GigaDevice Semiconductor Inc.

GD32H73x_75x
Arm® Cortex®-M7 32-bit MCU

适用于 GD32H737xx, GD32H757xx, GD32H759xx, GD32H75Exx 系列

用户手册

1.8 版本

（2026 年 3 月）

目录

目录 2
图索引 39
表索引 52
1. 系统及存储器架构 59
1.1. Arm® Cortex®-M7 处理器 59
1.2. 系统架构 60
1.2.1. 总线矩阵区域 0 64
1.2.2. 总线矩阵区域 1 64
1.2.3. 总线矩阵区域 2 65
1.3. 存储器映射 65
1.3.1. 片上 SRAM 存储器 81
1.3.2. 片上 FLASH 存储器概述 83
1.4. 引导配置 83
1.5. 系统配置控制器（SYSCFG） 85
1.6. 定时器中止输入锁定 86
1.7. AXI 互连矩阵（AXIIM） 86
1.7.1. 主要特性 86
1.7.2. 功能说明 86
1.8. 系统配置寄存器（SYSCFG） 88
1.8.1. 外设模式配置寄存器（SYSCFG_PMCFG） 88
1.8.2. EXTI 源选择寄存器 0（SYSCFG_EXTISS0） 90
1.8.3. EXTI 源选择寄存器 1（SYSCFG_EXTISS1） 91
1.8.4. EXTI 源选择寄存器 2（SYSCFG_EXTISS2） 92
1.8.5. EXTI 源选择寄存器 3（SYSCFG_EXTISS3） 93
1.8.6. 锁定控制寄存器（SYSCFG_LKCTL） 95
1.8.7. I/O 补偿控制寄存器（SYSCFG_CPSCTL） 96
1.8.8. I/O 补偿单元代码配置寄存器（SYSCFG_CPSCCCFG） 97
1.8.9. TIMER 输入选择寄存器 0（SYSCFG_TIMERCISEL0） 97
1.8.10. TIMER 输入选择寄存器 1（SYSCFG_TIMERCISEL1） 99
1.8.11. TIMER 输入选择寄存器 2（SYSCFG_TIMERCISEL2） 100
1.8.12. TIMER 输入选择寄存器 3（SYSCFG_TIMERCISEL3） 101
1.8.13. TIMER 输入选择寄存器 4（SYSCFG_TIMERCISEL4） 102
1.8.14. TIMER 输入选择寄存器 5（SYSCFG_TIMERCISEL5） 103
1.8.15. TIMER 输入选择寄存器 6（SYSCFG_TIMERCISEL6） 105
1.8.16. CPU ICACHE 错误状态寄存器（SYSCFG_CPUICAC） 107
1.8.17. CPU DCACHE 错误状态寄存器（SYSCFG_CPUDCAC） 107
1.8.18. FPU 中断使能寄存器（SYSCFG_FPUINTEN） 108

1.8.19. SRAM 配置寄存器 0（SYSCFG_SRAMCFG0） 109
1.8.20. SRAM 配置寄存器 1（SYSCFG_SRAMCFG1） 109
1.8.21. TIMERx 配置寄存器 0（SYSCFG_TIMERxCFG0, x=0, 7） 110
1.8.22. TIMERx 配置寄存器 1（SYSCFG_TIMERxCFG1, x=0, 7） 112
1.8.23. TIMERx 配置寄存器 2（SYSCFG_TIMERxCFG2, x=0, 7） 114
1.8.24. TIMERx 配置寄存器 0（SYSCFG_TIMERxCFG0, x=1, 2, 3, 4, 22, 23, 30, 31） 115
1.8.25. TIMERx 配置寄存器 1（SYSCFG_TIMERxCFG1, x=1, 2, 3, 4, 22, 23, 30, 31） 117
1.8.26. TIMERx 配置寄存器 2（SYSCFG_TIMERxCFG2, x=1, 2, 3, 4, 22, 23, 30, 31） 119
1.8.27. TIMERx 配置寄存器 0（SYSCFG_TIMERxCFG0, x=14, 40, 41, 42, 43, 44） 120
1.8.28. TIMERx 配置寄存器 1（SYSCFG_TIMERxCFG1, x=14, 40, 41, 42, 43, 44） 122
1.8.29. TIMERx 配置寄存器 2（SYSCFG_TIMERxCFG2, x=14, 40, 41, 42, 43, 44） 123
1.8.30. 用户配置寄存器（SYSCFG_USERCFG） 124

1.9. AXI 互联寄存器 125
1.9.1. AXI 外设 ID4 寄存器（AXI_PERIPH_ID4） 125
1.9.2. AXI 外设 ID0 寄存器（AXI_PERIPH_ID0） 125
1.9.3. AXI 外设 ID1 寄存器（AXI_PERIPH_ID1） 126
1.9.4. AXI 外设 ID2 寄存器（AXI_PERIPH_ID2） 126
1.9.5. AXI 外设寄存器 ID3 寄存器（AXI_PERIPH_ID3） 127
1.9.6. AXI 组件 ID0 寄存器（AXI_COMP_ID0） 127
1.9.7. AXI 组件 ID1 寄存器（AXI_COMP_ID1） 128
1.9.8. AXI 组件 ID2 寄存器（AXI_COMP_ID2） 128
1.9.9. AXI 组件 ID3 寄存器（AXI_COMP_ID3） 128
1.9.10. AXI 主端口 x 总线矩阵发布功能控制寄存器（AXI_MPxBM_ISS_CTL） 129
1.9.11. AXI 主端口 x 总线矩阵功能控制寄存器（AXI_MPxBM_CTL） 129
1.9.12. AXI 主端口 x 长突发功能控制寄存器（AXI_MPx_LB_CTL） 130
1.9.13. AXