CHxCOMVAL_ADD 位确定。

如果 CHxCOMCTL = 4’b0110（PWM 模式 0）且 DIR = 1’b0（向上计数模式），或者 CHxCOMCTL = 4’b0111（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 =4’b0111（PWM 模式 1）且 DIR = 1’b0（向上计数模式），或者 CHxCOMCTL =4’b0110（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 23-15. 复合 PWM 脉冲宽度*计算。

表 23-15. 复合 PWM 脉冲宽度

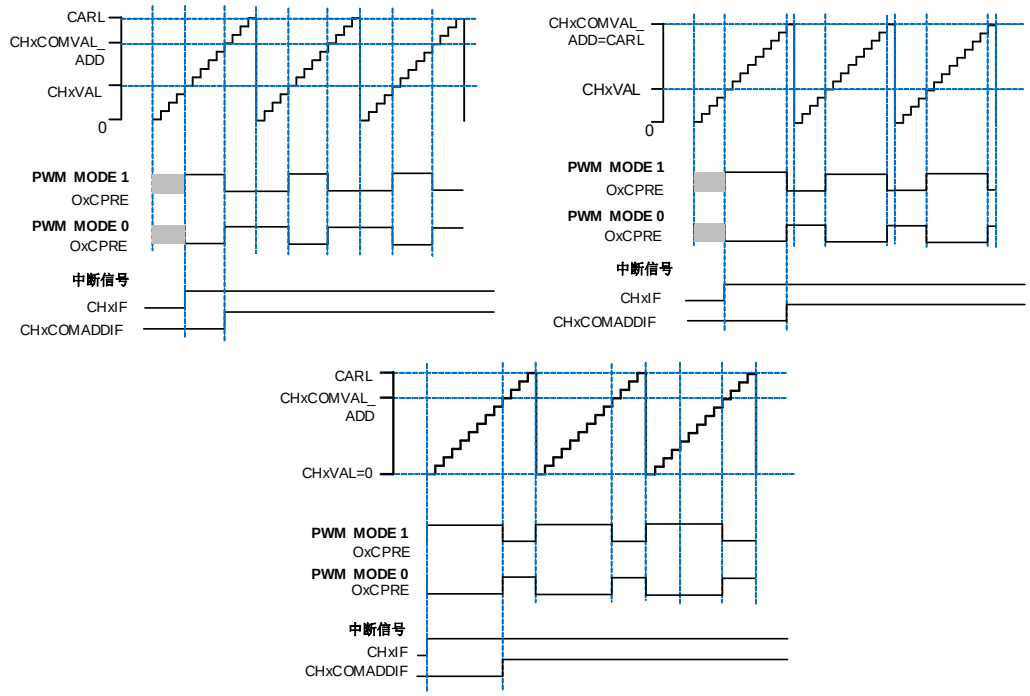
条件	模式	PWM 脉冲宽度
CHxVAL < CHxCOMVAL_ADD ≤ CARL	PWM 模式 0	（CARL + 0x0001） + （CHxVAL – CHxCOMVAL_ADD）
	PWM 模式 1	（CHxCOMVAL_ADD – CHxVAL）
CHxCOMVAL_ADD < CHxVAL ≤ CARL	PWM 模式 0	（CHxVAL - CHxCOMVAL_ADD）
	PWM 模式 1	（CARL + 0x0001） + （CHxCOMVAL_ADD – CHxVAL）
（CHxVAL = CHxCOMVAL_ADD ≤ CARL）或（CHxVAL > CARL > CHxCOMVAL_ADD）	PWM 模式 0（向上计数）或 PWM 模式 1（向下计数）	100%
	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：

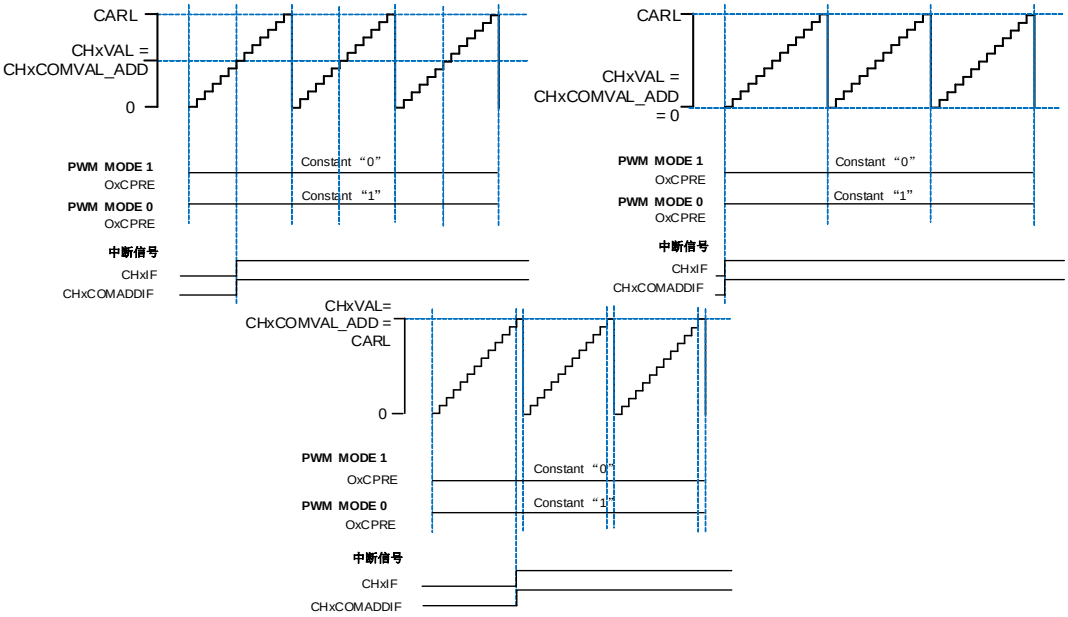
- CHxVAL < CHxCOMVAL_ADD，CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 23-79 通道 x 输出 PWM（CHxVAL < CHxCOMVAL_ADD）



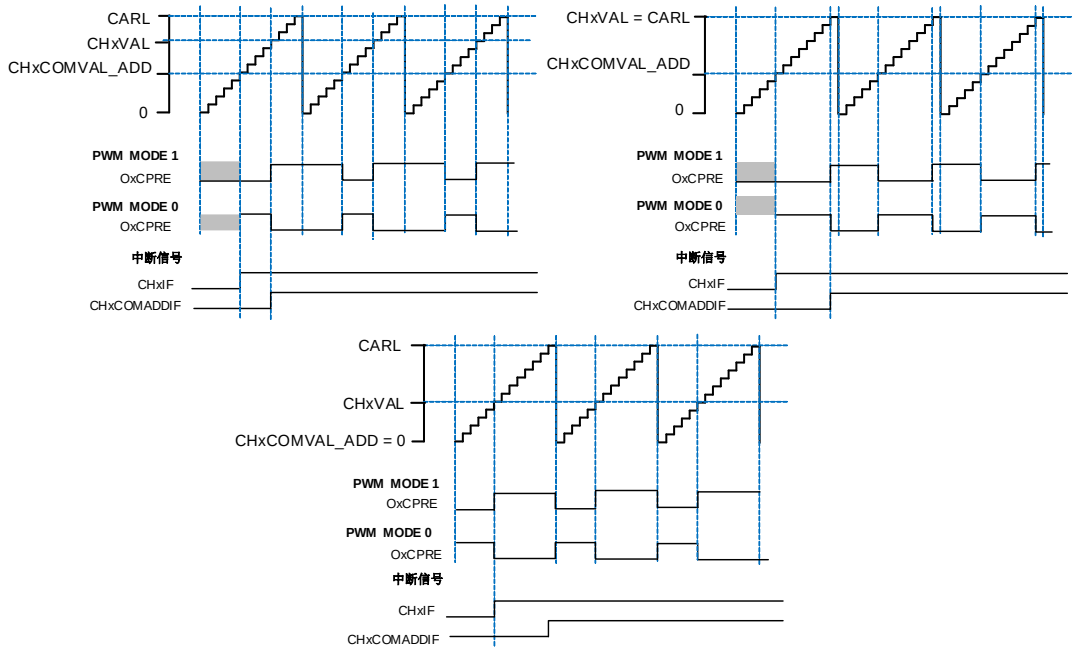
- CHxVAL = CHxCOMVAL_ADD, CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 23-80. 通道 x 输出 PWM（CHxVAL = CHxCOMVAL_ADD）



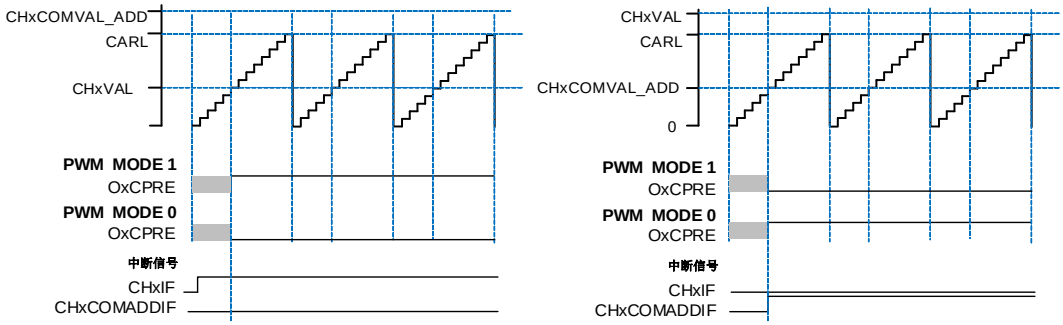
- CHxVAL > CHxCOMVAL_ADD, CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 23-81. 通道 x 输出 PWM（CHxVAL > CHxCOMVAL_ADD）



- CHxVAL或CHxCOMVAL_ADD值大于CARL。

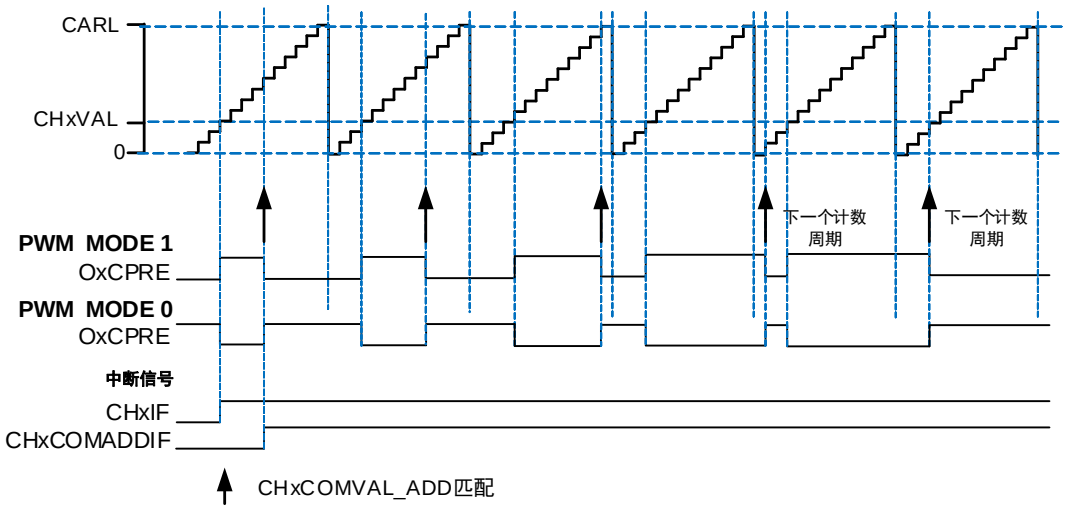
图 23-82. 通道 x 输出 PWM（CHxVAL 或 CHxCOMVAL_ADD > CARL）



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23-83. 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。

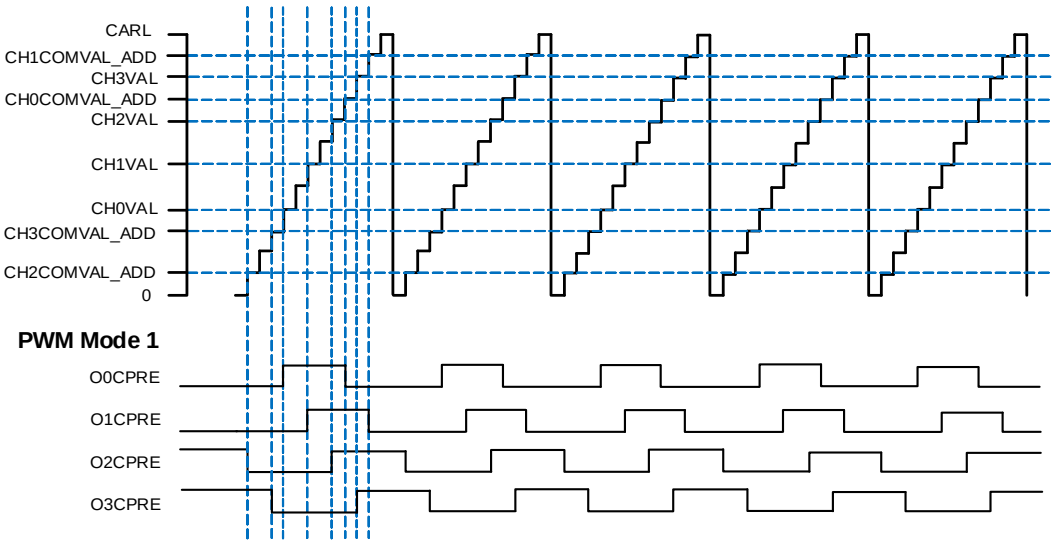
在某些情况下，CHxCOMVAL_ADD的匹配事件可以发生在下一个计数周期（CHxCOMVAL_ADD值在计数器到达CHxVAL值之后被写入，且CHxCOMVAL_ADD值小于或者等于CHxVAL值）。

图 23-83. 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 23-84. 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[3:0]（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[3:0]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 23-85. 边沿对齐模式下 CHx_O 输出脉冲（CHxOMPSEL ≠ 2’b00）

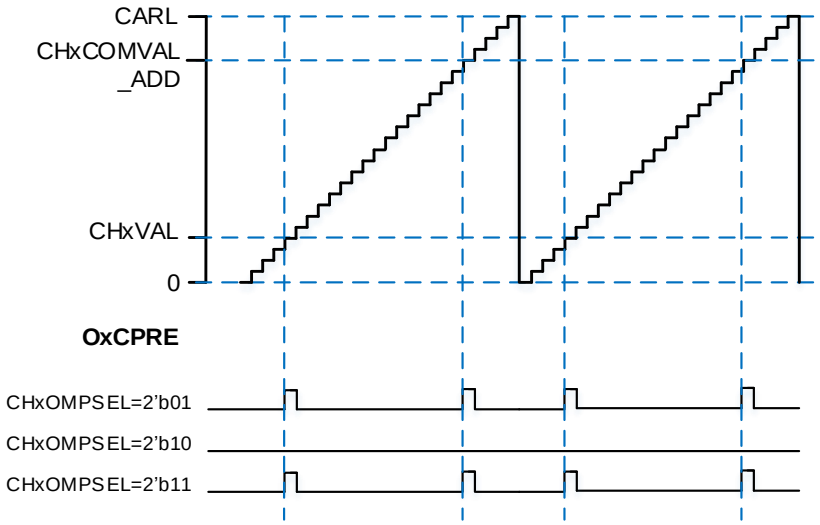
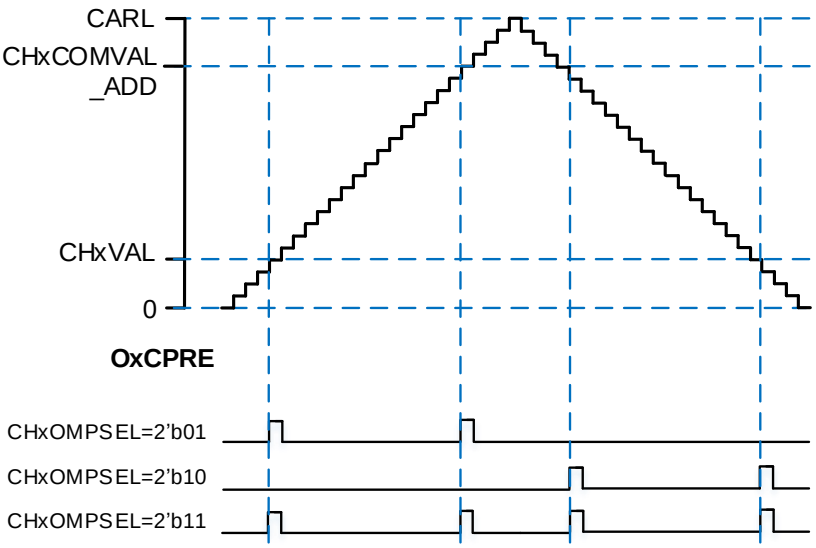


图 23-86. 中央对齐模式下 CHx_O 输出脉冲（CHxOMPSEL ≠ 2’b00）



通道输出准备信号

根据*图 23-73. 输出比较逻辑（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 ETIFP 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。

清除通道输出准备信号

当 CHxCOMCEN 位（在 TIMERx_CHCTLy 寄存器中）置 1 时，OxCPRE 信号可以由 OCPRE_CLR_INT信号清除。该功能用于CHxCOMCTL[3:0]位域位域（4'b0100和4'b0101除外）中配置的比较输出模式。

可以通过TIMERx_SMCFG寄存器中的OCRC位来选择OCPRE_CLR_INT的信号源。

OCRC 位清 0 时，OCPRE_CLR_INT 连接到 OCPRE_CLR 输入。OxCPRE 信号被 OCPRE_CLR_INT 信号的高电平清除，直到下一个更新事件发生时才会恢复输出。在 TIMERx_AFCTL1 寄存器的 OCRINSEL[2:0]位域中选择 OCPRE_CLR 的输入。

OCRC 位置 1 时，OCPRE_CLR_INT 连接到 ETIF。由 TIMERx_SMCFG 寄存器中的 ETP 位配置 OCPRE_CLR_INT 的输入极性。此时，ETPSC[1:0]位域必须设置为 2'b00。

正交译码器

参考*高级定时器(TIMERx,x = 0,7)正交编码器*。

译码器

参考*高级定时器(TIMERx,x = 0,7)译码器*。

正交译码器和译码器的时钟输出

参考*高级定时器(TIMERx,x = 0,7)正交译码器和译码器的时钟输出*

索引输入功能

参考*高级定时器(TIMERx,x = 0,7)索引输入功能*。

霍尔传感器接口功能

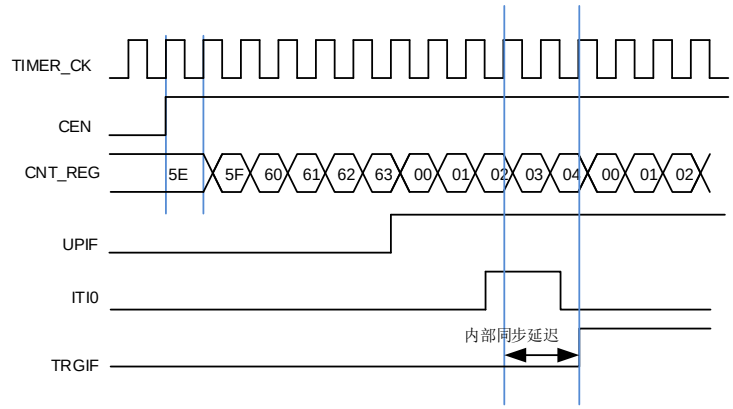
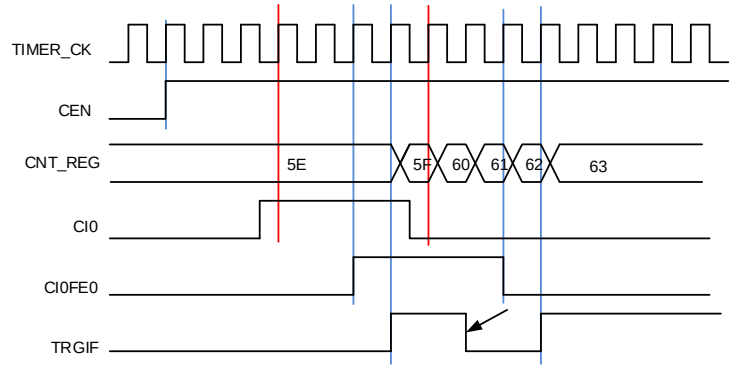
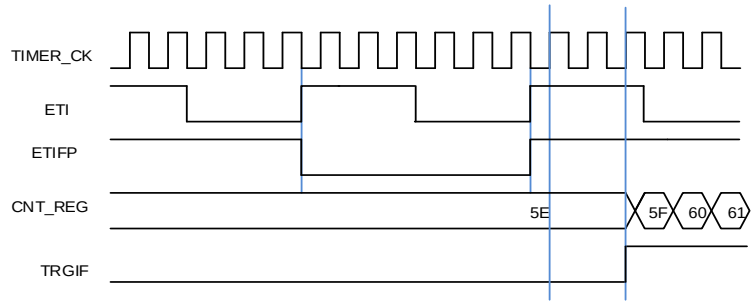
参考*高级定时器(TIMERx,x = 0,7)霍尔传感器接口功能*。

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等，可以通过设置SYSCFG_TIMERxCFG(x = 1..4)寄存器中的 TSCFGy[4:0] (y = 3..7)位域来确定，具体的输入触发源可以通过 TSCFGy[4:0] (y = 3..7)位域值来选择。

表 23-16. 从模式列表和举例（通用定时器 L0）

	模式选择	触发源选择	极性选择	滤波和预分频
列举	TSCFGy[4:0] y = 3：复位模式 y = 4：暂停模式 y = 5：事件模式 y = 6：外部时钟模式0 y = 7：复位+事件模式 y = 8：暂停 + 事件模式	TSCFGy[4:0] 00000：模式禁能 00001：ITI0 00010：ITI1 00011：ITI2 00100：ITI3 00101：CI0F_ED 00110：CI0FE0 00111：CI1FE1 01000：ETIFP(1) 01001：ITI4 01010：ITI5 01011：ITI6 01100：ITI7 01101：ITI8 01110：ITI9 01111：ITI10 10000：保留 10001：保留 10010：保留 10011：ITI14	如果触发源是CI0FE0或者CI1FE1，配置CHxP和CHxNP来选择极性和反相。 如果触发源是ETIFP（滤波后的ETI外部触发输入），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ETIFP，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	TSCFG3[4:0] 5’b00001，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 23-87. 复位模式
 
</td></tr>
<tr><td>例2</td><td>暂停模式
当触发输入为低的时候，计数器暂停计数</td><td>TSCFG4[4:0]
=5’b00110，
选择CI0FE0为触发源</td><td>TI0S=0（非异或）
[CH0NP=0, CH0P=0]
不反相，在上升沿捕获</td><td>在这个例子中滤波被旁路</td></tr>
<tr><td></td><td colspan="4">图 23-88. 暂停模式
 
</td></tr>
<tr><td>例3</td><td>事件模式
触发输入的上升沿计数器开始计数</td><td>TSCFG5[4:0]
=5’b01000，
选择ETIFP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></td><td colspan="4">图 23-89. 事件模式
 
</td></tr>
<tr><td>例 4</td><td colspan="4">复位 + 事件模式
当触发输入的上升沿到来时，计数器被重新初始化并开始计数。该模式仅用于可延时的单脉冲模式。</td></tr>
</table>

	模式选择	触发源选择	极性选择	滤波和预分频
例 5	**暂停 + 复位模式** 当触发输入的上升沿或下降沿（由 TIMERx_SMCFG 寄存器中的 PRMRPSEL 位配置）到来时，计数器将复位。当触发输入高时计数器计数，当触发输入低时计数器停止。在这种模式下，计数器的开始和停止可以被控制。			

(1) ETI 信号可以从外部 ETI 引脚输入，也可由片上外设提供，具体情况可以参考 *TIMER1_ETI 触发选择寄存器（TRIGSEL_TIMER1ETI）*、*TIMER2_ETI 触发选择寄存器（TRIGSEL_TIMER2ETI）*、*TIMER3_ETI 触发选择寄存器（TRIGSEL_TIMER3ETI）*和 *TIMER4_ETI 触发选择寄存器（TRIGSEL_TIMER4ETI）*模块。

单脉冲模式

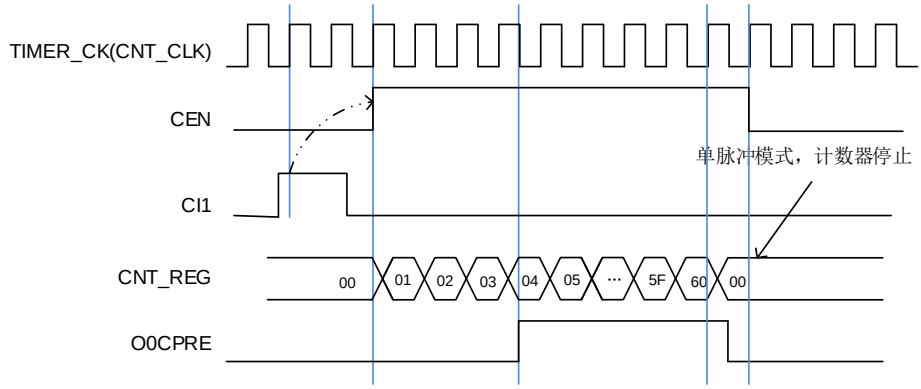
单脉冲模式与重复模式是相反的，设置 TIMERx_CTL0 寄存器的 SPM 位置 1，则使能单脉冲模式。当 SPM 置 1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置 CHxCOMCTL 配置 TIMERx 为 PWM 模式或者比较模式。

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。如果 CEN 值被硬件更新事件自动清 0，计数器将被再次初始化。

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。

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 23-90. 单脉冲模式，TIMERx_CHxCV = 0x04，TIMERx_CAR = 0x60



可延时的单脉冲模式

可以通过将TIMERx_CHCTLx寄存器中的CHxCOMCTL[3:0]位置1来使能可延时的单脉冲模式。在

这个模式下，通道输出准备信号OxCPRE的脉冲宽度由TIMERx_CAR寄存器值确定。

一旦设置定时器运行在可延时的单脉冲模式下，需进行以下配置：

- 定时器必须工作在从模式下，SYSCFG_TIMERxCFG(x = 1..4)寄存器中的 TSCFG7[4:0] != 5'b00000，从模式选择复位 + 事件模式；
- CHxCOMCTL[3:0]位设置为 4'b1000（可延时单脉冲模式 0）或 4'b1001（可延时单脉冲模式 1）

在可延时单脉冲模式0下，OxCPRE的输出情况类似与PWM模式0。在向上计数模式时，OxCPRE先输出有效电平，当外部触发事件发生时，立即输出无效电平，当下一次更新事件发生时，再变成有效电平；在向下计数模式时，OxCPRE先输出无效电平，当外部触发事件发生时，立即输出有效电平，当下一次更新事件发生时，再变成无效电平。

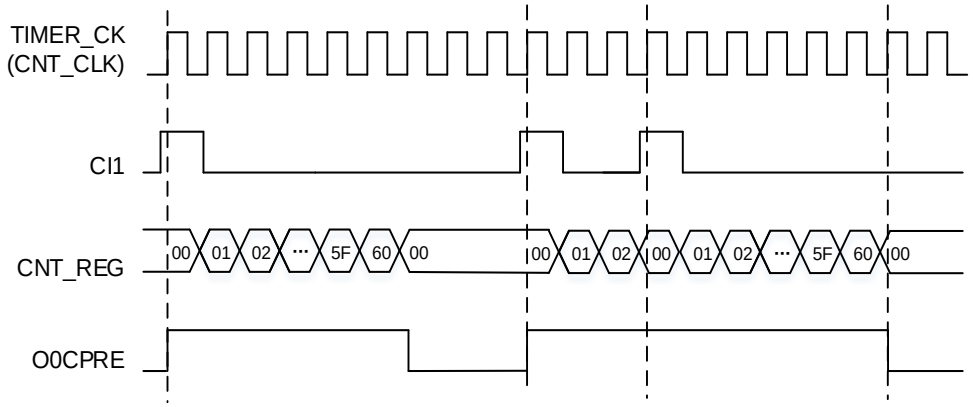
在可延时单脉冲模式1下，OxCPRE的输出情况类似与PWM模式1。在向上计数模式时，OxCPRE先输出无效电平，当外部触发事件发生时，立即输出有效电平，当下一次更新事件发生时，再变成无效电平；在向下计数模式时，OxCPRE先输出有效电平，当外部触发事件发生时，立即输出无效电平，当下一次更新事件发生时，再变成有效电平。

PWM微调模式也可用于可延迟的单脉冲模式。

注意：

- 不能使用中央对齐模式，TIMERx_CTL0 寄存器中的 CAM[1:0] = 2'b00；
- 在向上计数时（TIMERx_CTL0 寄存器中的 DIR = 0），TIMERx_CHxCV 的值设置为 0；在向下计数时，TIMERx_CHxCV 的值应大于或等于 TIMERx_CAR 的值。

图 23-91. 可延时单脉冲模式（TIMERx_CHxCV = 0x00，TIMERx_CAR = 0x60）



可编程的脉冲输出

参考*高级定时器(TIMERx,x = 0,7)可编程的脉冲输出*。

定时器互连

参考*高级定时器(TIMERx,x = 0,7)定时器互连*。

定时器 DMA 模式

定时器 DMA 模式是指通过 DMA 模块配置定时器的寄存器。有两个跟定时器 DMA 模式相关的寄存器：TIMERx_DMACFG 和 TIMERx_DMATB。当然，必须要使能 DMA 请求，一些内部中断事件可以产生 DMA 请求。当中断事件发生，TIMERx 会给 DMA 发送请求。DMA 配置成 M2P 模式，PADDR 是 TIMERx_DMATB 寄存器地址，DMA 就会访问 TIMERx_DMATB 寄存器。实际上，TIMERx_DMATB 寄存器只是一个缓冲，定时器会将 TIMERx_DMATB 映射到一个内部寄存器，这个内部寄存器由 TIMERx_DMACFG 寄存器中的 DMATA 来指定。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值为 0，表示 1 次传输，定时器的发送 1 个 DMA 请求就可以完成。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值不为 1，例如其值为 3，表示 4 次传输，定时器就需要再多发 3 次 DMA 请求。在这 3 次请求下，DMA 对 TIMERx_DMATB 寄存器的访问会映射到访问定时器的 DMATA+0x4, DMATA+0x8，DMATA+0xc 寄存器。总之，发生一次 DMA 内部中断请求，定时器会连续发送（DMATC+1）次请求。

如果再来 1 次 DMA 请求事件，TIMERx 将会重复上面的过程。

输出 DIR 位

DIR位可以在CH2和CH3通道上输出，配置在TIMERx_CHCTL1寄存器中的CH2COMCTL[3:0]或CH3COMCTL[3:0]位域值为4b'1011来使能该功能。

当计数器工作在中央对齐模式时，该功能可用于指示计数器的计数方向。当计数器工作在译码器模式时，此功能可用于指示外部信号的旋转方向。

UPIF 位备份功能

可以通过配置TIMERx_CTL0寄存器中的UPIFBUEN位来使能UPIF位的备份功能，UPIF和UPIFBU位之间没有延迟，两者完全同步。

使能该功能后，TIMERx_INTF寄存器中的UPIF位将会被实时备份到TIMERx_CNT寄存器中的UPIFBU位。这可以避免在读计数器和中断处理时产生冲突的情况。

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。

23.2.5. TIMERx 寄存器（x = 1 / 2 / 3 / 4）

TIMER1基地址：0x4000 0000

TIMER2基地址：0x4000 0400

TIMER3基地址：0x4000 0800

TIMER4基地址：0x4000 0C00

控制寄存器 0（TIMERx_CTL0）

地址偏移：0x00

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

位/位域	名称	描述
31:13	保留	必须保持复位值。
12	ADMEN	微调模式使能 0：微调模式禁能 1：微调模式使能 注意：该位只能在 CEN 位为 0 时修改。
11	UPIFBUEN	UPIF位备份使能 0：备份禁能。UPIF位没有备份到TIMERx_CNT寄存器中的UPIFBU位 1：备份使能。UPIF 位备份到 TIMERx_CNT 寄存器中的 UPIFBU 位
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）与死区时间和采样时钟（DTS）之间的分频系数，死区发生器和数字滤波器会用到 DTS 时间。 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，通道的比较中断标志置 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，通道的比较中断标志置 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，通道的比较中断标志都会置 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，CEN 硬件清零并且计数器停止计数
2	UPS	更新请求源 软件配置该位，选择更新事件源. 0:使能后，下述任一事件产生更新中断或 DMA 请求： – UPG 位被置 1 – 计数器上溢/下溢 – 从模式控制器产生的更新 1:使能后只有计数器上溢/ 下溢才产生更新中断或 DMA 请求。
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。当以下事件之一发生时，更新事件产生，具有缓存的寄存器被装入它们的预装载值： – UPG 位被置 1 – 计数器上溢/下溢 – 从模式控制器产生一个更新事件

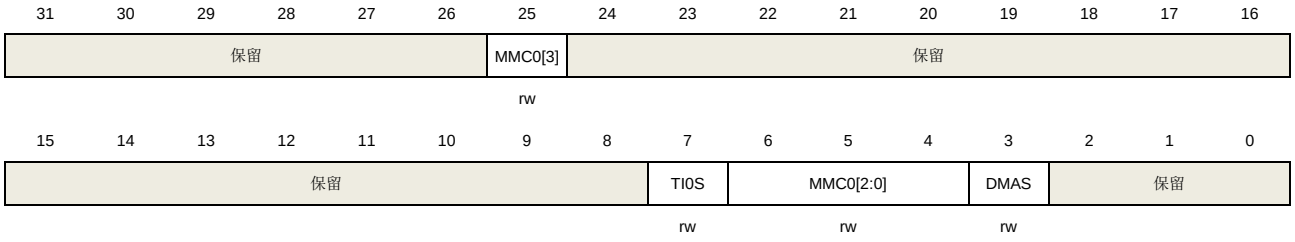
		1：更新事件禁能。带有缓存的寄存器保持原有值，如果 UPG 位被置 1 或者从模式控制器产生一个硬件复位事件，计数器和预分频器被重新初始化。
0	CEN	计数器使能 0：计数器禁能 1：计数器使能 在软件将 CEN 位置 1 后，外部时钟、暂停模式和译码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。

控制寄存器 1（TIMERx_CTL1）

地址偏移：0x04

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:26	保留	必须保持复位值。
25	MMC0[3]	主模式控制 0 请参考 MMC0[2:0]的描述。
24:8	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	MMC0[2:0]	主模式控制 0 这些位控制 TRGO0 信号的选择，TRGO0 信号由主定时器发给从定时器用于同步功能。 0000：复位。TIMERx_SWEVG 寄存器的 UPG 位被置 1 或从模式控制器产生复位触发一次 TRGO0 脉冲，后一种情况下，TRGO0 上的信号相对实际的复位会有一个延迟。 0001：使能。此模式可用于同时启动多个定时器或控制在一段时间内使能从定时器。主模式控制器选择计数器使能信号作为触发输出 TRGO0。当 CEN 控制位被置 1 或者暂停模式下触发输入为高电平时，计数器使能信号被置 1。在暂停模式下，

		计数器使能信号受控于触发输入，在触发输入和 TRGO0 上会有一个延迟，除非选择了主/ 从模式。 0010：更新。主模式控制器选择更新事件作为 TRGO0。 0011：捕获/比较脉冲。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 TRGO0 脉冲 0100：比较。在这种模式下主模式控制器选择 O0CPRE 信号被用于作为触发输出 TRGO0 0101：比较。在这种模式下主模式控制器选择 O1CPRE 信号被用于作为触发输出 TRGO0 0110：比较。在这种模式下主模式控制器选择 O2CPRE 信号被用于作为触发输出 TRGO0 0111：比较。在这种模式下主模式控制器选择O3CPRE信号被用于作为触发输出 TRGO0 1000：译码器时钟输出。在这种模式下主模式控制器选择译码器时钟信号作为触发输出TRGO0。此值仅用于正交译码器模式0~4和译码器模式0~3。 1001~1111：保留
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 CHx 的 DMA 请求 1：当更新事件发生，发送通道 CHx 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					PRMRPSEL	DECMODS	DECMODEN	保留							
					rw	rw	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	保留			OCRC	保留		
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	PRMRPSEL	暂停 + 复位模式复位极性选择 0：计数器在下降沿复位

		1：计数器在上升沿复位
25	DECMODS	译码器模式更新源 该位用于选择译码器模式的更新源。 0：模式修改后，通过 TIMER 的更新事件更新译码器模式 1：模式修改后，通过索引事件更新译码器模式
24	DECMODEN	译码器模式在线修改使能 0：译码器模式在线修改禁能 1：译码器模式在线修改使能 该位置 1 时，译码器模式可以从一种模式修改为另一种模式。
23:16	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	从模式的一部分为了使能外部时钟模式 1 在外部时钟模式 1，计数器由 ETIFP 信号上的任意有效边沿驱动。 0：外部时钟模式 1 禁能 1：外部时钟模式 1 使能 复位模式，暂停模式和事件模式可以与外部时钟模式 1 同时使用。但是 TSCFGy[4:0](y = 3,4,5)位域的值不能为 5'b01000。 如果外部时钟模式 0 和外部时钟模式 1 同时被使能，外部时钟的输入是 ETIFP。 **注意**：外部时钟模式 0 使能在 SYSCFG_TIMERxCFG1 寄存器中的 TSCFG6[4:0] 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 ETIFP 的频率。 00：预分频禁能 01：ETI 频率被 2 分频 10：ETI 频率被 4 分频 11：ETI 频率被 8 分频
11:8	ETFC[3:0]	外部触发滤波控制 数字滤波器是一个事件计数器，它记录到 N 个事件后会产生一个输出的跳变。这些位定义了对 ETI 信号采样的频率和对 ETI 数字滤波的带宽。 0000：滤波器禁能 $f_{SAMP}= f_{DTS}$，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

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 TRGO0，定时器被连接在一起，TRGO0 用做启动事件。 0：主从模式禁能 1：主从模式使能
6:4	保留	必须保持复位值。
3	OCRC	OxCPRE清除源选择 该位用于选择OxCPRE清除源。 0：OCPRE_CLR_INT连接OCPRE_CLR输入 1：OCPRE_CLR_INT 连接到 ETIF
2:0	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
CH3COMADDIE	CH2COMADDIE	CH1COMADDIE	CH0COMADDIE	保留								INDERRIE	DIRTRANIE	DECDISIE	DECJIE
rw	rw	rw	rw									rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INDIE	TRGDEN	保留	CH3DEN	CH2DEN	CH1DEN	CH0DEN	UPDEN	保留	TRGIE	保留	CH3IE	CH2IE	CH1IE	CH0IE	UPIE
rw	rw		rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

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。
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。
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。
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。
27:20	保留	必须保持复位值。
19	INDERRIE	索引错误中断使能 0：索引错误中断禁能 1：索引错误中断使能
18	DIRTRANIE	计数方向变化中断使能 0：计数方向变化中断禁能 1：计数方向变化中断禁能
17	DECDISIE	正交译码器信号断线检测使能 0：禁能 1：使能 **注意**：该位仅用于正交译码器信号断线检测使能（DECDISDEN =1）时。
16	DECJIE	正交译码器信号跳变（两个信号同时发生跳变）中断使能 0：禁能 1：使能 **注意**：该位仅用于正交译码器信号同时跳变检测使能（DECJDEN =1）时。
15	INDIE	索引中断使能 0：索引中断禁能 1：索引中断使能
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
CH3COMADDIF	CH2COMADDIF	CH1COMADDIF	CH0COMADDIF	保留								INDERRIF	DIRTRANIF	DECDISIF	DECJIF
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
INDIF	保留		CH3OF	CH2OF	CH1OF	CH0OF	保留		TRGIF	保留	CH3IF	CH2IF	CH1IF	CH0IF	UPIF
rc_w0			rc_w0	rc_w0	rc_w0	rc_w0			rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位/位域	名称	描述
31	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。
27:20	保留	必须保持复位值。
19	INDERRIF	索引错误中断标志位 此标志由硬件置 1 软件清 0。 0：无索引错误中断发生 1：索引错误中断发生

18	DIRTRANIF	计数方向变化中断标志位 译码器模式下，此标志在 TIMERx_CTL0 寄存器中的 DIR 位发生变化时由硬件置 1，并由软件清 0。 0：无计数方向变化中断发生 1：计数方向变化中断发生
17	DECDISIF	正交译码器信号断线中断标志位 0：无正交译码器信号断线中断发生 1：正交译码器信号断线中断发生 **注意**：该位仅用于正交译码器信号断线检测使能（DECDISDEN =1）时。
16	DECJIF	正交译码器信号跳变（两个信号同时发生跳变）中断标志位 0：无正交译码器信号跳变中断发生 1：正交译码器信号跳变中断发生 **注意**：该位仅用于正交译码器信号同时跳变检测使能（DECJDEN =1）时。
15	INDIF	索引中断标志位 此标志由硬件置 1 软件清 0。 0：无索引中断发生 1：索引中断发生
14:13	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，此标志由硬件置 1。此位由软件清 0。当从模式控制器处于除暂停模式外的其它模式时，在 TRGI 输入端检测到有效边沿，产生触发事件。当从模式控制器处于暂停模式时，TRGI 的任意边沿都可以产生触发事件。 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
CH3COMADDG	CH2COMADDG	CH1COMADDG	CH0COMADDG	保留											
w	w	w	w												
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	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。
27:7	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
CH1MS[2]	CH0MS[2]	CH1COMADDSEN	CH0COMADDSEN	CH1ADDUPS	CH0ADDUPS	保留	CH1COMCTL[3]	保留							CH0COMCTL[3]
		保留	保留	保留	保留		保留								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
CH1COMCEN	CH1COMCTL[2:0]			CH1COMSEN	CH1COMFEN	CH1MS[1:0]		CH0COMCEN	CH0COMCTL[2:0]			CH0COMSEN	CH0COMFEN	CH0MS[1:0]	
CH1CAPFLT[3:0]				CH1CAPPSC[1:0]				CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

输出比较模式：

位/位域	名称	描述
31	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。
27	CH1ADDUPS	通道1附加寄存器更新源 0：在发生更新事件时，更新TIMERx_CH1COMV_ADD寄存器 1：在发生计数器计数值匹配 CH1VAL 值时，更新 TIMERx_CH1COMV_ADD 寄存器

26	CH0ADDUPS	通道0附加寄存器更新源 0：在发生更新事件时，更新TIMERx_CH0COMV_ADD寄存器 1：在发生计数器计数值匹配 CH0VAL 值时，更新 TIMERx_CH0COMV_ADD 寄存器
25	保留	必须保持复位值。
24	CH1COMCTL[3]	通道 1 输出比较控制 参见 CH0COMCTL[2:0]描述
23:17	保留	必须保持复位值。
16	CH0COMCTL[3]	通道 0 输出比较控制 参见 CH0COMCTL[2:0]描述
15	CH1COMCEN	通道 1 输出比较清 0 使能 参见 CH0COMCEN 描述。
14:12	CH1COMCTL[2:0]	通道 1 输出比较模式 参见 CH0COMCTL 描述。
11	CH1COMSEN	通道 1 输出比较影子寄存器使能 参见 CH0COMSEN 描述。
10	CH1COMFEN	通道1输出比较快速使能 参见 CH0COMFEN 描述
9:8	CH1MS[1:0]	通道 1 模式选择 CH1MS[2:0]位域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。 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器输入被选中时（由 SYSCFG_TIMERxCFG2(x = 1..4)寄存器中的 TSCFG15[4:0]位域选择）。 100~111：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 CH0COMCTL[3]和 CH0COMCTL[2:0]位域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。

0000：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

0001：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。

0010：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。

0011：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。

0100：强制为低。强制 O0CPRE 为低电平

0101：强制为高。强制 O0CPRE 为高电平

0110：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。

0111：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。

1000：可延时的单脉冲模式0。O0CPRE的输出情况类似与PWM模式0。在向上计数模式时，O0CPRE先输出有效电平，当外部触发事件发生时，立即输出无效电平，当下一次更新事件发生时，再变成有效电平；在向下计数模式时，O0CPRE先输出无效电平，当外部触发事件发生时，立即输出有效电平，当下一次更新事件发生时，再变成无效电平。

1001：可延时的单脉冲模式1。O0CPRE的输出情况类似与PWM模式1。在向上计数模式时，O0CPRE先输出无效电平，当外部触发事件发生时，立即输出有效电平，当下一次更新事件发生时，再变成无效电平；在向下计数模式时，O0CPRE先输出有效电平，当外部触发事件发生时，立即输出无效电平，当下一次更新事件发生时，再变成有效电平。

1010 / 1011 / 1100 / 1101 / 1110 / 1111：保留

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***。

在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或者输出比较模式中从时基模式切换到 PWM 模式时，O0CPRE 电平才改变。

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（TIMERx_CTL0 寄存器的 SPM =1），可以在未确认预装载寄存器情况下使用 PWM 模式。

2	CH0COMFEN	通道0输出比较快速使能 当该位为1时，如果通道配置为PWM模式0或者PWM模式1，会加快捕获 / 比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH0_O被设置为比较电平而与比较结果无关。 0：通道0输出比较快速禁能。当触发器的输入有一个有效沿时，激活CH0_O输出的最小延时为5个时钟周期 1：通道0输出比较快速使能。当触发器的输入有一个有效沿时，激活CH0_O输出的最小延时为 3 个时钟周期
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，CH0MS[2:0]位才可以写。 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（由 SYSCFG_TIMERxCFG2(x = 1..4)寄存器中的 TSCFG15[4:0]位域选择）。 100~111：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 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 CI0 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	CH0CAPPSC[1:0]	通道 0 输入捕获预分频器 这 2 位定义了通道 0 输入的预分频系数。当 TIMERx_CHCTL2 寄存器中的 CH0EN =0 时，则预分频器复位。 00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获 01：每 2 个事件触发一次捕获 10：每 4 个事件触发一次捕获 11：每 8 个事件触发一次捕获
1:0	CH0MS[1:0]	通道 0 模式选择 与输出比较模式相同。

通道控制寄存器 1（TIMERx_CHCTL1）

地址偏移：0x1C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3MS[2]	CH2MS[2]	CH3COMADDSEN	CH2COMADDSEN	CH3ADDUPS	CH2ADDUPS	保留	CH3COMCTL[3]	保留							CH2COMCTL[3]
		保留	保留	保留	保留		保留								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
CH3COMCEN	CH3COMCTL[2:0]			CH3COMSEN	CH3COMFEN	CH3MS[1:0]		CH2COMCEN	CH2COMCTL[2:0]			CH2COMSEN	CH2COMFEN	CH2MS[1:0]	
CH3CAPFLT[3:0]				CH3CAPPSC[1:0]				CH2CAPFLT[3:0]				CH2CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

输出比较模式：

位/位域	名称	描述
31	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。
27	CH3ADDUPS	通道3附加寄存器更新源 0：在发生更新事件时，更新TIMERx_CH3COMV_ADD寄存器 1：在发生计数器计数值匹配 CH3VAL 值时，更新 TIMERx_CH3COMV_ADD 寄存器
26	CH2ADDUPS	通道2附加寄存器更新源 0：在发生更新事件时，更新TIMERx_CH2COMV_ADD寄存器 1：在发生计数器计数值匹配 CH2VAL 值时，更新 TIMERx_CH2COMV_ADD 寄存器
25	保留	必须保持复位值。
24	CH3COMCTL[3]	通道 3 输出比较控制 请参考 CH2COMCTL[2:0]描述
23:17	保留	必须保持复位值。
16	CH2COMCTL[3]	通道 2 输出比较控制 请参考 CH2COMCTL[2:0]描述
15	CH3COMCEN	通道 3 输出比较清 0 使能 参见 CH0COMCEN 描述。
14:12	CH3COMCTL[2:0]	通道 3 输出比较模式 参见 CH0COMCTL 描述。
11	CH3COMSEN	通道 3 输出比较影子寄存器使能 参见 CH0COMSEN 描述。

10	CH3COMFEN	通道3输出比较快速使能 参见 CH2COMFEN 描述
9:8	CH3MS[1:0]	通道 3 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 CH3EN 位被清 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器输入被选中时（由 SYSCFG_TIMERxCFG2(x = 1..4)寄存器中的 TSCFG15[4:0]位域选择）。 100~111：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 ETIF 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。 0000：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 0001：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。 0010：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。 0011：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。 0100：强制为低。强制 O2CPRE 为低电平 0101：强制为高。强制 O2CPRE 为高电平 0110：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。 0111：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。 1000：可延时的单脉冲模式0。O2CPRE的输出情况类似与PWM模式0。在向上计数模式时，O2CPRE先输出有效电平，当外部触发事件发生时，立即输出无效电平，当下一次更新事件发生时，再变成有效电平；在向下计数模式时，O2CPRE先输出无效电平，当外部触发事件发生时，立即输出有效电平，当下一次更新事件发生时，再变成无效电平。

1001：可延时的单脉冲模式1。O2CPRE的输出情况类似与PWM模式1。在向上计数模式时，O2CPRE先输出无效电平，当外部触发事件发生时，立即输出有效电平，当下一次更新事件发生时，再变成无效电平；在向下计数模式时，O2CPRE先输出有效电平，当外部触发事件发生时，立即输出无效电平，当下一次更新事件发生时，再变成有效电平。

1010：可编程的脉冲输出。当计数器计数值与输出比较寄存器TIMERx_CH2CV匹配时，CH2_O上输出可编程的脉冲。脉冲由TIMERx_DECCTL寄存器中的OPPSC[2:0]和OPWID[7:0]位域配置。

1011：输出DIR位。O2CPRE信号表示DIR位（在TIMERx_CTL0寄存器中）的值。

1100 / 1101 / 1110 / 1111：保留

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***。

在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或者输出比较模式中从时基模式切换到 PWM 模式时，O2CPRE 电平才改变。

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（TIMERx_CTL0 寄存器的 SPM =1），可以在未确认预装载寄存器情况下使用 PWM 模式。
2	CH2COMFEN	通道2输出比较快速使能 当该位为1时，如果通道配置为PWM模式0或者PWM模式1，会加快捕获 / 比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH2_O被设置为比较电平而与比较结果无关。 0：通道2输出比较快速禁能。当触发器的输入有一个有效沿时，激活CH2_O输出的最小延时为5个时钟周期 1：通道2输出比较快速使能。当触发器的输入有一个有效沿时，激活CH2_O输出的最小延时为 3 个时钟周期
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。 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（由 SYSCFG_TIMERxCFG2(x = 1..4)寄存器中的 TSCFG15[4:0]位域选择）。

100~111：保留

输入捕获模式：

位/位域	名称	描述
31	CH3MS[2]	通道 3 模式选择 与输出模式相同。
30	CH2MS[2]	通道 2 模式选择 与输出模式相同。
29:16	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 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 CI2 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	CH2CAPPSC[1:0]	通道 2 输入捕获预分频器 这 2 位定义了通道 2 输入的预分频系数。当 TIMERx_CHCTL2 寄存器中的 CH2EN =0 时，则预分频器复位。

		00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获 01：每 2 个事件触发一次捕获 10：每 4 个事件触发一次捕获 11：每 8 个事件触发一次捕获
1:0	CH2MS[1:0]	通道 2 模式选择 与输出比较模式相同。

通道控制寄存器 2（TIMERx_CHCTL2）

地址偏移：0x20

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3NP	保留	CH3P	CH3EN	CH2NP	保留	CH2P	CH2EN	CH1NP	保留	CH1P	CH1EN	CH0NP	保留	CH0P	CH0EN
rw		rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	CH3NP	通道 3 互补输出极性 参考 CH0NP 描述。
14	保留	必须保持复位值。
13	CH3P	通道 3 极性 参考 CH0P 描述。
12	CH3EN	通道 3 使能 参考 CH0EN 描述。
11	CH2NP	通道 2 互补输出极性 参考 CH0NP 描述。
10	保留	必须保持复位值。
9	CH2P	通道 2 极性 参考 CH0P 描述。
8	CH2EN	通道 2 使能

		参考 CH0EN 描述。
7	CH1NP	通道 1 互补输出极性 参考 CH0NP 描述。
6	保留	必须保持复位值。
5	CH1P	通道 1 极性 参考 CH0P 描述。
4	CH1EN	通道 1 使能 参考 CH0EN 描述。
3	CH0NP	通道 0 互补输出极性 当通道 0 配置为输出模式，该位必须保持复位值。 当通道 0 配置为输入模式时，此位和 CH0P 联合使用，作为输入信号 CI0 的极性选择控制信号。
2	保留	必须保持复位值。
1	CH0P	通道 0 极性 当通道 0 配置为输出模式时，此位定义了输出信号极性。 0：通道 0 高电平有效 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。
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）（TIMERx, x = 2, 3）

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
UPIFBU	保留														
r															

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	UPIFBU	UPIF位备份 该位只读，是 TIMERx_INTF 寄存器的 UPIF 位的备份值。当 UPIFBUEN = 1 时，该位有效，若 UPIFBUEN =0，该位保留，读取该位值为零。
30:16	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。 当 PWMADMEN = 0 时，该位域表示当前计数器的值。写操作该位域可以改变计数器的值。 当 PWMADMEN = 1 时，该位域仅用于表示计数器值的整数部分，不包含小数部分。

计数器寄存器（TIMERx_CNT）（TIMERx, x = 1, 4）

地址偏移：0x24

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

UPIFBUEN = 0:

位/位域	名称	描述
31:0	CNT[31:0]	这些位是当前的计数值。写操作能改变计数器值。 当 PWMADMEN = 0 时，该位域表示当前计数器的值。写操作该位域可以改变计数器的值。 当PWMADMEN = 1时，该位域仅用于表示计数器值的整数部分，不包含小数部分。

UPIFBUEN = 1:

位/位域	名称	描述
31	UPIFBU	UPIF位备份 该位只读，是TIMERx_INTF寄存器的UPIF位的备份值。当UPIFBUEN = 1时，该位有效，若UPIFBUEN =0，该位保留，读取该位值为零。
30	CNT[30:0]	这些位是当前的计数值。写操作能改变计数器值。 当 PWMADMEN = 0 时，该位域表示当前计数器的值。写操作该位域可以改变计数器的值。 当PWMADMEN = 1时，该位域仅用于表示计数器值的整数部分，不包含小数部分。

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 PSC 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）（TIMERx, x = 2, 3）

地址偏移：0x2C
复位值：0x FFFF 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
CARL[19:16]				保留											
rw															

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

位/位域	名称	描述
31:28	CARL[19:16]	计数器自动重载值（16 位至 19 位） 当 PWMADMEN = 0 时，CARL[19:16]位域值 0000。 当 PWMADMEN = 1 时，CARL[19:16]位域用于表示自动重载值的小数部分。
27:16	保留	必须保持复位值。
15:0	CARL[15:0]	计数器自动重载值（0 位至 15 位） 这些位定义了计数器的自动重载值。 当 PWMADMEN = 0 时，CARL[15:0]位域用于表示计数器的自动重载值。 当 PWMADMEN = 1 时，CARL[15:0]位域用于表示自动重载值的整数部分。

计数器自动重载寄存器（TIMERx_CAR）（TIMERx, x = 1, 4）

地址偏移：0x2C

复位值：0x FFFF 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
CARL[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARL[15:0]															
rw															

位/位域	名称	描述
31:0	CARL[31:0]	计数器自动重载值 这些位定义了计数器的自动重载值。 当 PWMADMEN = 0 时，CARL[31:0]位域用于表示计数器的自动重载值。 当PWMADMEN = 1时，CARL[27:0]位域用于表示自动重载值的整数部分，CARL[31:28]位域用于表示自动重载值的整数部分。

通道 0 捕获/比较值寄存器（TIMERx_CH0CV）（TIMERx, x = 2, 3）

地址偏移：0x34

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH0VAL[19:16]				保留											
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															
rw															

位/位域	名称	描述
31:28	CH0VAL[19:16]	通道 0 的捕获或比较值（16 位到 19 位） 当通道 0 配置为输入模式时，CH0VAL[19:16]位域值为 0000。 当通道 0 配置为输出模式时，该位域包含了即将和计数器比较的值。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。 当 PWMADMEN = 0 时，CH0VAL[19:16]位域值为 0000。 当 PWMADMEN = 1 时，CH0VAL[19:16]用于表示比较值的小数部分。
27:16	保留	必须保持复位值。
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值（0 位至 15 位） 当通道 0 配置为输入模式时，CH0VAL[15:0]决定了上次捕获事件的计数器值，且该位域只读。 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。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。 当 PWMADMEN = 0 时，CH0VAL[15:0]位域值表示比较值。 当 PWMADMEN = 1 时，CH0VAL[15:0]用于表示比较值的整数部分。

通道 0 捕获/比较值寄存器（TIMERx_CH0CV）（TIMERx, x = 1, 4）

地址偏移：0x34

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH0VAL[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															
rw															

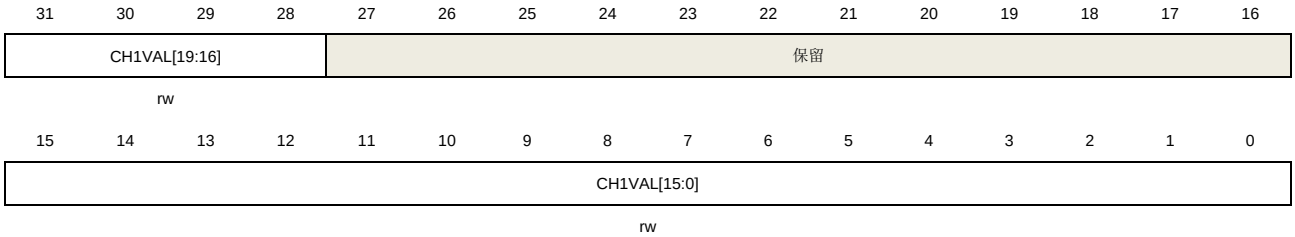
位/位域	名称	描述
31:0	CH0VAL[31:0]	通道 0 的捕获或比较值 当通道 0 配置为输入模式时，CH0VAL[31:28]位域值为 0000，CH0VAL[27:0]位域表示上次捕获事件的计数器值，且该位域只读。 当通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。

当 PWMADMEN = 0 时，CH0VAL[31:0]位域值表示计数器的比较值。
当 PWMADMEN = 1 时，CH0VAL[27:0]位域用于表示比较值的整数部分，CH0VAL[31:28]用于表示比较值的小数部分。

通道 1 捕获/比较值寄存器（TIMERx_CH1CV）（TIMERx, x = 2, 3）

地址偏移：0x38
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

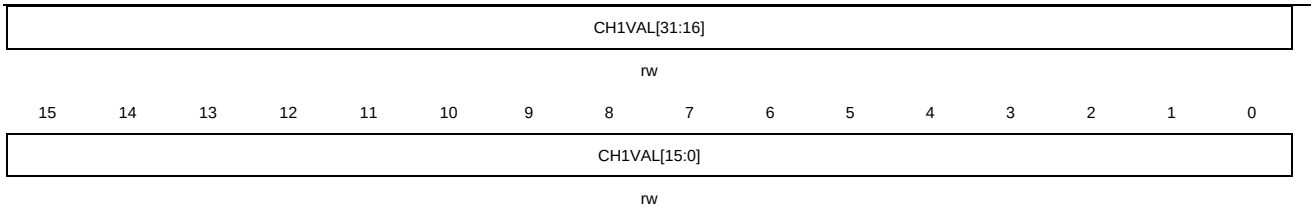


位/位域	名称	描述
31:28	CH1VAL[19:16]	通道 1 的捕获或比较值（16 位到 19 位） 当通道 1 配置为输入模式时，CH1VAL[19:16]位域值为 0000。 当通道 1 配置为输出模式时，该位域包含了即将和计数器比较的值。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。 当 PWMADMEN = 0 时，CH1VAL[19:16]位域值为 0000。 当 PWMADMEN = 1 时，CH1VAL[19:16]用于表示比较值的小数部分。
27:16	保留	必须保持复位值。
15:0	CH1VAL[15:0]	通道 1 的捕获或比较值（0 位至 15 位） 当通道 1 配置为输入模式时，CH1VAL[15:0]决定了上次捕获事件的计数器值，且该位域只读。 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。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。 当 PWMADMEN = 0 时，CH1VAL[15:0]位域值表示比较值。 当 PWMADMEN = 1 时，CH1VAL[15:0]用于表示比较值的整数部分。

通道 1 捕获/比较值寄存器（TIMERx_CH1CV）（TIMERx, x = 1, 4）

地址偏移：0x38
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

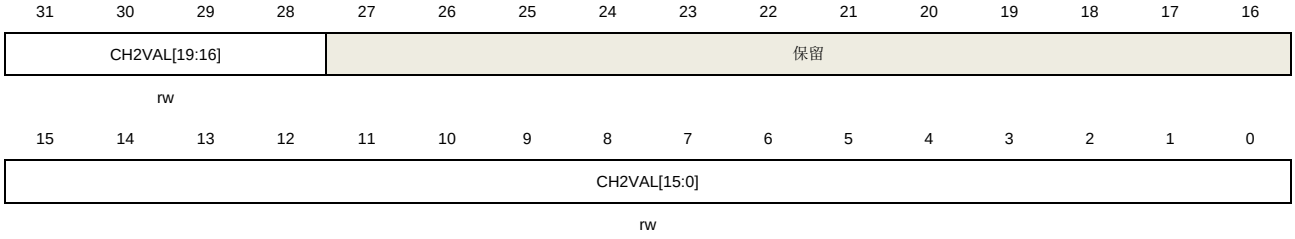


位/位域	名称	描述
31:0	CH1VAL[31:0]	通道 1 的捕获或比较值 当通道 1 配置为输入模式时，CH1VAL[31:28]位域值为 0000，CH1VAL[27:0]位域表示上次捕获事件的计数器值，且该位域只读。 当通道 1 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。 当 PWMADMEN = 0 时，CH1VAL[31:0]位域值表示计数器的比较值。 当 PWMADMEN = 1 时，CH1VAL[27:0]位域用于表示比较值的整数部分，CH1VAL[31:28]用于表示比较值的小数部分。

通道 2 捕获/比较值寄存器（TIMERx_CH2CV）（TIMERx, x = 2, 3）

地址偏移：0x3C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:28	CH2VAL[19:16]	通道 2 的捕获或比较值（16 位到 19 位） 当通道 2 配置为输入模式时，CH2VAL[19:16]位域值为 0000。 当通道 2 配置为输出模式时，该位域包含了即将和计数器比较的值。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。 当 PWMADMEN = 0 时，CH2VAL[19:16]位域值为 0000。 当 PWMADMEN = 1 时，CH2VAL[19:16]用于表示比较值的小数部分。
27:16	保留	必须保持复位值。
15:0	CH2VAL[15:0]	通道 2 的捕获或比较值（0 位至 15 位） 当通道 2 配置为输入模式时，CH2VAL[15:0]决定了上次捕获事件的计数器值，且该

位域只读。

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。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。

当 PWMADMEN = 0 时，CH2VAL[15:0]位域值表示比较值。

当 PWMADMEN = 1 时，CH2VAL[15:0]用于表示比较值的整数部分。

通道 2 捕获/比较值寄存器（TIMERx_CH2CV）（TIMERx, x = 1, 4）

地址偏移：0x3C

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH2VAL[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2VAL[15:0]															
rw															

位/位域	名称	描述
31:0	CH2VAL[31:0]	通道 2 的捕获或比较值 当通道 2 配置为输入模式时，CH2VAL[31:28]位域值为 0000，CH2VAL[27:0]位域表示上次捕获事件的计数器值，且该位域只读。 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。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。 当 PWMADMEN = 0 时，CH2VAL[31:0]位域值表示计数器的比较值。 当 PWMADMEN = 1 时，CH2VAL[27:0]位域用于表示比较值的整数部分，CH2VAL[31:28]用于表示比较值的小数部分。

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

地址偏移：0x40

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3VAL[19:16]				保留											
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3VAL[15:0]															
rw															

位/位域	名称	描述
31:28	CH3VAL[19:16]	通道 3 的捕获或比较值（16 位到 19 位） 当通道 3 配置为输入模式时，CH3VAL[19:16]位域值为 0000。 当通道 3 配置为输出模式时，该位域包含了即将和计数器比较的值。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。 当 PWMADMEN = 0 时，CH3VAL[19:16]位域值为 0000。 当 PWMADMEN = 1 时，CH3VAL[19:16]用于表示比较值的小数部分。
27:16	保留	必须保持复位值。
15:0	CH3VAL[15:0]	通道 3 的捕获或比较值（0 位至 15 位） 当通道 3 配置为输入模式时，CH3VAL[15:0]决定了上次捕获事件的计数器值，且该位域只读。 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。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。 当 PWMADMEN = 0 时，CH3VAL[15:0]位域值表示比较值。 当 PWMADMEN = 1 时，CH3VAL[15:0]用于表示比较值的整数部分。

通道 3 捕获/比较值寄存器（TIMERx_CH3CV）（TIMERx, x = 1, 4）

地址偏移：0x40

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3VAL[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3VAL[15:0]															
rw															

位/位域	名称	描述
31:0	CH3VAL[31:0]	通道 3 的捕获或比较值 当通道 3 配置为输入模式时，CH3VAL[31:28]位域值为 0000，CH3VAL[27:0]位域表示上次捕获事件的计数器值，且该位域只读。 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。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。 当 PWMADMEN = 0 时，CH3VAL[31:0]位域值表示计数器的比较值。 当 PWMADMEN = 1 时，CH3VAL[27:0]位域用于表示比较值的整数部分，CH3VAL[31:28]用于表示比较值的小数部分。

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
CH0COMVAL_ADD[31:16]															
rw															
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	CH0COMVAL_ADD [31:16]	通道0附加比较值（位16到位31） 该位域仅用于TIMER1/ 4。
15:0	CH0COMVAL_ADD [15:0]	通道0附加比较值（位0到位15） 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
CH1COMVAL_ADD[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COMVAL_ADD[15:0]															
rw															

位/位域	名称	描述
31:16	CH1COMVAL_ADD [31:16]	通道1附加比较值（位16到位31） 该位域仅用于TIMER1/ 4。
15:0	CH1COMVAL_ADD [15:0]	通道1附加比较值（位0到位15） 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。

注意：该寄存器仅用于复合PWM模式（当CH0CPWMEN=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
CH2COMVAL_ADD[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2COMVAL_ADD[15:0]															
rw															

位/位域	名称	描述
31:16	CH2COMVAL_ADD [31:16]	通道2附加比较值（位16到位31） 该位域仅用于TIMER1/ 4。
15:0	CH2COMVAL_ADD [15:0]	通道2附加比较值（位0到位15） 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。 **注意：**该寄存器仅用于复合PWM模式（当CH0CPWMEN=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
CH3COMVAL_ADD[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3COMVAL_ADD[15:0]															
rw															

位/位域	名称	描述
31:16	CH3COMVAL_ADD [31:16]	通道3附加比较值（位16到位31） 该位域仅用于TIMER1/ 4。
15:0	CH3COMVAL_ADD	通道3附加比较值（位0到位15）

[15:0]	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。 **注意**：该寄存器仅用于复合PWM模式（当CH0CPWMEN=1时）。

控制寄存器 2（TIMERx_CTL2）

地址偏移：0x74

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3CPWMEN	CH2CPWMEN	CH1CPWMEN	CH0CPWMEN	保留								DECDISDEN	DECJDEN	保留	
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]		保留							
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
27:20	保留	必须保持复位值。
19	DECDISDEN	正交译码器信号断线检测使能 0：正交译码器信号断线检测禁能 1：正交译码器信号断线检测使能
18	DECJDEN	正交译码器信号跳变（两个信号同时发生跳变沿）检测使能 0：正交译码器信号跳变（两个信号同时发生跳变沿）检测禁能

		1：正交译码器信号跳变（两个信号同时发生跳变沿）检测使能
17:16	保留	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。

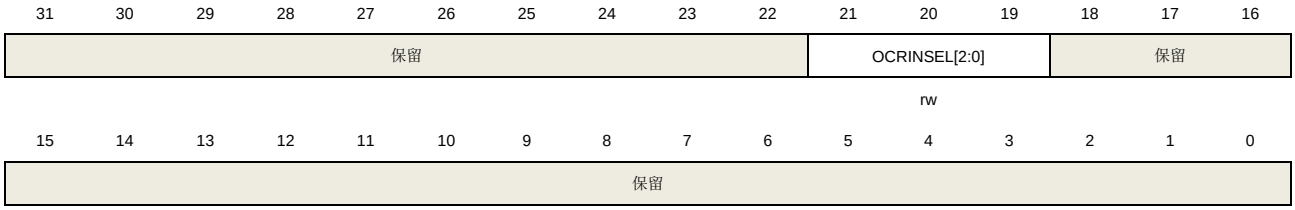
7:0	保留	必须保持复位值。

备用功能控制寄存器 1（TIMERx_AFCTL1）

地址偏移：0x90

复位值：0x0000 0007

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:22	保留	必须保持复位值。
21:19	OCRINSEL[2:0]	OCPRE_CLR输入选择 000：OCPRE_CLR0 001：OCPRE_CLR1 … 111：OCPRE_CLR7

OCPRE_CLR 输入选择	TIMER1 / 2	TIMER3 / 4
OCPRE_CLR0	CMP0_OUT	保留
OCPRE_CLR1	CMP1_OUT	保留
OCPRE_CLR2	CMP2_OUT	保留
OCPRE_CLR3	CMP3_OUT	保留
OCPRE_CLR4	CMP4_OUT	保留
OCPRE_CLR5	CMP5_OUT	保留
OCPRE_CLR6	CMP6_OUT	保留
OCPRE_CLR7	CMP7_OUT	保留

18:0	保留	必须保持复位值。

看门狗计数器周期寄存器（TIMERx_WDGPER）

地址偏移：0x94

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
WDGPER[31:16]															
rw															

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

位/位域	名称	描述
31:0	WDGPER[31:0]	看门狗计数器周期值 这些位用于配置两个看门狗的计数器周期。当看门狗计数器连续计数到该值时，计数器计数超时且中断标志位DECDISIF位置位。若DECDISIE=1，则相应的中断产生。 **注意**：该寄存器位仅用于正交译码器信号断线检测功能（DECDISDEN =1）使能。

译码器控制寄存器（TIMERx_DECCTL）

地址偏移：0xA0

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留					OPPSC[2:0]			OPWID[7:0]							
					rw			rw							

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

位/位域	名称	描述
31:27	保留	必须保持复位值。
26:24	OPPSC[2:0]	输出脉冲预分频 该位域用于配置脉冲发生器的时钟分频系数。 $t_{opclk} = (2^{(OPPSC\ [2:0])}) \times t_{CK_TIMER}$
23:16	OPWID[7:0]	输出脉冲宽度 该位域用于配置脉冲的宽度 $t_{opw} = OPWID\ [7:0] \times t_{opclk}$
15:8	保留	必须保持复位值。
7:6	INDP[1:0]	索引定位 在正交解码器模式0~4中，该位域表示索引事件将在哪种类型的AB输入中复位计数

		器： 00：当AB相输入为00时，索引事件复位计数器 01：当AB相输入为01时，索引事件复位计数器 10：当AB相输入为10时，索引事件复位计数器 11：当AB相输入为11时，索引事件复位计数器 在译码器模式0~3中，该位域表示索引事件将在哪种类型的AB输入中复位计数器： x0：当时钟信号为低电平时，索引事件复位计数器 x1：当时钟信号为高电平时，索引事件复位计数器 **注意**：在译码器模式0~3中，INDP[1]无效。
5	FINDRST	第一个索引信号复位计数器 0：所有的索引信号都可以复位计数器 1：只有第一个索引信号有效，可以复位计数器
4:3	保留	必须保持复位值。
2:1	INDRSTDIR[1:0]	索引信号复位计数器时的计数方向 该位域用于配置计数器的计数方向和索引 00：当计数器向上和向下计数时，索引信号复位计数器 01：仅在计数器向上计数时，索引信号复位计数器 10：仅在计数器向下计数时，索引信号复位计数器 11：保留 注意：仅在INDRSTEN = 0时，可以修改INDRSTDIR[1:0]位域。
0	INDRSTEN	索引信号复位使能 0：索引信号复位计数器禁能 1：索引信号复位计数器使能

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 寄存器的数量。 6'b000000：传输1次 6'b000001：传输2次 … 6'b111000：传输57次
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。 6'b000000：TIMERx_CTL0 6'b000001：TIMERx_CTL1 … 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
DMATB[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMATB[15:0]															
rw															

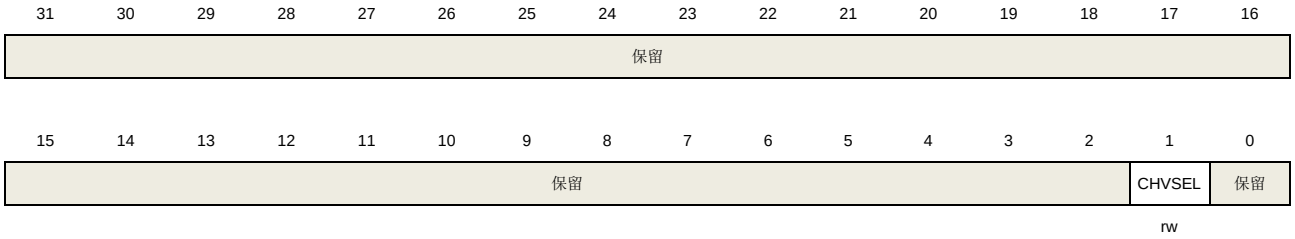
位/位域	名称	描述
31:0	DMATB [31:0]	DMA 发送缓冲 对这个寄存器的读或写，（起始地址+传输次数*4）地址范围内的寄存器会被访问 传输次数由硬件计算，范围为 0 到 DMATC。

配置寄存器（TIMERx_CFG）

地址偏移：0xFC

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:2	保留	必须保持复位值。
1	CHVSEL	写捕获比较寄存器选择位 此位由软件写 1 或清 0。 1：当写入捕获比较寄存器的值与寄存器当前值相等时，写入操作无效 0：无影响
0	保留	必须保持复位值。

23.3. 通用定时器 L3（TIMERx, x = 14）

23.3.1. 简介

通用定时器 L3（TIMER14）是 3 通道定时器，支持输入捕获和输出比较。可以产生 PWM 信号控制电机和电源管理。通用定时器 L3 含有一个 16 位无符号计数器。

通用定时器 L3 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

通用定时器 L3 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，但是他们可以被同步在一起形成一个更大的定时器，这些定时器的计数器一致地增加。

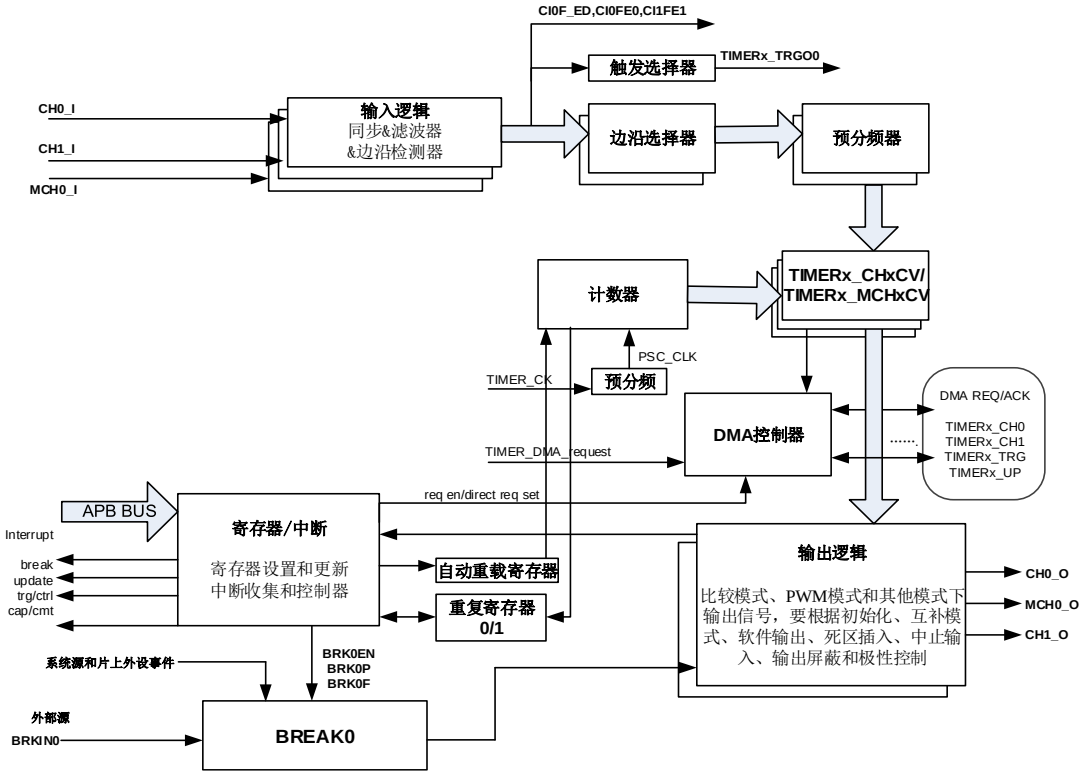
23.3.2. 主要特征

- 总通道数：3；
- 计数器宽度：16位；
- 时钟源可选：内部时钟，内部触发，外部输入；
- 计数模式：向上计数；
- 可编程的预分频器：16位，运行时可以被改变；
- 每个通道可配置：输入捕获模式，输出比较模式，可编程的PWM模式，单脉冲模式；
- 可编程的死区时间；
- 自动重装载功能；
- 可编程的计数器重复功能；
- 中止输入功能：BREAK0；
- 中断输出和DMA请求：更新事件，比较/捕获事件和中止事件；
- 多个定时器的菊链使得一个定时器可以同时启动多个定时器；
- 定时器的同步允许被选择的定时器在同一个时钟周期开始计数；
- 定时器主-从管理。

23.3.3. 结构框图

*图 23-92. 通用定时器 L3 结构框图*提供了通用定时器 L3 的内部配置细节

图 23-92. 通用定时器 L3 结构框图



23.3.4. 功能描述

时钟源选择

通用定时器 L3 可以由内部时钟源 TIMER_CK 或者由 SYSCFG_TIMERxCFG(x = 14)寄存器中的 TSCFGy[4:0] (y = 3..7, 15)位域。

- 当SYSCFG_TIMERxCFG(x = 14)寄存器中的TSCFGy[4:0]=5′b00000(y = 3..7, 15)，定时器选择内部时钟源（连接到RCU模块的CK_TIMER）

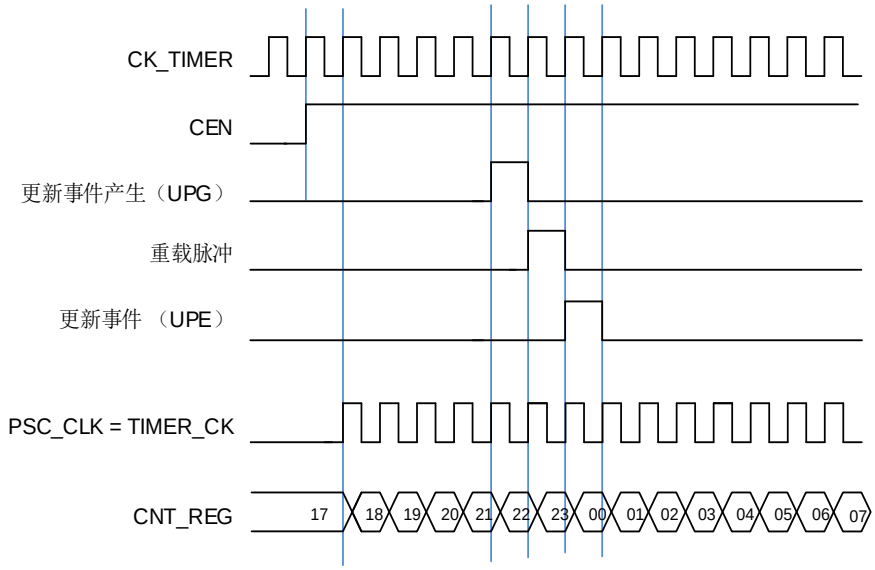
如果 SYSCFG_TIMERxCFG(x = 14)寄存器中的 TSCFGy[4:0] =5′b00000 (y = 3..7, 15)，默认用来驱动计数器预分频器的是内部时钟源 CK_TIMER。当 CEN 置位，CK_TIMER 经过预分频器（预分频值由 TIMERx_PSC 寄存器确定）产生 PSC_CLK。

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 SYSCFG_TIMERxCFG(x = 14)寄存器中的 TSCFG6[4:0]位域设置为非零值，预分频器被其他时钟源驱动，具体在下文说明。当 TSCFGy[4:0] (y = 3, 4, 5, 7)被设置为非零值时，计数器预分

频器时钟源由内部时钟 TIMER_CK 驱动。

图 23-93. 内部时钟分频为 1 时正常模式下的控制电路



- TSCFG6[4:0]设置为非零值（外部时钟模式0），定时器选择外部输入引脚作为时钟源

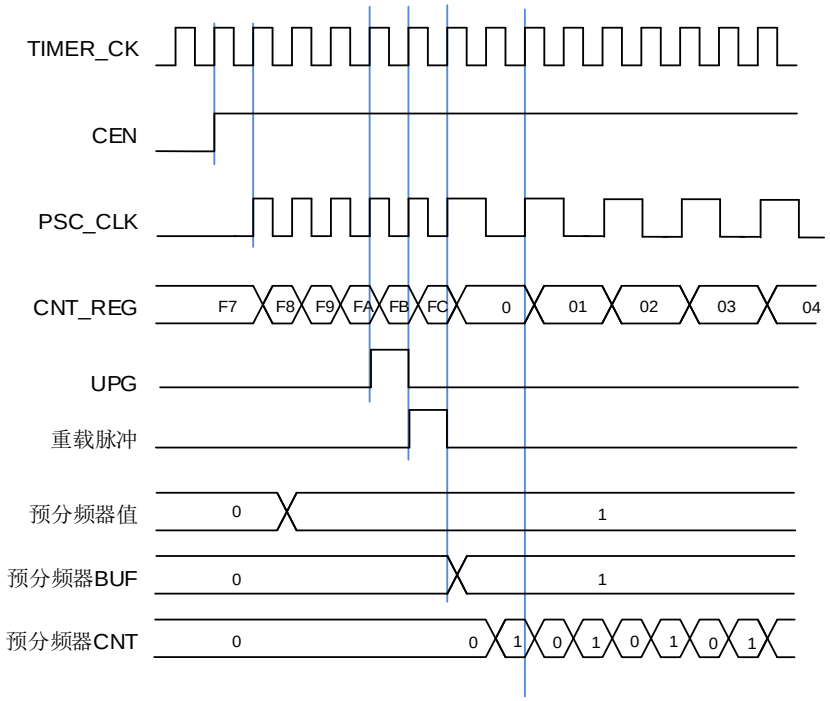
计数器预分频器可以在 CI0 / CI1 引脚的每个上升沿或下降沿计数。这种模式可以通过设置 TSCFG6[4:0]为 0x5 ~ 0x7 来选择。

计数器预分频器也可以在内部触发信号 ITI0 ~ 10 / ITI14 的上升沿计数。这种模式可以通过设置 TSCFG6[4:0]为 0x1 ~ 0x4，0x9 ~ 0xF 和 0x13 来选择。

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 23-94. 当预分频器的参数从 1 变到 2 时，计数器的时序图



向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从 0 开始向上连续计数到自动加载值（定义在 TIMERx_CAR 寄存器中），一旦计数器计数到自动加载值，会重新从 0 开始向上计数。如果设置了重复计数器，在（TIMERx_CREP0/1 + 1）次上溢后产生更新事件，否则在每次上溢时都会产生更新事件。在向上计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 0。

当通过 TIMERx_SWEVG 寄存器的 UPG 位置 1 来设置更新事件时，计数值会被清 0，并产生更新事件。

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

当发生更新事件时，所有的寄存器（重复计数器，自动重载寄存器，预分频寄存器）都将被更新。

图 23-95. 向上计数时序图,PSC=0 / 2 和 *图 23-96. 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 23-95. 向上计数时序图，PSC=0 / 2

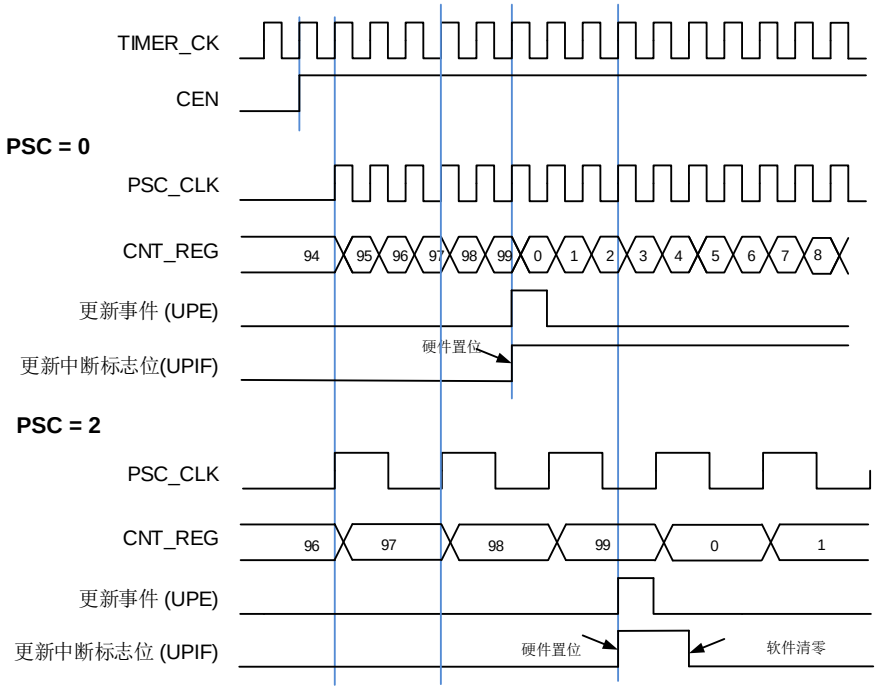
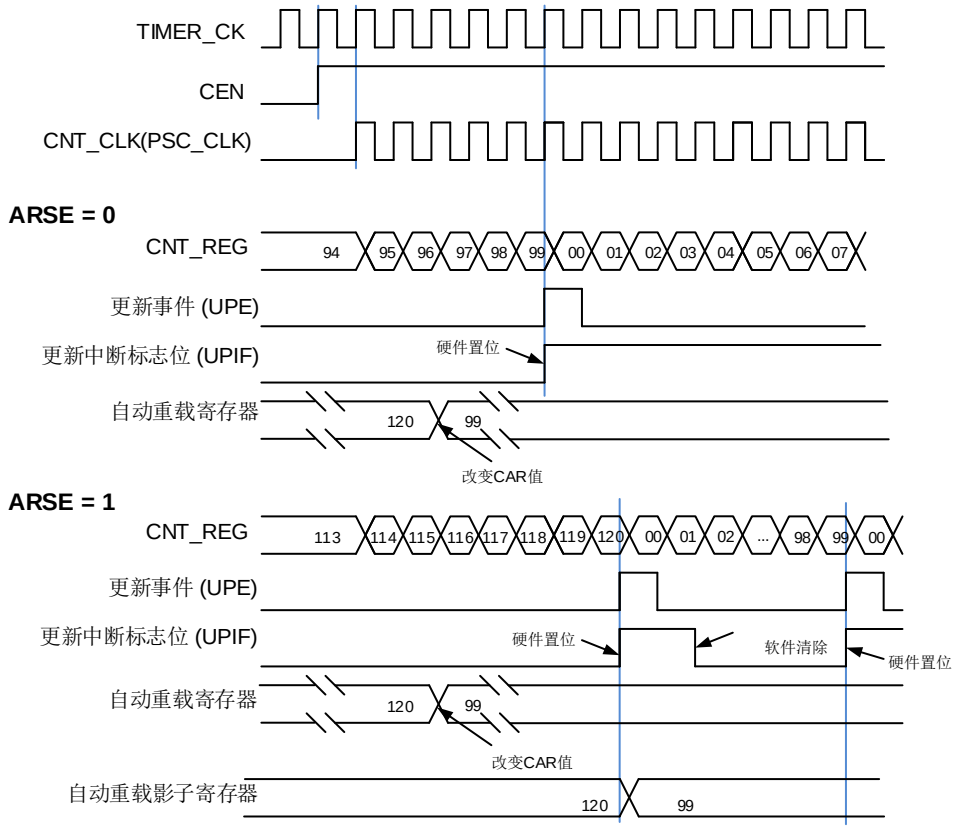


图 23-96. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



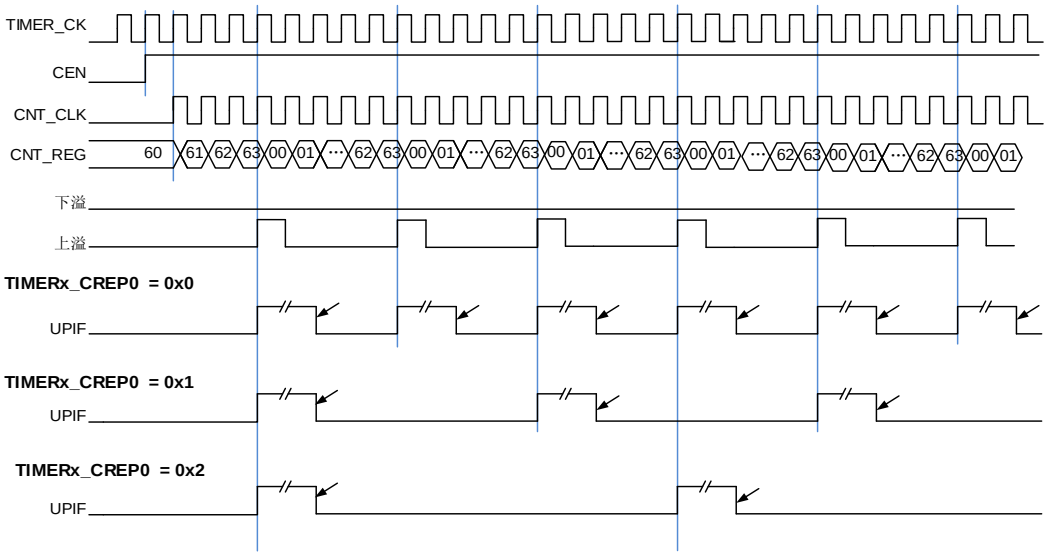
重复计数器

通用L3定时器有两个重复寄存器TIMERx_CREP0 / 1，可通过配置TIMERx_CFG寄存器中的CREPSEL位来选择。其中TIMERx_CREP0寄存器中的CREP0[7:0]是8位的，TIMERx_CREP1寄存器中的CREP1[31:0]是32位，用户可根据需求选择使用。

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_CREP0/1 寄存器的 CREP0/1。向上计数模式下，重复计数器在每次计数器上溢时递减。

将 TIMERx_SWEVG 寄存器的 UPG 位置 1 可以重载 TIMERx_CREP0/1 寄存器中 CREP0/1 的值并产生一个更新事件。

图 23-97. 在向上计数模式下计数器重复时序图



捕获/比较通道

通用定时器 L3 拥有 3 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 可输出独立和互补。

■ 输入捕获模式

当 MCHxMSEL=2'b00（独立模式）时，通道 x 和多模式通道 x 才可以独立进行输入捕获。

捕获模式允许通道测量一个波形时序，频率，周期，占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，TIMERx_CHxCV / TIMERx_MCHxCV（x = 0, 1）寄存器会捕获计数器当前的值，同时 CHxIF / MCHxIF（x = 0,1）位置 1，如果 CHxIE/ MCHxIE =1（x = 0, 1），则产生相应的通道中断。

图 23-98. 通道 0 输入捕获逻辑

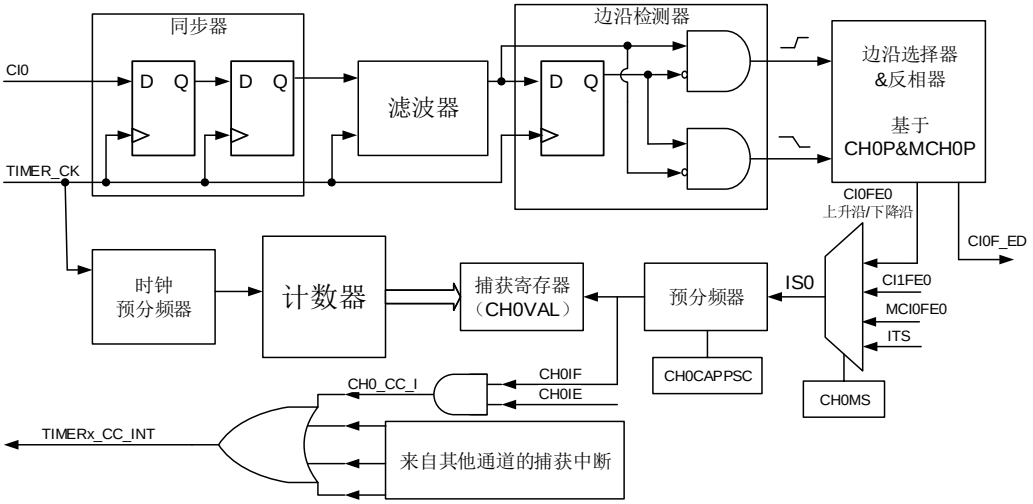
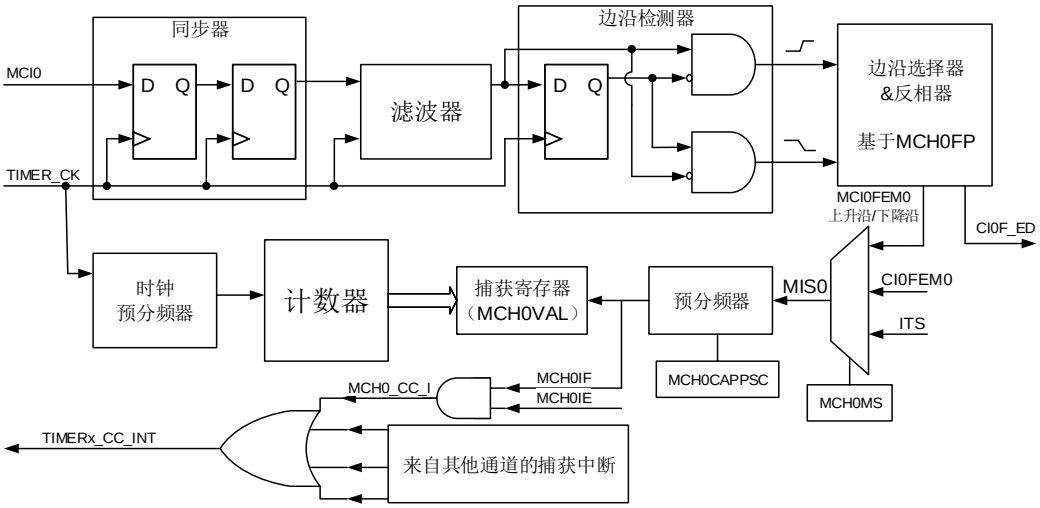


图 23-99. 多模式通道 0 输入捕获逻辑



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。

输入捕获模式也可用来测量 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 23-100. 输出比较逻辑（当 MCHxMSEL = 2'b00 时，x = 0），*图 23-101. 输出比较逻辑（当 MCHxMSEL = 2'b11 时，x = 0）*和*图 23-102. 输出比较逻辑（x = 1）*给出了通道的输出比较逻辑。

图 23-100. 输出比较逻辑（当 MCHxMSEL = 2'b00 时，x = 0）

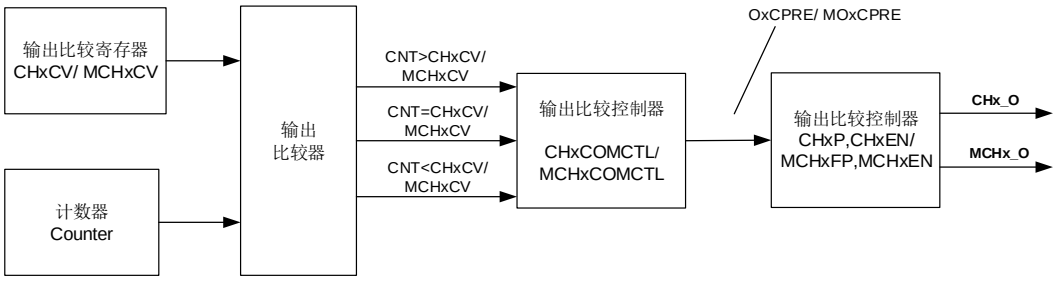


图 23-101. 输出比较逻辑（当 MCHxMSEL = 2’b11 时，x = 0）

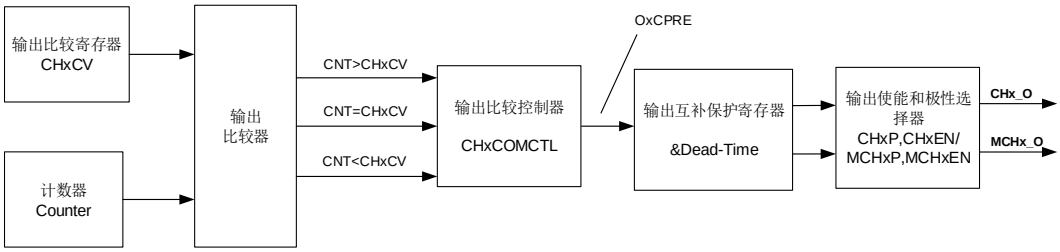
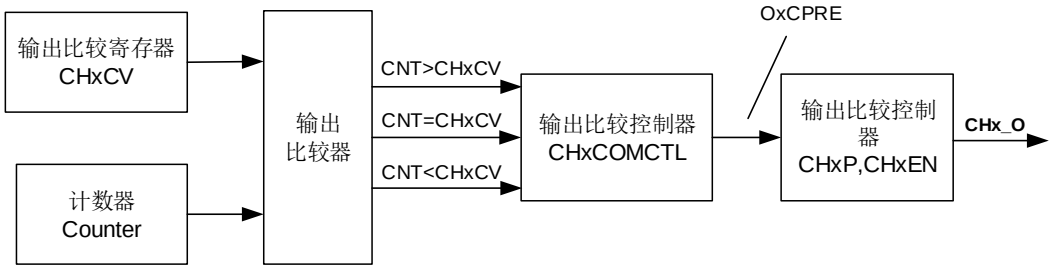


图 23-102. 输出比较逻辑（x = 1）



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和TIMERx_MCHCTL2寄存器）。请参考*图23-100. 输出比较逻辑（当MCHxMSEL = 2’b00时，x = 0）*。
- 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23-101. 输出比较逻辑（当MCHxMSEL = 2’b11时，x = 0）*。

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_CCHP0寄存器中的相关位（ROS、IOS、POE和DTCFG等位）有关。详情请见*互补输出*。

在输出比较模式，TIMERx 可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的 TIMERx_CHxCV / 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。

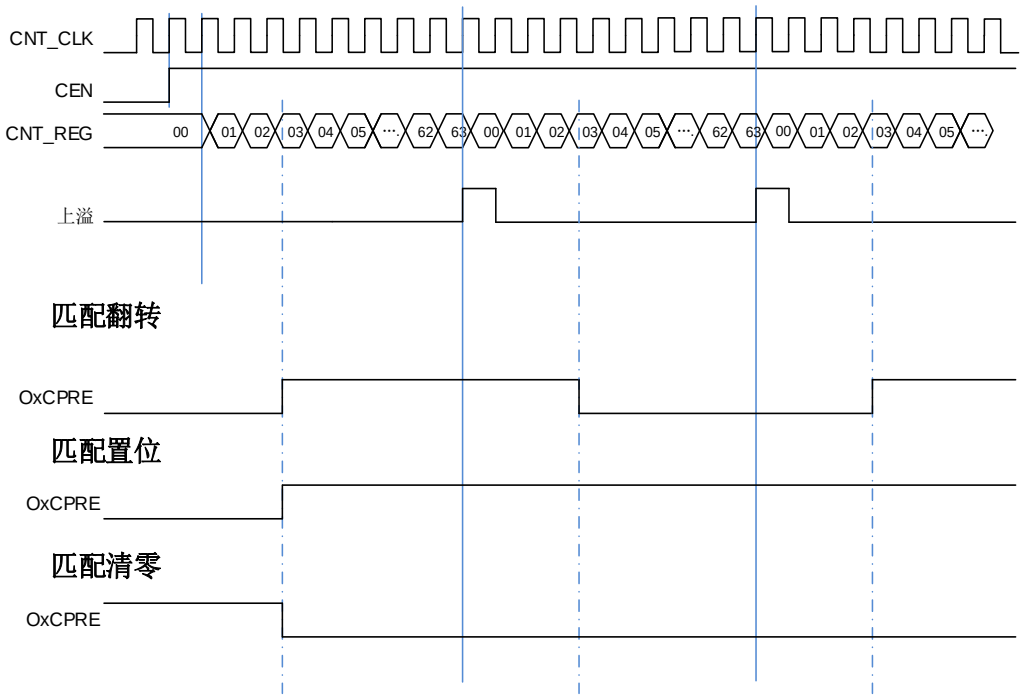
第四步：通过TIMERx_CAR寄存器和TIMERx_CHxCV寄存器配置输出比较时基：

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23-103. 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 23-103. 三种输出比较模式



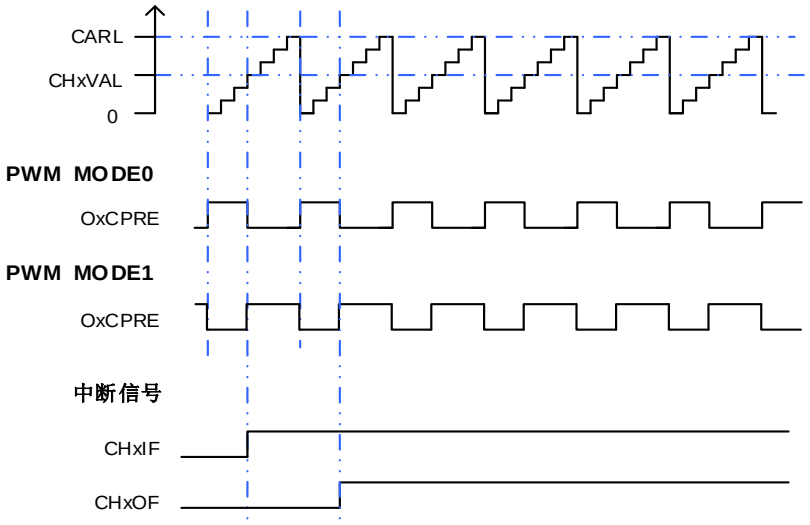
PWM 模式

在 PWM 输出模式下（PWM 模式 0 是配置 CHxCOMCTL/ MCHxCOMCTL 为 4'b0110，PWM 模式 1 是配置 CHxCOMCTL/ MCHxCOMCTL 为 4'b0111），通道根据 TIMERx_CAR 寄存器和 TIMERx_CHxCV/ TIMERx_MCHxCV 寄存器的值，输出 PWM 波形。

EAPWM 的周期由 TIMERx_CAR 寄存器值决定，占空比由 TIMERx_CHxCV/ TIMERx_MCHxCV 寄存器值决定。*图 23-104. PWM 时序图*显示了 EAPWM 的输出波形和中断。

当计数器向上计数时，在PWM0模式下（CHxCOMCTL/ MCHxCOMCTL =4'b0110），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=4'b0111），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 23-104. PWM 时序图

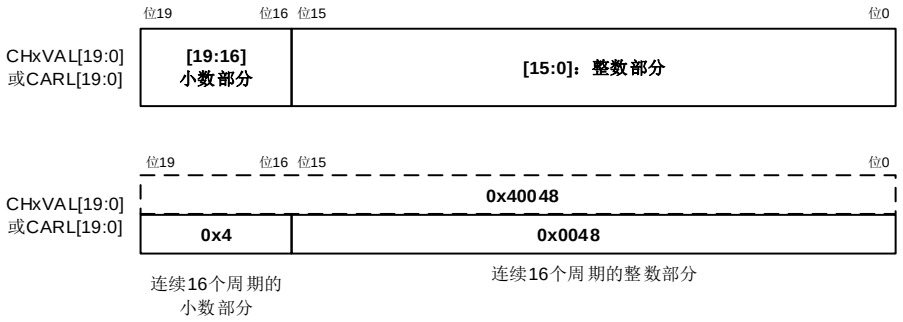


微调模式

通过配置TIMERx_CTL0寄存器中的ADMEN位为1，可以使能微调模式。该模式可以提高输出PWM波的有效分辨率，通过TIMERx_CHxCV寄存器中的CHxVAL[19:0]位域可以提高占空比分辨率，通过TIMERx_CAR寄存器中的CARL[19:0]位域可以提高PWM频率的分辨率。

当微调模式使能时，CHxVAL位域和CARL位域的低16位[15:0]用于整数部分，高4位[19:16]用于微调的小数部分。通过预定义的方式，在连续16个周期内对CHxVAL值或CARL值进行微调（每次调整不超过一个TIMER时钟周期），可增加16倍的分辨率。

图 23-105. 微调模式：数据格式和寄存器位域



根据ADMEN位的配置（置位或清零），CHxVAL位域和CARL位域将自动更新。当需要对ADMEN位进行清零时，需要遵循以下步骤：

1. CEN位和ARSE位必须清零；
2. CARL[19:16]位域必须清零；
3. ADMEN位必须清零；

4. CHxIF位必须清零；
5. 可以将CEN位置1。

以下公式可以计算PWM分辨率：

$$Resolution = f_{PSC_CLK} \ / \ f_{pwm} \quad (23\text{-}9)$$

由式*(23-9)*可得，微调模式禁能时（ADMEN=0），PWM的最小频率f_{pwm}:

$$(f_{pwm})_{min} = f_{PSC_CLK} \ / \ 65536 \quad (23\text{-}10)$$

微调模式使能时（ADMEN=1），

$$(f_{pwm})_{min} = f_{PSC_CLK} \ / \ (65535 + 15 \ / \ 16) \quad (23\text{-}11)$$

当微调模式使能时，CHxVAL[19:0]位域和CARL[19:0]位域的最大值为0xFFFFE（整数部分为0xFFFE，小数部分为0xF）。

在连续 16 个周期内，占空比和周期的变化情况，具体如*图 23-106. PWM 微调模式原理*和*表 23-17. 边沿对齐模式中 CHxVAL 和 CARL 位域的变化*所示。

图 23-106. PWM 微调模式原理

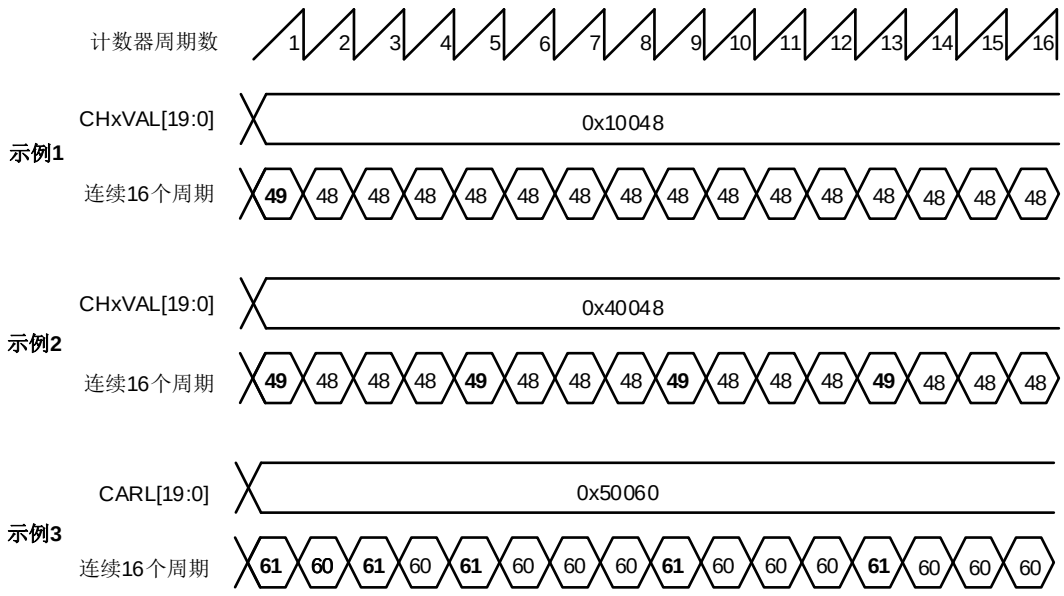


表 23-17. 边沿对齐模式中 CHxVAL 和 CARL 位域的变化

CHxVAL[19:16] / CARL[19:16]	周期															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

CHxVAL[19:16] / CARL[19:16]	周期															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0010	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-
0011	+1	-	-	-	+1	-	-	-	+1	-	-	-	-	-	-	-
0100	+1	-	-	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0101	+1	-	+1	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0110	+1	-	+1	-	+1	-	-	-	+1	-	+1	-	+1	-	-	-
0111	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	-	-
1000	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1001	+1	+1	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1010	+1	+1	+1	-	+1	-	+1	-	+1	+1	+1	-	+1	-	+1	-
1011	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	-	+1	-
1100	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1101	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1110	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	+1	+1	+1	+1	-
1111	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	-

复合 PWM 模式

在复合 PWM 模式中（CHxCPWMEN = 1’b1，CHxMS[2:0] = 3’b000 和 CHxCOMCTL = 4’b0110、4’b0111），通道 x（x = 0, 1））上的 PWM 输出信号由 CHxVAL 和 CHxCOMVAL_ADD 位确定。

如果 CHxCOMCTL = 4’b0110（PWM 模式 0）且 DIR = 1’b0（向上计数模式），或者 CHxCOMCTL = 4’b0111（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 =4’b0111（PWM 模式 1）且 DIR = 1’b0（向上计数模式），或者 CHxCOMCTL =4’b0110（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 23-18. 复合 PWM 脉冲宽度*计算。

表 23-18. 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%

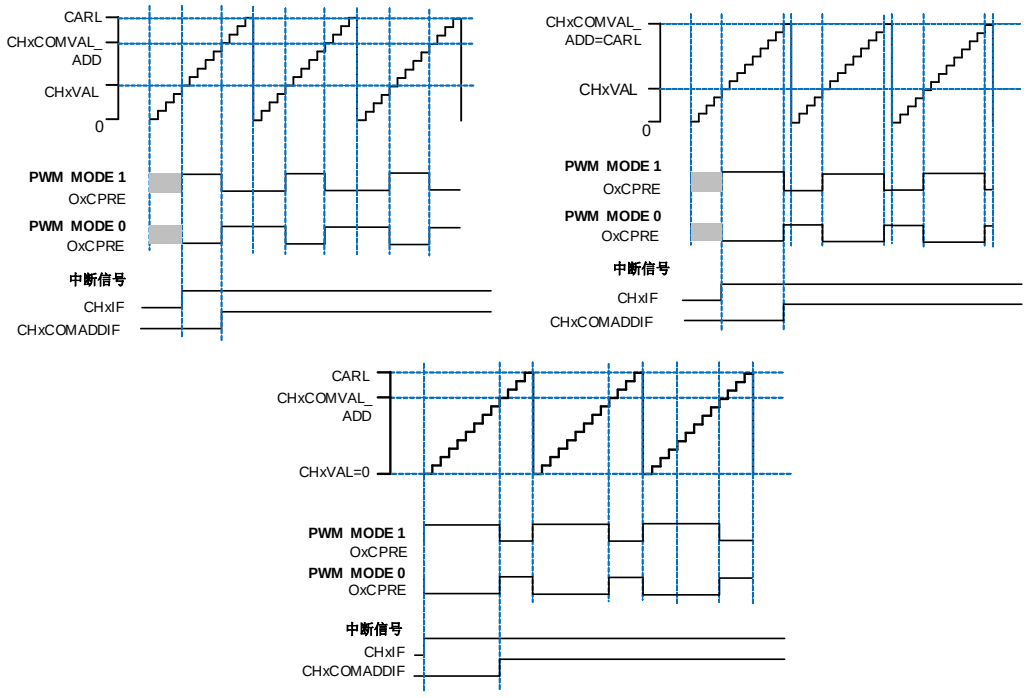
条件	模式	PWM 脉冲宽度
CARL）或（CHxVAL > CARL > CHxCOMVAL_ADD）	或 PWM 模式 1（向下计数）	
	PWM 模式 0（向下计数）或 PWM 模式 1（向上计数）	0%
CHxCOMVAL_ADD > CARL > CHxVAL	PWM 模式 0（向上计数）或 PWM 模式 1（向下计数）	0%
	PWM 模式 0（向下计数）或 PWM 模式 1（向上计数）	100%
（CHxVAL>CARL）且（CHxCOMVAL_ADD > CARL）	-	CHx_O 输出保持

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：

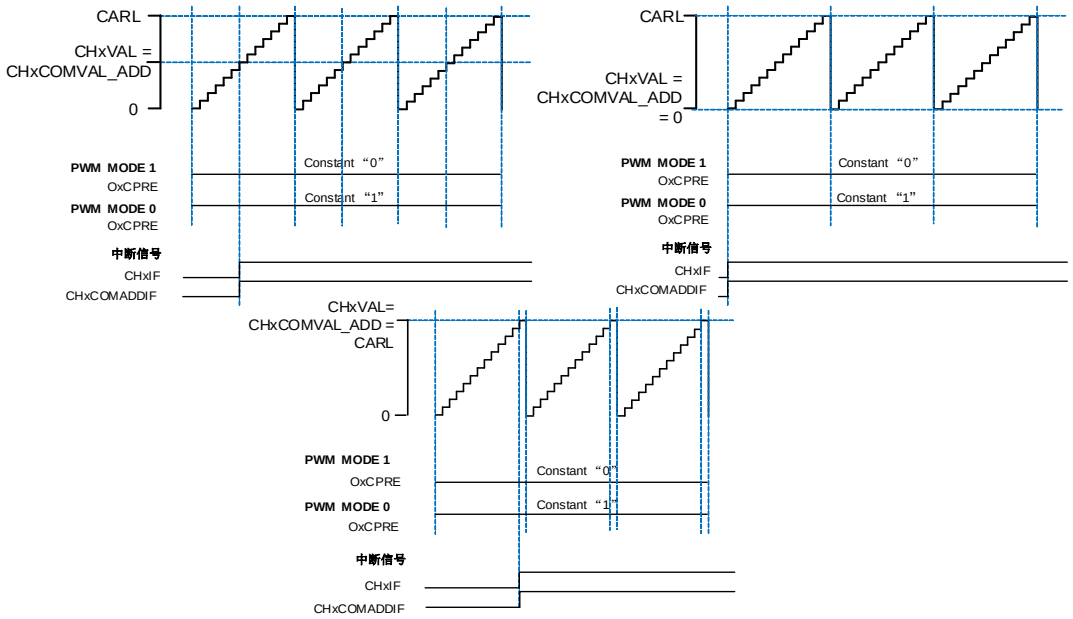
- CHxVAL < CHxCOMVAL_ADD，CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 23-107 通道 x 输出 PWM（CHxVAL < CHxCOMVAL_ADD）



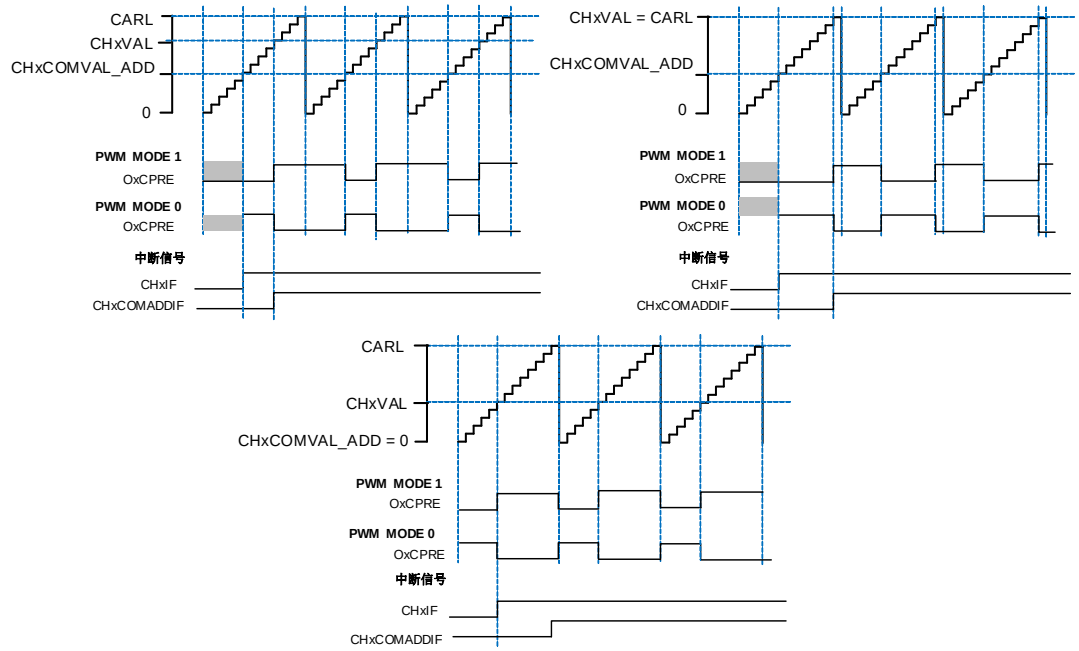
- CHxVAL = CHxCOMVAL_ADD, CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 23-108. 通道 x 输出 PWM（CHxVAL = CHxCOMVAL_ADD）



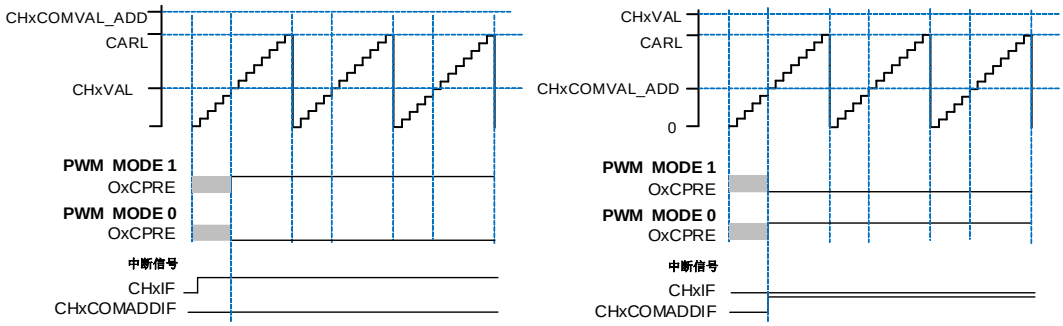
- CHxVAL > CHxCOMVAL_ADD, CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 23-109. 通道 x 输出 PWM（CHxVAL > CHxCOMVAL_ADD）



- CHxVAL或CHxCOMVAL_ADD值大于CARL。

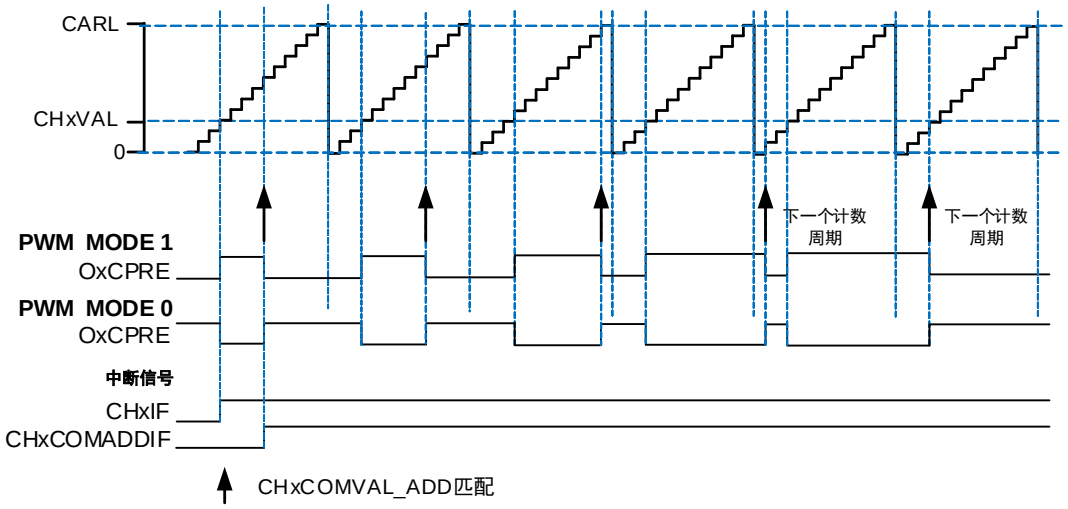
图 23-110. 通道 x 输出 PWM（CHxVAL 或 CHxCOMVAL_ADD > CARL）



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23-111. 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 23-111. 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。

输出匹配脉冲选择

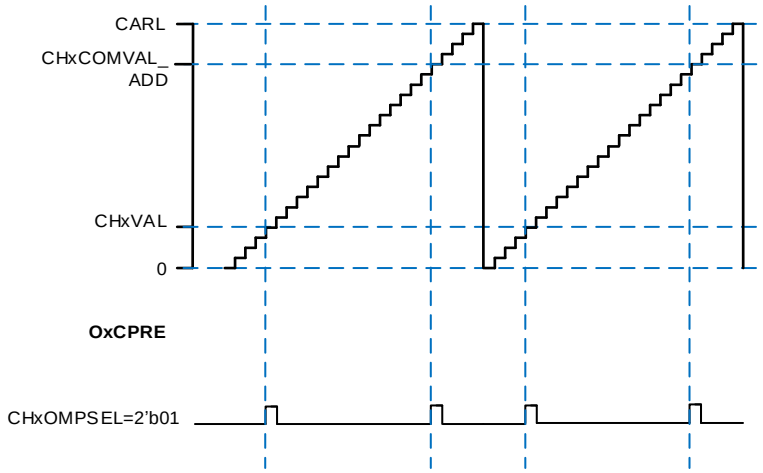
当发生匹配事件时，CHx_O（x = 0, 1）的输出由CHxCOMCTL[3:0]（x = 0, 1）位设置，通过配置CHxOMPSEL[1:0]（x = 0, 1）位，可选择CHx_O（x = 0, 1）的输出信号正常或者脉冲。

当匹配事件发生时，CHxOMPSEL[1:0]（x = 0, 1）用于选择OxCPRE信号输出（驱动CHx_O）：

- CHxOMPSEL = 2’b00，OxCPRE信号根据CHxCOMCTL[3:0]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；

图 23-112. 边沿对齐模式下 CHx_O 输出脉冲（CHxOMPSEL = 2'b01）



通道输出准备信号

如*图 23-100. 输出比较逻辑（当 MCHxMSEL = 2'b00 时，x = 0）*，*图 23-101. 输出比较逻辑（当 MCHxMSEL = 2'b11 时，x = 0）*和*图 23-102. 输出比较逻辑（x = 1）*所示，当 TIMERx 用于输出匹配比较模式下，在通道输出信号之前将产生一个中间信号，即 OxCPRE 或 MOxCPRE 信号（通道 x 或多模式通道 x 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 ETIFP 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。

清除通道输出准备信号

当CHxCOMCEN位或MCHxCOMCEN位（在TIMERx_CHCTLy / TIMERx_MCHCTLy寄存器中）

置 1 时， OxCPRE 和 MOxCPRE 信号可以由 OCPRE_CLR_INT 信号清除。该功能用于 CHxCOMCTL[3:0]位域或MCHxCOMCTL[3:0]位域（4'b0100和4'b0101除外）中配置的比较输出模式。

OCPRE_CLR_INT 连接到 OCPRE_CLR 输入。OxCPRE /MOxCPRE 信号被 OCPRE_CLR_INT 信号的高电平清除，直到下一个更新事件发生时才会恢复输出。在 TIMERx_AFCTL1 寄存器的 OCRINSEL[2:0]位域中选择 OCPRE_CLR 的输入。

互补输出

CHx_O 和 MCHx_O 的输出具有两种情况：

- MCHxMSEL = 2'b00：MCHx_O输出独立于CHx_O输出。
- MCHxMSEL = 2'b11：MCHx_O 输出与 CHx_O 输出互补，且 MCHx_O 的输出不由 MCHxCOMCTL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 的 1 对通道具有此功能。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_CCHP0 寄存器中和 TIMERx_CTL1 寄存器中的 POEN、ROS、IOS、ISOx 和 ISOxN 位。输出极性由 TIMERx_CHCTL2 寄存器中的 CHxP 和 MCHxP 位来决定。

当 CHx_O 和 MCHx_O 的输出互补时，有三种输出情况：输出使能、输出关闭状态和输出禁能，具体情况可参考*表 23-19. 由参数控制的互补输出表（MCHxMSEL=2'b11）*。

表 23-19. 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="3">0</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O / MCHx_O = LOW
CHx_O / MCHx_O 输出禁能(1)</td></tr>
<tr><td>1</td><td colspan="2" rowspan="3">CHx_O/MCHx_O输出关闭状态(2)：
通道先输出无效电平：CHx_O = CHxP，MCHx_O = CHxNP）；如果死区产生时钟未失效，在死区时间之后：CHx_O = ISOx，MCHx_O = 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/MCHx_O输出关闭状态：
通道先输出无效电平：CHx_O = CHxP，MCHx_O = CHxNP）；如果死区产生时钟未失效，在死区时间之后：CHx_O = ISOx，MCHx_O = ISOxN</td></tr>
<tr><td rowspan="3">1</td><td rowspan="3">0</td><td rowspan="3">0/1</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O/MCHx_O = LOW
CHx_O/MCHx_O输出禁能</td></tr>
<tr><td>1</td><td>CHx_O = LOW
CHx_O输出禁能</td><td>MCHx_O=OxCPRE⊕(2)MCHxP
MCHx_O输出使能</td></tr>
<tr><td>1</td><td>0</td><td>CHx_O=OxCPRE⊕CHxP</td><td>MCHx_O = LOW</td></tr>
</table>

互补参数					输出状态	
POEN	ROS	IOS	CHxEN	MCHxEN	CHx_O	MCHx_O
					CHx_O输出使能	MCHx_O输出禁能
				1	CHx_O=OxCPRE⊕CHxP CHx_O输出使能	MCHx_O=(!OxCPRE)(3)⊕ MCHxP MCHx_O输出使能
	1		0	0	CHx_O = CHxP CHx_O输出关闭状态	MCHx_O = MCHxP MCHx_O输出关闭状态
				1	CHx_O = CHxP CHx_O输出关闭状态	MCHx_O=OxCPRE⊕MCHxP MCHx_O输出使能
			1	0	CHx_O=OxCPRE⊕CHxP CHx_O输出使能	MCHx_O = MCHxP MCHx_O输出关闭状态
				1	CHx_O=OxCPRE⊕CHxP CHx_O输出使能	MCHx_O=(!OxCPRE)⊕ MCHxP MCHx_O输出使能

注意：

（1） 输出禁能：CHx_O / MCHx_O 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 / MCHx_O 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。

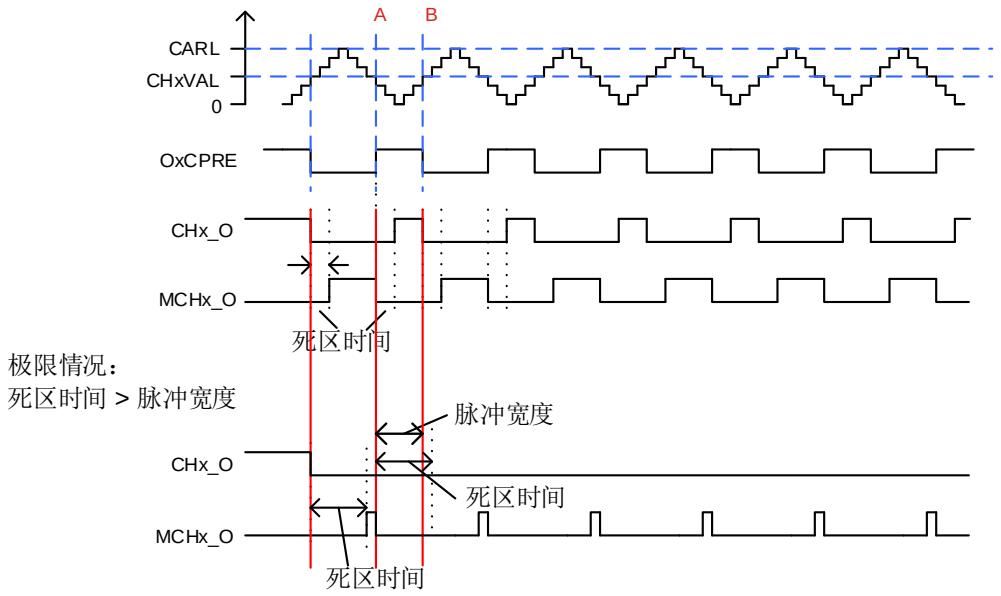
死区时间插入

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 0（TIMERx_CCHP0）*。

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 23-113. 带死区时间的互补输出*中的 A 点，CHx_O 信号在死区时间内为低电平，直到死区时间过后才变为高电平，而 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 23-113. 带死区时间的互补输出*所示。

图 23-113. 带死区时间的互补输出



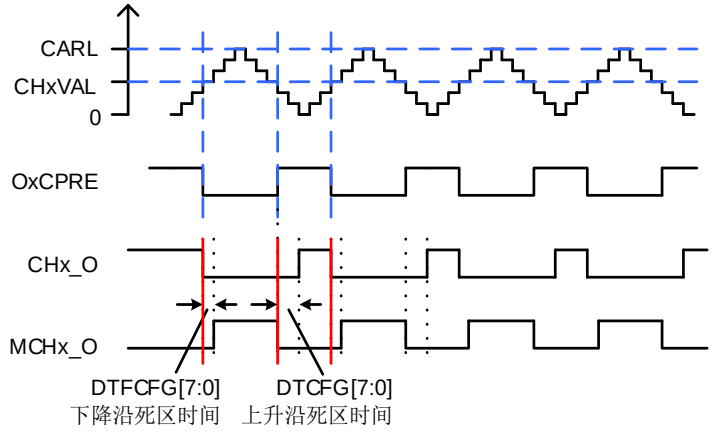
不同的死区时间插入

当DTDIFEN位（在TIMERx_CCHP1寄存器中）设置为1时，CHx_O和MCHx_O信号可以输出不同的死区时间，具体如*图23-114.不同死区时间的互补输出（DTDIFEN=1）*所示。

通道输出准备信号OxCPRE上升沿的死区时间由TIMERx_CCHP0寄存器或TIMERx_FCCHPy寄存器中的DTCFG[7:0]位域配置。OxCPRE信号的下降沿的死区时间由TIMERx_CCHP1寄存器或TIMERx_FCCHPy寄存器中的DTFCFG[7:0]位域配置。

可以在CHx_O和MCHx_O信号输出时修改死区时间。当TIMERx_CCHP1寄存器中的DTMODEN位置1时，可以使能该功能。DTCFG[7:0]位域和DTFCFG[7:0]位域的新值将会在下一次更新事件发生时生效。

图 23-114.不同死区时间的互补输出（DTDIFEN=1）



中止功能

当 MCHxMSEL = 2'b11（MCHx_O 的输出不使用 MCHxCOMCTL 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。

通用 L3 定时器具有 BREAK0 中止功能。可以通过将 TIMERx_CCHP0 寄存器中的 BRK0EN 位置 1 来使能。中止输入极性由 TIMERx_CCHP0 寄存器中的 BRK0P 位配置，电平有效。

使用 BREAK0 功能时，CHx_O 和 MCHx_O 信号的输出电平由以下位控制：TIMERx_CCHP0 寄存器的 POEN、IOS 和 ROS 位，TIMERx_CTL1 寄存器的 ISOx 和 ISOxN 位。

中止事件是所有源逻辑或运算的结果。中止功能可以处理三种类型的事件源：

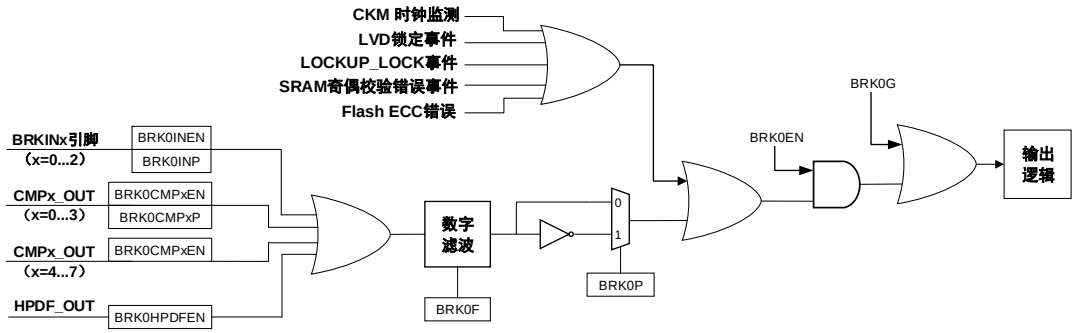
- 外部信号源：来自BRKIN0输入；
- 系统源：由RCU中的时钟监视器CKM生成的HXTAL卡住事件、LVD锁定事件，Cortex®-M33锁定事件、SRAM奇偶校验错误或FLASH ECC错误事件；
- 片上外设源：比较器输出、HPDF的看门狗输出。

BREAK0中止事件也可以由软件置位TIMERx_SWEVG寄存器中的BRK0G位产生。

如*所示，BRKIN0 可以从 TRIGSEL 模块选择 GPIO 引脚，具体可参考 TIMER14_BRKIN 触发选择寄存器（TRIGSEL_TIMER14BRKIN）。*

图 23-115. BREAK0 中止功能逻辑图所示，BRKIN0 可以从 TRIGSEL 模块选择 GPIO 引脚，具体可参考 *TIMER14_BRKIN 触发选择寄存器（TRIGSEL_TIMER14BRKIN）*。

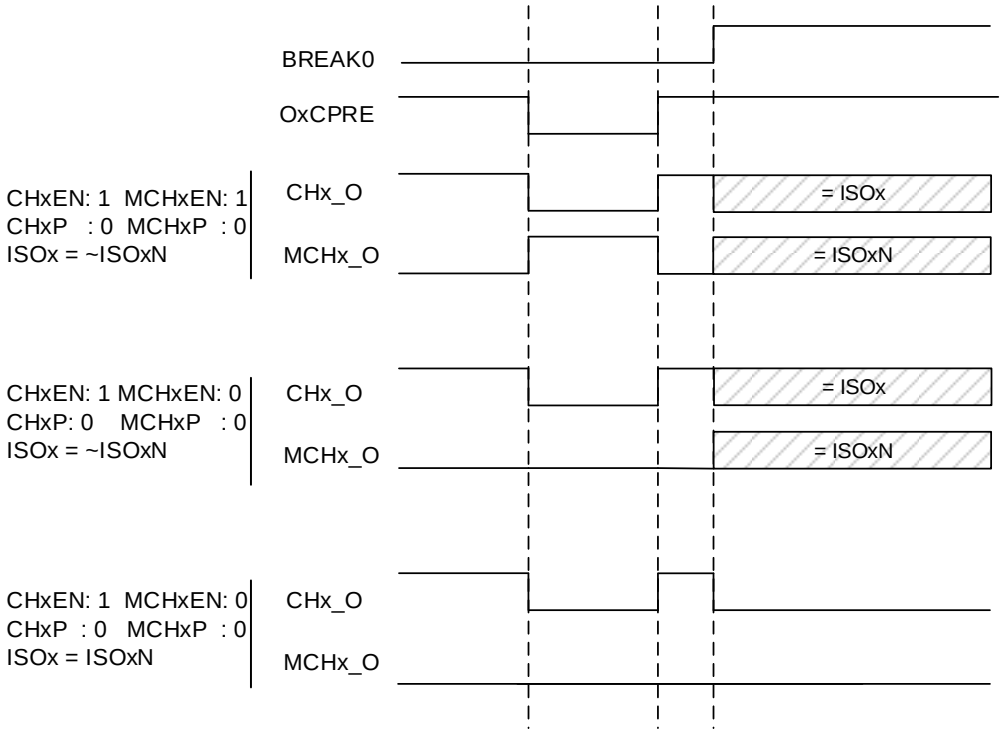
图 23-115. BREAK0 中止功能逻辑图



BREAK0可用于处理系统源、片上外设和外部输入信号源的故障，当发生BREAK0中止事件时，输出强制为无效电平，或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。

当 MCHxMSEL = 2'b11 且发生 BREAK0 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。当 IOS=1 时，通道输出情况如*图 23-116. 通道响应 BREAK0 中止输入（高电平有效）时，输出信号的行为（IOS=1）*所示，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。

图 23-116. 通道响应 BREAK0 中止输入（高电平有效）时，输出信号的行为（IOS=1）



发生中止事件时，TIMERx_INTF 寄存器的 BRK0IF/BRK1IF 位被置 1。如果 BRKIE=1，中断产生。

锁存中止功能

高级定时器的中止输入引脚BRKIN0具有锁存中止功能，可通过设置TIMERx_CCHP0寄存器中的BRK0LK位为1，将相应的BRKIN0配置为锁存中止功能。

当使能了锁存中止功能时，需要将 BRKIN0 引脚设置为开漏模式，且低电平有效（BRK0P =0，BRK0IN0P=0）。任何中止源请求发生时，都可以将相应的 BRKIN0 引脚强制为低电平。若 BRKIN0 引脚设置为高电平有效（BRK0 =1，BRK0IN0P=1），则锁存中止功能被禁止。

当中止功能使能（将TIMERx_CCHP0寄存器中的BRK0EN=1）时，通过软件将TIMERx_SWEVG寄存器中的BRK0G位置1也可以将BRKIN0引脚强制为低电平。

当中止功能未使能（将TIMERx_CCHP0寄存器中的BRK0EN位为0）时，通过软件将BRK0G位置1，对BRKIN0引脚无影响。但BRK0F标志位会置位，通道输出为安全状态。

将 TIMERx_CCHP0 寄存器中的 BRK0REL 位置 1，可以释放 BRKIN0 引脚，当中止输入源无效时，BRK0REL 位由硬件清零，BRKIN0 引脚将恢复锁存中止功能。

在下面两种情况下，不能释放中止输入引脚 BRKIN0：

- 中止输入源有效：虽然BRK0REL位置1，释放了BRKIN0引脚，但由于中止源仍然存在，故中止事件仍然有效；
- POEN=1：通道输出使能时，即使BRK0REL位置1，也不能释放BRKIN0引脚。

表 23-20. 中止功能锁存/释放条件

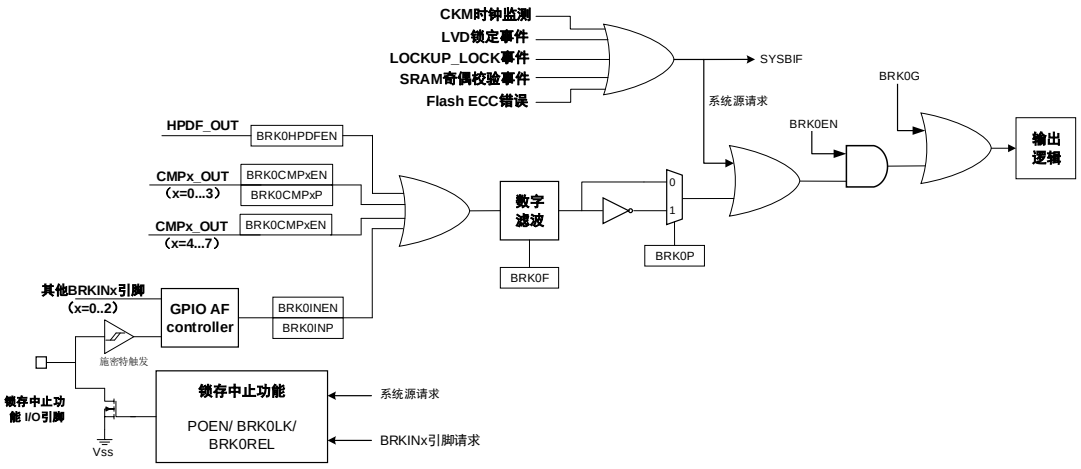
POEN	BRK0LK	BRK0REL	中止输入引脚状态
0	1	0	锁存
	1	1	释放

BREAK0 输入引脚 BRKIN0 的锁存中止功能默认是使能的（BRK0REL=0），当 BREAK0 中止事件发生时，可以通过下面的方法来重新配置锁存中止功能：

- BRK0REL=1，释放BRKIN0引脚；
- 软件等待系统中止源无效，可通过软件清除SYSBIF标志位；
- 软件轮询BRK0REL位，直到BRK0REL=0（硬件实现）。

上述过程完成后，BREAK0 锁存中止功能重新使能，此时，可通过软件将 POEN 置 1 来恢复 PWM 输出。

图 23-117. BREAK0 的 BRKIN0 引脚锁存功能逻辑图

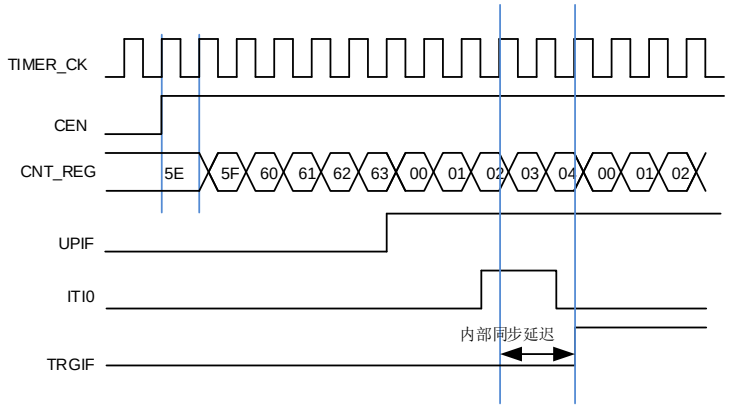


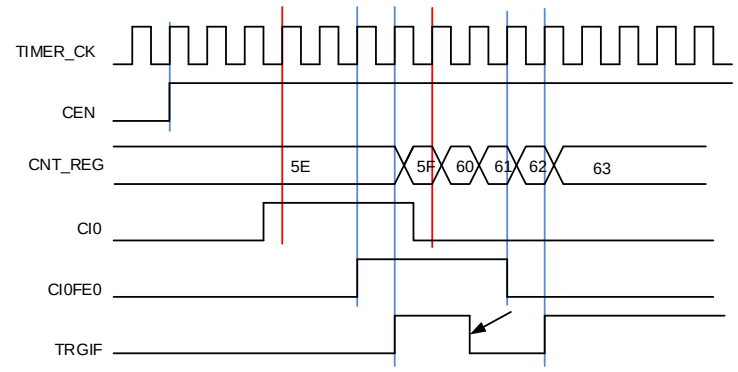
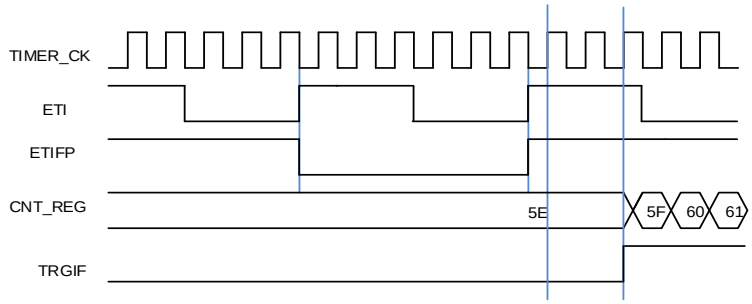
主-从管理

TIMERx 能在多种模式下同步外部触发，包括复位模式，暂停模式和事件模式等，可以通过设置 SYSCFG_TIMERxCFG(x = 14)寄存器中的 TSCFGy[4:0] (y = 3..7)位域来确定，具体的输入触发源可以通过 TSCFGy[4:0] (y = 3..7)位域的值来选择。

表 23-21. 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	TSCFGy[4:0] y = 3：复位模式 y = 4：暂停模式 y = 5：事件模式 y = 6：外部时钟模式 0	TSCFGy[4:0] 00000：模式禁能 00001：ITI0 00010：ITI1 00011：ITI2 00100：ITI3	如果触发源是CIxFEx（x = 0, 1）或者MCIxFEMx（x = 0, 1），配置CHxP、MCHxP和MCHxFP来选择极性和反相。	触发源ITIx，滤波和预分频不可用 触发源CIx/MCIx，配置CHxCAPFLT/MCHxCAPFLT设置滤波，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>y = 7：复位+事件模式</td><td>00101：CI0F_ED
00110：CI0FE0
00111：CI1FE1
01000：保留
01001：ITI4
01010：ITI5
01011：ITI6
01100：ITI7
01101：ITI8
01110：ITI9
01111：ITI10
10000：保留
10001：保留
10010：保留
10011：ITI14
其他：保留</td><td></td><td></td></tr>
<tr><td rowspan="2">例1</td><td>复位模式
当触发输入上升沿，计数器清零重启</td><td>TSCFG3[4:0]
5’b00001，选择ITI0为触发源</td><td>触发源是ITI0，极性选择不可用</td><td>触发源是ITI0，滤波和预分频不可用</td></tr>
<tr><td colspan="4">图 23-118. 复位模式下的控制电路
 
</td></tr>
<tr><td>例2</td><td>暂停模式
当触发输入为低的时候，计数器暂停计数</td><td>TSCFG4[4:0]
=5’b00110，
选择CI0FE0为触发源</td><td>TI0S=0（非异或）
[MCH0P=0,CH0P=0]
CI0FE0不反相，在上升沿捕获</td><td>在这个例子中滤波被旁路</td></tr>
</table>

<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 23-119. 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>TSCFG5[4:0]
=5’b01000，选择
ETIFP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 23-120. 事件模式下的控制电路
 
</td></tr>
<tr><td>例 4</td><td colspan="4">复位 + 事件模式
当触发输入的上升沿到来时，计数器被重新初始化并开始计数。该模式仅用于可延时的单脉冲模式。</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，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。

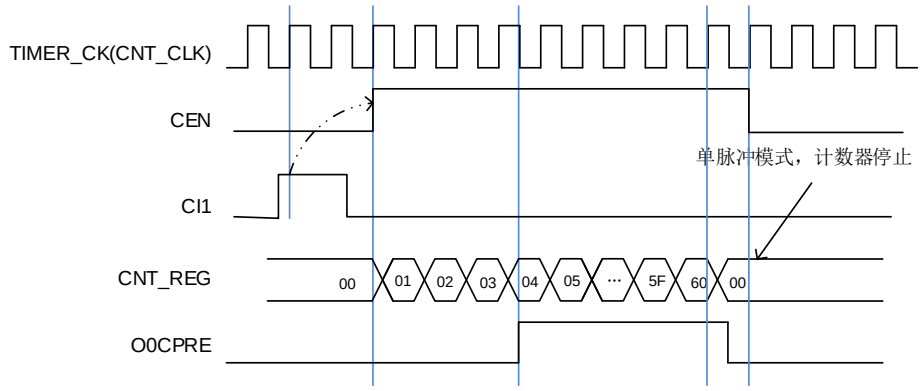
一旦设置定时器运行在单脉冲模式下，没有必要设置 TIMERx_CTL0 寄存器的定时器使能位 CEN=1 来使能计数器。触发信号沿或者软件写 CEN=1 都可以产生一个脉冲，此后 CEN 位一直保持为 1 直到更新事件发生或者 CEN 位被软件写 0。如果 CEN 位被软件清 0，计数器停止工作，计数值被保持。如果 CEN 值被硬件更新事件自动清 0，计数器将被再次初始化。

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 23-121. 单脉冲模式，TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60



可延时的单脉冲模式

可以通过将 TIMERx_CHCTLx / TIMERx_MCHCTLx 寄存器中的 CHxCOMCTL[3:0] / MCHxCOMCTL[3:0]位置1来使能可延时的单脉冲模式。在这个模式下，通道输出准备信号 OxCPRE/ MOxCPRE的脉冲宽度由TIMERx_CAR寄存器值确定。

一旦设置定时器运行在可延时的单脉冲模式下，需进行以下配置：

- 定时器必须工作在从模式下，SYSCFG_TIMERxCFG(x = 14)寄存器中的 TSCFG7[4:0] != 5’b00000，从模式选择复位 + 事件模式；
- CHxCOMCTL[3:0]/ MCHxCOMCTL[3:0]位设置为 4’b1000（可延时单脉冲模式 0）或 4’b1001（可延时单脉冲模式 1）

在可延时单脉冲模式0下，OxCPRE / MOxCPRE的输出情况类似与PWM模式0。在向上计数模式时，OxCPRE / MOxCPRE先输出有效电平，当外部触发事件发生时，立即输出无效电平，当下一次更新事件发生时，再变成有效电平；在向下计数模式时，OxCPRE / MOxCPRE先输出无效电平，当外部触发事件发生时，立即输出有效电平，当下一次更新事件发生时，再变成无效电平。

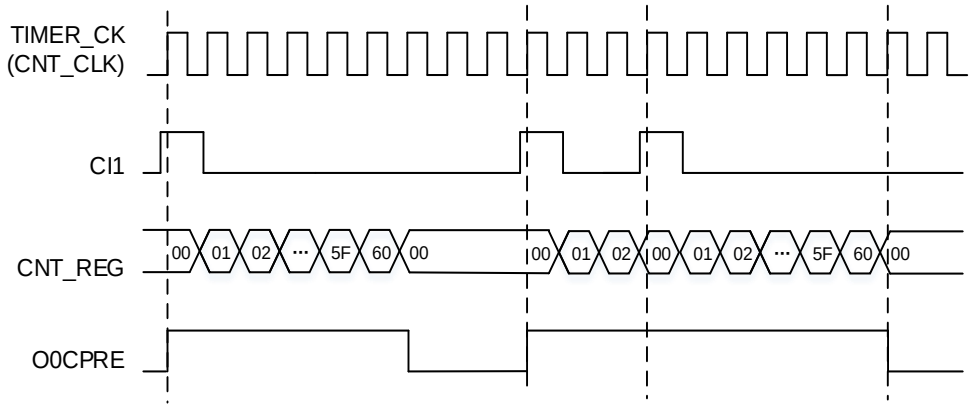
在可延时单脉冲模式1下，OxCPRE / MOxCPRE的输出情况类似与PWM模式1。在向上计数模式时，OxCPRE / MOxCPRE先输出无效电平，当外部触发事件发生时，立即输出有效电平，当下一次更新事件发生时，再变成无效电平；在向下计数模式时，OxCPRE / MOxCPRE先输出有效电平，当外部触发事件发生时，立即输出无效电平，当下一次更新事件发生时，再变成有效电平。

PWM微调模式也可用于可延迟的单脉冲模式。

注意：

- 不能使用中央对齐模式，TIMERx_CTL0 寄存器中的 CAM[1:0] = 2'b00；
- 在向上计数时（TIMERx_CTL0 寄存器中的 DIR = 0），TIMERx_CHxCV / TIMERx_MCHxCV 的值设置为 0；在向下计数时，TIMERx_CHxCV / TIMERx_MCHxCV 的值应大于或等于 TIMERx_CAR 的值。

图 23-122. 可延时单脉冲模式（TIMERx_CHxCV = 0x00，TIMERx_CAR = 0x60）



定时器互连

参考*高级定时器(TIMERx,x = 0,7)定时器互连*。

定时器 DMA 模式

定时器 DMA 模式是指通过 DMA 模块配置定时器的寄存器。有两个跟定时器 DMA 模式相关的寄存器：TIMERx_DMACFG 和 TIMERx_DMATB。当然，必须要使能 DMA 请求，一些内部中断事件可以产生 DMA 请求。当中断事件发生，TIMERx 会给 DMA 发送请求。DMA 配置成 M2P 模式，PADDR 是 TIMERx_DMATB 寄存器地址，DMA 就会访问 TIMERx_DMATB 寄存器。实际上，TIMERx_DMATB 寄存器只是一个缓冲，定时器会将 TIMERx_DMATB 映射到一个内部寄存器，这个内部寄存器由 TIMERx_DMACFG 寄存器中的 DMATA 来指定。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值为 0，表示 1 次传输，定时器的发送 1 个 DMA 请求就可以完成。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值不为 1，例如其值为 3，表示 4 次传输，定时器就需要再多发 3 次 DMA 请求。在这 3 次请求下，DMA 对 TIMERx_DMATB 寄存器的访问会映射到访问定时器的 DMATA+0x4, DMATA+0x8, DMATA+0xc 寄存器。总之，发生一次 DMA 内部中断请求，定时器会连续发送（DMATC+1）次请求。

如果再来 1 次 DMA 请求事件，TIMERx 将会重复上面的过程。

UPIF 位备份功能

可以通过配置TIMERx_CTL0寄存器中的UPIFBUEN位来使能UPIF位的备份功能，UPIF和UPIFBU位之间没有延迟，两者完全同步。

使能该功能后，TIMERx_INTF寄存器中的UPIF位将会被实时备份到TIMERx_CNT寄存器中的

UPIFBU位。这可以避免在读计数器和中断处理时产生冲突的情况。

定时器调试模式

当 Cortex®-M33 内核停止，DBG_CTL1 寄存器中的 TIMERx_HOLD 配置位被置 1，定时器计数器停止。

23.3.5. TIMERx 寄存器(x = 14)

TIMER14 基地址：0x4001 4000

控制寄存器 0（TIMERx_CTL0）

地址偏移：0x00
复位值：0x0000 0000

该寄存器通过字访问（32位）。

31	30	29	28	27	26	25	24	23	22	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			ADMEN	UPIFBUEN	保留	CKDIV[1:0]		ARSE	保留			SPM	UPS	UPDIS	CEN
			rw	rw		rw		rw				rw	rw	rw	rw

位/位域	名称	描述
31:13	保留	必须保持复位值。
12	ADMEN	微调模式使能 0：微调模式禁能 1：微调模式使能 注意：该位只能在CEN位为0时修改。
11	UPIFBUEN	UPIF位备份使能 0：备份禁能。UPIF位没有备份到TIMERx_CNT寄存器中的UPIFBU位 1：备份使能。UPIF 位备份到 TIMERx_CNT 寄存器中的 UPIFBU 位
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（TIMER_CK）与死区时间和采样时钟（DTS）之间的分频系数，死区发生器和数字滤波器会用到 DTS 时间。 00：$f_{DTS}=f_{TIMER_CK}$ 01：$f_{DTS}= f_{TIMER_CK}$ /2 10：$f_{DTS}= f_{TIMER_CK}$ /4 11：保留
7	ARSE	自动重载影子使能 0：禁能 TIMERx_CAR 寄存器的影子寄存器 1：使能 TIMERx_CAR 寄存器的影子寄存器

6:4	保留	必须保持复位值。
3	SPM	单脉冲模式 0：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，CEN 硬件清零并且计数器停止计数
2	UPS	更新请求源 软件配置该为，选择更新事件源. 0：使能后，下述任一事件产生更新中断或 DMA 请求： – UPG 位被置 1 – 计数器上溢 – 从模式控制器产生的更新 1：使能后只有计数器上溢才产生更新中断或 DMA 请求
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.当以下事件之一发生时，更新事件产生，具有缓存的寄存器被装入它们的预装载值： – UPG 位被置 1 – 计数器上溢 – 从模式控制器产生一个更新事件 1：更新事件禁能。带有缓存的寄存器保持原有值，如果 UPG 位被置 1 或者从模式控制器产生一个硬件复位事件，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。触发模式可以自动地通过硬件设置 CEN 位。

控制寄存器 1（TIMERx_CTL1）

地址偏移：0x04
复位值：0x0000 0000

该寄存器通过字访问（32位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCUC[2:1]		保留													
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					ISO1	ISO0N	ISO0	保留	MMC0[2:0]			DMAS	CCUC[0]	保留	CCSE
					rw	rw	rw		rw			rw	rw		rw

位/位域	名称	描述
31:30	CCUC[2:1]	换相控制影子寄存器更新控制 请参考 CCUC [0]的描述。
29:11	保留	必须保持复位值。
10	ISO1	通道 1 的空闲状态输出 参考 ISO0 位。
9	ISO0N	多模式通道 0 的互补通道空闲状态输出 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_CCHP0 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_CCHP0 寄存器的 PROT [1:0]位为 00 的时候可以被更改。
7	保留	必须保持复位值。
6:4	MMC0[2:0]	主模式控制 0 这些位控制 TRGO0 信号的选择，TRGO0 信号由主定时器发给从定时器用于同步功能 000：复位。TIMERx_SWEVG 寄存器的 UPG 位被置 1 或从模式控制器产生复位触发一次 TRGO0 脉冲，后一种情况下，TRGO0 上的信号相对实际的复位会有一个延迟。 001：使能。此模式可用于同时启动多个定时器或控制在一段时间内使能从定时器。主模式控制器选择计数器使能信号作为触发输出 TRGO0。当 CEN 控制位被置 1 或者暂停模式下触发输入为高电平时，计数器使能信号被置 1。在暂停模式下，计数器使能信号受控于触发输入，在触发输入和 TRGO0 上会有一个延迟，除非选择了主/从模式。 010：更新。主模式控制器选择更新事件作为 TRGO0。 011：捕获/比较脉冲.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 TRGO0 脉冲 100：比较。在这种模式下主模式控制器选择 O0CPRE 信号被用于作为触发输出 TRGO0 101：比较。在这种模式下主模式控制器选择 O1CPRE 信号被用于作为触发输出 TRGO0 110：保留 111：保留
3	DMAS	DMA请求源选择

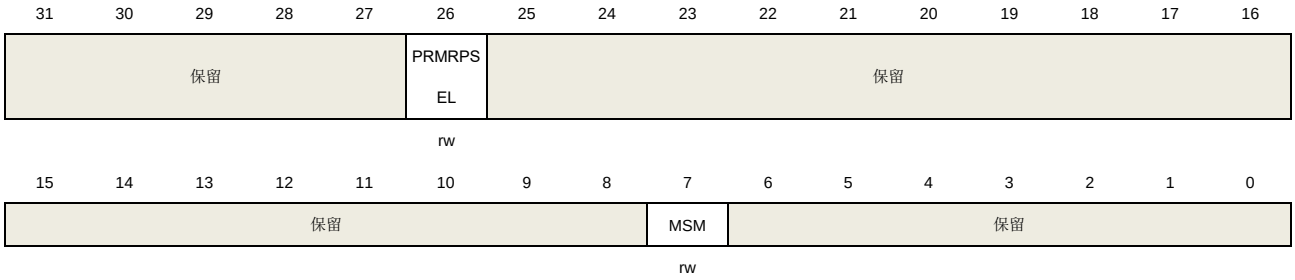
		0：当通道捕获/比较事件发生时，发送通道 CHx/MCHx 的 DMA 请求 . 1：当更新事件发生，发送通道 CHx/MCHx 的 DMA 请求
2	CCUC	换相控制影子寄存器更新控制 当换相控制影子寄存器（CHxEN，MCHxEN 和 CHxCOMCTL 位）使能（CCSE=1），这些影子寄存器更新控制如下： 000：CMTG 位被置 1 时，更新影子寄存器 001：当 CMTG 位被置 1 或检测到 TRIGI 上升沿时，影子寄存器更新 100：当计数器上溢事件发生时，影子寄存器更新 其他值：保留 当通道没有互补输出时，此位无效。
1	保留	必须保持复位值。
0	CCSE	换相控制影子使能 0：影子寄存器 CHxEN、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

该寄存器通过字访问（32位）。



位/位域	名称	描述
31:27	保留	必须保持复位值。
26	PRMRPSEL	暂停 + 复位模式复位极性选择 0：计数器在下降沿复位 1：计数器在上升沿复位
25:8	保留	必须保持复位值。
7	MSM	主-从模式

		该位被用来同步被选择的定时器同时开始计数。通过 TRIGI 和 TRGO0，定时器被连接在一起，TRGO0 用做启动事件。 0：主从模式禁能 1：主从模式使能
6:0	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

该寄存器通过字访问（32位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		CH1COMADDIE	CH0COMADDIE	保留			MCH0DEN	保留			MCH0IE	保留			
		rw	rw				rw				rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TRGDEN	CMTDEN	保留		CH1DEN	CH0DEN	UPDEN	BRKIE	TRGIE	CMTIE	保留		CH1IE	CH0IE	UPIE
	rw	rw			rw	rw	rw	rw	rw	rw			rw	rw	rw

位/位域	名称	描述
31:30	保留	必须保持复位值。
29	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。
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。
27:25	保留	必须保持复位值。
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] = 2'b00 时）。
23:21	保留	必须保持复位值。

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] = 2'b00 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
12:11	保留	必须保持复位值。
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
4:3	保留	必须保持复位值。
2	CH1IE	通道 1 比较/捕获中断使能 0：禁止通道 1 中断

		1：使能通道 1 中断
1	CH0IE	通道 0 比较/捕获中断使能 0：禁止通道 0 中断 1：使能通道 0 中断
0	UPIE	更新中断使能 0：禁止更新中断 1：使能更新中断

中断标志寄存器（TIMERx_INTF）

地址偏移：0x10

复位值：0x0000 0000

该寄存器通过字访问（32位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		CH1COMADDIF	CH0COMADDIF	保留			MCH0OF	保留			MCH0IF	保留			
		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					CH1OF	CH0OF	保留	BRK0IF	TRGIF	CMTIF	保留		CH1IF	CH0IF	UPIF
					rc_w0	rc_w0		rc_w0	rc_w0	rc_w0			rc_w0	rc_w0	rc_w0

位/位域	名称	描述
31:30	保留	必须保持复位值。
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。
27:25	保留	必须保持复位值。
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
23:21	保留	必须保持复位值。
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
19:11	保留	必须保持复位值。
10	CH1OF	通道 1 捕获溢出标志 参见 CH0OF 描述
9	CH0OF	通道 0 捕获溢出标志 当通道 0 被配置为输入模式时，在 CH0IF 标志位已经被置 1 后，捕获事件再次发生时，该标志位可以由硬件置 1。该标志位由软件清 0 0：无捕获溢出中断发生 1：发生了捕获溢出中断
8	保留	必须保持复位值。
7	BRK0IF	BREAK0 中断标志位 一旦 BREAK0 输入有效，由硬件对该位置‘1’。如果 BREAK0 输入无效，则该位可由软件清‘0’。 0：无 BREAK0 事件产生 1：BREAK0 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，此标志由硬件置 1。此位由软件清 0。当从模式控制器处于除暂停模式外的其它模式时，在触发输入端检测到有效边沿，产生触发事件。当从模式控制器处于暂停模式时，触发输入的任意边沿都可以产生触发事件。 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
4:3	保留	必须保持复位值。
2	CH1IF	通道 1 比较/捕获中断标志

参见 CH0IF 描述

1	CH0IF	通道 0 比较/捕获中断标志 此标志由硬件置 1 软件清 0。当通道 0 在输入模式下时，捕获事件发生时此标志位被置 1；当通道 0 在输出模式下时，此标志位在一个比较事件发生时被置 1。 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

该寄存器通过字访问（32位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		CH1COMADDG	CH0COMADDG	保留							MCH0G	保留			
		w	w								w				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								BRK0G	TRGG	CMTG	保留		CH1G	CH0G	UPG
								w	w	w			w	w	w

位/位域	名称	描述
31:30	保留	必须保持复位值。
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。
27:21	保留	必须保持复位值。

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	BRK0G	产生 BREAK0 事件 该位由软件置 1，用于产生一个 BREAK0 事件，由硬件自动清 0。当此位被置 1 时，POEN 位被清 0 且 BRK0IF 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 BREAK0 事件 1：产生 BREAK0 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） 的互补输出被更新（根据 TIMERx_CTL1 中 CCSE 值）。 0：不产生通道控制更新事件 1：产生通道控制更新事件
4:3	保留	必须保持复位值。
2	CH1G	通道 1 捕获或比较事件发生 参见 CH0G 描述。
1	CH0G	通道 0 捕获或比较事件发生 该位由软件置 1，用于在通道 0 产生一个捕获/比较事件，由硬件自动清 0。当此位被置 1，CH0IF 标志位被置 1，若开启对应的中断和 DMA，则发出相应的中断和 DMA 请求。此外，如果通道 0 配置为输入模式，计数器的当前值被 TIMERx_CH0CV 寄存器捕获，如果 CH0IF 标志位已经为 1，则 CH0OF 标志位被置 1。 0：不产生通道 0 捕获或比较事件 1：发生通道 0 捕获或比较事件

0	UPG	更新事件产生 此位由软件置 1，被硬件自动清 0。当此位被置 1，向上计数模式，计数器被清 0，预分频计数器将同时被清除。 0：无更新事件产生 1：产生更新事件

通道控制寄存器 0（TIMERx_CHCTL0）

地址偏移：0x18

复位值：0x0000 0000

该寄存器通过字访问（32位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH1MS[2]	CH0MS[2]	CH1COMADDSEN	CH0COMADDSEN	CH1ADDUPS	CH0ADDUPS	保留	CH1COMCTL[3]	保留							CH0COMCTL[3]
		保留	保留	保留	保留		保留								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
CH1COMCEN	CH1COMCTL[2:0]			CH1COMSEN	CH1COMFEN	CH1MS[1:0]		CH0COMCEN	CH0COMCTL[2:0]			CH0COMSEN	CH0COMFEN	CH0MS[1:0]	
CH1CAPFLT[3:0]				CH1CAPPSC[1:0]				CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

输出比较模式：

位/位域	名称	描述
31	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：通道 0 附加比较输出影子寄存器禁能 1：通道 0 附加比较输出影子寄存器使能 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_CCHP0 寄存器的 PROT [1:0]=11 且 CH0MS =000 时此位不能被改变。
27	CH1ADDUPS	通道1附加寄存器更新源

		0：在发生更新事件时，更新TIMERx_CH1COMV_ADD寄存器 1：在发生计数器计数值匹配 CH1VAL 值时，更新 TIMERx_CH1COMV_ADD 寄存器
26	CH0ADDUPS	通道0附加寄存器更新源 0：在发生更新事件时，更新TIMERx_CH0COMV_ADD寄存器 1：在发生计数器计数值匹配 CH0VAL 值时，更新 TIMERx_CH0COMV_ADD 寄存器
25	保留	必须保持复位值。
24	CH1COMCTL[3]	通道 1 输出比较控制 参见 CH0COMCTL[2:0]描述
23:17	保留	必须保持复位值。
16	CH0COMCTL[3]	通道 0 输出比较控制 参见 CH0COMCTL[2:0]描述
15	CH1COMCEN	通道 1 输出比较清 0 使能 参见 CH0COMCEN 描述
14:12	CH1COMCTL[2:0]	通道 1 输出比较模式 参见 CH0COMCTL 描述
11	CH1COMSEN	通道 1 输出比较影子寄存器使能 参见 CH0COMSEN 描述
10	CH1COMFEN	通道1输出比较快速使能 参见 CH0COMFEN 描述
9:8	CH1MS[1:0]	通道 1 模式选择 CH1MS[2:0]位域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] = 2'b00 时，TIMERx_CHCTL2 寄存器的 CH1EN 位清 0；当 MCH1MSEL[1:0] = 2'b01 或 2'b11 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器输入被选中时（由 SYSCFG_TIMERxCFG2(x = 14)寄存器中的 TSCFG15[4:0]位域选择）。 100~111：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

CH0COMCTL[3]和 CH0COMCTL[2:0]位域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] = 2'b11 时，CH0COMCTL[3]和 CH0COMCTL[2:0]位域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。

0000：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

0001：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。

0010：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。

0011：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。

0100：强制为低。强制 O0CPRE 为低电平

0101：强制为高。强制 O0CPRE 为高电平

0110：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。

0111：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。

1000：可延时的单脉冲模式0。O0CPRE的输出情况类似与PWM模式0。在向上计数模式时，O0CPRE先输出有效电平，当外部触发事件发生时，立即输出无效电平，当下一次更新事件发生时，再变成有效电平；在向下计数模式时，O0CPRE先输出无效电平，当外部触发事件发生时，立即输出有效电平，当下一次更新事件发生时，再变成无效电平。

1001：可延时的单脉冲模式1。O0CPRE的输出情况类似与PWM模式1。在向上计数模式时，O0CPRE先输出无效电平，当外部触发事件发生时，立即输出有效电平，当下一次更新事件发生时，再变成无效电平；在向下计数模式时，O0CPRE先输出有效电平，当外部触发事件发生时，立即输出无效电平，当下一次更新事件发生时，再变成有效电平。

1010~1111：保留

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***。

在 PWM 模式 0 或 PWM 模式 1 中，只有当比较结果改变了或者输出比较模式中从

		时基模式切换到 PWM 模式时，O0CPRE 电平才改变。 当 CH0 和 MCH0 输出互补时，该位域预装载。若 CCSE =1，则该位域只在通道换相事件发生时更新。 当 TIMERx_CCHP0 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（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_CCHP0 寄存器的 PROT [1:0]=11 且 CH0MS =000 时此位不能被改变。
2	CH0COMFEN	通道0输出比较快速使能 当该位为1时，如果通道配置为PWM模式0或者PWM模式1，会加快捕获 / 比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH0_O被设置为比较电平而与比较结果无关。 0：通道0输出比较快速禁能。当触发器的输入有一个有效沿时，激活CH0_O输出的最小延时为5个时钟周期 1：通道0输出比较快速使能。当触发器的输入有一个有效沿时，激活CH0_O输出的最小延时为 3 个时钟周期
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] = 2'b00 时，TIMERx_CHCTL2 寄存器的 CH1EN 位清 0；当 MCH0MSEL[1:0] = 2'b01 或 2'b11 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（由 SYSCFG_TIMERxCFG2(x = 14)寄存器中的 TSCFG15[4:0]位域选择）。 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 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 CI0 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	CH0CAPPSC[1:0]	通道 0 输入捕获预分频器 这 2 位定义了通道 0 输入的预分频系数。当 TIMERx_CHCTL2 寄存器中的 CH0EN=0 时，则预分频器复位。 00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获 01：每 2 个事件触发一次捕获 10：每 4 个事件触发一次捕获 11：每 8 个事件触发一次捕获
1:0	CH0MS[1:0]	通道 0 模式选择

与输出比较模式相同

通道控制寄存器 2（TIMERx_CHCTL2）

地址偏移：0x20

复位值：0x0000 0000

该寄存器通过字访问（32位）。

31	30	29	28	27	26	25	24	23	22	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.								MCH1P	保留	CH1P	CH1EN	MCH0P	MCH0EN	CH0P	CH0EN
								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	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] = 2'b11 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_CCHP0 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_CCHP0 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

该寄存器通过字访问（32位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UPIFBU	保留														
r															
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	UPIFBU	UPIF位备份 该位只读，是 TIMERx_INTF 寄存器的 UPIF 位的备份值。当 UPIFBUEN = 1 时，该位有效，若 UPIFBUEN =0，该位保留，读取该位值为零。
30:16	保留	必须保持复位值。

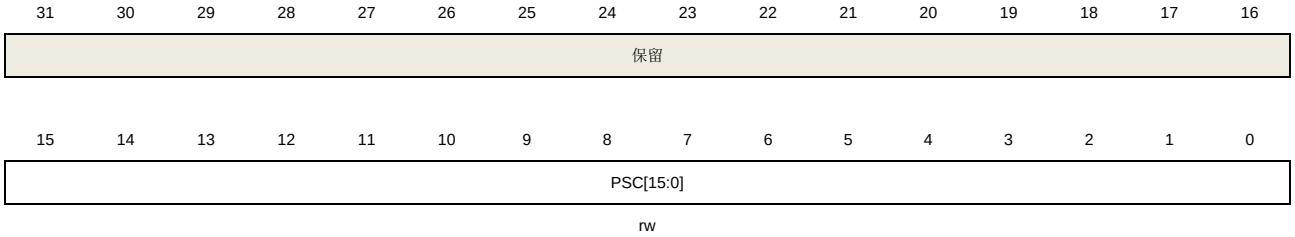
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。 当 PWMADMEN = 0 时，该位域表示当前计数器的值。写操作该位域可以改变计数器的值。 当 PWMADMEN = 1 时，该位域仅用于表示计数器值的整数部分，不包含小数部分。

预分频寄存器（TIMERx_PSC）

地址偏移：0x28

复位值：0x0000 0000

该寄存器通过字访问（32位）。



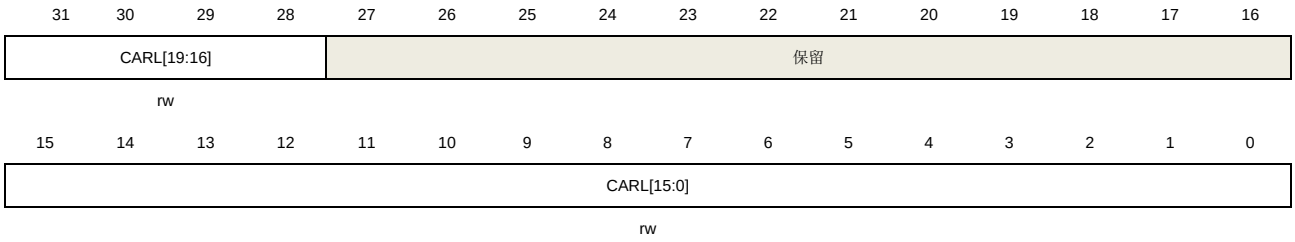
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 PSC 时钟除以（PSC+1），每次当更新事件产生时，PSC 的值被装入当前预分频寄存器。

计数器自动重载寄存器（TIMERx_CAR）

地址偏移：0x2C

复位值：0x0000 FFFF

该寄存器通过字访问（32位）。



位/位域	名称	描述
31:28	CARL[19:16]	计数器自动重载值（16 位至 19 位） 当 PWMADMEN = 0 时，CARL[19:16]位域值 0000。

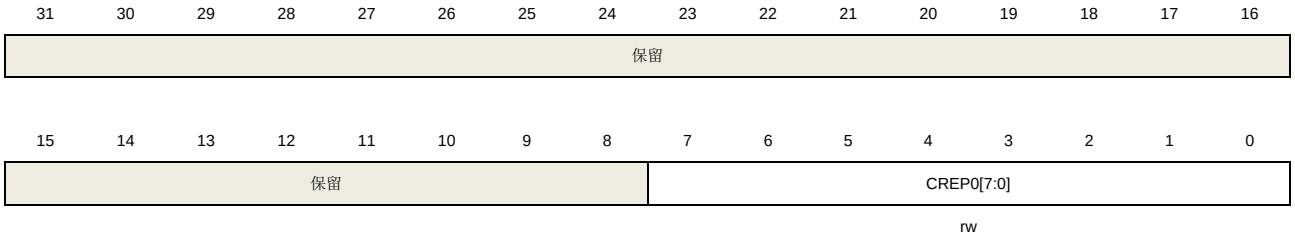
		当 PWMADMEN = 1 时，CARL[19:16]位域用于表示自动重载值的小数部分。
27:16	保留	必须保持复位值。
15:0	CARL[15:0]	计数器自动重载值（0 位至 15 位） 这些位定义了计数器的自动重载值。 当 PWMADMEN = 0 时，CARL[15:0]位域用于表示计数器的自动重载值。 当 PWMADMEN = 1 时，CARL[15:0]位域用于表示自动重载值的整数部分。

重复计数寄存器 0（TIMERx_CREP0）

地址偏移：0x30

复位值：0x0000 0000

该寄存器通过字访问（32位）。



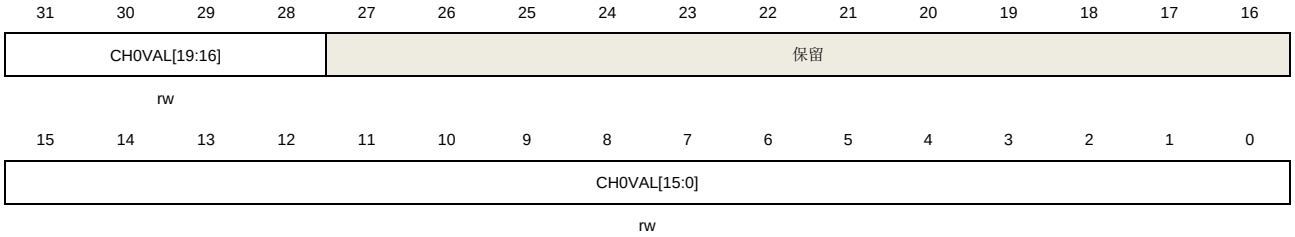
位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	CREP0[7:0]	重复计数器的值 0 这些位定义了更新事件的产生速率。重复计数器计数值减为 0 时产生更新事件。影子寄存器的更新速率也会受这些位影响（前提是影子寄存器被使能）。 **注意**：当 TIMERx_CFG 寄存器中的 CREPSEL =0 时，使用该位。

通道 0 捕获 / 比较寄存器（TIMERx_CH0CV）

地址偏移：0x34

复位值：0x0000 0000

该寄存器通过字访问（32位）。



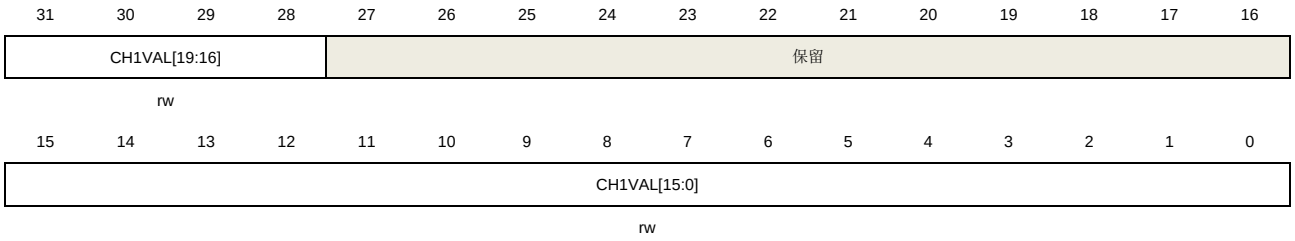
位/位域	名称	描述
31:28	CH0VAL[19:16]	通道 0 的捕获或比较值（16 位到 19 位） 当通道 0 配置为输入模式时，CH0VAL[19:16]位域值为 0000。 当通道 0 配置为输出模式时，该位域包含了即将和计数器比较的值。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。 当 PWMADMEN = 0 时，CH0VAL[19:16]位域值为 0000。 当 PWMADMEN = 1 时，CH0VAL[19:16]位域用于表示比较值的小数部分。
27:16	保留	必须保持复位值。
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值（0 位至 15 位） 当通道 0 配置为输入模式时，CH0VAL[15:0]决定了上次捕获事件的计数器值，且该位域只读。 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。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。 当 PWMADMEN = 0 时，CH0VAL[15:0]位域值表示比较值。 当 PWMADMEN = 1 时，CH0VAL[15:0]位域用于表示比较值的整数部分。

通道 1 捕获 / 比较寄存器（TIMERx_CH1CV）

地址偏移：0x38

复位值：0x0000 0000

该寄存器通过字访问（32位）。



位/位域	名称	描述
31:28	CH1VAL[19:16]	通道 1 的捕获或比较值（16 位到 19 位） 当通道 1 配置为输入模式时，CH1VAL[19:16]位域值为 0000。 当通道 1 配置为输出模式时，该位域包含了即将和计数器比较的值。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。 当 PWMADMEN = 0 时，CH1VAL[19:16]位域值为 0000。 当 PWMADMEN = 1 时，CH1VAL[19:16]位域用于表示比较值的小数部分。
27:16	保留	必须保持复位值。
15:0	CH1VAL[15:0]	通道 1 的捕获或比较值（0 位至 15 位） 当通道 1 配置为输入模式时，CH1VAL[15:0]决定了上次捕获事件的计数器值，且该

位域只读。

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。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。

当 PWMADMEN = 0 时，CH1VAL[15:0]位域值表示比较值。

当 PWMADMEN = 1 时，CH1VAL[15:0]位域用于表示比较值的整数部分。

互补通道保护寄存器 0（TIMERx_CCHP0）

地址偏移：0x44

复位值：0x0000 0000

该寄存器通过字访问（32位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			BRK0LK	保留	BRK0REL	保留						BRK0F[3:0]			
			rw		rw								rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POEN	OAEN	BRK0P	BRK0EN	ROS	IOS	PROT[1:0]		DTCFG[7:0]							
rw	rw	rw	rw	rw	rw	rw		rw							

位/位域	名称	描述
31:29	保留	必须保持复位值。
28	BRK0LK	BREAK0输入锁存 0：BREAK0输入为输入模式 1：BREAK0输入为锁存模式 当BRK0LK置1时，BREAK0输入配置为开漏输出模式。 任何有效的BREAK0事件都会拉低BREAK0输入引脚电平，用于向外部设备提示有内部BREAK0事件发生。 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。 **注意**：对该位的每一次写操作，需要延时 1 个 APB 时钟才有效。
27	保留	必须保持复位值。
26	BRK0REL	BREAK0输入释放 当BREAK0输入无效时，该位由硬件清零。 0：BREAK0输入锁存 1：BREAK0输入释放 当软件将该位置1时，将释放锁存输出控制（高阻态的开漏模式）。当BREAK0事件无效时，该位由硬件清零。 **注意**：对该位的每一次写操作，需要延时 1 个 APB 时钟才有效。
25:20	保留	必须保持复位值。

19:16	BRK0F[3:0]	BREAK0输入信号滤波 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 BREAK0 输入信号的采样频率和数字滤波器的长度。 0000：无滤波器，BREAK0 异步有效，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 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0] = 00 时才可修改。
15	POEN	所有的通道输出使能 根据 OAEN 位，该位可以软件设置或者硬件自动设置。一旦中止输入有效，该位被硬件异步清 0。如果一个通道配置为输出模式，如果设置了相应的使能位（TIMERx_CHCTL2 寄存器的 CHxEN，CHxNEN 位），则开启 CHx_O 和 MCHx_O 输出。 0：禁止通道输出或强制为空闲状态 1：使能通道输出
14	OAEN	自动输出使能 此位定义了 POEN 位是否可以被硬件自动置 1。 0：POEN 位不能被硬件置 1 1：如果中止输入无效，下一次更新事件发生时，POEN 位能被硬件自动置 1 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0] = 00 时才可修改。
13	BRK0P	BREAK0 输入信号极性 此位定义了 BREAK0 输入的极性。 0：BREAK0 输入低电平有效 1：BREAK0 输入高电平有效 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0] = 00 时才可修改。
12	BRK0EN	BREAK0 输入信号使能

		此位置 1 使能 BREAK0 输入信号。 0：BREAK0 输入禁能 1：BREAK0 输入使能 此位只有在 TIMERx_CCHP0 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 23-19. 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_CCHP0 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 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 23-19. 由参数控制的互补输出表（MCHxMSEL=2'b11）***。 0：输出“关闭状态”禁能。当 CHxE 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_CCHP0 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_CCHP0 寄存器中 BRK0EN / BRK0P / 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_CCHP0 寄存器中 ROS / IOS 位。 11：PROT 模式 2。除了 PROT 模式 1 下的寄存器写保护外，还有 TIMERx_CHCTLR0/1 中 CHxCOMCTL / CHxCOMSEN 位（如果相关通道配置为输出模式）写保护。 11：PROT 模式 2。除了 PROT 模式 1 下的寄存器写保护外，还有 TIMERx_CHCTLR0 及 TIMERx_MCHCTL0 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_CCHP0 寄存器被写入，这两位被写

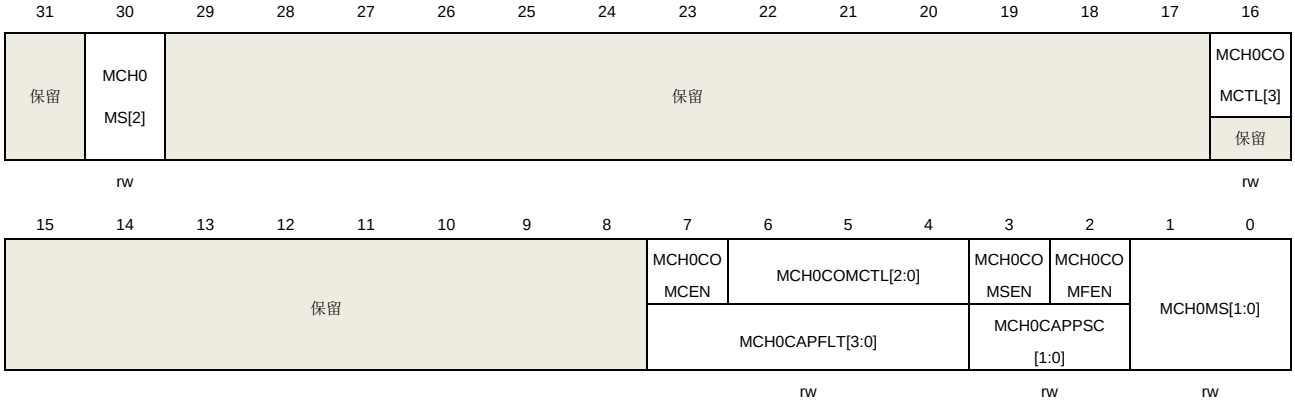
		保护
7:0	DTCFG[7:0]	死区时间控制 这些位定义了插入互补输出之间的死区持续时间。DTCFG 值和死区时间的关系如下： DTCFG [7:5] = 3'b0xx：DTvalue = DTCFG [7:0] x t_{DT}, t_{DT} = t_{DTS} DTCFG [7:5] = 3'b 10x：DTvalue =（64+DTCFG [5:0]）x t_{DT}，t_{DT} = t_{DTS} * 2 DTCFG [7:5] = 3'b 110：DTvalue =（32+DTCFG [4:0]）x t_{DT}，t_{DT}= t_{DTS} * 8 DTCFG [7:5] = 3'b 111：DTvalue =（32+DTCFG [4:0]）x t_{DT}，t_{DT} = t_{DTS} * 16 此位只有在 TIMERx_CCHP0 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。



输出比较模式：

位/位域	名称	描述
31	保留	必须保持复位值。
30	MCH0MS[2]	多模式通道 0 I/O 模式选择 参考MCH0MS[1:0]描述。
29:17	保留	必须保持复位值。
16	MCH0COMCTL[3]	多模式通道 0 输出比较控制 请参考MCH0COMCTL[2:0]描述。
15:8	保留	必须保持复位值。
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 输出比较控制 当多模式通道 0 配置为输出模式，并且 MCH0MSEL[1:0] = 2'b00，MCH0COMCTL[3]和 MCH0COMCTL[2:0]位域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] = 2'b11 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。 0000：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 0001：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。 0010：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。 0011：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。 0100：强制为低。强制 MO0CPRE 为低电平 0101：强制为高。强制 MO0CPRE 为高电平 0110：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。 0111：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。 1000：可延时的单脉冲模式0。MO0CPRE的输出情况类似与PWM模式0。在向上计数模式时，MO0CPRE先输出有效电平，当外部触发事件发生时，立即输出无效电平，当下一次更新事件发生时，再变成有效电平；在向下计数模式时，MO0CPRE先输出无效电平，当外部触发事件发生时，立即输出有效电平，当下一次更新事件发生时，再变成无效电平。 1001：可延时的单脉冲模式1。MO0CPRE的输出情况类似与PWM模式1。在向上计数模式时，MO0CPRE先输出无效电平，当外部触发事件发生时，立即输出有效电平，当下一次更新事件发生时，再变成无效电平；在向下计数模式时，MO0CPRE先输出有效电平，当外部触发事件发生时，立即输出无效电平，当下一次更新事件发生时，再变成有效电平。 1010~1111：保留 在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或者输出比较模式中从时基模式切换到 PWM 模式时，MO0CPRE 电平才改变。

		当 CH0 和 MCH0 输出互补时，该位域预装载。若 CCSE =1，则该位域只在通道换相事件发生时更新。 当TIMERx_CCHP0寄存器的PROT[1:0]=11且MCH0MS =000（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_CCHP0寄存器的PROT [1:0]=11且CH0MS=00时此位不能被改变。
2	MCH0COMFEN	多模式通道0输出比较快速使能 当该位为1时，如果通道配置为PWM模式0或者PWM模式1，会加快捕获 / 比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，MCH0_O被设置为比较电平而与比较结果无关。 0：多模式通道0输出比较快速禁能。当触发器的输入有一个有效沿时，激活MCH0_O输出的最小延时为5个时钟周期 1：多模式通道0输出比较快速使能。当触发器的输入有一个有效沿时，激活MCH0_O输出的最小延时为3个时钟周期
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：保留 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（由 SYSCFG_TIMERxCFG2(x = 14)寄存器中的 TSCFG15[4:0]位域选择）。 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	保留	必须保持复位值。
30	MCH0MS[2]	多模式通道 0 I/O 模式选择 参考MCH0MS[1:0]描述。

29:8	保留	必须保持复位值。
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 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														MCH0FP[1:0]	
														rw	

位/位域	名称	描述
31:2	保留	必须保持复位值。
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] = 2'b00，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_CCHP0 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。

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
CH1COMVAL_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CH0CPWMEN=1时）。

控制寄存器 2（TIMERx_CTL2）

地址偏移：0x74

复位值：0x0030 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		CH1CPWMEN	CH0CPWMEN	保留						MCH0MSEL[1:0]		保留			
		rw	rw							rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				CH1OMPSEL[1:0]		CH0OMPSEL[1:0]		保留							
				rw		rw									

位/位域	名称	描述
31:30	保留	必须保持复位值。
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
27:22	保留	必须保持复位值。
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：保留 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
19:12	保留	必须保持复位值。
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：保留 11：保留
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：保留 11：保留
7:0	保留	必须保持复位值。

TIMERx 备用功能控制寄存器 0（TIMERx_AFCTL0）

地址偏移：0x8C

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			BRK0CMP3P	BRK0CMP2P	BRK0CMP1P	BRK0CMP0P	保留								BRK0IN0P
			rw	rw	rw	rw									rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRK0CMP6EN	BRK0CMP5EN	BRK0CMP4EN	BRK0CMP3EN	BRK0CMP2EN	BRK0CMP1EN	BRK0CMP0EN	BRK0HPDFEN	BRK0CMP7EN	保留						BRK0IN0EN
rw	rw	rw	rw	rw	rw	rw	rw	rw							rw

位/位域	名称	描述
31:29	保留	必须保持复位值。
28	BRK0CMP3P	BREAK0 CMP3输入极性 该位用于配置CMP3输入极性，具体极性是由该位和BRK0P位共同确定。 0：CMP3输入信号不反相（BRK0P = 0，输入信号低有效；BRK0P = 1，输入信号高有效） 1：CMP3输入信号反相（BRK0P = 0，输入信号高有效；BRK0P = 1，输入信号低有效） 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
27	BRK0CMP2P	BREAK0 CMP2输入极性 该位用于配置CMP2输入极性，具体极性是由该位和BRK0P位共同确定。

		0：CMP2输入信号不反相（BRK0P = 0，输入信号低有效；BRK0P = 1，输入信号高有效） 1：CMP2输入信号反相（BRK0P = 0，输入信号高有效；BRK0P = 1，输入信号低有效） 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
26	BRK0CMP1P	BREAK0 CMP1输入极性 该位用于配置CMP1输入极性，具体极性是由该位和BRK0P位共同确定。 0：CMP1输入信号不反相（BRK0P =0，输入信号低有效；BRK0P =1，输入信号高有效） 1：CMP1输入信号反相（BRK0P =0，输入信号高有效；BRK0P =1，输入信号低有效） 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。
25	BRK0CMP0P	BREAK0 CMP0输入极性 0：CMP0输入信号不反相（BRK0P =0，输入信号低有效；BRK0P =1，输入信号高有效） 1：CMP输入信号反相（BRK0P =0，输入信号高有效；BRK0P =1，输入信号低有效） 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。
24:17	保留	必须保持复位值。
16	BRK0IN0P	BREAK0 BRKIN0备用输入极性 该位用于配置BRKIN0输入极性，具体极性是由该位和BRK0P位共同确定。 0：BRKIN0输入信号不反相（BRK0P =0，输入信号低有效；BRK0P =1，输入信号高有效） 1：BRKIN0输入信号反相（BRK0P =0，输入信号高有效；BRK0P =1，输入信号低有效） 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。
15	BRK0CMP6EN	BREAK0 CMP6输入使能 0：CMP6输入禁能 1：CMP6输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
14	BRK0CMP5EN	BREAK0 CMP5输入使能 0：CMP5输入禁能 1：CMP5输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
13	BRK0CMP4EN	BREAK0 CMP4输入使能 0：CMP4输入禁能 1：CMP4输入使能

		此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
12	BRK0CMP3EN	BREAK0 CMP3输入使能 0：CMP3输入禁能 1：CMP3输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
11	BRK0CMP2EN	BREAK0 CMP2输入使能 0：CMP2输入禁能 1：CMP2输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
10	BRK0CMP1EN	BREAK0 CMP1输入使能 0：CMP1输入禁能 1：CMP1输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。
9	BRK0CMP0EN	BREAK0 CMP0输入使能 0：CMP0输入禁能 1：CMP0输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。
8	BRK0HPDFEN	BREAK0 HPDF输入（HPDF_BREAK[0]）使能 0：HPDF输入禁能 1：HPDF输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。
7	BRK0CMP7EN	BREAK0 CMP7输入使能 0：CMP7输入禁能 1：CMP7输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
6:1	保留	必须保持复位值。
0	BRK0IN0EN	BREAK0 BRKIN0备用输入使能 0：BRKIN0输入禁能 1：BRKIN0输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。

备用功能控制寄存器 1（TIMERx_AFCTL1）

地址偏移：0x90
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

位/位域	名称	描述
31:22	保留	必须保持复位值。
21:19	OCRINSEL[2:0]	OCPRE_CLR输入选择 000：OCPRE_CLR0 001：OCPRE_CLR1 … 111：OCPRE_CLR7

OCPRE_CLR 输入选择	TIMER14
OCPRE_CLR0	CMP0_OUT
OCPRE_CLR1	CMP1_OUT
OCPRE_CLR2	CMP2_OUT
OCPRE_CLR3	CMP3_OUT
OCPRE_CLR4	CMP4_OUT
OCPRE_CLR5	CMP5_OUT
OCPRE_CLR6	CMP6_OUT
OCPRE_CLR7	CMP7_OUT

此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。

18:0	保留	必须保持复位值。

重复计数寄存器 1（TIMERx_CREP1）

地址偏移：0x98
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CREP1[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CREP1[15:0]															
rw															

位/位域	名称	描述
31:0	CREP1[31:0]	重复计数器值 1 该位域为 32 位，只读。 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）。 **注意**：当TIMERx_CFG寄存器中的CREPSEL=1时，使用该位域。

互补通道保护寄存器 1（TIMERx_CCHP1）

地址偏移：0x09C

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:18	保留	必须保持复位值。
17	DTMODEN	死区时间在线修改功能使能 0：死区时间在线修改功能禁能 1：死区时间在线修改功能使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。 **注意**：如果在CEN=1时将该位置1，则在上次更新事件之后写入的任何值（在DTCFG[7:0]位域和DTCFG[7:0]位域）都是无效的，只使用前一次的值。
16	DTDIFEN	死区时间不同配置使能 0：上升沿和下降沿的死区时间（在 TIMERx_CCHP1 寄存器的 DTCFG[7:0]位域配置）相同。 1：上升沿的死区时间由 TIMERx_CCHP1 寄存器的 DTCFG[7:0]位域配置；下降沿的死区时间由 TIMERx_CCHP1 寄存器的 DTFCFG[7:0]位域配置。 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
15:8	保留	必须保持复位值。
7:0	DTFCFG[7:0]	下降沿死区时间控制 该位域控制 OxCPRE 信号下降沿的死区时间值，该死区时间在输出转换之前插入。 DTCFG 值与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’b 10x：DTvalue =（64+DTCFG [5:0]）x t_{DT}，t_{DT} = t_{DTS} * 2

DTCFG [7:5] = 3’b 110：DTvalue =（32+DTCFG [4:0]）x t_{DT}，t_{DT}= t_{DTS} * 8

DTCFG [7:5] = 3’b 111：DTvalue =（32+DTCFG [4:0]）x t_{DT}，t_{DT} = t_{DTS} * 16

此位只有在TIMERx_CCHP0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。 6’b000000：传输1次 6’b000001：传输2次 … 6’b111000：传输 57 次
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_DMAVB 寄存器的第一个地址。当通过 TIMERx_DMA 第一次访问时，访问的就是该位域指定的地址。第二次访问 TIMERx_DMATB 时，将访问起始地址+0x4。 6’b0_0000：TIMERx_CTL0 6’b0_0001：TIMERx_CTL1 … 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
DMATB[31:16]															
rw															

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

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

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												CCUSEL	CREPSEL	CHVSEL	OUTSEL
												rw	rw	rw	rw

位/位域	名称	描述
31:4	保留	必须保持复位值。
3	CCUSEL	换相控制影子寄存器更新选择 只有当CCUC[2:0]位域配置为100,101和110时，该位才有效。 0：当计数器产生一个上溢/下溢事件时，影子寄存器才更新 1：当重复计数器值为0，且计数器产生一个上溢/下溢事件时，影子寄存器才更新
2	CREPSEL	计数器重复寄存器选择 该位用于选择重复计数寄存器。 0：更新事件的速率由TIMERx_CREP0寄存器确定 1：更新事件的速率由 TIMERx_CREP1 寄存器确定
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

23.4. 通用定时器 L4（TIMERx, x = 15, 16）

23.4.1. 简介

通用定时器 L4（TIMER15 / 16）是 2 通道定时器，支持输入捕获和输出比较。可以产生 PWM 信号控制电机和电源管理。通用定时器 L4 含有一个 16 位无符号计数器。

通用定时器 L4 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。

通用定时器 L4 包含了一个死区时间插入模块，非常适合电机控制。

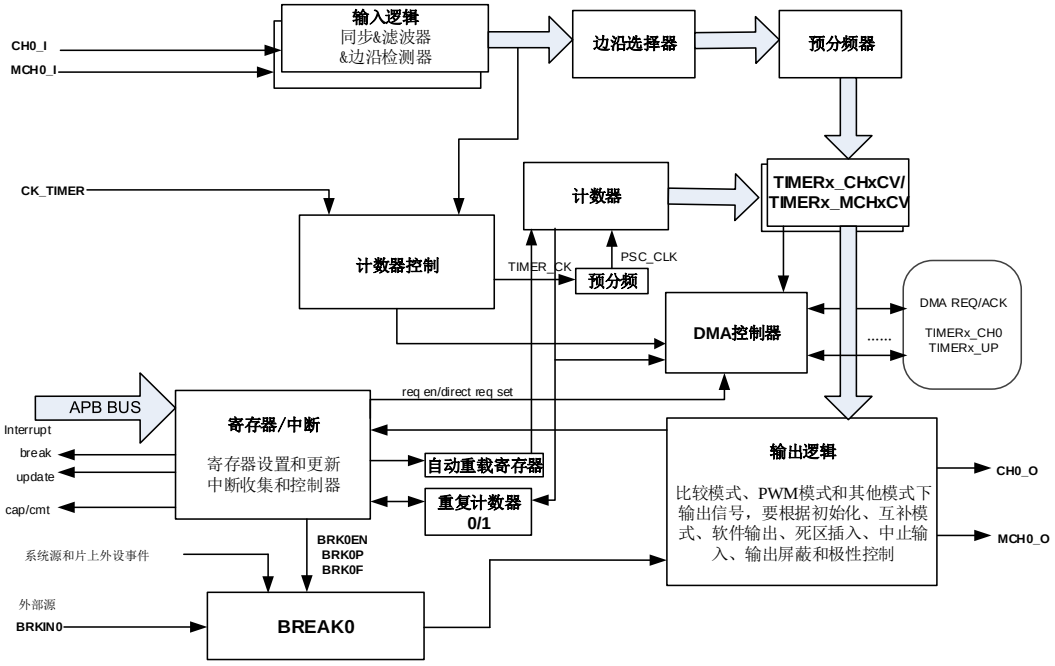
23.4.2. 主要特征

- 总通道数：2；
- 计数器宽度：16位；
- 时钟源可选：内部时钟；
- 计数模式：向上计数；
- 可编程的预分频器：16位，运行时可以被改变；
- 每个通道可配置：输入捕获模式，输出比较模式，可编程的PWM模式，单脉冲模式；
- 可编程的死区时间；
- 自动重装载功能；
- 可编程的计数器重复功能；
- 中止输入功能：BREAK0；
- 中断输出和DMA请求：更新事件，比较/捕获事件和中止事件。

23.4.3. 结构框图

*图 23-123. 通用定时器 L4 结构框图*提供了通用定时器 L4 的内部配置细节

图 23-123. 通用定时器 L4 结构框图



23.4.4. 功能描述

时钟源选择

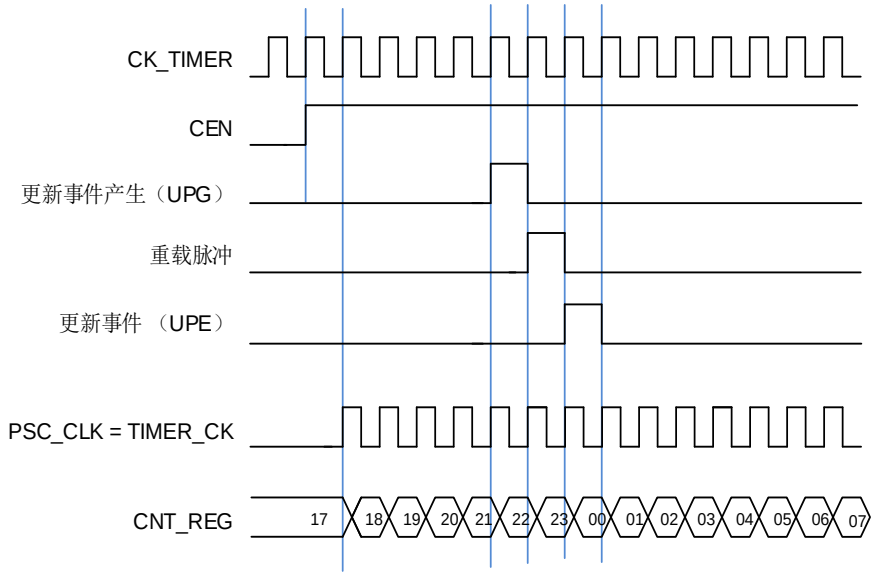
通用定时器 L4 由内部时钟源 TIMER_CK.

■ 定时器选择内部时钟源（连接到RCU模块的CK_TIMER）

通用定时器 L4 只有一个时钟源：内部时钟源。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。

驱动预分频器计数的 TIMER_CK 等于来自于 RCU 模块的 CK_TIMER。

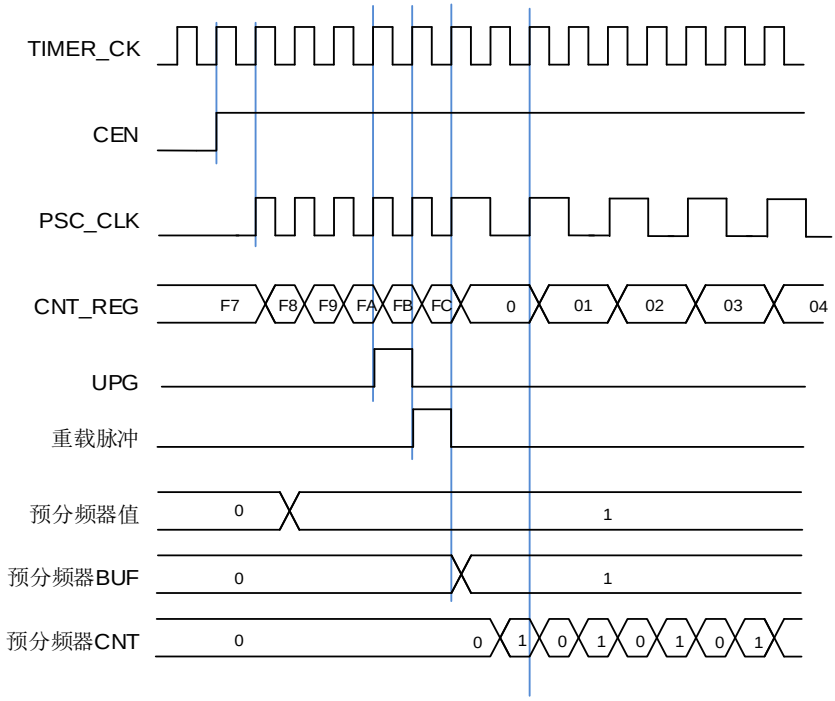
图 23-124. 内部时钟分频为 1 时正常模式下的控制电路



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 23-125. 当预分频器的参数从 1 变到 2 时，计数器的时序图



向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从 0 开始向上连续计数到自动加载值（定义在 TIMERx_CAR 寄存器中），一旦计数器计数到自动加载值，会重新从 0 开始向上计数。如果设置了重复计数器，在（TIMERx_CREP0 / 1 + 1）次上溢后产生更新事件，否则在每次上溢时都会产生更新事件。在向上计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 0。

当通过 TIMERx_SWEVG 寄存器的 UPG 位置 1 来设置更新事件时，计数值会被清 0，并产生更新事件。

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

当发生更新事件时，所有的寄存器（重复计数器，自动重载寄存器，预分频寄存器）都将被更新。

图 23-126. 向上计数时序图，PSC = 0 / 2 和 *图 23-127. 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 23-126. 向上计数时序图，PSC = 0 / 2

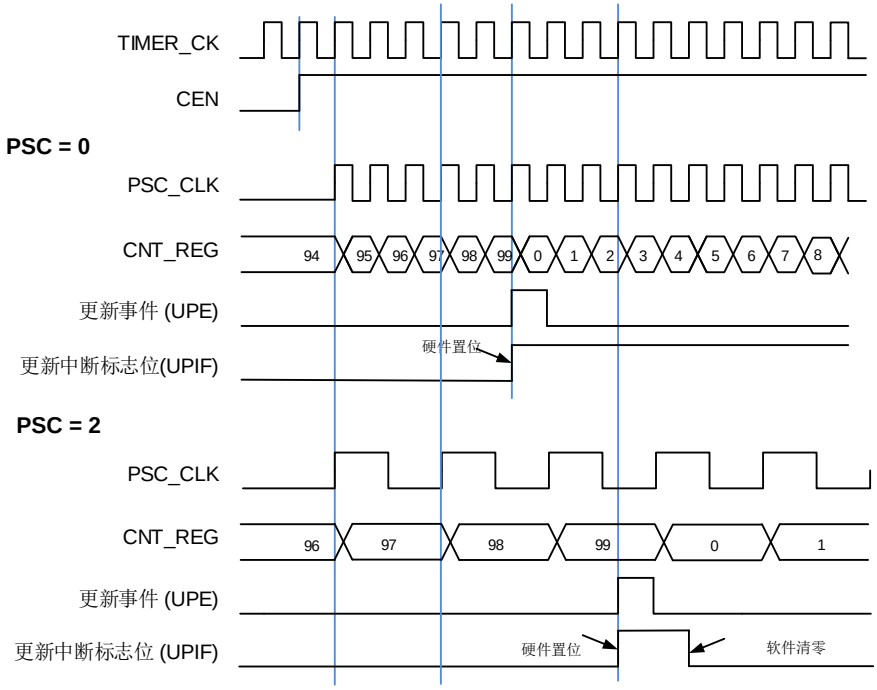
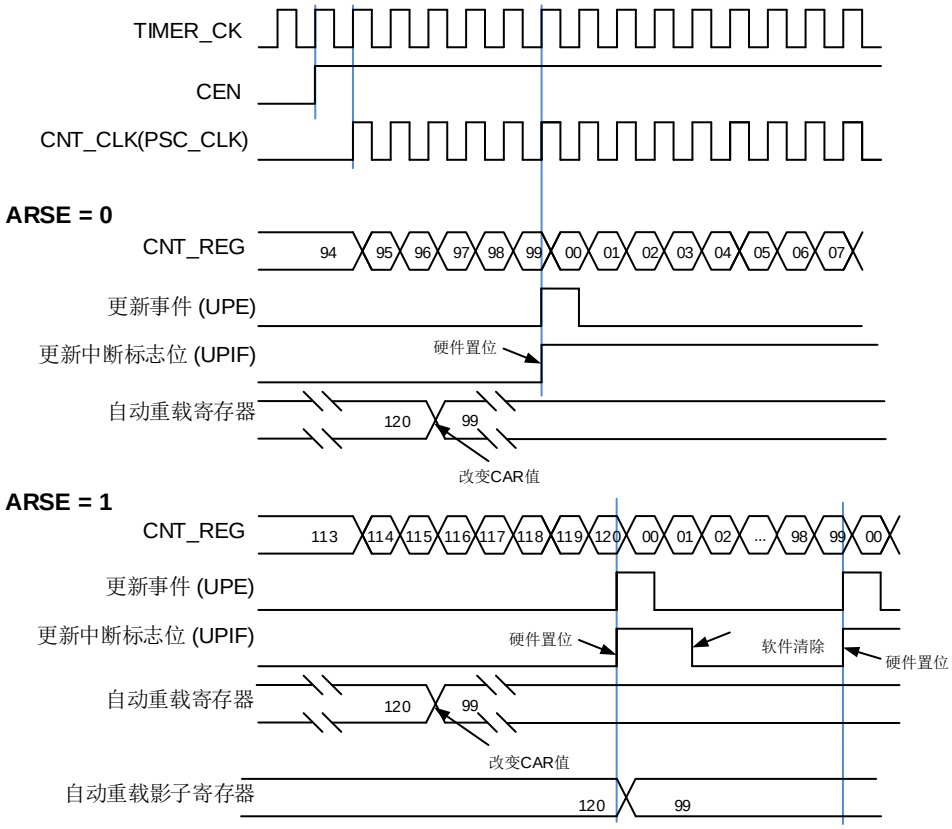


图 23-127. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



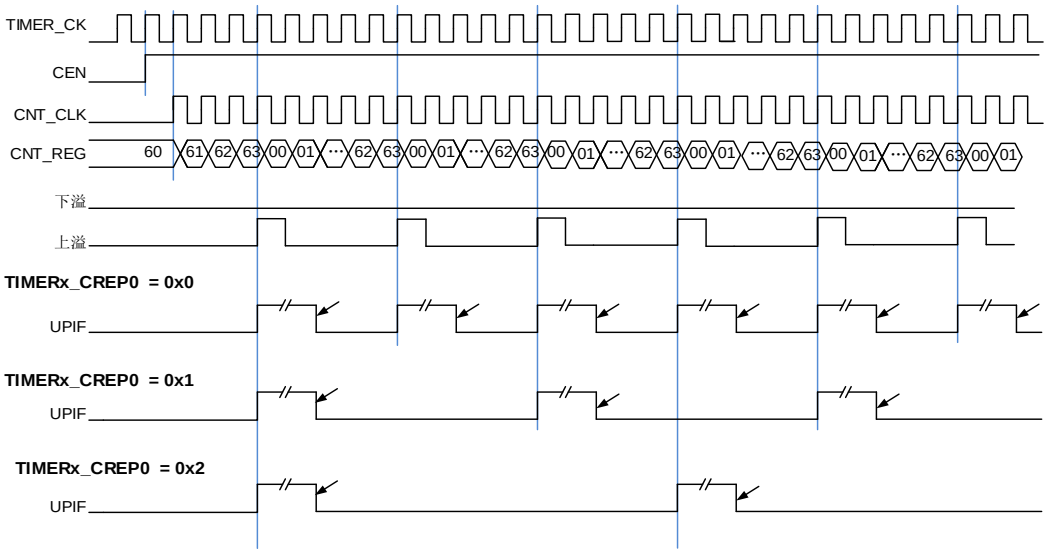
重复计数器

通用L3定时器有两个重复寄存器TIMERx_CREP0 / 1，可通过配置TIMERx_CFG寄存器中的CREPSEL位来选择。其中TIMERx_CREP0寄存器中的CREP0[7:0]是8位的，TIMERx_CREP1寄存器中的CREP1[31:0]是32位，用户可根据需求选择使用。

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_CREP0 / 1 寄存器的 CREP0 / 1。向上计数模式下，重复计数器在每次计数器上溢时递减。

将 TIMERx_SWEVG 寄存器的 UPG 位置 1 可以重载 TIMERx_CREP0 / 1 寄存器中 CREP0 / 1 的值并产生一个更新事件。

图 23-128. 在向上计数模式下计数器重复时序图



捕获/比较通道

通用定时器 L4 拥有 2 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 可输出独立和互补。

- 输入捕获模式

当 MCHxMSEL=2'b00（独立模式）时，通道 x 和多模式通道 x 才可以独立进行输入捕获。

捕获模式允许通道测量一个波形时序，频率，周期，占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，TIMERx_CHxCV / TIMERx_MCHxCV（x = 0）寄存器会捕获计数器当前的值，同时 CHxIF / MCHxIF（x = 0）位置 1，如果 CHxIE / MCHxIE =1（x = 0），则产生相应的通道中断。

图 23-129. 通道 0 输入捕获逻辑

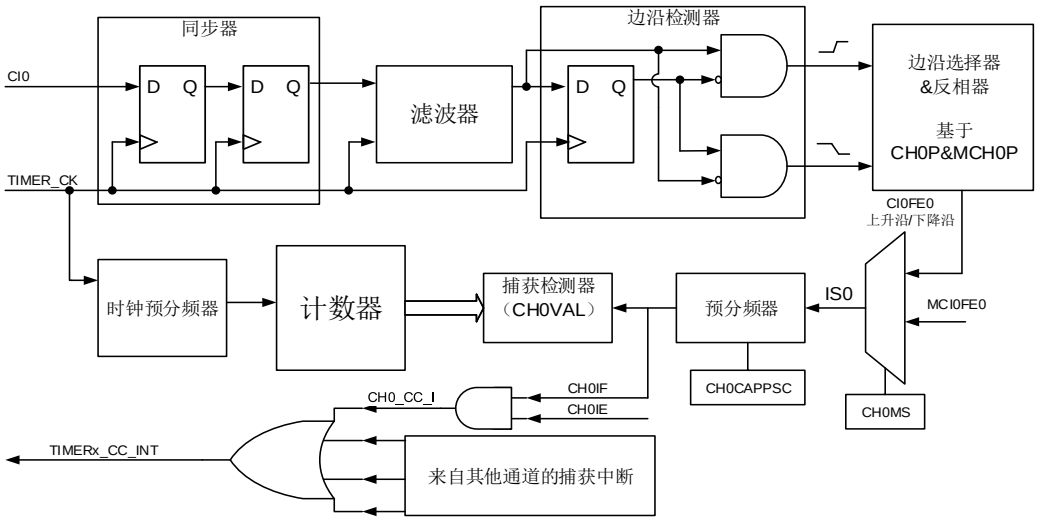
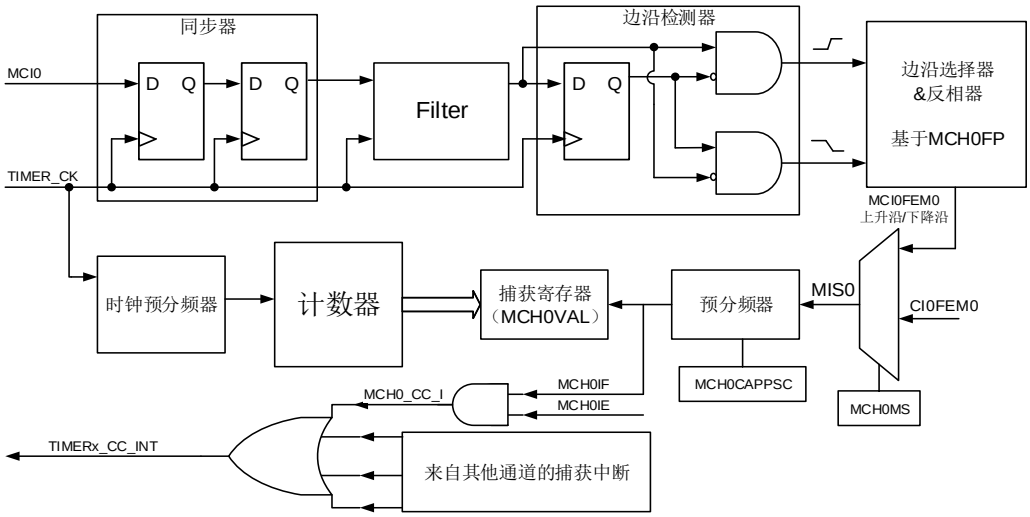


图 23-130. 多模式通道 0 输入捕获逻辑



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。

■ 输出比较模式

*图 23-131. 输出比较逻辑（当 MCHxMSEL = 2'b00 时，x = 0）*和*图 23-132. 输出比较逻辑（当 MCHxMSEL = 2'b11 时，x = 0）*给出了通道的输出比较逻辑。

图 23-131. 输出比较逻辑（当 MCHxMSEL = 2'b00 时，x = 0）

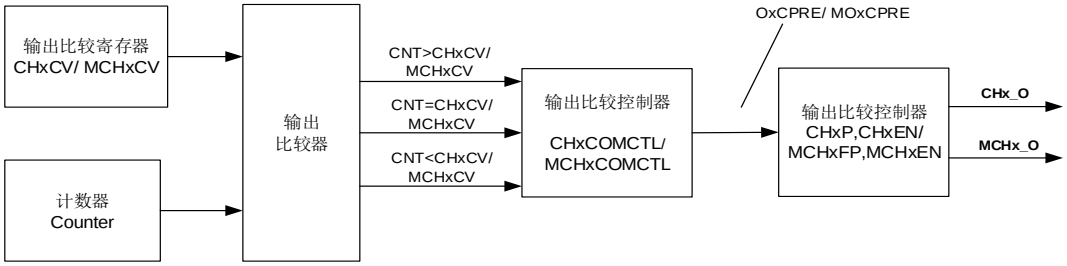
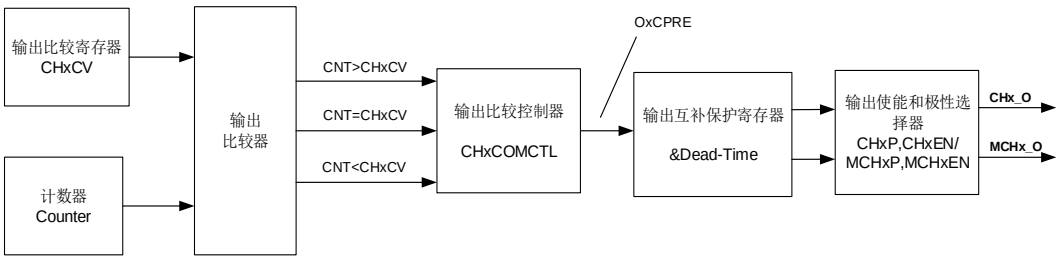


图 23-132. 输出比较逻辑（当 MCHxMSEL = 2'b11 时，x = 0）



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和TIMERx_MCHCTL2寄存器）。请参考*图23-131. 输出比较逻辑（当MCHxMSEL = 2'b00时，x = 0）*。
- 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23-132. 输出比较逻辑（当MCHxMSEL = 2'b11时，x = 0）*。

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_CCHP0寄存器中的相关位（ROS、IOS、POE和DTCFG等位）有关。详情请见*互补输出*。

在输出比较模式，TIMERx 可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的 TIMERx_CHxCV / 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。

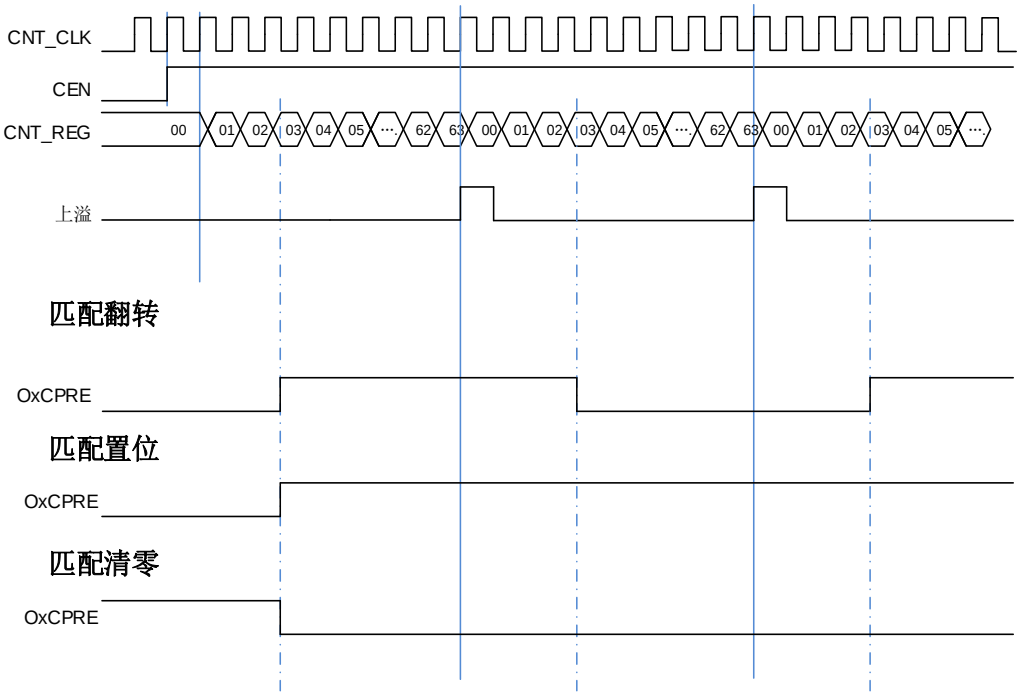
第三步：通过CHxIE/ MCHxIE / CHxDEN / MCHxDEN位配置中断 / DMA请求使能。
第四步：通过TIMERx_CAR寄存器和TIMERx_CHxCV寄存器配置输出比较时基：

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 23-133. 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 23-133. 三种输出比较模式



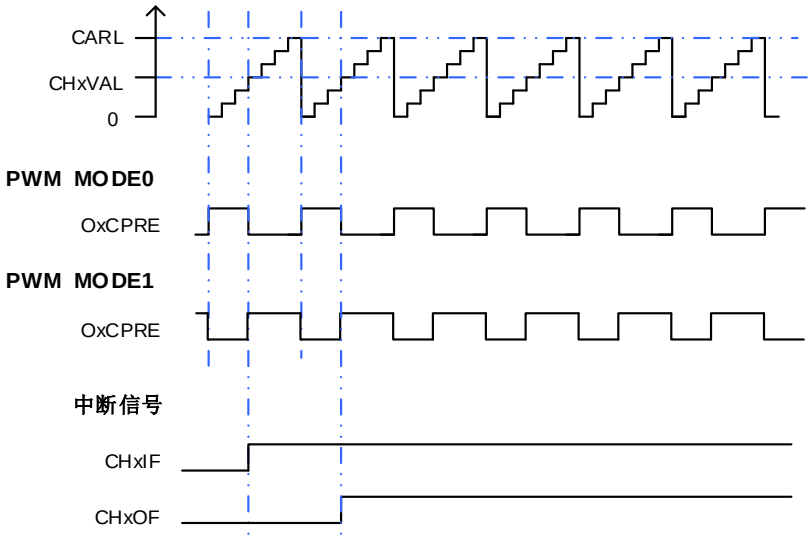
PWM 模式

在 PWM 输出模式下（PWM 模式 0 是配置 CHxCOMCTL / MCHxCOMCTL 为 4'b0110，PWM 模式 1 是配置 CHxCOMCTL / MCHxCOMCTL 为 4'b0111），通道根据 TIMERx_CAR 寄存器和 TIMERx_CHxCV / TIMERx_MCHxCV 寄存器的值，输出 PWM 波形。

EAPWM 的周期由 TIMERx_CAR 寄存器值决定，占空比由 TIMERx_CHxCV / TIMERx_MCHxCV 寄存器值决定。*图 23-134. PWM 时序图*显示了 EAPWM 的输出波形和中断。

当计数器向上计数时，在PWM0模式下（CHxCOMCTL / MCHxCOMCTL =4'b0110），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=4'b0111），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 23-134. PWM 时序图



微调模式

请参考*通用定时器 L3（TIMERx, x = 14）微调模式*。

通道输出准备信号

*图 23-131. 输出比较逻辑（当 MCHxMSEL = 2'b00 时，x = 0）*和*图 23-132. 输出比较逻辑（当 MCHxMSEL = 2'b11 时，x = 0）*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 或多模式通道 x 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 位域为 0x06 或 0x07 可以配置 PWM 模式 0 / PWM 模式 1。在这些模式中，根据计数器值和 TIMERx_CHxCV 寄存器值的关系以及计数方向，OxCPRE 信号改变其电平。具体细节描述，请参考相应的位。

设置 CHxCOMCTL =0x04 或 0x05 可以实现 OxCPRE 信号的强制输出功能。输出比较信号能够直接由软件强置为有效或无效状态，而不依赖于 TIMERx_CHxCV 的值和计数器值之间的比较结果。

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。

清除通道输出准备信号

请参考*通用定时器 L3（TIMERx, x = 14）通道输出准备信号*。

互补输出

CHx_O 和 MCHx_O 的输出具有两种情况：

- MCHxMSEL=2'b00：MCHx_O输出独立于CHx_O输出。
- MCHxMSEL=2'b11 ： MCHx_O 输出与 CHx_O 输出互补， 且 MCHx_O 的输出不由 MCHxCOMCTL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 1 对通道具有此功能。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_CCHP0 寄存器中和 TIMERx_CTL1 寄存器中的 POEN、ROS、IOS、ISOx 和 ISOxN 位。输出极性由 TIMERx_CHCTL2 寄存器中的 CHxP 和 MCHxP 位来决定。

当 CHx_O 和 MCHx_O 的输出互补时，有三种输出情况：输出使能、输出关闭状态和输出禁能，具体情况可参考*表 23-22. 由参数控制的互补输出表（MCHxMSEL = 2'b11）*。

表 23-22. 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="3">0</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O / MCHx_O = LOW
CHx_O / MCHx_O 输出禁能(1)</td></tr>
<tr><td>1</td><td colspan="2" rowspan="3">CHx_O/MCHx_O输出关闭状态(2)：
通道先输出无效电平：CHx_O = CHxP，MCHx_O = CHxNP）；如果死区产生时钟未失效，在死区时间之后：
CHx_O = ISOx，MCHx_O = ISOxN (3)</td></tr>
<tr><td rowspan="2">1</td><td>0</td></tr>
<tr><td>1</td><td>1</td></tr>
<tr><td>1</td><td>x</td><td>x</td><td colspan="2">CHx_O/MCHx_O输出关闭状态：
通道先输出无效电平：CHx_O = CHxP，MCHx_O = CHxNP）；如果死区产生时钟未失效，在死区时间之后：
CHx_O = ISOx，MCHx_O = ISOxN</td></tr>
<tr><td rowspan="4">1</td><td rowspan="4">0</td><td rowspan="4">0/1</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O/MCHx_O = LOW
CHx_O/MCHx_O输出禁能</td></tr>
<tr><td>1</td><td>CHx_O = LOW
CHx_O输出禁能</td><td>MCHx_O=OxCPRE ⊕ (2)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输出禁能</td></tr>
<tr><td>1</td><td>CHx_O=OxCPRE ⊕ CHxP
CHx_O输出使能</td><td>MCHx_O=(!OxCPRE)(3) ⊕ MCHxP
MCHx_O输出使能</td></tr>
</table>

互补参数					输出状态	
POEN	ROS	IOS	CHxEN	MCHxEN	CHx_O	MCHx_O
	1		0	0	CHx_O = CHxP CHx_O输出关闭状态	MCHx_O = MCHxP MCHx_O输出关闭状态
				1	CHx_O = CHxP CHx_O输出关闭状态	MCHx_O=OxCPRE⊕MCHxP MCHx_O输出使能
			1	0	CHx_O=OxCPRE⊕CHxP CHx_O输出使能	MCHx_O = MCHxP MCHx_O输出关闭状态
				1	CHx_O=OxCPRE⊕CHxP CHx_O输出使能	MCHx_O=(!OxCPRE)⊕ MCHxP MCHx_O输出使能

注意：

（1） 输出禁能：CHx_O / MCHx_O 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 / MCHx_O 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。

死区时间插入

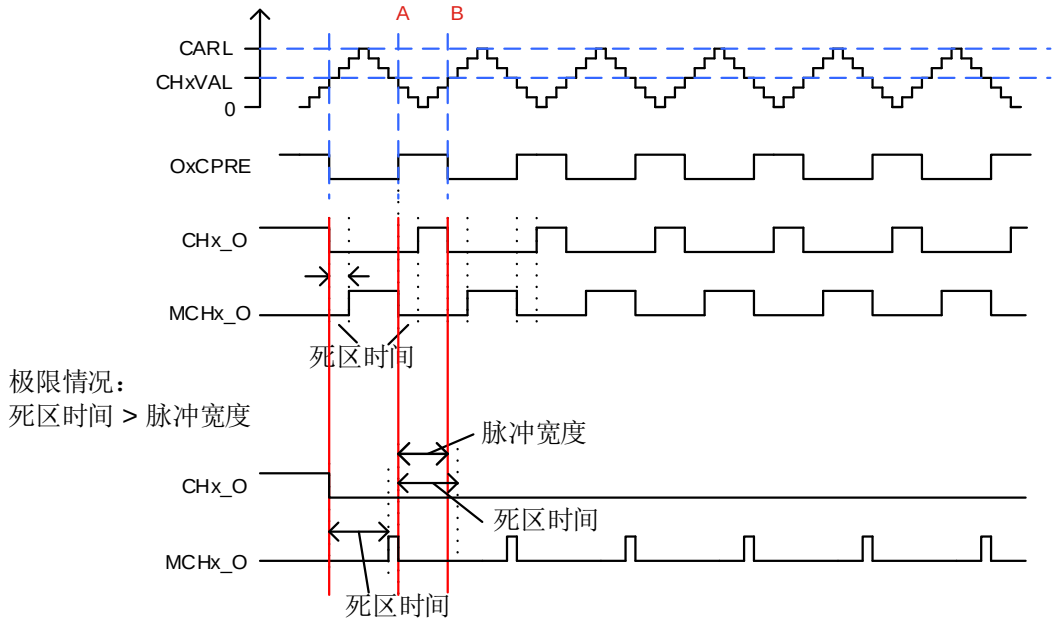
设置 MCHxMSEL=2'b11，CHxEN 和 MCHxEN 为 1'b1，同时设置 POEN=1，就可以使能死区插入功能。DTCFG 位域定义了死区时间，死区时间对所有通道有效。死区时间设置的细节请参考*互补通道保护寄存器 0（TIMERx_CCHP0）*。

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 23-135. 带死区时间的互补输出*中的 A 点，CHx_O 信号在死区时间内为低电平，直到死区时间过后才变为高电平，而 MCHx_O 信号立刻变为低电平。同样，在 B 点，计数器再次匹配（TIMERx 计数器值等于 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 23-135. 带死区时间的互补输出*所示。

图 23-135. 带死区时间的互补输出



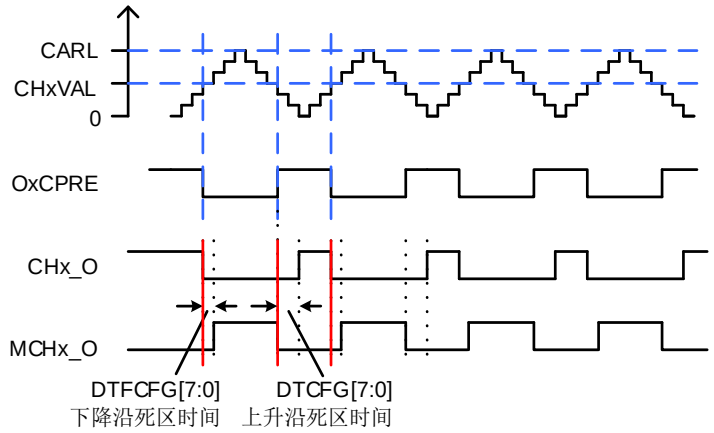
不同的死区时间插入

当DTDIFEN位（在TIMERx_CCHP1寄存器中）设置为1时，CHx_O和MCHx_O信号可以输出不同的死区时间，具体如*图23-136.不同死区时间的互补输出（DTDIFEN=1）*所示。

通道输出准备信号OxCPRE上升沿的死区时间由TIMERx_CCHP0寄存器或TIMERx_FCCHPy寄存器中的DTCFG[7:0]位域配置。OxCPRE信号的下降沿的死区时间由TIMERx_CCHP1寄存器或TIMERx_FCCHPy寄存器中的DTFCFG[7:0]位域配置。

可以在CHx_O和MCHx_O信号输出时修改死区时间。当TIMERx_CCHP1寄存器中的DTMODEN位置1时，可以使能该功能。DTCFG[7:0]位域和DTFCFG[7:0]位域的新值将会在下一次更新事件发生时生效。

图 23-136.不同死区时间的互补输出（DTDIFEN=1）



中止功能

当 MCHxMSEL = 2'b11（MCHx_O 的输出不使用 MCHxCOMCTL 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。

通用 L3 定时器具有 BREAK0 中止功能。可以通过将 TIMERx_CCHP0 寄存器中的 BRK0EN 位置 1 来使能。中止输入极性由 TIMERx_CCHP0 寄存器中的 BRK0P 位配置，电平有效。

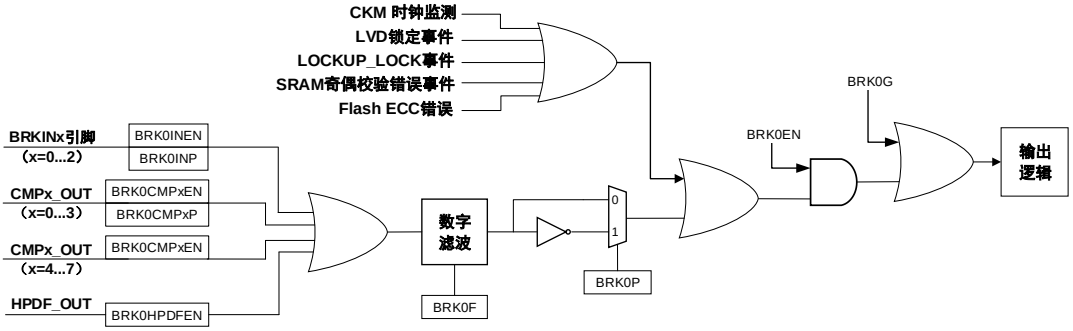
使用 BREAK0 功能时，CHx_O 和 MCHx_O 信号的输出电平由以下位控制：TIMERx_CCHP0 寄存器的 POEN、IOS 和 ROS 位，TIMERx_CTL1 寄存器的 ISOx 和 ISOxN 位。

中止事件是所有源逻辑或运算的结果。中止功能可以处理三种类型的事件源：

- 外部信号源：来自BRKIN0输入；
- 系统源：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、SRAM奇偶校验错误或FLASH ECC错误事件；
- 片上外设源：比较器输出、HPDF的看门狗输出。

BREAK0中止事件也可以由软件置位TIMERx_SWEVG寄存器中的BRK0G位产生。如*图23-137. BREAK0 中止功能逻辑图*所示，BRKIN0可以从TRIGSEL模块选择GPIO引脚，具体可参考*TIMER15_BRKIN触发选择寄存器（TRIGSEL_TIMER15BRKIN）*和*TIMER16_BRKIN触发寄存器（TRIGSEL_TIMER16BRKIN）*。

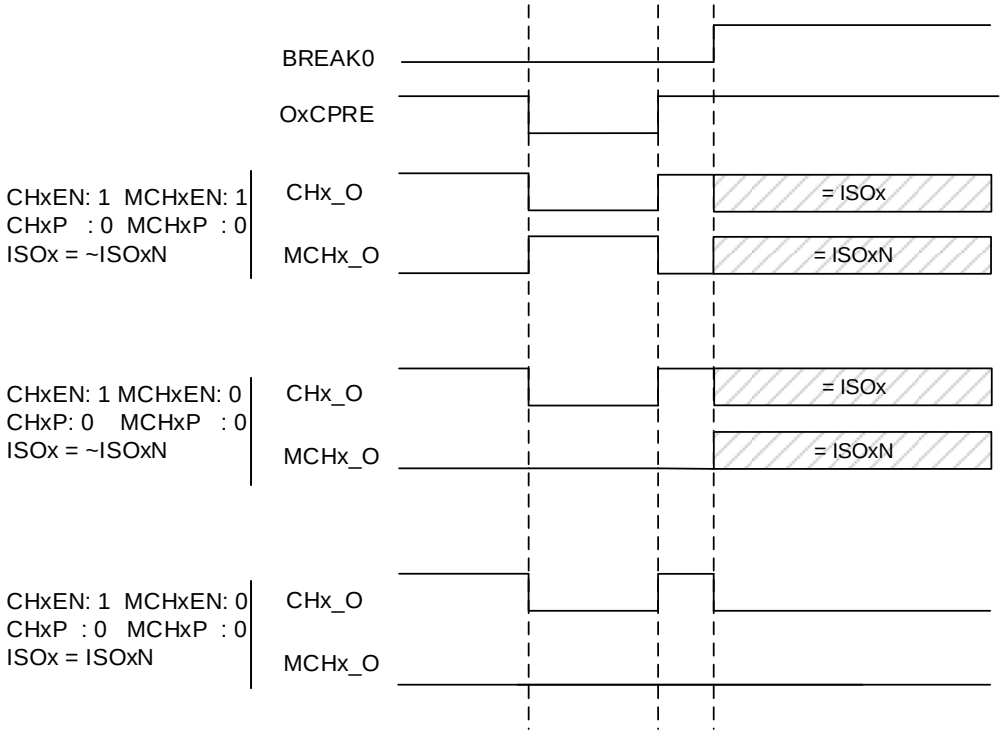
图 23-137. BREAK0 中止功能逻辑图



BREAK0可用于处理系统源、片上外设和外部输入信号源的故障，当发生BREAK0中止事件时，输出强制为无效电平，或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。

当 MCHxMSEL = 2'b11 且发生 BREAK0 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。当 IOS=1 时，通道输出情况如*图 23-138. 通道响应 BREAK0 中止输入（高电平有效）时，输出信号的行为（IOS=1）*所示，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。

图 23-138. 通道响应 BREAK0 中止输入（高电平有效）时，输出信号的行为（IOS=1）



发生中止事件时，TIMERx_INTF 寄存器的 BRK0IF/BRK1IF 位被置 1。如果 BRKIE=1，中断产生。

锁存中止功能

通用定时器L3的中止输入引脚BRKIN0具有锁存中止功能，可通过设置TIMERx_CCHP0寄存器中的BRK0LK位为1，将相应的BRKIN0引脚配置为锁存中止功能。

当使能了锁存中止功能时，需要将 BRKIN0 引脚设置为开漏模式，且低电平有效（BRK0P =0，BRK0IN0P =0）。任何中止源请求发生时，都可以将相应的 BRKIN0 引脚强制为低电平。若 BRKIN0 引脚设置为高电平有效（BRK0P=1，BRK0IN0P=1），则锁存中止功能被禁止。

当中止功能使能（将TIMERx_CCHP0寄存器中的BRK0EN=1）时，通过软件将TIMERx_SWEVG寄存器中的BRK0G位置1也可以将BRKIN0引脚强制为低电平。

当中止功能未使能（将TIMERx_CCHP0寄存器中的BRK0EN位为0）时，通过软件将BRK0G位置1，对BRKIN0引脚无影响。但BRK0F标志位会置位，通道输出为安全状态。

将 TIMERx_CCHP0 寄存器中的 BRK0REL 位置 1，可以释放 BRKIN0 引脚，当中止输入源无效时，BRK0REL 位由硬件清零，BRKIN0 引脚将恢复锁存中止功能。

在下面两种情况下，不能释放中止输入引脚 BRKIN0：

- 中止输入源有效：虽然BRK0REL位置1，释放了BRKIN0引脚，但由于中止源仍然存在，故中止事件仍然有效；
- POEN=1：通道输出使能时，即使BRK0REL位置1，也不能释放BRKIN0引脚。

表 23-23. 中止功能锁存/释放条件

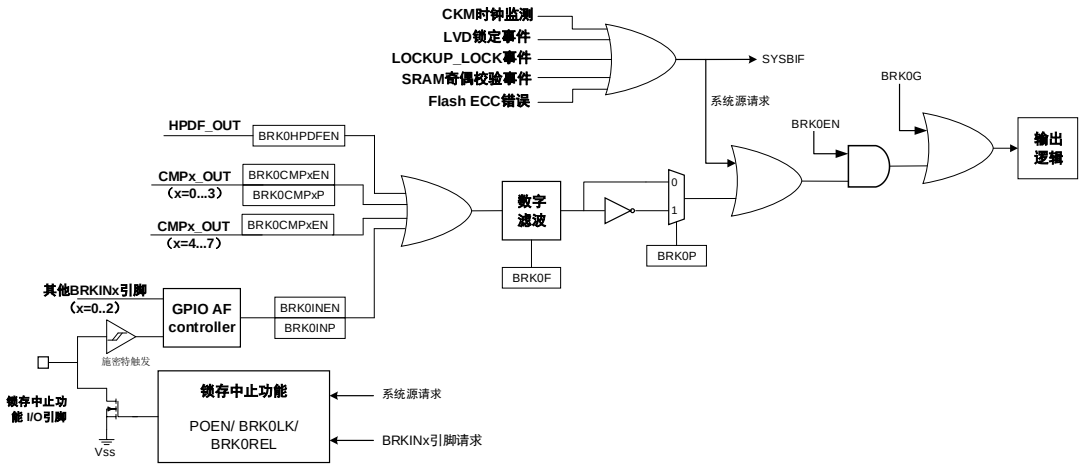
POEN	BRK0LK	BRK0REL	中止输入引脚状态
0	1	0	锁存
	1	1	释放

BREAK0 输入引脚 BRKIN0 的锁存中止功能默认是使能的（BRK0REL=0），当 BREAK0 中止事件发生时，可以通过下面的方法来重新配置锁存中止功能：

- BRK0REL=1，释放BRKIN0引脚；
- 软件等待系统中止源无效，可通过软件清除SYSBIF标志位；
- 软件轮询BRK0REL位，直到BRK0REL=0（硬件实现）。

上述过程完成后，BREAK0 锁存中止功能重新使能，此时，可通过软件将 POEN 置 1 来恢复 PWM 输出。

图 23-139. BREAK0 的 BRKIN0 引脚锁存功能逻辑图



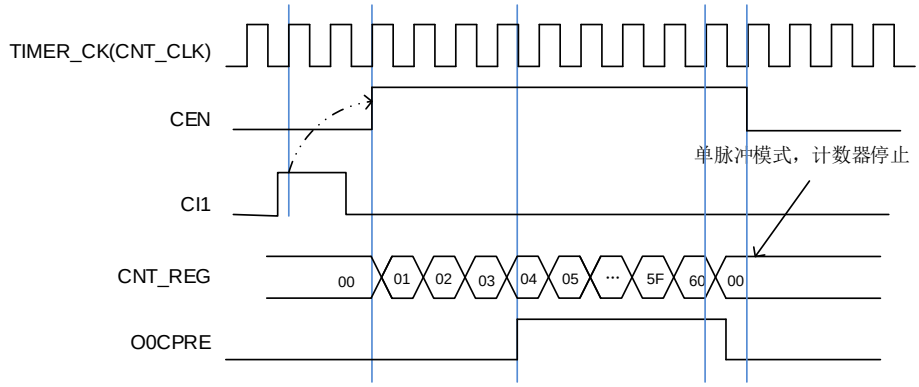
单脉冲模式

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。如果 CEN 值被硬件更新事件自动清 0，计数器将被再次初始化。

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。

图 23-140. 单脉冲模式，TIMERx_CHxCV = 0x04，TIMERx_CAR=0x60



定时器 DMA 模式

定时器 DMA 模式是指通过 DMA 模块配置定时器的寄存器。有两个跟定时器 DMA 模式相关的寄存器：TIMERx_DMACFG 和 TIMERx_DMATB。当然，必须要使能 DMA 请求，一些内部中断事件可以产生 DMA 请求。当中断事件发生，TIMERx 会给 DMA 发送请求。DMA 配置成 M2P 模式，PADDR 是 TIMERx_DMATB 寄存器地址，DMA 就会访问 TIMERx_DMATB 寄存器。实际上，TIMERx_DMATB 寄存器只是一个缓冲，定时器会将 TIMERx_DMATB 映射到一个内部寄存器，这个内部寄存器由 TIMERx_DMACFG 寄存器中的 DMATA 来指定。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值为 0，表示 1 次传输，定时器的发送 1 个 DMA 请求就可以完成。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值不为 1，例如其值为 3，表示 4 次传输，定时器就需要再多发 3 次 DMA 请求。在这 3 次请求下，DMA 对 TIMERx_DMATB 寄存器的访问会映射到访问定时器的 DMATA+0x4，DMATA+0x8, DMATA+0xc 寄存器。总之，发生一次 DMA 内部中断请求，定时器会连续发送（DMATC+1）次请求。

如果再来 1 次 DMA 请求事件，TIMERx 将会重复上面的过程。

UPIF 位备份功能

可以通过配置TIMERx_CTL0寄存器中的UPIFBUEN位来使能UPIF位的备份功能，UPIF和UPIFBU位之间没有延迟，两者完全同步。

使能该功能后，TIMERx_INTF寄存器中的UPIF位将会被实时备份到TIMERx_CNT寄存器中的UPIFBU位。这可以避免在读计数器和中断处理时产生冲突的情况。

定时器调试模式

当 Cortex®-M33 内核停止，DBG_CTL1 寄存器中的 TIMERx_HOLD 配置位被置 1，定时器计数器停止。

23.4.5. TIMERx 寄存器（x = 15 / 16）

TIMER15 基地址：0x4001 4400

TIMER16 基地址：0x4001 4800

控制寄存器 0（TIMERx_CTL0）

地址偏移：0x00
复位值：0x0000 0000

该寄存器通过字访问（32位）。

31	30	29	28	27	26	25	24	23	22	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			ADMEN	UPIFBUEN	保留	CKDIV[1:0]		ARSE	保留			SPM	UPS	UPDIS	CEN
			rw	rw		rw		rw				rw	rw	rw	rw

位/位域	名称	描述
31:13	保留	必须保持复位值。
12	ADMEN	微调模式使能 0：微调模式禁能 1：微调模式使能 注意：该位只能在CEN位为0时修改。
11	UPIFBUEN	UPIF位备份使能 0：备份禁能。UPIF位没有备份到TIMERx_CNT寄存器中的UPIFBU位 1：备份使能。UPIF 位备份到 TIMERx_CNT 寄存器中的 UPIFBU 位
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（TIMER_CK）与死区时间和采样时钟（DTS）之间的分频系数，死区发生器和数字滤波器会用到 DTS 时间。 00：$f_{DTS}=f_{TIMER_CK}$ 01：$f_{DTS}= f_{TIMER_CK}$ /2 10：$f_{DTS}= f_{TIMER_CK}$ /4 11：保留
7	ARSE	自动重载影子使能 0：禁能 TIMERx_CAR 寄存器的影子寄存器

		1：使能 TIMERx_CAR 寄存器的影子寄存器
6:4	保留	必须保持复位值。
3	SPM	单脉冲模式 0：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，CEN 硬件清零并且计数器停止计数
2	UPS	更新请求源 软件配置该为，选择更新事件源. 0：使能后，下述任一事件产生更新中断或 DMA 请求： – UPG 位被置 1 – 计数器上溢 – 从模式控制器产生的更新 1：使能后只有计数器上溢才产生更新中断或 DMA 请求
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.当以下事件之一发生时，更新事件产生，具有缓存的寄存器被装入它们的预装载值： – UPG 位被置 1 – 计数器上溢 – 从模式控制器产生一个更新事件 1：更新事件禁能。带有缓存的寄存器保持原有值，如果 UPG 位被置 1 或者从模式控制器产生一个硬件复位事件，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。触发模式可以自动地通过硬件设置 CEN 位。

控制寄存器 1（TIMERx_CTL1）

地址偏移：0x04
复位值：0x0000 0000

该寄存器通过字访问（32位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCUC[2:1]		保留													
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						ISO0N	ISO0	保留				DMAS	CCUC[0]	保留	CCSE

rw rw rw rw rw

位/位域	名称	描述
31:30	CCUC[2:1]	换相控制影子寄存器更新控制 请参考 CCUC [0]的描述。
29:10	保留	必须保持复位值。
9	ISO0N	多模式通道 0 的互补通道空闲状态输出 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_CCHP0 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_CCHP0 寄存器的 PROT[1:0]位为 00 的时候可以被更改。
7:4	保留	必须保持复位值。
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 CHx / MCHx 的 DMA 请求 1：当更新事件发生，发送通道 CHx / MCHx 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： 000：CMTG 位被置 1 时，更新影子寄存器 001：保留 100：当计数器上溢事件发生时，影子寄存器更新 其他值：保留 当通道没有互补输出时，此位无效。
1	保留	必须保持复位值。
0	CCSE	换相控制影子使能 0：影子寄存器 CHxEN、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

DMA 和中断使能寄存器（TIMERx_DMAINTEN）

地址偏移：0x0C

复位值：0x0000 0000

该寄存器通过字访问（32位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留							MCH0DEN	保留			MCH0IE	保留			
							rw				rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CH0DEN	UPDEN	BRKIE	保留	CMTIE	保留			CH0IE	UPIE
						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	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] = 2'b00 时）。
23:21	保留	必须保持复位值。
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] = 2'b00 时）。
19:10	保留	必须保持复位值。
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	保留	必须保持复位值。
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
4:2	保留	必须保持复位值。
1	CH0IE	通道 0 比较/捕获中断使能 0：禁止通道 0 中断 1：使能通道 0 中断
0	UPIE	更新中断使能 0：禁止更新中断 1：使能更新中断

中断标志寄存器（TIMERx_INTF）

地址偏移：0x10

复位值：0x0000 0000

该寄存器通过字访问（32位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留							MCH0OF	保留			MCH0IF	保留			
							rc_w0				rc_w0				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CH0OF	保留	BRK0IF	保留	CMTIF	保留			CH0IF	UPIF
						rc_w0		rc_w0		rc_w0				rc_w0	rc_w0

位/位域	名称	描述
31:25	保留	必须保持复位值。
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
23:21	保留	必须保持复位值。
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
19:10	保留	必须保持复位值。
9	CH0OF	通道 0 捕获溢出标志 当通道 0 被配置为输入模式时，在 CH0IF 标志位已经被置 1 后，捕获事件再次发生时，该标志位可以由硬件置 1。该标志位由软件清 0。 0：无捕获溢出中断发生 1：发生了捕获溢出中断
8	保留	必须保持复位值。
7	BRK0IF	BREAK0 中断标志位 一旦 BREAK0 输入有效，由硬件对该位置 1。如果 BREAK0 输入无效，则该位可由软件清 0。 0：无 BREAK0 事件产生 1：BREAK0 输入上检测到有效电平
6	保留	必须保持复位值。
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
4:2	保留	必须保持复位值。
1	CH0IF	通道 0 比较/捕获中断标志 此标志由硬件置 1 软件清 0。当通道 0 在输入模式下时，捕获事件发生时此标志位被置 1；当通道 0 在输出模式下时，此标志位在一个比较事件发生时被置 1。 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

该寄存器通过字访问（32位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留											MCH0G	保留			
											w				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								BRK0G	保留	CMTG	保留			CH0G	UPG
								w		w				w	w

位/位域	名称	描述
31:21	保留	必须保持复位值。
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	BRK0G	产生 BREAK0 事件 该位由软件置 1，用于产生一个 BREAK0 事件，由硬件自动清 0。当此位被置 1 时，POEN 位被清 0 且 BRK0IF 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 BREAK0 事件 1：产生 BREAK0 事件
6	保留	必须保持复位值。
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） 的互补输出被更新（根据 TIMERx_CTL1 中 CCSE 值）。 0：不产生通道控制更新事件 1：产生通道控制更新事件
4:2	保留	必须保持复位值。
1	CH0G	通道 0 捕获或比较事件发生 该位由软件置 1，用于在通道 0 产生一个捕获/比较事件，由硬件自动清 0。当此位被置 1，CH0IF 标志位被置 1，若开启对应的中断和 DMA，则发出相应的中断和 DMA 请求。此外，如果通道 0 配置为输入模式，计数器的当前值被 TIMERx_CH0CV 寄存器捕获，如果 CH0IF 标志位已经为 1，则 CH0OF 标志位被置 1。

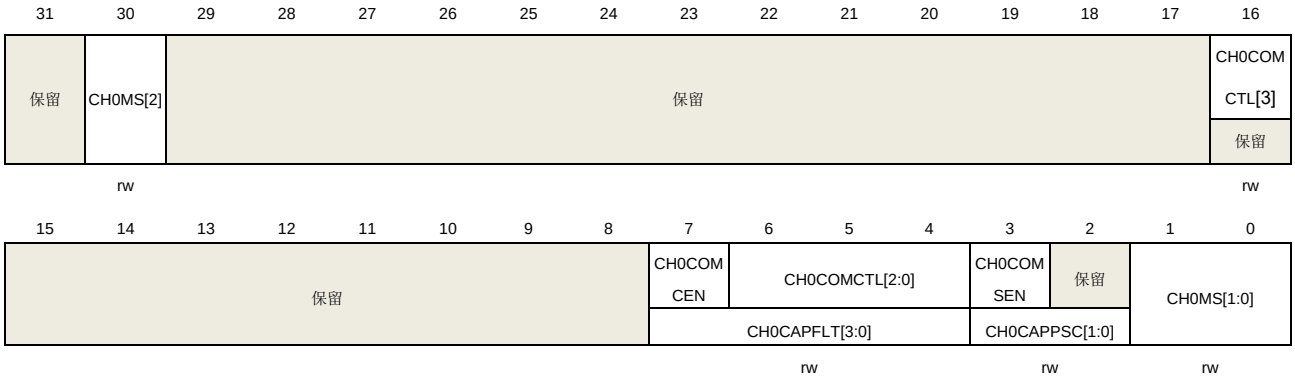
0：不产生通道 0 捕获或比较事件
1：发生通道 0 捕获或比较事件

0 UPG 更新事件产生
此位由软件置 1，被硬件自动清 0。当此位被置 1，向上计数模式，计数器被清 0，预分频计数器将同时被清除。
0：无更新事件产生
1：产生更新事件

通道控制寄存器 0（TIMERx_CHCTL0）

地址偏移：0x18
复位值：0x0000 0000

该寄存器通过字访问（32位）。



输出比较模式：

位/位域	名称	描述
31	保留	必须保持复位值。
30	CH0MS[2]	通道 0 I/O 模式选择 参考 CH0MS[1:0]描述。
29:17	保留	必须保持复位值。
16	CH0COMCTL[3]	通道 0 输出比较控制 参见 CH0COMCTL[2:0]描述
15:8	保留	必须保持复位值。
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 CH0COMCTL[3]和 CH0COMCTL[2:0]位域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] = 2'b11 时，CH0COMCTL[3]和 CH0COMCTL[2:0]位域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。 0000：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 0001：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。 0010：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。 0011：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。 0100：强制为低。强制 O0CPRE 为低电平 0101：强制为高。强制 O0CPRE 为高电平 0110：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。 0111：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。 1000~1111：保留 **注意**： 在 PWM 模式 0 或 PWM 模式 1 中，只有当比较结果改变了或者输出比较模式中从时基模式切换到 PWM 模式时，O0CPRE 电平才改变。 当 CH0 和 MCH0 输出互补时，该位域预装载。若 CCSE =1，则该位域只在通道换相事件发生时更新。 当 TIMERx_CCHP0 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（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_CCHP0 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] = 2'b00 时，TIMERx_CHCTL2 寄存器的 CH1EN 位清 0；当 MCH0MSEL[1:0] = 2'b01 或 2'b11 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：保留 011：保留 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	保留	必须保持复位值。
30	CH1CAPFLT[3:0]	通道 1 输入捕获滤波控制 参见 CH0CAPFLT 描述。
29:8	保留	必须保持复位值。
7:4	CH0CAPFLT[3:0]	通道 0 输入捕获滤波控制 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 CI0 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	CH0CAPPSC[1:0]	通道 0 输入捕获预分频器 这 2 位定义了通道 0 输入的预分频系数。当 TIMERx_CHCTL2 寄存器中的 CH0EN =0 时，则预分频器复位。 00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获 01：每 2 个事件触发一次捕获 10：每 4 个事件触发一次捕获 11：每 8 个事件触发一次捕获
1:0	CH0MS[1:0]	通道 0 模式选择 与输出比较模式相同。

通道控制寄存器 2（TIMERx_CHCTL2）

地址偏移：0x20

复位值：0x0000 0000

该寄存器通过字访问（32位）。

31	30	29	28	27	26	25	24	23	22	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												MCH0P	MCH0EN	CH0P	CH0EN
												rw	rw	rw	rw

位/位域	名称	描述
31:4	保留	必须保持复位值。
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] = 2'b11 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_CCHP0 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_CCHP0 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

该寄存器通过字访问（32位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UPIFBU	保留														
r															
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	UPIFBU	UPIF位备份 该位只读，是 TIMERx_INTF 寄存器的 UPIF 位的备份值。当 UPIFBUEN = 1 时，该位有效，若 UPIFBUEN =0，该位保留，读取该位值为零。
30:16	保留	必须保持复位值。

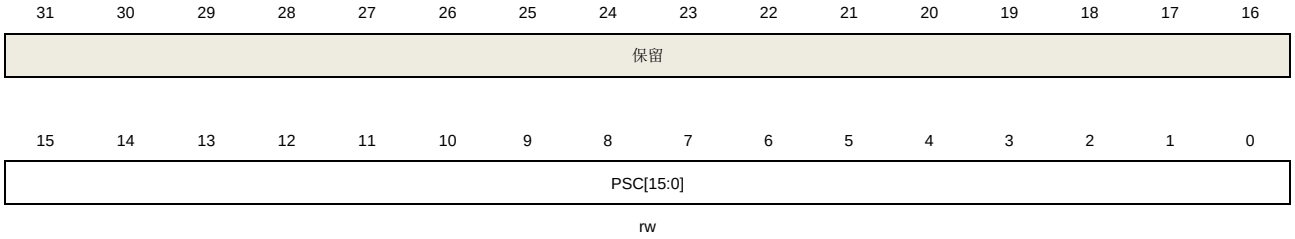
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。 当 PWMADMEN = 0 时，该位域表示当前计数器的值。写操作该位域可以改变计数器的值。 当 PWMADMEN = 1 时，该位域仅用于表示计数器值的整数部分，不包含小数部分。

预分频寄存器（TIMERx_PSC）

地址偏移： 0x28

复位值：0x0000 0000

该寄存器通过字访问（32位）。



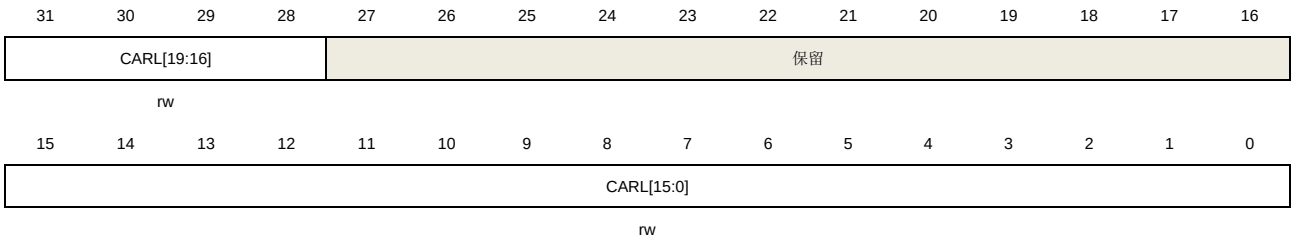
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 PSC 时钟除以（PSC+1），每次当更新事件产生时，PSC 的值被装入当前预分频寄存器。

计数器自动重载寄存器（TIMERx_CAR）

地址偏移：0x2C

复位值：0x0000 FFFF

该寄存器通过字访问（32位）。



位/位域	名称	描述
31:28	CARL[19:16]	计数器自动重载值（16 位至 19 位） 当 PWMADMEN = 0 时，CARL[19:16]位域值 0000。 当 PWMADMEN = 1 时，CARL[19:16]位域用于表示自动重载值的小数部分。

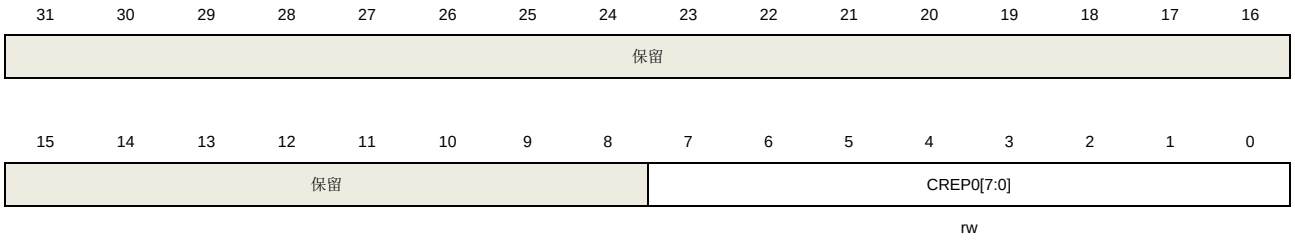
27:16	保留	必须保持复位值。
15:0	CARL[15:0]	计数器自动重载值（0 位至 15 位） 这些位定义了计数器的自动重载值。 当 PWMADMEN = 0 时，CARL[15:0]位域用于表示计数器的自动重载值。 当 PWMADMEN = 1 时，CARL[15:0]位域用于表示自动重载值的整数部分。

重复计数寄存器 0（TIMERx_CREP0）

地址偏移：0x30

复位值：0x0000 0000

该寄存器通过字访问（32位）。



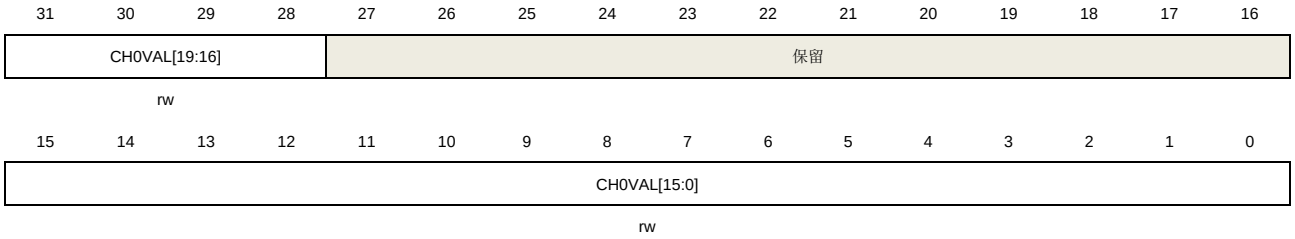
位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	CREP0[7:0]	重复计数器值 0 这些位定义了更新事件的产生速率。重复计数器计数值减为 0 时产生更新事件。影子寄存器的更新速率也会受这些位影响（前提是影子寄存器被使能）。 **注意**：当 TIMERx_CFG 寄存器中的 CREPSEL =0 时，使用该位。

通道 0 捕获/比较寄存器（TIMERx_CH0CV）

地址偏移：0x34

复位值：0x0000 0000

该寄存器通过字访问（32位）。



位/位域	名称	描述

31:28	CH0VAL[19:16]	通道 0 的捕获或比较值（16 位到 19 位） 当通道 0 配置为输入模式时，CH0VAL[19:16]位域值为 0000。 当通道 0 配置为输出模式时，该位域包含了即将和计数器比较的值。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。 当 PWMADMEN = 0 时，CH0VAL[19:16]位域值为 0000。 当 PWMADMEN = 1 时，CH0VAL[19:16]位域用于表示比较值的小数部分。
27:16	保留	必须保持复位值。
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值（0 位至 15 位） 当通道 0 配置为输入模式时，CH0VAL[15:0]决定了上次捕获事件的计数器值，且该位域只读。 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。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。 当 PWMADMEN = 0 时，CH0VAL[15:0]位域值表示比较值。 当 PWMADMEN = 1 时，CH0VAL[15:0]位域用于表示比较值的整数部分。

互补通道保护寄存器 0（TIMERx_CCHP0）

地址偏移：0x44

复位值：0x0000 0000

该寄存器通过字访问（32位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			BRK0LK	保留	BRK0REL	保留						BRK0F[3:0]			
	rw		rw		rw									rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POEN	OAEN	BRK0P	BRK0EN	ROS	IOS	PROT[1:0]		DTCFG[7:0]							
rw	rw	rw	rw	rw	rw	rw		rw							

位/位域	名称	描述
31:29	保留	必须保持复位值。
28	BRK0LK	BREAK0输入锁存 0：BREAK0输入为输入模式 1：BREAK0输入为锁存模式 当BRK0LK置1时，BREAK0输入配置为开漏输出模式。 任何有效的BREAK0事件都会拉低BREAK0输入引脚电平，用于向外部设备提示有内部BREAK0事件发生。 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。 **注意**：对该位的每一次写操作，需要延时 1 个 APB 时钟才有效。

27	保留	必须保持复位值。
26	BRK0REL	BREAK0输入释放 当BREAK0输入无效时，该位由硬件清零。 0：BREAK0输入锁存 1：BREAK0输入释放 当软件将该位置1时，将释放锁存输出控制（高阻态的开漏模式）。当BREAK0事件无效时，该位由硬件清零。 **注意**：对该位的每一次写操作，需要延时 1 个 APB 时钟才有效。
25:20	保留	必须保持复位值。
19:16	BRK0F[3:0]	BREAK0输入信号滤波 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 BREAK0 输入信号的采样频率和数字滤波器的长度。 0000：无滤波器，BREAK0 异步有效，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 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0] = 00 时才可修改。
15	POEN	所有的通道输出使能 根据 OAEN 位，该位可以软件设置或者硬件自动设置。一旦中止输入有效，该位被硬件异步清 0。如果一个通道配置为输出模式，如果设置了相应的使能位（TIMERx_CHCTL2 寄存器的 CHxEN，CHxNEN 位），则开启 CHx_O 和 MCHx_O 输出。 0：禁止通道输出或强制为空闲状态 1：使能通道输出
14	OAEN	自动输出使能 此位定义了 POEN 位是否可以被硬件自动置 1。

		0：POEN 位不能被硬件置 1 1：如果中止输入无效，下一次更新事件发生时，POEN 位能被硬件自动置 1 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0] = 00 时才可修改。
13	BRK0P	BREAK0 输入信号极性 此位定义了 BREAK0 输入的极性。 0：BREAK0 输入低电平有效 1：BREAK0 输入高电平有效 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0] = 00 时才可修改。
12	BRK0EN	BREAK0 输入信号使能 此位置 1 使能 BREAK0 输入信号。 0：BREAK0 输入禁能 1：BREAK0 输入使能 此位只有在 TIMERx_CCHP0 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 23-22. 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_CCHP0 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 23-22. 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_CCHP0 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_CCHP0 寄存器中 BRK0EN / BRK0P / 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_CCHP0 寄

存器中 ROS / IOS 位。

11：PROT 模式 2。除了 PROT 模式 1 下的寄存器写保护外，还有 TIMERx_CHCTLR0 / 1 中 CHxCOMCTL / CHxCOMSEN 位（如果相关通道配置为输出模式）写保护。

11：PROT 模式 2。除了 PROT 模式 1 下的寄存器写保护外，还有 TIMERx_CHCTLR0 及 TIMERx_MCHCTL0 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_CCHP0 寄存器被写入，这两位被写保护

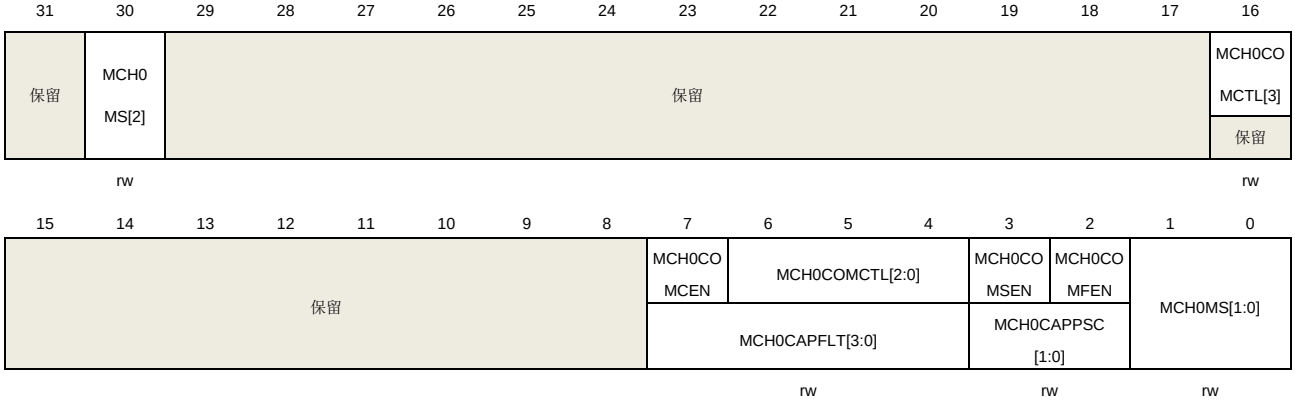
7:0	DTCFG[7:0]	死区时间控制 这些位定义了插入互补输出之间的死区持续时间。DTCFG 值和死区时间的关系如下： DTCFG [7:5] = 3'b0xx：DTvalue = DTCFG [7:0] x t_{DT}, t_{DT} = t_{DTS} DTCFG [7:5] = 3'b 10x：DTvalue =（64+DTCFG [5:0]）x t_{DT}，t_{DT} = t_{DTS} * 2 DTCFG [7:5] = 3'b 110：DTvalue =（32+DTCFG [4:0]）x t_{DT}，t_{DT}= t_{DTS} * 8 DTCFG [7:5] = 3'b 111：DTvalue =（32+DTCFG [4:0]）x t_{DT}，t_{DT} = t_{DTS} * 16 此位只有在 TIMERx_CCHP0 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。



输出比较模式：

位/位域	名称	描述
31	保留	必须保持复位值。
30	MCH0MS[2]	多模式通道 0 I/O 模式选择 参考MCH0MS[1:0]描述。

29:17	保留	必须保持复位值。
16	MCH0COMCTL[3]	多模式通道 0 输出比较控制 请参考MCH0COMCTL[2:0]描述。
15:8	保留	必须保持复位值。
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 输出比较控制 当多模式通道 0 配置为输出模式，并且 MCH0MSEL[1:0] = 2'b00，MCH0COMCTL[3]和 MCH0COMCTL[2:0]位域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] = 2'b11 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。 0000：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 0001：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。 0010：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。 0011：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。 0100：强制为低。强制 MO0CPRE 为低电平 0101：强制为高。强制 MO0CPRE 为高电平 0110：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。 0111：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。 1000~1111：保留 **注意**： 在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或者输出比较模式中从时基模式切换到 PWM 模式时，MO0CPRE 电平才改变。 当 CH0 和 MCH0 输出互补时，该位域预装载。若 CCSE = 1，则该位域只在通道换

		相事件发生时更新。 当TIMERx_CCHP0寄存器的PROT[1:0] = 11且MCH0MS = 000（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_CCHP0寄存器的PROT [1:0] = 11且CH0MS = 00时此位不能被改变。
2	MCH0COMFEN	多模式通道0输出比较快速使能 当该位为1时，如果通道配置为PWM模式0或者PWM模式1，会加快捕获 / 比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，MCH0_O被设置为比较电平而与比较结果无关。 0：多模式通道0输出比较快速禁能。当触发器的输入有一个有效沿时，激活MCH0_O输出的最小延时为5个时钟周期 1：多模式通道0输出比较快速使能。当触发器的输入有一个有效沿时，激活MCH0_O输出的最小延时为3个时钟周期
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：保留 011：保留 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	保留	必须保持复位值。
30	MCH0MS[2]	多模式通道 0 I/O 模式选择 参考MCH0MS[1:0]描述。
29:8	保留	必须保持复位值。
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 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														MCH0FP[1:0]	
														rw	

位/位域	名称	描述
31:2	保留	必须保持复位值。
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] = 2'b00，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_CCHP0 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。

控制寄存器 2（TIMERx_CTL2）

地址偏移：0x74

复位值：0x0030 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留										MCH0MSEL[1:0]		保留			
										rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															

位/位域	名称	描述
31:22	保留	必须保持复位值。
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：保留 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
19:0	保留	必须保持复位值。

TIMERx 备用功能控制寄存器 0（TIMERx_AFCTL0）

地址偏移：0x8C

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			BRK0CMP3P	BRK0CMP2P	BRK0CMP1P	BRK0CMP0P	保留								BRK0IN0P
			rw	rw	rw	rw									rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRK0CMP6EN	BRK0CMP5EN	BRK0CMP4EN	BRK0CMP3EN	BRK0CMP2EN	BRK0CMP1EN	BRK0CMP0EN	BRK0HPDFEN	BRK0CMP7EN	保留						BRK0IN0EN
rw	rw	rw	rw	rw	rw	rw	rw	rw							rw

位/位域	名称	描述

31:29	保留	必须保持复位值。
28	BRK0CMP3P	BREAK0 CMP3输入极性 该位用于配置CMP3输入极性，具体极性是由该位和BRK0P位共同确定。 0：CMP3输入信号不反相（BRK0P = 0，输入信号低有效；BRK0P = 1，输入信号高有效） 1：CMP3输入信号反相（BRK0P = 0，输入信号高有效；BRK0P = 1，输入信号低有效） 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
27	BRK0CMP2P	BREAK0 CMP2输入极性 该位用于配置CMP2输入极性，具体极性是由该位和BRK0P位共同确定。 0：CMP2输入信号不反相（BRK0P = 0，输入信号低有效；BRK0P = 1，输入信号高有效） 1：CMP2输入信号反相（BRK0P = 0，输入信号高有效；BRK0P = 1，输入信号低有效） 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
26	BRK0CMP1P	BREAK0 CMP1输入极性 该位用于配置CMP1输入极性，具体极性是由该位和BRK0P位共同确定。 0：CMP1输入信号不反相（BRK0P = 0，输入信号低有效；BRK0P = 1，输入信号高有效） 1：CMP1输入信号反相（BRK0P = 0，输入信号高有效；BRK0P = 1，输入信号低有效） 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
25	BRK0CMP0P	BREAK0 CMP0输入极性 0：CMP0输入信号不反相（BRK0P = 0，输入信号低有效；BRK0P = 1，输入信号高有效） 1：CMP输入信号反相（BRK0P = 0，输入信号高有效；BRK0P = 1，输入信号低有效） 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
24:17	保留	必须保持复位值。
16	BRK0IN0P	BREAK0 BRKIN0备用输入极性 该位用于配置BRKIN0输入极性，具体极性是由该位和BRK0P位共同确定。 0：BRKIN0输入信号不反相（BRK0P =0，输入信号低有效；BRK0P = 1，输入信号高有效） 1：BRKIN0输入信号反相（BRK0P =0，输入信号高有效；BRK0P = 1，输入信号低有效） 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。

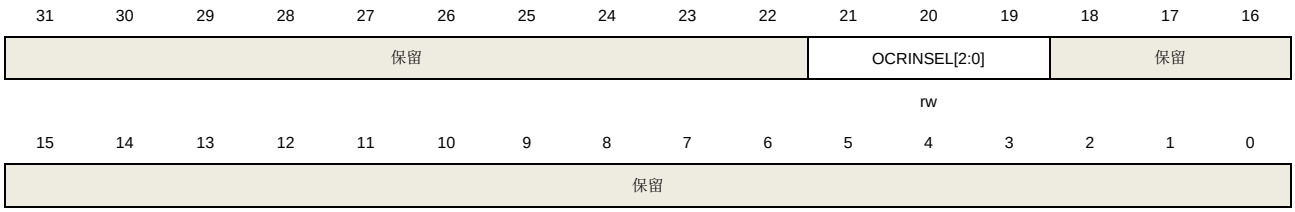
15	BRK0CMP6EN	BREAK0 CMP6输入使能 0：CMP6输入禁能 1：CMP6输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
14	BRK0CMP5EN	BREAK0 CMP5输入使能 0：CMP5输入禁能 1：CMP5输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
13	BRK0CMP4EN	BREAK0 CMP4输入使能 0：CMP4输入禁能 1：CMP4输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
12	BRK0CMP3EN	BREAK0 CMP3输入使能 0：CMP3输入禁能 1：CMP3输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
11	BRK0CMP2EN	BREAK0 CMP2输入使能 0：CMP2输入禁能 1：CMP2输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
10	BRK0CMP1EN	BREAK0 CMP1输入使能 0：CMP1输入禁能 1：CMP1输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
9	BRK0CMP0EN	BREAK0 CMP0输入使能 0：CMP0输入禁能 1：CMP0输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
8	BRK0HPDFEN	BREAK0 HPDF输入使能 0：HPDF输入禁能 1：HPDF输入使能 **注意**：HPDF输入对于TIMER15（HPDF_BREAK[1]）和TIMER16（HPDF_BREAK[2]）是不同的。 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
7	BRK0CMP7EN	BREAK0 CMP7输入使能 0：CMP7输入禁能

		1：CMP7输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
6:1	保留	必须保持复位值。
0	BRK0IN0EN	BREAK0 BRKIN0备用输入使能 0：BRKIN0输入禁能 1：BRKIN0输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。

备用功能控制寄存器 1（TIMERx_AFCTL1）

地址偏移：0x90
复位值：0x0000 0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:22	保留	必须保持复位值。
21:19	OCRINSEL[2:0]	OCPRE_CLR输入选择 000：OCPRE_CLR0 001：OCPRE_CLR1 … 111：OCPRE_CLR7

OCPRE_CLR 输入选择	TIMER15 / 16
OCPRE_CLR0	CMP0_OUT
OCPRE_CLR1	CMP1_OUT
OCPRE_CLR2	CMP2_OUT
OCPRE_CLR3	CMP3_OUT
OCPRE_CLR4	CMP4_OUT
OCPRE_CLR5	CMP5_OUT
OCPRE_CLR6	CMP6_OUT
OCPRE_CLR7	CMP7_OUT

此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。

18:0	保留	必须保持复位值。

重复计数寄存器 1（TIMERx_CREP1）

地址偏移：0x98
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CREP1[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CREP1[15:0]															
rw															

位/位域	名称	描述
31:0	CREP1[31:0]	重复计数器的值 1 该位域为 32 位，只读。 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）。 **注意**：当TIMERx_CFG寄存器中的CREPSEL=1时，使用该位域。

互补通道保护寄存器 1（TIMERx_CCHP1）

地址偏移：0x09C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:18	保留	必须保持复位值。
17	DTMODEN	死区时间在线修改功能使能 0：死区时间在线修改功能禁能

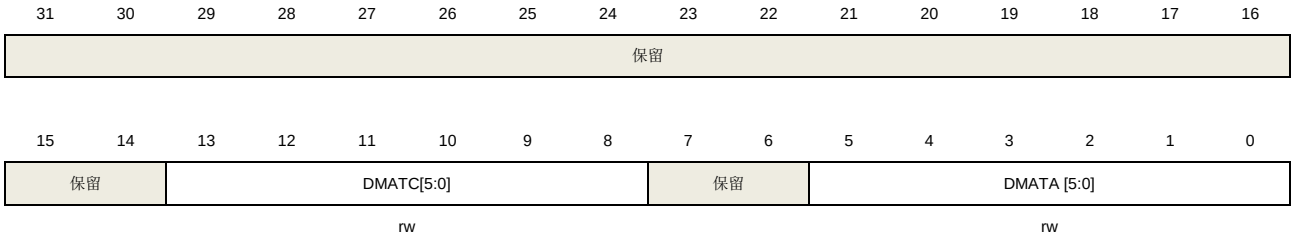
		1：死区时间在线修改功能使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。 **注意**：如果在CEN=1时将该位置1，则在上次更新事件之后写入的任何值（在DTCFG[7:0]位域和DTCFG[7:0]位域）都是无效的，只使用前一次的值。
16	DTDIFEN	死区时间不同配置使能 0：上升沿和下降沿的死区时间（在 TIMERx_CCHP1 寄存器的 DTCFG[7:0]位域配置）相同。 1：上升沿的死区时间由 TIMERx_CCHP1 寄存器的 DTCFG[7:0]位域配置；下降沿的死区时间由 TIMERx_CCHP1 寄存器的 DTFCFG[7:0]位域配置。 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
15:8	保留	必须保持复位值。
7:0	DTFCFG[7:0]	下降沿死区时间控制 该位域控制 OxCPRE 信号下降沿的死区时间值，该死区时间在输出转换之前插入。 DTCFG 值与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’b 10x：DTvalue =（64+DTCFG [5:0]）x t_{DT}，t_{DT} = t_{DTS} * 2 DTCFG [7:5] = 3’b 110：DTvalue =（32+DTCFG [4:0]）x t_{DT}，t_{DT}= t_{DTS} * 8 DTCFG [7:5] = 3’b 111：DTvalue =（32+DTCFG [4:0]）x t_{DT}，t_{DT} = t_{DTS} * 16 此位只有在 TIMERx_CCHP0 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。



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。 6’b000000：传输1次 6’b000001：传输2次

		… 6’b111000：传输 57 次
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_DMAVB 寄存器的第一个地址。当通过 TIMERx_DMA 第一次访问时，访问的就是该位域指定的地址。第二次访问 TIMERx_DMATB 时，将访问起始地址+0x4。 6’b0_0000：TIMERx_CTL0 6’b0_0001：TIMERx_CTL1 … 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
DMATB[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMATB[15:0]															
rw															

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

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	CCUSEL	CREPSEL	CHVSEL	OUTSEL
	rw	rw	rw	rw

位/位域	名称	描述
31:4	保留	必须保持复位值。
3	CCUSEL	换相控制影子寄存器更新选择 只有当CCUC[2:0]位域配置为100,101和110时，该位才有效。 0：当计数器产生一个上溢/下溢事件时，影子寄存器才更新 1：当重复计数器值为0，且计数器产生一个上溢/下溢事件时，影子寄存器才更新
2	CREPSEL	计数器重复寄存器选择 该位用于选择重复计数寄存器。 0：更新事件的速率由TIMERx_CREP0寄存器确定 1：更新事件的速率由 TIMERx_CREP1 寄存器确定
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

23.5. 基本定时器（TIMERx, x = 5, 6）

23.5.1. 简介

基本定时器（TIMER5 / 6）包含一个无符号 16 位计数器。可以被用作通用定时器和为 DAC（数字到模拟转换器）提供时钟。基本定时器可以配置产生 DMA 请求，TRGO0 触发连接到 DAC。

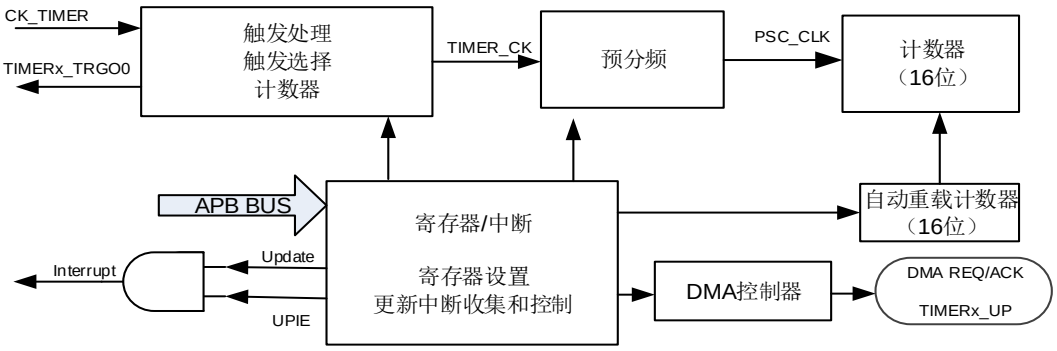
23.5.2. 主要特征

- 计数器宽度：16位（TIMER5 / 6）；
- 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。

23.5.3. 结构框图

*图 23-141. 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 23-141. 基本定时器结构框图



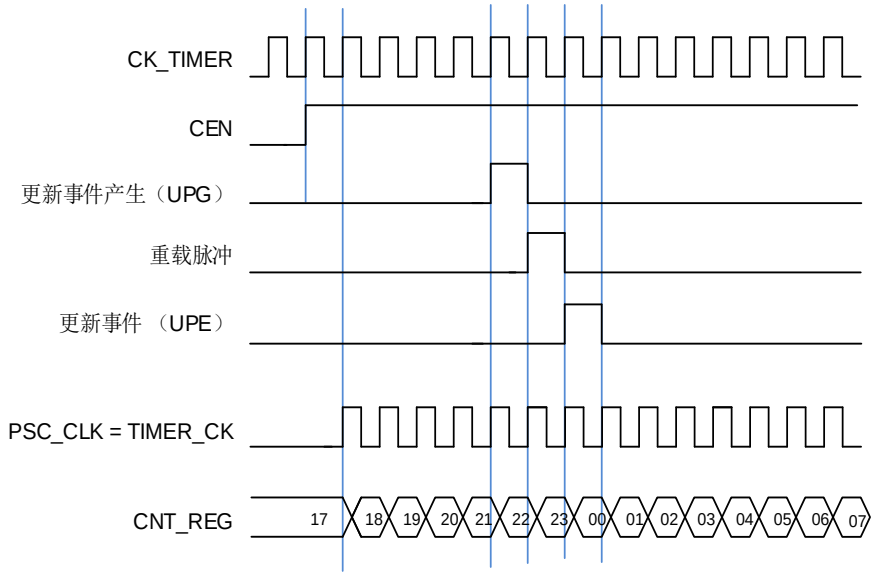
23.5.4. 功能描述

时钟源选择

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。

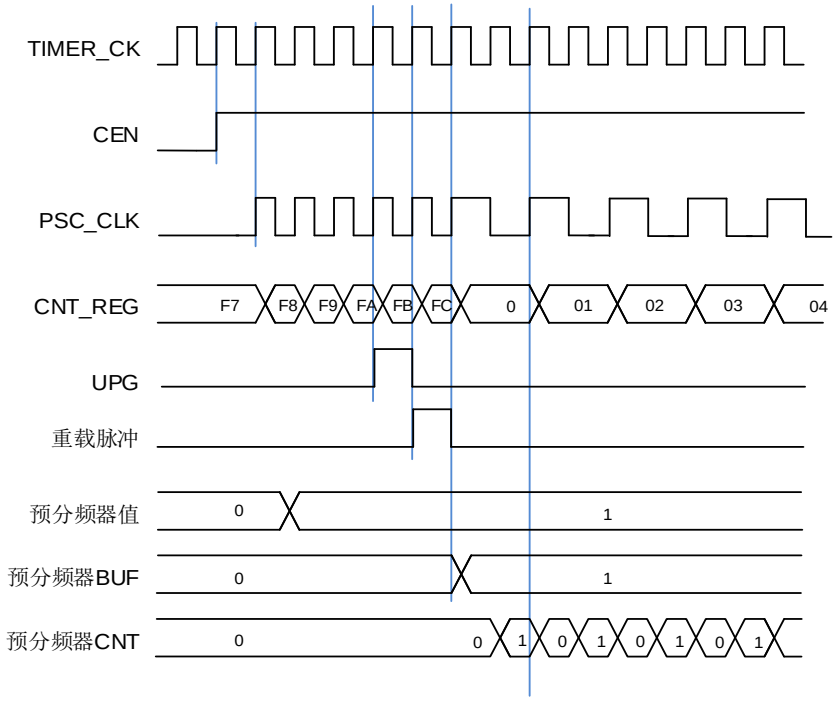
图 23-142. 内部时钟分频为 1 时正常模式下的控制电路



预分频

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 23-143. 当预分频器的参数从 1 变到 2 时，计数器的时序图



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，所有的寄存器（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 23-144. 向上计数时序图，PSC = 0 / 2

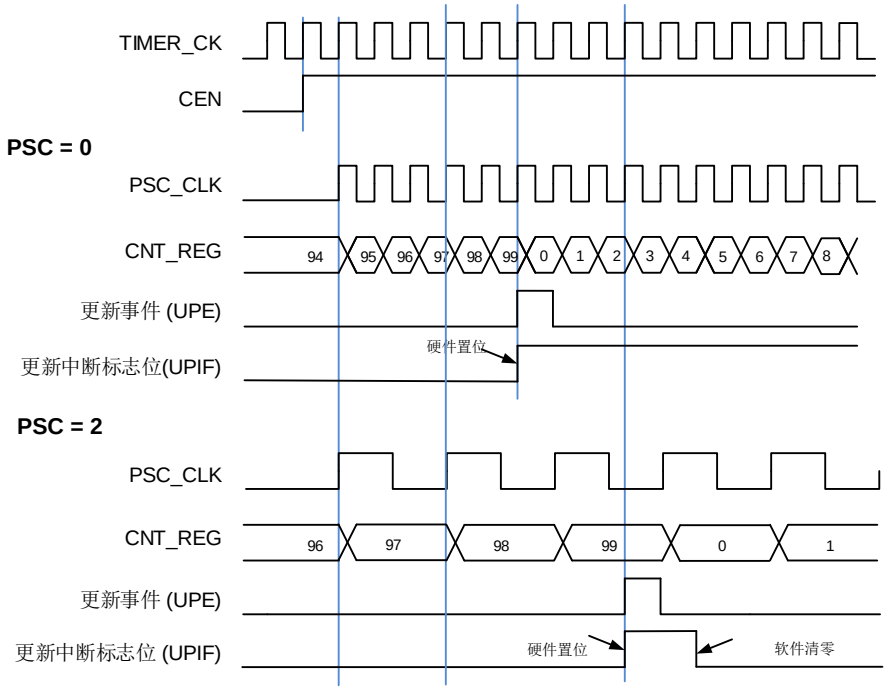
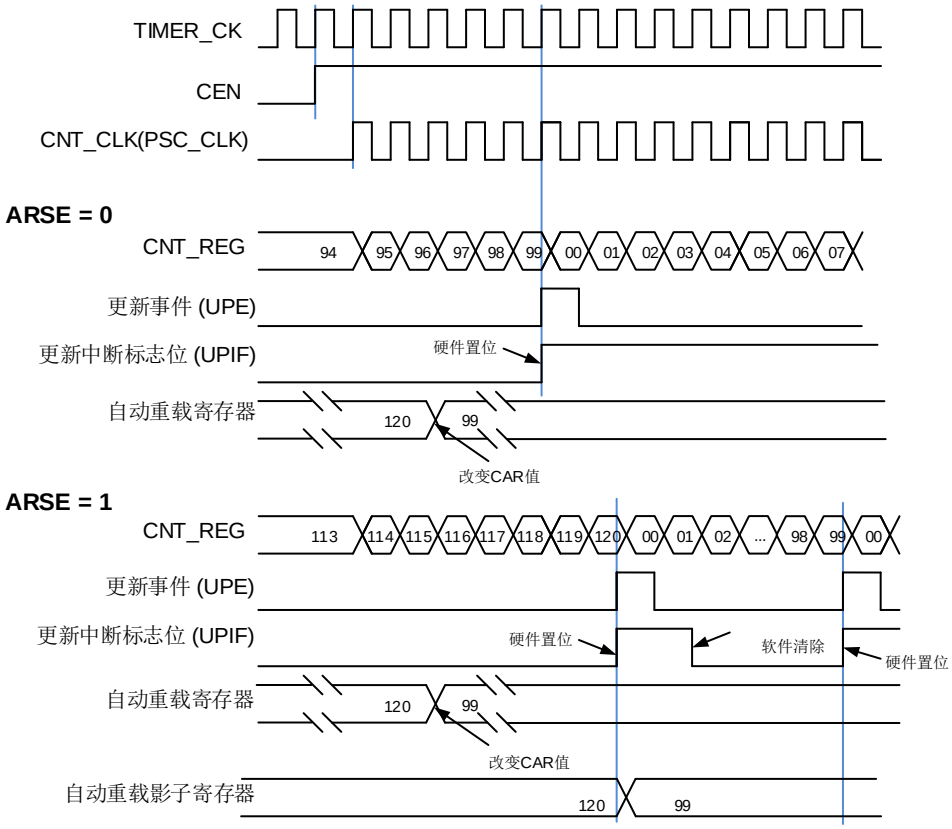


图 23-145. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值

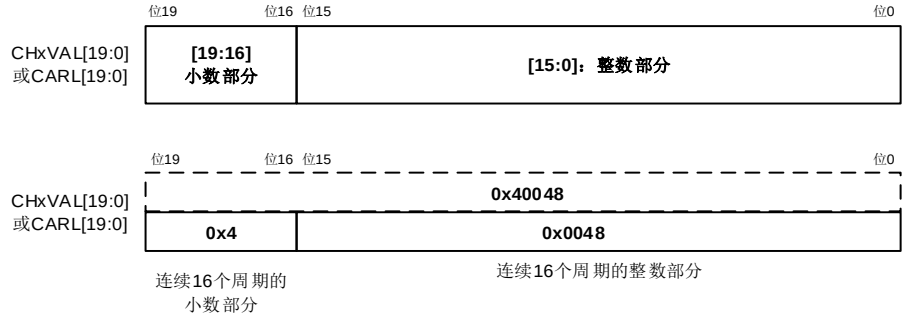


微调模式

通过配置TIMERx_CTL0寄存器中的ADMEN位为1，可以使能微调模式。该模式可以提高输出PWM波的有效分辨率，通过TIMERx_CHxCV寄存器中的CHxVAL[19:0]位域可以提高占空比分辨率，通过TIMERx_CAR寄存器中的CARL[19:0]位域可以提高PWM频率的分辨率。

当微调模式使能时，CHxVAL位域和CARL位域的低16位[15:0]用于整数部分，高4位[19:16]用于微调的小数部分。通过预定义的方式，在连续16个周期内对CHxVAL值或CARL值进行微调（每次调整不超过一个TIMER时钟周期），可增加16倍的分辨率。

图 23-146. 微调模式：数据格式和寄存器位域



根据ADMEN位的配置（置位或清零），CHxVAL位域和CARL位域将自动更新。当需要对ADMEN位进行清零时，需要遵循以下步骤：

1. CEN位和ARSE位必须清零；
2. CARL[19:16]位域必须清零；
3. ADMEN位必须清零；
4. CHxIF位必须清零；
5. 可以将CEN位置1。

以下公式可以计算PWM分辨率：

$$Resolution = f_{PSC_CLK} \ / \ f_{pwm} \quad (23\text{-}12)$$

由式*(23-12)*可得，微调模式禁能时（ADMEN=0），PWM的最小频率f_{pwm}：

$$(f_{pwm})_{min} = f_{PSC_CLK} \ / \ 65536 \quad (23\text{-}13)$$

微调模式使能时（ADMEN=1），

$$(f_{pwm})_{min} = f_{PSC_CLK} \ / \ (65535 + 15 \ / \ 16) \quad (23\text{-}14)$$

当微调模式使能时，CHxVAL[19:0]位域和CARL[19:0]位域的最大值为0xFFFFE（整数部分为0xFFFE，小数部分为0xF）。

在连续16个周期内，占空比和周期的变化情况，具体如*图23-147. PWM微调模式原理*和*表23-24. 边沿对齐模式中CHxVAL和CARL位域的变化*所示。

图 23-147. PWM 微调模式原理

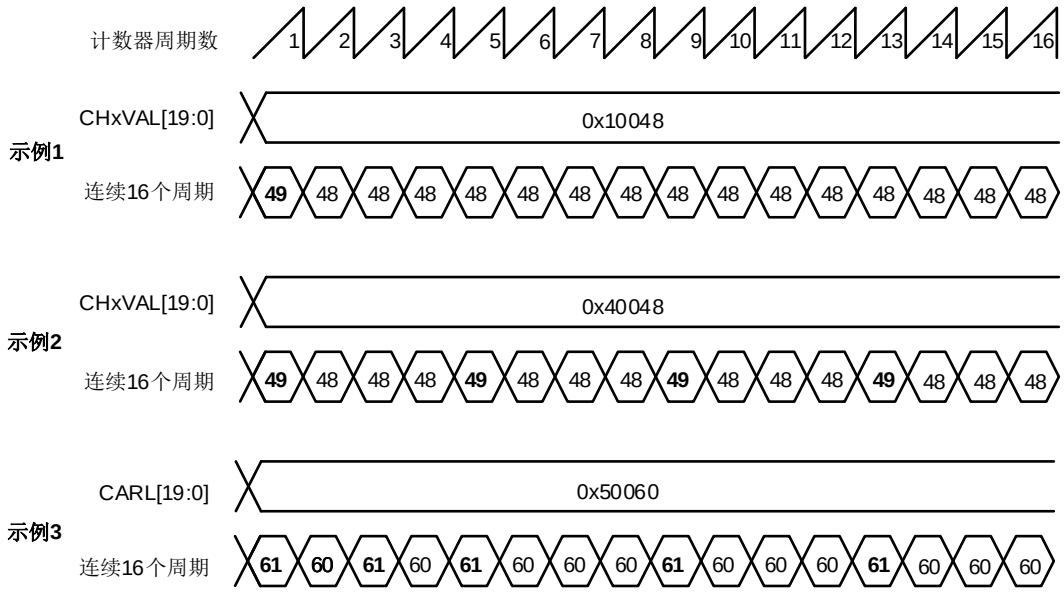


表 23-24. 边沿对齐模式中 CHxVAL 和 CARL 位域的变化

CHxVAL[19:16] / CARL[19:16]	周期															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0010	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-
0011	+1	-	-	-	+1	-	-	-	+1	-	-	-	-	-	-	-
0100	+1	-	-	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0101	+1	-	+1	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0110	+1	-	+1	-	+1	-	-	-	+1	-	+1	-	+1	-	-	-
0111	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	-	-
1000	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1001	+1	+1	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1010	+1	+1	+1	-	+1	-	+1	-	+1	+1	+1	-	+1	-	+1	-
1011	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	-	+1	-
1100	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1101	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1110	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	+1	+1	+1	+1	-
1111	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	-

UPIF 位备份功能

可以通过配置TIMERx_CTL0寄存器中的UPIFBUEN位来使能UPIF位的备份功能，UPIF和UPIFBU位之间没有延迟，两者完全同步。

使能该功能后，TIMERx_INTF寄存器中的UPIF位将会被实时备份到TIMERx_CNT寄存器中的UPIFBU位。这可以避免在读计数器和中断处理时产生冲突的情况。

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。

23.5.5. TIMERx 寄存器（x = 5 / 6）

TIMER5基地址：0x4000 1000

TIMER6基地址：0x4000 1400

控制寄存器 0（TIMERx_CTL0）

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	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			ADMEN	UPIFBUEN	保留			ARSE	保留			SPM	UPS	UPDIS	CEN
			rw	rw				rw				rw	rw	rw	rw

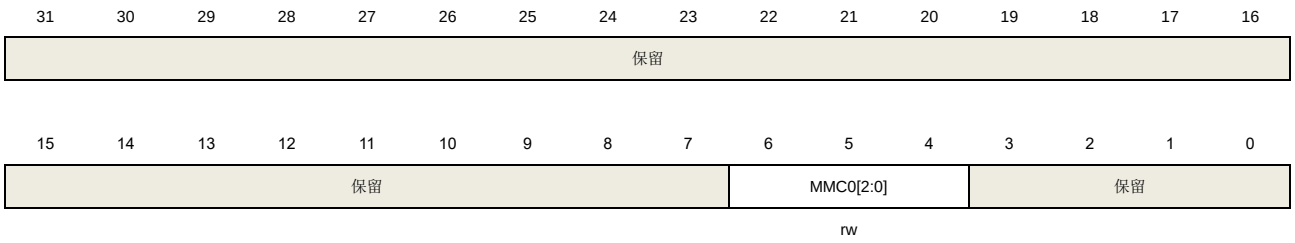
位/位域	名称	描述
31:13	保留	必须保持复位值。
12	ADMEN	微调模式使能 0：微调模式禁能 1：微调模式使能 注意：该位只能在CEN位为0时修改。
11	UPIFBUEN	UPIF位备份使能 0：备份禁能。UPIF位没有备份到TIMERx_CNT寄存器中的UPIFBU位 1：备份使能。UPIF位备份到TIMERx_CNT寄存器中的UPIFBU位
10:8	保留	必须保持复位值。
7	ARSE	自动重载影子使能 0：禁能 TIMERx_CAR 寄存器的影子寄存器 1：使能 TIMERx_CAR 寄存器的影子寄存器
6:4	保留	必须保持复位值。
3	SPM	单脉冲模式 0：更新事件发生后，计数器继续计数 1：在下一次更新事件发生时，CEN 硬件清零并且计数器停止计数
2	UPS	更新请求源

		软件配置该位，选择更新事件源. 0：使能后，下述任一事件产生更新中断或 DMA 请求： – UPG 位被置 1 – 计数器上溢 – 从模式控制器产生的更新 1：使能后只有计数器上溢才产生更新中断或 DMA 请求。
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。当以下事件之一发生时，更新事件产生，具有缓存的寄存器被装入它们的预装载值： – UPG 位被置 1 – 计数器上溢 – 从模式控制器产生一个更新事件 1：更新事件禁能。带有缓存的寄存器保持原有值，如果 UPG 位被置 1 或者从模式控制器产生一个硬件复位事件，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。触发模式可以自动地通过硬件设置 CEN 位。

控制寄存器 1（TIMERx_CTL1）

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:7	保留	必须保持复位值。
6:4	MMC0[2:0]	这些位控制 TRGO0 信号的选择，TRGO0 信号由主定时器发给从定时器用于同步功能

		000：复位。TIMERx_SWEVG 寄存器的 UPG 位被置 1 或从模式控制器产生复位触发一次 TRGO0 脉冲，后一种情况下，TRGO0 上的信号相对实际的复位会有一个延迟。 001：使能。此模式可用于同时启动多个定时器或控制在一段时间内使能从定时器。主模式控制器选择计数器使能信号作为触发输出 TRGO0。当 CEN 控制位被置 1 或者暂停模式下触发输入为高电平时，计数器使能信号被置 1。在暂停模式下，计数器使能信号受控于触发输入，在触发输入和 TRGO0 上会有一个延迟，除非选择了主 / 从模式。 010：更新。主模式控制器选择更新事件作为 TRGO0。 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

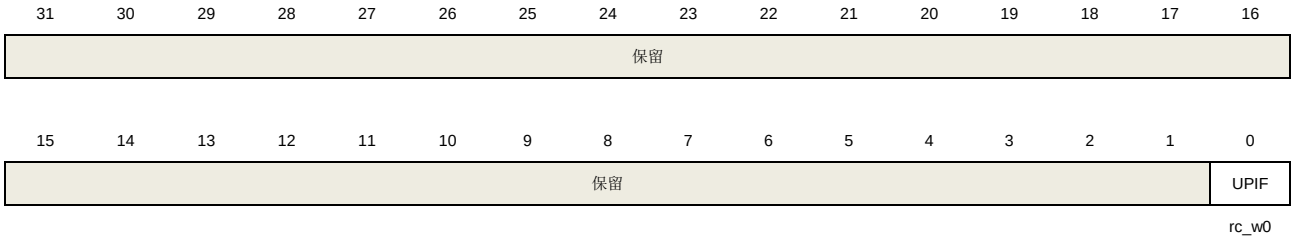
位/位域	名称	描述
31:9	保留	必须保持复位值。
8	UPDEN	更新 DMA 请求使能 0：禁止更新 DMA 请求 1：使能更新 DMA 请求
7:1	保留	必须保持复位值。
0	UPIE	更新中断使能 0：禁止更新中断 1：使能更新中断

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。



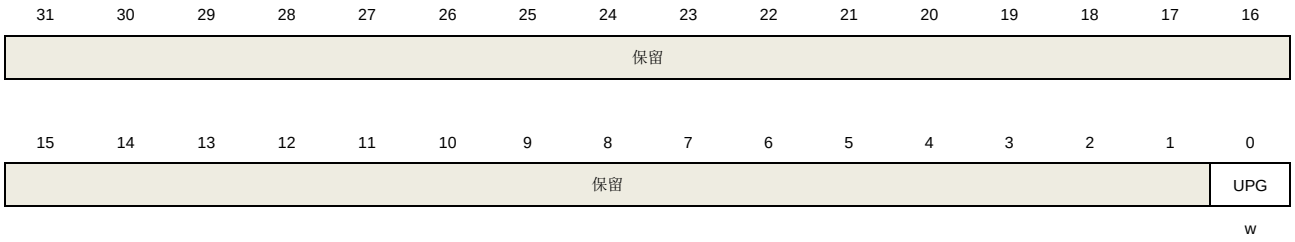
位/位域	名称	描述
31:1	保留	必须保持复位值。
0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置 1，软件清 0。 0：无更新中断发生 1：发生更新中断

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。

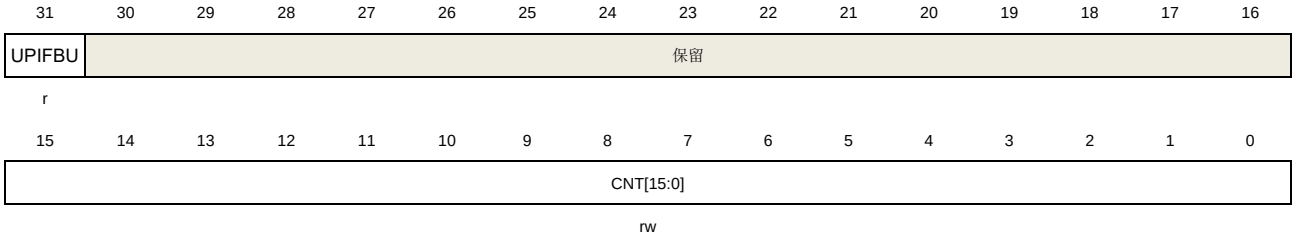


位/位域	名称	描述
31:1	保留	必须保持复位值。
0	UPG	更新事件产生 此位由软件置 1，被硬件自动清 0。当此位被置 1 并且向上计数模式，计数器被清 0，预分频计数器将同时被清除。 0：无更新事件产生 1：产生更新事件

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。

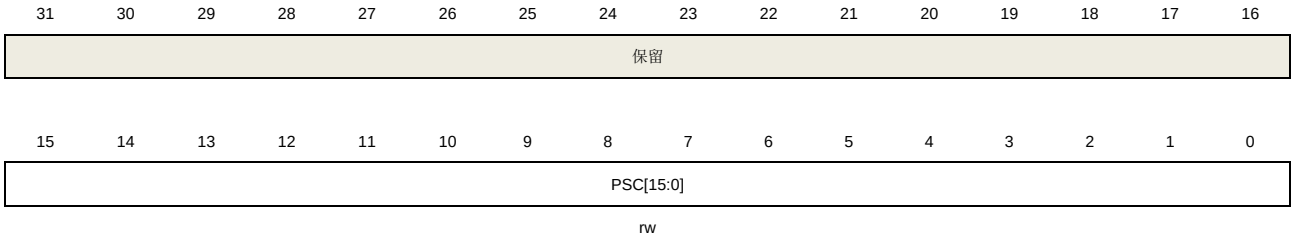


位/位域	名称	描述
31	UPIFBU	UPIF位备份 该位只读，是 TIMERx_INTF 寄存器的 UPIF 位的备份值。当 UPIFBUEN = 1 时，该位有效，若 UPIFBUEN =0，该位保留，读取该位值为零。
30:16	保留	必须保持复位值。
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。 当 PWMADMEN = 0 时，该位域表示当前计数器的值。写操作该位域可以改变计数器的值。 当 PWMADMEN = 1 时，该位域仅用于表示计数器值的整数部分，不包含小数部分。

预分频寄存器（TIMERx_PSC）

地址偏移：0x28
复位值：0x0000 0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	PSC[15:0]	计数器时钟预分频值

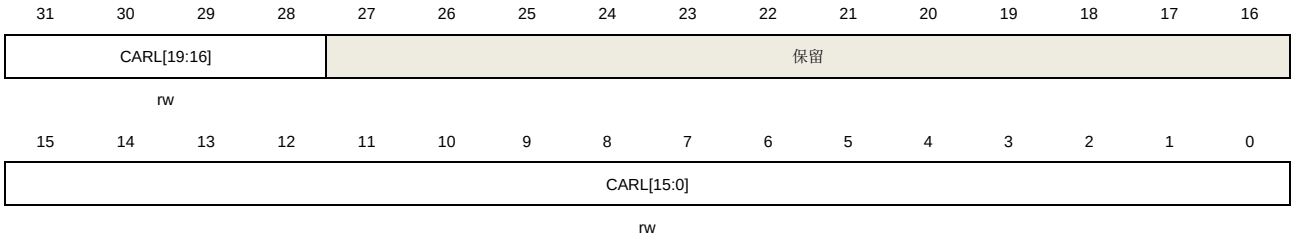
计数器时钟等于 PSC 时钟除以（PSC+1），每次当更新事件产生时，PSC 的值被装入当前预分频寄存器。

计数器自动重载寄存器（TIMERx_CAR）

地址偏移：0x2C

复位值：0x0000 FFFF

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:28	CARL[19:16]	计数器自动重载值（16 位至 19 位） 当 PWMADMEN = 0 时，CARL[19:16]位域值 0000。 当 PWMADMEN = 1 时，CARL[19:16]位域用于表示自动重载值的小数部分。
27:16	保留	必须保持复位值。
15:0	CARL[15:0]	计数器自动重载值（0 位至 15 位） 这些位定义了计数器的自动重载值。 当 PWMADMEN = 0 时，CARL[15:0]位域用于表示计数器的自动重载值。 当 PWMADMEN = 1 时，CARL[15:0]位域用于表示自动重载值的整数部分。

24. 低功耗定时器（LPTIMER）

24.1. 简介

LPTIMER 是一个 16 位的定时器，基于多样的时钟源，它能够在除待机模式（Standby mode）以外的所有功耗模式下运行。LPTIMER 提供了灵活的时钟机制，在将功耗降至最低的同时，还可以实现所需的功能和性能。

LPTIMER 可以用作没有内部时钟源的脉冲计数器。LPTIMER 可以将系统从低功耗模式唤醒，非常适合于以极低的功耗实现超时模式的场合。

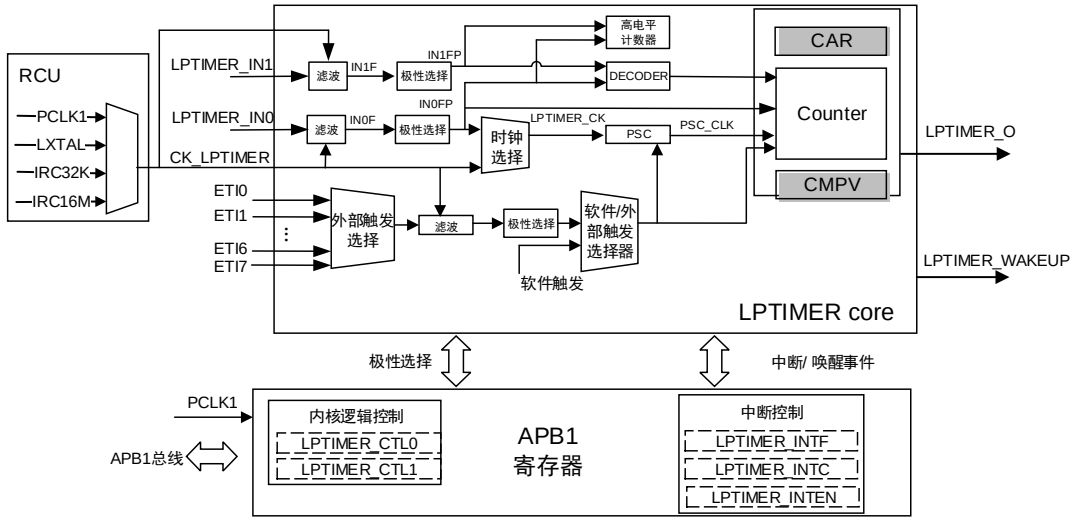
24.2. 主要特征

- 计数器宽度：16位
- 时钟源可选：
  - 内部时钟源：内部8MHz RC晶振（IRC8M），内部32KHz RC晶振（IRC32K），32.768 KHz低速晶振（LXTAL）和APB1时钟（PCLK1）
  - 外部时钟源：来自于LPTIMER_IN0引脚上的时钟源（作为脉冲计数器）
- 计数模式：向上计数
- 运行模式：连续计数模式或单次计数模式
- 可编程的预分频器：3位
- 通道输出可配置：可编程的PWM模式，单脉冲模式，置位模式
- 自动重装载功能
- 中断输出
- 可选择的触发：软件触发或硬件输入触发
- 译码器模式：译码器模式0和译码器模式1

24.3. 结构框图

*图 24-1. LPTIMER 结构框图*提供了低功耗定时器的内部配置细节。

图 24-1. LPTIMER 结构框图



24.4. 功能描述

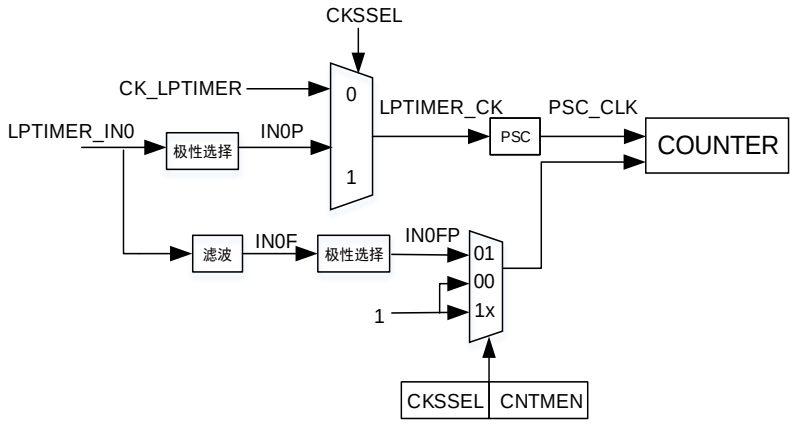
24.4.1. 时钟源配置

LPTIMER 可以由多个时钟源提供时钟，如内部时钟源有：内部 8MHz RC 晶振（IRC8M），内部 32KHz RC 晶振（IRC32K），32.768 KHz 低速晶振（LXTAL）和 APB1 时钟（PCLK1），这些时钟源来自于复位和时钟单元 RCU。

LPTIMER 还可以使用外部引脚 LPTIMER_IN0 上的外部时钟信号作为时钟，当使用外部时钟作为时钟源时，LPTIMER 有以下两种配置方式：

- **Case 0**：当LPTIMER由外部信号提供时钟时，还需要APB1或其他晶振（如IRC8M、IRC32K和LXTAL）同时提供内部时钟信号；
- **Case 1**：LPTIMER的时钟仅由LPTIMER_IN0引脚上外部时钟信号提供。在进入低功耗模式后，所有晶振关闭，此配置可用于实现超时模式或脉冲计数器功能。

图 24-2. LPTIMER 时钟源选择



LPTIMER 可以由内部时钟信号或外部时钟信号（由 LPTIMER_CTL0 寄存器中的 CNTMEN 位和 CKSSEL 位控制）驱动。CKSSEL 位用于选择哪个时钟驱动计数器预分频器，默认时钟源为 PCLK1。CNTMEN 位用于选择哪个时钟信号驱动 LPTIMER 计数器。

当 LPTIMER 使用外部时钟信号时，CKPSEL 用于配置计数器的有效边沿。计数器可以由外部时钟的上升沿、下降沿或双边沿更新，具体由 CKPSEL[1:0]位域配置。

需要注意的是，当由外部引脚 LPTIMER_IN0 提供外部时钟信号时，如果有效边沿选择双边沿（CKPSEL=2'b10）或者 LPTIMER_IN0 引脚由数字滤波器采样（ECKFLT≠2'b00），则还需要提供内部时钟信号（Case 0）。在这种情况下，内部时钟信号频率至少是外部时钟频率的 4 倍。

可以根据 CKSSEL 位和 CNTMEN 位的配置，选择以下的时钟模式：

- CKSSEL = 0：LPTIMER时钟由内部时钟信号提供

  - 内部时钟模式0（CNTMEN = 0）

  LPTIMER由内部时钟信号提供时钟，计数器在内部时钟的每个脉冲进行计数。

  - 内部时钟模式1（CNTMEN = 1）

  外部时钟信号（LPTIMER_IN0）由内部时钟进行采样，因此，为了保证不丢失任何事件，外部时钟的变化频率不能超过内部时钟的频率。并且，LPTIMER的内部时钟信号不能预分频（PSC [2:0] = 000）。

- CKSSEL = 1：LPTIMER时钟由外部时钟信号提供

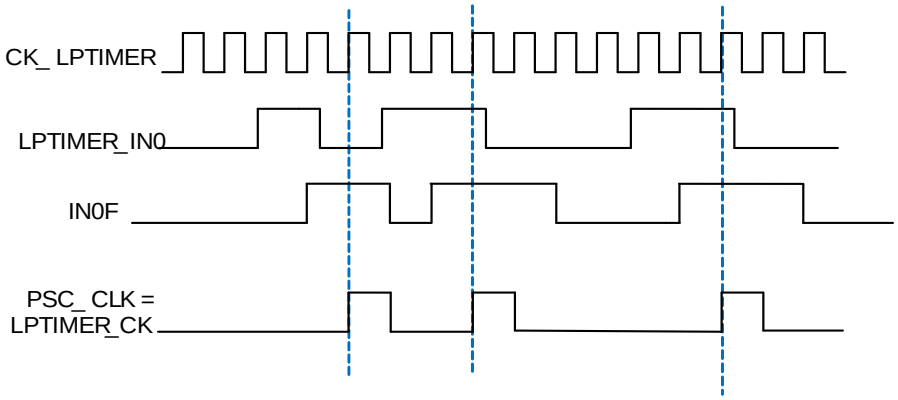
  这种情况下，可以将CNTMEN位置1或清零，LPTIMER不需要使用内部时钟源（除非使能了输入滤波或是选择了双边沿作为外部时钟的有效边沿）。LPTIMER_IN0引脚上的外部信号作为LPTIMER的系统时钟，该情况适用于没有嵌入式晶振的工作情况；

  这种情况下，LPTIMER的计数器可以在外部时钟信号的上升沿或下降沿计数，不能在双边沿计数。

  由于LPTIMER_IN0引脚上的外部时钟信号也用于驱动LPTIMER的内核逻辑部分，因此，在计

数器计数之前会有一些初始延迟（LPTIMER使能之后）。因此，在使能LPTIMER之后，LPTIMER_IN0引脚上的外部时钟信号前5个有效边沿将会丢失。

图 24-3. 内部时钟模式 1（CKSSEL = 0，CNTMEN = 1，PSC[2:0] = 000）



24.4.2. LPTIMER 使能

LPTIMER_CTL1 寄存器的 LPTEN 位用于使能 LPTIMER 的内核逻辑模块。在 LPTEN 位置 1 后，实际使能 LPTIMER 之前需要延迟两个 LPTIMER_CK 时钟周期。

只有在 LPTIEMR 禁能时，才能修改 LPTIMER_CTL0 和 LPTIMER_INTEN 寄存器（INHLCOIE 位和 HLCMVUPIF 位除外）。

24.4.3. 预分频器

预分频器可以将 LPTIMER 的时钟 LPTIMER_CK 除以 2 的乘幂分频为计数器时钟 PSC_CLK。只有在 LPTIEMR 禁能（LPTEN=0）时，才能修改 PSC[2:0]位域。下表列出了所有的分频系数：

表 24-1. 预分频器的分频系数

预分频系数	PSC[2:0]位域
1/1	000
1/2	001
1/4	010
1/8	011
1/16	100
1/32	101
1/64	110
1/128	111

24.4.4. 输入滤波

LPTIMER_Inx 引脚上的外部（映射到 GPIO）或内部（映射到片上外设，如比较器）信号需要通过数字滤波器进行滤波，以防止毛刺和噪声干扰在 LPTIMER 中扩散，这可以有效防止误计数和误触发。

在使用数字滤波器之前，需要先给 LPTIMER 提供内部时钟源，这样可以确保滤波器的正确运行。

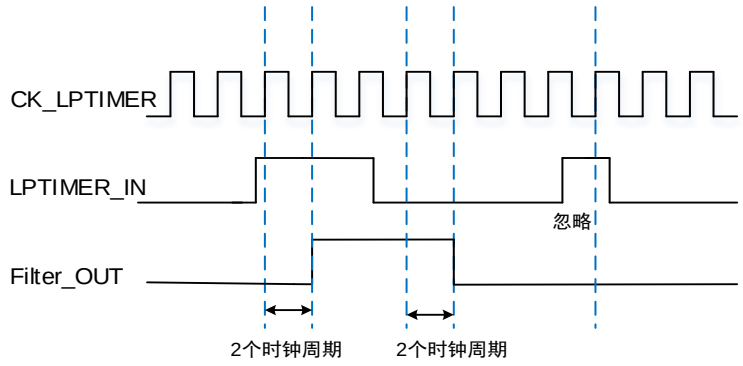
数字滤波器有两种类型：

- 第1种：用于保护LPTIMER的外部输入（LPTIMER_IN0/ LPTIMER_IN1），数字滤波器由ECKFLT[1:0]位进行配置；
- 第2种：用于保护LPTIMER的触发输入（ETIx），数字滤波器由TFLT[1:0]位进行配置；

注意：相同类型的数字滤波器应该保持相同的配置。

数字滤波器的灵敏度取决于 LPTIMER 输入引脚上连续相同采样的数量，并将信号电平变化视为有效。*图 24-4. 输入滤波时序图（ECKFLT=2'b01）*显示了输入滤波器 2 次连续采样的示例。

图 24-4. 输入滤波时序图（ECKFLT=2'b01）



注意：如果没有内部时钟信号，则必须禁能数字滤波器（设置 ECKFLT=0，TFLT=0）。这种情况下，可以使用外部模拟滤波器来保证 LPTIMER 的外部输入不受干扰。

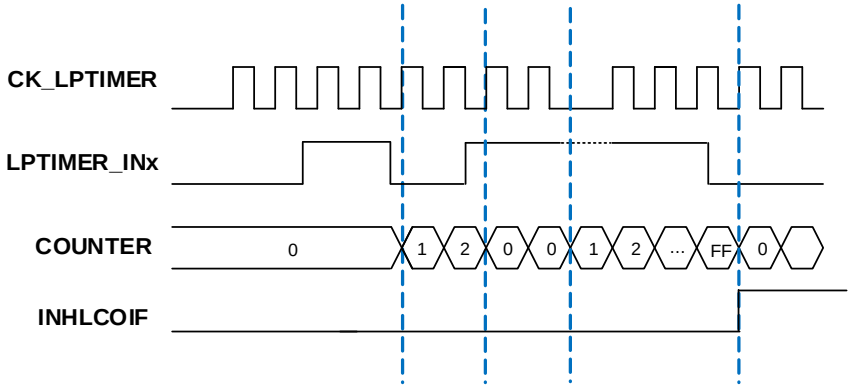
24.4.5. 外部输入高电平计数器

将 INHLCEN 位置 1 可以使能外部输入 LPTIMER_Inx 的高电平计数器功能，高电平计数器的时钟由内部时钟 CK_LPTIMER 提供。当 LPTIMER_Inx 引脚上出现高电平时，计数器开始计数，一旦出现低电平，计数器清零。

当高电平计数器的计数值等于 INHLCMVAL 位域（在 LPTIMER_INHLCMV 寄存器中）定义的数值时，LPTIMER_INTF 寄存器中的 INHLCOIF 位由硬件置位。若使能了 LPTIMER_INTEN 寄存器中的 INHLCOIE 位，则会产生中断。可以通过向 INTC 寄存器中的 INHLCOIC 位写 1 来清除 INHLCOIF 中断标志位。

*图 24-5. 外部输入高电平计数器*给出了外部输入高电平计数器的示例。

图 24-5. 外部输入高电平计数器



APB 总线和 CK_LPTIMER 使用不同的时钟，因此，APB 写操作和 LPTIMER_INHLCMV 寄存器实际使用这些值的时间存在一些延迟。在此延迟时间内，应当避免对该寄存器的任何其他写操作。

LPTIMER_INTF 寄存器中的 HLCMVUPIF 位用于说明对 LPTIMER_INHLCMV 寄存器的写操作何时完成。

24.4.6. 计数器启动

LPTIMER 的计数器可以通过软件触发或通过检测 8 个触发输入上的有效边沿来启动。ETMEN [1:0]位域用于配置 LPTIMER 的触发模式：

- ETMEN[1:0] = 2'b 00：一旦软件置位CTNMST位或SMST位，LPTIMER的计数器就会启动；
- ETMEN[1:0] ≠ 2'b00：ETSEL[2:0]位用于选择8个触发输入中的一个来启动LPTIMER。ETMEN[1:0]位域的其余3个非零值用于配置触发输入所使用的有效边沿。一旦检测到有效边沿，LPTIMER计数器就会启动。

外部触发可视为 LPTIMER 的异步信号，因此，一旦检测到触发信号，为了实现同步，在 LPTIMER 开始运行之前需要延迟 2 个计数器时钟周期。如果在 LPTIMER 启动后发生新的触发事件，该触发事件将会被忽略（除非使能了超时模式）。

注意：在置位 SMST 位和 CTNMST 位之前，必须将 LPTEN 位置位。当 LPTEN=0 时，对这些位的任何写操作都将被硬件丢弃。

24.4.7. 外部触发映射

LPTIMER 外部触发连接的情况如下*表 24-2. 外部触发映射*所示：

表 24-2. 外部触发映射

ETSEL[2:0]	外部触发映射
ETI0	GPIO
ETI1	RTC闹钟0

ETSEL[2:0]	外部触发映射
ETI2	RTC闹钟1
ETI3	RTC_TAMP0
ETI4	RTC_TAMP1
ETI5	RTC_TAMP2
ETI6	CMP0_OUT
ETI7	CMP1_OUT
ETI8	CMP2_OUT
ETI9	CMP3_OUT
ETI10	CMP4_OUT
ETI11	CMP5_OUT
ETI12	CMP6_OUT
ETI13	CMP7_OUT

24.4.8. 计数器运行模式

LPTIMER计数器运行在两种模式下：

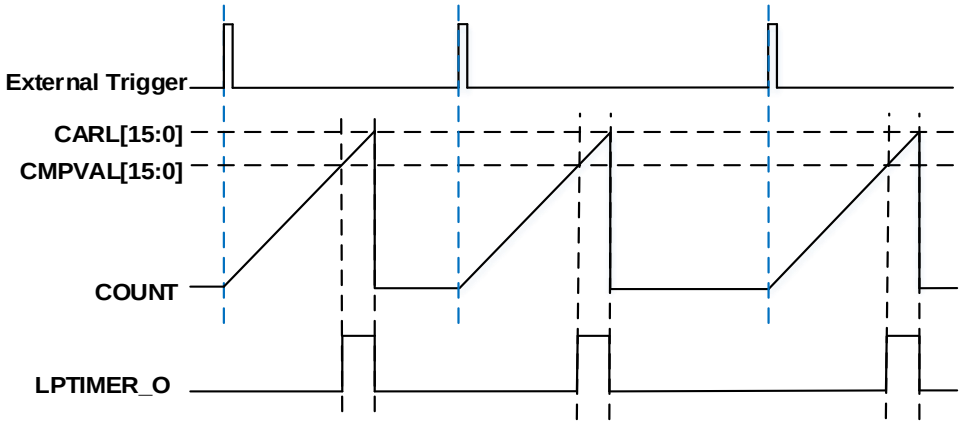
- 连续计数模式：LPTIMER计数器由触发事件启动（软件触发或外部触发）后连续运行，直到LPTIMER禁能后才会停止；
- 单次计数模式：LPTIMER计数器由触发事件启动（软件触发或外部触发），在计数到CARL位域（在LPTIMER_CAR寄存器中）定义的值后停止；

单次计数模式

将 LPTIMER_CTL0 寄存器中的 SMST 位置 1，可以使能 LPTIMER 计数器的单次计数模式。该模式下，一次新的触发事件将重新启动 LPTIMER 计数器。在计数器启动之后且计数器达到 CARL 位域定义的值之前发生的任何触发事件都将被忽略。

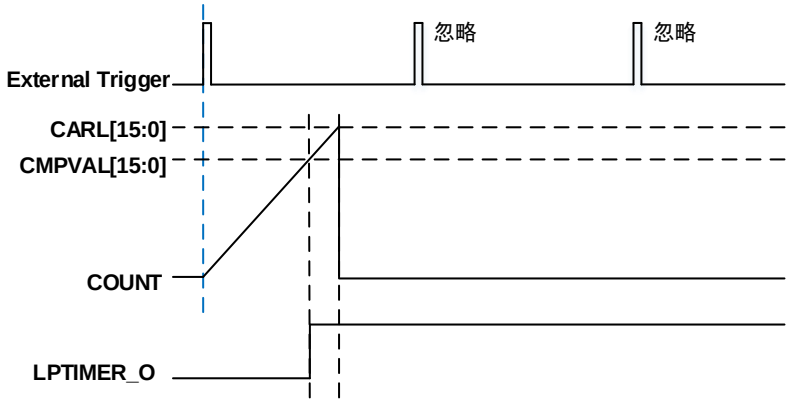
如果选择了外部触发来启动 LPTIMER 计数器，则当 SMST 位置 1 时，在计数器停止计数后（CNT 位域的值为 0）到达的每一个外部触发事件都将启动计数器进行新的计数周期计数，具体如*图 24-6. LPTIMER 输出（SMST = 1，16 位）*所示。

图 24-6. LPTIMER 输出（SMST = 1，16 位）



将 LPTIMER_CTL0 寄存器中的 OMSEL 位置 1，可以使能 LPTIMER 的置位模式。该模式下，LPTIMER 的计数器仅在第一次触发后启动，之后的所有触发事件都将被忽略，具体如*图 24-7. LPTIMER 输出（OMSEL = 1，16 位）*所示。

图 24-7. LPTIMER 输出（OMSEL = 1，16 位）



当 ETMEN[1:0] = 2’b 00 时，软件触发使能，将 SMST 位置 1，LPTIMER 以单次计数模式启动。

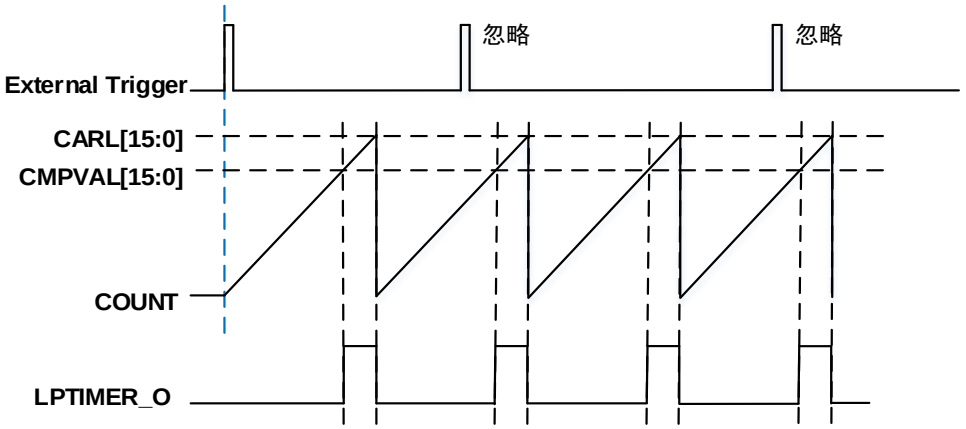
连续计数模式

将 LPTIMER_CTL0 寄存器中的 CTNMST 位置 1，可以使能 LPTIMER 计数器的连续计数模式。

如果选择了外部触发来启动 LPTIMER 计数器，则在 CTNMST 为置 1 之后到达的外部触发事件将启动计数器的连续计数模式，在 LPTIMER 启动后到达的任何触发事件都将被忽略，具体如*图 24-8. LPTIMER 输出（CTNMST = 1，16 位）*所示。

当 ETMEN[1:0] = 2’b 00 时，软件触发使能，将 CTNMST 位置 1，LPTIMER 以连续计数模式启动。

图 24-8. LPTIMER 输出（CTNMST = 1，16 位）



只有当 LPTIMER 使能（LPTEN=1）时，才能修改 SMST 位和 CTNMST 位，单次计数模式和连续计数模式可以实时进行修改。

若 LPTIMER 工作在连续计数模式，将 SMST 位置 1 后将切换到单次计数模式。计数器计数到 CARL 位域定义的值之后将停止计数。

若 LPTIMER 工作在单次计数模式，将 CTNMST 位置 1 后将切换到连续计数模式。计数器计数到 CARL 位域定义的值之后将重新开始计数。

24.4.9. 计数器复位

LPTIMER 的计数器可以通过软件异步或者同步复位为 0。两种复位方式的详情如下：

- 计数器异步复位：当设置LPTIMER_CTL1寄存器的RDRSTEN位为1时，LPTIMER_CNT寄存器会在每次读访问后被复位为0。
- 计数器同步复位：LPTIMER计数器可以通过LPTIMER_CTL1寄存器的CNTRST位写1来同步复位。同步复位意味着复位操作需要3个LPTIMER内核时钟周期来同步，当LPTIMER使用异步时钟（与APB时钟异步）来计数时。在同步过程中，计数器依旧保持计数。在计数器复位操作完成后，CNTRST位被硬件自动清零。

注意: 两种复位方式不能同时使用。CNTRST 位只能在已经被硬件清 0 后才能再次写 1。软件需要在试图对 CNTRST 写 1 时检查 CNTRST 是否已经被清零。

24.4.10. 输出模式

通过配置 LPTIMER_CARL 寄存器和 LPTIMER_CMPV 寄存器，LPTIMER 可以输出几种不同的波形。

LPTIMER 可以输出以下 3 种波形：

- PWM模式：当LPTIMER_CNT的值和LPTIMER_CMPV寄存器的值匹配时，LPTIMER输出置

位。当LPTIMER_CNT的值和LPTIMER_CAR寄存器的值匹配时，LPTIMER输出复位；

- 单脉冲模式：输出波形与PWM模式的第一个脉冲相同，之后始终输出复位；
- 置位模式：输出波形与单脉冲模式类似，输出保持为信号的最后电平（具体由LPTIMER_CTL0寄存器中的OPSEL位确定）。

这三种输出模式都要求 LPTIMER_CAR 寄存器的值大于 LPTIMER_CMPV 寄存器的值。

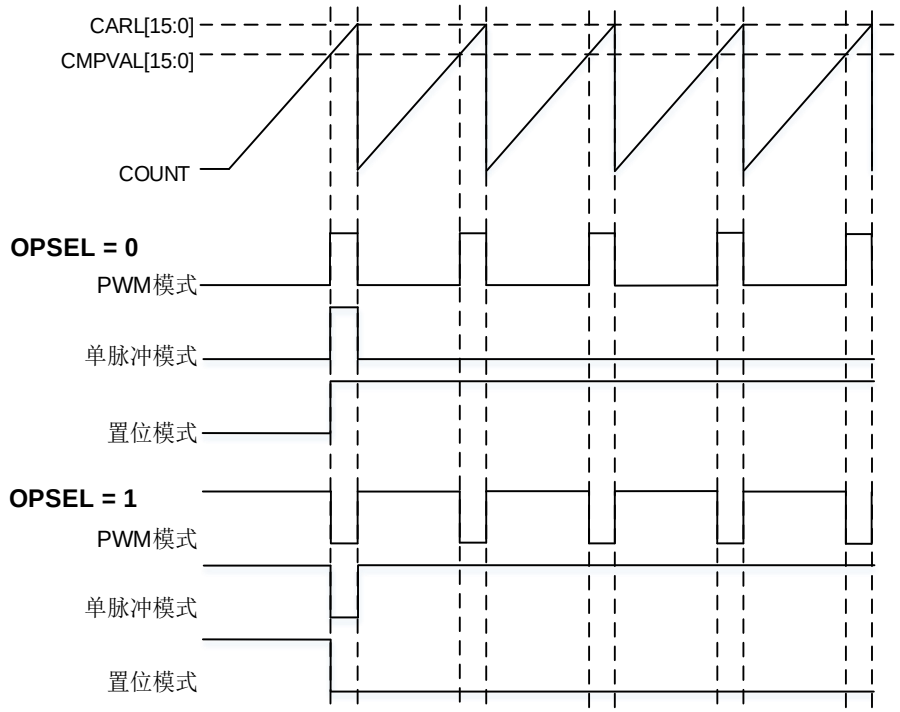
LPTIMER_CTL0 寄存器中的 OMSEL 位用于选择这三种输出模式。

- OMSEL = 0：LPTIMER输出为PWM模式或单脉冲模式（具体由CTNMST位或SPMST位配置）；
- OMSEL = 1：LPTIMER输出置位模式。

OPSEL 位用于配置 LPTIMER 的输出极性，修改该位立即生效。因此，在使能 LPTIMER 之前，只要修改极性配置位，输出默认值就会立即改变。

LPTIMER 输出波形的最大频率为 LPTIMER 时钟频率的二分之一，*图 24-9. LPTIMER_O 输出（OPSEL=0/1，16 位）*所示为 LPTIMER 输出的三种波形，和 OPSEL 位的值对输出波形极性的影响。

图 24-9. LPTIMER_O 输出（OPSEL=0/1，16 位）



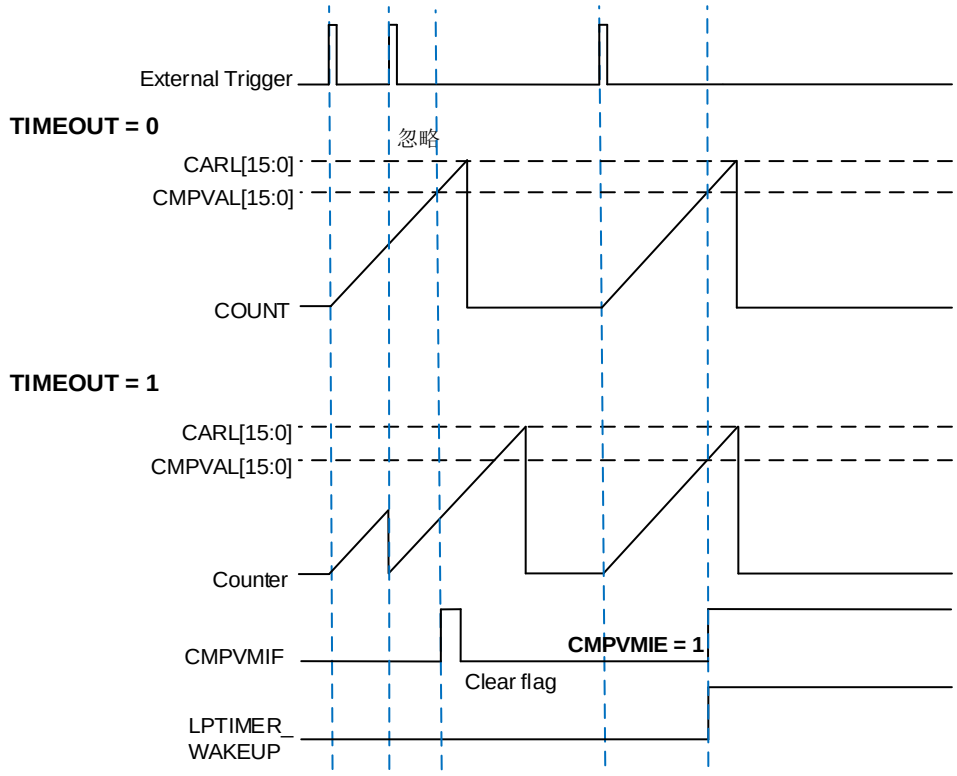
24.4.11. 超时模式

将 TIMEOUT 位置 1，可以通过在选定触发输入上检测到的有效边沿来复位 LPTIMER。第 1 个触发事件将用于启动 LPTIMER，之后的触发事件将复位和重新启动 LPTIMER。

LPTIMER 可以实现低功耗的超时模式，超时值可由 LPTIMER_CMPV 寄存器的值进行配置。

在配置好的比较寄存器值范围内若没有触发发生，当计数器计数到比较寄存器值时，将产生比较匹配中断唤醒 MCU。

图 24-10. LPTIMER 超时模式（16 位）



24.4.12. 译码器模式

LPTIMER由两种译码器模式：

- 译码器模式0：LPTIMER_IN0和LPTIMER_IN1输入正交信号，当DECMEN=1且DECMSEL=0使能该模式；
- 译码器模式1：LPTIMER_IN0和LPTIMER_IN1输入非交信号，当DECMEN=1且DECMSEL=1使能该模式。

译码器模式 0

译码器模式 0 用于 LPTIMER_IN0 和 LPTIMER_IN1 输入为正交信号的情况，两个正交信号相互作用产生计数。

首先，将 CTNMST 位置 1 使能连续计数模式，同时 DECMEN 位置 1 使能译码器模式。设置 DECMSEL = 0 选择译码器模式 0，然后设置 CKPSEL[1:0] = 2’b00、2’b01 或 2’b10 来选择计数器上升沿、下降沿或双边沿计数。

在 IN0F（LPTIMER_IN0 滤波信号）信号和 IN1F（LPTIMER_IN1 滤波信号）信号电平变化期间，硬件会自动改变计数方向。计数方向变化机制如*表 24-3. 计数方向与译码器信号之间的关系*所示。译码器可以看作是一个带有方向选择的外部时钟，这意味着计数器会在 0 和自动重载值之间连续的计数。

因此，必须在计数器计数之前配置 LPTIMER_CAR 寄存器。

当计数器计数方向改变时，相应的标志位会置 1。当计数器从向上计数改为向下计数时，DOWNIF 位置 1；当计数器从向下计数改为向上计数时，UPIF 位置 1。当寄存器中的 DOWNIE = 1 或 UPIE = 1 时，相应的中断产生。

表 24-3. 计数方向与译码器信号之间的关系

计数模式（CKPSEL[1:0]）	电平	IN0F		IN1F	
		上升沿	下降沿	上升沿	下降沿
上升沿计数	IN0F = 1	x	x	向上	-
	IN0F = 0	x	x	向下	-
	IN1F = 1	向下	-	x	x
	IN1F = 0	向上	-	x	x
下降沿计数	IN0F = 1	x	x	-	向下
	IN0F= 0	x	x	-	向上
	IN1F = 1	-	向上	x	x
	IN1F = 0	-	向下	x	x
双边沿计数	IN0F = 1	x	x	向上	向下
	IN0F = 0	x	x	向下	向上
	IN1F = 1	向下	向上	x	x
	IN1F = 0	向上	向下	x	x

注意： “-”是不计数；“x”是不可能。

*图24-11. 计数器运行在译码器模式0（上升沿计数）*和*图24-12. 计数器运行在译码器模式0（下降沿计数）*分别给出了上升沿计数和下降沿计数的示例。

图 24-11. 计数器运行在译码器模式 0（上升沿计数）

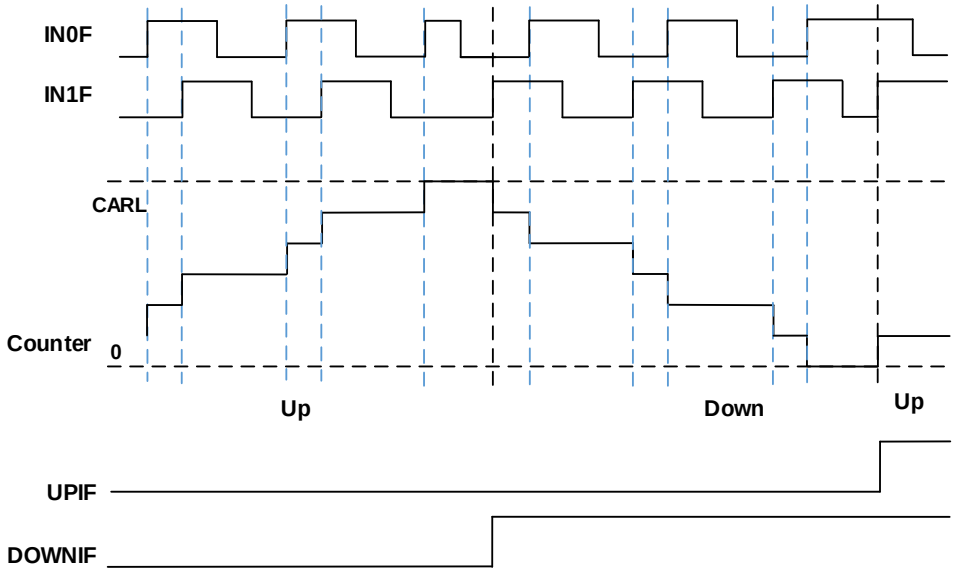
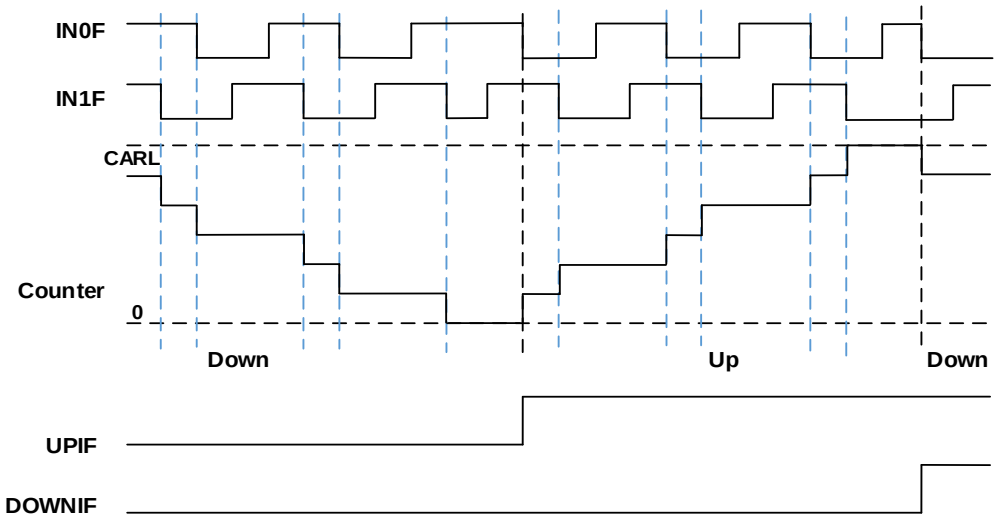


图 24-12. 计数器运行在译码器模式 0（下降沿计数）



译码器模式 1

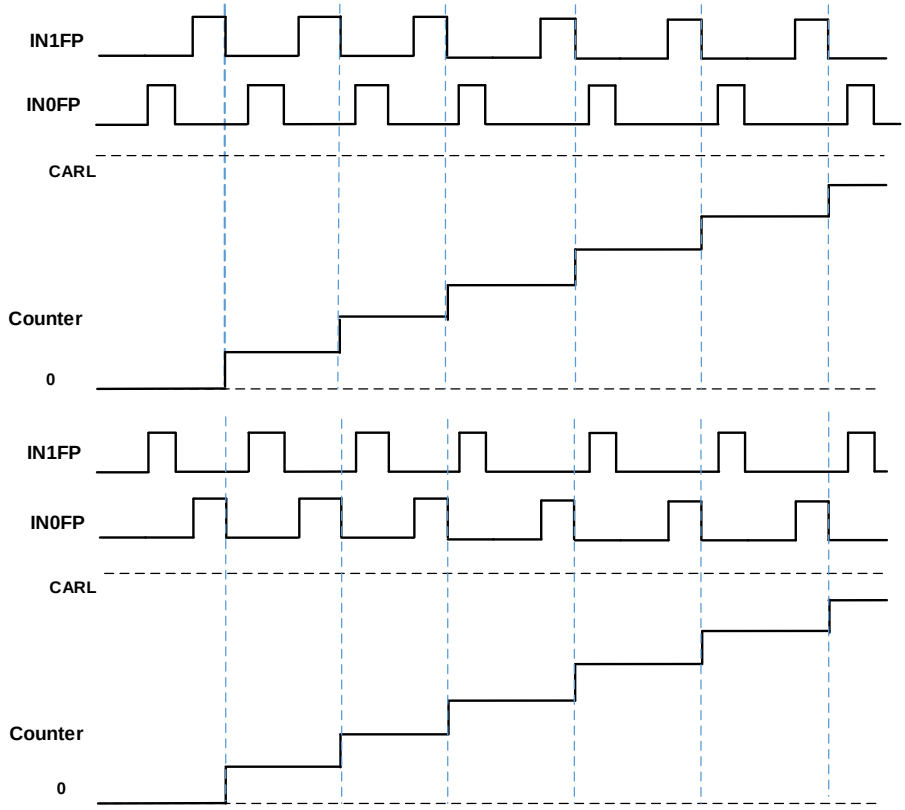
译码器模式 1 用于 LPTIMER_IN0 和 LPTIMER_IN1 输入为非交信号的情况，两个非交信号相互作用产生计数。

首先，将 CTNMST 位置 1 使能连续计数模式，同时 DECMEN 位置 1 使能译码器模式。设置 DECMSEL = 1 选择译码器模式 1，然后设置 CKPSEL[1:0] = 2'b00、2'b01 来选择 LPTIMER_IN0 和 LPTIMER_IN1 输入同相或反相。

当 IN0F 信号和 IN1F 信号依次出现两个不重叠脉冲时，计数器增计数一次。*图 24-13. 计数器运行在译码器模式 1（同相）*所示在译码器模式 1 下正确计数的信号波形时序图。IN0F 信号和 IN1F 信号的高电平不重叠。

译码器可以看作是一个带有方向选择的外部时钟，这意味着计数器会在 0 和自动重载值之间连续的计数。因此，必须在计数器计数之前配置 LPTIMER_CAR 寄存器。

图 24-13. 计数器运行在译码器模式 1（同相）



当 IN0F 信号和 IN1F 信号波形不满足*图 24-13. 计数器运行在译码器模式 1（同相）*所示的时序关系时，计数器不能计数。根据两个信号输入波形的情况，相应的标志位（IN1EIF，IN0EIF，INRFOEIF，INHLOEIF）将置位，若 LPTIMER_INTEN 寄存器中的 IN1EIE 位、IN0EIE 位、INRFOEIE 位和 INHLOEIE 位置 1，则将产生相应的中断。

图 24-14. 计数器运行在译码器模式 1（同相，IN1EIF）

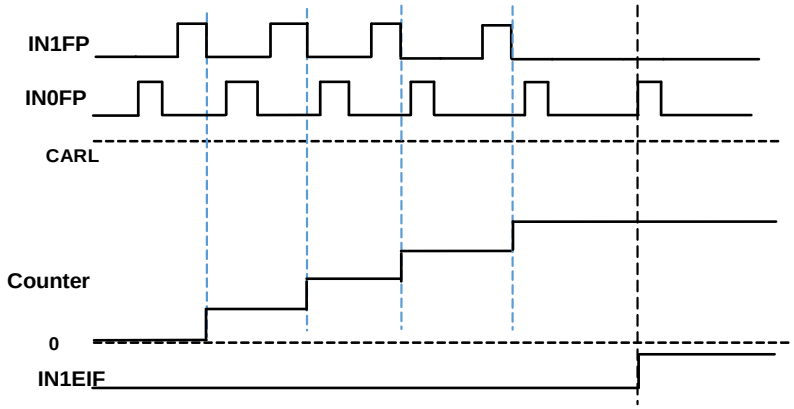


图 24-15. 计数器运行在译码器模式 1（同相，IN0EIF）

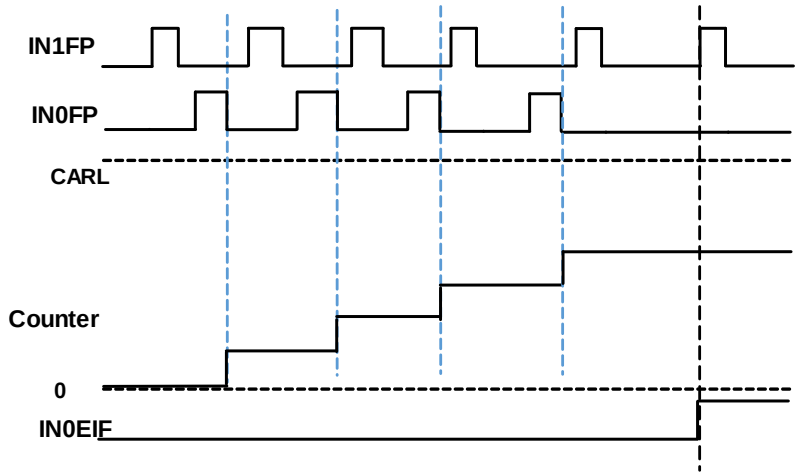


图 24-16. 计数器运行在译码器模式 1（同相，INRFOEIF）

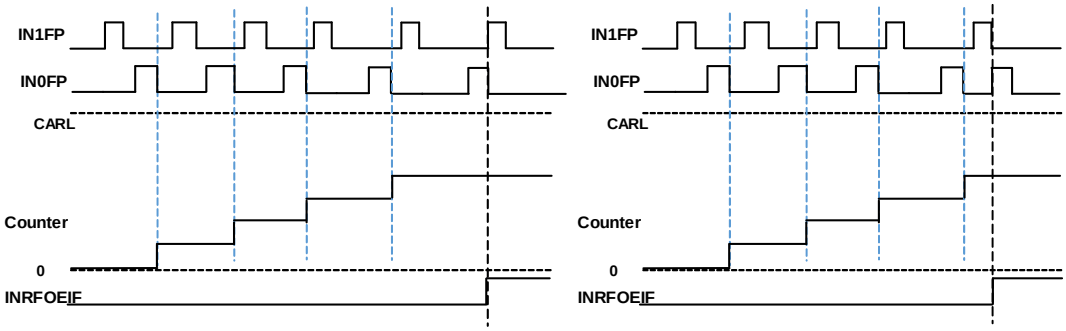
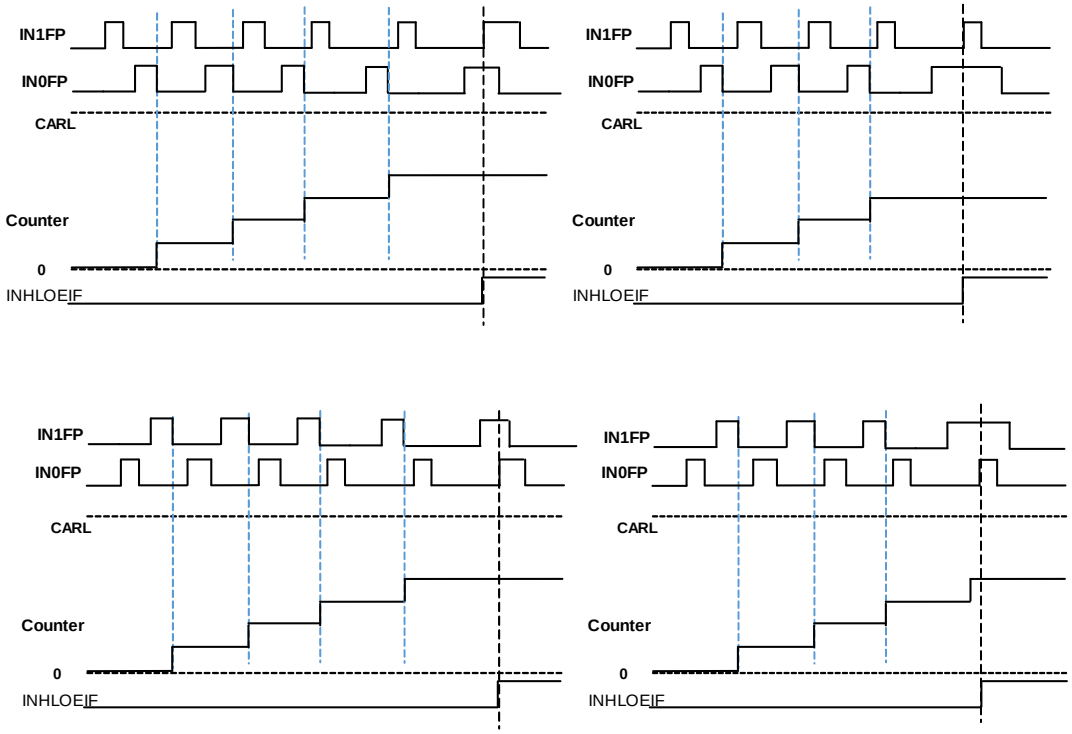


图 24-17. 计数器运行在译码器模式 1（同相，INHLOEIF）



注意，LPTIMER 使用译码器模式时，还应提供一个内部时钟信号（CKSSEL = 0），且该时钟信号不能进行预分频（PSC[2:0] = 000）。在这种情况下，内部时钟信号的频率至少是外部时钟信号频率的四倍。

24.4.13. 寄存器更新操作

LPTIMER_CAR 寄存器和 LPTIMER_CMPV 寄存器在 APB 总线完成写操作之后立即更新，或当 LPTIMER 启动后在当前周期完成后更新。SHWEN 位用于配置 LPTIMER_CAR 寄存器和 LPTIMER_CMPV 寄存器的更新情况：

- SHWEN = 0：在APB总线写操作后，LPTIMER_CAR寄存器和LPTIMER_CMPV寄存器立即更新；
- SHWEN = 1：当LPTIMER启动之后，LPTIMER_CAR寄存器和LPTIMER_CMPV寄存器在当前周期完成后更新。

APB 总线和 CK_LPTIMER 使用不同的时钟，因此，APB 写操作和 LPTIMER_CAR 寄存器和 LPTIMER_CMPV 寄存器实际使用这些值的时间存在一些延迟。在此延迟时间内，应当避免对这些寄存器的任何其他写操作。

LPTIMER_INTF 寄存器中的 CARUPIF 位和 CMPVUPIF 位分别用于说明 LPTIMER_CAR 寄存器和 LPTIMER_CMPV 寄存器的写操作何时完成。

在写 LPTIMER_CAR 寄存器和 LPTIMER_CMPV 寄存器之后，只有完成了当前的写操作，才能对

同一寄存器进行新的写操作。在 CMPVUPIF 标志位和 CARUPIF 标志位分别置 1 前，任何连续的写操作将造成不可预测的结果。

24.4.14. 低功耗模式

LPTIMER 具有多种时钟源，可以在除待机模式（Standby mode）以外的所有功耗模式下保持运行。LPTIMER 可以将系统从低功耗模式唤醒，非常适合于以极低的功耗实现超时模式的场合。

表 24-4. LPTIMER 工作在低功耗模式

模式	描述
睡眠模式	正常运行，LPTIMER的中断会使设备退出睡眠模式
深度睡眠模式	当LPTIMER由LXTAL或内部低速RC振荡器提供时钟时，LPTIMER的中断会使设备退出深度睡眠模式。

24.4.15. 中断

若在 LPTIMER_INTEN 寄存器中使能了相应的位，下面的事件可以产生中断或唤醒事件：

- LPTIMER_IN1错误
- LPTIMER_IN0错误
- LPTIMER_IN0和LPTIMER_IN1下降沿和上升沿重叠错误
- LPTIMER_IN0和LPTIMER_IN1高电平重叠错误
- LPTIMER_Inx（x=0,1）高电平计数器溢出
- 输入高电平计数最大值寄存器更新中断标志位
- LPTIMER计数器由向上计数改为向下计数
- LPTIMER计数器由向下计数改为向上计数
- 计数器自动重载寄存器更新
- 比较寄存器更新
- 外部触发边沿事件
- 计数器自动重载寄存器匹配
- 比较寄存器匹配

如果 LPTIMER_INTF 寄存器中的中断标志位在相应的中断使能位置 1（LPTIMER_INTEN 寄存器中）前置位了，则该中断无效。

表 24-5. LPTIMER 中断事件

中断事件	描述
LPTIMER_IN1错误	当LPTIMER_IN1信号不在LPTIMER_IN0信号的两个连续上升沿之间发生跳变时，该标志位置1（仅用于译码器模式1）。
LPTIMER_IN0错误	当LPTIMER_IN0信号不在LPTIMER_IN1信号的两个连续上升沿之间发生跳变时，该标志位置1（仅用于译码器模式1）。
LPTIMER_IN0和LPTIMER_IN1下	当LPTIMER_IN0下降沿和LPTIMER_IN1上升沿同时发生或者

中断事件	描述
降沿和上升沿重叠	LPTIMER_IN0上升沿和LPTIMER_IN1下降沿同时发生时，该标志位置1（仅用于译码器模式1）。
LPTIMER_IN0和LPTIMER_IN1高电平重叠	当LPTIMER_IN0和LPTIMER_IN1的高电平重叠时，该标志位置1（仅用于译码器模式1）。
LPTIMER_Inx（x=0,1）高电平计数器溢出	当LPTIMER_Inx的高电平计数器与外部输入高电平计数最大值寄存器（LPTIMER_INHLCMV）值相等时，该标志位置1。
输入高电平计数最大值寄存器更新	当APB总线完成对LPTIMER_INHLCMV寄存器的写操作时，该标志位置1。
LPTIMER计数器由向上计数改为向下计数	在译码器模式中，当计数器由向上计数改为向下计数时，该标志位置1。
LPTIMER计数器由向下计数改为向上计数	在译码器模式中，当计数器由向下计数改为向上计数时，该标志位置1。
计数器自动重载寄存器更新	当APB总线完成对LPTIMER_CAR寄存器的写操作时，该标志位置1。
比较寄存器更新	当APB总线完成对LPTIMER_CMPV寄存器的写操作时，该标志位置1。
外部触发边沿事件	当外部触发的有效边沿发生时，该标志位置1。
计数器自动重载寄存器匹配	当LPTIMER_CNT的值与LPTIMER_CAR寄存器的值相等时，该标志位置1。
比较寄存器匹配	当LPTIMER_CNT的值与LPTIMER_CMPV寄存器的值相等时，该标志位置1。

24.4.16. LPTIMER 调试模式

当Cortex™-M33内核停止时，DBG_CTL1寄存器中相应的LPTIMER_HOLD配置位被置1，LPTIMER的计数器停止。

24.5. LPTIMER 寄存器

LPTIMER 基地址：0x4000 9400

24.5.1. 中断标志寄存器（LPTIMER_INTF）

地址偏移：0x00
复位值：0x0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IN1EIF	IN0EIF	INRFOEIF	INHLOEIF	INHLCOIF	HLCMV UPIF	保留									
r	r	r	r	r	r										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									DOWNIF	UPIF	CARUPIF	CMPV UPIF	ETED EVIF	CARMIF	CMPV MIF
									r	r	r	r	r	r	r

位/位域	名称	描述
31	IN1EIF	LPTIMER_IN1 错误中断标志位 当 LPTIMER_IN1 信号不在 LPTIMER_IN0 信号的两个连续上升沿之间发生跳变时，该标志位由硬件置 1。可以通过向 INTC 寄存器的 IN1EIC 位写入 1 来清除 IN1EIF 标志。 **注意**：该标志位仅用于译码器模式 1。
30	IN0EIF	LPTIMER_IN0 错误中断标志位 当 LPTIMER_IN0 信号不在 LPTIMER_IN1 信号的两个连续上升沿之间发生跳变时，该标志位由硬件置 1。可以通过向 INTC 寄存器的 IN0EIC 位写入 1 来清除 IN0EIF 标志。 **注意**：该标志位仅用于译码器模式 1。
29	INRFOEIF	LPTIMER_IN0 和 LPTIMER_IN1 下降沿和上升沿重叠错误中断标志位 当 LPTIMER_IN0 下降沿和 LPTIMER_IN1 上升沿同时发生或者 LPTIMER_IN0 上升沿和 LPTIMER_IN1 下降沿同时发生时，该标志位由硬件置 1。可以通过向 INTC 寄存器的 INRFOEIC 位写入 1 来清除 INRFOEIF 标志。 **注意**：该标志位仅用于译码器模式 1。
28	INHLOEIF	LPTIMER_IN0 和 LPTIMER_IN1 高电平重叠错误中断标志位 当 LPTIMER_IN0 和 LPTIMER_IN1 的高电平重叠时，该标志位由硬件置 1。可以通过向 INTC 寄存器的 INHLOEIC 位写入 1 来清除 INHLOEIF 标志。

		注意：该标志位仅用于译码器模式 1。
27	INHLCOIF	LPTIMER_Inx（x=0,1）高电平计数器溢出中断标志位 当 LPTIMER_Inx 的高电平计数器与外部输入高电平计数最大值寄存器（LPTIMER_INHLCMV）值相等时，该标志位由硬件置 1。可以通过向 INTC 寄存器的 INHLCOIC 位写入 1 来清除 INHLCOIF 标志。
26	HLCMVUPIF	输入高电平计数最大值寄存器更新中断标志位 当 APB 总线完成对 LPTIMER_INHLCMV 寄存器的写操作时，该标志位由硬件置 1。可以通过向 INTC 寄存器的 HLCMVUPIC 位写入 1 来清除 HLCMVUPIF 标志。
25:7	保留	必须保持复位值。
6	DOWNIF	LPTIMER计数器由向上计数改为向下计数中断标志位 在译码器 0 模式中，当计数器由向上计数改为向下计数时，该标志位由硬件置 1。可以通过向 INTC 寄存器的 DOWNIC 位写入 1 来清除 DOWNIF 标志。
5	UPIF	LPTIMER计数器由向下计数改为向上计数中断标志位 在译码器0模式中，当计数器由向下计数改为向上计数时，该标志位由硬件置1。可以通过向INTC寄存器的UPIC位写入1来清除UPIF标志。
4	CARUPIF	计数器自动重载寄存器更新中断标志位 当 APB 总线完成对 LPTIMER_CAR 寄存器的写操作时，该标志位由硬件置 1。可以通过向 INTC 寄存器的 CARUPIC 位写入 1 来清除 CARUPIF 标志。
3	CMPVUPIF	比较寄存器更新中断标志位 当 APB 总线完成对 LPTIMER_CMPV 寄存器的写操作时，该标志位由硬件置 1。可以通过向 INTC 寄存器的 CMPVUPIC 位写入 1 来清除 CMPVUPIF 标志。
2	ETEDEVIF	外部触发边沿事件中断标志位 当外部触发的有效边沿发生时，该标志位由硬件置 1。可以通过向 INTC 寄存器的 ETEDEVIC 位写入 1 来清除 ETEDEVIF 标志。 **注意**：当外部触发的有效边沿发生在 LPTIMER 启动之后，该标志位不会置位。
1	CARMIF	计数器自动重载寄存器匹配中断标志位 当 LPTIMER_CNT 的值与 LPTIMER_CAR 寄存器的值相等时，该标志位由硬件置 1。可以通过向 INTC 寄存器的 CARMIC 位写入 1 来清除 CARMIF 标志。
0	CMPVMIF	比较寄存器匹配中断标志位 当 LPTIMER_CNT 的值与 LPTIMER_CMPV 寄存器的值相等时，该标志位由硬件置 1。可以通过向 INTC 寄存器的 CMPVMIC 位写入 1 来清除 CMPVMIF 标志。

24.5.2. 中断标志清除寄存器（LPTIMER_INTC）

地址偏移：0x04

复位值：0x0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IN1EIC	IN0EIC	INRFOEIC	INHLOEIC	INHLCOIC	HLCMV UPIC	保留									
w	w	w	w	w	w										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									DOWNIC	UPIC	CARUPIC	CMPV UPIC	ETED EVIC	CARMIC	CMPV MIC
									w	w	w	w	w	w	w

位/位域	名称	描述
31	IN1EIC	LPTIMER_IN1 错误中断标志清除位 向该位写 1 来清除 IN1EIF 标志，写 0 无影响。
30	IN0EIC	LPTIMER_IN0 错误中断标志清除位 向该位写 1 来清除 IN0EIF 标志，写 0 无影响。
29	INRFOEIC	LPTIMER_IN0 和 LPTIMER_IN1 下降沿和上升沿重叠错误中断标志清除位 向该位写 1 来清除 INRFOEIF 标志，写 0 无影响。
28	INHLOEIC	LPTIMER_IN0 和 LPTIMER_IN1 高电平重叠错误中断标志清除位 向该位写 1 来清除 INHLOEIF 标志，写 0 无影响。
27	INHLCOIC	LPTIMER_Inx（x=0,1）高电平计数器溢出中断标志清除位 向该位写 1 来清除 INHLCOIF 标志，写 0 无影响。
26	HLCMVUPIC	输入高电平计数最大值寄存器更新中断标志清除位 向该位写 1 来清除 HLCMVUPIF 标志，写 0 无影响。
25:7	保留	必须保持复位值。
6	DOWNIC	LPTIMER计数器由向上计数改为向下计数中断标志清除位 向该位写 1 来清除 DOWNIF 标志，写 0 无影响。
5	UPIC	LPTIMER计数器由向下计数改为向上计数中断标志清除位 向该位写 1 来清除 UPIF 标志，写 0 无影响。
4	CARUPIC	计数器自动重载寄存器更新中断标志清除位 向该位写 1 来清除 CARUPIF 标志，写 0 无影响。
3	CMPVUPIC	比较寄存器更新中断标志清除位 向该位写 1 来清除 CMPVUPIF 标志，写 0 无影响。
2	ETEDEVIC	外部触发边沿事件中断标志清除位

		向该位写 1 来清除 ETEDEVIF 标志，写 0 无影响。
1	CARMIC	计数器自动重载寄存器匹配中断标志清除位 向该位写 1 来清除 CARMIF 标志，写 0 无影响。
0	CMPVMIC	比较寄存器匹配中断标志清除位 向该位写 1 来清除 CMPVMIF 标志，写 0 无影响。

24.5.3. 中断使能寄存器（LPTIMER_INTEN）

地址偏移：0x08
复位值：0x0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IN1EIE	IN0EIE	INRFOEIE	INHLOEIE	INHLCOIE	HLCMV UPIE	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									DOWNIE	UPIE	CARUPIE	CMPV UPIE	ETED EVIE	CARMIE	CMPV MIE
									rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	IN1EIE	LPTIMER_IN1 错误中断使能位 0：禁止 LPTIMER_IN1 错误中断 1：使能 LPTIMER_IN1 错误中断 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
30	IN0EIE	LPTIMER_IN0 错误中断使能位 0：禁止 LPTIMER_IN0 错误中断 1：使能 LPTIMER_IN0 错误中断 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
29	INRFOEIE	LPTIMER_IN0 和 LPTIMER_IN1 下降沿和上升沿重叠错误中断使能位 0：禁止 LPTIMER_IN0 和 LPTIMER_IN1 下降沿和上升沿重叠错误中断 1：使能 LPTIMER_IN0 和 LPTIMER_IN1 下降沿和上升沿重叠错误中断 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
28	INHLOEIE	LPTIMER_IN0 和 LPTIMER_IN1 高电平重叠错误中断使能位

		0：禁止 LPTIMER_IN0 和 LPTIMER_IN1 高电平重叠错误中断 1：使能 LPTIMER_IN0 和 LPTIMER_IN1 高电平重叠错误中断 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
27	INHLCOIE	LPTIMER_IN0 和 LPTIMER_IN1 高电平计数器溢出中断使能位 0：禁止 LPTIMER_IN0 和 LPTIMER_IN1 高电平计数器溢出中断 1：使能 LPTIMER_IN0 和 LPTIMER_IN1 高电平计数器溢出中断 只有在 LPTIMER 的外部输入高电平计数器禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 INHLCEN 位为 0）。
26	HLCMVUPIE	输入高电平计数最大值寄存器更新中断使能位 0：禁止输入高电平计数最大值寄存器更新中断 1：使能输入高电平计数最大值寄存器更新中断 只有在 LPTIMER 的外部输入高电平计数器禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 INHLCEN 位为 0）。
25:7	保留	必须保持复位值。
6	DOWNIE	LPTIMER计数器由向上计数改为向下计数中断使能位 0：禁止计数器由向上计数改为向下计数中断 1：使能计数器由向上计数改为向下计数中断 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
5	UPIE	LPTIMER计数器由向下计数改为向上计数中断使能位 0：禁止计数器由向下计数改为向上计数中断 1：使能计数器由向下计数改为向上计数中断 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
4	CARUPIE	计数器自动重载寄存器更新中断使能位 0：禁止计数器自动重载寄存器更新中断 1：使能计数器自动重载寄存器更新中断 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
3	CMPVUPIE	比较寄存器更新中断使能位 0：禁止比较寄存器更新中断 1：使能比较寄存器更新中断 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
2	ETEDEVIE	外部触发边沿事件中断使能位

0：禁止外部触发边沿事件中断
1：使能外部触发边沿事件中断
只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。

1	CARMIE	计数器自动重载寄存器匹配中断使能位 0：禁止计数器自动重载寄存器匹配中断 1：使能计数器自动重载寄存器匹配中断 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
0	CMPVMIE	比较寄存器匹配中断使能位 0：禁止比较寄存器匹配中断 1：使能比较寄存器匹配中断 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。

24.5.4. 控制寄存器 0（LPTIMER_CTL0）

地址偏移：0x0C
复位值：0x0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		ETSEL[3]	保留			DECMSEL	DECMEN	CNTMEN	SHWEN	OPSEL	OMSEL	TIMEOUT	ETMEN[1:0]		保留
		rw				rw	rw	rw	rw	rw	rw	rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETSEL[2:0]			保留	PSC[2:0]			保留	TFLT [1:0]		保留	ECKFLT[1:0]		CKPSEL[1:0]		CKSSEL
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	ETSEL[3]	外部触发选择 参考 ETSEL[2:0].
28:26	保留	必须保持复位值。
25	DECMSEL	译码器模式选择 0：译码器模式 0 1：译码器模式 1 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。

24	DECMEN	译码器模式使能 0：译码器模式禁能 1：译码器模式使能 只有在LPTIMER禁能时，才能修改该位（LPTIMER_CTL1寄存器中的LPTEN位为0）。
23	CNTMEN	计数器模式选择 该位用于选择 LPTIMER 计数器的时钟源。 0：计数器在内部时钟每一个脉冲都计数 1：计数器在 LPTIMER_IN0 引脚上的每一个有效脉冲计数 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
22	SHWEN	LPTIMER_CAR 和 LPTIMER_CMPV 影子寄存器使能 0：影子寄存器禁能。在每一次 APB 写操作之后，这两个寄存器立即更新 1：影子寄存器使能。这两个寄存器在 LPTIME 周期结束之后更新 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
21	OPSEL	输出极性选择 该位用于控制 LPTIMER 输出的极性。 0：输出同相。向上计数时，当计数器值与 LPTIMER_CMPV 的值匹配，输出高电平；当计数器值与 LPTIMER_CAR 的值匹配，输出低电平。 1：输出反相。向上计数时，当计数器值与 LPTIMER_CMPV 的值匹配，输出低电平；当计数器值与 LPTIMER_CAR 的值匹配，输出高电平。 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
20	OMSEL	输出模式选择 该位用于控制 LPTIMER 的输出模式。 0：PWM 模式或单脉冲模式（CTNMST 位选择 PWM 模式，SMST 位选择单脉冲模式） 1：置位模式 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
19	TIMEOUT	超时模式使能 该位用于控制 LPTIMER 的超时模式。 0：LPTIMER 启动后，新的触发事件会被忽略 1：LPTIMER 启动后，新的触发事件会复位和重新启动 LPTIMER 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
18:17	ETMEN[1:0]	外部触发模式使能 该位域用于配置 LPTIMER 的外部触发模式。 00：外部触发禁能（软件触发） 01：外部触发上升沿有效 10：外部触发下降沿有效 11：外部触发上升沿和下降沿都有效 只有在 LPTIMER 禁能时，才能修改该位域（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。

16	保留	必须保持复位值。
15:13	ETSEL[2:0]	外部触发选择 该位域用于选择 LPTIMER 的外部触发源 0000: ETI0 (GPIO) 0001: ETI1 (RTC 闹钟 0) 0010: ETI2 (RTC 闹钟 1) 0011: ETI3 (RTC_TAMP0) 0100: ETI4 (RTC_TAMP1) 0101: ETI5 (RTC_TAMP2) 0110: ETI6 (CMP0_OUT) 0111: ETI7 (CMP1_OUT) 1000: ETI8 (CMP2_OUT) 1001: ETI9 (CMP3_OUT) 1010: ETI10 (CMP4_OUT) 1011: ETI11 (CMP5_OUT) 1100: ETI12 (CMP6_OUT) 1101: ETI13 (CMP7_OUT) 只有在 LPTIMER 禁能时，才能修改该位域（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
12	保留	必须保持复位值。
11:9	PSC[2:0]	时钟预分频器选择 该位域用于配置预分频器，将 LPTIMER 的时钟 LPTIMER_CK 分频到计数器时钟 PSC_CLK。 000：$f_{PSC_CLK}=f_{LPTIMER_CK}$ 001：$f_{PSC_CLK}=f_{LPTIMER_CK}/2$ 010：$f_{PSC_CLK}=f_{LPTIMER_CK}/4$ 011：$f_{PSC_CLK}=f_{LPTIMER_CK}/8$ 100：$f_{PSC_CLK}=f_{LPTIMER_CK}/16$ 101：$f_{PSC_CLK}=f_{LPTIMER_CK}/32$ 110：$f_{PSC_CLK}=f_{LPTIMER_CK}/64$ 111：$f_{PSC_CLK}=f_{LPTIMER_CK}/128$ 只有在 LPTIMER 禁能时，才能修改该位域（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
8	保留	必须保持复位值。
7:6	TFLT[1:0]	触发滤波 该位域用于配置触发的数字滤波器，使用该功能时，必须使用内部时钟源。 00：无滤波器，触发信号的每个有效电平都有效 01：触发信号的有效电平变化必须保持 2 个时钟周期 10：触发信号的有效电平变化必须保持 4 个时钟周期 11：触发信号的有效电平变化必须保持 8 个时钟周期

只有在 LPTIMER 禁能时，才能修改该位域（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。

5	保留	必须保持复位值。
4:3	ECKFLT[1:0]	外部时钟滤波 该位域用于配置外部时钟的数字滤波器，使用该功能时，必须使用内部时钟源。 00：无滤波器，外部时钟的每个有效电平变化都有效 01：外部时钟的有效电平变化必须保持 2 个时钟周期 10：外部时钟的有效电平变化必须保持 4 个时钟周期 11：外部时钟的有效电平变化必须保持 8 个时钟周期 只有在 LPTIMER 禁能时，才能修改该位域（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
2:1	CKPSEL[1:0]	时钟极性选择 当 LPTIMER 使用外部时钟源时，该位域用于配置计数器计数的有效边沿。 00：上升沿计数 若 LPTIMER 配置为译码器模式 0（DECMEN=1,DECMSEL=0），译码器的上升沿计数模式有效； 若 LPTIMER 配置为译码器模式 1（DECMEN=1,DECMSEL=1），LPTIMER_IN0 和 LPTIMER_IN1 的输入同相； 若 LPTIMER 外部输入高电平计数器使能（INHLCEN=1），LPTIMER_IN0 和 LPTIMER_IN1 的输入同相。 01：下降沿计数 若 LPTIMER 配置为译码器模式 0（DECMEN=1,DECMSEL=0），译码器的下降沿计数模式有效； 若 LPTIMER 配置为译码器模式 1（DECMEN=1,DECMSEL=1），LPTIMER_IN0 和 LPTIMER_IN1 的输入反相； 若 LPTIMER 外部输入高电平计数器使能（INHLCEN=1），LPTIMER_IN0 和 LPTIMER_IN1 的输入反相。 10：双边沿计数 当外部时钟的双边沿都能有效计数时，LPTIMER 必须使用内部时钟源，并且内部时钟源的频率最少是外部时钟源频率的 4 倍。 若 LPTIMER 配置为译码器模式 0（DECMEN=1，DECMSEL=0），译码器的双边沿计数模式有效； LPTIMER 不能配置为译码器模式 1； 若 LPTIMER 外部输入高电平计数器使能（INHLCEN=1），LPTIMER_IN0 和 LPTIMER_IN1 的输入同相。 11：保留 只有在 LPTIMER 禁能时，才能修改该位域（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
0	CKSSEL	时钟源选择 该位用于选择 LPTIMER 的时钟源。 0：LPTIMER 使用内部时钟源

1：LPTIMER 使用外部时钟源（LPTIMER_IN0）

只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。

注意：当 DECMEN 位置 1 使能译码器模式时，CKSSEL 会自动清零。

24.5.5. 控制寄存器 1（LPTIMER_CTL1）

地址偏移：0x10

复位值：0x0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INHLCEN	LPTENF	保留													
rw	r														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											RDRSTEN	CNTRST	CTNMST	SMST	LPTEN
											rw	rs	rw	rw	rw

位/位域	名称	描述
31	INHLCEN	LPTIMER 外部输入高电平计数器使能 0：禁能 1：使能
30	LPTENF	LPTIMER 从 LPTIMER 内核使能标志位 该位由硬件置位和清零。 0：LPTIMER 禁能 1：LPTIMER 使能
29:5	保留	必须保持复位值。
4	RDRSTEN	LPTIMER_CNT 读异步复位使能 该位由软件置位和清零。 0: 读 LPTIMER_CNT 寄存器使其异步复位禁能 1: 读 LPTIMER_CNT 寄存器使其异步复位使能 只有在 LPTIMER 使能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 1）。
3	CNTRST	计数器值(LPTIMER_CNT)同步复位 该位由软件置位和硬件清零，写 0 无效，软件读返回复位操作状态。 写 1 同步复位计数器（LPTIMER_CNT）寄存器。当 LPTIMER 使用的是与 APB 时钟异步的时钟，复位操作需要 3 个 LPTIMER 内核时钟周期来同步。 **注意：** CNTRST 只可以在硬件清零后被软件写 1。软件连续写 1 时需要检查确认 CNTRST 已经被硬件清零。

		只有在 LPTIMER 使能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 1）。
2	CTNMST	LPTIMER 以连续计数模式启动 该位由软件置位和硬件清零。 0：无效 1：写 1 启动连续计数模式 只有在 LPTIMER 使能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 1）。
1	SMST	LPTIMER 以单次计数模式启动 该位由软件置位和硬件清零。 0：无效 1：写 1 启动单次计数模式 只有在 LPTIMER 使能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 1）。
0	LPTEN	LPTIMER 使能 该位由软件置位和清零。 0：LPTIMER 禁能 1：LPTIMER 使能

24.5.6. 比较寄存器（LPTIMER_CMPV）

地址偏移：0x14
复位值：0x0000

该寄存器只能按字（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
CMPVAL[15:0]															
rw															

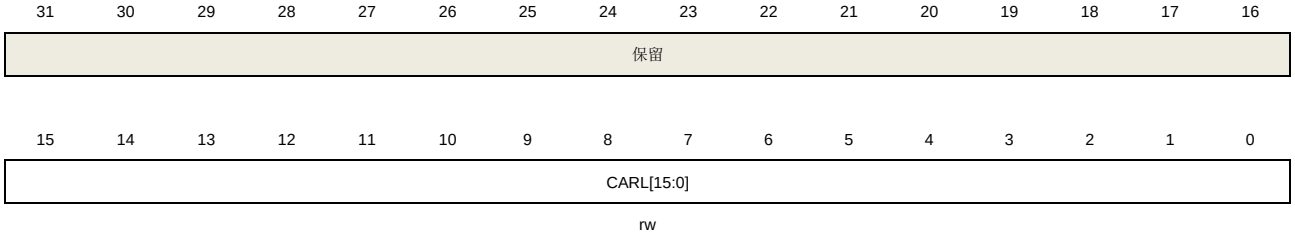
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CMPVAL[15:0]	比较值 这些位定义了计数器的比较值。 只有在 LPTIMER 使能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位

为 1）。

24.5.7. 计数器自动重载寄存器（LPTIMER_CAR）

地址偏移：0x18
复位值：0x0001

该寄存器只能按字（32位）访问。

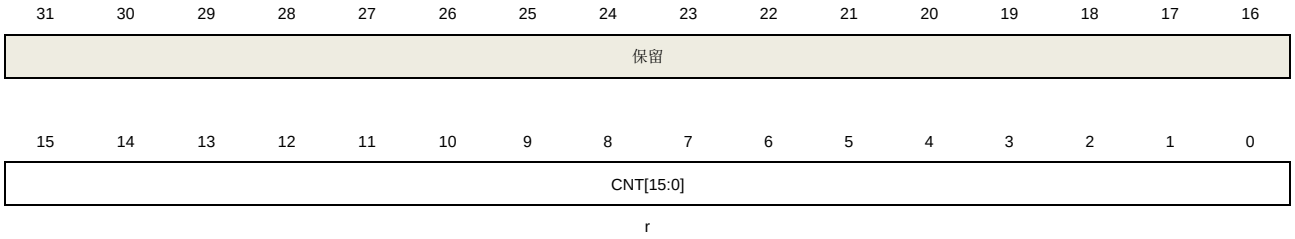


位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CARL[15:0]	计数器自动重载寄存器值 这些位定义了计数器的自动重载值。 只有在 LPTIMER 使能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 1）。

24.5.8. 计数器寄存器（LPTIMER_CNT）

地址偏移：0x1C
复位值：0x0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CNT[15:0]	计数器值 **注意**：当 LPTIMER 使用异步时钟时，对 LPTIMER_CNT 寄存器的读操作可能会返

回不可靠的值。因此，需要执行两个连续的读操作，并且确认两次的读取值是否相同。

24.5.9. 外部输入映射寄存器（LPTIMER_EIRMP）

地址偏移：0x20

复位值：0x0000

该寄存器只能按字（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										IN1_RMP[2:1]		IN0_RMP[2:1]		IN1_RMP	IN0_RMP
										rw		rw		rw	rw

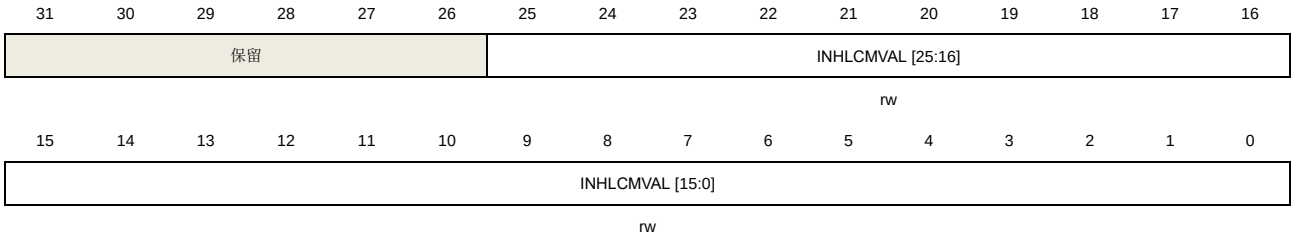
位/位域	名称	描述
31:6	保留	必须保持复位值。
5:4	IN1_RMP[2:1]	外部输入 LPTIMER_IN1 映射扩展位 00: 外部输入引脚 0 映射到 CMP1_OUT 01: 外部输入引脚 0 映射到 CMP3_OUT 10: 外部输入引脚 0 映射到 CMP5_OUT 11: 外部输入引脚 0 映射到 CMP7_OUT
3:2	IN0_RMP[2:1]	外部输入 LPTIMER_IN0 映射扩展位 00: 外部输入引脚 0 映射到 CMP0_OUT 01: 外部输入引脚 0 映射到 CMP2_OUT 10: 外部输入引脚 0 映射到 CMP4_OUT 11: 外部输入引脚 0 映射到 CMP6_OUT
1	IN1_RMP	外部输入 LPTIMER_IN1 映射 0：外部输入引脚 1 映射到 GPIO 1：外部输入引脚 1 映射到 CMPx_OUT，具体 CMP 根据 IN1_RMP[2:1]配置
0	IN0_RMP	外部输入 LPTIMER_IN0 映射 0：外部输入引脚 0 映射到 GPIO 1：外部输入引脚 0 映射到 CMPx_OUT，具体 CMP 根据 IN0_RMP[2:1]配置

24.5.10. 输入高电平计数最大值寄存器（LPTIMER_INHLCMV）

地址偏移：0X24

复位值：0x0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:26	保留	必须保持复位值。
25:0	INHLCMVAL	输入高电平计数最大值 只有在 LPTIMER 的外部输入高电平计数器使能时，才能修改该位（LPTIMER_CTL1 寄存器中的 INHLCEN 位为 1）。

25. 高分辨率定时器（HRTIMER）

25.1. 简介

HRTIMER 具有高分辨率计数时钟，可用于高精度定时。它可以产生 16 个高分辨率的数字信号来灵活地控制电动机或用于电源管理应用。这 16 个数字信号可以独立输出，也可以耦合成 8 对互补信号输出。

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

为了安全起见，HRTIMER 可处理各种故障输入。

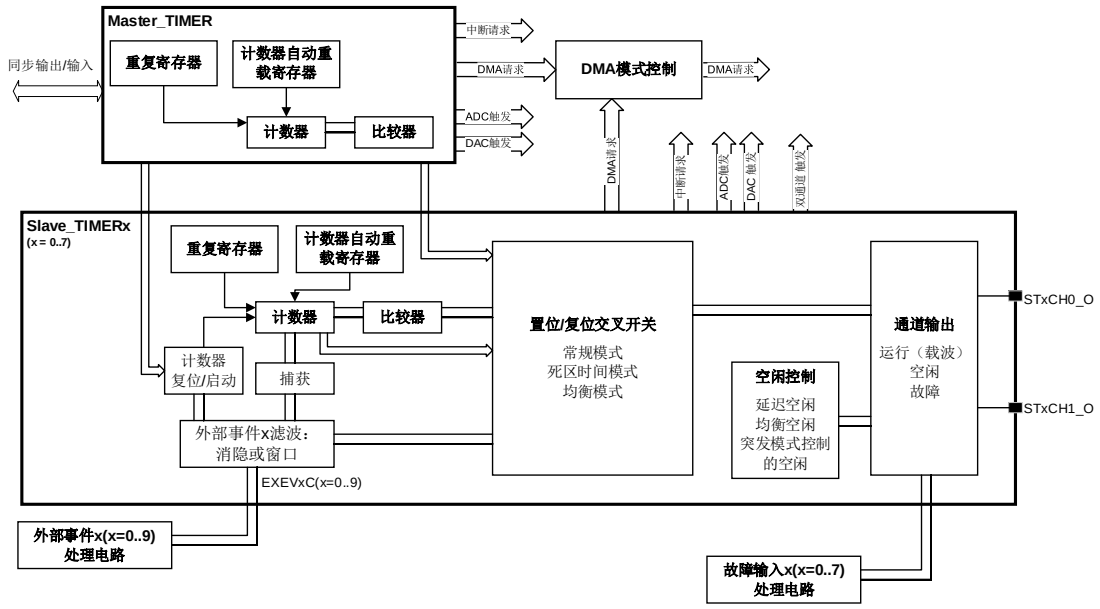
25.2. 主要特征

- 高分辨率定时单元：Master_TIMER，Slave_TIMERx (x=0..7)；
- 16个数字信号输出：它们可由任意一个定时单元控制，可独立输出也可耦合成8对互补输出；
- 同步输出：作为主机同步外部资源；
- 同步输入：作为从机与外部资源同步；
- 多个内部信号连接到ADC和DAC；
- 多种故障输入保护机制：故障输入通道和系统故障；
- 突发模式控制器应用于轻载操作；
- 10个中断向量：Master_TIMER中断，Slave_TIMERx（x = 0..7）中断和故障中断；
- 9个DMA请求：Master_TIMER请求和Slave_TIMERx（x = 0..7）请求。DMA模式可以更新多个寄存器；
- DMA模式用于多个寄存器的更新。

25.3. 结构框图

*图 25-1. HRTIMER 结构框图*给出了 HRTIMER 的内部细节配置。

图 25-1. HRTIMER 结构框图



25.4. 功能说明

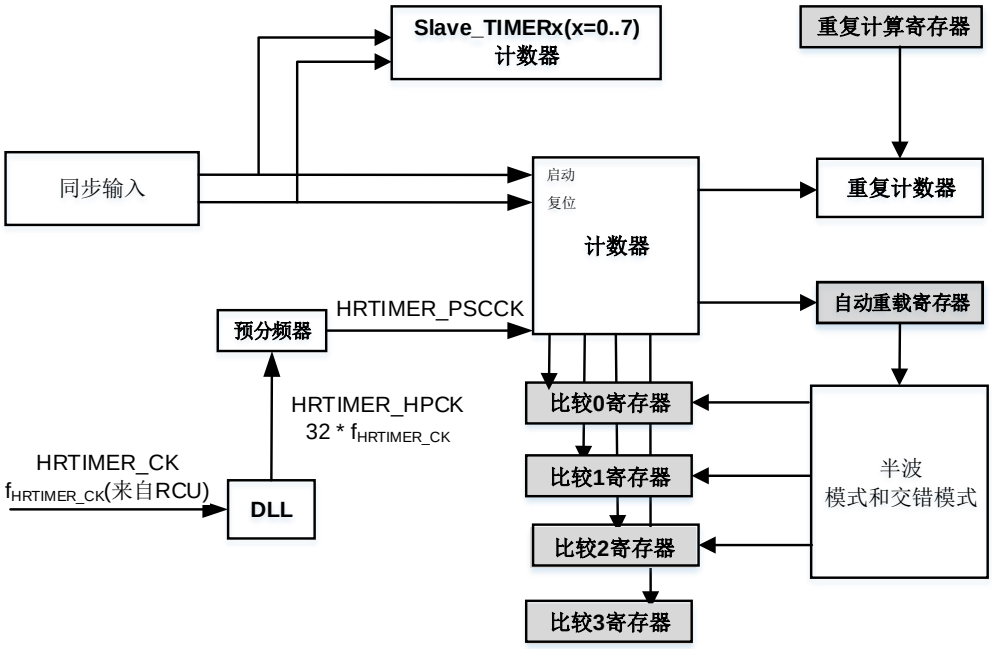
25.4.1. Master_TIMER 单元

Master_TIMER 单元由以下模块组成：

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

*图 25-2. Master_TIMER 结构图*给出了 Master_TIMER 的内部细节配置。

图 25-2. Master_TIMER 结构图



自动重载寄存器和比较 y（y = 0..3）寄存器具有以下限值：

当 CNTCKDIV[2:0] < 3'b101

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

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

当 CNTCKDIV[2:0] >= 3'b101

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

具体请见*表 25-1. 自动重载寄存器和比较 y（y = 0..3）寄存器的限值*。

表 25-1. 自动重载寄存器和比较 y（y = 0..3）寄存器的限值

CNTCKDIV[2:0]	最小值	最大值
3'b000	0x0060	0xFFDF
3'b001	0x0030	0xFFEF
3'b010	0x0018	0xFFF7
3'b011	0x000C	0xFFFB
3'b100	0x0006	0xFFFD
3'b101	0x0003	0xFFFE
3'b110	0x0003	0xFFFE
3'b111	0x0003	0xFFFE

计数器时钟

Master_TIMER 的时钟源是由 RCU 模块提供的 HRTIMER_CK，DLL 用于产生高分辨率时钟 HRTIMER_HPCK（$f_{HRTIMER_HPCK}$ = 32 * $f_{HRTIMER_CK}$），更多信息请参考 ***DLL 校准***。

预分频器（PSC）将高分辨率时钟（HRTIMER_HPCK）除以分频因子 $2^{CNTCKDIV[2:0]}$，得到计数器时钟（HRTIMER_PSCCK）。该分频因子由 HRTIMER_MTCTL0 寄存器中的 CNTCKDIV[2:0]位域控制。它们之间的频率关系可以表示如下：

$$f_{HRTIMER_PSCCK} = f_{HRTIMER_HPCK} / 2^{CNTCKDIV[2:0]} \quad (25\text{-}1)$$

注意：一旦 Master_TIMER 使能了，就不能修改时钟分频 CNTCKDIV[2:0]的值。

*图 25-3. 分频为 16 时计数器时钟*显示了将寄存器 HRTIMER_MTCAR 设置为 0x0104，HRTIMER_MTCTL0 寄存器中的位域 CNTCKDIV[2: 0]设置为 3'b100 时计数器的动作。

图 25-3. 分频为 16 时计数器时钟

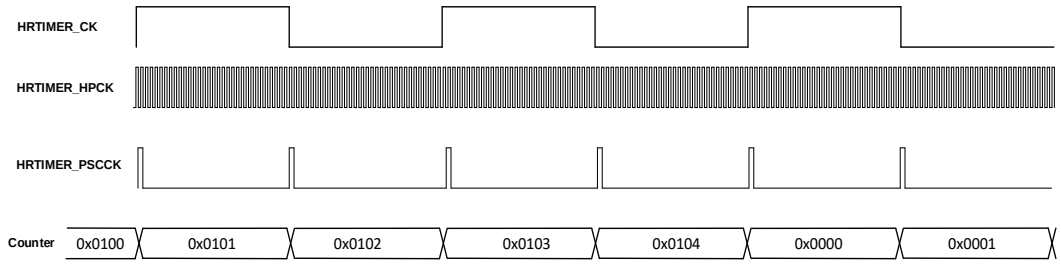


*表 25-2. fHRTIMER_CK = 216MHz 的分辨率*列出了 $f_{HRTIMER_CK}$ 为 216MHz 时的不同分辨率。

表 25-2. $f_{HRTIMER_CK}$ = 216MHz 的分辨率

CNTCKDIV[2:0]	$f_{HRTIMER_PSCCK}$	分辨率
3'b000	216*32MHz=6.912GHz	144.68ps
3'b001	216*16MHz=3.456GHz	289.35ps
3'b010	216*8MHz=1.728GHz	578.70ps
3'b011	216*4MHz=864MHz	1.16ns
3'b100	216*2MHz=432MHz	2.31ns
3'b101	216*1MHz=216MHz	4.63ns
3'b110	216/2MHz=108MHz	9.26ns
3'b111	216/4MHz=54MHz	18.52ns

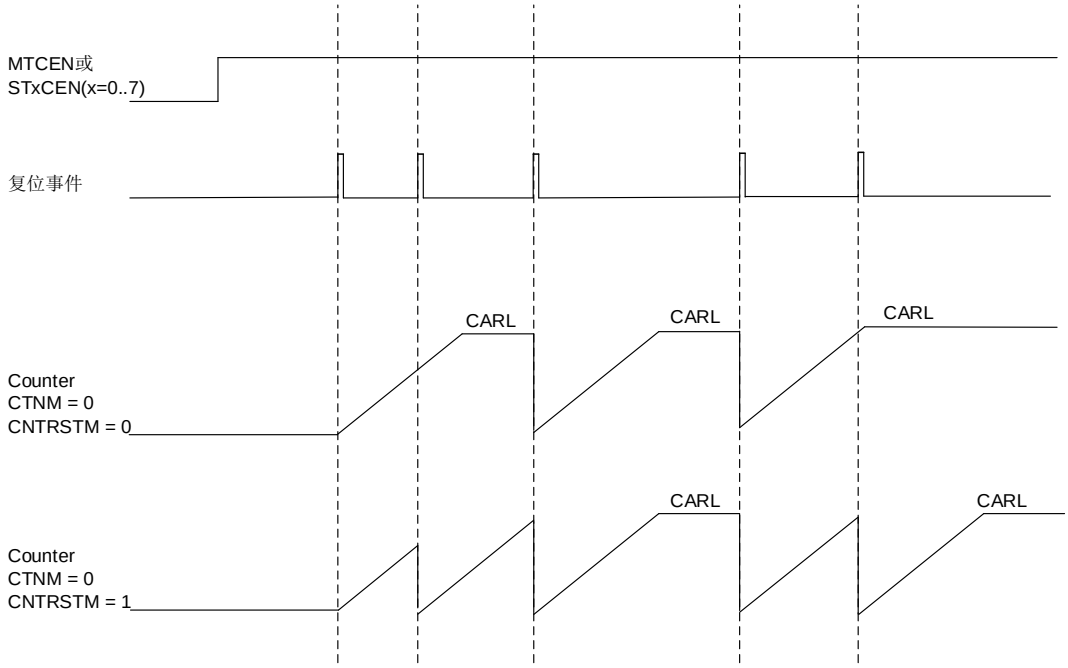
向上计数模式

计数器从 0 连续递增到计数器重载值，该值在 HRTIMER_MTCAR 寄存器中定义。计数器有两种工作模式：单脉冲模式（HRTIMER_MTCTL0 寄存器中的 CTNM = 0）或连续模式（CTNM = 1）。

在单脉冲模式下，将 HRTIMER_MTCTL0 寄存器中的 MTCEN 位置 1 后，第一个复位事件将启动计数器。当计数到计数器重载值时，计数器停止并生成周期事件。然后，其他的复位事件将复位并

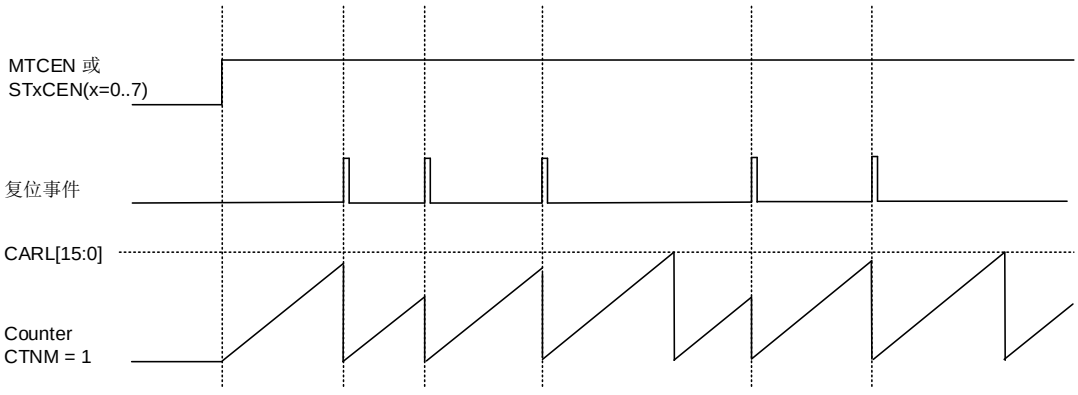
重新启动计数器。在计数过程中，如果 HRTIMER_MTCTL0 寄存器中的 CNTRSTM = 1，则复位事件将复位并重新启动计数器，否则将被忽略。*图 25-4. 单脉冲模式下计数器的动作*显示了单脉冲模式下的计数器运行情况。

图 25-4. 单脉冲模式下计数器的动作



在连续模式下，一旦 HRTIMER_MTCTL0 寄存器中的 MTCEN 位置 1，计数器将立即启动。当计数到计数器重载值时，计数器从 0 重新启动，并产生翻转事件。与单脉冲模式不同，随时生成的复位事件将复位并重启计数器。*图 25-5. 连续模式下计数器的动作*显示了连续模式下的计数器运行情况。

图 25-5. 连续模式下计数器的动作



重复计数器

HRTIMER_MTCTL0 寄存器中的 MTCEN 位置 1 时，重复计数器将加载 HRTIMER_MTCREP 寄

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

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

*图 25-6. 连续模式下重复计数器的动作*显示了在连续模式下重复计数器的运行情况。

图 25-6. 连续模式下重复计数器的动作

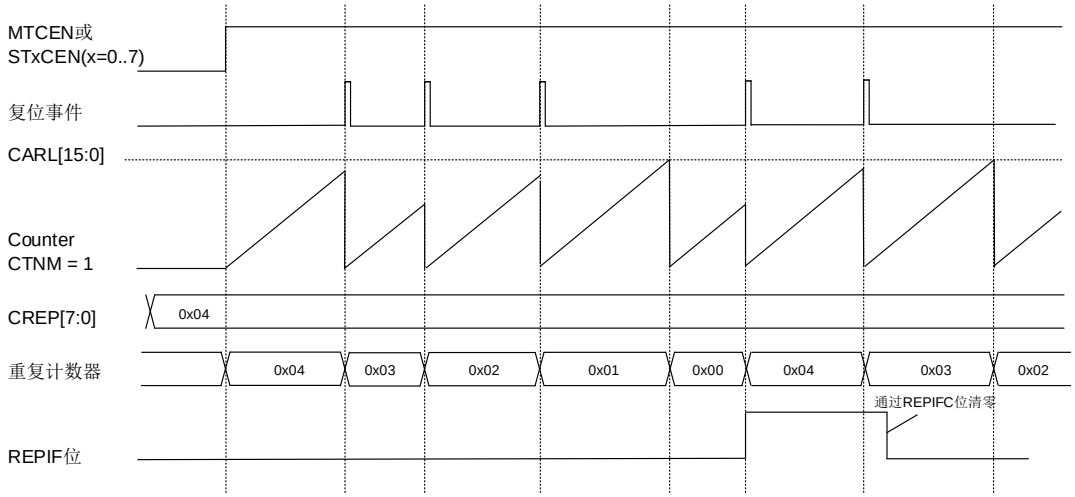


*图 25-7. 单脉冲模式下，CNTRSTM = 0 时重复计数器的动作*显示了单脉冲模式下，CNTRSTM = 0 时重复计数器的运行情况。

图 25-7. 单脉冲模式下，CNTRSTM = 0 时重复计数器的动作

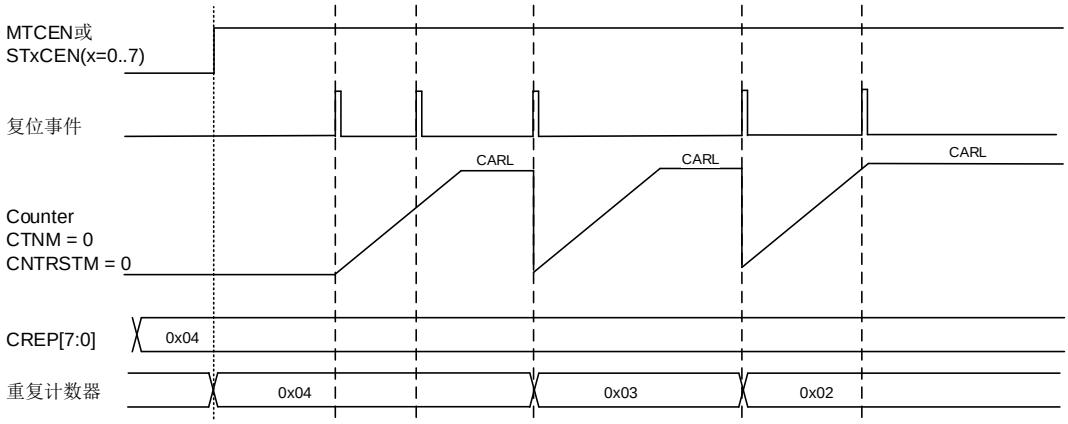
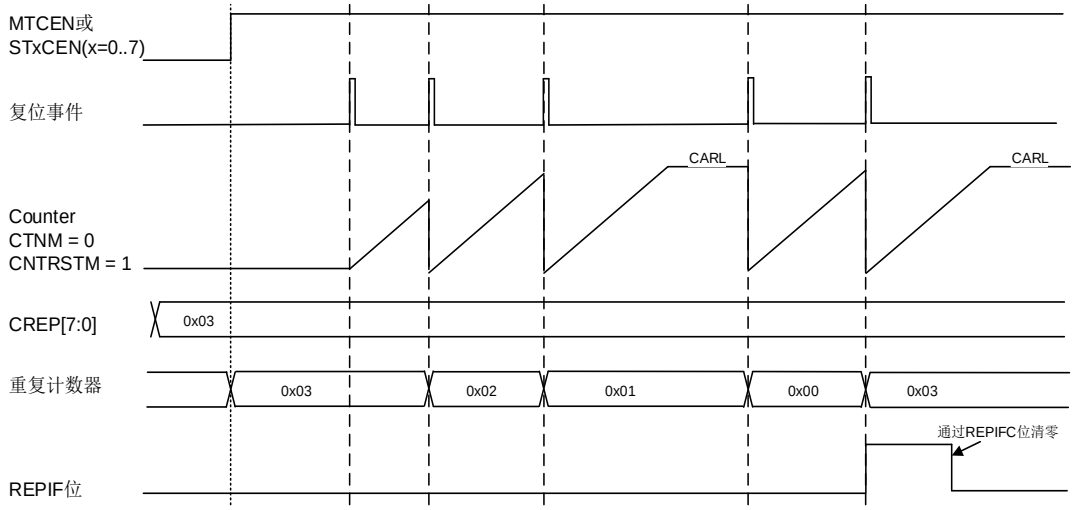


*图 25-8. 单脉冲模式下，CNTRSTM = 1 时重复计数器的动作*显示了单脉冲模式下，CNTRSTM = 1 时重复计数器的运行情况。

图 25-8. 单脉冲模式下，CNTRSTM = 1 时重复计数器的动作



计数器复位

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

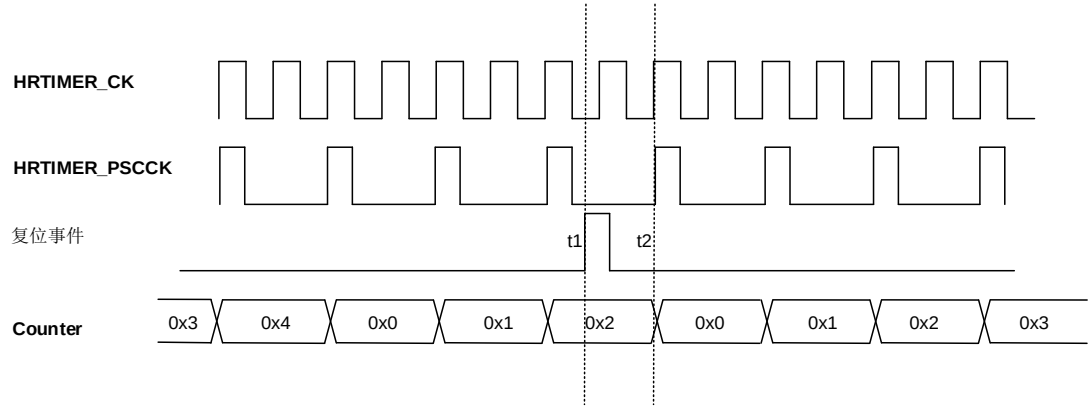
将 MTSRST 位置 1（由硬件自动清除）将使计数器复位。

当 HRTIMER_MTCTL0 寄存器中的 SYNIRST 位置 1 时，同步输入可以复位计数器。详细信息请参考*同步输入*。

当计数器时钟 HRTIMER_PSCCK 的预分频系数大于 32（CNTCKDIV [2:0] > 3'b101）时，计数器复位事件将延迟到 HRTIMER_PSCCK 的下一个上升沿。

*图 25-9. 当预分频系数为 64 时，复位事件重新同步*显示了连续模式下，CNTCKDIV[2:0] = 3'b110，HRTIMER_MTCAR = 0x4 时的运行情况。

图 25-9. 当预分频系数为 64 时，复位事件重新同步



比较

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

比较事件会将相应的比较中断标志位置 1（HRTIMER_MTINTF 寄存器中的 CMPxIF 位，x = 0..3），如果比较中断或 DMA 请求使能（HRTIMER_MTDMAINTEN 寄存器中的 CMPxIE = 1 或 CMPxDEN = 1，x = 0..3），则会产生一个比较中断或 DMA 请求。通过写 1 到 HRTIMER_MTINTFC 中的 CMPxIF 位（x = 0..3）可以清除比较中断标志。

半波模式

当 HRTIMER_MTCTL0 中的 HALFM 位置 1 时，半波模式使能。此模式将比较 0 有效寄存器的值强制为计数器重载值的一半，但 HRTIMER_MTCMP0V 寄存器的值不会更新为（HRTIMER_MTCAR / 2）的值。半波模式主要用于生成固定占空比为 50％的方波。

当 HRTIMER_MTCTL0 寄存器中的 SHWEN 位置 1 时，将使能影子寄存器，比较 0 有效寄存器的值在更新事件时刷新。反之，比较 0 有效寄存器在新值写入后立即刷新。

交错模式

此模式有助于实现与半模式相辅相成的替代拓扑结构。当 HRTIMER_MTCAR 值更新时，比较值寄存器会被自动重新计算。通过 HRTIMER_MTCTL0 和 HRTIMER_STxCTL0 中的 ALTM[1:0]位选择交错模式。

表 25-3. 交错模式选择

ALTM[1:0]	交错模式
00	禁止
01	三重交错（120°）
10	四重交错(90°)
11	保留

*表 25-4. 交错模式的比较值*显示了交错模式的比较值，比较寄存器的内容将被覆盖，相应的比较事件可用作触发器，以设置或复位从定时器。

表 25-4. 交错模式的比较值

模式	三重交错（120°）	四重交错(90°)
HRTIMER_MTCMP0V	HRTIMER_MTCAR /3	HRTIMER_MTCAR /4
HRTIMER_MTCMP1V	2 x HRTIMER_MTCAR /3	HRTIMER_MTCAR /2
HRTIMER_MTCMP2V	无影响	3 x HRTIMER_MTCAR /4

注意：在交错模式中，比较寄存器由硬件控制，写入这些寄存器时没有影响。然而，预加载寄存器存储了比较值，并且在退出交错模式时将变为活动状态。

同步输入启动/复位计数器

当 HRTIMER_MTCTL0 寄存器中的 SYNIRST 位置 1 时，同步输入可以产生计数器复位事件；当 HRTIMER_MTCTL0 寄存器中的 SYNISTRT 位置 1 时，同步输入可以启动计数器。更多信息请参考*同步输入*。

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

更新事件和影子寄存器

Master_TIMER 中的某些寄存器具有影子寄存器。MCU 复位后，影子寄存器被禁用。如果将 HRTIMER_MTCTL0 寄存器中的 SHWEN 位清 0，则将禁用影子寄存器。写入这些寄存器的值将转移到活动寄存器中并生效。

HRTIMER_MTCTL0 寄存器中的 SHWEN 位置 1，将使能影子寄存器并预加载这些寄存器。写入这些寄存器的值将被传送到影子寄存器，且不会立即生效。当发生更新事件时，影子寄存器内容将转移到活动寄存器中并立即生效。

*表 25-5. Master_TIMER 影子寄存器和更新事件*列出了具有影子寄存器的寄存器和相应的更新事件。

表 25-5. Master_TIMER 影子寄存器和更新事件

具有影子寄存器的寄存器	影子寄存器使能位	更新事件
HRTIMER_MTDMAINTEN	HRTIMER_MTCTL0 寄存器中的 SHWEN 位	软件（MTSUP位）
HRTIMER_MTCAR		重复事件（UPREP = 1）
HRTIMER_MTCREP		DMA模式结束事件（UPSEL[1:0] = 2'b01）
HRTIMER_MTCMP0V		DMA模式结束事件之后的翻转事件（UPSEL[1:0] = 2'b10）
HRTIMER_MTCMP1V		
HRTIMER_MTCMP2V		
HRTIMER_MTCMP3V		

Master_TIMER 有 4 个更新选项：

1. 软件生成更新事件。写 1 到 HRTIMER_CTL1 寄存器的 MTSUP 位可以产生更新事件。此时，无论 HRTIMER_MTCTL0 寄存器中的 UPSEL[1:0]位如何配置，所有挂起的硬件更新事件都将被忽略；
2. 重复事件生成更新事件。如果 HRTIMER_MTCTL0 寄存器中的 UPREP 位置 1，由翻转事件或复位事件引起的重复事件会生成更新事件。HRTIMER_MTCTL0 寄存器中的 UPSEL[1:0] = 2'b10，则重复事件不生成更新事件；
3. 当 DMA 模式下的 DMA 传输完成时，生成更新事件。如果 HRTIMER_MTCTL0 寄存器中的

UPSEL[1:0] = 2'b01，则在 DMA 模式下的 DMA 传输完成时，硬件会自动生成更新事件。也可以通过软件或重复事件来生成更新事件。

4. 当 DMA 模式下的 DMA 传输完成后，计数器的翻转会产生更新事件。如果 HRTIMER_MTCTL0 寄存器中的 UPSEL [1:0] = 2'b10，则在 DMA 模式下的 DMA 传输完成后，计数器发生翻转事件时，硬件会自动生成更新事件。也可以通过软件生成更新事件。

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

DAC 触发

当 Master_TIMER 的更新事件发生时，如果 HRTIMER_MTCTL0 寄存器中的 DACTRGS[1:0] != 2'b00，则在 HRTIMER_DACTRIGOx（x = 0..3）上生成 DAC 触发请求。如果 HRTIMER_MTCTL0 寄存器中的 DACTRGS [1:0] = 2'b00，则不会生成 DAC 触发请求。HRTIMER_DACTRIGOx（x = 0..3）是从 Master_TIMER 连接到 DAC 模块的内部信号。有关更多信息，请参考 ***DAC 触发***。

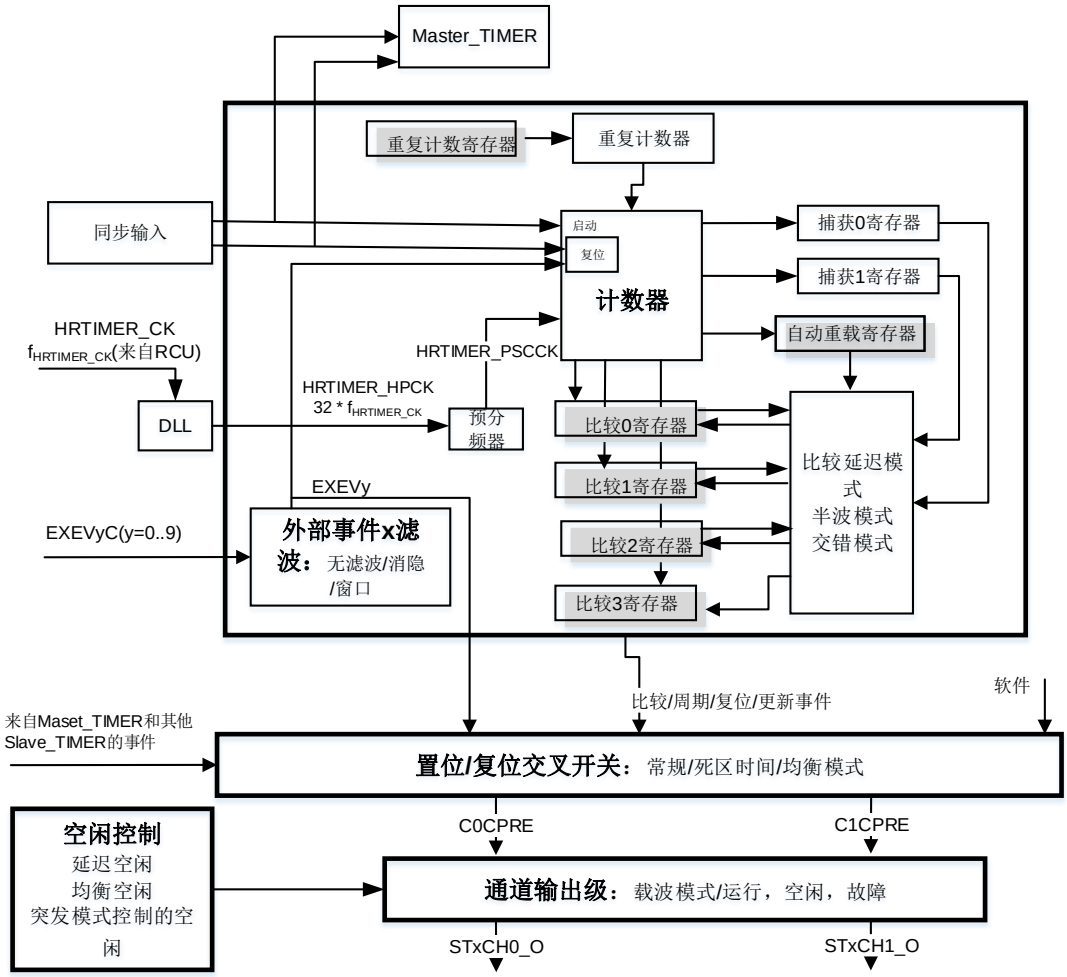
25.4.2. Slave_TIMERx(x=0..7)单元

HRTIMER 具有 8 个相同结构的从定时器：Slave_TIMERx（x = 0..7）。每个从定时器都由以下组件构成：

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

图 25-10. Slave_TIMERx 结构框图显示了 Slave_TIMERx 的结构框图。

图 25-10. Slave_TIMERx 结构框图



自动重载寄存器和比较 y（y = 0..3）寄存器具有以下限值：

当 CNTCKDIV[2:0] < 3'b101

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

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

当 CNTCKDIV[2:0] >= 3'b101

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

具体请见*表 25-1. 自动重载寄存器和比较 y（y = 0..3）寄存器的限值*。

计数器和捕获 y（y = 0,1）寄存器还具有以下限制：对于计数器时钟分频低于 32（CNTCKDIV [2:0] <5），最低有效位忽略。它们不能进行写操作和读操作时值为零。详见*表 25-6. 计数器和捕获*

y(y=0,1)寄存器限值。

表 25-6. 计数器和捕获 y(y=0,1)寄存器限值

CNTCKDIV[2:0]	无效位
3'b000	位 4~位 0
3'b001	位 3~位 0
3'b010	位 2~位 0
3'b011	位 1~位 0
3'b100	位 0
3'b101	x
3'b110	x
3'b111	x

注意："x"表示所有位都有效。

计数器时钟

Slave_TIMERx 的时钟源是来自 RCU 模块的 HRTIMER_CK，DLL 用于产生高分辨率时钟 HRTIMER_HPCK（$f_{HRTIMER_HPCK} = 32 * f_{HRTIMER_CK}$），更多信息请参考 *DLL 校准*。

预分频器（PSC）将高分辨率时钟（HRTIMER_HPCK）除以分频因子 $2^{CNTCKDIV[2:0]}$，得到计数器时钟（HRTIMER_PSCCK）。该分频因子由 HRTIMER_STxCTL0 寄存器中的 CNTCKDIV[2:0]位域控制。它们之间的频率关系可以表示如下：

$$f_{HRTIMER_PSCCK} = f_{HRTIMER_HPCK} / 2^{CNTCKDIV[2:0]} \quad (25\text{-}2)$$

注意：一旦 Slave_TIMERx 使能了，就不能修改时钟分频 CNTCKDIV[2:0]的值，CNTCKDIV[2:0]在 HRTIMER_STxCTL0 寄存器中。

参考*图 25-3. 分频为 16 时计数器时钟*和*表 25-2. fHRTIMER CK = 216MHz 的分辨率*可得更多细节。

向上计数模式

计数器从 0 连续递增到计数器重载值，该值在 HRTIMER_STxCAR 寄存器中定义。计数器有两种工作模式：单脉冲模式（HRTIMER_STxCTL0 寄存器中的 CTNM = 0）和连续模式（HRTIMER_STxCTL0 寄存器中的 CTNM = 1）。

在单脉冲模式下，将 HRTIMER_MTCTL0 寄存器中的 STxCEN 位置 1 后，第一个复位事件将启动计数器。当计数到计数器重载值时，计数器停止并生成周期事件。然后，其他的复位事件将复位并重新启动计数器。在计数过程中，如果 HRTIMER_STxCTL0 寄存器中的 CNTRSTM = 1，则复位事件将复位并重新启动计数器，否则将被忽略。*图 25-4. 单脉冲模式下计数器的动作*显示了单脉冲模式下的计数器运行情况。

在连续模式下，一旦 HRTIMER_MTCTL0 寄存器中的 STxCEN 位置 1，计数器将立即启动。当计

数到计数器重载值时，计数器从 0 重新启动，并产生翻转事件。与单脉冲模式不同，随时生成的复位事件将复位并重启计数器。*图 25-5. 连续模式下计数器的动作*显示了连续模式下的计数器运行情况。

重复计数器

HRTIMER_MTCTL0 寄存器中的 STxCEN 位置 1 时，重复计数器将加载 HRTIMER_STxCREP 寄存器的值。当由于复位事件或连续模式下的翻转事件清零计数器时，重复计数器值递减。当重复计数器值达到零时，复位事件或连续模式下的翻转事件将产生一个重复事件并重新加载 HRTIMER_STxCREP 寄存器的值。

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

连续模式下重复计数器的运行情况如*图 25-6. 连续模式下重复计数器的动作*所示。

单脉冲模式下，CNTRSTM = 0 时重复计数器的运行情况如*图 25-7. 单脉冲模式下，CNTRSTM = 0 时重复计数器的动作*所示。

单脉冲模式下，CNTRSTM = 1 时重复计数器的运行情况如*图 25-8. 单脉冲模式下，CNTRSTM = 1 时重复计数器的动作*所示。

计数器复位

计数器可以通过以下三种信号源复位：

1.软件。软件写 1 到 HRTIMER_CTL1 寄存器的 STxSRST 位。

2.同步输入启动/复位计数器。

3.HRTIMER_STxCNTRST 寄存器和 HRTIMER_STxCNTRSTA 寄存器中配置的事件。

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

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

写 1 到 STxSRST 位（由硬件自动清除）使计数器复位。Master_TIMER 和 Slave_TIMERx（x = 0..7）的这些控制位都在 HRTIMER_CTL1 寄存器中，可以同时复位多个计数器。

当 HRTIMER_STxCTL0 寄存器中的 SYNIRST 位置 1 时，同步输入可以复位计数器。请参考*同步输入*。

可以在 HRTIMER_STxCNTRST 和 HRTIMER_STxCNTRSTA 寄存器中同时配置 39 个事件来复位计数器，这些事件可以分为四类：

- Slave_TIMERx：更新事件，比较1事件和比较3事件；
- 其他Slave_TIMERy（例如x = 1，则y = 0，2..7）：比较0事件，比较1事件和比较3事件；
- Master_TIMER：比较0事件，比较1事件，比较2事件，比较3事件和复位事件；
- 外部事件y（y = 0..9）：EXEVy为Slave_TIMERx中的外部事件的滤波信号。

当计数器时钟 HRTIMER_PSCCK 的预分频系数大于 16（CNTCKDIV [2:0] > 3'b101）时，计数器复位事件将延迟到 HRTIMER_PSCCK 的下一个上升沿。具体请见*图 25-9. 当预分频系数为64 时，复位事件重新同步*。

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

捕获

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

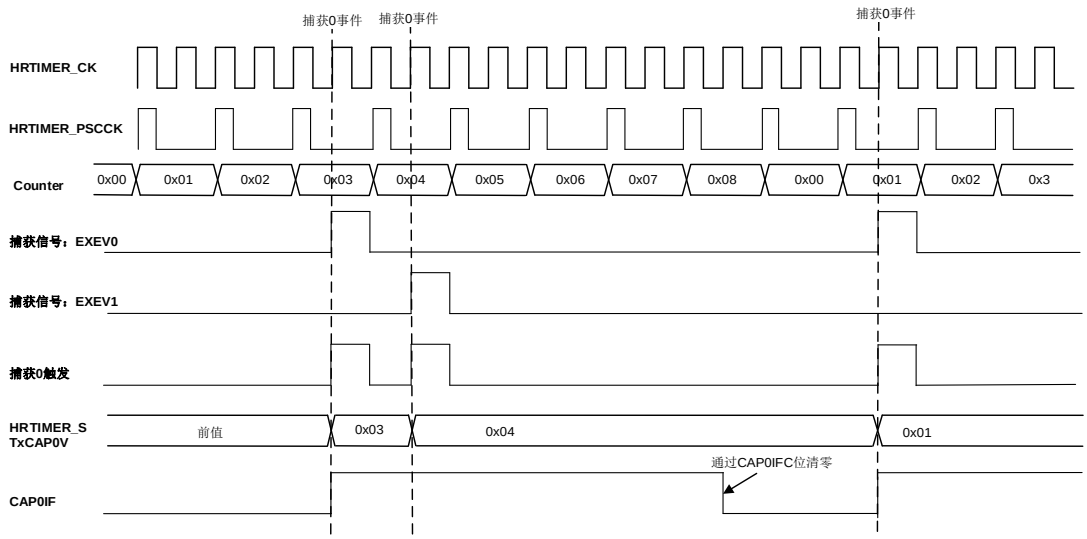
当选定的触发信号发生时，计数器的当前值被捕获到 HRTIMER_STxCAPyV（y = 0,1）寄存器中。同时，HRTIMER_STxINTF 寄存器中的 CAPyIF（y = 0,1）位置 1，如果 HRTIMER_STxDMAINTEN 寄存器中的 CAPyIE（y = 0,1）= 1 或 CAPyDEN（y = 0,1）= 1，则生成相应的捕获中断和 DMA 请求。可以通过写 1 到 HRTIMER_STxINTFC 寄存器中的 CAPyIFC 位来清除捕获中断标志位 CAPyIF。

捕获 0 触发事件在 HRTIMER_STxCAP0TRG 寄存器中定义，捕获 1 触发事件在 HRTIMER_STxCAP1TRG 寄存器中定义。当选择了多个触发事件时，所有的触发事件是逻辑“或”运算的。

注意： 如果将外部事件配置为具有电平有效，则只能在 HRTIMER_STxCAP0TRG 和 HRTIMER_STxCAP1TRG 寄存器中使能一个外部事件。

捕获溢出是无法防止的，即使先前的捕获值未读取或捕获标志未清除，新的捕获仍将被触发，并且新的捕获值将覆盖先前的值。请参考*图 25-11. EXEV0 和 EXEV1 触发捕获 0*。

图 25-11. EXEV0 和 EXEV1 触发捕获 0

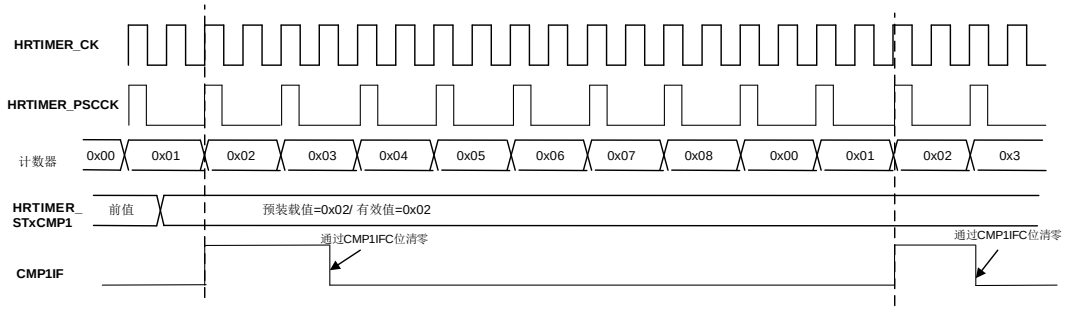


比较

Slave_TIMERx 单元有四个比较寄存器：HRTIMER_STxCMPyV（y = 0..3）。当计数器值与比较寄存器值匹配时，将产生一个比较事件。具体请见*图 25-12. STxCAR=0x8, STxCMP1V=0x02 时，比较 1 寄存器的动作*。

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

图 25-12. STxCAR=0x8, STxCMP1V=0x02 时，比较 1 寄存器的动作



半波模式

当 HRTIMER_STxCTL0 中的 HALFM 位置 1 时，半波模式使能。此模式将比较 0 有效寄存器的值强制为计数器重载值的一半，但 HRTIMER_STxCMP0V 寄存器的值不会更新为（HRTIMER_STxCAR / 2）的值。半波模式主要用于生成固定占空比为 50%的方波。

当 HRTIMER_STxCTL0 寄存器中的 SHWEN 位置 1 时，将使能影子寄存器，比较 0 有效寄存器

的值在更新事件时刷新。反之，比较 0 有效寄存器在新值写入后立即刷新。

交错模式

此模式有助于实现与半波模式相辅相成的替代拓扑结构。当 HRTIMER_STxCAR 值更新时，比较值寄存器会被自动重新计算。通过 HRTIMER_STxCTL0 和 HRTIMER_STxCTL0 中的 HALFM 位和 ALTM[1:0]位选择交错模式，交错模式仅在 HALFM 位置 0 时有效。

表 25-7. 交错模式选择

ALTM[1:0]	交错模式
00	禁止
01	三重交错（120°）
10	四重交错(90°)
11	保留

*表 25-4. 交错模式的比较值*显示了交错模式的有效寄存器比较值，比较有效寄存器的内容将被覆盖，相应的比较事件可用作触发器，以设置或复位从定时器。

表 25-8. 两种交错模式比较值

模式	三重交错模式(120°)	四重交错模式(90°)
HRTIMER_STxCMP0V	HRTIMER_STxCAR /3	HRTIMER_STxCAR /4
HRTIMER_STxCMP1V	2 x HRTIMER_STxCAR /3	2 x HRTIMER_STxCAR /4
HRTIMER_STxCMP2V	无影响	3 x HRTIMER_STxCAR /4

注意：在交错模式中，比较寄存器由硬件控制，软件写入对比较寄存器无效。预装载寄存器保存比较值，并在退出交错模式后生效。

空占空比异常情况

HRTIMER 不支持小于 3 个 t_{HPTMER_CK} 时钟周期的输出脉冲，具体参考*输出准备信号：窄脉冲管理*。例如，如果 CNTCKDIV[2:0] = 3'b000，则为 0x60，如果 CNTCKDIV[2:0] = 3'b001，则为 0x30，如果 CNTCKDIV[2:0] = 3'b010，则为 0x18。小于上述值的脉冲无法正常输出。

通过在 HRTIMER_STxCMP0V 和 HRTIMER_STxCMP2V 中写入空值，并配置以下操作。在 HRTIMER 周期中将跳过一个脉冲。

- 更新事件生成通道 0 的“置位请求”。
- 比较 0（比较 2）事件生成“复位请求”。
- 比较 0（比较 2）事件仅在比较事件所属的计时器单元中有效。

在上述应用条件下，重新编写比较值 HRTIMER_STxCMP0V 和 HRTIMER_STxCMP2V，它们之间的差值大于 3，将会恢复正常输出。

交换模式

HRTIMER_CTL1 中设置 EXCx，可以交换 Slave timer 的两个输出 CH0 和 CH1，并且两个通道的输出在下一个更新事件发生时生效。在置位和复位之前，两个输出 CH0 和 CH1 将按照以下交叉方式进行：

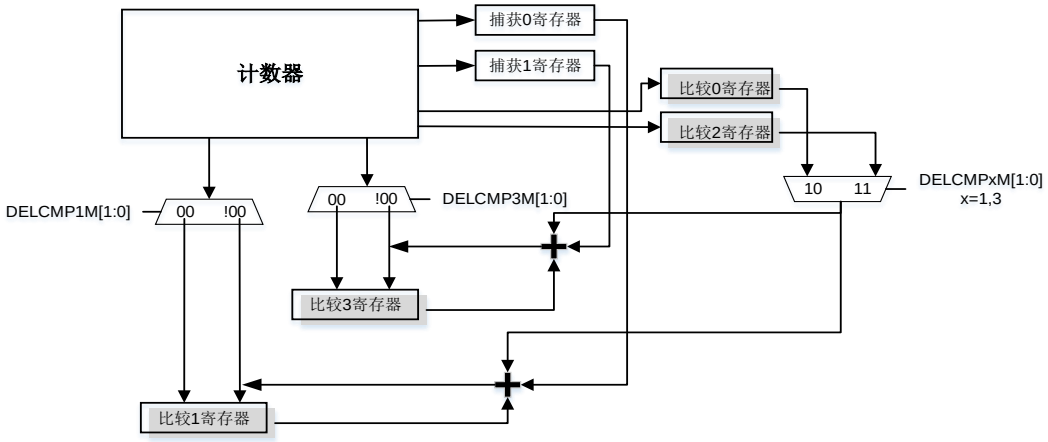
- 如果EXCx = 0，则STxCH0SET和STxCH0RST控制CH0的输出，STxCH1SET和STxCH1RST控制CH1的输出。
- 如果EXCx = 1，则STxCH0SET和STxCH0RST控制CH1的输出，STxCH1SET和STxCH1RST控制CH0的输出。

注意：交换模式仅影响预装载寄存器；在使用交换模式时，必须启用影子寄存器。

比较延迟模式

此模式仅用于比较 y（y = 1,3）寄存器，并由 HRTIMER_STxCTL0 寄存器中的 DELCMPyM[1:0] 位域控制。比较寄存器与计数器比较的实际值是重新计算值，该值在捕获 0/1 触发或比较 0/ 2 事件之后重新计算得到，具体如*图 25-13. 比较延迟模式框图*。此模式允许通过硬件将生成的波形与捕获触发同步。

图 25-13. 比较延迟模式框图



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

当没有捕获触发或比较事件发生时，不生成比较 y 事件。捕获触发事件发生后，将比较 y 有效寄存器中的值与对应的 HRTIMER_STxCAP0V 或 HRTIMER_STxCAP1V 寄存器值相加，然后将其与计数器进行比较。比较 1 寄存器与捕获 0 寄存器和比较 0 寄存器/比较 2 寄存器关联，而比较 3 寄存器与捕获 1 寄存器和比较 0 寄存器/比较 2 寄存器关联。

注意：重新计算的值被传输到一个内部寄存器，且无法读取。

HRTIMER_STxCTL0 寄存器中的 DELCMP1M[1:0]位域（比较 1 事件）和 DELCMP3M[1:0]位域（比较 3 事件）可用于配置延迟模式。下面以 DELCMP1M[1:0]为例：

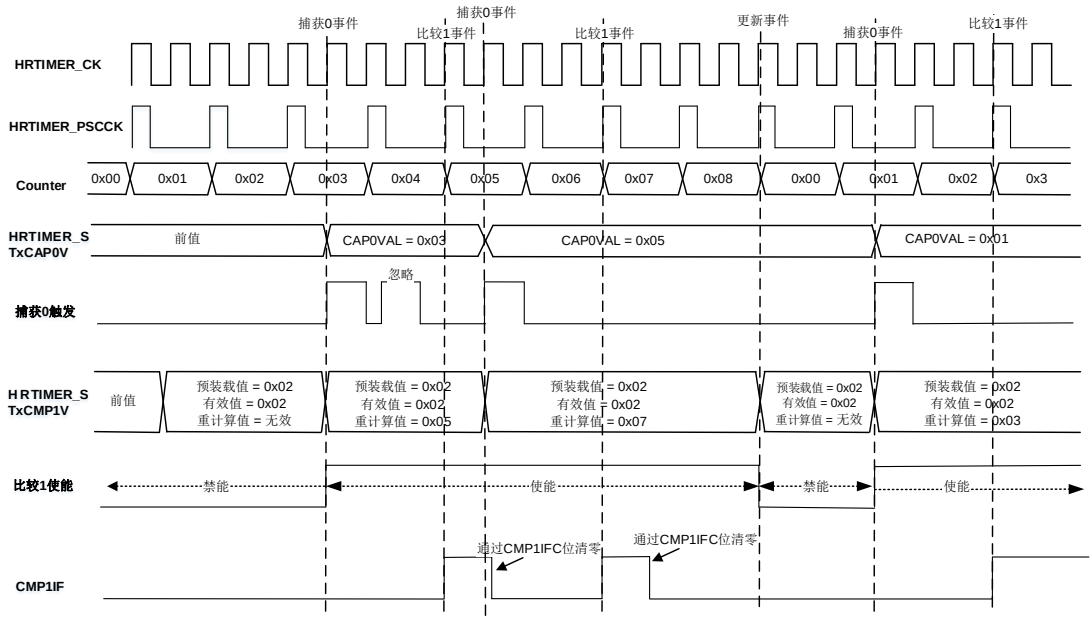
■ 2’b00，比较1延迟模式禁能

比较 1 延迟模式禁用。一旦计数器值等于比较 1 寄存器的值，就会发生比较匹配。参见*图 25-12. STxCAR=0x8, STxCMP1V=0x02 时，比较 1 寄存器的动作*。

■ 2’b01，比较1延迟模式0

捕获 0 事件发生后，将重新计算比较 1 寄存器的值（比较 1 有效寄存器值+捕获 0 寄存器值）。一旦计数器值等于重新计算后的比较 1 寄存器值，就会发生比较 1 事件。参见*图 25-14. 比较 1 延迟模式 0*。

图 25-14. 比较 1 延迟模式 0



■ 2’b10，比较1延迟模式1

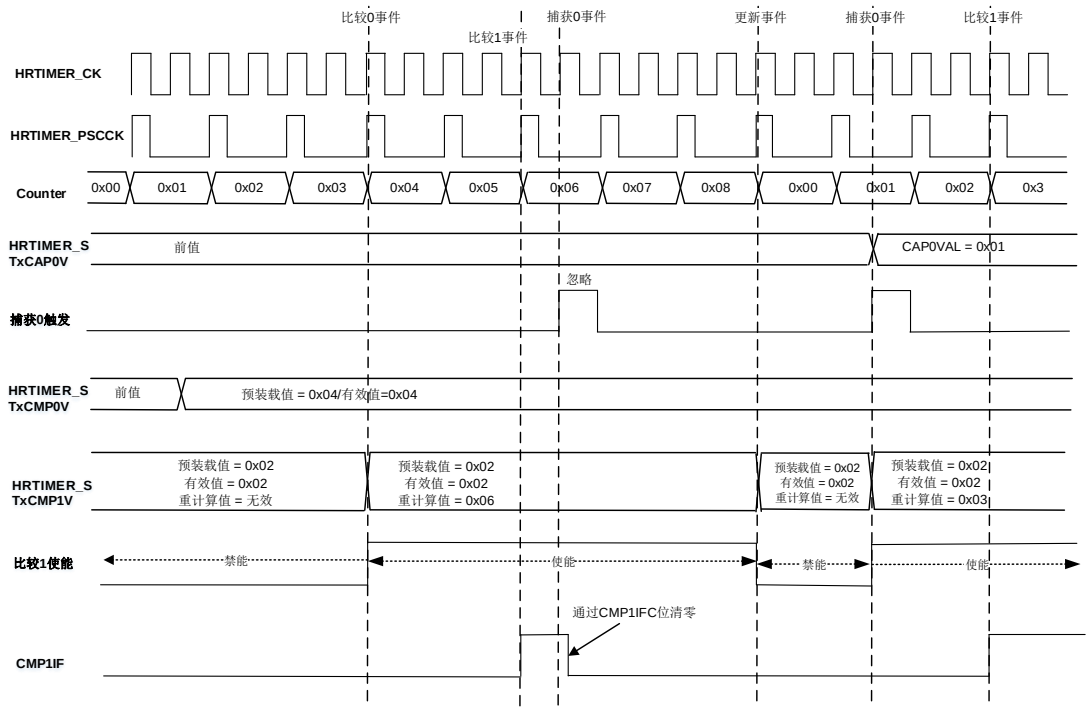
在捕获 0 事件或比较 0 事件之后，将重新计算比较 1 寄存器的值。

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

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

一旦计数器值等于重新计算的比较 1 寄存器值，就会发生比较 1 事件。如果捕获 0 事件先发生，则比较 0 事件将被忽略。同样，如果先发生比较 0 事件，则将忽略捕获 0 事件。详情请见*图 25-15. 比较 1 延迟模式 1*。

图 25-15. 比较 1 延迟模式 1



■ 2'b11，比较1延迟模式2

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

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

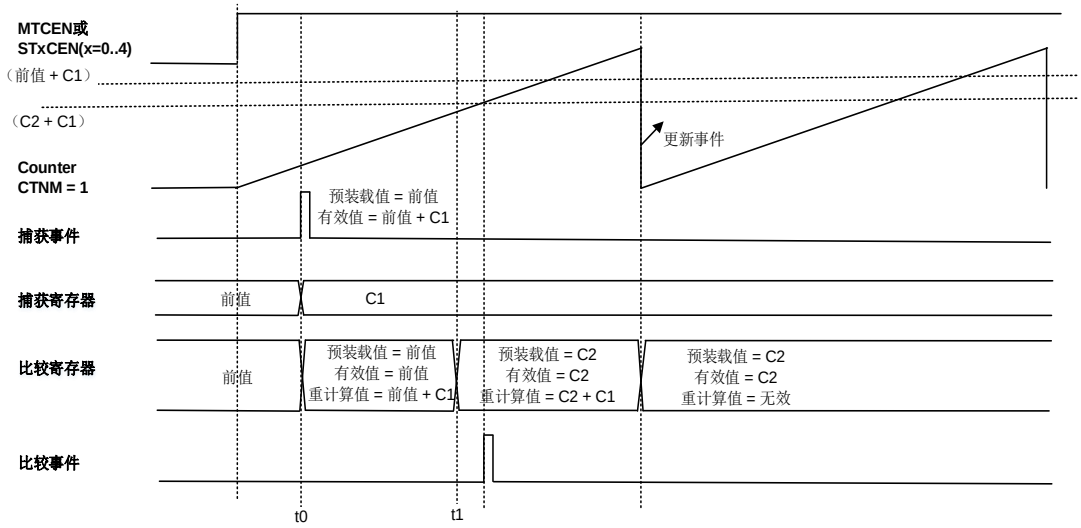
发生比较 2 事件时，比较 1 寄存器的重新计算值 = 比较 1 有效寄存器值 + 比较 2 有效寄存器值。

一旦计数器值等于重新计算的比较 1 寄存器值，就会发生比较 1 事件。如果捕获 0 事件先发生，则比较 2 事件将被忽略。同样，如果先发生比较 2 事件，则将忽略捕获 0 事件。详见*图 25-15. 比较 1 延迟模式 1*。

影子寄存器（SHWEN = 0）禁能时，即使在发生捕获事件后修改 HRTIMER_STxCMP0V 位或 HRTIMER_STxCMP2V 位的值，新的比较值也会立即被带入有效寄存器。*图 25-16.比较延迟模式（SHWEN=0）*显示了一个示例：

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

图 25-16.比较延迟模式（SHWEN=0）



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

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

可变频率半波模式

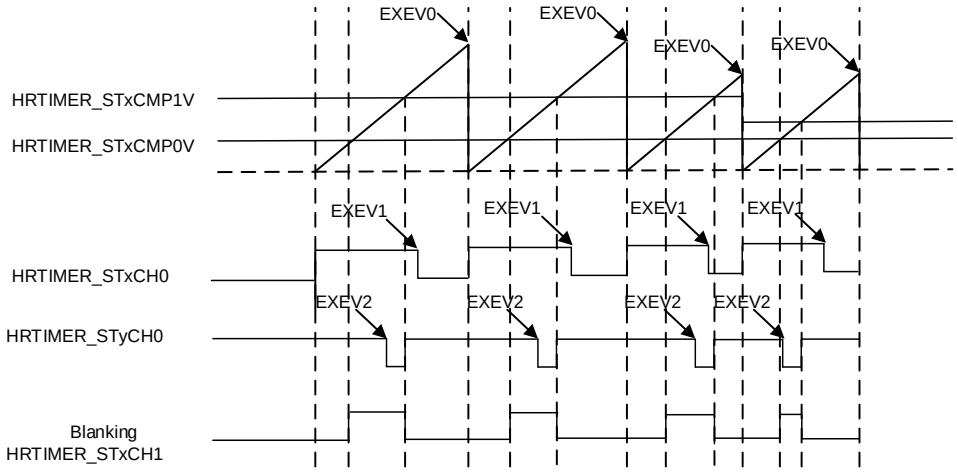
可变频率半波模式是半波模式的补充，该模式允许调节输出信号频率，同时保持 180°相位移。该模式的主要原理是主从模式，从定时器（Slave_TIMERy）通过主定时器（Slave_TIMERx）的捕获事件不断进行调整。

主定时器（Slave_TIMERx）产生捕获事件后，硬件自动将捕获值的一半存储在 HRTIMER_STxCMP1V 中。从定时器（Slave_TIMERy）置位或复位事件可以由主定时器（Slave_TIMERx）的比较 1 事件触发。

通过在 HRTIMER_STxCTL1 中设置 TRGHALFM，可启用可变频率半波模式，当 HRTIMER_MTCTL0 中的 STxCEN 置位时，此位不能更改。

用户可以写入 HRTIMER_STyCMP1V 的初始值，当第一次捕获发生时，初始值将被忽略。并且当 HRTIMER_STxCTL1 中的 TRGHALFM 被复位时，HRTIMER_STyCMP1V 将不会被预加载。

图 25-17.可变频率半波模式



如*图 25-17.可变频率半波模式*所示。

HRTIMER_STxCH0 由外部事件 0（EXEV0）置位，并由外部事件 1（EXEV1）复位，同时外部事件 0 触发产生捕获事件。

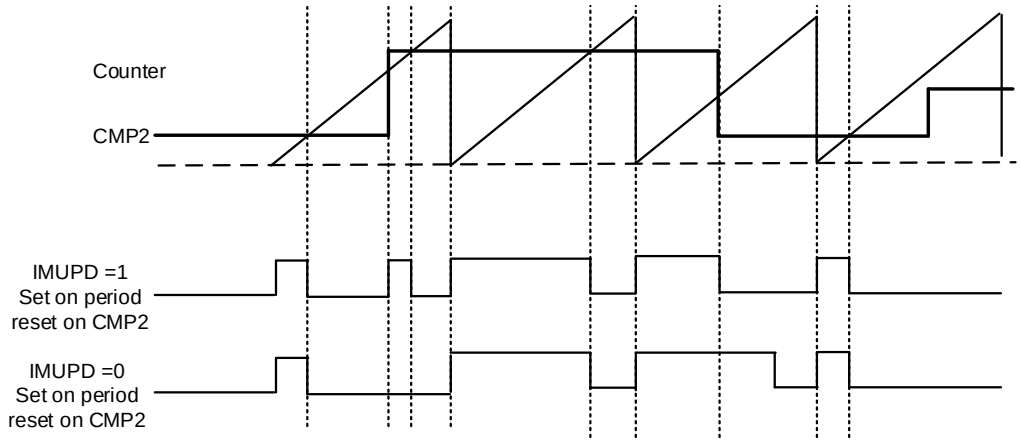
- HRTIMER_STyCH0由HRTIMER_STxCMP1V置位，并由EXEV2复位
- HRTIMER_STxCH1由HRTIMER_STxCMP0V置位，并由HRTIMER_STxCMP1V复位

立即更新模式

HRTIMER 的立即更新模式适用于比较 0 复位事件和比较 2 复位事件，并通过在 HRTIMER_STxCTL1 寄存器中设置 IMUPDxV 位来启用此模式。当启用立即更新模式时，PWM 波形会立即更新，而无需等待当前周期结束。在以下情况下，PWM 波形会立即更改。

- 在运行过程中更改比较值时，如果新的比较值小于计数器值且当前比较值大于计数器值，则会立即复位输出PWM波形。
- 在运行过程中更改比较值时，如果新的比较值大于计数器值且当前比较值大于计数器值，则会立即设置输出PWM波形。
- 当新的比较值和当前值均小于计数器值时，输出PWM波形不会更改。

图 25-18. 当 IMPUD2V = 1 和 IMPUD2V = 0 时输出的 PWM 波形



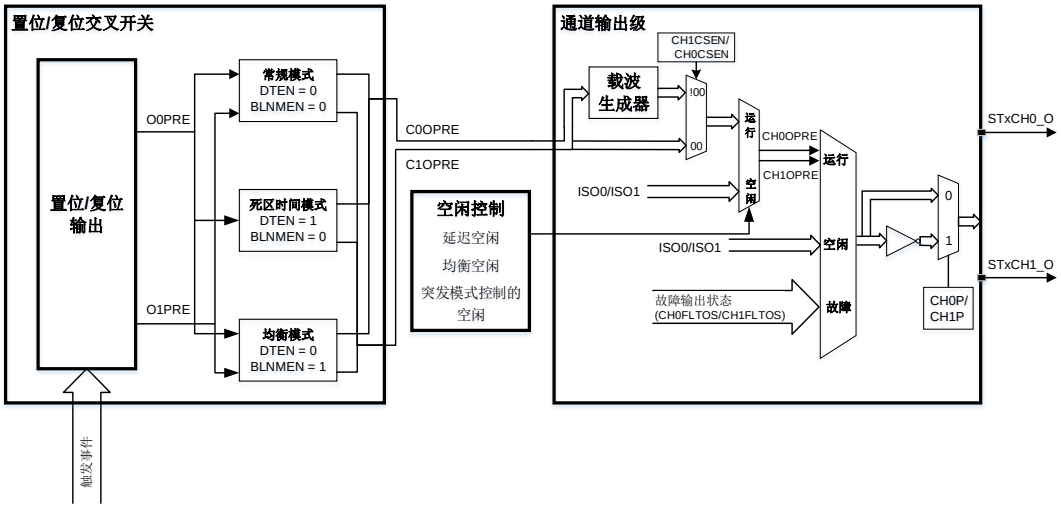
置位/复位交叉开关

通道输出波形功能由三个部分实现：

- 置位/复位交叉开关
- 空闲控制
- 通道输出级

*图 25-19. 通道输出结构图*显示了这三个部分的结构。

图 25-19. 通道输出结构图



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

向上计数模式下输出准备信号

Slave_TIMERx 有一个置位/复位输出模块。该模块可以生成两个输出准备信号：O0PRE 和 O1PRE。其中，O0PRE 由 HRTIMER_STxCH0SET 和 HRTIMER_STxCH0RST 寄存器控制。O1PRE 由 HRTIMER_STxCH1SET 和 HRTIMER_STxCH1RST 寄存器控制。OyPRE（y = 0,1）的高电平为有效电平，低电平为无效电平。

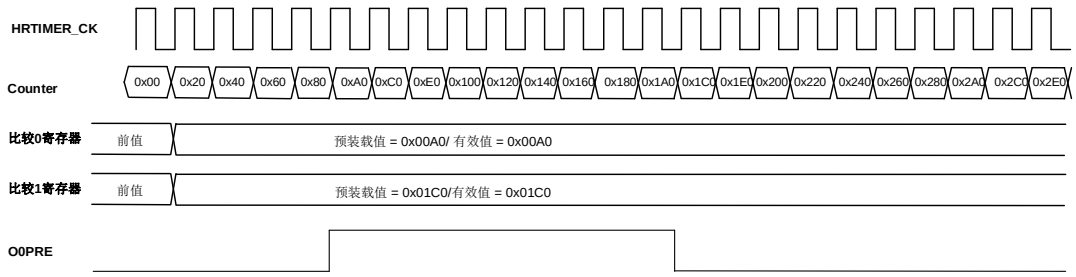
当 HRTIMER_STxCHySET 寄存器中配置的事件发生时，此模块将产生置位请求，并使 OyPRE 输出高电平。当 HRTIMER_STxCHyRST 寄存器中配置的事件发生时，此模块会产生复位请求，并使 OyPRE 输出低电平。如果在 HRTIMER_STxCHySET 和 HRTIMER_STxCHyRST 寄存器中配置了相同的事件，则此模块会生成输出翻转请求，并在配置的事件发生时使 OyPRE 输出翻转。

注意：如果 HRTIMER_STxCTL0 和 HRTIMER_STxACTL 寄存器中的 CNTCKDIV[2:0]位域等于 3'b110 或 3'b111，则不得同时设置 HRTIMER_STxCH0SET 和 HRTIMER_STxCH0RST 寄存器中的同一事件。

*图 25-20. O0PRE 波形：CMP0 事件置位，CMP1 事件复位*显示了以下配置时的 O0PRE 输出波形：

- HRTIMER_STxCTL0寄存器中的CNTCKDIV [2:0] = 3'b000；
- HRTIMER_STxCH0SET= 0x0000 0008：比较0事件产生置位请求，O0PRE将输出高电平；
- HRTIMER_STxCH0RST= 0x0000 0010：比较1事件产生复位请求，O0PRE将输出低电平；
- HRTIMER_STxCMP0V = 0x00A0；
- HRTIMER_STxCMP1V = 0x01C0。

图 25-20. O0PRE 波形：CMP0 事件置位，CMP1 事件复位



OyPRE（y = 0,1）最多可以选择 34 个事件：

- Slave_TIMERx：更新事件，复位事件，周期事件和比较y（y = 0..3）事件；
- Master_TIMER：周期事件，比较y（y = 0..3）事件；
- Slave_TIMERx互连事件：其他Slave_TIMERy有11个互连事件（例如x = 1，然后y = 0，2..7），参见*表25-9. Slave_TIMER内部连接事件*；
- 外部事件y（y = 0..9）：EXEVy为Slave_TIMERx中的外部事件的滤波信号；
- 软件事件。

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

表 25-9. Slave_TIMER 内部连接事件

内部连接		到 ST0	到 ST1	到 ST2	到 ST3	到 ST4	到 ST5	到 ST6	到 ST7
来自 ST0	CMP0	×	0	×	0	×	×	×	0
	CMP1	×	1	0	×	×	×	0	×
	CMP2	×	×	1	×	×	0	×	×
	CMP3	×	×	×	1	0	×	×	×
来自 ST1	CMP0	0	×	×	×	×	1	×	1
	CMP1	1	×	2	2	×	×	×	2
	CMP2	×	×	3	×	1	×	1	×
	CMP3	×	×	×	3	2	2	2	×
来自 ST2	CMP0	×	×	×	×	3	3	3	×
	CMP1	2	×	×	×	4	×	×	×
	CMP2	3	2	×	×	×	×	4	3
	CMP3	×	3	×	4	×	4	×	4
来自 ST3	CMP0	4	×	×	×	5	×	5	×
	CMP1	5	×	4	×	6	×	6	5
	CMP2	×	4	×	×	×	5	×	×
	CMP3	×	5	5	×	×	6	×	6
来自 ST4	CMP0	×	6	×	5	×	×	×	7
	CMP1	×	7	×	×	×	7	7	×
	CMP2	6	×	6	×	×	8	8	×
	CMP3	7	×	7	6	×	×	×	8
来自 ST5	CMP0	×	×	×	7	×	×	×	×
	CMP1	×	×	8	×	×	×	×	9
	CMP2	×	8	×	8	7	×	9	×
	CMP3	8	×	×	×	8	×	×	×
来自 ST6	CMP0	×	×	9	×	×	×	×	×
	CMP1	×	9	×	9	×	×	×	×
	CMP2	9	×	×	×	9	×	×	10
	CMP3	×	×	×	×	×	9	×	×
来自 ST7	CMP0	×	×	×	10	×	×	×	×
	CMP1	×	×	10	×	10	×	10	×
	CMP2	×	10	×	×	×	10	×	×
	CMP3	10	×	×	×	×	×	×	×

注意：（1）表中的数字表示 Slave_TIMERx 互连事件。

（2）“x”代表无效。

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

行仲裁。

中央对齐模式下输出准备信号

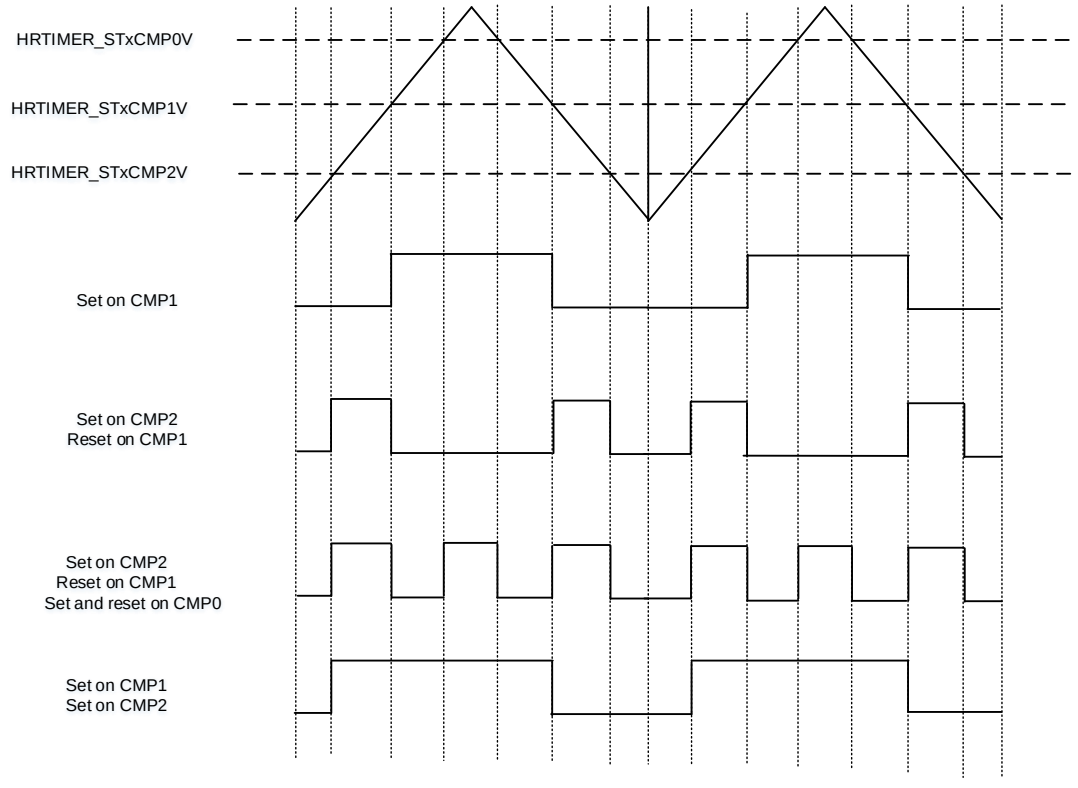
在中央对齐计数模式下，计数器从 0 计数到计数器重载值，然后从该值减少到 0。通过在 HRTIMER_STxCTL1 中设置 CAM 位来启用该模式，中央对齐计数模式仅适用于 Slave_TIMERx（x = 0..7），而不适用于 Master_TIMER。

只有在发生周期事件或复位事件时，Slave_TIMERx 的周期 HRTIMER_STxCAR 的预装载值会被更新。

当在 HRTIMER_STxCHySET 寄存器中配置的事件发生时，此模块产生一个置位请求，并使 OyPRE 在上升计数期间保持高电平，在下降计数期间保持低电平。当在 HRTIMER_STxCHyRST 中配置的事件发生时，此模块产生一个复位请求，并使 OyPRE 在上升计数期间保持低电平，在下降计数期间保持高电平。如果将相同的事件配置在 HRTIMER_STxCHySET 和 HRTIMER_STxCHyRST 寄存器中，此模块将产生一个翻转请求，并在配置的事件发生时使 OyPRE 输出翻转。

有多达 34 个事件可用于选择 OyPRE(y=0,1)，参考*向上计数模式下输出准备信号*。

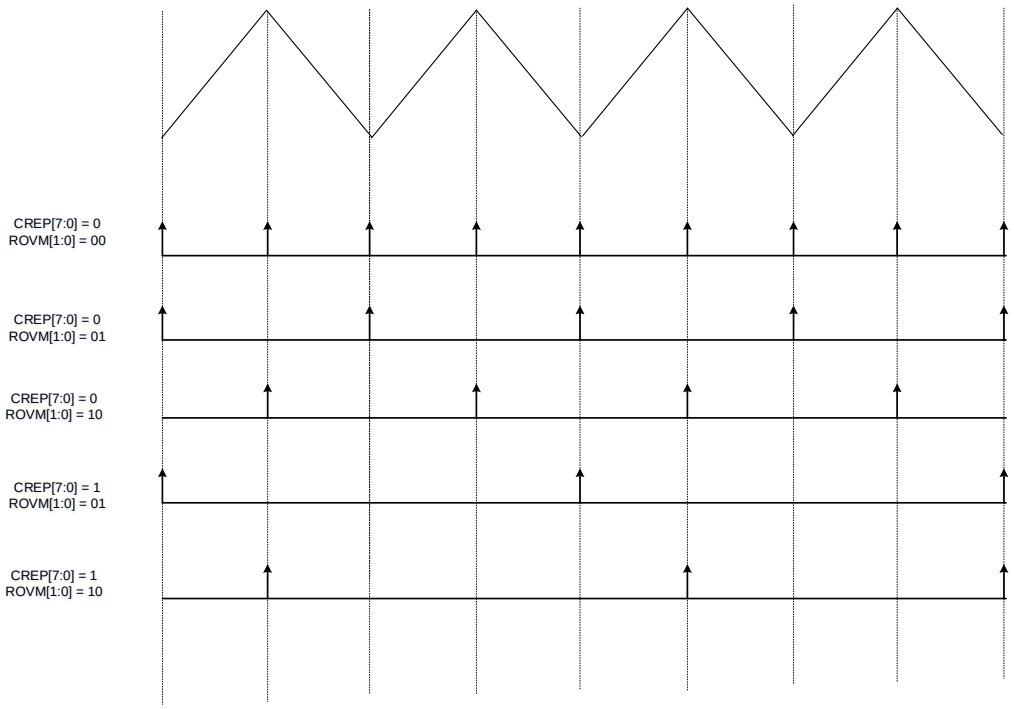
图 25-21. 中央对齐模式下 OyPRE



中央对齐模式可以在半波模式、死区模式、均衡模式、延迟空闲模式、突发模式和立即更新模式中使用。

HRTIMER_STxCAPyV（x=0..7，y=0..3）的捕获值在向上计数时参考起始值，HRTIMER_STxCAPyV = HRTIMER_STxCNT，HRTIMER_STxCAPyV（x=0..7，y=0..3）的捕获值在向下计数时参考自动重载值，HRTIMER_STxCAPyV = HRTIMER_STxCAR - HRTIMER_STxCNT。HRTIMER_STxCAPyV 寄存器中的 DIR 显示计数方向。用于置位或复位 OxPRE 的周期事件在 HRTIMER_STxCTL1 寄存器中由 ROVM[1:0]位定义。当 ROVM[1:0] = 2b00 时，周期事件在计数器等于 0 或等于 HRTIMER_STxCAR 值时生成；当 ROVM[1:0] = 2b01 时，周期事件在计数器等于 0 时生成；当 ROVM[1:0] = 2b10 时，周期事件在计数器等于 HRTIMER_STxCAR 时生成。

图 25-22. 中央对齐模式下重复计数器值 CREP[7:0]和 ROVM[1:0]



HRTIMER 具有消隐模式和窗口模式，在向上计数模式和中央对齐模式中，消隐时间和窗口时间不同。

表 25-10. 消隐模式和窗口模式在向上计数模式和中央对齐模式

EXEV0FM[4:0]	向上计数模式	中央对齐模式
00010	消隐从周期事件到比较 1 事件	消隐从比较0事件到比较1事件，仅在中央对齐模式向上计数有效
00100	消隐从周期事件到比较 3 事件	消隐从比较2事件到比较1事件，仅在中央对齐模式向上计数有效
01101	窗口从周期事件到比较 1 事件	窗口从比较1事件到比较2事件，仅在中央对齐模式向上计数有效
01110	窗口从周期事件到比较 2 事件	窗口从比较1事件到比较2事件，仅在中央对齐模式向下计数有效

EXEV0FM[4:0]	向上计数模式	中央对齐模式
01111	窗口从周期事件来自其他定时器	窗口从比较1事件（向上计数）到比较2事件（向下计数），仅在中央对齐模式有效

仲裁机制

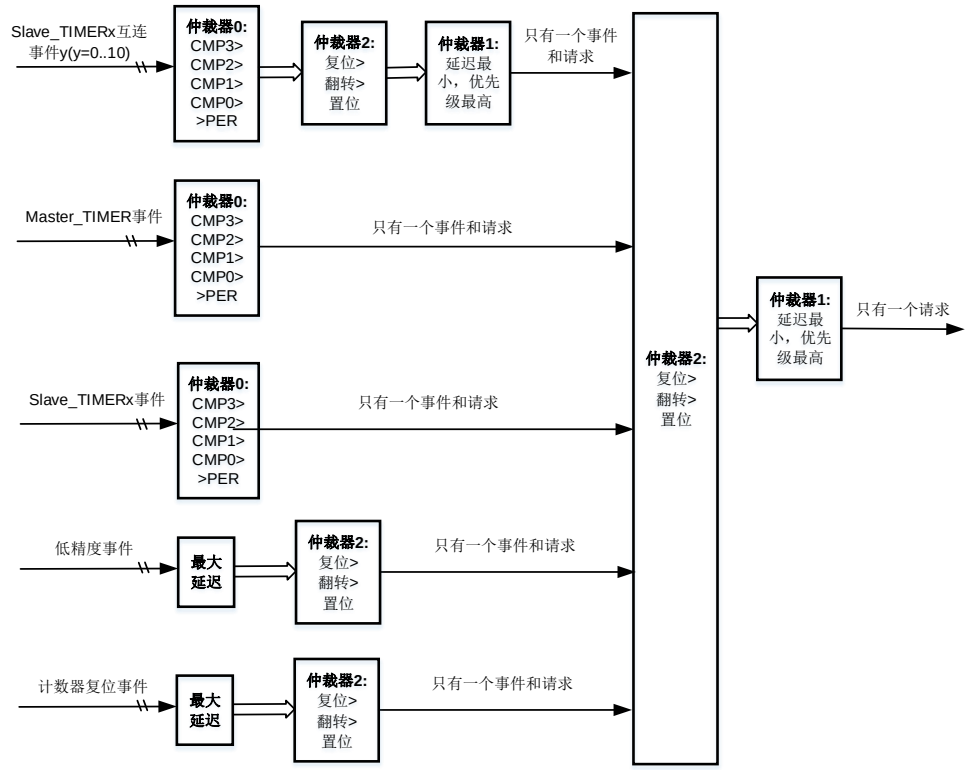
当 HRTIMER_STxCH1SET 和 HRTIMER_STxCH1RST 寄存器中配置的多个事件发生在同一个 t_{HPTMER_CK} 周期内时，将执行仲裁过程，且只有一个事件有效，可以更改 OyPRE（y = 0,1）的输出。

这 35 个事件可以分为五种类型：

- Slave_TIMERx：比较y（y = 0..3）事件，周期事件。
- Master_TIMER：比较y（y = 0..3）事件，周期事件。
- Slave_TIMERx互连事件：互连事件y（y = 0..10）
- 低精度事件：Slave_TIMERx的更新事件和复位事件，外部事件y（y = 0..9），软件事件。
- 计数器复位事件：使用最大延时。

具体的仲裁过程*图 25-23. 每个 tHRTMER_CK 周期的仲裁机制过程*所示。

图 25-23. 每个 t_{HRTMER_CK} 周期的仲裁机制过程



三个仲裁器的功能如下：

- 仲裁器 0 的优先级顺序（从最高优先级到最低优先级）：

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

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

延迟越小，优先级越高。

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

复位请求 > 输出翻转请求 > 置位请求。

以 Slave_TIMER0 中的 O0PRE 输出为例，配置如下：

- HRTIMER_STxCH0SET = 0x0060 5898，选定产生置位请求的事件是：

Master_TIMER：比较 2 事件，周期事件；

Slave_TIMER0：比较 1 事件，比较 0 事件；

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

低精度事件：外部事件 0（EXEV0），外部事件 1（EXEV1）。

计数器复位事件。

- HRTIMER_STxCH0RST = 0x0198 0344，选定产生复位请求的事件是：

Master_TIMER：比较 0 事件，比较 1 事件；

Slave_TIMER0：比较 3 事件，周期事件；

Slave_TIMER0 的互连事件：互连事件 7（Slave_TIMER4 比较 2 事件），互连事件 8（Slave_TIMER4 比较 3 事件）；

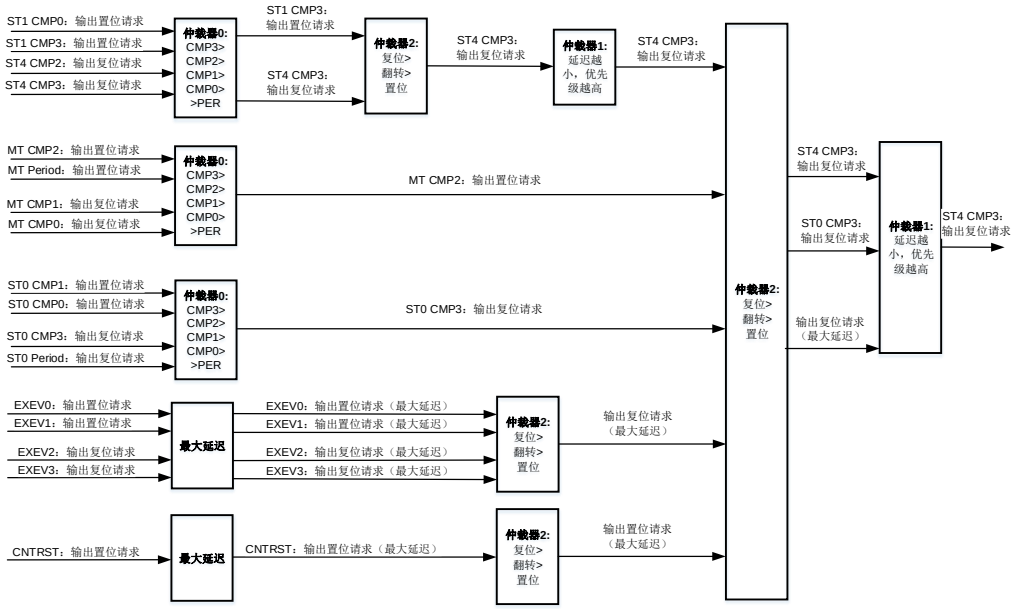
低精度事件：外部事件 2（EXEV2），外部事件 3（EXEV3）。

- 延迟：Slave_TIMER4 比较 3 < Slave_TIMER0 比较 3

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

图 25-24. 仲裁机制示例所示。最终，Slave_TIMER4 比较 3 事件产生的复位请求在该 t_{HRTMER_CK} 期间有效，并且 O0PRE 将被设置为低电平。

图 25-24. 仲裁机制示例



输出准备信号：窄脉冲管理

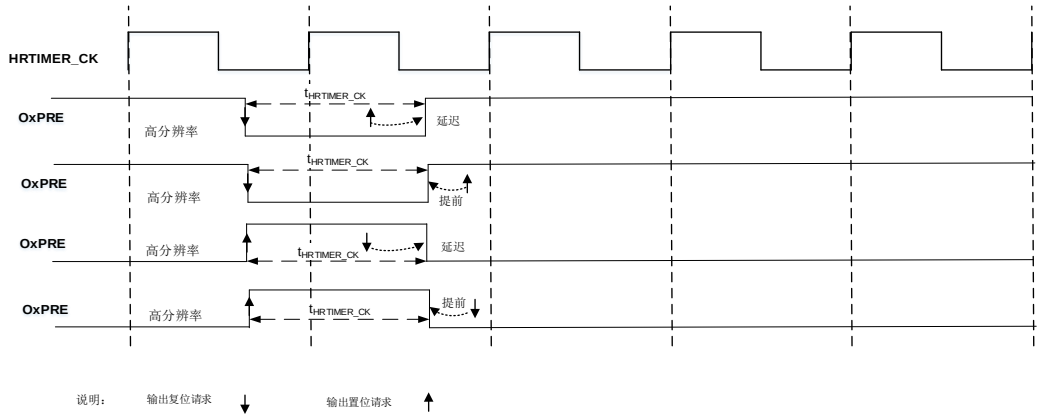
当几个输出置位和/或复位请求在 3 个连续的 t_{HRTMER_CK} 周期内发生时，OyPRE（y = 0,1）输出信号是一个窄脉冲。窄脉冲的输出管理由 HRTIMER_STxCTL0 寄存器中的 CNTCKDIV[2:0]位域配置，有以下两种情形：

- 情形0：CNTCKDIV[2:0] < 3'b101
- 情形1：CNTCKDIV[2:0] >= 3'b101

情形 0：CNTCKDIV[2:0] < 3'b101

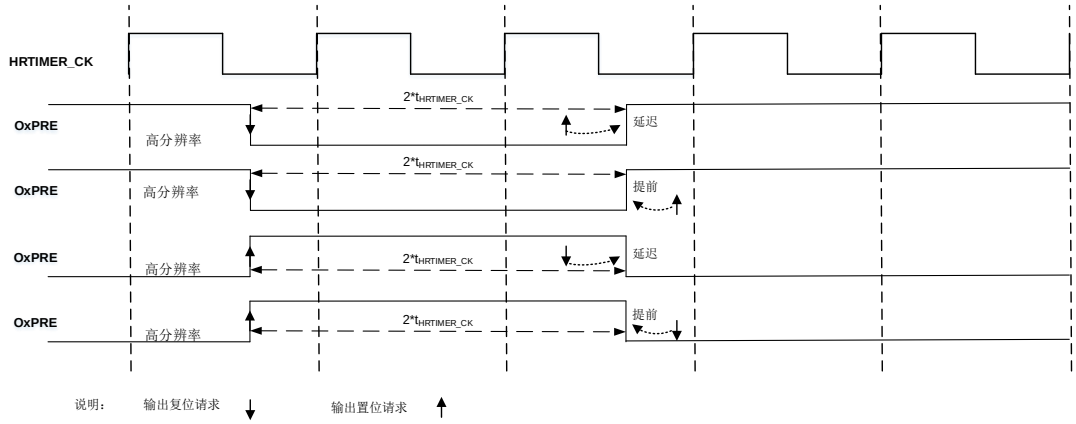
如果输出置位和复位请求在两个连续的 t_{HRTMER_CK} 周期内产生，则会生成脉宽为 1 个 t_{HPTMER_CK} 周期的脉冲。具体如*图 25-25. 脉冲宽度为 1 个 tHRTMER_CK 周期*所示。

图 25-25. 脉冲宽度为 1 个 t_{HRTMER_CK} 周期



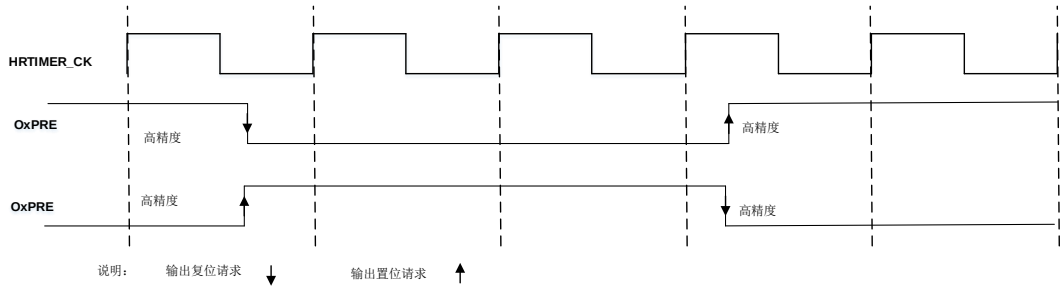
如果输出置位和复位请求的时间间隔包括一个完整的 t_{HRTMER_CK} 周期，则会生成脉宽为 2 个 t_{HRTMER_CK} 周期的脉冲。具体如*图 25-26.脉冲宽度为 2 个 tHRTMER_CK 周期*所示。

图 25-26.脉冲宽度为 2 个 t_{HRTMER_CK} 周期



如果输出置位和复位请求的时间间隔大于两个完整的 t_{HPTMER_CK} 周期，则需要使用超高分辨率时钟。具体如*图 25-27. 高分辨率 OxPRE 波形*所示。

图 25-27. 高分辨率 OxPRE 波形

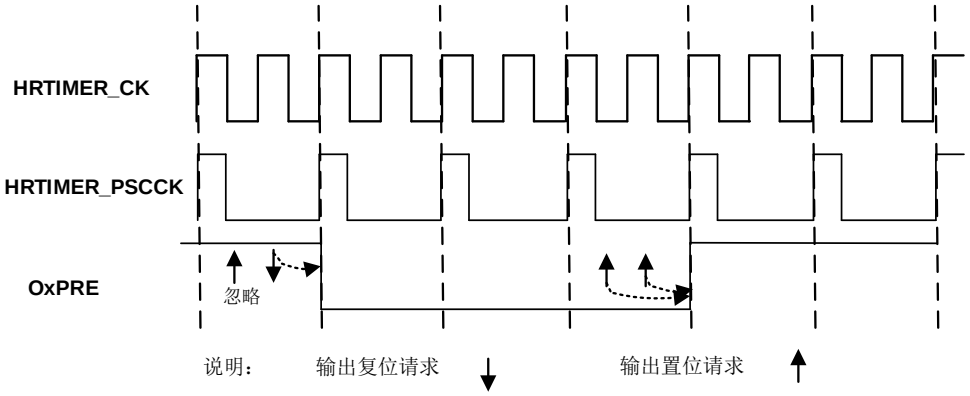


情形 1: CNTCKDIV[2:0] >= 3'b101

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

当来自不同事件源的置位请求和复位请求在 1 个 $t_{HRTIMER_CK}$ 周期中同时发生时，复位请求具有最高优先级，详见*图 25-28. CNTCKDIV[2:0] = 3'b110 时的 OxPRE 波形*。

图 25-28. CNTCKDIV[2:0] = 3'b110 时的 OxPRE 波形



常规模式

当 HRTIMER_STxCHOCTL 寄存器中的 DTEN = 0，HRTIMER_STxCTL0 寄存器中的 BLNMEN = 0 时，置位/复位交叉开关以常规模式运行。

该模式中，C0OPRE 和 C1OPRE 是独立的。C0OPRE（C1OPRE）直接连接到 O0PRE（O1PRE）。

当 Slave_TIMERx（x = 0..7）运行在 RUN 或 IDLE 状态下，C0OPRE 从无效状态变为有效状态时，HRTIMER_STxINTF 寄存器中的 CH0OAIF 位将被设置为 1。如果使能了中断或 DMA 请求（HRTIMER_STxDMAINTEN 寄存器中的 CH0OAIE = 1 或 CH0OADEN = 1），则产生相应的中断或 DMA 请求。通过将 1 写到 HRTIMER_STxINTFC 中的 CH0OAIFC 位可以清除 CH0OAIF 中断标志。

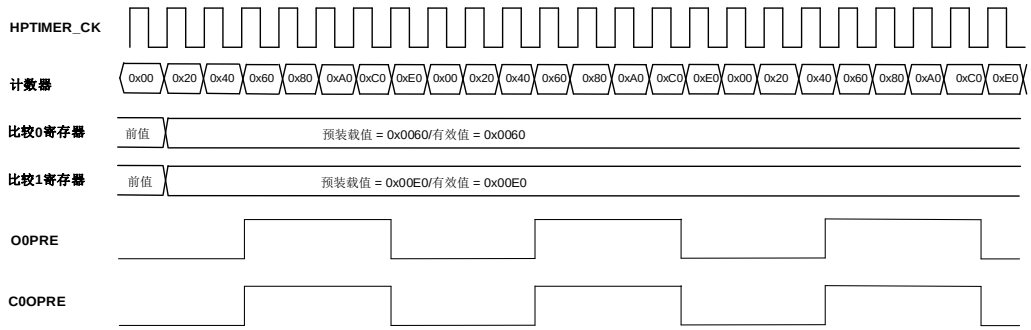
当 Slave_TIMERx（x = 0..7）运行在 RUN 或 IDLE 状态下，C0OPRE 从有效状态变为无效状态时，HRTIMER_STxINTF 寄存器中的 CH0ONAIF 位将置 1，如果使能了中断或 DMA 请求（HRTIMER_STxDMAINTEN 寄存器中的 CH0ONAIE = 1 或 CH0ONADEN = 1），则产生相应的中断或 DMA 请求。通过将 1 写到 HRTIMER_STxINTFC 中的 CH0ONAIFC 位可以清除 CH0ONAIF 中断标志。

通道 1 与通道 0 输出情况相同。

*图 25-29. 常规模式下的 C0OPRE 波形*显示了以下配置时的 C0OPRE 波形：

- HRTIMER_STxCTL0寄存器中的CNTCKDIV[2:0] = 3'b000；
- HRTIMER_STxCH0SET = 0x0000 0008：比较0事件产生置位请求，O0PRE输出高电平；
- HRTIMER_STxCH0RST = 0x0000 0010：比较1事件产生复位请求，O0PRE输出低电平；
- HRTIMER_STxCMP0V = 0x0060；
- HRTIMER_STxCMP1V = 0x00E0。

图 25-29. 常规模式下的 C0OPRE 波形



死区时间模式

当 HRTIMER_STxCHOCTL 寄存器中的 DTEN = 1，HRTIMER_STxCTL0 寄存器中的 BLNMEN = 0 时，置位/复位交叉开关在死区模式下运行。

死区模式中，只对 O0PRE 进行编程，以驱动 C0OPRE 和 C1OPRE 的输出。C0OPRE 和 C1OPRE 是一对互补信号，在有效状态转换之间插入可编程的死区时间。

死区时间值是由 HRTIMER_STxDTCTL 寄存器中的 DTFCFG [15:0]位域和 DTRCFG [15:0]位域确定的。DTFCFG [15:0]位域定义在 O0PRE 下降沿之后的死区时间，而 DTRCFG [15:0]位域定义在 O0PRE 上升沿之后的死区时间。

注意：DTFCFG [8:0]和 DTRCFG [8:0]位域在 HRTIMER_STxDTCTL 寄存器中，DTFCFG [15:9]和 DTRCFG [15:9]位域在 HRTIMER_STxACTL 寄存器中

死区时间值可以由 HRTIMER_STxDTCTL 寄存器中的 DTRS 位和 DTFS 位配置为正或负。当需要某些波重叠时，可以定义负的死区时间值。

死区时间由 HRTIMER_STxDTCTL 寄存器中的 DTGCKDIV [3:0]位域定义的时钟确定。

注意：DTGCKDIV[3:0]位域在 HRTIMER_STxDTCTL 寄存器中。

当 Slave_TIMERx（x = 0..7）运行在 RUN 或 IDLE 状态下，C0OPRE 从无效状态变为有效状态时，HRTIMER_STxINTF 寄存器中的 CH0OAIF 位将被设置为 1。如果使能了中断或 DMA 请求（HRTIMER_STxDMAINTEN 寄存器中的 CH0OAIE = 1 或 CH0OADEN = 1），则产生相应的中断或 DMA 请求。通过将 1 写到 HRTIMER_STxINTFC 中的 CH0OAIFC 位可以清除 CH0OAIF 中断标志。

当 Slave_TIMERx（x = 0..7）运行在 RUN 或 IDLE 状态下，C0OPRE 从有效状态变为无效状态时，HRTIMER_STxINTF 寄存器中的 CH0ONAIF 位将置 1，如果使能了中断或 DMA 请求（HRTIMER_STxDMAINTEN 寄存器中的 CH0ONAIE = 1 或 CH0ONADEN = 1），则产生相应的中断或 DMA 请求。通过将 1 写到 HRTIMER_STxINTFC 中的 CH0ONAIFC 位可以清除 CH0ONAIF 中断标志。

通道 1 与通道 0 输出情况相同。

*图 25-30. 具有死区时间的 C0OPRE 和 C1OPRE 互补输出波形*显示了 O0PRE 脉冲宽度大于死区时间时的 C0OPRE 和 C1OPRE 波形。

图 25-30. 具有死区时间的 C0OPRE 和 C1OPRE 互补输出波形

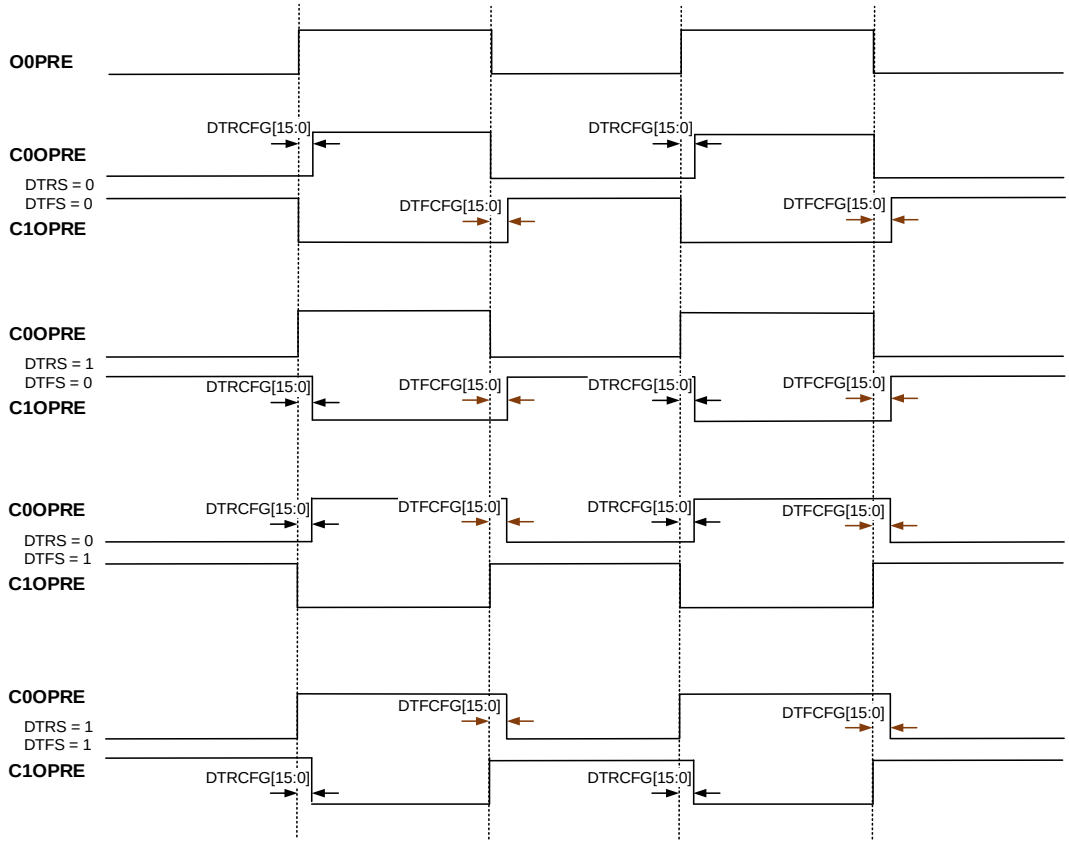
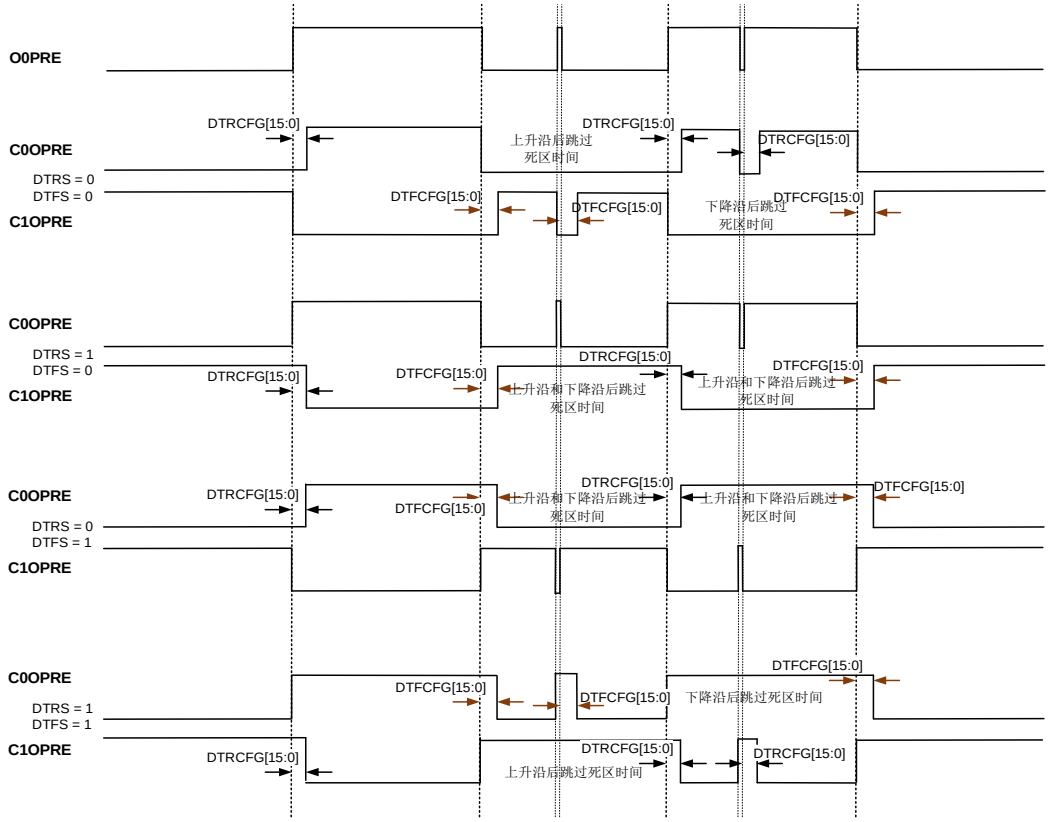


*图 25-31. 脉宽小于死区时间时的互补波形*显示了 O0PRE 脉冲宽度小于死区时间时的 C0OPRE 和 C1OPRE 互补波形。

图 25-31. 脉宽小于死区时间时的互补波形

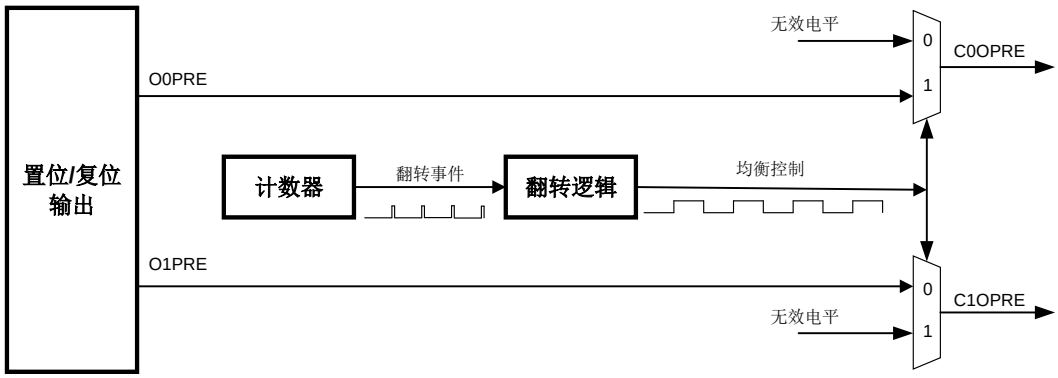


均衡模式

当 HRTIMER_STxCHOCTL 寄存器中的 DTEN = 0，HRTIMER_STxCTL0 寄存器中的 BLNMEN = 1 时，置位/复位交叉开关在均衡模式下运行。当计数器在连续模式下运行时才能使用均衡模式，并且一旦使能了计数器就不得使其复位。

*图 25-32. 均衡模式的结构框图*显示了均衡模式的信号控制过程。

图 25-32. 均衡模式的结构框图



一旦接收到翻转事件，翻转逻辑模块的输出就会翻转。当翻转逻辑模块的输出为 1（高电平）时，

C0OPRE 连接到 O0PRE，C1OPRE 为无效电平（低电平）。当翻转逻辑模块的输出为 0（低电平）时，C1OPRE 连接到 O1PRE，C0OPRE 为无效电平（低电平）。

建议配置 HRTIMER_STxCH0SET = HRTIMER_STxCH1SET，HRTIMER_STxCH0RST = HRTIMER_STxCH1RST，实现相同波形的均衡操作。在进行其他应用时，也可以对两个输出进行不同的配置。

均衡模式禁能时，HRTIMER_STxINTF 寄存器中的 CBLNF 位将复位，该位用于指示当前哪个通道正在输出信号（O0PRE 或 O1PRE）。

当 Slave_TIMERx（x = 0..7）运行在 RUN 或 IDLE 状态下，C0OPRE 从无效状态变为有效状态时，HRTIMER_STxINTF 寄存器中的 CH0OAIF 位将被设置为 1。如果使能了中断或 DMA 请求（HRTIMER_STxDMAINTEN 寄存器中的 CH0OAIE = 1 或 CH0OADEN = 1），则产生相应的中断或 DMA 请求。通过将 1 写到 HRTIMER_STxINTFC 中的 CH0OAIFC 位可以清除 CH0OAIF 中断标志。

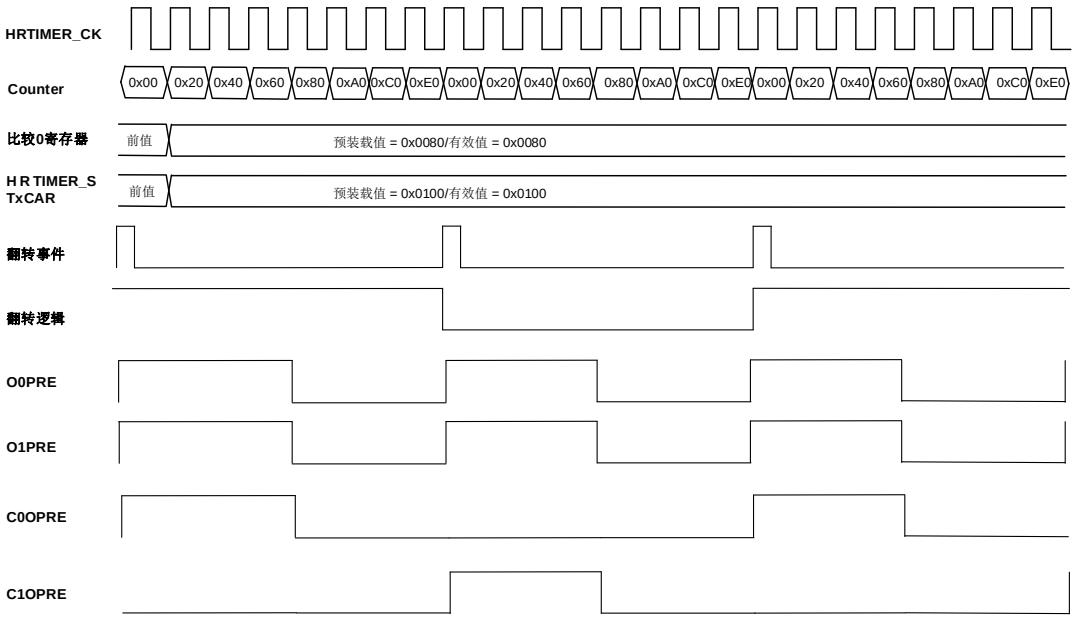
当 Slave_TIMERx（x = 0..7）运行在 RUN 或 IDLE 状态下，C0OPRE 从有效状态变为无效状态时，HRTIMER_STxINTF 寄存器中的 CH0ONAIF 位将置 1，如果使能了中断或 DMA 请求（HRTIMER_STxDMAINTEN 寄存器中的 CH0ONAIE = 1 或 CH0ONADEN = 1），则产生相应的中断或 DMA 请求。通过将 1 写到 HRTIMER_STxINTFC 中的 CH0ONAIFC 位可以清除 CH0ONAIF 中断标志。

通道 1 与通道 0 输出情况相同。

*图 25-33. 均衡模式下的 C0OPRE 和 C1OPRE 波形*显示了以下配置时的 C0OPRE 和 C1OPRE 波形：

- HRTIMER_STxCH0SET = HRTIMER_STxCH1SET = 0x0000 0004：周期事件产生置位请求，O0PRE和O1PRE输出高电平；
- HRTIMER_STxCH0RST = HRTIMER_STxCH1RST = 0x0000 0008：比较0事件产生复位请求，O0PRE和O1PRE输出低电平。

图 25-33. 均衡模式下的 C0OPRE 和 C1OPRE 波形



空闲控制

空闲控制级有三种控制空闲状态的方式：

- 延迟空闲模式
- 均衡空闲模式
- 突发模式控制的空闲模式

延迟空闲和均衡空闲模式不能同时使用。均衡空闲仅在均衡模式下可用。延迟空闲或均衡空闲可以与突发模式控制的空闲模式同时使用，但突发模式的优先级最低。当置位/复位开关在不同模式下操作时，可以使用不同的空闲控制模式，详见*表 25-11. 交叉开关和空闲控制同时运行*。

表 25-11. 交叉开关和空闲控制同时运行

置位/复位交叉开关运行模式	IDLE 控制级运行模式
常规模式	延迟空闲，突发模式控制的空闲
死区时间模式	延迟空闲，突发模式控制的空闲
均衡模式	延迟空闲，均衡控制和突发模式控制空闲

HRTIMER_STxINTF 寄存器中的 CHyF(y=0,1)位，指示了 CHyOPRE 的输出电平。

延迟空闲

在延迟空闲模式，所选外部事件（对于 Slave_TIMER0/ 1/ 2 为 EXEV5/ 6，对于 Slave_TIMER3/4/5/6/7 为 EXEV7/ 8）之后的置位请求或复位请求会导致 CHyOPRE（y = 0，1）输出进入空闲状态。具体情况与 HRTIMER_STxCHOCTL 寄存器中的 ISOy/ CHyP（y = 0,1）有

关，详见*表 25-12. 进入和退出空闲状态的请求*。ISOy 位用于定义空闲状态时 CHyOPRE 的输出电平。空闲模式会永久保持，计数器将继续运行，直到重新使能输出才退出该模式。将 STxCH0EN 和 STxCH1EN 位重新置 1 后，通过置位请求或复位请求可以重新使能延迟空闲模式。

表 25-12. 进入和退出空闲状态的请求

ISOy/CHyP (y=0,1)值	进入空闲状态的请求	退出空闲状态的请求
ISOy = 0 CHyP = 0	复位请求	置位请求和复位请求
ISOy = 1 CHyP = 0	置位请求	置位请求和复位请求
ISOy = 0 CHyP = 1	置位请求	置位请求和复位请求
ISOy = 1 CHyP = 1	复位请求	置位请求和复位请求

延迟空闲模式可以应用于单个输出（CHyOPRE）或两个输出（CH0OPRE 和 CH1OPRE）情况（由 HRTIMER_STxCHOCTL 寄存器中的 DLYISCH[2:0]位域定义）：

- DLYISCH[2:0] = 3'b000：延迟空闲模式应用于CH0OPRE；
- DLYISCH[2:0] = 3'b001：延迟空闲模式应用于CH1OPRE；
- DLYISCH[2:0] = 3'b010：延迟空闲模式应用于CH0OPRE和CH1OPRE。

一旦选定的外部事件（EXEV5/ 6 或 EXEV7/ 8）到达，HRTIMER_STxINTF 寄存器中的 DLYIIF 位置 1，如果使能了相应的中断和 DMA（HRTIMER_STxDMAINTEN 寄存器中的 DLYIIE = 1 或 DLYIDEN = 1），则产生中断或 DMA 请求。通过写 1 到 HRTIMER_STxINTFC 寄存器中的 DLYIIFC 位可以清除中断标志。

当选定的外部事件（EXEV5 / 6 或 EXEV7/ 8）触发延迟空闲模式时，HRTIMER_STxINTF 寄存器中的 CHyDLYF（y = 0,1）位可以指示 CHyOPRE 信号的状态。

下面四张图显示了延迟空闲模式中的 CH0OPRE 波形：

- C0OPRE 运行在常规模式：HRTIMER_STxCHOCTL 寄存器中的 DTEN = 0，HRTIMER_STxCTL0寄存器中的BLNMEN = 0；
- 比较0事件产生置位请求；
- 比较1事件产生复位请求。

图 25-34. 延迟空闲模式，ISO0 = 0 和 CHOP = 0

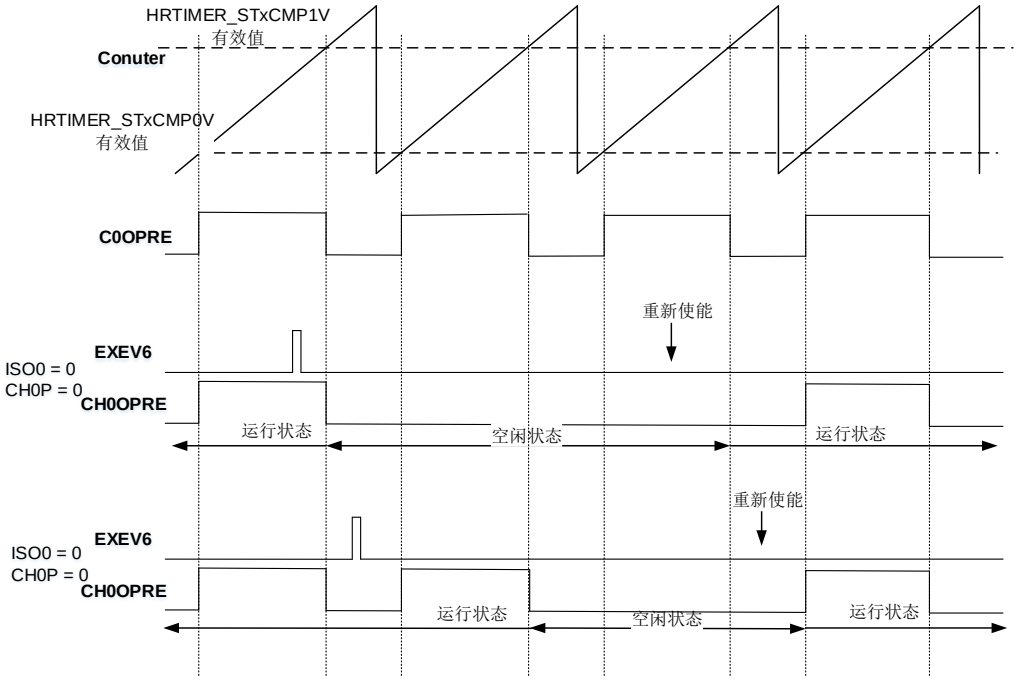


图 25-35. 延迟空闲模式，ISO0 = 1 和 CHOP = 0

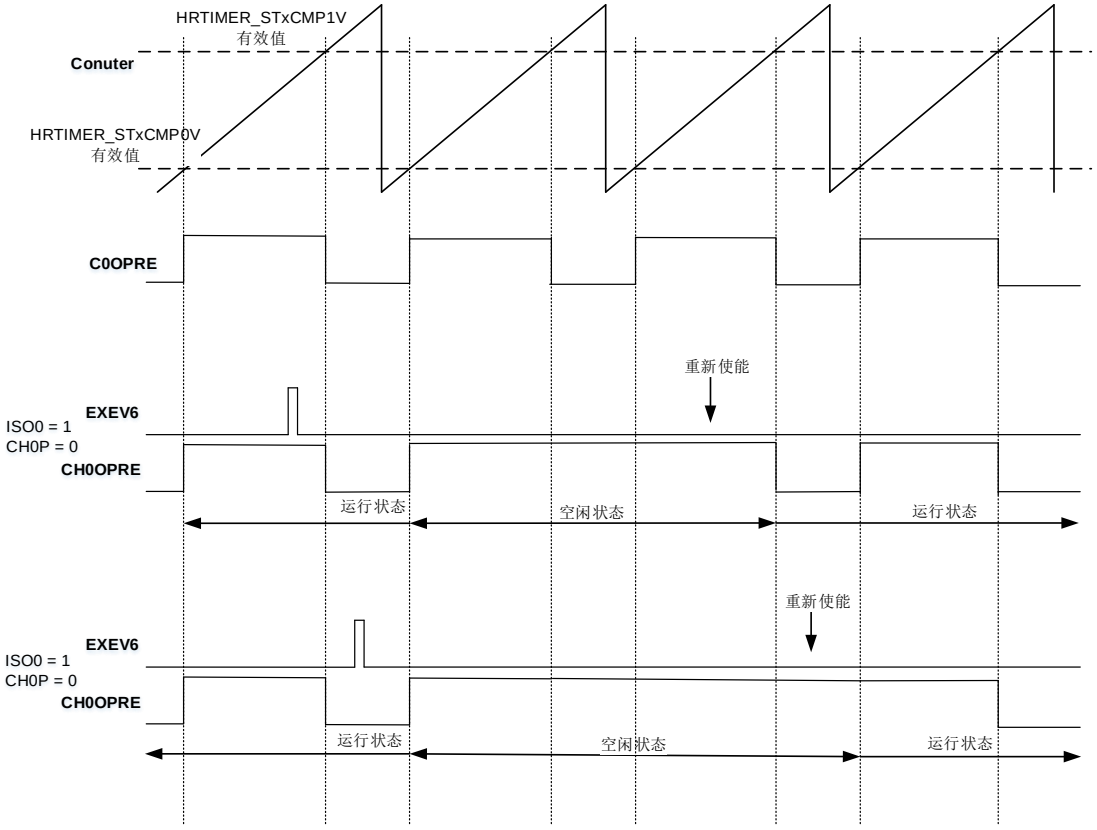


图 25-36. 延迟空闲模式，ISO0 = 0 和 CHOP = 1

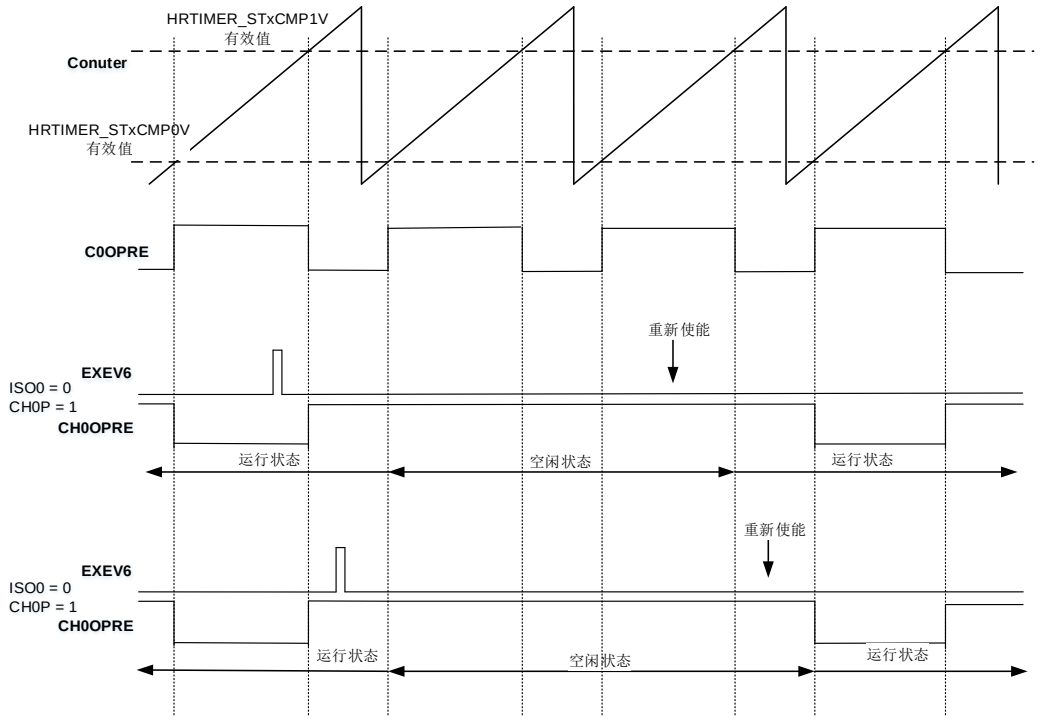
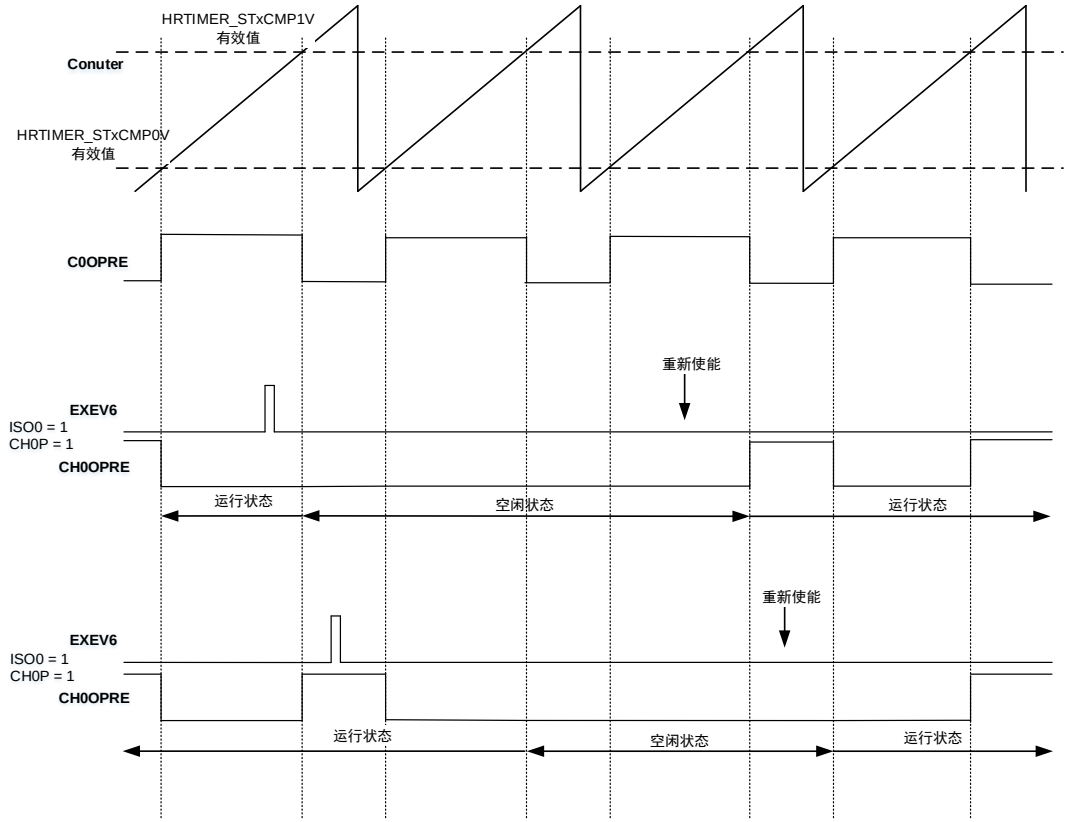


图 25-37. 延迟空闲模式，ISO0 = 1 和 CHOP = 1



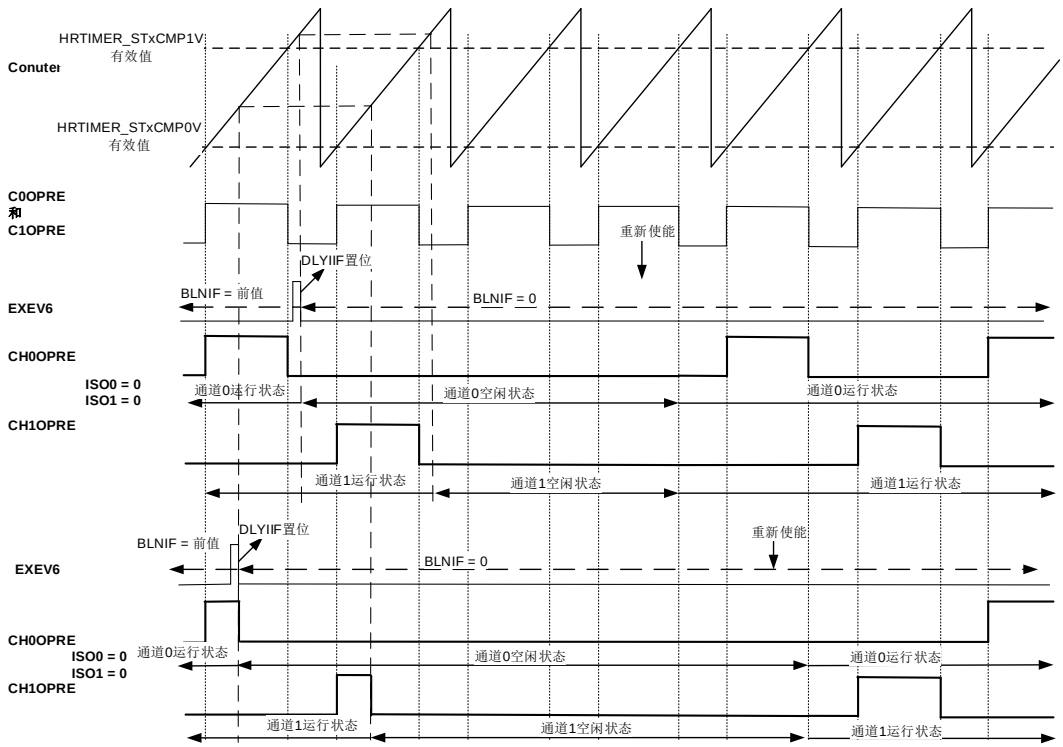
均衡空闲

均衡空闲模式仅在均衡模式下可用。通过将 HRTIMER_STxCHOCTL 寄存器的 DLYISCH[2:0]位域设为 3'bx011，使能均衡空闲模式。Slave_TIMER0/1/2 的外部事件 5/6（EXEV5/6）和 Slave_TIMER3/4/5/6/7 的外部事件 7/8（EXEV7/8）可使用均衡空闲模式。

外部事件发生时，CHyOPRE（y = 0,1）进入空闲状态，并输出由 HRTIMER_STxCHOCTL 寄存器中的 ISOy 位定义的电平，且 HRTIMER_STxINTF 寄存器中的 DLYIIF 位置 1。该外部事件触发捕获，将计数器值捕获到比较 3 有效寄存器中（该值用户不可访问）。均衡模式会再维持一个周期，使互补输出 CHzOPRE（z = 0，1 且 z≠y）可以重复 CHyOPRE 上的短脉冲：*图 25-38. 均衡空闲模式，ISO0 = 0 和 ISO1 = 0* 显示了均衡空闲模式下 Slave_TIMER0 的 CH0OPRE/ CH1OPRE 波形，配置如下：

- C0OPRE 处于均衡模式：HRTIMER_STxCHOCTL 寄存器中的 DTEN = 0，HRTIMER_STxCTL0寄存器中的BLNMEN = 1；
- 比较0事件产生置位请求；
- 比较1事件产生复位请求；
- 在外部事件6发生时，通道0和通道1输出为均衡空闲模式：Slave_TIMER0的DLYISCH [2：0] = 3'b111。

图 25-38. 均衡空闲模式，ISO0 = 0 和 ISO1 = 0



HRTIMER_STxINTF 寄存器中的 BLNIF 位指示了发生均衡空闲模式时哪个通道正在输出信号。如*图 25-38. 均衡空闲模式，ISO0 = 0 和 ISO1 = 0*，外部事件 6（EXEV6）到来时，通道 0 输出信

号，通道 1 输出无效，且 BLNIF 位复位为零。

在计数器继续运行时，IDLE 模式将永久保持，直到重新使能输出才退出均衡空闲模式。将 STxCH0EN 和 STxCH1EN 位同时重新置 1 后，通过置位请求或复位请求可以重新使能均衡空闲模式。

在以下情况下，均衡空闲模式可以与突发模式一起使用：

- BMSTx位必须复位（保持计数器时钟HRTIMER_PSCCK，且计数器正常运行）；
- 当输出配置为突发模式控制的空闲状态时，不会触发均衡空闲模式。

均衡空闲自动恢复模式

C0OPRE 和 C1OPRE 在设置 BALIAR 时可以自动从均衡空闲中恢复，并且在 DLYISCH[2:0] = 0x011 或 0x111 时可以使用均衡空闲自动恢复模式，该位位于 HRTIMER_STxCHOCTL 寄存器中。当窄脉冲已加载到输出寄存器时，C0OPRE 和 C1OPRE 的脉冲将恢复为正常输出。

均衡空闲自动恢复模式只能在 HRTIMER_STxCAR 中的计数器自动重载值 CARL[15:0]大于 6 个 $t_{HRTIMER_CK}$ 周期时使用，例如当 CNTCKDIV[2:0] = 0 时为 0xC0，当 CNTCKDIV[2:0] = 1 时为 0x60。

突发模式控制的空闲模式

在突发模式中，空闲状态由突发模式控制器控制。具体请参考*突发模式*。

均衡空闲和延迟空闲的优先级高于突发模式：一旦触发均衡空闲和延迟空闲模式，任何突发模式的退出请求都将被丢弃。相反，如果在均衡空闲或延迟空闲退出时，突发模式仍有效，则突发模式将正常恢复。

突发模式控制器可以对任意两个输出 CHyOPRE（y = 0,1）进行控制。见*表 25-13. 突发模式控制的空闲状态时的输出*，HRTIMER_STxCHOCTL 寄存器中的 ISOy 位和 BMCHyIEN（y = 0,1）位可以对突发模式控制的空闲模式期间的每个输出的状态进行配置。

表 25-13. 突发模式控制的空闲状态时的输出

ISOy	BMCHyIEN	CHyOPRE (y=0,1)
x	0	无影响：输出不受突发模式控制器影响
0	1	在突发模式控制的空闲模式中输出复位电平
1	1	在突发模式控制的空闲模式中输出置位电平

通道输出级

每个 Slave_TIMERx 可以控制一对输出（STxCH0_O 和 STxCH1_O）。输出级有三种工作状态：

- 运行状态：STxCHy_O（y = 0,1）输出CHyOPRE（y = 0,1）的电平。
- 空闲状态：STxCHy_O（y = 0,1）输出由HRTIMER_STxCHOCTL寄存器中ISOy位定义的电平。
- 故障状态：STxCHy_O（y = 0,1）可以永久有效，无效或Hi-Z（由HRTIMER_STxCHOCTL寄

存器的CHyFLTOS位定义）。详见*故障输入*。

HRTIMER_CHOUTEN 寄存器中的 STxCHyEN 位和 HRTIMER_CHOUTDISF 寄存器中的 STxCHyDISF 位可以指示输出的状态，如*表 25-14. 输出级状态编程（x=0..7, y=0,1）*所述。

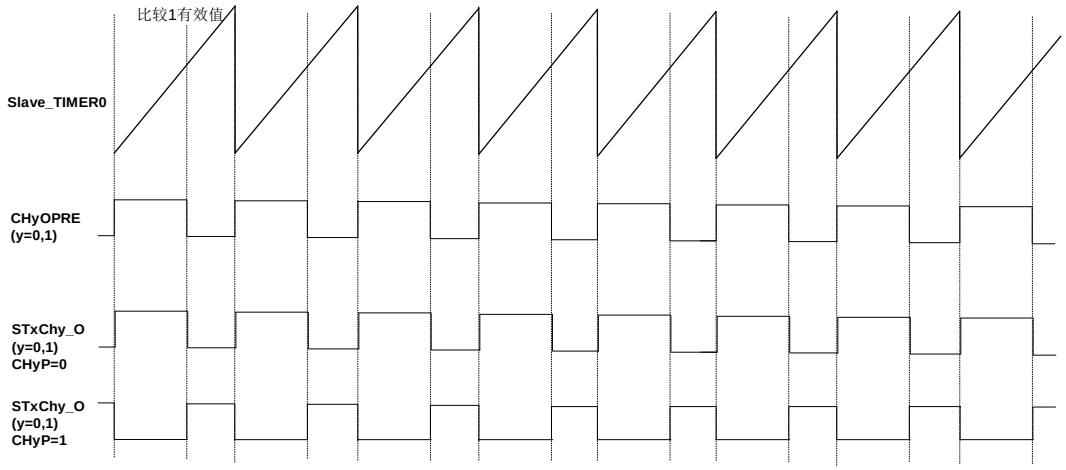
表 25-14. 输出级状态编程（x=0..7, y=0,1）

STxCHyEN	STxCHyDISF	输出级状态
1	x	运行状态
0	0	空闲状态
0	1	故障状态

将 HRTIMER_CHOUTDIS 寄存器中的 STxCHyDIS 位置 1，输出禁能，并使输出进入空闲状态。三种工作状态的优先级顺序为：空闲状态 > 故障状态 > 运行状态。

HRTIMER_STxCHOCTL 寄存器中的 CHyP 位可以设置输出极性。当 CHyP = 0 时，输出极性为高电平有效。当 CHyP = 1 时，输出极性为低电平有效。详见*图 25-39. CHyP=0 或 CHyP=1 时的 STxCHy_O 波形*。

图 25-39. CHyP=0 或 CHyP=1 时的 STxCHy_O 波形



使用 HRTIMER_STxCHOCTL 寄存器中的 CHyFLTOS [1:0]位域可以配置故障状态下的输出电平，如下所示：

- 2'b00：输出永远不会进入故障状态，并保持运行或空闲状态；
- 2'b01：故障状态时，输出有效电平；
- 2'b10：故障状态时，输出无效电平；
- 2'b11：故障状态时，输出为三态。

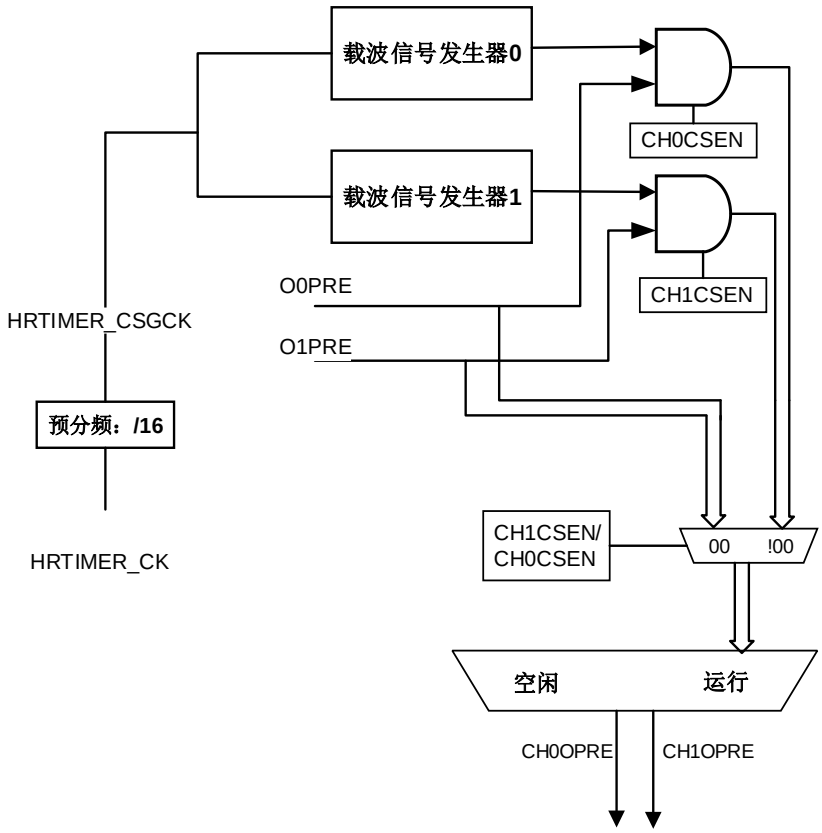
使用 HRTIMER_STxCHOCTL 寄存器中的 ISOy 位配置处于空闲状态的输出电平，如下：

- 2'b0：空闲状态时，输出无效电平；
- 2'b1：空闲状态时，输出有效电平。

载波信号模式

可以在 OyPRE(y=0,1)信号顶部添加一个高频载波信号，如*图 25-40. 载波信号结构框图*所示。

图 25-40. 载波信号结构框图



载波信号模式中，可以在载波信号开始之前定义一个特定的脉冲宽度。载波信号的频率和占空比是可配置的。详见*图 25-41. 载波模式使能时的 HRTIMER 输出*。

将 HRTIMER_STxCHOCTL 寄存器中的 CH0CSEN 和 CH1CSEN 位置 1，可以分别在通道 0 和 1 上使能载波信号模式。

第一个脉冲的脉冲宽度由 HRTIMER_STxCSCTL 寄存器中的 CSFSTPW [3:0]位域配置，公式如下：

$$t_{CSFSTPW}=(CSFSTPW[3:0]+1)* t_{HRTIMER_CSGCK}，其中t_{HRTIMER_CSGCK}=16 *t_{HRTIMER_CK}$$

载波信号的频率由 HRTIMER_STxCSCTL 寄存器中的 CSPRD [3:0]位域配置，公式如下：

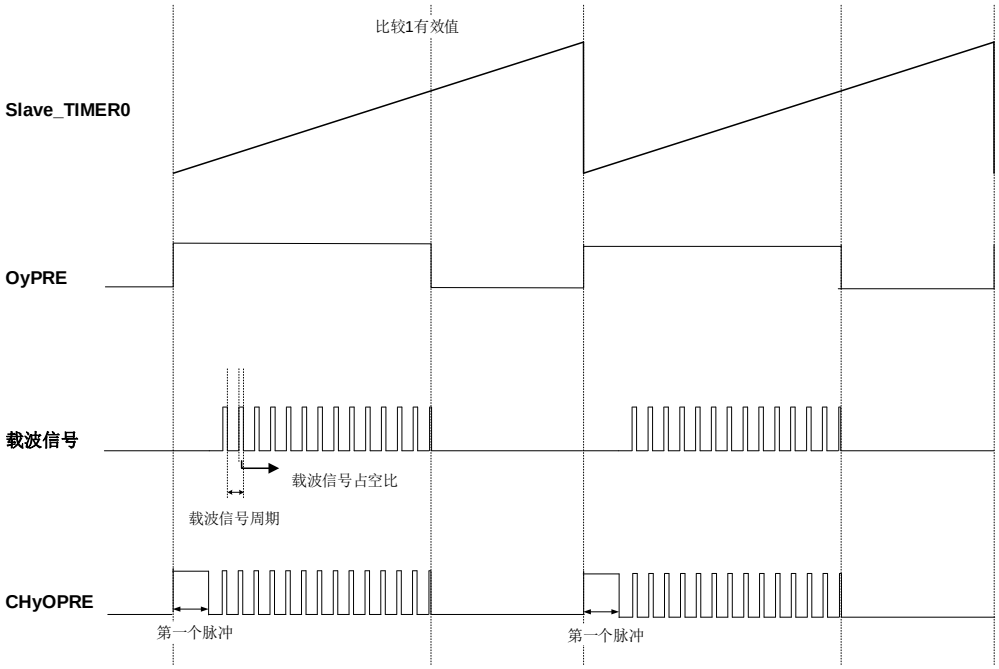
$$t_{CSPRD}=(CSPRD[3:0]+1)* t_{HRTIMER_CSGCK}，其中t_{HRTIMER_CSGCK}=16 *t_{HRTIMER_CK}$$

载波信号的占空比由 HRTIMER_STxCSCTL 寄存器中的 CSDTY[2:0]位域配置，步长为 12.5%。

载波信号模式中，载波信号发生器的输出与 OyPRE 逻辑与运算后输出。OyPRE（y = 0,1）输出

无效时，载波信号会立即停止，即使当前的载波周期未完成。具体请参考*图 25-41. 载波模式使能时的 HRTIMER 输出*。

图 25-41. 载波模式使能时的 HRTIMER 输出



同步输入启动/复位计数器

当 HRTIMER_MTCTL0 寄存器中的 SYNIRST 位置 1 时，同步输入可以产生计数器复位事件；当 HRTIMER_MTCTL0 寄存器中的 SYNISTRT 位置 1 时，同步输入可以启动计数器。详见*同步输入*。

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

更新事件和影子寄存器

Slave_TIMER 中的某些寄存器具有影子寄存器。MCU 复位后，影子寄存器被禁能。如果将 HRTIMER_STxCTL0 寄存器中的 SHWEN 位清 0，则禁能影子寄存器。写入这些寄存器的值将立即转移到有效寄存器中并生效。

如果 HRTIMER_STxCTL0 寄存器中的 SHWEN 位置 1，则使能影子寄存器，*表 25-15. Slave_TIMERx 影子寄存器和通用寄存器和更新事件*中列出的寄存器被预加载。写入这些寄存器的值将被传送到影子寄存器，且不会立即生效。当发生更新事件时，影子寄存器内容将转移到有效寄存器中并立即生效。

注意：当 SHWEN=1 时，才会产生更新事件。

*表 25-15. Slave_TIMERx 影子寄存器和通用寄存器和更新事件*列出了具有影子寄存器的寄存器和相应的更新事件。

表 25-15. Slave_TIMERx 影子寄存器和通用寄存器和更新事件

具有影子寄存器的寄存器	影子寄存器使能位	更新事件
HRTIMER_STxDMAINTEN HRTIMER_STxCAR HRTIMER_STxCREP HRTIMER_STxCMP0V HRTIMER_STxCMP0CP HRTIMER_STxCMP1V HRTIMER_STxCMP2V HRTIMER_STxCMP3V HRTIMER_STxDTCTL HRTIMER_STxCH0SET HRTIMER_STxCH0RST HRTIMER_STxCH1SET HRTIMER_STxCH1RST HRTIMER_STxCNTRST HRTIMER_STxCNTRSTA HRTIMER_STxACTL 寄存器中的 DTFCFG[15:9]和 DTRCFG[15:9]	HRTIMER_STxCTL0寄存器中的SHWEN位	软件（STxSUP 位） 重复事件（UPREP = 1） 计数器复位或翻转事件（UPRST = 1） 来自其他定时器的更新事件（Slave_TIMERx 是 UPBSTX，Master_TIMER 是 UPBMT） DMA 模式结束事件（UPSEL[3:0] = 4'b0001） DMA 模式结束事件之后的更新事件（UPSEL[3:0] = 4'b0010） STxUPINy(y=0..2)的上升沿产生更新事件 STxUPINy(y=0..2)的上升沿之后产生更新事件
HETIMER_ADCTRIGS0 HETIMER_ADCTRIGS0A HETIMER_ADCTRIGS1 HETIMER_ADCTRIGS1A HETIMER_ADCTRIGS2 HETIMER_ADCTRIGS2A HETIMER_ADCTRIGS3 HETIMER_ADCTRIGS3A	Master_TIMER或Slave_TIMERx(x = 0..7)的更新事件，由HRTIMER_MTCTL0或HRTIMER_STxCTL0中的SHWEN位域	

更新使能输入 STxUPINy（y = 0..2）是来自通用定时器的芯片内部信号，上升沿有效。具体请见*表 25-16. STxUPINy（y=0..2）和芯片内部信号*。

表 25-16. STxUPINy（y=0..2）和芯片内部信号

更新使能输入 STxUPINy（y = 0..2）	芯片内部信号
STxUPIN0	TIMER15_CH0_O
STxUPIN1	TIMER16_CH0_O

更新使能输入 STxUPINy（y = 0..2）	芯片内部信号
STxUPIN2	TIMER5_TRGO

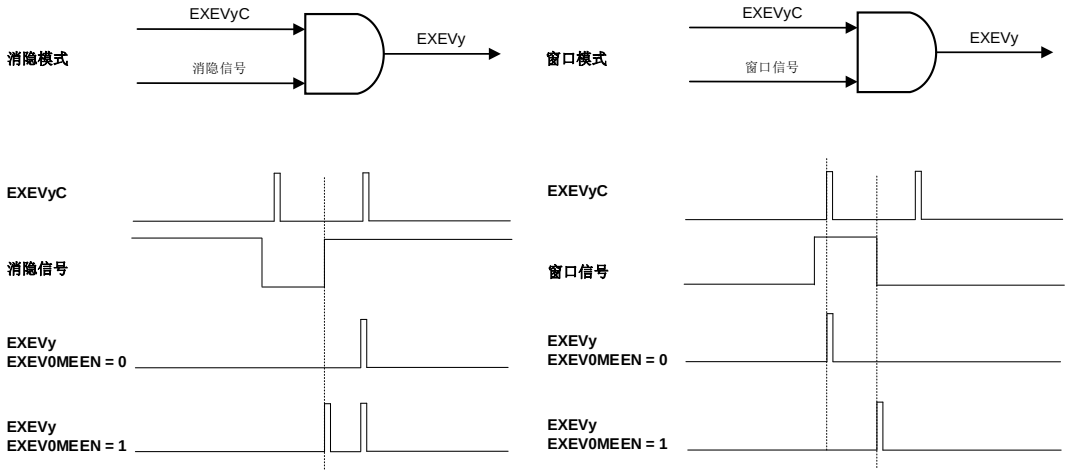
外部事件滤波

外部事件 EXEVyC（y = 0..9）在指定时间内被滤波，有两种滤波模式：

- 消隐模式：在指定时间内发生的外部事件被忽略；
- 窗口模式：在指定时间内发生的外部事件有效。

具体请见*图 25-42. 消隐模式和窗口模式*。

图 25-42. 消隐模式和窗口模式



参考*外部事件*章节可得更多关于外部事件 EXEVyC(y=0..9)的信息。

消隐模式

消隐模式中，在指定时间内发生的外部事件 EXEVyC（y = 0..9）被忽略，其他时间发生的外部事件有效。在指定的时间内，消隐信号为低电平。该模式由 EXEVyFM [4:0]位域配置，范围从 4’b0001 到 4’b1100。

消隐信号源有两种类型：

- Slave_TIMERx本身：指定的时间是指从计数器复位到比较事件发生持续的时间。（EXEVyFM [4:0] = 5’b00001~5’b00100，用于设置比较0~比较3事件）；
- 来自其他Slave_TIMER单元（EXEVyFM [5:0] = 5’b00101~5’b01100）的STBLKSRCz（z = 0..9）：指定的时间是指从选定的Slave_TIMER计数器复位到比较事件发生持续的时间。也可以是选定Slave_TIMER中的CH1OPRE信号（在这种情况下，只要CH1OPRE为低电平，事件将被忽略）具体请见*表25-17. 消隐模式下的滤波信号映射*。

表 25-17. 消隐模式下的滤波信号映射

		到 ST0	到 ST1	到 ST2	到 ST3	到 ST4	到 ST5	到 ST6	到 ST7
来自 ST0	CMP0	×	STBLK SRC0	×	STBLK SRC0	×	×	×	STBLK SRC0
	CMP1	×	×	STBLK SRC0	×	STBLK SRC0	×	×	×
	CMP3	×	STBLK SRC1	×	×	×	STBLK SRC0	×	×
	CH1OPRE	×	STBLK SRC2	×	×	×	×	STBLK SRC0	×
来自 ST1	CMP0	STBLK SRC0	×	STBLK SRC1	×	STBLK SRC1	×	STBLK SRC1	×
	CMP1	×	×	×	STBLK SRC1	×	STBLK SRC1	×	STBLK SRC1
	CMP3	STBLK SRC1	×	STBLK SRC2	×	×	×	×	×
	CH1OPRE	STBLK SRC2	×	×	×	×	×	×	×
来自 ST2	CMP0	STBLK SRC3	STBLK SRC3	×	STBLK SRC2	STBLK SRC2	×	×	×
	CMP1	×	STBLK SRC4	×	STBLK SRC3	×	×	×	×
	CMP3	STBLK SRC4	×	×	×	×	STBLK SRC2	×	STBLK SRC2
	CH1OPRE	×	×	×	STBLK SRC4	×	×	STBLK SRC2	×
来自 ST3	CMP0	STBLK SRC6	×	STBLK SRC4	×	STBLK SRC5	×	×	×
	CMP1	×	STBLK SRC6	×	×	×	STBLK SRC3	STBLK SRC3	×
	CMP3	×	×	STBLK SRC5	×	STBLK SRC6	STBLK SRC4	×	STBLK SRC3
	CH1OPRE	×	×	STBLK SRC6	×	STBLK SRC7	×	×	×
来自 ST4	CMP0	×	STBLK SRC7	×	STBLK SRC5	×	STBLK SRC5	×	×
	CMP1	STBLK SRC7	×	×	×	×	×	×	STBLK SRC4
	CMP3	×	×	STBLK	STBLK	×	STBLK	STBLK	×

		到 ST0	到 ST1	到 ST2	到 ST3	到 ST4	到 ST5	到 ST6	到 ST7
				SRC7	SRC6		SRC6	SRC4	
	CH1OPRE	×	×	×	×	×	STBLK SRC7	×	×
来自 ST5	CMP0	STBLK SRC5	×	STBLK SRC3	×	×	×	STBLK SRC5	STBLK SRC5
	CMP1	×	STBLK SRC5	×	×	×	×	×	×
	CMP3	×	×	×	STBLK SRC7	STBLK SRC3	×	STBLK SRC6	STBLK SRC6
	CH1OPRE	×	×	×	×	STBLK SRC4	×	STBLK SRC7	×
来自 ST6	CMP0	×	STBLK SRC8	×	STBLK SRC8	×	STBLK SRC8	×	STBLK SRC7
	CMP1	STBLK SRC8	×	STBLK SRC8	×	STBLK SRC8	×	×	STBLK SRC8
	CMP3	×	×	×	×	×	×	×	×
	CH1OPRE	×	×	×	×	×	×	×	STBLK SRC9
来自 ST7	CMP0	STBLK SRC9	×	STBLK SRC9	×	STBLK SRC9	×	STBLK SRC8	×
	CMP1	×	STBLK SRC9	×	STBLK SRC9	×	STBLK SRC9	STBLK SRC9	×
	CMP3	×	×	×	×	×	×	×	×
	CH1OPRE	×	×	×	×	×	×	×	×

当 EXEVyMEEN 位置 1 时，将使能外部事件的存储功能，外部事件不会立即生效。一旦指定的时间完成，该外部事件将被存储并生成。

窗口模式

窗口模式中，在指定时间内发生的外部事件 EXEVyC（y = 0..9）有效，其他时间发生的则被忽略。在指定的时间内，窗口信号为高电平。此模式由 EXEVyFM[4:0]位域配置，范围从 5'b01101 到 5'b01111。

如果在指定时间内未发生外部事件 EXEVyC(y=0..9)，则在指定时间结束时将产生超时事件。

窗口信号源有两种类型：

- Slave_TIMERx本身：指定的时间是指从计数器复位到比较事件发生持续的时间。（EXEVyFM [4:0] = 5'b01101和5'b01110，用于分别设置比较1和比较2事件）；
- 来自其他Slave_TIMER单元（EXEVyFM [4:0] = 5'b00101~5'b01100）的STWDSRC：指定的

时间是指从选定的Slave_TIMER计数器复位到比较事件发生持续的时间。具体请见*表25-18. 窗口模式的滤波信号映射*。

表 25-18. 窗口模式的滤波信号映射

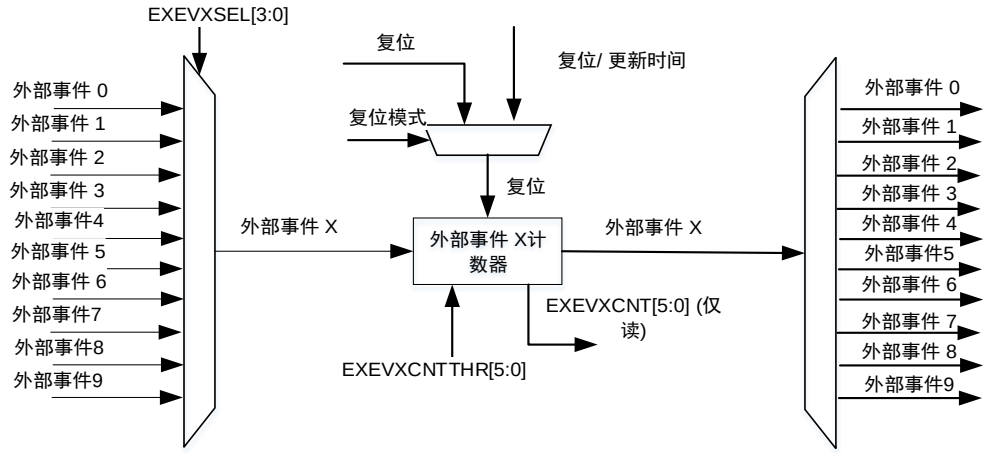
到 / 来自	T0 ST0	T0 ST1	T0 ST2	T0 ST3	T0 ST4	T0 ST5	T0 ST6	T0 ST7
STWDSRC	Slave_TIMER1 比较 1 事件	Slave_TIMER0 比较 1 事件	Slave_TIMER3 比较 1 事件	Slave_TIMER2 比较 1 事件	Slave_TIMER5 比较 1 事件	Slave_TIMER4 比较 1 事件	Slave_TIMER7 比较 1 事件	Slave_TIMER6 比较 1 事件

当 EXEVyMEEN 位置 1 时，将使能外部事件的存储功能，外部事件不会立即生效。一旦指定的时间完成，该外部事件将被存储并生成。

外部事件计数

10 个外部事件可以由外部事件 X 计数模块进行过滤。如*图 25-43. 外部事件 X 计数器*所示。

图 25-43. 外部事件 X 计数器

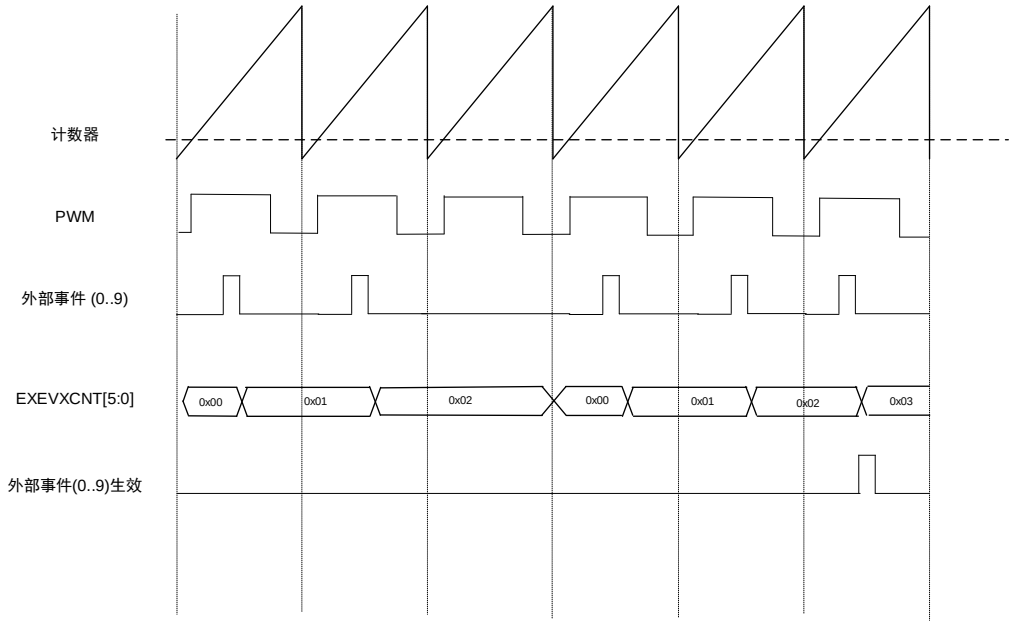


外部事件 X 计数器通过在 HRTIMER_STxEXEVFCFG2 寄存器中使能 EXEVXCEN 位启用。仅当 EXEVXCNT[5:0]的值大于或等于 EXEVXCNTTHR[5:0] + 1 时，外部事件才被视为有效。

当 EXEVXRSTM 位复位时，发生复位和更新事件时，EXEVXCNT[5:0]位被复位,外部事件仅在其在一个 PWM 周期内发生多次时被视为有效。当 EXEVXRSTM 位置位时，在每个 PWM 周期内生成外部事件时，EXEVXCNT[5:0]位会累积，并且在上一个 PWM 周期内没有外部事件出现时，EXEVXCNT[5:0]位将被复位。

在写入 EXEVXCNTTHR[5:0]位之后应置位 EXEVXCEN 位，且在启用外部事件 X 计数器时可以更改 EXEVXCNTTHR[5:0]位。新值 EXEVXCNTTHR[5:0]将在下一个周期生效。在启用外部事件 X 计数器时不能更改 EXEVXSEL[3:0]。

图 25-44. 当 EXEVXCEN 为 1 和 EXEVXCNTTHR[5:0]为 2 时外部事件 X 计数器



快速外部事件模式

可以根据实际需求动态调整外部事件的处理时间。当在 HRTIMER_EXEVCFG0 寄存器中将 EXEVxFAST 位复位时，需要在生效之前重新采样外部事件，从而增加一些延迟，并且可以生成高分辨率的脉冲。当在 HRTIMER_EXEVCFG0 寄存器中设置 EXEVxFAST 位时，启用了快速外部事件模式。在此模式下，将延迟最小化。

快速外部事件模式仅对电平事件有效，不适用于边沿事件，因此 HRTIMER_EXEVCFG0 寄存器中的 EXEVxEG[1:0] 位必须为 0x00。在此模式下，可以使用外部事件过滤，并且必须禁用外部事件锁存。

同一事件不能同时配置在 HRTIMER_STxCHySET 和 HRTIMER_STxCHyRST 寄存器中，当复位事件和置位事件同时发生时，复位事件具有更高的优先级。

在快速外部事件模式下，当外部事件发生并生效时，其他在 11 个 $t_{HRTIMER_CK}$ 内发生的外部事件无效。

25.4.3. DLL 校准

DLL 可以产生并校准高分辨率时钟 HRTIMER_HPCK（$f_{HRTIMER_HPCK} = 32 * f_{HRTIMER_CK}$）。DLL 模块可以一次或定期校准高分辨率时钟 HRTIMER_HPCK。

当 HRTIMER_DLLCCTL 寄存器中的 CLBPEREN 位置 1 时，使能定期的 DLL 校准，CLBPER [1:0] 位域配置校准周期。DLL 将在整个 HRTIMER 运行期间定期校准时钟。

当 HRTIMER_DLLCCTL 寄存器中的 CLBPEREN 位清 0 时，将 CLBSTRT 置 1，DLL 只校准一

次高分辨率时钟 HRTIMER_HPCK。

注意：仅当系统时钟频率在 150MHz 到 216MHz 时，DLL 校准有效。

25.4.4. 突发模式

突发模式控制器允许通过硬件使 CHyOPRE（y = 0,1）交替输出空闲和运行状态。该模式由 HRTIMER_BMCTL 寄存器中的 BMEN 位使能，通常在轻载情况中使用。

突发模式控制器包括：

- 1个计数器（BM-counter）；
- 1个比较寄存器：HRTIMER_BMCMPV，用于定义空闲状态的持续时间；
- 1个周期寄存器：HRTIMER_BMCAR，用于定义空闲和运行状态持续时间的总和。

BM-counter 的计数模式

BM-counter 可以运行在连续模式或单脉冲模式下。

当 BMCTN = 1 时，BM-counter 运行在连续模式下。BM-counter 从 0 连续计数到计数器重载值（HRTIMER_BMCAR）。当计数到计数器重载值时，计数器将从 0 重新启动。突发模式过程一直持续到 HRTIMER_BMCTL 中的 BMOPTF 位被复位。

当 BMCTN = 0 时，BM-counter 运行在单脉冲模式下。BM-counter 从 0 连续计数到计数器重载值（HRTIMER_BMCAR）。当计数到计数器重载值时，BM-counter 停止计数。

当计数到计数器重载值（HRTIMER_BMCAR）时，HRTIMER_INTF 寄存器中的 BMPERIF 位置 1，如果 BMPERIE = 1（在 HRTIMER_INTEN 寄存器中），则突发模式控制器产生突发模式周期中断请求。可以写 1 到 HRTIMER_INTC 中的 BMPERIFC 位来清除 BMPERIF 位。

突发模式的时序

BM-counter 由几个时钟源提供时钟，可以通过 HRTIMER_BMCTL 寄存器中的 BMCLKS[3:0]位选择。当选定的时钟源信号的上升沿到达时，BM-counter 计数值加 1。

当 BMCLKS[3:0] = 4'b1010 时，BM-counter 的时钟源是 $f_{HRTIMER_CK}$ 分频后得到的，分频系数由 HRTIMER_BMCTL 寄存器中的 BMPSC [3:0]位域定义。

当 BMCLKS[3:0] = 4'b0110~4'b1001 时，BM-counter 的时钟源是芯片内部信号：BMCLKy（y = 0..3），具体请见*表 25-19. 突发模式的芯片内部信号*。

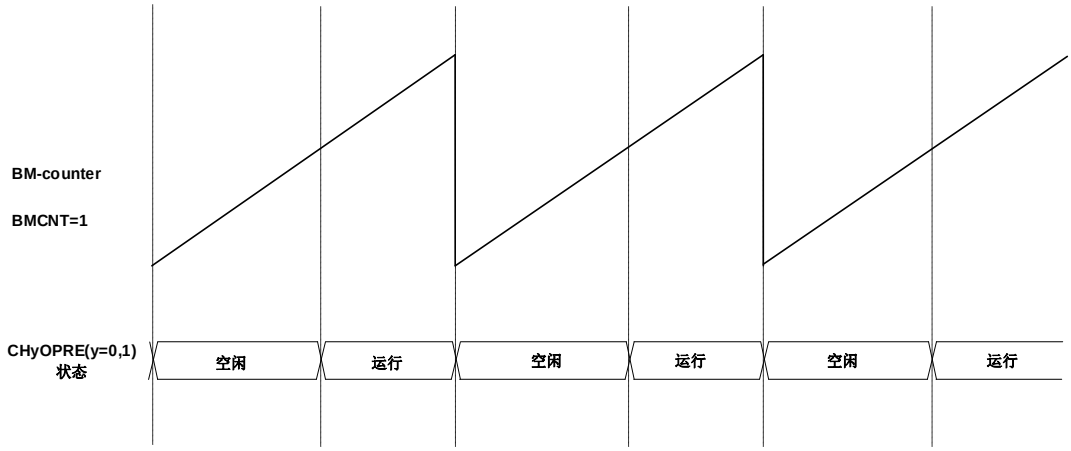
表 25-19. 突发模式的芯片内部信号

BMCLKy(y=0..3)	芯片内部信号
BMCLK0	TIMER15_CH0_O
BMCLK1	TIMER16_CH0_O

BMCLKy(y=0..3)	芯片内部信号
BMCLK2	TIMER6_TRGO
BMCLK3	保留

空闲状态的持续时间由 HRTIMER_BMCMPV 寄存器定义，并且 HRTIMER_BMCAR 寄存器定义了突发模式的周期，该周期值是空闲状态和运行状态持续时间之和，具体请见*图 25-45. 突发模式时序图*。

图 25-45. 突发模式时序图



当 BMSE 置位时，HRTIMER_BMCMPV 和 HRTIMER_BMCAR 寄存器是预装载，在下列情形下会从预装载传输到有效寄存器：

- 当使能突发模式时（BMEN=1）；
- 当突发模式周期结束时。

注意：当写 HRTIMER_BMCAR 后会暂时禁能更新，直到写 HRTIMER_BMCMPV 寄存器后才恢复更新。

突发模式进入

HRTIMER_BMSTRG 寄存器和 HRTIMER_BMSTRGA 寄存器中定义了 44 个可触发突发模式的事件。这些触发事件可以同时选择，然后再进行逻辑或运算。而在 BM-counter 的计数过程中，这些触发事件被忽略。这些触发事件分为七种：

1. Master_TIMER 事件：重复事件，复位/翻转事件，比较 0/1/2/3 事件；

2. Slave_TIMERx 事件：重复事件，复位/翻转事件，比较 0 和比较 1 事件；

3.外部事件：EXEV6 和 EXEV7；

4. EXEV6 事件之后的 Slave_TIMER0 周期事件；

5. EXEV7 事件之后的 Slave_TIMER3 周期事件；

6.芯片内部信号：TIMER6_TRGO；

7.软件：写 1 到 HRTIMER_BMSTRG 寄存器的 SWTRG 位。

触发事件发生时，有两种进入突发模式的方式：常规进入和延迟进入。

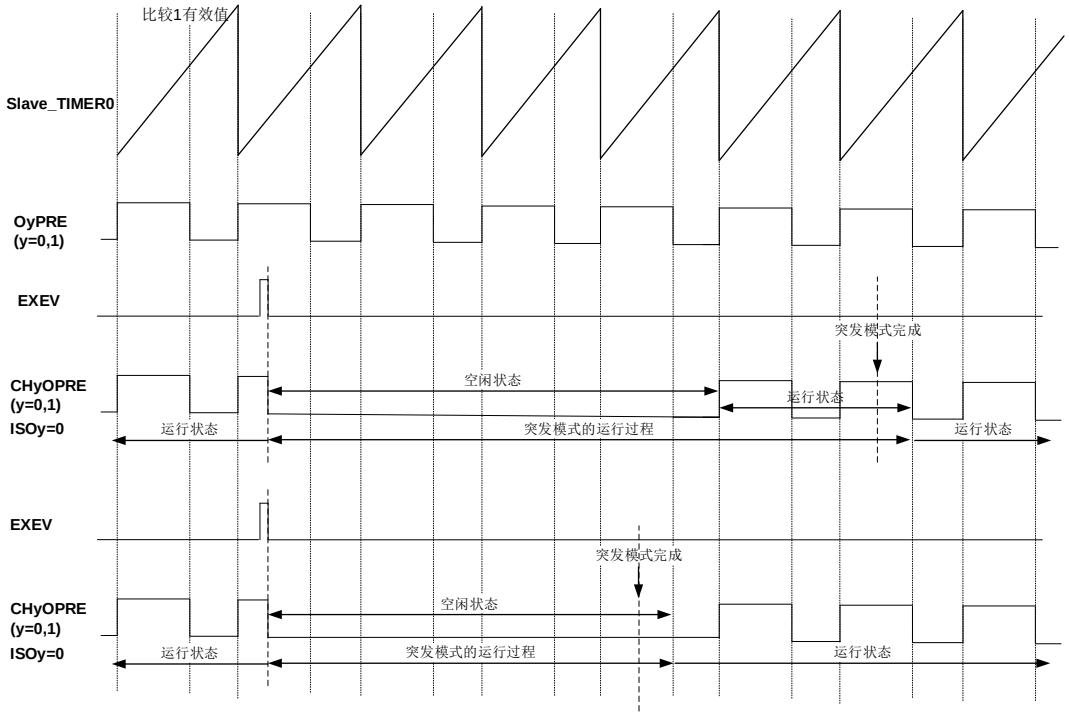
常规进入

当 HRTIMER_STxCHOCTL 寄存器中的 BMCHyDTI（y = 0，1）位为 0 时，突发模式是常规进入模式。选定事件发生后的第一个 BM-counter 计数时钟到来时，输出将进入突发模式，并输出空闲电平（根据 ISO0 位和 ISO1 位设置）。

*图 25-46. 突发模式的常规进入*显示了在以下配置时，Slave_TIMER0 中的 CHyOPRE 波形：

- CyOPRE 处于常规模式：HRTIMER_ST0CHOCTL 寄存器中的 DTEN = 0，HRTIMER_ST0CTL0寄存器中的BLNMEN = 0；
- 周期事件产生置位请求；
- 比较1事件产生复位请求；
- BM-counter的时钟源是Slave_TIMER0的翻转事件：BMCLKS [3:0] = 4’b0001。

图 25-46. 突发模式的常规进入



延迟进入

当 HRTIMER_STxCHOCTL 寄存器中的 BMCHyDTI（y = 0,1）位为 1 时，突发模式的进入被延迟。在进入突发模式之前，CHyOPRE 被强制插入死区时间。

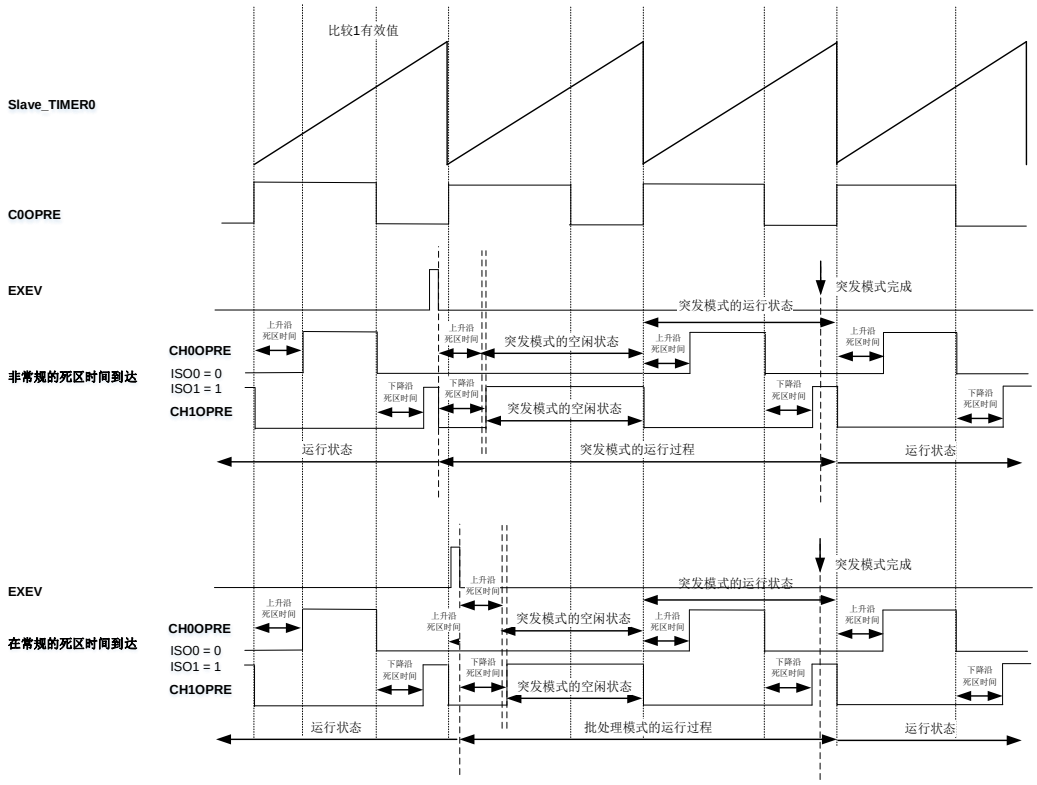
每个 CHyOPRE 都有自己的死区插入值：

- BMCH0DTI = 1时，DTRCFG[15:0]用于配置CH0OPRE的死区时间；
- BMCH1DTI = 1时，DTFCFG[15:0]用于配置CH1OPRE的死区时间。

延迟进入模式适用于以下情况：CHyOPRE(y=0,1)之一具有有效的空闲电平（ISOy = 1），且死区时间为正（DTRS /DTFS 设置为 0）。

在常规死区时间内，突发模式被触发，当前死区过程中止，将重新开始新的死区插入过程。详见*图 25-47. 突发模式的延迟进入*。

图 25-47. 突发模式的延迟进入



突发模式退出

在连续模式下，突发模式由软件强制退出。BMOPTF 或 BMEN 位重写为 0 后，发生输出置位/复位请求时，会退出突发模式。详情请见*图 25-46. 突发模式的常规进入*和*图 25-47. 突发模式的延迟进入*。

在单脉冲模式下，一旦经过空闲周期，就退出突发模式。

突发模式的时钟

可以在突发模式工作（运行，空闲）期间停止并复位 Master_TIMER 和 Slave_TIMERx（x = 0..7）

单元的计数器，可通过 HRTIMER_BMCTL 寄存器中的 BMMT 位和 BMSTx（x = 0..7）位进行配置：

- BMMT或BMSTx（x = 0..7）= 0：保持Master_TIMER或Slave_TIMERx（x = 0..7）的计数器时钟（HRTIMER_PSCCK），且计数器正常运行；
- BMMT或BMSTx（x = 0..7）= 1：Master_TIMER或Slave_TIMERx（x = 0..7）计数器时钟（HRTIMER_PSCCK）停止，并复位计数器。

使用 HRTIMER_STxCMP0CP 寄存器模拟突发模式

可以使用 HRTIMER_STxCMP0CP 寄存器来生成类似于突发模式控制的波形，配置如下：

- 比较0事件用于产生复位请求；
- 周期事件用于产生置位请求；
- 使用DMA（重复事件）连续将两个32位数据写入HRTIMER_STxCMP0CP寄存器，如下所示：

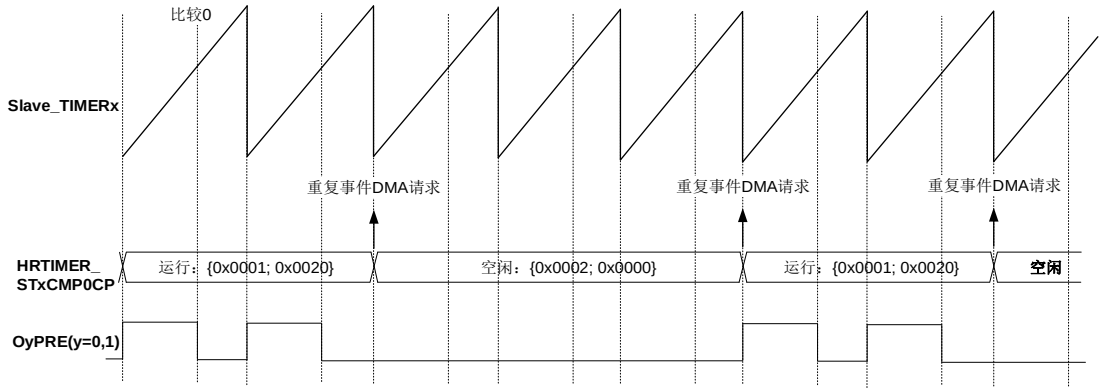
HRTIMER_STxCMP0CP = {CREP [7:0] =运行周期数-1；CMP0VAL [15:0] =占空比}

HRTIMER_STxCMP0CP = {CREP [7:0] =空闲周期数-1；CMP0VAL [15:0] = 0}

例如，要生成每 5 个周期中有 2 个周期输出置位的 PWM 波，可进行如下配置：

- 运行：HRTIMER_STxCMP0CP = {0x0001; 0x0020}；
- 空闲：HRTIMER_STxCMP0CP = {0x0002; 0x0000}。

图 25-48. 模拟突发模式示例



25.4.5. 同步输入/输出

同步电路在 Master_TIMER 内部：

- 同步输出：HRTIMER可以作为主机产生同步信号；
- 同步输入：HRTIMER也可以作为从机等待触发同步。

同步输出

可以将 HRTIMER 配置为主机，同步外部资源；

可以配置 HRTIMER_MTCTL0 寄存器中的 SYNOSRC[1:0]位域，选择发送到同步输出上的源。有以下四个源：

- 2'b00：Master_TIMER启动事件。在以下三种情况下可以将生成的启动事件用作同步输出，当MTCEN位置1时，当计数器在单脉冲模式下达到周期值后重新启动时，还有当CTNM或CNTRSTM位置1时，在计数期间发生的复位事件；
- 2'b01：Master_TIMER比较0事件；
- 2'b10：Slave_TIMER0复位和启动事件。它与Master_TIMER启动事件类似，除以下情况外：连续模式下的计数器翻转，在CNTRSTM = 0时的单脉冲模式下放弃的复位请求；
- 2'b11：Slave_TIMER0比较0事件。

HRTIMER_MTCTL0 寄存器中的 SYNOPLS[1:0]位域确定同步输出信号的极性：

- 2'b00：脉冲产生禁能。同步输出引脚HRTIMER_SCOUT上没有脉冲；
- 2'b01：保留；
- 2'b10：在同步输出引脚HRTIMER_SCOUT上生成正脉冲。正脉冲的长度为16 $t_{HRTIMER_CK}$个周期；
- 2'b11：在同步输出引脚HRTIMER_SCOUT上生成负脉冲。负脉冲的长度为16 $t_{HRTIMER_CK}$个周期。

同步输入

HRTIMER 可以作为从机等待触发同步。可以通过 HRTIMER_MTCTL0 寄存器中的 SYNISRC [1:0]位域选择同步输入源。有四个输入触发源可选：

- 2'b00：同步输入禁能；
- 2'b01：保留；
- 2'b10：芯片内部信号。高级定时器TIMER0的TIMER0_TRGO信号；
- 2'b11：芯片外部引脚。芯片外部引脚（HRTIMER_SCIN）上的正脉冲（上升沿有效）。

Master_TIMER 由 HRTIMER_MTCTL0 寄存器中的 SYNISTRT 位和 SYNIRST 位配置。Slave_TIMERx 由 HRTIMER_STxCTL0 寄存器中的 SYNISTRT 位和 SYNIRST 位配置。

当 SYNISTRT 置 1 时，必须先使能定时器（将 STxCEN 位或 MTCEN 位置 1），则同步输入信号将启动计数器。在连续模式下，即使 STxCEN 位或 MTCEN 位被置 1，计数器也不会启动，只有在同步输入信号到达后才会启动。

当 SYNIRST 置 1 时，同步输入信号将复位计数器，并像其他任何重置事件一样递减重复计数器。

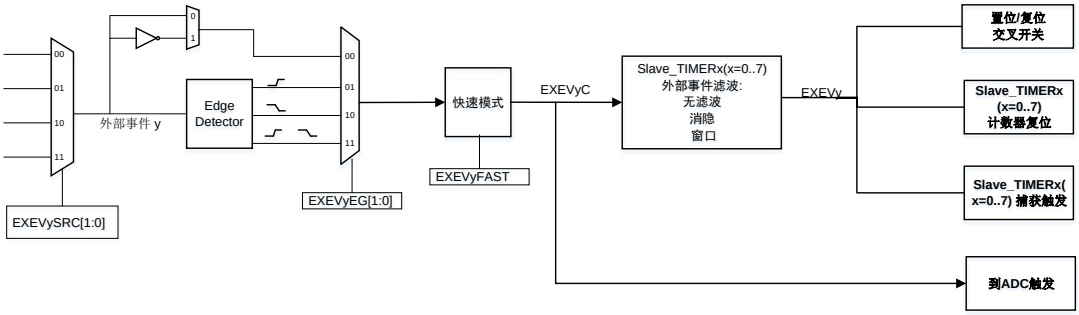
25.4.6. 外部事件

10 个外部事件可以同时用于 8 个 Slave_TIMER 中的任意 1 个。通过 HRTIMER_EXEVCFG0 寄

存器配置外部事件 y（y = 0..4），通过 HRTIMER_EXEVCFG1 和 HRTIMER_EXEVDFCTL 寄存器配置外部事件 y（y = 5..9）。

处理外部事件 y（y = 0..4）的过程如*图 25-49. 外部事件 y（y=0..4）处理过程框图*所示。

图 25-49. 外部事件 y（y=0..4）处理过程框图

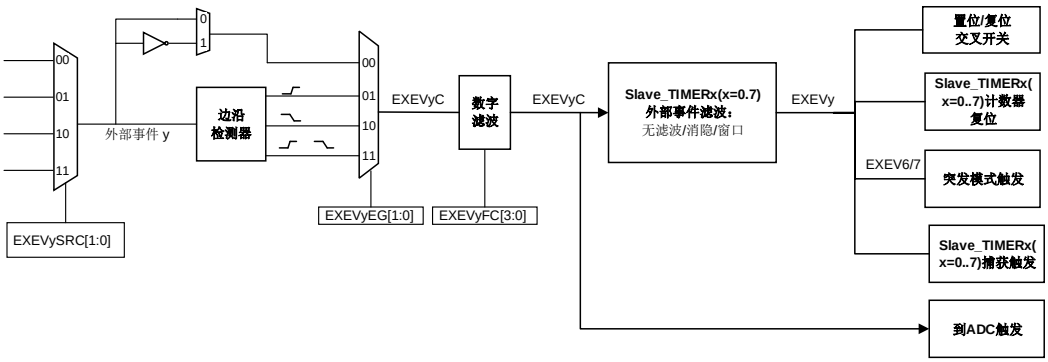


外部事件 y（y = 0..4）的配置如下：

- 4个源：通过EXEVySRC[1:0]位域进行配置；
- 有效沿选择：通过EXEVyEG[1:0]位域进行配置。可以是电平有效的或边沿有效（上升沿，下降沿或两者兼有）；
- 极性选择：在电平有效（EXEVyEG [1:0] = 2'b00）时，由EXEVyP位进行配置。
- 当EXEVyFAST位置1时，外部事件工作在异步模式。

处理外部事件 y（y = 5..9）的过程如*图 25-50. 外部事件 y（y=5..9）处理过程框图*所示。

图 25-50. 外部事件 y（y=5..9）处理过程框图



外部事件 y（y=5..9）的配置如下：

- 4个源：通过EXEVySRC[1:0]位域进行配置；
- 有效沿选择：通过EXEVyEG[1:0]位域进行配置。可以是电平有效的或边沿有效（上升沿，下降沿或两者兼有）；
- 极性选择：在电平有效（EXEVyEG [1:0] = 2'b00）时，由EXEV0P位进行配置；
- 数字滤波配置：配置HRTIMER_EXEVDFCTL寄存器中的EXEVyFC[3:0]位域。

数字滤波器的采样时钟 $f_{HRTIMER_EXEVFCK}$ 由 HRTIMER_EXEVDFCTL 寄存器中的 EXEVFDIV[2:0]位

域定义。

这些外部事件源 EXEVySRCz（y = 0..9，z = 0..4）可以来自比较器、数字输入引脚、ADC 的模拟看门狗和 TIMER_TRGO。具体请参考*表 25-20. 外部事件映射*。

表 25-20. 外部事件映射

外部事件	EXEVySRC0	EXEVySRC1	EXEVySRC2	EXEVySRC3
外部事件 0	PC12	比较器 1	TIMER0_TRGO	ADC0_WD0_OUT
外部事件 1	PC11	比较器 3	TIMER1_TRGO	ADC0_WD1_OUT
外部事件 2	PB7	比较器 5	TIMER2_TRGO	ADC0_WD2_OUT
外部事件 3	PB6	比较器 0	比较器 4	ADC1_WD0_OUT
外部事件 4	PB9	比较器 2	比较器 6	ADC1_WD1_OUT
外部事件 5	PB5	比较器 1	比较器 0	ADC1_WD2_OUT
外部事件 6	PB4	比较器 3	TIMER6_TRGO	ADC2_WD0_OUT
外部事件 7	PB8	比较器 5	比较器 2	ADC3_WD0_OUT
外部事件 8	PB3	比较器 4	TIMER14_TRGO	比较器 3
外部事件 9	PC5/PC6	比较器 6	TIMER5_TRGO	ADC3_WD0_OUT

注意：“×”表示不可用。

可以直接使用外部事件 y（y = 0..9），也可以对其进行滤波处理（以在指定时间内限制其操作）。具体参考*外部事件滤波*。

25.4.7. 故障输入

HRTIMER 具有故障保护机制，可用于每个 Slave_TIMERx。具体请参考*图 25-51. 故障输入结构图*。

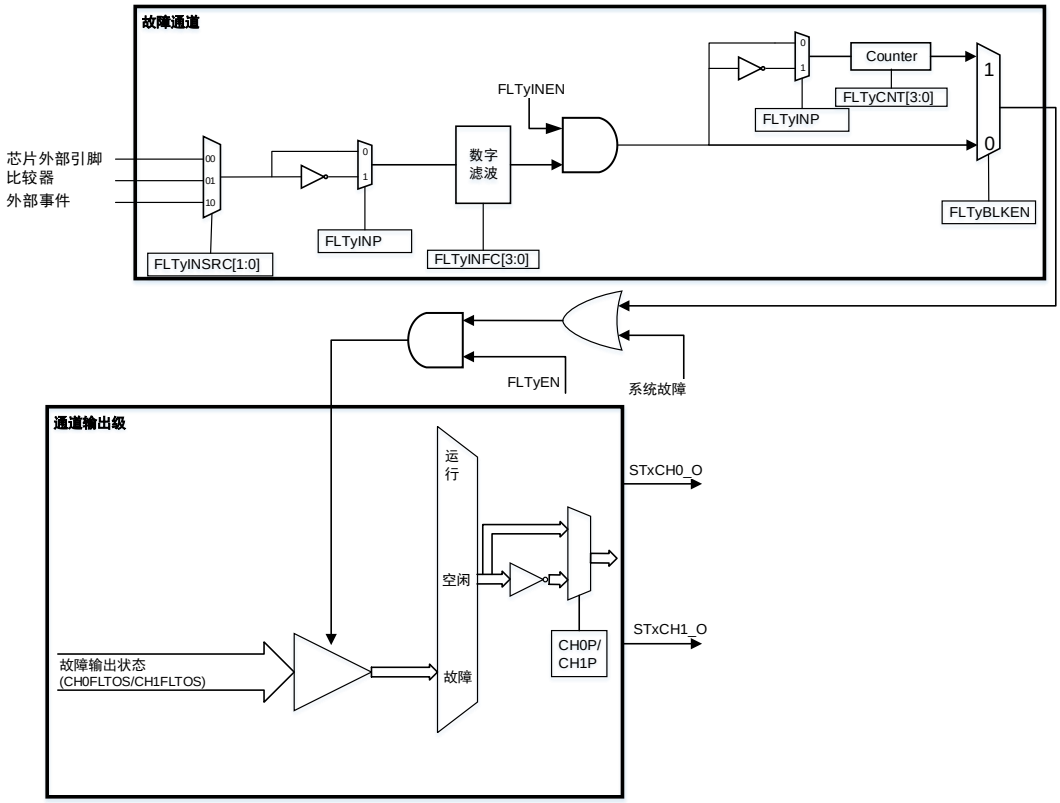
发生故障事件时，输出（STxCHy_O，x = 0..7，y = 0,1）为预置的电平。

预置电平由 HRTIMER_STxCHOCTL 寄存器中的 CHyFLTOS[1:0]位域配置，保护机制可以处理三种类型的故障源：

- 故障通道：来自数字输入引脚或比较器的故障事件；

- 系统故障：来自MCU内部的信号，例如SRAM奇偶校验器；
- 故障事件：故障事件来自外部事件y(y=0..9)。

图 25-51. 故障输入结构图



通过 HRTIMER_STxFLTCTL 寄存器中的 FLTyEN 位使能故障输入。对 HRTIMER_STxFLTCTL 寄存器中的 FLTENPROT 位的一次写入可以保护 FLTyEN 位。当 FLTENPROT 位置 1 时，FLTyEN 位写保护（只读）。

故障通道

可以通过 HRTIMER_FLTINCFG0 和 HRTIMER_FLTINCFG1 寄存器配置所有的故障通道。

FLTyINSRC（y = 0..7）位用于选择故障通道源，即可以是数字输入引脚，也可以是比较器输出或外部事件。每个 Slave_TIMERx 都有 8 个故障通道（故障通道 0-7）可以使用。具体请见*表 25-21. 故障通道映射*。

表 25-21. 故障通道映射

故障通道	FLTyINSRC = 00 (外部引脚)	FLTyINSRC = 01 (内部信号)	FLTyINSRC = 10 (外部事件)
故障通道 0	PA12	比较器 1	外部事件 0
故障通道 1	PA15	比较器 3	外部事件 1

故障通道	FLTyINSRC = 00 (外部引脚)	FLTyINSRC = 01 (内部信号)	FLTyINSRC = 10 (外部事件)
故障通道 2	PB10	比较器 5	外部事件 2
故障通道 3	PB11	比较器 0	外部事件 3
故障通道 4	PB0/PC7	比较器 2	外部事件 4
故障通道 5	PC10	比较器 4	外部事件 5
故障通道 6	PC3	比较器 6	外部事件 6
故障通道 7	PC4	比较器 7	外部事件 7

注意：“×”表示不可用。

可以通过 HRTIMER_FLTINCFG0 和 HRTIMER_FLTINCFG1 寄存器中的 FLTyINP 位来配置故障信号的极性。如果 FLTyINP = 0，信号低电平有效；如果 FLTyINP = 1，则高电平有效。

可通过 HRTIMER_FLTINCFG0 和 HRTIMER_FLTINCFG1 寄存器中的 FLTyINFC[3:0]位域，对设置极性后的数字信号滤波器进行配置。数字滤波器采样时钟 $f_{HRTIMER_FLTFCK}$ 由 HRTIMER_FLTINCFG1 寄存器中的 FLTFDIV[2:0]位域定义。

可通过 HRTIMER_FLTINCFG0 和 HRTIMER_FLTINCFG1 寄存器中的 FLTyINEN 位来使能故障通道 y（y = 0..7），所有通道可同时使能。

对 HRTIMER_FLTINCFG0 和 HRTIMER_FLTINCFG1 寄存器中 FLTyINPROT 位的一次写入，可保护 FLT0INEN 位，FLT0INP 位，FLT0INSRC 位和 FLT0INFC [3:0]位域。当 FLTyINPROT 位置 1 时，这些位写保护（只读）。

故障消隐

在故障消隐模式下，在指定时间内发生的故障通道将被忽略，请参考*表 25-22. 故障通道消隐*。

表 25-22. 故障通道消隐

故障通道	FLTxBLKS = 0,复位和比较窗口		FLTxBLKS = 1，比较和比较窗口	
	消隐开始	消隐结束	消隐开始	消隐结束
故障通道 0	Slave_TIMER0 复位 / 更新	Slave_TIMER0 比较 2	Slave_TIMER0 比较 3	Slave_TIMER0 比较 2
故障通道 1	Slave_TIMER1 复位 / 更新	Slave_TIMER1 比较 2	Slave_TIMER1 比较 3	Slave_TIMER1 比较 2
故障通道 2	Slave_TIMER2 复位 / 更新	Slave_TIMER2 比较 2	Slave_TIMER2 比较 3	Slave_TIMER2 比较 2
故障通道 3	Slave_TIMER3 复位 / 更新	Slave_TIMER3 比较 2	Slave_TIMER3 比较 3	Slave_TIMER3 比较 2
故障通道 4	Slave_TIMER4 复位 / 更新	Slave_TIMER4 比较 2	Slave_TIMER4 比较 3	Slave_TIMER4 比较 2
故障通道 5	Slave_TIMER5 复	Slave_TIMER5 比	Slave_TIMER5 比	Slave_TIMER5 比

故障通道	FLTxBLKS = 0,复位和比较窗口		FLTxBLKS = 1, 比较和比较窗口	
	消隐开始	消隐结束	消隐开始	消隐结束
	位 / 更新	较 2	较 3	较 2
故障通道 6	Slave_TIMER6 复位 / 更新	Slave_TIMER6 比较 2	Slave_TIMER6 比较 3	Slave_TIMER6 比较 2
故障通道 7	Slave_TIMER7 复位 / 更新	Slave_TIMER7 比较 2	Slave_TIMER7 比较 3	Slave_TIMER7 比较 2

故障计数阈值由 HRTIMER_FLTINCFG2 寄存器和 HRTIMER_FLTINCFG3 寄存器中的 FLTyCNT[3:0]（y = 0..7）配置。当故障通道事件的数量等于(FLTyCNT[3:0]+1) 时，故障生效。

FLTxRST 位复位，当复位或翻转事件发生时，故障计数器被复位，可参考*表 25-23. 故障通道复位源*。

当 FLTxRST 位置位时，故障计数器复位模式由 HRTIMER_FLTRECCTL 决定。当 HRTIMER_FLTRECCTL 寄存器中的 FLTRECCTL 位置 0 时，故障计数器仅在没有故障事件处于有效状态时，每次复位/翻转事件时复位。当 HRTIMER_FLTRECCTL 寄存器中的 FLTRECCTL 位置 1 时，故障计数器仅在一个周期内没有发生故障事件时，每次复位/翻转事件时复位。

注意：当 FLTxRST 位复位，故障计数器复位源取决于*表 25-23. 故障通道复位源*。当 FLTxRST 位置位时，故障计数器复位源来自与故障通道连接的 Slave_TIMERx，与故障通道连接的 Slave_TIMERx 的复位/翻转事件用于复位故障计数器。

图 25-52. 当 FLTxRST 为 1，FLTRECCTL 为 1 和 FLTxCNT[3:0] = 0x03 时故障计数器

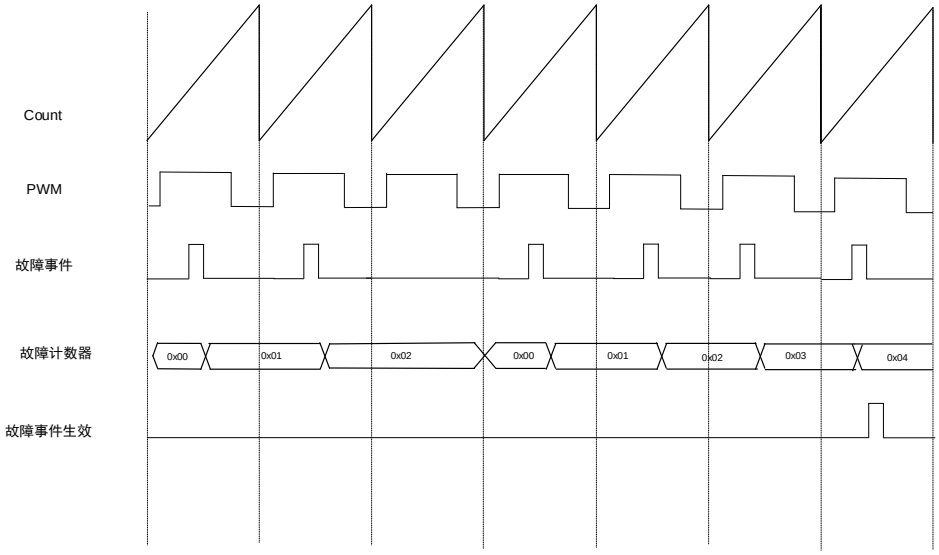


表 25-23. 故障通道复位源

故障通道	故障计数器复位源
故障通道 0	Slave_TIMER0 复位 / 翻转

故障通道	故障计数器复位源
故障通道 1	Slave_TIMER1 复位 / 翻转
故障通道 2	Slave_TIMER2 复位 / 翻转
故障通道 3	Slave_TIMER3 复位 / 翻转
故障通道 4	Slave_TIMER4 复位 / 翻转
故障通道 5	Slave_TIMER5 复位 / 翻转
故障通道 6	Slave_TIMER6 复位 / 翻转
故障通道 7	Slave_TIMER7 复位 / 翻转

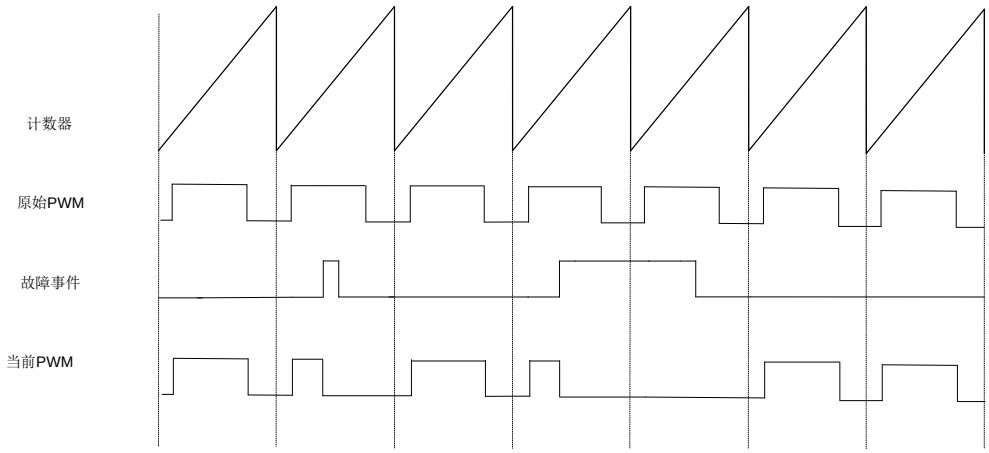
故障恢复

故障硬件自动恢复

当故障源消失后，HRTIMER 通道恢复 PWM 波输出。

HRTIMER_STxACTL 寄存器中的 FLTAR 位，HRTIMER_FLTINCFG2 寄存器和 HRTIMER_FLTINCFG3 寄存器中的 FLTxRST 位同时使用，即同时置位 FLTxRST（x = 0..7）和 FLTAR 位，能够实现故障消失后，下周期通道自动恢复 PWM 波输出。

图 25-53. 当 FLTxRST 为 1 和 FLTAR 为 1 时故障自动恢复



故障恢复保护

当 FLTxRST（x = 0..7）位置 1 和 FLTRECCTL 位置 0 时，故障事件没有消失前，软件不能恢复 PWM 输出。直到故障源消失且重新使能通道输出（置位 HRTIMER_CHOUTEN 寄存器的 STxCHyEN 位, x = 0..7, y = 0,1）才能恢复 PWM 输出。

注意：HRTIMER_CHOUTEN 寄存器中的 STxCHyEN 位无法正确指示对应通道的实际输出状态。具体问题如下：

1）在硬件故障自动恢复后，PWM 通道恢复输出且 PWM 波形恢复正常，但 STxCHyEN 位仍错误

地保持为 0，而不是恢复为 1。

2）在故障有效时，即使通过软件手动设置 STxCHyEN 位，PWM 输出仍保持禁用状态，而 STxCHyEN 位却错误地显示为 1。

由于这些问题，在使用故障输入功能时，STxCHyEN 位无法可靠地指示通道状态。因此，建议避免依赖读取该位进行状态验证。

系统故障

系统故障来自芯片内部的信号：

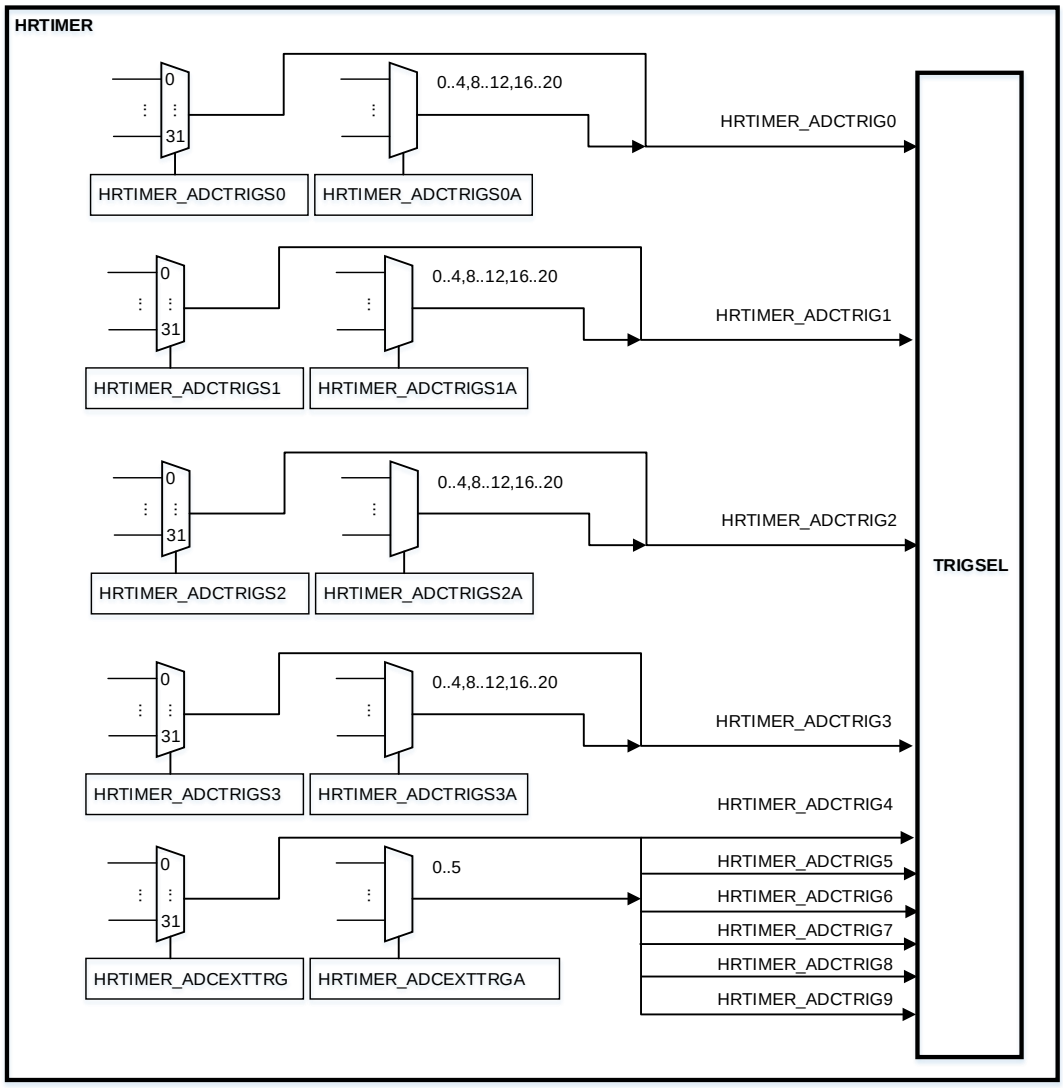
- 时钟监视生成的HXTAL故障事件；
- Cortex®-M33锁定信号；
- 低压检测器（LVD）的输出。

当 HRTIMER_STxFLTCTL 寄存器中的 FLTyEN 位置 1 时，系统故障才有效。系统故障可以覆盖故障通道输入（逻辑或）。

25.4.8. ADC 触发

TRIGSEL 可以触发 ADC，10 个独立的 ADC 触发（HRTIMER_ADCTRIG0 - HRTIMER_ADCTRIG9）可用于使能 ADC 的常规组。具体请参考*图 25-54. ADC 触发源选择图*。

图 25-54. ADC 触发源选择图



对于 ADC 触发 0 到 3（HRTIMER_ADCTRIG0 - HRTIMER_ADCTRIG3），每个触发输出最多可以连接（逻辑或运算）47 个事件。32 个事件在 HRTIMER_ADCTRIGSy（y = 0..3）寄存器中定义，15 个事件在 HRTIMER_ADCTRIGSyA（y = 0..3）寄存器中定义。

对于 ADC 触发 4 到 9（HRTIMER_ADCTRIG4 - HRTIMER_ADCTRIG9），每个触发输出一次只能连接 1 个事件（从 47 个事件中选择一个）。47 个事件在 HRTIMER_ADCEXTTRG、HRTIMER_ADCEXTTRGA 寄存器中定义。

HRTIMER_ADCTRIGSy（y = 0..3）寄存器，HRTIMER_ADCTRIGSyA（y = 0..3）寄存器，HRTIMER_ADCEXTTRG 寄存器和 HRTIMER_ADCEXTTRGA 寄存器使能预加载，并可以使用与其相关的定时器进行同步更新。更新源由 HRTIMER_CTL0 寄存器和 HRTIMER_ADCTRGUPD 寄存器中的 ADTGyUSRC[3:0]位域定义。例如，ADTGyUSRC [3:0] = 3’b0001，Slaver_TIMER0 是更新源：

- 如果HRTIMER_STxCTL0寄存器中的SHWEN = 1，HRTIMER_ADCTRIGSy（y = 0..3）寄存器和HRTIMER_ADCTRIGSyA（y = 0..3）被预加载，可以与Slaver_TIMER0同步更新；
- 如果HRTIMER_STxCTL0寄存器中的SHWEN = 0，HRTIMER_ADCTRIGSy（y = 0..3）寄存器和HRTIMER_ADCTRIGSyA（y = 0..3）不会被预加载，写访问将使触发源立即更新。

HRTIMER_ADCTRIGS0- HRTIMER_ADCTRIGS9为了更兼容ADC触发，设置了触发分频等功能，需要注意的是，这几种触发源连接到TRIGSEL，具体触发什么模块由TRIGSEL决定。

ADC 触发分频

ADC 的触发频率可以通过配置 HRTIMER_ADCPSCRy（y=0,1）寄存器来进行调整。具体的配置涉及设置 HRTIMER_ADCPSCRy 寄存器中的 ADCxPSC[4:0] 位。

在向上计数模式下，触发频率仅与 HRTIMER_ADCPSCRy 寄存器中的 ADCxPSC[4:0] 位有关。然而，在中央对齐计数模式下，ADC 触发频率不仅受到 ADCxPSC[4:0]的影响，还受到 HRTIMER_STxCTL1 寄存器中的 ADCROVM[1:0] 位的影响。ADCROVM[1:0] 的不同配置如下：

- ADCROVM[1:0] = 00：计数器在向上计数和向下计数方向时生成 ADC 触发事件。
- ADCROVM[1:0] = 01: 计数器在向下计数方向时生成 ADC 触发事件。
- ADCROVM[1:0] = 10：计数器在向上计数方向时生成 ADC 触发事件。

图 25-55. 向上计数模式下 ADC 触发分频

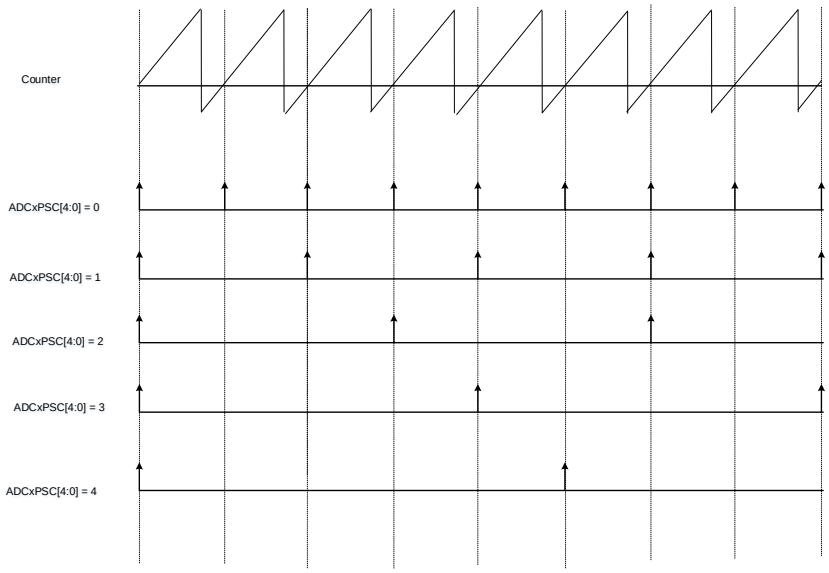
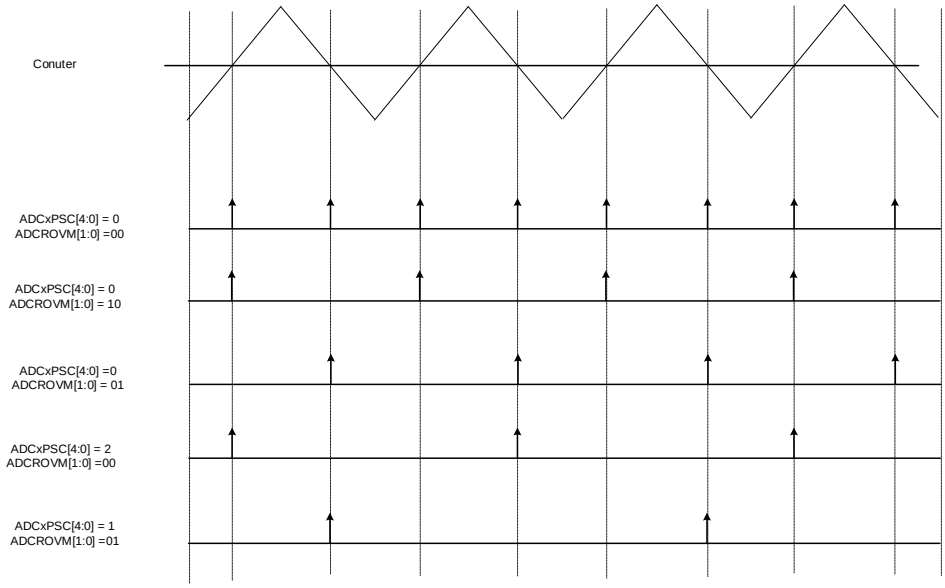


图 25-56. 中央对齐计数模式下 ADC 触发分频



25.4.9. DAC 触发

HRTIMER 允许使用定时器更新同步更新片上 DAC。Master_TIMER 和 Slave_TIMERx 的更新事件可以在 HRTIMER_DACTRIGy（y = 0..2）上生成 DAC 更新触发。

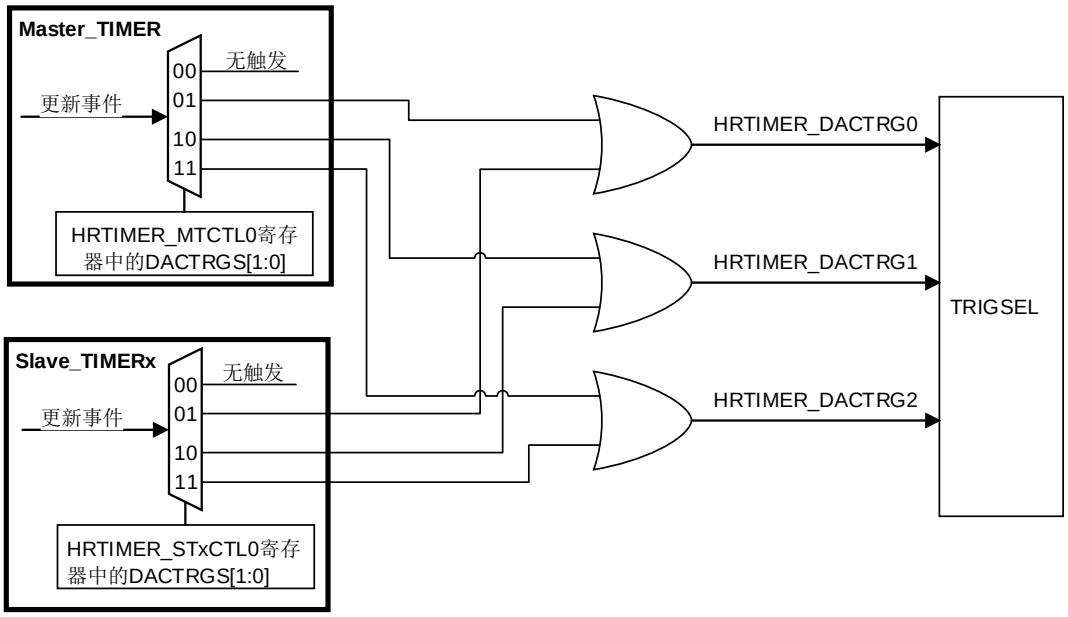
HRTIMER_MTCTL0 和 HRTIMER_STxCTL0 寄存器中的 DACTRGS [1:0]位域配置如下：

- 00：没有DAC触发事件发生；
- 01：在HRTIMER_DACTRIG0上生成DAC触发事件；
- 10：在HRTIMER_DACTRIG1上生成DAC触发事件；
- 11：在HRTIMER_DACTRIG2上生成DAC触发事件。

HRTIMER_DACTRG0- HRTIMER_DACTRG2兼容DAC触发，需要注意的是，这几种触发源连接到TRIGSEL，具体触发什么模块由TRIGSEL决定。

在多个计时器中使能 DACTRGS [1:0]位域时，HRTIMER_DACTRIGy（y = 0..2）将由所有定时器的更新事件或组成。具体请参考*图 25-57. DAC 触发源选择图*。

图 25-57. DAC 触发源选择图



在 DAC 触发输出上生成一个持续 32 个 f_{APB2} 时钟周期的脉冲。同步脉冲在这 32 个 APB2 时钟周期的空闲期，在此期间，任何新的 DAC 更新请求都将被忽略。因此，最大同步频率为 f_{APB2} / 64。

HRTIMER_DACTRIG0- HRTIMER_DACTRIG2 触发源连接到 TRIGSEL，具体触发什么模块由 TRIGSEL 决定。

25.4.10. 双通道触发模式

启用双通道触发模式需要设置 HRTIMER_STxCTL1 寄存器中的 TRIGEN 位。并且定时器运行中(STxCEN 位置位)，就无法更改 TRIGEN 位。此时 HRTIMER_STxCTL1 寄存器中的 TRIG0M 和 TRIG1M 配置 Slave_TIMERx 输出的触发信号 TRIG0 和 TRIG1 连接到 TRIGSEL（在 TRIGSEL 模块分别为 HRTIMER_STx_TRIG0 和 HRTIMER_STx_TRIG1），可通过配置 HRTIMER_STxCTL1 寄存器中的 TRIG0M 位和 TRIG1M 位，选择触发信号源。

TRIG0 信号通过 TRIG0M 进行配置。

- TRIG0M = 0：比较1事件生成TRIG0触发信号。
- TRIG0M = 1：输出0复位事件生成TRIG0触发信号。

TRIG1 信号通过 TRIG1M 进行配置。

- TRIG1M = 0：计数器复位或翻转事件生成TRIG1触发信号。
- TRIG1M = 1：输出0置位事件生成TRIG1触发信号。

下面以 DAC 为例来说明双通道触发模式的使用场景，通过双通道触发模式可以轻松实现斜坡补偿技术和滞环控制。在此模式下，DAC 输出一个逐渐减小的锯齿信号，锯齿波的周期与 PWM 波的

周期同步。

注意：在双通道触发模式下，当 TRIG0M = 0 和 TRIG1M = 0 时，只有 Slave timer 0 和 Slave timer 3 支持硬件斜坡补偿，但比较 1 寄存器具有以下限制：

当 CNTCKDIV[2:0] < 3'b101

- 最小值必须大于或等于8个$t_{HRTIMER_CK}$周期对应的计数值；

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

当 CNTCKDIV[2:0] >= 3'b101

- 最小值必须大于或等于0x0008。

DAC 模块可根据来自 TRIGSEL 的触发信号自定义选择哪个信号触发 DAC 产生步进或者复位信号。此处举例描述以来自 HRTIMER 的 TRIG0 产生步进信号，TRIG1 产生复位信号。需要注意的是触发 DAC 步进或复位的信号不仅仅是这两个信号，凡是来自 TRIGSEL 的信号均可触发。

图 25-58. 当 TRIG0M= 0 和 TRIG1M = 0 时 DAC 触发

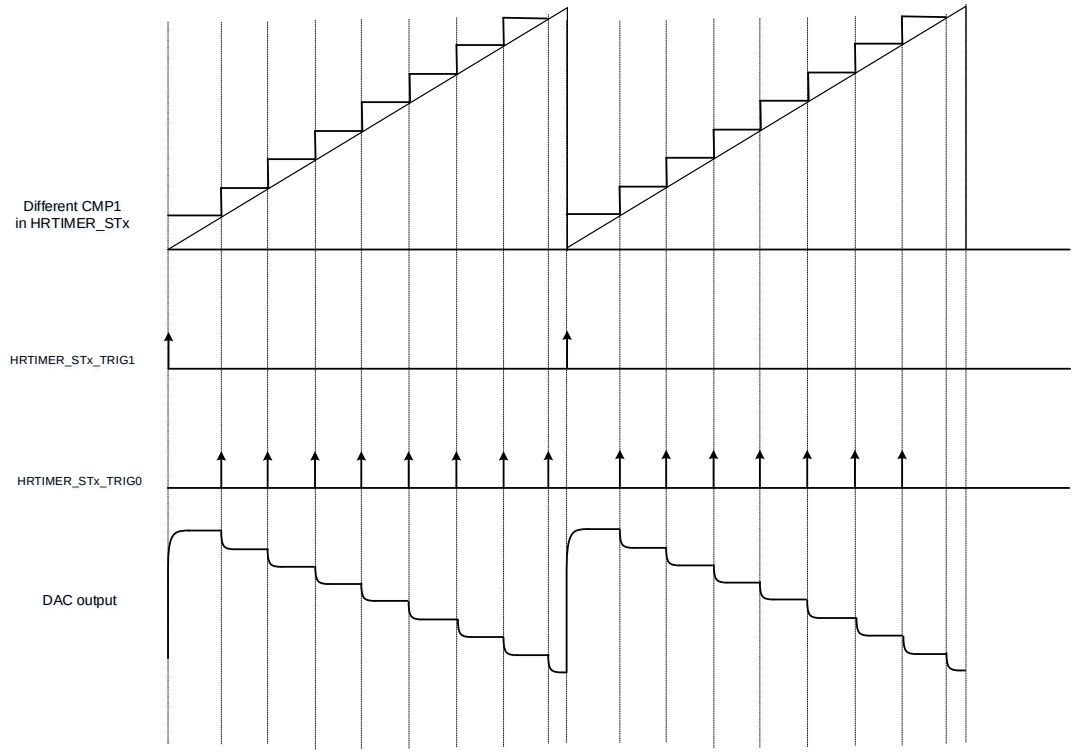
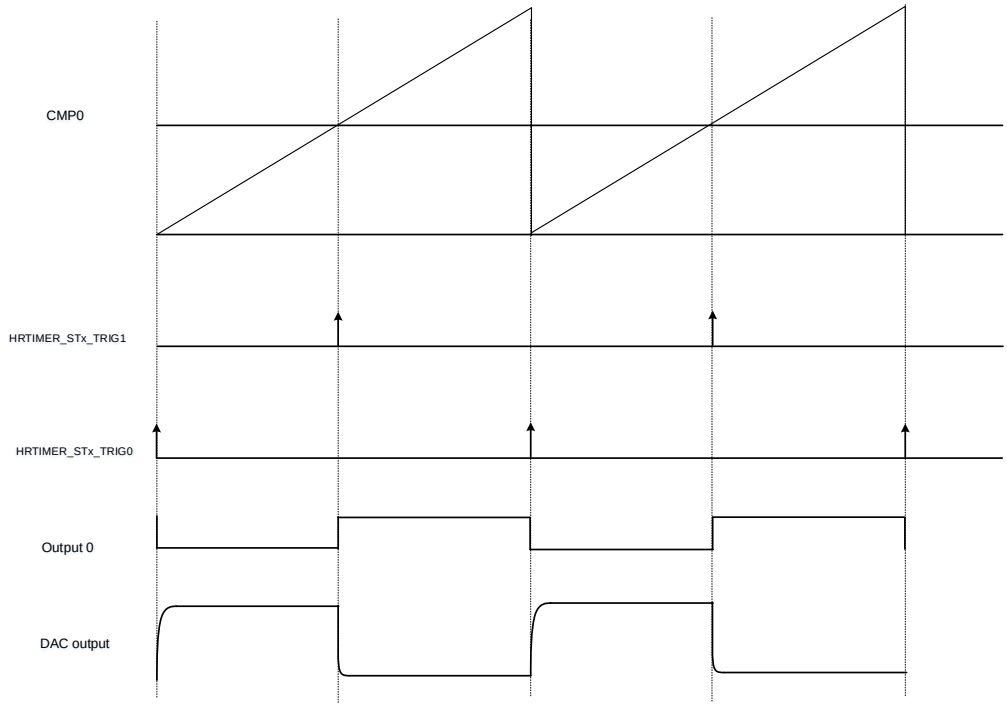


图 25-59. 当 TRIG0M = 1 和 TRIG1M = 1 时 DAC 触发



25.4.11. 中断

大多数事件可以生成中断请求，所有的中断请求可分组到 10 个中断向量(HRTIMER_IRQy，y=0..9)。详见*表 25-24. 中断映射*。

表 25-24. 中断映射

中断号	事件	Control bit
Master_TIMER：HRTIMER_IRQ0	更新事件	HRTIMER_MTDMAINTEN 中的 UPIE 位
	同步输入事件	HRTIMER_MTDMAINTEN 中的 SYNIIE 位
	重复事件	HRTIMER_MTDMAINTEN 中的 REPIE 位
	比较 0 事件	HRTIMER_MTDMAINTEN 中的 CMP0IE 位
	比较 1 事件	HRTIMER_MTDMAINTEN 中的 CMP1IE 位
	比较 2 事件	HRTIMER_MTDMAINTEN 中的 CMP2IE 位
	比较 3 事件	HRTIMER_MTDMAINTEN 中的 CMP3IE 位
Slave_TIMER0：HRTIMER_IRQ1 Slave_TIMER1：HRTIMER_IRQ2 Slave_TIMER2：	延迟空闲模式进入	HRTIMER_STxDMAINTEN 中的 DLYIIE 位
	计数器复位事件	HRTIMER_STxDMAINTEN 中的 RSTIE 位
	C1OPRE 从有效到无效	HRTIMER_STxDMAINTEN 中的 CH1ONAIE 位
	C1OPRE 从无效到有效	HRTIMER_STxDMAINTEN 中的 CH1OAIE 位

中断号	事件	Control bit
HRTIMER_IRQ3 Slave_TIMER3： HRTIMER_IRQ4 Slave_TIMER4： HRTIMER_IRQ5 Slave_TIMER5： HRTIMER_IRQ6 Slave_TIMER6： HRTIMER_IRQ7 Slave_TIMER7： HRTIMER_IRQ8	C0OPRE 从有效到无效	HRTIMER_STxDMAINTEN 中的 CH0ONAIE 位
	C0OPRE 从无效到有效	HRTIMER_STxDMAINTEN 中的 CH0OAIE 位
	捕获 1 事件	HRTIMER_STxDMAINTEN 中的 CAP1IE 位
	捕获 0 事件	HRTIMER_STxDMAINTEN 中的 CAP0IE 位
	更新事件	HRTIMER_STxDMAINTEN 中的 UPIE 位
	重复事件	HRTIMER_STxDMAINTEN 中的 REPIE 位
	比较 3 事件	HRTIMER_STxDMAINTEN 中的 CMP3IE 位
	比较 2 事件	HRTIMER_STxDMAINTEN 中的 CMP2IE 位
	比较 1 事件	HRTIMER_STxDMAINTEN 中的 CMP1IE 位
	比较 0 事件	HRTIMER_STxDMAINTEN 中的 CMP0IE 位
HRTIMER_IRQ0	突发模式周期事件	HRTIMER_INTEN 中的 BMPERIE 位
	DLL 校准完成	HRTIMER_INTEN 中的 DLLCALIE 位
HRTIMER_IRQ9	系统故障	HRTIMER_INTEN 中的 SYSFLTIE 位
	故障 x (x = 0..7)	HRTIMER_INTEN 中的 FLTxIE (x = 0..7)位

25.4.12. DMA 请求

大多数事件可以生成 DMA 请求，每个定时器对应一个 DMA 通道，详见*表 25-25. DMA 请求映射*。

表 25-25. DMA 请求映射

DMA channel	Event	Control bit
Master_TIMER：	更新事件	HRTIMER_MTDMAINTEN 中的 UPDEN 位
	同步输入事件	HRTIMER_MTDMAINTEN 中的 SYNIDEN 位
	重复事件	HRTIMER_MTDMAINTEN 中的 REPDEN 位
	比较 0 事件	HRTIMER_MTDMAINTEN 中的 CMP0DEN 位
	比较 1 事件	HRTIMER_MTDMAINTEN 中的 CMP1DEN 位
	比较 2 事件	HRTIMER_MTDMAINTEN 中的 CMP2DEN 位
	比较 3 事件	HRTIMER_MTDMAINTEN 中的 CMP3DEN 位
Slave_TIMER0： Slave_TIMER1： Slave_TIMER2： Slave_TIMER3： Slave_TIMER4： Slave_TIMER5: Slave_TIMER6: Slave_TIMER7:	延迟空闲模式进入	HRTIMER_STxDMAINTEN 中的 DLYIDEN 位
	计数器复位事件	HRTIMER_STxDMAINTEN 中的 RSTDEN 位
	C1OPRE 从有效到无效	HRTIMER_STxDMAINTEN 中的 CH1ONADEN 位
	C1OPRE 从无效到有效	HRTIMER_STxDMAINTEN 中的 CH1OADEN 位
	C0OPRE 从有效到无效	HRTIMER_STxDMAINTEN 中的 CH0ONADEN 位
	C0OPRE 从无效到有效	HRTIMER_STxDMAINTEN 中的 CH0OADEN 位
	捕获 1 事件	HRTIMER_STxDMAINTEN 中的 CAP1DEN 位
	捕获 0 事件	HRTIMER_STxDMAINTEN 中的 CAP0DEN 位
	更新事件	HRTIMER_STxDMAINTEN 中的 UPDEN 位
	重复事件	HRTIMER_STxDMAINTEN 中的 REPDEN 位

DMA channel	Event	Control bit
	比较 3 事件	HRTIMER_STxDMAINTEN 中的 CMP3DEN 位
	比较 2 事件	HRTIMER_STxDMAINTEN 中的 CMP2DEN 位
	比较 1 事件	HRTIMER_STxDMAINTEN 中的 CMP1DEN 位
	比较 0 事件	HRTIMER_STxDMAINTEN 中的 CMP0DEN 位

注意：必须先禁能 DMA 控制器，然后再禁能 DMA 请求。

25.4.13. DMA 模式

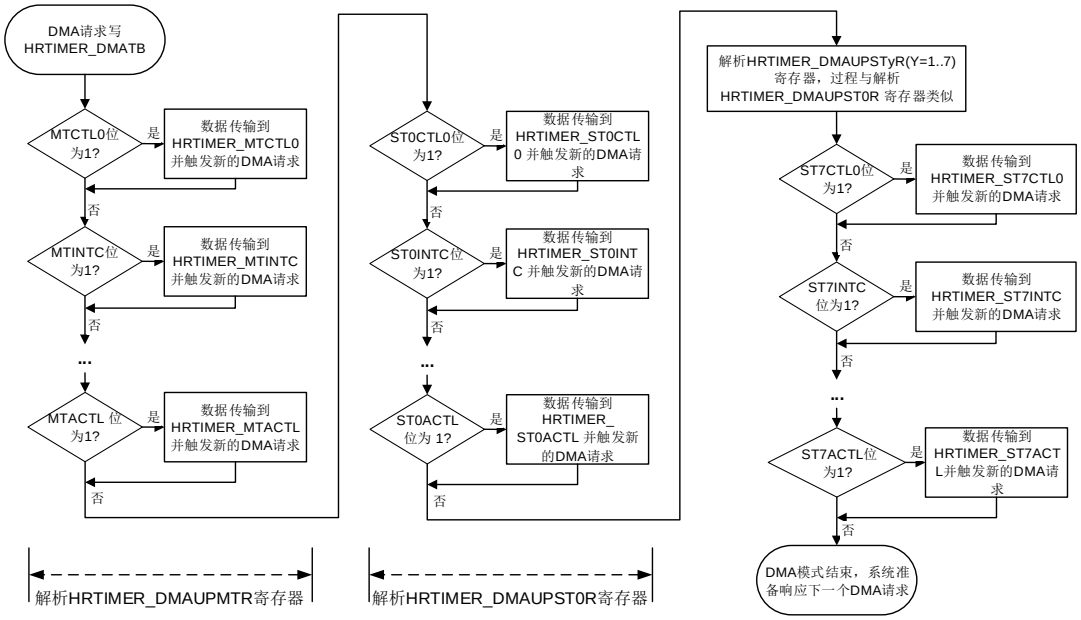
定时器的 DMA 模式是通过 DMA 模块，实现单个 DMA 请求配置 HRTIMER 的多个寄存器的功能。相关的寄存器（总共十个寄存器）如下：

- HRTIMER_DMAUPMTR：定义更新Master_TIMER中的哪些寄存器。Master_TIMER的大多数控制和数据寄存器都与一个选择位关联。如果该选择位置位，则写访问将重定向到关联的寄存器；
- HRTIMER_DMAUPSTxR(x=0..7)：定义更新Slave_TIMERx中的哪些寄存器。大多数Slave_TIMERx控制和数据寄存器都与一个选择位相关联。如果该选择位被置位，则写访问将重定向到关联的寄存器。
- HRTIMER_DMATB：DMA传输缓冲区寄存器。只需要将指向HRTIMER_DMATB寄存器的DMA模块作为目标，并禁能外设增量模式的外设配置。所有对该寄存器的写访问都将通过重定向机制在内部重新传输到最终目标寄存器。

DMA 模式是永久使能的（没有使能位）。通过对 HRTIMER_DMATB 寄存器的首次写访问来启动 DMA 操作。

发生 DMA 请求时，HRTIMER 会生成多个 32 位 DMA 请求并解析要更新的寄存器（在 HRTIMER_DMAUPMTR 和 HRTIMER_DMAUPSTxR 寄存器中定义）。如果选择位置 1，则写访问将重定向到关联的寄存器。如果选择位为 0，则跳过相关寄存器更新，并继续进行寄存器解析，直到检测到新的位置 1，触发新的 DMA 请求。9 个寄存器（HRTIMER_DMAUPMTR 和 HRTIMER_DMAUPSTxR 寄存器）全部解析后，DMA 模式完成，系统已准备好等待下一个 DMA 触发。若再有 DMA 请求事件发生，则 HRTIMER 将重复上述过程。详见*图 25-60. DMA 模式运行流程图*。

图 25-60. DMA 模式运行流程图



25.4.14. Debug 模式

当 Cortex®-M33 内核暂停时，DBG_CTL0 寄存器中的 HRTIMER_HOLD 位决定了计数器是否停止运行。

HRTIMER_HOLD = 0

若 HRTIMER_HOLD = 0，则 HRTIMER 继续正常运行。

HRTIMER_HOLD = 1

若 HRTIMER_HOLD = 1，则将停止 Master_TIMER 和所有 Slave_TIMERx 中的计数器。

如果 CHyFLTOS[1:0] = 2'b01、2'b10、2'b11，则输出进入 FAULT 状态。可以通过将 HRTIMER_CHOUTEN 寄存器中的 STxCHyEN 位置 1，清零 HRTIMER_HOLD 位来再次使能输出。如果 CHyFLTOS [1:0] = 2'b00，则输出保持其当前状态。退出调试模式时，输出将返回其原始状态。

所有计数器的复位/启动和捕获触发功能都禁能。除 ADC 触发外，所有外部事件的触发都被禁能。更新事件将被丢弃。突发模式电路被冻结：触发都被忽略，突发模式计数器停止。

DLL 校准正常运行。在运行模式下驱动正常输出的单元不受调试影响，例如死区时间单元，载波信号和置位/复位交叉开关等。

25.5. HRTIMER 寄存器

HRTIMER 基地址：0x4001 5800

寄存器进行分段寻址：

HRTIMER Master_TIMER 寄存器基地址：0x4001 5800

HRTIMER Slave_TIMER0 寄存器基地址：0x4001 5880

HRTIMER Slave_TIMER1 寄存器基地址：0x4001 5900

HRTIMER Slave_TIMER2 寄存器基地址：0x4001 5980

HRTIMER Slave_TIMER3 寄存器基地址：0x4001 5A00

HRTIMER Slave_TIMER4 寄存器基地址：0x4001 5A80

HRTIMER Slave_TIMER5 寄存器基地址：0x4001 5B00

HRTIMER Slave_TIMER6 寄存器基地址：0x4001 6000

HRTIMER Slave_TIMER7 寄存器基地址：0x4001 6080

HRTIMER 通用寄存器基地址：0x4001 5B80

25.5.1. Master_TIMER 寄存器

HRTIMER Master_TIMER 寄存器基地址：0x4001 5800

Master_TIMER 控制寄存器 0 (HRTIMER_MTCTL0)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UPSEL[1:0]		UPREP	保留	SHWEN	DACTRGS[1:0]		ST7CEN	ST6CEN	ST5CEN	ST4CEN	ST3CEN	ST2CEN	ST1CEN	ST0CEN	MTCEN
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
SYNOSRC[1:0]		SYNOPLS[1:0]		SYNISTRT	SYNIRST	SYNISRC[1:0]		ALTM[1:0]		HALFM	CNTRSTM	CTNM	CNTCKDIV[2:0]		
rw		rw		rw	rw	rw		rw		rw	rw	rw	rw		

位/位域	名称	描述
31:30	UPSEL[1:0]	更新事件选择

		该位域用于配置更新事件与 DMA 模式的关系。 00：更新事件产生与 DMA 模式无关。 01：在 DMA 模式下完成 DMA 传输时生成更新事件。 10：在 DMA 模式下 DMA 传输完成后，计数器翻转产生更新事件。仅适用于连续模式。 11：保留。
29	UPREP	重复事件生成更新事件 该位用于使能重复事件生成更新事件。 0：重复事件生成更新事件禁能 1：重复事件生成更新事件使能 **注意：**仅当 UPSEL [1:0] = 2'b00 或 2'b01 时才能设置 UPREP。
28	保留	必须保持复位值
27	SHWEN	影子寄存器使能 0：影子寄存器禁能 1：影子寄存器使能
26:25	DACTRGS[1:0]	DAC 触发源 发生更新事件时，定时器生成 DAC 触发事件。该位域用于配置哪个触发源生成 DAC 触发事件。 00：不生成 DAC 触发事件 01：在 HRTIMER_DACTRIG0 上生成 DAC 触发事件 10：在 HRTIMER_DACTRIG1 上生成 DAC 触发事件 11：在 HRTIMER_DACTRIG2 上生成 DAC 触发事件
24	ST7CEN	Slave_TIMER7 计数器使能 0：Slave_TIMER7 计数器禁能 1：Slave_TIMER7 计数器使能 **注意：**不得在小于 8 个 $t_{HRTIMER_CK}$ 时钟周期内修改该位。
23	ST6CEN	Slave_TIMER6 计数器使能 0：Slave_TIMER6 计数器禁能 1：Slave_TIMER6 计数器使能 **注意：**不得在小于 8 个 $t_{HRTIMER_CK}$ 时钟周期内修改该位。
22	ST5CEN	Slave_TIMER5 计数器使能 0：Slave_TIMER5 计数器禁能 1：Slave_TIMER5 计数器使能 **注意：**不得在小于 8 个 $t_{HRTIMER_CK}$ 时钟周期内修改该位。
21	ST4CEN	Slave_TIMER4 计数器使能 0：Slave_TIMER4 计数器禁能

		1：Slave_TIMER4 计数器使能 **注意：**不得在小于 8 个 $t_{HRTIMER_CK}$ 时钟周期内修改该位。
20	ST3CEN	Slave_TIMER3 计数器使能 0：Slave_TIMER3 计数器禁能 1：Slave_TIMER3 计数器使能 **注意：**不得在小于 8 个 $t_{HRTIMER_CK}$ 时钟周期内修改该位。
19	ST2CEN	Slave_TIMER2 计数器使能 0：Slave_TIMER2 计数器禁能 1：Slave_TIMER2 计数器使能 **注意：**不得在小于 8 个 $t_{HRTIMER_CK}$ 时钟周期内修改该位。
18	ST1CEN	Slave_TIMER1 计数器使能 0：Slave_TIMER1 计数器禁能 1：Slave_TIMER1 计数器使能 **注意：**不得在小于 8 个 $t_{HRTIMER_CK}$ 时钟周期内修改该位。
17	ST0CEN	Slave_TIMER0 计数器使能 0：Slave_TIMER0 计数器禁能 1：Slave_TIMER0 计数器使能 **注意：**不得在小于 8 个 $t_{HRTIMER_CK}$ 时钟周期内修改该位。
16	MTCEN	Master_TIMER 计数器使能 0：Master_TIMER 计数器禁能 1：Master_TIMER 计数器使能 **注意：**不得在小于 8 个 $t_{HRTIMER_CK}$ 时钟周期内修改该位。
15:14	SYNOSRC[1:0]	同步输出源 该位域用于配置发送到同步输出 HRTIMER_SCOUT 上的事件。 00：Master_TIMER 启动事件。 01：Master_TIMER 比较 0 事件 10：Slave_TIMER0 复位和启动事件 11：Slave_TIMER0 比较 0 事件
13:12	SYNOPLS[1:0]	同步输出脉冲 该位域用于配置同步输出 HRTIMER_SCOUT 上的脉冲。 00：脉冲生成禁能。HRTIMER_SCOUT 上无脉冲。 01：保留。 10：在 HRTIMER_SCOUT 上产生正脉冲。它的长度是 16 个 $t_{HRTIMER_CK}$ 个周期。 11：在 HRTIMER_SCOUT 上产生负脉冲。它的长度是 16 个 $t_{HRTIMER_CK}$ 个周期。
11	SYNISTRT	同步输入启动计数器 该位用于配置同步输入启动计数器。

		0：同步输入不能启动计数器。 1：同步输入可以启动计数器。
10	SYNIRST	同步输入复位计数器 该位用于配置同步输入复位计数器。 0：同步输入不能复位计数器。 1：同步输入可以复位计数器。
9:8	SYNISRC[1:0]	同步输入源 该位域用于配置同步输入源。 00：同步输入禁能。 01：保留。 10：内部信号：高级定时器 TIMER0 中的 TIMER0_TRGO。 11：外部信号：HRTIMER_SCIN 引脚上的正脉冲。 **注意：**相应的计时器使能后，将无法修改此位字段
7:6	ALTM[1:0]	交错模式 此位字段仅在 HALFM 位被复位时具有重要意义。它启用交错模式。 00: 禁用交错模式 01: 三重交错模式：当写入 HRTIMER_MTCAR 寄存器时，HRTIMER_MTCMP0V 活动寄存器将自动更新为 HRTIMER_MTCAR/3 的值，HRTIMER_MTCMP1V 活动寄存器将自动更新为 2x(HRTIMER_MTCAR/3) 的值。 10: 四重交错模式：当写入 HRTIMER_MTCAR 寄存器时，HRTIMER_MTCMP0V 活动寄存器将自动更新为 HRTIMER_MTCAR/4 的值，HRTIMER_MTCMP1V 活动寄存器将自动更新为 HRTIMER_MTCAR/2 的值，HRTIMER_MTCMP2V 活动寄存器将自动更新为 3x(HRTIMER_MTCAR/4) 的值。 11: 禁用交错模式
5	HALFM	半波模式 该位置 1 时，HRTIMER_MTCMP0V 有效寄存器始终是计数器自动重载值（HRTIMER_MTCAR）的一半。 0：半波模式禁能。 1：半波模式使能。
4	CNTRSTM	计数器复位模式 该位用于定义单脉冲模式下计数器的行为。 0：计数器只能计数到周期值后才能复位。 1：可以随时复位计数器（运行或停止）。
3	CTNM	连续模式。 0：单脉冲模式。当计数器达到 HRTIMER_MTCAR 值时，它将由硬件停止。 1：连续模式。计数器在达到 HRTIMER_MTCAR 值时，翻转到 0 并连续计数。
2:0	CNTCKDIV[2:0]	计数器时钟分频

该位域可以由软件配置，确定高分辨率时钟（HRTIMER_HPCK）和计数器时钟（HRTIMER_PSCCK）的分频比。

$f_{HRTIMER_PSCCK} = f_{HRTIMER_HPCK} / 2^{CNTCKDIV[2:0]}$。

000：$f_{HRTIMER_PSSCK} = f_{HPTIM_HPCK}$

001：$f_{HRTIMER_PSSCK} = f_{HPTIM_HPCK} / 2$

010：$f_{HRTIMER_PSSCK} = f_{HPTIM_HPCK} / 4$

011：$f_{HRTIMER_PSSCK} = f_{HPTIM_HPCK} / 8$

100：$f_{HRTIMER_PSSCK} = f_{HPTIM_HPCK} / 16$

101：$f_{HRTIMER_PSSCK} = f_{HPTIM_HPCK} / 32$

110：$f_{HRTIMER_PSSCK} = f_{HPTIM_HPCK} / 64$

111：$f_{HRTIMER_PSSCK} = f_{HPTIM_HPCK} / 128$

其他值保留。

注意：一旦使能定时器，就不能修改 CNTCKDIV [2:0]位域。

Master_TIMER 中断标志寄存器(HRTIMER_MTINTF)

地址偏移：0x04

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									UPIF	SYNIIF	REPIF	CMP3IF	CMP2IF	CMP1IF	CMP0IF
									r	r	r	r	r	r	r

位/位域	名称	描述
31:7	保留	必须保持复位值
6	UPIF	更新中断标志 发生更新事件时，此标志由硬件置位。 0：更新中断未发生 1：更新中断发生
5	SYNIIF	同步输入中断标志 同步输入到来时，此标志由硬件置位。 0：同步输入未发生 1：同步输入发生
4	REPIF	重复中断标志 发生重复事件时，此标志由硬件置位。

0：重复中断未发生
1：重复中断发生

3	CMP3IF	比较 3 中断标志 当发生比较 3 事件时，此标志由硬件置位。 0：比较 3 中断未发生 1：比较 3 中断发生
2	CMP2IF	比较 2 中断标志 当发生比较 2 事件时，此标志由硬件置位。 0：比较 2 中断未发生 1：比较 2 中断发生
1	CMP1IF	比较 1 中断标志 当发生比较 3 事件时，此标志由硬件置位。 0：比较 1 中断未发生 1：比较 1 中断发生
0	CMP0IF	比较 0 中断标志 当发生比较 0 事件时，此标志由硬件置位。 0：比较 0 中断未发生 1：比较 0 中断发生

Master_TIMER 中断标志清除寄存器(HRTIMER_MTINTC)

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									UPIFC	SYNIIFC	REPIFC	CMP3IFC	CMP2IFC	CMP1IFC	CMP0IFC
									w	w	w	w	w	w	w

位/位域	名称	描述
31:7	保留	必须保持复位值
6	UPIFC	更新中断标志清除 0：没有影响 1：清除更新中断标志
5	SYNIIFC	同步输入中断标志清除

0：没有影响
1：清除同步输入中断标志

4	REPIFC	重复中断标志清除 0：没有影响 1：清除重复中断标志
3	CMP3IFC	比较 3 中断标志清除 0：没有影响 1：清除比较 3 中断标志
2	CMP2IFC	比较 2 中断标志清除 0：没有影响 1：清除比较 2 中断标志
1	CMP1IFC	比较 1 中断标志清除 0：没有影响 1：清除比较 1 中断标志
0	CMP0IFC	比较 0 中断标志清除 0：没有影响 1：清除比较 0 中断标志

Master_TIMER DMA 和中断使能寄存器 (HRTIMER_MTDMAINTEN)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留									UPDEN	SYNIDEN	REPDEN	CMP3DEN	CMP2DEN	CMP1DEN	CMP0DEN
									rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									UPIE	SYNIIE	REPIE	CMP3IE	CMP2IE	CMP1IE	CMP0IE
									rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:23	保留	必须保持复位值
22	UPDEN	更新 DMA 请求使能 0：禁能 1：使能

21	SYNIDEN	同步输入 DMA 请求使能 0：禁能 1：使能
20	REPDEN	重复 DMA 请求使能 0：禁能 1：使能
19	CMP3DEN	比较 3 DMA 请求使能 0：禁能 1：使能
18	CMP2DEN	比较 2 DMA 请求使能 0：禁能 1：使能
17	CMP1DEN	比较 1 DMA 请求使能 0：禁能 1：使能
16	CMP0DEN	比较 0 DMA 请求使能 0：禁能 1：使能
15:7	保留	必须保持复位值
6	UPIE	更新中断使能 0：禁能 1：使能
5	SYNIIE	同步输入中断使能 0：禁能 1：使能
4	REPIE	重复中断使能 0：禁能 1：使能
3	CMP3IE	比较 3 中断使能 0：禁能 1：使能
2	CMP2IE	比较 2 中断使能 0：禁能 1：使能

1	CMP1IE	比较 1 中断使能 0：禁能 1：使能
0	CMP0IE	比较 0 中断使能 0：禁能 1：使能

Master_TIMER 计数器寄存器(HRTIMER_MTCNT)

地址偏移：0x10

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

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]	当前计数器值。对该位域进行写操作可以更改计数器的值。 仅当 Master_TIMER 停止（HRTIMER_MTCTL0 寄存器中的 MTCEN = 0）时，对其进行写操作才能更改计数器的值。 **注意：** （1）计数器时钟分频系数小于 32（CNTCKDIV [2:0] < 5）时，计数器的最低有效位无效，它们不能被写入，读出值为 0。 （2）如果写入该位域的值高于 HRTIM_MPER 寄存器值，则定时器的行为是不可预测的。

Master_TIMER 计数器自动重载寄存器(HRTIMER_MTCAR)

地址偏移：0x14

复位值：0x0000 FFDF

该寄存器只能进行字（32 位）访问。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CARL[15:0]	计数器自动重载值 该位域定义了计数器的自动重载值。该寄存器具有影子寄存器。如果影子寄存器被禁能（SHWEN = 0），它将保存有效寄存器的内容；否则，它将保存影子寄存器的内容。 **注意：** （1）最小值必须大于或等于（3*$t_{HRTIMER_CK}$）。例如：当 CNTCKDIV [2:0] = 3'b000 时，CARL [15:0] > = 0x60。 （2）最大值必须小于或等于（0xFFFF － 1*$t_{HRTIMER_CK}$）。例如：当 CNTCKDIV [2:0] = 3'b000 时，CARL [15:0] <= 0xFFDF。

Master_TIMER 重复计数寄存器(HRTIMER_MTCREP)

地址偏移：0x18

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

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

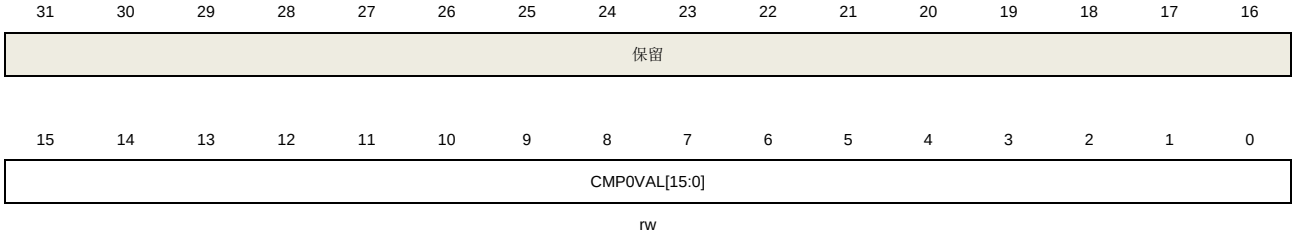
位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	CREP[7:0]	重复计数器值 该位域用于定义重复事件的发生率。当重复计数器递减计数到零时，连续模式下即将发生的翻转事件或复位事件将产生一个重复事件。 该寄存器具有影子寄存器。如果影子寄存器被禁能（SHWEN = 0），它将保存有效寄存器的内容；否则，它将保存影子寄存器的内容。

Master_TIMER 比较 0 寄存器(HRTIMER_MTCMP0V)

地址偏移：0x1C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。



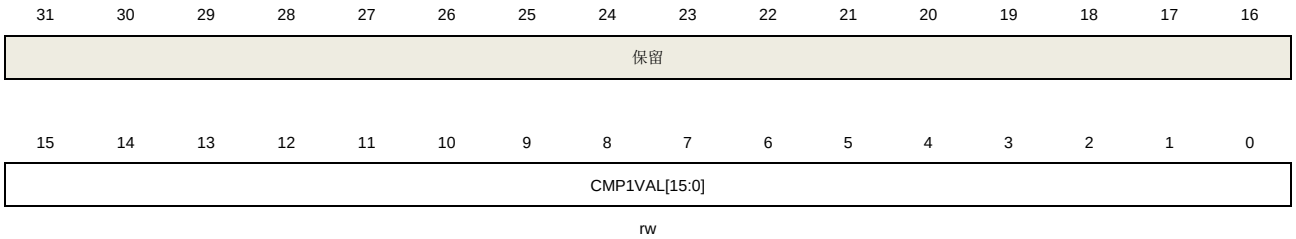
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CMP0VAL[15:0]	比较 0 值 该位域用于配置与计数器进行比较的值。 该寄存器具有影子寄存器。如果影子寄存器被禁能（SHWEN = 0），它将保存有效寄存器的内容；否则，它将保存影子寄存器的内容。 **注意：** （1）最小值必须大于或等于 3 个 $t_{HRTIMER_CK}$。例如：当 CNTCKDIV [2:0] = 3'b000 时，CARL [15:0] > = 0x60。 （2）最大值必须小于或等于（0xFFFF －1*$t_{HRTIMER_CK}$）。例如：当 CNTCKDIV [2:0] = 3'b000 时，CARL [15:0] <= 0xFFDF。

Master_TIMER 比较 1 寄存器(HRTIMER_MTCMP1V)

地址偏移：0x24

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。



位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CMP1VAL[15:0]	比较 1 值 该位域用于配置与计数器进行比较的值。 该寄存器具有影子寄存器。如果影子寄存器被禁能（SHWEN = 0），它将保存有效

寄存器的内容；否则，它将保存影子寄存器的内容。

注意：

（1）最小值必须大于或等于 3 个 $t_{HRTIMER_CK}$。例如：当 CNTCKDIV [2:0] = 3'b000 时，CARL [15:0] > = 0x60。

（2）最大值必须小于或等于（0xFFFF －1*$t_{HRTIMER_CK}$）。例如：当 CNTCKDIV [2:0] = 3'b000 时，CARL [15:0] <= 0xFFDF。

Master_TIMER 比较 2 寄存器(HRTIMER_MTCMP2V)

地址偏移：0x28

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP2VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CMP2VAL[15:0]	比较 2 值 该位域用于配置与计数器进行比较的值。 该寄存器具有影子寄存器。如果影子寄存器被禁能（SHWEN = 0），它将保存有效寄存器的内容；否则，它将保存影子寄存器的内容。 **注意：** （1）最小值必须大于或等于 3 个 $t_{HRTIMER_CK}$。例如：当 CNTCKDIV [2:0] = 3'b000 时，CARL [15:0] > = 0x60。 （2）最大值必须小于或等于（0xFFFF －1*$t_{HRTIMER_CK}$）。例如：当 CNTCKDIV [2:0] = 3'b000 时，CARL [15:0] <= 0xFFDF。

Master_TIMER 比较 3 寄存器(HRTIMER_MTCMP3V)

地址偏移：0x2C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP3VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CMP3VAL[15:0]	比较 3 值 该位域用于配置与计数器进行比较的值。 该寄存器具有影子寄存器。如果影子寄存器被禁能（SHWEN = 0），它将保存有效寄存器的内容；否则，它将保存影子寄存器的内容。 **注意：** （1）最小值必须大于或等于 3 个 $t_{HRTIMER_CK}$。例如：当 CNTCKDIV [2:0] = 3'b000 时，CARL [15:0] > = 0x60。 （2）最大值必须小于或等于（0xFFFF －1*$t_{HRTIMER_CK}$）。例如：当 CNTCKDIV [2:0] = 3'b000 时，CARL [15:0] <= 0xFFDF。

25.5.2. Slave_TIMERx 寄存器(x=0..7)

HRTIMER Slave_TIMER0 寄存器基地址：0x4001 5880

HRTIMER Slave_TIMER1 寄存器基地址：0x4001 5900

HRTIMER Slave_TIMER2 寄存器基地址：0x4001 5980

HRTIMER Slave_TIMER3 寄存器基地址：0x4001 5A00

HRTIMER Slave_TIMER4 寄存器基地址：0x4001 5A80

HRTIMER Slave_TIMER5 寄存器基地址：0x4001 5B00

HRTIMER Slave_TIMER6 寄存器基地址：0x4001 6000

HRTIMER Slave_TIMER7 寄存器基地址：0x4001 6080

Slave_TIMERx 控制寄存器 0 (HRTIMER_STxCTL0)

地址偏移：0x00

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UPSEL[3:0]				SHWEN	DACTRGS[1:0]		UPBMT	UPBST4	UPBST3	UPBST2	UPBST1	UPBST0	UPRST	UPREP	UPBST5
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

DELCMP3M[1:0]	DELCMP1M[1:0]	SYNISTRT	SYNIRST	RSYNUPD	ALTM[1:0]	BLNMEN	HALFM	CNTRSTM	CTNM	CNTCKDIV[2:0]
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:28	UPSEL[3:0]	更新事件选择 该位域用于配置更新事件与 DMA 模式的关系。 0000：更新事件的生成独立于 DMA 模式。 0001：在 DMA 模式下完成 DMA 传输时生成更新事件。 0010：在 DMA 模式下完成 DMA 传输后的更新事件，生成更新事件。 0011：在 STxUPIN0 的上升沿生成更新事件。 0100：在 STxUPIN1 的上升沿生成更新事件。 0101：在 STxUPIN2 的上升沿生成更新事件。 0110：在 STxUPIN0 的上升沿到来之后的更新事件，生成更新事件。 0111：在 STxUPIN1 的上升沿到来之后的更新事件，生成更新事件。 1000：在 STxUPIN2 的上升沿到来之后的更新事件，生成更新事件。 其他值保留。 **注意：** （1）在写入新值之前，必须先复位该位域。 （2）当 UPSEL [3:0] = 4'b0001，4'b0011，4'b0100 和 4'b0101 时，可以有多个并发更新源。例如，通过 Master_TIMER（UPBMT = 1）和 DMA 模式进行更新。
27	SHWEN	影子寄存器使能 0：影子寄存器禁能 1：影子寄存器使能
26:25	DACTRGS[1:0]	DAC 触发源 发生更新事件时，定时器生成 DAC 触发事件。该位域用于配置哪个触发源生成 DAC 触发事件。 00：不生成 DAC 触发事件 01：在 HRTIMER_DACTRIG0 上生成 DAC 触发事件 10：在 HRTIMER_DACTRIG1 上生成 DAC 触发事件 11：在 HRTIMER_DACTRIG2 上生成 DAC 触发事件
24	UPBMT	通过 Master_TIMER 更新事件进行更新 该位置 1 时，Slave_TIMERx（x = 0..7）更新事件与 Master_TIMER 更新事件同步，且它们的有效寄存器由 Master_TIMER 更新事件进行更新。 0：有效寄存器不由 Master_TIMER 更新。 1：有效寄存器由 Master_TIMER 更新。
23	UPBST4	通过 Slave_TIMER4 更新事件进行更新 该位置 1 时，Slave_TIMERx（x=0..3,5..7）更新事件与 Slave_TIMER4 更新事件同

步，且它们的有效寄存器由 Slave_TIMER4 更新事件进行更新。
0：有效寄存器不由 Slave_TIMER4 更新。
1：有效寄存器由 Slave_TIMER4 更新。
注意：Slave_TIMER4 的寄存器中不存在此位。

22	UPBST3	通过 Slave_TIMER3 更新事件进行更新 该位置 1 时，Slave_TIMERx（x=0..2,4..7）更新事件与 Slave_TIMER3 更新事件同步，且它们的有效寄存器由 Slave_TIMER3 更新事件进行更新。 0：有效寄存器不由 Slave_TIMER3 更新。 1：有效寄存器由 Slave_TIMER3 更新。 **注意：**Slave_TIMER3 的寄存器中不存在此位。
21	UPBST2	通过 Slave_TIMER2 更新事件进行更新 该位置 1 时，Slave_TIMERx（x=0,1,3..7）更新事件与 Slave_TIMER2 更新事件同步，且它们的有效寄存器由 Slave_TIMER2 更新事件进行更新。 0：有效寄存器不由 Slave_TIMER2 更新。 1：有效寄存器由 Slave_TIMER2 更新。 **注意：**Slave_TIMER2 的寄存器中不存在此位。
20	UPBST1	通过 Slave_TIMER1 更新事件进行更新 该位置 1 时，Slave_TIMERx（x = 0,2..7）更新事件与 Slave_TIMER1 更新事件同步，且它们的有效寄存器由 Slave_TIMER1 更新事件进行更新。 0：有效寄存器不由 Slave_TIMER1 更新。 1：有效寄存器由 Slave_TIMER1 更新。 **注意：**Slave_TIMER1 的寄存器中不存在此位。
19	UPBST0	通过 Slave_TIMER0 更新事件进行更新 该位置 1 时，Slave_TIMERx（x = 1..7）更新事件与 Slave_TIMER0 更新事件同步，且它们的有效寄存器由 Slave_TIMER0 更新事件进行更新。 0：有效寄存器不由 Slave_TIMER0 更新。 1：有效寄存器由 Slave_TIMER0 更新。 **注意：**Slave_TIMER0 的寄存器中不存在此位。
18	UPRST	更新事件由复位事件生成 该位用于使能计数器复位事件或翻转事件生成更新事件。 0：复位事件或翻转事件生成更新事件禁能 1：复位事件或翻转事件生成更新事件使能
17	UPREP	更新事件由重复事件生成 该位用于使能重复事件生成更新事件。 0：重复事件生成更新事件禁能 1：重复事件生成更新事件使能
16	UPBST5	通过 Slave_TIMER5 更新事件进行更新

		该位置 1 时，Slave_TIMERx（x=0..4,6,7）更新事件与 Slave_TIMER5 更新事件同步，且它们的有效寄存器由 Slave_TIMER4 更新事件进行更新。 0：有效寄存器不由 Slave_TIMER5 更新。 1：有效寄存器由 Slave_TIMER5 更新。 **注意：**Slave_TIMER5 的寄存器中不存在此位。
15:14	DELCMP3M[1:0]	比较 3 延迟模式 00：比较 3 延迟模式禁能。只要计数器值等于比较 3 有效寄存器值，就会发生比较匹配。 01：比较 3 延迟模式 0。捕获 1 事件后，比较 3 的重新计算值为：（比较 3 有效寄存器值+捕获 1 值）。一旦计数器等于重新计算的值，就会发生比较匹配。 10：比较 3 延迟模式 1。在捕获 1 事件或比较 0 事件之后，比较 3 的重新计算值是：（比较 3 有效寄存器值+捕获 1 值，或者比较 3 有效寄存器值+比较 0 值）。一旦计数器等于重新计算的值，就会发生比较匹配。 11：比较 3 延迟模式 2。在捕获 1 事件或比较 2 事件之后，比较 3 的重新计算值为：（比较 3 有效寄存器值+捕获 1 的值，或比较 3 有效寄存器值+比较 2 的值）。一旦计数器等于重新计算的值，就会发生比较匹配。 **注意：**一旦使能计数器（HRTIMER_MTCTL0 寄存器中的 STxCEN = 1），就不得修改此位域。
13:12	DELCMP1M[1:0]	比较 1 延迟模式 00：比较 1 延迟模式禁能。只要计数器值等于比较 1 有效寄存器值，就会发生比较匹配。 01：比较 1 延迟模式 0。捕获 0 事件后，比较 1 的重新计算值为：（比较 1 有效寄存器值+捕获 0 值）。一旦计数器等于重新计算的值，就会发生比较匹配。 10：比较 1 延迟模式 1。在捕获 0 事件或比较 0 事件之后，比较 1 的重新计算值是：（比较 1 有效寄存器值+捕获 0 值，或者比较 1 有效寄存器值+比较 0 值）。一旦计数器等于重新计算的值，就会发生比较匹配。 11：比较 1 延迟模式 2。在捕获 0 事件或比较 2 事件之后，比较 1 的重新计算值为：（比较 1 有效寄存器值+捕获 0 的值，或比较 1 有效寄存器值+比较 2 的值）。一旦计数器等于重新计算的值，就会发生比较匹配。 **注意：**一旦使能计数器（HRTIMER_MTCTL0 寄存器中的 STxCEN = 1），就不得修改此位域。
11	SYNISTRT	同步输入启动计数器 该位用于配置同步输入启动计数器。 0：同步输入不能启动计数器。 1：同步输入可以启动计数器。
10	SYNIRST	同步输入复位计数器 该位用于配置同步输入复位计数器。 0：同步输入不能复位计数器。

		1：同步输入可以复位计数器。
9	RSYNUPD	重新同步更新 此位指定来自定时单元外部的更新源是否必须同步： 0: 来自其他定时器（当 UPBMT、UPBST0、UPBST1、UPBST2、UPBST3、UPBST4、UPBST5、UPBST6、UPBST7 位置 1 时）或软件更新（STxSUP 位）的更新立即生效。 1: 来自其他定时器（当 UPBMT、UPBST0、UPBST1、UPBST2、UPBST3、UPBST4、UPBST5、UPBST6、UPBST7 位置 1 时）或软件更新（STxSUP 位）的更新将在产生复位/翻转事件生效。 注意：当 UPSEL [3:0] = 0000 时，此位才具有意义，否则将被忽略。
8:7	ALTM[1:0]	交错模式 此位字段仅在 HALFM 位被复位时具有重要意义。它启用交错模式。 00: 禁用交错模式 01: 三重交错模式：当写入 HRTIMER_STxCAR 寄存器时，HRTIMER_STxCMP0V 活动寄存器将自动更新为 HRTIMER_ STxCAR/3 的值，HRTIMER_STxCMP1V 活动寄存器将自动更新为 2x(HRTIMER_STxCAR/3) 的值。 10: 四重交错模式：当写入 HRTIMER_STxCAR 寄存器时，HRTIMER_STxCMP0V 活动寄存器将自动更新为 HRTIMER_STxCAR/4 的值，HRTIMER_STxCMP1V 活动寄存器将自动更新为 HRTIMER_STxCAR/2 的值，HRTIMER_STxCMP2V 活动寄存器将自动更新为 3x(HRTIMER_STxCAR/4) 的值。 11: 禁用交错模式
6	BLNMEN	均衡模式使能 0：均衡模式禁能 1：均衡模式使能 **注意：**一旦使能计数器（HRTIMER_MTCTL0 寄存器中的 STxCEN = 1），就不得修改此位域。
5	HALFM	半波模式 该位置 1 时，HRTIMER_STxCMP0V 有效寄存器始终是计数器自动重载值（HRTIMER_STxCAR）的一半。 0：半波模式禁能。 1：半波模式使能。
4	CNTRSTM	计数器复位模式 该位用于定义单脉冲模式下定时器计数器的行为。 0：计数器只能计数到周期值后才能复位 1：可以随时复位计数器（运行或停止）。
3	CTNM	连续模式。 0：单脉冲模式。当计数器达到 HRTIMER_STxCAR 值时，它将由硬件停止。

1：连续模式。计数器在达到HRTIMER_STxCAR值时，翻转到0并连续计数。

2:0	CNTCKDIV[2:0]	计数器时钟分频 该位域可以由软件配置，确定高分辨率时钟（HRTIMER_HPCK）和计数器时钟（HRTIMER_PSCCK）的分频比。 $f_{HRTIMER_PSCCK} = f_{HRTIMER_HPCK} / 2^{CNTCKDIV[2:0]}$。 000：$f_{HRTIMER_PSSCK} = f_{HPTIM_HPCK}$ 001：$f_{HRTIMER_PSSCK} = f_{HPTIM_HPCK} / 2$ 010：$f_{HRTIMER_PSSCK} = f_{HPTIM_HPCK} / 4$ 011：$f_{HRTIMER_PSSCK} = f_{HPTIM_HPCK} / 8$ 100：$f_{HRTIMER_PSSCK} = f_{HPTIM_HPCK} / 16$ 101：$f_{HRTIMER_PSSCK} = f_{HPTIM_HPCK} / 32$ 110：$f_{HRTIMER_PSSCK} = f_{HPTIM_HPCK} / 64$ 111：$f_{HRTIMER_PSSCK} = f_{HPTIM_HPCK} / 128$ 其他值保留。 **注意：**一旦使能定时器（HRTIMER_MTCTL0 寄存器中的 STxCEN =1），就不能修改 CNTCKDIV [2:0]位域。

Slave_TIMERx 中断标志寄存器 (HRTIMER_STxINTF)

地址偏移：0x04

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIR	保留									CH1F	CH0F	CH1SF	CH0SF	BLNIF	CBLNF
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	DLYIIF	RSTIF	CH1ONAIF	CH1OAIF	CH0ONAIF	CH0OAIF	CAP1IF	CAP0IF	UPIF	保留	REPIF	CMP3IF	CMP2IF	CMP1IF	CMP0IF
	r	r	r	r	r	r	r	r	r		r	r	r	r	r

位/位域	名称	描述
31	DIR	计数方向： 0: 正向计数 1: 反向计数
30:22	保留	必须保持复位值
21	CH1F	通道 1 输出标志 该位用于指示通道 1 的输出电平状态。 0：通道 1 输出无效电平。

		1：通道 1 输出有效电平。
20	CH0F	通道 0 输出标志 该位用于指示通道 0 的输出电平状态。 0：通道 0 输出无效电平。 1：通道 0 输出有效电平。
19	CH1SF	通道 1 输出状态标志 此位指示通道 1 在延迟空闲模式进入时的输出状态，该位仅当任意延时保护进入时更新。 0: 通道 1 输出处于非活动电平。 1: 通道 1 输出处于活动电平。
18	CH0SF	通道 0 输出状态标志 此位指示通道 0 在延迟空闲模式进入时的输出状态，该位仅当任意延时保护进入时更新。 0: 通道 0 输出处于非活动电平。 1: 通道 0 输出处于活动电平。
17	BLNIF	均衡空闲标志 该位用于指示在进入均衡空闲状态时，哪个通道正在输出信号。 0：当进入均衡空闲模式时，通道 0 输出 CH0OPRE 信号，而通道 1 输出无效电平。 1：当进入均衡空闲模式时，通道 1 输出 CH1OPRE 信号，而通道 0 输出无效电平。
16	CBLNF	当前的均衡状态标志 该位仅在均衡模式下有效。该位用于指示当前正在输出信号的通道。 0：通道 0 输出 CH0OPRE 信号，通道 1 输出无效电平。 1：通道 1 输出 CH1OPRE 信号，通道 0 输出无效电平。
15	保留	必须保持复位值
14	DLYIIF	延迟空闲模式进入中断标志 进入延迟空闲或均衡空闲模式时，此标志由硬件置位。 0：延迟空闲模式进入中断未发生 1：延迟空闲模式进入中断发生
13	RSTIF	计数器复位中断标志 计数器复位或翻转事件发生时，此标志由硬件置位。 0：计数器复位或翻转事件中断未发生 1：计数器复位或翻转事件中断发生
12	CH1ONAIF	通道 1 输出无效中断标志 请参阅 CH0ONAIF 说明。
11	CH1OAIF	通道 1 输出有效中断标志 请参阅 CH0OAIF 说明。
10	CH0ONAIF	通道 0 输出无效中断标志

		当通道 0 输出无效（C0OPRE 从有效变为无效）发生时，该标志由硬件置位。 0：通道 0 输出无效中断未发生 1：通道 0 输出无效中断发生
9	CH0OAIF	通道 0 输出有效中断标志 当通道 0 输出有效（C0OPRE 从无效变为有效）发生时，该标志由硬件置位。 0：通道 0 输出有效中断未发生 1：通道 0 输出有效中断发生
8	CAP1IF	捕获 1 中断标志 当捕获 1 事件发生，该标志由硬件置位。 0：捕获 1 中断未发生 1：捕获 1 中断发生
7	CAP0IF	捕获 0 中断标志 当捕获 0 事件发生，该标志由硬件置位。 0：捕获 0 中断未发生 1：捕获 0 中断发生
6	UPIF	更新中断标志 当更新事件发生，该标志由硬件置位。 0：更新中断未发生 1：更新中断发生
5	保留	必须保持复位值
4	REPIF	重复中断标志 当重复事件发生时，此标志由硬件置位。 0：重复中断未发生 1：重复中断发生
3	CMP3IF	比较 3 中断标志 当比较 3 事件发生，该标志由硬件置位。 0：比较 3 中断未发生 1：比较 3 中断发生
2	CMP2IF	比较 2 中断标志 当比较 2 事件发生，该标志由硬件置位。 0：比较 2 中断未发生 1：比较 2 中断发生
1	CMP1IF	比较 1 中断标志 当比较 1 事件发生，该标志由硬件置位。 0：比较 1 中断未发生

1：比较 1 中断发生

0	CMP0IF	比较 0 中断标志 当比较 2 事件发生，该标志由硬件置位。 0：比较 0 中断未发生 1：比较 0 中断发生

Slave_TIMERx 中断标志清除寄存器 (HRTIMER_STxINTC)

地址偏移：0x08

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	DLYIIFC	RSTIFC	CH1ONAIFC	CH1OAIFC	CH0ONAIFC	CH0OAIFC	CAP1IFC	CAP0IFC	UPIFC	保留	REPIFC	CMP3IFC	CMP2IFC	CMP1IFC	CMP0IFC
	w	w	w	w	w	w	w	w	w		w	w	w	w	w

位/位域	名称	描述
31:15	保留	必须保持复位值
14	DLYIIFC	延迟空闲模式进入中断标志清除 0：无效 1：清除延迟空闲模式进入中断标志（HRTIMER_STxINTF 寄存器中的 DLYIIF 位）
13	RSTIFC	计数器复位中断标志清除 0：无效 1：清除计数器复位中断标志（HRTIMER_STxINTF 寄存器中的 RSTIF 位）
12	CH1ONAIFC	通道 1 输出无效中断标志清除 清除 HRTIMER_STxINTF 寄存器中的 CH1ONAIF 位 请参考 CH0ONAIFC 的描述
11	CH1OAIFC	通道 1 输出有效中断标志清除 清除 HRTIMER_STxINTF 寄存器中的 CH1OAIF 位 请参考 CH0OAIFC 的描述
10	CH0ONAIFC	通道 0 输出无效中断标志清除 0：无效 1：清除通道 0 输出无效中断标志（HRTIMER_STxINTF 寄存器中的 CH0ONAIF 位）

9	CH0OAIFC	通道 0 输出有效中断标志清除 0：无效 1：清除通道 0 输出有效中断标志（HRTIMER_STxINTF 寄存器中的 CH0OAIF 位）
8	CAP1IFC	比较 1 捕获中断标志清除 0：无效 1：清除比较 1 捕获中断标志（HRTIMER_STxINTF 寄存器中的 CAP1IF 位）
7	CAP0IFC	比较 0 捕获中断标志清除 0：无效 1：清除比较 0 捕获中断标志（HRTIMER_STxINTF 寄存器中的 CAP0IF 位）
6	UPIFC	更新中断标志清除 0：无效 1：清除更新中断标志（HRTIMER_STxINTF 寄存器中的 UPIF 位）
5	保留	必须保持复位值
4	REPIFC	重复中断标志清除 0：无效 1：清除重复中断标志（HRTIMER_STxINTF 寄存器中的 REPIF 位）
3	CMP3IFC	比较 3 中断标志清除 0：无效 1：清除比较 3 中断标志（HRTIMER_STxINTF 寄存器中的 CMP3IF 位）
2	CMP2IFC	比较 2 中断标志清除 0：无效 1：清除比较 2 中断标志（HRTIMER_STxINTF 寄存器中的 CMP2IF 位）
1	CMP1IFC	比较 1 中断标志清除 0：无效 1：清除比较 1 中断标志（HRTIMER_STxINTF 寄存器中的 CMP1IF 位）
0	CMP0IFC	比较 0 中断标志清除 0：无效 1：清除比较 0 中断标志（HRTIMER_STxINTF 寄存器中的 CMP0IF 位）

Slave_TIMERx DMA 和中断使能寄存器 (HRTIMER_STxDMAINTEN)

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

保留	DLYIDEN	RSTDEN	CH1ONADEN	CH1OADEN	CH0ONADEN	CH0OADEN	CAP1DEN	CAP0DEN	UPDEN	保留	REPDEN	CMP3DEN	CMP2DEN	CMP1DEN	CMP0DEN
	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

保留	DLYIIE	RSTIE	CH1ONAIE	CH1OAIE	CH0ONAIE	CH0OAIE	CAP1IE	CAP0IE	UPIE	保留	REPIE	CMP3IE	CMP2IE	CMP1IE	CMP0IE
	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

位/位域	名称	描述
31	保留	必须保持复位值
30	DLYIDEN	延迟空闲模式进入 DMA 请求使能 0：禁能 1：使能
29	RSTDEN	计数器复位 DMA 请求使能 0：禁能 1：使能
28	CH1ONADEN	通道 1 输出无效 DMA 请求使能 请参考 CH0ONADEN 位描述。
27	CH1OADEN	通道 1 输出有效 DMA 请求使能 请参考 CH0OADEN 位描述。
26	CH0ONADEN	通道 0 输出无效 DMA 请求使能 0：禁能 1：使能
25	CH0OADEN	通道 0 输出有效 DMA 请求使能 0：禁能 1：使能
24	CAP1DEN	捕获 1 DMA 请求使能 0：禁能 1：使能
23	CAP0DEN	捕获 0 DMA 请求使能 0：禁能 1：使能
22	UPDEN	更新 DMA 请求使能 0：禁能 1：使能

21	保留	必须保持复位值
20	REPDEN	重复 DMA 请求使能 0：禁能 1：使能
19	CMP3DEN	比较 3 DMA 请求使能 0：禁能 1：使能
18	CMP2DEN	比较 2 DMA 请求使能 0：禁能 1：使能
17	CMP1DEN	比较 1 DMA 请求使能 0：禁能 1：使能
16	CMP0DEN	比较 0 DMA 请求使能 0：禁能 1：使能
15	保留	必须保持复位值
14	DLYIIE	延迟空闲模式进入中断使能 0：禁能 1：使能
13	RSTIE	计数器复位中断使能 0：禁能 1：使能
12	CH1ONAIE	通道 1 输出无效中断使能 请参考 CH0ONAIE 位描述。
11	CH1OAIE	通道 1 输出有效中断使能 请参考 CH0OAIE 位描述。
10	CH0ONAIE	通道 0 输出无效中断使能 0：禁能 1：使能
9	CH0OAIE	通道 0 输出有效中断使能 0：禁能 1：使能
8	CAP1IE	捕获 1 中断使能

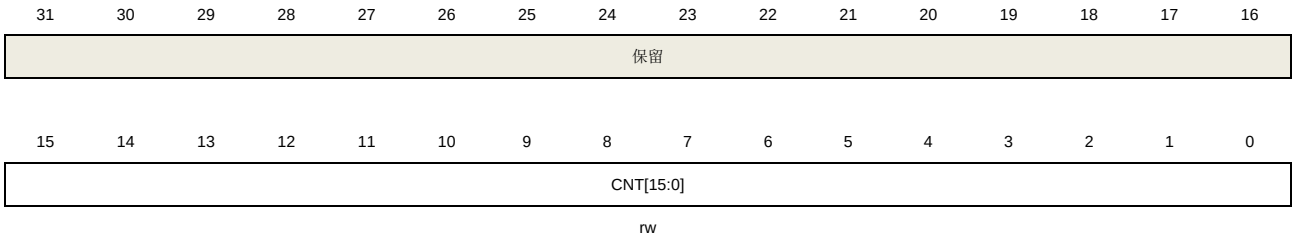
		0：禁能 1：使能
7	CAP0IE	捕获 0 中断使能 0：禁能 1：使能
6	UPIE	更新中断使能 0：禁能 1：使能
5	保留	必须保持复位值
4	REPIE	重复中断使能 0：禁能 1：使能
3	CMP3IE	比较 3 中断使能 0：禁能 1：使能
2	CMP2IE	比较 2 中断使能 0：禁能 1：使能
1	CMP1IE	比较 1 中断使能 0：禁能 1：使能
0	CMP0IE	比较 0 中断使能 0：禁能 1：使能

Slave_TIMERx 计数器寄存器 (HRTIMER_STxCNT)

地址偏移：0x10

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。



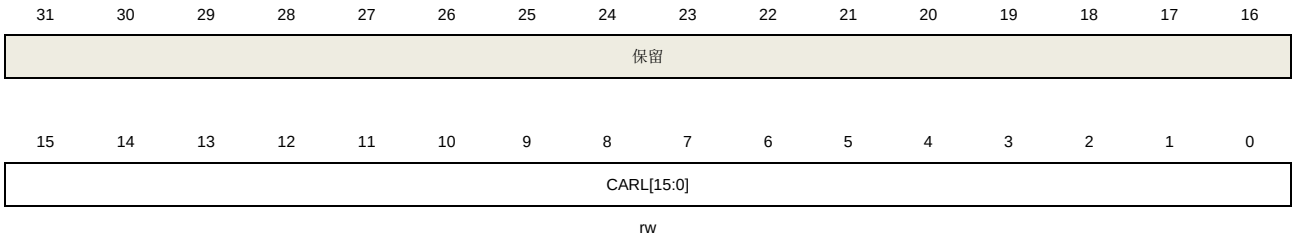
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CNT[15:0]	该位域用于配置当前计数器值。对该位域进行写操作可以更改计数器的值。 仅当 Slave_TIMERx 停止（HRTIMER_MTCTL0 寄存器中的 STxCEN = 0）时，才能对其进行写操作，更改计数器的值。 **注意：** （1）计数器时钟分频系数小于 64（CNTCKDIV [2:0] < 5）时，计数器的最低有效位无效，它们不能被写入，读出值为 0。 （2）如果写入该位域的值高于 HRTIM_MPER 寄存器值，则定时器的行为是不可预测的。

Slave_TIMERx 计数器自动重载寄存器 (HRTIMER_STxCAR)

地址偏移：0x14

复位值：0x0000 FFDF

该寄存器只能进行字（32 位）访问。



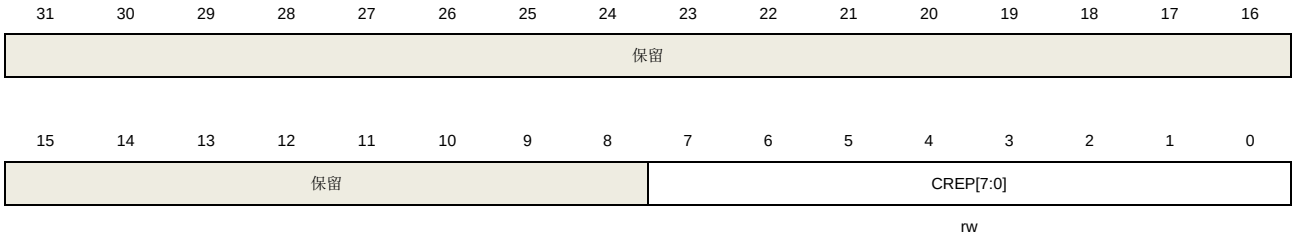
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CARL[15:0]	计数器自动重载值 该位域定义计数器的自动重载值。该寄存器具有影子寄存器。如果影子寄存器被禁能（SHWEN = 0），它将保存有效寄存器的内容；否则，它将保存影子寄存器的内容。 **注意：** （1）最小值必须大于或等于（3*$t_{HRTIMER_CK}$）。例如：当 CNTCKDIV [2:0] = 3'b000 时，CARL [15:0]> = 0x60。 （2）最大值必须小于或等于（0xFFFF －1*$t_{HRTIMER_CK}$）。例如：当 CNTCKDIV [2:0] = 3'b000 时，CARL [15:0] <= 0xFFDF。

Slave_TIMERx 重复计数寄存器 (HRTIMER_STxCREP)

地址偏移：0x18

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。



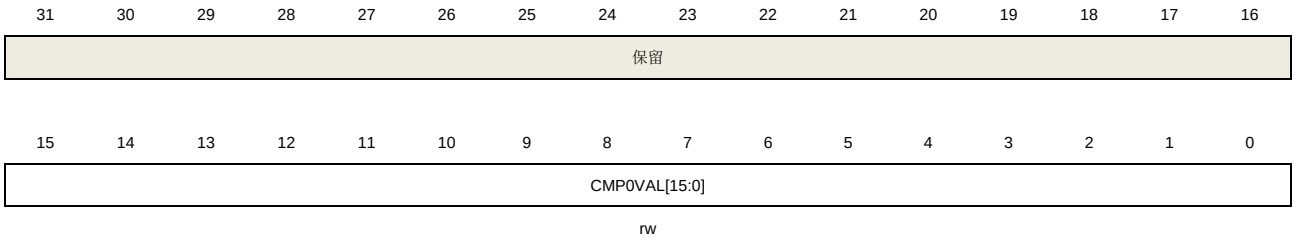
位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	CREP[7:0]	重复计数器值 该位域用于定义重复事件的发生率。当重复计数器递减计数到零时，连续模式下即将发生的翻转事件或复位事件将产生一个重复事件。 该寄存器具有影子寄存器。如果影子寄存器被禁能（SHWEN = 0），它将保存有效寄存器的内容；否则，它将保存影子寄存器的内容。

Slave_TIMERx 比较 0 寄存器 (HRTIMER_STxCMP0V)

地址偏移：0x1C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。



位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CMP0VAL[15:0]	比较 0 值 该位域包含要与计数器进行比较的值。 该寄存器具有影子寄存器。如果影子寄存器被禁能（SHWEN = 0），它将保存有效寄存器的内容；否则，它将保存影子寄存器的内容。 **注意：** （1）最小值必须大于或等于 3 个 $t_{HRTIMER_CK}$。例如：当 CNTCKDIV [2:0] = 3'b000 时，CARL [15:0] > = 0x60。 （2）最大值必须小于或等于（0xFFFF －1*$t_{HRTIMER_CK}$）。例如：当 CNTCKDIV [2:0]

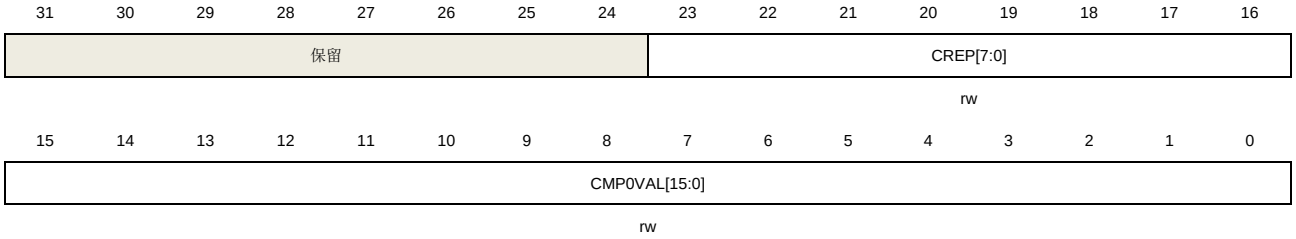
= 3’b000 时，CARL [15:0] <= 0xFFDF。

Slave_TIMERx 比较 0 复合寄存器 (HRTIMER_STxCMP0CP)

地址偏移：0x20

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。



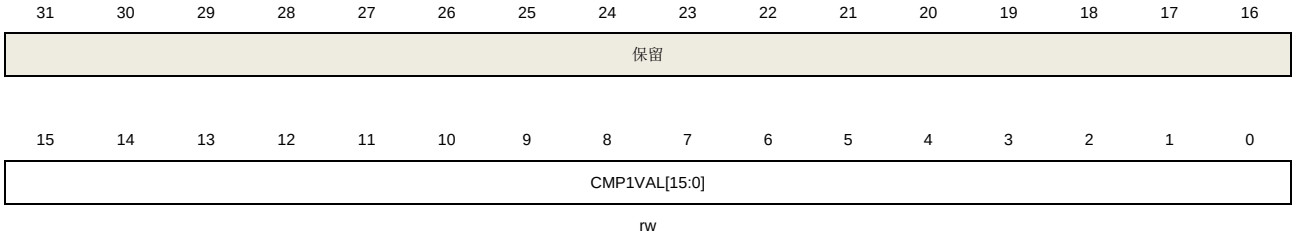
位/位域	名称	描述
31:24	保留	必须保持复位值
23:16	CREP[7:0]	计数器重复值 该位域是 HRTIMER_STxCREP 寄存器中 CREP[7:0]的别名。
15:0	CMP0VAL[15:0]	比较 0 值 该位域是 HRTIMER_STxCMP0V 寄存器中 CMP0VAL[15:0]的别名。

Slave_TIMERx 比较 1 寄存器 (HRTIMER_STxCMP1V)

地址偏移：0x24

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。



位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CMP1VAL[15:0]	比较 1 值 该位域包含要与计数器进行比较的值。 该寄存器具有影子寄存器。如果影子寄存器被禁能（SHWEN = 0），它将保存有效

寄存器的内容；否则，它将保存影子寄存器的内容。

延迟模式中，有效寄存器的值会重新进行计算。

注意：

（1）最小值必须大于或等于 3 个 $t_{HRTIMER_CK}$。例如：当 CNTCKDIV [2:0] = 3'b000 时，CARL [15:0] > = 0x60。

（2）最大值必须小于或等于（0xFFFF －1*$t_{HRTIMER_CK}$）。例如：当 CNTCKDIV [2:0] = 3'b000 时，CARL [15:0] <= 0xFFDF。

Slave_TIMERx 比较 2 寄存器 (HRTIMER_STxCMP2V)

地址偏移：0x28

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP2VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CMP2VAL[15:0]	比较 2 值 该位域包含要与计数器进行比较的值。 该寄存器具有影子寄存器。如果影子寄存器被禁能（SHWEN = 0），它将保存有效寄存器的内容；否则，它将保存影子寄存器的内容。 **注意：** （1）最小值必须大于或等于 3 个 $t_{HRTIMER_CK}$。例如：当 CNTCKDIV [2:0] = 3'b000 时，CARL [15:0] > = 0x60。 （2）最大值必须小于或等于（0xFFFF －1*$t_{HRTIMER_CK}$）。例如：当 CNTCKDIV [2:0] = 3'b000 时，CARL [15:0] <= 0xFFDF。

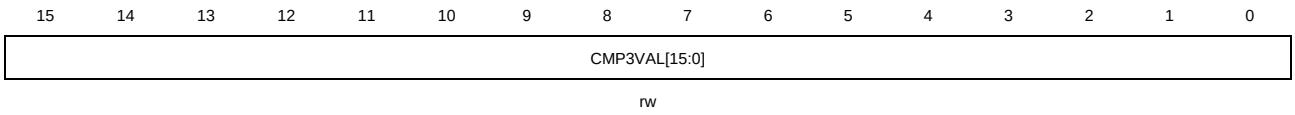
Slave_TIMERx 比较 3 寄存器 (HRTIMER_STxCMP3V)

地址偏移：0x2C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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



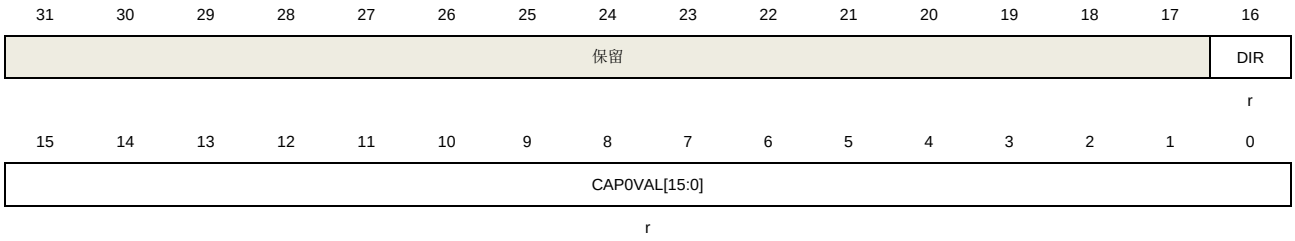
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CMP3VAL[15:0]	比较 3 值 该位域包含要与计数器进行比较的值。 该寄存器具有影子寄存器。如果影子寄存器被禁能（SHWEN = 0），它将保存有效寄存器的内容；否则，它将保存影子寄存器的内容。 延迟模式中，有效寄存器的值会重新进行计算。 **注意：** （1）最小值必须大于或等于 3 个 $t_{HRTIMER_CK}$。例如：当 CNTCKDIV [2:0] = 3'b000 时，CARL [15:0] > = 0x60。 （2）最大值必须小于或等于（0xFFFF －1*$t_{HRTIMER_CK}$）。例如：当 CNTCKDIV [2:0] = 3'b000 时，CARL [15:0] <= 0xFFDF。

Slave_TIMERx 捕获 0 寄存器 (HRTIMER_STxCAP0V)

地址偏移：0x30

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。



位/位域	名称	描述
31:17	保留	必须保持复位值
16	DIR	Slave_TIMERx 捕获 1 方向状态： 该寄存器保存捕获 1 事件发生时的计数方向值： 0: 计时器正向计数 1: 计时器反向计数 在正向计数模式下（UPDNM 位复位），DIR 位始终读取为 0。
15:0	CAP0VAL[15:0]	捕获 0 值

该位域保持上一个捕获事件发生时的计数器值，且该位域只读。

注意：计数器时钟分频系数小于 32（CNTCKDIV [2:0] <5）时，计数器的最低有效位无效，它们不能被写入，读出值为 0。

Slave_TIMERx 捕获 1 寄存器 (HRTIMER_STxCAP1V)

地址偏移：0x34

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP1VAL[15:0]															
r															

位/位域	名称	描述
31:17	保留	必须保持复位值
16	DIR	Slave_TIMERx 捕获 1 方向状态： 该寄存器保存捕获 1 事件发生时的计数方向值： 0: 计时器正向计数 1: 计时器反向计数 在正向计数模式下（UPDNM 位复位），DIR 位始终读取为 0。
15:0	CAP1VAL[15:0]	捕获 1 值 该位域保持上一个捕获事件发生时的计数器值，且该位域只读。 **注意：**计数器时钟分频系数小于 32（CNTCKDIV [2:0] <5）时，计数器的最低有效位无效，它们不能被写入，读出值为 0。

Slave_TIMERx 死区控制寄存器 (HRTIMER_STxDTCTL)

地址偏移：0x38

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DTFSVP ROT	DTFSPR OT	保留				DTFS	DTFCFG[8:0]								
rwo	rwo					rw	rw								

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

DTRSVPROT	DTRSPROT	DTGCKDIV[3:0]	DTRS	DTRCFG[8:0]
rwo	rwo	rw	rw	rw

位/位域	名称	描述
31	DTFSVPROT	死区下降沿（值和符号）保护 该位域用于死区下降沿（值和符号）的写保护。 0：保护禁能。DTFS 位和 DTFCFG [15:0]位域是可写的。 1：保护使能。DTFS 位和 DTFCFG [15:0]位域是只读的。 **注意：** （1）DTFCFG [15:9]位域在 HRTIMER_STxACTL 寄存器中。
30	DTFSPROT	死区下降沿（符号）保护 该位域用于死区下降沿（符号）的写保护。 0：保护禁能。HRTIMER_STxDTCTL 寄存器中的 DTFS 位是可写的。 1：保护使能。HRTIMER_STxDTCTL 寄存器中的 DTFS 位是只读的。
29:26	保留	必须保持复位值
25	DTFS	死区下降沿值的符号 0：死区下降沿值的符号为正。 1：死区下降沿值的符号为负。 **注意：**当 HRTIMER_STxDTCTL 寄存器中的 DTFSPROT 位或 DTFSVPROT 位置 1 时，无法修改此位。
24:16	DTFCFG[8:0]	死区下降沿值 该位域用于配置跟随输出准备信号（OyPRE，y = 0,1）下降沿之后的死区时间值。 DTF 值 = DTFCFG [15:0] x $t_{HRTIMER_DTGCK}$，其中，$t_{HRTIMER_DTGCK}$ = 1/ $f_{HRTIMER_DTGCK}$。 写入该位域可以更改 DTFCFG [15:0]位域的低 9 位。 **注意：** （1）DTFCFG [15:9] 位域在 HRTIMER_STxACTL 寄存器中。 （2）当 HRTIMER_STxDTCTL 寄存器中的 DTFSVPROT 位置 1 时，无法修改此位域。
15	DTRSVPROT	死区上升沿（值和符号）保护 该位域用于死区上升沿（值和符号）的写保护。 0：保护禁能。DTRS 位和 DTRCFG [15:0]位域是可写的。 1：保护使能。DTRS 位和 DTRCFG [15:0]位域是只读的。 **注意：** （1）DTRCFG [15:9]位域在 HRTIMER_STxACTL 寄存器中。
14	DTRSPROT	死区上升沿（符号）保护 该位域用于死区上升沿（符号）的写保护。

		0：保护禁能。HRTIMER_STxDTCTL 寄存器中的 DTRS 位是可写的。 1：保护使能。HRTIMER_STxDTCTL 寄存器中的 DTRS 位是只读的。
13:10	DTGCKDIV[3:0]	死区时间发生器时钟分频 该位域可以通过软件配置，用于确定 HRTIMER 时钟（HRTIMER_CK）和死区时间发生器时钟（HRTIMER_DTGCK）的分频系数。 DTGCKDIV[3] is '0'时，$f_{HRTIMER_DTGCK} = (8*f_{HRTIMER_CK})/2^{DTGCKDIV[2:0]}$. DTGCKDIV[3] is '1' 时，$f_{HRTIMER_DTGCK} = 2^{(DTGCKDIV[2:0]+4)}*f_{HRTIMER_CK}$ 0000：$f_{HRTIMER_DTGCK} = 8*f_{HRTIMER_CK}$ 0001：$f_{HRTIMER_DTGCK} = (8*f_{HRTIMER_CK})/2$ 0010：$f_{HRTIMER_DTGCK} = (8*f_{HRTIMER_CK})/4$ 0011：$f_{HRTIMER_DTGCK} = (8*f_{HRTIMER_CK})/8$ 0100：$f_{HRTIMER_DTGCK} = (8*f_{HRTIMER_CK})/16$ 0101：$f_{HRTIMER_DTGCK} = (8*f_{HRTIMER_CK})/32$ 0110：$f_{HRTIMER_DTGCK} = (8*f_{HRTIMER_CK})/64$ 0111：$f_{HRTIMER_DTGCK} = (8*f_{HRTIMER_CK})/128$ 1000: $f_{HRTIMER_DTGCK} = 16* f_{HRTIMER_CK}$ 1001: $f_{HRTIMER_DTGCK} = 32* f_{HRTIMER_CK}$ 其他位保留。 **注意：**如果任何一个保护位（DTFSPROT，DTFSVPROT，DTRSPROT 和 DTRSVPROT）置位，则不能修改此位域。
9	DTRS	死区上升沿值的符号 0：死区上升沿值的符号为正。 1：死区上升沿值的符号为负。 **注意：**当 HRTIMER_STxDTCTL 寄存器中的 DTRSPROT 位或 DTRSVPROT 位置 1 时，无法修改此位。
8:0	DTRCFG[8:0]	死区上升沿值 该位域用于配置跟随输出准备信号（OyPRE，y = 0,1）上升沿之后的死区时间值。 DTR 值 = DTRCFG[15:0] x $t_{HRTIMER_DTGCK}$，其中，$t_{HRTIMER_DTGCK} = 1/ f_{HRTIMER_DTGCK}$。 写入该位域可以更改 DTRCFG[15:0]位域的低 9 位。 **注意：** （1）DTRCFG [15:9]位域在 HRTIMER_STxACTL 寄存器中。 （2）当 HRTIMER_STxDTCTL 寄存器中的 DTRSVPROT 位置 1 时，无法修改此位域。

Slave_TIMERx 通道 0 置位请求寄存器 (HRTIMER_STxCH0SET)

地址偏移：0x3C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH0SUP	CH0SEXEV9	CH0SEXEV8	CH0SEXEV7	CH0SEXEV6	CH0SEXEV5	CH0SEXEV4	CH0SEXEV3	CH0SEXEV2	CH0SEXEV1	CH0SEXEV0	CH0SSTEV8	CH0SSTEV7	CH0SSTEV6	CH0SSTEV5	CH0SSTEV4
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0SSTEV3	CH0SSTEV2	CH0SSTEV1	CH0SSTEV0	CH0SMTCMP3	CH0SMTCMP2	CH0SMTCMP1	CH0SMTCMP0	CH0SMTPER	CH0SCMP3	CH0SCMP2	CH0SCMP1	CH0SCMP0	CH0SPER	CH0SRST	CH0SSEV
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	CH0SUP	更新事件生成通道 0 置位请求 该位置 1 时，更新事件可以产生置位请求。 0：更新事件不生成置位请求。 1：更新事件生成置位请求。
30	CH0SEXEV9	外部事件 9 生成通道 0 置位请求 请参考 CH0SEXEV0 说明。
29	CH0SEXEV8	外部事件 8 生成通道 0 置位请求 请参考 CH0SEXEV0 说明。
28	CH0SEXEV7	外部事件 7 生成通道 0 置位请求 请参考 CH0SEXEV0 说明。
27	CH0SEXEV6	外部事件 6 生成通道 0 置位请求 请参考 CH0SEXEV0 说明。
26	CH0SEXEV5	外部事件 5 生成通道 0 置位请求 请参考 CH0SEXEV0 说明。
25	CH0SEXEV4	外部事件 4 生成通道 0 置位请求 请参考 CH0SEXEV0 说明。
24	CH0SEXEV3	外部事件 3 生成通道 0 置位请求 请参考 CH0SEXEV0 说明。
23	CH0SEXEV2	外部事件 2 生成通道 0 置位请求 请参考 CH0SEXEV0 说明。
22	CH0SEXEV1	外部事件 1 生成通道 0 置位请求 请参考 CH0SEXEV0 说明。
21	CH0SEXEV0	外部事件 0 生成通道 0 置位请求 该位置 1 时，外部事件 0 可以产生置位请求。

0：该事件不生成置位请求。
1：该事件生成置位请求。

20	CH0SSTEV8	Slave_TIMERx 互连事件 8 生成通道 0 置位请求 请参考 CH0SSTEV0 说明。
19	CH0SSTEV7	Slave_TIMERx 互连事件 7 生成通道 0 置位请求 请参考 CH0SSTEV0 说明。
18	CH0SSTEV6	Slave_TIMERx 互连事件 6 生成通道 0 置位请求 请参考 CH0SSTEV0 说明。
17	CH0SSTEV5	Slave_TIMERx 互连事件 5 生成通道 0 置位请求 请参考 CH0SSTEV0 说明。
16	CH0SSTEV4	Slave_TIMERx 互连事件 4 生成通道 0 置位请求 请参考 CH0SSTEV0 说明。
15	CH0SSTEV3	Slave_TIMERx 互连事件 3 生成通道 0 置位请求 请参考 CH0SSTEV0 说明。
14	CH0SSTEV2	Slave_TIMERx 互连事件 2 生成通道 0 置位请求 请参考 CH0SSTEV0 说明。
13	CH0SSTEV1	Slave_TIMERx 互连事件 1 生成通道 0 置位请求 请参考 CH0SSTEV0 说明。
12	CH0SSTEV0	Slave_TIMERx 互连事件 0 生成通道 0 置位请求 该位置 1 时，Slave_TIMERx 互连事件 0 可以产生置位请求。具体请参考***表 25-9. Slave_TIMER 内部连接事件***。 0：该事件不生成置位请求。 1：该事件生成置位请求。
11	CH0SMTCMP3	Master_TIMER 比较 3 事件生成通道 0 置位请求 该位置 1 时，Master_TIMER 比较 3 事件可以产生置位请求。 0：Master_TIMER 比较 3 事件不生成置位请求。 1：Master_TIMER 比较 3 事件生成置位请求。
10	CH0SMTCMP2	Master_TIMER 比较 2 事件生成通道 0 置位请求 该位置 1 时，Master_TIMER 比较 2 事件可以产生置位请求。 0：Master_TIMER 比较 2 事件不生成置位请求。 1：Master_TIMER 比较 2 事件生成置位请求。
9	CH0SMTCMP1	Master_TIMER 比较 1 事件生成通道 0 置位请求 该位置 1 时，Master_TIMER 比较 1 事件可以产生置位请求。 0：Master_TIMER 比较 1 事件不生成置位请求。

		1：Master_TIMER 比较 1 事件生成置位请求。
8	CH0SMTCMP0	Master_TIMER 比较 0 事件生成通道 0 置位请求 该位置 1 时，Master_TIMER 比较 0 事件可以产生置位请求。 0：Master_TIMER 比较 0 事件不生成置位请求。 1：Master_TIMER 比较 0 事件生成置位请求。
7	CH0SMTPER	Master_TIMER 周期事件生成通道 0 置位请求 连续模式下，Master_TIMER 计数器的翻转事件可以产生置位请求。在单脉冲模式下，Master_TIMER 计数器的复位事件可以产生置位请求。 0：该事件不生成置位请求。 1：该事件生成置位请求。
6	CH0SCMP3	Slave_TIMERx 比较 3 事件生成通道 0 置位请求 该位置 1 时，Slave_TIMERx 比较 3 事件可以产生置位请求。 0：Slave_TIMERx 比较 3 事件不生成置位请求。 1：Slave_TIMERx 比较 3 事件生成置位请求。
5	CH0SCMP2	Slave_TIMERx 比较 2 事件生成通道 0 置位请求 该位置 1 时，Slave_TIMERx 比较 2 事件可以产生置位请求。 0：Slave_TIMERx 比较 2 事件不生成置位请求。 1：Slave_TIMERx 比较 2 事件生成置位请求。
4	CH0SCMP1	Slave_TIMERx 比较 1 事件生成通道 0 置位请求 该位置 1 时，Slave_TIMERx 比较 1 事件可以产生置位请求。 0：Slave_TIMERx 比较 1 事件不生成置位请求。 1：Slave_TIMERx 比较 1 事件以生成置位请求。
3	CH0SCMP0	Slave_TIMERx 比较 0 事件生成通道 0 置位请求 该位置 1 时，Slave_TIMERx 比较 0 事件可以产生置位请求。 0：Slave_TIMERx 比较 0 事件不生成置位请求。 1：Slave_TIMERx 比较 0 事件生成置位请求。
2	CH0SPER	Slave_TIMERx 周期事件生成通道 0 置位请求 该位置 1 时，Slave_TIMERx 周期事件可以产生置位请求。 0：Slave_TIMERx 周期事件不生成置位请求。 1：Slave_TIMERx 周期事件生成置位请求。
1	CH0SRST	Slave_TIMERx 复位事件生成通道 0 置位请求 该位置 1 时，由软件和同步输入引起的 Slave_TIMERx 复位事件，生成通道 0 置位请求。 0：该事件不生成置位请求。 1：该事件生成置位请求。 **注意：**该位置 1 时，其他的定时器复位事件不会影响输出。

0	CH0SSEV	软件事件生成通道 0 置位请求 该位由软件置 1，由硬件自动清除。该位置 1 时，生成通道 0 置位请求。 0：该事件不生成置位请求 1：该事件生成置位请求 **注意：**该位不进行预装载。

Slave_TIMERx 通道 0 复位请求寄存器 (HRTIMER_STxCH0RST)

地址偏移：0x40

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH0RSUP	CH0RSEXEV9	CH0RSEXEV8	CH0RSEXEV7	CH0RSEXEV6	CH0RSEXEV5	CH0RSEXEV4	CH0RSEXEV3	CH0RSEXEV2	CH0RSEXEV1	CH0RSEXEV0	CH0RSSTEV8	CH0RSSTEV7	CH0RSSTEV6	CH0RSSTEV5	CH0RSSTEV4
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0RSSTEV3	CH0RSSTEV2	CH0RSSTEV1	CH0RSSTEV0	CH0RSMTCMP3	CH0RSMTCMP2	CH0RSMTCMP1	CH0RSMTCMP0	CH0RSMTPER	CH0RSCMP3	CH0RSCMP2	CH0RSCMP1	CH0RSCMP0	CH0RSPER	CH0RSRST	CH0RSSEV
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	CH0RSUP	更新事件生成通道 0 复位请求 该位置 1 时，更新事件可以产生复位请求。 0：更新事件不生成复位请求。 1：更新事件生成复位请求。
30	CH0RSEXEV9	外部事件 9 生成通道 0 复位请求 请参考 CH0RSEXEV0 说明。
29	CH0RSEXEV8	外部事件 8 生成通道 0 复位请求 请参考 CH0RSEXEV0 说明。
28	CH0RSEXEV7	外部事件 7 生成通道 0 复位请求 请参考 CH0RSEXEV0 说明。
27	CH0RSEXEV6	外部事件 6 生成通道 0 复位请求 请参考 CH0RSEXEV0 说明。
26	CH0RSEXEV5	外部事件 5 生成通道 0 复位请求 请参考 CH0RSEXEV0 说明。
25	CH0RSEXEV4	外部事件 4 生成通道 0 复位请求

		请参考 CH0RSEXEV0 说明。
24	CH0RSEXEV3	外部事件 3 生成通道 0 复位请求 请参考 CH0RSEXEV0 说明。
23	CH0RSEXEV2	外部事件 2 生成通道 0 复位请求 请参考 CH0RSEXEV0 说明。
22	CH0RSEXEV1	外部事件 1 生成通道 0 复位请求 请参考 CH0RSEXEV0 说明。
21	CH0RSEXEV0	外部事件 0 生成通道 0 复位请求 当该位置 1 时，外部事件 0 可以产生复位请求。 0：该事件不生成复位请求。 1：该事件生成复位请求。
20	CH0RSSTEV8	Slave_TIMERx 互连事件 8 生成通道 0 复位请求 请参考 CH0RSSTEV0 说明。
19	CH0RSSTEV7	Slave_TIMERx 互连事件 7 生成通道 0 复位请求 请参考 CH0RSSTEV0 说明。
18	CH0RSSTEV6	Slave_TIMERx 互连事件 6 生成通道 0 复位请求 请参考 CH0RSSTEV0 说明。
17	CH0RSSTEV5	Slave_TIMERx 互连事件 5 生成通道 0 复位请求 请参考 CH0RSSTEV0 说明。
16	CH0RSSTEV4	Slave_TIMERx 互连事件 4 生成通道 0 复位请求 请参考 CH0RSSTEV0 说明。
15	CH0RSSTEV3	Slave_TIMERx 互连事件 3 生成通道 0 复位请求 请参考 CH0RSSTEV0 说明。
14	CH0RSSTEV2	Slave_TIMERx 互连事件 2 生成通道 0 复位请求 请参考 CH0RSSTEV0 说明。
13	CH0RSSTEV1	Slave_TIMERx 互连事件 1 生成通道 0 复位请求 请参考 CH0RSSTEV0 说明。
12	CH0RSSTEV0	Slave_TIMERx 互连事件 0 生成通道 0 复位请求 该位置 1 时，Slave_TIMERx 互连事件 0 可以产生复位请求。具体请参考***表 25-9. Slave_TIMER 内部连接事件***。 0：该事件不生成复位请求。 1：该事件生成复位请求。
11	CH0RSMTCMP3	Master_TIMER 比较 3 事件生成通道 0 复位请求

		该位置 1 时，Master_TIMER 比较 3 事件可以产生复位请求。 0：Master_TIMER 比较 3 事件不生成复位请求。 1：Master_TIMER 比较 3 事件生成复位请求。
10	CH0RSMTCMP2	Master_TIMER 比较 2 事件生成通道 0 复位请求 该位置 1 时，Master_TIMER 比较 2 事件可以产生复位请求。 0：Master_TIMER 比较 2 事件不生成复位请求。 1：Master_TIMER 比较 2 事件生成复位请求。
9	CH0RSMTCMP1	Master_TIMER 比较 1 事件生成通道 0 复位请求 该位置 1 时，Master_TIMER 比较 1 事件可以产生复位请求。 0：Master_TIMER 比较 1 事件不生成复位请求。 1：Master_TIMER 比较 1 事件生成复位请求。
8	CH0RSMTCMP0	Master_TIMER 比较 0 事件生成通道 0 复位请求 该位置 1 时，Master_TIMER 比较 0 事件可以产生复位请求。 0：Master_TIMER 比较 0 事件不生成复位请求。 1：Master_TIMER 比较 0 事件生成复位请求。
7	CH0RSMTPER	Master_TIMER 周期事件生成通道 0 复位请求 连续模式下，Master_TIMER 计数器翻转事件可以产生复位请求。在单脉冲模式下，Master_TIMER 计数器复位事件可以产生复位请求。 0：该事件不生成复位请求。 1：该事件生成复位请求。
6	CH0RSCMP3	Slave_TIMERx 比较 3 事件生成通道 0 复位请求 该位置 1 时，Slave_TIMERx 比较 3 事件可以产生复位请求。 0：Slave_TIMERx 比较 3 事件不生成复位请求。 1：Slave_TIMERx 比较 3 事件生成复位请求。
5	CH0RSCMP2	Slave_TIMERx 比较 2 事件生成通道 0 复位请求 该位置 1 时，Slave_TIMERx 比较 2 事件可以产生复位请求。 0：Slave_TIMERx 比较 2 事件不生成复位请求。 1：Slave_TIMERx 比较 2 事件生成复位请求。
4	CH0RSCMP1	Slave_TIMERx 比较 1 事件生成通道 0 复位请求 该位置 1 时，Slave_TIMERx 比较 1 事件可以产生复位请求。 0：Slave_TIMERx 比较 1 事件不生成复位请求。 1：Slave_TIMERx 比较 1 事件生成复位请求。
3	CH0RSCMP0	Slave_TIMERx 比较 0 事件生成通道 0 复位请求 该位置 1 时，Slave_TIMERx 比较 0 事件可以产生复位请求。 0：Slave_TIMERx 比较 0 事件不生成复位请求。 1：Slave_TIMERx 比较 0 事件生成复位请求。

2	CH0RSPER	Slave_TIMERx 周期事件生成通道 0 复位请求 该位置 1 时，Slave_TIMERx 周期事件可以产生复位请求。 0：Slave_TIMERx 周期事件不生成复位请求。 1：Slave_TIMERx 周期事件生成复位请求。
1	CH0RSRST	Slave_TIMERx 复位事件生成通道 0 复位请求 该位置 1 时，由软件和同步输入引起的 Slave_TIMERx 复位事件，生成通道 0 复位请求。 0：该事件不生成复位请求。 1：该事件生成复位请求。 **注意：**该位置 1 时，其他的定时器复位事件不会影响输出。
0	CH0RSSEV	软件事件生成通道 0 复位请求 该位由软件置 1，由硬件自动清除。该位置 1 时，生成通道 0 复位请求。 0：该事件不生成复位请求 1：该事件生成复位请求 **注意：**该位不进行预装载。

Slave_TIMERx 通道 1 置位请求寄存器 (HRTIMER_STxCH1SET)

地址偏移：0x44

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH1SUP	CH1SEXEV9	CH1SEXEV8	CH1SEXEV7	CH1SEXEV6	CH1SEXEV5	CH1SEXEV4	CH1SEXEV3	CH1SEXEV2	CH1SEXEV1	CH1SEXEV0	CH1SSTEV8	CH1SSTEV7	CH1SSTEV6	CH1SSTEV5	CH1SSTEV4
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1SSTEV3	CH1SSTEV2	CH1SSTEV1	CH1SSTEV0	CH1SMTCMP3	CH1SMTCMP2	CH1SMTCMP1	CH1SMTCMP0	CH1SMTPER	CH1SCMP3	CH1SCMP2	CH1SCMP1	CH1SCMP0	CH1SPER	CH1SRST	CH1SSEV
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	CH1SUP	更新事件生成通道 1 置位请求 该位置 1 时，更新事件可以产生置位请求。 0：更新事件不生成置位请求。 1：更新事件生成置位请求。
30	CH1SEXEV9	外部事件 9 生成通道 1 置位请求 请参考 CH1SEXEV0 说明。
29	CH1SEXEV8	外部事件 8 生成通道 1 置位请求

		请参考 CH1SEXEV0 说明。
28	CH1SEXEV7	外部事件 7 生成通道 1 置位请求 请参考 CH1SEXEV0 说明。
27	CH1SEXEV6	外部事件 6 生成通道 1 置位请求 请参考 CH1SEXEV0 说明。
26	CH1SEXEV5	外部事件 6 生成通道 1 置位请求 请参考 CH1SEXEV0 说明。
25	CH1SEXEV4	外部事件 4 生成通道 1 置位请求 请参考 CH1SEXEV0 说明。
24	CH1SEXEV3	外部事件 3 生成通道 1 置位请求 请参考 CH1SEXEV0 说明。
23	CH1SEXEV2	外部事件 2 生成通道 1 置位请求 请参考 CH1SEXEV0 说明。
22	CH1SEXEV1	外部事件 1 生成通道 1 置位请求 请参考 CH1SEXEV0 说明。
21	CH1SEXEV0	外部事件 0 生成通道 1 置位请求 当该位置 1 时，外部事件 1 可以产生置位请求。 0：该事件不生成置位请求。 1：该事件生成置位请求。
20	CH1SSTEV8	Slave_TIMERx 互连事件 8 生成通道 1 置位请求 请参考 CH1SSTEV0 说明。
19	CH1SSTEV7	Slave_TIMERx 互连事件 7 生成通道 1 置位请求 请参考 CH1SSTEV0 说明。
18	CH1SSTEV6	Slave_TIMERx 互连事件 6 生成通道 1 置位请求 请参考 CH1SSTEV0 说明。
17	CH1SSTEV5	Slave_TIMERx 互连事件 5 生成通道 1 置位请求 请参考 CH1SSTEV0 说明。
16	CH1SSTEV4	Slave_TIMERx 互连事件 4 生成通道 1 置位请求 请参考 CH1SSTEV0 说明。
15	CH1SSTEV3	Slave_TIMERx 互连事件 3 生成通道 1 置位请求 请参考 CH1SSTEV0 说明。
14	CH1SSTEV2	Slave_TIMERx 互连事件 2 生成通道 1 置位请求

请参考 CH1SSTEV0 说明。

13 CH1SSTEV1 Slave_TIMERx 互连事件 1 生成通道 1 置位请求
请参考 CH1SSTEV0 说明。

12 CH1SSTEV0 Slave_TIMERx 互连事件 0 生成通道 1 置位请求
该位置 1 时，Slave_TIMERx 互连事件 0 可以产生置位请求。具体请参考***表 25-9. Slave_TIMER 内部连接事件***。
0：该事件不生成置位请求。
1：该事件生成置位请求。

11 CH1SMTCMP3 Master_TIMER 比较 3 事件生成通道 1 置位请求
该位置 1 时，Master_TIMER 比较 3 事件可以产生置位请求。
0：Master_TIMER 比较 3 事件不生成置位请求。
1：Master_TIMER 比较 3 事件生成置位请求。

10 CH1SMTCMP2 Master_TIMER 比较 2 事件生成通道 1 置位请求
该位置 1 时，Master_TIMER 比较 2 事件可以产生置位请求。
0：Master_TIMER 比较 2 事件不生成置位请求。
1：Master_TIMER 比较 2 事件生成置位请求。

9 CH1SMTCMP1 Master_TIMER 比较 1 事件生成通道 1 置位请求
该位置 1 时，Master_TIMER 比较 1 事件可以产生置位请求。
0：Master_TIMER 比较 1 事件不生成置位请求。
1：Master_TIMER 比较 1 事件生成置位请求。

8 CH1SMTCMP0 Master_TIMER 比较 0 事件生成通道 1 置位请求
该位置 1 时，Master_TIMER 比较 0 事件可以产生置位请求。
0：Master_TIMER 比较 0 事件不生成置位请求。
1：Master_TIMER 比较 0 事件生成置位请求。

7 CH1SMTPER Master_TIMER 周期事件生成通道 1 置位请求
连续模式下，Master_TIMER 计数器翻转事件可以产生置位请求。在单脉冲模式下，Master_TIMER 计数器复位事件可以产生置位请求。
0：该事件不生成置位请求。
1：该事件生成置位请求。

6 CH1SCMP3 Slave_TIMERx 比较 3 事件生成通道 1 置位请求
该位置 1 时，Slave_TIMERx 比较 3 事件可以产生置位请求。
0：Slave_TIMERx 比较 3 事件不生成置位请求。
1：Slave_TIMERx 比较 3 事件生成置位请求。

5 CH1SCMP2 Slave_TIMERx 比较 2 事件生成通道 1 置位请求
该位置 1 时，Slave_TIMERx 比较 2 事件可以产生置位请求。
0：Slave_TIMERx 比较 2 事件不生成置位请求。

1：Slave_TIMERx 比较 2 事件生成置位请求。

4	CH1SCMP1	Slave_TIMERx 比较 1 事件生成通道 1 置位请求 该位置 1 时，Slave_TIMERx 比较 1 事件可以产生置位请求。 0：Slave_TIMERx 比较 1 事件不生成置位请求。 1：Slave_TIMERx 比较 1 事件生成置位请求。
3	CH1SCMP0	Slave_TIMERx 比较 0 事件生成通道 1 置位请求 该位置 1 时，Slave_TIMERx 比较 0 事件可以产生置位请求。 0：Slave_TIMERx 比较 0 事件不生成置位请求。 1：Slave_TIMERx 比较 0 事件生成置位请求。
2	CH1SPER	Slave_TIMERx 周期事件生成通道 1 置位请求 该位置 1 时，Slave_TIMERx 周期事件可以产生置位请求。 0：Slave_TIMERx 周期事件不生成置位请求。 1：Slave_TIMERx 周期事件生成置位请求。
1	CH1SRST	Slave_TIMERx 复位事件生成通道 1 置位请求 该位置 1 时，由软件和同步输入引起的 Slave_TIMERx 复位事件，生成通道 1 置位请求。 0：该事件不生成置位请求。 1：该事件生成置位请求。 **注意：**该位置 1 时，其他的定时器复位事件不会影响输出。
0	CH1SSEV	软件事件生成通道 1 置位请求 该位由软件置 1，由硬件自动清除。该位置 1 时，生成通道 1 置位请求。 0：该事件不生成置位请求 1：该事件生成置位请求 **注意：**该位不进行预装载。

Slave_TIMERx 通道 1 复位请求寄存器 (HRTIMER_STxCH1RST)

地址偏移：0x48

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH1RSUP	CH1RSEXEV9	CH1RSEXEV8	CH1RSEXEV7	CH1RSEXEV6	CH1RSEXEV5	CH1RSEXEV4	CH1RSEXEV3	CH1RSEXEV2	CH1RSEXEV1	CH1RSEXEV0	CH1RSSTEV8	CH1RSSTEV7	CH1RSSTEV6	CH1RSSTEV5	CH1RSSTEV4
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1RSSTEV3	CH1RSSTEV2	CH1RSSTEV1	CH1RSSTEV0	CH1RSMTCMP3	CH1RSMTCMP2	CH1RSMTCMP1	CH1RSMTCMP0	CH1RSMTPER	CH1RSCMP3	CH1RSCMP2	CH1RSCMP1	CH1RSCMP0	CH1RSPER	CH1RSRST	CH1RSSEV

rw rw rw rw rw rw rw rw rw rw rw rw rw rw rw rw

位/位域	名称	描述
31	CH1RSUP	更新事件生成通道 1 复位请求 该位置 1 时，更新事件可以产生复位请求。 0：更新事件不生成复位请求。 1：更新事件生成复位请求。
30	CH1RSEXEV9	外部事件 9 生成通道 1 复位请求 请参考 CH1RSEXEV0 说明。
29	CH1RSEXEV8	外部事件 8 生成通道 1 复位请求 请参考 CH1RSEXEV0 说明。
28	CH1RSEXEV7	外部事件 9 生成通道 1 复位请求 请参考 CH1RSEXEV0 说明。
27	CH1RSEXEV6	外部事件 6 生成通道 1 复位请求 请参考 CH1RSEXEV0 说明。
26	CH1RSEXEV5	外部事件 5 生成通道 1 复位请求 请参考 CH1RSEXEV0 说明。
25	CH1RSEXEV4	外部事件 4 生成通道 1 复位请求 请参考 CH1RSEXEV0 说明。
24	CH1RSEXEV3	外部事件 3 生成通道 1 复位请求 请参考 CH1RSEXEV0 说明。
23	CH1RSEXEV2	外部事件 2 生成通道 1 复位请求 请参考 CH1RSEXEV0 说明。
22	CH1RSEXEV1	外部事件 1 生成通道 1 复位请求 请参考 CH1RSEXEV0 说明。
21	CH1RSEXEV0	外部事件 0 生成通道 1 复位请求 当该位置 1 时，外部事件 0 可以产生复位请求。 0：该事件不生成复位请求。 1：该事件生成复位请求。
20	CH1RSSTEV8	Slave_TIMERx 互连事件 8 生成通道 1 复位请求 请参考 CH1RSSTEV0 说明。
19	CH1RSSTEV7	Slave_TIMERx 互连事件 7 生成通道 1 复位请求 请参考 CH1RSSTEV0 说明。
18	CH1RSSTEV6	Slave_TIMERx 互连事件 6 生成通道 1 复位请求

请参考 CH1RSSTEV0 说明。

17 CH1RSSTEV5 Slave_TIMERx 互连事件 5 生成通道 1 复位请求
请参考 CH1RSSTEV0 说明。

16 CH1RSSTEV4 Slave_TIMERx 互连事件 4 生成通道 1 复位请求
请参考 CH1RSSTEV0 说明。

15 CH1RSSTEV3 Slave_TIMERx 互连事件 3 生成通道 1 复位请求
请参考 CH1RSSTEV0 说明。

14 CH1RSSTEV2 Slave_TIMERx 互连事件 2 生成通道 1 复位请求
请参考 CH1RSSTEV0 说明。

13 CH1RSSTEV1 Slave_TIMERx 互连事件 1 生成通道 1 复位请求
请参考 CH1RSSTEV0 说明。

12 CH1RSSTEV0 Slave_TIMERx 互连事件 0 生成通道 1 复位请求
该位置 1 时，Slave_TIMERx 互连事件 1 可以产生复位请求。具体请参考***表 25-9. Slave_TIMER 内部连接事件***。
0：该事件不生成复位请求。
1：该事件生成复位请求。

11 CH1RSMTCMP3 Master_TIMER 比较 3 事件生成通道 1 复位请求
该位置 1 时，Master_TIMER 比较 3 事件可以产生复位请求。
0：Master_TIMER 比较 3 事件不生成复位请求。
1：Master_TIMER 比较 3 事件生成复位请求。

10 CH1RSMTCMP2 Master_TIMER 比较 2 事件生成通道 1 复位请求
该位置 1 时，Master_TIMER 比较 2 事件可以产生复位请求。
0：Master_TIMER 比较 2 事件不生成复位请求。
1：Master_TIMER 比较 2 事件生成复位请求。

9 CH1RSMTCMP1 Master_TIMER 比较 1 事件生成通道 1 复位请求
该位置 1 时，Master_TIMER 比较 1 事件可以产生复位请求。
0：Master_TIMER 比较 1 事件不生成复位请求。
1：Master_TIMER 比较 1 事件生成复位请求。

8 CH1RSMTCMP0 Master_TIMER 比较 0 事件生成通道 1 复位请求
该位置 1 时，Master_TIMER 比较 0 事件可以产生复位请求。
0：Master_TIMER 比较 0 事件不生成复位请求。
1：Master_TIMER 比较 0 事件生成复位请求。

7 CH1RSMTPER Master_TIMER 周期事件生成通道 1 复位请求
连续模式下，Master_TIMER 计数器翻转事件可以产生复位请求。在单脉冲模式下，Master_TIMER 计数器复位事件可以产生复位请求。

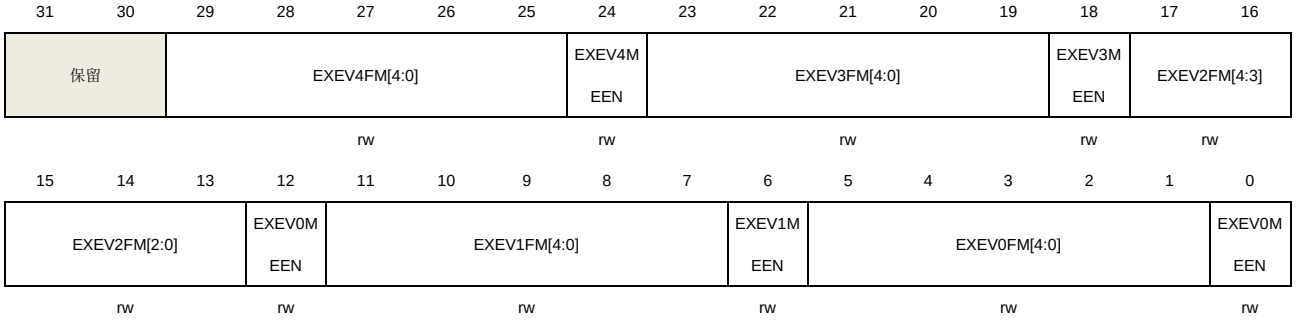
		0：该事件不生成复位请求。 1：该事件生成复位请求。
6	CH1RSCMP3	Slave_TIMERx 比较 3 事件生成通道 1 复位请求 该位置 1 时，Slave_TIMERx 比较 3 事件可以产生复位请求。 0：Slave_TIMERx 比较 3 事件不生成复位请求。 1：Slave_TIMERx 比较 3 事件生成复位请求。
5	CH1RSCMP2	Slave_TIMERx 比较 2 事件生成通道 1 复位请求 该位置 1 时，Slave_TIMERx 比较 2 事件可以产生复位请求。 0：Slave_TIMERx 比较 2 事件不生成复位请求。 1：Slave_TIMERx 比较 2 事件生成复位请求。
4	CH1RSCMP1	Slave_TIMERx 比较 1 事件生成通道 1 复位请求 该位置 1 时，Slave_TIMERx 比较 1 事件可以产生复位请求。 0：Slave_TIMERx 比较 1 事件不生成复位请求。 1：Slave_TIMERx 比较 1 事件生成复位请求。
3	CH1RSCMP0	Slave_TIMERx 比较 0 事件生成通道 1 复位请求 该位置 1 时，Slave_TIMERx 比较 0 事件可以产生复位请求。 0：Slave_TIMERx 比较 0 事件不生成复位请求。 1：Slave_TIMERx 比较 0 事件生成复位请求。
2	CH1RSPER	Slave_TIMERx 周期事件生成通道 1 复位请求 该位置 1 时，Slave_TIMERx 周期事件可以产生复位请求。 0：Slave_TIMERx 周期事件不生成复位请求。 1：Slave_TIMERx 周期事件生成复位请求。
1	CH1RSRST	Slave_TIMERx 复位事件生成通道 1 复位请求 该位置 1 时，由软件和同步输入引起的 Slave_TIMERx 复位事件，生成通道 1 复位请求。 0：该事件不生成复位请求。 1：该事件生成复位请求。 **注意：**该位置 1 时，其他的定时器复位事件不会影响输出。
0	CH1RSSEV	软件事件生成通道 1 复位请求 该位由软件置 1，由硬件自动清除。该位置 1 时，生成通道 1 复位请求。 0：该事件不生成复位请求 1：该事件生成复位请求 **注意：**该位不进行预装载。

Slave_TIMERx 外部事件滤波配置寄存器 0 (HRTIMER_STxEXEVFCFG0)

地址偏移：0x4C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。



位/位域	名称	描述
31:30	保留	必须保持复位值
29:25	EXEV4FM[4:0]	外部事件 4 滤波模式 请参考 EXEV0FM [4:0]的描述。
24	EXEV4MEEN	外部事件 4 存储功能使能 请参考 EXEV0MEEN 的描述。
23:19	EXEV3FM[4:0]	外部事件 3 滤波模式 请参考 EXEV0FM [4:0]的描述。
18	EXEV3MEEN	外部事件 3 存储功能使能 请参考 EXEV0MEEN 的描述。
17:13	EXEV2FM[4:0]	外部事件 2 滤波模式 请参考 EXEV0FM [4:0]的描述。
12	EXEV2MEEN	外部事件 2 存储功能使能 请参考 EXEV0MEEN 的描述。
11:7	EXEV1FM[4:0]	外部事件 1 滤波模式 请参考 EXEV0FM [4:0]的描述。
6	EXEV1MEEN	外部事件 1 存储功能使能 请参考 EXEV0MEEN 的描述。
5:1	EXEV0FM[4:0]	外部事件 0 滤波模式 在消隐模式下，如果外部事件在消隐期间发生，则将其忽略。在窗口模式下，仅当外部事件发生在给定的时间窗口内时，才考虑该外部事件。 00000：滤波模式禁能。 00001：消隐模式。消隐时间是从计数器复位到 HRTIMER_STxCMP0V 比较事件发生持续的时间。

00010：消隐模式。消隐时间是从计数器复位到 HRTIMER_STxCMP1V 比较事件发生持续的时间。
00011：消隐模式。消隐时间是从计数器复位到 HRTIMER_STxCMP2V 比较事件发生持续的时间。
00100：消隐模式。消隐时间是从计数器复位到 HRTIMER_STxCMP3V 比较事件发生持续的时间。
00101：消隐模式。消隐时间为其他 Slave_TIMERy（除了 Slave_TIMERx）：STBLKSRC0。
00110：消隐模式。消隐时间为其他 Slave_TIMERy（除了 Slave_TIMERx）：STBLKSRC1。
00111：消隐模式。消隐时间为其他 Slave_TIMERy（除了 Slave_TIMERx）：STBLKSRC2。
01000：消隐模式。消隐时间为其他 Slave_TIMERy（除了 Slave_TIMERx）：STBLKSRC3。
01001：消隐模式。消隐时间为其他 Slave_TIMERy（除了 Slave_TIMERx）：STBLKSRC4。
01010：消隐模式。消隐时间为其他 Slave_TIMERy（除了 Slave_TIMERx）：STBLKSRC5。
01011：消隐模式。消隐时间为其他 Slave_TIMERy（除了 Slave_TIMERx）：STBLKSRC6。
01100：消隐模式。消隐时间为其他 Slave_TIMERy（除了 Slave_TIMERx）：STBLKSRC7。
01101：窗口模式。窗口时间是从计数器复位到 HRTIMER_STxCMP1V 比较事件发生持续的时间。
01110：窗口模式。窗口时间是从计数器复位到 HRTIMER_STxCMP2V 比较事件发生持续的时间。
01111：窗口模式。窗口时间为其他 Slave_TIMERy（除了 Slave_TIMERx）：STWDSRC。
10000：消隐模式。消隐时间为其他 Slave_TIMERy（除了 Slave_TIMERx）：STBLKSRC8。
10001：消隐模式。消隐时间为其他 Slave_TIMERy（除了 Slave_TIMERx）：STBLKSRC9。
注意：
（1）一旦计数器使能（STxCEN 位置 1），不得修改该位域。
（2）用于滤波的比较寄存器的值必须大于 0。

0 EXEV0MEEN 外部事件 0 存储功能使能
0：外部事件 0 存储功能禁能。
1：外部事件 0 存储功能使能。一旦消隐周期或窗口周期完成，便会产生存储事件。
注意：
（1）一旦计数器使能（STxCEN 位置 1），不得修改该位域。

（2）该位置 1 时，可以在窗口模式下生成超时事件。

Slave_TIMERx 外部事件滤波配置寄存器 1 (HRTIMER_STxEXEVFCFG1)

地址偏移：0x50

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		EXEV9FM[4:0]					EXEV9MEEN	EXEV8FM[4:0]					EXEV8MEEN	EXEV7FM[4:3]	
		rw					rw	rw					rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXEV7FM[2:0]			EXEV7MEEN	EXEV6FM[4:0]					EXEV6MEEN	EXEV5FM[4:0]					EXEV5MEEN
rw			rw	rw					rw	rw					rw

位/位域	名称	描述
31:30	保留	必须保持复位值
29:25	EXEV9FM[4:0]	外部事件 9 滤波模式 请参考 HRTIMER_STxEXEVFCFG0 寄存器中的 EXEV0FM [4:0]的描述。
24	EXEV9MEEN	外部事件 9 存储功能使能 请参考 HRTIMER_STxEXEVFCFG0 寄存器中的 EXEV0MEEN 的描述。
23:19	EXEV8FM[4:0]	外部事件 8 滤波模式 请参考 HRTIMER_STxEXEVFCFG0 寄存器中的 EXEV0FM [4:0]的描述。
18	EXEV8MEEN	外部事件 8 存储功能使能 请参考 HRTIMER_STxEXEVFCFG0 寄存器中的 EXEV0MEEN 的描述。
17:13	EXEV7FM[4:0]	外部事件 7 滤波模式 请参考 HRTIMER_STxEXEVFCFG0 寄存器中的 EXEV0FM [4:0]的描述。
12	EXEV7MEEN	外部事件 7 存储功能使能 请参考 HRTIMER_STxEXEVFCFG0 寄存器中的 EXEV0MEEN 的描述。
11:7	EXEV6FM[3:0]	外部事件 6 滤波模式 请参考 HRTIMER_STxEXEVFCFG0 寄存器中的 EXEV0FM [4:0]的描述。
6	EXEV6MEEN	外部事件 6 存储功能使能 请参考 HRTIMER_STxEXEVFCFG0 寄存器中的 EXEV0MEEN 的描述。
5:1	EXEV5FM[4:0]	外部事件 5 滤波模式

请参考 HRTIMER_STxEXEVFCFG0 寄存器中的 EXEV0FM [4:0]的描述。

0	EXEV5MEEN	外部事件 5 存储功能使能 请参考 HRTIMER_STxEXEVFCFG0 寄存器中的 EXEV0MEEN 的描述。

Slave_TIMERx 计数器复位寄存器(HRTIMER_STxCNTRST)

地址偏移：0x54

复位值：0x0000 0000

For Slave_TIMER0

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ST5CMP1RST	ST4CMP3RST	ST4CMP1RST	ST4CMP0RST	ST3CMP3RST	ST3CMP1RST	ST3CMP0RST	ST2CMP3RST	ST2CMP1RST	ST2CMP0RST	ST1CMP3RST	ST1CMP1RST	ST1CMP0RST	EXEV9RST	EXEV8RST	EXEV7RST
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXEV6RST	EXEV5RST	EXEV4RST	EXEV3RST	EXEV2RST	EXEV1RST	EXEV0RST	MTCMP3RST	MTCMP2RST	MTCMP1RST	MTCMP0RST	MTPERRST	CMP3RST	CMP1RST	UPRST	ST5CMP0RST
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	ST5CMP1RST	Slave_TIMER5 比较 1 事件复位计数器 请参考 ST5CMP1RST 说明。
30	ST4CMP3RST	Slave_TIMER4 比较 3 事件复位计数器 请参考 ST4CMP0RST 说明。
29	ST4CMP1RST	Slave_TIMER4 比较 1 事件复位计数器 请参考 ST4CMP0RST 说明。
28	ST4CMP0RST	Slave_TIMER4 比较 0 事件复位计数器 该位用于配置 Slave_TIMER4 比较 0 事件复位计数器。 0：Slave_TIMER4 比较 0 事件不复位计数器。 1：Slave_TIMER4 比较 0 事件复位计数器。
27	ST3CMP3RST	Slave_TIMER3 比较 3 事件复位计数器 请参考 ST3CMP0RST 说明。
26	ST3CMP1RST	Slave_TIMER3 比较 1 事件复位计数器 请参考 ST3CMP0RST 说明。
25	ST3CMP0RST	Slave_TIMER3 比较 0 事件复位计数器

		该位用于配置 Slave_TIMER3 比较 0 事件复位计数器。 0：Slave_TIMER3 比较 0 事件不复位计数器。 1：Slave_TIMER3 比较 0 事件复位计数器。
24	ST2CMP3RST	Slave_TIMER2 比较 3 事件复位计数器 请参考 ST2CMP0RST 说明。
23	ST2CMP1RST	Slave_TIMER2 比较 1 事件复位计数器 请参考 ST2CMP0RST 说明。
22	ST2CMP0RST	Slave_TIMER2 比较 0 事件复位计数器 该位用于配置 Slave_TIMER2 比较 0 事件复位计数器。 0：Slave_TIMER2 比较 0 事件不复位计数器。 1：Slave_TIMER2 比较 0 事件复位计数器。
21	ST1CMP3RST	Slave_TIMER1 比较 3 事件复位计数器 请参考 ST1CMP0RST 说明。
20	ST1CMP1RST	Slave_TIMER1 比较 1 事件复位计数器 请参考 ST1CMP0RST 说明。
19	ST1CMP0RST	Slave_TIMER1 比较 0 事件复位计数器 该位用于配置 Slave_TIMER1 比较 0 事件复位计数器。 0：Slave_TIMER1 比较 0 事件不复位计数器。 1：Slave_TIMER1 比较 0 事件复位计数器。
18	EXEV9RST	外部事件 9 复位计数器 请参考 EXEV0RST 说明。
17	EXEV8RST	外部事件 8 复位计数器 请参考 EXEV0RST 说明。
16	EXEV7RST	外部事件 7 复位计数器 请参考 EXEV0RST 说明。
15	EXEV6RST	外部事件 6 复位计数器 请参考 EXEV0RST 说明。.
14	EXEV5RST	外部事件 5 复位计数器 请参考 EXEV0RST 说明。
13	EXEV4RST	外部事件 4 复位计数器 请参考 EXEV0RST 说明。
12	EXEV3RST	外部事件 3 复位计数器 请参考 EXEV0RST 说明。

11	EXEV2RST	外部事件 2 复位计数器 请参考 EXEV0RST 说明。
10	EXEV1RST	外部事件 1 复位计数器 请参考 EXEV0RST 说明。
9	EXEV0RST	外部事件 0 复位计数器 0：外部事件 0 不复位计数器。 1：外部事件 0 复位计数器。
8	MTCMP3RST	Master_TIMER 比较 3 事件复位计数器 请参考 MTCMP0RST 说明。
7	MTCMP2RST	Master_TIMER 比较 2 事件复位计数器 请参考 MTCMP0RST 说明。
6	MTCMP1RST	Master_TIMER 比较 1 事件复位计数器 请参考 MTCMP0RST 说明。
5	MTCMP0RST	Master_TIMER 比较 0 事件复位计数器 该位用于配置 Master_TIMER 比较 0 事件复位计数器。 0：Master_TIMER 比较 0 事件不复位计数器。 1：Master_TIMER 比较 0 事件复位计数器。
4	MTPERRST	Master_TIMER 周期事件复位计数器 该位用于配置 Master_TIMER 周期事件复位计数器。 0：Master_TIMER 周期事件不复位计数器。 1：Master_TIMER 周期事件复位计数器。
3	CMP3RST	Slave_TIMER0 比较 3 事件复位计数器 请参考 CMP1RST 说明。
2	CMP1RST	Slave_TIMER0 比较 1 事件复位计数器 该位用于配置比较 1 事件复位计数器。 0：比较 1 事件不复位计数器。 1：比较 1 事件复位计数器。
1	UPRST	Slave_TIMER0 更新事件复位计数器 该位用于配置更新事件复位计数器。 0：更新事件不复位计数器。 1：更新事件复位计数器。
0	ST5CMP0RST	Slave_TIMER5 比较 0 事件复位计数器 请参考 ST1CMP0RST 说明。

For Slave_TIMER1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ST5CMP1RST	ST4CMP3RST	ST4CMP1RST	ST4CMP0RST	ST3CMP3RST	ST3CMP1RST	ST3CMP0RST	ST2CMP3RST	ST2CMP1RST	ST2CMP0RST	ST0CMP3RST	ST0CMP1RST	ST0CMP0RST	EXEV9RST	EXEV8RST	EXEV7RST
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
EXEV6RST	EXEV5RST	EXEV4RST	EXEV3RST	EXEV2RST	EXEV1RST	EXEV0RST	MTCMP3RST	MTCMP2RST	MTCMP1RST	MTCMP0RST	MTPERRST	CMP3RST	CMP1RST	UPRST	ST5CMP0RST
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	ST5CMP1RST	Slave_TIMER5 比较 1 事件复位计数器 请参考 ST5CMP1RST 说明。
30	ST4CMP3RST	Slave_TIMER4 比较 3 事件复位计数器 请参考 ST4CMP0RST 说明。
29	ST4CMP1RST	Slave_TIMER4 比较 1 事件复位计数器 请参考 ST4CMP0RST 说明。
28	ST4CMP0RST	Slave_TIMER4 比较 0 事件复位计数器 该位用于配置 Slave_TIMER4 比较 0 事件复位计数器。 0：Slave_TIMER4 比较 0 事件不复位计数器 1：Slave_TIMER4 比较 0 事件复位计数器
27	ST3CMP3RST	Slave_TIMER3 比较 3 事件复位计数器 请参考 ST3CMP0RST 说明。
26	ST3CMP1RST	Slave_TIMER3 比较 1 事件复位计数器 请参考 ST3CMP0RST 说明。
25	ST3CMP0RST	Slave_TIMER3 比较 0 事件复位计数器 该位用于配置 Slave_TIMER3 比较 0 事件复位计数器。 0：Slave_TIMER3 比较 0 事件不复位计数器 1：Slave_TIMER3 比较 0 事件复位计数器
24	ST2CMP3RST	Slave_TIMER2 比较 3 事件复位计数器 请参考 ST2CMP0RST 说明。
23	ST2CMP1RST	Slave_TIMER2 比较 1 事件复位计数器 请参考 ST2CMP0RST 说明。
22	ST2CMP0RST	Slave_TIMER2 比较 0 事件复位计数器 该位用于配置 Slave_TIMER2 比较 0 事件复位计数器。

		0：Slave_TIMER2 比较 0 事件不复位计数器 1：Slave_TIMER2 比较 0 事件复位计数器
21	ST0CMP3RST	Slave_TIMER0 比较 3 事件复位计数器 请参考 ST0CMP0RST 说明。
20	ST0CMP1RST	Slave_TIMER0 比较 1 事件复位计数器 请参考 ST0CMP0RST 说明。
19	ST0CMP0RST	Slave_TIMER0 比较 0 事件复位计数器 该位用于配置 Slave_TIMER0 比较 0 事件复位计数器。 0：Slave_TIMER0 比较 0 事件不复位计数器 1：Slave_TIMER0 比较 0 事件复位计数器
18	EXEV9RST	外部事件 9 复位计数器 请参考 EXEV0RST 说明。
17	EXEV8RST	外部事件 8 复位计数器 请参考 EXEV0RST 说明。
16	EXEV7RST	外部事件 7 复位计数器 请参考 EXEV0RST 说明。
15	EXEV6RST	外部事件 6 复位计数器 请参考 EXEV0RST 说明。
14	EXEV5RST	外部事件 5 复位计数器 请参考 EXEV0RST 说明。
13	EXEV4RST	外部事件 4 复位计数器 请参考 EXEV0RST 说明。
12	EXEV3RST	外部事件 3 复位计数器 请参考 EXEV0RST 说明。
11	EXEV2RST	外部事件 2 复位计数器 请参考 EXEV0RST 说明。
10	EXEV1RST	外部事件 1 复位计数器 请参考 EXEV0RST 说明。
9	EXEV0RST	外部事件 0 复位计数器 0：外部事件 0 不复位计数器 1：外部事件 0 复位计数器
8	MTCMP3RST	Master_TIMER 比较 3 事件复位计数器 请参考 MTCMP0RST 说明。

7	MTCMP2RST	Master_TIMER 比较 2 事件复位计数器 请参考 MTCMP0RST 说明。
6	MTCMP1RST	Master_TIMER 比较 1 事件复位计数器 请参考 MTCMP0RST 说明。
5	MTCMP0RST	Master_TIMER 比较 0 事件复位计数器 该位用于配置 Master_TIMER 比较 0 事件复位计数器。 0：Master_TIMER 比较 0 事件不复位计数器 1：Master_TIMER 比较 0 事件复位计数器
4	MTPERRST	Master_TIMER 周期事件复位计数器 该位用于配置 Master_TIMER 周期事件复位计数器。 0：Master_TIMER 周期事件不复位计数器 1：Master_TIMER 周期事件复位计数器
3	CMP3RST	Slave_TIMER1 比较 3 事件复位计数器 请参考 CMP1RST 说明。
2	CMP1RST	Slave_TIMER1 比较 1 事件复位计数器 该位用于配置比较 1 事件复位计数器。 0：比较 1 事件不复位计数器 1：比较 1 事件复位计数器
1	UPRST	Slave_TIMER1 更新事件复位计数器 该位用于配置更新事件复位计数器。 0：更新事件不复位计数器 1：更新事件复位计数器
0	ST5CMP0RST	Slave_TIMER5 比较 0 事件复位计数器 请参考 ST5CMP0RST 说明。

For Slave_TIMER2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ST5CMP1RST	ST4CMP3RST	ST4CMP1RST	ST4CMP0RST	ST3CMP3RST	ST3CMP1RST	ST3CMP0RST	ST1CMP3RST	ST1CMP1RST	ST1CMP0RST	ST0CMP3RST	ST0CMP1RST	ST0CMP0RST	EXEV9RST	EXEV8RST	EXEV7RST
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
EXEV6RST	EXEV5RST	EXEV4RST	EXEV3RST	EXEV2RST	EXEV1RST	EXEV0RST	MTCMP3RST	MTCMP2RST	MTCMP1RST	MTCMP0RST	MTPERRST	CMP3RST	CMP1RST	UPRST	ST5CMP0RST
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述

31	ST5CMP1RST	Slave_TIMER5 比较 1 事件复位计数器 请参考 ST4CMP0RST 说明。
30	ST4CMP3RST	Slave_TIMER4 比较 3 事件复位计数器 请参考 ST4CMP0RST 说明。
29	ST4CMP1RST	Slave_TIMER4 比较 1 事件复位计数器 请参考 ST4CMP0RST 说明。
28	ST4CMP0RST	Slave_TIMER4 比较 0 事件复位计数器 该位用于配置 Slave_TIMER4 比较 0 事件复位计数器。 0：Slave_TIMER4 比较 0 事件不复位计数器 1：Slave_TIMER4 比较 0 事件复位计数器
27	ST3CMP3RST	Slave_TIMER3 比较 3 事件复位计数器 请参考 ST3CMP0RST 说明。
26	ST3CMP1RST	Slave_TIMER3 比较 1 事件复位计数器 请参考 ST3CMP0RST 说明。
25	ST3CMP0RST	Slave_TIMER3 比较 0 事件复位计数器 该位用于配置 Slave_TIMER3 比较 0 事件复位计数器。 0：Slave_TIMER3 比较 0 事件不复位计数器 1：Slave_TIMER3 比较 0 事件复位计数器
24	ST1CMP3RST	Slave_TIMER1 比较 3 事件复位计数器 请参考 ST1CMP0RST 说明。
23	ST1CMP1RST	Slave_TIMER1 比较 1 事件复位计数器 请参考 ST1CMP0RST 说明。
22	ST1CMP0RST	Slave_TIMER1 比较 0 事件复位计数器 该位用于配置 Slave_TIMER1 比较 0 事件复位计数器。 0：Slave_TIMER1 比较 0 事件不复位计数器 1：Slave_TIMER1 比较 0 事件复位计数器
21	ST0CMP3RST	Slave_TIMER0 比较 3 事件复位计数器 请参考 ST0CMP0RST 说明。
20	ST0CMP1RST	Slave_TIMER0 比较 1 事件复位计数器 请参考 ST0CMP0RST 说明。
19	ST0CMP0RST	Slave_TIMER0 比较 0 事件复位计数器 该位用于配置 Slave_TIMER0 比较 0 事件复位计数器。 0：Slave_TIMER0 比较 0 事件不复位计数器 1：Slave_TIMER0 比较 0 事件复位计数器

18	EXEV9RST	外部事件 9 复位计数器 请参考 EXEV0RST 说明。
17	EXEV8RST	外部事件 8 复位计数器 请参考 EXEV0RST 说明。
16	EXEV7RST	外部事件 7 复位计数器 请参考 EXEV0RST 说明。
15	EXEV6RST	外部事件 6 复位计数器 请参考 EXEV0RST 说明。
14	EXEV5RST	外部事件 5 复位计数器 请参考 EXEV0RST 说明。
13	EXEV4RST	外部事件 4 复位计数器 请参考 EXEV0RST 说明。
12	EXEV3RST	外部事件 3 复位计数器 请参考 EXEV0RST 说明。
11	EXEV2RST	外部事件 2 复位计数器 请参考 EXEV0RST 说明。
10	EXEV1RST	外部事件 1 复位计数器 请参考 EXEV0RST 说明。
9	EXEV0RST	外部事件 0 复位计数器 0：外部事件 0 不复位计数器 1：外部事件 0 复位计数器
8	MTCMP3RST	Master_TIMER 比较 3 事件复位计数器 请参考 MTCMP0RST 说明。
7	MTCMP2RST	Master_TIMER 比较 2 事件复位计数器 请参考 MTCMP0RST 说明。
6	MTCMP1RST	Master_TIMER 比较 1 事件复位计数器 请参考 MTCMP0RST 说明。
5	MTCMP0RST	Master_TIMER 比较 0 事件复位计数器 该位用于配置 Master_TIMER 比较 0 事件复位计数器。 0：Master_TIMER 比较 0 事件不复位计数器。 1：Master_TIMER 比较 0 事件复位计数器。
4	MTPERRST	Master_TIMER 周期事件复位计数器 该位用于配置 Master_TIMER 周期事件复位计数器。

0：Master_TIMER 周期事件不复位计数器。
1：Master_TIMER 周期事件复位计数器。

3 CMP3RST Slave_TIMER2 比较 3 事件复位计数器
请参考 CMP1RST 说明。

2 CMP1RST Slave_TIMER2 比较 1 事件复位计数器
该位用于配置比较 1 事件复位计数器。
0：比较 1 事件不复位计数器。
1：比较 1 事件复位计数器。

1 UPRST Slave_TIMER2 更新事件复位计数器
该位用于配置更新事件复位计数器。
0：更新事件不复位计数器
1：更新事件复位计数器

0 ST5CMP0RST Slave_TIMER5 比较 0 事件复位计数器
请参考 ST5CMP0RST 说明。

For Slave_TIMER3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ST5CMP1RST	ST4CMP3RST	ST4CMP1RST	ST4CMP0RST	ST2CMP3RST	ST2CMP1RST	ST2CMP0RST	ST1CMP3RST	ST1CMP1RST	ST1CMP0RST	ST0CMP3RST	ST0CMP1RST	ST0CMP0RST	EXEV9RST	EXEV8RST	EXEV7RST
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXEV6RST	EXEV5RST	EXEV4RST	EXEV3RST	EXEV2RST	EXEV1RST	EXEV0RST	MTCMP3RST	MTCMP2RST	MTCMP1RST	MTCMP0RST	MTPERRST	CMP3RST	CMP1RST	UPRST	ST5CMP0RST
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	ST5CMP1RST	Slave_TIMER5 比较 1 事件复位计数器 请参考 ST4CMP0RST 说明。
30	ST4CMP3RST	Slave_TIMER4 比较 3 事件复位计数器 请参考 ST4CMP0RST 说明。
29	ST4CMP1RST	Slave_TIMER4 比较 1 事件复位计数器 请参考 ST4CMP0RST 说明。
28	ST4CMP0RST	Slave_TIMER4 比较 0 事件复位计数器 该位用于配置 Slave_TIMER4 比较 0 事件复位计数器。 0：Slave_TIMER4 比较 0 事件不复位计数器。 1：Slave_TIMER4 比较 0 事件复位计数器。

27	ST2CMP3RST	Slave_TIMER2 比较 3 事件复位计数器 请参考 ST2CMP0RST 说明。
26	ST2CMP1RST	Slave_TIMER2 比较 1 事件复位计数器 请参考 ST2CMP0RST 说明。
25	ST2CMP0RST	Slave_TIMER2 比较 0 事件复位计数器 该位用于配置 Slave_TIMER2 比较 0 事件复位计数器。 0：Slave_TIMER2 比较 0 事件不复位计数器。 1：Slave_TIMER2 比较 0 事件复位计数器。
24	ST1CMP3RST	Slave_TIMER1 比较 3 事件复位计数器 请参考 ST1CMP0RST 说明。
23	ST1CMP1RST	Slave_TIMER1 比较 1 事件复位计数器 请参考 ST1CMP0RST 说明。
22	ST1CMP0RST	Slave_TIMER1 比较 0 事件复位计数器 该位用于配置 Slave_TIMER1 比较 0 事件复位计数器。 0：Slave_TIMER2 比较 0 事件不复位计数器。 1：Slave_TIMER1 比较 0 事件复位计数器。
21	ST0CMP3RST	Slave_TIMER0 比较 3 事件复位计数器 请参考 ST0CMP0RST 说明。
20	ST0CMP1RST	Slave_TIMER0 比较 1 事件复位计数器 请参考 ST0CMP0RST 说明。
19	ST0CMP0RST	Slave_TIMER0 比较 0 事件复位计数器 该位用于配置 Slave_TIMER0 比较 0 事件复位计数器。 0：Slave_TIMER0 比较 0 事件不复位计数器。 1：Slave_TIMER0 比较 0 事件复位计数器。
18	EXEV9RST	外部事件 9 复位计数器 请参考 EXEV0RST 说明。
17	EXEV8RST	外部事件 8 复位计数器 请参考 EXEV0RST 说明。
16	EXEV7RST	外部事件 7 复位计数器 请参考 EXEV0RST 说明。
15	EXEV6RST	外部事件 6 复位计数器 请参考 EXEV0RST 说明。
14	EXEV5RST	外部事件 5 复位计数器

		请参考 EXEV0RST 说明。
13	EXEV4RST	外部事件 4 复位计数器 请参考 EXEV0RST 说明。
12	EXEV3RST	外部事件 3 复位计数器 请参考 EXEV0RST 说明。
11	EXEV2RST	外部事件 2 复位计数器 请参考 EXEV0RST 说明。
10	EXEV1RST	外部事件 1 复位计数器 请参考 EXEV0RST 说明。
9	EXEV0RST	外部事件 0 复位计数器 该位用于配置外部事件 0 复位计数器。 0：外部事件 0 不复位计数器。 1：外部事件 0 复位计数器。
8	MTCMP3RST	Master_TIMER 比较 3 事件复位计数器 请参考 MTCMP0RST 说明。
7	MTCMP2RST	Master_TIMER 比较 2 事件复位计数器 请参考 MTCMP0RST 说明。
6	MTCMP1RST	Master_TIMER 比较 1 事件复位计数器 请参考 MTCMP0RST 说明。
5	MTCMP0RST	Master_TIMER 比较 0 事件复位计数器 该位用于配置 Master_TIMER 比较 0 事件复位计数器。 0：Master_TIMER 比较 0 事件不复位计数器。 1：Master_TIMER 比较 0 事件复位计数器。
4	MTPERRST	Master_TIMER 周期事件复位计数器 该位用于配置 Master_TIMER 周期事件复位计数器。 0：Master_TIMER 周期事件不复位计数器。 1：Master_TIMER 周期事件复位计数器。
3	CMP3RST	Slave_TIMER3 比较 3 事件复位计数器 请参考 CMP1RST 说明。
2	CMP1RST	Slave_TIMER3 比较 1 事件复位计数器 该位用于配置比较 1 事件复位计数器。 0：比较 1 事件不复位计数器。 1：比较 1 事件复位计数器。
1	UPRST	Slave_TIMER3 更新事件复位计数器

该位用于配置更新事件复位计数器。
0：更新事件不复位计数器。
1：更新事件复位计数器。

0 ST5CMP0RST Slave_TIMER5 比较 0 事件复位计数器
请参考 ST5CMP0RST 说明。

For Slave_TIMER4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ST5CMP1RST	ST3CMP3RST	ST3CMP1RST	ST3CMP0RST	ST2CMP3RST	ST2CMP1RST	ST2CMP0RST	ST1CMP3RST	ST1CMP1RST	ST1CMP0RST	ST0CMP3RST	ST0CMP1RST	ST0CMP0RST	EXEV9RST	EXEV8RST	EXEV7RST
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXEV6RST	EXEV5RST	EXEV4RST	EXEV3RST	EXEV2RST	EXEV1RST	EXEV0RST	MTCMP3RST	MTCMP2RST	MTCMP1RST	MTCMP0RST	MTPERRST	CMP3RST	CMP1RST	UPRST	ST5CMP0RST
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	ST5CMP1RST	Slave_TIMER5 比较 1 事件复位计数器 请参考 ST4CMP0RST 说明。
30	ST3CMP3RST	Slave_TIMER3 比较 3 事件复位计数器 请参考 ST3CMP0RST 说明。
29	ST3CMP1RST	Slave_TIMER3 比较 1 事件复位计数器 请参考 ST3CMP0RST 说明。
28	ST3CMP0RST	Slave_TIMER3 比较 0 事件复位计数器 该位用于配置 Slave_TIMER3 比较 0 事件复位计数器。 0：Slave_TIMER3 比较 0 事件不复位计数器。 1：Slave_TIMER3 比较 0 事件复位计数器。
27	ST2CMP3RST	Slave_TIMER2 比较 3 事件复位计数器 请参考 ST2CMP0RST 说明。
26	ST2CMP1RST	Slave_TIMER2 比较 1 事件复位计数器 请参考 ST2CMP0RST 说明。
25	ST2CMP0RST	Slave_TIMER2 比较 0 事件复位计数器 该位用于配置 Slave_TIMER2 比较 0 事件复位计数器。 0：Slave_TIMER2 比较 0 事件不复位计数器。 1：Slave_TIMER2 比较 0 事件复位计数器。
24	ST1CMP3RST	Slave_TIMER1 比较 3 事件复位计数器

		请参考 ST1CMP0RST 说明。
23	ST1CMP1RST	Slave_TIMER1 比较 1 事件复位计数器 请参考 ST1CMP0RST 说明。
22	ST1CMP0RST	Slave_TIMER1 比较 0 事件复位计数器 该位用于配置 Slave_TIMER1 比较 0 事件复位计数器。 0：Slave_TIMER1 比较 0 事件不复位计数器。 1：Slave_TIMER1 比较 0 事件复位计数器。
21	ST0CMP3RST	Slave_TIMER0 比较 3 事件复位计数器 请参考 ST0CMP0RST 说明。
20	ST0CMP1RST	Slave_TIMER0 比较 1 事件复位计数器 请参考 ST0CMP0RST 说明。
19	ST0CMP0RST	Slave_TIMER0 比较 0 事件复位计数器 该位用于配置 Slave_TIMER0 比较 0 事件复位计数器。 0：Slave_TIMER0 比较 0 事件不复位计数器。 1：Slave_TIMER0 比较 0 事件复位计数器。
18	EXEV9RST	外部事件 9 复位计数器 请参考 EXEV0RST 说明。
17	EXEV8RST	外部事件 8 复位计数器 请参考 EXEV0RST 说明。
16	EXEV7RST	外部事件 7 复位计数器 请参考 EXEV0RST 说明。
15	EXEV6RST	外部事件 6 复位计数器 请参考 EXEV0RST 说明。
14	EXEV5RST	外部事件 5 复位计数器 请参考 EXEV0RST 说明。
13	EXEV4RST	外部事件 4 复位计数器 请参考 EXEV0RST 说明。
12	EXEV3RST	外部事件 3 复位计数器 请参考 EXEV0RST 说明。
11	EXEV2RST	外部事件 2 复位计数器 请参考 EXEV0RST 说明。
10	EXEV1RST	外部事件 1 复位计数器 请参考 EXEV0RST 说明。

9	EXEV0RST	外部事件 0 复位计数器 0：外部事件 0 不复位计数器 1：外部事件 0 复位计数器
8	MTCMP3RST	Master_TIMER 比较 3 事件复位计数器 请参考 MTCMP0RST 说明。
7	MTCMP2RST	Master_TIMER 比较 2 事件复位计数器 请参考 MTCMP0RST 说明。
6	MTCMP1RST	Master_TIMER 比较 1 事件复位计数器 请参考 MTCMP0RST 说明。
5	MTCMP0RST	Master_TIMER 比较 0 事件复位计数器 该位用于配置 Master_TIMER 比较 0 事件复位计数器。 0：Master_TIMER 比较 0 事件不复位计数器。 1：Master_TIMER 比较 0 事件复位计数器。
4	MTPERRST	Master_TIMER 周期事件复位计数器 该位用于配置 Master_TIMER 周期事件复位计数器。 0：Master_TIMER 周期事件不复位计数器。 1：Master_TIMER 周期事件复位计数器。
3	CMP3RST	Slave_TIMER4 比较 3 事件复位计数器 请参考 CMP1RST 说明。
2	CMP1RST	Slave_TIMER4 比较 1 事件复位计数器 该位用于配置比较 1 事件复位计数器。 0：比较 1 事件不复位计数器。 1：比较 1 事件复位计数器。
1	UPRST	Slave_TIMER4 更新事件复位计数器 该位用于配置更新事件复位计数器。 0：更新事件不复位计数器。 1：更新事件复位计数器。
0	ST5CMP0RST	Slave_TIMER5 比较 0 事件复位计数器 请参考 ST5CMP0RST 说明。

For Slave_TIMER5 - For Slave_TIMER7

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ST4CMP1RST	ST3CMP3RST	ST3CMP1RST	ST3CMP0RST	ST2CMP3RST	ST2CMP1RST	ST2CMP0RST	ST1CMP3RST	ST1CMP1RST	ST1CMP0RST	ST0CMP3RST	ST0CMP1RST	ST0CMP0RST	EXEV9RST	EXEV8RST	EXEV7RST
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXEV6RST	EXEV5RST	EXEV4RST	EXEV3RST	EXEV2RST	EXEV1RST	EXEV0RST	MTCMP3RST	MTCMP2RST	MTCMP1RST	MTCMP0RST	MTPERRST	CMP3RST	CMP1RST	UPRST	ST4CMP0RST
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	ST4CMP1RST	Slave_TIMER4 比较 1 事件复位计数器 请参考 ST4CMP0RST 说明。
30	ST3CMP3RST	Slave_TIMER3 比较 3 事件复位计数器 请参考 ST3CMP0RST 说明。
29	ST3CMP1RST	Slave_TIMER3 比较 1 事件复位计数器 请参考 ST3CMP0RST 说明。
28	ST3CMP0RST	Slave_TIMER3 比较 0 事件复位计数器 该位用于配置 Slave_TIMER3 比较 0 事件复位计数器。 0：Slave_TIMER3 比较 0 事件不复位计数器。 1：Slave_TIMER3 比较 0 事件复位计数器。
27	ST2CMP3RST	Slave_TIMER2 比较 3 事件复位计数器 请参考 ST2CMP0RST 说明。
26	ST2CMP1RST	Slave_TIMER2 比较 1 事件复位计数器 请参考 ST2CMP0RST 说明。
25	ST2CMP0RST	Slave_TIMER2 比较 0 事件复位计数器 该位用于配置 Slave_TIMER2 比较 0 事件复位计数器。 0：Slave_TIMER2 比较 0 事件不复位计数器。 1：Slave_TIMER2 比较 0 事件复位计数器。
24	ST1CMP3RST	Slave_TIMER1 比较 3 事件复位计数器 请参考 ST1CMP0RST 说明。
23	ST1CMP1RST	Slave_TIMER1 比较 1 事件复位计数器 请参考 ST1CMP0RST 说明。
22	ST1CMP0RST	Slave_TIMER1 比较 0 事件复位计数器 该位用于配置 Slave_TIMER1 比较 0 事件复位计数器。 0：Slave_TIMER1 比较 0 事件不复位计数器。 1：Slave_TIMER1 比较 0 事件复位计数器。
21	ST0CMP3RST	Slave_TIMER0 比较 3 事件复位计数器 请参考 ST0CMP0RST 说明。
20	ST0CMP1RST	Slave_TIMER0 比较 1 事件复位计数器

		请参考 ST0CMP0RST 说明。
19	ST0CMP0RST	Slave_TIMER0 比较 0 事件复位计数器 该位用于配置 Slave_TIMER0 比较 0 事件复位计数器。 0：Slave_TIMER0 比较 0 事件不复位计数器。 1：Slave_TIMER0 比较 0 事件复位计数器。
18	EXEV9RST	外部事件 9 复位计数器 请参考 EXEV0RST 说明。
17	EXEV8RST	外部事件 8 复位计数器 请参考 EXEV0RST 说明。
16	EXEV7RST	外部事件 7 复位计数器 请参考 EXEV0RST 说明。
15	EXEV6RST	外部事件 6 复位计数器 请参考 EXEV0RST 说明。
14	EXEV5RST	外部事件 5 复位计数器 请参考 EXEV0RST 说明。
13	EXEV4RST	外部事件 4 复位计数器 请参考 EXEV0RST 说明。
12	EXEV3RST	外部事件 3 复位计数器 请参考 EXEV0RST 说明。
11	EXEV2RST	外部事件 2 复位计数器 请参考 EXEV0RST 说明。
10	EXEV1RST	外部事件 1 复位计数器 请参考 EXEV0RST 说明。
9	EXEV0RST	外部事件 0 复位计数器 0：外部事件 0 不复位计数器 1：外部事件 0 复位计数器
8	MTCMP3RST	Master_TIMER 比较 3 事件复位计数器 请参考 MTCMP0RST 说明。
7	MTCMP2RST	Master_TIMER 比较 2 事件复位计数器 请参考 MTCMP0RST 说明。
6	MTCMP1RST	Master_TIMER 比较 1 事件复位计数器 请参考 MTCMP0RST 说明。
5	MTCMP0RST	Master_TIMER 比较 0 事件复位计数器

		该位用于配置 Master_TIMER 比较 0 事件复位计数器。 0：Master_TIMER 比较 0 事件不复位计数器。 1：Master_TIMER 比较 0 事件复位计数器。
4	MTPERRST	Master_TIMER 周期事件复位计数器 该位用于配置 Master_TIMER 周期事件复位计数器。 0：Master_TIMER 周期事件不复位计数器。 1：Master_TIMER 周期事件复位计数器。
3	CMP3RST	Slave_TIMERy (y = 5 - 7)比较 3 事件复位计数器 请参考 CMP1RST 说明。
2	CMP1RST	Slave_TIMERy (y = 5 - 7)比较 1 事件复位计数器 该位用于配置比较 1 事件复位计数器。 0：比较 1 事件不复位计数器。 1：比较 1 事件复位计数器。
1	UPRST	Slave_TIMERy (y = 5 - 7)更新事件复位计数器 该位用于配置更新事件复位计数器。 0：更新事件不复位计数器。 1：更新事件复位计数器。
0	ST4CMP0RST	Slave_TIMER4 比较 0 事件复位计数器 请参考 ST4CMP0RST 说明。

Slave_TIMERx 载波控制寄存器 (HRTIMER_STxCSCTL)

地址偏移：0x58

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

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

位/位域	名称	描述
31:11	保留	必须保持复位值
10:7	CSFSTPW[3:0]	第一个载波信号的脉冲宽度 该位域定义了在通道输出准备信号（CHxOPRE）上升沿之后的第一个载波信号的脉冲宽度。

		$t_{CSFSTPW}$ = (CSFSTPW[3:0]+1) x $t_{HRTIMER_CSGCK}$，$t_{HRTIMER_CSGCK}$ =16 x $t_{HRTIMER_CK}$. 0000：$t_{CSFSTPW}$ = $t_{HRTIMER_CSGCK}$ 0001：$t_{CSFSTPW}$ = 2*$t_{HRTIMER_CSGCK}$ … 1110：$t_{CSFSTPW}$ = 15*$t_{HRTIMER_CSGCK}$ 1111：$t_{CSFSTPW}$ = 16*$t_{HRTIMER_CSGCK}$
6:4	CSDTY[2:0]	载波信号占空比 该位域定义了载波信号（第一个脉冲除外）的占空比为 CSDTY [2:0] / 8。 000：0%（仅出现第一个脉冲）。 001：12.5% 010：25.0% 011：37.5% 100：50.0% 101：62.5% 110：75.0% 111：87.5%
3:0	CSPRD[3:0]	载波信号周期 该位域定义了载波信号的周期（第一个脉冲除外）。 t_{CSPRD} = (CSPRD[3:0]+1) x $t_{HRTIMER_CSGCK}$, $t_{HRTIMER_CSGCK}$ =16 x $t_{HRTIMER_CK}$. 0000: 16 x $t_{HRTIMER_CK}$ 0001: 32 x $t_{HRTIMER_CK}$ … 1111: 256 x $t_{HRTIMER_CK}$

Slave_TIMERx 捕获 0 触发寄存器 (HRTIMER_STxCAP0TRG)

地址偏移：0x5C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

For Slave_TIMER0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CP0BST4 CMP1	CP0BST4 CMP0	CP0BST4 NA	CP0BST4 A	CP0BST3 CMP1	CP0BST3 CMP0	CP0BST3 NA	CP0BST3 A	CP0BST2 MP1	CP0BST2 CMP0	CP0BST2 NA	CP0BST2 A	CP0BST1 CMP1	CP0BST1 CMP0	CP0BST1 NA	CP0BST1 A
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				CP0BEX EV9	CP0BEX EV8	CP0BEX EV7	CP0BEX EV6	CP0BEX EV5	CP0BEX EV4	CP0BEX EV3	CP0BEX EV2	CP0BEX EV1	CP0BEX EV0	CP0BUP	CP0BSW
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	CP0BST4CMP1	Slave_TIMER4 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP0BST0CMP1 描述。
30	CP0BST4CMP0	Slave_TIMER4 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP0BST0CMP0 描述。
29	CP0BST4NA	ST4CH0_O 输出有效到无效的变化触发捕获 0 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP0BST0NA 描述。
28	CP0BST4A	ST4CH0_O 输出无效到有效的变化触发捕获 0 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP0BST0A 描述。
27	CP0BST3CMP1	Slave_TIMER3 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP0BST0CMP1 描述。
26	CP0BST3CMP0	Slave_TIMER3 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP0BST0CMP0 描述。
25	CP0BST3NA	ST3CH0_O 输出有效到无效的变化触发捕获 0 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP0BST0NA 描述。
24	CP0BST3A	ST3CH0_O 输出无效到有效的变化触发捕获 0 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP0BST0A 描述。
23	CP0BST2CMP1	Slave_TIMER2 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP0BST0CMP1 描述。
22	CP0BST2CMP0	Slave_TIMER2 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP0BST0CMP0 描述。
21	CP0BST2NA	ST2CH0_O 输出有效到无效的变化触发捕获 0 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP0BST0NA 描述。

20	CP0BST2A	ST2CH0_O 输出无效到有效的变化触发捕获 0 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP0BST0A 描述。
19	CP0BST1CMP1	Slave_TIMER1 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP0BST0CMP1 描述。
18	CP0BST1CMP0	Slave_TIMER1 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP0BST0CMP0 描述。
17	CP0BST1NA	ST1CH0_O 输出有效到无效的变化触发捕获 0 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP0BST0NA 描述。
16	CP0BST1A	ST1CH0_O 输出无效到有效的变化触发捕获 0 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP0BST0A 描述。
15:12	保留	必须保持复位值
11	CP0BEXEV9	外部事件 9 触发捕获 0 请参考 CP0BEXEV0 描述。
10	CP0BEXEV8	外部事件 8 触发捕获 0 请参考 CP0BEXEV0 描述。
9	CP0BEXEV7	外部事件 7 触发捕获 0 请参考 CP0BEXEV0 描述。
8	CP0BEXEV6	外部事件 6 触发捕获 0 请参考 CP0BEXEV0 描述。
7	CP0BEXEV5	外部事件 5 触发捕获 0 请参考 CP0BEXEV0 描述。
6	CP0BEXEV4	外部事件 4 触发捕获 0 请参考 CP0BEXEV0 描述。
5	CP0BEXEV3	外部事件 3 触发捕获 0 请参考 CP0BEXEV0 描述。
4	CP0BEXEV2	外部事件 2 触发捕获 0 请参考 CP0BEXEV0 描述。
3	CP0BEXEV1	外部事件 1 触发捕获 0

请参考 CP0BEXEV0 描述。

2 CP0BEXEV0 外部事件 0 触发捕获 0
当该位置 1 时，外部事件 0 触发捕获 0。
0：外部事件 0 不触发捕获 0
1：外部事件 0 触发捕获 0

1 CP0BUP 更新事件触发捕获 0
当该位置 1 时，更新事件触发捕获 0。
0：更新事件不触发捕获 0
1：更新事件触发捕获 0

0 CP0BSW 软件触发捕获 0
该位由软件置 1，硬件自动清零。该位置 1 时，软件触发捕获 0。
0：软件不触发捕获 0
1：软件触发捕获 0

For Slave_TIMER1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CP0BST4CMP1	CP0BST4CMP0	CP0BST4NA	CP0BST4A	CP0BST3CMP1	CP0BST3CMP0	CP0BST3NA	CP0BST3A	CP0BST2MP1	CP0BST2CMP0	CP0BST2NA	CP0BST2A	保留			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CP0BST0CMP1	CP0BST0CMP0	CP0BST0NA	CP0BST0A	CP0BEXEV9	CP0BEXEV8	CP0BEXEV7	CP0BEXEV6	CP0BEXEV5	CP0BEXEV4	CP0BEXEV3	CP0BEXEV2	CP0BEXEV1	CP0BEXEV0	CP0BUP	CP0BSW
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	CP0BST4CMP1	Slave_TIMER4 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP0BST0CMP1 描述。
30	CP0BST4CMP0	Slave_TIMER4 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP0BST0CMP0 描述。
29	CP0BST4NA	ST4CH0_O 输出有效到无效的变化触发捕获 0 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP0BST0NA 描述。
28	CP0BST4A	ST4CH0_O 输出无效到有效的变化触发捕获 0 该位仅在 Slave_TIMER4 的寄存器中保留。

		请参考 CP0BST0A 描述。
27	CP0BST3CMP1	Slave_TIMER3 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP0BST0CMP1 描述。
26	CP0BST3CMP0	Slave_TIMER3 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP0BST0CMP0 描述。
25	CP0BST3NA	ST3CH0_O 输出有效到无效的变化触发捕获 0 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP0BST0NA 描述。
24	CP0BST3A	ST3CH0_O 输出无效到有效的变化触发捕获 0 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP0BST0A 描述。
23	CP0BST2CMP1	Slave_TIMER2 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP0BST0CMP1 描述。
22	CP0BST2CMP0	Slave_TIMER2 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP0BST0CMP0 描述。
21	CP0BST2NA	ST2CH0_O 输出有效到无效的变化触发捕获 0 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP0BST0NA 描述。
20	CP0BST2A	ST2CH0_O 输出无效到有效的变化触发捕获 0 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP0BST0A 描述。
19:16	保留	必须保持复位值
15	CP0BST0CMP1	Slave_TIMER0 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER0 的寄存器中保留。 0：Slave_TIMER0 的比较 1 事件不触发捕获 0。 1：Slave_TIMER0 的比较 1 事件触发捕获 0。
14	CP0BST0CMP0	Slave_TIMER0 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER0 的寄存器中保留。 0：Slave_TIMER0 的比较 0 事件不触发捕获 0。 1：Slave_TIMER0 的比较 0 事件触发捕获 0。

13	CP0BST0NA	ST0CH0_O 输出有效到无效的变化触发捕获 0 当 Slave_TIMER0 通道 0 的输出从有效电平转换为无效电平时，捕获 0 由 ST0CH0_O 触发。 该位仅在 Slave_TIMER0 中保留。 0：ST0CH0_O 输出有效到无效的变化不触发捕获 0。 1：ST0CH0_O 输出有效到无效的变化触发捕获 0。
12	CP0BST0A	ST0CH0_O 输出无效到有效的变化触发捕获 0 当 Slave_TIMER0 通道 0 的输出从无效电平转换为有效电平时，捕获 0 由 ST0CH0_O 触发。 该位仅在 Slave_TIMER0 中保留。 0：ST0CH0_O 输出无效到有效的变化不触发捕获 0。 1：ST0CH0_O 输出无效到有效的变化触发捕获 0。
11	CP0BEXEV9	外部事件 9 触发捕获 0 请参考 CP0BEXEV0 描述。
10	CP0BEXEV8	外部事件 8 触发捕获 0 请参考 CP0BEXEV0 描述。
9	CP0BEXEV7	外部事件 7 触发捕获 0 请参考 CP0BEXEV0 描述。
8	CP0BEXEV6	外部事件 6 触发捕获 0 请参考 CP0BEXEV0 描述。
7	CP0BEXEV5	外部事件 5 触发捕获 0 请参考 CP0BEXEV0 描述。
6	CP0BEXEV4	外部事件 4 触发捕获 0 请参考 CP0BEXEV0 描述。
5	CP0BEXEV3	外部事件 3 触发捕获 0 请参考 CP0BEXEV0 描述。
4	CP0BEXEV2	外部事件 2 触发捕获 0 请参考 CP0BEXEV0 描述。
3	CP0BEXEV1	外部事件 1 触发捕获 0 请参考 CP0BEXEV0 描述。
2	CP0BEXEV0	外部事件 0 触发捕获 0 当该位置 1 时，外部事件 0 触发捕获 0。 0：外部事件 0 不触发捕获 0 1：外部事件 0 触发捕获 0

1	CP0BUP	更新事件触发捕获 0 当该位置 1 时，更新事件触发捕获 0。 0：更新事件不触发捕获 0 1：更新事件触发捕获 0
0	CP0BSW	软件触发捕获 0 该位由软件置 1，硬件自动清零。该位置 1 时，软件触发捕获 0。 0：软件不触发捕获 0 1：软件触发捕获 0

For Slave_TIMER2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CP0BST4CMP1	CP0BST4CMP0	CP0BST4NA	CP0BST4A	CP0BST3CMP1	CP0BST3CMP0	CP0BST3NA	CP0BST3A	保留				CP0BST1CMP1	CP0BST1CMP0	CP0BST1NA	CP0BST1A
rw	rw	rw	rw	rw	rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CP0BST0CMP1	CP0BST0CMP0	CP0BST0NA	CP0BST0A	CP0BEXEV9	CP0BEXEV8	CP0BEXEV7	CP0BEXEV6	CP0BEXEV5	CP0BEXEV4	CP0BEXEV3	CP0BEXEV2	CP0BEXEV1	CP0BEXEV0	CP0BUP	CP0BSW
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	CP0BST4CMP1	Slave_TIMER4 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP0BST0CMP1 描述。
30	CP0BST4CMP0	Slave_TIMER4 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP0BST0CMP0 描述。
29	CP0BST4NA	ST4CH0_O 输出有效到无效的变化触发捕获 0 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP0BST0NA 描述。
28	CP0BST4A	ST4CH0_O 输出无效到有效的变化触发捕获 0 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP0BST0A 描述。
27	CP0BST3CMP1	Slave_TIMER3 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP0BST0CMP1 描述。
26	CP0BST3CMP0	Slave_TIMER3 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER3 的寄存器中保留。

		请参考 CP0BST0CMP0 描述。
25	CP0BST3NA	ST3CH0_O 输出有效到无效的变化触发捕获 0 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP0BST0NA 描述。
24	CP0BST3A	ST3CH0_O 输出无效到有效的变化触发捕获 0 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP0BST0A 描述。
23:20	保留	必须保持复位值
19	CP0BST1CMP1	Slave_TIMER1 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP0BST0CMP1 描述。
18	CP0BST1CMP0	Slave_TIMER1 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP0BST0CMP0 描述。
17	CP0BST1NA	ST1CH0_O 输出有效到无效的变化触发捕获 0 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP0BST0NA 描述。
16	CP0BST1A	ST1CH0_O 输出无效到有效的变化触发捕获 0 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP0BST0A 描述。
15	CP0BST0CMP1	Slave_TIMER0 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER0 的寄存器中保留。 0：Slave_TIMER0 的比较 1 事件不触发捕获 0。 1：Slave_TIMER0 的比较 1 事件触发捕获 0。
14	CP0BST0CMP0	Slave_TIMER0 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER0 的寄存器中保留。 0：Slave_TIMER0 的比较 0 事件不触发捕获 0。 1：Slave_TIMER0 的比较 0 事件触发捕获 0。
13	CP0BST0NA	ST0CH0_O 输出有效到无效的变化触发捕获 0 当 Slave_TIMER0 通道 0 的输出从有效电平转换为无效电平时，捕获 0 由 ST0CH0_O 触发。 该位仅在 Slave_TIMER0 中保留。 0：ST0CH0_O 输出有效到无效的变化不触发捕获 0。 1：ST0CH0_O 输出有效到无效的变化触发捕获 0。
12	CP0BST0A	ST0CH0_O 输出无效到有效的变化触发捕获 0

		当Slave_TIMER0通道0的输出从无效电平转换为有效电平时，捕获0由ST0CH0_O触发。 该位仅在 Slave_TIMER0 中保留。 0：ST0CH0_O 输出无效到有效的变化不触发捕获 0。 1：ST0CH0_O 输出无效到有效的变化触发捕获 0。
11	CP0BEXEV9	外部事件 9 触发捕获 0 请参考 CP0BEXEV0 描述。
10	CP0BEXEV8	外部事件 8 触发捕获 0 请参考 CP0BEXEV0 描述。
9	CP0BEXEV7	外部事件 7 触发捕获 0 请参考 CP0BEXEV0 描述。
8	CP0BEXEV6	外部事件 6 触发捕获 0 请参考 CP0BEXEV0 描述。
7	CP0BEXEV5	外部事件 5 触发捕获 0 请参考 CP0BEXEV0 描述。
6	CP0BEXEV4	外部事件 4 触发捕获 0 请参考 CP0BEXEV0 描述。
5	CP0BEXEV3	外部事件 3 触发捕获 0 请参考 CP0BEXEV0 描述。
4	CP0BEXEV2	外部事件 2 触发捕获 0 请参考 CP0BEXEV0 描述。
3	CP0BEXEV1	外部事件 1 触发捕获 0 请参考 CP0BEXEV0 描述。

For Slave_TIMER3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CP0BST4CMP1	CP0BST4CMP0	CP0BST4NA	CP0BST4A	保留				CP0BST2MP1	CP0BST2CMP0	CP0BST2NA	CP0BST2A	CP0BST1CMP1	CP0BST1CMP0	CP0BST1NA	CP0BST1A
rw	rw	rw	rw					rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CP0BST0CMP1	CP0BST0CMP0	CP0BST0NA	CP0BST0A	CP0BEXEV9	CP0BEXEV8	CP0BEXEV7	CP0BEXEV6	CP0BEXEV5	CP0BEXEV4	CP0BEXEV3	CP0BEXEV2	CP0BEXEV1	CP0BEXEV0	CP0BUP	CP0BSW
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述

31	CP0BST4CMP1	Slave_TIMER4 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP0BST0CMP1 描述。
30	CP0BST4CMP0	Slave_TIMER4 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP0BST0CMP0 描述。
29	CP0BST4NA	ST4CH0_O 输出有效到无效的变化触发捕获 0 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP0BST0NA 描述。
28	CP0BST4A	ST4CH0_O 输出无效到有效的变化触发捕获 0 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP0BST0A 描述。
27:24	保留	必须保持复位值
23	CP0BST2CMP1	Slave_TIMER2 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP0BST0CMP1 描述。
22	CP0BST2CMP0	Slave_TIMER2 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP0BST0CMP0 描述。
21	CP0BST2NA	ST2CH0_O 输出有效到无效的变化触发捕获 0 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP0BST0NA 描述。
20	CP0BST2A	ST2CH0_O 输出无效到有效的变化触发捕获 0 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP0BST0A 描述。
19	CP0BST1CMP1	Slave_TIMER1 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP0BST0CMP1 描述。
18	CP0BST1CMP0	Slave_TIMER1 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP0BST0CMP0 描述。
17	CP0BST1NA	ST1CH0_O 输出有效到无效的变化触发捕获 0 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP0BST0NA 描述。

16	CP0BST1A	ST1CH0_O 输出无效到有效的变化触发捕获 0 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP0BST0A 描述。
15	CP0BST0CMP1	Slave_TIMER0 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER0 的寄存器中保留。 0：Slave_TIMER0 的比较 1 事件不触发捕获 0。 1：Slave_TIMER0 的比较 1 事件触发捕获 0。
14	CP0BST0CMP0	Slave_TIMER0 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER0 的寄存器中保留。 0：Slave_TIMER0 的比较 0 事件不触发捕获 0。 1：Slave_TIMER0 的比较 0 事件触发捕获 0。
13	CP0BST0NA	ST0CH0_O 输出有效到无效的变化触发捕获 0 当 Slave_TIMER0 通道 0 的输出从有效电平转换为无效电平时，捕获 0 由 ST0CH0_O 触发。 该位仅在 Slave_TIMER0 中保留。 0：ST0CH0_O 输出有效到无效的变化不触发捕获 0。 1：ST0CH0_O 输出有效到无效的变化触发捕获 0。
12	CP0BST0A	ST0CH0_O 输出无效到有效的变化触发捕获 0 当 Slave_TIMER0 通道 0 的输出从无效电平转换为有效电平时，捕获 0 由 ST0CH0_O 触发。 该位仅在 Slave_TIMER0 中保留。 0：ST0CH0_O 输出无效到有效的变化不触发捕获 0。 1：ST0CH0_O 输出无效到有效的变化触发捕获 0。
11	CP0BEXEV9	外部事件 9 触发捕获 0 请参考 CP0BEXEV0 描述。
10	CP0BEXEV8	外部事件 8 触发捕获 0 请参考 CP0BEXEV0 描述。
9	CP0BEXEV7	外部事件 7 触发捕获 0 请参考 CP0BEXEV0 描述。
8	CP0BEXEV6	外部事件 6 触发捕获 0 请参考 CP0BEXEV0 描述。
7	CP0BEXEV5	外部事件 5 触发捕获 0 请参考 CP0BEXEV0 描述。
6	CP0BEXEV4	外部事件 4 触发捕获 0 请参考 CP0BEXEV0 描述。

5	CP0BEXEV3	外部事件 3 触发捕获 0 请参考 CP0BEXEV0 描述。
4	CP0BEXEV2	外部事件 2 触发捕获 0 请参考 CP0BEXEV0 描述。
3	CP0BEXEV1	外部事件 1 触发捕获 0 请参考 CP0BEXEV0 描述。
2	CP0BEXEV0	外部事件 0 触发捕获 0 当该位置 1 时，外部事件 0 触发捕获 0。 0：外部事件 0 不触发捕获 0 1：外部事件 0 触发捕获 0
1	CP0BUP	更新事件触发捕获 0 当该位置 1 时，更新事件触发捕获 0。 0：更新事件不触发捕获 0 1：更新事件触发捕获 0
0	CP0BSW	软件触发捕获 0 该位由软件置 1，硬件自动清零。该位置 1 时，软件触发捕获 0。 0：软件不触发捕获 0 1：软件触发捕获 0

For Slave_TIMER4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				CP0BST3CMP1	CP0BST3CMP0	CP0BST3NA	CP0BST3A	CP0BST2MP1	CP0BST2CMP0	CP0BST2NA	CP0BST2A	CP0BST1CMP1	CP0BST1CMP0	CP0BST1NA	CP0BST1A
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CP0BST0CMP1	CP0BST0CMP0	CP0BST0NA	CP0BST0A	CP0BEXEV9	CP0BEXEV8	CP0BEXEV7	CP0BEXEV6	CP0BEXEV5	CP0BEXEV4	CP0BEXEV3	CP0BEXEV2	CP0BEXEV1	CP0BEXEV0	CP0BUP	CP0BSW
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:28	保留	必须保持复位值
27	CP0BST3CMP1	Slave_TIMER3 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP0BST0CMP1 描述。
26	CP0BST3CMP0	Slave_TIMER3 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER3 的寄存器中保留。

		请参考 CP0BST0CMP0 描述。
25	CP0BST3NA	ST3CH0_O 输出有效到无效的变化触发捕获 0 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP0BST0NA 描述。
24	CP0BST3A	ST3CH0_O 输出无效到有效的变化触发捕获 0 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP0BST0A 描述。
23	CP0BST2CMP1	Slave_TIMER2 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP0BST0CMP1 描述。
22	CP0BST2CMP0	Slave_TIMER2 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP0BST0CMP0 描述。
21	CP0BST2NA	ST2CH0_O 输出有效到无效的变化触发捕获 0 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP0BST0NA 描述。
20	CP0BST2A	ST2CH0_O 输出无效到有效的变化触发捕获 0 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP0BST0A 描述。
19	CP0BST1CMP1	Slave_TIMER1 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP0BST0CMP1 描述。
18	CP0BST1CMP0	Slave_TIMER1 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP0BST0CMP0 描述。
17	CP0BST1NA	ST1CH0_O 输出有效到无效的变化触发捕获 0 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP0BST0NA 描述。
16	CP0BST1A	ST1CH0_O 输出无效到有效的变化触发捕获 0 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP0BST0A 描述。
15	CP0BST0CMP1	Slave_TIMER0 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER0 的寄存器中保留。 0：Slave_TIMER0 的比较 1 事件不触发捕获 0。

		1：Slave_TIMER0 的比较 1 事件触发捕获 0。
14	CP0BST0CMP0	Slave_TIMER0 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER0 的寄存器中保留。 0：Slave_TIMER0 的比较 0 事件不触发捕获 0。 1：Slave_TIMER0 的比较 0 事件触发捕获 0。
13	CP0BST0NA	ST0CH0_O 输出有效到无效的变化触发捕获 0 当 Slave_TIMER0 通道 0 的输出从有效电平转换为无效电平时，捕获 0 由 ST0CH0_O 触发。 该位仅在 Slave_TIMER0 中保留。 0：ST0CH0_O 输出有效到无效的变化不触发捕获 0。 1：ST0CH0_O 输出有效到无效的变化触发捕获 0。
12	CP0BST0A	ST0CH0_O 输出无效到有效的变化触发捕获 0 当 Slave_TIMER0 通道 0 的输出从无效电平转换为有效电平时，捕获 0 由 ST0CH0_O 触发。 该位仅在 Slave_TIMER0 中保留。 0：ST0CH0_O 输出无效到有效的变化不触发捕获 0。 1：ST0CH0_O 输出无效到有效的变化触发捕获 0。
11	CP0BEXEV9	外部事件 9 触发捕获 0 请参考 CP0BEXEV0 描述。
10	CP0BEXEV8	外部事件 8 触发捕获 0 请参考 CP0BEXEV0 描述。
9	CP0BEXEV7	外部事件 7 触发捕获 0 请参考 CP0BEXEV0 描述。
8	CP0BEXEV6	外部事件 6 触发捕获 0 请参考 CP0BEXEV0 描述。
7	CP0BEXEV5	外部事件 5 触发捕获 0 请参考 CP0BEXEV0 描述。
6	CP0BEXEV4	外部事件 4 触发捕获 0 请参考 CP0BEXEV0 描述。
5	CP0BEXEV3	外部事件 3 触发捕获 0 请参考 CP0BEXEV0 描述。
4	CP0BEXEV2	外部事件 2 触发捕获 0 请参考 CP0BEXEV0 描述。
3	CP0BEXEV1	外部事件 1 触发捕获 0

请参考 CP0BEXEV0 描述。

2 CP0BEXEV0 外部事件 0 触发捕获 0
当该位置 1 时，外部事件 0 触发捕获 0。
0：外部事件 0 不触发捕获 0
1：外部事件 0 触发捕获 0

1 CP0BUP 更新事件触发捕获 0
当该位置 1 时，更新事件触发捕获 0。
0：更新事件不触发捕获 0
1：更新事件触发捕获 0

0 CP0BSW 软件触发捕获 0
该位由软件置 1，硬件自动清零。该位置 1 时，软件触发捕获 0。
0：软件不触发捕获 0
1：软件触发捕获 0

For Slave_TIMER5 - For Slave_TIMER7

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CP0BST4CMP1	CP0BST4CMP0	CP0BST4NA	CP0BST4A	CP0BST3CMP1	CP0BST3CMP0	CP0BST3NA	CP0BST3A	CP0BST2MP1	CP0BST2CMP0	CP0BST2NA	CP0BST2A	CP0BST1CMP1	CP0BST1CMP0	CP0BST1NA	CP0BST1A
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CP0BST0CMP1	CP0BST0CMP0	CP0BST0NA	CP0BST0A	CP0BEXEV9	CP0BEXEV8	CP0BEXEV7	CP0BEXEV6	CP0BEXEV5	CP0BEXEV4	CP0BEXEV3	CP0BEXEV2	CP0BEXEV1	CP0BEXEV0	CP0BUP	CP0BSW
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	CP0BST4CMP1	Slave_TIMER4 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP0BST0CMP1 描述。
30	CP0BST4CMP0	Slave_TIMER4 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP0BST0CMP0 描述。
29	CP0BST4NA	ST4CH0_O 输出有效到无效的变化触发捕获 0 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP0BST0NA 描述。
28	CP0BST4A	ST4CH0_O 输出无效到有效的变化触发捕获 0 该位仅在 Slave_TIMER4 的寄存器中保留。

		请参考 CP0BST0A 描述。
27	CP0BST3CMP1	Slave_TIMER3 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP0BST0CMP1 描述。
26	CP0BST3CMP0	Slave_TIMER3 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP0BST0CMP0 描述。
25	CP0BST3NA	ST3CH0_O 输出有效到无效的变化触发捕获 0 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP0BST0NA 描述。
24	CP0BST3A	ST3CH0_O 输出无效到有效的变化触发捕获 0 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP0BST0A 描述。
23	CP0BST2CMP1	Slave_TIMER2 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP0BST0CMP1 描述。
22	CP0BST2CMP0	Slave_TIMER2 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP0BST0CMP0 描述。
21	CP0BST2NA	ST2CH0_O 输出有效到无效的变化触发捕获 0 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP0BST0NA 描述。
20	CP0BST2A	ST2CH0_O 输出无效到有效的变化触发捕获 0 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP0BST0A 描述。
19	CP0BST1CMP1	Slave_TIMER1 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP0BST0CMP1 描述。
18	CP0BST1CMP0	Slave_TIMER1 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP0BST0CMP0 描述。
17	CP0BST1NA	ST1CH0_O 输出有效到无效的变化触发捕获 0 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP0BST0NA 描述。

16	CP0BST1A	ST1CH0_O 输出无效到有效的变化触发捕获 0 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP0BST0A 描述。
15	CP0BST0CMP1	Slave_TIMER0 的比较 1 事件触发捕获 0 该位仅在 Slave_TIMER0 的寄存器中保留。 0：Slave_TIMER0 的比较 1 事件不触发捕获 0。 1：Slave_TIMER0 的比较 1 事件触发捕获 0。
14	CP0BST0CMP0	Slave_TIMER0 的比较 0 事件触发捕获 0 该位仅在 Slave_TIMER0 的寄存器中保留。 0：Slave_TIMER0 的比较 0 事件不触发捕获 0。 1：Slave_TIMER0 的比较 0 事件触发捕获 0。
13	CP0BST0NA	ST0CH0_O 输出有效到无效的变化触发捕获 0 当 Slave_TIMER0 通道 0 的输出从有效电平转换为无效电平时，捕获 0 由 ST0CH0_O 触发。 该位仅在 Slave_TIMER0 中保留。 0：ST0CH0_O 输出有效到无效的变化不触发捕获 0。 1：ST0CH0_O 输出有效到无效的变化触发捕获 0。
12	CP0BST0A	ST0CH0_O 输出无效到有效的变化触发捕获 0 当 Slave_TIMER0 通道 0 的输出从无效电平转换为有效电平时，捕获 0 由 ST0CH0_O 触发。 该位仅在 Slave_TIMER0 中保留。 0：ST0CH0_O 输出无效到有效的变化不触发捕获 0。 1：ST0CH0_O 输出无效到有效的变化触发捕获 0。
11	CP0BEXEV9	外部事件 9 触发捕获 0 请参考 CP0BEXEV0 描述。
10	CP0BEXEV8	外部事件 8 触发捕获 0 请参考 CP0BEXEV0 描述。
9	CP0BEXEV7	外部事件 7 触发捕获 0 请参考 CP0BEXEV0 描述。
8	CP0BEXEV6	外部事件 6 触发捕获 0 请参考 CP0BEXEV0 描述。
7	CP0BEXEV5	外部事件 5 触发捕获 0 请参考 CP0BEXEV0 描述。
6	CP0BEXEV4	外部事件 4 触发捕获 0 请参考 CP0BEXEV0 描述。

5	CP0BEXEV3	外部事件 3 触发捕获 0 请参考 CP0BEXEV0 描述。
4	CP0BEXEV2	外部事件 2 触发捕获 0 请参考 CP0BEXEV0 描述。
3	CP0BEXEV1	外部事件 1 触发捕获 0 请参考 CP0BEXEV0 描述。
2	CP0BEXEV0	外部事件 0 触发捕获 0 当该位置 1 时，外部事件 0 触发捕获 0。 0：外部事件 0 不触发捕获 0 1：外部事件 0 触发捕获 0
1	CP0BUP	更新事件触发捕获 0 当该位置 1 时，更新事件触发捕获 0。 0：更新事件不触发捕获 0 1：更新事件触发捕获 0
0	CP0BSW	软件触发捕获 0 该位由软件置 1，硬件自动清零。该位置 1 时，软件触发捕获 0。 0：软件不触发捕获 0 1：软件触发捕获 0

Slave_TIMERx 捕获 1 触发寄存器 (HRTIMER_STxCAP1TRG)

地址偏移：0x60

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

For Slave_TIMER0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CP1BST4CMP1	CP1BST4CMP0	CP1BST4NA	CP1BST4A	CP1BST3CMP1	CP1BST3CMP0	CP1BST3NA	CP1BST3A	CP1BST2MP1	CP1BST2CMP0	CP1BST2NA	CP1BST2A	CP1BST1CMP1	CP1BST1CMP0	CP1BST1NA	CP1BST1A
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
保留				CP1BEXEV9	CP1BEXEV8	CP1BEXEV7	CP1BEXEV6	CP1BEXEV5	CP1BEXEV4	CP1BEXEV3	CP1BEXEV2	CP1BEXEV1	CP1BEXEV0	CP1BUP	CP1BSW
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	CP1BST4CMP1	Slave_TIMER4 的比较 1 事件触发捕获 1

		该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP1BST0CMP1 描述。
30	CP1BST4CMP0	Slave_TIMER4 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP1BST0CMP0 描述。
29	CP1BST4NA	ST4CH0_O 输出有效到无效的变化触发捕获 1 该位仅在 Slave_TIMER4 的寄存器中。 请参考 CP1BST0NA 描述。
28	CP1BST4A	ST4CH0_O 输出无效到有效的变化触发捕获 1 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP1BST0A 描述。
27	CP1BST3CMP1	Slave_TIMER3 的比较 1 事件触发捕获 1 该位仅存在于 Slave_TIMER3 的寄存器中保留。 请参考 CP1BST0CMP1 描述。
26	CP1BST3CMP0	Slave_TIMER3 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP1BST0CMP0 描述。
25	CP1BST3NA	ST3CH0_O 输出有效到无效的变化触发捕获 1 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP1BST0NA 描述。
24	CP1BST3A	ST3CH0_O 输出无效到有效的变化触发捕获 1 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP1BST0A 描述。
23	CP1BST2CMP1	Slave_TIMER2 的比较 1 事件触发捕获 1 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP1BST0CMP1 描述。
22	CP1BST2CMP0	Slave_TIMER2 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP1BST0CMP0 描述。
21	CP1BST2NA	ST2CH0_O 输出有效到无效的变化触发捕获 1 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP1BST0NA 描述。
20	CP1BST2A	ST2CH0_O 输出无效到有效的变化触发捕获 1 该位仅在 Slave_TIMER2 的寄存器中保留。

		请参考 CP1BST0A 描述。
19	CP1BST1CMP1	Slave_TIMER1 的比较 1 事件触发捕获 1 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP1BST0CMP1 描述。
18	CP1BST1CMP0	Slave_TIMER1 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP1BST0CMP0 描述。
17	CP1BST1NA	ST1CH0_O 输出有效到无效的变化触发捕获 1 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP1BST0NA 描述。
16	CP1BST1A	ST1CH0_O 输出无效到有效的变化触发捕获 1 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP1BST0A 描述。
15:12	保留	必须保持复位值
11	CP1BEXEV9	外部事件 9 触发捕获 1 请参考 CP1BEXEV0 描述。
10	CP1BEXEV8	外部事件 8 触发捕获 1 请参考 CP1BEXEV0 描述。
9	CP1BEXEV7	外部事件 7 触发捕获 1 请参考 CP1BEXEV0 描述。
8	CP1BEXEV6	外部事件 6 触发捕获 1 请参考 CP1BEXEV0 描述。
7	CP1BEXEV5	外部事件 5 触发捕获 1 请参考 CP1BEXEV0 描述。
6	CP1BEXEV4	外部事件 4 触发捕获 1 请参考 CP1BEXEV0 描述。
5	CP1BEXEV3	外部事件 3 触发捕获 1 请参考 CP1BEXEV0 描述。
4	CP1BEXEV2	外部事件 2 触发捕获 1 请参考 CP1BEXEV0 描述。
3	CP1BEXEV1	外部事件 1 触发捕获 1 请参考 CP1BEXEV0 描述。
2	CP1BEXEV0	外部事件 0 触发捕获 1

当该位置 1 时，外部事件 0 触发捕获 1。
0：外部事件 0 不触发捕获 1
1：外部事件 0 触发捕获 1

1 CP1BUP 更新事件触发捕获 1
当该位置 1 时，更新事件触发捕获 1。
0：更新事件不触发捕获 1
1：更新事件触发捕获 1

0 CP1BSW 软件触发捕获 1
该位由软件置 1，硬件自动清零。该位置 1 时，软件触发捕获 1。
0：软件不触发捕获 1
1：软件触发捕获 1

For Slave_TIMER1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CP1BST4CMP1	CP1BST4CMP0	CP1BST4NA	CP1BST4A	CP1BST3CMP1	CP1BST3CMP0	CP1BST3NA	CP1BST3A	CP1BST2MP1	CP1BST2CMP0	CP1BST2NA	CP1BST2A	保留			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CP1BST0CMP1	CP1BST0CMP0	CP1BST0NA	CP1BST0A	CP1BEXEV9	CP1BEXEV8	CP1BEXEV7	CP1BEXEV6	CP1BEXEV5	CP1BEXEV4	CP1BEXEV3	CP1BEXEV2	CP1BEXEV1	CP1BEXEV0	CP1BUP	CP1BSW
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	CP1BST4CMP1	Slave_TIMER4 的比较 1 事件触发捕获 1 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP1BST0CMP1 描述。
30	CP1BST4CMP0	Slave_TIMER4 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP1BST0CMP0 描述。
29	CP1BST4NA	ST4CH0_O 输出有效到无效的变化触发捕获 1 该位仅在 Slave_TIMER4 的寄存器中。 请参考 CP1BST0NA 描述。
28	CP1BST4A	ST4CH0_O 输出无效到有效的变化触发捕获 1 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP1BST0A 描述。
27	CP1BST3CMP1	Slave_TIMER3 的比较 1 事件触发捕获 1 该位仅存在于 Slave_TIMER3 的寄存器中保留。

		请参考 CP1BST0CMP1 描述。
26	CP1BST3CMP0	Slave_TIMER3 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP1BST0CMP0 描述。
25	CP1BST3NA	ST3CH0_O 输出有效到无效的变化触发捕获 1 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP1BST0NA 描述。
24	CP1BST3A	ST3CH0_O 输出无效到有效的变化触发捕获 1 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP1BST0A 描述。
23	CP1BST2CMP1	Slave_TIMER2 的比较 1 事件触发捕获 1 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP1BST0CMP1 描述。
22	CP1BST2CMP0	Slave_TIMER2 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP1BST0CMP0 描述。
21	CP1BST2NA	ST2CH0_O 输出有效到无效的变化触发捕获 1 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP1BST0NA 描述。
20	CP1BST2A	ST2CH0_O 输出无效到有效的变化触发捕获 1 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP1BST0A 描述。
19:16	保留	必须保持复位值
15	CP1BST0CMP1	Slave_TIMER0 的比较 1 事件触发捕获 1 该位仅在 Slave_TIMER0 的寄存器中保留。 0：Slave_TIMER0 的比较 1 事件不触发捕获 1。 1：Slave_TIMER0 的比较 1 事件触发捕获 1。
14	CP1BST0CMP0	Slave_TIMER0 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER0 的寄存器中保留。 0：Slave_TIMER0 的比较 0 事件不触发捕获 1。 1：Slave_TIMER0 的比较 0 事件触发捕获 1。
13	CP1BST0NA	ST0CH0_O 输出有效到无效的变化触发捕获 1 当 Slave_TIMER0 通道 0 的输出从有效电平转换为无效电平时，捕获 1 由 ST0CH0_O 触发。 该位仅在 Slave_TIMER0 中保留。

		0：ST0CH0_O 输出有效到无效的变化不触发捕获 1。 1：ST0CH0_O 输出有效到无效的变化触发捕获 1。
12	CP1BST0A	ST0CH0_O 输出无效到有效的变化触发捕获 1 当 Slave_TIMER0 通道 0 的输出从无效电平转换为有效电平时，捕获 1 由 ST0CH0_O 触发。 该位仅在 Slave_TIMER0 中保留。 0：ST0CH0_O 输出无效到有效的变化不触发捕获 1。 1：ST0CH0_O 输出无效到有效的变化触发捕获 1。
11	CP1BEXEV9	外部事件 9 触发捕获 1 请参考 CP1BEXEV0 描述。
10	CP1BEXEV8	外部事件 8 触发捕获 1 请参考 CP1BEXEV0 描述。
9	CP1BEXEV7	外部事件 7 触发捕获 1 请参考 CP1BEXEV0 描述。
8	CP1BEXEV6	外部事件 6 触发捕获 1 请参考 CP1BEXEV0 描述。
7	CP1BEXEV5	外部事件 5 触发捕获 1 请参考 CP1BEXEV0 描述。
6	CP1BEXEV4	外部事件 4 触发捕获 1 请参考 CP1BEXEV0 描述。
5	CP1BEXEV3	外部事件 3 触发捕获 1 请参考 CP1BEXEV0 描述。
4	CP1BEXEV2	外部事件 2 触发捕获 1 请参考 CP1BEXEV0 描述。
3	CP1BEXEV1	外部事件 1 触发捕获 1 请参考 CP1BEXEV0 描述。
2	CP1BEXEV0	外部事件 0 触发捕获 1 当该位置 1 时，外部事件 0 触发捕获 1。 0：外部事件 0 不触发捕获 1 1：外部事件 0 触发捕获 1
1	CP1BUP	更新事件触发捕获 1 当该位置 1 时，更新事件触发捕获 1。 0：更新事件不触发捕获 1 1：更新事件触发捕获 1

0	CP1BSW	软件触发捕获 1 该位由软件置 1，硬件自动清零。该位置 1 时，软件触发捕获 1。 0：软件不触发捕获 1 1：软件触发捕获 1

For Slave_TIMER2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CP1BST4CMP1	CP1BST4CMP0	CP1BST4NA	CP1BST4A	CP1BST3CMP1	CP1BST3CMP0	CP1BST3NA	CP1BST3A	保留				CP1BST1CMP1	CP1BST1CMP0	CP1BST1NA	CP1BST1A
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
CP1BST0CMP1	CP1BST0CMP0	CP1BST0NA	CP1BST0A	CP1BEXEV9	CP1BEXEV8	CP1BEXEV7	CP1BEXEV6	CP1BEXEV5	CP1BEXEV4	CP1BEXEV3	CP1BEXEV2	CP1BEXEV1	CP1BEXEV0	CP1BUP	CP1BSW
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	CP1BST4CMP1	Slave_TIMER4 的比较 1 事件触发捕获 1 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP1BST0CMP1 描述。
30	CP1BST4CMP0	Slave_TIMER4 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP1BST0CMP0 描述。
29	CP1BST4NA	ST4CH0_O 输出有效到无效的变化触发捕获 1 该位仅在 Slave_TIMER4 的寄存器中。 请参考 CP1BST0NA 描述。
28	CP1BST4A	ST4CH0_O 输出无效到有效的变化触发捕获 1 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP1BST0A 描述。
27	CP1BST3CMP1	Slave_TIMER3 的比较 1 事件触发捕获 1 该位仅存在于 Slave_TIMER3 的寄存器中保留。 请参考 CP1BST0CMP1 描述。
26	CP1BST3CMP0	Slave_TIMER3 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP1BST0CMP0 描述。
25	CP1BST3NA	ST3CH0_O 输出有效到无效的变化触发捕获 1 该位仅在 Slave_TIMER3 的寄存器中保留。

		请参考 CP1BST0NA 描述。
24	CP1BST3A	ST3CH0_O 输出无效到有效的变化触发捕获 1 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP1BST0A 描述。
23:20	保留	必须保持复位值
19	CP1BST1CMP1	Slave_TIMER1 的比较 1 事件触发捕获 1 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP1BST0CMP1 描述。
18	CP1BST1CMP0	Slave_TIMER1 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP1BST0CMP0 描述。
17	CP1BST1NA	ST1CH0_O 输出有效到无效的变化触发捕获 1 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP1BST0NA 描述。
16	CP1BST1A	ST1CH0_O 输出无效到有效的变化触发捕获 1 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP1BST0A 描述。
15	CP1BST0CMP1	Slave_TIMER0 的比较 1 事件触发捕获 1 该位仅在 Slave_TIMER0 的寄存器中保留。 0：Slave_TIMER0 的比较 1 事件不触发捕获 1。 1：Slave_TIMER0 的比较 1 事件触发捕获 1。
14	CP1BST0CMP0	Slave_TIMER0 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER0 的寄存器中保留。 0：Slave_TIMER0 的比较 0 事件不触发捕获 1。 1：Slave_TIMER0 的比较 0 事件触发捕获 1。
13	CP1BST0NA	ST0CH0_O 输出有效到无效的变化触发捕获 1 当 Slave_TIMER0 通道 0 的输出从有效电平转换为无效电平时，捕获 1 由 ST0CH0_O 触发。 该位仅在 Slave_TIMER0 中保留。 0：ST0CH0_O 输出有效到无效的变化不触发捕获 1。 1：ST0CH0_O 输出有效到无效的变化触发捕获 1。
12	CP1BST0A	ST0CH0_O 输出无效到有效的变化触发捕获 1 当 Slave_TIMER0 通道 0 的输出从无效电平转换为有效电平时，捕获 1 由 ST0CH0_O 触发。 该位仅在 Slave_TIMER0 中保留。

0：ST0CH0_O 输出无效到有效的变化不触发捕获 1。
1：ST0CH0_O 输出无效到有效的变化触发捕获 1。

11	CP1BEXEV9	外部事件 9 触发捕获 1 请参考 CP1BEXEV0 描述。
10	CP1BEXEV8	外部事件 8 触发捕获 1 请参考 CP1BEXEV0 描述。
9	CP1BEXEV7	外部事件 7 触发捕获 1 请参考 CP1BEXEV0 描述。
8	CP1BEXEV6	外部事件 6 触发捕获 1 请参考 CP1BEXEV0 描述。
7	CP1BEXEV5	外部事件 5 触发捕获 1 请参考 CP1BEXEV0 描述。
6	CP1BEXEV4	外部事件 4 触发捕获 1 请参考 CP1BEXEV0 描述。
5	CP1BEXEV3	外部事件 3 触发捕获 1 请参考 CP1BEXEV0 描述。
4	CP1BEXEV2	外部事件 2 触发捕获 1 请参考 CP1BEXEV0 描述。
3	CP1BEXEV1	外部事件 1 触发捕获 1 请参考 CP1BEXEV0 描述。
2	CP1BEXEV0	外部事件 0 触发捕获 1 当该位置 1 时，外部事件 0 触发捕获 1。 0：外部事件 0 不触发捕获 1 1：外部事件 0 触发捕获 1
1	CP1BUP	更新事件触发捕获 1 当该位置 1 时，更新事件触发捕获 1。 0：更新事件不触发捕获 1 1：更新事件触发捕获 1
0	CP1BSW	软件触发捕获 1 该位由软件置 1，硬件自动清零。该位置 1 时，软件触发捕获 1。 0：软件不触发捕获 1 1：软件触发捕获 1

For Slave_TIMER3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CP1BST4CMP1	CP1BST4CMP0	CP1BST4NA	CP1BST4A	保留				CP1BST2MP1	CP1BST2CMP0	CP1BST2NA	CP1BST2A	CP1BST1CMP1	CP1BST1CMP0	CP1BST1NA	CP1BST1A
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
CP1BST0CMP1	CP1BST0CMP0	CP1BST0NA	CP1BST0A	CP1BEXEV9	CP1BEXEV8	CP1BEXEV7	CP1BEXEV6	CP1BEXEV5	CP1BEXEV4	CP1BEXEV3	CP1BEXEV2	CP1BEXEV1	CP1BEXEV0	CP1BUP	CP1BSW
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	CP1BST4CMP1	Slave_TIMER4 的比较 1 事件触发捕获 1 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP1BST0CMP1 描述。
30	CP1BST4CMP0	Slave_TIMER4 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP1BST0CMP0 描述。
29	CP1BST4NA	ST4CH0_O 输出有效到无效的变化触发捕获 1 该位仅在 Slave_TIMER4 的寄存器中。 请参考 CP1BST0NA 描述。
28	CP1BST4A	ST4CH0_O 输出无效到有效的变化触发捕获 1 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP1BST0A 描述。
27:24	保留	必须保持复位值
23	CP1BST2CMP1	Slave_TIMER2 的比较 1 事件触发捕获 1 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP1BST0CMP1 描述。
22	CP1BST2CMP0	Slave_TIMER2 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP1BST0CMP0 描述。
21	CP1BST2NA	ST2CH0_O 输出有效到无效的变化触发捕获 1 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP1BST0NA 描述。
20	CP1BST2A	ST2CH0_O 输出无效到有效的变化触发捕获 1 该位仅在 Slave_TIMER2 的寄存器中保留。

请参考 CP1BST0A 描述。

19	CP1BST1CMP1	Slave_TIMER1 的比较 1 事件触发捕获 1 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP1BST0CMP1 描述。
18	CP1BST1CMP0	Slave_TIMER1 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP1BST0CMP0 描述。
17	CP1BST1NA	ST1CH0_O 输出有效到无效的变化触发捕获 1 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP1BST0NA 描述。
16	CP1BST1A	ST1CH0_O 输出无效到有效的变化触发捕获 1 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP1BST0A 描述。
15	CP1BST0CMP1	Slave_TIMER0 的比较 1 事件触发捕获 1 该位仅在 Slave_TIMER0 的寄存器中保留。 0：Slave_TIMER0 的比较 1 事件不触发捕获 1。 1：Slave_TIMER0 的比较 1 事件触发捕获 1。
14	CP1BST0CMP0	Slave_TIMER0 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER0 的寄存器中保留。 0：Slave_TIMER0 的比较 0 事件不触发捕获 1。 1：Slave_TIMER0 的比较 0 事件触发捕获 1。
13	CP1BST0NA	ST0CH0_O 输出有效到无效的变化触发捕获 1 当 Slave_TIMER0 通道 0 的输出从有效电平转换为无效电平时，捕获 1 由 ST0CH0_O 触发。 该位仅在 Slave_TIMER0 中保留。 0：ST0CH0_O 输出有效到无效的变化不触发捕获 1。 1：ST0CH0_O 输出有效到无效的变化触发捕获 1。
12	CP1BST0A	ST0CH0_O 输出无效到有效的变化触发捕获 1 当 Slave_TIMER0 通道 0 的输出从无效电平转换为有效电平时，捕获 1 由 ST0CH0_O 触发。 该位仅在 Slave_TIMER0 中保留。 0：ST0CH0_O 输出无效到有效的变化不触发捕获 1。 1：ST0CH0_O 输出无效到有效的变化触发捕获 1。
11	CP1BEXEV9	外部事件 9 触发捕获 1 请参考 CP1BEXEV0 描述。

10	CP1BEXEV8	外部事件 8 触发捕获 1 请参考 CP1BEXEV0 描述。
9	CP1BEXEV7	外部事件 7 触发捕获 1 请参考 CP1BEXEV0 描述。
8	CP1BEXEV6	外部事件 6 触发捕获 1 请参考 CP1BEXEV0 描述。
7	CP1BEXEV5	外部事件 5 触发捕获 1 请参考 CP1BEXEV0 描述。
6	CP1BEXEV4	外部事件 4 触发捕获 1 请参考 CP1BEXEV0 描述。
5	CP1BEXEV3	外部事件 3 触发捕获 1 请参考 CP1BEXEV0 描述。
4	CP1BEXEV2	外部事件 2 触发捕获 1 请参考 CP1BEXEV0 描述。
3	CP1BEXEV1	外部事件 1 触发捕获 1 请参考 CP1BEXEV0 描述。
2	CP1BEXEV0	外部事件 0 触发捕获 1 当该位置 1 时，外部事件 0 触发捕获 1。 0：外部事件 0 不触发捕获 1 1：外部事件 0 触发捕获 1
1	CP1BUP	更新事件触发捕获 1 当该位置 1 时，更新事件触发捕获 1。 0：更新事件不触发捕获 1 1：更新事件触发捕获 1
0	CP1BSW	软件触发捕获 1 该位由软件置 1，硬件自动清零。该位置 1 时，软件触发捕获 1。 0：软件不触发捕获 1 1：软件触发捕获 1

For Slave_TIMER4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				CP1BST3CMP1	CP1BST3CMP0	CP1BST3NA	CP1BST3A	CP1BST2MP1	CP1BST2CMP0	CP1BST2NA	CP1BST2A	CP1BST1CMP1	CP1BST1CMP0	CP1BST1NA	CP1BST1A
				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

CP1BST0CMP1	CP1BST0CMP0	CP1BST0NA	CP1BST0A	CP1BEXEV9	CP1BEXEV8	CP1BEXEV7	CP1BEXEV6	CP1BEXEV5	CP1BEXEV4	CP1BEXEV3	CP1BEXEV2	CP1BEXEV1	CP1BEXEV0	CP1BUP	CP1BSW
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:28	保留	必须保持复位值
27	CP1BST3CMP1	Slave_TIMER3 的比较 1 事件触发捕获 1 该位仅存在于 Slave_TIMER3 的寄存器中保留。 请参考 CP1BST0CMP1 描述。
26	CP1BST3CMP0	Slave_TIMER3 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP1BST0CMP0 描述。
25	CP1BST3NA	ST3CH0_O 输出有效到无效的变化触发捕获 1 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP1BST0NA 描述。
24	CP1BST3A	ST3CH0_O 输出无效到有效的变化触发捕获 1 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP1BST0A 描述。
23	CP1BST2CMP1	Slave_TIMER2 的比较 1 事件触发捕获 1 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP1BST0CMP1 描述。
22	CP1BST2CMP0	Slave_TIMER2 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP1BST0CMP0 描述。
21	CP1BST2NA	ST2CH0_O 输出有效到无效的变化触发捕获 1 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP1BST0NA 描述。
20	CP1BST2A	ST2CH0_O 输出无效到有效的变化触发捕获 1 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP1BST0A 描述。
19	CP1BST1CMP1	Slave_TIMER1 的比较 1 事件触发捕获 1 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP1BST0CMP1 描述。
18	CP1BST1CMP0	Slave_TIMER1 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER1 的寄存器中保留。

		请参考 CP1BST0CMP0 描述。
17	CP1BST1NA	ST1CH0_O 输出有效到无效的变化触发捕获 1 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP1BST0NA 描述。
16	CP1BST1A	ST1CH0_O 输出无效到有效的变化触发捕获 1 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP1BST0A 描述。
15	CP1BST0CMP1	Slave_TIMER0 的比较 1 事件触发捕获 1 该位仅在 Slave_TIMER0 的寄存器中保留。 0：Slave_TIMER0 的比较 1 事件不触发捕获 1。 1：Slave_TIMER0 的比较 1 事件触发捕获 1。
14	CP1BST0CMP0	Slave_TIMER0 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER0 的寄存器中保留。 0：Slave_TIMER0 的比较 0 事件不触发捕获 1。 1：Slave_TIMER0 的比较 0 事件触发捕获 1。
13	CP1BST0NA	ST0CH0_O 输出有效到无效的变化触发捕获 1 当 Slave_TIMER0 通道 0 的输出从有效电平转换为无效电平时，捕获 1 由 ST0CH0_O 触发。 该位仅在 Slave_TIMER0 中保留。 0：ST0CH0_O 输出有效到无效的变化不触发捕获 1。 1：ST0CH0_O 输出有效到无效的变化触发捕获 1。
12	CP1BST0A	ST0CH0_O 输出无效到有效的变化触发捕获 1 当 Slave_TIMER0 通道 0 的输出从无效电平转换为有效电平时，捕获 1 由 ST0CH0_O 触发。 该位仅在 Slave_TIMER0 中保留。 0：ST0CH0_O 输出无效到有效的变化不触发捕获 1。 1：ST0CH0_O 输出无效到有效的变化触发捕获 1。
11	CP1BEXEV9	外部事件 9 触发捕获 1 请参考 CP1BEXEV0 描述。
10	CP1BEXEV8	外部事件 8 触发捕获 1 请参考 CP1BEXEV0 描述。
9	CP1BEXEV7	外部事件 7 触发捕获 1 请参考 CP1BEXEV0 描述。
8	CP1BEXEV6	外部事件 6 触发捕获 1 请参考 CP1BEXEV0 描述。

7	CP1BEXEV5	外部事件 5 触发捕获 1 请参考 CP1BEXEV0 描述。
6	CP1BEXEV4	外部事件 4 触发捕获 1 请参考 CP1BEXEV0 描述。
5	CP1BEXEV3	外部事件 3 触发捕获 1 请参考 CP1BEXEV0 描述。
4	CP1BEXEV2	外部事件 2 触发捕获 1 请参考 CP1BEXEV0 描述。
3	CP1BEXEV1	外部事件 1 触发捕获 1 请参考 CP1BEXEV0 描述。
2	CP1BEXEV0	外部事件 0 触发捕获 1 当该位置 1 时，外部事件 0 触发捕获 1。 0：外部事件 0 不触发捕获 1 1：外部事件 0 触发捕获 1
1	CP1BUP	更新事件触发捕获 1 当该位置 1 时，更新事件触发捕获 1。 0：更新事件不触发捕获 1 1：更新事件触发捕获 1
0	CP1BSW	软件触发捕获 1 该位由软件置 1，硬件自动清零。该位置 1 时，软件触发捕获 1。 0：软件不触发捕获 1 1：软件触发捕获 1

For Slave_TIMER5 - For Slave_TIMER7

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CP1BST4CMP1	CP1BST4CMP0	CP1BST4NA	CP1BST4A	CP1BST3CMP1	CP1BST3CMP0	CP1BST3NA	CP1BST3A	CP1BST2MP1	CP1BST2CMP0	CP1BST2NA	CP1BST2A	CP1BST1CMP1	CP1BST1CMP0	CP1BST1NA	CP1BST1A
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
CP1BST0CMP1	CP1BST0CMP0	CP1BST0NA	CP1BST0A	CP1BEXEV9	CP1BEXEV8	CP1BEXEV7	CP1BEXEV6	CP1BEXEV5	CP1BEXEV4	CP1BEXEV3	CP1BEXEV2	CP1BEXEV1	CP1BEXEV0	CP1BUP	CP1BSW
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	CP1BST4CMP1	Slave_TIMER4 的比较 1 事件触发捕获 1

		该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP1BST0CMP1 描述。
30	CP1BST4CMP0	Slave_TIMER4 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP1BST0CMP0 描述。
29	CP1BST4NA	ST4CH0_O 输出有效到无效的变化触发捕获 1 该位仅在 Slave_TIMER4 的寄存器中。 请参考 CP1BST0NA 描述。
28	CP1BST4A	ST4CH0_O 输出无效到有效的变化触发捕获 1 该位仅在 Slave_TIMER4 的寄存器中保留。 请参考 CP1BST0A 描述。
27	CP1BST3CMP1	Slave_TIMER3 的比较 1 事件触发捕获 1 该位仅存在于 Slave_TIMER3 的寄存器中保留。 请参考 CP1BST0CMP1 描述。
26	CP1BST3CMP0	Slave_TIMER3 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP1BST0CMP0 描述。
25	CP1BST3NA	ST3CH0_O 输出有效到无效的变化触发捕获 1 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP1BST0NA 描述。
24	CP1BST3A	ST3CH0_O 输出无效到有效的变化触发捕获 1 该位仅在 Slave_TIMER3 的寄存器中保留。 请参考 CP1BST0A 描述。
23	CP1BST2CMP1	Slave_TIMER2 的比较 1 事件触发捕获 1 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP1BST0CMP1 描述。
22	CP1BST2CMP0	Slave_TIMER2 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP1BST0CMP0 描述。
21	CP1BST2NA	ST2CH0_O 输出有效到无效的变化触发捕获 1 该位仅在 Slave_TIMER2 的寄存器中保留。 请参考 CP1BST0NA 描述。
20	CP1BST2A	ST2CH0_O 输出无效到有效的变化触发捕获 1 该位仅在 Slave_TIMER2 的寄存器中保留。

请参考 CP1BST0A 描述。

19	CP1BST1CMP1	Slave_TIMER1 的比较 1 事件触发捕获 1 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP1BST0CMP1 描述。
18	CP1BST1CMP0	Slave_TIMER1 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP1BST0CMP0 描述。
17	CP1BST1NA	ST1CH0_O 输出有效到无效的变化触发捕获 1 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP1BST0NA 描述。
16	CP1BST1A	ST1CH0_O 输出无效到有效的变化触发捕获 1 该位仅在 Slave_TIMER1 的寄存器中保留。 请参考 CP1BST0A 描述。
15	CP1BST0CMP1	Slave_TIMER0 的比较 1 事件触发捕获 1 该位仅在 Slave_TIMER0 的寄存器中保留。 0：Slave_TIMER0 的比较 1 事件不触发捕获 1。 1：Slave_TIMER0 的比较 1 事件触发捕获 1。
14	CP1BST0CMP0	Slave_TIMER0 的比较 0 事件触发捕获 1 该位仅在 Slave_TIMER0 的寄存器中保留。 0：Slave_TIMER0 的比较 0 事件不触发捕获 1。 1：Slave_TIMER0 的比较 0 事件触发捕获 1。
13	CP1BST0NA	ST0CH0_O 输出有效到无效的变化触发捕获 1 当 Slave_TIMER0 通道 0 的输出从有效电平转换为无效电平时，捕获 1 由 ST0CH0_O 触发。 该位仅在 Slave_TIMER0 中保留。 0：ST0CH0_O 输出有效到无效的变化不触发捕获 1。 1：ST0CH0_O 输出有效到无效的变化触发捕获 1。
12	CP1BST0A	ST0CH0_O 输出无效到有效的变化触发捕获 1 当 Slave_TIMER0 通道 0 的输出从无效电平转换为有效电平时，捕获 1 由 ST0CH0_O 触发。 该位仅在 Slave_TIMER0 中保留。 0：ST0CH0_O 输出无效到有效的变化不触发捕获 1。 1：ST0CH0_O 输出无效到有效的变化触发捕获 1。
11	CP1BEXEV9	外部事件 9 触发捕获 1 请参考 CP1BEXEV0 描述。

10	CP1BEXEV8	外部事件 8 触发捕获 1 请参考 CP1BEXEV0 描述。
9	CP1BEXEV7	外部事件 7 触发捕获 1 请参考 CP1BEXEV0 描述。
8	CP1BEXEV6	外部事件 6 触发捕获 1 请参考 CP1BEXEV0 描述。
7	CP1BEXEV5	外部事件 5 触发捕获 1 请参考 CP1BEXEV0 描述。
6	CP1BEXEV4	外部事件 4 触发捕获 1 请参考 CP1BEXEV0 描述。
5	CP1BEXEV3	外部事件 3 触发捕获 1 请参考 CP1BEXEV0 描述。
4	CP1BEXEV2	外部事件 2 触发捕获 1 请参考 CP1BEXEV0 描述。
3	CP1BEXEV1	外部事件 1 触发捕获 1 请参考 CP1BEXEV0 描述。
2	CP1BEXEV0	外部事件 0 触发捕获 1 当该位置 1 时，外部事件 0 触发捕获 1。 0：外部事件 0 不触发捕获 1 1：外部事件 0 触发捕获 1
1	CP1BUP	更新事件触发捕获 1 当该位置 1 时，更新事件触发捕获 1。 0：更新事件不触发捕获 1 1：更新事件触发捕获 1
0	CP1BSW	软件触发捕获 1 该位由软件置 1，硬件自动清零。该位置 1 时，软件触发捕获 1。 0：软件不触发捕获 1 1：软件触发捕获 1

Slave_TIMERx 通道输出控制寄存器 (HRTIMER_STxCHOCTL)

地址偏移：0x64

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

保留								BMCH1DTI	CH1CSEN	CH1FLTOS[1:0]		ISO1	BMCH1IEN	CH1P	保留
								rw	rw	rw		rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	BALIAR	保留	DLYISCH[2:0]			DLYISMEN	DTEN	BMCH0DTI	CH0CSEN	CH0FLTOS[1:0]		ISO0	BMCH0IEN	CH0P	保留
	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	BMCH1DTI	突发模式中通道 1 的死区时间 在突发模式下，可以在输出进入空闲状态之前插入死区时间。 0：输出立即为空闲状态。 1：输出在进入空闲状态之前插入死区时间。 **注意：** （1）一旦计数器使能（STxCEN 位置 1），就不得修改该位。 （2）仅在突发模式的空闲模式下，输出空闲状态之一为有效状态（ISOy = 1，y = 0,1），且死区时间值为正（DTFSPROT 位 / DTRSPROT 位为 0）时，才可以设置此位）。
22	CH1CSEN	通道 1 载波信号模式使能 0：通道 1 载波信号模式禁能。 1：通道 1 载波信号模式使能。 **注意：**一旦计数器使能（STxCEN 位置 1），就不得修改该位。
21:20	CH1FLTOS[1:0]	通道 1 故障输出状态 该位域配置了故障事件发生时，通道 1 的输出状态。 00：没有影响。发生故障事件时，输出处于运行模式。 01：输出为有效电平。 10：输出无效电平。 11：输出为高阻状态。 **注意：**如果 HRTIMER_STxFLTCTL 寄存器中的 FLTyEN（y = 0..4）位置 1，或输出处于故障状态，,一旦计数器使能（将 STxCEN 位置 1），就不能修改该位域。
19	ISO1	通道 1 输出空闲状态 0：通道 1 输出空闲状态为无效电平。 1：通道 1 输出空闲状态为有效电平。 **注意：**必须在 HRTIMER 控制输出前，配置该位。
18	BMCH1IEN	在突发模式中使能通道 1 空闲状态 该位用于配置在突发模式下，通道 1 输出空闲状态。 0：通道 1 输出不受突发模式影响。

		1：在突发模式下，通道 1 的输出可以为空闲状态。 **注意：**该位已预加载，可以在运行时更改，但在突发模式下不得更改。
17	CH1P	通道 1 输出极性 该位确定通道 1 输出信号的极性。 0：通道 1 高电平有效。 1：通道 1 低电平有效。 **注意：**一旦计数器使能（STxCEN 位置 1），就不得修改该位。
16:15	保留	必须保持复位值
14	BALIAR	均衡空闲自动恢复 此位确定在均衡空闲事件后是否自动重新激活输出。此位仅在 DLYISCH [2:0] 被设置为 011 或 111 时具有重要意义；否则将被忽略。 0：禁能 1：使能 **注意：**一旦计数器使能（STxCEN 位置 1），就不得修改该位。
13	保留	必须保持复位值
12:10	DLYISCH[2:0]	延迟空闲的源和通道 该位域配置了延迟空闲模式（DLYISMEN = 1）使能时的源和通道。 在 HRTIMER_STyCHOCTL（y = 0, 1, 2）寄存器中： 000：外部事件 5 到来时，通道 0 输出延迟的空闲状态。 001：外部事件 5 到来时，通道 1 输出延迟的空闲状态。 010：外部事件 5 到来时，通道 0 和通道 1 输出延迟的空闲状态。 011：在均衡模式下，外部事件 5 到来时，通道 0 和通道 1 输出均衡空闲状态（HRTIMER_STyCTL0（y = 0, 1, 2）寄存器中的 BLNMEN = 1）。 100：外部事件 6 到来时，通道 0 输出延迟的空闲状态。 101：外部事件 6 到来时，通道 1 输出延迟的空闲状态。 110：外部事件 6 到来时，通道 0 和通道 1 输出延迟的空闲状态。 111：在均衡模式下，外部事件 6 到来时，通道 0 和通道 1 输出均衡空闲状态（HRTIMER_STyCTL0（y = 0, 1, 2）寄存器中的 BLNMEN = 1）。 在 HRTIMER_STyCHOCTL（y = 3..7）寄存器中： 000：外部事件 7 到来时，通道 0 输出延迟的空闲状态。 001：外部事件 7 到来时，通道 1 输出延迟的空闲状态。 010：外部事件 7 到来时，通道 0 和通道 1 输出延迟的空闲状态。 011：在均衡模式下，外部事件 7 到来时，通道 0 和通道 1 输出均衡空闲状态（HRTIMER_STyCTL0（y = 3..7）寄存器中的 BLNMEN = 1）。 100：外部事件 8 到来时，通道 0 输出延迟的空闲状态。 101：外部事件 8 到来时，通道 1 输出延迟的空闲状态。 110：外部事件 8 到来时，通道 0 和通道 1 输出延迟的空闲状态。 111：在均衡模式下，外部事件 8 到来时，通道 0 和通道 1 输出均衡空闲状态

		（HRTIMER_STyCTL0（y = 3...7）寄存器中的 BLNMEN = 1）。 **注意：**一旦延迟空闲模式（DLYISMEN 位置 1）使能，就不得修改此位域。
9	DLYISMEN	延迟空闲模式使能 0：延迟空闲模式禁能 1：延迟空闲模式使能 **注意：**一旦计数器使能（STxCEN 位置 1），就不得修改该位
8	DTEN	死区时间使能 0：通道 0 和通道 1 的输出是独立的。 1：通道 0 和通道 1 的输出是互补的，在通道 0 和通道 1 的输出之间插入死区时间。 **注意：**一旦计数器使能（STxCEN 位置 1），或其输出被其他定时器使能和控制，就不得修改该位。
7	BMCH0DTI	突发模式中通道 0 的死区时间 在突发模式下，可以在输出进入空闲状态之前插入死区时间。 0：输出立即为空闲状态。 1：输出在进入空闲状态之前插入死区时间。 **注意：** （1）一旦计数器使能（STxCEN 位置 1），就不得修改该位。 （2）仅在突发模式的空闲模式下，输出空闲状态之一为有效状态（ISOy = 1，y = 0,1），且死区时间值为正（DTFSPROT 位 / DTRSPROT 位为 0）时，才可以设置此位）。
6	CH0CSEN	通道 0 载波信号模式使能 0：通道 0 载波信号模式禁能。 1：通道 0 载波信号模式使能。 **注意：**一旦计数器使能（STxCEN 位置 1），就不得修改该位。
5:4	CH0FLTOS[1:0]	通道 0 故障输出状态 该位域配置了在故障事件发生时，通道 0 的输出状态。 00：没有影响。发生故障事件时，输出处于运行模式。 01：输出为有效电平。 10：输出无效电平。 11：输出为高阻状态。 **注意：**如果 HRTIMER_STxFLTCTL 寄存器中的 FLTyEN（y = 0..4）位置 1，或输出处于故障状态，一旦计数器使能（将 STxCEN 位置 1），就不能修改该位域。
3	ISO0	通道 0 输出空闲状态 0：通道 0 输出空闲状态为无效电平。 1：通道 0 输出空闲状态为有效电平。 注意：必须在 HRTIMER 控制输出前，配置该位。
2	BMCH0IEN	在突发模式中使能通道 0 空闲状态

		该位用于配置在突发模式下，通道 0 输出空闲状态 0：通道 0 输出不受突发模式影响。 1：在突发模式下，通道 0 的输出可以为空闲状态。 **注意：**该位已预加载，可以在运行时更改，但在突发模式下不得更改。
1	CH0P	通道 0 输出极性 该位确定通道 0 输出信号的极性。 0：通道 0 高电平有效 1：通道 0 低电平有效 **注意：**一旦计数器使能（STxCEN 位置 1），就不得修改该位。
0	保留	必须保持复位值

Slave_TIMERx 故障控制寄存器 (HRTIMER_STxFLTCTL)

地址偏移：0x68

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLTENPROT								保留							
rwo															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								FLT7EN	FLT6EN	FLT5EN	FLT4EN	FLT3EN	FLT2EN	FLT1EN	FLT0EN
								rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	FLTENPROT	故障保护使能 该位用于使能写保护，该位只能写一次，当它置 1 时，只能通过系统复位清零。 0：故障保护禁能。FLTyEN (y=0..7)可以写操作。 1：故障保护使能。FLTyEN (y=0..7)为只读。
30:8	保留	必须保持复位值
7	FLT7EN	故障 7 使能 0：故障 7 禁能 1：故障 7 使能
6	FLT6EN	故障 6 使能 0：故障 6 禁能 1：故障 6 使能
5	FLT5EN	故障 5 使能

		0：故障 5 禁能 1：故障 5 使能
4	FLT4EN	故障 4 使能 0：故障 4 禁能 1：故障 4 使能
3	FLT3EN	故障 3 使能 0：故障 3 禁能 1：故障 3 使能
2	FLT2EN	故障 2 使能 0：故障 2 禁能 1：故障 2 使能
1	FLT1EN	故障 1 使能 0：故障 1 禁能 1：故障 1 使能
0	FLT0EN	故障 0 使能 0：故障 0 禁能 1：故障 0 使能

Slave_TIMERx 控制寄存器 1 (HRTIMER_STxCTL1)

地址偏移：0x6C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留											TRGHALFM	保留		IMUPD2V	IMUPD0V
											rw			rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FEROVM[1:0]		BMROVM[1:0]		ADCROVM[1:0]		OUTROVM[1:0]		ROVM[1:0]		保留	CAM	保留	TRIG1M	TRIG0M	TRIGEN
rw		rw		rw		rw		rw			rw		rw	rw	rw

位/位域	名称	描述
31:21	保留	必须保持复位值
20	TRGHALFM	可变频率半波模式 此位域确定比较 1 寄存器的工作模式，是采用标准模式（只要计数器等于比较值，就立即发生比较匹配），还是采用可变频率半波模式（参见：*可变频率半波模式*）。 0: 仅由用户写入 CMP1 寄存器（标准比较模式）

		1: 当发生捕获 0 事件时，硬件会立即设置 CMP1 的值。它加载为（捕获 0 除以 2）的值。初始值可以由用户写入（只要 TRGHALFM 被复位），但一旦第一个捕获被触发，该值将被忽略（当 TRGHALFM 位被设置时，对于 CMP1，预加载机制将被禁用）。
19:18	保留	必须保持复位值
17	IMUPD2V	立即更新比较 2 值 PWM 模式： 此位确定比较 2 的操作模式。 0：当计数器等于比较值时，发生比较 2 事件。 1：当计数器超过比较值时，发生比较 2 事件。如果比较值在运行时更改，新值将与当前计数器值进行比较，可能导致生成输出的设置或复位。
16	IMUPD0V	立即更新比较 0 值 PWM 模式： 此位确定比较 0 的操作模式。 0：当计数器等于比较值时，发生比较 0 事件。 1：当计数器超过比较值时，发生比较 0 事件。如果比较值在运行时更改，新值将与当前计数器值进行比较，可能导致生成输出的设置或复位。
15:14	FEROVM[1:0]	故障和事件翻转模式 此位定义了在上下计数模式下何时生成翻转事件，仅涉及故障和事件计数器使用的翻转事件。 00: 当计数器等于 0 或等于 HRTIMER_STxCAR 值时生成事件 01: 当计数器等于 0 时生成事件 10: 当计数器等于 HRTIMER_STxCAR 时生成事件 11: 保留 **注意：**此设置仅在 CAM 位被设置时适用。否则，它不具有重要意义。 注意：一旦定时器正在运行（STxCEN 位被设置），此位域将无法更改。
13:12	BMROVM[1:0]	突发模式下的翻转模式 此位定义了在上下计数模式下何时生成翻转事件，在突发模式下生效。 00: 当计数器等于 0 或等于 HRTIMER_STxCAR 值时生成事件 01: 当计数器等于 0 时生成事件 10: 当计数器等于 HRTIMER_STxCAR 时生成事件 11: 保留 **注意：**此设置仅在 CAM 位被设置时适用。否则，它不具有重要意义。 注意：一旦定时器正在运行（STxCEN 位被设置），此位域将无法更改。
11:10	ADCROVM[1:0]	上下计数模式下触发 ADC 此位定义了在上下计数模式下何时生成 ADC 触发事件。 00: 计数器在向上计数和向下计数方向时生成 ADC 触发事件 01: 计数器在向下计数方向时生成 ADC 触发事件 10: 计数器在向上计数方向时生成 ADC 触发事件 11: 保留 **注意：**此设置仅在 CAM 位被设置时适用。否则，它不具有重要意义。 注意：一旦定

		时器正在运行（STxCEN 位被设置），此位域将无法更改。
9:8	OUTROVM[1:0]	此位定义了在上下计数模式下何时生成翻转事件，根据 HRTIMER_STxCHySET 和 HRTIMER_STxCHyRST 的设置对输出进行设置和/或复位。 00: 当计数器等于 0 或等于 HRTIMER_STxCAR 值时生成事件 01: 当计数器等于 0 时生成事件 10: 当计数器等于 HRTIMER_STxCAR 时生成事件 11: 保留 **注意：**此设置仅在 CAM 位被设置时适用。否则，它不具有重要意义。 注意：一旦定时器正在运行（STxCEN 位被设置），此位域将无法更改。
7:6	ROVM[1:0]	此位定义了在上下计数模式下何时生成翻转事件，该翻转事件用于更新触发器（将内容从预加载传输到活动寄存器），IRQ 和 DMA 请求，减少重复计数器，以及外部事件过滤。 00: 当计数器等于 0 或等于 HRTIMER_STxCAR 值时生成事件 01: 当计数器等于 0 时生成事件 10: 当计数器等于 HRTIMER_STxCAR 时生成事件 11: 保留 **注意：**此设置仅在 CAM 位被设置时适用。否则，它不具有重要意义。 注意：一旦定时器正在运行（STxCEN 位被设置），此位域将无法更改。
5	保留	必须保持复位值
4	CAM	此位定义计数器是在向上计数模式还是在中央对齐计数模式下运行。 0：计数器正在向上计数模式下运行。 1：计数器正在中央对齐计数模式下运行。 注意：一旦定时器开始运行（STxCEN 位被设置），此位将无法更改。
3	保留	必须保持复位值
2	TRIG1M	此位定义何时生成 TRIG1 触发器信号。 0：触发器在计数器复位或翻转事件时生成。 1：触发器在输出 0 置位事件时生成。 **注意：**当 TRIGEN 位复位时（双触发禁用），TRIG1 位无效。
1	TRIG0M	此位定义何时生成 TRIG0 触发器信号。 0：触发器在比较 1 事件时生成。 1：触发器在输出 0 复位事件时生成。 注意：当 TRIGEN 位复位时（双触发禁用），TRIG0 位无效。
0	TRIGEN	双触发使能 此位启用双触发机制。 0：双触发禁用 1：双触发使能

注意：一旦定时器开始运行（STxCEN 位被设置），此位将无法更改。

Slave_TIMERx 外部事件滤波配置寄存器 2 (HRTIMER_STxEXEVFCFG2)

地址偏移：0x70

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

位/位域	名称	描述
31:26	EXEVXCNT[5:0]	外部事件 X 计数器数值 这个位域记录了外部事件 X 的计数器数值。
25:14	保留	必须保持复位值
13:8	EXEVXCNTTHR[5:0]	外部事件 X 计数器阈值 此位域确定外部事件 X 计数器的阈值。当事件数量等于 (EXEVXCNTTHR[5:0]+1) 时，事件发生。
7:4	EXEVXSEL[3:0]	外部事件 X 选择 此位确定外部事件 X 的来源。 0：外部事件 0 作为外部事件 X 的来源。 1：外部事件 1 作为外部事件 X 的来源。 ... 9：外部事件 9 作为外部事件 X 的来源。 其他：保留
3	保留	必须保持复位值
2	EXEVXRSTM	外部事件 X 复位模式 此位确定外部事件 X 计数器的复位模式。 0：在每次复位/翻转事件时，外部事件计数器 X 被复位。 1：仅在上一个计数周期内没有事件发生时，在每次复位/翻转事件时，外部事件计数器 X 被复位。
1	EXEVXCRST	外部事件 X 计数器复位 此位启动外部事件 X 计数器的复位。由软件置位，由硬件复位。 0：无动作

		1：复位外部事件计数器 X
0	EXEVXCEN	外部事件 X 计数器使能 此位启用外部事件 X 计数器。 0：禁用外部事件 X 计数器 1：启用外部事件 X 计数器

Slave_TIMERx 捕获触发复合寄存器 (HRTIMER_STxCAPTRGCOM)

地址偏移：0x74

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

For Slave_TIMER0 - Slave_TIMER4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				CP1BST7CMP1	CP1BST7CMP0	CP1BST7NA	CP1BST7A	CP1BST6CMP1	CP1BST6CMP0	CP1BST6NA	CP1BST6A	CP1BST5CMP1	CP1BST5CMP0	CP1BST5NA	CP1BST5A
				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				CP0BST7CMP1	CP0BST7CMP0	CP0BST7NA	CP0BST7A	CP0BST6CMP1	CP0BST6CMP0	CP0BST6NA	CP0BST6A	CP0BST5CMP1	CP0BST5CMP0	CP0BST5NA	CP0BST5A
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:28	保留	必须保持复位值
27	CP1BST7CMP1	Slave_TIMER7 的比较 1 事件触发捕获 1 请参考 CP0BST5CMP1 描述。
26	CP1BST7CMP0	Slave_TIMER7 的比较 1 事件触发捕获 1 请参考 CP0BST5CMP0 描述。
25	CP1BST7NA	ST7CH0_O 输出有效到无效的变化触发捕获 1 请参考 CP0BST5NA 描述。
24	CP1BST7A	ST7CH0_O 输出无效到有效的变化触发捕获 1 请参考 CP0BST5A 描述。
23	CP1BST6CMP1	Slave_TIMER6 的比较 1 事件触发捕获 1 请参考 CP0BST5CMP1 描述。
22	CP1BST6CMP0	Slave_TIMER6 的比较 1 事件触发捕获 1 请参考 CP0BST5CMP0 描述。

21	CP1BST6NA	ST6CH0_O 输出有效到无效的变化触发捕获 1 请参考 CP0BST5NA 描述。
20	CP1BST6A	ST6CH0_O 输出无效到有效的变化触发捕获 1 请参考 CP0BST5A 描述。
19	CP1BST5CMP1	Slave_TIMER5 的比较 1 事件触发捕获 1 请参考 CP0BST5CMP1 描述。
18	CP1BST5CMP0	Slave_TIMER5 的比较 1 事件触发捕获 1 请参考 CP0BST5CMP0 描述。
17	CP1BST5NA	ST5CH0_O 输出有效到无效的变化触发捕获 1 请参考 CP0BST5NA 描述。
16	CP1BST5A	ST5CH0_O 输出无效到有效的变化触发捕获 1 请参考 CP0BST5A 描述。
15:12	保留	必须保持复位值
11	CP0BST7CMP1	Slave_TIMER7 的比较 1 事件触发捕获 0 请参考 CP0BST5CMP1 描述。
10	CP0BST7CMP0	Slave_TIMER7 的比较 0 事件触发捕获 0 请参考 CP0BST5CMP0 描述。
9	CP0BST7NA	ST7CH0_O 输出有效到无效的变化触发捕获 0 请参考 CP0BST5NA 描述。
8	CP0BST7A	ST7CH0_O 输出无效到有效的变化触发捕获 0 请参考 CP0BST5A 描述。
7	CP0BST6CMP1	Slave_TIMER6 的比较 1 事件触发捕获 0 请参考 CP0BST5CMP1 描述。
6	CP0BST6CMP0	Slave_TIMER6 的比较 0 事件触发捕获 0 请参考 CP0BST5CMP0 描述。
5	CP0BST6NA	ST6CH0_O 输出有效到无效的变化触发捕获 0 请参考 CP0BST5NA 描述。
4	CP0BST6A	ST6CH0_O 输出无效到有效的变化触发捕获 0 请参考 CP0BST5A 描述。
3	CP0BST5CMP1	Slave_TIMER5 的比较 1 事件触发捕获 0 0：Slave_TIMER5 的比较 1 事件不触发捕获 0。 1：Slave_TIMER5 的比较 1 事件触发捕获 0。

2	CP0BST5CMP0	Slave_TIMER5 的比较 0 事件触发捕获 0 0：Slave_TIMER5 的比较 0 事件不触发捕获 0。 1：Slave_TIMER5 的比较 0 事件触发捕获 0。
1	CP0BST5NA	ST5CH0_O 输出有效到无效的变化触发捕获 0 当 Slave_TIMER5 通道 0 的输出从有效电平转换为无效电平时，捕获 1 由 ST5CH0_O 触发。 0：ST5CH0_O 输出有效到无效的变化不触发捕获 0。 1：ST5CH0_O 输出有效到无效的变化触发捕获 0。
0	CP0BST5A	ST5CH0_O 输出无效到有效的变化触发捕获 0 当 Slave_TIMER5 通道 0 的输出从无效电平转换为有效电平时，捕获 0 由 ST5CH0_O 触发。 0：ST5CH0_O 输出无效到有效的变化不触发捕获 0。 1：ST5CH0_O 输出无效到有效的变化触发捕获 0。

For Slave_TIMER5

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				CP1BST7CMP1	CP1BST7CMP0	CP1BST7NA	CP1BST7A	CP1BST6CMP1	CP1BST6CMP0	CP1BST6NA	CP1BST6A	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
保留				CP0BST7CMP1	CP0BST7CMP0	CP0BST7NA	CP0BST7A	CP0BST6CMP1	CP0BST6CMP0	CP0BST6NA	CP0BST6A	保留			
				rw	rw	rw	rw	rw	rw	rw	rw				

位/位域	名称	描述
31:28	保留	必须保持复位值
27	CP1BST7CMP1	Slave_TIMER7 的比较 1 事件触发捕获 1 请参考 CP0BST6CMP1 描述。
26	CP1BST7CMP0	Slave_TIMER7 的比较 1 事件触发捕获 1 请参考 CP0BST6CMP0 描述。
25	CP1BST7NA	ST7CH0_O 输出有效到无效的变化触发捕获 1 请参考 CP0BST6NA 描述。
24	CP1BST7A	ST7CH0_O 输出无效到有效的变化触发捕获 1 请参考 CP0BST6A 描述。
23	CP1BST6CMP1	Slave_TIMER6 的比较 1 事件触发捕获 1 请参考 CP0BST6CMP1 描述。

22	CP1BST6CMP0	Slave_TIMER6 的比较 1 事件触发捕获 1 请参考 CP0BST6CMP0 描述。
21	CP1BST6NA	ST6CH0_O 输出有效到无效的变化触发捕获 1 请参考 CP0BST6NA 描述。
20	CP1BST6A	ST6CH0_O 输出无效到有效的变化触发捕获 1 请参考 CP0BST6A 描述。
19:12	保留	必须保持复位值
11	CP0BST7CMP1	Slave_TIMER7 的比较 1 事件触发捕获 0 请参考 CP0BST6CMP1 描述。
10	CP0BST7CMP0	Slave_TIMER7 的比较 0 事件触发捕获 0 请参考 CP0BST6CMP0 描述。
9	CP0BST7NA	ST7CH0_O 输出有效到无效的变化触发捕获 0 请参考 CP0BST6NA 描述。
8	CP0BST7A	ST7CH0_O 输出无效到有效的变化触发捕获 0 请参考 CP0BST6A 描述。
7	CP0BST6CMP1	Slave_TIMER6 的比较 1 事件触发捕获 0 0：Slave_TIMER6 的比较 1 事件不触发捕获 0。 1：Slave_TIMER6 的比较 1 事件触发捕获 0。
6	CP0BST6CMP0	Slave_TIMER6 的比较 0 事件触发捕获 0 0：Slave_TIMER6 的比较 0 事件不触发捕获 0。 1：Slave_TIMER6 的比较 0 事件触发捕获 0。
5	CP0BST6NA	ST6CH0_O 输出有效到无效的变化触发捕获 0 当 Slave_TIMER6 通道 0 的输出从有效电平转换为无效电平时，捕获 1 由 ST6CH0_O 触发。 0：ST6CH0_O 输出有效到无效的变化不触发捕获 0。 1：ST6CH0_O 输出有效到无效的变化触发捕获 0。
4	CP0BST6A	ST6CH0_O 输出无效到有效的变化触发捕获 0 当 Slave_TIMER6 通道 0 的输出从无效电平转换为有效电平时，捕获 0 由 ST6CH0_O 触发。 0：ST6CH0_O 输出无效到有效的变化不触发捕获 0。 1：ST6CH0_O 输出无效到有效的变化触发捕获 0。
3:0	保留	必须保持复位值

For Slave_TIMER6

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				CP1BST7CMP1	CP1BST7CMP0	CP1BST7NA	CP1BST7A	保留				CP1BST5CMP1	CP1BST5CMP0	CP1BST5NA	CP1BST5A
				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				CP0BST7CMP1	CP0BST7CMP0	CP0BST7NA	CP0BST7A	保留				CP0BST5CMP1	CP0BST5CMP0	CP0BST5NA	CP0BST5A
				rw	rw	rw	rw					rw	rw	rw	rw

位/位域	名称	描述
31:28	保留	必须保持复位值
27	CP1BST7CMP1	Slave_TIMER7 的比较 1 事件触发捕获 1 请参考 CP0BST5CMP1 描述。
26	CP1BST7CMP0	Slave_TIMER7 的比较 1 事件触发捕获 1 请参考 CP0BST5CMP0 描述。
25	CP1BST7NA	ST7CH0_O 输出有效到无效的变化触发捕获 1 请参考 CP0BST5NA 描述。
24	CP1BST7A	ST7CH0_O 输出无效到有效的变化触发捕获 1 请参考 CP0BST5A 描述。
23:20	保留	必须保持复位值
19	CP1BST5CMP1	Slave_TIMER5 的比较 1 事件触发捕获 1 请参考 CP0BST5CMP1 描述。
18	CP1BST5CMP0	Slave_TIMER5 的比较 1 事件触发捕获 1 请参考 CP0BST5CMP0 描述。
17	CP1BST5NA	ST5CH0_O 输出有效到无效的变化触发捕获 1 请参考 CP0BST5NA 描述。
16	CP1BST5A	ST5CH0_O 输出无效到有效的变化触发捕获 1 请参考 CP0BST5A 描述。

15:12	保留	必须保持复位值
11	CP0BST7CMP1	Slave_TIMER7 的比较 1 事件触发捕获 0 请参考 CP0BST5CMP1 描述。
10	CP0BST7CMP0	Slave_TIMER7 的比较 0 事件触发捕获 0 请参考 CP0BST5CMP0 描述。
9	CP0BST7NA	ST7CH0_O 输出有效到无效的变化触发捕获 0 请参考 CP0BST5NA 描述。
8	CP0BST7A	ST7CH0_O 输出无效到有效的变化触发捕获 0 请参考 CP0BST5A 描述。
7:4	保留	必须保持复位值
3	CP0BST5CMP1	Slave_TIMER5 的比较 1 事件触发捕获 0 0：Slave_TIMER5 的比较 1 事件不触发捕获 0。 1：Slave_TIMER5 的比较 1 事件触发捕获 0。
2	CP0BST5CMP0	Slave_TIMER5 的比较 0 事件触发捕获 0 0：Slave_TIMER5 的比较 0 事件不触发捕获 0。 1：Slave_TIMER5 的比较 0 事件触发捕获 0。
1	CP0BST5NA	ST5CH0_O 输出有效到无效的变化触发捕获 0 当 Slave_TIMER5 通道 0 的输出从有效电平转换为无效电平时，捕获 1 由 ST5CH0_O 触发。 0：ST5CH0_O 输出有效到无效的变化不触发捕获 0。 1：ST5CH0_O 输出有效到无效的变化触发捕获 0。
0	CP0BST5A	ST5CH0_O 输出无效到有效的变化触发捕获 0 当 Slave_TIMER5 通道 0 的输出从无效电平转换为有效电平时，捕获 0 由 ST5CH0_O 触发。 0：ST5CH0_O 输出无效到有效的变化不触发捕获 0。 1：ST5CH0_O 输出无效到有效的变化触发捕获 0。

For Slave_TIMER7

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								CP1BST6CMP1	CP1BST6CMP0	CP1BST6NA	CP1BST6A	CP1BST5CMP1	CP1BST5CMP0	CP1BST5NA	CP1BST5A
								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								CP0BST6CMP1	CP0BST6CMP0	CP0BST6NA	CP0BST6A	CP0BST5CMP1	CP0BST5CMP0	CP0BST5NA	CP0BST5A

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	CP1BST6CMP1	Slave_TIMER6 的比较 1 事件触发捕获 1 请参考 CP0BST5CMP1 描述。
22	CP1BST6CMP0	Slave_TIMER6 的比较 1 事件触发捕获 1 请参考 CP0BST5CMP0 描述。
21	CP1BST6NA	ST6CH0_O 输出有效到无效的变化触发捕获 1 请参考 CP0BST5NA 描述。
20	CP1BST6A	ST6CH0_O 输出无效到有效的变化触发捕获 1 请参考 CP0BST5A 描述。
19	CP1BST5CMP1	Slave_TIMER5 的比较 1 事件触发捕获 1 请参考 CP0BST5CMP1 描述。
18	CP1BST5CMP0	Slave_TIMER5 的比较 1 事件触发捕获 1 请参考 CP0BST5CMP0 描述。
17	CP1BST5NA	ST5CH0_O 输出有效到无效的变化触发捕获 1 请参考 CP0BST5NA 描述。
16	CP1BST5A	ST5CH0_O 输出无效到有效的变化触发捕获 1 请参考 CP0BST5A 描述。
15:8	保留	必须保持复位值
7	CP0BST6CMP1	Slave_TIMER6 的比较 1 事件触发捕获 0 请参考 CP0BST5CMP1 描述。
6	CP0BST6CMP0	Slave_TIMER6 的比较 0 事件触发捕获 0 请参考 CP0BST5CMP0 描述。
5	CP0BST6NA	ST6CH0_O 输出有效到无效的变化触发捕获 0 请参考 CP0BST5NA 描述。
4	CP0BST6A	ST6CH0_O 输出无效到有效的变化触发捕获 0 请参考 CP0BST5A 描述。
3	CP0BST5CMP1	Slave_TIMER5 的比较 1 事件触发捕获 0 0：Slave_TIMER5 的比较 1 事件不触发捕获 0。 1：Slave_TIMER5 的比较 1 事件触发捕获 0。
2	CP0BST5CMP0	Slave_TIMER5 的比较 0 事件触发捕获 0

0：Slave_TIMER5 的比较 0 事件不触发捕获 0。
1：Slave_TIMER5 的比较 0 事件触发捕获 0。

1	CP0BST5NA	ST5CH0_O 输出有效到无效的变化触发捕获 0 当 Slave_TIMER5 通道 0 的输出从有效电平转换为无效电平时，捕获 1 由 ST5CH0_O 触发。 0：ST5CH0_O 输出有效到无效的变化不触发捕获 0。 1：ST5CH0_O 输出有效到无效的变化触发捕获 0。
0	CP0BST5A	ST5CH0_O 输出无效到有效的变化触发捕获 0 当 Slave_TIMER5 通道 0 的输出从无效电平转换为有效电平时，捕获 0 由 ST5CH0_O 触发。 0：ST5CH0_O 输出无效到有效的变化不触发捕获 0。 1：ST5CH0_O 输出无效到有效的变化触发捕获 0。

Slave_TIMERx 计数器复位和通道 0/通道 1 置位复位请求附加寄存器 (HRTIMER_STxCNTRSTA)

地址偏移：0x78
复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

For Slave_TIMER0 - Slave_TIMER4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		CH1RSSTCNTRST	CH0RSSTCNTRST	CH1RSSTEV10	CH1RSSTEV9	CH0RSSTEV10	CH0RSSTEV9	保留		CH1SSTCNTRST	CH0SSTCNTRST	CH1SSTEV10	CH1SSTEV9	CH0SSTEV10	CH0SSTEV9
		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									ST7CMP3RST	ST7CMP1RST	ST7CMP0RST	ST6CMP3RST	ST6CMP1RST	ST6CMP0RST	ST5CMP3RST
									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	CH1RSSTCNTRST	计数器复位事件生成通道 1 复位请求 请参考 CH0SSTCNTRST 说明。
28	CH0RSSTCNTRST	计数器复位事件生成通道 0 复位请求 请参考 CH0SSTCNTRST 说明。

27	CH1RSSTEV10	Slave_TIMERx 互连事件 10 生成通道 1 复位请求 请参考 CH0SSTEV0 说明。
26	CH1RSSTEV9	Slave_TIMERx 互连事件 9 生成通道 1 复位请求 请参考 CH0SEXEV0 说明。
25	CH0RSSTEV10	Slave_TIMERx 互连事件 10 生成通道 0 复位请求 请参考 CH0SSTEV0 说明。
24	CH0RSSTEV9	Slave_TIMERx 互连事件 9 生成通道 0 复位请求 0：该事件不生成通道 0 复位请求。 1：该事件生成通道 0 复位请求。
23:22	保留	必须保持复位值
21	CH1SSTCNTRST	计数器复位事件生成通道 1 置位请求 请参考 CH0SSTCNTRS 说明。
20	CH0SSTCNTRST	计数器复位事件生成通道 0 置位请求 0：计数器复位事件不生成通道 0 置位请求。 1：计数器复位事件生成通道 0 置位请求。
19	CH1SSTEV10	Slave_TIMERx 互连事件 10 生成通道 1 置位请求 请参考 CH0SSTEV9 说明。
18	CH1SSTEV9	Slave_TIMERx 互连事件 9 生成通道 1 置位请求 请参考 CH0SSTEV9 说明。
17	CH0SSTEV10	Slave_TIMERx 互连事件 10 生成通道 0 置位请求 请参考 CH0SSTEV9 说明。
16	CH0SSTEV9	Slave_TIMERx 互连事件 9 生成通道 0 置位请求 0：该事件不生成通道 0 置位请求。 1：该事件生成通道 0 置位请求。
15:7	保留	必须保持复位值
6	ST7CMP3RST	Slave_TIMER7 比较 3 事件复位计数器 请参考 ST5CMP3RST 说明。
5	ST7CMP1RST	Slave_TIMER7 比较 1 事件复位计数器 请参考 ST5CMP3RST 说明。
4	ST7CMP0RST	Slave_TIMER7 比较 0 事件复位计数器 请参考 ST5CMP3RST 说明。
3	ST6CMP3RST	Slave_TIMER6 比较 3 事件复位计数器

请参考 ST5CMP3RST 说明。

2 ST6CMP1RST Slave_TIMER6 比较 1 事件复位计数器
请参考 ST5CMP3RST 说明。

1 ST6CMP0RST Slave_TIMER6 比较 0 事件复位计数器
请参考 ST5CMP3RST 说明。

0 ST5CMP3RST Slave_TIMER5 比较 3 事件复位计数器
0：Slave_TIMER5 比较 3 事件不复位计数器。
1：Slave_TIMER5 比较 3 事件复位计数器。

For Slave_TIMER5

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		CH1RSSTCNTRST	CH0RSSTCNTRST	CH1RSSTEV10	CH1RSSTEV9	CH0RSSTEV10	CH0RSSTEV9	保留		CH1SSTCNTRST	CH0SSTCNTRST	CH1SSTEV10	CH1SSTEV9	CH0SSTEV10	CH0SSTEV9
		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									ST7CMP3RST	ST7CMP1RST	ST7CMP0RST	ST6CMP3RST	ST6CMP1RST	ST6CMP0RST	ST4CMP3RST
									rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	保留	必须保持复位值
29	CH1RSSTCNTRST	计数器复位事件生成通道 1 复位请求 请参考 CH0SSTCNTRST 说明。
28	CH0RSSTCNTRST	计数器复位事件生成通道 0 复位请求 请参考 CH0SSTCNTRST 说明。
27	CH1RSSTEV10	Slave_TIMERx 互连事件 10 生成通道 1 复位请求 请参考 CH0SSTEV0 说明。
26	CH1RSSTEV9	Slave_TIMERx 互连事件 9 生成通道 1 复位请求 请参考 CH0SEXEV0 说明。
25	CH0RSSTEV10	Slave_TIMERx 互连事件 10 生成通道 0 复位请求 请参考 CH0SSTEV0 说明。
24	CH0RSSTEV9	Slave_TIMERx 互连事件 9 生成通道 0 复位请求 0：该事件不生成复位请求。 1：该事件生成复位请求。

23:22	保留	必须保持复位值
21	CH1SSTCNTRST	计数器复位事件生成通道 1 置位请求 请参考 CH0SSTCNTRST 说明。
20	CH0SSTCNTRST	计数器复位事件生成通道 0 置位请求 0：计数器复位事件不生成通道 0 置位请求。 1：计数器复位事件生成通道 0 置位请求。
19	CH1SSTEV10	Slave_TIMERx 互连事件 10 生成通道 1 置位请求 请参考 CH0SSTEV9 说明。
18	CH1SSTEV9	Slave_TIMERx 互连事件 9 生成通道 1 置位请求 请参考 CH0SSTEV9 说明。
17	CH0SSTEV10	Slave_TIMERx 互连事件 10 生成通道 0 置位请求 请参考 CH0SSTEV9 说明。
16	CH0SSTEV9	Slave_TIMERx 互连事件 9 生成通道 0 置位请求 0：该事件不生成置位请求。 1：该事件生成置位请求。
15:7	保留	必须保持复位值
6	ST7CMP3RST	Slave_TIMER7 比较 3 事件复位计数器 请参考 ST4CMP3RST 说明。
5	ST7CMP1RST	Slave_TIMER7 比较 1 事件复位计数器 请参考 ST4CMP3RST 说明。
4	ST7CMP0RST	Slave_TIMER7 比较 0 事件复位计数器 请参考 ST4CMP3RST 说明。
3	ST6CMP3RST	Slave_TIMER6 比较 3 事件复位计数器 请参考 ST4CMP3RST 说明。
2	ST6CMP1RST	Slave_TIMER6 比较 1 事件复位计数器 请参考 ST4CMP3RST 说明。
1	ST6CMP0RST	Slave_TIMER6 比较 0 事件复位计数器 请参考 ST4CMP3RST 说明。
0	ST4CMP3RST	Slave_TIMER4 比较 3 事件复位计数器 0：Slave_TIMER4 比较 3 事件不复位计数器。 1：Slave_TIMER4 比较 3 事件复位计数器。

For Slave_TIMER6

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		CH1RSSTCNTRST	CH0RSSTCNTRST	CH1RSSTEV10	CH1RSSTEV9	CH0RSSTEV10	CH0RSSTEV9	保留		CH1SSTCNTRST	CH0SSTCNTRST	CH1SSTEV10	CH1SSTEV9	CH0SSTEV10	CH0SSTEV9
		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									ST7CMP3RST	ST7CMP1RST	ST7CMP0RST	ST5CMP3RST	ST5CMP1RST	ST5CMP0RST	ST4CMP3RST
									rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	保留	必须保持复位值
29	CH1RSSTCNTRST	计数器复位事件生成通道 1 复位请求 请参考 CH0SSTCNTRST 说明。
28	CH0RSSTCNTRST	计数器复位事件生成通道 0 复位请求 请参考 CH0SSTCNTRST 说明。
27	CH1RSSTEV10	Slave_TIMERx 互连事件 10 生成通道 1 复位请求 请参考 CH0SSTEV0 说明。
26	CH1RSSTEV9	Slave_TIMERx 互连事件 9 生成通道 1 复位请求 请参考 CH0SEXEV0 说明。
25	CH0RSSTEV10	Slave_TIMERx 互连事件 10 生成通道 0 复位请求 请参考 CH0SSTEV0 说明。
24	CH0RSSTEV9	Slave_TIMERx 互连事件 9 生成通道 0 复位请求 0：该事件不生成复位请求。 1：该事件生成复位请求。
23:22	保留	必须保持复位值
21	CH1SSTCNTRST	计数器复位事件生成通道 1 置位请求 请参考 CH0SSTCNTRST 说明。
20	CH0SSTCNTRST	计数器复位事件生成通道 0 置位请求 0：计数器复位事件不生成通道 0 置位请求。 1：计数器复位事件生成通道 0 置位请求。
19	CH1SSTEV10	Slave_TIMERx 互连事件 10 生成通道 1 置位请求 请参考 CH0SSTEV9 说明。

18	CH1SSTEV9	Slave_TIMERx 互连事件 9 生成通道 1 置位请求 请参考 CH0SSTEV9 说明。
17	CH0SSTEV10	Slave_TIMERx 互连事件 10 生成通道 0 置位请求 请参考 CH0SSTEV9 说明。
16	CH0SSTEV9	Slave_TIMERx 互连事件 9 生成通道 0 置位请求 0：该事件不生成置位请求。 1：该事件生成置位请求。
15:7	保留	必须保持复位值
6	ST7CMP3RST	Slave_TIMER7 比较 3 事件复位计数器 请参考 ST4CMP3RST 说明。
5	ST7CMP1RST	Slave_TIMER7 比较 1 事件复位计数器 请参考 ST4CMP3RST 说明。
4	ST7CMP0RST	Slave_TIMER7 比较 0 事件复位计数器 请参考 ST4CMP3RST 说明。
3	ST5CMP3RST	Slave_TIMER5 比较 3 事件复位计数器 请参考 ST4CMP3RST 说明。
2	ST5CMP1RST	Slave_TIMER5 比较 1 事件复位计数器 请参考 ST4CMP3RST 说明。
1	ST5CMP0RST	Slave_TIMER5 比较 0 事件复位计数器 请参考 ST4CMP3RST 说明。
0	ST4CMP3RST	Slave_TIMER4 比较 3 事件复位计数器 0：Slave_TIMER4 比较 3 事件不复位计数器。 1：Slave_TIMER4 比较 3 事件复位计数器。

For Slave_TIMER7

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		CH1RSSTCNTRST	CH0RSSTCNTRST	CH1RSSTEV10	CH1RSSTEV9	CH0RSSTEV10	CH0RSSTEV9	保留		CH1SSTCNTRST	CH0SSTCNTRST	CH1SSTEV10	CH1SSTEV9	CH0SSTEV10	CH0SSTEV9
		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									ST6CMP3RST	ST6CMP1RST	ST6CMP0RST	ST5CMP3RST	ST5CMP1RST	ST5CMP0RST	ST4CMP3RST
									rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	保留	必须保持复位值
29	CH1RSSTCNTRST	计数器复位事件生成通道 1 复位请求 请参考 CH0SSTCNTRST 说明。
28	CH0RSSTCNTRST	计数器复位事件生成通道 0 复位请求 请参考 CH0SSTCNTRST 说明。
27	CH1RSSTEV10	Slave_TIMERx 互连事件 10 生成通道 1 复位请求 请参考 CH0SSTEV0 说明。
26	CH1RSSTEV9	Slave_TIMERx 互连事件 9 生成通道 1 复位请求 请参考 CH0SEXEV0 说明。
25	CH0RSSTEV10	Slave_TIMERx 互连事件 10 生成通道 0 复位请求 请参考 CH0SSTEV0 说明。
24	CH0RSSTEV9	Slave_TIMERx 互连事件 9 生成通道 0 复位请求 0：该事件不生成复位请求。 1：该事件生成复位请求。
23:22	保留	必须保持复位值
21	CH1SSTCNTRST	计数器复位事件生成通道 1 置位请求 请参考 CH0SSTCNTRST 说明。
20	CH0SSTCNTRST	计数器复位事件生成通道 0 置位请求 0：计数器复位事件不生成通道 0 置位请求。 1：计数器复位事件生成通道 0 置位请求。
19	CH1SSTEV10	Slave_TIMERx 互连事件 10 生成通道 1 置位请求 请参考 CH0SSTEV9 说明。
18	CH1SSTEV9	Slave_TIMERx 互连事件 9 生成通道 1 置位请求 请参考 CH0SSTEV9 说明。
17	CH0SSTEV10	Slave_TIMERx 互连事件 10 生成通道 0 置位请求 请参考 CH0SSTEV9 说明。
16	CH0SSTEV9	Slave_TIMERx 互连事件 9 生成通道 0 置位请求 0：该事件不生成置位请求。 1：该事件生成置位请求。
15:7	保留	必须保持复位值
6	ST6CMP3RST	Slave_TIMER6 比较 3 事件复位计数器

		请参考 ST4CMP3RST 说明。
5	ST6CMP1RST	Slave_TIMER6 比较 1 事件复位计数器 请参考 ST4CMP3RST 说明。
4	ST6CMP0RST	Slave_TIMER6 比较 0 事件复位计数器 请参考 ST4CMP3RST 说明。
3	ST5CMP3RST	Slave_TIMER5 比较 3 事件复位计数器 请参考 ST4CMP3RST 说明。
2	ST5CMP1RST	Slave_TIMER5 比较 1 事件复位计数器 请参考 ST4CMP3RST 说明。
1	ST5CMP0RST	Slave_TIMER5 比较 0 事件复位计数器 请参考 ST4CMP3RST 说明。
0	ST4CMP3RST	Slave_TIMER4 比较 3 事件复位计数器 0：Slave_TIMER4 比较 3 事件不复位计数器。 1：Slave_TIMER4 比较 3 事件复位计数器。

Slave_TIMERx 附加控制寄存器 (HRTIMER_STxACTL)

地址偏移：0x7C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DTFCFG[15:9]							保留								
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTRCFG[15:9]							FLTAR	保留						UPBST7	UPBST6
rw							rw							rw	rw

位/位域	名称	描述
31:25	DTFCFG[15:9]	下降沿死区值配置 该位域用于配置跟随输出准备信号（OyPRE，y = 0,1）下降沿之后的死区时间值。 DTF 值 = DTFCFG [15:0] x $t_{HRTIMER_DTGCK}$，其中，$t_{HRTIMER_DTGCK}$ = 1/ $f_{HRTIMER_DTGCK}$。 写入该位域可以更改 DTFCFG [15:0]位域的高 7 位。 **注意：** （1）当 HRTIMER_STxDTCTL 寄存器中的 DTFSVPROT 位置 1 时，无法修改此位域。 （2）该位是预装载的。

24:16	保留	必须保持复位值
15:9	DTRCFG[15:9]	上升沿死区值配置 该位域用于配置跟随输出准备信号（OyPRE，y = 0,1）上升沿之后的死区时间值。 DTR 值= DTRCFG[15:0] x $t_{HRTIMER_DTGCK}$，其中，$t_{HRTIMER_DTGCK}$ = 1/ $f_{HRTIMER_DTGCK}$。 写入该位域可以更改 DTRCFG[15:0]位域的高 7 位。 **注意：** （1）当 HRTIMER_STxDTCTL 寄存器中的 DTRSVPROT 位置 1 时，无法修改此位域。 （2）该位是预装载的。
8	FLTAR	故障自动恢复 该位用于选择当所有故障源都处于失效时，通道输出是否自动重新恢复。 0：禁能。 1：使能。 **注意：**一旦定时器开始运行（ STxCEN 位被设置），此位将无法更改。
7:2	保留	必须保持复位值
1	UPBST7	通过 Slave_TIMER7 更新事件进行更新 该位置 1 时，Slave_TIMERx（x=0,1,2,3,4,5,6）更新事件与 Slave_TIMER7 更新事件同步，且它们的有效寄存器由 Slave_TIMER7 更新事件进行更新。 0：有效寄存器不由 Slave_TIMER7 更新。 1：有效寄存器由 Slave_TIMER7 更新。 **注意：**Slave_TIMER7 的寄存器中不存在此位。
0	UPBST6	通过 Slave_TIMER6 更新事件进行更新 该位置 1 时，Slave_TIMERx（x=0,1,2,3,4,5,7）更新事件与 Slave_TIMER6 更新事件同步，且它们的有效寄存器由 Slave_TIMER6 更新事件进行更新。 0：有效寄存器不由 Slave_TIMER6 更新。 1：有效寄存器由 Slave_TIMER6 更新。 **注意：**Slave_TIMER6 的寄存器中不存在此位。

25.5.3. 通用寄存器

HRTIMER 通用寄存器基地址：0x4001 5B80

HRTIMER 控制寄存器 0 (HRTIMER_CTL0)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADTG3USRC[3]	ADTG2USRC[3]	ADTG1USRC[3]	ADTG0USRC[3]	ADTG3USRC[2:0]			ADTG2USRC[2:0]			ADTG1USRC[2:0]			ADTG0USRC[2:0]		
rw	rw	rw	rw	rw			rw			rw			rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							ST7UPDIS	ST6UPDIS	ST5UPDIS	ST4UPDIS	ST3UPDIS	ST2UPDIS	ST1UPDIS	ST0UPDIS	MTUPDIS
							rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	ADTG3USRC[3]	请参考 ADTG3USRC[2:0]说明。
30	ADTG2USRC[3]	请参考 ADTG2USRC[2:0]说明。
29	ADTG1USRC[3]	请参考 ADTG1USRC[2:0]说明。
28	ADTG0USRC[3]	请参考 ADTG0USRC[2:0]说明。
27:25	ADTG3USRC[2:0]	HRTIMER_ADCTRIG3 更新源 该位域可以由软件配置，配置 HRTIMER_ADCTRIGS3 寄存器的更新源。 0000：Master_TIMER 更新事件 0001：Slaver_TIMER0 更新事件 0010：Slaver_TIMER1 更新事件 0011：Slaver_TIMER2 更新事件 0100：Slaver_TIMER3 更新事件 0101：Slaver_TIMER4 更新事件 0110：Slaver_TIMER5 更新事件 0111：Slaver_TIMER6 更新事件 1000：Slaver_TIMER7 更新事件 其他值保留。
24:22	ADTG2USRC[2:0]	HRTIMER_ADCTRIG2 更新源 该位域可以由软件配置，配置 HRTIMER_ADCTRIGS2 寄存器的更新源。 0000：Master_TIMER 更新事件 0001：Slaver_TIMER0 更新事件 0010：Slaver_TIMER1 更新事件 0011：Slaver_TIMER2 更新事件 0100：Slaver_TIMER3 更新事件 0101：Slaver_TIMER4 更新事件 0110：Slaver_TIMER5 更新事件 0111：Slaver_TIMER6 更新事件 1000：Slaver_TIMER7 更新事件

		其他值保留。
21:19	ADTG1USRC[2:0]	HRTIMER_ADCTRIG1 更新源 该位域可以由软件配置，配置 HRTIMER_ADCTRIGS1 寄存器的更新源。 0000：Master_TIMER 更新事件 0001：Slaver_TIMER0 更新事件 0010：Slaver_TIMER1 更新事件 0011：Slaver_TIMER2 更新事件 0100：Slaver_TIMER3 更新事件 0101：Slaver_TIMER4 更新事件 0110：Slaver_TIMER5 更新事件 0111：Slaver_TIMER6 更新事件 1000：Slaver_TIMER7 更新事件 其他值保留。
18:16	ADTG0USRC[2:0]	HRTIMER_ADCTRIG0 更新源 该位域可以由软件配置，配置 HRTIMER_ADCTRIGS0 寄存器的更新源。 0000：Master_TIMER 更新事件 0001：Slaver_TIMER0 更新事件 0010：Slaver_TIMER1 更新事件 0011：Slaver_TIMER2 更新事件 0100：Slaver_TIMER3 更新事件 0101：Slaver_TIMER4 更新事件 0110：Slaver_TIMER5 更新事件 0111：Slaver_TIMER6 更新事件 1000：Slaver_TIMER7 更新事件 其他值保留。
15:9	保留	必须保持复位值
8	ST7UPDIS	Slave_TIMER7 更新事件禁能 该位用于使能或禁能更新事件的生成。 0：更新事件使能。 1：更新事件禁能。
7	ST6UPDIS	Slave_TIMER6 更新事件禁能 该位用于使能或禁能更新事件的生成。 0：更新事件使能。 1：更新事件禁能。
6	ST5UPDIS	Slave_TIMER5 更新事件禁能 该位用于使能或禁能更新事件的生成。 0：更新事件使能。

1：更新事件禁能。

5	ST4UPDIS	Slave_TIMER4 更新事件禁能 该位用于使能或禁能更新事件的生成。 0：更新事件使能。 1：更新事件禁能。
4	ST3UPDIS	Slave_TIMER3 更新事件禁能 该位用于使能或禁能更新事件的生成。 0：更新事件使能。 1：更新事件禁能。
3	ST2UPDIS	Slave_TIMER2 更新事件禁能 该位用于使能或禁能更新事件的生成。 0：更新事件使能。 1：更新事件禁能。
2	ST1UPDIS	Slave_TIMER1 更新事件禁能 该位用于使能或禁能更新事件的生成。 0：更新事件使能。 1：更新事件禁能。
1	ST0UPDIS	Slave_TIMER0 更新事件禁能 该位用于使能或禁能更新事件的生成。 0：更新事件使能。 1：更新事件禁能。
0	MTUPDIS	Master_TIMER 更新事件禁能 该位用于使能或禁能更新事件的生成。 0：更新事件使能。 1：更新事件禁能。

HRTIMER 控制寄存器 1 (HRTIMER_CTL1)

地址偏移：0x04

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		ST7SRST	ST7SUP	保留				EXC7	EXC6	EXC5	EXC4	EXC3	EXC2	EXC1	EXC0
		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
ST6SRST	ST5SRST	ST4SRST	ST3SRST	ST2SRST	ST1SRST	ST0SRST	MTSRST	ST6SUP	ST5SUP	ST4SUP	ST3SUP	ST2SUP	ST1SUP	ST0SUP	MTSUP
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	保留	必须保持复位值
29	ST7SRST	Slave_TIMER7 软件复位 该位可由软件置位，硬件自动清除。该位置 1 时，计数器复位。 0：无影响。 1：计数器复位。
28	ST7SUP	Slave_TIMER7 软件更新 该位可由软件置位，硬件自动清除。该位置 1 时，影子寄存器的内容被传送到有效寄存器，并且所有挂起的更新请求都被取消。 0：无影响。 1：更新生成。
27:24	保留	必须保持复位值
23	EXC7	交换 Slave_TIMER7 输出 请参考 EXC0 说明。
22	EXC6	交换 Slave_TIMER6 输出 请参考 EXC0 说明
21	EXC5	交换 Slave_TIMER5 输出 请参考 EXC0 说明
20	EXC4	交换 Slave_TIMER4 输出 请参考 EXC0 说明
19	EXC3	交换 Slave_TIMER3 输出 请参考 EXC0 说明
18	EXC2	交换 Slave_TIMER2 输出 请参考 EXC0 说明
17	EXC1	交换 Slave_TIMER1 输出 请参考 EXC0 说明
16	EXC0	交换 Slave_TIMER0 输出 0: HRTIMER_STxCH0SET 和 HRTIMER_STxCH0RST 编码 CHO 输出，而 HRTIMER_STxCH1SET 和 HRTIMER_STxCH1RST 编码 CH1 输出。 1: HRTIMER_STxCH0SET 和 HRTIMER_STxCH0RST 编码 CH1 输出，而 HRTIMER_STxCH1SET 和 HRTIMER_STxCH1RST 编码 CH0 输出。
15	ST6SRST	Slave_TIMER6 软件复位 该位可由软件置位，硬件自动清除。该位置 1 时，计数器复位。

0：无影响。
1：计数器复位。

14 ST5SRST Slave_TIMER5 软件复位
该位可由软件置位，硬件自动清除。该位置 1 时，计数器复位。
0：无影响。
1：计数器复位。

13 ST4SRST Slave_TIMER4 软件复位
该位可由软件置位，硬件自动清除。该位置 1 时，计数器复位。
0：无影响。
1：计数器复位。

12 ST3SRST Slave_TIMER3 软件复位
该位可由软件置位，硬件自动清除。该位置 1 时，计数器复位。
0：无影响。
1：计数器复位。

11 ST2SRST Slave_TIMER2 软件复位
该位可由软件置位，硬件自动清除。该位置 1 时，计数器复位。
0：无影响。
1：计数器复位。

10 ST1SRST Slave_TIMER1 软件复位
该位可由软件置位，硬件自动清除。该位置 1 时，计数器复位。
0：无影响。
1：计数器复位。

9 ST0SRST Slave_TIMER0 软件复位
该位可由软件置位，硬件自动清除。该位置 1 时，计数器复位。
0：无影响。
1：计数器复位。

8 MTSRST Master_TIMER 软件复位
该位可由软件置位，硬件自动清除。该位置 1 时，计数器复位。
0：无影响。
1：计数器复位。

7 ST6SUP Slave_TIMER6 软件更新
该位可由软件置位，硬件自动清除。该位置 1 时，影子寄存器的内容被传送到有效寄存器，并且所有挂起的更新请求都被取消。
0：无影响。
1：更新生成。

6 ST5SUP Slave_TIMER5 软件更新

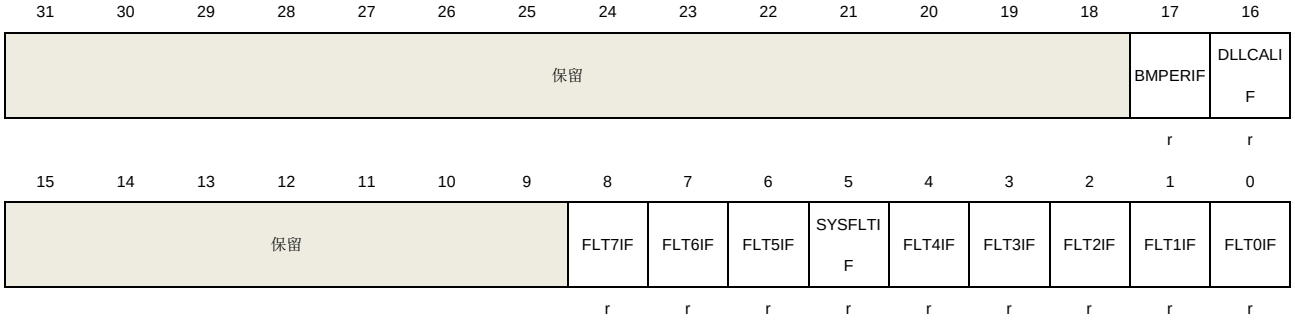
		该位可由软件置位，硬件自动清除。该位置 1 时，影子寄存器的内容被传送到有效寄存器，并且所有挂起的更新请求都被取消。 0：无影响。 1：更新生成。
5	ST4SUP	Slave_TIMER4 软件更新 该位可由软件置位，硬件自动清除。该位置 1 时，影子寄存器的内容被传送到有效寄存器，并且所有挂起的更新请求都被取消。 0：无影响。 1：更新生成。
4	ST3SUP	Slave_TIMER3 软件更新 该位可由软件置位，硬件自动清除。该位置 1 时，影子寄存器的内容被传送到有效寄存器，并且所有挂起的更新请求都被取消。 0：无影响。 1：更新生成。
3	ST2SUP	Slave_TIMER2 软件更新 该位可由软件置位，硬件自动清除。该位置 1 时，影子寄存器的内容被传送到有效寄存器，并且所有挂起的更新请求都被取消。 0：无影响。 1：更新生成。
2	ST1SUP	Slave_TIMER1 软件更新 该位可由软件置位，硬件自动清除。该位置 1 时，影子寄存器的内容被传送到有效寄存器，并且所有挂起的更新请求都被取消。 0：无影响。 1：更新生成。
1	ST0SUP	Slave_TIMER0 软件更新 该位可由软件置位，硬件自动清除。该位置 1 时，影子寄存器的内容被传送到有效寄存器，并且所有挂起的更新请求都被取消。 0：无影响。 1：更新生成。
0	MTSUP	Master_TIMER 软件更新 该位可由软件置位，硬件自动清除。该位置 1 时，影子寄存器的内容被传送到有效寄存器，并且所有挂起的更新请求都被取消。 0：无影响。 1：更新生成。

HRTIMER 中断标志寄存器 (HRTIMER_INTF)

地址偏移：0x08

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。



位/位域	名称	描述
31:18	保留	必须保持复位值
17	BMPERIF	突发模式周期中断标志 突发模式的周期时间到达时，该位由硬件置位。可以通过软件写 1 清零。 0：突发模式周期中断未发生 1：突发模式周期中断发生
16	DLLCALIF	DLL 校准完成中断标志 DLL 校准完成后，此标志由硬件置 1。可以通过软件写 1 清零。 0：DLL 校准完成中断未发生 1：DLL 校准完成中断发生
15:9	保留	必须保持复位值
8	FLT7IF	故障 7 中断标志 请参考 FLT0IF 描述。
7	FLT6IF	故障 6 中断标志 请参考 FLT0IF 描述。
6	FLT5IF	故障 5 中断标志 请参考 FLT0IF 描述。
5	SYSFLTIF	系统故障中断标志 系统故障发生时，此标志由硬件置 1。可以通过软件写 1 清零。 0：系统故障中断未发生 1：系统故障中断发生
4	FLT4IF	故障 4 中断标志

请参考 FLT0IF 描述。

3	FLT3IF	故障 3 中断标志 请参考 FLT0IF 描述。
2	FLT2IF	故障 2 中断标志 请参考 FLT0IF 描述。
1	FLT1IF	故障 1 中断标志 请参考 FLT0IF 描述。
0	FLT0IF	故障 0 中断标志 故障 0 发生时，此标志由硬件置 1。可以通过软件写 1 清零。 0：故障 0 中断未发生 1：故障 0 中断发生

HRTIMER 中断标志清除寄存器 (HRTIMER_INTC)

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							FLT7IFC	FLT6IFC	FLT5IFC	SYSFLTIFC	FLT4IFC	FLT3IFC	FLT2IFC	FLT1IFC	FLT0IFC
							w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:18	保留	必须保持复位值
17	BMPERIFC	突发模式周期中断标志清除 该位软件写 1 可以清零 HRTIMER_INTF 寄存器中的 BMPERIF 位。 0：无影响 1：突发模式周期中断标志清除
16	DLLCALIFC	DLL 校准完成中断标志清除 该位软件写 1 可以清零 HRTIMER_INTF 寄存器中的 DLLCALIF 位。 0：无影响 1：DLL 校准完成中断标志清除

15:9	保留	必须保持复位值
8	FLT7IFC	故障 7 中断标志清除 请参考 FLT0IF 描述。
7	FLT6IFC	故障 6 中断标志清除 请参考 FLT0IF 描述。
6	FLT5IFC	故障 5 中断标志清除 请参考 FLT0IF 描述。
5	SYSFLTIFC	系统故障中断标志清除 该位软件写 1 可以清零 HRTIMER_INTF 寄存器中的 SYSFLTIF 位。 0：无影响 1：系统故障中断标志清除
4	FLT4IFC	故障 4 中断标志清除 请参考 FLT0IF 描述。
3	FLT3IFC	故障 3 中断标志清除 请参考 FLT0IF 描述。
2	FLT2IFC	故障 2 中断标志清除 请参考 FLT0IF 描述。
1	FLT1IFC	故障 1 中断标志清除 请参考 FLT0IF 描述。
0	FLT0IFC	故障 0 中断标志清除 该位软件写 1 可以清零 HRTIMER_INTF 寄存器中的 FLT0IF 位。 0：无影响 1：故障 0 中断标志清除

HRTIMER 中断使能寄存器 (HRTIMER_INTEN)

地址偏移：0x10

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

保留	FLT7IE	FLT6IE	FLT5IE	SYSFLTIE	FLT4IE	FLT3IE	FLT2IE	FLT1IE	FLT0IE
	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:18	保留	必须保持复位值
17	BMPERIE	突发模式周期中断使能 0：禁能 1：使能
16	DLLCALIE	DLL 校准完成中断使能 0：禁能 1：使能
15:9	保留	必须保持复位值
8	FLT7IE	故障 7 中断使能 请参考 FLT0IE 描述。
7	FLT6IE	故障 6 中断使能 请参考 FLT0IE 描述。
6	FLT5IE	故障 5 中断使能 请参考 FLT0IE 描述。
5	SYSFLTIE	系统故障中断使能 0：禁能 1：使能
4	FLT4IE	故障 4 中断使能 请参考 FLT0IE 描述。
3	FLT3IE	故障 3 中断使能 请参考 FLT0IE 描述。
2	FLT2IE	故障 2 中断使能 请参考 FLT0IE 描述。
1	FLT1IE	故障 1 中断使能 请参考 FLT0IE 描述。
0	FLT0IE	故障 0 中断使能 0：禁能 1：使能

HRTIMER 通道输出使能寄存器 (HRTIMER_CHOUTEN)

地址偏移：0x14

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST7CH1EN	ST7CH0EN	ST6CH1EN	ST6CH0EN	ST5CH1EN	ST5CH0EN	ST4CH1EN	ST4CH0EN	ST3CH1EN	ST3CH0EN	ST2CH1EN	ST2CH0EN	ST1CH1EN	ST1CH0EN	ST0CH1EN	ST0CH0EN
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	ST7CH1EN	Slave_TIMER7 通道 1 输出（ST7CH1_O）使能 请参考 ST0CH0EN 描述。 **注意：** 禁能状态对应空闲和故障状态，由 HRTIMER_CHOUTDISF 寄存器中的 ST7CH1DISF 位配置。
14	ST7CH0EN	Slave_TIMER7 通道 0 输出（ST7CH0_O）使能 请参考 ST0CH0EN 描述。 **注意：** 禁能状态对应空闲和故障状态，由 HRTIMER_CHOUTDISF 寄存器中的 ST7CH0DISF 位配置。
13	ST6CH1EN	Slave_TIMER6 通道 1 输出（ST6CH1_O）使能 请参考 ST0CH0EN 描述。 **注意：** 禁能状态对应空闲和故障状态，由 HRTIMER_CHOUTDISF 寄存器中的 ST7CH1DISF 位配置。
12	ST6CH0EN	Slave_TIMER6 通道 0 输出（ST6CH0_O）使能 请参考 ST0CH0EN 描述。 **注意：** 禁能状态对应空闲和故障状态，由 HRTIMER_CHOUTDISF 寄存器中的 ST6CH0DISF 位配置。
11	ST5CH1EN	Slave_TIMER5 通道 1 输出（ST5CH1_O）使能 请参考 ST0CH0EN 描述。 **注意：** 禁能状态对应空闲和故障状态，由 HRTIMER_CHOUTDISF 寄存器中的 ST7CH1DISF 位配置。
10	ST5CH0EN	Slave_TIMER5 通道 0 输出（ST5CH0_O）使能 请参考 ST0CH0EN 描述。

		注意：禁能状态对应空闲和故障状态，由 HRTIMER_CHOUTDISF 寄存器中的 ST5CH0DISF 位配置。
9	ST4CH1EN	Slave_TIMER4 通道 1 输出（ST4CH1_O）使能 请参考 ST0CH0EN 描述。 **注意：**禁能状态对应空闲和故障状态，由 HRTIMER_CHOUTDISF 寄存器中的 ST4CH1DISF 位配置。
8	ST4CH0EN	Slave_TIMER4 通道 0 输出（ST4CH0_O）使能 请参考 ST0CH0EN 描述。 **注意：**禁能状态对应空闲和故障状态，由 HRTIMER_CHOUTDISF 寄存器中的 ST4CH0DISF 位配置。
7	ST3CH1EN	Slave_TIMER3 通道 1 输出（ST3CH1_O）使能 请参考 ST0CH0EN 描述。 **注意：**禁能状态对应空闲和故障状态，由 HRTIMER_CHOUTDISF 寄存器中的 ST3CH1DISF 位配置。
6	ST3CH0EN	Slave_TIMER3 通道 0 输出（ST3CH0_O）使能 请参考 ST0CH0EN 描述。 **注意：**禁能状态对应空闲和故障状态，由 HRTIMER_CHOUTDISF 寄存器中的 ST3CH0DISF 位配置。
5	ST2CH1EN	Slave_TIMER2 通道 1 输出（ST2CH1_O）使能 请参考 ST0CH0EN 描述。 **注意：**禁能状态对应空闲和故障状态，由 HRTIMER_CHOUTDISF 寄存器中的 ST2CH1DISF 位配置。
4	ST2CH0EN	Slave_TIMER2 通道 0 输出（ST2CH0_O）使能 请参考 ST0CH0EN 描述。 **注意：**禁能状态对应空闲和故障状态，由 HRTIMER_CHOUTDISF 寄存器中的 ST2CH0DISF 位配置。
3	ST1CH1EN	Slave_TIMER1 通道 1 输出（ST1CH1_O）使能 请参考 ST0CH0EN 描述。 **注意：**禁能状态对应空闲和故障状态，由 HRTIMER_CHOUTDISF 寄存器中的 ST1CH1DISF 位配置。
2	ST1CH0EN	Slave_TIMER1 通道 0 输出（ST1CH0_O）使能 请参考 ST0CH0EN 描述。 **注意：**禁能状态对应空闲和故障状态，由 HRTIMER_CHOUTDISF 寄存器中的 ST1CH0DISF 位配置。
1	ST0CH1EN	Slave_TIMER0 通道 1 输出（ST0CH1_O）使能 请参考 ST0CH0EN 描述。

注意：禁能状态对应空闲和故障状态，由 HRTIMER_CHOUTDISF 寄存器中的 ST0CH1DISF 位配置。

0	ST0CH0EN	Slave_TIMER0 通道 0 输出（ST0CH0_O）使能 该位写 1 使能输出，写 0 无影响。 故障输入有效时，该位由硬件异步清除。 读该位内容，将返回输出使能或禁能的状态。 0：Slave_TIMER0 通道 0 输出 ST0CH0_O 禁能。输出处于故障状态或空闲状态。 1：Slave_TIMER0 通道 0 输出 ST0CH0_O 使能。 **注意：**禁能状态对应空闲和故障状态，由 HRTIMER_CHOUTDISF 寄存器中的 ST0CH0DISF 位配置。

HRTIMER 通道输出禁能寄存器 (HRTIMER_CHOUTDIS)

地址偏移：0x18

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST7CH1DIS	ST7CH0DIS	ST6CH1DIS	ST6CH0DIS	ST5CH1DIS	ST5CH0DIS	ST4CH1DIS	ST4CH0DIS	ST3CH1DIS	ST3CH0DIS	ST2CH1DIS	ST2CH0DIS	ST1CH1DIS	ST1CH0DIS	ST0CH1DIS	ST0CH0DIS
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	ST7CH1DIS	Slave_TIMER7 通道 1 输出（ST7CH1_O）禁能 请参考 ST0CH0DIS 描述。
14	ST7CH0DIS	Slave_TIMER7 通道 0 输出（ST7CH0_O）禁能 请参考 ST0CH0DIS 描述。
13	ST6CH1DIS	Slave_TIMER6 通道 1 输出（ST6CH1_O）禁能 请参考 ST0CH0DIS 描述。
12	ST6CH0DIS	Slave_TIMER6 通道 0 输出（ST6CH0_O）禁能 请参考 ST0CH0DIS 描述。
11	ST5CH1DIS	Slave_TIMER5 通道 1 输出（ST5CH1_O）禁能 请参考 ST0CH0DIS 描述。

10	ST5CH0DIS	Slave_TIMER5 通道 0 输出（ST5CH0_O）禁能 请参考 ST0CH0DIS 描述。
9	ST4CH1DIS	Slave_TIMER4 通道 1 输出（ST4CH1_O）禁能 请参考 ST0CH0DIS 描述。
8	ST4CH0DIS	Slave_TIMER4 通道 0 输出（ST4CH0_O）禁能 请参考 ST0CH0DIS 描述。
7	ST3CH1DIS	Slave_TIMER3 通道 1 输出（ST3CH1_O）禁能 请参考 ST0CH0DIS 描述。
6	ST3CH0DIS	Slave_TIMER3 通道 0 输出（ST3CH0_O）禁能 请参考 ST0CH0DIS 描述。
5	ST2CH1DIS	Slave_TIMER2 通道 1 输出（ST2CH1_O）禁能 请参考 ST0CH0DIS 描述。
4	ST2CH0DIS	Slave_TIMER2 通道 0 输出（ST2CH0_O）禁能 请参考 ST0CH0DIS 描述。
3	ST1CH1DIS	Slave_TIMER1 通道 1 输出（ST1CH1_O）禁能 请参考 ST0CH0DIS 描述。
2	ST1CH0DIS	Slave_TIMER1 通道 0 输出（ST1CH0_O）禁能 请参考 ST0CH0DIS 描述。
1	ST0CH1DIS	Slave_TIMER0 通道 1 输出（ST0CH1_O）禁能 请参考 ST0CH0DIS 描述。
0	ST0CH0DIS	Slave_TIMER1 通道 0 输出（ST1CH0_O）禁能 该位写 1 禁能输出，通道 0 进入空闲状态。写 0 无影响。 0：无影响。 1：Slave_TIMER1 通道 0 输出 ST0CH0_O 禁能。输出从故障状态或运行状态进入空闲状态。

HRTIMER 通道输出禁能标志寄存器 (HRTIMER_CHOUTDISF)

地址偏移：0x1C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

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

ST7CH1DISF	ST7CH0DISF	ST6CH1DISF	ST6CH0DISF	ST5CH1DISF	ST5CH0DISF	ST4CH1DISF	ST4CH0DISF	ST3CH1DISF	ST3CH0DISF	ST2CH1DISF	ST2CH0DISF	ST1CH1DISF	ST1CH0DISF	ST0CH1DISF	ST0CH0DISF
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	ST7CH1DISF	Slave_TIMER7 通道 1 输出（ST7CH1_O）禁能标志 请参考 ST0CH0DISF 描述。
14	ST7CH0DISF	Slave_TIMER7 通道 0 输出（ST7CH0_O）禁能标志 请参考 ST0CH0DISF 描述。
13	ST6CH1DISF	Slave_TIMER6 通道 1 输出（ST6CH1_O）禁能标志 请参考 ST0CH0DISF 描述。
12	ST6CH0DISF	Slave_TIMER6 通道 0 输出（ST6CH0_O）禁能标志 请参考 ST0CH0DISF 描述。
11	ST5CH1DISF	Slave_TIMER5 通道 1 输出（ST5CH1_O）禁能标志 请参考 ST0CH0DISF 描述。
10	ST5CH0DISF	Slave_TIMER5 通道 0 输出（ST5CH0_O）禁能标志 请参考 ST0CH0DISF 描述。
9	ST4CH1DISF	Slave_TIMER4 通道 1 输出（ST4CH1_O）禁能标志 请参考 ST0CH0DISF 描述。
8	ST4CH0DISF	Slave_TIMER4 通道 0 输出（ST4CH0_O）禁能标志 请参考 ST0CH0DISF 描述。
7	ST3CH1DISF	Slave_TIMER3 通道 1 输出（ST3CH1_O）禁能标志 请参考 ST0CH0DISF 描述。
6	ST3CH0DISF	Slave_TIMER3 通道 0 输出（ST3CH0_O）禁能标志 请参考 ST0CH0DISF 描述。
5	ST2CH1DISF	Slave_TIMER2 通道 1 输出（ST2CH1_O）禁能标志 请参考 ST0CH0DISF 描述。
4	ST2CH0DISF	Slave_TIMER2 通道 0 输出（ST2CH0_O）禁能标志 请参考 ST0CH0DISF 描述。
3	ST1CH1DISF	Slave_TIMER1 通道 1 输出（ST1CH1_O）禁能标志 请参考 ST0CH0DISF 描述。
2	ST1CH0DISF	Slave_TIMER1 通道 0 输出（ST1CH0_O）禁能标志

请参考 ST0CH0DISF 描述。

1	ST0CH1DISF	Slave_TIMER0 通道 1 输出（ST0CH1_O）禁能标志 请参考 ST0CH0DISF 描述。
0	ST0CH0DISF	Slave_TIMER0 通道 0 输出（ST0CH0_O）禁能标志 读该位内容，将返回通道 0 输出禁能状态。输出使能时无效。 0：空闲状态下，Slave_TIMER0 通道 0 输出 ST0CH0_O 禁能。 1：故障状态下，Slave_TIMER0 通道 0 输出 ST0CH0_O 禁能。

HRTIMER 突发模式控制寄存器 (HRTIMER_BMCTL)

地址偏移：0x20

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BMOPTF	保留						BMST7	BMST6	BMST5	BMST4	BMST3	BMST2	BMST1	BMST0	BMMT
rc_w0							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					BMSE	BMPSC[3:0]				BMCLKS[3:0]				BMCTN	BMEN
					rw	rw				rw				rw	rw

位/位域	名称	描述
31	BMOPTF	突发模式运行标志 突发模式正在运行时，该标志位由硬件置位。该位写 0 将停止突发模式。 0：常规运行，突发模式不起作用。 1：突发模式正在进行。
30:25	保留	必须保持复位值
24	BMST7	Slave_TIMER7 突发模式 0：Slave_TIMER7 计数器时钟（HRTIMER_PSCCK）保持，且计数器正常运行。 1：Slave_TIMER7 计数器时钟（HRTIMER_PSCCK）停止，且计数器复位。 **注意：** （1）突发模式使能时，无法更改此位。 （2）当均衡空闲模式有效（HRTIMER_STxCHOCTL 寄存器中的 DLYISCH [2:0] = 3'bx11）时，不能将此位置位。
23	BMST6	Slave_TIMER6 突发模式 0：Slave_TIMER6 计数器时钟（HRTIMER_PSCCK）保持，且计数器正常运行。 1：Slave_TIMER6 计数器时钟（HRTIMER_PSCCK）停止，且计数器复位。 **注意：**

（1）突发模式使能时，无法更改此位。

（2）当均衡空闲模式有效（HRTIMER_STxCHOCTL 寄存器中的 DLYISCH [2:0] = 3’bx11）时，不能将此位置位。

22 BMST5

Slave_TIMER5 突发模式

0：Slave_TIMER5 计数器时钟（HRTIMER_PSCCK）保持，且计数器正常运行。

1：Slave_TIMER5 计数器时钟（HRTIMER_PSCCK）停止，且计数器复位。

注意：

（1）突发模式使能时，无法更改此位。

（2）当均衡空闲模式有效（HRTIMER_STxCHOCTL 寄存器中的 DLYISCH [2:0] = 3’bx11）时，不能将此位置位。

21 BMST4

Slave_TIMER4 突发模式

0：Slave_TIMER4 计数器时钟（HRTIMER_PSCCK）保持，且计数器正常运行。

1：Slave_TIMER4 计数器时钟（HRTIMER_PSCCK）停止，且计数器复位。

注意：

（1）突发模式使能时，无法更改此位。

（2）当均衡空闲模式有效（HRTIMER_STxCHOCTL 寄存器中的 DLYISCH [2:0] = 3’bx11）时，不能将此位置位。

20 BMST3

Slave_TIMER3 突发模式

0：Slave_TIMER3 计数器时钟（HRTIMER_PSCCK）保持，且计数器正常运行。

1：Slave_TIMER3 计数器时钟（HRTIMER_PSCCK）停止，且计数器复位。

注意：

（1）突发模式使能时，无法更改此位。

（2）当均衡空闲模式有效（HRTIMER_STxCHOCTL 寄存器中的 DLYISCH [2:0] = 3’bx11）时，不能将此位置位。

19 BMST2

Slave_TIMER2 突发模式

0：Slave_TIMER2 计数器时钟（HRTIMER_PSCCK）保持，且计数器正常运行。

1：Slave_TIMER2 计数器时钟（HRTIMER_PSCCK）停止，且计数器复位。

注意：

（1）突发模式使能时，无法更改此位。

（2）当均衡空闲模式有效（HRTIMER_STxCHOCTL 寄存器中的 DLYISCH [2:0] = 3’bx11）时，不能将此位置位。

18 BMST1

Slave_TIMER1 突发模式

0：Slave_TIMER1 计数器时钟（HRTIMER_PSCCK）保持，且计数器正常运行。

1：Slave_TIMER1 计数器时钟（HRTIMER_PSCCK）停止，且计数器复位。

注意：

（1）突发模式使能时，无法更改此位。

（2）当均衡空闲模式有效（HRTIMER_STxCHOCTL 寄存器中的 DLYISCH [2:0] =

		3’bx11）时，不能将此位置位。
17	BMST0	Slave_TIMER0 突发模式 0：Slave_TIMER0 计数器时钟（HRTIMER_PSCCK）保持，且计数器正常运行。 1：Slave_TIMER0 计数器时钟（HRTIMER_PSCCK）停止，且计数器复位。 **注意：** （1）突发模式使能时，无法更改此位。 （2）当均衡空闲模式有效（HRTIMER_STxCHOCTL 寄存器中的 DLYISCH [2:0] = 3’bx11）时，不能将此位置位。
16	BMMT	Master_TIMER 突发模式 0：Master_TIMER 计数器时钟（HRTIMER_PSCCK）保持，且计数器正常运行。 1：Master_TIMER 计数器时钟（HRTIMER_PSCCK）停止，且计数器复位。
15:11	保留	必须保持复位值
10	BMSE	突发模式影子寄存器使能 0：HRTIMER_BMCMPV 和 HRTIMER_BMCAR 寄存器的影子寄存器禁能。 1：HRTIMER_BMCMPV 和 HRTIMER_BMCAR 寄存器的影子寄存器使能。 **注意：**突发模式使能时不能更改此位。
9:6	BMPSC[3:0]	突发模式时钟分频 该位域可以由软件配置，当 HRTIMER_BMCTL 寄存器中的 BMCLKS [3:0] = 4’b1010 时，确定高分辨率时钟（HRTIMER_HPCK）和突发模式计数器时钟（HRTIMER_BMCNTCK）的分频比。 $f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/2^{BMPSC[3:0]}$ 0000：$f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}$ 0001：$f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/2$ 0010：$f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/4$ 0011：$f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/8$ 0100：$f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/16$ 0101：$f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/32$ 0110：$f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/64$ 0111：$f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/128$ 1000：$f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/256$ 1001：$f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/512$ 1010：$f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/1024$ 1011：$f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/2048$ 1100：$f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/4096$ 1101：$f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/8192$ 1110：$f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/16384$ 1111：$f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/32768$

注意：突发模式使能时，该位不能修改。

5:2 BMCLKS[3:0] 突发模式时钟源

该位域用于配置突发模式计数器的时钟源。

0000：Master_TIMER 计数器复位/翻转事件。

0001：Slave_TIMER0 计数器复位/翻转事件。

0010：Slave_TIMER1 计数器复位/翻转事件。

0011：Slave_TIMER2 计数器复位/翻转事件。

0100：Slave_TIMER3 计数器复位/翻转事件。

0101：Slave_TIMER4 计数器复位/翻转事件。

0110：芯片内部信号 0，BMCLK0。

0111：芯片内部信号 1，BMCLK1。

1000：芯片内部信号 2，BMCLK2。

1001：芯片内部信号 3，BMCLK3。

1010：$F_{HRTIMER_CK}$ 时钟预分频（根据 BMPRSC [3:0]设置）。

1011：Slave_TIMER5 计数器复位/翻转事件。

1100：Slave_TIMER6 计数器复位/翻转事件。

1101：Slave_TIMER7 计数器复位/翻转事件。

其他值保留。

注意：

（1）突发模式使能时，无法更改此位

（2）BMCLKy（y = 0..3）：请参考*表 25-19. 突发模式的芯片内部信号*。

1 BMCTN 突发模式下的连续模式

0：单脉冲模式。BM 计数器达到 HRTIMER_BMCAR 值时，由硬件停止。

1：连续模式。BM 计数器达到 HRTIMER_BMCAR 值时，翻转到零并连续计数。

0 BMEN 突发模式使能

该位置 1 时，突发模式控制器已准备好接收突发模式启动触发。该位写 0 将终止突发模式。

0：突发模式禁能。

1：突发模式使能。

HRTIMER 突发模式启动触发寄存器 (HRTIMER_BMSTRG)

地址偏移：0x24

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CISGN	EXEV7	EXEV6	ST3EXEV7	ST0EXEV6	ST4CMP1	ST4CMP0	ST4REP	ST4RST	ST3CMP1	ST3CMP0	ST3REP	ST3RST	ST2CMP1	ST2CMP0	ST2REP

rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST2RST	ST1CMP1	ST1CMP0	ST1REP	ST1RST	ST0CMP1	ST0CMP0	ST0REP	ST0RST	MTCMP3	MTCMP2	MTCMP1	MTCMP0	MTREP	MTRST	SWTRG
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	CISGN	芯片内部信号触发突发模式 芯片内部信号（TIMER6_TRGO）启动突发模式。 0：对突发模式无影响。 1：芯片内部信号启动突发模式。
30	EXEV7	外部事件 7 触发突发模式 外部事件 7 启动突发模式。 0：对突发模式无影响。 1：外部事件 7 启动突发模式。
29	EXEV6	外部事件 6 触发突发模式 外部事件 6 启动突发模式。 0：对突发模式无影响。 1：外部事件 6 启动突发模式。
28	ST3EXEV7	外部事件 7 之后的 Slave_TIMER3 周期事件触发突发模式 0：对突发模式无影响。 1：外部事件 7 之后的 Slave_TIMER3 周期事件启动突发模式。
27	ST0EXEV6	外部事件 6 之后的 Slave_TIMER0 周期事件触发突发模式 0：对突发模式无影响。 1：外部事件 6 之后的 Slave_TIMER0 周期事件启动突发模式。
26	ST4CMP1	Slave_TIMER4 比较 1 事件触发突发模式。 请参考 MTCMP1 模式。
25	ST4CMP0	Slave_TIMER4 比较 0 事件触发突发模式。 请参考 MTCMP0 模式。
24	ST4REP	Slave_TIMER4 重复事件触发突发模式。 请参考 MTREP 模式。
23	ST4RST	Slave_TIMER4 复位事件触发突发模式。 请参考 MTRST 模式。
22	ST3CMP1	Slave_TIMER3 比较 1 事件触发突发模式。 请参考 MTCMP1 模式。

21	ST3CMP0	Slave_TIMER3 比较 0 事件触发突发模式。 请参考 MTCMP0 模式。
20	ST3REP	Slave_TIMER3 重复事件触发突发模式。 请参考 MTREP 模式。
19	ST3RST	Slave_TIMER3 复位事件触发突发模式。 请参考 MTRST 模式。
18	ST2CMP1	Slave_TIMER2 比较 1 事件触发突发模式。 请参考 MTCMP1 模式。
17	ST2CMP0	Slave_TIMER2 比较 0 事件触发突发模式。 请参考 MTCMP0 模式。
16	ST2REP	Slave_TIMER2 重复事件触发突发模式。 请参考 MTREP 模式。
15	ST2RST	Slave_TIMER2 复位事件触发突发模式。 请参考 MTRST 模式。
14	ST1CMP1	Slave_TIMER1 比较 1 事件触发突发模式。 请参考 MTCMP1 模式。
13	ST1CMP0	Slave_TIMER1 比较 0 事件触发突发模式。 请参考 MTCMP0 模式。
12	ST1REP	Slave_TIMER1 重复事件触发突发模式。 请参考 MTREP 模式。
11	ST1RST	Slave_TIMER1 复位事件触发突发模式。 请参考 MTRST 模式。
10	ST0CMP1	Slave_TIMER0 比较 1 事件触发突发模式。 请参考 MTCMP1 模式。
9	ST0CMP0	Slave_TIMER0 比较 0 事件触发突发模式。 请参考 MTCMP0 模式。
8	ST0REP	Slave_TIMER0 重复事件触发突发模式。 请参考 MTREP 模式。
7	ST0RST	Slave_TIMER0 复位事件触发突发模式。 请参考 MTRST 模式。
6	MTCMP3	Master_TIMER 比较 3 事件触发突发模式。 请参考 MTCMP0 模式。

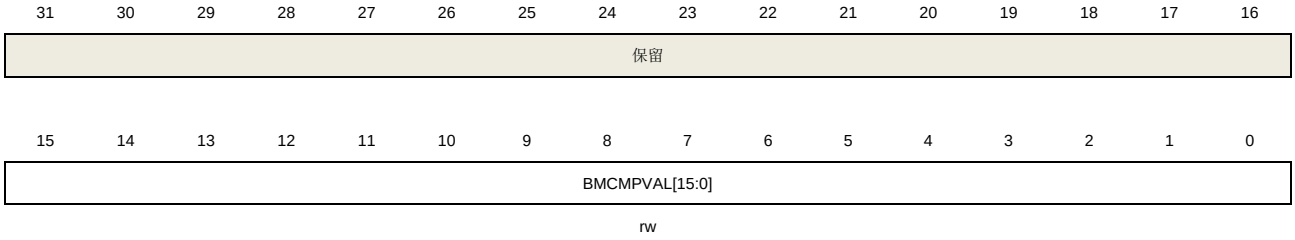
5	MTCMP2	Master_TIMER 比较 2 事件触发突发模式。 请参考 MTCMP0 模式。
4	MTCMP1	Master_TIMER 比较 1 事件触发突发模式。 请参考 MTCMP0 模式。
3	MTCMP0	Master_TIMER 比较 0 事件触发突发模式。 0：对突发模式无影响。 1：Master_TIMER 比较 0 事件启动突发模式。
2	MTREP	Master_TIMER 重复事件触发突发模式。 0：对突发模式无影响。 1：Master_TIMER 重复事件启动突发模式。
1	MTRST	Master_TIMER 复位事件触发突发模式。 0：对突发模式无影响。 1：Master_TIMER 复位事件启动突发模式。
0	SWTRG	软件触发突发模式 该位由软件置 1，硬件自动清零。该位置 1 时，触发突发模式。如果突发模式未使能，则该位无效（HRTIMER_BMCTL 寄存器中的 BMEN 位复位）。 0：对突发模式无影响。 1：软件触发启动突发模式。 **注意：**如果突发模式未使能（BMEN 位复位），则此位无效。

HRTIMER 突发模式比较值寄存器 (HRTIMER_BMCMPV)

地址偏移：0x28

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。



位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	BMCMPVAL[15:0]	突发模式比较值 该位域包含了与BM计数器进行比较的值，并定义了空闲的持续时间。 该寄存器有影子寄存器，可以进行预装载。

注意：当预分频系数为0，$f_{HRTIMER_CK}$时钟直接作为突发模式时钟源（BMCLKS [3:0] = 4'b1010和BMPSC [3:0] = 4）时，BMCMPVAL[15:0]不能设置为0x0000。

HRTIMER 突发模式计数器自动重载寄存器 (HRTIMER_BMCAR)

地址偏移：0x2C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	BMCARL[15:0]	突发模式计数器自动重载值 该位域配置了 BM 计数器的自动重载值，并定义了突发模式的周期，该周期是空闲状态和运行状态持续的时间之和。 该寄存器具有影子寄存器，可以进行预加载。 **注意：**突发模式使能时，该位域不能为零。

HRTIMER 外部事件配置寄存器 0 (HRTIMER_EXEVCFG0)

地址偏移：0x30

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		EXEV4FAST	EXEV4EG[1:0]		EXEV4P	EXEV4SRC[1:0]		EXEV3FAST	EXEV3EG[1:0]		EXEV3P	EXEV3SRC[1:0]		EXEV2FAST	EXEV2EG[1]
		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
EXEV2EG[0]	EXEV2P	EXEV2SRC[1:0]		EXEV1FAST	EXEV1EG[1:0]		EXEV1P	EXEV1SRC[1:0]		EXEV0FAST	EXEV0EG[1:0]		EXEV0P	EXEV0SRC[1:0]	
rw	rw	rw		rw	rw		rw	rw			rw		rw	rw	

位/位域	名称	描述

31:30	保留	必须保持复位值
29	EXEV4FAST	外部事件 4 快速模式 请参考 EXEV0FAST 说明。
28:27	EXEV4EG[1:0]	外部事件 4 有效沿 请参考 EXEV0EG [1:0]说明。
26	EXEV4P	外部事件 4 极性 请参考 EXEV0P 说明。
25:24	EXEV4SRC[1:0]	外部事件 4 的源 请参考 EXEV0SRC[1:0]说明。
23	EXEV3FAST	外部事件 3 快速模式 请参考 EXEV0FAST 说明。
22:21	EXEV3EG[1:0]	外部事件 3 有效沿 请参考 EXEV0EG [1:0]说明。
20	EXEV3P	外部事件 3 极性 请参考 EXEV0P 说明。
19:18	EXEV3SRC[1:0]	外部事件 3 的源 请参考 EXEV0SRC[1:0]说明。
17	EXEV2FAST	外部事件 2 快速模式 请参考 EXEV0FAST 说明。
16:15	EXEV2EG[1:0]	外部事件 2 有效沿 请参考 EXEV0EG [1:0]说明。
14	EXEV2P	外部事件 2 极性 请参考 EXEV0P 说明。
13:12	EXEV2SRC[1:0]	外部事件 2 的源 请参考 EXEV0SRC[1:0]说明。
11	EXEV1FAST	外部事件 1 快速模式 请参考 EXEV0FAST 说明。
10:9	EXEV1EG[1:0]	外部事件 1 有效沿 请参考 EXEV0EG [1:0]说明。
8	EXEV1P	外部事件 1 极性 请参考 EXEV0P 说明。
7:6	EXEV1SRC[1:0]	外部事件 1 的源

请参考 EXEV0SRC[1:0]说明。

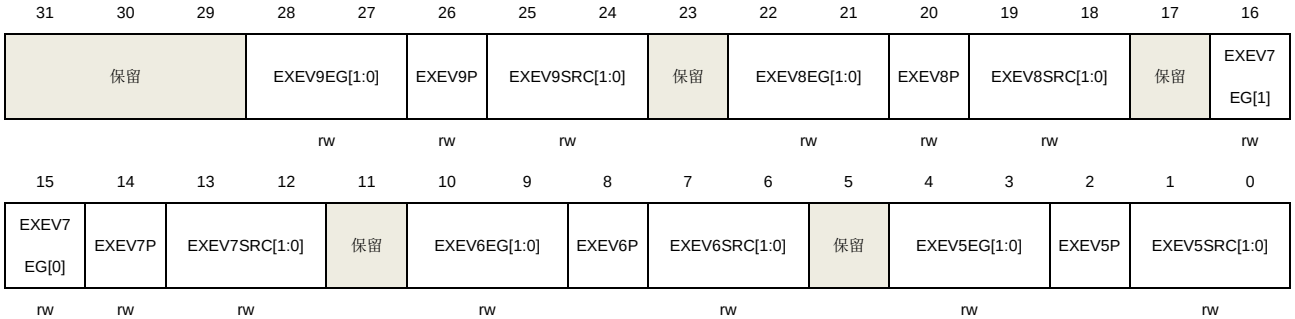
5	EXEV0FAST	外部事件 0 快速模式 0：在输出之前，外部事件 0 通过 HRTIMER 逻辑进行重新同步，引入了与 HRTIMER 时钟相关的延迟。 1：外部事件 0 在输出上以异步方式运行（低延迟模式）。
4:3	EXEV0EG[1:0]	外部事件 0 有效沿 该位域配置了外部事件 0 的有效沿。 00：电平有效。有效电平由 EXEV0P 位定义。 01：上升沿有效，EXEV0P 位无效。 10：下降沿有效，EXEV0P 位无效 11：上升沿和下降沿均有效，EXEV0P 位无效。
2	EXEV0P	外部事件 0 的极性 当 EXEV0EG [1:0] = 2'b00 时，该位确定了外部事件 0 的有效电平。 0：外部事件 0 高电平有效。 1：外部事件 0 低电平有效。 **注意：**一旦 Slave_TIMERx 使能，就不能更改该位。
1:0	EXEV0SRC[1:0]	外部事件 0 的源 00：外部事件 0 的源为 EXEV0SRC 0。 01：外部事件 0 的源为 EXEV0SRC 1。 10：外部事件 0 的源为 EXEV0SRC 2。 11：外部事件 0 的源为 EXEV0SRC 3。 **注意：**一旦 Slave_TIMERx 使能，就不能更改该位。

HRTIMER 外部事件配置寄存器 1 (HRTIMER_EXEVCFG1)

地址偏移：0x34

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。



位/位域	名称	描述

31:29	保留	必须保持复位值
28:27	EXEV9EG[1:0]	外部事件 9 有效沿 请参考 HRTIMER_EXEVCFG0 寄存器中的 EXEV0EG [1:0]位域说明。
26	EXEV9P	外部事件 9 极性 请参考 HRTIMER_EXEVCFG0 寄存器中的 EXEV0P 位说明。
25:24	EXEV9SRC[1:0]	外部事件 9 的源 请参考 HRTIMER_EXEVCFG0 寄存器中的 EXEV0SRC[1:0]位域说明。
23	保留	必须保持复位值
22:21	EXEV8EG[1:0]	外部事件 8 有效沿 请参考 HRTIMER_EXEVCFG0 寄存器中的 EXEV0EG [1:0]位域说明。
20	EXEV8P	外部事件 8 极性 请参考 HRTIMER_EXEVCFG0 寄存器中的 EXEV0P 位说明。
19:18	EXEV8SRC[1:0]	外部事件 8 的源 请参考 HRTIMER_EXEVCFG0 寄存器中的 EXEV0SRC[1:0]位域说明。
17	保留	必须保持复位值
16:15	EXEV7EG[1:0]	外部事件 7 有效沿 请参考 HRTIMER_EXEVCFG0 寄存器中的 EXEV0EG [1:0]位域说明。
14	EXEV7P	外部事件 7 极性 请参考 HRTIMER_EXEVCFG0 寄存器中的 EXEV0P 位说明。
13:12	EXEV7SRC[1:0]	外部事件 7 的源 请参考 HRTIMER_EXEVCFG0 寄存器中的 EXEV0SRC[1:0]位域说明。
11	保留	必须保持复位值
10:9	EXEV6EG[1:0]	外部事件 6 有效沿 请参考 HRTIMER_EXEVCFG0 寄存器中的 EXEV0EG [1:0]位域说明。
8	EXEV6P	外部事件 6 极性 请参考 HRTIMER_EXEVCFG0 寄存器中的 EXEV0P 位说明。
7:6	EXEV6SRC[1:0]	外部事件 6 的源 请参考 HRTIMER_EXEVCFG0 寄存器中的 EXEV0SRC[1:0]位域说明。
5	保留	必须保持复位值
4:3	EXEV5EG[1:0]	外部事件 5 有效沿 请参考 HRTIMER_EXEVCFG0 寄存器中的 EXEV0EG [1:0]位域说明。

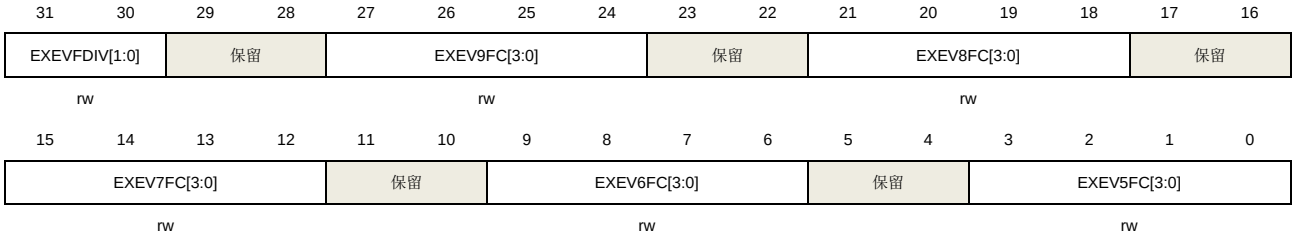
2	EXEV5P	外部事件 5 极性 请参考 HRTIMER_EXEVCFG0 寄存器中的 EXEV0P 位说明。
1:0	EXEV5SRC[1:0]	外部事件 5 的源 请参考 HRTIMER_EXEVCFG0 寄存器中的 EXEV0SRC[1:0]位域说明。

HRTIMER 外部事件数字滤波控制寄存器 (HRTIMER_EXEVDFCTL)

地址偏移：0x38

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。



位/位域	名称	描述
31:30	EXEVFDIV[1:0]	外部事件数字滤波器时钟分频 该位域由软件配置，确定 HRTIMER 时钟（HRTIMER_CK）和外部事件数字滤波器时钟（HRTIMER_EXEVFCK）之间的分频比。 $f_{HRTIMER_EXEVFCK} = f_{HRTIMER_CK}/2^{EXEVFDIV[2:0]}$。 00：$f_{HRTIMER_EXEVFCK} = f_{HRTIMER_CK}$。 01：$f_{HRTIMER_EXEVFCK} = f_{HRTIMER_CK}/2$。 10：$f_{HRTIMER_EXEVFCK} = f_{HRTIMER_CK}/4$。 11：$f_{HRTIMER_EXEVFCK} = f_{HRTIMER_CK}/8$。
29:28	保留	必须保持复位值
27:24	EXEV9FC[3:0]	外部事件 9 滤波控制 请参考 EXEV5FC [3:0]说明。
23:22	保留	必须保持复位值
21:18	EXEV8FC[3:0]	外部事件 8 滤波控制 请参考 EXEV5FC [3:0]说明。
17:16	保留	必须保持复位值
15:12	EXEV7FC[3:0]	外部事件 7 滤波控制 请参考 EXEV5FC [3:0]说明。

11:10	保留	必须保持复位值
9:6	EXEV6FC[3:0]	外部事件 5 滤波控制 请参考 EXEV5FC [3:0]说明。
5:4	保留	必须保持复位值
3:0	EXEV5FC[3:0]	外部事件 5 滤波控制 在数字滤波器中使用事件计数器，N 次输入事件后，输出才会发生转变。该位域用于确定采样外部事件的频率（f_{SAMP}）以及应用于外部事件的数字滤波器的长度。 0000：无滤波。 0001：$f_{SAMP} = f_{HRTIMER_CK}$，N=2。 0010：$f_{SAMP} = f_{HRTIMER_CK}$，N=4。 0011：$f_{SAMP} = f_{HRTIMER_CK}$，N=8。 0100：$f_{SAMP} = f_{HRTIMER_EXEVFCK}$ /2，N=6。 0101：$f_{SAMP} = f_{HRTIMER_EXEVFCK}$ /2，N=8。 0110：$f_{SAMP} = f_{HRTIMER_EXEVFCK}$ /4，N=6。 0111：$f_{SAMP} = f_{HRTIMER_EXEVFCK}$ /4，N=8。 1000：$f_{SAMP} = f_{HRTIMER_EXEVFCK}$ /8，N=6。 1001：$f_{SAMP} = f_{HRTIMER_EXEVFCK}$ /8，N=8。 1010：$f_{SAMP} = f_{HRTIMER_EXEVFCK}$ /16，N=5。 1011：$f_{SAMP} = f_{HRTIMER_EXEVFCK}$ /16，N=6。 1100：$f_{SAMP} = f_{HRTIMER_EXEVFCK}$ /16，N=8。 1101：$f_{SAMP} = f_{HRTIMER_EXEVFCK}$ /32，N=5。 1110：$f_{SAMP} = f_{HRTIMER_EXEVFCK}$ /32，N=6。 1111：$f_{SAMP} = f_{HRTIMER_EXEVFCK}$ /32，N=8。

HRTIMER ADC 触发源 0 寄存器 (HRTIMER_ADCTRIGS0)

地址偏移：0x3C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TRG0ST4 PER	TRG0ST4 C3	TRG0ST4 C2	TRG0ST4 C1	TRG0ST3 PER	TRG0ST3 C3	TRG0ST3 C2	TRG0ST3 C1	TRG0ST2 PER	TRG0ST2 C3	TRG0ST2 C2	TRG0ST2 C1	TRG0ST1 RST	TRG0ST1 PER	TRG0ST1 C3	TRG0ST1 C2
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
TRG0ST1 C1	TRG0ST0 RST	TRG0ST0 PER	TRG0ST0 C3	TRG0ST0 C2	TRG0ST0 C1	TRG0EX EV4	TRG0EX EV3	TRG0EX EV2	TRG0EX EV1	TRG0EX EV0	TRG0MT PER	TRG0MT C3	TRG0MT C2	TRG0MT C1	TRG0MT C0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	TRG0ST4PER	Slave_TIMER4周期事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置周期事件生成ADC触发事件。 0：HRTIMER Slave_TIMER4周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER4周期事件生成ADC触发事件。
30	TRG0ST4C3	Slave_TIMER4比较3事件生成HRTIMER_ADCTRIG0上的ADC触发事件 请参考 TRG0ST4C1 描述。
29	TRG0ST4C2	Slave_TIMER4比较2事件生成HRTIMER_ADCTRIG0上的ADC触发事件 请参考 TRG0ST4C1 描述。
28	TRG0ST4C1	Slave_TIMER4比较1事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置比较1事件生成ADC触发事件。 0：HRTIMER Slave_TIMER4比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER4比较1事件生成ADC触发事件。
27	TRG0ST3PER	Slave_TIMER3周期事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置周期事件生成ADC触发事件。 0：HRTIMER Slave_TIMER3周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER3周期事件生成ADC触发事件。
26	TRG0ST3C3	Slave_TIMER3比较3事件生成HRTIMER_ADCTRIG0上的ADC触发事件 请参考 TRG0ST3C1 描述。
25	TRG0ST3C2	Slave_TIMER3比较2事件生成HRTIMER_ADCTRIG0上的ADC触发事件 请参考 TRG0ST3C1 描述。
24	TRG0ST3C1	Slave_TIMER3比较1事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置比较1事件生成ADC触发事件。 0：HRTIMER Slave_TIMER3比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER3比较1事件生成ADC触发事件。
23	TRG0ST2PER	Slave_TIMER2周期事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置周期事件生成ADC触发事件。 0：HRTIMER Slave_TIMER2周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER2周期事件生成ADC触发事件。
22	TRG0ST2C3	Slave_TIMER3比较3事件生成HRTIMER_ADCTRIG0上的ADC触发事件

		请参考 TRG0ST2C1 描述。
21	TRG0ST2C2	Slave_TIMER3比较2事件生成HRTIMER_ADCTRIG0 请参考 TRG0ST2C1 描述。
20	TRG0ST2C1	Slave_TIMER2比较1事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置比较1事件是否生成ADC触发事件。 0：HRTIMER Slave_TIMER2比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER2比较1事件生成ADC触发事件。
19	TRG0ST1RST	Slave_TIMER1复位事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置生成ADC触发事件。 0：HRTIMER Slave_TIMER1复位事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER1复位事件生成ADC触发事件。
18	TRG0ST1PER	Slave_TIMER1周期事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置周期事件生成ADC触发事件。 0：HRTIMER Slave_TIMER1周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER1周期事件生成ADC触发事件。
17	TRG0ST1C3	Slave_TIMER1比较3事件生成HRTIMER_ADCTRIG0 请参考 TRG0ST1C1 描述。
16	TRG0ST1C2	Slave_TIMER1比较2事件生成HRTIMER_ADCTRIG0 请参考 TRG0ST1C1 描述。
15	TRG0ST1C1	Slave_TIMER1比较1事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置比较1事件生成ADC触发事件。 0：HRTIMER Slave_TIMER1比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER1比较1事件生成ADC触发事件。
14	TRG0ST0RST	Slave_TIMER0复位事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置事件是否生成ADC触发事件。 0：HRTIMER Slave_TIMER0复位事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER0复位事件生成ADC触发事件。
13	TRG0ST0PER	Slave_TIMER0周期事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置周期事件生成ADC触发事件。 0：HRTIMER Slave_TIMER0周期事件不生成ADC触发事件。

		1：HRTIMER Slave_TIMER0周期事件生成ADC触发事件。
12	TRG0ST0C3	Slave_TIMER0比较3事件生成HRTIMER_ADCTRIG0上的ADC触发事件 请参考 TRG0ST0C1 描述。
11	TRG0ST0C2	Slave_TIMER0比较2事件生成HRTIMER_ADCTRIG0上的ADC触发事件 请参考 TRG0ST0C1 描述。
10	TRG0ST0C1	Slave_TIMER0比较1事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置比较1事件生成ADC触发事件。 0：HRTIMER Slave_TIMER0比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER0比较1事件生成ADC触发事件。
9	TRG0EXEV4	外部事件4生成生成HRTIMER_ADCTRIG0上的ADC触发事件 请参考 TRG0EXEV0 描述。
8	TRG0EXEV3	外部事件3生成生成HRTIMER_ADCTRIG0上的ADC触发事件 请参考 TRG0EXEV0 描述。
7	TRG0EXEV2	外部事件2生成生成HRTIMER_ADCTRIG0上的ADC触发事件 请参考 TRG0EXEV0 描述。
6	TRG0EXEV1	外部事件1生成生成HRTIMER_ADCTRIG0上的ADC触发事件 请参考 TRG0EXEV0 描述。
5	TRG0EXEV0	外部事件0生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：外部事件0（EXEV0C）不生成ADC触发事件。 1：外部事件0（EXEV0C）生成ADC触发事件。
4	TRG0MTPER	Master_TIMER周期事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置周期事件生成ADC触发事件。 0：HRTIMER Master_TIMER周期事件不生成ADC触发事件。 1：HRTIMER Master_TIMER周期事件生成ADC触发事件。
3	TRG0MTC3	Master_TIMER比较3事件生成HRTIMER_ADCTRIG0上的ADC触发事件 请参考TRG0MTC0描述。
2	TRG0MTC2	Master_TIMER比较2事件生成HRTIMER_ADCTRIG0上的ADC触发事件 请参考TRG0MTC0描述。
1	TRG0MTC1	Master_TIMER比较1事件生成HRTIMER_ADCTRIG0上的ADC触发事件 请参考TRG0MTC0描述。

0	TRG0MTC0	Master_TIMER比较0事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置比较1事件生成ADC触发事件。 0：HRTIMER Master_TIMER比较0事件不生成ADC触发事件。 1：HRTIMER Master_TIMER比较0事件生成ADC触发事件。

HRTIMER ADC 触发源 1 寄存器 (HRTIMER_ADCTRIGS1)

地址偏移：0x40

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TRG1ST4RST	TRG1ST4C3	TRG1ST4C2	TRG1ST4C1	TRG1ST3RST	TRG1ST3PER	TRG1ST3C3	TRG1ST3C2	TRG1ST3C1	TRG1ST2RST	TRG1ST2PER	TRG1ST2C3	TRG1ST2C2	TRG1ST2C1	TRG1ST1PER	TRG1ST1C3
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRG1ST1C2	TRG1ST1C1	TRG1ST0PER	TRG1ST0C3	TRG1ST0C2	TRG1ST0C1	TRG1EXEV9	TRG1EXEV8	TRG1EXEV7	TRG1EXEV6	TRG1EXEV5	TRG1MTPER	TRG1MTC3	TRG1MTC2	TRG1MTC1	TRG1MTC0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	TRG1ST4RST	Slave_TIMER4复位事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER4复位事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER4复位事件生成ADC触发事件。
30	TRG1ST4C3	Slave_TIMER4比较3事件生成HRTIMER_ADCTRIG1上的ADC触发事件 请参考 TRG1ST4C1 描述。
29	TRG1ST4C2	Slave_TIMER4比较2事件生成HRTIMER_ADCTRIG1上的ADC触发事件 请参考 TRG1ST4C1 描述。
28	TRG1ST4C1	Slave_TIMER4比较1事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于确定事件生成ADC触发事件。 0：HRTIMER Slave_TIMER4比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER4比较1事件生成ADC触发事件。
27	TRG1ST3RST	Slave_TIMER3复位事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件

		生成ADC触发事件。 0：HRTIMER Slave_TIMER3复位事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER3复位事件生成ADC触发事件。
26	TRG1ST3PER	Slave_TIMER3周期事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER3周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER3周期事件生成ADC触发事件。
25	TRG1ST3C3	Slave_TIMER3比较3事件生成HRTIMER_ADCTRIG1上的ADC触发事件 请参考 TRG1ST3C1 描述。
24	TRG1ST3C2	Slave_TIMER3比较2事件生成HRTIMER_ADCTRIG1上的ADC触发事件 请参考 TRG1ST3C1 描述。
23	TRG1ST3C1	Slave_TIMER3比较1事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER3比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER3比较1事件生成ADC触发事件。
22	TRG1ST2RST	Slave_TIMER2复位事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER2复位事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER2复位事件生成ADC触发事件。
21	TRG1ST2PER	Slave_TIMER2周期事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER2周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER2周期事件生成ADC触发事件。
20	TRG1ST2C3	Slave_TIMER2比较3事件生成HRTIMER_ADCTRIG1上的ADC触发事件 请参考 TRG1ST2C1 描述。
19	TRG1ST2C2	Slave_TIMER2比较2事件生成HRTIMER_ADCTRIG1上的ADC触发事件 请参考 TRG1ST2C1 描述。
18	TRG1ST2C1	Slave_TIMER2比较1事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER2比较1事件不生成ADC触发事件。

		1：HRTIMER Slave_TIMER2比较1事件生成ADC触发事件。
17	TRG1ST1PER	Slave_TIMER1周期事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER1周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER1周期事件生成ADC触发事件。
16	TRG1ST1C3	Slave_TIMER1比较3事件生成HRTIMER_ADCTRIG1上的ADC触发事件 请参考 TRG1ST1C1 描述。
15	TRG1ST1C2	Slave_TIMER1比较2事件生成HRTIMER_ADCTRIG1上的ADC触发事件 请参考 TRG1ST1C1 描述。
14	TRG1ST1C1	Slave_TIMER1比较1事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER1比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER1比较1事件生成ADC触发事件。
13	TRG1ST0PER	Slave_TIMER0周期事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER0周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER0周期事件生成ADC触发事件。
12	TRG1ST0C3	Slave_TIMER0比较3事件生成HRTIMER_ADCTRIG1上的ADC触发事件 请参考 TRG1ST0C1 描述。
11	TRG1ST0C2	Slave_TIMER0比较2事件生成HRTIMER_ADCTRIG1上的ADC触发事件 请参考 TRG1ST0C1 描述。
10	TRG1ST0C1	Slave_TIMER0比较1事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER0比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER0比较1事件生成ADC触发事件。
9	TRG1EXEV9	外部事件9事件生成HRTIMER_ADCTRIG1上的ADC触发事件 请参考 TRG1EXEV5 描述。
8	TRG1EXEV8	外部事件8事件生成HRTIMER_ADCTRIG1上的ADC触发事件 请参考 TRG1EXEV5 描述。
7	TRG1EXEV7	外部事件7事件生成HRTIMER_ADCTRIG1上的ADC触发事件

请参考 TRG1EXEV5 描述。

6	TRG1EXEV6	外部事件6事件生成HRTIMER_ADCTRIG1上的ADC触发事件 请参考 TRG1EXEV5 描述。
5	TRG1EXEV5	外部事件5生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：外部事件5（EXEV5C）不生成ADC触发事件。 1：外部事件5（EXEV5C）生成ADC触发事件。
4	TRG1MTPER	Master_TIMER周期事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Master_TIMER周期事件不生成ADC触发事件。 1：HRTIMER Master_TIMER周期事件生成ADC触发事件。
3	TRG1MTC3	Master_TIMER比较3事件生成HRTIMER_ADCTRIG1上的ADC触发事件 请参考TRG1MTC0描述。
2	TRG1MTC2	Master_TIMER比较2事件生成HRTIMER_ADCTRIG1上的ADC触发事件 请参考TRG1MTC0描述。
1	TRG1MTC1	Master_TIMER比较1事件生成HRTIMER_ADCTRIG1上的ADC触发事件 请参考TRG1MTC0描述。
0	TRG1MTC0	Master_TIMER比较0事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Master_TIMER比较0事件不生成ADC触发事件。 1：HRTIMER Master_TIMER比较0事件生成ADC触发事件。

HRTIMER ADC 触发源 2 寄存器 (HRTIMER_ADCTRIGS2)

地址偏移：0x44

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TRG2ST4 PER	TRG2ST4 C3	TRG2ST4 C2	TRG2ST4 C1	TRG2ST3 PER	TRG2ST3 C3	TRG2ST3 C2	TRG2ST3 C1	TRG2ST2 PER	TRG2ST2 C3	TRG2ST2 C2	TRG2ST2 C1	TRG2ST1 RST	TRG2ST1 PER	TRG2ST1 C3	TRG2ST1 C2
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

TRG2ST1C1	TRG2ST0RST	TRG2ST0PER	TRG2ST0C3	TRG2ST0C2	TRG2ST0C1	TRG2EXEV4	TRG2EXEV3	TRG2EXEV2	TRG2EXEV1	TRG2EXEV0	TRG2MTPER	TRG2MTC3	TRG2MTC2	TRG2MTC1	TRG2MTC0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	TRG2ST4PER	Slave_TIMER4周期事件生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER4周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER4周期事件生成ADC触发事件。
30	TRG2ST4C3	Slave_TIMER4比较3事件生成HRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2ST4C1 描述。
29	TRG2ST4C2	Slave_TIMER4比较2事件生成HRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2ST4C1 描述。
28	TRG2ST4C1	Slave_TIMER4比较1事件生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER4比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER4比较1事件生成ADC触发事件。
27	TRG2ST3PER	Slave_TIMER3周期事件生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER3周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER3周期事件生成ADC触发事件。
26	TRG2ST3C3	Slave_TIMER3比较3事件生成HRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2ST3C1 描述。
25	TRG2ST3C2	Slave_TIMER3比较2事件生成HRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2ST3C1 描述。
24	TRG2ST3C1	Slave_TIMER3比较1事件生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER3比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER3比较1事件生成ADC触发事件。
23	TRG2ST2PER	Slave_TIMER2周期事件生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。

		0：HRTIMER Slave_TIMER2周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER2周期事件生成ADC触发事件。
22	TRG2ST2C3	Slave_TIMER2比较3事件生成HRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2ST2C1 描述。
21	TRG2ST2C2	Slave_TIMER2比较2事件生成HRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2ST2C1 描述。
20	TRG2ST2C1	Slave_TIMER2比较1事件生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER2比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER2比较1事件生成ADC触发事件。
19	TRG2ST1RST	Slave_TIMER1复位事件生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER1复位事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER1复位事件生成ADC触发事件。
18	TRG2ST1PER	Slave_TIMER1周期事件生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER1周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER1周期事件生成ADC触发事件。
17	TRG2ST1C3	Slave_TIMER1比较3事件生成HRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2ST1C1 描述。
16	TRG2ST1C2	Slave_TIMER1比较2事件生成HRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2ST1C1 描述。
15	TRG2ST1C1	Slave_TIMER1比较1事件生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER1比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER1比较1事件生成ADC触发事件。
14	TRG2ST0RST	Slave_TIMER0复位事件生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER0复位事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER0复位事件生成ADC触发事件。

13	TRG2ST0PER	Slave_TIMER0周期事件生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER0周期事件上生成ADC触发事件。 1：HRTIMER Slave_TIMER0周期事件上成ADC触发事件。
12	TRG2ST0C3	Slave_TIMER0比较3事件生成HRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2ST0C1 描述。
11	TRG2ST0C2	Slave_TIMER0比较2事件生成HRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2ST0C1 描述。
10	TRG2ST0C1	Slave_TIMER0比较1事件生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER0比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER0比较1事件生成ADC触发事件。
9	TRG2EXEV4	外部事件4生成HRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2EXEV0 描述。
8	TRG2EXEV3	外部事件3生成HRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2EXEV0 描述。
7	TRG2EXEV2	外部事件2生成HRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2EXEV0 描述。
6	TRG2EXEV1	外部事件1生成HRTIMER_ADCTRIG2上的ADC触发事件 请参考 TRG2EXEV0 描述。
5	TRG2EXEV0	外部事件0生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：外部事件0（EXEV0C）不生成ADC触发事件。 1：外部事件0（EXEV0C）生成ADC触发事件。
4	TRG2MTPER	Master_TIMER周期事件生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Master_TIMER周期事件不生成ADC触发事件。 1：HRTIMER Master_TIMER周期事件生成ADC触发事件。
3	TRG2MTC3	Master_TIMER比较3事件生成HRTIMER_ADCTRIG2上的ADC触发事件 请参考TRG2MTC0描述。
2	TRG2MTC2	Master_TIMER比较2事件生成HRTIMER_ADCTRIG2上的ADC触发事件

请参考TRG2MTC0描述。

1	TRG2MTC1	Master_TIMER比较31事件生成HRTIMER_ADCTRIG2上的ADC触发事件 请参考TRG2MTC0描述。
0	TRG2MTC0	Master_TIMER比较0事件生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Master_TIMER比较0事件不生成ADC触发事件。 1：HRTIMER Master_TIMER比较0事件生成ADC触发事件。

HRTIMER ADC 触发源 3 寄存器 (HRTIMER_ADCTRIGS3)

地址偏移：0x48

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TRG3ST4RST	TRG3ST4C3	TRG3ST4C2	TRG3ST4C1	TRG3ST3RST	TRG3ST3PER	TRG3ST3C3	TRG3ST3C2	TRG3ST3C1	TRG3ST2RST	TRG3ST2PER	TRG3ST2C3	TRG3ST2C2	TRG3ST2C1	TRG3ST1PER	TRG3ST1C3
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRG3ST1C2	TRG3ST1C1	TRG3ST0PER	TRG3ST0C3	TRG3ST0C2	TRG3ST0C1	TRG3EXEV9	TRG3EXEV8	TRG3EXEV7	TRG3EXEV6	TRG3EXEV5	TRG3MTPER	TRG3MTC3	TRG3MTC2	TRG3MTC1	TRG3MTC0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	TRG3ST4RST	Slave_TIMER4复位事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER4复位事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER4复位事件生成ADC触发事件。
30	TRG3ST4C3	Slave_TIMER4比较3事件生成HRTIMER_ADCTRIG3上的ADC触发事件 请参考 TRG3ST4C1 描述。
29	TRG3ST4C2	Slave_TIMER4比较2事件生成HRTIMER_ADCTRIG3上的ADC触发事件 请参考 TRG3ST4C1 描述。
28	TRG3ST4C1	Slave_TIMER4比较1事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER4比较1事件不生成ADC触发事件。

		1：HRTIMER Slave_TIMER4比较1事件生成ADC触发事件。
27	TRG3ST3RST	Slave_TIMER3复位事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER3复位事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER3复位事件生成ADC触发事件。
26	TRG3ST3PER	Slave_TIMER3周期事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER3周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER3周期事件生成ADC触发事件。
25	TRG3ST3C3	Slave_TIMER3比较3事件生成HRTIMER_ADCTRIG3上的ADC触发事件 请参考 TRG3ST3C1 描述。
24	TRG3ST3C2	Slave_TIMER3比较2事件生成HRTIMER_ADCTRIG3上的ADC触发事件 请参考 TRG3ST3C1 描述。
23	TRG3ST3C1	Slave_TIMER3比较1事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER3比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER3比较1事件生成ADC触发事件。
22	TRG3ST2RST	Slave_TIMER2复位事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER2复位事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER2复位事件生成ADC触发事件。
21	TRG3ST2PER	Slave_TIMER2周期事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER2周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER2周期事件生成ADC触发事件。
20	TRG3ST2C3	Slave_TIMER2比较3事件生成HRTIMER_ADCTRIG3上的ADC触发事件 请参考 TRG3ST2C1 描述。
19	TRG3ST2C2	Slave_TIMER2比较2事件生成HRTIMER_ADCTRIG3上的ADC触发事件 请参考 TRG3ST2C1 描述。
18	TRG3ST2C1	Slave_TIMER2比较1事件生成HRTIMER_ADCTRIG3上的ADC触发事件

		HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER2比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER2比较1事件生成ADC触发事件。
17	TRG3ST1PER	Slave_TIMER1周期事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER1周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER1周期事件生成ADC触发事件。
16	TRG3ST1C3	Slave_TIMER1比较3事件生成HRTIMER_ADCTRIG3上的ADC触发事件 请参考 TRG3ST1C1 描述。
15	TRG3ST1C2	Slave_TIMER1比较2事件生成HRTIMER_ADCTRIG3上的ADC触发事件 请参考 TRG3ST1C1 描述。
14	TRG3ST1C1	Slave_TIMER1比较1事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER1比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER1比较1事件生成ADC触发事件。
13	TRG3ST0PER	Slave_TIMER0周期事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER0周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER0周期事件生成ADC触发事件。
12	TRG3ST0C3	Slave_TIMER0比较3事件生成HRTIMER_ADCTRIG3上的ADC触发事件 请参考 TRG3ST0C1 描述。
11	TRG3ST0C2	Slave_TIMER0比较2事件生成HRTIMER_ADCTRIG3上的ADC触发事件 请参考 TRG3ST0C1 描述。
10	TRG3ST0C1	Slave_TIMER0比较1事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Slave_TIMER0比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER0比较1事件生成ADC触发事件。
9	TRG3EXEV9	外部事件9事件生成HRTIMER_ADCTRIG3上的ADC触发事件 请参考 TRG3EXEV5 描述。
8	TRG3EXEV8	外部事件8事件生成HRTIMER_ADCTRIG3上的ADC触发事件

		请参考 TRG3EXEV5 描述。
7	TRG3EXEV7	外部事件7事件生成HRTIMER_ADCTRIG3上的ADC触发事件 请参考 TRG3EXEV5 描述。
6	TRG3EXEV6	外部事件6事件生成HRTIMER_ADCTRIG3上的ADC触发事件 请参考 TRG3EXEV5 描述。
5	TRG3EXEV5	外部事件5生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：外部事件5（EXEV5C）不生成ADC触发事件。 1：外部事件5（EXEV5C）生成ADC触发事件。
4	TRG3MTPER	Master_TIMER周期事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Master_TIMER周期事件不生成ADC触发事件。 1：HRTIMER Master_TIMER周期事件生成ADC触发事件。
3	TRG3MTC3	Master_TIMER比较3事件生成HRTIMER_ADCTRIG3上的ADC触发事件 请参考TRG3MTC0描述。
2	TRG3MTC2	Master_TIMER比较2事件生成HRTIMER_ADCTRIG3上的ADC触发事件 请参考TRG3MTC0描述。
1	TRG3MTC1	Master_TIMER比较1事件生成HRTIMER_ADCTRIG3上的ADC触发事件 请参考TRG3MTC0描述。
0	TRG3MTC0	Master_TIMER比较0事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置事件生成ADC触发事件。 0：HRTIMER Master_TIMER比较0事件不生成ADC触发事件。 1：HRTIMER Master_TIMER比较0事件生成ADC触发事件。

HRTIMER DLL 校准控制寄存器 (HRTIMER_DLLCCTL)

地址偏移：0x4C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

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

保留	CLBPER[1:0]	CLBPEREN	CLBSTRT
	rw	rw	wo

位/位域	名称	描述
31:4	保留	必须保持复位值
3:2	CLBPER[1:0]	DLL 校准周期 该位域定义了 DLL 校准周期的长度。 00：1048576 * $t_{HRTIMER_CK}$ 01：131072 * $t_{HRTIMER_CK}$ 10：16384 * $t_{HRTIMER_CK}$ 11：2048 * $t_{HRTIMER_CK}$
1	CLBPEREN	DLL 定期校准使能 该位用于使能定期的 DLL 校准。CLBPER [1:0]位域用于设置校准周期。 0：DLL 定期校准禁能。 1：DLL 定期校准使能。 **注意：**不能同时设置 CLBPEREN 位和 CLBSTRT 位。
0	CLBSTRT	DLL 校准开始 当 CLBPEREN = 0 时，向该位写入 1 将启动 DLL 校准。该位只能进行写操作。 0：无效。 1：DLL 校准开始。 **注意：**不能同时设置 CLBPEREN 位和 CLBSTRT 位。

HRTIMER 故障输入配置寄存器 0 (HRTIMER_FLTINCFG0)

地址偏移：0x50

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLT3INPROT	FLT3INFC[3:0]				FLT3INSRC[0]	FLT3INP	FLT3INEN	FLT2INPROT	FLT2INFC[3:0]				FLT2INSRC[0]	FLT2INP	FLT2INEN
rwo	rw				rw	rw	rw	rwo	rw				rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLT1INPROT	FLT1INFC[3:0]				FLT1INSRC[0]	FLT1INP	FLT1INEN	FLT0INPROT	FLT0INFC[3:0]				FLT0INSRC[0]	FLT0INP	FLT0INEN
rwo	rw				rw	rw	rw	rwo	rw				rw	rw	rw

位/位域	名称	描述

31	FLT3INPROT	保护故障 3 输入配置 请参考 FLT0INPROT 描述。
30:27	FLT3INFC[3:0]	故障 3 输入滤波配置 请参考 FLT0INFC[3:0]描述。
26	FLT3INSRC[0]	故障 3 输入源 请参考 FLT0INSRC[0]描述。
25	FLT3INP	故障 3 输入极性 请参考 FLT0INP 描述。
24	FLT3INEN	故障 3 输入使能 请参考 FLT0INEN 描述。
23	FLT2INPROT	保护故障 2 输入配置 请参考 FLT0INPROT 描述。
22:19	FLT2INFC[3:0]	故障 3 输入滤波控制 请参考 FLT0INFC[3:0]描述。
18	FLT2INSRC[0]	故障 2 输入源 请参考 FLT0INSRC[0]描述。
17	FLT2INP	故障 2 输入极性 请参考 FLT0INP 描述。
16	FLT2INEN	故障 2 输入使能 请参考 FLT0INEN 描述。
15	FLT1INPROT	保护故障 1 输入配置 请参考 FLT0INPROT 描述。
14:11	FLT1INFC[3:0]	故障 1 输入滤波控制 请参考 FLT0INFC[3:0]描述。
10	FLT1INSRC[0]	故障 1 输入源 请参考 FLT0INSRC[0]描述。
9	FLT1INP	故障 1 输入极性 请参考 FLT0INP 描述。
8	FLT1INEN	故障 1 输入使能 请参考 FLT0INEN 描述。
7	FLT0INPROT	保护故障 0 输入配置 该位域用于配置故障 0 输入配置的写保护功能。该位写一次有效。一旦由软件置位，只能通过系统复位将其清除。

0：保护禁能。FLT0INEN，FLT0INP，FLT0INSRC 和 FLT0INFC [3:0]是可写的。
1：保护使能。FLT0INEN，FLT0INP，FLT0INSRC 和 FLT0INFC [3:0]是只读的。

6:3 FLT0INFC[3:0] 故障 0 输入滤波控制

在数字滤波器中使用事件计数器，N 次输入事件后，输出才会发生转变。该位域用于确定采样故障输入的频率（f_{SAMP}）以及应用于故障输入的数字滤波器的长度。

0000：无滤波。

0001：$f_{SAMP} = f_{HRTIMER_CK}$, N=2.

0010：$f_{SAMP} = f_{HRTIMER_CK}$, N=4.

0011：$f_{SAMP} = f_{HRTIMER_CK}$, N=8.

0100：$f_{SAMP} = f_{HRTIMER_FLTFCK}$ /2, N=6.

0101：$f_{SAMP} = f_{HRTIMER_FLTFCK}$ /2, N=8.

0110：$f_{SAMP} = f_{HRTIMER_FLTFCK}$ /4, N=6.

0111：$f_{SAMP} = f_{HRTIMER_FLTFCK}$ /4, N=8.

1000：$f_{SAMP} = f_{HRTIMER_FLTFCK}$ /8, N=6.

1001：$f_{SAMP} = f_{HRTIMER_FLTFCK}$ /8, N=8.

1010：$f_{SAMP} = f_{HRTIMER_FLTFCK}$ /16, N=5.

1011：$f_{SAMP} = f_{HRTIMER_FLTFCK}$ /16, N=6.

1100：$f_{SAMP} = f_{HRTIMER_FLTFCK}$ /16, N=8.

1101：$f_{SAMP} = f_{HRTIMER_FLTFCK}$ /32, N=5.

1110：$f_{SAMP} = f_{HRTIMER_FLTFCK}$ /32, N=6.

1111：$f_{SAMP} = f_{HRTIMER_FLTFCK}$ /32, N=8.

注意：

（1）仅当 FLT0INEN 位复位时，才能对该位进行写操作。

（2）当 FLT0INPROT 位配置时，不能修改该位域。

2 FLT0INSRC[0] 故障 0 输入源，与 FLT0INSRC[1]结合。

00：故障 0 的输入源是芯片外部引脚。

01：故障 0 的输入源是芯片内部信号（如比较器）。

10：故障 0 的输入源是外部事件。

其他值保留。

注意：仅当 FLT0INEN 位复位时，才能对该位进行写操作。

1 FLT0INP 故障 0 输入极性

该位用于配置故障 0 的输入极性。

0：故障 0 输入为低电平有效。

1：故障 0 输入为高电平有效。

0 FLT0INEN 故障 0 输入使能

该位置 1 时，使能故障 0 输入。

0：故障 0 输入禁能。

1：故障 0 输入使能。

HRTIMER 故障输入配置寄存器 1 (HRTIMER_FLTINCFG1)

地址偏移：0x54

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						FLTFDIV[2:0]		FLT7INSRC[1]	FLT6INSRC[1]	FLT5INSRC[1]	FLT4INSRC[1]	FLT3INSRC[1]	FLT2INSRC[1]	FLT1INSRC[1]	FLT0INSRC[1]
						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
FLT5INPROT	FLT5INFC[3:0]				FLT5INSRC[0]	FLT5INP	FLT5INEN	FLT4INPROT	FLT4INFC[3:0]				FLT4INSRC[0]	FLT4INP	FLT4INEN
rw	rw				rw	rw	rw	rwo	rw				rw	rw	rw

位/位域	名称	描述
31:26	保留	必须保持复位值
25:24	FLTFDIV[2:0]	故障输入数字滤波器时钟分频 该位域可由软件配置，以确定 HRTIMER 时钟（HRTIMER_CK）和故障输入数字滤波器时钟（HRTIMER_FLTFCK）之间的分频比。 $f_{HRTIMER_FLTFCK} = f_{HRTIMER_CK}/2^{FLTFDIV[2:0]}$。 00：$f_{HRTIMER_FLTFCK} = f_{HRTIMER_CK}$。 01：$f_{HRTIMER_FLTFCK} = f_{HRTIMER_CK}/2$。 10：$f_{HRTIMER_FLTFCK} = f_{HRTIMER_CK}/4$。 11：$f_{HRTIMER_FLTFCK} = f_{HRTIMER_CK}/8$。 **注意：**必须在设置 FLTyINEN（y=0..4）之前配置该位。
23	FLT7INSRC[1]	故障 7 输入源 请参考 FLT0INSRC[0]描述。
22	FLT6INSRC[1]	故障 6 输入源 请参考 FLT0INSRC[0]描述。
21	FLT5INSRC[1]	故障 5 输入源 请参考 FLT0INSRC[0]描述。
20	FLT4INSRC[1]	故障 4 输入源 请参考 FLT0INSRC[0]描述。
19	FLT3INSRC[1]	故障 3 输入源 请参考 FLT0INSRC[0]描述。
18	FLT2INSRC[1]	故障 2 输入源

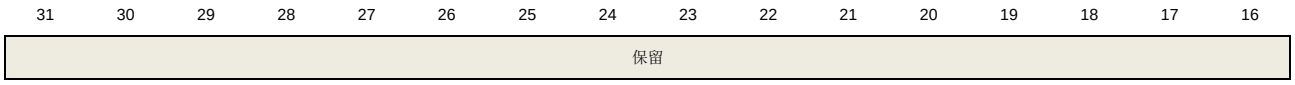
		请参考 FLT0INSRC[0]描述。
17	FLT1INSRC[1]	故障 1 输入源 请参考 FLT0INSRC[0]描述。
16	FLT0INSRC[1]	故障 0 输入源 请参考 FLT0INSRC[0]描述。
15	FLT5INPROT	保护故障 5 输入配置 请参考 HRTIMER_FLTINCFG0 寄存器中的 FLT0INPROT 位描述。
14:11	FLT5INFC[3:0]	故障 5 输入滤波控制 请参考 HRTIMER_FLTINCFG0 寄存器中的 FLT0INFC[3:0]位域描述。
10	FLT5INSRC[0]	故障 5 输入源 请参考 FLT0INSRC[0]描述。
9	FLT5INP	故障 5 输入极性 请参考 HRTIMER_FLTINCFG0 寄存器中的 FLT0INP 位描述。
8	FLT5INEN	故障 5 输入使能 请参考 HRTIMER_FLTINCFG0 寄存器中的 FLT0INEN 位描述。
7	FLT4INPROT	保护故障 4 输入配置 请参考 HRTIMER_FLTINCFG0 寄存器中的 FLT0INPROT 位描述。
6:3	FLT4INFC[3:0]	故障 4 输入滤波控制 请参考 HRTIMER_FLTINCFG0 寄存器中的 FLT0INFC[3:0]位域描述。
2	FLT4INSRC[0]	故障 4 输入源 请参考 HRTIMER_FLTINCFG0 寄存器中的 FLT0INSRC[0]位描述。
1	FLT4INP	故障 4 输入极性 请参考 HRTIMER_FLTINCFG0 寄存器中的 FLT0INP 位描述。
0	FLT4INEN	故障 4 输入使能 请参考 HRTIMER_FLTINCFG0 寄存器中的 FLT0INEN 位描述。

HRTIMER DMA 更新 Master_TIMER 寄存器 (HRTIMER_DMAUPMTR)

地址偏移：0x58

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						MTCMP3V	MTCMP2V	MTCMP1V	MTCMP0V	MTCREP	MTCAR	MTCNT	MTDMAINTEN	MTINTC	MTCTL0
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	保留	必须保持复位值
30:10	保留	必须保持复位值
9	MTCMP3V	通过 DMA 模式更新 HRTIMER_MTCMP3V 寄存器 请参考MTCTL0描述。
8	MTCMP2V	通过 DMA 模式更新 HRTIMER_MTCMP2V 寄存器 请参考 MTCTL0 描述。
7	MTCMP1V	通过 DMA 模式更新 HRTIMER_MTCMP1V 寄存器 请参考 MTCTL0 描述。
6	MTCMP0V	通过 DMA 模式更新 HRTIMER_MTCMP0V 寄存器 请参考 MTCTL0 描述。
5	MTCREP	通过 DMA 模式更新 HRTIMER_MTCREP 寄存器 请参考 MTCTL0 描述。
4	MTCAR	通过 DMA 模式更新 HRTIMER_MTCAR 寄存器 请参考 MTCTL0 描述。
3	MTCNT	通过 DMA 模式更新 HRTIMER_MTCNT 寄存器 请参考 MTCTL0 描述。
2	MTDMAINTEN	通过 DMA 模式更新 HRTIMER_MTDMAINTEN 寄存器 请参考 MTCTL0 描述。
1	MTINTC	通过 DMA 模式更新 HRTIMER_MTINTC 寄存器 请参考 MTCTL0 描述。
0	MTCTL0	通过 DMA 模式更新 HRTIMER_MTCTL0 寄存器 该位用于配置通过 DMA 模式更新 HRTIMER_MTCTL0 寄存器。 0：DMA 模式不能更新 HRTIMER_MTCTL0 寄存器。 1：通过 DMA 模式更新 HRTIMER_MTCTL0 寄存器。

HRTIMER DMA 更新 Slavex_TIMER 寄存器 (HRTIMER_DMAUPSTxR)(x=0..7)

地址偏移：Slave_TIMER0: 0x5C

地址偏移：Slave_TIMER1: 0x60
地址偏移：Slave_TIMER2: 0x64
地址偏移：Slave_TIMER3: 0x68
地址偏移：Slave_TIMER4: 0x6C
地址偏移：Slave_TIMER5: 0x74
地址偏移：Slave_TIMER6: 0x100
地址偏移：Slave_TIMER7: 0x104
复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STxACTL	保留								STxEXEVFCFG2	STxCTL1	STxFLTCTL	STxCHOCTL	STxCSCTL	STxCNTRST	STxEXEVFCFG1
rw									rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STxEXEVFCFG0	STxCH1RST	STxCH1SET	STxCH0RST	STxCH0SET	STxDTCTL	STxCMP3V	STxCMP2V	STxCMP1V	STxCMP0V	STxCREP	STxCAR	STxCNT	STxDMAINTEN	STxINTC	STxCTL0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	STxACTL	通过 DMA 模式更新 HRTIMER_STxACTL 寄存器 请参考 STxCTL0 描述。
30:23	保留	必须保持复位值
22	STxEXEVFCFG2	通过 DMA 模式更新 HRTIMER_STxEXEVFCFG2 寄存器 请参考 STxCTL0 描述。
21	STxCTL1	通过 DMA 模式更新 STxCTL1 寄存器 请参考 STxCTL0 描述。
20	STxFLTCTL	通过 DMA 模式更新 HRTIMER_STxFLTCTL 寄存器 请参考 STxCTL0 描述。
19	STxCHOCTL	通过 DMA 模式更新 HRTIMER_STxCHOCTL 寄存器 请参考 STxCTL0 描述。
18	STxCSCTL	通过 DMA 模式更新 HRTIMER_STxCSCTL 寄存器 请参考 STxCTL0 描述。
17	STxCNTRST	通过 DMA 模式更新 HRTIMER_STxCNTRST 寄存器 请参考 STxCTL0 描述。
16	STxEXEVFCFG1	通过 DMA 模式更新 HRTIMER_STxEXEVFCFG1 寄存器

		请参考 STxCTL0 描述。
15	STxEXEVFCFG0	通过 DMA 模式更新 HRTIMER_STxEXEVFCFG0 寄存器 请参考 STxCTL0 描述。
14	STxCH1RST	通过 DMA 模式更新 HRTIMER_STxCH1RST 寄存器 请参考 STxCTL0 描述。
13	STxCH1SET	通过 DMA 模式更新 HRTIMER_STxCH1SET 寄存器 请参考 STxCTL0 描述。
12	STxCH0RST	通过 DMA 模式更新 HRTIMER_STxCH0RST 寄存器 请参考 STxCTL0 描述。
11	STxCH0SET	通过 DMA 模式更新 HRTIMER_STxCH0SET 寄存器 请参考 STxCTL0 描述。
10	STxDTCTL	通过 DMA 模式更新 HRTIMER_STxDTCTL 寄存器 请参考 STxCTL0 描述。
9	STxCMP3V	通过 DMA 模式更新 HRTIMER_STxCMP3V 寄存器 请参考 STxCTL0 描述。
8	STxCMP2V	通过 DMA 模式更新 HRTIMER_STxCMP2V 寄存器 请参考 STxCTL0 描述。
7	STxCMP1V	通过 DMA 模式更新 HRTIMER_STxCMP1V 寄存器 请参考 STxCTL0 描述。
6	STxCMP0V	通过 DMA 模式更新 HRTIMER_STxCMP0V 寄存器 请参考 STxCTL0 描述。
5	STxCREP	通过 DMA 模式更新 HRTIMER_STxCREP 寄存器 请参考 STxCTL0 描述。
4	STxCAR	通过 DMA 模式更新 HRTIMER_STxCAR 寄存器 请参考 STxCTL0 描述。
3	STxCNT	通过 DMA 模式更新 HRTIMER_STxCNT 寄存器 请参考 STxCTL0 描述。
2	STxDMAINTEN	通过 DMA 模式更新 HRTIMER_STxDMAINTEN 寄存器 请参考 STxCTL0 描述。
1	STxINTC	通过 DMA 模式更新 HRTIMER_STxINTC 寄存器 请参考 STxCTL0 描述。
0	STxCTL0	通过 DMA 模式更新 HRTIMER_STxCTL0 寄存器

该位用于配置 DMA 模式更新 HRTIMER_STxCTL0 寄存器。

0：DMA 模式不能更新 HRTIMER_STxCTL0 寄存器。

1：通过 DMA 模式更新 HRTIMER_STxCTL0 寄存器。

HRTIMER DMA 传输缓冲寄存器 (HRTIMER_DMATB)

地址偏移：0x70

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

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

位/位域	名称	描述
31:0	DMATB[31:16]	DMA 传输缓冲区 对该寄存器进行写操作时，实际访问的寄存器是 HRTIMER_DMAUPMTR 寄存器和 HRTIMER_DMAUPSTxR（x = 0..4）寄存器中使能的寄存器。 寄存器指针的增量由硬件计算。

HRTIMER ADC 外部触发寄存器 (HRTIMER_ADCEXTTRG)

地址偏移：0x78

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	ADC9TRG[4:0]					ADC8TRG[4:0]					ADC7TRG[4:0]				
	rw					rw					rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	ADC6TRG[4:0]					ADC5TRG[4:0]					ADC4TRG[4:0]				
	rw					rw					rw				

位/位域	名称	描述
31	保留	必须保持复位值
30:26	ADC9TRG[4:0]	ADC 触发源 9 选择，该位域选择 ADC 触发源 9

		请参考 ADC5TRG[4:0]描述。
25:21	ADC8TRG[4:0]	ADC 触发源 8 选择，该位域选择 ADC 触发源 8 请参考 ADC4TRG[4:0]描述。
20:16	ADC7TRG[4:0]	ADC 触发源 7 选择，该位域选择 ADC 触发源 7 请参考ADC5TRG[4:0]描述。
15	保留	必须保持复位值
14:10	ADC6TRG[4:0]	ADC 触发源 6 选择，该位域选择 ADC 触发源 6 请参考 ADC4TRG[4:0]描述。
9:5	ADC5TRG[4:0]	ADC 触发源 5 选择，该位域选择 ADC 触发源 5 000000: 由 Master_TIMER 的比较 0 事件触发 ADC 000001: 由 Master_TIMER 的比较 1 事件触发 ADC 000010: 由 Master_TIMER 的比较 2 事件触发 ADC 000011: 由 Master_TIMER 的比较 3 事件触发 ADC 000100: 由 Master_TIMER 的周期事件触发 ADC 000101: 由外部事件 5 触发 ADC 000110: 由外部事件 6 触发 ADC 000111: 由外部事件 7 触发 ADC 001000: 由外部事件 8 触发 ADC 001001: 由外部事件 9 触发 ADC 001010: 由 Slave_TIMER0 的比较 1 事件触发 ADC 001011: 由 Slave_TIMER0 的比较 3 事件触发 ADC 001100: 由 Slave_TIMER0 的周期事件触发 ADC 001101: 由 Slave_TIMER1 的比较 1 事件触发 ADC 001110: 由 Slave_TIMER1 的比较 3 事件触发 ADC 001111: 由 Slave_TIMER1 的周期事件触发 ADC 010000: 由 Slave_TIMER2 的比较 1 事件触发 ADC 010001: 由 Slave_TIMER2 的比较 3 事件触发 ADC 010010: 由 Slave_TIMER2 的周期事件触发 ADC 010011: 由 Slave_TIMER2 的复位和计数翻转事件触发 ADC 010100: 由 Slave_TIMER3 的比较 1 事件触发 ADC 010101: 由 Slave_TIMER3 的比较 3 事件触发 ADC 010110: 由 Slave_TIMER3 的周期事件触发 ADC 010111: 由 Slave_TIMER3 的复位和计数翻转事件触发 ADC 011000: 由 Slave_TIMER4 的比较 1 事件触发 ADC 011001: 由 Slave_TIMER4 的比较 2 事件触发 ADC 011010: 由 Slave_TIMER4 的比较 3 事件触发 ADC 011011: 由 Slave_TIMER4 的复位和计数翻转事件触发 ADC 011100: 由 Slave_TIMER5 的比较 1 事件触发 ADC

011101: 由 Slave_TIMER5 的比较 2 事件触发 ADC
011110: 由 Slave_TIMER5 的比较 3 事件触发 ADC
011111: 由 Slave_TIMER5 的周期事件触发 ADC
100000: 由 Slave_TIMER6 的比较 1 事件触发 ADC
100001: 由 Slave_TIMER6 的比较 2 事件触发 ADC
100010: 由 Slave_TIMER6 的比较 3 事件触发 ADC
100011: 由 Slave_TIMER6 的周期事件触发 ADC
100100: 由 Slave_TIMER6 的复位和计数翻转事件触发 ADC
100101: 由 Slave_TIMER7 的比较 1 事件触发 ADC
100110: 由 Slave_TIMER7 的比较 2 事件触发 ADC
100111: 由 Slave_TIMER7 的比较 3 事件触发 ADC
101000: 由 Slave_TIMER7 的周期事件触发 ADC
101001: 由 Slave_TIMER7 的复位和计数翻转事件触发 ADC
其他：保留。
注意：ADC5TRG[5]见 HRTIMER_ADCEXTTRGA。

4:0	ADC4TRG[4:0]	ADC 触发源 4 选择，该位域选择 ADC 触发源 4 000000: 由Master_TIMER的比较0事件触发ADC 000001: 由Master_TIMER的比较1事件触发ADC 000010: 由Master_TIMER的比较2事件触发ADC 000011: 由Master_TIMER的比较3事件触发ADC 000100: 由Master_TIMER的周期事件触发ADC 000101: 由外部事件0触发ADC 000110: 由外部事件1触发ADC 000111: 由外部事件2触发ADC 001000: 由外部事件3触发ADC 001001: 由外部事件4触发ADC 001010: 由Slave_TIMER0的比较2事件触发ADC 001011: 由Slave_TIMER0的比较3事件触发ADC 001100: 由Slave_TIMER0的周期事件触发ADC 001101: 由Slave_TIMER0的复位和计数翻转事件触发ADC 001110: 由Slave_TIMER1的比较2事件触发ADC 001111: 由Slave_TIMER1的比较3事件触发ADC 010000: 由Slave_TIMER1的周期事件触发ADC 010001: 由Slave_TIMER1的复位和计数翻转事件触发ADC 010010: 由Slave_TIMER2的比较2事件触发ADC 010011: 由Slave_TIMER2的比较3事件触发ADC 010100: 由Slave_TIMER2的周期事件触发ADC 010101: 由Slave_TIMER3的比较2事件触发ADC 010110: 由Slave_TIMER3的比较3事件触发ADC 010111: 由Slave_TIMER3的周期事件触发ADC

011000: 由Slave_TIMER4的比较2事件触发ADC
011001: 由Slave_TIMER4的比较3事件触发ADC
011010: 由Slave_TIMER4的周期事件触发ADC
011011: 由Slave_TIMER5的比较1事件触发ADC
011100: 由Slave_TIMER5的比较2事件触发ADC
011101: 由Slave_TIMER5的比较3事件触发ADC
011110: 由Slave_TIMER5的周期事件触发ADC
011111: 由Slave_TIMER5的复位和计数翻转事件触发ADC
100000: 由Slave_TIMER6的比较1事件触发ADC
100001: 由Slave_TIMER6的比较2事件触发ADC
100010: 由Slave_TIMER6的比较3事件触发ADC
100011: 由Slave_TIMER6的周期事件触发ADC
100100: 由Slave_TIMER6的复位和计数翻转事件触发ADC
100101: 由Slave_TIMER7的比较1事件触发ADC
100110: 由Slave_TIMER7的比较2事件触发ADC
100111: 由Slave_TIMER7的比较3事件触发ADC
101000: 由Slave_TIMER7的周期事件触发ADC
101001: 由Slave_TIMER7的复位和计数翻转事件触发ADC
其他: 保留

注意：ADC4TRG[5]见HRTIMER_ADCEXTTRGA。

HRTIMER ADC 外部触发寄存器 (HRTIMER_ADCTRGUPD)

地址偏移：0x7C
复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								ADTG9USRC[3:0]				ADTG8USRC[3:0]			
								rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADTG7USRC[3:0]				ADTG6USRC[3:0]				ADTG5USRC[3:0]				ADTG4USRC[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:24	保留	必须保持复位值
23:20	ADTG9USRC[3:0]	ADC9TRG 更新源 请参考 ADTG4USRC[3:0]描述。
19:16	ADTG8USRC[3:0]	ADC8TRG 更新源

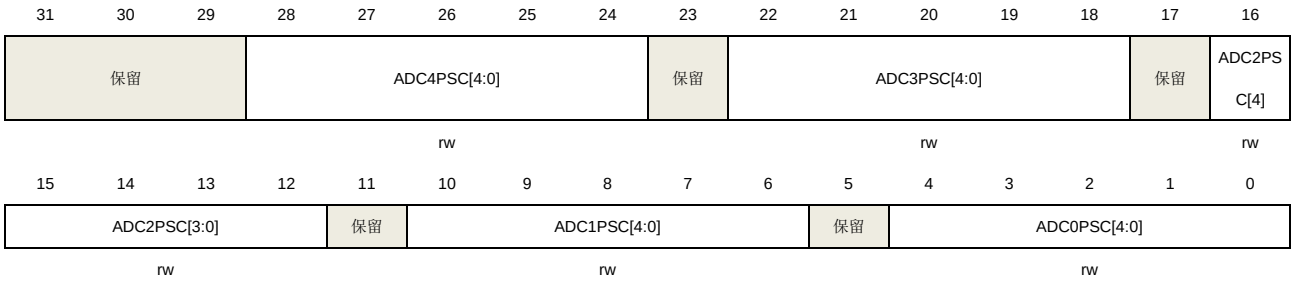
		请参考 ADTG4USRC[3:0]描述。
15:12	ADTG7USRC[3:0]	ADC7TRG 更新源 请参考ADTG4USRC[3:0]描述。
11:8	ADTG6USRC[3:0]	ADC6TRG 更新源 请参考ADTG4USRC[3:0]描述。
7:4	ADTG5USRC[3:0]	ADC5TRG 更新源 请参考 ADTG4USRC[3:0]描述。
3:0	ADTG4USRC[3:0]	ADC4TRG 更新源 该位域可以由软件配置，配置寄存器 HRTIMER_ADCEXTTRG 和 HRTIMER_ADCEXTTRGA 中的 ADC4TRG[5:0]的更新源。 0000：Master_TIMER 更新事件 0001：Slaver_TIMER0 更新事件 0010：Slaver_TIMER1 更新事件 0011：Slaver_TIMER2 更新事件 0100：Slaver_TIMER3 更新事件 0101：Slaver_TIMER4 更新事件 0110：Slaver_TIMER5 更新事件 0111：Slaver_TIMER6 更新事件 1000：Slaver_TIMER7 更新事件 其他值保留。

HRTIMER ADC 预分频寄存器 0 (HRTIMER_ADCPSCR0)

地址偏移：0x80

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。



位/位域	名称	描述
31:29	保留	必须保持复位值
28:24	ADC4PSC[4:0]	ADC 外部触发 4 预分频

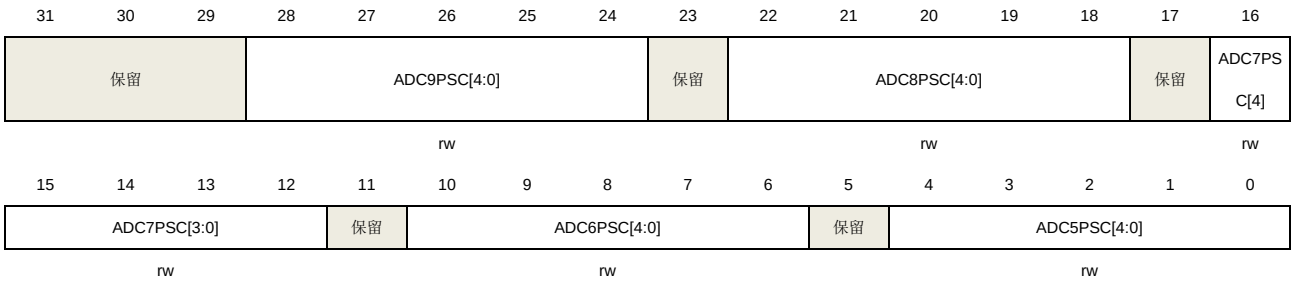
		该位域选择 ADC 外部触发 4 预分频
23	保留	必须保持复位值
22:18	ADC3PSC[4:0]	ADC 外部触发 3 预分频 该位域选择ADC外部触发3预分频
17	保留	必须保持复位值
16:12	ADC2PSC[4:0]	ADC 外部触发 2 预分频 该位域选择 ADC 外部触发 2 预分频
11	保留	必须保持复位值
10:6	ADC1PSC[4:0]	ADC 外部触发 1 预分频 该位域选择ADC外部触发1预分频
5	保留	必须保持复位值
4:0	ADC0PSC[4:0]	ADC 外部触发 0 预分频 该位域选择 ADC 外部触发 0 预分频

HRTIMER ADC 预分频寄存器 1 (HRTIMER_ADCPSCR1)

地址偏移：0x84

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。



位/位域	名称	描述
31:29	保留	必须保持复位值
28:24	ADC9PSC[4:0]	ADC 外部触发 9 预分频 该位域选择 ADC 外部触发 9 预分频
23	保留	必须保持复位值
22:18	ADC8PSC[4:0]	ADC 外部触发 8 预分频 该位域选择ADC外部触发8预分频

17	保留	必须保持复位值
16:12	ADC7PSC[4:0]	ADC 外部触发 7 预分频 该位域选择 ADC 外部触发 7 预分频
11	保留	必须保持复位值
10:6	ADC6PSC[4:0]	ADC 外部触发 6 预分频 该位域选择ADC外部触发6预分频
5	保留	必须保持复位值
4:0	ADC5PSC[4:0]	ADC 外部触发 5 预分频 该位域选择 ADC 外部触发 5 预分频

HRTIMER 故障输入配置寄存器 2 (HRTIMER_FLTINCFG2)

地址偏移：0x88

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLT3RST	FLT3CNTRST	FLT3CNT[3:0]				FLT3BLKS	FLT3BLKEN	FLT2RST	FLT2CNTRST	FLT2CNT[3:0]				FLT2BLKS	FLT2BLKEN
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
FLT1RST	FLT1CNTRST	FLT1CNT[3:0]				FLT1BLKS	FLT1BLKEN	FLT0RST	FLT0CNTRST	FLT0CNT[3:0]				FLT0BLKS	FLT0BLKEN
rw	rw	rw				rw	rw	rw	rw	rw				rw	rw

位/位域	名称	描述
31	FLT3RST	故障 3 复位模式 请参考FLT0RST描述。
30	FLT3CNTRST	故障 3 计数器复位 请参考 FLT0CNTRST 描述。
29:26	FLT3CNT[3:0]	故障 3 计数器 请参考 FLT0CNT[3:0]描述。
25	FLT3BLKS	故障 3 消隐源 请参考FLT0BLKS描述。
24	FLT3BLKEN	故障 3 消隐使能 请参考 FLT0BLKEN 描述

23	FLT2RST	故障 2 复位模式 请参考 FLT0RST 描述。
22	FLT2 CNTRST	故障 2 计数器复位 请参考 FLT0CNTRST 描述。
21:18	FLT2CNT[3:0]	故障 2 计数器 请参考FLT0CNT[3:0]描述。
17	FLT2BLKS	故障 2 消隐源 请参考 FLT0BLKS 描述。
16	FLT2BLKEN	故障 2 消隐使能 请参考 FLT0BLKEN 描述
15	FLT1RST	故障 1 复位模式 请参考 FLT0RST 描述。
14	FLT1CNTRST	故障 1 计数器复位 请参考 FLT0CNTRST 描述。
13:10	FLT1CNT[3:0]	故障 1 计数器 请参考 FLT0CNT[3:0]描述。
9	FLT1BLKS	故障 1 消隐源 请参考 FLT0BLKS 描述。
8	FLT1BLKEN	故障 1 消隐使能 请参考 FLT0BLKEN 描述
7	FLT0RST	故障 0 复位模式： 此位指定了故障 0 计数器的复位模式。 0: 在每次复位/翻转事件上复位故障 0 计数器。 1: 当 HRTIMER_FLTRECCTL 寄存器中的 FLTRECCTL 位置 0 时，故障计数器仅在没有故障事件处于有效状态时，每次复位/翻转事件时复位。当 HRTIMER_FLTRECCTL 寄存器中的 FLTRECCTL 位置 1 时，故障计数器仅在一个周期内没有发生故障事件时，每次复位/翻转事件时复位。 注意：此配置仅在 FLT0INEN 使能位处于复位状态时生效。
6	FLT0CNTRST	故障 0 计数器复位 此位用于复位故障 0 计数器，由软件设置并由硬件复位。 0: 无操作 1: 故障 0 计数器复位
5:2	FLT0CNT[3:0]	故障 0 计数器 此位域确定故障 0 计数器的阈值。当事件计数等于(FLT0CNT[3:0]+1)时，故障被视

为有效。

1	FLT0BLKS	故障 0 消隐源 FTL0BLKS 位选择了故障 0 的消隐源。 0: 故障 0 复位和比较窗口 1: 故障 0 比较和比较窗口 注意：仅在 FLT0INEN 使能位被复位时，才可配置此位域。
0	FLT0BLKEN	故障 0 消隐使能 FTL0BLKEN 位选择故障 0 的消隐模式。消隐源由 FLT0BLKS 位定义。 0: 故障 0 不进行消隐 1: 故障 0 消隐模式 **注意：**仅当 FLT0INEN 使能位被清除时，才写入此位域。

HRTIMER 故障输入配置寄存器 3 (HRTIMER_FLTINCFG3)

地址偏移：0x8C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLT7RST	FLT7CNTRST	FLT7CNT[3:0]				FLT7BLKS	FLT7BLKEN	FLT6RST	FLT6CNTRST	FLT6CNT[3:0]				FLT6BLKS	FLT6BLKEN
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
FLT5RST	FLT5CNTRST	FLT5CNT[3:0]				FLT5BLKS	FLT5BLKEN	FLT4RST	FLT4CNTRST	FLT4CNT[3:0]				FLT4BLKS	FLT4BLKEN
rw	rw	rw				rw	rw	rw	rw	rw				rw	rw

位/位域	名称	描述
31	FLT7RST	故障 7 复位模式 请参考FLT0RST描述
30	FLT7CNTRST	故障 7 计数器复位 请参考 FLT0CNTRST 描述
29:26	FLT7CNT[3:0]	故障 7 计数器 请参考 FLT0CNT[3:0]描述
25	FLT7BLKS	故障 7 消隐源 请参考FLT0BLKS描述
24	FLT7BLKEN	故障 7 消隐使能 请参考 FLT0BLKEN 描述
23	FLT6RST	故障 6 复位模式 请参考 FLT0RST 描述
22	FLT6 CNTRST	故障 6 计数器复位 请参考 FLT0CNTRST 描述
21:18	FLT6CNT[3:0]	故障 6 计数器 请参考FLT0CNT[3:0]描述
17	FLT6BLKS	故障 6 消隐源 请参考 FLT0BLKS 描述。
16	FLT6BLKEN	故障 6 消隐使能 请参考 FLT0BLKEN 描述

15	FLT5RST	故障 5 复位模式 请参考 FLT0RST 描述。
14	FLT5CNTRST	故障 5 计数器复位 请参考 FLT0CNTRST 描述
13:10	FLT5CNT[3:0]	故障 5 计数器 请参考 FLT0CNT[3:0]描述
9	FLT5BLKS	故障 5 消隐源 请参考 FLT0BLKS 描述
8	FLT5BLKEN	故障 5 消隐使能 请参考 FLT0BLKEN 描述
7	FLT4RST	故障 4 复位模式 请参考 FLT0RST 描述。
6	FLT4CNTRST	故障 4 计数器复位 请参考 FLT0CNTRST 描述
5:2	FLT0CNT[3:0]	故障 4 计数器 请参考 FLT0CNT[3:0]描述
1	FLT0BLKS	故障 4 消隐源 请参考 FLT0BLKS 描述
0	FLT0BLKEN	故障 4 消隐使能 请参考 FLT0BLKEN 描述

HRTIMER 突发模式启动触发附加寄存器 (HRTIMER_BMSTRGA)

地址偏移：0x108

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				ST7CMP1	ST7CMP0	ST7REP	ST7RST	ST6CMP1	ST6CMP0	ST6REP	ST6RST	ST5CMP1	ST5CMP0	ST5REP	ST5RST
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述

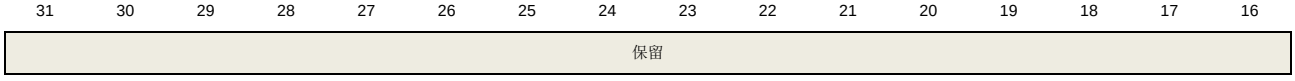
31:12	保留	必须保持复位值
11	ST7CMP1	Slave_TIMER7 比较 1 事件触发突发模式。 请参考 MTCMP1 模式。
10	ST7CMP0	Slave_TIMER7 比较 0 事件触发突发模式。 请参考 MTCMP0 模式。
9	ST7REP	Slave_TIMER7 重复事件触发突发模式。 请参考 MTREP 模式。
8	ST7RST	Slave_TIMER7 复位事件触发突发模式。 请参考 MTRST 模式。
7	ST6CMP1	Slave_TIMER6 比较 1 事件触发突发模式。 请参考 MTCMP1 模式。
6	ST6CMP0	Slave_TIMER6 比较 0 事件触发突发模式。 请参考 MTCMP0 模式。
5	ST6REP	Slave_TIMER6 重复事件触发突发模式。 请参考 MTREP 模式。
4	ST6RST	Slave_TIMER6 复位事件触发突发模式。 请参考 MTRST 模式。
3	ST5CMP1	Slave_TIMER5 比较 1 事件触发突发模式。 请参考 MTCMP1 模式。
2	ST5CMP0	Slave_TIMER5 比较 0 事件触发突发模式。 请参考 MTCMP0 模式。
1	ST5REP	Slave_TIMER5 重复事件触发突发模式。 请参考 MTREP 模式。
0	ST5RST	Slave_TIMER5 复位事件触发突发模式。 请参考 MTRST 模式。

HRTIMER 故障输入配置寄存器 4 (HRTIMER_FLTINCFG4)

地址偏移：0x10C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLT7INPROT	FLT7INFC[3:0]				FLT7INSRC[0]	FLT7INP	FLT7INEN	FLT6INPROT	FLT6INFC[3:0]				FLT6INSRC[0]	FLT6INP	FLT6INEN
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	FLT7INPROT	保护故障 7 输入配置 请参考 FLT0INPROT 描述。
14:11	FLT7INFC[3:0]	故障 7 输入滤波控制 请参考 FLT0INFC[3:0]描述。
10	FLT7INSRC[0]	故障 7 输入源 请参考 FLT0INSRC 描述。
9	FLT7INP	故障 7 输入极性 请参考 FLT0INP 描述。
8	FLT7INEN	故障 7 输入使能 请参考 FLT0INEN 描述。
7	FLT6INPROT	保护故障 6 输入配置 请参考 FLT0INPROT 描述。
6:3	FLT6INFC[3:0]	故障 6 输入滤波控制 请参考 FLT0INFC[3:0]描述。
2	FLT6INSRC[0]	故障 6 输入源 请参考 FLT0INSRC 描述。
1	FLT6INP	故障 6 输入极性 请参考 FLT0INP 描述。
0	FLT6INEN	故障 6 输入使能 请参考 FLT0INEN 描述。

EXEVyFM HRTIMER ADC 外部触发附加寄存器 (HRTIMER_ADCEXTTRGA)

地址偏移：0x110

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resreved										ADC9TRG[5]	ADC8TRG[5]	ADC7TRG[5]	ADC6TRG[5]	ADC5TRG[5]	ADC4TRG[5]
										rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31: 6	保留	必须保持复位值
5	ADC9TRG[5]	ADC 触发源 9 选择，该位域选择 ADC 触发源 9 请参考 ADC4TRG[4:0]描述。
4	ADC8TRG[5]	ADC 触发源 8 选择，该位域选择 ADC 触发源 8 请参考ADC4TRG[4:0]描述。
3	ADC7TRG[5]	ADC 触发源 7 选择，该位域选择 ADC 触发源 7 请参考ADC4TRG[4:0]描述。
2	ADC6TRG[5]	ADC 触发源 6 选择，该位域选择 ADC 触发源 6 请参考 ADC4TRG[4:0]描述。
1	ADC5TRG[5]	ADC 触发源 5 选择，该位域选择 ADC 触发源 5 请参考 ADC4TRG[4:0]描述。
0	ADC4TRG[5]	ADC 触发源 4 选择，该位域选择 ADC 触发源 4

HRTIMER ADC 触发源 0 附加寄存器 (HRTIMER_ADCTRIGS0A)

地址偏移：0x114

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Resreved											TRG0ST7RST	TRG0ST7PER	TRG0ST7C3	TRG0ST7C2	TRG0ST7C1
											rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resreved			TRG0ST6RST	TRG0ST6PER	TRG0ST6C3	TRG0ST6C2	TRG0ST6C1	Resreved			TRG0ST5RST	TRG0ST5PER	TRG0ST5C3	TRG0ST5C2	TRG0ST5C1
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位/位域	名称	描述
31:21	保留	必须保持复位值

20	TRG0ST7RST	Slave_TIMER7复位事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置生成ADC触发事件。 0：HRTIMER Slave_TIMER7复位事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER7复位事件生成ADC触发事件。
19	TRG0ST7PER	Slave_TIMER7周期事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置周期事件生成ADC触发事件。 0：HRTIMER Slave_TIMER7周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER7周期事件生成ADC触发事件。
18	TRG0ST7C3	Slave_TIMER7比较3事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置比较1事件是否生成ADC触发事件。 0：HRTIMER Slave_TIMER7比较3事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER7 比较 3 事件生成 ADC 触发事件。
17	TRG0ST7C2	Slave_TIMER7比较2事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置比较1事件是否生成ADC触发事件。 0：HRTIMER Slave_TIMER7比较2事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER7 比较 2 事件生成 ADC 触发事件。
16	TRG0ST7C1	Slave_TIMER7比较1事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置比较1事件生成ADC触发事件。 0：HRTIMER Slave_TIMER7比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER7比较1事件生成ADC触发事件。
15:13	保留	必须保持复位值
12	TRG0ST6RST	Slave_TIMER6复位事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置生成ADC触发事件。 0：HRTIMER Slave_TIMER6复位事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER6复位事件生成ADC触发事件。
11	TRG0ST6PER	Slave_TIMER6周期事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置周期事件生成ADC触发事件。 0：HRTIMER Slave_TIMER6周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER6周期事件生成ADC触发事件。

10	TRG0ST6C3	Slave_TIMER6比较3事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置比较1事件是否生成ADC触发事件。 0：HRTIMER Slave_TIMER6比较3事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER6 比较 3 事件生成 ADC 触发事件。
9	TRG0ST6C2	Slave_TIMER6比较2事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置比较1事件是否生成ADC触发事件。 0：HRTIMER Slave_TIMER6比较2事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER6 比较 2 事件生成 ADC 触发事件。
8	TRG0ST6C1	Slave_TIMER6比较1事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置比较1事件生成ADC触发事件。 0：HRTIMER Slave_TIMER6比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER6比较1事件生成ADC触发事件。
7:5	保留	必须保持复位值
4	TRG0ST5RST	Slave_TIMER5复位事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置生成ADC触发事件。 0：HRTIMER Slave_TIMER5复位事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER5 复位事件生成 ADC 触发事件。
3	TRG0ST5PER	Slave_TIMER5周期事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置周期事件生成ADC触发事件。 0：HRTIMER Slave_TIMER5周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER5 周期事件生成 ADC 触发事件。
2	TRG0ST5C3	Slave_TIMER5比较3事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置比较1事件是否生成ADC触发事件。 0：HRTIMER Slave_TIMER5比较3事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER5 比较 3 事件生成 ADC 触发事件。
1	TRG0ST5C2	Slave_TIMER5比较2事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置比较1事件是否生成ADC触发事件。 0：HRTIMER Slave_TIMER5比较2事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER5比较2事件生成ADC触发事件。

0	TRG0ST5C1	Slave_TIMER5比较1事件生成HRTIMER_ADCTRIG0上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG0上生成ADC触发事件。该位用于配置比较1事件生成ADC触发事件。 0：HRTIMER Slave_TIMER5比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER5比较1事件生成ADC触发事件。

HRTIMER ADC 触发源 1 附加寄存器 (HRTIMER_ADCTRIGS1A)

地址偏移：0x118

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Resreved											TRG1ST7RST	TRG1ST7PER	TRG1ST7C3	TRG1ST7C2	TRG1ST7C1
											rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resreved			TRG1ST6RST	TRG1ST6PER	TRG1ST6C3	TRG1ST6C2	TRG1ST6C1	Resreved			TRG1ST5RST	TRG1ST5PER	TRG1ST5C3	TRG1ST5C2	TRG1ST5C1
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位/位域	名称	描述
31:21	保留	必须保持复位值
20	TRG1ST7RST	Slave_TIMER7复位事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置生成ADC触发事件。 0：HRTIMER Slave_TIMER7复位事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER7复位事件生成ADC触发事件。
19	TRG1ST7PER	Slave_TIMER7周期事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置周期事件生成ADC触发事件。 0：HRTIMER Slave_TIMER7周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER7周期事件生成ADC触发事件。
18	TRG1ST7C3	Slave_TIMER7比较3事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置比较1事件是否生成ADC触发事件。 0：HRTIMER Slave_TIMER7比较3事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER7 比较 3 事件生成 ADC 触发事件。
17	TRG1ST7C2	Slave_TIMER7比较2事件生成HRTIMER_ADCTRIG1上的ADC触发事件

		HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置比较1事件是否生成ADC触发事件。 0：HRTIMER Slave_TIMER7比较2事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER7 比较 2 事件生成 ADC 触发事件。
16	TRG1ST7C1	Slave_TIMER7比较1事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置比较1事件生成ADC触发事件。 0：HRTIMER Slave_TIMER7比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER7比较1事件生成ADC触发事件。
15:13	保留	必须保持复位值
12	TRG1ST6RST	Slave_TIMER6复位事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置生成ADC触发事件。 0：HRTIMER Slave_TIMER6复位事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER6复位事件生成ADC触发事件。
11	TRG1ST6PER	Slave_TIMER6周期事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置周期事件生成ADC触发事件。 0：HRTIMER Slave_TIMER6周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER6周期事件生成ADC触发事件。
10	TRG1ST6C3	Slave_TIMER6比较3事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置比较1事件是否生成ADC触发事件。 0：HRTIMER Slave_TIMER6比较3事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER6 比较 3 事件生成 ADC 触发事件。
9	TRG1ST6C2	Slave_TIMER6比较2事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置比较1事件是否生成ADC触发事件。 0：HRTIMER Slave_TIMER6比较2事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER6 比较 2 事件生成 ADC 触发事件。
8	TRG1ST6C1	Slave_TIMER6比较1事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置比较1事件生成ADC触发事件。 0：HRTIMER Slave_TIMER6比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER6比较1事件生成ADC触发事件。
7:5	保留	必须保持复位值

4	TRG1ST5RST	Slave_TIMER5复位事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置生成ADC触发事件。 0：HRTIMER Slave_TIMER5复位事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER5 复位事件生成 ADC 触发事件。
3	TRG1ST5PER	Slave_TIMER5周期事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置周期事件生成ADC触发事件。 0：HRTIMER Slave_TIMER5周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER5 周期事件生成 ADC 触发事件。
2	TRG1ST5C3	Slave_TIMER5比较3事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置比较1事件是否生成ADC触发事件。 0：HRTIMER Slave_TIMER5比较3事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER5 比较 3 事件生成 ADC 触发事件。
1	TRG1ST5C2	Slave_TIMER5比较2事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置比较1事件是否生成ADC触发事件。 0：HRTIMER Slave_TIMER5比较2事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER5比较2事件生成ADC触发事件。
0	TRG1ST5C1	Slave_TIMER5比较1事件生成HRTIMER_ADCTRIG1上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG1上生成ADC触发事件。该位用于配置比较1事件生成ADC触发事件。 0：HRTIMER Slave_TIMER5比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER5比较1事件生成ADC触发事件。

HRTIMER ADC 触发源 2 附加寄存器 (HRTIMER_ADCTRIGS2A)

地址偏移：0x11C

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Resreved											TRG2ST7 RST	TRG2ST7 PER	TRG2ST7 C3	TRG2ST7 C2	TRG2ST7 C1
											rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resreved			TRG2ST6 RST	TRG2ST6 PER	TRG2ST6 C3	TRG2ST6 C2	TRG2ST6 C1	Resreved			TRG2ST5 RST	TRG2ST5 PER	TRG2ST5 C3	TRG2ST5 C2	TRG2ST5 C1

rw rw rw rw rw rw rw rw rw rw

位/位域	名称	描述
31:21	保留	必须保持复位值
20	TRG2ST7RST	Slave_TIMER7复位事件生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置生成ADC触发事件。 0：HRTIMER Slave_TIMER7复位事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER7复位事件生成ADC触发事件。
19	TRG2ST7PER	Slave_TIMER7周期事件生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置周期事件生成ADC触发事件。 0：HRTIMER Slave_TIMER7周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER7周期事件生成ADC触发事件。
18	TRG2ST7C3	Slave_TIMER7比较3事件生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置比较1事件是否生成ADC触发事件。 0：HRTIMER Slave_TIMER7比较3事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER7 比较 3 事件生成 ADC 触发事件。
17	TRG2ST7C2	Slave_TIMER7比较2事件生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置比较1事件是否生成ADC触发事件。 0：HRTIMER Slave_TIMER7比较2事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER7 比较 2 事件生成 ADC 触发事件。
16	TRG2ST7C1	Slave_TIMER7比较1事件生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置比较1事件生成ADC触发事件。 0：HRTIMER Slave_TIMER7比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER7比较1事件生成ADC触发事件。
15:13	保留	必须保持复位值
12	TRG2ST6RST	Slave_TIMER6复位事件生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置生成ADC触发事件。 0：HRTIMER Slave_TIMER6复位事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER6复位事件生成ADC触发事件。
11	TRG2ST6PER	Slave_TIMER6周期事件生成HRTIMER_ADCTRIG2上的ADC触发事件

		HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置周期事件生成ADC触发事件。 0：HRTIMER Slave_TIMER6周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER6周期事件生成ADC触发事件。
10	TRG2ST6C3	Slave_TIMER6比较3事件生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置比较1事件是否生成ADC触发事件。 0：HRTIMER Slave_TIMER6比较3事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER6 比较 3 事件生成 ADC 触发事件。
9	TRG2ST6C2	Slave_TIMER6比较2事件生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置比较1事件是否生成ADC触发事件。 0：HRTIMER Slave_TIMER6比较2事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER6 比较 2 事件生成 ADC 触发事件。
8	TRG2ST6C1	Slave_TIMER6比较1事件生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置比较1事件生成ADC触发事件。 0：HRTIMER Slave_TIMER6比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER6比较1事件生成ADC触发事件。
7:5	保留	必须保持复位值
4	TRG2ST5RST	Slave_TIMER5复位事件生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置生成ADC触发事件。 0：HRTIMER Slave_TIMER5复位事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER5 复位事件生成 ADC 触发事件。
3	TRG2ST5PER	Slave_TIMER5周期事件生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置周期事件生成ADC触发事件。 0：HRTIMER Slave_TIMER5周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER5 周期事件生成 ADC 触发事件。
2	TRG2ST5C3	Slave_TIMER5比较3事件生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置比较1事件是否生成ADC触发事件。 0：HRTIMER Slave_TIMER5比较3事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER5 比较 3 事件生成 ADC 触发事件。
1	TRG2ST5C2	Slave_TIMER5比较2事件生成HRTIMER_ADCTRIG2上的ADC触发事件

		HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置比较1事件是否生成ADC触发事件。 0：HRTIMER Slave_TIMER5比较2事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER5比较2事件生成ADC触发事件。
0	TRG2ST5C1	Slave_TIMER5比较1事件生成HRTIMER_ADCTRIG2上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置比较1事件生成ADC触发事件。 0：HRTIMER Slave_TIMER5比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER5比较1事件生成ADC触发事件。

HRTIMER ADC 触发源 3 附加寄存器 (HRTIMER_ADCTRIGS3A)

地址偏移：0x120

复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Resreved											TRG3ST7RST	TRG3ST7PER	TRG3ST7C3	TRG3ST7C2	TRG3ST7C1
											rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resreved			TRG3ST6RST	TRG3ST6PER	TRG3ST6C3	TRG3ST6C2	TRG3ST6C1	Resreved			TRG3ST5RST	TRG3ST5PER	TRG3ST5C3	TRG3ST5C2	TRG3ST5C1
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位/位域	名称	描述
31:21	保留	必须保持复位值
20	TRG3ST7RST	Slave_TIMER7复位事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置生成ADC触发事件。 0：HRTIMER Slave_TIMER7复位事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER7复位事件生成ADC触发事件。
19	TRG3ST7PER	Slave_TIMER7周期事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置周期事件生成ADC触发事件。 0：HRTIMER Slave_TIMER7周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER7周期事件生成ADC触发事件。
18	TRG3ST7C3	Slave_TIMER7比较3事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG2上生成ADC触发事件。该位用于配置比较1

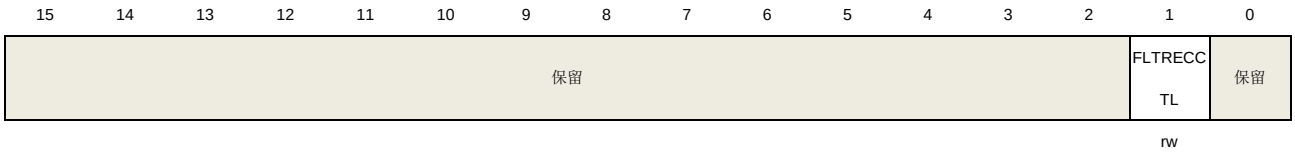
		事件是否生成ADC触发事件。 0：HRTIMER Slave_TIMER7比较3事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER7 比较 3 事件生成 ADC 触发事件。
17	TRG3ST7C2	Slave_TIMER7比较2事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置比较1事件是否生成ADC触发事件。 0：HRTIMER Slave_TIMER7比较2事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER7 比较 2 事件生成 ADC 触发事件。
16	TRG3ST7C1	Slave_TIMER7比较1事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置比较1事件生成ADC触发事件。 0：HRTIMER Slave_TIMER7比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER7比较1事件生成ADC触发事件。
15:13	保留	必须保持复位值
12	TRG3ST6RST	Slave_TIMER6复位事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置生成ADC触发事件。 0：HRTIMER Slave_TIMER6复位事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER6复位事件生成ADC触发事件。
11	TRG3ST6PER	Slave_TIMER6周期事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置周期事件生成ADC触发事件。 0：HRTIMER Slave_TIMER6周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER6周期事件生成ADC触发事件。
10	TRG3ST6C3	Slave_TIMER6比较3事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置比较1事件是否生成ADC触发事件。 0：HRTIMER Slave_TIMER6比较3事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER6 比较 3 事件生成 ADC 触发事件。
9	TRG3ST6C2	Slave_TIMER6比较2事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置比较1事件是否生成ADC触发事件。 0：HRTIMER Slave_TIMER6比较2事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER6 比较 2 事件生成 ADC 触发事件。
8	TRG3ST6C1	Slave_TIMER6比较1事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置比较1

		事件生成ADC触发事件。 0：HRTIMER Slave_TIMER6比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER6比较1事件生成ADC触发事件。
7:5	保留	必须保持复位值
4	TRG3ST5RST	Slave_TIMER5复位事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置生成ADC触发事件。 0：HRTIMER Slave_TIMER5复位事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER5 复位事件生成 ADC 触发事件。
3	TRG3ST5PER	Slave_TIMER5周期事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置周期事件生成ADC触发事件。 0：HRTIMER Slave_TIMER5周期事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER5 周期事件生成 ADC 触发事件。
2	TRG3ST5C3	Slave_TIMER5比较3事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置比较1事件是否生成ADC触发事件。 0：HRTIMER Slave_TIMER5比较3事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER5 比较 3 事件生成 ADC 触发事件。
1	TRG3ST5C2	Slave_TIMER5比较2事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置比较1事件是否生成ADC触发事件。 0：HRTIMER Slave_TIMER5比较2事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER5比较2事件生成ADC触发事件。
0	TRG3ST5C1	Slave_TIMER5比较1事件生成HRTIMER_ADCTRIG3上的ADC触发事件 HRTIMER可以在HRTIMER_ADCTRIG3上生成ADC触发事件。该位用于配置比较1事件生成ADC触发事件。 0：HRTIMER Slave_TIMER5比较1事件不生成ADC触发事件。 1：HRTIMER Slave_TIMER5比较1事件生成ADC触发事件。

HRTIMER 故障恢复控制寄存器 (HRTIMER_FLTRECCTL)

地址偏移：0x17C
复位值：0x0000 0000

该寄存器只能进行字（32 位）访问。



Bits	Fields	Descriptions
31:2	保留	必须保持复位值
1	FLTRECCTL	故障恢复控制 0：故障计数器仅在没有故障事件处于有效状态时，每次复位/翻转事件时复位。 1：故障计数器仅在一个周期内没有发生故障事件时，每次复位/翻转事件时复位。 **注意：**该位仅在 FLTxRST 位置 1 时才会生效。
0	保留	必须保持复位值

26. 红外接口（IFRP）

26.1. 简介

红外接口(IFRP)用于控制红外光 LED，并发送红外数据实现红外遥控。

该模块没有寄存器，由 TIMER15 和 TIMER16 控制。通过将 GPIO 引脚设置为快速模式，可以将模块的输出提高到高电流容量。

26.2. 主要特性

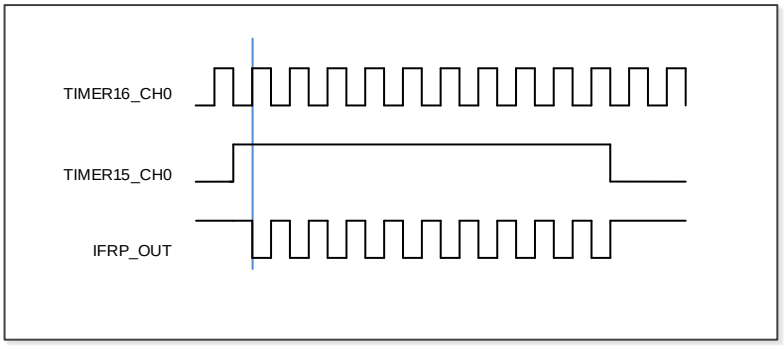
- IFRP 输出信号是由 TIMER15_CH0 和 TIMER16_CH0 决定；
- 为了得到正确的红外线信号，TIMER15 应产生低频调制包络信号，TIMER16 应产生高频载波信号；
- 通过在 SYSCFG_CFG0 中设置 PB9FMPEN，IFRP 输出（PB9）可以控制 LED 接口。

26.3. 功能描述

IFRP 能够集成 TIMER15 和 TIMER16 的输出，以生成红外线信号。

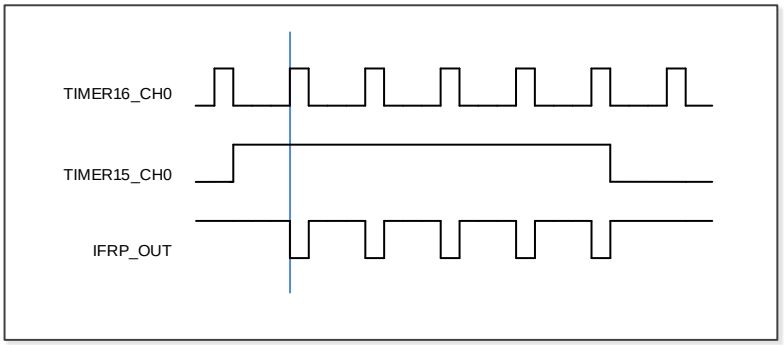
1. 对 TIMER15 的 CH0 进行编程，产生低频 PWM 信号，即调制值信号。对 TIMER16 的 CH0 进行编程，生成高频率 PWM 信号，即载波信号。在产生这些信号之前，信道需要被激活。
2. 设置 GPIO 重映射寄存器并使能 PIN。

图 26-1. IFRP 输出时序图 1



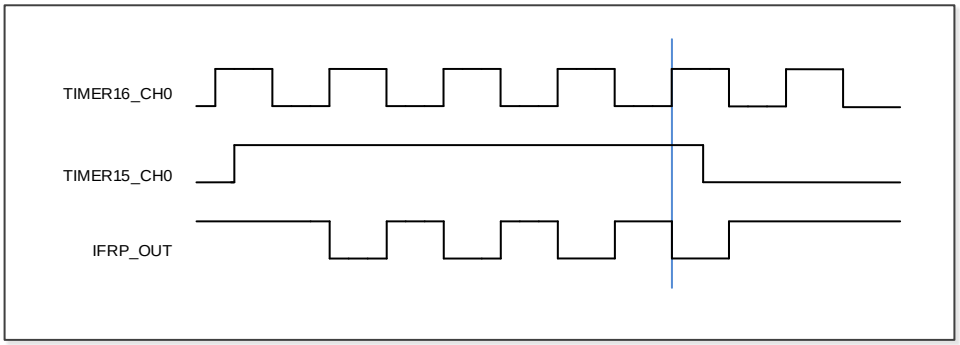
注意: IFRP_OUT与TIMER16_CH0相比有一个APB时钟延迟。

图 26-2. IFRP 输出时序图 2



注意: 载波(TIMER15_CH0)的占空比可以改变，当TIMER15_CH0占空比较高时，IFRP_OUT与TIMER16_CH0呈反向关系。

图 26-3. IFRP 输出时序图 3



注意: IFRP_OUT将保持TIMER16_CH0的完整性，即使包络信号(TIMER15_CH0)不是活动的。

27. 通用同步异步收发器（USART）

27.1. 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（CK_APBx，CK_AHB，CK_LXTAL或CK_IRC8M）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。

27.2. 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/发送FIFO功能；
- 双时钟域：
  - 互为异步关系的PCLK和USART时钟；
  - 不依赖PCLK设置的波特率设置。
- 可编程的波特率产生器，当时钟频率为216MHz，过采样为8，最高速度可达27MBits/s；
- 完全可编程的串口特性：
  - 数据位（7、8、9或10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；
- 可配置的多级缓存通信DMA访问数据缓冲区；
- 发送器和接收器可分别使能；
- 奇偶校验位控制：
  - 发送奇偶校验位；
  - 检测接收的数据字节的奇偶校验位。
- LIN断开帧的产生和检测；
- 支持红外数据协议（IrDA）；

- 同步传输模式以及为同步传输输出发送时钟；
- 支持兼容ISO7816-3的智能卡接口：
  - 字节模式（T = 0）；
  - 块模式（T = 1）；
  - 直接和反向转换。
- 多处理器通信：
  - 如果地址不匹配，则进入静默模式；
  - 通过线路空闲检测或者地址匹配检测从静默模式唤醒。
- 支持ModBus通信：
  - 超时功能；
  - CR / LF字符识别。
- 从深度睡眠模式唤醒：
  - 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），接收FIFO为空（RFE），接收FIFO达到阈值（RFT），接收FIFO非空（RFNE），发送缓冲区为空（TBE），传输完成（TC），发送FIFO不满（TFNF），发送FIFO空（TFE），发送FIFO到达阈值（TFTIF）；
  - 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、USART2完全实现上述功能，但是UART3 / UART4只实现了上面所介绍的部分功能，下面这些功能在UART3 / UART4中没有实现：

- 智能卡模式；
- IrDA SIR ENDEC模块；
- LIN模式；
- 双时钟域和从深度睡眠模式唤醒；
- 接收超时中断；
- ModBus通信；
- 同步模式。

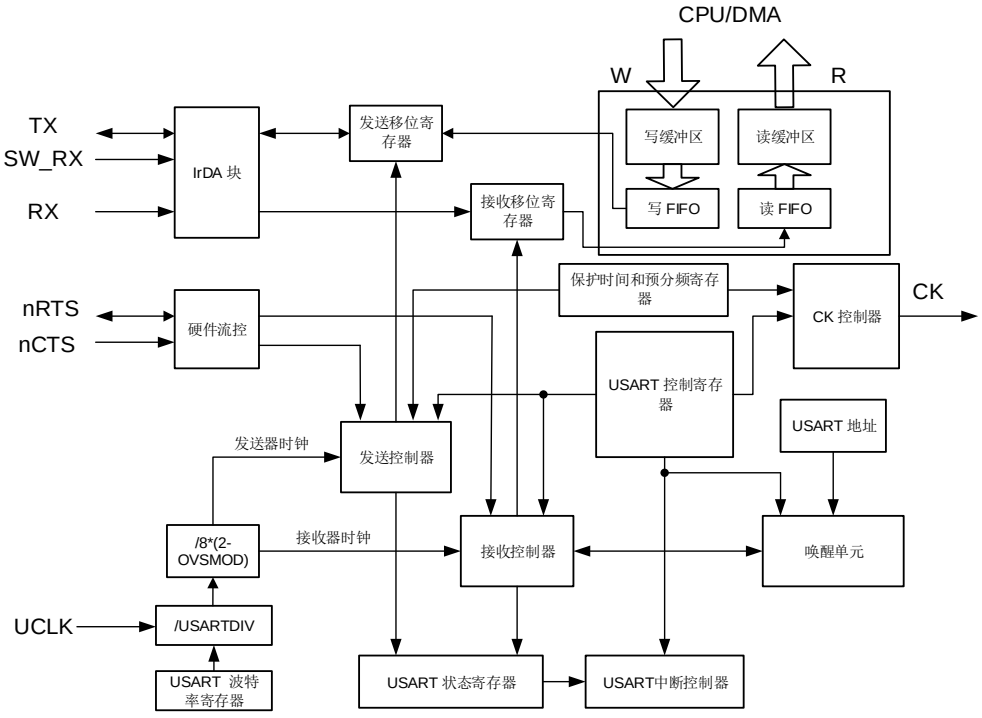
27.3. 功能说明

USART 接口通过*表 27-1. USART 重要引脚描述*中主要引脚从外部连接到其他设备。

表 27-1. 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

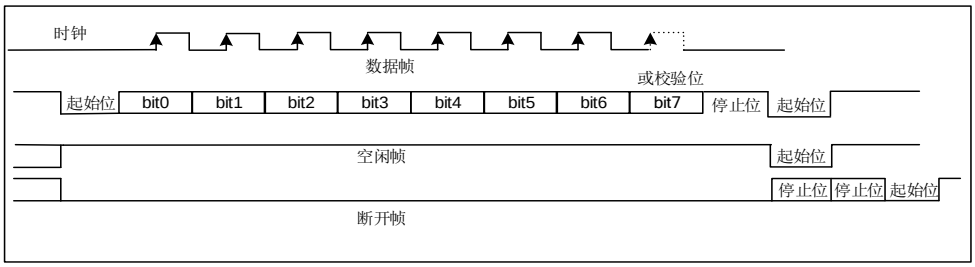
图 27-1. USART 模块内部框图



27.3.1. 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WL0和WL1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WL1位为0且WL0位为0，第七位为校验位。若WL1位为0且WL0位置1，第八位为校验位。若WL1位置1且WL0位为0，第六位位校验位。若WL1位置1且WL0位置1，第九位为校验位。USART_CTL0寄存器中PM位用于选择校验位的计算方法。

图 27-2. 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 27-2. 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。

27.3.2. 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UCLK有如下关系：

如果过采样率是16，公式为：

$$\text{USARTDIV} = \frac{\text{UCLK}}{16 \times \text{Baud Rate}} \quad (27\text{-}1)$$

如果过采样是8，公式为：

$$\text{USARTDIV} = \frac{\text{UCLK}}{8 \times \text{Baud Rate}} \quad (27\text{-}2)$$

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。

27.3.3. 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。

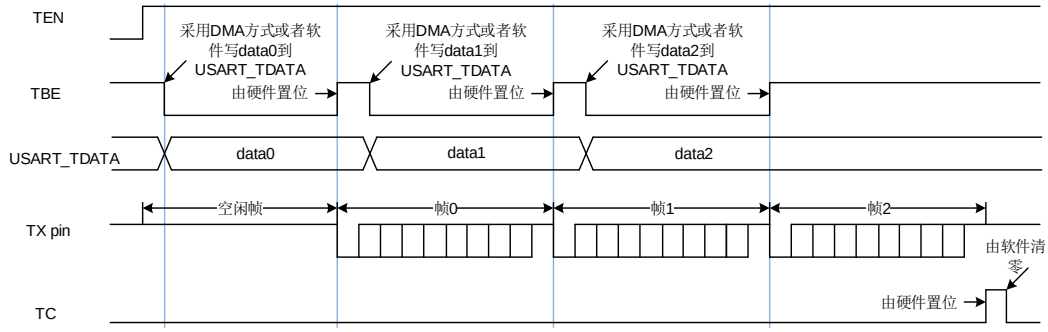
系统上电后，TBE默认为高电平。在USART_STAT寄存器中TBE置位时，数据可以在不覆盖前一个数据的情况下写入USART_TDATA寄存器。当数据写入USART_TDATA寄存器，TBE位将被清0。在数据由USART_TDATA移入移位寄存器后，该位由硬件置1。如果数据在一个发送过程正在进行时被写入USART_TDATA寄存器，它将首先被存入发送缓冲区，在当前发送过程完成时传输到发送移位寄存器中。如果数据在写入USART_TDATA寄存器时，没有发送过程正在进行，TBE位将被清零然后迅速置位，原因是数据被立刻传输到发送移位寄存器。

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 27-3. 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 27-3. 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。通过向USART_INTC寄存器的TCC位写1可将TC位清0。

当SBKCMD置位时，会发送一个断开帧，发送完成后，SBKCMD将被清0。

27.3.4. 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。

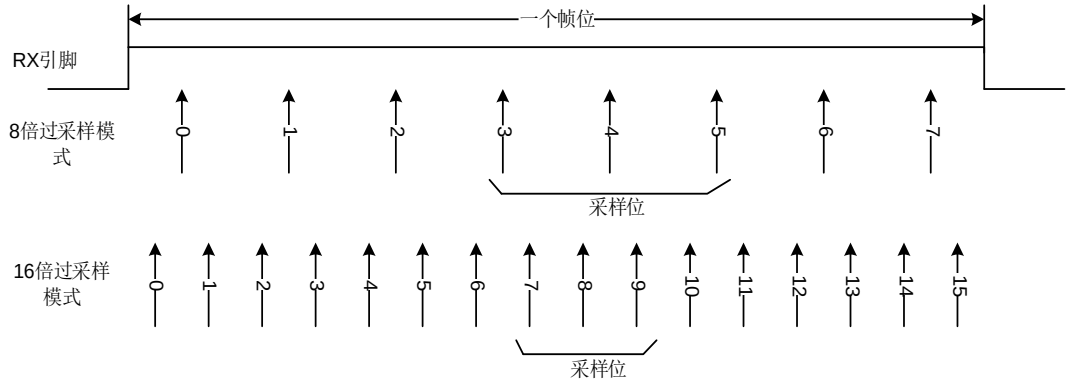
当接收到一个数据帧，USART_STAT寄存器中的RBNE置位，如果设置了USART_CTL0寄存器中相应的中断使能位RBNEIE，将会产生中断。在USART_STAT寄存器中可以观察接收状态标志。

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1、2、3个采样点；如果是16倍过采样模式，选择第6、7、8个采样点。如果在3个采样点中有2个或3个为0，该数据位被视为0，否则为1。如果3个采样点中有一个采样点的值与其他两个不同，不管是数据位，奇偶校验位或者停止位，都将产生噪声错误（NERR）。如果使能DMA，并置位USART_CTL2寄存器中ERRIE，将会产生中断。如果在USART_CTL2中置位OSB，接收器将仅获取一个采样点来估计一个数据位的值。在这种情况下将不会检测到噪声错误。

图 27-4. 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。如果使能DMA并置位USART_CTL2寄存器中ERRIE位或者置位RBNEIE，将产生中断。

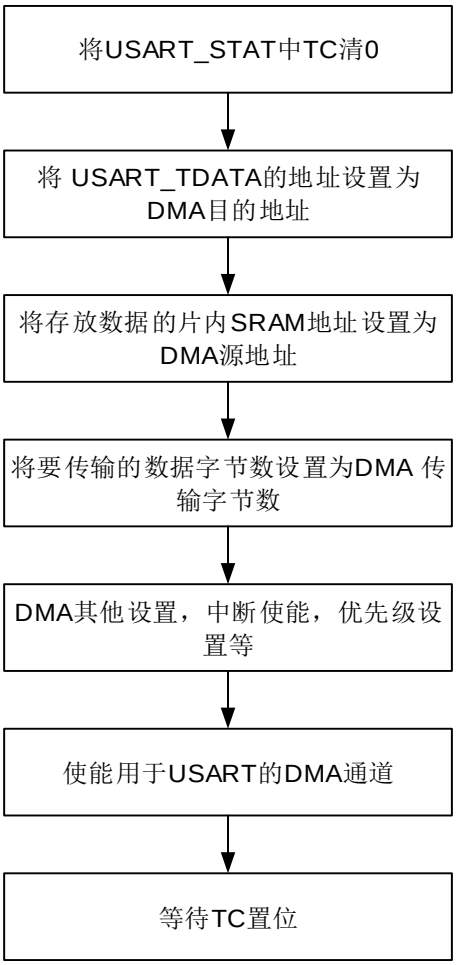
若接收过程中，产生了噪声错误（NERR）、校验错误（PERR）、帧错误（FERR）或溢出错误（ORERR），则NERR、PERR、FERR或ORERR将和RBNE同时置位。如果没有使能DMA，RBNE中断发生时，软件需检查是否有噪声错误、校验错误、帧错误或溢出错误产生。

27.3.5. 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 27-5. 采用 DMA 方式实现 USART 数据发送配置步骤*所示。

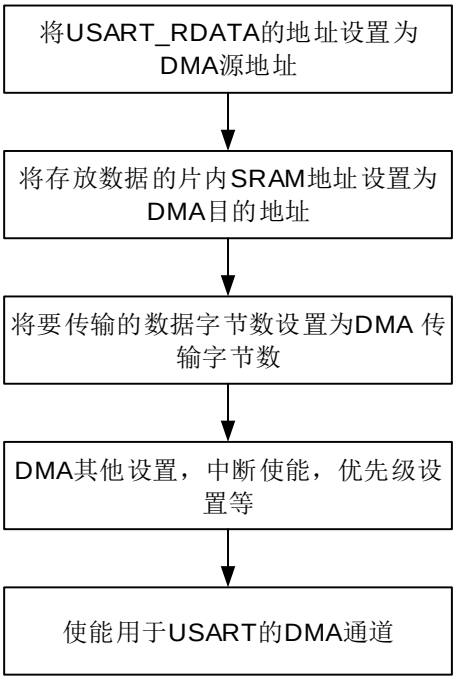
图 27-5. 采用 DMA 方式实现 USART 数据发送配置步骤



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 27-6. 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 27-6. 采用 DMA 方式实现 USART 数据接收配置步骤

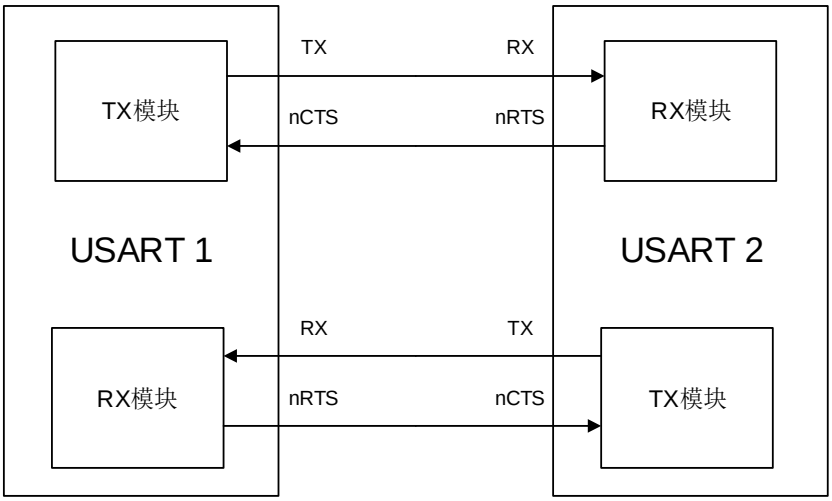


当USART接收到的数据数量达到了DMA传输数据数量，DMA模块将产生传输完成中断。

27.3.6. 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 27-7. 两个 USART 之间的硬件流控制



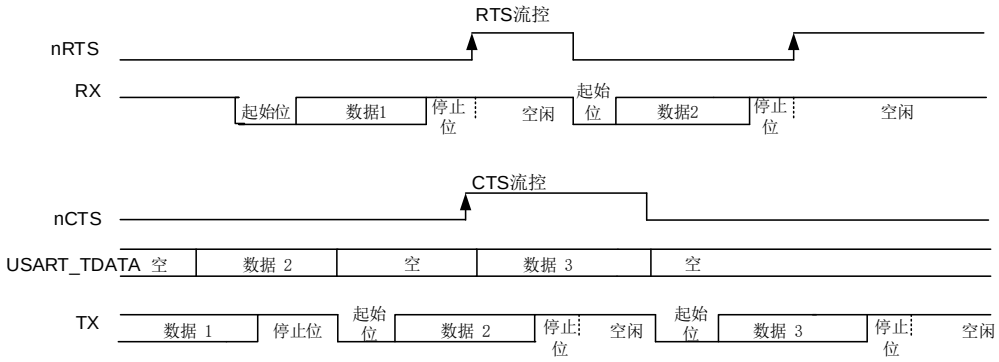
RTS 流控

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 27-8. 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。

27.3.7. 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被置位，数据最高位会被认为是地址标志位。如果地址标志位为1，该字节被认为是地址字节。如果地址标志位是0，该字节被认为是数据字节。通过USART_CTL1寄存器中ADDM0位或USART_CTL2寄存器中ADDM1位配置地址检测模式，如果地址字节的低4位或低7位与USART_CTL1寄存器中的ADDR0位或USART_CTL2寄存器中ADDR1位相同，硬件会将RWU清零，并退出静默模式。接收到将USART唤醒的数据帧，RBNE将置位。状态标志可以从USART_STAT寄存器中获取。如果地址字节的低4位或低7位与USART_CTL1寄存器中的ADDR0位或USART_CTL2寄存器中ADDR1位不相同，硬件会置位RWU并自动进入静默模式。在这种情况下，RBNE不会被置位。

如果USART_CTL0寄存器中PCEN位被置位，地址字节最高位被视为校验位，其余位被视为地址位。如果ADDM0或ADDM1位被置位，且接收帧为8位的数据，其中最低的7位将与ADDR0[6:0]或ADDR1[6:0]进行比较。如果ADDM0或ADDM1位被置位，且接收帧为9位的数据，其中最低的8位将与将与ADDR0[7:0]或ADDR1[7:0]进行比较。

ADDR0和ADDR1的匹配地址操作功能相同。当数据格式为8位和9位且启用奇偶校验时，地址标志的位置与校验位位置相同。

如果仅AMEN0或AMEN1被置位，则匹配地址仅与关联的ADDR0或ADDR1进行比较，且仅在地址匹配时将数据传输到接收数据缓冲区。

如果AMEN0和AMEN1均被置位，则匹配地址将与ADDR0和ADDR1进行比较，且数据仅在与ADDR0或ADDR1匹配时传输。因此，第二个匹配地址可作为串行总线的广播呼叫地址。

注意：如果MEN位被置位，且WM位和RWU位被复位，在RX引脚上检测到空闲帧时，IDLEF位将被置位。如果RWU被置位，则IDLEF不会被置位。此外，当使用地址匹配方法将USART从静默模式唤醒时，需要置位AMEN0或AMEN1。

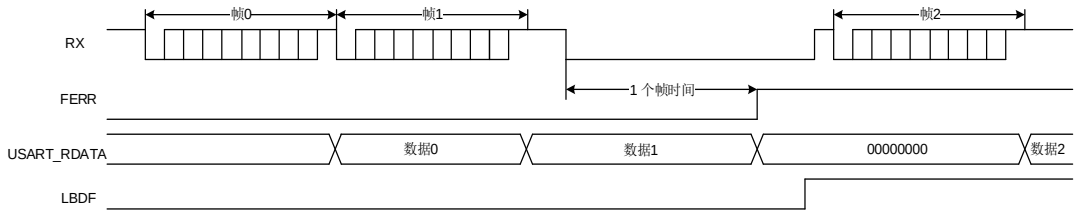
27.3.8. LIN 模式

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。

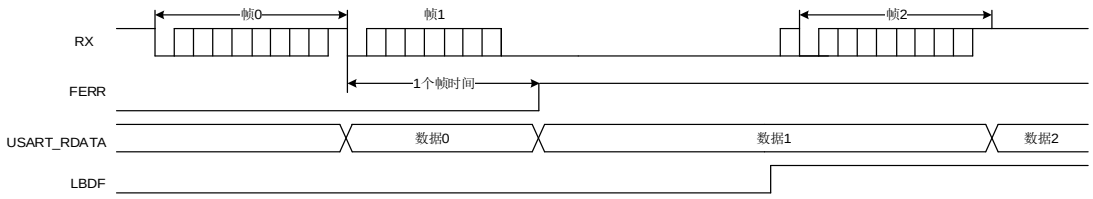
断开检测功能完全独立于普通USART接收器。因此，断开检测可以是在空闲状态下，也可以在数据传输过程中。USART_CTL1寄存器中LBLEN位可以选择断开帧的长度。如果在RX引脚检测到大于或等于与预期的断开帧长度的0（LBLEN = 0时，10个0；LBLEN = 1时，11个0），USART_STAT寄存器中LBDF置位。如果USART_CTL1寄存器中LBDIE被置位，将产生中断。如*图27-9. 空闲状态下检测断开帧*所示，如果断开帧发生在空闲状态下，USART接收器会接收到一个全0数据帧，同时FERR置位。

图 27-9. 空闲状态下检测断开帧



如*图 27-10. 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 27-10. 数据传输过程中检测断开帧



27.3.9. 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 27-11. 同步模式下的 USART 示例

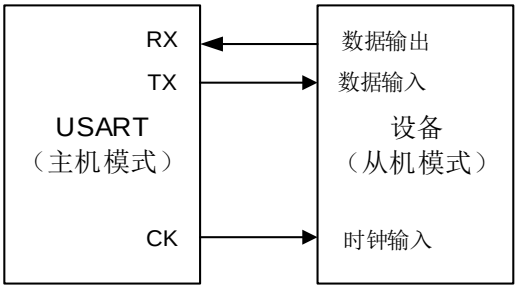
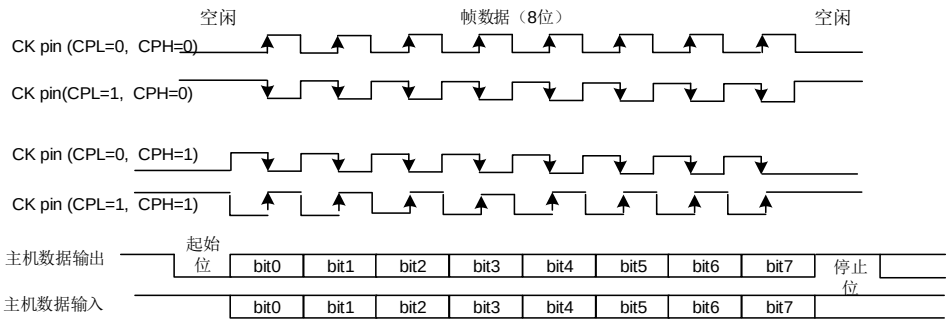


图 27-12. 8-bit 格式的 USART 同步通信波形（CLEN = 1）

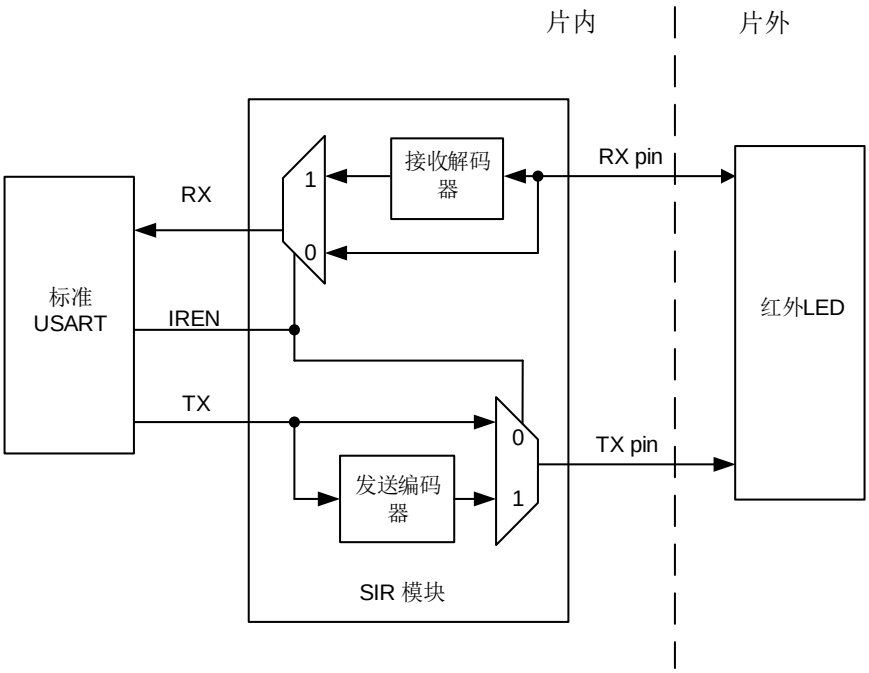


27.3.10. 串行红外（IrDA SIR）编解码功能模块

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发送译码器进行调制，调制后的信号经由红外LED进行发送，经解调后将数据发送至USART接收器。对于译码器而言，波特率应小于115200。

图 27-13. IrDA SIR ENDEC 模块

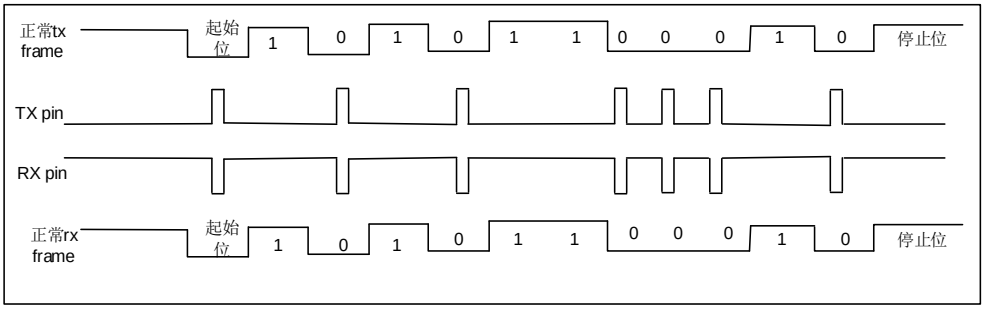


在IrDA模式下，TX引脚与RX引脚电平不同。TX引脚通常为低电平，RX引脚通常为高电平。IrDA引脚电平保持稳定代表逻辑‘1’，红外光源脉冲（RTZ信号）代表逻辑‘0’。其脉冲宽度通常占一个位时间的3/16。IrDA无法检测到宽度小于1个PSC时钟的脉冲。如果脉冲宽度大于1但是小于2倍

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 27-14. 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。发送译码器由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，USART_BAUD寄存器需配置16*PSC[7:0]。TX引脚脉冲宽度可以为低功耗波特率的3倍。接收解码器工作模式与正常IrDA模式相同。

27.3.11. 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。

图片中的文字内容如下：

半双工模式下仅用单线通信，TX和RX引脚在内部连接到一起，RX引脚不再使用，TX引脚应配置为开漏模式，通信冲突由软件处理。

27.3.12. 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 27-15. ISO7816-3 数据帧格式

字节模式（T = 0）

相较于正常操作模式下的时序，从发送移位寄存器到TX引脚的传递时间延迟了半个波特率时钟，并且TC标志的置位将根据USART_GP寄存器的GUAT[7:0]设置延迟某一特定时间。在智能卡模式下，在最后一帧数据的停止位之后，内部保护时间计数器将开始计数，GUAT[7:0]的值配置为ISO7816-3协议的CGT减12。在保护时间寄存器向上计数这段时间TC将被强制拉低，当计数达到设定值时，TC被置位。

在USART发送期间，如果检测到有奇偶校验错误，TX引脚在停止位最后一个位时间内被拉低，智能卡发送一个NACK信号。根据协议，USART会自动重发SCRTNUM次。在重发数据帧前面会插入2位的帧间隔。最后一次重发字节后，TC会立即被置位。如果在最大重发次数后仍然收到NACK信号，USART将会停止发送，帧错误标志被置位。USART不会将NACK信号作为起始位。

在USART接收期间，如果在当前数据帧检测到校验错误，TX引脚在停止位的最后一个位时间内会被拉低。智能卡会接收到NACK信号。然后在智能卡端会产生一个帧错误。如果接收到的字节是错误的，RBNE中断和接收DMA请求都不会被激活。根据协议，智能卡将重新发送数据。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。

27.3.13. 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。

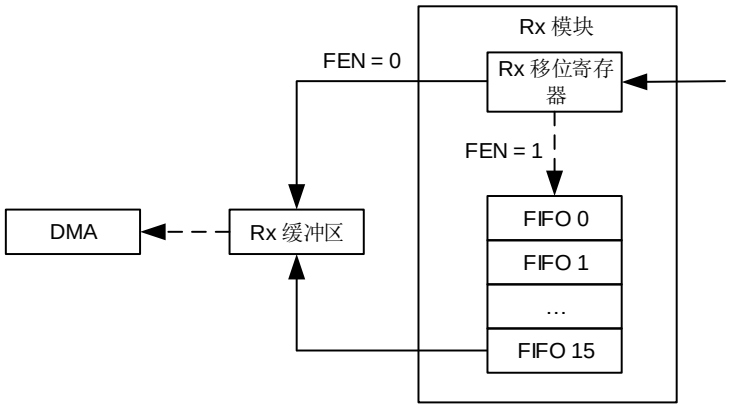
为了检测空闲状态，必须置位USART_CTL1寄存器的RTEN位和USART_CTL0寄存器的RTIE位。USART_RT寄存器必须被设置成与2个字节超时所对应的值。在最后一个停止位被接收后，当接收线在这期间是空闲的，将产生一个中断，通知软件当前块接收已经完成。

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ADDR0或ADDR1激活地址匹配中断（AMIE0 = 1或AMIE1 = 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。

27.3.14. 接收 FIFO

通过将USART_FCS寄存器的FEN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17帧的数据。若接收FIFO满，RFF位将被置位，如果RFFIE被置位，将产生中断。若接收FIFO达到RFTCFG[2:0]位域配置的阈值，RFT位将被置位，如果RFTIE位被置位，将产生中断。若接收FIFO非空且RFNEIE被置位，将产生中断。

图 27-16. USART 接收 FIFO 结构

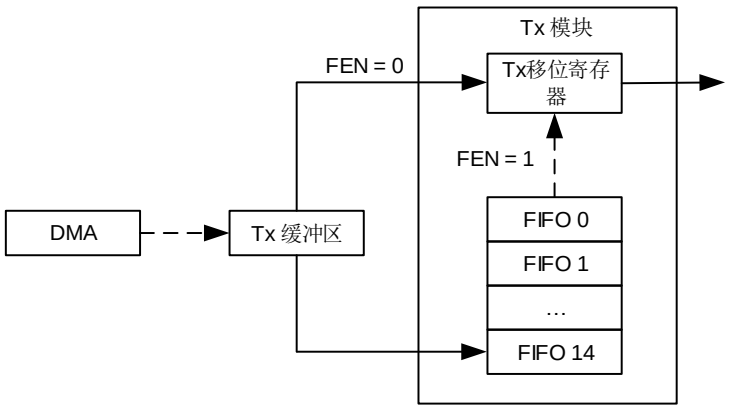


如果软件在响应RBNE中断时读数据接收缓冲区，在响应开始时，RBNEIE位应清0。当所有接收的数据被读出后，RBNEIE位应置位。在读出接收的数据前，PERR，NERR，FERR，EBF都应被清0。

27.3.15. 发送 FIFO

通过将USART_FCS寄存器的FEN置位使能发送FIFO，发送FIFO和发送缓存区可储存多至16帧的数据。若发送FIFO空，TFE将被置位，如果TFEIE位被置位，将产生中断。若发送FIFO中可用空间达到TFTCFG[2:0]位域配置的阈值，TFTIF将被置位(或者TFT将被清除)，如果TFTIE被置位，将产生中断。若发送FIFO非满，TFNF位被置位，如果TFNFIE被置位，将产生中断。

图 27-17. USART 发送 FIFO 结构



27.3.16. 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CK_IRC8M或CK_LXTAL。

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。

注意：当使用地址匹配方法将MCU从深度睡眠模式唤醒时，需要置位AMEN0或AMEN1。

27.3.17. USART 中断

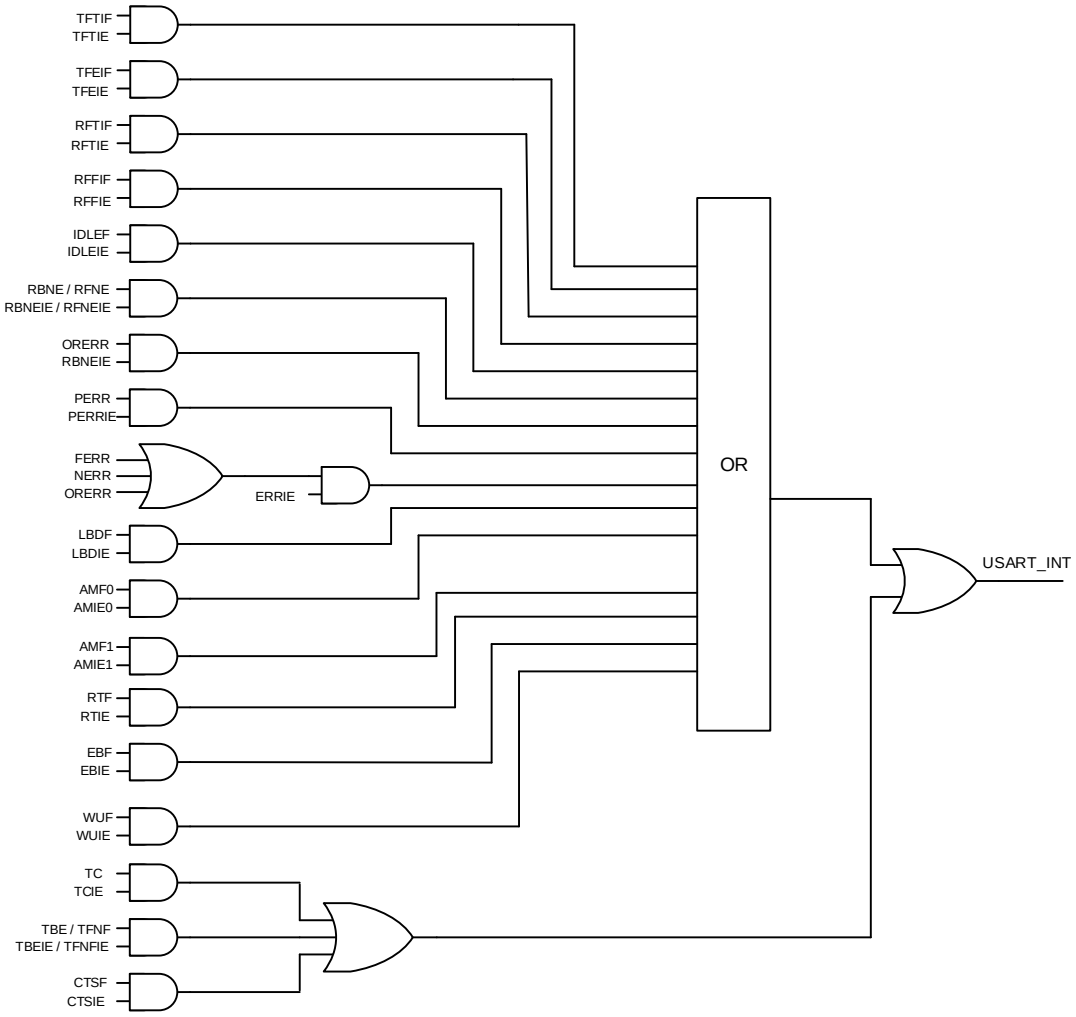
USART 中断事件和标志如*表 27-3. USART 中断请求*所示：

表 27-3. USART 中断请求

中断事件	事件标志	使能控制位
发送数据寄存器空或发送FIFO非满	TBE / TFNF	TBEIE / TFNFIE
CTS标志	CTSF	CTSIE
发送结束	TC	TCIE
接收到的数据可以读取或接收FIFO非空	RBNE / RFNE	RBNEIE / RFNEIE
检测到过载错误	ORERR	
接收FIFO满	RFFIF	RFFIE
接收FIFO达到阈值	RFTIF	RFTIE
发送FIFO空	TFEIF	TFEIE
发送FIFO中可用空间达到阈值	TFTIF	TFTIE
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或ORERR或FERR	ERRIE
字符0匹配	AMF0	AMIE0
字符1匹配	AMF1	AMIE1
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 27-18. USART 中断映射框图



27.4. USART 寄存器

USART0 基地址：0x4001 3800

USART1 基地址：0x4000 4400

USART2 基地址：0x4000 4800

UART3 基地址：0x4000 4C00

UART4 基地址：0x4000 5000

27.4.1. 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
AMIE1	保留		WL1	EBIE	RTIE	DEA[4:0]					DED[4:0]				
rw			rw	rw	rw			rw					rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVSMOD	AMIE0	MEN	WL0	WM	PCEN	PM	PERRIE	TBEIE / TFNFIE	TCIE	RBNEIE / RFNEIE	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	AMIE1	ADDR1中字符匹配中断使能 0：ADDR1中字符匹配中断禁用 1：ADDR1中字符匹配中断使能
30:29	保留	必须保持复位值。
28	WL1	字长1 WL1与WL0位决定字长。 00：8数据位 01：9数据位 10：7数据位 11：10数据位 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 在UART3 / UART4中，该位保留。
26	RTIE	接收超时中断使能 0：中断禁止 1：中断使能 在UART3 / UART4中，该位保留。
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	AMIE0	ADDR0中字符匹配中断使能 0：ADDR0中字符匹配中断禁用 1：ADDR0中字符匹配中断使能
13	MEN	静默模式使能 0：静默模式禁用 1：静默模式被使能
12	WL0	字长0 WL1与WL0位决定字长。 0：8数据位 1：9数据位 当USART被使能（UEN = 1）时，该位域不能被改写。
11	WM	从静默模式唤醒方法 0：空闲线 1：地址标记 当USART被使能（UEN = 1）时，该位域不能被改写。
10	PCEN	校验控制使能

		0：校验控制禁用 1：校验控制被使能 当USART被使能（UEN = 1）时，该位域不能被改写。
9	PM	校验模式 0：偶校验 1：奇校验 当USART被使能（UEN = 1）时，该位域不能被改写。
8	PERRIE	校验错误中断使能 0：校验错误中断禁用 1：当USART_STAT寄存器的PERR位置位时，将触发中断。
7	TBEIE	当FIFO禁用： 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。
	TFNFIE	当FIFO使能： 发送FIFO非满中断使能 0：中断禁止 1：当USART_STAT寄存器的TFNF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	当FIFO禁用： 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。
	RFNEIE	当FIFO使能： 接收FIFO非空中断使能和过载错误中断使能 0：接收FIFO非空中断和过载错误中断禁用 1：当USART_STAT寄存器的ORERR或RFNE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 CK_IRC8M 或 CK_LXTAL。 在UART3 / UART4中，该位保留。
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

27.4.2. 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
ADDR0[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	ADDM0	保留			AMEN0
rw	rw	rw		rw	rw	rw	rw		rw	rw	rw				rw

位/位域	名称	描述
31:24	ADDR0[7:0]	USART的节点地址0 这些位给出USART的节点地址0。 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匹配的检测。接收到的最高位为1的数据帧将和这些位进行比较。当ADDM0位被清零时，仅仅ADDR0[3:0]被用来比较。 在正常的接收期间，这些位也用来进行字符检测。所有接收到的字符（8位）与ADDR0[7:0]的值进行比较，如果匹配，AMF0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

		在UART3 / UART4中，该位保留。
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。 在UART3 / UART4中，该位保留。
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。 **注意：**0.5停止位和1.5停止位不适用于UART3 / UART4。
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。 在UART3 / UART4中，该位保留。
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。 在UART3 / UART4中，该位保留。
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。 在UART3 / UART4中，该位保留。
4	ADDM0	地址0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（ADDR0[5:0]，ADDR0[6:0]和ADDR0[7:0]）执行。 USART被使能（UEN = 1）时，该位域不能被改写。
3:1	保留	必须保持复位值。
0	AMEN0	地址0匹配使能 0：地址0匹配模式禁用 1：地址0匹配模式使能

27.4.3. 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
ADDR1[7:0]								ADDM1	WUIE	WUM[1:0]		SCRTNUM[2:0]			AMEN1
rw								rw	rw	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
31:24	ADDR1[7:0]	USART的节点地址1 这些位给出USART的节点地址1。 在多处理器通信并且静默模式或者深度睡眠模式期间，这些位用来唤醒进行地址标记的检测。接收到的最高位为1的数据帧将和这些位进行比较。当ADDM1位被清零时，仅仅ADDR1[3:0]被用来比较。 在正常的接收期间，这些位也用来进行字符检测。所有接收到的字符（8位）与ADDR1[7:0]的值进行比较，如果匹配，AMF1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	ADDM1	地址1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（ADDR1[5:0]，ADDR1[6:0]和ADDR1[7:0]）执行。 USART被使能（UEN = 1）时，该位域不能被改写。
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 在UART3 / UART4，该位保留。
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。 在UART3 / UART4，该位保留。
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。 在UART3 / UART4中，该位保留。
16	AMEN1	地址1匹配使能 0：地址1匹配模式禁用 1：地址1匹配模式使能
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（DENR = 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。 在UART3 / UART4中，该位保留。
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。 在UART3 / UART4中，该位保留。
3	HDEN	半双工使能 0：禁用半双工模式 1：开启半双工模式

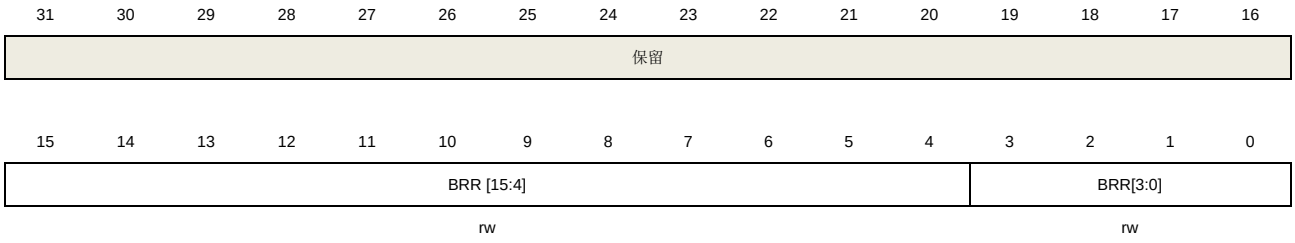
		USART被使能（UEN = 1）时，该位域不能被改写。
2	IRLP	IrDA低功耗模式 0：正常模式 1：低功耗模式 USART被使能（UEN = 1）时，该位域不能被改写。
1	IREN	IrDA模式使能 0：IrDA禁用 1：IrDA被使能 USART被使能（UEN = 1）时，该位域不能被改写。 在UART3 / UART4中，该位保留。
0	ERRIE	多级缓存通信模式的错误中断使能 0：禁用错误中断 1：在多级缓存通信时，当USART_STAT寄存器的FERR位，ORERR位或NERR位被置位时，会产生中断。

27.4.4. USART 波特率寄存器（USART_BAUD）

地址偏移：0x0C
复位值：0x0000 0000

当USART（UEN = 1）被使能时，该寄存器不能被改写。

该寄存器只能按字（32位）访问。



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。

27.4.5. 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

USART被使能（UEN = 1）时，该寄存器不能被改写。

在UART3 / UART4中，该寄存器保留。

该寄存器只能按字（32位）访问。

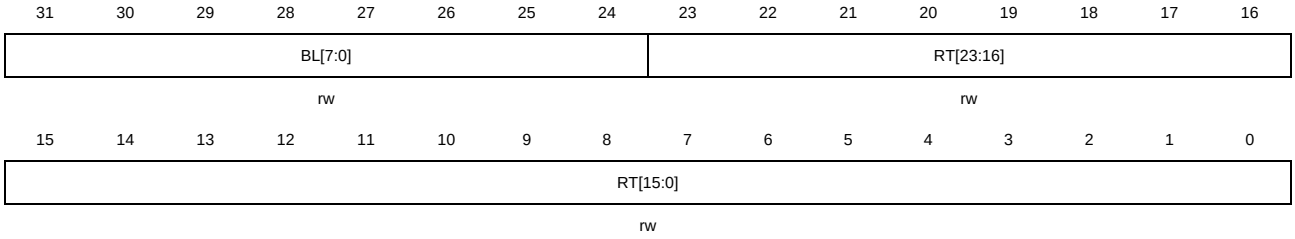
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	GUAT[7:0]	在智能卡模式下的保护时间值 USART被使能（UEN = 1）时，该位域不能被改写。
7:0	PSC[7:0]	预分频器值 在红外低功耗模式下，对系统时钟进行分频已获得低功耗模式下的频率。寄存器的值是分频系数 00000000：保留 – 不设置这个值 00000001：1分频 00000010：2分频 ... 在IrDA正常模式下的分频值 00000001：仅能设为这个值 在智能卡模式下，对系统时钟进行分频的值存于PSC[4:0]位域中。PSC[7:5]位保持为复位值。分频系数是寄存器中值的两倍。 00000：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。

27.4.6. USART 接收超时寄存器（USART_RT）

地址偏移：0x14
复位值：0x0000 0000

在UART3 / UART4中，该寄存器保留。

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:24	BL[7:0]	块长度 这些位给出了智能卡T=1的接收时块的长度。它的值等于信息字节的长度+结束部分的长度（1-LEC/2-CRC）–1。 这个值可以在块接收开始时设置（用于需要从块的序言提取块的长度的情形），这个只在每一个接收时钟周期只能设置一次。在智能卡模式下，当TBE=0时，块的长度计数器被清0。 在其他模式下，当REN = 0（禁用接收器）并且/或者当EBC位被写1时块的长度计数器被清0。
23:0	RT[23:0]	接收器超时门限 该位域指定接收超时值，单位是波特时钟的时长 标准模式下，如果在最后一个字节接收后，在RT规定的时长内，没有检测到新的起始位，RTF标志被置位。 在智能卡模式，这个值被用来实现CWT和BWT。在这种情况下，超时检测是从最后一个接收字节的起始位开始。 这些位可以在工作时改写。假如一个新数据到来的时间比RT规定的晚，RTF标志会被置位。对于每个接收字符，这个值只能改写一次。 **注意**：当16倍过采样时，RT[23:0]不能被配置为0xFFFFFF。

27.4.7. 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SBF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。

27.4.8. 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	AMF0	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		AMF1	EBF	RTF	CTS	CTSF	LBDF	TBE TFNF	TC	RBNE RFNE	IDLEF	ORERR	NERR	FERR	PERR
		r	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 在UART3/UART4中，该位保留。
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。 在UART3 / UART4中，该位保留。
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	AMF0	ADDR0中字符匹配标志 0：ADDR0中字符和接收到的字符不匹配 1：ADDR0中字符和接收到的字符匹配，如果USART_CTL0寄存器的AMIE0 = 1，将引发一个中断。 当接收到ADDR0[7:0]中定义的字符时，硬件置位。 通过向USART_INTC寄存器的AMC0写1清0。
16	BSY	忙标志

		0：USART处于空闲 1：USART正在接收
15:14	保留	必须保持复位值。
13	AMF1	ADDR1中字符匹配标志 0：ADDR1中字符和接收到的字符不匹配 1：ADDR1中字符和接收到的字符匹配，如果USART_CTL0寄存器的AMIE1 = 1，将引发一个中断。 当接收到ADDR1[7:0]中定义的字符时，硬件置位。 通过向USART_INTC寄存器的AMC1写1清0。
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。 在UART3 / UART4中，该位保留。
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。 在UART3 / UART4中，该位保留。
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。 在UART3 / UART4中，该位保留。
7	TBE	当FIFO模式禁用： 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。
	TFNF	当FIFO模式使能： 传输FIFO非满 0：传输FIFO满 1：传输FIFO非满。如果USART_CTL0寄存器的TFNFIE位置位，将会有中断产生。 当发送FIFO非满时，由硬件置位。当FIFO满时，由硬件置1。 **注意：**在TXCMD置位期间，TFNF保持复位直到发送FIFO空。
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或USART_FCS寄存器中TFE位被置位，该位由硬件置位。 通过向USART_INTC寄存器的TCC位写1清0。 **注意：**当TEN清零时，TC位被立即置位，传输结束。
5	RBNE	当FIFO模式禁用： 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。
	RFNE	当FIFO模式使能： 接收FIFO非空 0：接收FIFO为空 1：接收FIFO非空。当寄存器USART_CTL0的RFNEIE位被置位，将会有中断产生。 当接收FIFO非空时，由硬件置位。当接收FIFO为空时，由硬件清零。该位也可以通过RXFCMD置位来清零。
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或RFNEIE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或RFF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。 注意：当该位与RBNE位或RFNE位同时置位时，将不会产生中断。当FIFO使能时，噪声错误与USART_RDATA中的数据有关。
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。 注意：当FIFO使能时，噪声错误与USART_RDATA中的数据有关。
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。 注意：当FIFO使能时，噪声错误与USART_RDATA中的数据有关。

27.4.9. 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		AMC0	AMC1
											w			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

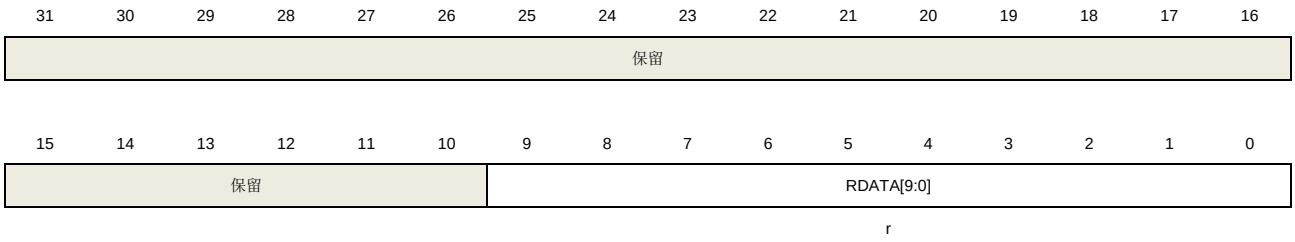
位/位域	名称	描述
31:21	保留	必须保持复位值。
20	WUC	从深度睡眠模式唤醒标志的清除 向该位写1清除USART_STAT寄存器的WUF位。 在UART3 / UART4中，该位保留。
19:18	保留	必须保持复位值。
17	AMC0	ADDR0中字符匹配标志清除 向该位写1清除USART_STAT寄存器的AMF0位。
16	AMC1	ADDR1中字符匹配标志清除 向该位写1清除USART_STAT寄存器的AMF1位。
15:13	保留	必须保持复位值。
12	EBC	块结束标志清除 向该位写1清除USART_STAT寄存器的EBF位。 在UART3 / UART4中，该位保留。
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。 在UART3 / UART4中，该位保留。
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。

		在UART3 / UART4中，该位保留。
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。

27.4.10. 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。



位/位域	名称	描述
31:10	保留	必须保持复位值。
9:0	RDATA[9:0]	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6位、7位、8位或9位，取决于数据的长度）是奇偶校验位。

27.4.11. USART 数据发送寄存器（USART_TDATA）

地址偏移：0x28
复位值：未定义

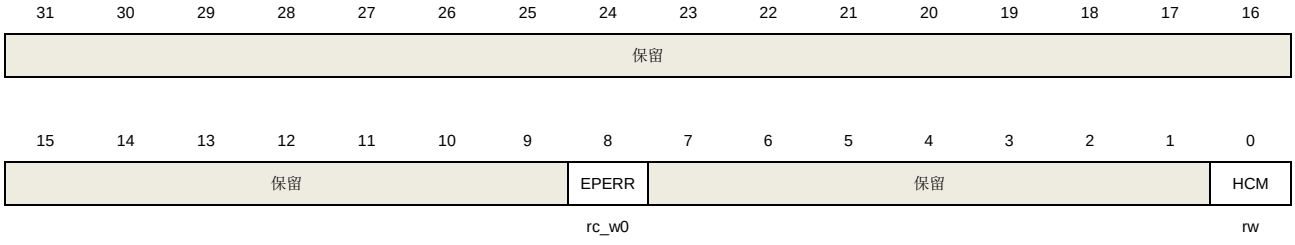
该寄存器只能按字（32 位）访问。

位/位域	名称	描述
31:10	保留	必须保持复位值。
9:0	TDATA[9:0]	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6位、7位、8位或9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。

27.4.12. 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。



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

27.4.13. USART FIFO 控制和状态寄存器（USART_FCS）

地址偏移：0xD0

复位值：0x0300 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
TFEIE	保留	TFTIE	保留	RFTIE	TFEC	TFTIF	TFEIF	保留	RFTIF	TFTCFG[2:0]			RFTCFG[2:0]		
rw		rw		rw	rw	r	r		rc_w0	rw			rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RFFIF	RFCNT[2:0]			RFF	RFE	RFFIE	FEN	TFF	TFE	TFT	RFT	保留	RFCNT[4:3]		ELNACK
rc_w0	r			r	r	rw	rw	r	r	r	r		r		rw

位/位域	名称	描述
31	TFEIE	发送FIFO空中断使能 如果该位置位，当TFE位置位时，中断发生。 0：禁止发送FIFO空中断 1：使能发送FIFO空中断
30	保留	必须保持复位值。
29	TFTIE	发送FIFO到达阈值中断使能 如果该位置位，当发送FIFO中的可用空间到达TFTCFG[2:0]配置的阈值时，中断发生。 0：禁止发送FIFO到达阈值中断 1：使能发送FIFO到达阈值中断
28	保留	必须保持复位值。
27	RFTIE	接收FIFO到达阈值中断使能 如果该位置位，当发送FIFO到达RFTCFG[2:0]配置的阈值时，中断发生。 0：禁止接收FIFO到达阈值中断

		1：使能接收FIFO到达阈值中断
26	TFEC	发送FIFO空标志清除 写1清除TFE标志
25	TFTIF	发送FIFO到达阈值中断标志 当TFTIE位置位时，该位有效。 0：发送FIFO中的可用空间未到达可编程阈值 1：发送FIFO中的可用空间到达可编程阈值中断标志。当TFTIE置位时，中断发生。 当发送FIFO中的可用空间TFTCFG[2:0]配置的阈值时，由硬件置位。
24	TFEIF	发送FIFO空中断标志 当TFEIE置位时，该位有效。 0：发送FIF非空 1：发送FIFO为空中断标志
23	保留	必须保持复位值。
22	RFTIF	接收FIFO到达阈值中断标志 当RFTIE位置位时，该位有效。 0：接收FIFO到达可编程阈值 1：接收FIFO到达可编程阈值中断标志
21:19	TFTCFG[2:0]	发送FIFO阈值配置 000：发送FIFO到达FIFO深度的1/8 001：发送FIFO到达FIFO深度的1/4 010：发送FIFO到达FIFO深度的1/2 011：发送FIFO到达FIFO深度的3/4 100：发送FIFO到达FIFO深度的7/8 101：发送FIFO为空 11x：保留
18:16	RFTCFG[2:0]	接收FIFO阈值配置 000：接收FIFO到达FIFO深度的1/8 001：接收FIFO到达FIFO深度的1/4 010：接收FIFO到达FIFO深度的1/2 011：接收FIFO到达FIFO深度的3/4 100：接收FIFO到达FIFO深度的7/8 101：接收FIFO为满 11x：保留
15	RFFIF	接收FIFO满中断标志 当RFFIE置位时，该位有效。 0：接收FIFO非满

		1：接收FIFO满中断标志
14:12	RFCNT[2:0]	接收FIFO计数值 该位域与RFCNT[4:3]位域决定接收FIFO计数值。
11	RFF	接收FIFO满标志 0：接收FIFO不为满 1：接收FIFO满。当RFFIE置位时，中断发生。 当接收数据个数为RXFIFO 大小加1时，由硬件置1。
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 如果该位置位，当RFF位置位时，中断发生。 0：禁止接收FIFO满中断 1：使能接收FIFO满中断
8	FEN	FIFO使能 0：禁止使用FIFO 1：使能FIFO 当USART被使能（UEN = 1）时，该位域不能被改写。 **注意：**当接收或发送数据未完成时，不要改变该位。当UEN位清零且不改变该位，在重配UEN位时，如果之前FIFO的值不在需要，需要先刷新FIFO。
7	TFF	发送FIFO满标志 0：发送FIFO不为满 1：发送FIFO满。
6	TFE	发送FIFO空标志 0：发送FIFO不为空 1：发送FIFO空。当TFEIE置位时，中断发生。 当发送FIFO为空时，由硬件置位。当发送FIFO中只少有一个数据时，由硬件清0。向USART_CMD寄存器的TXFCMD位写1时，由硬件置位
5	TFT	发送FIFO阈值标志 0：发送FIFO中的可用空间到达可编程阈值 1：发送FIFO中的可用空间未到达可编程阈值。 当发送FIFO中的可用空间到达TFTCFG[2:0]配置的阈值时，由硬件复位。
4	RFT	接收FIFO阈值标志 0：接收FIFO未到达可编程阈值 1：接收FIFO到达可编程阈值。当RFTIE置位时，中断发生。 当接收FIFO到达RFTCFG[2:0]配置的阈值时，由硬件置1。这意味着接收FIFO中有

		RFTCFG[2:0] -1个数据，USART_RDATA寄存器中有一个数据。 **注意：**当RTFCFG[2:0]=0b101且接收到16个数据时，RFT被置位。
3	保留	必须保持复位值。
2:1	RFCNT[4:3]	接收FIFO计数值 该位域与RFCNT[2:0]位域决定接收FIFO计数值。
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 在UART3 / UART4中，该位保留。

28. 内部集成电路总线接口（I2C）

28.1. 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 的负载。

28.2. 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，该模式必须在 SYSCFG_CFG1 中使能）；
- 从机模式下可配置的 SCL 主动拉低；
- 支持 DMA 模式；
- 兼容 SMBus 3.0 和 PMBus 1.3；
- 可选择的 PEC（报文错误校验）生成和校验；
- 可编程模拟过滤器和数字过滤器；
- I2C 地址匹配时，由睡眠模式、深度睡眠模式唤醒；
- 独立于 PCLK 的时钟。

28.3. 功能说明

I2C 接口的内部结构如*图 28-1. I2C 模块框图*所示。

图 28-1. I2C 模块框图

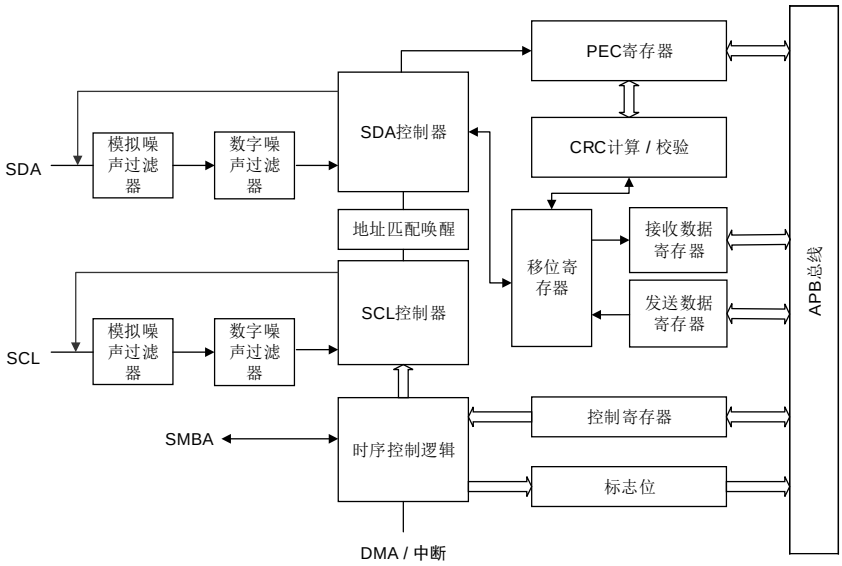


表 28-1. I2C 总线术语说明（参考飞利浦 I2C 规范）

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

28.3.1. 时钟要求

I2C 时钟独立于 PCLK 时钟，因此可以独立操作 I2C。

I2C 时钟（I2CCLK）可以从以下三个时钟源中选择：

- CK_APB1：AP1 时钟（默认值）
- CK_SYS：系统时钟
- CK_IRC8M：内部高速 8M RC 振荡器

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 130ns，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。

28.3.2. I2C 通讯流程

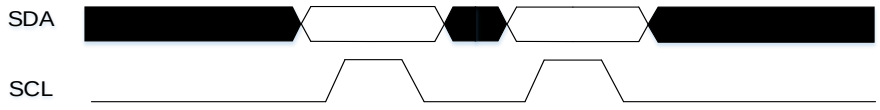
主机和从机都能实现数据收发，因此，I2C 可以实现四种工作模式：

- 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 28-2. 数据有效性*）。每个数据比特传输需要一个时钟脉冲。

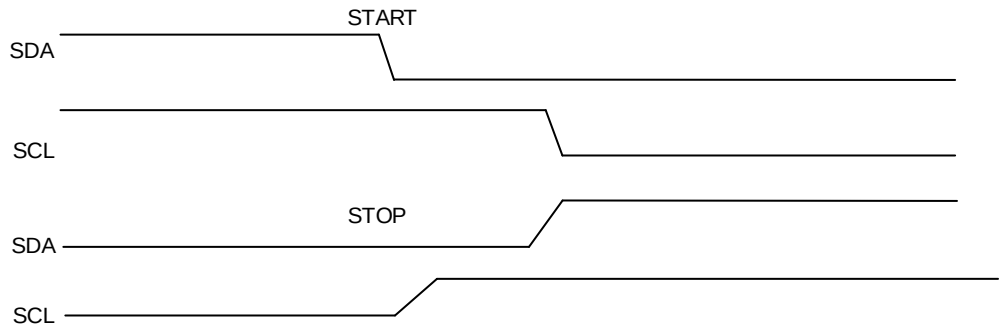
图 28-2. 数据有效性



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 28-3. 开始和停止*）。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 28-3. 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 28-4. 10 位地址的 I2C 通讯流程（主机发送）

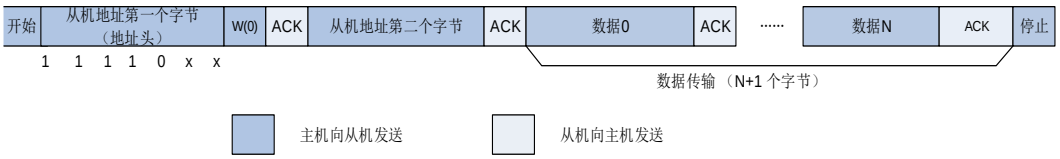


图 28-5. 7 位地址的 I2C 通讯流程（主机发送）

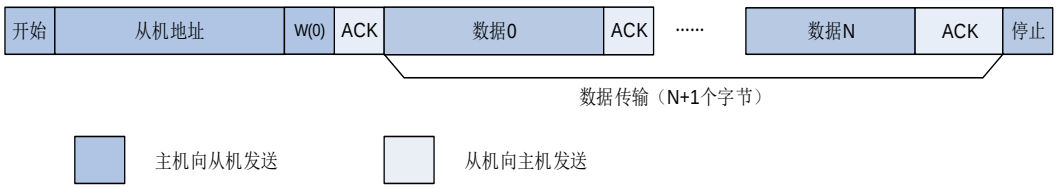
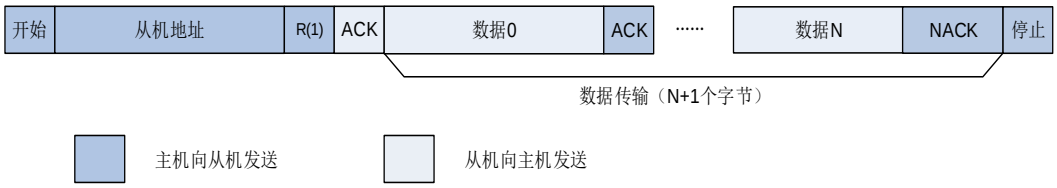


图 28-6. 7 位地址的 I2C 通讯流程（主机接收）



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 28-7. 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 28-8. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R = 1）*所示。

图 28-7. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R = 0）

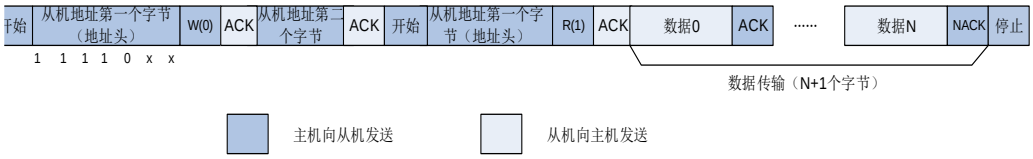
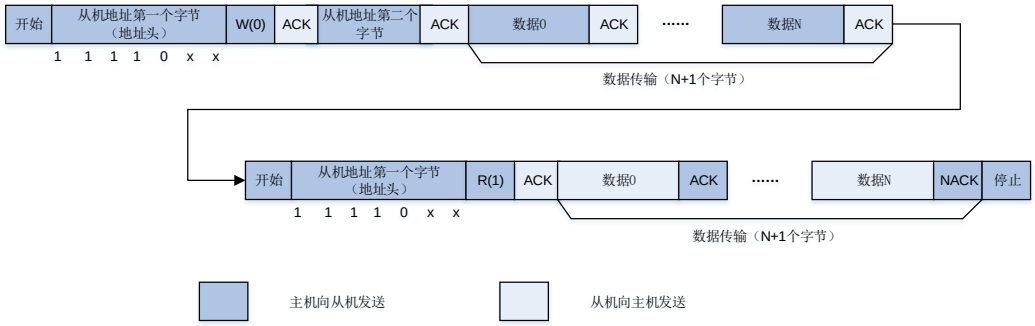


图 28-8. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R = 1）



28.3.3. 噪声滤波器

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。

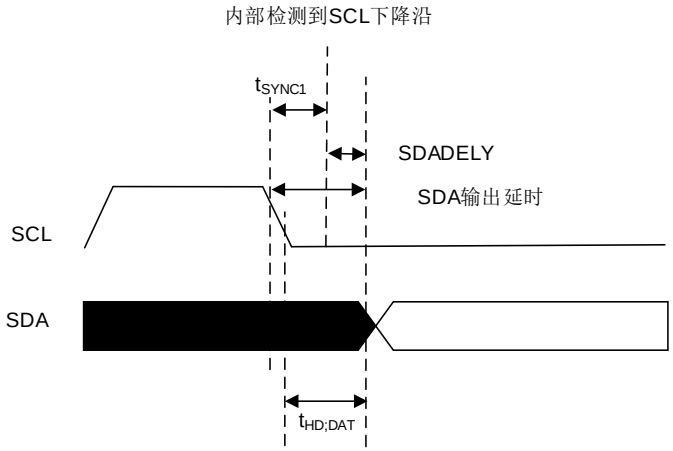
数字滤波器由 I2C_CTL0 寄存器中 DNF[3:0]位来配置。当数字滤波器使能时，SCL 和 SDA 电平保持稳定的时间大于DNF[3:0]×t_{I2CCLK}才会发生内部变化。抑制峰值宽度可由 DNF[3:0]配置。

28.3.4. 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。

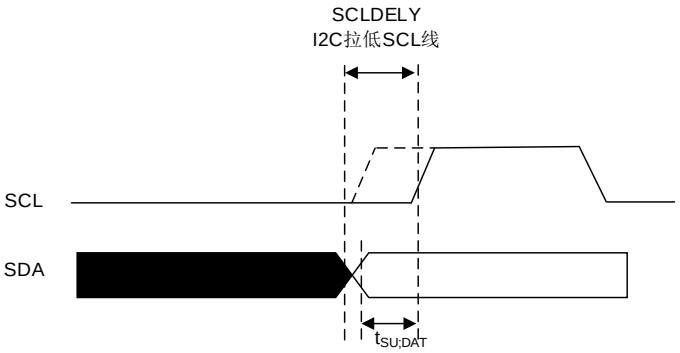
如果数据已经在 I2C_TDATA 寄存器中，在经历 SDADELY 延时后，数据由 SDA 发送，如*图 28-9. 数据保持时间*所示。

图 28-9. 数据保持时间



当数据经过 SDA 发送时，SCLDELY 计数器开启。如图*图 28-10. 数据建立时间*所示。

图 28-10. 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}。

注意：当 SDADELY = 0 时，$t_{SDADELY}$ = t_{PSC}。

SDADELY 必须符合以下条件：

- SDADELY≥{t_f(max)+$t_{HD;DAT}$(min)-t_{AF}(min)-[(DNF+3)*t_{I2CCLK}]}/[(PSC+1)*t_{I2CCLK}]
- SDADELY≤{$t_{HD;DAT}$(max)-t_{AF}(max)-[(DNF+4)*t_{I2CCLK}]}/[(PSC+1)*t_{I2CCLK}]

注意：t_{AF}为模拟滤波器延时，$t_{HD;DAT}$必须小于$t_{VD;DAT}$的最大值。

当 SS=0 时，经过延时$t_{SDADELY}$，在数据写入 I2C_TDATA 寄存器之前，从机会拉低时钟线。在数据建立时间期间 SCL 保持低电平。数据建立时间$t_{SCLDELY}$=(SCLDELY+1)*t_{PSC}。$t_{SCLDELY}$影响$t_{SU;DAT}$。

SCLDELY 必须符合以下条件：

- SCLDELY≥{[t_r(max)+$t_{SU;DAT}$(min)]/[(PSC+1)*t_{I2CCLK}]}-1

在主机模式下，SCL 时钟高低电平由 I2C_TIMING 寄存器中 PSC[3:0]，SCLH[7:0]和 SCLL[7:0]控制。

当内部检测到 SCL 下降沿，在释放 SCL 输出之前会插入一个延时，该延时为t_{SCLL}=(SCLL+1)*t_{PSC}，其中t_{PSC}=(PSC+1)*t_{I2CCLK}。t_{SCLL}影响 SCL 低电平持续时间t_{LOW}。

当内部检测到 SCL 上升沿，在将 SCL 拉低之前会插入一个延时，该延时为t_{SCLH}=(SCLH+1)*t_{PSC}，其中t_{PSC}=(PSC+1)*t_{I2CCLK}。t_{SCLH}影响 SCL 高电平持续时间t_{HIGH}。

注意：时序配置和 SS 位在 I2C 外设使能时是不能改变的。

表 28-2. 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	

28.3.5. 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

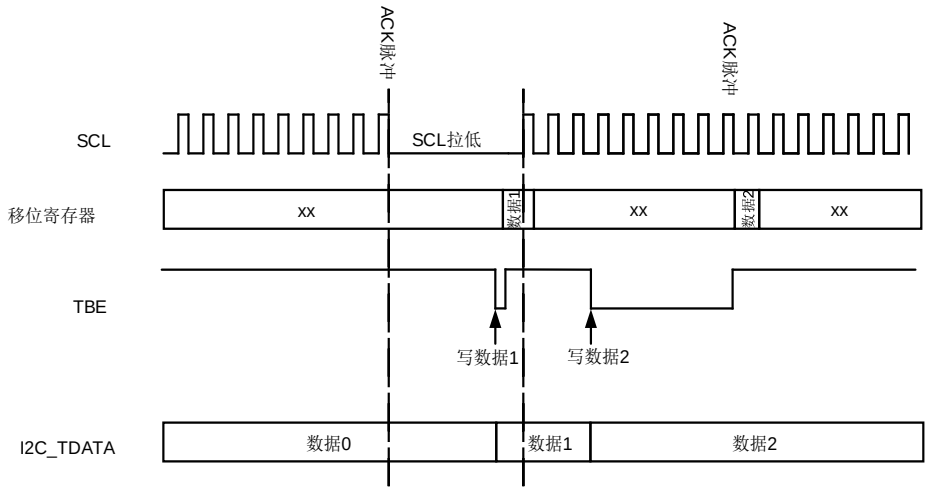
- I2CEN 写 1

28.3.6. 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。

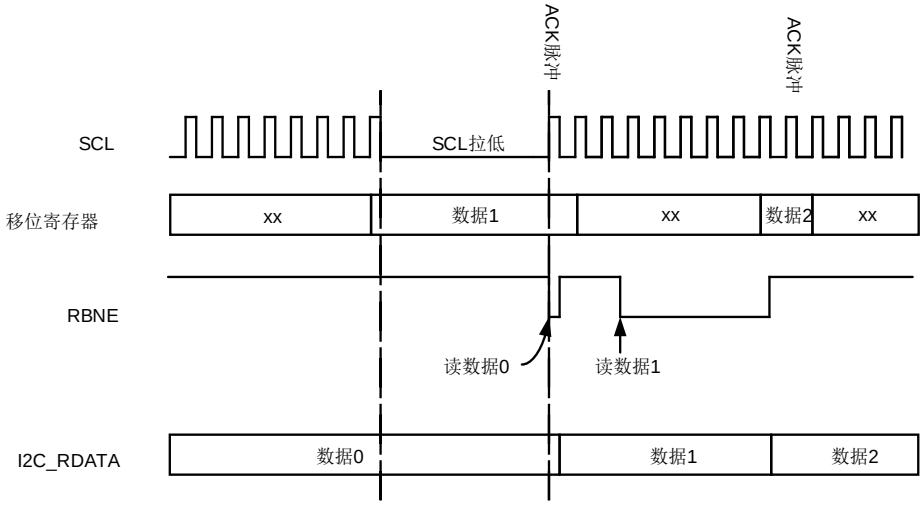
图 28-11. 数据发送



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 28-12. 数据接收



硬件传输管理重载和自动结束模式

为了管理字节传输和中断如*表 28-3. 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 28-3. 可关闭通信模式

工作模式	行为
主机模式	产生 NACK，STOP 和 RESTART
从机接收模式	ACK 控制
SMBus 模式	PEC 生成 / 校验

传输的字节数由 BYTENUM[7:0]在 I2C_CTL1 寄存器中配置。如果 BYTENUM 大于 255，或者处于从机字节控制模式，则必须通过将 I2C_CTL1 寄存器中 RELOAD 位置 1 来使能重载模式。在重载模式下，当 BYTENUM 计数到 0 时，TCR 位将置 1，如果 TCIE 位置 1 将产生中断。当 TCR 位置 1 时，SCL 将被拉低。在 BYTENUM 写一个非零值将清除 TCR 位。

注意：重载模式必须在 BYTENUM[7:0]最后一次重载后禁用。

当使能自动结束模式时，必须禁用重载模式。在自动结束模式下，当 BYTENUM[7:0]计数到 0 时，主机将自动发送一个 STOP 信号。

当重载模式和自动结束模式都被禁用时，I2C 通信进程需要由软件终止。如果 BYTENUM[7:0]中的字节数已经传输完成，软件应将 STOP 位置 1 来产生一个 STOP 信号，然后清除 TC。

28.3.7. 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：

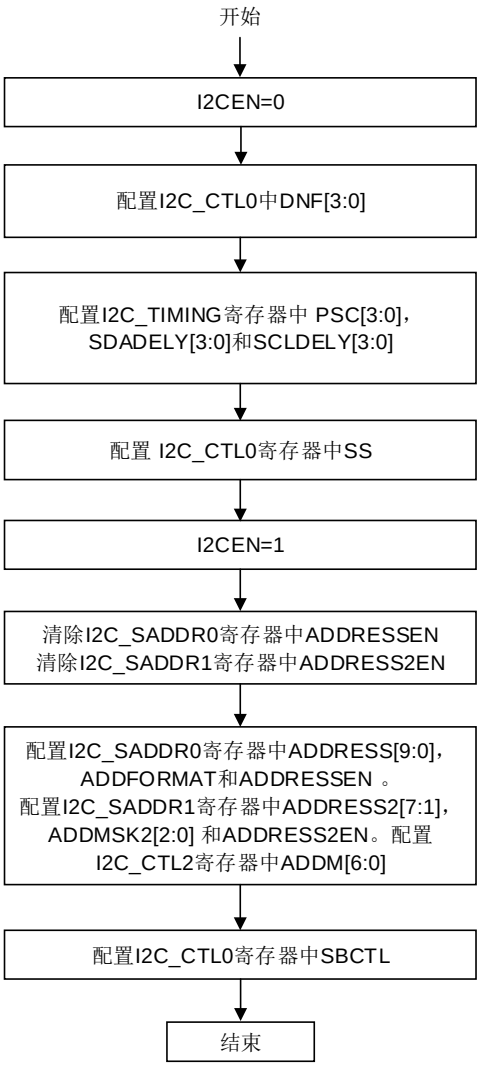
1、 I2CEN = 0

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 28-13. 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 28-14. I2C 从机发送编程模型（SS=0）

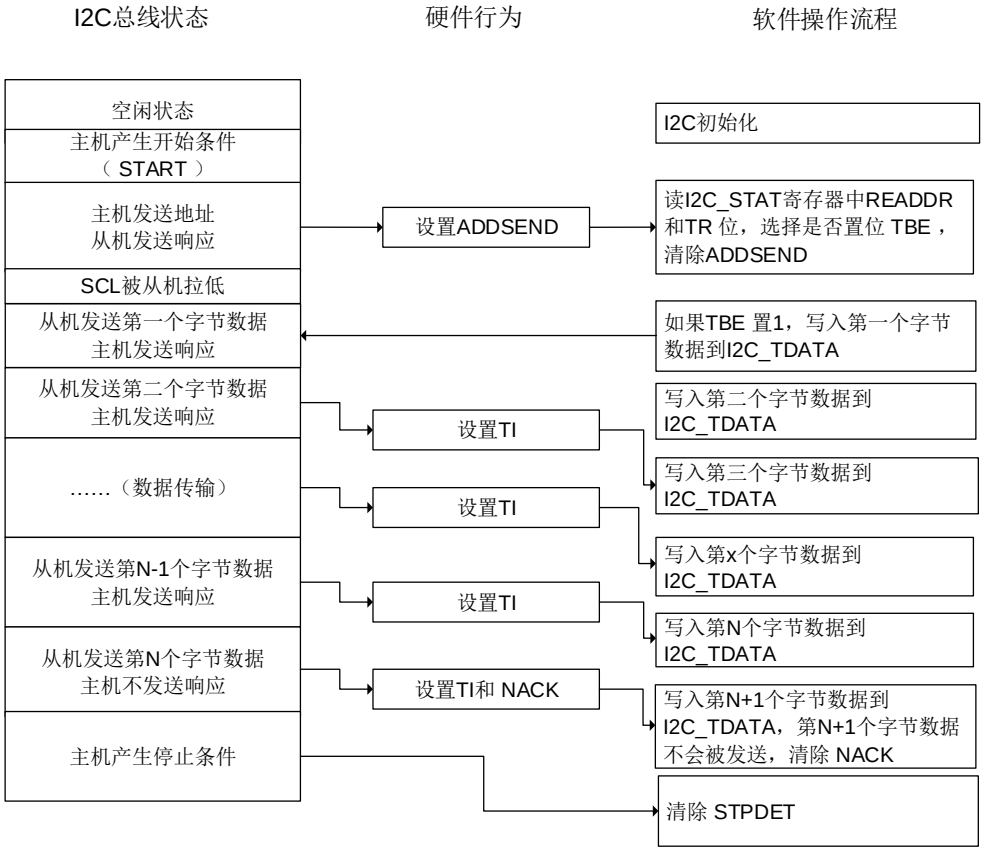
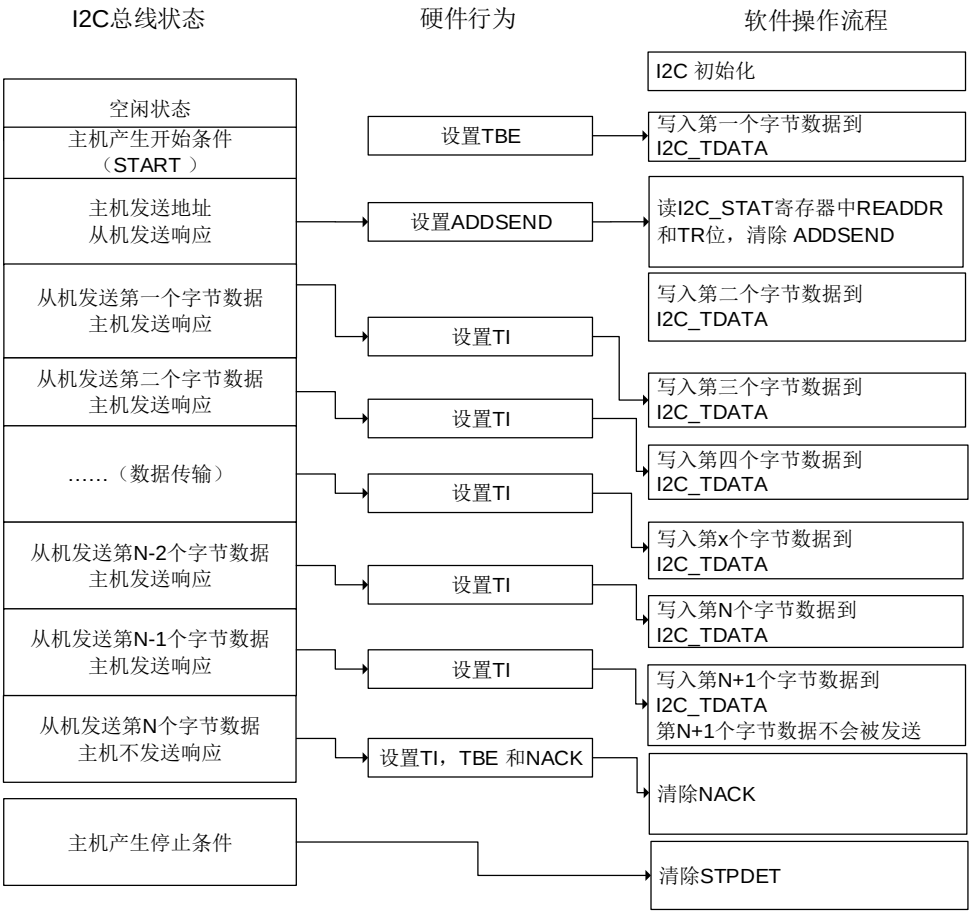


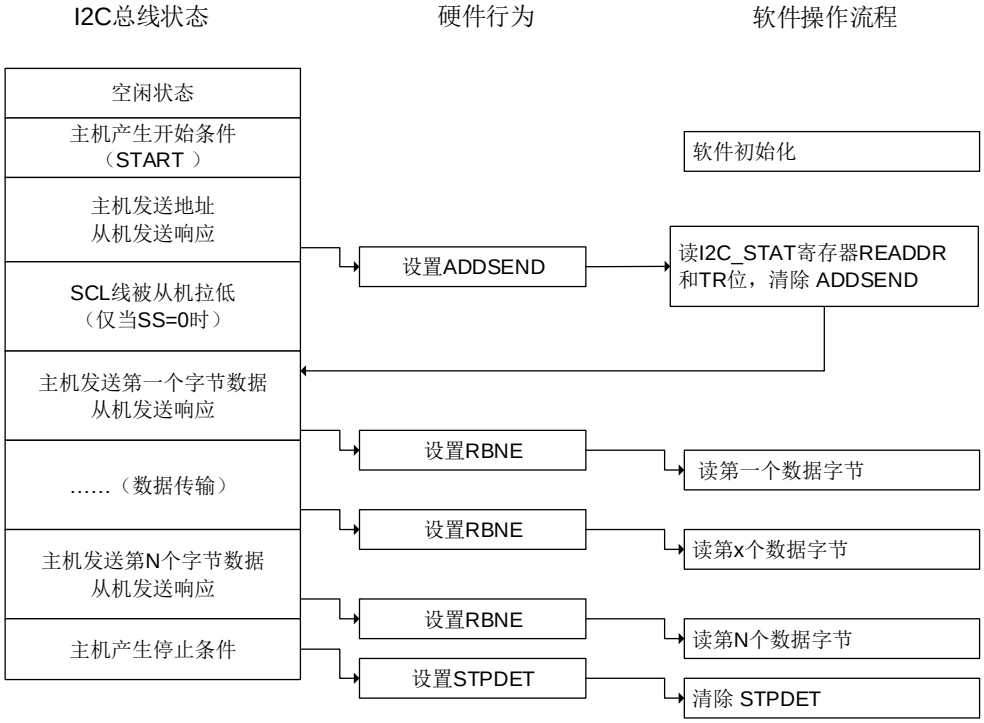
图 28-15. I2C 从机发送编程模型（SS=1）



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 28-16. I2C 从机接收编程模型



28.3.8. 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

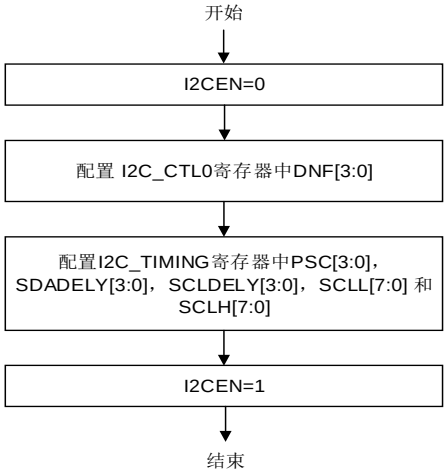
大于或者等于 255，必须将 BYTENUM[7:0]配置为 0xFF。然后主机发送 START 信号。以上提到的所有位必须在 START 位置 1 之前配置。START 信号发送完成之后，待 I2C_STAT 寄存器 I2CBSY 位为 0 时，发送从机地址。当仲裁丢失时，主机切换成从机模式，START 位由硬件清零。当从机地址发送完成时，START 位由硬件清零。

注意：BYTENUM[7:0]不能配置为 0。

在 10 位寻址模式下，在发送 10 位地址头之后，如果主机接收到 NACK，主机将重发 10 位地址头直到收到 ACK。将 ADDSENDC 置 1 可以停止重发从机地址。

如果 START 位置 1 时，I2C 作为从机被寻址成功，ADDSEND 置 1，主机将切换为从机模式。START 位将在 ADDSENDC 置 1 时清零。

图 28-17. 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_CTL1 寄存器 BYTENUM[7:0]。如果发送字节数大于 255，必须通过将 I2C_CTL1 寄存器 RELOAD 位置 1 来使能重载模式。在重载模式下，当 BYTENUM[7:0]个字节传输完成，I2C_STAT 寄存器 TCR 位将置 1，并且在 BYTENUM[7:0]更新一个非零值之前，SCL 被拉低。

如果接收到 NACK，TI 位将不会置 1。

- 如果 BYTENUM[7:0]个字节传输完成且 RELOAD=0，将 I2C_CTL1 寄存器中 AUTOEND 置 1 可以自动产生 STOP 信号。当 AUTOEND=0 时，I2C_STAT 寄存器 TC 位将置 1 且 SCL 被拉低。在这种情况下，主机可以通过将 I2C_CTL1 寄存器中 STOP 位置 1 来产生 STOP 信号。或者产生 RESTART 信号来开始一个新的数据传输过程。将 START / STOP 置 1 可以清除 TC 位。
- 如果接收到 NACK 信号，I2C 将自动产生 STOP 信号。I2C_CTL0 寄存器中 NACK 将置 1，如果 NACKIE 位置 1，将产生中断。

注意：当 RELOAD=1 时，AUTOEND 位无效。

图 28-18. I2C 主机发送编程模型（N<=255）

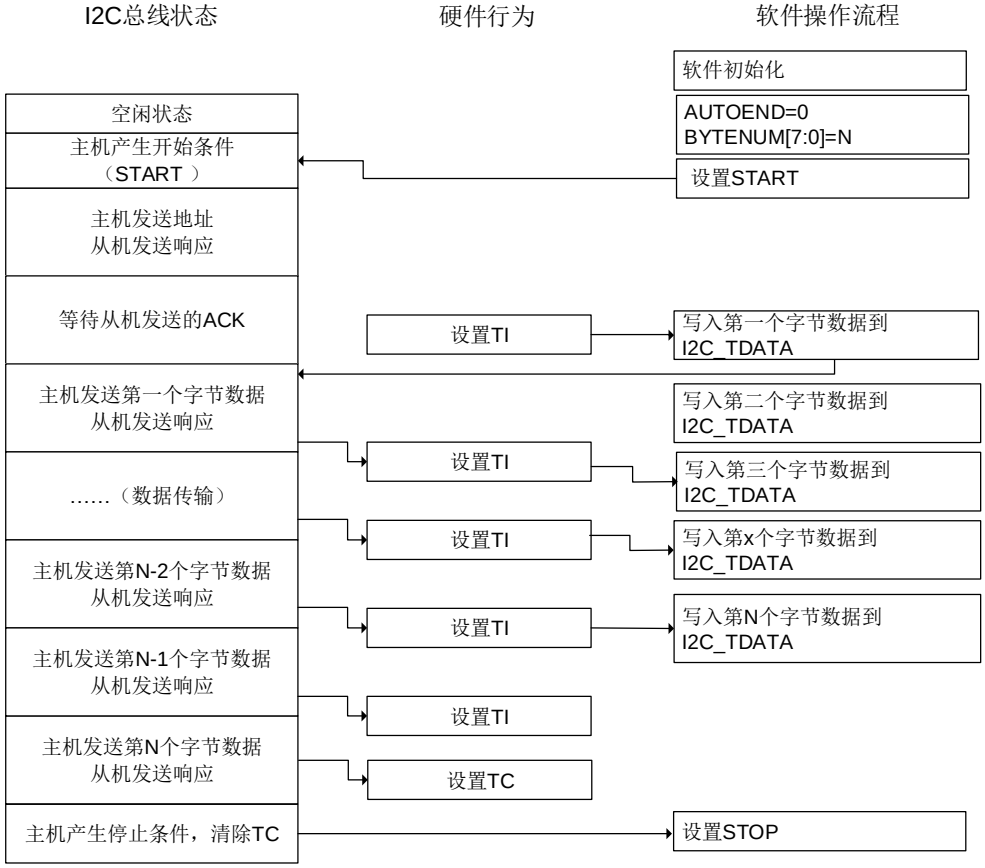


图 28-19. I2C 主机发送编程模型（N>255）

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_CTL1 寄存器中 RELOAD 位置 1 来使能重载模式。在重载模式下，当 BYTENUM[7:0]个字节传输完成，I2C_STAT 寄存器中 TCR 位将置 1，在 BYTENUM[7:0]中写入一个非零值之前，SCL 被拉低。

如果 BYTENUM[7:0]个字节传输完成且 RELOAD=0，将 I2C_CTL1 寄存器中 AUTOEND 置 1 可以自动产生 STOP 信号。当 AUTOEND=0 时，I2C_STAT 寄存器 TC 位将置 1 且 SCL 被拉低。在这种情况下，主机可以通过将 I2C_CTL1 寄存器中 STOP 位置 1 来产生 STOP 信号。或者产生 RESTART 信号来开始一个新的数据传输过程。将 START / STOP 置 1 可以清除 TC 位。

图 28-20. I2C 主机接收编程模型（N<=255）

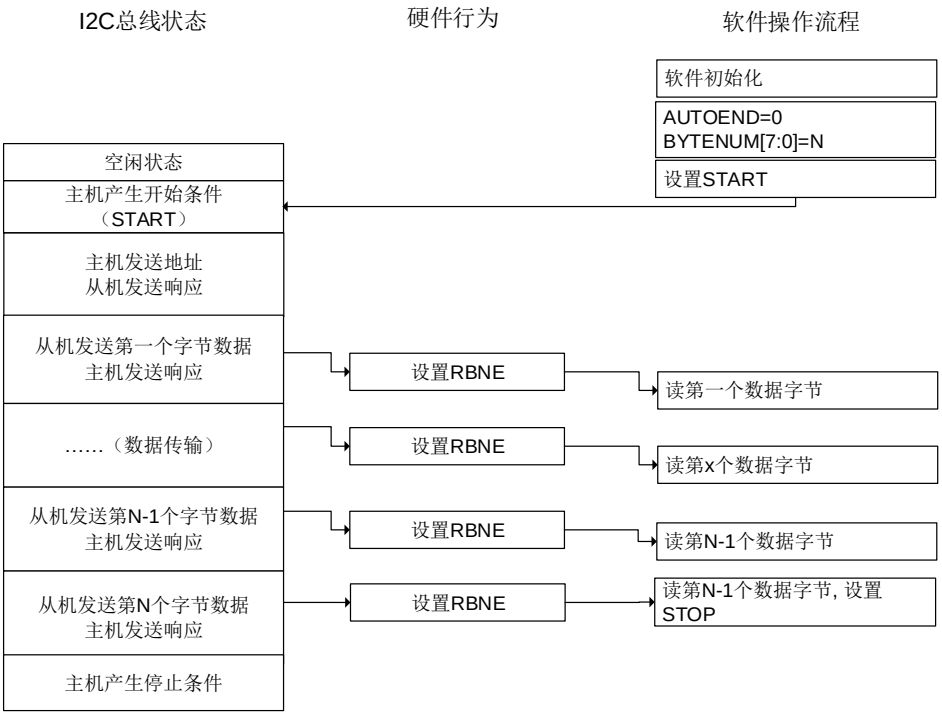


图 28-21. I2C 主机接收编程模型（N>255）

28.3.9. 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。

BUSTOB[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）。

28.3.10. 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 寄存器中的内容将在 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。

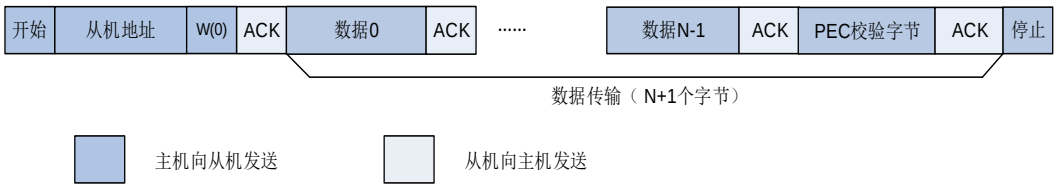
SMBus 作为从机接收器时，为了在数据发送完成时进行 PEC 校验，SBCTL 位必须置 1。要对每个字节进行 ACK 控制，必须通过将 RELOAD 位置 1 来使能 RELOAD 模式。如果要校验 PEC 字节，必须将 RELOAD 位清零同时将 PECTRANS 置 1。在 BYTENUM-1 个字节接收完成后，接收的下一个字节将与 I2C_PEC 寄存器中的数据进行比较。如果校验值不匹配，将自动产生 NACK 信号；如果校验值匹配将自动产生 ACK 信号，将忽略 NACKEN 位的值。当接收到 PEC 字节时，PEC 字节会存到 I2C_RDATA 寄存器中，RBNE 位将置 1。如果 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：

1. 在 RELAOD 位置 1 之后，PECTRANS 不可以被修改。

2. SMBus 作为从机发送器，并且使用 PEC 功能时，必须在检测到 START 信号前配置 BYTENUM[7:0]。

图 28-22. 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

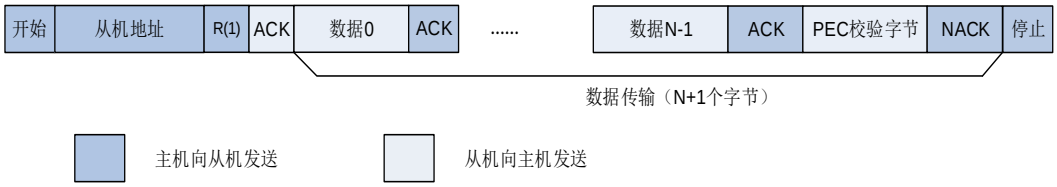
字节之后会给出 NACK 响应。

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 28-23. SMBus 主机接收器和从机发送器通信流程



28.3.11. 从省电模式唤醒

当 I2C 地址匹配成功时，MCU 从睡眠模式、深度睡眠模式被唤醒。为了将 MCU 从这些省电模式唤醒，I2C_CTL0 寄存器中 WUEN 位必须置 1，同时 I2CCLK 时钟源选择 IRC8M。在深度睡眠模式下，IRC8M 关闭。当 I2C 检测到 START 信号时，IRC8M 打开，I2C 会将 SCL 拉低直到 IRC8M 被唤醒。在接收地址期间，IRC8M 为 I2C 提供时钟。当地址匹配时，在 MCU 唤醒期间，I2C 的 SCL 线被拉低。当 ADDSEND 清除时，SCL 线被释放，数据传输过程恢复正常。如果检测到的地址不匹配，IRC8M 会再次关闭，MCU 将不会被唤醒。

只有地址匹配中断（ADDMIE = 1）能唤醒 MCU。如果 I2C 的时钟源是系统时钟，或者 WUEN = 0，IRC8M 在接收到 START 信号之后将不会打开。当从省电模式唤醒使能时，数字滤波器必须禁能，I2C_CTL0 寄存器中 SS 位也必须清 0。如果禁止从省电模式唤醒（WUEN = 0），则在进入省电模式之前必须禁能 I2C 外设（I2CEN = 0）。

28.3.12. 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。

28.3.13. I2C 错误和中断

I2C 错误标志如*表 28-4. I2C 错误标志*所示。

表 28-4. 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 28-5. I2C 中断事件*所示。

表 28-5. 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	
PEC 错误	PECERR	
超时错误	TIMEOUT	
SMBus 报警	SMBALT	

28.3.14. 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。

28.4. I2C 寄存器

I2C0 基地址：0x4000 5400

I2C1 基地址：0x4000 5800

I2C2 基地址：0x4000 C000

I2C3 基地址：0x4000 5C00

28.4.1. 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	WUEN	SS	SBCTL
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DENR	DENT	保留	ANOFF	DNF[3:0]				ERRIE	TCIE	STPDETIE	NACKIE	ADDMIE	RBNEIE	TIE	I2CEN
rw	rw		rw	rw				rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:24	保留	必须保持复位值。
23	PECEN	PEC 计算开关。 0：PEC 计算关闭。 1：PEC 计算打开。
22	SMBALTEN	SMBus 报警使能。 0：从机模式下 SMBA 引脚高电平或主机模式下 SMBus 报警引脚 SMBA 禁能。 1：从机模式下 SMBA 引脚低电平或主机模式下 SMBus 报警引脚 SMBA 使能。
21	SMBDAEN	SMBus 设备默认地址使能。 0：设备默认地址禁能，对默认地址 0b1100001x 进行 NACK 应答。 1：设备默认地址使能，对默认地址 0b1100001x 进行 ACK 应答。
20	SMBHAEN	SMBus 主机地址使能。 0：主机地址禁能，对地址 0b0001000x 进行 NACK 应答。 1：主机地址使能，对地址 0b0001000x 进行 ACK 应答。

19	GCEN	是否响应对地址（0x00）的广播呼叫。 0：从机不响应广播呼叫。 1：从机将响应广播呼叫。
18	WUEN	使能从省电模式中唤醒，包含睡眠模式和深度睡眠模式。 当 MCU 从省电模式唤醒时该位清零。 0：禁止从省电模式中唤醒。 1：使能从省电模式中唤醒。 **注意：**当 DNF[3:0] = 0 时，WUEN 才能被置 1。
17	SS	在从机模式下数据未就绪时是否将 SCL 拉低。 软件置 1 和清 0。 0：拉低 SCL 1：不拉低 SCL **注意：**在主机模式下，该位必须为 0。该位只能在 I2CEN=0 时被修改。
16	SBCTL	从机模式下字节控制。 该位用于在从机模式下使能硬件字节控制。 0：从机模式下字节控制禁能。 1：从机模式下字节控制使能。
15	DENR	DMA 接收使能 0：DMA 接收禁能 1：DMA 接收使能
14	DENT	DMA 发送使能 0：DMA 发送禁能 1：DMA 发送使能
13	保留	必须保持复位值。
12	ANOFF	模拟噪声滤波器禁能 0：模拟噪声滤波器使能。 1：模拟噪声滤波器禁能。 **注意：**该位只有在 I2C 禁能（I2CEN=0）时被编程。
11:8	DNF[3:0]	数字噪声滤波器 0000：数字噪声滤波器禁能。 0001：数字噪声滤波使能并且可以滤除脉宽宽度不大于 1 t_{I2CCLK}的尖峰。 ... 1111：数字噪声滤波使能并且可以滤除脉宽宽度不大于 15 t_{I2CCLK}的尖峰。 这些位只能在 I2C 禁能（I2CEN = 0）时修改。
7	ERRIE	错误中断使能 0：错误中断禁能

		1：错误中断使能，当 BERR，LOSTARB，OUERR，PECERR，TIMEOUT 或 SMBALT 位置 1 时，将产生中断。
6	TCIE	传输完成中断使能 0：传输完成中断禁能。 1：传输完成中断使能。
5	STPDETIE	停止信号检测中断使能 0：停止信号（STPDET）检测中断禁能。 1：停止信号（STPDET）检测中断使能。
4	NACKIE	接收到 NACK 应答中断使能 0：接收到 NACK 应答中断禁能。 1：接收到 NACK 应答中断使能。
3	ADDMIE	从机模式下地址匹配中断使能 0：地址匹配中断禁能。 1：地址匹配中断使能。
2	RBNEIE	接收中断使能 0：接收（RBNE）中断禁能。 1：接收（RBNE）中断使能。
1	TIE	发送中断使能 0：发送中断（TI）禁能。 1：发送中断（TI）使能。
0	I2CEN	I2C 外设使能 0：I2C 禁能。 1：I2C 使能。

28.4.2. 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，该位不能被修改，传输过程中软件不能修改该位。
15	NACKEN	从机模式下产生 NACK 0：在接收到新的字节时，发送 ACK。 1：在接收到新的字节时，发送 NACK。 **注意：**该位可由软件置 1，并在以下情况下由硬件清零：NACK 发送完成或检测到 STOP 信号或 ADDSEND 置 1，或 I2CEN=0。当 PEC 使能时，发送 ACK 还是 NACK 与 NACKEN 值无关。当 SS=1 时，且 OUERR 位置 1，NACKEN 的值会被忽略，并且发送 NACK。
14	STOP	I2C 总线上产生一个 STOP 结束信号。 该位由软件置 1，并在 I2CEN=0 或检测到 STOP 信号时由硬件清零。 0：不发送 STOP。 1：发送 STOP。
13	START	I2C 总线上产生一个 START 信号 该位由软件置 1，并在从机地址发送后由硬件清零。当仲裁丢失时，或发生超时错误，或 I2CEN=0 时，该位也可以由硬件清零。将 I2C_STATC 寄存器中 ADDSENDC 位置 1 可以软件清除该位。

0：不发送 START。

1：发送 START。

12	HEAD10R	在主机接收模式下仅执行 10 位地址头读操作。 0：主机发送 10 位从机地址读序列为 START + 10 位地址头（写） + 第二个地址字节 + RESTART + 10 位地址头（读）。 1：主机寻址读序列为 RESTART + 10 位地址头（读）。 **注意：**当 START 位置 1 时，该位不能被修改，传输过程中软件不能修改该位。
11	ADD10EN	主机模式下使能 10 位寻址模式 0：主机工作在 7 位寻址模式下。 1：主机工作在 10 位寻址模式下。 **注意：**当 START 位置 1 时，该位不能被修改，传输过程中软件不能修改该位。
10	TRDIR	主机模式下传输方向 0：主机发送 1：主机接收 **注意：**当 START 位置 1 时，该位不能被修改，传输过程中软件不能修改该位。
9:0	SADDRESS[9:0]	待发送的从机地址 SADDRESS[9:8]：从机地址 9:8 位。 如果 ADD10EN = 0，该位域无效。 如果 ADD10EN = 1，将该位域写入待发送从机地址的 9:8 位。 SADDRESS[7:1]：从机地址 7:1 位。 如果 ADD10EN = 0，在这些位写入待发送 7 位从机地址。 如果 ADD10EN = 1，在这些位写入待发送从机地址的 7:1 位。 SADDRESS0：从机地址 0 位。 如果 ADD10EN = 0，这些位无效。 如果 ADD10EN = 1，在这些位写入待发送从机地址的 0 位。 **注意：**当 START 位置 1 时，该位不能被修改，传输过程中软件不能修改该位。

28.4.3. 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[9:8]		ADDRESS[7:1]							ADDRESS0

rw rw rw rw rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	ADDRESSEN	I2C 地址使能 0：I2C 地址禁能。 1：I2C 地址使能。
14:11	保留	必须保持复位值。
10	ADDFORMAT	I2C 从机地址模式 0：7 位地址。 1：10 位地址。 **注意：**当 ADDRESSEN =1 时，该位不能被改写。
9:8	ADDRESS[9:8]	10 位地址的最高两位 **注意：**当 ADDRESSEN =1 时，该位不能被改写。
7:1	ADDRESS[7:1]	7 位地址或者 10 位地址的第 7-1 位 **注意：** 1. 当 ADDRESSEN =1 时，该位不能被改写。 2. 当从机地址设置为保留地址 0b'1111000 时，如果地址匹配，从机不响应。
0	ADDRESS0	10 位地址的第 0 位 **注意：**当 ADDRESSEN =1 时，该位不能被改写。

28.4.4. 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。

位/位域	名称	描述
31:16	保留	必须保持复位值。

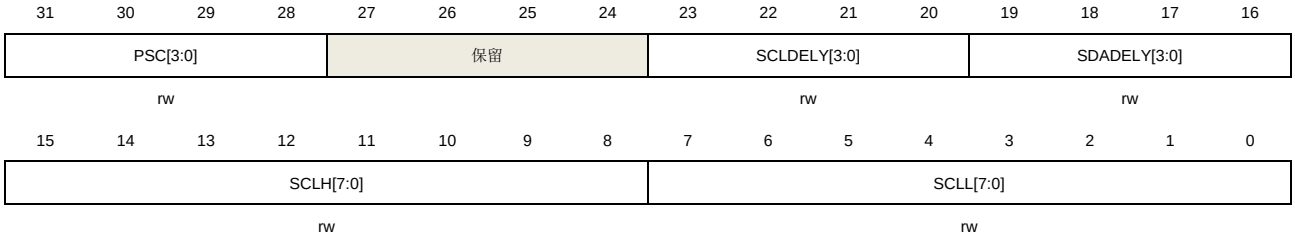
15	ADDRESS2EN	I2C 第二个地址使能 0：I2C 第二个地址禁能 1：I2C 第二个地址使能
14:11	保留	必须保持复位值。
10:8	ADDMSK2[2:0]	ADDRESS2[7:1]掩码 定义接收到的地址哪些位需要与 ADDRESS2[7:1]进行比较，哪些位屏蔽（不比较）。 000：不屏蔽，所有的位都进行比较。 N（001~110）：ADDRESS2[n:1]屏蔽。ADDRESS2[7:n+1]需要进行比较。 111：ADDRESS2[7:1]屏蔽。对于接收到的所有 7 位地址都会进行 ACK 应答，保留地址（0b0000xxx 和 0b1111xxx）除外。 **注意：**当 ADDRESS2EN =1 时，该位不能被改写。如果 ADDMSK2 不等于 0，即使所有位都匹配，I2C 保留地址（0b0000xxx 和 0b1111xxx）也不会进行 ACK 应答。
7:1	ADDRESS2[7:1]	I2C 从机的第二个地址 **注意：** 1. 当 ADDRESS2EN=1 时，该位不能被改写。 2. 当从机地址设置为保留地址 0b'1111000 时，如果地址匹配，从机不响应。
0	保留	必须保持复位值。

28.4.5. 时序寄存器（I2C_TIMING）

地址偏移：0x10
复位值：0x0000 0000

当 I2C（I2CEN=1）被使能时，该寄存器不能被改写。

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:28	PSC[3:0]	时序预分频 为了生成用于数据建立和数据保持的计数器的时钟周期t_{PSC}，这些位用于配置 I2CCLK 时钟预分频。t_{PSC}也用于 SCL 高电平和低电平计数器。 $t_{PSC}=(PSC+1)*t_{I2CCLK}$。

27:24	保留	必须保持复位值。
23:20	SCLDELY[3:0]	数据建立时间 这些位用于在 SDA 边沿和 SCL 上升沿之间生成延时$t_{SCLDELY}$。在主机模式下和在从机模式下 SS=0 时，在$t_{SCLDELY}$期间 SCL 线被拉低。 $t_{SCLDELY}$=(SCLDELY+1)*t_{PSC}。
19:16	SDADELY[3:0]	数据保持时间 这些位用于在 SCL 下降沿和 SDA 边沿之间生成延时$t_{SDADELY}$。在主机模式下和在从机模式下 SS=0 时，在$t_{SDADELY}$期间 SCL 线被拉低。 0x0: $t_{SDADELY}$ = 1*t_{PSC} 0x1-0xF: $t_{SDADELY}$=SDADELY*t_{PSC}
15:8	SCLH[7:0]	SCL 高电平周期 SCL 高电平周期可以通过配置这些位来产生。 t_{SCLH}=(SCLH+1)*t_{PSC}。 **注意：**这些位只能用于主机模式。
7:0	SCLL[7:0]	SCL 低电平周期 SCL 低电平周期可以通过配置这些位来产生。 t_{SCLL}=(SCLL+1)*t_{PSC}。 **注意：**这些位只能用于主机模式。

28.4.6. 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。

28.4.7. 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。

28.4.8. 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			

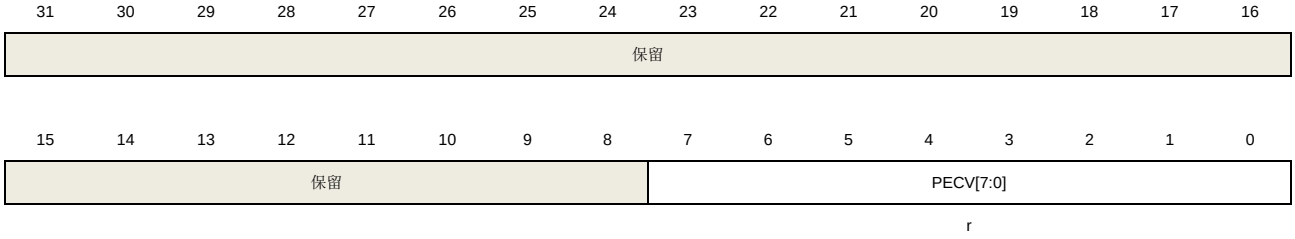
位/位域	名称	描述
31:14	保留	必须保持复位值。
13	SMBALTC	SMBus 报警标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 SMBALT 位。
12	TIMEOUTC	TIMEOUT 标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 TIMEOUT 位。
11	PECERRC	PEC 错误标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 PECERR 位。
10	OUERRC	上溢 / 下溢标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 OUERR 位。

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。

28.4.9. 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。

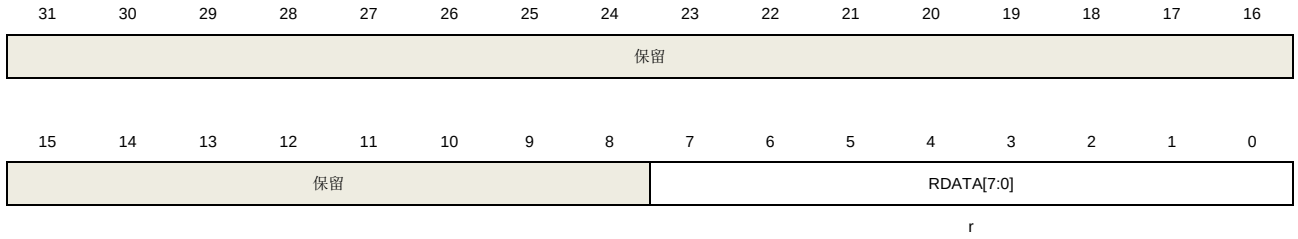


位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	PECV[7:0]	在 PEC 使能时，由硬件计算出来的 PEC 值。 当 I2CEN = 0 时，PECV 由硬件清零。

28.4.10. 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。

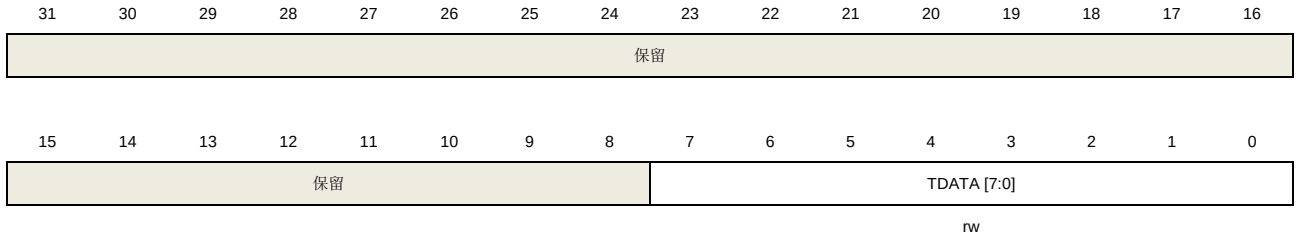


位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	RDATA[7:0]	接收到的数据

28.4.11. 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。

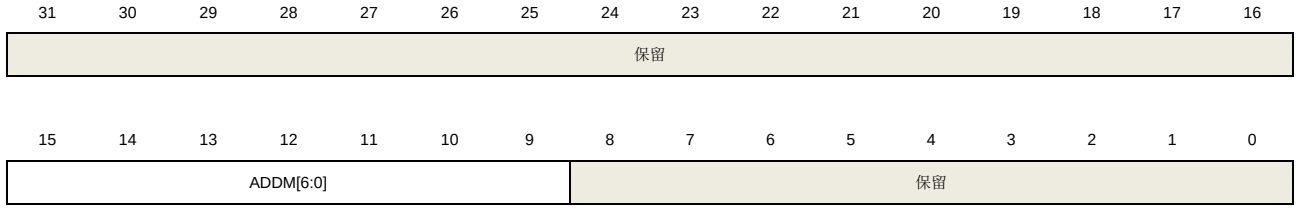


位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	TDATA[7:0]	发送的数据

28.4.12. 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。



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。

29. 控制器局域网络（CAN）

29.1. 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。

29.2. 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。

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

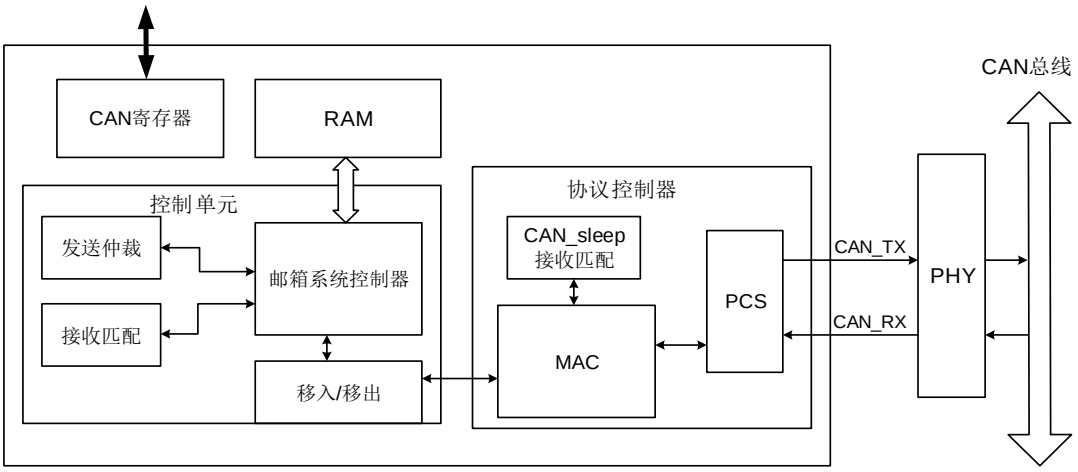
- 接收私有过滤寄存器用于每个接收邮箱或者接收FIFO；
- 接收邮箱公有过滤寄存器用于接收邮箱，接收FIFO公有过滤寄存器用于接收FIFO；
- 支持接收优先级，可配置在匹配阶段的接收邮箱和接收FIFO的优先级；

- 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。

29.3. 功能说明

CAN 模块结构框图如*图 29-1. CAN 模块结构框图*所示。

图 29-1. 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。

29.3.1. 邮箱描述符

邮箱描述符如*表29-1. 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 29-1. 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 29-3. 接收邮箱 CODE*** 和***表 29-4. 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 29-2. 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
11	20
12	24
13	32
14	48
15	64

表 29-3. 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 OVERRUN	-	-	表示邮箱正在更新。

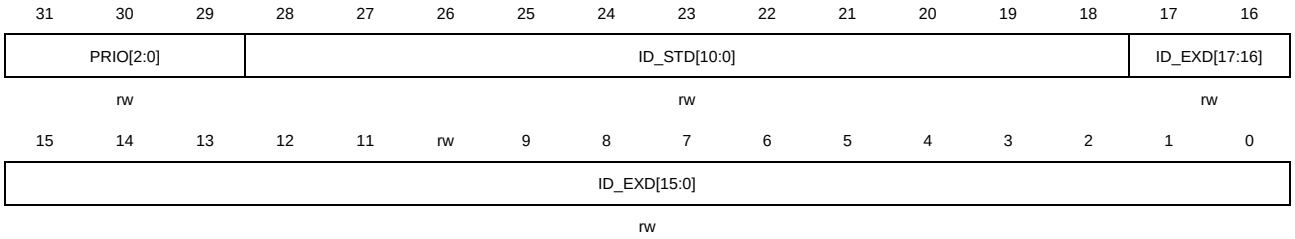
1. 完成服务：邮箱被CPU读取过，并且通过读取CAN_TIMER寄存器或者读取其他邮箱的方式解锁了该邮箱。
2. 远程请求帧存储位，参考***控制寄存器2（CAN_CTL2）***。
3. 邮箱代码为0b1010的不可被中止。
4. 对于接收邮箱，如果CODE[0]位置位，则对应的邮箱将不参与接收匹配过程。注意，对于Tx邮箱，读取时应该忽略BUSY位，除非设置了CAN_CTL0寄存器中的MST位。

表 29-4. 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。
	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



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 29-5. 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

29.3.2. 接收 FIFO 描述符

接收FIFO描述符如*表29-6. 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0xDC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。

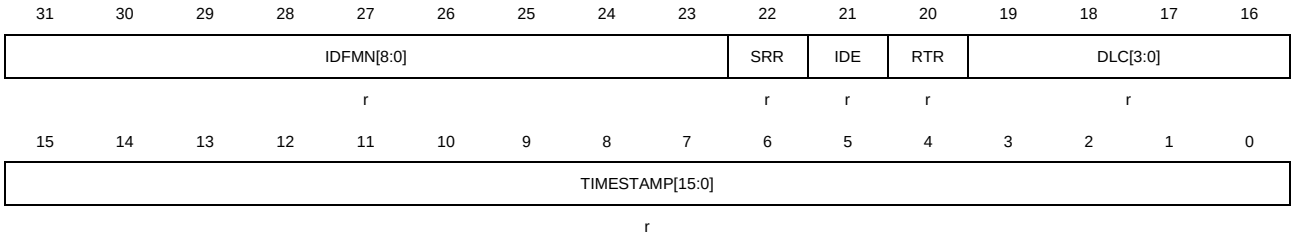
复位时默认标识符过滤表包含8个过滤元素，从FDES4到FDES11。

表 29-6. 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

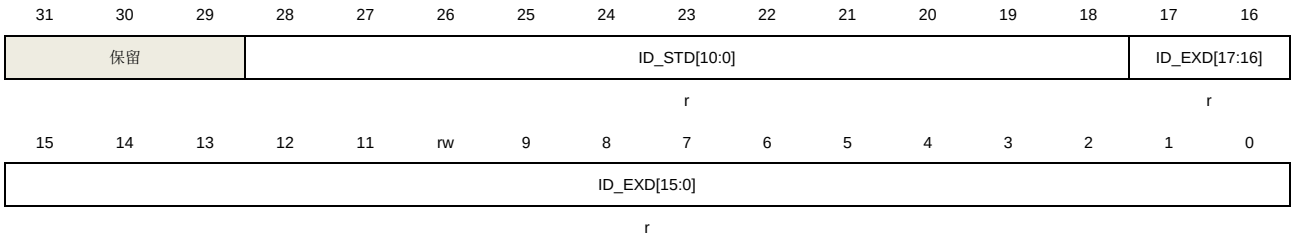
地址偏移：0x80



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

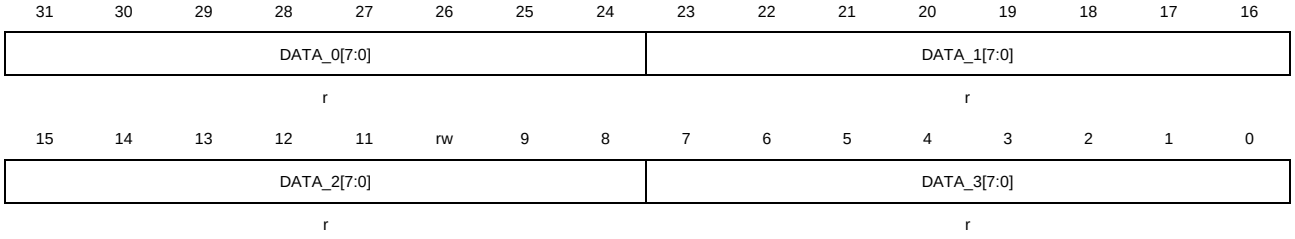


位/位域	名称	描述

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

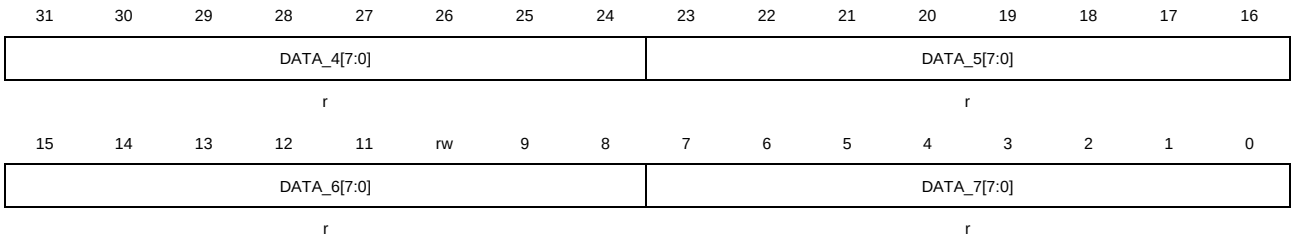
地址偏移：0x88



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



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。

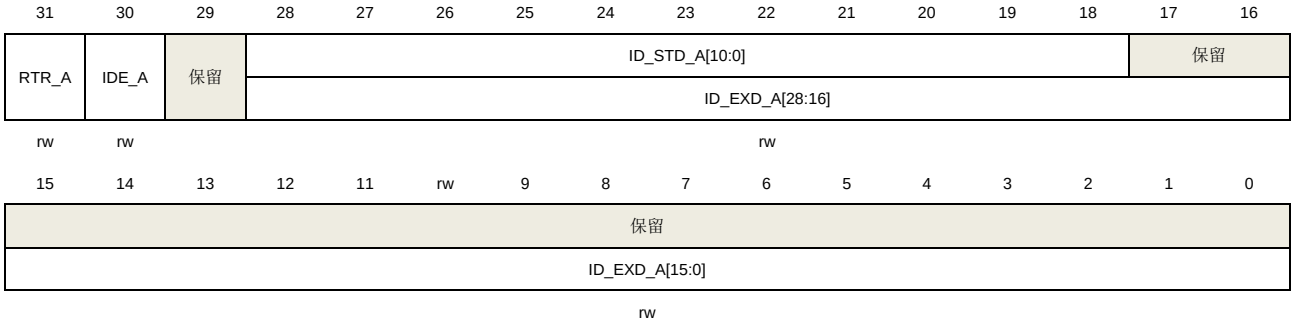
FDESx：接收 FIFO 描述符字 x（x = 4..107）

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。

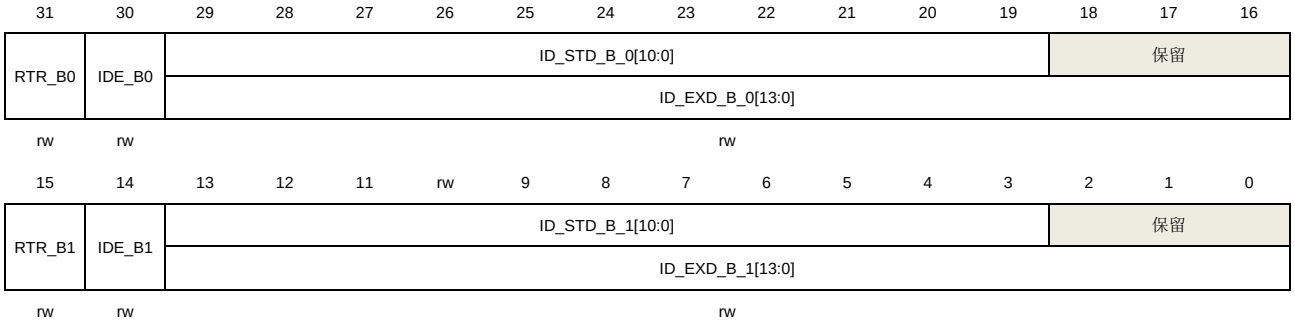
格式A模式：



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：



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。

29.3.3. 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，或者发出进入虚拟联网模式的请求，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。

29.3.4. 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LPS和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LPS位和PNS位，恢复到正常模式，CAN模块将重新与CAN总线同步。

29.3.5. 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29-5. 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*接收邮箱失活*或*发送邮箱失活*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29-7. 当本地优先级禁用时的邮箱仲裁值（32位）*和*表29-8. 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29-7. 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29-8. 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 29-7. 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 29-8. 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。

29.3.6. 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29-6. 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。

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或者最后一个非空状态的匹配邮箱为获胜者。

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29-9. 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 29-9. 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29-10. 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 29-10. 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。

29.3.7. 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29-11. 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。
- 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。
- 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。

29.3.8. 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。正在发送CAN FD帧的节点在发送该帧的数据阶段时不执行再同步。

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29-2. 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：

$$t_{compensation} = t_{measure} + t_{offset} \quad (29\text{-}1)$$

其中：

$$t_{offset} = TDCO[4:0] \times t_{CANCLK} \quad (29\text{-}2)$$

$$t'_{offset} = t_{PBS1_FD} + t_{PTS_FD} + t_{SYNC_SEG} \quad (29\text{-}3)$$

$$t_{PBS1_FD} = (DPBS1[2:0] + 1) \times t_{q_FD} \quad (29\text{-}4)$$

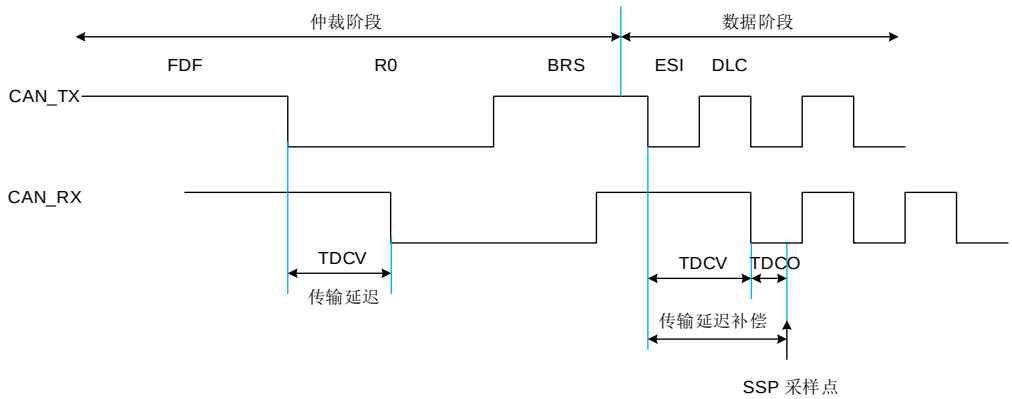
$$t_{PTS_FD} = DPTS[4:0] \times t_{q_FD} \quad (29\text{-}5)$$

$$t_{q_FD} = (DBAUDPSC[9:0] + 1) \times t_{CANCLK} \quad (29\text{-}6)$$

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 29-2. 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。

29.3.9. 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。

29.3.10. 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（$1\times t_q$）。

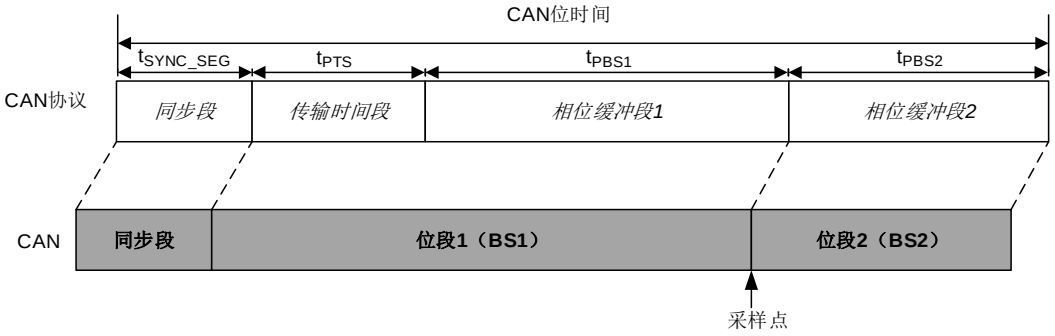
位段1（BS1）：该段包括CAN协议中的传播时间段和相位缓冲段1。该段可自动延长来补偿网络节点的频率不同引起的相位正漂移。

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29-3. CAN位时间*所示。

图 29-3. 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。

有效跳变沿定义为只有当在前一个采样点检测到的总线状态为隐性（recessive）时，一个位时间

内从隐性位到显性位的第一次转变。

如果有效跳变沿在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。

注：有关硬同步和再同步的详细说明，请参考ISO 11898标准。

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IRC64MDIV内部时钟。

CAN波特率计算如下：

$$BaudRate = \frac{1}{CAN\ Bit\ Time} \tag{29-7}$$

$$CAN\ Bit\ Time = t_{SYNC_SEG} + t_{PTS} + t_{PBS1} + t_{PBS2} \tag{29-8}$$

其中：

$$t_{SYNC_SEG} = 1 \times t_q \tag{29-9}$$

$$t_{PTS} = (N_{PTS} + 1) \times t_q \ \ or \ \ t_{PTS} = N_{DPTS} \times t_q \tag{29-10}$$

$$t_{PBS1} = (N_{PBS1} + 1) \times t_q \tag{29-11}$$

$$t_{PBS2} = (N_{PBS2} + 1) \times t_q \tag{29-12}$$

$$t_q = (N_{BAUDPSC} + 1) \times t_{CANCLK} \tag{29-13}$$

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。

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。

29.3.11. 中断

CAN中断事件与标志如*表29-11. 中断事件*所示。

表 29-11. 中断事件

中断事件		标志			使能控制			
		位		寄存器	使能位	控制位	使能寄存器	控制寄存器
离线		BOF		CAN_ERR1	BOIE		CAN_CTL1	
离线恢复		BORF			BORIE		CAN_CTL2	
错误汇总	位隐性错误	ERRSF	BRERR		ERRSIE		CAN_CTL1	
	位显性错误		BDERR					
	ACK错误		ACKERR					
	CRC错误		CRCERR					
	格式错误		FMERR					

	填充错误		STFERR					
FD帧数据位时间的错误汇总	位隐性错误	ERRFSF	BRFERR		ERRFSIE		CAN_CTL2	
	位显性错误		BDFERR					
	CRC错误		CRCFERR					
	格式错误		FMFERR					
	填充错误		STFFERR					
Tx错误警告		TWERRIF			TWERRIE	WERREN	CAN_CTL1	CAN_CTL0
Rx错误警告		RWERRIF			RWERRIE			
匹配唤醒		WMS		CAN_PN_STAT	WMIE		CAN_PN_CTL0	
超时唤醒		WTOS			WTOIE			
邮箱成功发送或接收帧		所有位		CAN_STAT	所有位	RFEN = 0	CAN_INTEN	CAN_CTL0
		MSx			MIEx	RFEN = 1		
Rx FIFO非空		MS5_RFNE			MIE5	RFEN = 1 & DMAEN = 0		
Rx FIFO警告		MS6_RFW			MIE6			
Rx FIFO溢出		MS7_RFO			MIE7			

29.4. 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PNEN位和SLEPMOD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。

29.5. CAN 寄存器

CAN0基地址：0x4001 A000

CAN1基地址：0x4001 B000

CAN2基地址：0x4001 C000

29.5.1. 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 该位不会被CAN_CTL0寄存器中的软件复位SWRST位影响。 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。 当CAN_CTL0寄存器的LPS位置位时，软件复位不起作用。 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

29.5.2. 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	BSSSEL	BSPMOD	ABORDIS	TSYNC	MTO	MMOD	保留		
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	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	保留	必须保持复位值。
8	BSSSEL	位采样同步选择 0: CAN总线采样的两阶段同步 1: CAN总线采样单级同步
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。

29.5.3. 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。

29.5.4. 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 **注：**对于标准帧，MDF18~MFD28 位域用于过滤邮箱描述符 ID 字段。

29.5.5. 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

29.5.6. 错误寄存器 1（CAN_ERR1）

地址偏移：0x20
复位值：0x0004 0009

该寄存器只能按字（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。

29.5.7. 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

29.5.8. 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*。

29.5.9. 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 29-12. 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

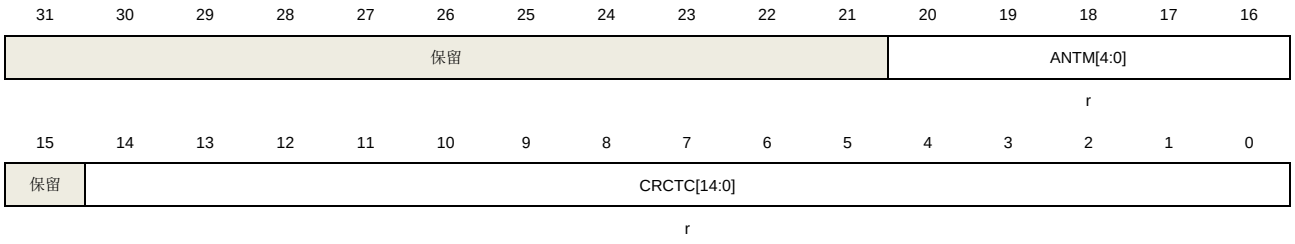
		如同数据帧一样存储起来，而不自动发送远程应答帧。
16	IDERTR_RMF	邮箱接收时IDE和RTR域的过滤类型 该位定义了接收邮箱描述符中IDE和RTR域与接收的位的匹配类型。 0：总是比较IDE域，从不比较RTR域。忽略相关过滤寄存器中的过滤数据配置。 1：过滤比较IDE和RTR域，使用相关过滤寄存器中的过滤数据配置。
15	ITSRC	内部计数器时钟源 0：CAN波特率 1：TRIGSEL输出的外部触发CANx_EX_TIME_TICK
14	PREEN	CAN规范中的协议异常检测使能 0：禁能协议异常检测 1：使能协议异常检测
13	保留	必须保持复位值。
12	ISO	ISO CAN 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。

29.5.10. 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。



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。

29.5.11. 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

29.5.12. 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。

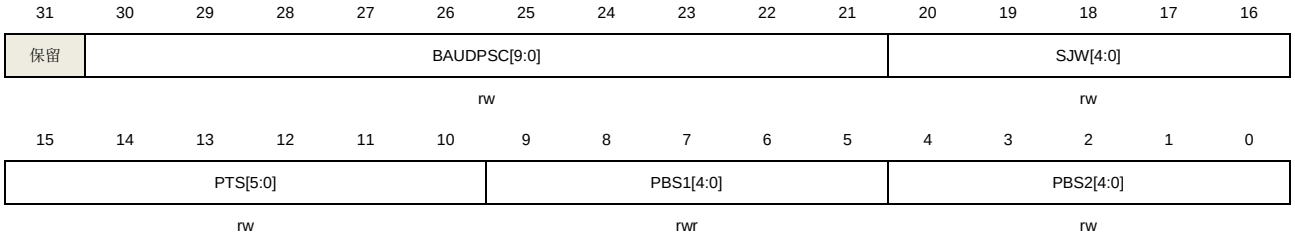
29.5.13. 位时间寄存器（CAN_BT）

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。



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= 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

29.5.14. 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

29.5.15. 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

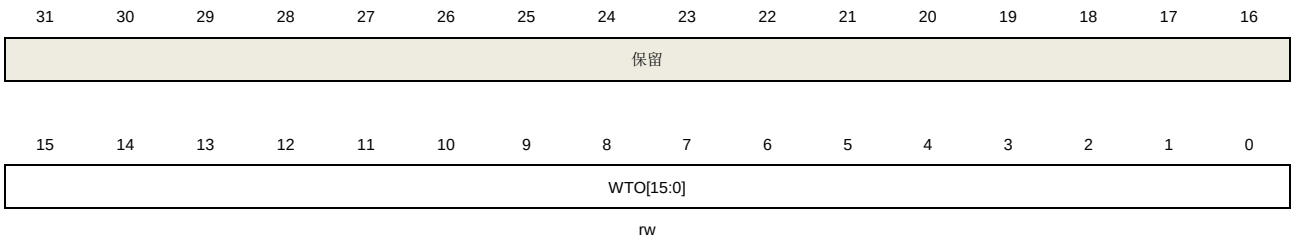
29.5.16. 虚拟联网模式超时寄存器（CAN_PN_TO）

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。



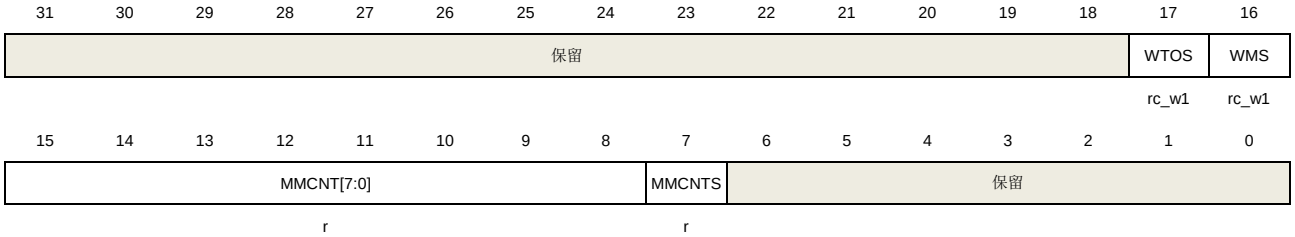
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	WTO[15:0]	超时唤醒 该超时值按照CAN位时间的64倍进行计数。默认关闭超时唤醒。

29.5.17. 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。



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。

29.5.18. 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。

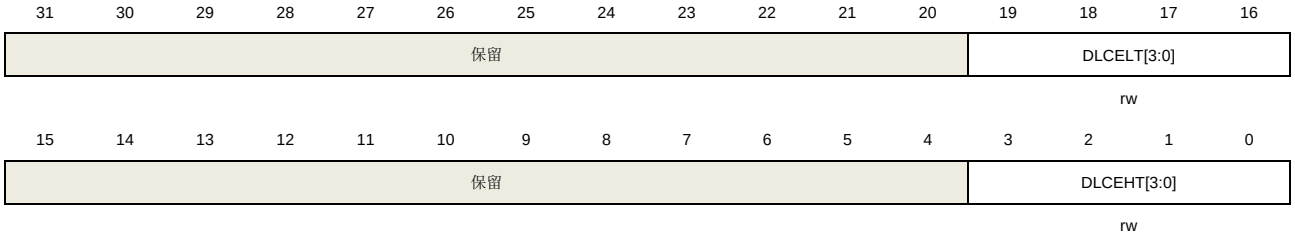
29.5.19. 虚拟联网模式期望 DLC 寄存器（CAN_PN_EDLC）

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。



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

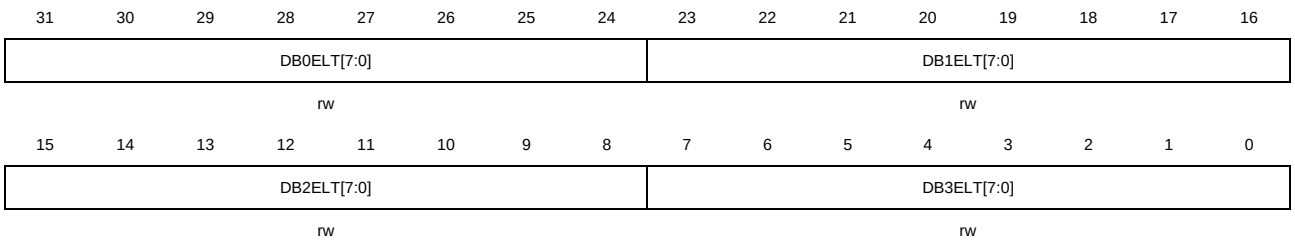
29.5.20. 虚拟联网模式期望数据低字 0 寄存器（CAN_PN_EDL0）

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。



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。

29.5.21. 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。

29.5.22. 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。

29.5.23. 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）。

29.5.24. 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。

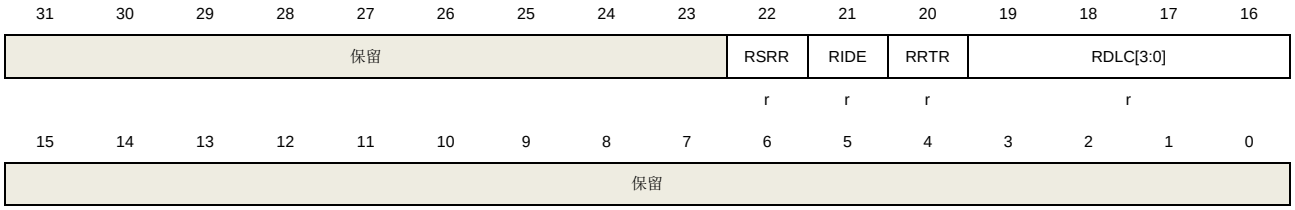
23:16	DB5FD_HTF[7:0]	在虚拟联网模式下的数据字节5过滤数据 / 数据字节5期望上限值 参考DB3FD_EHT[7:0]描述。
15:8	DB6FD_HTF[7:0]	在虚拟联网模式下的数据字节6过滤数据 / 数据字节6期望上限值 参考DB3FD_EHT[7:0]描述。
7:0	DB7FD_HTF[7:0]	在虚拟联网模式下的数据字节7过滤数据 / 数据字节7期望上限值 参考DB3FD_EHT[7:0]描述。

29.5.25. 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。



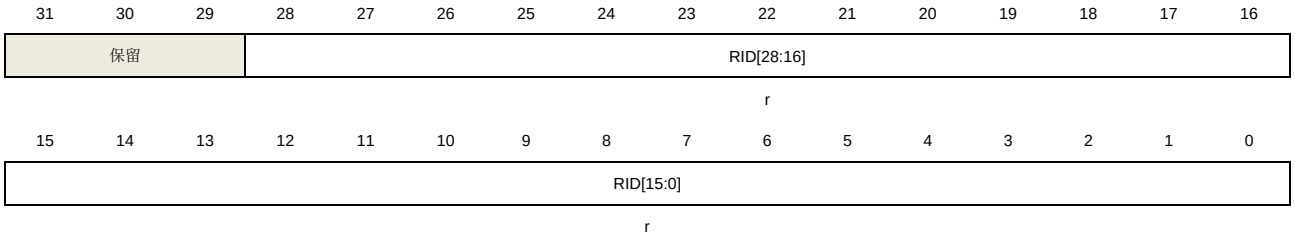
位/位域	名称	描述
31:23	保留	必须保持复位值。
22	RSRR	接收到的SRR位
21	RIDE	接收到的IDE位 0：帧为标准格式 1：帧为扩展格式
20	RRTR	接收到的RTR位 0：帧为数据帧 1：帧为远程帧
19:16	RDLC[3:0]	接收到的DLC域 该位域指示了有效的数据字节长度。
15:0	保留	必须保持复位值。

29.5.26. 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。



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。

29.5.27. 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。

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

29.5.28. 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

29.5.29. 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。

29.5.30. 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

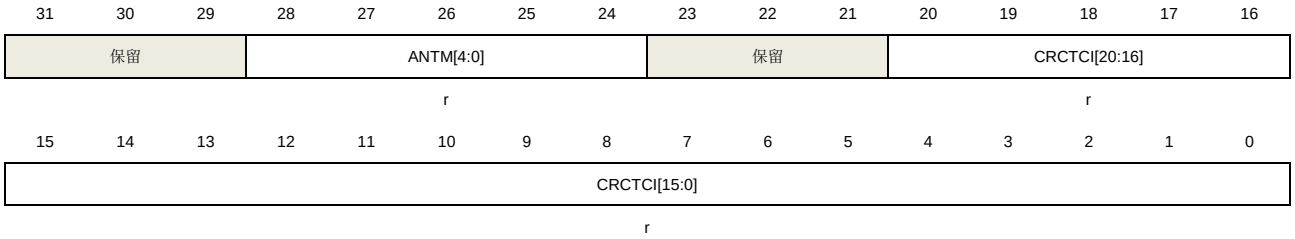
31:30	保留	必须保持复位值。
29:20	DBAUDPSC[9:0]	数据位时间的波特率分频系数 CAN数据位时间的波特率分配系数= DBAUDPSC[9:0] + 1。
19	保留	必须保持复位值。
18:16	DSJW[2:0]	数据位时间的再同步补偿宽度 再同步补偿占用的时间单元数量 = DSJW[2:0] + 1
15	保留	必须保持复位值。
14:10	DPTS[4:0]	数据位时间的传播时间段 传播时间段占用的时间单元数量 = DPTS[4:0]
9:8	保留	必须保持复位值。
7:5	DPBS1[2:0]	数据位时间的相位缓冲段1 相位缓冲段 1 占用的时间单元数量 = DPBS1[2:0] + 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

29.5.31. 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。



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。

30. 串行外设接口（SPI）

30.1. 简介

SPI模块可以通过SPI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。

30.2. 主要特性

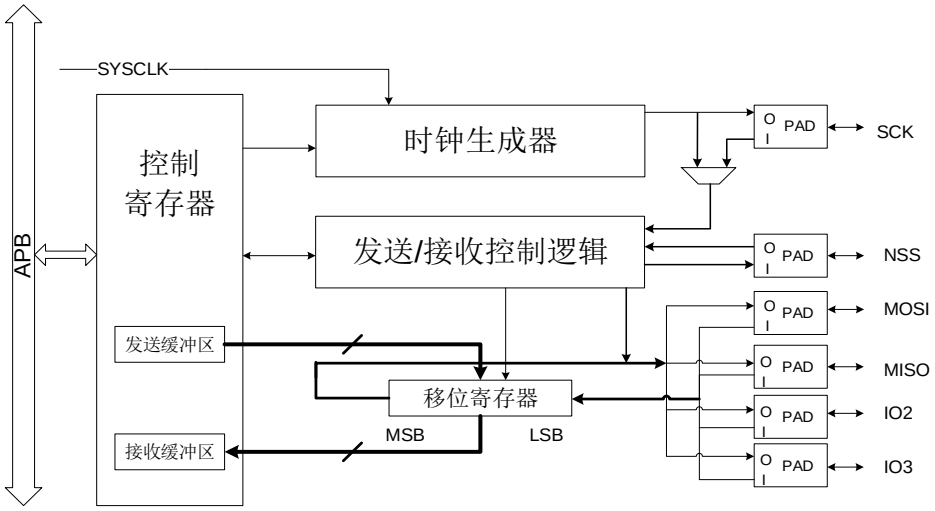
30.2.1. SPI 主要特性

- 具有全双工和单工模式的主从操作。
- 32位宽度，独立的发送和接收缓冲区。
- 4位至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）。

30.3. SPI 功能说明

30.3.1. SPI 结构框图

图 30-1. SPI 结构框图



30.3.2. SPI 信号线描述

常规配置（非 SPI 四线模式）

表 30-1. SPI 信号描述

引脚名称	方向	描述
SCK	I/O	主机：SPI 时钟输出 从机：SPI 时钟输入
MISO	I/O	主机：数据接收线 从机：数据发送线 主机双向线模式：不使用 从机双向线模式：数据发送和接收线
MOSI	I/O	主机：数据发送线 从机：数据接收线 主机双向线模式：数据发送和接收线 从机双向线模式：不使用
NSS	I/O	软件 NSS 模式：不使用 主机硬件 NSS 模式：NSSDRV=1 时，为 NSS 输出，适用于单主机模式；NSSDRV=0 时，为 NSS 输入，适用于多主机模式。

		从机硬件 NSS 模式：为 NSS 输入，作为从机的片选信号。

SPI 四线配置

SPI默认配置为单路模式，当SPI_QCTL中的QMOD位置1时，配置为SPI四线模式（只适用于SPI0）。SPI四线模式只能工作在主机模式。

在SPI四线模式下，SPI通过以下6个引脚与外部设备连接：

表 30-2. 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

30.3.3. 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。

在SPI0常规模式中，通过SPI_CTL1中的DZ[3:0]位域配置数据长度，可以设置为4位至16位。该设置不仅适用于数据的发送也适用于数据的接收。不论设置的数据长度是多少，对FIFO的读访问必须与SPI_CTL1寄存器中的BYTEN位设置的对齐。在SPI四线模式下，数据长度固定为8位。

同样，通过设置SPI_CTL0中的LF位可以配置数据顺序，当LF=1时，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。

当访问SPI_DATA寄存器时，数据帧总是右对齐成一个字节（如果数据长度小于或等于一个字节）或一个半字。通讯时，只有数据长度内的位会随时钟输出。

图 30-2. SPI0 常规模式下的时序图

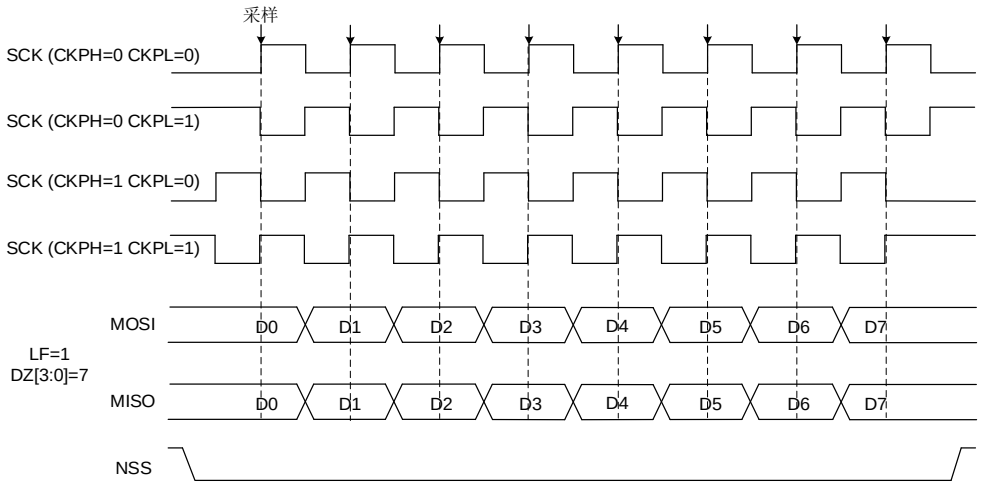
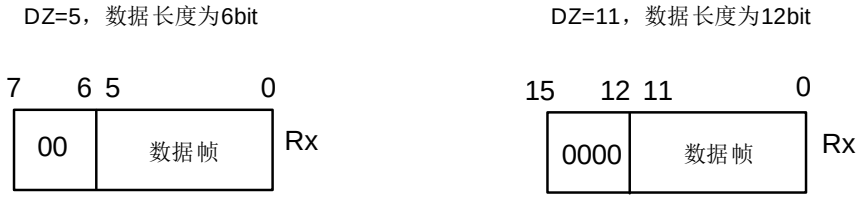
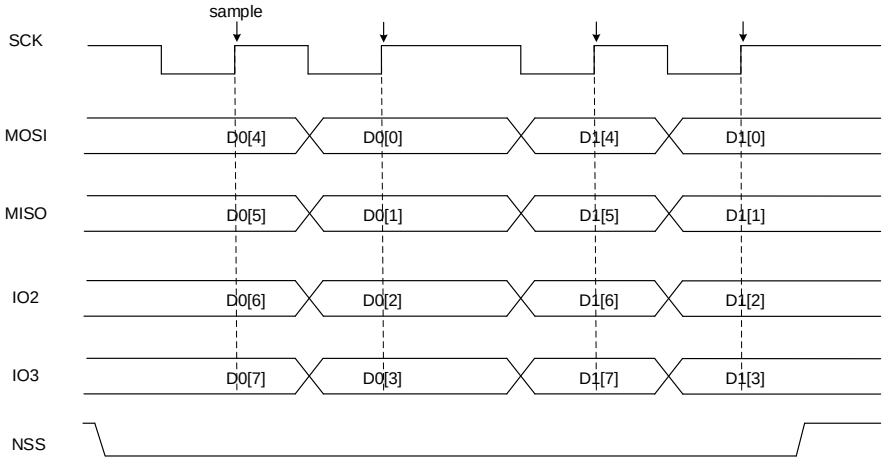


图 30-3. SPI0 数据帧右对齐示意图



通过设置SPI_CTL0中的LF位可以配置数据顺序，当LF=1时，SPI1先发送LSB位，当LF=0时，则先发送MSB位。在TI模式中，数据顺序固定为先发MSB位。

图 30-4. SPI 四线模式下的 SPI 时序图（CKPL=1, CKPH=1, LF=0）

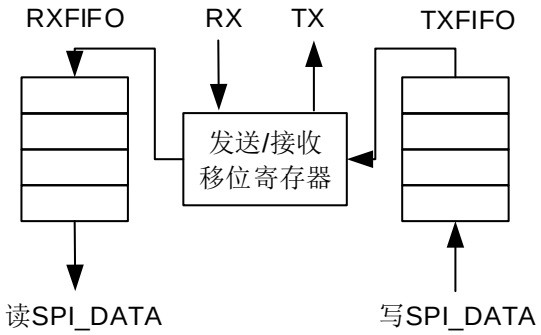


30.3.4. 独立发送和接收缓冲区

独立的32位的接收缓冲区（RXFIFO）和发送缓冲区（TXFIFO）分别用于SPI数据传输的不同方向，

它们使得SPI可以连续工作。

图 30-5. 发送/接收缓冲区



当当前TXFIFO的存储量小于或等于整体存储能力的一半时，TXFIFO被视为空(1)并且此时TBE被硬件置1。当TBE位置位时，向SPI_DATA寄存器写数据，会把数据存入发送FIFO的末尾。当当RXFIFO被视为非空(2)时硬件将RBNE位置1。当RBNE位置位时，从SPI_DATA寄存器读数据，将从接收FIFO获得最早数据。

注意：

（1）对于SPI，TXFIFO空意味着TXFIFO当前的存储量小于或等于TXFIFO整体存储能力的一半。TXFIFO满的意义与之相反。所以，当数据长度不大于8位时，TXFIFO最多能存储3个数据帧。如果下文出现TXFIFO空或者满，如无特殊说明，意义与这里说明的相同。

（2）对于SPI，RXFIFO空的意义分为以下两种情况：如果SPI_CTL1中BYTEN位为1时，RXFIFO空意味着当前RXFIFO的存储量小于RXFIFO整体存储能力的四分之一。此时，当数据长度不大于8位时，RXFIFO最多可以存储4个数据帧。如果SPI_CTL1中BYTEN位为0时，RXFIFO空意味着当前RXFIFO的存储量小于RXFIFO整体存储能力的一半。RXFIFO满的意义与之相反。如果下文出现RXFIFO空或者满，如无特殊说明，意义与这里说明的相同。

数据合并

在SPI_CTL1寄存器中DZ[3:0]配置传输数据位宽为8位或者小于8位的情况下，通过配置SPI_CTL1寄存器中BYTEN位为0，开启数据合并传输模式功能。在配置SPI_CTL1寄存器中DZ[3:0]配置传输数据位宽为小于等于8位时，该功能可以实现当对SPI_DATA寄存器进行16位写访问时，两个数据帧的发送是并行方式而不是串行方式。

同样的，在接收端接收器通过对SPI_DATA的一次16位读访问，获取这两个数据帧，并且这两帧数据在接收时，仅会产生一个RBNE事件。

注意：当被传输的数据为奇数个字节时，在发送端，需要用8位访问SPI_DATA，发出最后一个数据帧。在接收端，为了产生最后一个字节的RBNE事件，接收器必须在接收最后一个数据帧时，改变BYTEN位。

30.3.5. 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 30-3. 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保持高电平，当发送或接收过程开始时，NSS变为低电平。当禁用SPI时，NSS变为高电平。

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 30-4. 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

30.3.6. SPI 运行模式

表 30-5. SPI 运行模式

模式	描述	寄存器配置	数据引脚用法
MFD	全双工主机模式	MSTMOD = 1 RO = 0 BDEN = 0 BDOEN：不要求	MOSI：发送 MISO：接收
MTU	单向线连接主机发送模式	MSTMOD = 1 RO = 0 BDEN = 0 BDOEN：不要求	MOSI：发送 MISO：不使用
MRU	单向线连接主机接收模式	MSTMOD = 1 RO = 1 BDEN = 0 BDOEN：不要求	MOSI：不使用 MISO：接收
MTB	双向线连接主机发送模式	MSTMOD = 1 RO = 0 BDEN = 1 BDOEN = 1	MOSI：发送 MISO：不使用
MRB	双向线连接主机接收模式	MSTMOD = 1	MOSI：接收

模式	描述	寄存器配置	数据引脚用法
		RO = 0 BDEN = 1 BDOEN = 0	MISO：不使用
SFD	全双工从机模式	MSTMOD = 0 RO = 0 BDEN = 0 BDOEN：不要求	MOSI：接收 MISO：发送
STU	单向线连接从机发送模式	MSTMOD = 0 RO = 0 BDEN = 0 BDOEN：不要求	MOSI：不使用 MISO：发送
SRU	单向线连接从机接收模式	MSTMOD = 0 RO = 1 BDEN = 0 BDOEN：不要求	MOSI：接收 MISO：不使用
STB	双向线连接从机发送模式	MSTMOD = 0 RO = 0 BDEN = 1 BDOEN = 1	MOSI：不使用 MISO：发送
SRB	双向线连接从机接收模式	MSTMOD = 0 RO = 0 BDEN = 1 BDOEN = 0	MOSI：不使用 MISO：接收

图 30-6. 典型的全双工模式连接

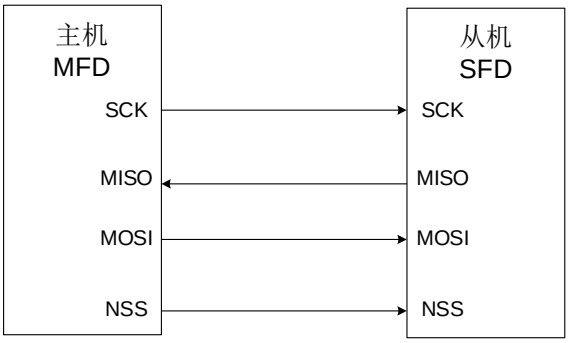


图 30-7. 典型的单工模式连接（主机：接收，从机：发送）

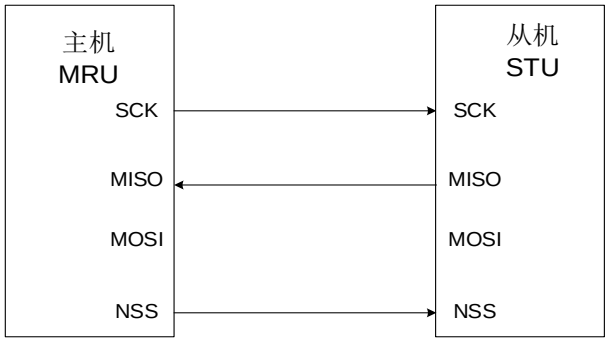


图 30-8. 典型的单工模式连接（主机：只发送，从机：接收）

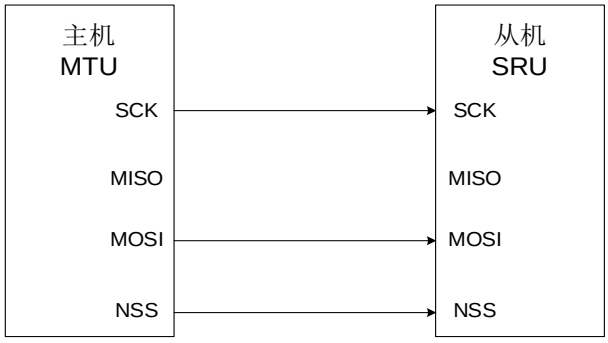
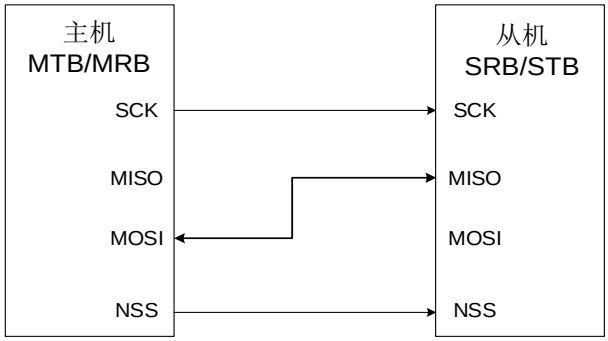


图 30-9. 典型的双向线连接



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. 配置时钟时序（SPI_CTL0中的CKPL位和CKPH位）。
3. 配置帧格式（SPI_CTL0中的LF位）。
4. 配置数据格式（SPI_CTL1中的DZ[3:0]位域）和SPI_DATA的访问方式（SPI_CTL1中的

BYTEN）。

5. 按照上文*NSS功能*的描述，根据应用程序的需求，配置NSS模式（SPI_CTL0中的SWNSSEN位和NSSDRV位）。
6. 如果工作在TI模式，需要将SPI_CTL1中的TMOD位置1，否则，忽略此步骤。
7. 如果工作在 NSSP 模式，需要将 SPI_CTL1 中的 NSSP 位置 1，否则，忽略此步骤。
8. 根据*表30-5. SPI运行模式*，配置MSTMOD位、RO位、BDEN位和BDOEN位。
9. 根据应用程序的需求，配置TXDMA_ODD和RXDMA_ODD位。
10. 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。
11. 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 30-10. 主机 TI 模式在不连续发送时的时序图

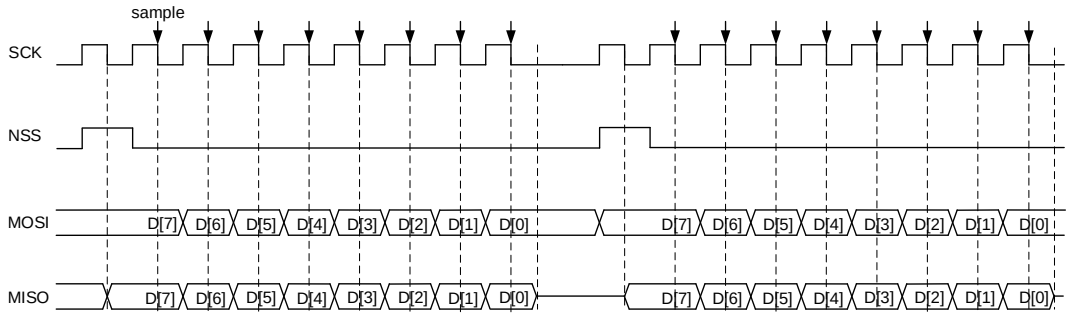
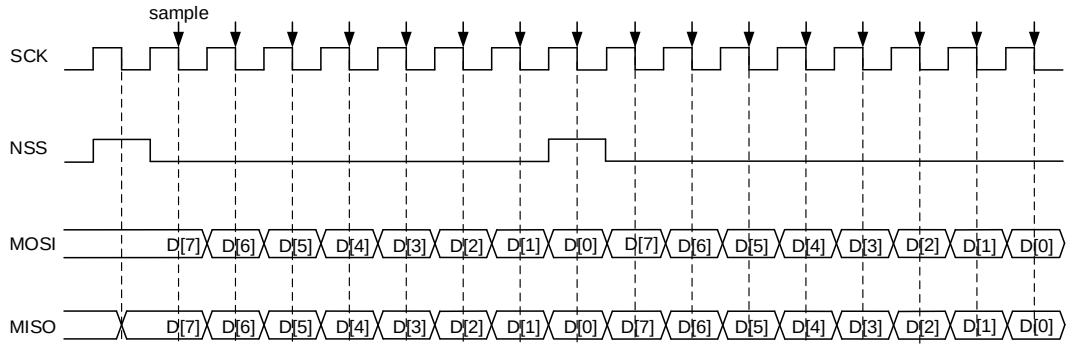
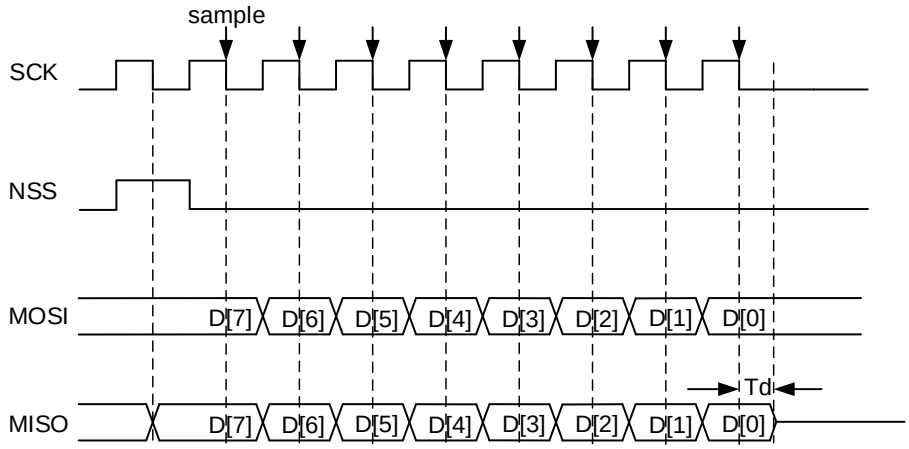


图 30-11. 主机 TI 模式在连续发送时的时序图



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 30-12. 从机 TI 模式时序图



在从机TI模式中，在SCK信号的最后一个上升沿，从机开始发送最后一个字节的LSB位，在半位的时间之后，主机开始采集数据。为了确保主机采集到正确的数据，在释放该引脚之前，从机需要在SCK信号的下降沿之后继续驱动该位一段时间，这段时间称为T_d，T_d通过SPI_CTL0寄存器中的PSC[2:0]位来设置。

$$T_d = \frac{T_{bit}}{2} + 5 * T_{pclk} \quad (30\text{-}1)$$

例如，如果PSC[2:0] = 010，那么T_d数值为$9*T_{pclk}$。

在从机模式下，从机需要监视NSS信号，如果检测到错误的NSS信号，将会置位FERR标志位。例如，NSS信号在一个字节的中间位发生翻转。

NSS 脉冲模式操作流程

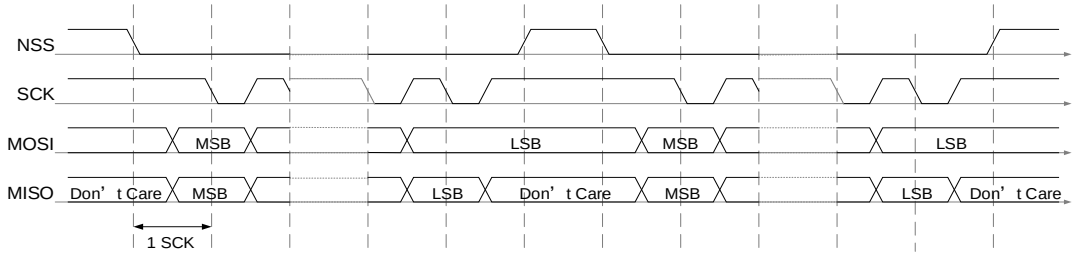
配置SPI_CTL1寄存器中的NSSP位使能该功能，为了确保使用该功能实现，需满足以下几个条件：配置设备为主机模式，使用普通SPI协议的数据帧格式，同时在第一个时钟跳变沿采样数据。

总之：MSTMOD = 1，NSSP = 1，CKPH = 0。

当使用NSS脉冲模式时，根据内部数据发送缓冲区的状态，NSS脉冲会在两个连续的数据帧之间产生，且持续时间至少为1个SCK时钟周期。如果数据发送缓冲区保持为空，可能会持续多个SCK时钟周期。NSS脉冲功能专为单一的主从应用设计，支持从机锁存数据。

下图描述了NSS脉冲模式在主机连续发送时的时序图。

图 30-13. NSS 脉冲模式时序图（主机连续发送）



SPI 四线模式操作流程

SPI四线模式用于控制四线SPI Flash外设。

要配置成SPI四线模式，首先要确认TBE位置1，且TRANS位清零，然后将SPI_QCTL寄存器中的QMOD位置1。在SPI四线模式，SPI_CTL0寄存器中BDEN位、BDOEN位、CRCEN位、CRCNT位、CRCNT位、RO位和LF位保持清零，DZ[3:0]位域配置数据长度为8位，且MSTMOD位置1，以保证SPI工作于主机模式。SPIEN位、PSC位、CKPL位和CKPH位根据需要进行配置。

SPI四线模式有两种运行模式：四线写模式和四线读模式，通过SPI_QCTL寄存器中的QRD位进行配置。

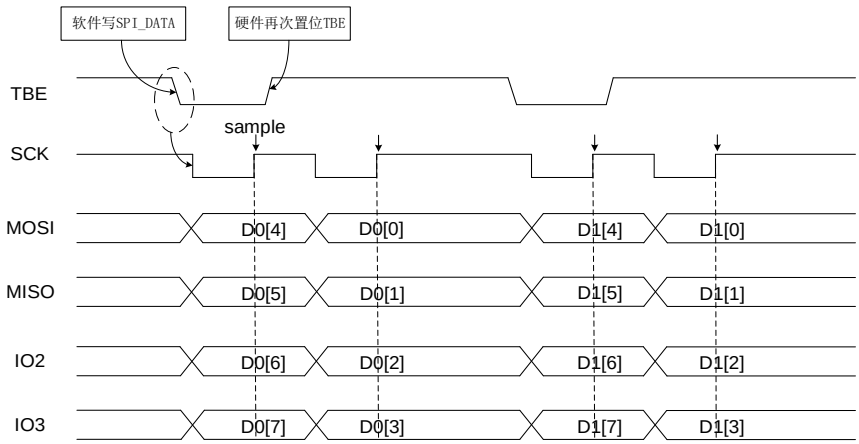
四线写模式

当SPI_QCTL寄存器中的QMOD位置1且QRD位清零时，SPI工作在四线写模式。在四线写模式中，MOSI、MISO、IO2和IO3都用作输出引脚，在SCK产生时钟信号后，一旦数据写入SPI_DATA寄存器（TBE位清零）且SPIEN位置1时，将会通过这四个引脚发送写入的数据。SPI开始数据传输之后，每发送一个数据帧都要检测TBE标志位，若不能满足条件则停止传输。

四线模式下发送操作流程：

1. 根据应用需求，配置SPI_CTL0和SPI_CTL1中的时钟预分频、时钟极性、相位等参数；
2. 将SPI_QCTL中的QMOD位置1，然后将SPI_CTL0中的SPIEN位置1来使能SPI功能；
3. 向SPI_DATA寄存器中写入一个字节的数据，TBE标志位将会清零；
4. 等待硬件将TBE位重新置位，然后写入下一个字节数据。

图 30-14. SPI 四线模式写操作时序图



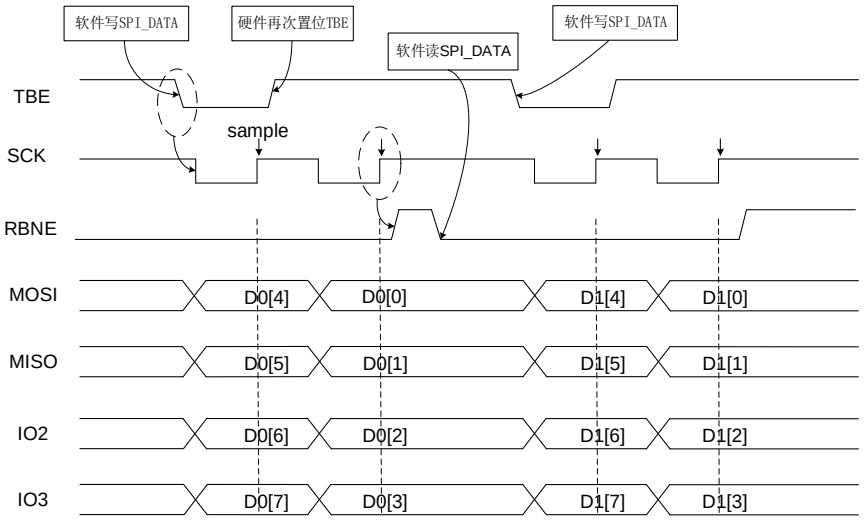
四线读模式

当SPI_QCTL寄存器中的QMOD位和QRD位都置1时，SPI工作在四线读模式。在四线读模式中，MOSI、MISO、IO2和IO3都用作输入引脚，一旦数据写入SPI_DATA寄存器（TBE位清零）且SPIEN位置1时，在SCK信号线产生时钟信号。写数据到SPI_DATA寄存器只是为了产生SCK时钟信号，所以可以写入任何数据。SPI开始数据传输之后，每发送一个数据帧都要检测SPIEN位和TBE位，若条件不满足则停止传输。所以软件需要一直向SPI_DATA写空闲数据，以产生SCK时钟信号。

四线模式下接收操作流程：

1. 根据应用需求，配置SPI_CTL0和SPI_CTL1中时钟预分频、时钟极性、相位等参数；
2. 将SPI_QCTL中的QMOD位和QRD位置1，然后将SPI_CTL0中的SPIEN位置1来使能SPI功能；
3. 写任意数据（例如0xFF）到SPI_DATA寄存器；
4. 等待RBNE位置1，然后读SPI_DATA寄存器来获取接收的数据；
5. 写任意数据（例如0xFF）到SPI_DATA寄存器，以接收下一个字节数据。

图 30-15. SPI 四线模式读操作时序图



SPI 停止流程

不同运行模式下采用不同的流程来停止SPI功能。

MFD SFD

等待TXLVL[1:0]=00和TRANS=0，接着通过清零SPIEN位关闭SPI。最后，读取数据直到RXLVL[1:0]=00。

MTU MTB STU STB

等待TXLVL[1:0]=00和TRANS=0，接着通过清零SPIEN位关闭SPI。

MRU MRB

应用程序可以在任何时候关闭SPI功能，然后等待TRANS=0，读取数据直到RXLVL[1:0]=00。

SRU SRB

应用程序可以在任何时候关闭SPI功能，然后等待TRANS=0，读取数据直到RXLVL[1:0]=00。

TI模式

TI模式的停止流程与上面描述过程相同。

NSS脉冲模式

NSS脉冲模式的停止流程与上面描述过程相同。

SPI四线模式

在禁用SPI四线模式和关闭SPI功能之前，软件应该先检查：TBE位置1，TRANS位清零，SPI_QCTL中的QMOD位和SPI_CTL0中的SPIEN位清零。

30.3.7. DMA 功能

DMA功能在传输过程中将应用程序从数据读写过程中释放出来，从而提高了系统效率。

通过置位SPI_CTL1寄存器中的DMATEN位和DMAREN位，使能SPI模式的DMA功能。为了使用DMA功能，软件首先应当正确配置DMA模块，然后通过初始化流程配置SPI模块，最后使能SPI。

SPI使能后，如果DMATEN位置1，每当TBE=1时，SPI将会发出一个DMA请求，然后DMA应答该请求，并自动写数据到SPI_DATA寄存器。如果DMAREN位置1，每当RBNE=1时，发出一个DMA请求，然后DMA应答该请求，并自动从SPI_DATA寄存器读取数据。

DMA 数据合并传输

采用DMA进行数据传输，当BYTEN设置为0且DZ[3:0]配置的数据长度小于或等于8位且数据合并模式使能时，DMA将会以16位方式访问SPI_DATA寄存器，自动完成数据的发送。

在数据合并模式使能且传输数据帧的帧数不是偶数倍的情况下，为了避免最后一次DMA传输多一帧数据的问题，需要将SPI_CTL1寄存器中TXDMA_ODD/RXDMA_ODD位为设置为1。

30.3.8. 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。

对于SPI，只有数据长度为8位或者16位时，SPI提供CRC计算，且独立于数据长度，可以固定设置为8位或16位CRC计算。对于其他所有的数据长度，CRC无效。CRC数据交换，通常需要在数据序列结束后，再占用一个或多个数据通信的时间。例如，当设置为8位的数据长度并做16位CRC检查时，发送完整的CRC数据就要两帧。如果使能了DMA功能，硬件将会自动处理CRC传输和校验，但SPI需设置DMA发送通道和接收通道的计数器值。发送DMA计数器值为不包括CRC帧的数据帧的数量。接收DMA计数器值的配置如下：

1.全双工模式：假设SPI接收的数据量为L，当CRCL = 0且DZ = 8时，，则DMA接收通道的计数值等于L + 1，否则DMA接收通道的计数值等于L + 2。

2.只接收模式：DMA接收通道计数值只等于接收的数据量。接收数据完成后，通过软件读取SPI_RCRC寄存器的方式获取CRC值。

注意：当SPI处于从机模式且CRC功能使能时，无论SPI是否使能，CRC计算器都对输入SCK时钟敏感。只有当时钟稳定时，软件才能启用CRC，以避免错误的CRC计算。当SPI作为从机工作时，在数据阶段和CRC阶段之间，内部NSS信号需要保持低电平。

当配置SPI为从模式并且使用CRC功能时，即使NSS引脚为高时仍然会执行CRC的计算(当NSS信号为高时，只要SCK引脚上有时钟脉冲，则CRC计算会继续执行)。当主设备交替地与多个从设备进行通信时，将会出现这种情况,此时建议在NSS信号为低时重启CRC功能)。当从设备未选中(NSS信号为高)转换到被选中为一个新的从设备(NSS信号为低)的时候，为了保持主从设备端下次CRC计算结果的同步，应该清除主从两端的CRC数值。建议按照下述步骤清除CRC数值：

1. 关闭SPI模块（SPIEN=0）；
2. 清除CRCEN位（CRCEN=0）；
3. 设置CRCEN位（CRCEN=1）；
4. 使能SPI模块（SPIEN=1）。

30.3.9. SPI 中断

状态标志位

- **发送缓冲区空标志位（TBE）**

当发送缓冲区为空或当前发送FIFO的存储量小于或等于总存储量的一半时，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/发送FIFO。

- **接收缓冲区非空标志位（RBNE）**

对于SPI，该位根据SPI_CTL1中的BYTEN位设置：如果BYTEN=0，则当前接收FIFO的存储量大于或等于总存储量的1/2时，RBNE置位。如果BYTEN=1，则当前接收FIFO的存储量大于或等于总存储量的1/4时，RBNE置位。表示此时接收到数据，并已存入接收FIFO中，软件可以通过读SPI_DATA寄存器来读取此数据。

- **SPI通信进行中标志位（TRANS）**

TRANS位是用来指示当前传输是否正在进行或结束的状态标志位，它由内部硬件置位和清除，无法通过软件控制。该标志位不会产生任何中断。

错误标志

- **配置错误标志（CONFERR）**

在主机模式中，CONFERR位是一个错误标志位。在硬件NSS模式中，如果NSSDRV没有使能，当NSS被拉低时，CONFERR位被置1。在软件NSS模式中，当SWNSS位为0时，CONFERR位置1。当CONFERR位置1时，SPIEN位和MSTMOD位由硬件清除，SPI关闭，设备强制进入从机模式。

在CONFERR位清零之前，SPIEN位和MSTMOD位保持写保护，从机的CONFERR位不能置1。在多主机配置中，设备可以在CONFERR位置1时进入从机模式，这意味着发生了系统控制的多主冲突。

- **接收过载错误（RXORERR）**

在RBNE位为1时，如果再有数据被接收，RXORERR位将会置1。这说明，上一帧数据还未被读出而新的数据已经接收了。接收缓冲区的内容不会被新接收的数据覆盖，所以新接收的数据丢失。

- **帧错误（FERR）**

在TI从机模式下，从机也要监视NSS信号，如果检测到错误的NSS信号，将会置位FERR标志位。例如，NSS信号在一个字节的中间位发生翻转。

- **CRC错误（CRCERR）**

当CRCEN位置1时，SPI_RCRC寄存器中接收到的数据的CRC计算值将会和紧随着最后一帧数据后接收到的CRC值进行比较，当两者不同时，CRCERR位将会置1。

表 30-6. 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	

30.4. SPI 寄存器

SPI0 基地址：0x4001 3000
SPI1 基地址：0x4000 3800
SPI2 基地址：0x4000 3C00

30.4.1. 控制寄存器 0（SPI_CTL0）

地址偏移：0x00
复位值：0x0000 0000

该寄存器可以按字节（8位）或半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
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
BDEN	BDOEN	CRCEN	CRCNT	CRCL	RO	SWNSSEN	SWNSS	LF	SPIEN	PSC[2:0]			MSTMOD	CKPL	CKPH
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	BDEN	双向数据模式使能 0：2线单向传输模式。 1：1线双向传输模式。数据在主机的MOSI引脚和从机的MISO引脚之间传输。
14	BDOEN	双向传输输出使能 当BDEN置位时，该位决定了数据的传输方向。 0：工作在只接收模式。 1：工作在只发送模式。
13	CRCEN	CRC计算使能 0：禁能CRC计算。 1：使能CRC计算。
12	CRCNT	下一次传输CRC 0：下一次传输值为数据。 1：下一次传输值为CRC值。 当数据传输由DMA管理时，CRC值由硬件传输，该位应该被清零。 在全双工和只发送模式下，当最后一个数据写入SPI_DATA寄存器后应将该位置1。

		在只接收模式下，在接收完倒数第二个数据后应将该位置1。
11	CRCL	CRC长度 0：8位CRC数据长度 1：16位CRC数据长度
10	RO	只接收模式 当BDEN清零时，该位决定了数据的传输方向。 0：全双工模式 1：只接收模式
9	SWNSSEN	NSS软件模式使能 0：NSS硬件模式，NSS电平取决于NSS引脚。 1：NSS软件模式，NSS电平取决于SWNSS位。 该位在SPI TI模式下没有意义。
8	SWNSS	NSS软件模式下NSS引脚选择 0：NSS引脚拉低。 1：NSS引脚拉高。 只有在SWNSSEN置位时，该位有效。 该位在SPI TI模式下没有意义。
7	LF	最低有效位先发模式 0：先发送最高有效位。 1：先发送最低有效位。 该位在SPI TI模式下没有意义。
6	SPIEN	SPI使能 0：禁能SP 1：使能SPI
5:3	PSC[2:0]	主时钟预分频选择 000：PCLK/2 001：PCLK/4 010：PCLK/8 011：PCLK/16 100：PCLK/32 101：PCLK/64 110：PCLK/128 111：PCLK/256 当使用SPI0时，PCLK=PCLK2。当使用SPI1或者SPI2时，PCLK=PCLK1。
2	MSTMOD	主从模式使能 0：从机模式

1：主机模式

1	CKPL	时钟极性选择 0：SPI为空闲状态时，CLK引脚拉低。 1：SPI为空闲状态时，CLK引脚拉高。
0	CKPH	时钟相位选择 0：在第一个时钟跳变沿采集第一个数据。 1：在第二个时钟跳变沿采集第一个数据。

30.4.2. 控制寄存器 1（SPI_CTL1）

地址偏移：0x04
复位值：0x0000 0700

该寄存器可以按字节（8位）或半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TXDMA_ODD	RXDMA_ODD	BYTEN	DZ[3:0]				TBEIE	RBNEIE	ERRIE	TMOD	NSSP	NSSDRV	DMATEN	DMAREN
	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	TXDMA_ODD	DMA发送通道奇数字节 在数据合并传输模式中，当通过DMA发送的数据总数为奇数时置位。仅在DMA功能开启且合并模式开启时（数据长度小于等于8位且对SPI_DATA写入访问是16位宽）有效。 必须在SPI禁止时写入。 0：通过DMA发送的数据总量为偶数个。 1：通过DMA发送的数据总量为奇数个。
13	RXDMA_ODD	DMA接收通道奇数字节 在数据合并传输模式中，当通过DMA接收的数据总数为奇数时置位。仅在DMA功能开启且合并模式开启时（数据长度小于等于8位且对SPI_DATA写入访问是16位宽）有效。 必须在SPI禁止时写入。 0：通过DMA接收的数据总量为偶数个。 1：通过DMA接收的数据总量为奇数个。

12	BYTEN	字节访问使能 该位用于指示对FIFO的访问宽度，并设置产生RBNE的RXFIFO的阈值。 0：半字访问，且当RXLVL>=2时，RBNE置位。 1：字节访问，且当RXLVL>=1时，RBNE置位。
11:8	DZ[3:0]	数据位宽 这些位配置SPI传输数据的位宽： 0000：强制为“0111” 0001：强制为“0111” 0010：强制为“0111” 0011：4位 0100：5位 …… 1111：16位
7	TBEIE	发送缓冲区空中断使能 0：禁能TBE中断 1：使能TBE中断。当TBE置位时，产生中断。
6	RBNEIE	接收缓冲区非空中断使能 0：禁能RBNE中断 1：使能RBNE中断。当RBNE置位时，产生中断。
5	ERRIE	错误中断使能 0：禁能错误中断 1：使能错误中断。当CRCERR位，CONFERR位，FERR位，RXORERR位或者TXURERR位置1时，产生中断。
4	TMOD	SPI TI模式使能 0：禁能SPI TI模式 1：使能SPI TI模式
3	NSSP	SPI NSS脉冲模式使能 0：禁能SPI NSS脉冲模式 1：使能SPI NSS脉冲模式
2	NSSDRV	NSS输出使能 0：禁能主机NSS输出 1：使能主机NSS输出 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：禁能发送缓冲区DMA

		1：使能发送缓冲区DMA。当SPI_STAT中的TBE置位时，将会在相应的DMA通道上产生一个DMA请求。
0	DMAREN	接收缓冲区DMA使能 0：禁能接收缓冲区DMA 1：使能接收缓冲区DMA。当SPI_STAT中的RBNE置位时，将会在相应的DMA通道上产生一个DMA请求。

30.4.3. 状态寄存器（SPI_STAT）

地址偏移：0x08

复位值：0x0000 0002

该寄存器可以按字节（8位）或半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			TXLVL[1:0]		RXLVL[1:0]		FERR	TRANS	RXORERR	CONFERR	CRCERR	保留		TBE	RBNE
			r		r		rc_w0	r	r	r	rc_w0			r	r

位/位域	名称	描述
31:13	保留	必须保持复位值。
12:11	TXLVL[1:0]	发送FIFO状态 00：空 01：1/4满 10：1/2满 11：满 **注意：**这里的FIFO状态是指FIFO当前实际的存储量。在这里，当FIFO存储量大于总存储量的1/2时认为FIFO已满。
10:9	RXLVL[1:0]	接收FIFO状态 00：空 01：1/4满 10：1/2满 11：满 这些位在打开了CRC计算功能时的SPI只接收模式下，不使用。 **注意：**这里的FIFO状态是指FIFO当前实际的存储量。在这里，当FIFO存储量大于总存储量的1/2时认为FIFO已满。

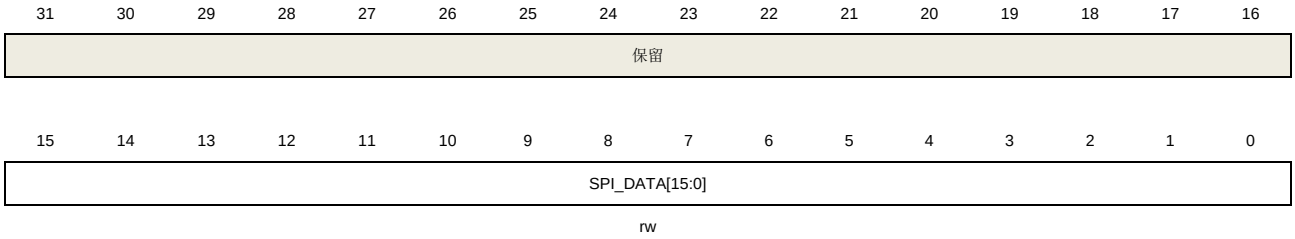
8	FERR	帧错误 SPI TI 模式： 0：没有 TI 模式帧错误发生 1：TI 模式帧错误发生 该位由硬件置位，可以通过写0清除。
7	TRANS	通信进行中标志 0：SPI空闲 1：SPI当前正在发送且/或接收数据。 该位由硬件置位和清除。
6	RXORERR	接收过载错误标志 0：没有接收过载错误发生 1：接收过载错误发生 该位由硬件置位，软件序列清零。软件序列为：先读SPI_DATA寄存器，然后读SPI_STAT寄存器。
5	CONFERR	SPI配置错误 0：无配置错误发生 1：配置错误发生（主机模式下，在硬件NSS模式时NSS引脚被拉低，或者软件NSS模式时SWNSS位为0，都会产生CONFERR错误。） 该位由硬件置位，软件序列清零。软件序列为：读或写SPI_STAT寄存器，然后写SPI_CTL0寄存器。 I2S模式下不使用该位。
4	CRCERR	SPI CRC错误标志 0：SPI_RCRC值等于最后接收到的CRC值。 1：SPI_RCRC值不等于最后接收到的CRC值该位由硬件置位，可以通过写0清除。
3:2	保留	必须保持复位值。
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

30.4.4. 数据寄存器（SPI_DATA）

地址偏移：0x0C

复位值：0x0000 0000

该寄存器可以按字节（8位）或半字（16位）或字（32位）访问。

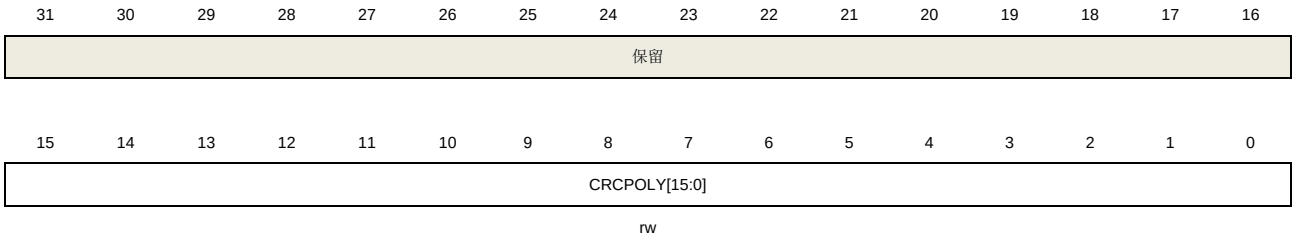


位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	SPI_DATA[15:0]	数据传输寄存器值 对于SPI，硬件有两个FIFO：TXFIFO和RXFIFO。向SPI_DATA写数据将会把数据存入发送FIFO，从SPI_DATA读数据，将从接收FIFO获得数据。 **注意：** 对于SPI，实际上硬件只根据配置好的BYTEN这一位来判断每一次访问SPI_DATA的位宽，与软件当前操作所使用的位宽无关。

30.4.5. CRC 多项式寄存器（SPI_CRCPOLY）

地址偏移：0x10
复位值：0x0000 0007

该寄存器可以按字节（8位）或半字（16位）或字（32位）访问。



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。

30.4.6. 接收 CRC 寄存器（SPI_RCRC）

地址偏移：0x14
复位值：0x0000 0000

该寄存器可以按字节（8位）或半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	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。对于SPI，只有当数据长度为8位或16位时，CRC有效。当CRC长度设置为8位并且数据长度等于8位时，CRC计算基于CRC8标准进行，并将值保存在RCRC[7：0]中，否则CRC计算基于CRC16标准进行，并将值保存在RCRC[15：0]中。 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。

30.4.7. 发送 CRC 寄存器（SPI_TCRC）

地址偏移：0x18
复位值：0x0000 0000

该寄存器可以按字节（8位）或半字（16位）或字（32位）访问。

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

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

r

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	TCRC[15:0]	发送CRC寄存器值 当SPI_CTL0中的CRCEN置位时，硬件计算发送数据的CRC值，并保存到TCRC寄存器中。对于SPI，只有当数据长度为8位或16位时，CRC有效。当CRC长度设置为8位并且数据长度等于8位时，CRC计算基于CRC8标准进行，并将值保存在TCRC[7：0]中，否则CRC计算基于CRC16标准进行，并将值保存在TCRC[15：0]中。

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。

30.4.8. SPI0 四线 SPI 控制寄存器（SPI_QCTL）

地址偏移：0x80

复位值：0x0000 0000

该寄存器可以按字节（8位）或半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	QRD	四线SPI模式读选择 0：SPI四线模式写操作 1：SPI四线模式读操作 该位仅能在SPI未通信时配置（TRANS位清零）。 该位仅适用于SPI0。
0	QMOD	四线SPI模式使能 0：SPI工作在单路模式 1：SPI工作在四线模式 该位仅能在SPI未通信时配置（TRANS位清零）。 该位仅适用于SPI0。

31. 四线 SPI 接口（QSPI）

31.1. 简介

QSPI是一种专用于和flash存储器通信的接口，可以支持单线，双线，四线SPI Flash。可以工作在普通模式、读轮询模式和内存映射模式。

31.2. 主要特征

- 三种模式：普通模式（外部地址），读轮询模式和内存映射模式；
- 可用于普通模式和内存映射模式的完全可编程的命令格式；
- 集成用于接收和发送的FIFO；
- 支持SDR和DDR模式；
- 支持DQS信号；
- 允许8位、16位或32位数据访问；
- 普通模式支持DMA操作。

31.3. 功能描述

31.3.1. QSPI 结构框图

QSPI使用7根信号线和外部flash存储器连接，引脚在*表31-1. QSPI信号线描述*中描述。

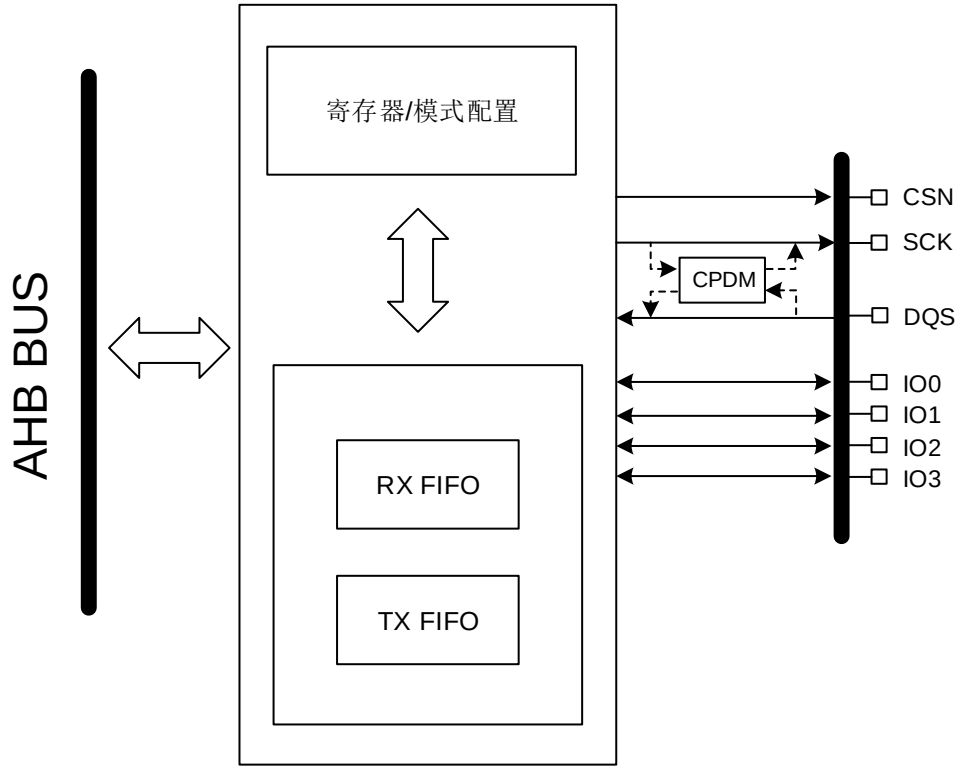
表 31-1. QSPI 信号线描述

引脚	方向	描述
CSN	O	片选输出（低电平有效）
SCK	O	时钟输出
IO0/SO	I/O	单线模式：数据输出 双线模式：数据输入或输出 四线模式：数据输入或输出
IO1/SI	I/O	单线模式：数据输入 双线模式：数据输入或输出 四线模式：数据输入或输出
IO2	I/O	单线模式：连接 flash 的 WP 引脚，控制“写保护”功能 双线模式：连接 flash 的 WP 引脚，控制“写保护”功能 四线模式：数据输入或输出
IO3	I/O	单线模式：连接 flash 的 HOLD 引脚，控制“保持”功能

引脚	方向	描述
		双线模式：连接 flash 的 HOLD 引脚，控制“保持”功能 四线模式：数据输入或输出
DQS	I	数据选通信号

QSPI 结构图框图如*图 31-1. QSPI 结构框图*。

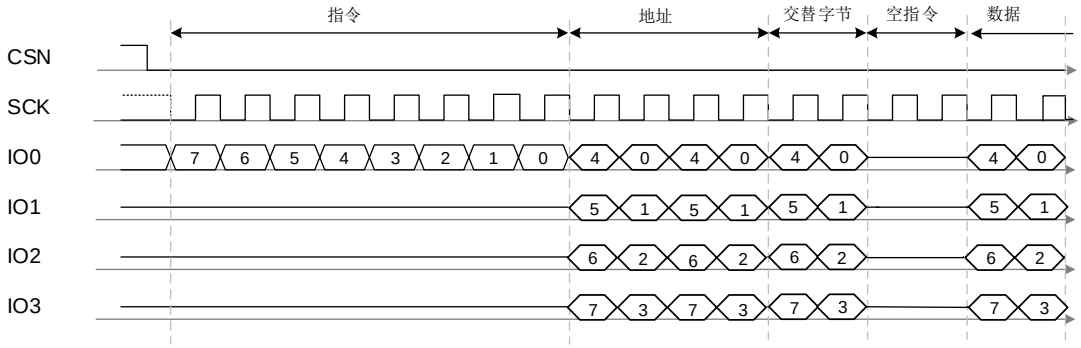
图 31-1. QSPI 结构框图



31.3.2. QSPI 命令格式

QSPI 使用不同格式的命令与 flash 存储器通信。一共最多有五个阶段：指令阶段、地址阶段、交替字节阶段、空指令阶段、数据阶段。任一阶段都可以跳过，但是至少需要包含指令阶段、地址阶段、交替字节、数据阶段的其中一个阶段，这是由软件保证，硬件设计没有任何保护方法。另外，命令的高位始终占用高位信号线。

图 31-2. QSPI 命令格式



命令和相应的配置如*表 31-2. QSPI 命令描述*所示。

表 31-2. QSPI 命令描述

命令	发送信息	配置	注意
指令	8位指令	QSPI_TCFG寄存器定义指令和信号线模式	-
地址	1-4字节地址	QSPI_ADDR寄存器定义地址信息。QSPI_TCFG寄存器定义发送地址的字节数和信号线模式	-
交替字节	1-4交替字节	QSPI_ALTE 寄存器定义交替字节信息，QSPI_TCFG 寄存器定义交替字节的个数和信号线模式	-
空闲	0-31周期	QSPI_TCFG 寄存器定义周期，DATAMOD 位域（QSPI_TCFG 寄存器）定义空闲信号线模式	这期间与外部存储器没有数据交互，等待外部存储器，准备数据。
数据	任意数量的字节	在普通模式下，QSPI_DTLEN 寄存器定义字节数。DATAMOD 位域（QSPI_TCFG 寄存器）定义数据信号线模式，DATAMOD = 00 的配置只能在普通写模式下使用。	在内存映射模式下，传输的字节个数确定 AHB 总线的访问操作，可以 8 位，16 位或者 32 位读写访问，相应传输 1 个，2 个，4 个字节。

注意：信号线模式可以为无指令，单线，双线或者四线。

31.3.3. QSPI 信号线的模式

对于 QSPI 的信号线模式，指令阶段、地址阶段、字节交替阶段、数据阶段都可以通过设置 IMOD / ADDRMOD / ALTEMOD / DATAMOD 位域进行独立配置。

表 31-3. QSPI 信号线模式

信号线模式		单线模式	双线模式	四线模式
配置位域	IMOD	01或者00	10或者00	11或者00
	ADDRMOD			
	ALTEMOD			
	DATAMOD			
引脚	IO0（SO）	输出	输入：数据阶段读操作（高阻状态） 输出：所有其他阶段	输入：数据阶段读操作（高阻状态） 输出：所有其他阶段
	IO1（SI）	输入（高阻状态）		
	IO2	输出0（禁止“写保护”功能）		
	IO3	输出1（禁止“保持”功能）		
描述		DATAMOD = 01时，空闲阶段IO0输出，IO1输入（高阻状态）。	DATAMOD = 10 时，空闲阶段 IO0 / IO1 一直高阻状态。	DATAMOD = 11时，空闲阶段IO0 / IO1 / IO2 / IO3 一直高阻状态。

IO2 / IO3 仅用于四线模式，如果五个阶段都没有配置为四线模式，IO2 / IO3 引脚被释放，即使 QSPI 被使能也可以用于其他功能。

QSPI 引脚

QSPI 最大支持 200M 时钟，但使用不同组的引脚支持的最高通信时钟频率不同，请参考*表 31-4. QSPI 使用引脚与支持的最高通信时钟对应关系*。

表 31-4. QSPI 使用引脚与支持的最高通信时钟对应关系

引脚功能	最高时钟 120M	最高时钟 120M	最高时钟 200M	最高时钟 120M
QSPI_CSN	PA2	PB11\PE11	PD3	PA2
QSPI_SCK	PA3	PB10\PE10	PD2	PF10
QSPI_IO0	PB1	PE12	PD4	PC1\PF8
QSPI_IO1	PB0\PB2	PE13	PD5	PC2\PF9
QSPI_IO2	PA7	PE14	PD6	PC3\PF7
QSPI_IO3	PA6\PC4	PE15	PD7	PF6
QSPI_DQS	PB7	PB7	PD1	PB7

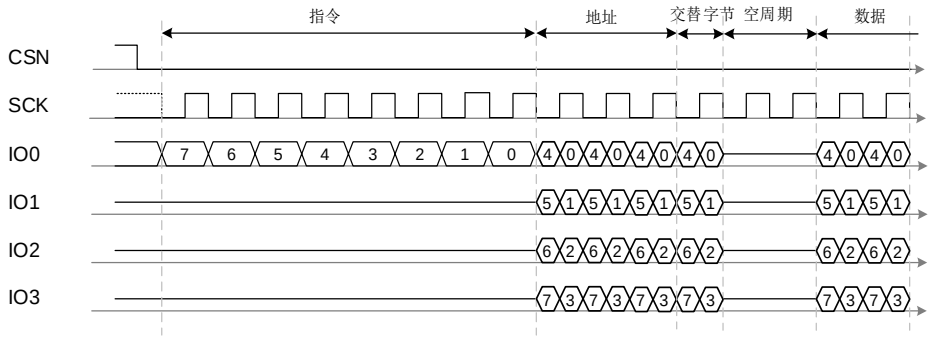
注意：强烈推荐按组使用引脚，否则无法保证 QSPI 支持的最大通信时钟频率。

31.3.4. QSPI DDR 模式

默认情况下，QSPI 工作在 SDR（单倍数据速率）模式，当 QSPI_TCFG 寄存器 DDREN 位置 1

时，QSPI 工作在 DDR 模式（双倍数据速率）下。DDR 模式下，地址 / 字节交替 / 数据阶段，IO0、IO1、IO2、IO3 信号均在 SCK 信号的两个时钟沿（上升沿和下降沿）传输数据，指令阶段仍采用 SDR 模式传输，IO0、IO1、IO2、IO3 信号在 SCK 的下降沿采样。

图 31-3. QSPI DDR 模式

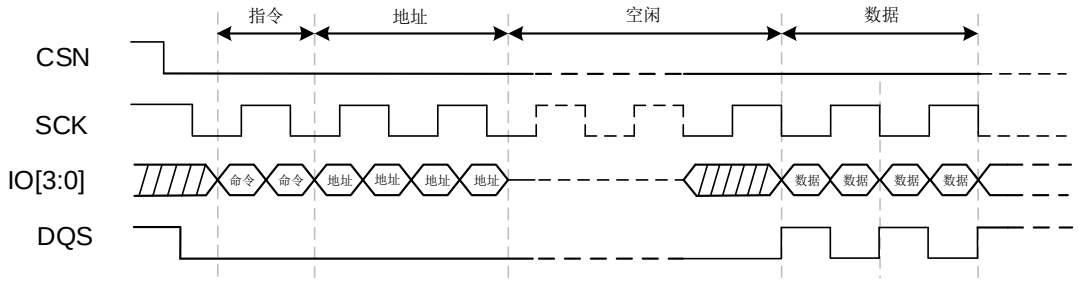


在 DDR 模式时，根据实际使用需求，可能需要通过设置 QSPI_DCFG 寄存器中 DLYSCEN 位为 1 使用 CPDM 微调 QSPI 接收时钟相位。请参考*时钟相位延迟模块（CPDM）*。

31.3.5. DQS 信号

DQS信号通常用于高速应用程序，以指示何时存储器的输出数据可被MCU读取。开启DQS功能时，频率与SCK频率相同。对于SDR读取操作，数据只能锁存在DQS信号的上升沿上。对于DDR读操作，数据应该锁存在DQS信号的上升沿和下降沿。

图 31-4. 使用 DQS 信号 DDR 模式读



31.3.6. CSN 和 SCK 的行为

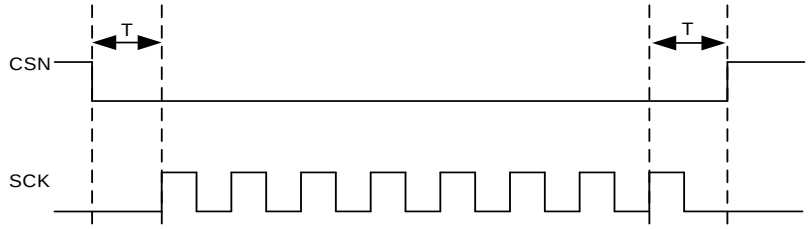
CSN 默认为高电平，它在命令开始时拉低，结束时拉高。

SCK 是从内部 sck 信号输出的一个门信号，内部 sck 信号是一直存在的。

为了适应一些高速设备，QSPI 支持通过配置 QSPI_DCFG 寄存器中 CSNCKM 位选择 CSN 在第一个 SCK 有效上升沿的之前一个或者两个 SCK 时钟周期拉低及在最后一个 SCK 有效上升沿之后

一个或者两个 SCK 时钟周期拉高。

图 31-5. CSN 和 SCK 的行为（CSNCKM = 0）



当 FIFO 在写操作时为空或者读操作时为满，SCK 会停止并且保持低电平直到 FIFO 可以再次工作。在这时，如果 CSN 为高电平，SCK 会在 CSN 上升沿之后的半个 SCK 时钟周期拉高电平。

31.4. 操作模式

QSPI 可以工作在普通模式、读轮询模式和内存映射模式。在普通模式下，所有的操作都依赖于 QSPI 寄存器。在读轮询模式下，定时读取并检查外部闪存存储器中的状态寄存器值。在内存映射模式下，外部闪存被映射到微控制器地址空间（范围从 0x9000 0000 到 0x9FFF FFFF），并作为内部存储器访问。

31.4.1. 普通模式

普通模式写操作是通过将 QSPI_TCFG 寄存器中的 FMOD[1:0]为“00”来选择的。待传输的数据写入 QSPI_DATA。通过将 QSPI_TCFG 寄存器中的 FMOD[1:0]设置为“01”，选择普通模式读操作，接收的数据从 QSPI_DATA 读取。

QSPI_DTLEN 寄存器中的 DTLEN[31:0]定义了待传输的字节数。如果 DTLEN 是 0xFFFF FFFF，则认为传输的字节数是未定义的，在传输字节数达到 QSPI_DCFG 寄存器中 FMSZ[4:0]规定的存储器大小边界时停止传输。如果 DTLEN 为 0xFFFF FFFF 且 FMSZ[4:0]被配置为 0x1F 时，闪存存储器容量为 4GB，传输会一直持续到发生中止请求或 QSPI 被禁用。

当传输数据的字节数达到 DTLEN 寄存器中设定的值时，传输完成标志 TC 会被置 1。如果 DTLEN 为 0xFFFF FFFF，则发送 / 接收字节数等于 FMSZ[4:0]规定的外部存储器大小时，TC 会被置 1。如果 TCIE 和 TC 都被置 1，则会产生中断，通过将 QSPI_STATC 寄存器的 TCC 位置 1 可以清除 TC 位。

初始化一个命令序列

命令序列在根据通信需求配置好最后信息之后立即开始。

当没有地址并且没有数据时，在访问 QSPI_TCFG 寄存器之后立即开始命令序列。

当存在地址但没有数据时，在访问 QSPI_ADDR 寄存器之后立即开始命令序列。

当在普通模式写操作时需要地址并且有数据，在访问 QSPI_DATA 寄存器之后立即开始命令序列。

FIFO

16 字节的 FIFO 用于传输数据。在普通模式写操作时，AHB 写访问方式与 FIFO 增加的字节数的关系如*表 31-5. AHB 写访问方式与 FIFO 增加的字节数的关系*所示。

表 31-5. AHB 写访问方式与 FIFO 增加的字节数的关系

AHB 写访问方式	FIFO 增加的字节数
32 位	4 字节
16 位	2 字节
8 位	1 字节

注意：当 AHB 写访问模式为 8 位或 16 位时，QSPI_DATA 寄存器中最低有效字节是有效的。

FIFO 阈值由 QSPI_CTL 寄存器中的 FTL[3:0]定义，在普通模式读操作时，当 FIFO 中的字节数等于者超过定义的阈值时，QSPI_STAT 寄存器中的 FIFO 阈值标志 FT 将置 1。在数据阶段完成后如果 FIFO 不为空，FT 也会被置 1。在普通模式写操作时，当 FIFO 的空字节数超过阈值时，FT 会被置 1。

如果 FTIE 和 FT 都被置 1，将产生中断。如果 QSPI DMA 使能，DMA 请求由 FT 产生，直到标志清除。

在普通模式读操作时，当 FIFO 变为满时，QSPI 暂时停止 SCK 以避免溢出。读序列不能恢复直到 FIFO 中有大于等于 4 个字节剩余空间。

31.4.2. 读轮询模式

通过将 FMOD[1:0]配置为“10”可以来选择读轮询模式。在读轮询模式下，QSPI 周期性地启动一个读命令，其中最多包含 4 字节的数据。接收到的数据可以按位屏蔽，并与定义的数据内容进行比较，如果匹配发生，且 RPMFIE 位置 1，将生成一个中断。

读轮询访问序列的启动与普通模式读操作相同。在周期性间隔时 BUSY 位保持高电平。

轮询匹配模式位 RPMM 控制比较匹配模式，如果 RPMM = 0，与模式被选择。在该模式下，只在所有的未屏蔽位都有匹配时，状态匹配标志 RPMF 将置 1。如果 RPMM = 1，或模式被选择。在该模式下，任何非屏蔽位只要有一位匹配，RPMF 将置 1。

在读轮询模式下，如果设置了 RPMS 位，当检测到匹配时，读轮询序列将停止，并在数据阶段结束时清除 BUSY 标志。否则，周期序列将继续，直到 ABORT 位置 1 或 QSPI 被关闭。

在读轮询模式下，FIFO 是避开的，读取的状态字节存放在 QSPI_DATA 中，存储的状态字节不会被 MASK 控制域影响。如果有数据阶段，QSPI_DATA 中内容在数据阶段开始时更新。

如果 FT 位在数据阶段结束时被置位，此时认为外部闪存状态字节已经被读取，读取 QSPI_DATA 清除 FT 位。

31.4.3. 内存映射模式

通过将 FMOD[1:0]配置为“11”，可以选择内存映射模式。在内存映射模式下，外部闪存被认为是内部存储器来访问，最大访问地址为 256MB，即使外部闪存大于 256MB。即使 FMSZ 定义的地址范围在 256MB 范围内，内存映射模式也不允许地址超过 FMSZ 定义的范围。否则，将生成一个错误。如果 AHB 主机是 CPU，会产生硬故障中断。如果 AHB 主机是 DMA，将产生一个传输错误，并且相应 DMA 通道会关闭。

在该模式下，支持字节、半字、字单次访问或突发访问。

内存映射模式支持顺序访问时的预取功能。QSPI 在访问数据之前会首先在下一个地址加载数据，如果下次访问确实是在下一个地址，那么访问速度会更快，因为数据已经被预取了。否则，将重新启动读取序列。在读取序列开始之前拉低 CSN。

当 FIFO 满时，SCK 停止输出，在此期间 CSN 保持低电平。如果 QSPI_CTL 寄存器中 TMOUTEN 位置 1，当低电平的持续时间达到 QSPI_TMOUT 寄存器中指定的 SCK 时钟周期数时，CSN 将被拉高。

在开始传输时，在 CSN 拉低之前 BUSY 位会变为高电平，在发生超时、中止或者 QSPI 被关闭后变为低电平。

31.5. QSPI 配置

31.5.1. Flash 配置

QSPI_DCFG 寄存器的配置可以用来指定外部闪存存储器的特性，从而使 QSPI 可以持续工作。

QSPI_DCFG 寄存器中 FMSZ[4:0]定义了外部存储器大小，FMSZ+1 是外部存储器的地址位数。在普通模式下，flash 容量最大可达 4GB。

CSHC[2:0]定义了片选高电平时间，它规定了在两个命令序列之间保持高电平最少的 SCK 周期数。

31.5.2. IP 配置

QSPI_CTL 寄存器中的配置可以用来指定 QSPI IP 的特征。

QSPI_CTL 寄存器中 PSC[7:0]定义了时钟分频系数。

SSAMPLE 定义哪个 SCK 边沿用于采样数据。默认情况下，QSPI 在外部存储器驱动数据后的半个 SCK 周期采样数据。然而，因为外部信号的延迟，需要推迟采样数据。采样边沿可以使用 SSAMPLE 移位半个 SCK 周期。

通过在 QSPI_STAT 寄存器中设置 DMAEN 位来启用 DMA 请求。在 QSPI_CTL 寄存器中设置 FTL[3:0]位来配置 FIFO 阈值等级。

31.6. 只发送一次指令

将 QSPI_TCFG 寄存器中 SIOO 位置 1，可以使能只发送一次指令模式，该功能对所有模式有效。如果 SIOO 置 1，在访问 QSPI_TCFG 后该指令只发送一次，后续命令序列会跳过指令阶段，直到 QSPI_TCFG 再次被访问。

31.7. 错误和中断

当*表 31-6. TERR 和 AHB 错误条件*中一个条件发生时，会产生 TERR 和 AHB 错误。

表 31-6. TERR 和 AHB 错误条件

错误名称	条件
TERR	1. 在普通模式或者读轮询模式下，超出FMSZ规定的地址范围，在QSPI_ADDR寄存器中编写了一个错误的地址。 2. 在普通模式下，地址（ADDR）加数据长度（DTLEN）大于外部内存。
AHB错误	1. 在内存映射模式下，AHB主机执行超出范围访问，或QSPI被关闭。 2. AHB主机正在访问内存映射空间，但内存映射模式未使能。

QSPI 中断事件和标志如*表 31-7. QSPI 中断事件*所示。

表 31-7. QSPI 中断事件

中断事件	事件标志	中断使能位	清除方式
FIFO阈值中断	FT	FTIE	硬件清除
传输完成中断	TC	TCIE	QSPI_STATC寄存器中TCC位置1
传输错误中断	TERR	TERRIE	QSPI_STATC寄存器中TERRC位置1
超时中断	TMOUT	TMOUTIE	QSPI_STATC寄存器中TMOUTC位置1
状态匹配中断	RPMF	RPMFIE	QSPI_STATC寄存器中RPMFC位置1

31.8. QSPI 寄存器

QSPI 访问基地址：0xA000 1000

31.8.1. 控制寄存器（QSPI_CTL）

地址偏移：0x00
复位值：0x0000 0010

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PSC[7:0]								RPMM	RPMS	保留	TMOUTIE	RPMFIE	FTIE	TCIE	TERRIE
rw								rw	rw		rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCKDV[3:0]				FTL[3:0]				保留		OCKDEN	SSAMPLE	TMOUTEN	DMAEN	ABORT	QSPIEN
rw				rw						rw	rw	rw	rw	w1s	rw

位/位域	名称	描述
31:24	PSC[7:0]	该位域定义了从 AHB 时钟分频产生 QSPI 时钟的分频因子。SCK 和 AHB 之间的频率关系为： $f_{SCK}=f_{AHB}/(PSC+1)$. 00000000: $f_{SCK}=f_{AHB}$ 00000001: $f_{SCK}=f_{AHB}/2$ 00000010: $f_{SCK}=f_{AHB}/3$ ... 11111111: $f_{SCK}=f_{AHB}/256$ 对于奇数时钟分频因子，时钟的占空比没有50%，时钟信号保持低电平时间要比高电平时间少一个周期。 该位只能在BUSY = 0时才能修改。
23	RPMM	读轮询匹配模式 该位表明在读轮询时采用什么方式定义产生匹配 0：与模式，如果flash返回的字节所有非屏蔽位都和QSPI_STATMATCH寄存器相应位匹配，状态匹配标志RPMF被置位。 1：或模式，如果flash返回的字节任何一个非屏蔽位都和QSPI_STATMATCH寄存器相应位匹配，状态匹配标志RPMF被置位。 该位只能在BUSY = 0时修改。
22	RPMS	读轮询模式停止 该位表明在产生匹配后停止读轮询模式 0：在ABORT置位或者禁能QSPI模块时读轮询停止。

		1：在产生匹配后读轮询停止。 该位只能在BUSY = 0时修改。
21	保留	必须保持复位值
20	TMOUTIE	超时中断使能 0：中断禁能 1：中断使能
19	RPMFIE	读轮询模式匹配中断使能 0：中断禁能 1：中断使能
18	FTIE	FIFO阈值中断使能 0：中断禁能 1：中断使能
17	TCIE	传输完成中断使能 0：中断禁能 1：中断使能
16	TERRIE	传输错误中断使能 0：中断禁能 1：中断使能
15:12	OCKDV[3:0]	输出时钟延时值 该位域仅在OCKDEN使能时有效。 输出时钟延迟功能仅在QSPI时钟不分频时生效。
11:8	FTL[3:0]	FIFO阈值等级 该位在普通模式下使用，FIFO中的字节数会触发FIFO阈值标志被置位。 普通模式写操作时(FMOD = 00)： 0000：FT会被置位，如果有1个或者更多字节可以有效写入FIFO 0001：FT会被置位，如果有2个或者更多字节可以有效写入FIFO ... 1111：FT会被置位，如果有16个字节可以有效写入FIFO 普通模式读操作时 (FMOD = 01)： 0000：FT会被置位，如果有1个或者更多有效数据能从FIFO中读取 0001：FT会被置位，如果有2个或者更多有效数据能从FIFO中读取 ... 1111：FT会被置位，如果有16个有效数据能从FIFO中读取 如果DMAEN为1，在改变FTL之前，DMA控制器的相应通道必须禁能。
7:6	保留	必须保持复位值

5	OCKDEN	向flash写数据时输出SCK延时使能 0：SCK延时禁能 1：SCK延时使能
4	SSAMPLE	采样延时 默认情况下，QSPI在FLASH存储器驱动数据后二分之一个SCK时钟周期采样。考虑到外部信号的延迟，该位域可以配置为允许数据稍后对数据进行采样。 0：不延时 1：延时半个周期 该位只能在BUSY = 0时才能被修改。
3	TMOUTEN	超时计数器使能 在内存映射模式（FMOD=11）下，如果将该位置1，且在TMOUTCYC[15:0]定义的时间之后，没有访问外部flash，片选（CSN）会释放。 0：超时计数器失能，在内存映射模式下访问后片选（CSN）保持低电平 1：超时计数器使能，在内存映射模式下，如果flash在超过TMOUTCYC [15:0]个周期后没有访问外部flash，片选（CSN）会释放。 该位只能在BUSY = 0时修改。
2	DMAEN	DMA使能 普通模式下，可以使用DMA传输数据。当FT位置1时，DMA传输开始。 0：DMA禁能 1：DMA使能
1	ABORT	终止请求 该位停止当前命令，终止请求完成后会自动清除。 读轮询模式或者内存映射模式，如果将该位置1，RPMS位或者DMEN位被清除。 0：无终止请求 1：终止请求
0	QSPIEN	使能QSPI 0：QSPI禁能 1：QSPI使能

31.8.2. 设备配置寄存器（QSPI_DCFG）

地址偏移：0x04
复位值：0x001F 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	CSNCKM	保留									FMSZ[4:0]				
	rw										rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					CSHC[2:0]			保留		DLYSCEN	RCKSEL	RXSFT[2:0]			CKMOD
					rw					rw	rw	rw			rw

位/位域	名称	描述
31	保留	必须保持复位值
30	CSNCKM	选择CSN在第一个SCK有效上升沿的之前一个还是两个SCK时钟周期拉低和在最后一个SCK有效上升沿之后一个还是两个SCK时钟周期拉高。 0：1个SCK周期 1：2个SCK周期
29:21	保留	必须保持复位值。
20:16	FMSZ[4:0]	flash存储器大小 该位域定义外部存储器大小： Flash存储器字节数为$2^{[FMSZ+1]}$ FMSZ+1是flash存储器地址位数。普通模式下，flash存储器容量最大到4GB。在内存映射模式下，最大256MB。 该位只能在BUSY = 0时修改。
15:11	保留	必须保持复位值。
10:8	CSHC[2:0]	片选高电平周期数 CSHC+1定义了在两个命令序列之间保持高电平最少的SCK周期数 000：CSN保持高电平至少1个SCK周期 001：CSN保持高电平至少2个SCK周期 ... 111：CSN保持高电平至少8个SCK周期 该位只能在BUSY = 0时修改。
7:6	保留	必须保持复位值。
5	DLYSCEN	延迟扫描功能使能 0：禁能 1：使能 **注意**：该位仅在RCKSEL = 0时生效。该位使能SCK相位微调功能，通过CPDM模块实现。CPDM调整时钟完成后，需再设置DLYSCEN = 0。
4	RCKSEL	接收时钟选择 0：SCK 1：DQS 选择接收时钟源为QSPI内部产生的SCK还是外部设备提供的DQS。当选择DQS时钟时，也可以使用CPDM进行时钟调整，而且不需要将DLYSCEN位置位。

3:1	RXSFT[2:0]	接收移位步长，数据延长超过0.5个周期时。 000：不延迟 001：1个周期 010：2个周期 011：3个周期 100：4个周期 101：5个周期 110：6个周期 111：7个周期 该位域可和SSAMPLE位一起调整接收采样点。
0	CKMOD	该位表明QSPI空闲时SCK电平 0：当CSN为高时，SCK保持低电平 1：当CSN为高时，SCK保持高电平 该位只能在BUSY = 0时修改。

31.8.3. 状态寄存器（QSPI_STAT）

地址偏移：0x08

复位值：0x0000 0004

该寄存器只能按字（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			FL[4:0]					保留		BUSY	TMOUT	RPMF	FT	TC	TERR
			rw							r	r	r	r	r	r

位/位域	名称	描述
31:13	保留	必须保持复位值。
12:8	FL[4:0]	FIFO等级 该位域给出FIFO在普通模式下存储的有效字节数。在内存映射模式和读轮询模式下，FL为0。
7:6	保留	必须保持复位值。
5	BUSY	忙状态 flash该位在命令传输时设置。在普通模式下，当一次操作完成后，该位将被清除。如果在普通读模式下，FIFO也必须为空。
4	TMOUT	超时标志

		当TMOUTEN被设置并且在TMOUTCYC[15:0]个周期之后没有访问闪存时，该位置1。
3	RPMF	读轮询匹配标志 在读轮询模式下，当接收到数据匹配QSPI_STATMATCH寄存器中期望值时，该位置1。
2	FT	FIFO阈值标志 在普通模式下，当FIFO阈值到达或者最后一次读操作时FIFO非空，该位置1。在阈值条件不再满足时由硬件清0。 在读轮询模式下，每次从外部flash读取状态寄存器时置位，DATA寄存器被读取时清0。
1	TC	传输完成标志 在普通模式下，当QSPI_DTLEN寄存器中已编程的数据长度以普通模式或终止操作完成时，该位置1。
0	TERR	传输错误标志 在普通模式下，当访问了无效地址时，该位置1。

31.8.4. 状态清除寄存器（QSPI_STATC）

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											TMOUTC	RPMFC	保留	TCC	TERRC
											w	w		w	w

位/位域	名称	描述
31:5	保留	必须保持复位值。
4	TMOUTC	清除超时标志 对该位写1清除QSPI_STAT寄存器的TMOUT标志。
3	RPMFC	清除读轮询匹配标志 对该位写1清除QSPI_STAT寄存器的RPMF标志。
2	保留	必须保持复位值。

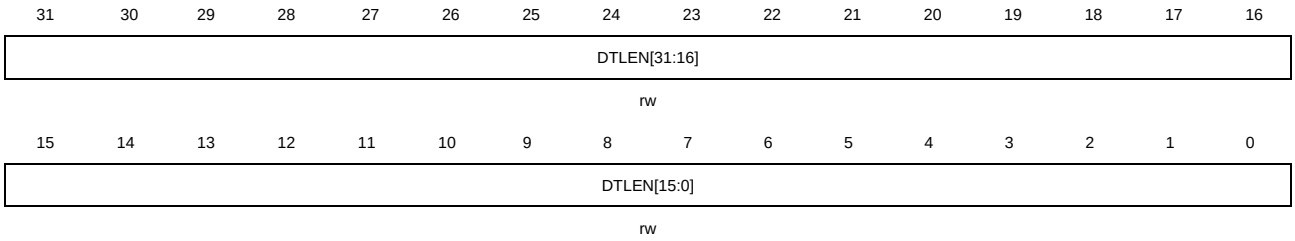
1	TCC	清除传输完成标志 对该位写1清除QSPI_STAT寄存器的TC标志。
0	TERRC	清除传输错误标志 对该位写1清除QSPI_STAT寄存器的TERR标志。

31.8.5. 数据长度寄存器（QSPI_DTLEN）

地址偏移：0x10

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:0	DTLEN[31:0]	数据长度 该位域指定在普通模式和读轮询模式下要传输的数据的个数n（值+1）。在读轮询模式下，该位域的值应该是一个不大于3（表示4字节）的值。在普通模式下，如果该位域配置为0xFFFF FFFF，表示未定义的长度，QSPI将继续传输数据，直到QSPI_DCFG寄存器中的FMSZ[4:0]定义内存地址结束。 0x0000 0000：待传输字节数为1 0x0000 0001：待传输字节数为2 0x0000 0002：待传输字节数为3 0x0000 0003：待传输字节数为4 ... 0xFFFF FFFD：待传输字节数为4,294,967,294 (4G-2) 0xFFFF FFFE：待传输字节数为4,294,967,295 (4G-1) 0xFFFF FFFF：未定义长度，QSPI_DCFG寄存器中由FMSZ[4:0]定义的所有字节，直到内存结束都将被传输。 如果FMSZ[4:0]为0x1F，将无限地继续读取。 内存映射模式下，该位无影响。 该位只能在BUSY = 0时修改。

31.8.6. 传输配置寄存器（QSPI_TCFG）

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
DDREN	DDRHEN	保留	SIOO	FMOD		DATAMOD[1:0]		保留	DUMYC[4:0]					ALTESZ[1:0]	
rw	rw		rw	rw		rw			rw					rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALTEMOD[1:0]		ADDRSZ[1:0]		ADDRMOD[1:0]		IMOD[1:0]		INSTRUCTION[7:0]							
rw		rw		rw		rw		rw							

位/位域	名称	描述
31	DDREN	双倍传输速率模式使能 0：禁能 1：使能
30	DDRHEN	DDR输出保持使能 在 DDR 模式下，延迟1/4个QSPI输出时钟周期再输出数据 0：禁能 1：使能 在 QSPI 分频时使用。
29	保留	必须保持复位值。
28	SIOO	只发送一次指令模式 当IMOD = 00时，该位没有影响。 0：每次命令序列都发送指令 1：命令序列第一次时发送指令 该位只能在BUSY = 0时修改。
27::26	FMOD[1:0]	工作状态 00：普通模式写操作 01：普通模式读操作 10：读轮询模式 11：内存映射模式 如果DMAEN位置1，在改变FMOD位域之前，DMA控制器的相应通道必须关闭。 该位域只能在BUSY = 0时修改。
25:24	DATAMOD[1:0]	数据模式 该位定义数据阶段的操作模式。 00：无数据 01：单线传输数据 10：双线传输数据 11：四线传输数据

		该位同时决定空闲阶段操作模式 该位域只能在BUSY = 0时修改。
23	保留	必须保持复位值。
22:18	DUMYC[4:0]	空指令周期数 该位域定义空闲阶段持续时间。 该位域只能在BUSY = 0时修改。
17:16	ALTESZ[1:0]	交替字节大小 00：8位交替字节 01：16位交替字节 10：24位交替字节 11：32位交替字节 该位域只能在BUSY = 0时修改。
15:14	ALTEMOD[1:0]	交替字节模式 00：无交替字节 01：单线传输交替字节 10：双线传输交替字节 11：四线传输交替字节 该位域只能在BUSY = 0时修改。
13:12	ADDRSZ[1:0]	地址大小 00：8位地址 01：16位地址 10：24位地址 11：32位地址 该位域只能在BUSY = 0时修改。
11:10	ADDRMOD[1:0]	地址模式 00：无地址 01：单线传输地址 10：双线传输地址 11：四线传输地址 该位域只能在BUSY = 0时修改。
9:8	IMOD[1:0]	命令模式 00：无指令 01：单线传输指令 10：双线传输指令 11：四线传输指令 该位域只能在BUSY = 0时修改。

7:0	INSTRUCTION[7:0]	指令 发送到flash存储器的命令信息。 该位域只能在BUSY = 0时修改。

31.8.7. 地址寄存器（QSPI_ADDR）

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
ADDR [31:16]															
rw															

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

位/位域	名称	描述
31:0	ADDR [31:0]	地址 发送到flash存储器的访问地址。 当BUSY=0或在内存映射模式下，对该位域写入值将被忽略。

31.8.8. 交替字节寄存器（QSPI_ALTE）

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
ALTE[31:16]															
rw															

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

位/位域	名称	描述
31:0	ALTE [31:0]	交替字节 紧随地址之后，发送给flash存储器的可选数据。 该位只能在BUSY = 0时修改。

31.8.9. 数据寄存器（QSPI_DATA）

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
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]	将要与flash存储器交互的数据。 在普通模式下写操作时，在发送到flash存储器之前，写入到该寄存器数据会被存储到FIFO中。如果FIFO为满，写操作会停止直到FIFO有足够空间。 在普通模式下读操作时，读该寄存器获取从flash存储器接收的数据。如果FIFO没有足够的字节数来满足读命令请求，并且BUSY位为1，那么读操作会被停止直到FIFO中有足够的数据或者传输已经完成。 在读轮询模式下，该寄存器包含从flash读取的最后数据。

31.8.10. 状态屏蔽寄存器（QSPI_STATMK）

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
MASK[31:16]															
rw															

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

位/位域	名称	描述
31:0	MASK[31:0]	读轮询模式下状态屏蔽 读轮询模式下从flash接收的状态字节掩码。 对于MASK[31:0]第n位： 0：接收数据的第n位不参与匹配 1：接收数据的第n位参与匹配

该位域只能在BUSY = 0时修改。

31.8.11. 状态匹配寄存器（QSPI_STATMATCH）

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
MATCH[31:16]															
rw															

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

位/位域	名称	描述
31:0	MATCH[31:0]	读轮询模式下状态匹配 与QSPI_STATMK寄存器中值进行比较匹配的期望值。 该位域只能在BUSY = 0时修改。

31.8.12. 间隔寄存器（QSPI_INTERVAL）

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
INTERVAL[15:0]															
rw															

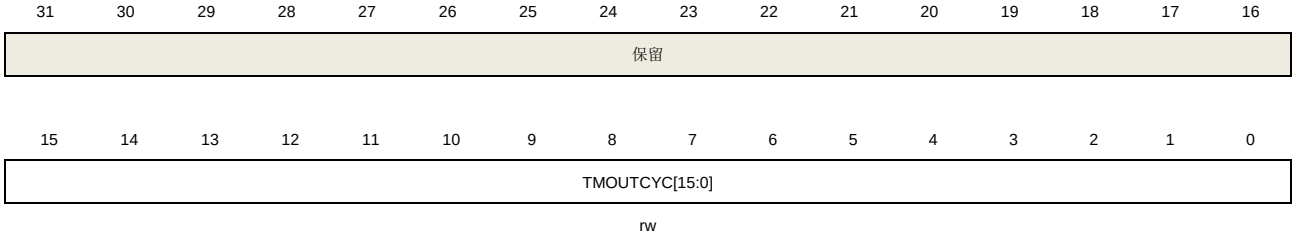
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	INTERVAL[15:0]	间隔周期 读轮询模式下两次读命令之间的SCK周期数。 该位域只能在BUSY = 0时修改。

31.8.13. 超时寄存器（QSPI_TMOUT）

地址偏移：0x30

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



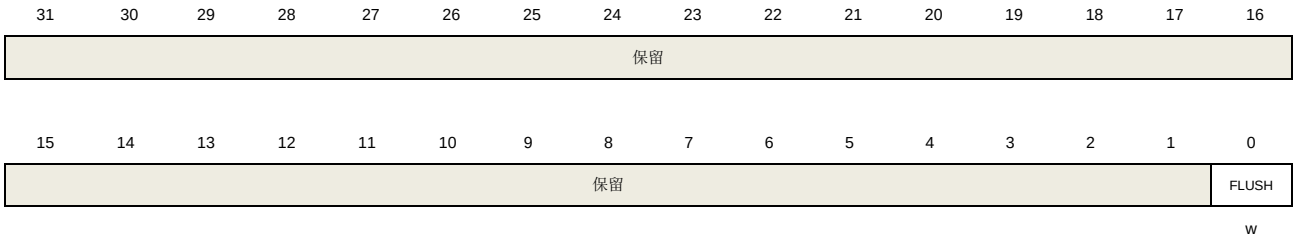
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	TMOUTCYC[15:0]	超时周期 当内存映射模式，FIFO满时，该位域表明在下次访问到来时片选保持低电平的SCK周期数。 该位域只能在BUSY = 0时修改。 **注意：**该位域不能设置为0，如果超时功能打开。

31.8.14. FIFO 刷新寄存器（QSPI_FLUSH）

地址偏移：0x34

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:1	保留	必须保持复位值。
0	FLUSH	用于刷新所有内部FIFO

32. 时钟相位延迟模块（CPDM）

32.1. 简介

时钟相位延迟模块（CPDM）模块用于将输入时钟的相位延迟后再输出时钟。使用时，应用程序需要先对输出时钟的相位进行编程，再在其他外设中使用该输出时钟接收数据。

该延迟与电压和温度相关，当参数变化时，可能需要重新配置应用程序和重新确定输出时钟与接收数据之间的相位关系。

32.2. 主要特征

- 支持输入时钟的频率范围：25MHz ~ 208MHz。
- 支持多达12个输出时钟相位选择。

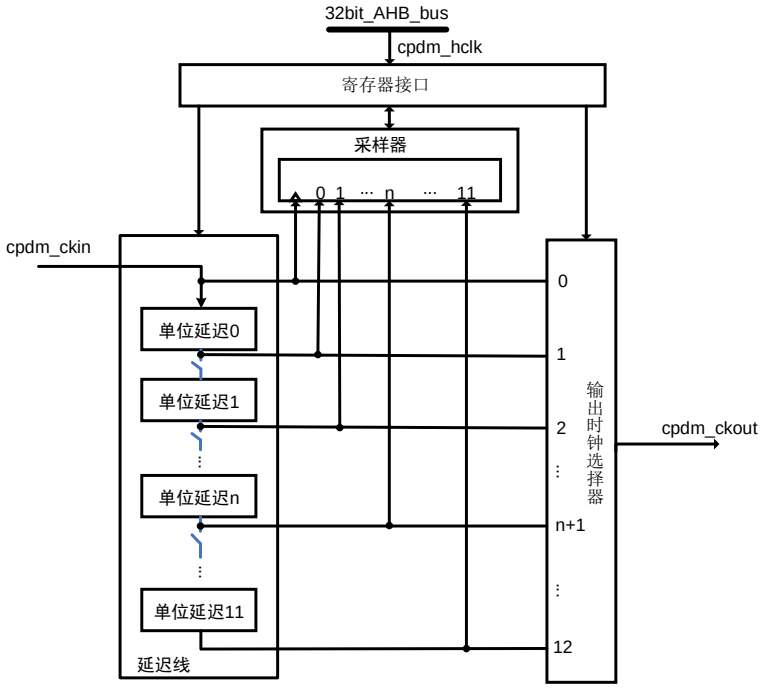
32.3. 功能说明

时钟相位延迟模式（CPDM）主要包括寄存器接口模块、延迟线模块、延迟线长度采样模块及输出时钟选择复用器等模块。延迟线模块用于产生单位延迟。

注意：

CPDM 仅在 QSPI 中使用，请参考*图 31-1. QSPI 结构框图*。

图 32-1. CPDM 模块结构框图



注意：

cpdm_hclk：时钟相位延迟模块的寄存器接口时钟。
cpdm_ckin：时钟相位延迟模块的输入时钟。
cpdm_ckout：时钟相位延迟模块的输出时钟。

32.3.1. 概述

配置CPDM_CTL寄存器中的CPDMEN位可以使能CPDM模块。配置CPDM_CTL寄存器中的DLSEN位可以使能延迟线长度采样模块。

在对输入时钟移相之前，应将延迟线长度配置为一个输入时钟周期。当CPDM模块使能且延迟线长度采样模块使能（DLSEN = 1）时，通过配置CPDM_CFG寄存器中DLSTCNT[11:0]位域定义一个单位延迟单元的所需的延迟步长的数目。当延迟线长度采样模块使能（DLSEN = 1）时，长度采样器可以访问CPDM_CFG寄存器中的延迟线长度（DLLEN）和长度有效标志（DLLENF），因此可以通过使能长度采样器来配置延迟线长度。当延迟线长度采样模块禁能（DLSEN = 0）时，CPDM_CFG寄存器CPSEL位来选择输出时钟相位。

延迟线长度配置完成且延迟线长度采样模块禁能（DLSEN = 0）时，就可以通过时钟相位选择器最终输出经过移相的时钟。

- 当CPDMEN = 0且DLSEN = 0时，CPDM输出时钟使能，此时输出时钟为输入时钟。

- 当DLSEN = 1时，延迟线采样模块使能，CPDM输出时钟禁能。
- 当CPDMEN = 1且DLSEN = 0时，CPDM输出时钟使能，输出时钟相位由CPDM_CFG寄存器中CPSEL[3:0]位域配置决定。

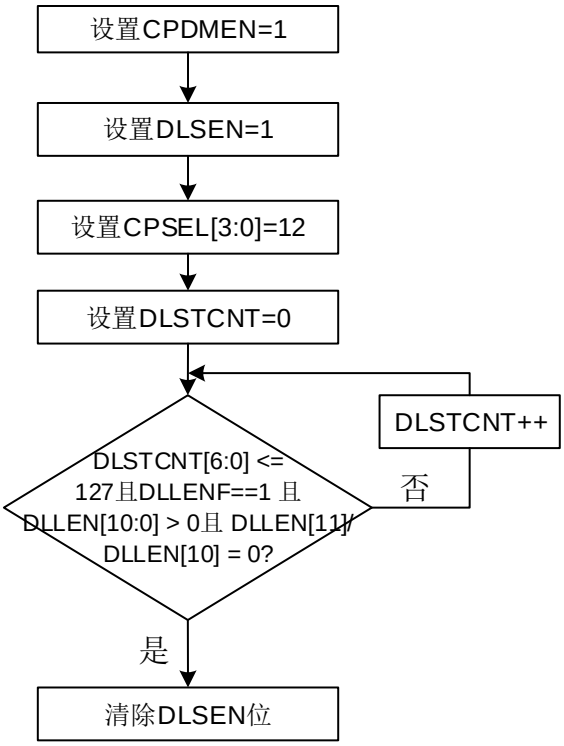
注意：只有当DLSEN为1时才能更改单位延迟及输出时钟相位。

32.3.2. 操作流程

步骤一

配置输出时钟相位之前需要先配置延迟线长度。配置的延迟线长度应当覆盖一个完整的输入时钟周期。具体配置流程如*图32-2. CPDM延迟线长度配置流程图*。

图 32-2. CPDM 延迟线长度配置流程图



注意：

1.如果 DLLEN[10:0] > 0 且 DLLEN[11] / DLLEN[10] = 0，则将延迟线长度配置为一个输入时钟周期。

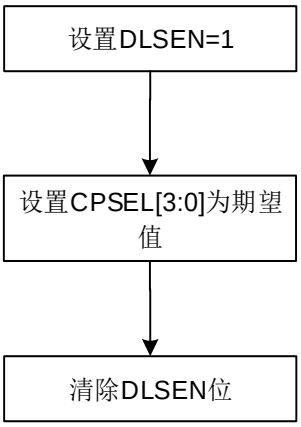
2.对于 N = 10~0，如果 DLLEN[N] = 1，则涵盖一个输入时钟周期的单位延迟数量为 N。

步骤二

当将延迟线长度配置为一个输入时钟周期后，可以在涵盖一个输入时钟周期的单位延迟 N 之间选

择输出时钟相位，具体方法如*图 32-3. CPDM 输出时钟相位配置流程图*。

图 32-3. CPDM 输出时钟相位配置流程图



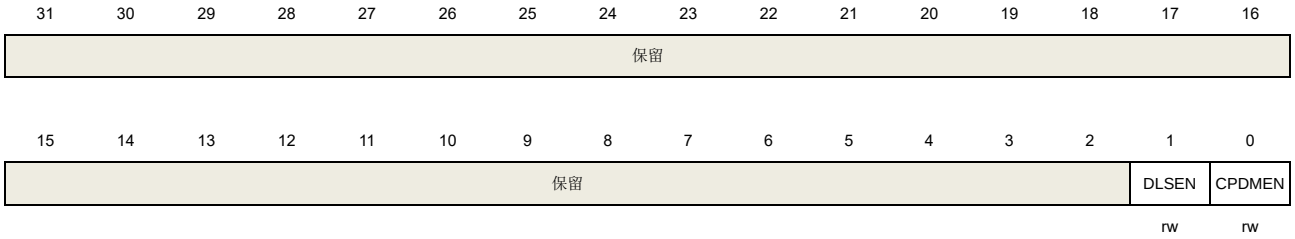
32.4. CPDM 寄存器

CPDM基地址：0x4802 2800

32.4.1. 控制寄存器（CPDM_CTL）

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

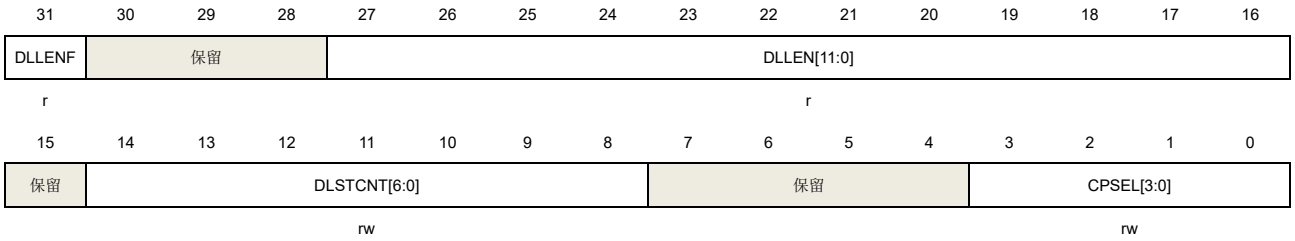


位/位域	名称	描述
31:2	保留	必须保持复位值。
1	DLSEN	延迟线采样模块使能位 0：禁能 CPDM 延迟线采样模块 1：使能 CPDM 延迟线采样模块
0	CPDMEN	CPDM 使能位 0：禁能 CPDM 1：使能 CPDM

32.4.2. 配置寄存器（CPDM_CFG）

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31	DLLENF	延迟线长度有效标志 0：DLLEN[11:0]中的长度值无效。 1：DLLEN[11:0]中的长度值有效。
30:28	保留	必须保持复位值。
27:16	DLLEN[11:0]	延迟线长度值 在输入时钟的上升沿采样的 12 个单位延迟值。 仅在 DLLENF = 1 时有效。
15	保留	必须保持复位值。
14:8	DLSTCNT[6:0]	定义一个单位延迟单元的所需的延迟步长的计数值 仅当 DLSEN = 1 时，才可写入。 0000000：单位延迟 = 初始延迟 0000001：单位延迟 = 初始延迟+ 1 *延迟步长 … 1111111：单位延迟 = 初始延迟+ 127 *延迟步长
7:4	保留	必须保持复位值。
3:0	CPSEL[3:0]	输出时钟相位选择 仅当 DLSEN = 1 时，才可写入。 输出时钟的相位 = 输入时钟 + CPSEL * 单位延迟 0000：输出时钟的相位 = 输入时钟 0001：输出时钟的相位 = 输入时钟 + 1 * 单位延迟 .. 1100：输出时钟的相位 = 输入时钟 + 12 * 单位延迟 1101 ~ 1111：保留

33. 外部存储器控制器（EXMC）

33.1. 简介

外部存储器控制器EXMC，用来访问各种片外存储器，通过配置寄存器，EXMC可以把AMBA协议转换为专用的片外存储器通信协议，包括SRAM，ROM，NOR Flash，PSRAM。用户还可以调整相关的时间参数来提高通信效率。EXMC的访问空间被划分为许多个块（Bank），每个块支持特定的存储器类型，用户可以通过对Bank的控制寄存器配置来控制外部存储器。

33.2. 主要特性

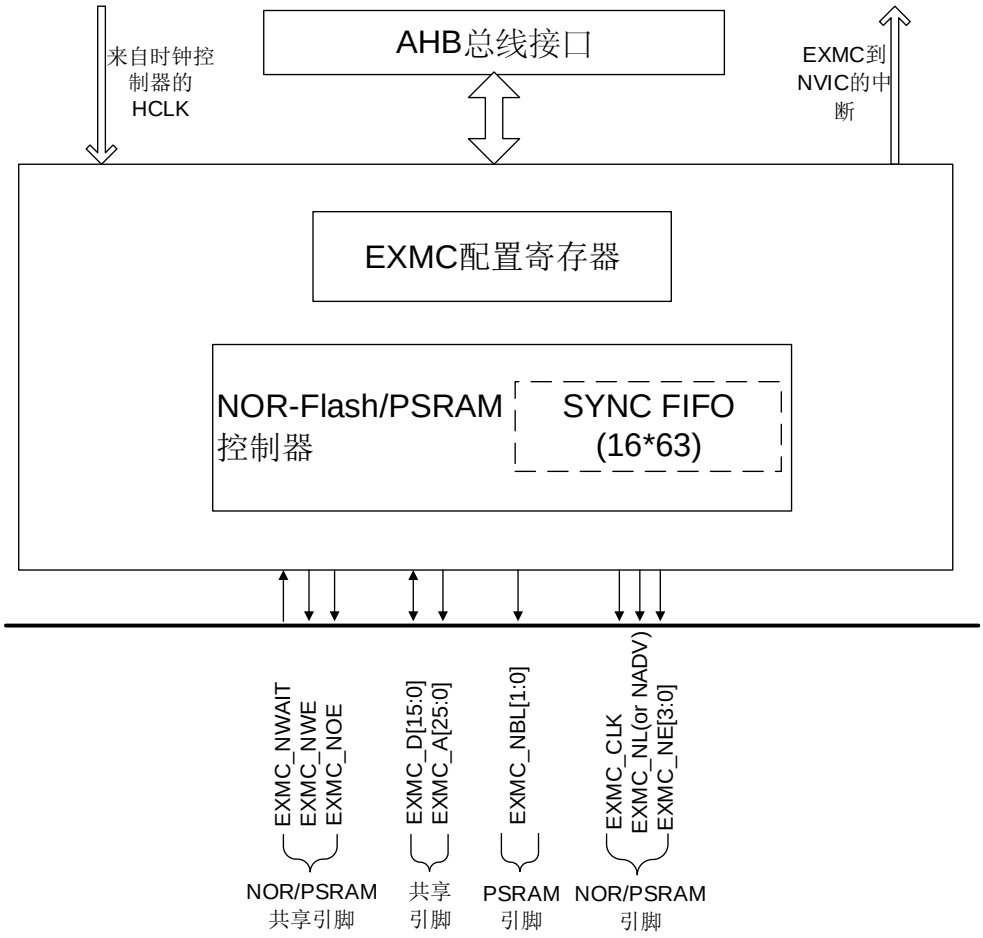
- 支持片外存储器类型：
  - SRAM
  - PSRAM
  - ROM
  - NOR Flash
- AMBA协议与各种片外存储器协议转换；
- 时序参数可编程可以满足用户特定需求；
- 每个Bank有独立的片选信号；
- 对于部分存储器类型支持独立的读写时序；
- 支持8位，16位总线带宽；
- NOR Flash和PSRAM支持地址总线和数据总线的复用；
- 提供写使能和字节选择信号；
- 当AMBA总线宽度与外部存储器数据宽度不同时，会自动分割操作；
- 写FIFO最多16个字的数据存储。

33.3. 功能说明

33.3.1. 结构框图

EXMC由4个模块组成：AHB总线接口，EXMC配置寄存器，NOR / PSRAM控制器，和外部设备接口。AHB时钟（HCLK）是参考时钟。

图 33-1. 系统架构



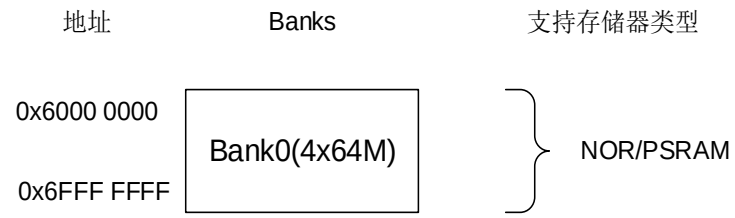
33.3.2. EXMC 访问基本规范

EXMC 是 AHB 总线至外部设备协议的转换接口。32 位的 AHB 读写操作可以转化为几个连续的 8 位或 16 位读写操作。在数据传输的过程中，AHB 数据宽度和存储器数据宽度可能不相同。为了保证数据传输的一致性，EXMC 读写访问需要遵从以下规范：

- AHB 访问宽度等于存储器宽度，则没有问题；
- AHB 访问宽度大于存储器宽度，则自动将 AHB 访问分割成几个连续的存储器数据宽度的传输；
- AHB 访问宽度小于存储器宽度。如果外部存储设备具有字节选择功能，如 SRAM、ROM、PSRAM，则可通过它的字节通道 EXMC_NBL[1:0]来访问对应的字节。否则禁止写操作，只允许读操作。

33.3.3. 外部设备地址映射

图 33-2. EXMC Bank 划分



EXMC 将外部存储器分成多个 Bank，每个 Bank 占 256M 字节，仅 Bank0 有效。Bank0 分为 4 个 Region，每个 Region 占 64M 字节。

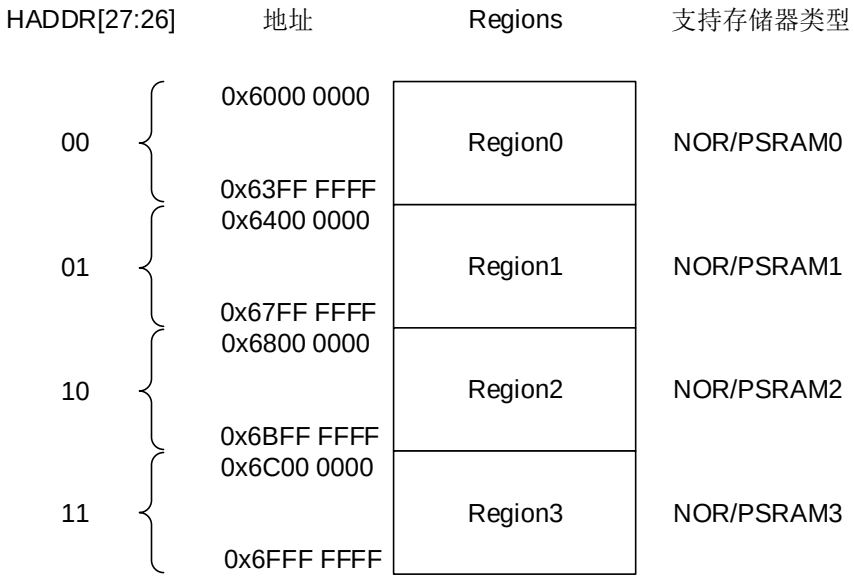
每个 Bank 和 Region 都有独立的片选控制信号，也都能进行独立的配置。

Bank0 用于访问 NOR、PSRAM 设备。

NOR 和 PSRAM 的地址映射

*图 33-3. Bank0 地址映射*是 Bank0 四个 Region 的地址映射。AHB 地址线 HADDR[27:26]作为四个 Region 的片选信号。

图 33-3. Bank0 地址映射



由于 HADDR[25:0]是字节地址，而外部存储器访问有可能不是按字节访问的，所以会出现地址不一致的情况，但 EXMC 能实现对 HADDR 的调整以适应外部存储器的数据宽度。具体规则如下：

- 如果外部存储器的数据宽度是 8 位按字节对齐，EXMC 内部将 HADDR[25:0]与 EXMC_A[25:0]相连，然后 EXMC_A[25:0]与外部存储器的地址线相连；
- 如果外部存储器的数据宽度是 16 位按半字对齐，就需要将 HADDR 的字节地址转化为半字地址之后再连接外存储器。EXMC 内部将 HADDR[25:1]与 EXMC_A[24:0]相连，然后 EXMC_A[24:0]与外部存储器的地址线相连。

33.3.4. NOR / PSRAM 控制器

EXMC 模块的 NOR / PSRAM 控制器控制 Bank0，它可以支持 NOR Flash、PSRAM、SRAM、ROM 和 CRAM 外部存储器。EXMC 对 Bank0 每个 Region 输出一个唯一的片选信号，NE[x](x=0...3)，用于在 4 个 Region 中进行片选，所有其他的信号都是共享的。每个 Region 都有专门的寄存器控制。

注意：

在异步模式下，所有控制器输出信号在内部 AHB 总线时钟（HCLK）的上升沿改变。

在同步模式下，所有控制器输出数据在外部存储器时钟（EXMC_CLK）的下降沿改变。

NOR / PSRAM 接口描述

表 33-1. NOR Flash 接口信号描述

EXMC 引脚	传输方向	模式	功能描述
EXMC_CLK	输出	同步	同步时钟信号
非复用 EXMC_A[25:0]	输出	异步/同步	地址总线
复用 EXMC_A[25:16]			
EXMC_D[15:0]	输入/输出	异步/同步（复用）	地址/数据总线
	输入/输出	异步/同步（非复用）	数据总线
EXMC_NE[x]	输出	异步/同步	片选，x=0/1/2/3
EXMC_NOE	输出	异步/同步	读使能
EXMC_NWE	输出	异步/同步	写使能
EXMC_NWAIT	输入	异步/同步	等待输入信号
EXMC_NL(NADV)	输出	异步/同步	地址有效

表 33-2. PSRAM 非复用接口信号描述

EXMC 引脚	传输方向	模式	功能描述
EXMC_CLK	输出	同步	同步时钟信号
EXMC_A[25:0]	输出	异步/同步	地址总线
EXMC_D[15:0]	输入/输出	异步/同步	数据总线
EXMC_NE[x]	输出	异步/同步	片选，x=0/1/2/3
EXMC_NOE	输出	异步/同步	读使能
EXMC_NWE	输出	异步/同步	写使能
EXMC_NWAIT	输入	异步/同步	等待输入信号
EXMC_NL(NADV)	输出	异步/同步	地址锁存信号（地址有效使能，NADV）
EXMC_NBL[1]	输出	异步/同步	高字节使能
EXMC_NBL[0]	输出	异步/同步	低字节使能

支持的存储器访问模式

*表 33-3. 支持的 16 位传输*和*表 33-4. 支持的 8 位传输*列出了当存储器访问总线是 8 位或 16 位时 EXMC 对 NOR，PSRAM 和 SRAM 支持的访问模式。

表 33-3. 支持的 16 位传输

存储器类型	访问模式	读/写	AHB 传输宽度	存储器传输宽度	注释
NOR Flash	异步	R	8	16	
	异步	W	8	16	不支持
	异步	R	16	16	
	异步	W	16	16	
	异步	R	32	16	分成 2 次 EXMC 访问
	异步	W	32	16	分成 2 次 EXMC 访问
	同步	R	8	16	不支持
	同步	R	16	16	
	同步	R	32	16	分成 2 次 EXMC 访问
PSRAM	异步	R	8	16	
	异步	W	8	16	使用字节信号 NBL[1:0]
	异步	R	16	16	
	异步	W	16	16	
	异步	R	32	16	分成 2 次 EXMC 访问
	异步	W	32	16	分成 2 次 EXMC 访问
	同步	R	8	16	不支持
	同步	R	16	16	
	同步	R	32	16	
	同步	W	8	16	使用字节信号 NBL[1:0]
	同步	W	16	16	
	同步	W	32	16	
SRAM 和 ROM	异步	R	8	16	
	异步	R	16	16	
	异步	R	32	16	分成 2 次 EXMC 访问
	异步	W	8	16	使用字节信号 NBL[1:0]
	异步	W	16	16	
	异步	W	32	16	

表 33-4. 支持的 8 位传输

存储器类型	访问模式	读/写	AHB 传输宽度	存储器传输宽度	注释
NOR Flash	异步	R	8	8	
	异步	W	8	8	
	异步	R	16	8	分成 2 次 EXMC 访问
	异步	W	16	8	分成 2 次 EXMC 访问
	异步	R	32	8	分成 4 次 EXMC 访问
	异步	W	32	8	分成 4 次 EXMC 访问

存储器类型	访问模式	读/写	AHB 传输宽度	存储器传输宽度	注释
	同步	R	8	8	
	同步	R	16	8	分成 2 次 EXMC 访问
	同步	R	32	8	分成 4 次 EXMC 访问
PSRAM	异步	R	8	8	
	异步	W	8	8	使用字节信号 NBL[1:0]
	异步	R	16	8	分成 2 次 EXMC 访问
	异步	W	16	8	分成 2 次 EXMC 访问
	异步	R	32	8	分成 4 次 EXMC 访问
	异步	W	32	8	分成 4 次 EXMC 访问
	同步	R	8	8	
	同步	R	16	8	分成 2 次 EXMC 访问
	同步	R	32	8	分成 4 次 EXMC 访问
	同步	W	8	8	使用字节信号 NBL[1:0]
	同步	W	16	8	分成 2 次 EXMC 访问
	同步	W	32	8	分成 4 次 EXMC 访问
SRAM 和 ROM	异步	R	8	8	
	异步	R	16	8	分成 2 次 EXMC 访问
	异步	R	32	8	分成 4 次 EXMC 访问
	异步	W	8	8	使用字节信号 NBL[1:0]
	异步	W	16	8	分成 2 次 EXMC 访问
	异步	W	32	8	分成 4 次 EXMC 访问

注意：在异步模式下，当一次 EXMC 访问被分成多个 EXMC 访问时，EXMC 可能无法达到最高频率。

NOR Flash / PSRAM 控制时序

EXMC 为 SRAM、ROM、PSRAM、NOR Flash 等外部静态存储器提供可编程的时序参数以及多种时序模型以满足不同的需求。

表 33-5. NOR / PSRAM 控制时序参数

参数	功能	访问模式	单位	最小值	最大值
CKDIV	同步时钟分频比	同步	HCLK	2	16
DLAT	数据延迟	同步	EXMC_CLK	2	17
BUSLAT	总线延迟	异步/同步读	HCLK	0	15
DSET	数据建立时间	异步	HCLK	1	255
AHLD	地址保持时间	异步（复用）	HCLK	1	15
ASET	地址建立时间	异步	HCLK	0	15
BLSET	字节信号建立时间	异步	HCLK	0	3

表 33-6. EXMC 时序模型

时序模型		扩展模式	模式描述	写时序参数	读时序参数
异步	模式 1	0	SRAM / PSRAM / CRAM	DSET ASET BLSET	DSET ASET BLSET
	模式 2	0	NOR Flash	DSET ASET	DSET ASET
	模式 A	1	SRAM / PSRAM / CRAM 在数据阶段 EXMC_NOE 翻转	WDSET WASET BLSET	DSET ASET BLSET
	模式 B	1	NOR Flash	WDSET WASET	DSET ASET
	模式 C	1	NOR Flash 在数据阶段 EXMC_NOE 翻转	WDSET WASET	DSET ASET
	模式 D	1	有地址保持功能	WDSET WAHLD WASET BLSET	DSET AHLD ASET BLSET
	模式 AM	0	NOR Flash 数据/地址复用	DSET AHLD ASET BUSLAT BLSET	DSET AHLD ASET BUSLAT BLSET
同步	模式 E	0	NOR / PSRAM / CRAM 同步读 PSRAM / CRAM 同步写	DLAT CKDIV	DLAT CKDIV
	模式 SM	0	NOR Flash/PSRAM/CRAM 数据/地址复用	DLAT CKDIV	DLAT CKDIV

如*表 33-6. EXMC 时序模型*所示，EXMC 模块 NOR Flash / PSRAM 控制器可以提供多种时序模型。用户可以通过修改*表 33-5. NOR / PSRAM 控制时序参数*中列出的参数来使之适合不同类型外部存储器的时序以及满足用户的要求。当将寄存器 EXMC_SNCTLx 位 EXMODEN 置 1 使能扩展模式后，可以通过寄存器 EXMC_SNTCFGx 和 EXMC_SNWTCFGx 将读写配置成独立的时序。

EXMC_CLK 可以通过 CCK 位来设置。如果 CCK 是 0，当 NOR Flash 使用同步模式时会产生 EXMC_CLK；如果 CCK 是 1，当 NOR Flash 同步模式和异步模式都会产生 EXMC_CLK。

异步访问时序

模式 1 – SRAM / CRAM

图 33-4. 模式 1 读时序

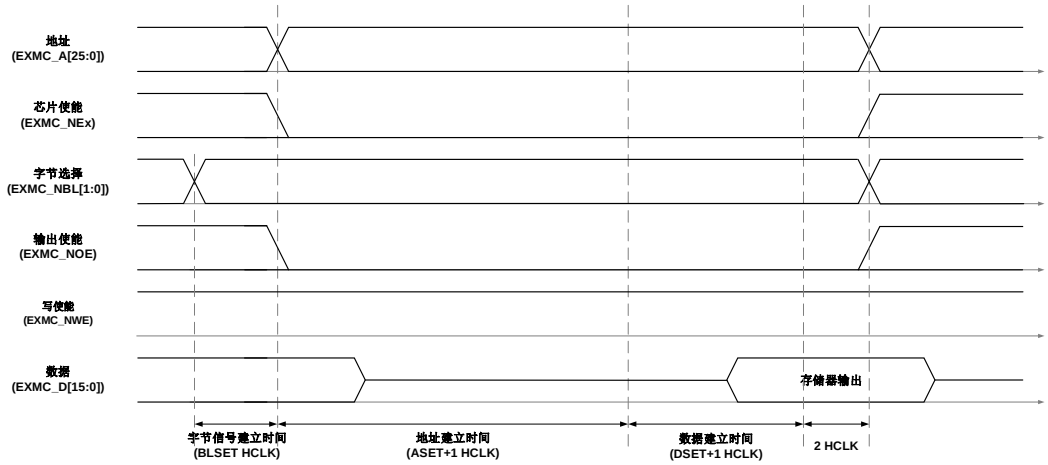


图 33-5. 模式 1 写时序

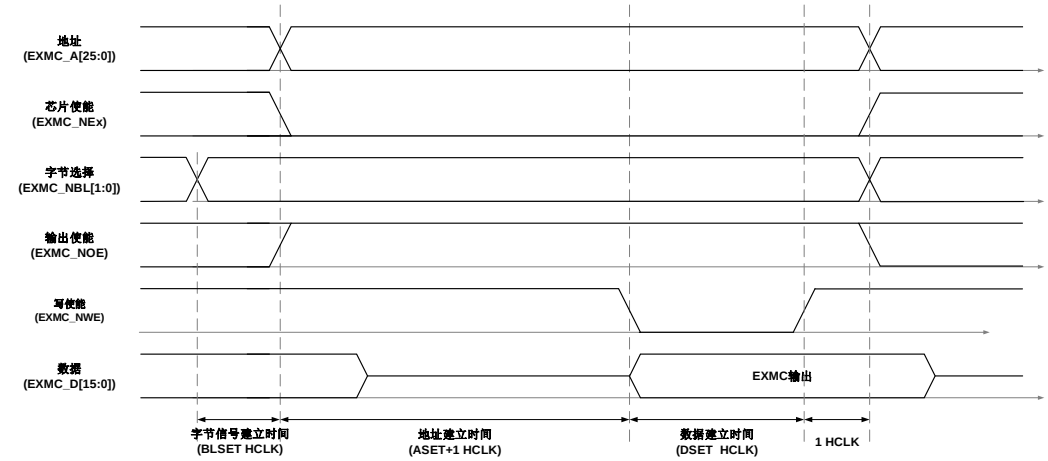


表 33-7. 模式 1 相关寄存器配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
31-24	保留	0x00
23-22	BLSET	取决于用户
21	WFIFODIS	取决于用户
20	CCK	取决于用户
19	SYNCWR	0x0
18-16	CPS	取决于存储器
15	ASYNCWTEN	取决于存储器
14	EXMODEN	0x0
13	NRWTEN	0x0
12	WEN	取决于用户

位域/位	名称	参考设定值
EXMC_SNCTLx		
11	NRWTCFG	无影响
10	WRAPEN	0x0
9	NRWTPOL	仅当位 15 为 1 时有效
8	SBRSTEN	0x0
7	保留	0x1
6	NREN	无影响
5-4	NRW	取决于存储器
3-2	NRTP	取决于存储器，Nor Flash：2
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFGx		
31-30	保留	0x0
29-28	ASYNCMOD	无影响
27-24	DLAT	无影响
23-20	CKDIV	无影响
19-16	BUSLAT	EXMC_NE[x]上升沿到下降沿的时间
15-8	DSET	取决于存储器与用户（写操作为 DSET+1 HCLK 时钟周期，读操作为 DSET+3 HCLK 时钟周期）
7-4	AHLD	无影响
3-0	ASET	取决于存储器与用户
EXMC_SNLATDECx		
31-3	保留	0x0
2-0	LATDEC	无影响

模式 A – SRAM / PSRAM(CRAM) OE 翻转

图 33-6. 模式 A 读时序

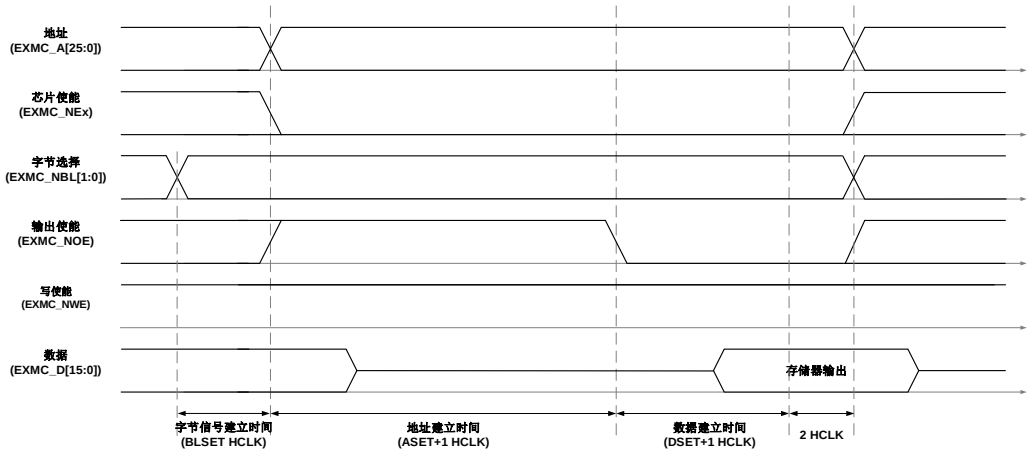
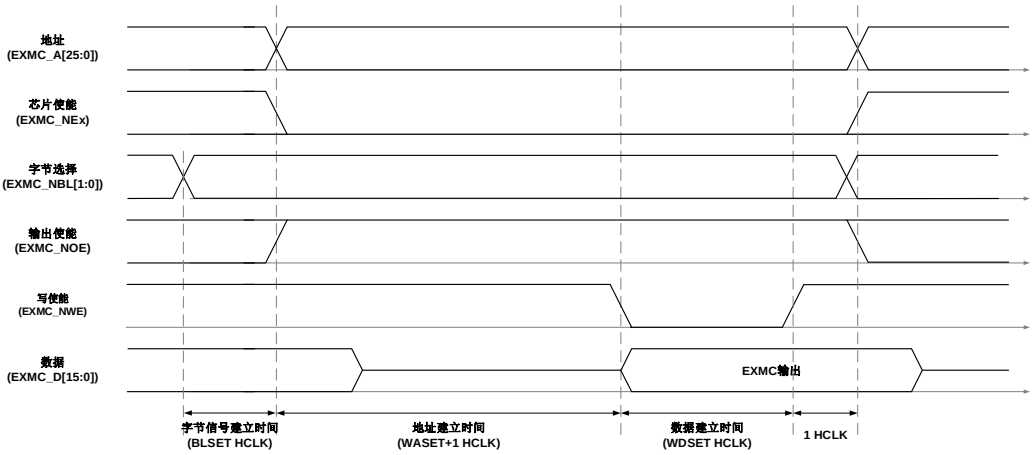


图 33-7. 模式 A 写时序



模式 A 和模式 1 的区别在于写时序，当两个模式的寄存器有相同的时序配置时，模式 A 的写时序独立于读时序。

表 33-8. 模式 A 相关寄存器配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
31-24	保留	0x00
23-22	BLSET	取决于用户
21	WFIFODIS	取决于用户
20	CCK	取决于用户
19	SYNCWR	0x0
18-16	保留	0x0
15	ASYNCWTEN	取决于存储器
14	EXMODEN	0x1

位域/位	名称	参考设定值
13	NRWTEN	0x0
12	WEN	取决于用户
11	NRWTCFG	无影响
10	WRAPEN	0x0
9	NRWTPOL	仅当位 15 为 1 时有效
8	SBRSTEN	0x0
7	保留	0x1
6	NREN	无影响
5-4	NRW	取决于存储器
3-2	NRTP	取决于存储器，Nor Flash：2
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFGx（读）		
31-30	保留	0x0
29-28	ASYNCMOD	0x0
27-24	DLAT	无影响
23-20	CKDIV	无影响
19-16	BUSLAT	EXMC_NE[x]上升沿到下降沿的时间
15-8	DSET	取决于存储器与用户（读操作为 DSET+3 HCLK 时钟周期）
7-4	AHLD	无影响
3-0	ASET	取决于存储器与用户
EXMC_SNWTCFGx（写）		
31-30	保留	0x0
29-28	WASYNCMOD	0x0
27-20	保留	0x0
19-16	WBUSLAT	EXMC_NE[x]上升沿到下降沿的时间
15-8	WDSET	取决于存储器与用户（写操作为 WDSET+1 HCLK 时钟周期）
7-4	WAHLD	0x0
3-0	WASET	取决于存储器与用户
EXMC_SNLATDECx		
31-3	保留	0x0
2-0	LATDEC	无影响

模式 2 / B – NOR Flash

图 33-8. 模式 2 / B 读时序

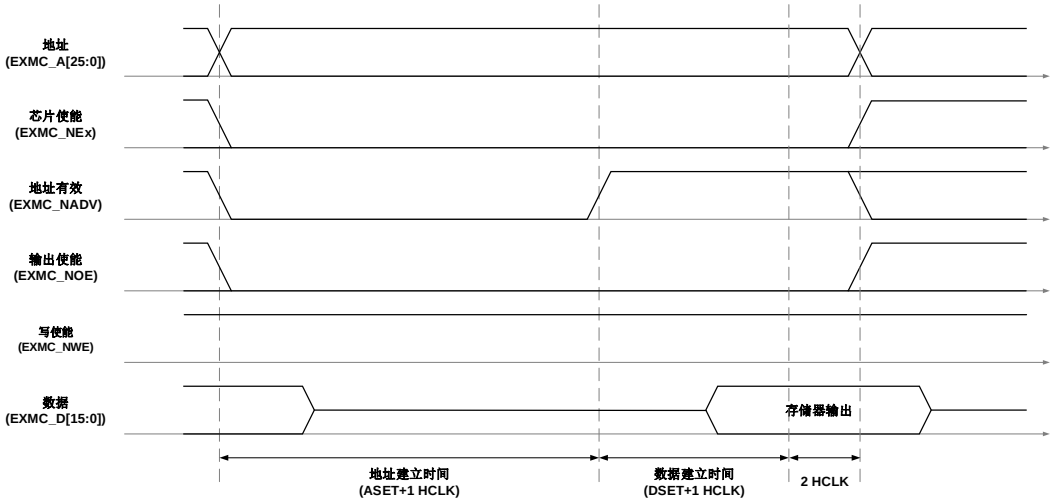


图 33-9. 模式 2 写时序

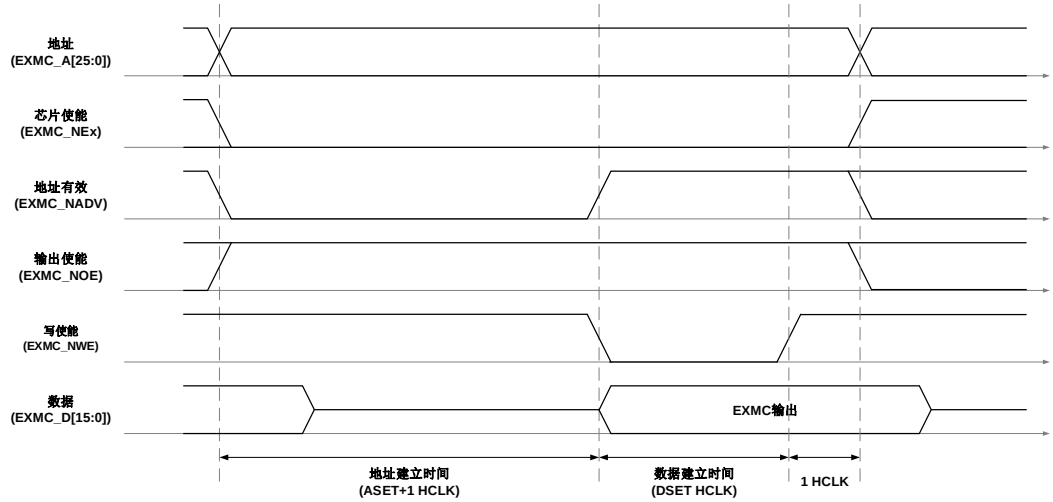


图 33-10. 模式 B 写时序

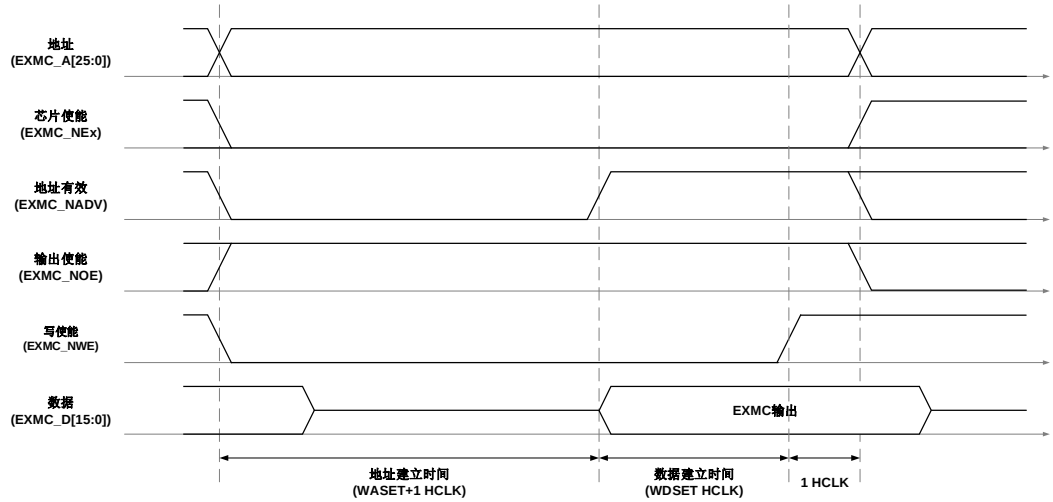


表 33-9. 模式 2 / B 相关寄存器配置

位域/位	名称	参考设定值
EXMC_SNCTLx（模式 2，模式 B）		
31-24	保留	0x00
23-22	BLSET	无影响
21	WFIFODIS	取决于用户
20	CCK	取决于用户
19	SYNCWR	0x0
18-16	保留	0x0
15	ASYNCWTEN	取决于存储器
14	EXMODEN	模式 2：0x0，模式 B：0x1
13	NRWTEN	0x0
12	WEN	取决于用户
11	NRWTCFG	无影响
10	WRAPEN	0x0
9	NRWTPOL	仅当位 15 为 1 时有效
8	SBRSTEN	0x0
7	保留	0x1
6	NREN	0x1
5-4	NRW	取决于存储器
3-2	NRTP	Nor Flash：2
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFGx（模式 2 读/写操作，模式 B 读操作）		
31-30	保留	0x0
29-28	ASYNCMOD	模式 B：0x1
27-24	DLAT	无影响
23-20	CKDIV	无影响
19-16	BUSLAT	EXMC_NE[x]上升沿到下降沿的时间
15-8	DSET	取决于存储器与用户（读操作为 DSET+3 HCLK 时钟周期）
7-4	AHLD	0x0
3-0	ASET	取决于存储器与用户
EXMC_SNWTCFGx（模式 B 写操作）		
31-30	保留	0x0
29-28	WASYNCMOD	模式 B：0x1
27-20	保留	0x0
19-16	WBUSLAT	EXMC_NE[x]上升沿到下降沿的时间

位域/位	名称	参考设定值
15-8	WDSET	取决于存储器与用户（写操作为 WDSET+1 HCLk 时钟周期）
7-4	WAHLD	0x0
3-0	WASET	取决于存储器与用户
EXMC_SNLATDECx		
31-3	保留	0x0
2-0	LATDEC	无影响

模式 C – NOR Flash OE 翻转

图 33-11. 模式 C 读时序

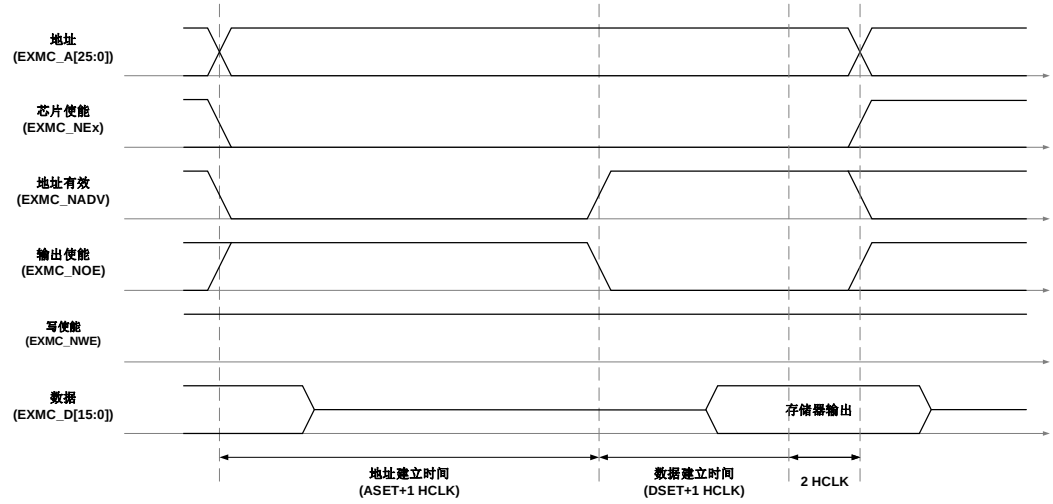
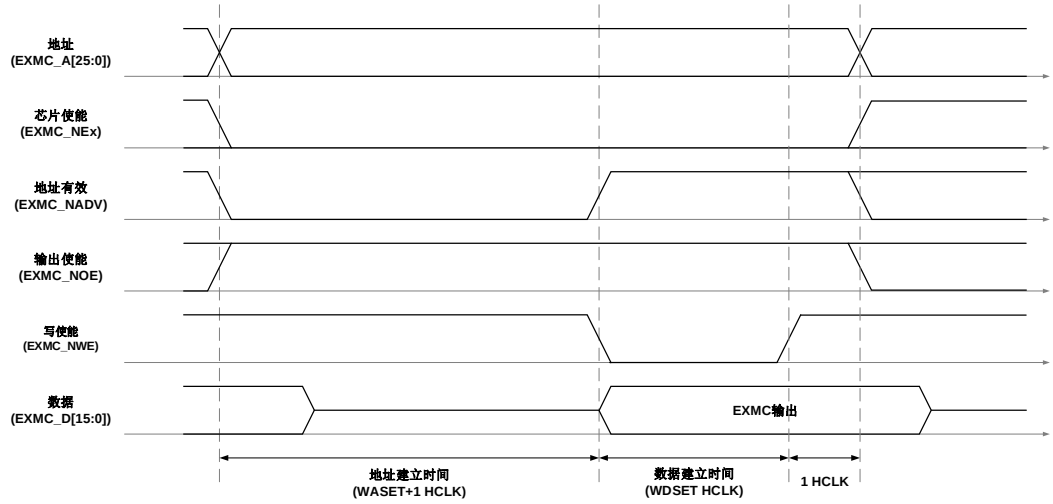


图 33-12. 模式 C 写时序



模式 C 和模式 1 的区别在于写时序，当两个模式的寄存器有相同的时序配置时，模式 C 的写时序独立于读时序并且 NOE 和 NADV 的翻转是不同的。

表 33-10. 模式 C 相关寄存器配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
31-24	保留	0x00
23-22	BLSET	无影响
21	WFIFODIS	取决于用户
20	CCK	取决于用户
19	SYNCWR	0x0
18-16	保留	0x0
15	ASYNCWTEN	取决于存储器
14	EXMODEN	0x1
13	NRWTEN	0x0
12	WEN	取决于用户
11	NRWTCFG	无影响
10	WRAPEN	0x0
9	NRWTPOL	仅当位 15 为 1 时有效
8	SBRSTEN	0x0
7	保留	0x1
6	NREN	0x1
5-4	NRW	取决于存储器
3-2	NRTP	Nor Flash：2
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFGx（读）		
31-30	保留	0x0
29-28	ASYNCMOD	模式 C：0x2
27-24	DLAT	0x0
23-20	CKDIV	0x0
19-16	BUSLAT	EXMC_NE[x]上升沿到下降沿的时间
15-8	DSET	取决于存储器与用户读操作为（DSET+3 HCLk 时钟周期）
7-4	AHLD	0x0
3-0	ASET	取决于存储器与用户
EXMC_SNWTCFGx（写）		
31-30	保留	0x0
29-28	WASYNCMOD	模式 C：0x2
27-20	保留	0x0
19-16	WBUSLAT	EXMC_NE[x]上升沿到下降沿的时间

位域/位	名称	参考设定值
15-8	WDSET	取决于存储器与用户（写操作为 WDSET+1 HCLK 时钟周期）
7-4	WAHLD	0x0
3-0	WASET	取决于存储器与用户
EXMC_SNLATDECx		
31-3	保留	0x0
2-0	LATDEC	无影响

模式 D – 带地址扩展的异步操作

图 33-13. 模式 D 读时序

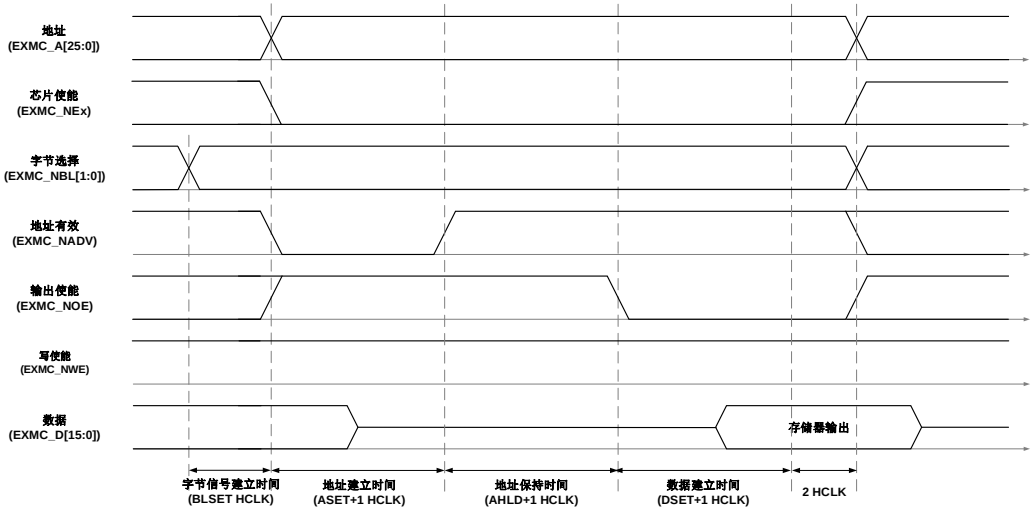


图 33-14. 模式 D 写时序

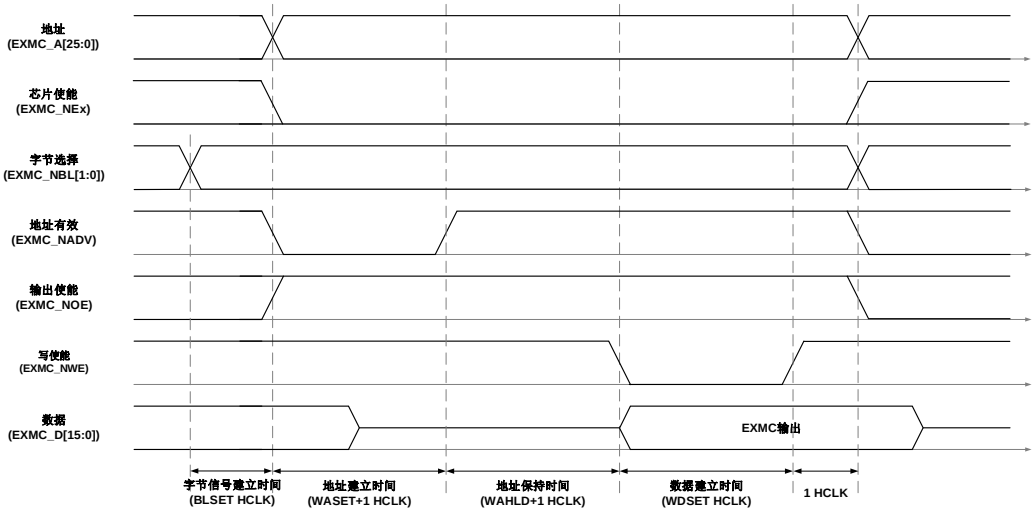


表 33-11. 模式 D 相关寄存器配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
31-24	保留	0x00
23-22	BLSET	取决于用户
21	WFIFODIS	取决于用户
20	CCK	取决于用户
19	SYNCWR	0x0
18-16	保留	0x0
15	ASYNCWTEN	取决于存储器
14	EXMODEN	0x1
13	NRWTEN	0x0
12	WEN	取决于用户
11	NRWTCFG	无影响
10	WRAPEN	0x0
9	NRWTPOL	仅当位 15 为 1 时有效
8	SBRSTEN	0x0
7	保留	0x1
6	NREN	取决于存储器
5-4	NRW	取决于存储器
3-2	NRTP	取决于存储器
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFGx（读）		
31-30	保留	0x0
29-28	ASYNCMOD	模式 D：0x3
27-24	DLAT	无关
23-20	CKDIV	无影响
19-16	BUSLAT	EXMC_NE[x]上升沿到下降沿的时间
15-8	DSET	取决于存储器与用户（读操作为 DSET+3 HCLK 时钟周期）
7-4	AHLD	取决于存储器与用户
3-0	ASET	取决于存储器与用户
EXMC_SNWTCFGx（写）		
31-30	保留	0x0
29-28	WASYNCMOD	模式 D：0x3
27-20	保留	0x0
19-16	WBUSLAT	EXMC_NE[x]上升沿到下降沿的时间

位域/位	名称	参考设定值
15-8	WDSET	取决于存储器与用户（写操作为 WSET+1 HCLk 时钟周期）
7-4	WAHLD	取决于存储器与用户
3-0	WASET	取决于存储器与用户
EXMC_SNLATDECx		
31-3	保留	0x0
2-0	LATDEC	无影响

模式 AM – NOR Flash 地址/数据总线复用

图 33-15. 复用模式读时序

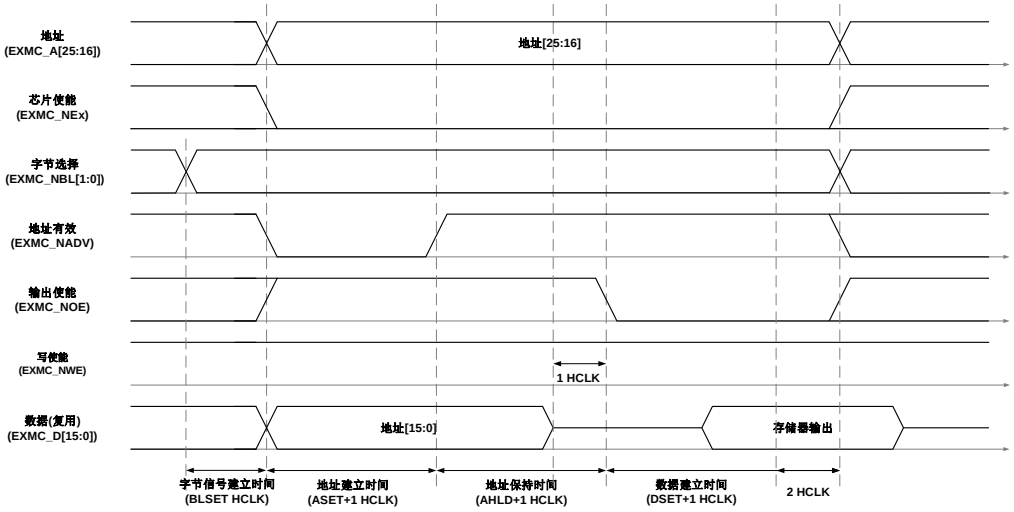


图 33-16. 复用模式写时序

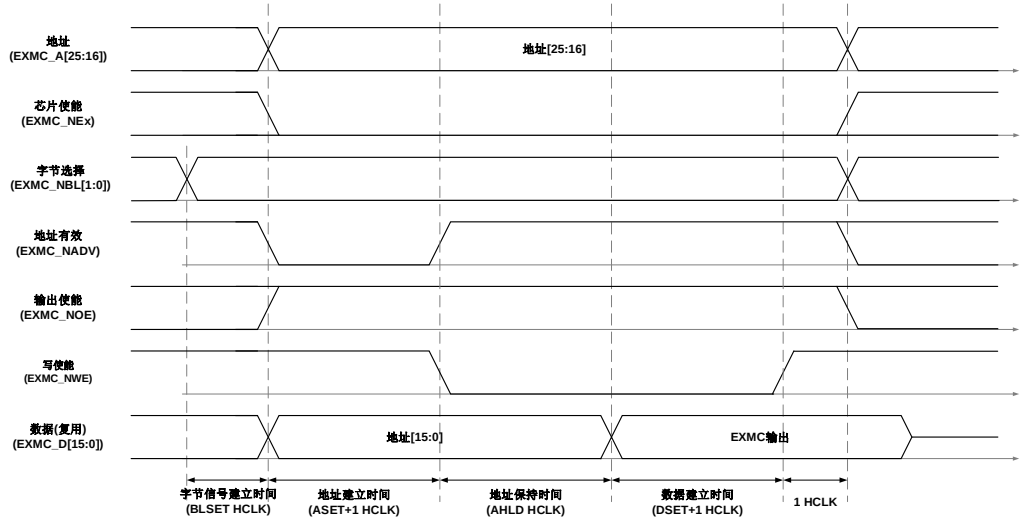


表 33-12. 复用模式相关寄存器配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
31-24	保留	0x00
23-22	BLSET	取决于用户
21	WFIFODIS	取决于用户
20	CCK	取决于用户
19	SYNCWR	0x0
18-16	保留	0x0
15	ASYNCWTEN	取决于存储器
14	EXMODEN	0x0
13	NRWTEN	0x0
12	WEN	取决于存储器
11	NRWTCFG	无影响
10	WRAPEN	0x0
9	NRWTPOL	仅当位 15 为 1 时有效
8	SBRSTEN	0x0
7	保留	0x1
6	NREN	0x1
5-4	NRW	取决于存储器
3-2	NRTP	Nor Flash：2
1	NRMUX	0x1
0	NRBKEN	0x1
EXMC_SNTCFGx		
31-30	保留	0x0
29-28	ASYNCMOD	0x0
27-24	DLAT	无影响
23-20	CKDIV	无影响
19-16	BUSLAT	EXMC_NE[x]上升沿到下降沿的时间
15-8	DSET	取决于存储器与用户（写操作为 DSET+2 HCLK 时钟周期，读操作为 DSET+3 HCLK 时钟周期）
7-4	AHLD	取决于存储器与用户
3-0	ASET	取决于存储器与用户
EXMC_SNLATDECx		
31-3	保留	0x0
2-0	LATDEC	无影响

异步通信的等待时间：

等待功能由寄存器 EXMC_SNCTL 位 ASYNCWTEN 控制。在访问外部存储器期间，若使能异步

等待功能（ASYNCWTEN=1），数据建立时间将会自动延长。延长时间的计算如下：

若存储器等待信号与 EXMC_NOE / EXMC_NWE 信号对齐：

$$T_{DATA_SETUP} \geq \max T_{WAIT_ASSERTION} + 4HCLK \quad (33\text{-}1)$$

若存储器等待信号与 EXMC_NE 信号对齐：

如果

$$\max T_{WAIT_ASSERTION} \geq T_{ADDRES_PHASE} + T_{HOLD_PHASE} \quad (33\text{-}2)$$

则

$$T_{DATA_SETUP} \geq (\max T_{WAIT_ASSERTION} - T_{ADDRES_PHASE} - T_{HOLD_PHASE}) + 4HCLK \quad (33\text{-}3)$$

否则

$$T_{DATA_SETUP} \geq 4HCLK \quad (33\text{-}4)$$

图 33-17. 异步等待有效时的读时序

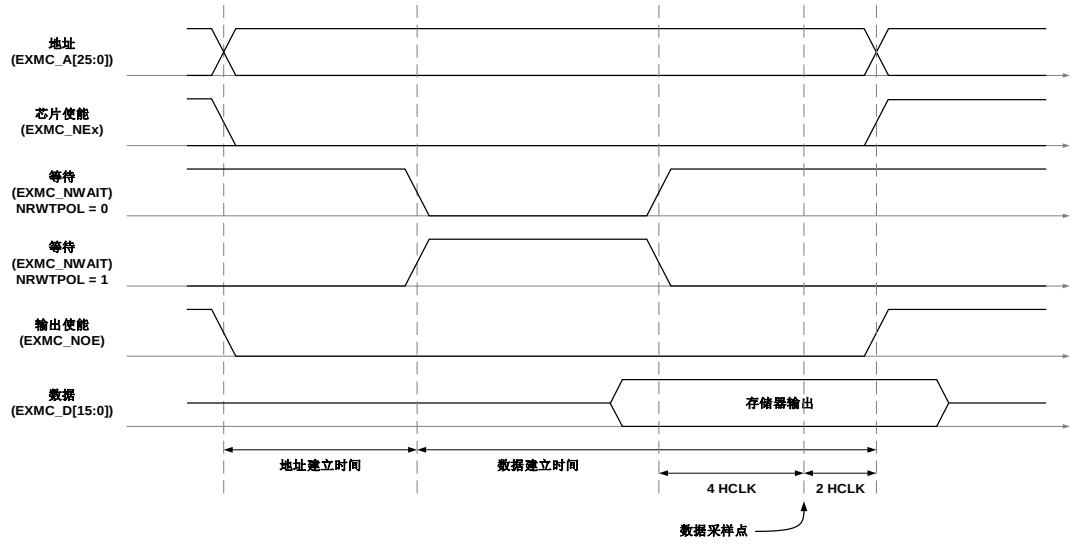
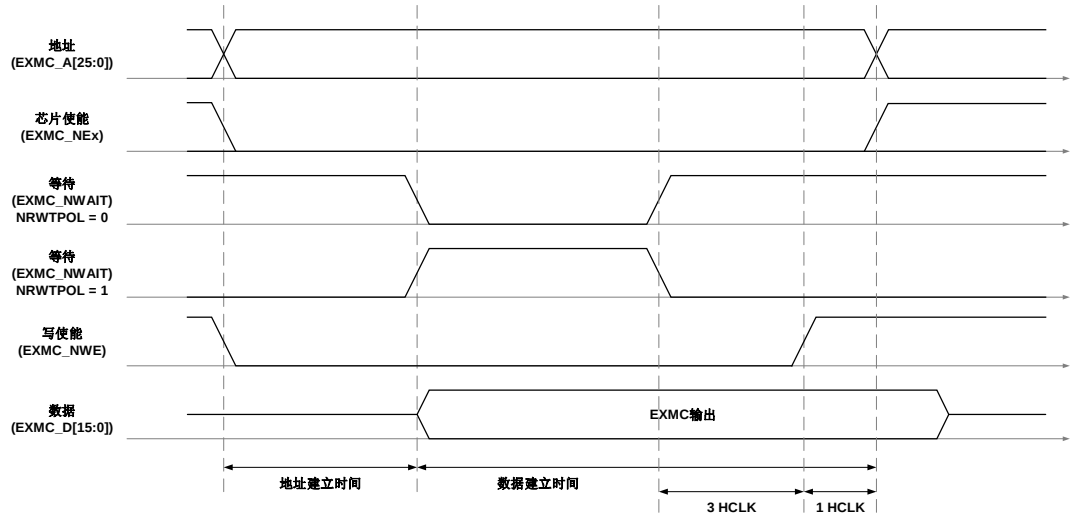


图 33-18. 异步等待有效时的写时序



同步访问时序

存储器时钟（EXMC_CLK）与系统时钟（HCLK）关系如下：

$$EXMC_CLK=\frac{HCLK}{CKDIV+1} \quad (33\text{-}5)$$

其中 CKDIV 是同步时钟分频比，通过配置寄存器 EXMC_SNTCFGx 中的 CKDIV 位来设置不同的值。

1. 数据延迟与 NOR Flash 延迟

数据延迟 DLAT 是指在采样数据之前需要等待的 EXMC_CLK 周期数。它和 NOR 闪存延迟的关系如下：

NOR 闪存延迟不包含 EXMC_NADV，二者之间的关系为：

$$NOR\ 闪存延迟=DLAT+2 \quad (33\text{-}6)$$

NOR 闪存延迟包含 EXMC_NADV，二者之间的关系为：

$$NOR\ 闪存延迟=DLAT+3 \quad (33\text{-}7)$$

注意：

在读访问时，数据延迟由 EXMC_SNTCFGx 寄存器中 DLAT 和 EXMC_SNLATDECx 寄存器中的 LATDEC 共同决定。具体请参考 ***SRAM / NOR Flash 数据延迟减少寄存器(EXMC_SNLATDECx)（x=0, 1, 2, 3）***。

2. 数据等待

用户需要保证 EXMC_NWAIT 信号与外部设备一致。该信号通过寄存器 EXMC_SNCTLx 来设置，

位 NRWTEN 使能，位 NRWTCFG 决定 EXMC_NWAIT 信号是等待状态同时有效，或者比等待状态提前一个时钟周期有效，位 NRWTPOL 设置 EXMC_NWAIT 信号极性。

在 NOR Flash 的同步突发模式中，当寄存器 EXMC_SNCTLx 位 NRWTEN 置 1，在数据延迟之后后检测到 EXMC_NWAIT 信号。如果 EXMC_NWAIT 有效，在 EXMC_NWAIT 无效之前会一直插入等待时钟。

EXMC_NWAIT 有效极性：

NRWTPOL = 1，EXMC_NWAIT 高电平有效

NRWTPOL = 0，EXMC_NWAIT 低电平有效

在同步突发模式中，EXMC_NWAIT 信号有两种配置：

NRWTCFG = 1，EXMC_NWAIT 信号有效时，当前时钟周期数据无效

NRWTCFG = 0，EXMC_NWAIT 信号有效时，下一个时钟周期数据无效，这是复位后的默认配置。

在 EXMC_NWAIT 信号有效的等待周期内，EXMC 会持续的给存储器发送时钟信号，保持片选和输出使能有效，并且忽视总线上的无效数据。

3. CRAM 页边界突发传输的自动分组

CRAM1.5 中禁止突发传输跨越页边界，EXMC 遇到边界会进行传输的自动分组。为了保证正确的突发分组操作，用户需要在寄存器 EXMC_SNCTLx 位 CPS 中需要设定 CRAM 的页大小。

4. 模式 SM – 单次突发传输

对于同步突发传输，如果 AHB 需要的数据为 16 位，则 EXMC 会执行一次长度为 1 的成组传输；如果 AHB 需要的数据为 32 位，则 EXMC 会把这次传输分成 2 次 16 位的传输，即执行一次长度为 2 的突发传输。

对于其他配置，请参考*支持的存储器访问模式*。

同步复用突发读时序 – NOR, PSRAM(CRAM)

图 33-19. 同步复用突发传输读时序

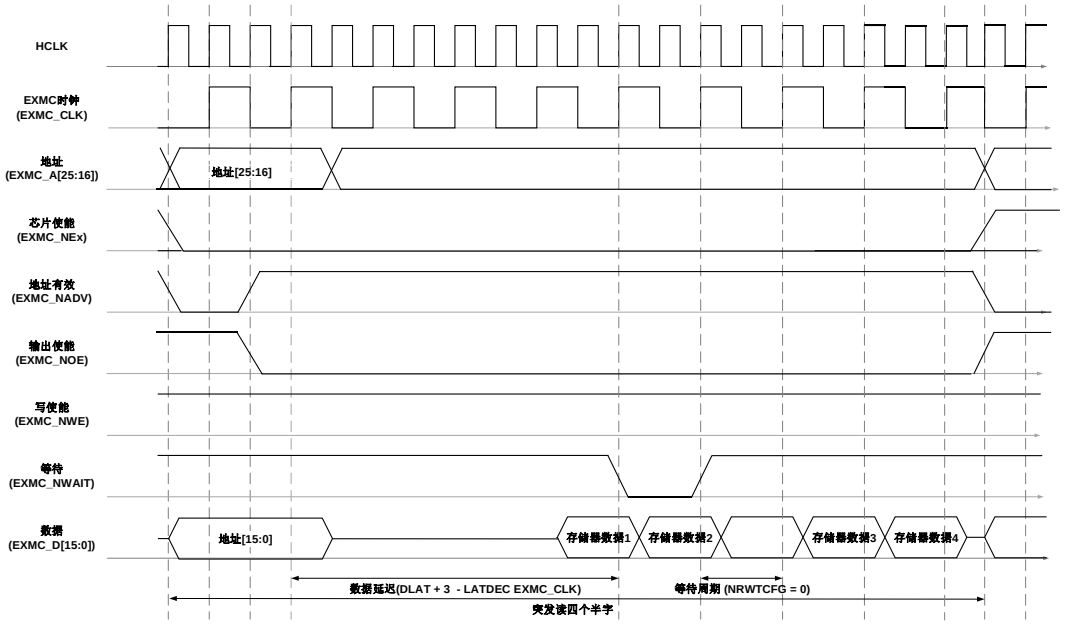


表 33-13. 同步复用模式读时序配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
31-24	保留	0x00
23-22	BLSET	无影响
21	WFIFODIS	无影响
20	CCK	取决于用户
19	SYNCWR	无影响
18-16	保留	0x0
15	ASYNCWTEN	0x0
14	EXMODEN	0x0
13	NRWTEN	取决于存储器
12	WEN	无影响
11	NRWTCFG	取决于存储器
10	WRAPEN	0x0
9	NRWTPOL	取决于存储器
8	SBRSTEN	0x1，突发读使能
7	保留	0x1
6	NREN	取决于存储器
5-4	NRW	0x1
3-2	NRTP	取决于存储器，0x1/0x2
1	NRMUX	0x1，取决于存储器与用户
0	NRBKEN	0x1

位域/位	名称	参考设定值
EXMC_SNTCFGx（读）		
31-30	保留	0x0
29-28	ASYNCMOD	0x0
27-24	DLAT	数据延迟
23-20	CKDIV	*图 33-19. 同步复用突发传输读时序*设置：0x1，EXMC_CLK=2HCLK
19-16	BUSLAT	EXMC_NE[x]上升沿到下降沿的时间
15-8	DSET	无影响
7-4	AHLD	无影响
3-0	ASET	无影响
EXMC_SNLATDECx		
31-3	保留	0x0
2-0	LATDEC	取决于存储器与用户

同步复用突发写时序 – NOR, PSRAM(CRAM)

图 33-20. 同步复用突发传输写时序

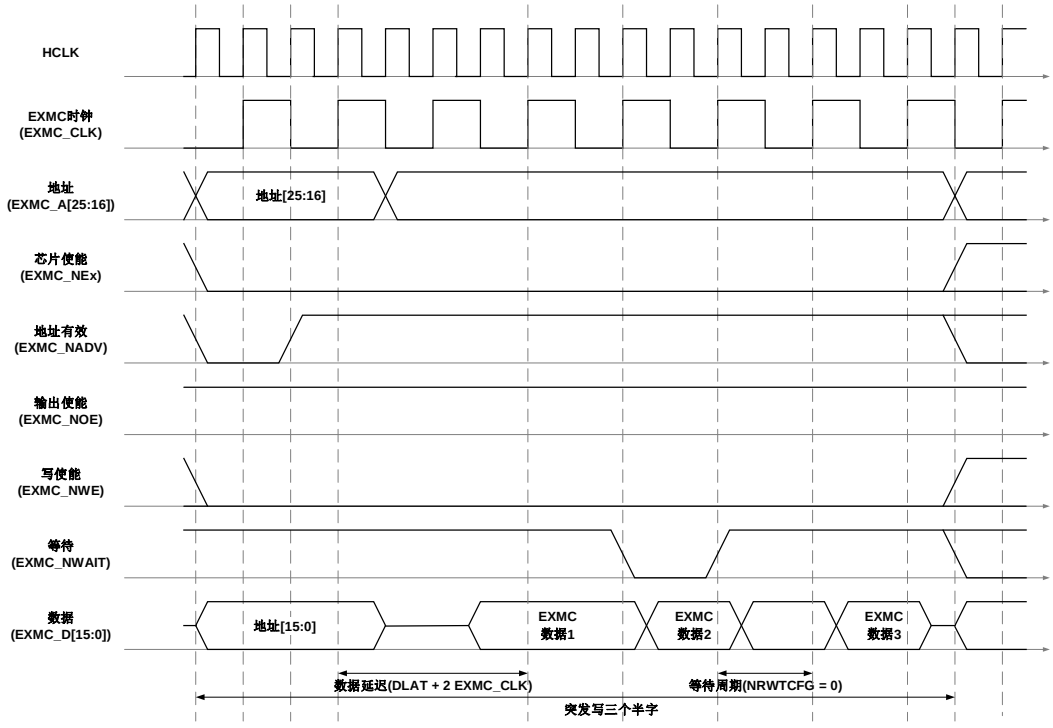


表 33-14. 同步复用模式写时序配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
31-24	保留	0x00

位域/位	名称	参考设定值
23-22	BLSET	无影响
21	WFIFODIS	无影响
20	CCK	取决于用户
19	SYNCWR	0x1，同步写使能
18-16	保留	0x0
15	AYSNCWAIT	0x0
14	EXMODEN	0x0
13	NRWTEN	取决于存储器
12	WEN	0x1
11	NRWTCFG	0x0（这里必须为 0）
10	WRAPEN	0x0
9	NTWTPOL	取决于存储器
8	SBRSTEN	无影响
7	保留	0x1
6	NREN	取决于存储器
5-4	NRW	0x1
3-2	NRTP	0x1
1	NRMUX	0x1，取决于用户
0	NRBKEN	0x1
EXMC_SNTCFGx（写）		
31-30	保留	0x0
29-28	ASYNCMOD	0x0
27-24	DLAT	数据延迟
23-20	CKDIV	*图 33-20. 同步复用突发传输写时序*设置：0x1，EXMC_CLK=2HCLK
19-16	BUSLAT	EXMC_NE[x]上升沿到下降沿的时间
15-8	DSET	无影响
7-4	AHLD	无影响
3-0	ASET	无影响
EXMC_SNLATDECx		
31-3	保留	0x0
2-0	LATDEC	无影响

写 FIFO 支持

写数据 FIFO 用于加速 AHB 对外部存储器的写访问，详细信息请参阅 EXMC_SNCTL 寄存器中的 WFIFODIS 位。

由于 FIFO 的最大深度为 16，建议使用不大于 16 的突发大小。在 FIFO 模式下（默认模式），当

FIFO 不为空时，所有控制寄存器的值都不能改变。

33.4. EXMC 寄存器

EXMC 基地址：0xA000 0000

33.4.1. NOR / PSRAM 控制器寄存器

SRAM / NOR Flash 控制寄存器（EXMC_SNCTLx）（x=0, 1, 2, 3）

偏移地址：0x00 + 8 * x（x = 0, 1, 2, 3）
复位值：0x0000 30DA（Region 0）
复位值：0x0000 30D2（Region 1）
复位值：0x0000 30D2（Region 2）
复位值：0x0000 30D2（Region 3）

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								BLSET[1:0]		WFIFODIS	CCK	SYNCWR	CPS[2:0]		
								rw		rw	rw	rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASYNCWAIT	EXMODEN	NRWTEN	WEN	NRWTCFG	WRAPEN	NRWTPOL	SBRSTEN	保留	NREN	NRW[1:0]		NRTP[1:0]		NRMUX	NRBKEN
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw		rw		rw	rw

位/位域	名称	描述
31:24	保留	必须保持复位值。
23:22	BLSET[1:0]	NBL（字节信号）建立时间 00：NBL 建立时间从 NBL 拉低到片选 NE 拉低为 0 个 AHB 时钟周期 01：NBL 建立时间从 NBL 拉低到片选 NE 拉低为 1 个 AHB 时钟周期 10：NBL 建立时间从 NBL 拉低到片选 NE 拉低为 2 个 AHB 时钟周期 11：NBL 建立时间从 NBL 拉低到片选 NE 拉低为 3 个 AHB 时钟周期
21	WFIFODIS	写 FIFO 禁能 该位控制写 FIFO 功能。 0：使能写 FIFO（复位默认值） 1：禁能写 FIFO **注意：**WFIFODIS位在EXMC_SNCTLx（x = 1..3）没有意义。只能在EXMC_SNCTL0寄存器中可以对其使能。
20	CCK	连续时钟配置

		0：EXMC_CLK只在同步模式产生 1：EXMC_CLK无条件产生 **注意：**该位只在EXMC_SNCTL0有效，EXMC_SNCTLx （x = 1..3）没有意义。当设置此位时，只有EXMC_SNCTL0 CKDIV[3:0]分频因子可以影响EXMC_CLK输出。
19	SYNCWR	选择写操作模式 0：异步写操作 1：同步写操作
18:16	CPS[2:0]	CRAM页大小 000：页边界自动突发分割 001：128字节 010：256字节 011：512字节 100：1024字节 其他：保留
15	ASYNCWTEN	异步等待功能使能位 0：禁用异步等待功能 1：使能异步等待功能
14	EXMODEN	扩展模式使能 0：禁用扩展模式，即不使用EXMC_SNWTCFGx 1：使能扩展模式
13	NRWTEN	NWAIT信号使能 对于存储器的突发模式访问，该位使能/禁用等待状态插入NWAIT信号功能 0：成组传输模式时，禁用NWAIT信号 1：成组传输模式时，使能NWAIT信号
12	WEN	写操作使能 0：禁止EXMC对外部存储器的写操作，否则产生一个AHB错误 1：允许EXMC对外部存储器的写操作（复位缺省值）
11	NRWTCFG	NWAIT信号配置，只在同步模式有效 0：NWAIT信号在等待状态前的一个数据周期有效 1：NWAIT信号在等待状态期间有效
10	WRAPEN	非对齐成组模式使能 0：禁止非对齐成组操作 1：允许非对齐成组操作
9	NRWTPOL	NWAIT信号极性 0：NWAIT低电平有效

		1：NWAIT高电平有效
8	SBRSTEN	同步突发模式使能 0：禁止同步突发模式 1：使能同步突发模式
7	保留	必须保持复位值。
6	NREN	NOR闪存访问使能 0：禁止NOR Flash访问 1：允许NOR Flash访问
5:4	NRW[1:0]	存储器数据宽度 00：8位 01：16位（复位缺省值） 10/11：保留
3:2	NRTP[1:0]	存储器类型 00：SRAM（Region1-Region3的复位缺省值） 01：PSRAM（CRAM） 10：NOR Flash（Region0的复位缺省值） 11：保留
1	NRMUX	数据线/地址线复用 0：禁用地址/数据复用功能 1：允许地址/数据复用功能
0	NRBKEN	存储块使能 0：禁用对应的存储器块 1：使能对应的存储器块

SRAM / NOR Flash 时序寄存器（EXMC_SNTCFGx）（x=0, 1, 2, 3）

偏移地址：0x04 + 8 * x（x = 0, 1, 2, 3）
复位值：0x0FFF FFFF

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		ASYNCMOD[1:0]		DLAT[3:0]				CKDIV[3:0]				BUSLAT[3:0]			
		rw		rw				rw				rw			

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

位/位域	名称	描述
31:30	保留	必须保持复位值。
29:28	ASYNCMOD[1:0]	异步访问模式 该位只有在扩展模式中使用 00：模式A 01：模式B 10：模式C 11：模式D
27:24	DLAT[3:0]	NOR Flash数据延时，仅在同步模式有效 0x0：首次突发访问的数据延迟时间为2个EXMC_CLK时钟周期 0x1：首次突发访问的数据延迟时间为3个EXMC_CLK时钟周期 …… 0xE ~ 0xF：首次突发访问的数据延迟时间为17个EXMC_CLK时钟周期
23:20	CKDIV[3:0]	同步模式时钟分频比，仅在同步模式有效 0x0：保留 0x1：EXMC_CLK周期=2个HCLK周期 …… 0xF：EXMC_CLK周期=16个HCLK周期
19:16	BUSLAT[3:0]	总线延迟时间 在复用读模式中使用，避免总线冲突，是总线恢复到高阻态的最小时间 0x0：总线延迟=0个HCLK周期 0x1：总线延迟=1个HCLK周期 …… 0xF：总线延迟=15个HCLK周期
15:8	DSET[7:0]	异步数据建立时间 该位域仅在异步模式有效 0x00：保留 0x01：数据建立时间=1个HCLK周期 …… 0xFF：数据建立时间=255个HCLK周期
7:4	AHLD[3:0]	异步地址保持时间 该位域设置地址保持时间，仅在模式D与复用模式有效 0x0：保留 0x1：地址保持时间=1个HCLK …… 0xF：地址保持时间=15个HCLK
3:0	ASET[3:0]	异步地址建立时间

该位域设置地址建立时间

注意：该位域仅在SRAM,ROM,NOR Flash的异步模式有效

0x0：地址建立时间= 0个HCLK

……

0xF：地址建立时间= 15个HCLK

SRAM / NOR Flash 写时序寄存器（EXMC_SNWTCFGx）（x=0, 1, 2, 3）

偏移地址：0x104 + 8 * x（x = 0, 1, 2, 3）

复位值：0x0FFF FFFF

该寄存器仅在扩展模式使能（寄存器 EXMC_SNCTL 位 EXMODEN 置 1）后有效。

该寄存器只能按字（32 位）访问。

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

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

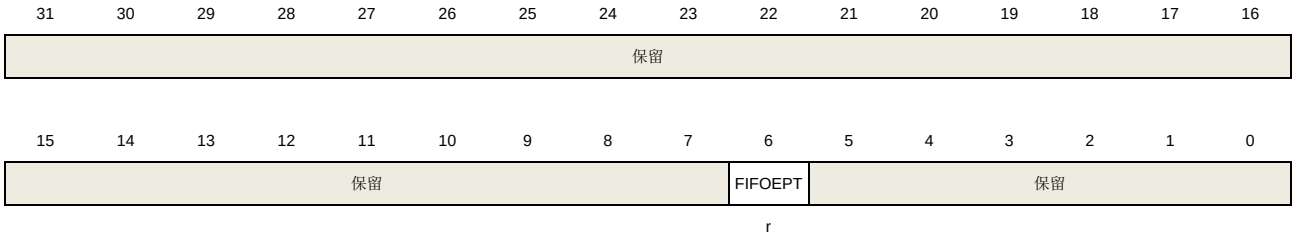
位/位域	名称	描述
31:30	保留	必须保持复位值。
29:28	WASYNCMOD[1:0]	异步访问模式 该位只有在扩展模式中使用 00：模式A 01：模式B 10：模式C 11：模式D
27:20	保留	必须保持复位值。
19:16	WBUSLAT[3:0]	总线延迟时间 在每个写事务结束时添加总线延迟，避免总线冲突，是总线恢复到高阻态的最小时间 0x0：总线延迟=0个HCLK周期 0x1：总线延迟=1个HCLK周期 …… 0xF：总线延迟=15个HCLK周期
15:8	WDSET[7:0]	异步数据建立时间 该位域仅在异步模式有效

		0x00：保留 0x01：数据建立时间=1个HCLK周期 …… 0xFF：数据建立时间=255个HCLK周期
7:4	WAHLD[3:0]	异步地址保持时间 该位域设置地址保持时间，仅在模式D与复用模式有效 0x0：保留 0x1：地址建立时间=1个HCLK …… 0xF：地址建立时间=15个HCLK
3:0	WASET[3:0]	异步地址建立时间 该位域设置地址建立时间 **注意**：该位域仅在SRAM,ROM,NOR Flash的异步模式有效 0x0：地址建立时间= 0个HCLK …… 0xF：地址建立时间= 15个HCLK

SRAM / NOR Flash 状态寄存器（EXMC_SNSTAT）

偏移地址：0x84
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



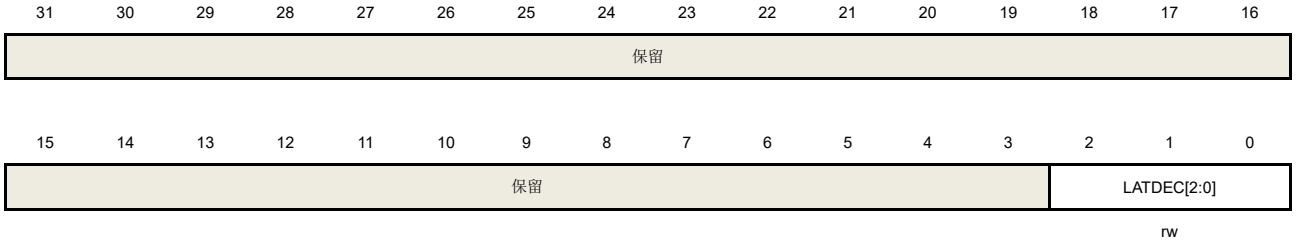
位/位域	名称	描述
31:7	保留	必须保持复位值。
6	FIFOEPT	FIFO空标志 0：FIFO非空 1：FIFO空
5:0	保留	必须保持复位值。

SRAM / NOR Flash 数据延迟减少寄存器（EXMC_SNLATDECx）（x=0, 1, 2, 3）

偏移地址：0x300 + 4 * x（x = 0, 1, 2, 3）
复位值：0x0000 0000

该寄存器仅在同步模式时有效。

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:3	保留	必须保持复位值。
2:0	LATDEC[2:0]	配置NOR Flash的数据延迟的减少值。仅在同步读访问时有效。 该字段与DLAT一起用于调整读访问时间。 同步读： 000：首次突发访问的数据延迟为(DLAT + 3) EXMC_CLK 001：首次突发访问的数据延迟为(DLAT + 2) EXMC_CLK 010：首次突发访问的数据延迟为(DLAT + 1) EXMC_CLK 011：首次突发访问的数据延迟为(DLAT + 0) EXMC_CLK 100：首次突发访问的数据延迟为(DLAT - 1) EXMC_CLK 101：首次突发访问的数据延迟为(DLAT - 2) EXMC_CLK 110：首次突发访问的数据延迟为(DLAT - 3) EXMC_CLK 111：首次突发访问的数据延迟为(DLAT - 4) EXMC_CLK **注意：**例如，如果读模式下的数据延迟需要配置为3个CLK，则DLAT[3:0]应为0b'0000，LATDEC[2:0]应为0b'010。

34. 高性能数字滤波器（HPDF）

34.1. 简介

GD32G553 内部集成了一种专门用于外部 Σ-Δ 调制器的高性能数字滤波器模块（HPDF）。HPDF 支持 SPI 接口和曼彻斯特编码单线接口，通过串行接口可将外部的 Σ-Δ 调制器与 MCU 连接，并对 Σ-Δ 调制器输出的串行数据流进行滤波。此外，HPDF 还支持并行数据流输入功能，实现对内部外设 ADC 或 MCU 内部存储器里的数据进行滤波处理。

34.2. 主要特性

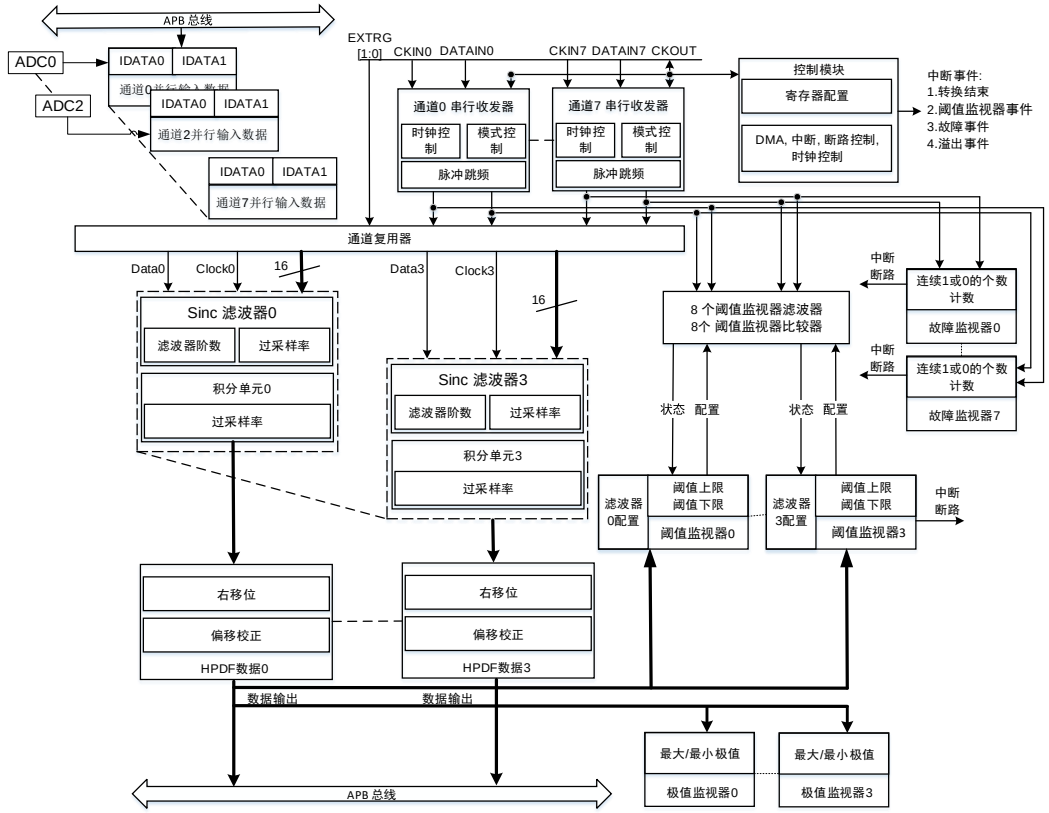
- 8 个复用数字串行输入通道；
  - 可配置的SPI和曼切斯特接口；
- 8 个内部数字并行输入通道；
  - 高达 16 位分辨率的输入；
  - 内部源：ADC 数据或内存（CPU / DMA 写）数据流；
- 可配置的 Sinc 滤波器和积分器；
  - 可配置 Sinc 滤波器的阶数、过采样率（抽取率）；；
  - 可配置积分器的采样率
- 阈值监视功能；
  - 独立的 Sinc 滤波器，可配置阶数和过采样率（抽取率）；
  - 可配置的数据输入源：串行通道输入数据或 HPDF 输出数据；
- 故障监视功能；
  - 拥有 8 位的计数器，用于监视串行通道输入数据流中连续的 0 或 1；
- 极值监视器功能；
  - 存储 HPDF 输出数据的最大值和最小值；
- 高达 24 位的输出数据分辨率；
- 可向外部 Σ-Δ 调制器提供时钟信号；
  - 通过 CKOUT 引脚提供可配置的时钟信号；
- 具有灵活的转换配置功能；
  - 转换通道分为规则组和注入组；
  - 支持多种转换模式和启动模式；
- HPDF 输出数据为有符号格式。

34.3. 功能描述

34.3.1. HPDF 结构框图

HPDF 的结构框图如*图 34-1. HPDF 结构框图*所示。

图 34-1. HPDF 结构框图



HPDF 接口通过*表 34-1. HPDF 引脚定义*中的引脚实现与外部 Σ-Δ 调制器的通信连接。

表 34-1. HPDF 引脚定义

引脚	类型	描述
EXTRG[1:0]	外部触发输入	外部触发信号源输入引脚，触发信号源为 EXTI11 和 EXTI15，作为注入组触发启动信号 HPDF_ITRG[24]和 HPDF_ITRG[25]。
CKOUT	时钟输出	HPDF 模块的时钟输出信号，给外部的 Σ-Δ 调制器提供时钟信号。
CKINx	时钟输入	外部 Σ-Δ 调制器提供给串行接口的时钟信号。
DATAINx	数据输入	外部 Σ-Δ 调制器通过该引脚向串行通道传输

引脚	类型	描述
		1bit 位的数据流。

表 34-2. HPDF 断路连接

断路名称	断路目标
HPDF_BREAK[0]	TIMER0 break0 / TIMER14 break0 / TIMER19 break0
HPDF_BREAK[1]	TIMER0 break1 / TIMER15 break0 / TIMER19 break1
HPDF_BREAK[2]	TIMER7 break0 / TIMER16 break0
HPDF_BREAK[3]	TIMER7 break1

34.3.2. HPDF 开关控制

在正常启动 HPDF 模块时，可通过将 HPDF_CH0CTL 寄存器中的 HPDFEN 置 1，从而全局使能 HPDF 模块。然后再将 HPDF_CHxCTL 中的 CHEN 位和 HPDF_FLTyCTL0 中的 FLTEN 位置 1，可分别使能输入通道和通道数字滤波器。此外只要输入通道使能，输入通道会立即开始接收串行数据。

HPDF 在工作期间，可通过将 FLTEN 清零的方式进入停止模式。进入停止模式之后，HPDF 模块正在进行的转换任务都会立即停止，且寄存器的配置保持不变（除 HPDF_FLTySTAT 和 HPDF_FLTyTMSTAT 寄存器被复位外）。

在停止模式下，HPDF 系统时钟会自动停止。在停止系统时钟，进入停止模式之前，必须清零 HPDFEN 位。

低功耗模式

HPDF 模块对降低功耗进行了优化，在正常的工作模式下，当未有执行的转换任务时，滤波器和积分器会自动进入空闲状态，以实现降低功耗的目的。

34.3.3. HPDF 时钟

HPDF 的时钟包含驱动内部模块的系统时钟和串行接口使用的串行时钟。

系统时钟

HPDF 的系统时钟 $f_{HPDFCLK}$ 用于驱动通道收发器、数字滤波器、积分器、阈值监视器、故障监视器、极值监视器和控制模块。HPDF 系统时钟源可由 RCU 章节 RCU_CFG1 寄存器中的 HPDFSEL 位进行配置。

串行输入时钟

HPDF 的串行接口通过 CKINx 引脚可接收来自外部 Σ-Δ 调制器的时钟信号，以此实现接收 Σ-Δ 调制器发送的串行数据流。

串行接口使用外部输入时钟会受到时钟频率的限制。如果使用标准的 SPI 接口，系统时钟 $f_{HPDFCLK} \geq 4f_{CKIN}$；如果使用曼切斯特编码接口，则需要系统时钟 $f_{HPDFCLK} \geq 6f_{CKIN}$。

串行输出时钟

HPDF 支持输出串行时钟的功能，可通过输出的串行时钟驱动与之相连接的 Σ-Δ 调制器。通过 HPDF_CH0CTL 寄存器中 CKOUTSEL 位可选择串行输出时钟的时钟源。当 CKOUTSEL=0 时，串行输出时钟源为 HPDF 系统时钟；当 CKOUTSEL=1 时，串行输出时钟源为音频时钟，音频时钟的配置可参考 RCU 章节的 RCU_CFG1 寄存器中的 HPDFAUDIOSEL[1:0]位域配置。

串行输出时钟源确定后，可通过配置 HPDF_CH0CTL 寄存器中的 CKOUTDIV[7:0]位域对输出时钟分频控制。当 CKOUTDIV[7:0]≠0 时，串行输出时钟分频器的值为 CKOUTDIV[7:0]+1。当 CKOUTDIV[7:0]=0 时，串行输出时钟被禁止，CKOUT 引脚保持低电平状态。

此外清零 HPDFEN 后，也可实现停止串行输出时钟信号。当串行输出时钟源为系统时钟时（CKOUTSEL=0），清零 HPDFEN，在 4 个系统时钟后停止串行输出时钟。当串行输出时钟源为音频时钟时（CKOUTSEL=1），清零 HPDFEN，在 1 个系统时钟和 3 个音频时钟后停止串行输出时钟。

串行输出时钟源只有在 HPDFEN=0 时，才可修改。为了避免 CKOUT 引脚上产生毛刺信号，软件必须在串行输出时钟停止后修改 HPDF_CH0CTL 寄存器中 CKOUTSEL 位的值。

串行输出时钟的频率的范围为 0-20MHz。

34.3.4. 复用串行数据通道

HPDF 有 8 个复用串行数据通道，支持 SPI 编码和曼切斯特编码。通过配置 HPDF_CHxCTL 寄存器中的 SITYP[1:0]位域选择当前通道支持的接口类型。

SPI 数据接口

在标准的 SPI 接口下，Σ-Δ 调制器通过 DATAINx 引脚向串行通道发送 1bit 的数据流。HPDF 与 Σ-Δ 调制器之间的时钟信号可以由 CKOUT 引脚输出，也可由 CKINx 引脚输入。

SPI 通信时的数据采样点由 HPDF_CHxCTL 寄存器中的 SITYP[1:0]位域和 SPICKSS[1:0]位域共同决定。SPI 通信时的数据采样点如表。

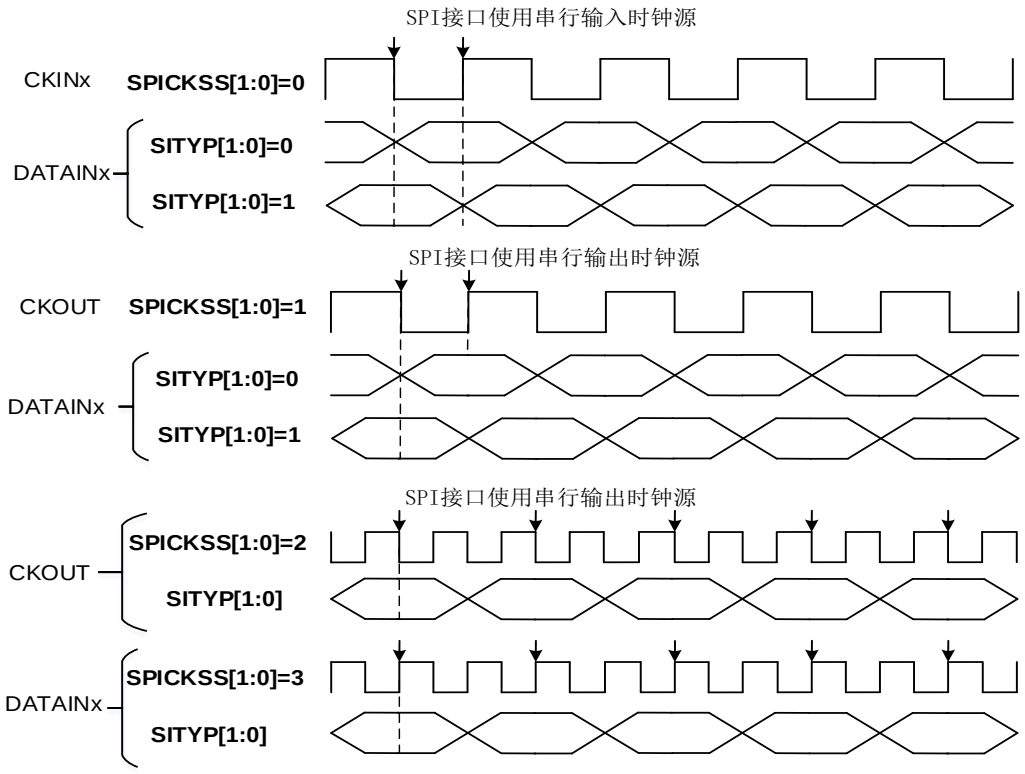
表 34-3. SPI 接口时钟配置

SPICKSS[1:0]	时钟源	SITYP[1:0]	采样点	描述
00	CKINx 信号	00	上升沿	数据在外部串行输入时钟信号的上升沿被采样
		01	下降沿	数据在外部串行输入时钟信号的下降沿被采样
01	CKOUT 信号	00	上升沿	数据在内部串行输出时钟信号的上升

SPICKSS[1:0]	时钟源	SITYP[1:0]	采样点	描述
				沿被采样
		01	下降沿	数据在内部串行输出时钟信号的下降沿被采样
10	CKOUT/2 信号（在 CKOUT 上升沿生成）	xx	每第二个 CKOUT 信号的下降沿	外部 Σ-Δ 调制器将 CKOUT 信号进行 2 分频，来生成串行输入通信时钟。数据在每第二个 CKOUT 下降沿被采样。
11	CKOUT/2 信号（在 CKOUT 下降沿生成）	xx	每第二个 CKOUT 信号的上升沿	外部 Σ-Δ 调制器将 CKOUT 信号进行 2 分频，来生成串行输入通信时钟。数据在每第二个 CKOUT 上升沿被采样。

根据*表 34-3. SPI 接口时钟配置*，SPI 数据传输的时序图如下图所示。

图 34-2. SPI 数据传输时序图



注意：如果采用 SPI 数据接口，时钟源的频率范围为 0-20MHz，且小于 $f_{HPDFCLK}/4$。

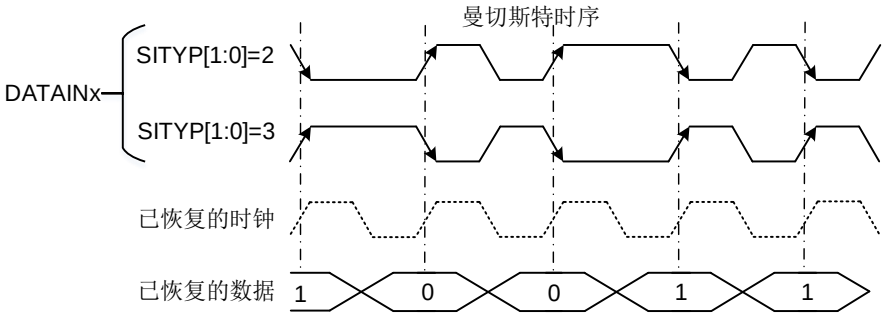
曼切斯特数据接口

HPDF 有 8 个复用串行数据通道可使用曼切斯特编码格式。通过 HPDF_CHxCTL 中的 SITYP[1:0] 位域可配置两种方式的编码格式：

1. 当 SITYP[1:0]=2 时，曼切斯特编码格式：上升沿=逻辑 0，下降沿=逻辑 1。
2. 当 SITYP[1:0]=3 时，曼切斯特编码格式：上升沿=逻辑 1，下降沿=逻辑 0。

采用曼切斯特编码时，外部的 Σ-Δ 调制器与 HPDF 之间只通过 DATAINx 引脚进行数据流的传输。经 HPDF 模块曼切斯特解码后，从串行的数据流中将时钟信号和数据恢复，恢复的时钟信号频率必须在 0-10MHz 之间，且小于 $f_{HPDFCLK}/6$。曼切斯特数据传输的时序图如下图所示。

图 34-3. 曼切斯特数据传输时序图



为了能够正确地接收曼切斯特数据及解码，需要根据预期的曼切斯特数据流速率对 CKOUTDIV[7:0]分频器进行配置。CKOUTDIV[7:0]的值参考以下格式计算：

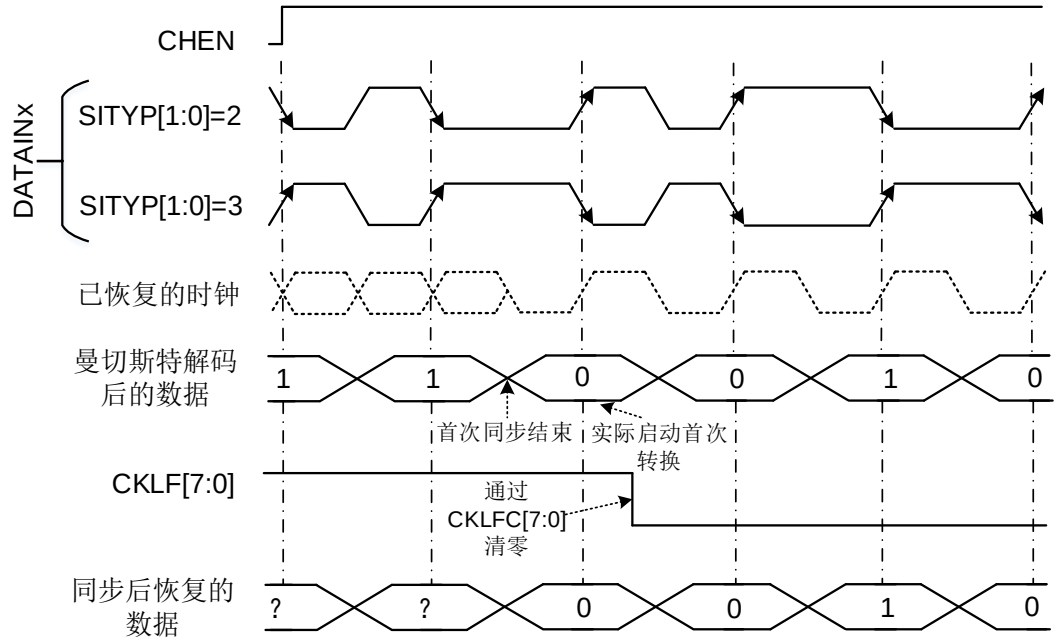
$$((CKOUTDIV+1)xT_{SYSCLK})<T_{Manchester_clock}<(2xCKOUTDIV\times T_{SYSCLK}) \quad (34\text{-}1)$$

串行通信编码同步

串行通道使能之后，必须实现成功同步后才能正确接收数据。对于 SPI 编码的同步发生在 SPI 数据流第一次检测到时钟输入信号之后。如果通道使用曼切斯特编码，首次同步发生在通道接收数据流由 1-0 或 0-1 的变化。

串行通道的收发器在未实现同步之前，通道的时钟丢失标志位被置 1，当成功同步之后，可通过 CKLFC[7:0]将时钟丢失标志位清零。在串行通道的收发器未实现同步时，无法通过 CKLFC[7:0]将时钟丢失标志位清零。因此，可通过软件循环地查询 CKLF[7:0]位的方式来判断串行通道的收发器是否成功同步。下图为曼切斯特编码首次同步的时序图。

图 34-4. 曼切斯特同步时序图



外部串行时钟频率测量

通道串行时钟输入频率的测量提供来自外部 Σ-Δ 调制器的实际数据速率，这对于应用目的很重要。

外部串行时钟输入频率可以通过在一个转换持续时间内计数 HPDF 时钟（$f_{HPDFCLK}$）的定时器来测量。计数从转换触发（常规或注入）后的第一个输入数据时钟开始，到转换结束前的最后一个输入数据时钟结束（设置转换结束标志）。当转换完成（ICEF = 1 或 RCEF = 1）时，每个转换持续时间（第一个串行采样和最后一个串行采样之间的时间）在寄存器 HPDF_FLTxCT 中的计数器 CNVCNT[27:0]中更新。然后，用户可以根据数字滤波器设置（SFO、SFOR、IOR、FAST）计算数据速率。外部串行频率测量只有在滤波器被旁路时才会停止（SFOR=0，只有积分器有效，HPDF_FLTxCT 寄存器中的 CNVCNT[27:0] = 0）。

在并行数据输入的情况下，测得的频率是一次转换期间的平均输入数据速率。

注意：当转换被中断（通过禁用 / 启用所选通道）时，中断时间也计入 CNVCNT[27:0]。因此，建议不要中断转换以获得正确的转换持续时间结果。

转换时间：

注入转换或 FAST = 0 的常规转换（如果 FAST=1，则为第一次转换）：

对于 Sincx 过滤器：

$T = CNVCNT / f_{HPDFCLK} = [SFOR * (IOR-1 + SFO) + SFO] / f_{CKIN}$

对于 FastSinc 滤波器：

T = CNVCNT / $f_{HPDFCLK}$ = [SFOR * (IOR-1 +4) + 2] / f_{CKIN})

FAST = 1 的常规转换（第一次转换除外）：

对于 Sincx 和 FastSinc 滤波器：

T = CNVCNT / $f_{HPDFCLK}$ = [SFOR * IOR-1] / f_{CKIN})

如果 FOSR = FOSR[9:0]+1 = 1（滤波器被旁路，仅激活积分器）：

T = IOR / f_{CKIN}（但 CNVCNT=0）

其中：

- f_{CKIN} 是通道输入时钟频率（在给定通道 CKINx 引脚上）或输入数据速率（在并行数据输入的情况下）
- SFOR 是滤波器过采样率：SFOR = SFOR[9:0]+1（见 HPDF_FLTxSCFFG 寄存器）
- IOR 是积分器过采样率：IOR = IOR[7:0]+1（见 HPDF_FLTxSFCFG 寄存器）
- SFO 是过滤器阶数：SFO = SFO[2:0]（见 HPDF_FLTxSCFFG 寄存器）

时钟丢失检测

时钟丢失检测是指通过检测通道串行输入时钟（CKINx 信号）是否丢失，以确保串行通道转换（或阈值监视器和故障监视器）的数据是否存在错误。如果产生了时钟信号丢失事件，则应丢弃给定的数据。使用时钟丢失检测功能时，必须将 CKOUT 信号源配置为系统时钟。

时钟丢失检测功能可由 HPDF_CHxCTL 中的 CKLEN 位使能或禁止。当使能时钟丢失检测功能和时钟丢失中断 CKLIE，若产生了时钟丢失事件，则通道的时钟丢失标志位会被置 1 并产生时钟丢失中断。可通过将 CKLFC[7:0]位域来清除相应的中断标志位。

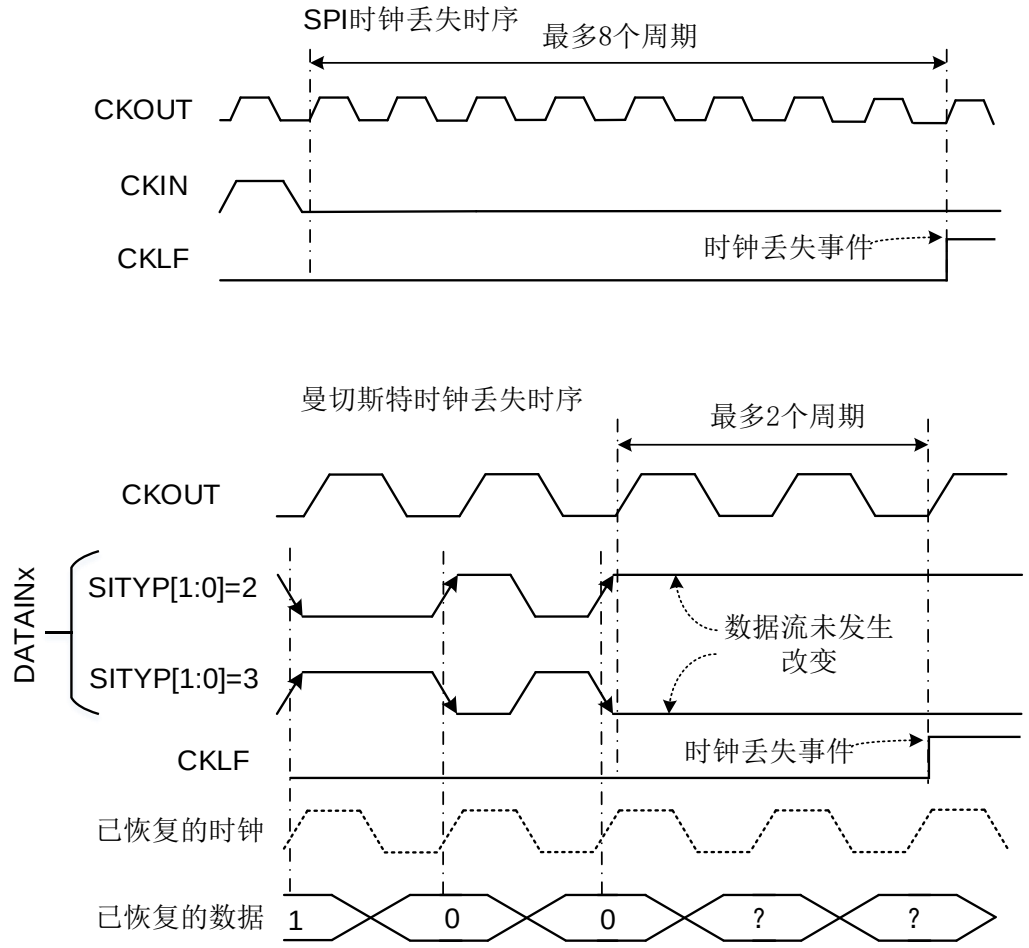
当通道的串行接口的收发器尚未被同步时，通道的时钟丢失标志位被置 1，且无法将相应的时钟丢失标志位清零。所以正确的使用时钟丢失功能的步骤如下：

1. 使能给定通道 CHEN=1。
2. 循环地查询时钟丢失标志位并对给定通道的 CKLFC 写 1，当确认相应的 CKLF 位被清零时，以此判断串行通道的收发器同步成功。
3. 使能时钟丢失检测功能 CKLEN=1，若要检测可能产生的时钟丢失，可使能时钟丢失中断 CKLIE=1。

如果串行通道使用 SPI 接口，使用时钟丢失检测功能时，将外部串行输入时钟（CKINx 信号）与串行输出时钟（CKOUT 信号）进行比较。外部串行输入时钟信号必须在每 8 个 CKOUT 信号周期内至少翻转一次，否则产生时钟丢失事件。

如果串行通道使用曼切斯特接口，时钟丢失检测在曼切斯特编码首次成功同步之后开始，将外部串行输入数据（DATAINx 信号）与串行输出时钟（CKOUT 信号）进行比较。串行输入数据 DATAINx 必须在每 2 个 CKOUT 信号周期内发送变化，否则产生时钟丢失事件。时钟丢失的时序如下图所示。

图 34-5. 时钟丢失检测时序图



注意：曼切斯特编码数据流最大速率必须小于时钟输出 CKOUT 信号。

通道引脚重定向

通道引脚重定向是指串行通道 0 的引脚可以配置为通道 1 的引脚，即通道 0 可从 DATAIN1 和 CKIN1 引脚读取信息。引脚重定向功能适用于采集 PDM 麦克风的音频数据。PDM 麦克风音频信号包含数据和时钟信号，数据分为左/右通道数据，左通道数据在时钟信号的上升沿采集，右通道数据在时钟信号的下降沿采集。

PDM 麦克风数据流输入串行通道时，其配置流程如下：

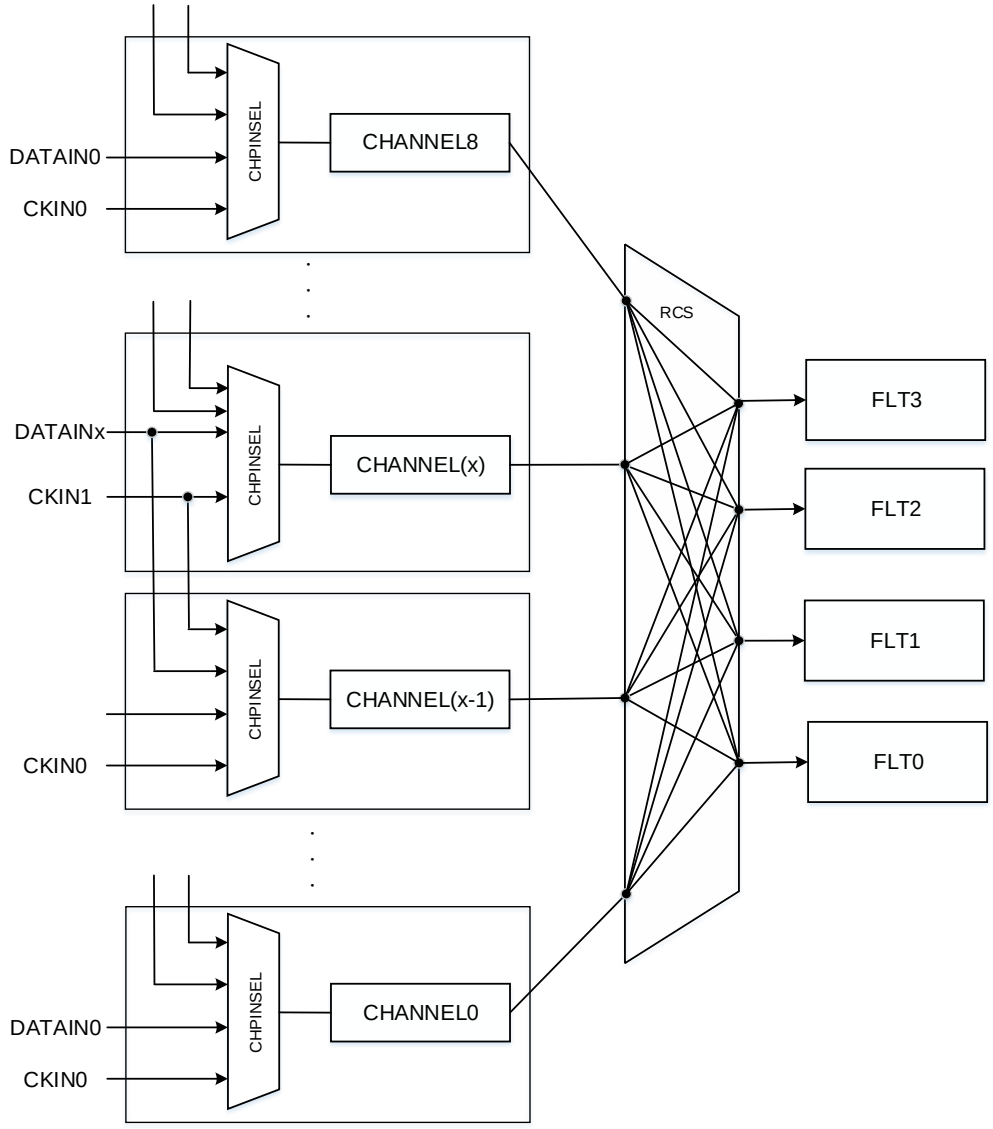
1. 选择 PDM 麦克风数据流输入的 HPDF 串行通道 1。
2. 将通道 1 的 HPDF_CHxCTL 寄存器中的 CHPINSEL 位写 0，通道 1 的输入引脚为自身引脚 DATAIN1 和 CKIN1。将 SITYP[1:0]=0，串行数据流在时钟信号的上升沿被采样，即通道 1 输入的为左通道数据。
3. 将通道 0 的 CHPINSEL 位置 1，通道 0 的输入引脚为引脚 DATAIN1 和 CKIN1。将 SITYP[1:0]=1，

串行数据流在时钟信号的下降沿被采样，即通道 0 输入的为右通道数据。

4. 将通道 0 和通道 1 配置相应的滤波器，对 PDM 麦克风左右通道数据进行滤波处理。

HPDF 模块的通道引脚重定向示意图如*图 34-6. 通道引脚重定向*所示。

图 34-6. 通道引脚重定向



脉冲跳频

脉冲跳频功能是指串行输入数据流在进入滤波器前，跳过指定数量的时钟脉冲后才进入滤波器进行滤波处理，以达到丢弃一定数量的 bit 位的目的。与未跳过的数据流相比，此操作将导致来自滤波器的最终输出样本（和下一个样本）将从后续的输入数据计算得出。

脉冲跳频要跳过的脉冲数由 HPDF_CHxPS 寄存器中的 PLSK[5:0]位域决定。将 PLSK[5:0]位域写

入值，指定通道将开始执行脉冲跳频功能。读取 PLSK[5:0]的值，表示剩余未执行的跳频脉冲数。对 PLSK[5:0]单次写操作时，执行的最大跳频脉冲数为 63 个。可通过多次写入 PLSK[5:0]位域来获得更多数量的跳频脉冲。

串行输入接口配置

HPDF 模块的串行输入接口配置步骤如下：

1. 配置时钟输出预分频器：通过配置 HPDF_CH0CTL 寄存器中的 CKOUTDIV[7:0]位域，预分频的系数为 CKOUTDIV[7:0]+1。
2. 配置串行接口类型和输入时钟相位：通过 HPDF_CHxCTL 寄存器中 SITYP[1:0]位域配置串行接口类型为 SPI 编码或曼切斯特编码，并确定时钟输入采样边沿。
3. 配置输入时钟源：通过配置 HPDF_CHxCTL 寄存器中 SPICKSS[1:0]选择串行接口的时钟源为串行输入时钟或串行输出时钟。
4. 配置数据偏移校正和右移位数：HPDF_CHxCFG0 寄存器中 DTRS[4:0]定义了最终数据右移的位数，数据移位后执行 CALOFF[23:0]位域定义的偏移校正。
5. 使能故障监视和时钟丢失检测功能：通过对 MMEN 和 CKLEN 置 1，使能故障监视和时钟丢失检测功能。
6. 设置阈值监视器的滤波器和故障监视器：阈值监视器的滤波器参数、故障监视器的短路信号分配及计数器阈值都由 HPDF_CHxCFG1 寄存器进行配置。

34.3.5. 并行数据输入

HPDF 模块可通过配置通道复用来选择并行数据作为通道的数据输入源。通过配置 HPDF_CHxCTL 中的 CMSD[1:0]位域来决定通道数据输入源是来自串行数据还是并行数据。每个通道提供了一个 32 位的并行数据输入寄存器 HPDF_CHxPDI，可通过 CPU / DMA 写入两个 16 位并行数据，该寄存器的两个 16 位数据均为有符号格式。

内部 ADC 输入

对于并行 ADC 数据输入（CMSD[1:0]=1），ADC[x]结果被分配至通道 x 输入（ADC1 填充 HPDF_CHxPDI 寄存器）。来自 ADC[x]的转换结束事件会导致更新通道 x 的数据（来自 ADC[x]的并行数据被用作数字滤波器的下一个采样）。转换结束事件发生时，来自 ADC[x]的数据被写入 HPDF_CHxPDI 寄存器（INDAT0[15:0]字段）。

数据封装模式设置（HPDF_CHxCTL 寄存器中的 DPM[1:0]）对 ADC 数据输入无影响。

CPU/DMA 写入并行数据

并行数据的写入方式有 2 种：CPU 直接写入和 DMA 写入方式。在使用 DMA 的方式写入并行数据时，DMA 应配置为存储器到存储器模式，其目标地址为并行数据输入寄存器 HPDF_CHxPDI 的地址。

注意：写入并行数据的 DMA 与读取 HPDF 模块最终转换数据的 DMA 不同。后者需要配置为外设到存储器模式。

并行数据封装模式

存储在 HPDF_CHxPDI 寄存器中的数据会经过通道滤波器进行滤波器处理。存储在 HPDF_CHxPDI 寄存器中的并行数据有 3 种模式。在不同数据封装模式下，允许加载的滤波器采样次数也不同，具体取决于 HPDF_CHxCTL 寄存器中的 DPM[1:0]位域的值。关于不同数据封装模式具体如下：

1. 标准模式（DPM[1:0]= 2'b00）：

此模式下，HPDF_CHxPDI 寄存器中的高 16 位被写保护，CPU/DMA 写入的 16 位数据存储在低 16 位的 DATAIN0[15:0]位域。CPU / DMA 配置为 16 位访问方式，写入一次 16 位数据时，通道滤波器必须执行一次输入采样来清除 HPDF_CHxPDI 寄存器。

2. 交错模式（DPM[1:0]= 2'b01）：

此模式下，CPU / DMA 配置为 32 位访问方式，数据存储在低 16 位的 DATAIN0[15:0]位域和高 16 位的 DATAIN1[15:0]位域。写入一次 32 位数据时，通道滤波器必须执行两次输入采样来清除 HPDF_CHxPDI 寄存器。通道滤波器第一次采样 DATAIN0[15:0]位域，第二次采样 DATAIN1[15:0]位域.

3. 双通道模式（DPM[1:0]= 2'b10）：

此模式下，CPU / DMA 配置为 32 位访问方式，数据存储在低 16 位的 DATAIN0[15:0]位域和高 16 位的 DATAIN1[15:0]位域。其中 DATAIN0[15:0]位域的数据用于当前通道 x，而 DATAIN1[15:0]位域的数据会自动被复制到通道 x+1 的并行数据输入寄存器的低 16 位，并将该数据用于通道 x+1。CPU / DMA 写入一次数据，数字滤波器执行两次采样，第一次执通道 x 的采样，第二次执行通道 x+1 采样。

HPDF 模块中只有偶数通道（通道 0）支持双通道模式，如果将奇数通道（通道 1）配置为双通道模式，则该通道的并行数据输入寄存器 HPDF_CHxPDI 被写保护。如果通道 x 为偶数通道，且被配置为双通道模式，则奇数通道 x+1 必须配置成标准模式。

并行数据输入寄存器 HPDF_CHxPDI 的操作模式如下表所示：

表 34-4. 并行数据封装模式

通道编号	封装模式					
	标准模式		交错模式		双通道模式	
	DATAIN1	DATAIN0	DATAIN1	DATAIN0	DATAIN1	DATAIN0
通道 0	写保护	CH0 采样	CH0 第二次采样	CH0 第一次采样	CH1 采样	CH0 采样
通道 1	写保护	CH1 采样	CH1 第二次采样	CH1 第一次采样	写保护	CH1 采样

通道编号	封装模式					
	标准模式		交错模式		双通道模式	
	DATAIN1	DATAIN0	DATAIN1	DATAIN0	DATAIN1	DATAIN0
通道 2	写保护	CH2 采样	CH2 第二次采样	CH2 第一次采样	CH3 采样	CH2 采样
通道 3	写保护	CH3 采样	CH3 第二次采样	CH3 第一次采样	写保护	CH3 采样
通道 4	写保护	CH4 采样	CH4 第二次采样	CH4 第一次采样	CH5 采样	CH4 采样
通道 5	写保护	CH5 采样	CH5 第二次采样	CH5 第一次采样	写保护	CH5 采样
通道 6	写保护	CH6 采样	CH6 第二次采样	CH6 第一次采样	CH7 采样	CH6 采样
通道 7	写保护	CH7 采样	CH7 第二次采样	CH7 第一次采样	写保护	CH7 采样

CPU / DMA 向 HPDF_CHxPDI 寄存器写操作应当在通道使能之后，因为在使能通道之后，通道转换会开启，在通道转换开启之前会丢弃 HPDF_CHxPDI 寄存器中的数据。

34.3.6. 规则组转换

HPDF 模块有 8 个复用通道，可分别用于规则组转换或注入组转换。如果通道被禁止（CHEN=0），使能通道转换，会导致通道一直处于转换状态。只有通过使能通道（CHEN=1）或禁止 HPDF 模块（HPDFEN=0）才能恢复正常。

规则组只选择 8 个通道中的一个，由 HPDF_FLTyCTL0 寄存器中的 RCS 位决定。在同一时刻内，只能有一个规则转换处于执行或待处理状态。如果已有规则转换请求尚未完成，则会忽略新的规则转换启动请求。规则转换的优先级低于注入组转换，能被注入组转换请求中断。

规则组的转换时间 t=CTCNT[27:0] / $f_{HPDFCLK}$。

转换启动模式

规则组转换只能通过软件的启动的方式实现。软件启动分为 2 种模式，具体方法如下：

1. 常规软件启动：向 HPDF_FLTyCTL0 寄存器中的 SRCS 位写 1。
2. 软件同步启动：将 HPDF_FLTyCTL0 寄存器中同步启动 RCSYN 位置 1，当使用常规软件启动 HPDF_FLT0 的常规转换时，则 HPDF_FLTy 也同步地启动规则转换。

转换模式

规则组转换支持连续模式和快速模式。

连续模式

通过将 HPDF_FLTyCTL0 中的 RCCM 位置 1 使能连续模式。在连续模式下，软件启动规则组转换后，重复执行转换规则组通道转换。清零 RCCM 位后，在连续模式下进行的规则转换会立即停止。

快速模式

通过将 HPDF_FLTyCTL0 中的 FAST 位置 1 使能快速模式。在快速模式，能够提升连续模式下的数据速率。因为在连续模式下，如果从一个通道连续转换，则无需新的数据填充滤波器，因为滤波器内的数据是来自先前连续模式下采样的有效数据。数据速率的提升由所选滤波器阶数决定。

启动连续转换后，在快速模式的首次转换于未开启快速模式的时间相同，然后会以较短的时间间隔完成后续的转换。

34.3.7. 注入组转换

注入组转换通道必须至少选择 8 个通道中的任意一个。可通过 HPDF_FLTyIGCS 寄存器中的 IGCSEL[7:0]位域选择哪个通道为注入组转换，IGCSEL[x]=1 表示通道 x 为注入组通道。

注入组的优先级高于规则组，正在进行中的规则组转换会被注入组转换请求中断，等待注入组完成转换后重启被中断的常规转换。在同一时刻内，只能有一个注入转换处于执行或待处理状态。如果已有注入转换请求尚未完成，则会忽略新的注入转换启动请求。

注入组的转换时间 t=CTCNT[27:0] / $f_{HPDFCLK}$。

启动转换方式

注入组转换可通过软件启动和触发启动的方式实现，具体方法如下：

1. 常规软件启动：向 HPDF_FLTyCTL0 寄存器中的 SICC 位写 1。
2. 软件同步启动：将 HPDF_FLTyCTL0 寄存器中同步启动 ICSYN 位置 1，当使用常规软件启动 HPDF_FLT0 的注入组转换时，则 HPDF_FLT1 也同步地启动注入转换。
3. 触发启动：当 HPDF_FLTyCTL0 寄存器中 ICTSSEL[4:0]位域写入非 0 的值时表示使能触发启动并同时选择了触发信号源。触发的有效边沿则由 ICTEEN[1:0]位域决定。

注入组的触发信号如下表所示：

表 34-5. 注入组的触发信号

触发信号名称	信号源
HPDF_ITRG0	TIMER0_TRGO0
HPDF_ITRG1	TIMER0_TRGO1
HPDF_ITRG2	TIMER7_TRGO0
HPDF_ITRG3	TIMER7_TRGO1
HPDF_ITRG4	TIMER2_TRGO0

触发信号名称	信号源
HPDF_ITRG5	TIMER3_TRGO0
HPDF_ITRG6	TIMER15_CH1
HPDF_ITRG7	TIMER5_TRGO0
HPDF_ITRG8	TIMER6_TRGO0
HPDF_ITRG[9~10]	保留
HPDF_ITRG11	保留
HPDF_ITRG12	保留
HPDF_ITRG[13~23]	保留
HPDF_ITRG24	EXTI11
HPDF_ITRG25	EXTI15
HPDF_ITRG26	-
HPDF_ITRG27	-
HPDF_ITRG28	-
HPDF_ITRG[29~30]	保留
HPDF_ITRG31	HPDF_ITRG

扫描转换模式

通过将 HPDF_FLTyCTL0 寄存器中 SCMOD 位置 1，可使能注入组转换的扫描转换模式。在扫描模式下，每次触发注入组转换时，注入组中的所有通道会从最低通道开始依次转换。

如果禁止扫描模式，则每次触发注入组转换时，只会转换注入组里的一个通道，下一次的触发会选择另一个通道。在禁止扫描模式下，对 IGCSEL[7:0]位域写操作会将最低通道作为选择的转换通道。

转换请求优先级

注入组的转换具有比规则组转换更高的优先级。已在进行的规则转换会被注入转换的请求立即中断。若注入转换序列结束时，如果 RCCM 仍处于置位状态，则连续的规则转换将再次启动。被打断的规则转换重新启动，RCHPDT 位的值表示被打断的规则转换延迟启动。

如果一个注入转换被挂起或已在进行中，则无法启动其他注入转换：只要 ICPF=1，启动注入转换的任何请求（软件或触发启动）都将被忽略。对于规则转换也是如此。

当注入转换正在进行（ICPF=1）时，对 HPDF_FLTyCTL0 的 SRCS 位写 1，请求规则转换。当注入序列完成时，优先级指示下一步执行规则转换，并以 RCHPDT 位表示延迟启动。

34.3.8. 数字滤波器

HPDF 模块的数字滤波器为 $Sinc^X$ 类型。输入的数据流经 $Sinc^X$ 进行滤波，从而降低输出数据速率并提高输出数据分辨率。通过 HPDF_FLTySFCFG 寄存器中的 SFO[2:0]位域和 SFOR[9:0]位域配置 $Sinc^X$ 滤波器的阶数和过采样率（抽取率）。用户可根据所需的分辨率配置 $Sinc^X$ 滤波器的阶数

和过采样率。Sinc^X 滤波的最大输出分辨率与过采样率的关系如下表：

表 34-6. Sinc^X 滤波的最大输出分辨率与过采样滤的关系

SFOR	Sinc	Sinc^2	FastSinc	Sinc^3	Sinc^4	Sinc^5
x	±x	±x^2	±2x^2	±x^3	±x^4	±x^5
4	±4	±16	±32	±64	±256	±1024
8	±8	±64	±64	±512	±4096	±32768
32	±32	±1024	±2048	±32768	±1048576	±33554432
64	±64	±4096	±8192	±262144	±16777216	±1073741824
128	±128	±16384	±32768	±2097152	±268435456	-
256	±256	±65536	±131072	±16777216	在满量程输入的条件下，结果会溢出	
1024	±1024	±1048576	±2097152	±1073741824		

注意：该表中最大输出分辨率来自滤波器输出的峰值数据值。

34.3.9. 积分器

积分器对来自数字滤波器的数据执行进一步的过采样率（抽取率）和分辨率提高。积分器对来自滤波器中给定数量的数据采样执行简单的求和操作。积分器输出数据是来自滤波器的数据采样求和而来，数据采样的数量由积分的过采样率决定。积分器的过采样率（抽取率）可由 HPDF_FLTySFCFG 寄存器中的 IOR[7:0]配置。积分器的最大输出分辨率、过采样率、Sinc 滤波器阶数的关系如下表：

表 34-7. 积分器的最大输出分辨率与 IOR、SFOR、SFO 之间的关系

滤波器类型	积分器最大输出分辨率
Sinc	±(SFOR×IOR)
Sinc^2	±(SFOR^2×IOR)
FastSinc	±(2SFOR^2×IOR)
Sinc^3	±(SFOR^3 ×IOR)
Sinc^4	±(SFOR^4×IOR)
Sinc^5	±(SFOR^5×IOR)

34.3.10. 阈值监视器

HPDF 模块的阈值监视器用于监视通道的串行输入数据或通道转换后最终输出的数据，当数据达到阈值监视器设定的阈值（上限或下限阈值）时，会产生中断或断路事件。高阈值由 HPDF_FLTyTMHT 寄存器中的 HTVAL[23:0]位决定，低阈值由 HPDF_FLTyTMLT 寄存器中的 LTVAL[23:0]位决定。

HPDF 模块的拥有 4 个阈值监视器，通过配置 HPDF_FLTyCTL1 寄存器中的 TMCHEN[7:0]位域

决定阈值监视器 x 是否监视输入通道。如 HPDF_FLT0CTL1 寄存器中的 TMCHEN[1]=1 表示阈值监视器 0 监视通道 1 的阈值。

阈值监视器工作模式

阈值监视器工作模式分为标准模式和快速模式。快速模式是配置阈值监视器监视通道的串行输入数据并与设定的阈值比较。标准模式是配置阈值监视器监视通道转换后输出的最终数据（存储在注入组数据寄存器 HPDF_FLTyIDATA 或规则组数据寄存器 HPDF_FLTyRDATA）。阈值监视器的快速模式可通过 HPDF_FLTyCTL0 中的 TMFM 位使能快速模式。两种模式的下的特性如下表：

表 34-8. 阈值监视器工作模式特点

模式	使能位	通道数据源	阈值监视器输入数据源	输入数据分辨率	详细描述
标准模式	TMFM=0	串行数据流、并行数据	HPDF 最终输出数据	24 位	阈值监视器监视通道转换后输出的最终数据。 响应时间慢，不适用于过流/过压等检测。
快速模式	TMFM=1	串行数据流	串行数据流	16 位	输入数据以连续模式提供，阈值监视器直接监视串行输入数据，与规则或注入转换无关。 响应时间快，适用于过流/过压等检测。

阈值监视器在快速模式下，只使用阈值（上限阈值 HTVAL[23:0]或下限阈值 LTVAL[23:0]）的高 16 位与通道的串行输入数据进行比较，即只用 HTVAL[23:0]和 LTVAL[23:0]的高 16 位定义阈值，这是因为阈值监视器的滤波器分辨率为 16 位。

阈值监视器在非快速模式下，完成右移位和偏移校正的最终数据会与 HTVAL[23:0]和 LTVAL[23:0]进行比较。

阈值监视器快速模式

在快速模式下，将使用阈值监视器自身的滤波器，在 HPDF_CHxCFG1 寄存器中可设置阈值监视器滤波器的过采样率（抽取率）和阶数。

阈值监视器的配置较为灵活，可通过 HPDF_FLTyCTL1 寄存器中的 TMCHEN[7:0]位域将一个阈值监视器可以配置监视多个通道。在此情况下，当多个通道发出请求时，阈值监视器优先处理通道编号小的请求，然后再处理通道编号大的请求。每个阈值监视器均有一个状态寄存器 HPDF_FLTyTMSTAT，当所监视的通道发生超出阈值错误事件时，在 HTF[7:0]或 LTF[7:0]位域中对应的标志会被置位。如 HTF[0]=2b'01，表示通道 0 发生超出上限阈值错误。

每个通道发出比较请求后，会在 8 个 HPDF 时钟周期内被执行。因此，每个通道的带宽被限制为

8 个 HPDF 时钟周期（如果 TMCHEN[7:0]=3）。由于输入通道最大采样频率为 $f_{HPDFCLK}/4$，因此在该输入时钟速度下，阈值监视器滤波器不能被旁路（TMFOR=0）。因此，用户必须根据输入采样时钟速度和 $f_{HPDFCLK}$ 正确配置阈值监视器滤波器参数和所监视的通道数。

在快速模式下，读取 HPDF_CHxTMFDT 寄存器中 TMDATA[15:0]位域可获得定通道 x 的阈值监视器滤波器数据。阈值监视器滤波器输出（在串行输入时钟频率 f_{CKIN}）一个结果所需的串行样本数如下：

1. 首次转换：

FastSinc 滤波器：采样数=(TMFOR[4:0]×4+ 2+1)。

$Sinc^X$ 滤波器（X=1...5）：采样数=(((TMFOR[4:0]+1)×TMSFO[1:0])+ TMSFO[1:0]+1)。

2. 除首次转换外的后续转换：

FastSinc 和 $Sinc^X$ 滤波器（X=1...5）：采样数=(TMFOR[4:0]+1)×(IOR[7:0]+1)。

阈值监视器状态标志

阈值监视器的全局状态为 HPDF_FLTySTAT 寄存器中的 TMEOF 标志位，当 TMEOF=1 时，表示至少产生了一个阈值监视器事件，即有超出（上限/下限）阈值的事件产生。如果使能 HPDF_FLTyCTL1 中的阈值监视器事件中断 TMIE=1，可产生看阈值监视器中断。当所有 HTF[7:0]和 LTF[7:0]都被清除时，TMEOF 位被清除。

HPDF_FLTyTMSTAT 寄存器中定义了通道发生超出阈值的错误事件标志，其中，HTF[7:0]位域表示通道 x 上是否发生超出上限阈值 HTVAL[23:0]值。LTF[7:0]位域表示通道 x 上是否发生超出下限阈值 LTVAL[23:0]值。通过将“1”写入 HPDF_FLTyTMFC 寄存器中相应的 HTFC[7:0]或 LTFC[7:0]位来清除超出阈值标志。

如*表 34-2. HPDF 断路连接*所示，HPDF 模块中有 4 个断路输出信号，通过配置 HPDF_FLTyTMHT 寄存器和 HPDF_FLTyTMLT 寄存器中的 HTBSD[3:0]和 LTBSD[3:0]位域将断路输出信号分配给阈值监视器超出阈值事件。

34.3.11. 故障监视器

故障监视器用于检测当前模拟信号的状态是否处于短路或开路故障（例如过电流/电压）。若故障监视器检测到上述两种状态之一时，能够以极快的响应时间产生断路事件，并输出断路信号。断路输出信号可以分配给故障监视器事件，可通过配置由 HPDF_CHxCFG1 寄存器中的 MMBSD[3:0]位域实现。断路输出信号与阈值监视器相同。

故障监视器的输入数据来自通道的串行输入数据，当通道输入数据源为并行数据时，禁止使用故障监视功能。在每个输入通道上都有一个递增计数器，用于记录在串行数据流有多少个连续的 0 或 1。当计数器达到故障阈值寄存器值（HPDF_CHxCFG1 寄存器中的 MMCT[7:0]位），则产生短路或开路故障事件。若监测数据流时遇到 0-1 或 1-0 的变化，则计数器的值会被自动清零并重新计

数。

用户可以通过设置 HPDF_CHxCTL 寄存器中的 MMEN 位来使能故障监视功能。当通道产生短路或开路故障事件时，相应的故障监视标志置位 MMF[7:0]被置位。可通过 HPDF_FLTyINTC 中的 MMFC[7:0]清除相应的标志，若通道 x 被禁用（CHEN=0），硬件也会清除故障监视标志。

34.3.12. 极值监视器

极值监视器被用于采集最终输出数据字的最小值和最大值（峰值到峰值）。通过配置 HPDF_FLTyCTL1 寄存器中的 EMCS[7:0]位域，可使一个极值监视器采集多个通道的极值。

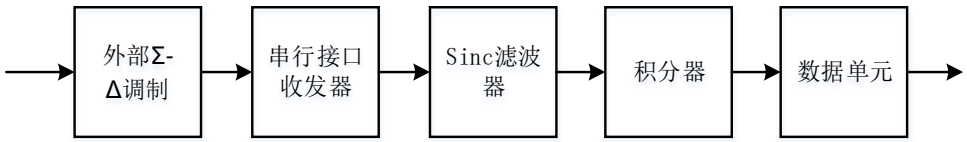
如果采集的最终输出数据字高于在极值监视器最大值寄存器中的值（HPDF_FLTyEMMAX 寄存器中的 MAXVAL[23:0]位），则用该寄存器的值被更新为当前的最终输出数据。如果采集的最终输出数据字小于在极值监视器最小值寄存器中的值（HPDF_FLTyEMMIN 寄存器中的 MINVAL[23:0]位），则该寄存器的值被更新为当前的最终输出数据。MAXDC 位和 MINDC 位的值分别指明了最大值/最小值来自哪个通道。

当读取 HPDF_FLTyEMMAX 或 HPDF_FLTyEMMIN 寄存器时，最大值或最小值被更新为复位值。

34.3.13. 数据单元

数据单元是整个 HPDF 模块中处理数据的最后一个部分，HPDF 模块处理数据的流程如下图所示。

图 34-7. HPDF 模块外部输入数据处理流程



输出数据速率取决于串行数据流速率、滤波器和积分器设置。最大输出数据速率如下表所示。

表 34-9. 最大输出速率

输入源	转换模式	滤波器类型	最大输出数据速率（采样/秒）
串行输入	非快速模式（FAST=0）	$Sinc^X$	$\frac{f_{CKIN}}{SFOR\times(IOR-1+SFO)+(SFO+1)}$
	非快速模式（FAST=0）	FastSinc	$\frac{f_{CKIN}}{SFOR\times(IOR-1+4)+(2+1)}$
	快速模式（FAST=1）	FastSinc 和 $Sinc^X$	$\frac{f_{CKIN}}{SFOR\times IOR}$
并行输入	非快速模式（FAST=0）	$Sinc^X$	$\frac{f_{DATA}}{SFOR\times(IOR-1+SFO)+(SFO+1)}$
	非快速模式（FAST=0）	FastSinc	$\frac{f_{DATA}}{SFOR\times(IOR-1+4)+(2+1)}$

输入源	转换模式	滤波器类型	最大输出数据速率（采样/秒）
	速模式（FAST=1）	FastSinc 和 $Sinc^X$	$\frac{f_{DATA}}{SFOR \times IOR}$

注意：表中 f_{DATA} 为 CPU / DMA 输入的并行数据速率，当滤波器被旁路时，必须满足 $f_{DATA} \leqslant f_{HPDFCLK}$。

有符号的数据格式

HPDF 模块中的有符号数据：并行数据寄存器、规则和注入组数据寄存器、阈值监视器阈值、极限监视器极值、偏移校正均为有符号格式。输出数据的最高有效位表示值的符号，数据采用二进制的补码格式。

由于数字处理中的所有操作都在 32 位有符号寄存器上执行，因此必须满足以下条件才能使结果不溢出：

1. 当使用 $Sinc^X$ 滤波器（x=1…5）时：$(SFOR^{SFO}) \times IOR \leq 2^{31}$。
2. 当使用 FastSinc 滤波器时：$2 \times (SFOR^2) \times IOR \leq 2^{31}$。

数据右位移

由于 HPDF 输出数据的最高分辨率为 24 位，并且来自处理路径的数据可以高达 32 位，因此在该模块中执行最终数据的右位移位。对于每个选定的输入通道，可在 HPDF_CHxCFG0 寄存器中的 DTRS[4:0]位域配置右移的位数，右移位是丢弃最低位的数，取近似值。

数据偏移校正

HPDF 模块中，每个通道都有一个数据偏移校正值，该值存储在 HPDF_CHxCFG0 寄存器的 CALOFF[23:0]位域。在进行偏移校正时，通道的输出数据中减去偏移校正值，以得到 HPDF 模块输出的最终数据。

数据偏移校正发生在数据右位移之后。

34.3.14. HPDF 中断

HPDF 的中断事件可分为通道转换中断事件、阈值监视器中断事件、故障监视器中断事件和通道时钟丢失中断事件。具体的中断事件描述如*表 34-10. HPDF 中断事件*所示。

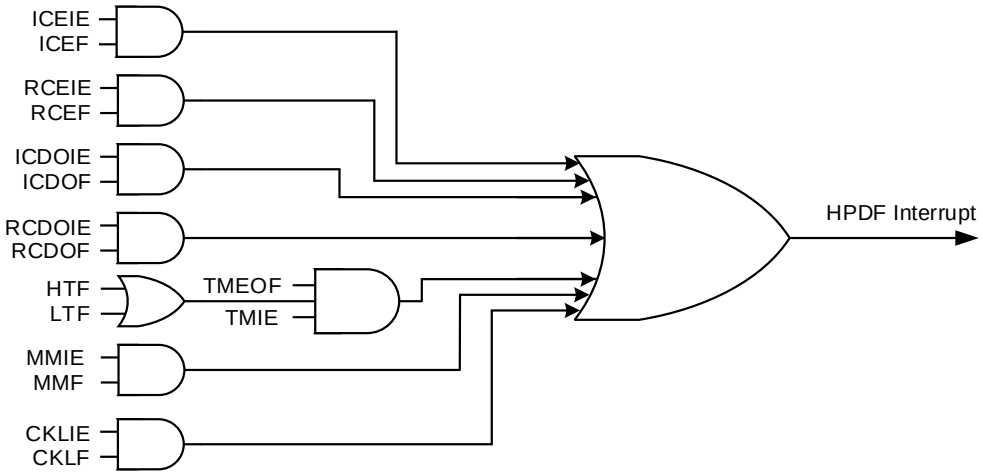
表 34-10. HPDF 中断事件

中断事件	描述	清除方式	中断使能位
ICEF	注入转换结束	读 HPDF_FLTyIDATA 寄存器	ICEIE
RCEF	规则转换结束	读 HPDF_FLTyRDATA 寄存器	RCEIE
ICDOF	注入转换数据溢出	写 1 到 ICDOFC 位	ICDOIE
RCDOF	规则转换数据溢出	写 1 到 RCDOFC 位	RCDOIE
TMEOF	阈值监视器事件	写 1 到 HTFC[7:0]位域	TMIE

中断事件	描述	清除方式	中断使能位
HTF[7:0] LTF[7:0]		写 1 到 LTFC[7:0]位域	
MMF	通道发生故障事件	写 1 到 MMFC[7:0]位	MMIE
CKLF	通道时钟丢失	写 1 到 CKLFC[7:0]位	CKLIE

HPDF 中断逻辑如*图 34-8. HPDF 中断逻辑图*所示。

图 34-8. HPDF 中断逻辑图



34.4. HPDF 寄存器

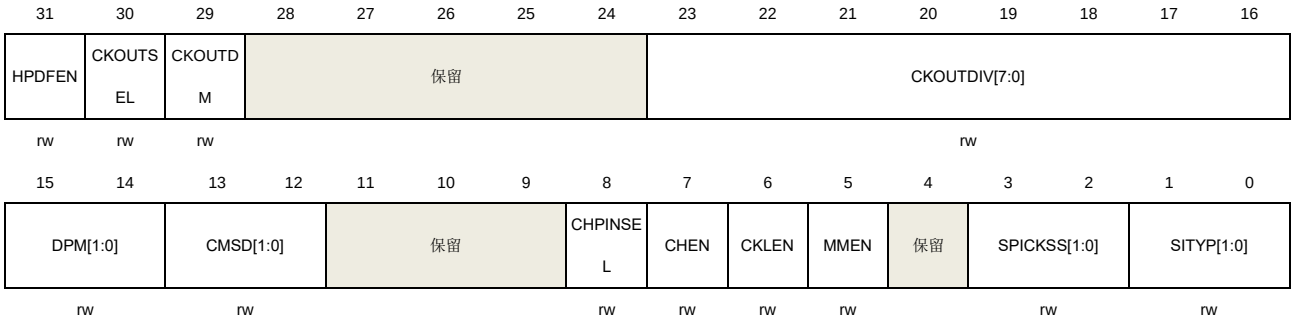
HPDF 访问基地址：0x4001 7000

34.4.1. HPDF 通道 x 寄存器（x = 0...7）

通道 x 控制寄存器（HPDF_CHxCTL）

地址偏移：0x00 + 0x20 * x，（x = 0...7）
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31	HPDFEN	HPDF 全局使能 0：禁止 HPDF 1：使能 HPDF 如果 HPDFEN=0，则复位 HPDF_FLTySTAT 寄存器和 HPDF_FLTyTMSTAT 寄存器。 此位仅在 HPDF_CH0CTL 寄存器中有效。
30	CKOUTSEL	串行输出时钟源选择 0：CK_HPDF 时钟作为串行输出时钟源 1：CK_HPDFAUDIO 时钟作为串行输出时钟源 此位仅在 HPDF_CH0CTL 寄存器中有效。
29	CKOUTDM	串行时钟输出占空比模式 0：禁止串行时钟输出占空比模式 1：使能串行时钟输出占空比模式，占空比为 1:1。 当 HPDFEN=0，此位才能修改。 此位仅在 HPDF_CH0CTL 寄存器中有效。
28:24	保留	必须保持复位值。

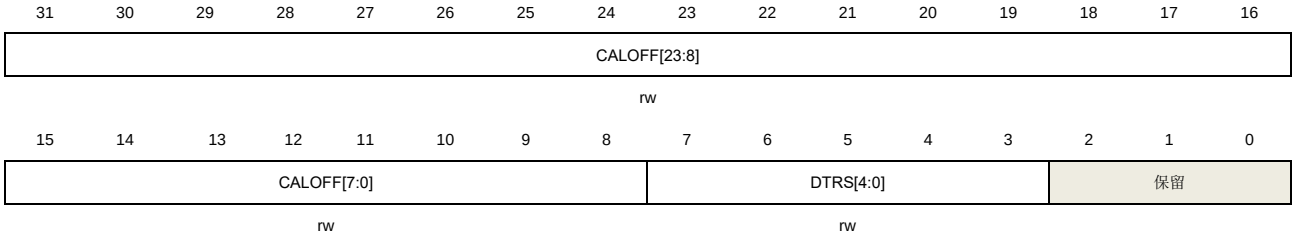
23:16	CKOUTDIV[7:0]	串行输出时钟预分频器 0：禁止串行输出时钟 1~255：串行输出时钟的分频系数为 CKOUTDIV+1 CKOUTDIV 还定义了时钟丢失检测的阈值。 当 HPDF 被禁止时 HPDENF=0，才能修改 CKOUTDIV 的值。HPDF 禁止后，在 1 个 HPDF 时钟内，输出时钟信号（CKOUT）变为低电平状态。 此位仅在 HPDF_CH0CTL 寄存器中有效。
15:14	DPM[1:0]	并行输入数据封装模式 00：标准模式 01：交错模式 10：双通道模式 11：保留 数据封装模式的详细介绍请参考*并行数据封装模式* 只有当 CHEN=0 时，此位域才能被修改。
13:12	CMSD[1:0]	选择复用通道 x 输入数据源 00：串行输入作为复用通道 x 输入数据源 01：内部模数转换器 ADC_x 作为复用通道 x 输入数据源 10：HPDF_CHxPDI 寄存器中内部数据作为复用通道 x 输入数据源 11：保留 当此位域的值为零时，HPDF_CHxPDI 寄存器被写保护。 只有当 CHEN=0 时，此位域才能修改。
11:9	保留	必须保持复位值。
8	CHPINSEL	通道输入引脚选择 0：选择当前通道 x 的引脚作为通道输入引脚 1：选择下个通道 x+1 的引脚作为通道输入引脚 只有当 CHEN=0 时，此位才能被修改。
7	CHEN	通道 x 使能 0：禁止通道 x 1：使能通道 x 如果通道 x 使能，该通道会根据已有的配置开始接收串行数据。
6	CKLEN	时钟丢失检测使能 0：禁止时钟丢失检测 1：使能时钟丢失检测
5	MMEN	故障监视器使能 0：禁止故障监视器 1：使能故障监视器

4	保留	必须保持复位值。
3:2	SPICKSS[1:0]	SPI 接口时钟源选择 00：选择外部输入时钟（CKINx）作为 SPI 时钟源-由 SITYP[1:0]位域决定采样点 01：选择内部输输出时钟（CKOUT）作为 SPI 时钟源-由 SITYP[1:0]位域决定采样点 10：选择内部输出时钟（CKOUT）作为 SPI 时钟源-采样点在 CKOUT 信号每第二个下降沿 11：选择内部输出时钟（CKOUT）作为 SPI 时钟源-采样点在 CKOUT 信号每第二个上升沿 只有当 CHEN=0 时，此位域才能被修改。
1:0	SITYP[1:0]	串行接口类型 00：SPI 接口，上升沿采样数据。 01：SPI 接口，下降沿采样数据。 10：曼切斯特编码接口：上升沿=逻辑 0，下降沿=逻辑 1。 11：曼切斯特编码接口：上升沿=逻辑 1，下降沿=逻辑 0。 只有当 CHEN=0 时，此位域才能修改。

通道 x 配置寄存器 0（HPDF_CHxCFG0）

地址偏移：0x04 + 0x20 * x，（x = 0…7）
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:8	CALOFF[23:0]	24 位偏移校正 通道的每一次转换数据后必须执行偏移校正。 此位域值由软件写入
7:3	DTRS[4:0]	数据右移位数 0~31：该值表示数据执行右移的位数 数据移位在偏移校正之间执行，数据移位将结果四舍五入到最接近的整数值，并保留符号位。 只有当 HPDF_CHxCTL 寄存器中的 CHEN=0 时，此位域才能被修改。

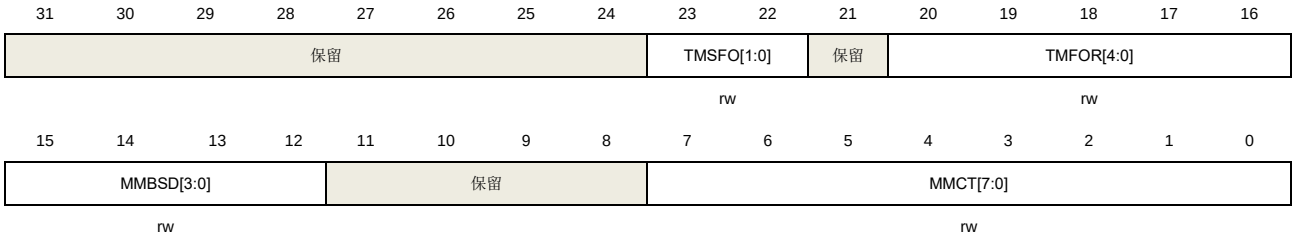
2:0	保留	必须保持复位值。

通道 x 配置寄存器 1（HPDF_CHxCFG1）

地址偏移：0x08 + 0x20 * x，（x = 0…7）

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:24	保留	必须保持复位值。
23:22	TMSFO[1:0]	阈值监视器 Sinc 滤波器阶数 00：FastSinc 滤波器 01：Sinc^1 滤波器 10：Sinc^2 滤波器 11：Sinc^3 滤波器 只有当 HPDF_CHxCTL 寄存器中的 CHEN=0 时，此位域才能被修改。
21	保留	必须保持复位值。
20:16	TMFOR[4:0]	阈值监视器 Sinc 滤波器过采样率（抽取率） 0~31：滤波器的抽取率为 TMFOR[4:0] + 1 如果 TMFOR=0，则滤波器被旁路 只有当 HPDF_CHxCTL 寄存器中的 CHEN=0 时，此位域才能被修改。
15:12	MMBSD[3:0]	故障监视器断路信号分配 MMBSD[i] = 0：断路信号 BREAK[i]未分配至通道 x 故障监视器。 MMBSD[i] = 1：断路信号 BREAK[i]已分配至通道 x 故障监视器。
11:8	保留	必须保持复位值。
7:0	MMCT[7:0]	故障监视器阈值 此位域的值表示故障监视器的计数器阈值，该值由软件写入。 如果故障监视器的计数器值达到阈值，则通道上产生故障监视事件。

通道 x 阈值监视器滤波器数据寄存器（HPDF_CHxTMFDT）

地址偏移：0x0C + 0x20 * x，（x = 0…7）

复位值：0x0000 0000

该寄存器只能按字（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
TMDATA[15:0]															
r															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	TMDATA[15:0]	阈值监视器数据 此数据来自阈值监视器的滤波器，该通道进行连续的数据转换。

通道 x 并行数据输入寄存器（HPDF_CHxPDI）

地址偏移：0x10 + 0x20 * x，（x = 0…7）

复位值：0x0000 0000

该寄存器只能按半字（16 位）或字（32 位）访问。

HPDF 模块的滤波器将对该寄存包含的 16 位数据进行滤波处理。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATAIN1[15:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAIN0[15:0]															
rw															

位/位域	名称	描述
31:16	DATAIN1[15:0]	通道 x 或通道 x+1 的输入数据 通过 CPU/DMA 的方式写入数据 如果 DPM[1:0]=0（标准模式），DATAIN1[15:0]被写保护。 如果 DPM[1:0]=1（交错模式），通道 x 的第二采样数据被保存到 DATAIN1[15:0]。通道 x 的第一个采样被保存到 DATAIN0[15:0]。HPDF_FTLx 滤波器依次读取两个采样。 如果 DPM[1:0]=2（双通道模式）： 通道 0：DATAIN1[15:0]中保存的采样数据被自动复制到通道 1 的 DATAIN0[15:0]位

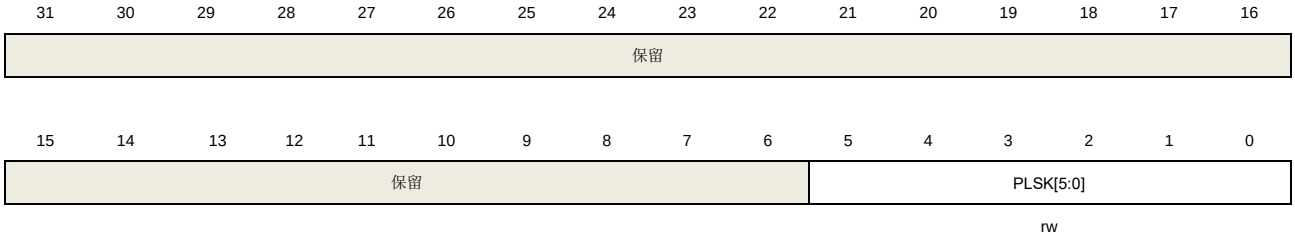
域中。
通道 1：DATAIN1[15:0]被写保护。
并行数据的详细操作模式请参考*并行数据封装模式*。
DATAIN1[15:0]采用 16 位有符号格式。

15:0	DATAIN0[15:0]	通道 x 的输入数据 通过 CPU/DMA 的方式写入数据 如果 DPM[1:0]=0（标准模式），通道 x 的数据采样保存在 DATAIN0[15:0]位域。 如果 DPM[1:0]=1（交错模式），通道 x 的第二采样数据被保存到 DATAIN1[15:0]。 通道 x 的第一个采样被保存到 DATAIN0[15:0]。HPDF_FTLx 滤波器依次读取两个采样。 如果 DPM[1:0]=2（双通道模式）： 通道 0：DATAIN0[15:0]位域的数据用于当前通道 x。 通道 1：DATAIN0[15:0]被写保护。 并行数据的详细操作模式请参考*并行数据封装模式*。 DATAIN0[15:0]采用 16 位有符号格式。

通道 x 跳频寄存器（HPDF_CHxPS）

地址偏移：0x14 + 0x20 * x，（x = 0…7）
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:6	保留	必须保持复位值。
5:0	PLSK[5:0]	输入数据跳频功能 0~63：该值表示将要跳过的串行输入采样 当此位域被写入非零的值时，跳频功能会立即执行。读取该位域，返回当前跳频剩余未执行的值。 当 PLSK[5:0]不为零时，仍可更新其值。

34.4.2. HPDF 滤波器 y 寄存器（y = 0...3）

滤波器 y 控制寄存器 0（HPDF_FLTyCTL0）

地址偏移：0x100 + 0x80 * y，（y = 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	TMFM	FAST	保留		RCS[2:0]			保留		RCDMAEN	保留	RCSYN	RCCM	SRCS	保留
	rw	rw			rw					rw		rw	rw	rt_w	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	ICTEEN[1:0]		ICTSSEL[4:0]					保留		ICDMAEN	SCMOD	ICSYN	保留	SICC	FLTEN
	rw		rw							rw	rw	rw		rt_w	rw

位/位域	名称	描述
31	保留	必须保持复位值。
30	TMFM	阈值监视器快速模式 0：阈值监视器监视的数据为最终数据，最终数据为执行过偏移校正和右移位后的数据。 1：阈值监视器监视的数据为通道串行输入数据
29	FAST	规则转换的快速转换模式 0：禁止快速转换模式 1：使能快速转换模式 如果使能快速转换模式，在规则转换的连续模式下，每次转换速度快于标志转换速度，但首次转换除外。该位对非连续转换没有影响。 只有当 FLTEN=0 时，此位才能被修改。
28:27	保留	必须保持复位值。
26:24	RCS[2:0]	选择规则转换通道 0：通道 0 作为规则转换通道 1：通道 1 作为规则转换通道 … 7：通道 7 作为规则转换通道 当 RCPF=1 时，写此位，被选中的通道在下一个规则转换开始转换。
23:22	保留	必须保持复位值。

21	RCDMAEN	使能读取规则转换数据的 DMA 0：禁止 DMA 读取规则转换数据 1：使能 DMA 读取规则转换数据 只有当 FLTEN=0 时，此位才能被修改。
20	保留	必须保持复位值。
19	RCSYN	同步启动规则转换 0：禁止同步启动规则转换 1：使用同步启动规则转换 如果 HPDF_FLT0CTL0 寄存器中的 RCSYN=1，其他 HPDF_FLTyCTL0 中 RCSYN=1 的规则通道同步启动。 只有当 FLTEN=0 时，此位才能被修改。
18	RCCM	规则转换连续模式 0：每个规则转换请求，只执行一次规则通道转换 1：每个规则转换请求，重复执行规则通道转换 在规则转换的连续模式下，清零此位，连续模式立即停止。
17	SRCS	软件启动规则转换 0：无影响 1：产生一个启动规则转换请求 如果 RCPF=1，对此位的写操操作是无效的。如果 RCSYN=1，此位置 1 将启动同步规则转换。 读此位，得到的值始终为零
16:15	保留	必须保持复位值。
14:13	ICTEEN[1:0]	注入转换触发边沿使能 00：禁止触发检测 01：触发信号的每个上升沿产生启动注入转换请求 10：触发信号的每个下降沿产生启动注入转换请求 11：触发信号的每个边沿（上升沿和下降沿）产生启动注入转换请求 只有当 FLTEN=0 时，此位域才能被修改。
12:8	ICTSSEL[4:0]	注入转换触发信号选择 0x00~0x1F：该值表示选择不同的触发信号开始转换 产生一个触发信号到同步启动触发的最大延迟为 1 个 $f_{HPDFCLK}$ 时钟周期，异步触发延时为 2-3 个 $f_{HPDFCLK}$ 时钟周期。 只有当 FLTEN=0 时，此位域才能被修改。
7:6	保留	必须保持复位值。
5	ICDMAEN	使能读取注入转换数据的 DMA 0：禁止 DMA 读取注入转换数据

		1：使能 DMA 读取注入转换数据 只有当 FLTEN=0 时，此位才能被修改。
4	SCMOD	注入转换扫描转换模式 0：对注入组通道执行一次转换，然后选中注入组的下一个通道 1：选择注入组最小编号通道开始，对注入组通道依次执行连续转换 如果 SCMOD=0，对 IGCSEL 位写操作将会导致通道选择复位为注入组中的最小通道。 只有当 FLTEN=0 时，此位才能被修改。
3	ICSYN	同步启动注入转换 0：禁止启动与 HPDF_FLT0CTL0 同步注入的转换 1：在 HPDF_FLT0CTL0 中 SICC 触发时，在 HPDF_FLTy 中同步启动注入转换。 只有当 FLTEN=0 时，此位才能被修改。
2	保留	必须保持复位值。
1	SICC	启动注入组转换 0：没有影响 1：产生一个注入组转换请求 如果 ICPF=1，对此位写操作是无效的。如果 ICSYN=1，此位置 1，将启动同步注入组转换。 读此位，得到的值始终为零
0	FLTEN	滤波器 y 使能 0：禁止滤波器 y 1：使能滤波器 y 如果滤波器 y 使能，滤波器 y 根据配置立即开始工作。 如果滤波器 y 禁止，滤波器 y 所有的转换和功能都立即停止，同时 HPDF_FLTySTAT 和 HPDF_FLTyTMSTAT 寄存器都被复位。

滤波器 y 控制寄存器 1（HPDF_FLTyCTL1）

地址偏移：0x104 + 0x80 * y，（y = 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								TMCHEN[7:0]							
											rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EMCS[7:0]								保留	CKLIE	MMIE	TMIE	RCDOIE	ICDOIE	RCEIE	ICEIE
rw									rw	rw	rw	rw	rw	rw	rw

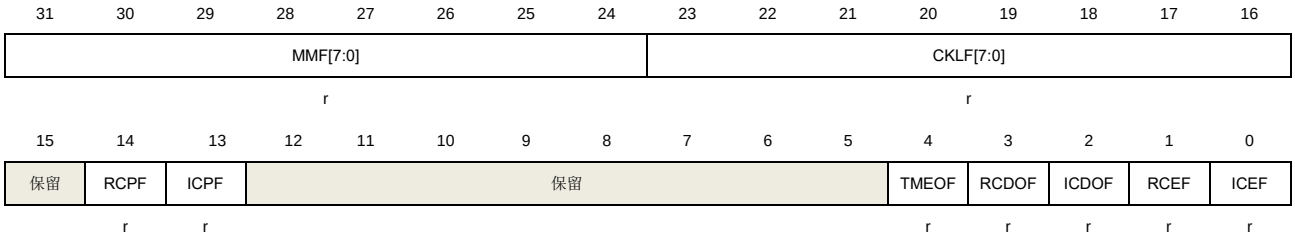
位/位域	名称	描述
31:24	保留	必须保持复位值。
23:16	TMCHEN[7:0]	阈值监视器通道使能 此位域决定阈值监视器持续监视的通道 TMCHEN[x] = 0：使能阈值监视器 y 监视通道 x TMCHEN[x] = 1：禁止阈值监视器 y 监视通道 x
15:8	EMCS[7:0]	极值监视器通道选择 此位域决定极值监视器要采样的通道 EMCS[x] = 0：极值监视器 y 不监视通道 x 的数据 EMCS[x] = 1：极值监视器 y 监视通道 x 的数据
7	保留	必须保持复位值。
6	CKLIE	时钟丢失中断使能 0：禁止时钟丢失中断 1：使能时钟丢失中断 此位仅在 HPDF_FLT0CTL1 寄存器中有效。
5	MMIE	故障监视器中断使能 0：禁止故障监视器中断 1：使能故障监视器中断 此位仅在 HPDF_FLT0CTL1 寄存器中有效。
4	TMIE	阈值监视器中断使能 0：禁止阈值监视器中断 1：使能阈值监视器中断
3	RCDOIE	规则转换数据溢出中断使能 0：禁止规则转换数据溢出中断 1：使能规则转换数据溢出中断
2	ICDOIE	注入转换数据溢出中断使能 0：禁止注入转换数据溢出中断 1：使能注入转换数据溢出中断
1	RCEIE	规则转换结束中断使能 0：禁止规则转换结束中断 1：使能规则转换结束中断
0	ICEIE	注入转换结束中断使能 0：禁止注入转换结束中断 1：使能注入转换结束中断

滤波器 y 状态寄存器（HPDF_FLTySTAT）

地址偏移：0x108 + 0x80 * y，（y = 0…3）
复位值：0x0003 0000

该寄存器只能按字（32 位）访问。

当 FTLEN=0 时，HPDF_FLTySTAT 寄存器被复位。



位/位域	名称	描述
31:24	MMF[7:0]	故障监视器标志 MMF[x]=0：通道 x 没有产生故障事件 MMF[x]=1：通道 x 产生故障事件 此位域由硬件置位，可通过 HPDF_FLTyINTC 中的 MMFC[7:0]位域清零。 通过禁止通道 CHEN=0，该位域由硬件清零。 此位域仅在 HPDF_FTL0STAT 寄存器中有效。
23:16	CKLF[7:0]	时钟丢失标志 CKLF[x]=0：通道 x 的时钟未丢失 CKLF[x]=1：通道 x 的时钟丢失 当 CHEN=0 时或串行接口尚未同步时，由硬件保持置位状态。串行接口同步完成后，若通道 x 的时钟丢失，CKLF[7:0]位域中相应的位由硬件置位。通过置位 HPDF_FLTyINTC 中的 CKLFC[7:0]位域，可清除 CKLF[7:0]位域中相应的位。 此位仅在 HPDF_FTL0STAT 寄存器中有效。
15	保留	必须保持复位值。
14	RCPF	规则转换正在进行标志 0：没有规则转换请求产生 1：规则转换正在进行或一个规则转换请求被挂起 如果 RCPF=1，将忽略启动规则转换的请求。当向 SRCS 位写 1，RCPF 被立即置位。
13	ICPF	注入转换正在进行标志 0：没有注入转换请求产生（软件或触发方式均未有） 1：规注入转换正在进行或一个注入转换请求被挂起 如果 ICPF=1，将忽略启动注入转换的请求。当向 SICC 位写 1，ICPF 被立即置位。

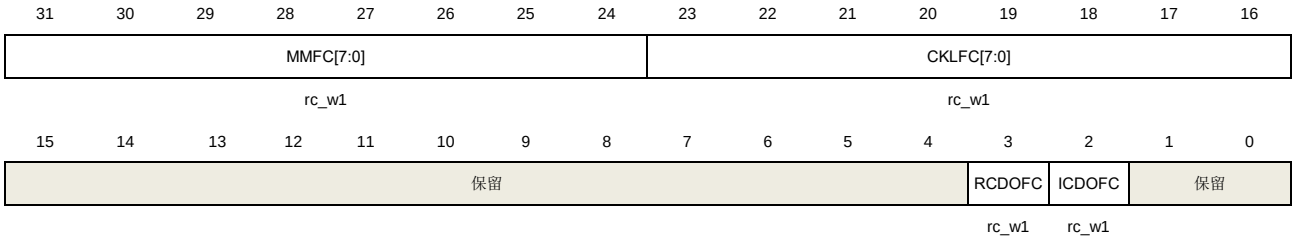
12:5	保留	必须保持复位值。
4	TMEOF	阈值监视器事件产生标志 0：没有阈值监视器事件产生 1：当检测数据超过阈值，阈值监视器产生阈值监视器事件 此位由硬件置位，通过清零 HPDF_FLTyTMSTAT 寄存器中的 HTF[7:0]和 LTF[7:0]位域将此位清零。
3	RCDOF	规则转换数据溢出标志 0：没有规则转换数据溢出产生 1：产生规则转换数据溢出 如果此位置位，表示规则转换已经完成，RCEF 也已经置位，FLTyRDATA 不受溢出影响。 此位由硬件置位，通过置位 HPDF_FLTyINTC 中的 RCDOFC 位，可清除此位。
2	ICDOF	注入转换溢出标志 0：没有注入转换数据溢出产生 1：产生注入转换数据溢出 如果此位置位，表示规则转换已经完成，ICEF 也已经置位，FLTyIDATA 不受溢出影响。 此位由硬件置位，通过置位 HPDF_FLTyINTC 中的 ICDOFC 位，可清除此位。.
1	RCEF	规则转结束标志 0：未完成规则转换 1：完成规则转换 如果 RCEF=1，表示转换数据可以被读取 此位由硬件置 1，当通过软件或 DMA 方式读 HPDF_FLTyRDATA 寄存器时，此位被清零。
0	ICEF	注入转结束标志 0：未完成注入转换 1：完成注入转换 如果 ICEF=1，表示转换数据可以被读取 此位由硬件置 1，当通过软件或 DMA 方式读 HPDF_FLTyIDATA 寄存器时，此位被清零。

滤波器 y 中断标志清除寄存器（HPDF_FLTyINTC）

地址偏移：0x10C + 0x80 * y，（y = 0…3）
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

注：读 HPDF_FLTyINTC 寄存器中的位，得到值始终为 0。



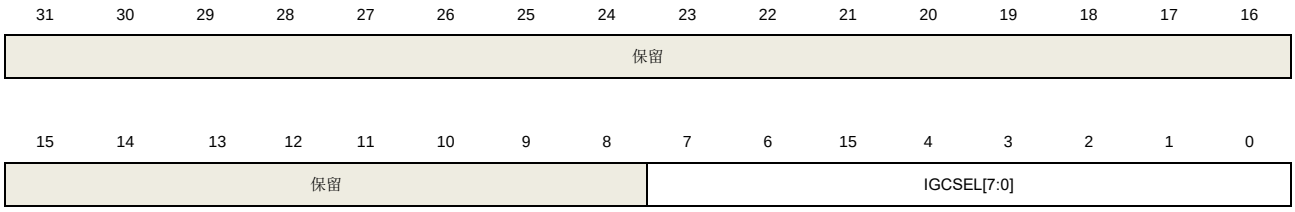
位/位域	名称	描述
31:24	MMFC[7:0]	清除故障监视器标志 MMFC[x]=0：没有影响 MMFC[x]=1：清除通道 x 的故障监视器标志 此位仅在 HPDF_FLT0INTC 寄存器中有效（滤波器 y=0）
23:16	CKLFC[7:0]	清除时钟丢失标志 CKLFC[x]=0：没有影响 CKLFC[x]=1：清除通道 x 的时钟丢失标志 当串行接口尚未完成时钟同步，时钟丢失标志被置位，此时不能通过 CKLFC[7:0]清除时钟丢失标志。 此位仅在 HPDF_FLT0INTC 寄存器中有效（滤波器 y=0）
15:4	保留	必须保持复位值。
3	RCDOFC	清除规则转换数据溢出标志 0：没有影响 1：清除规则转换数据溢出标志 RCDOF
2	ICDOFC	清除注入转换数据溢出标志 0：没有影响 1：清除注入转换数据溢出标志 ICDOF
1:0	保留	必须保持复位值。

滤波器 y 注入组通道选择寄存器（HPDF_FLTyIGCS）

地址偏移：0x110 + 0x80 * y，（y = 0…3）

复位值：0x0000 0001

该寄存器只能按字（32 位）访问。



rw

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	IGCSEL[7:0]	注入组通道选择 ICGSEL[x]=0：通道 x 不属于注入组 ICGSEL[x]=1：通道 x 属于注入组 如果 SCMOD=1，由最小编号的通道开始，依次转换每一个所选通道。 如果 SCMOD=0，只转换其中一个所选通道，然后选择下一个通道。 当 SCMOD=0 时，对 IGCSEL[7:0]写操作将通道选择复位为最小编号通道。 注入组中必须至少有 1 个通道，所有将 IGCSEL[7:0]变为 0 的写操作都被忽略。

滤波器 y SINC 滤波器配置寄存器（HPDF_FLTySFCFG）

地址偏移：0x114 + 0x80 * y，（y = 0…3）

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

位/位域	名称	描述
31:29	SFO[2:0]	滤波器阶数 000：FastSinc 滤波器 001：Sinc^1 滤波器 010：Sinc^2 滤波器 011：Sinc^3 滤波器 100：Sinc^4 滤波器 101：Sinc^5 滤波器 110~111：保留 只有当 HPDF_FLTyCTL0 寄存器中的 FLTEN=0 时，此位域才能被修改。
28:26	保留	必须保持复位值。
25:16	SFOR[9:0]	Sinc 滤波器过采样率（抽取率） 0~1023：Sinc 滤波器过采样率 SFOR= SFOR[9:0] +1 如果 SFOR[9:0]=0，即过采样率为 SFOR=1，表示滤波器被旁路

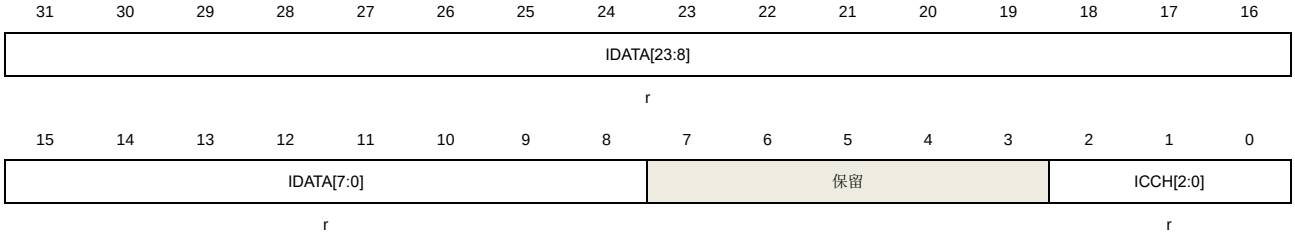
		只有当 HPDF_FLTyCTL0 寄存器中的 FLTEN=0 时，此位域才能被修改。
15:8	保留	必须保持复位值。
7:0	IOR[7:0]	积分器过采样率 0~255：积分器过采样率 IOR=IOR[7:0]+1 积分器的数据输出速率将减去该值 如果 IOR[7:0]=0，即过采样率为 IOR=1，表示积分器被旁路。 只有当 HPDF_FLTyCTL0 寄存器中的 FLTEN=0 时，此位域才能被修改。

滤波器 y 注入组转换数据寄存器（HPDF_FLTyIDATA）

地址偏移：0x118 + 0x80 * y，（y = 0…3）
复位值：0x0000 0000

该寄存器只能按半字（16 位）或字（32 位）访问。

注意：可使用半字访问只读取转换数据的高 16 位有效数据，读该寄存器可清除 ICEF 位。



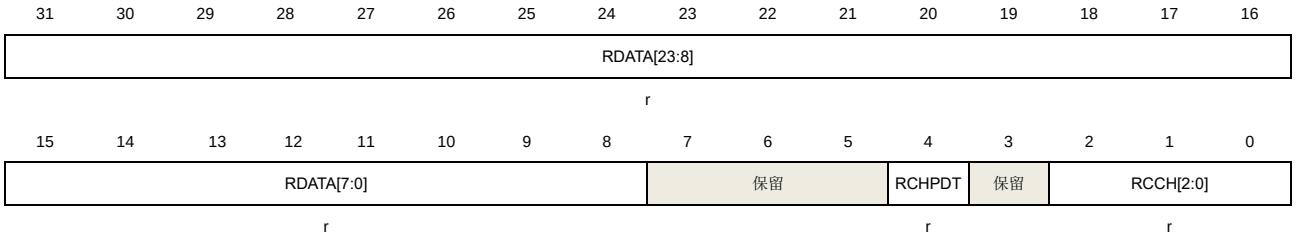
位/位域	名称	描述
31:8	IDATA[23:0]	注入通道转换数据 当每个注入中的一个通道转换完成，数据被保存在此位域。 当 ICEF=1 时，转换数据为有效的。读此寄存器清除 RCEF 位。
7:3	保留	必须保持复位值。
2:0	ICCH[2:0]	最近转换的注入通道 每个注入组通道转换完成时，ICCH[2:0]被更新，指示哪个通道完成了规则转换。故 IDATA[23:0]中的数据对应为该通道的值。

滤波器 y 规则通道转换数据寄存器（HPDF_FLTyRDATA）

地址偏移：0x11C + 0x80 * y，（y = 0…3）
复位值：0x0000 0000

该寄存器只能按半字（16 位）或字（32 位）访问。

注意：可使用半字访问只读取转换数据的高 16 位有效数据，读该寄存器可清除 RCEF 位。



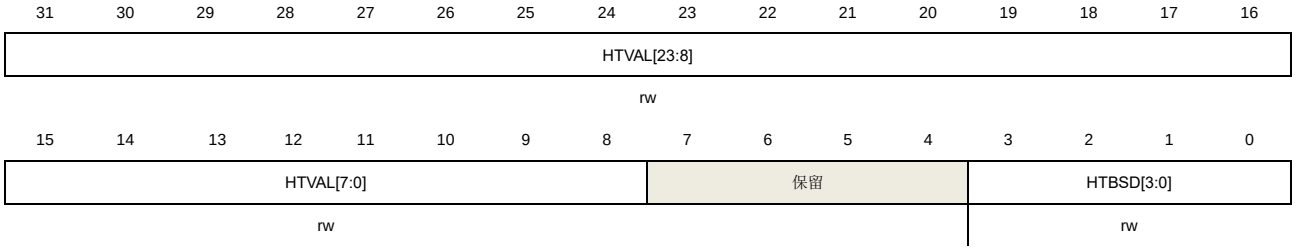
位/位域	名称	描述
31:8	RDATA[23:0]	规则通道转换数据 当每个规则转换完成，数据被保存在此位域。当 RCEF=1 时，转换数据为有效的。读此寄存器清除 RCEF 位。
7:5	保留	必须保持复位值。
4	RCHPDT	规则通道等待处理数据 在规则转换期间，被注入转换请求中断，导致 RDATA[23:0]中的规则转换数据被延迟处理。
3	保留	必须保持复位值。
2:0	RCCH[2:0]	最近转换的规则通道 每个规则转换完成时，RCCH[2:0]被更新，指示哪个通道完成了规则转换。故 RDATA[23:0]中的数据对应为该通道的值。

滤波器 y 阈值监视器上限阈值寄存器（HPDF_FLTyTMHT）

地址偏移：0x120 + 0x80 * y，（y = 0…3）

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:8	HTVAL[23:0]	阈值监视器上限阈值 此位域通过软写入阈值监视器的上限阈值 在阈值监视器快速模式下（TMFM=1），此位域的高 16 位定义上限阈值，并与阈值监视器数据寄存器中的 TMDATA[15:0]值比较。此时 HTVAL[7:0]被忽略。

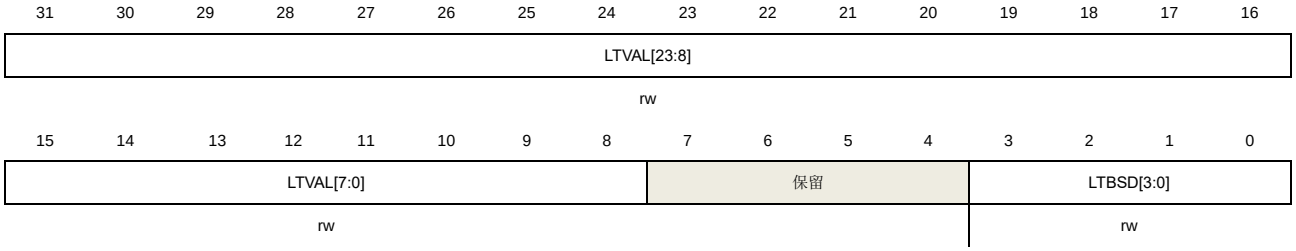
7:4	保留	必须保持复位值。
3:0	HTBSD[3:0]	上限阈值事件断路信号分配 HTBSD[i] = 0：断路信号 x 未分配到阈值监视器上限阈值事件。 HTBSD[i] = 1：断路信号 x 分配到阈值监视器上限阈值事件。

滤波器 y 阈值监视器下限阈值寄存器（HPDF_FLTyTMLT）

地址偏移：0x124 + 0x80 * y，（y = 0…3）

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



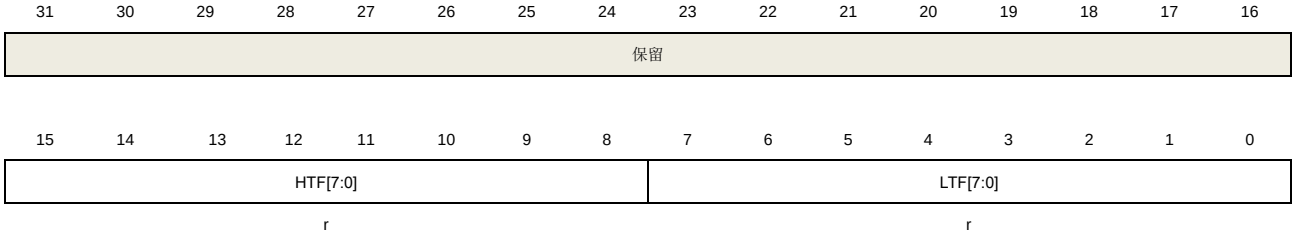
位/位域	名称	描述
31:8	LTVAL[23:0]	阈值监视器下限阈值 此位域通过软写入阈值监视器的下限阈值 在阈值监视器快速模式下（TMFM=1），此位域的高 16 位定义下限阈值，并与阈值监视器数据寄存器中的 TMDATA[15:0]值比较。此时 LTVAL[7:0]被忽略。
7:4	保留	必须保持复位值。
3:0	LTBSD[3:0]	下限阈值事件断路信号分配 LTBSD[i] = 0：断路信号 i 未分配到阈值监视器下限阈值事件 LTBSD[i] = 1：断路信号 i 分配到阈值监视器下限阈值事件

滤波器 y 阈值监视器状态寄存器（HPDF_FLTyTMSTAT）

地址偏移：0x128 + 0x80 * y，（y = 0…3）

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

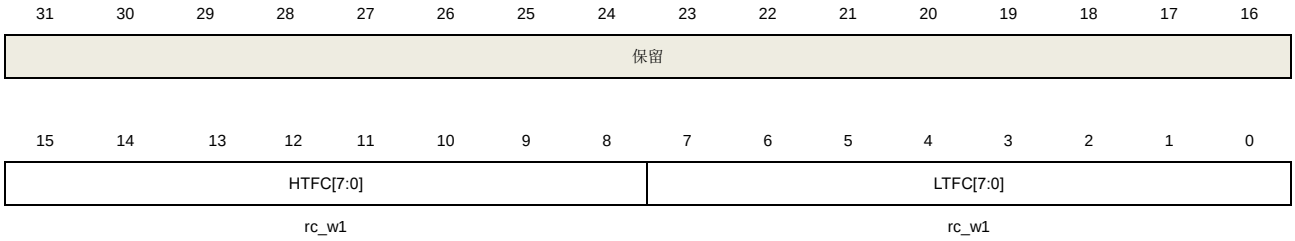


位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	HTF[7:0]	阈值监视器上限阈值标志 HTF[x]=0：通道 x 未超出上限阈值 HTF[x]=1：通道 x 超出上限阈值 此位域由硬件置位，可通过置位 HPDF_FLTyTMFC 寄存器 HTFC[7:0]位域中相应的位，清除对应的阈值监视器上限阈值标志。
7:0	LTF[7:0]	阈值监视器下限阈值标志 LTF[x]=0：通道 x 未超出下限阈值 LTF[x]=1：通道 x 超出下限阈值 此位域由硬件置位，可通过置位 HPDF_FLTyTMFC 寄存器 LTFC[7:0]位域中相应的位，清除对应的阈值监视器下限阈值标志。

滤波器 y 阈值监视器标志清除寄存器（HPDF_FLTyTMFC）

地址偏移：0x12C + 0x80 * y，（y = 0…3）
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



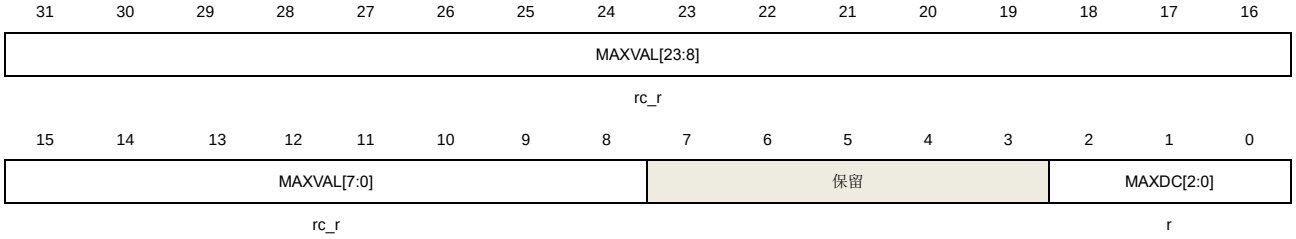
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	HTFC[7:0]	清除阈值监视器上限阈值标志 HTFC[x]=0：没有影响 HTFC[x]=1：清除通道 x 的阈值监视器上限阈值标志
7:0	LTFC[7:0]	清除阈值监视器下限阈值标志 LTFC[x]=0：没有影响 LTFC[x]=1：清除通道 x 的阈值监视器下限阈值标志

滤波器 y 极值监视器最大值寄存器（HPDF_FLTyEMMAX）

地址偏移：0x130 + 0x80 *y，（y = 0…3）

复位值：0x8000 0000

该寄存器只能按字（32 位）访问。



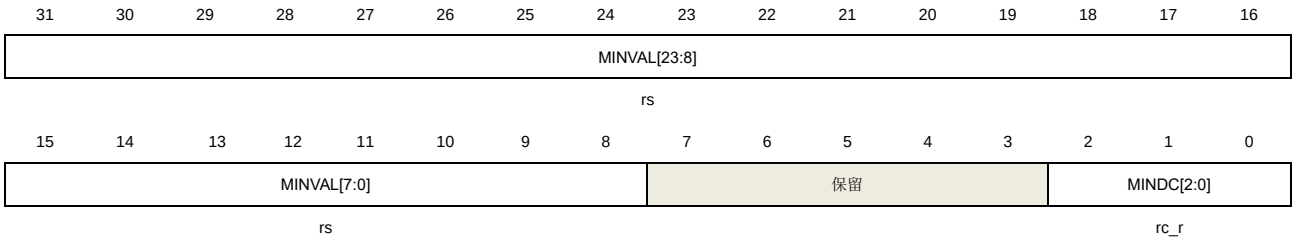
位/位域	名称	描述
31:8	MAXVAL[23:0]	极值监视器最大值 此位域通过硬件置位，表示 HPDF_FLTy 所转换的最大值。 读取该寄存器。此位域被复位。
7:3	保留	必须保持复位值。
2:0	MAXDC[2:0]	极值监视器最大值数据通道 该位域表示哪个通道的值被保存在 MAXVAL[23:0]位域中。 读取该寄存器此位被清 0

滤波器 y 极值监视器最小值寄存器（HPDF_FLTyEMMIN）

地址偏移：0x134 + 0x80 * y，（y = 0…3）

复位值：0x7FFF FF00

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:8	MINVAL[23:0]	极值监视器最小值 此位域通过硬件置位，表示 HPDF_FLTy 所转换的最小值。 读取该寄存器。此位域被复位。
7:3	保留	必须保持复位值。
2:0	MINDC[2:0]	极值监视器最小值数据通道 该位域表示哪个通道的值被保存在 MINVAL[23:0]位域中。

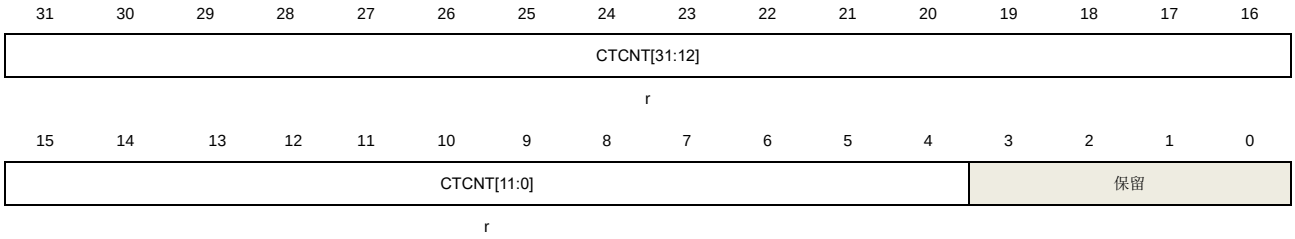
读取该寄存器。此位被清零。

滤波器 y 转换定时器寄存器（HPDF_FLTyCT）

地址偏移：0x138 + 0x80 * y，（y = 0…3）

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:4	CTCNT[27:0]	28 位定时器计数转换时间 t = CNVCNT[27:0] / $f_{HPDFCLK}$ 定时器的输入时钟来自 HPDF 时钟（系统时钟 $f_{HPDFCLK}$）。转换时间测量始于每次转换开始，并止于每 36 次转换结束（即第一次和最后一次串行采样之间的间隔）。只有在滤波器旁路时（SFOR[9:0]=0），转换时间测量才会停止，且 CNVCNT[27:0]=0.时间计时如下： 如果 FAST=0 （或者 FAST=1 连续模式下的第一次转换）： t = [SFOR * （IOR-1 + SFO） + SFO] / f_{CKIN}（适用于 $Sinc^x$ 滤波器） t = [SFOR * （IOR-1 + 4） + 2] / f_{CKIN}（适用于 FastSinc 滤波器） 如果在连续模式下 FAST=1 （第一次转换除外）： t = [SFOR * IOR-1] / f_{CKIN} 如果 SFOR = SFOR[9:0]+1 = 1（滤波器旁路，仅积分器有效）： CNVCNT = 0 （时间计时停止，转换时间: t = IOR / f_{CKIN}） 其中，f_{CKIN} 为给定通道 $CKIN_y$ 引脚上的通道输入时钟频率；在并行数据输入来自内部 ADC 或者来自 CPU/DMA 写操作的情况下，表示输入数据速率。 当转换被中断时（如禁能/使能所选通道）定时器也将该中断时间计算在内
3:0	保留	必须保持复位值。

35. FAC 滤波算法加速器（FAC）

35.1. 简介

滤波器算法加速器（FAC）包含乘法器、累加器和地址生成逻辑单元，该单元可以实现索引保存在本地存储中的矢量元素。FAC在输入端和输出端都支持循环缓冲区，以便于实现包含有限冲激响应（FIR）滤波器和无限冲激响应（IIR）滤波器的数字滤波器。FAC使得CPU免于频繁或冗长的滤波操作，相比于软件实现滤波，它能加速计算，提升关键任务的处理速度。

35.2. 主要特征

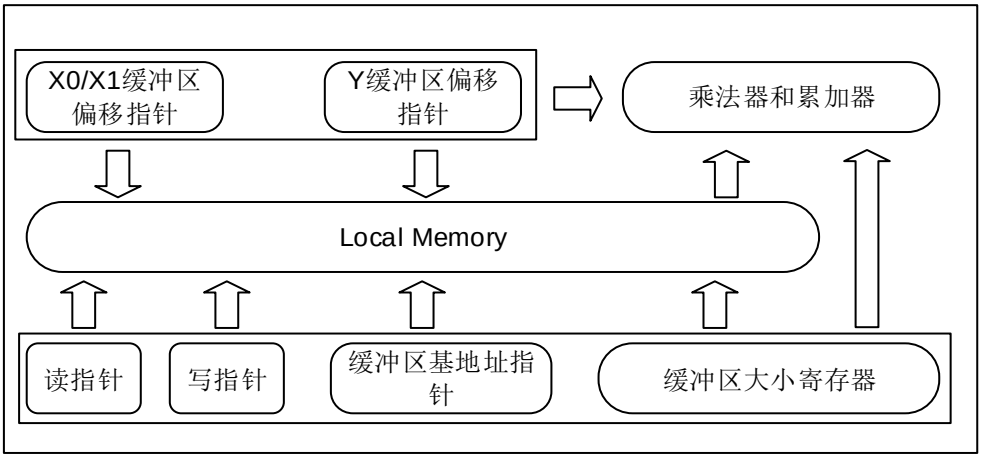
- 定点或浮点乘法器和累加器；
- 256 x 32位本地存储器；
- 16位定点和32位浮点输入端和输出端；
- 3个数据缓冲区，两个输入数据缓冲区一个输出数据缓冲区；
- 数据缓冲区可循环使用；
- 可实现FIR滤波和IIR滤波；
- 矢量函数支持卷积、点乘、相关函数；
- DMA读写数据通道。

35.3. 功能描述

35.3.1. 基本描述

FAC单元可以基于定点乘法器和累加器或浮点乘法器和累加器进行配置。两路16位有符号输入数据或32位浮点数据从本地存储被传输至MAC，二者相乘并累加至累加器。一个指针用以确定本地存储的输入数据的地址，该指针可以递增、递减、加载或由硬件复位。内置定序器控制指针和MAC操作，用以依序执行所请求的操作。

图 35-1. FAC 结构框图



内核和DMA控制器加载两路输入矢量数据到本地存储，用以计算点乘，选择和开启所请求的操作。两路输入向量元素是从本地存储中提取，二者相乘，然后将乘法器输出结果累加。当所请求的操作执行完毕后，累加器的内容被保存在本地存储，其相应的地址可以被内核或DMA读取。

有限冲激响应（FIR）滤波器重复计算点乘，包含反复计算滤波系数和输入采样数据的点乘，同时，丢弃最早的采样数据，新增一次采样数据。

无限冲激响应（IIR）滤波器操作是计算反馈滤波系数和先前的滤波输出数据之间的乘积，并将该计算结果与FIR卷积结果进行累加得到最后的滤波输出。

35.3.2. 本地存储和缓冲区

该单元包含256 x 32位读写存储：输入值保存在缓冲区X0和缓冲区X1，输出值保存在缓冲区Y。

缓冲区的地址和大小具体描述如下：

- x0_base:X0缓冲区基地址；
- x1_base:X1缓冲区基地址；
- y_base:Y缓冲区基地址。

缓冲区的长度为：

- x0_buf_size:X0缓冲区所分配的32位字的数目；
- x1_buf_size:X1缓冲区所分配的32位字的数目；
- y_buf_size:Y缓冲区所分配的32位字的数目。

通过使用初始化函数，X0缓冲区、X1缓冲区、Y缓冲区的内容可以被初始化。数据被传输到写指针所指示的目标缓冲区的位置。随着每次新写操作，写指针递增。如果指针达到所分配的缓冲区空间的终点，指针返回基地址。该特性用以预先加载滤波操作的向量元素，该操作同样适用于加载滤波参数和初始化滤波器。

缓冲区配置寄存器配置缓冲区大小和基地址。滤波器函数定义所需的缓冲区大小，而缓冲区基地址可以在内部存储中被随意配置，因此，考虑到各缓冲区所适配的地址范围为从0x00到0xFF，换言

之，缓冲区基地址与缓冲区长度之和小于256。缓冲区的位置和大小缺少限制，甚至于它们可以完全重叠。为了避免异常运行，不建议滤波器函数的缓冲区重叠。

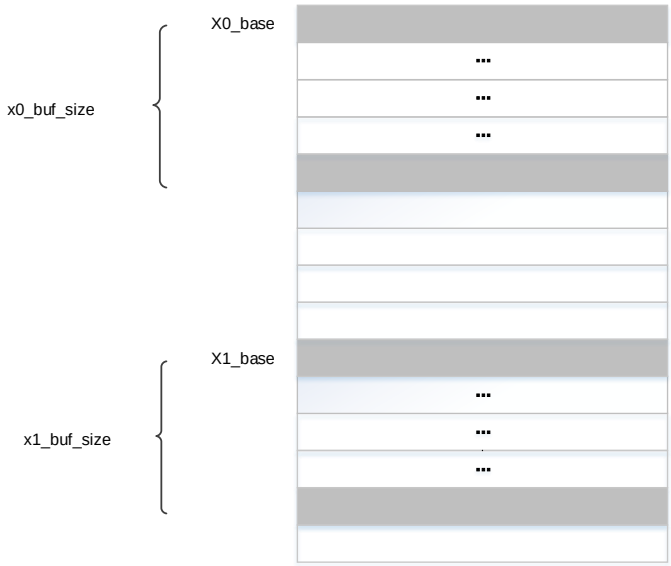
如果需要缓冲区循环使用，则可在缓冲区添加一个可选的净空区。此外，为了调节CPU或DMA的运行，有必要设置水印区级别。为了满足应用性能需求，净空区和水印区的数值需要配置。

通常，为了更高的数据吞吐量，输入缓冲区不应为空，净空区比水印区略微大些，从而能接受中断或DMA延迟。另一方面，如果输入数据的速度小于数据处理的速度，输入缓冲区可以为空，等待下一个数据被写入。因此，缓冲区和净空区相等，从而保证不会在输入端口出现上溢。

35.3.3. 输入缓冲区

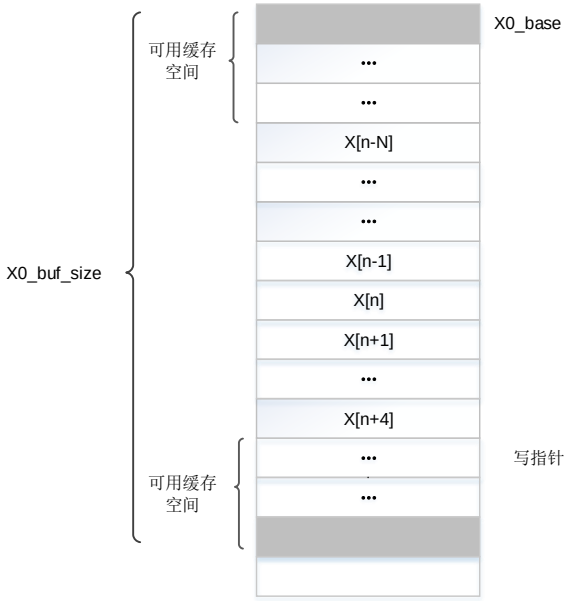
X0缓冲区和X1缓冲区用来保存输入到MAC的数据。每次乘法操作从X0缓冲区提取数值，从X1缓冲区提取数值，并将二者相乘。根据当前函数，控制单元的指针产生读地址偏移，该地址偏移是相对于缓冲区基地址。

图 35-2. 输入缓冲区



如果X0缓冲区工作在循环模式，在有空间可用的情况下，新数据不停地传输到输入缓冲区。对数字滤波器而言，预加载缓冲区是可选的。当滤波操作开始后，如果输入采样值没有被写入到X0缓冲区，缓冲区标识为‘空’。DMA或CPU会产生加载新的采样数据请求，该请求直到有足够的采样数据去参与滤波才会消失。

图 35-3. 循环输入缓冲区



X1缓冲区仅工作在非循环模式下。除非缓冲区内容不跟随FAC运行指令的改变而改变，X1缓冲区通常需要预加载。此外，X1缓冲区可以为滤波器函数保存滤波器系数。

对于循环缓冲区而言，所分配的缓冲区空间（x0_buf_size）应该比在使用的元素数量要大，因此，缓冲区中总是会出现可用的新值。*图35-3. 循环输入缓冲区*展示滤波操作的缓冲区布局。当计算输出采样y[n]时，FAC调用N+1个输入采样，从x[n-N]到x[n]。一旦计算结束，调用输入采样序列x[n-N+1]到x[n+1]，然后开始计算y[n+1]。输入序列丢弃最早的输入采样(x[n-N])，增加新的采样值(x[n+1])。

CPU或DMA有必要保证缓冲区空间中新采样x[n+1]是可用的。如果x[n+1]不可用，缓冲区标识为‘空’，阻塞当前操作的执行，除非增加新的采样。如果定时器或其他外设控制采样的流程，考虑到采样数据输入速度比滤波器处理速度慢，缓冲区通常工作在‘空’状态。

水印区阈值是由FAC_X0BCFG寄存器中的X0_WBFF位决定的，如果空闲空间数目少于水印区阈值，缓冲区被视为‘满’状态。当FAC使能时，如果充满标志没有被置位，中断被使能，生成中断，缓冲区请求更多的数据。不考虑上溢的风险，水印区允许在一次中断中传输若干数据。然而，如果上溢的情况出现，OFEF错误标志被置位，所写的数据被忽略，同时写指针不会递增。

*图35-4. 循环输入缓冲区区域*展示了滤波器运行时，X0缓冲区的改变过程，滤波器为长度为6的FIR滤波器，同时水印区设置为3。

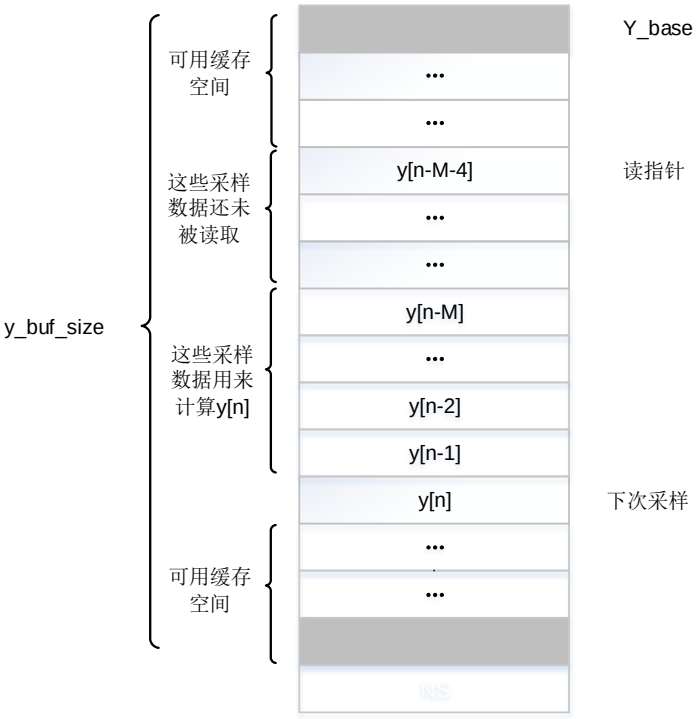
图 35-4. 循环输入缓冲区区域

35.3.4. 输出缓冲区

累加的输出保存在Y缓冲区中，当处理器或DMA读取输出值时，缓冲区空间将被释放。

读指针指向读操作时需要读取数据的地址，当有读命令发生时，所读的数据是从读指针地址中获取。随着每次读操作，其指针递增，当指针达到所分配的Y缓冲区空间末端时，将返回基地址。

图 35-5. 循环输出缓冲区



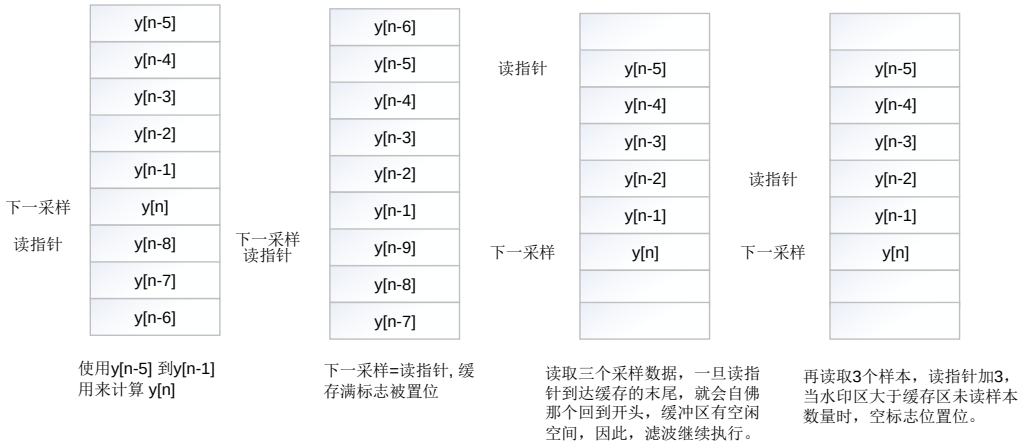
Y缓冲区也会工作在循环缓冲区模式。如果下一个输出数据的地址与读指针所指示的地址相等，缓冲区被标识为‘满’，然后滤波被中止，直至采样值被读取。

对于IIR滤波器，M个先前的输出采样值序列，从y[n-M]到y[n-1]，用来计算下一个输出采样y[n]。当增加一个新的采样时，最近最少的采样y[n-M]被丢弃。

如果水印区阈值不小于缓冲区中未读数据的数目，缓冲区被视为‘空’。水印阈值由YBCFG寄存器的Y_WBFF位设置。在‘空’标志未被置位的情况下，如果中断或DMA被使能，产生请求去从Y缓冲区读取数据。在不存在下溢的风险的前提下，通过水印区可实现若干数据在一次中断中被传输。然而，在下溢发生的情况下，UFEF标志被置位，读指针不增加，由读指针所寻址的存储内容返回到读操作。

*图35-6. 循环输出缓冲区区域*展示了滤波器运行时，Y缓冲区的改变过程，滤波器为长度为5的IIR滤波器，同时水印区设置为3。

图 35-6. 循环输出缓冲区区域



35.3.5. 初始化函数

当EXE位置1时，FAC单元的初始化是通过FAC_PARACFG寄存器的FUN位写入适当数值以触发下述函数实现的。IPP和IPQ位必须包含合适的参数，IPR不被使用，随着初始化函数完成，EXE位由硬件复位。

DMA请求和中断在初始化函数运行时应被关闭，当不需要数据流控制时，数据可通过DMA或软件传输到FAC存储。

X0 缓冲区加载函数

该加载函数从地址X0_BASE预加载值，写入数据从FAC_WDATA寄存器加载到X0缓冲区，同时，写入地址递增，当N个值被加载到X0缓冲区时，写指针最终指向X0_BASE+N。参数IPP中包含N个被加载到X0缓冲区的值，同时，IPQ和IPR未被使用，当N次对FAC_WDATA寄存器的写操作完成时，X0缓冲区加载操作完成。

X1 缓冲区加载函数

该加载函数从地址X1_BASE预加载值，写入数据从FAC_WDATA寄存器加载到X1缓冲区，同时，写入地址递增，在IIR滤波中，N个前馈滤波系数和M个反馈滤波系数被加载到X1缓冲区，滤波系数总和时N+M，在FIR滤波中，由于没有反馈滤波系数，因此M=0，N个前馈滤波系数被加载到X1缓冲区中。

参数IPP包含N个前馈滤波系数、参数IPQ包含M个反馈滤波系数，IPP和IPQ均被加载到X1缓冲区中，同时，IPP的起始地址是X1_BASE、IPQ的起始地址是X1_BASE+N、IPR未被使用。当N+M次对FAC_WDATA寄存器的写操作完成时，X1缓冲区加载操作完成。

Y 缓冲区加载函数

该加载函数从地址Y_BASE预加载值，写入数据从FAC_WDATA寄存器加载到Y缓冲区，同时，写

入地址递增，写入指针最终指向地址Y_BASE+N。此功能可用于预加载IIR滤波器中反馈存储参数，IPP中包含N个反馈存储参数，IPQ和IPR未被使用，当N次对FAC_WDATA寄存器的写操作完成时，Y缓冲区加载操作完成。

35.3.6. 滤波器函数

当EXE被置位时，向FAC_PARACFG寄存器中的FUN位域写入合适的值可触发FIR滤波或IIR滤波函数，IPP、IPQ、IPR中包含适用于每种滤波的滤波参数，滤波器会一直运行直到软件复位EXE位。

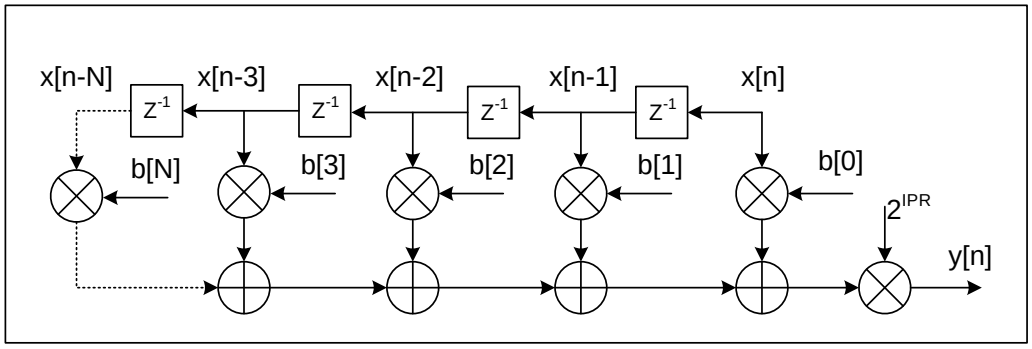
FIR filter: Y = B*X

$$y_n=2^{IPR}\sum_{k=0}^{N}(x_{n-k}b_k) \quad (35\text{-}1)$$

向量B包含N+1个滤波系数，X包含无限长的输入采样。Y的元素按照点乘$y_n = B.X_n$计算得到，$X_n = [x_{n-N},...,x_n]$，此函数用来实现有限冲激响应(FIR)滤波器。

FIR滤波器的结构如*图35-7. FIR滤波器结构图*所示。

图 35-7. FIR 滤波器结构图



X0缓冲区是一个循环缓冲区

，由向量X的元素组成，缓冲区的长度是N+1+d，d是净空区长度，X1缓冲区由向量B的元素组成，缓冲区的长度是N+1，Y缓冲区是一个循环缓冲区，由输出值（y_n）组成，缓冲区的长度是d，IPP的长度是N+1，向量B在[2:127]之间，参数IPR是增益，乘以2^{IPR}得到最终的滤波输出结果，IPR的范围是[0:7]，IPQ未使用。

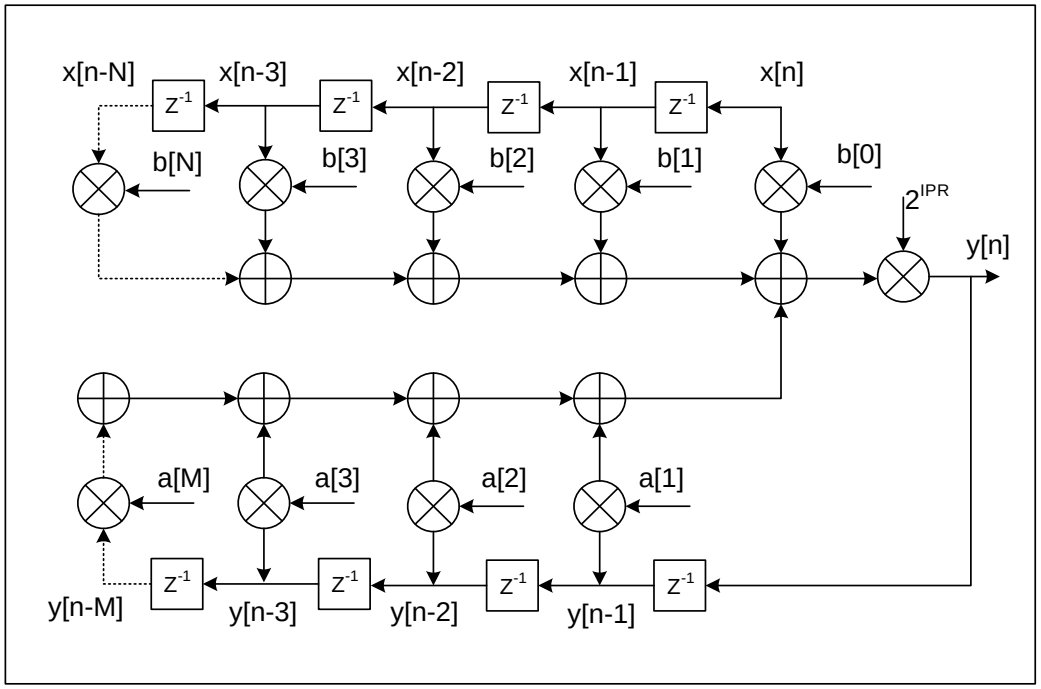
IIR filter: Y = B*X+A*Y

$$y_n=2^{IPR}\left(\sum_{k=0}^{N}(x_{n-k}\times b_k)+\sum_{k=1}^{M}\left(y_{n-k}\times a_k\right)\right) \quad (35\text{-}2)$$

无限冲激响应IIR滤波器的输出向量Y是系数向量B（长度为N+1）和无限长度的向量X之间的卷积，将向量Y和向量A的卷积相加，向量Y的元素由$B.X_n + A.Y_{n-1}$计算得到，其中，$X_n = [x_{n-N},...,x_n]$由N+1个元素组成，$Y_{n-1}= [y_{n-M},...,y_{n-1}]$由M个元素组成。

IIR滤波器的结构如*图35-8. IIR滤波器结构图*所示。

图 35-8. IIR 滤波器结构图



X0缓冲区是一个循环缓冲区，由向量X的元素组成，缓冲区的长度是N+1+d，X1缓冲区由向量B的元素和向量A的元素组成，缓冲区的长度是M+N+1，Y缓冲区是一个循环缓冲区，由输出值（y_n）组成，缓冲区的长度是M+d，IPP的长度是N+1，向量B在[2:64]之间，参数IPR是增益，乘以2^{IPR}得到最终的滤波输出结果，IPR的范围是[0:7]，参数IPQ的长度是M，向量A的长度是[1:63]。

35.3.7. 定点数据格式

FAC按照有符号定点数据格式（q1.15）操作输入和输出数据，该格式用1个符号位和15个小数位表述数据，其数值范围为-1（0x8000）到$1-2^{-15}$（0x7FFF）。

累加器数据格式是q4.22，该格式有26位，4位整型符号位和22位小数位，支持累加结果的范围为-8（0x2000000）到+7.99999976（0x1FFFFFF）。可编程增益的取值范围为0dB到42dB，以6dB为增幅，被应用到累加器输出。

如果超过数值范围，其数值大于+7.99999976或小于-8，累加器内容未饱和，所触发的绕回是无害的，因为随后的累加操作会取消绕回。然而，如果发生了绕回，FAC_STAT寄存器的STEF标志置位，此时，FAC_CTL寄存器的STEIE位置位，产生相应的中断。

通过置位FAC_CTL寄存器的CPEN位，在可编程增益应用后，累加器的数据输出可选饱和。如果CPEN位置1，任何数值超过q1.15输出的数值范围，根据其符号，会被置为-1或$1-2^{-15}$。如果CPEN位未置1，在应用增益后，未使用的累加器位会被简单地缩短。

35.3.8. 浮点数据格式

如*表35-1. IEEE 32位单精度浮点格式*所示，操作数据和计算结果数据格式，符合IEEE 32位单精度浮点格式

表 35-1. IEEE 32 位单精度浮点格式

S [31]	E [30:23]	M [22:0]	数值 (V)
0	0	0	零 （V = 0）
1	0	0	负零（V = -0）
0 + ve 1 - ve	0	非零	非规格化$(V=(-1)^{s}*2^{(-126)}*(0.M))$
0 + ve 1 - ve	1 to 254	0 to 0x7FFFFF	正常范围$(V=(-1)^{s}*2^{(E-127)}*(1.M))$
0	254	0x7FFFFF	正最大值（V = +Max）
1	254	0x7FFFFF	负最大值（V = -Max）
0	最大值 255	0	正无穷（V = + Infinity）
1	最大值 255	0	负无穷（V = - Infinity）
x	最大值 255	non zero	非数（V = NaN）

FAC关于不同的IEEE浮点数据格式的处理如下：

非规格化数据：非规格化操作数（E=0, M!=0）输入被视为0（E=0, M=0）。

溢出：当操作中的数据按照既定浮点格式显得过大会产生上溢，在这种情景下，返回正无穷或负无穷，并且FAC_STAT寄存器的STEF标志置位。

非数（NaN）：当NaN操作数（E=max, M!=0）数入被视为无穷（E=max, M=0）。

注意：配置FAC_CTL寄存器的FLTEN位置1，输入和输出数据符合32位IEEE数据格式，不支持数据截断和配置增益。

35.3.9. FIR 滤波器

FAC支持长度为N的FIR滤波器，N是抽头或滤波系数数量。根据FIR滤波器的本地存储需求，最小的长度N应是2N+1：N输入采样、1个输出采样和N个滤波系数。鉴于本地存储大小是256，N的最大值是127。为了最大的数据吞吐量，有必要分配少量的额外空间，d0是为了输入采样缓冲区，d1是为了输出采样缓冲区，为了保证滤波器不被中止，从而等待新的输入采样或读取输出采样。本地存储需求是2N+d0+d1。其相应的缓冲区应按照如下配置：X0_BUF_SIZE等于N + d0，X1_BUF_SIZE等于N，Y_BUF_SIZE等于d1。如果不需要额外空间，Y_BUF_SIZE可为1。

即使缓冲区基地址可以分配到任意地址，也有必要避免X1缓冲区与其他缓冲区重叠，否则，其系数被覆盖。示例配置如下：X1_BASE等于0，X0_BASE等于N，Y_BASE等于2N+d0。然而，如果存储空间是有限的，X0缓冲区和Y缓冲区可以是重叠的，没有必要保证X0_BASE等于N，X1_BASE

等于0，Y_BASE等于N。输出采样代替最旧的输入采样，因为Y_BUF_SIZE与X0_BUF_SIZE等于N+d0，因此缓冲区依然保持在同步状态。

注意：X0BCFG寄存器的X0_WBFF位必须由不多于$log_2(d0)$的数值编写，否则，在N个输入参数被写入前，缓冲区标识为‘满’，而后不会请求更多的采样。以此类推，YBCFG寄存器的Y_WBFF位必须由不多于$log_2(d1)$的数值编写。

X1缓冲区必须预加载滤波器系数。多达N采样的数量能预加载到X0缓冲区。因为FIR滤波器没有反馈路径，没有必要预加载Y缓冲区。

向FAC存储写数据和从FAC存储读数据的方式有三种：轮询、中断和DMA。在轮询方式中，在向FAC存储写数据和从FAC存储读数据前，软件需要确认X0BFF位或YBEF位是复位状态。在中断方式中，当与写操作相关的X0BFF标志为0时，或者当与读操作相关的YBEF标志为0时，会产生中断请求。在DMA方式中，当X0BFF标志或YBEF标志为复位状态时，会产生DMA写通道或读通道请求。

将以下值写入FAC_PARACFG寄存器，滤波器被使能。

FUN=8(FIR 滤波器);IPP=N(滤波系数数量);IPQ=“任意值”;IPR=增益;EXE=1。

如果预加载到X0缓冲区的数值少于$N + d – 2^{X0_WBFF}$，X0BFF标志保持为0。如果WIE位置1，写中断请求会触发通过FAC_WDATA寄存器写2^{X0_WBFF}额外采样值到X0缓冲区。中断处理函数需在每写2^{X0_WBFF}数值到FAC_WDATA寄存器后检查X0BFF标志，除非FAC_STAT寄存器的X0BFF位置位。以此类推，如果寄存器FAC_CTL的DWEN位置1，将会不断地产生DMA写通道请求，除非寄存器FAC_STAT的X0BFF位置1。

当最少N采样值被写入X0缓冲区，滤波器计算第一个输出采样值。当2^{Y_WBEF}输出采样被写入到Y缓冲区，FAC_STAT寄存器的YBEF位复位。如果寄存器FAC_CTL的RIE位置1，触发中断请求，从而处理器从缓冲区读取2^{Y_WBEF}采样值。中断处理函数需从FAC_RDATA寄存器每读2^{Y_WBFF}数值后检查YBEF标志，除非FAC_STAT寄存器的YBEF位置位。以此类推，如果寄存器FAC_CTL的DREN位置1，不断地产生DMA读通道请求，除非寄存器FAC_STAT的YBEF位置1。复位EXE位可以停止滤波器运行，否则，滤波器继续运行。

35.3.10. IIR 滤波器

FAC支持IIR滤波器长度N（滤波系数或前馈抽头的数量）和M(反馈滤波系数的数量，其数值的取值范围为从1到N-1)。

IIR滤波器的最小的存储需求是2N+2M，包含N个前馈滤波系数和M个反馈滤波系数，M个输出采样值和N个输入采样值。在M等于N-1的情况下，最大的可实现的滤波器长度是N=64。

为了最大的数据吞吐量，需要允许存在额外空间，空间d0是为了输入缓冲区，空间d1是为了输出缓冲区，所以存储需求的总计大小为2M+2N+d0+d1。X0_BUF_SIZE=N+d0,X1_BUF_SIZE =N+M和Y_BUF_SIZE=M+d1。缓冲区基地址必须不能重叠，即使其地址可以随意分配，比如X1_BASE=0, X0_BASE=N+M和Y_BASE=2N+M+d0。

注意：X0BCFG寄存器的X0_WBFF位必须由不多于$log_2(d0)$的数值编写，否则，在N个输入采样被写入前，缓冲区被标识为‘满’，而后不会请求更多的采样。以此类推，YBCFG寄存器的Y_WBFF位必须由不多于$log_2(d1)$的数值编写。

在加载X1缓冲区功能时，X1缓冲区必须预加载滤波器系数(N前馈滤波系数和M反馈滤波系数)。多达N的采样数据能预加载X0缓冲区。以此类推，多达M的采样数据能预加载Y缓冲区。

将以下值写入FAC_PARACFG寄存器，滤波器被使能。

FUN=9(IIR滤波器);IPP=N(前馈滤波系数数量);IPQ=M(反馈滤波系数数量);IPR=增益;EXE=1。

如果预加载到X0缓冲区的数值少于$N+d-2^{X0_WBFF}$，X0BFF标志保持为0。如果FAC_CTL寄存器的WIE位置1，通过FAC_WDATA寄存器写2^{X0_WBFF}额外采样值到缓冲区，触发写中断请求。中断处理函数需在每写2^{X0_WBFF}数值到FAC_WDATA寄存器后检查X0BFF标志，除非FAC_STAT寄存器的X0BFF位置位。以此类推，如果寄存器FAC_CTL的DWEN位置1，不断地产生DMA写通道请求，除非寄存器FAC_STAT的X0BFF位置1。

当最少N采样值被写入X0缓冲区，滤波器通过计算X0缓冲区最前面的N采样值和Y缓冲区最前面的M采样值，产生第一个输出采样值。第一个输出采样值写入Y缓冲区的地址是Y_BASE+M。

当2^{Y_WBEF}输出采样被写入到Y缓冲区，FAC_STAT寄存器的YBEF位是0。当处理器从缓冲区读取2^{Y_WBEF}采样值，如果寄存器FAC_CTL的RIE位置1，触发中断请求，从而处理器从缓冲区读取2^{Y_WBEF}采样值。中断处理函数需从FAC_RDATA寄存器每读2^{Y_WBFF}数值后检查YBEF标志，除非FAC_STAT寄存器的YBEF位置位。以此类推，如果寄存器FAC_CTL的DREN位置1，不断地产生DMA读通道请求，除非寄存器FAC_STAT的YBEF位置位。复位EXE位可以停止滤波器运行，否则，滤波器继续运行。

35.4. FAC 寄存器

FAC 基地址：0x4802 4800

35.4.1. FAC X0 缓冲区配置寄存器（FAC_X0BCFG）

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字（32位）访问，仅在FAC_PARACFG寄存器的EXE位为0时可修改。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						X0_WBFF[1:0]		保留							
						rw									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X0B_SIZE[7:0]								X0B_ADDR[7:0]							
rw								rw							

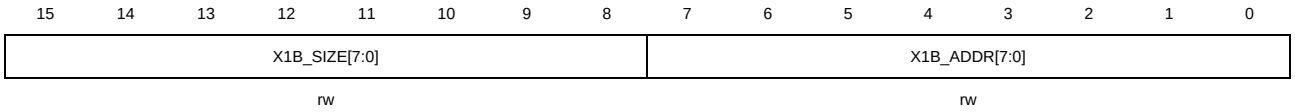
位/位域	名称	描述
31:26	保留	必须保持复位值。
25:24	X0_WBFF[1:0]	缓冲区水印区满标志位 如果缓冲区可用空间的数目少于 2^{X0_WBFF}，标志位置位 00：如果DMA写请求使能，阈值为1 01：阈值为2 10：阈值为4 11：阈值为8 如果若干数据在一次中断中传输到缓冲区，设置阈值大于1。
23:16	保留	必须保持复位值。
15:8	X0B_SIZE[7:0]	X0 缓冲区大小，滤波器前馈滤波系数数目。
7:0	X0B_ADDR[7:0]	X0 缓冲区基地址

35.4.2. FAC X1 缓冲区配置寄存器（FAC_X1BCFG）

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															



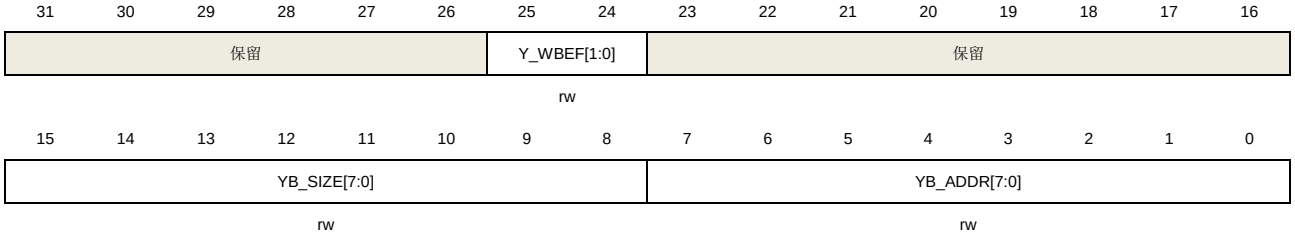
位/位域	名称	描述
31:26	保留	必须保持复位值。
15:8	X1B_SIZE[7:0]	X1 缓冲区分配大小 当 FAC 正在运行（EXE=1）时，该位域不可改变。
7:0	X1B_ADDR[7:0]	X1 缓冲区基地址 当 FAC 正在运行（EXE=1）时，该位域可进行改变，当改变滤波器系数值时，滤波器应该暂停，因为在滤波器计算过程中改变滤波器参数会影响输出结果。

35.4.3. FAC Y 缓冲区配置寄存器（FAC_YBCFG）

地址偏移：0x08

复位值：0x0000 0000

该寄存器只能按字（32位）访问。该寄存器仅在FAC_PARACFG寄存器的EXE位为0时可修改。



位/位域	名称	描述
31:26	保留	必须保持复位值。
25:24	Y_WBEF[1:0]	缓冲区水印区空标志 如果缓冲区水印区可用空间的数目少于 2^{Y_WBEF}，标志位置位。 00：阈值为1 01：阈值为2 10：阈值为4 11：阈值为8 如果若干数据在一次中断中传输到缓冲区，设置阈值大于 1。 如果 DMA 读数据指令被使能，阈值应设置为 1。
23:16	保留	必须保持复位值。
15:8	YB_SIZE[7:0]	Y 缓冲区分配大小 对于 FIR 滤波器，最小缓冲区大小为水印区阈值加 1。

		对于IIR滤波器，最小缓冲区大小为水印区阈值与反馈抽头数目之和。
7:0	YB_ADDR[7:0]	Y 缓冲区基地址。

35.4.4. FAC 参数配置寄存器（FAC_PARACFG）

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
EXE	FUN[6:0]							IPR[7:0]							
rw	rw							rw							

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

位/位域	名称	描述
31	EXE	执行 0：FAC停止运行 1：FAC 开始运行 使能该位，FAC 会执行 FUN 位域定义的功能，当产生软件复位时，FAC 会停止任何正在进行的功能。该位由硬件复位以实现初始化功能。
30:24	FUN[6:0]	功能 0000001：加载X0缓冲区 0000010：加载X1缓冲区 0000011：加载Y缓冲区 0001000：FIR滤波器 0001001：IIR 滤波器 其他：保留 该位仅在FAC_PARACFG寄存器的EXE位为0时可修改。
23:16	IPR	输入参数 IPR 该位仅在FAC_PARACFG寄存器的EXE位为0时可修改。
15:8	IPQ	输入参数 IPQ 该位仅在 FAC_PARACFG 寄存器的 EXE 位为 0 时可修改。
7:0	IPP	输入参数 IPP 该位仅在 FAC_PARACFG 寄存器的 EXE 位为 0 时可修改。

35.4.5. FAC 控制寄存器（FAC_CTL）

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															RST
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPEN	FLTEN	保留				DWEN	DREN	保留		GSTEIE	STEIE	UFEIE	OFEIE	WIE	RIE
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	RST	复位 FAC 单元 0：除能复位 1：使能复位 当 RST=1 时，写指针、读指针、EXE 位、FAC_STAT 寄存器、FAC_PARACFG 寄存器会产生复位。
15	CPEN	限幅使能 0：限幅禁能，累加器超出范围的值被截断。 1：限幅使能，累加器超出范围的值被限幅到最大正值或最小负值。
14	FLTEN	浮点格式使能 0：输入数据和结果支持定点有符号整数格式q1.15。 1：输入数据和结果支持32位单精度浮点格式。 该位仅在FAC_PARACFG寄存器的EXE位为0时可修改。
13:10	保留	必须保持复位值。
9	DWEN	DMA 写通道使能 0：未产生DMA请求 1：在 X0 缓冲区未满的情况下，产生 DMA 请求 该位仅在 FAC_PARACFG 寄存器的 EXE 位为 0 时可修改。
8	DREN	DMA 读通道使能 0：未产生DMA请求 1：在 Y 缓冲区未空的情况下，产生 DMA 请求 该位仅在 FAC_PARACFG 寄存器的 EXE 位为 0 时可修改。
7:6	保留	必须保持复位值。

5	GSTEIE	增益饱和错误中断使能 0：未产生中断. 1：如果 GSTEF 标志置 1，产生中断请求 软件置位和复位该位
4	STEIE	饱和错误中断使能 0：未产生中断. 1：如果 STEF 标志置 1，产生中断请求 软件置位和复位该位。
3	UFEIE	下溢错误中断使能 0：未产生中断. 1：如果 UFEF 标志置 1，产生中断请求 软件置位和复位该位。
2	OFEIE	上溢错误中断使能 0：未产生中断 1：如果 OFEF 标志置 1，产生中断请求 软件置位和复位该位。
1	WIE	写中断使能 0：未产生中断. 1：如果 X0BFF 标志置 1，产生中断请求 软件置位和复位该位。
0	RIE	读中断使能 0：未产生中断. 1：如果 YBEF 标志置 1，产生中断请求 软件置位和复位该位。

35.4.6. FAC 状态寄存器（FAC_STAT）

地址偏移：0x14

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															
				r	r	r	r							r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				GSTEF	STEF	UFEF	OFEF	保留						X0BFF	YBEF

位/位域	名称	描述
31:12	保留	必须保持复位值。
11	GSTEF	增益饱和错误标志，在增益后超过范围时置位。 0：未检测到增益饱和错误 1：增益饱和错误被检测
10	STEF	饱和错误标志 0：未检测到饱和错误 1：饱和错误被检测 当累积结果超出范围时发生饱和。
9	UFEF	下溢错误标志 0：未检测到下溢错误 1：检测到下溢错误 当 Y 缓冲区中没有可用有效数据时，从 FAC_RDATA 读取数据时发生下溢错误。
8	OFEF	上溢错误标志 0：未检测到上溢错误 1：检测到上溢错误 当 X1 缓冲区中没有空闲空间时，向 FAC_WDATA 写数据时会产生上溢错误。
7:2	保留	必须保持复位值。
1	X0BFF	X0 缓冲区满标志 0：X0缓冲区未满 1：X0 缓冲区已满 硬件或复位会置位或复位该标志。
0	YBEF	Y 缓冲区空标志 0：Y缓冲区未空 1：Y 缓冲区已空 硬件或复位会置位或复位该标志。

35.4.7. FAC 写数据寄存器（FAC_WDATA）

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

WDATA

rw

当FLTEN为1时，浮点数据被选择

位/位域	名称	描述
31:0	WDATA[31:0]	写数据 当对寄存器执行写命令时，写数据被传送到写指针指向的地址偏移，每次写入数据完成后，指针地址递增。

当FLTEN为0时，定点数据被选择

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	WDATA[15:0]	写数据 当对寄存器执行写命令时，写数据被传送到写指针指向的地址偏移，每次写入数据完成后，指针地址递增。

35.4.8. FAC 读数据寄存器（FAC_RDATA）

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

当FLTEN为1时，浮点数据被选择

位/位域	名称	描述
31:0	RDATA[31:0]	读数据 当对寄存器执行读命令时，读指针指向的 Y 缓冲区的内容就是读到的数据，每次读数据完成时，读指针递增。

当FLTEN为0时，定点数据被选择

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	RDATA[15:0]	读数据 当对寄存器执行读命令时，读指针指向的 Y 缓冲区的内容就是读到的数据，每次读数据完成时，读指针递增。

36. 附录

36.1. 寄存器表中使用的缩写列表

表 36-1. 寄存器功能位访问属性

寄存器表中缩写	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）	软件只能读该位，写入 1 触发事件，但对位值没有影响。
可读/可置位（rs）	软件可以读该位，也可以将这个位设置为 1。写入 0 对位值没有影响。
可读/读清零（rc_r）	软件可以读该位，读该位会自动清零。写入 0 对位值没有影响。
可读/读置位（rs_r）	软件可以读该位，读该位会置位该位。写对位值没有影响。
读/写一次（rwo）	软件只可写该位一次，可以读任意次。只有复位可将该位恢复为默认值。
读/写清零（rc_w）	软件可以读该位，对该位写可以清除这个位。写 0 和写 1 效果相同。
只读/写 1 触发（rt_w）	软件只能读该位，写 0 或 1 触发事件，但对位值没有影响。

36.2. 术语表

表 36-2. 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，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

术语	描述
APB	高级外设总线
RAZ	读为 0
WI	写忽略
RAZ/WI	读为 0/写忽略

36.3. 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。

37. 版本历史

表 37-1. 版本历史

版本号	描述	日期
1.0	首次发布	2024 年 11 月 08 日
1.1	1. 修改***配置寄存器(SYSCFG_CFG3)***，***配置寄存器(SYSCFG_CFG4)***和***配置寄存器(SYSCFG_CFG5)***低 2 比特的描述； 2. 修改 ***TCMSRAM 控制与状态寄存器(SYSCFG_TCMSRAMCS)***比特 0 的描述； 3. ***TCMSRAM 写保护***章节增加描述“当 TCMSRAM 设置写保护时，对其进行操作会产生 hardfault。” 4. 删除非对称 PWM 模式及其相关内容； 5. ***HRTIMER 故障输入配置寄存器 2 (HRTIMER_FLTINCFG2)***中比特 7 的描述 FTL4INEN 修改为 FTL0INEN； 6. 修改 ***25.4.10. 双通道触发模式***章节中***图 25-58. 当 TRIG0M = 1 和 TRIG1M = 1 时 DAC 触发***； 7. 修改 ***25.4.2. Slave_TIMERx(x=0...7)单元***章节中***表 25-17. 消隐模式下的滤波信号映射***，***图 25-36. 延迟空闲模式，ISO0=0 和 CHOP=1*** 和***图 25-37. 延迟空闲模式，ISO0=1 和 CHOP=1***； 8. 修改 ***25.4.4. 突发模式***章节下***突发模式***进入的描述； 9. 修改 ***Slave_TIMERx 控制寄存器 1（HRTIMER_STxCTL1）***中 TRGHALFM 位域描述； 10. HRTMER ***故障消隐***章节中描述“当故障通道事件的数量等于 FLTy[3:0]时，故障生效”修改为“当故障通道事件的数量等于 FLTy[3:0]+1 时，故障生效”； 11. 修改 ***HRTIMER 控制寄存器 1(HRTIMER_CTL1)***中位域 ST6SUP[1:0]描述； 12. 修改***图 17-1. ADC 模块框图***； 13. 修改***表 17-2. ADC 输入引脚定义***； 14. ***17.4.2. 双时钟域架构***章节增加“注意”描述； 15. 修改 ***33.3.3. 外部设备地址映射***章节中***图 33-3. Bank0 地址映射***； 16. 修改***表 33-3.支持的 16 位传输***；	2025 年 02 月 14 日

	17. 修改 ***SRAM/NORFlash 写时序寄存器（EXMC_SNWTCFGx）（x=0,1,2,3)***中 WBUSLAT[3:0]位域描述； 18. 修改***图 4-1.系统复位电路***; 19. 修改 ***4. 复位和时钟控制单元***章节中 ***4.2.1. 简介***描述，修改***图 4-2. 时钟树***; 20. 文档中 IRC40K 修改为 IRC32K； 21. ***CMP3 控制状态寄存器（CMP3_CS）***中位域 CMP3BLK[3:0]描述“0011：选择 TIMER14_CH2 输出比较信号为消隐源”修改为“0011：选择 TIMER14_CH1 输出比较信号为消隐源”； 22. 修改***控制寄存器（DBG_CTL2）***中部分位域描述； 23. 修改 ***24.4.7. 外部触发映射***章节***表 24-2. 外部触发映射***; 24. 修改 ***6.4. 内部连接***章节***表 6-1. 触发输入位域选择***; 25. ***29.3.9. 错误和状态***章节中***总线集成状态***增加“注意”描述； 26. 修改 ***29.3.10. 通信参数***章节中***内部计数器时钟源***描述； 27. 修改***位时间寄存器（CAN_BT）***和 ***FD 位时间寄存器(CAN_FDBT)***中位域 BAUDPSC[9:0]的描述； 28. 修改***接收邮箱公有过滤寄存器（CAN_RMPUBF)***位域描述； 29. 修改 ***7.4. GPIO 寄存器***中 GPIO 的 port 口复位值； 30. 修改***表 5-2. 中断向量表***; 31. 修改***图 31-1. QSPI 结构框图***; 32. 修改***设备配置寄存器（QSPI_DCFG）寄存器***中 DLYSCEN 位描述； 33. 在 ***2.3.7. 整片擦除***，***2.3.8. 主存储器闪存块编程***，***2.3.9. OTP 编程***和 ***2.4.5. 状态寄存器***章节中增加描述“当主存储块编程执行成功且使能操作结束中断(ENDIE = 1)时，FMC_STAT 寄存器中的 ENDF 置位”； 34. 删除***状态寄存器（SPI_STAT）***中有关 I2S 的描述； 35. 修改***图 3-1. 电源域概览***和增加***图 3-2. 1.1V 电源域概览***。	
1.2	1. 更新***表 31-1. QSPI 信号线描述***中 DQS 信号描述； 2. 将***表 25-4. 交错模式的比较值***中的	2025 年 8 月 8 日

	HRTIMER_MTCARL 改为 HRTIMER_MTCAR； 3. 将***表 25-8. 两种交错模式比较值***中的 HRTIMER_STxCARL 改为 HRTIMER_STxCAR； 4. ***7.3.11. I/O 补偿单元***章节中删除 SYSCFG_CPSCCCFG 寄存器描述； 5. ***17.4.4. 单端和差分输入通道***中删除差分模式的通道不能在单端采样的描述。 6. LPTIMER 章节中，编码器改成译码器；“内部 16 MHz RC”改成“内部 8MHz RC”； 7. 将 HRTIMER 章节***外部事件计数***中的“仅当 EXEVXCNT[5:0]的值大于或等于 EXEVXCNTTHR[5:0]时”改为“仅当 EXEVXCNT[5:0]的值大于或等于 EXEVXCNTTHR[5:0] + 1 时”； 8. 将 HRTIMER 章节***外部事件计数***中“在写入 EXEVXCNT[5:0]位之后应置位 EXEVXCEN 位，且在启用外部事件 X 计数器时可以更改 EXEVXCNT[5:0]位。新值 EXEVXCNT[5:0]将在下一个周期生效。在启用外部事件 X 计数器时不能更改 EXEVXSEL[3:0]。”中的 EXEVXCNT[5:0]改为 EXEVXCNTTHR[5:0]； 9. HRTIMER 章节 ***25.4.8. ADC 触发分频***中出现的 ROVM 改为 ADCROVM，修改图 25-56. 中央对齐计数模式下 ADC 触发分频，补充关于 ADC 触发事件的描述； 10. 修改 HRTIMER 章节 ***Slave_TIMERx 控制寄存器 1 (HRTIMER_STxCTL1)***中 ADCROVM 位域描述； 11. ***Slave_TIMERx 计数器复位和通道 0/通道 1 置位复位请求附加寄存器 (HRTIMER_STxCNTRSTA)*** 中，修改 Slave_TIMER5 - Slave_TIMER7 的 bit20、21、28、29 的位名和位域描述； 12. 修改 HRTIMER 章节***图 25-44. 当 EXEVXCEN 为 1 和 EXEVXCNTTHR[5:0]为 2 时外部事件 X 计数器***； 13. 修改 ***Master_TIMER 控制寄存器 0 (HRTIMER_MTCTL0)***中 ALTM[1:0]位域和 ***Slave_TIMERx 控制寄存器 0 (HRTIMER_STxCTL0)***中 ALTM[1:0]位域描述； 14. ***Slave_TIMERx 故障控制寄存器 (HRTIMER_STxFLTCTL)***中 FLTENPROT 比特位	

	域描述由“FLTyEN(y=0..4)”修改为“FLTyEN(y=0..7)”； 15. 将***死区时间模式***中以及 ***SlaVe_TIMERx 死区控制寄存器(HRTIMER_STxDTCTL)***中的 DTGCKDIV[2:0]修改为 DTGCKDIV[3:0]； 16. 修改***寄存器 HRTIMER ADC 触发源 1 附加寄存器 (HRTIMER_ADCTRIGS1A)***，***HRTIMER ADC 触发源 2 附加寄存器 (HRTIMER_ADCTRIGS2A)***，***HRTIMER ADC 触发源 3 附加寄存器 (HRTIMER_ADCTRIGS3A)***中的 bit0-bit4 名称。 17. 将 ***Slave_TIMERx 通道输出控制寄存器 (HRTIMER_STxCHOCTL)***中的 bit7 由 BMCH1DTI 改为 BMCH0DTI； 18. 修改 ***Slave_TIMERx 计数器复位寄存器 (HRTIMER_STxCNTRST)***中 bit1-3 位域描述； 19. 修改 ***27.3.11 半双工通信***章节引脚配置描述； 20. HRTIMER 章节新增 ***HRTIMER 故障恢复控制寄存器 (HRTIMER_FLTRECCTL)***； 21. 修改 HRTIMER 章节中***计数器时钟***小节描述； 22. ***HRTIMER 故障输入配置寄存器 2 (HRTIMER_FLTINCFG2)***中 FLT0RST 位域描述。 23. 修改***备份域控制寄存器（RCU_BDCTL）***中 LXTALDRI 位域描述； 24. 修改***图 25-3. 分频为 16 时计数器时钟***，***图 25-9. 当预分频系数为 64 时，复位事件重新同步***的题注； 25. ***25.4.7 故障输入***章节中新增小节***故障恢复***； 26. ***25.4.10. 双通道触发模式***章节中增加“注意：”,说明斜坡补偿的适用范围和限制； 27. ***25.4.7 故障输入***章节中，修改关于 FLTxRST 的描述； 28. ***25.4.7. 故障输入***章节的第二段修改为“发生故障事件时，输出（STxCHy_O， x = 0..7， y = 0,1）为预置的电平”； 29. 修改 ***25.4.7. 故障输入***章节***表 25-23. 故障通道复位源***； 30. 更新***图 25-52. 当 FLTxRST 为 1，FLTRECCTL 为 1 和 FLTxCNT[3:0] = 0x03 时故障计数器***内文字描述； 31. ADC 章节***采样保持模式***中只能使用 IRC32K 时钟，	

		删掉 LXTAL 时钟； 32. 修改 FFT 章节***控制和状态寄存器（FFT_CSR）***中 IMSEL[1:0]位域描述； 33. 修改 PMU 章节***图 3-6. BOR 波形图***，***图 3-10. VUVD 波形***； 34. 修改 TIMER 章节***图 23-31. BREAK0 中止功能逻辑图***，***图 23-115. BREAK0 中止功能逻辑图***，***图 23-137. BREAK0 中止功能逻辑图***； 35. 修改 I2C 章节***从机地址寄存器 1（I2C_SADDR1）*** ADDMSK2[2:0] 位域描述和***控制寄存器 2（I2C_CTL2）***复位值； 36. FMC 章节***单 bank 模式（DBS=0）***的“注意：”中增加描述；	
1.3		1. 修改***1.7.8. 系统状态寄存器（SYSCFG_STAT）***中 TCMSRAMECCMEIF、SRAM1ECCSEIF、SRAM1ECCMEIF 位域描述； 2. ***2. 闪存控制器（FMC）***章节，修改为系统电源上下电复位时，选项字节修改上传； 3. ***2.3.1. 闪存结构***章节中增加描述：BANK0 和 BANK1 地址是连续的； 4. ***2.3.10. 选项字节***章节***表 2-4. 选项字节***中地址 0x1fff 783e 修改为 0x1fff 782e； 5. ***2.3.10. 选项字节***章节增加***表 2-5. 出厂值***； 6. 修改***3.3.2. VDD/VDDA 电源域***章节中 VREFN 和 VREFP 相关的描述； 7. 1.2V 域修改为 V_{CORE} 域； 8. ***4.3.17. AHB1 睡眠和深度睡眠使能寄存器（RCU_AHB1SPDPEN）***中删除 FMCSPDPEN 比特位； 9. ***16.2.3. JTAG***章节中 BSD JTAG TAP 修改为 MCU JTAG TAP； 10. 修改***17.7.26. 同步控制寄存器（ADC_SYNCCTL）***中 ADCCK 位域描述； 11. 修改***表 18-2. DAC 触发与输出***； 12. ***21.1.1. 独立看门狗定时器***章节***功能说明***中增加注意：两次连续喂狗之间需插入 7 个及以上 IRC32K 时钟间隔； 13. 删除***22.3.16. RTC 引脚配置***章节 shut down 模式； 14. 修改***23.1.5. 计数器初始控制寄存器（TIMERx_CINITCTL）***中 CINITDIR 位域描述；	2026 年 1 月 27 日

	15. 修改表 ***23-9. 不同译码器模式下的计数方向***; 16. ***29.3.10. 通信参数***章节中，有效跳变沿修改为从隐性位到显性位； 17. 修改***表 25-6. 计数器和捕获 y(y=0,1)寄存器限值***，并将 ***25. 高分辨率定时器（HRTIMER）***章节中所有 CNTCKDIV[3：0]改成 CNTCKDIV[2：0]； 18. 修改 ***25.4.2. Slave_TIMER(x=0..7)***章节中***仲裁机制***描述； 19. 修改 ***25.4.2. Slave_TIMER(x=0..7)***章节中***交换模式*** EXCx = 1 时的功能描述； 20. ***25.4.3. DLL 校准***章节中增加注意：仅当系统时钟频率在 150MHz 到 216MHz 时，DLL 校准有效。 21. ***HRTIMER 外部事件配置寄存器 1 (HRTIMER_EXEVCFG1)***中，EXEV9EG[1:0]修改为 EXEV7EG[1:0]； 22. 修改 ***HRTIMER DMA 更新 Slavex_TIMER 寄存器 (HRTIMER_DMAUPSTxR)(x=0..7)***中 STxCMP 和 STxDTCT 位域描述； 23. 修改 ***27.3.10. 串行红外（IrDA SIR）编解码功能模块***章节描述； 24. ***27.4.6. USART 接收超时寄存器（USART_RT）***中的 RT[23:0]位域增加备注描述。 25. ***27.4.8. USART 状态寄存器（USART_STAT）***中的 REA 位域增加备注描述。 26. 修改 ***30.3.8. CRC 功能***章节中关于 CRC 校验的注释。 27. 修改 ***34.3.10. 阈值监视器***章节中***阈值监视器快速模式***下的转换公式； 28. 修改***滤波器 y 极值监视器最大值寄存器（HPDF_FLTyEMMAX）***中 MAXVAL 位域描述。	

Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the “Product”), is owned by the Company according to the laws of the People’s Republic of China and other applicable laws. The Company reserves all rights under such laws and no Intellectual Property Rights are transferred (either wholly or partially) or licensed by the Company (either expressly or impliedly) herein. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

To the maximum extent permitted by applicable law, the Company makes no representations or warranties of any kind, express or implied, with regard to the merchantability and the fitness for a particular purpose of the Product, nor does the Company assume any liability arising out of the application or use of any Product. Any information provided in this document is provided only for reference purposes. It is the sole responsibility of the user of this document to determine whether the Product is suitable and fit for its applications and products planned, and properly design, program, and test the functionality and safety of its applications and products planned using the Product. The Product is designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only, and the Product is not designed or intended for use in (i) safety critical applications such as weapons systems, nuclear facilities, atomic energy controller, combustion controller, aeronautic or aerospace applications, traffic signal instruments, pollution control or hazardous substance management; (ii) life-support systems, other medical equipment or systems (including life support equipment and surgical implants); (iii) automotive applications or environments, including but not limited to applications for active and passive safety of automobiles (regardless of front market or aftermarket), for example, EPS, braking, ADAS (camera/fusion), EMS, TCU, BMS, BSG, TPMS, Airbag, Suspension, DMS, ICMS, Domain, ESC, DCDC, e-clutch, advanced-lighting, etc.. Automobile herein means a vehicle propelled by a self-contained motor, engine or the like, such as, without limitation, cars, trucks, motorcycles, electric cars, and other transportation devices; and/or (iv) other uses where the failure of the device or the Product can reasonably be expected to result in personal injury, death, or severe property or environmental damage (collectively "Unintended Uses"). Customers shall take any and all actions to ensure the Product meets the applicable laws and regulations. The Company is not liable for, in whole or in part, and customers shall hereby release the Company as well as its suppliers and/or distributors from, any claim, damage, or other liability arising from or related to all Unintended Uses of the Product. Customers shall indemnify and hold the Company, and its officers, employees, subsidiaries, affiliates as well as its suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Product.

Information in this document is provided solely in connection with the Product. The Company reserves the right to make changes, corrections, modifications or improvements to this document and the Product described herein at any time without notice. The Company shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2026 GigaDevice Semiconductor Inc. – All rights reserved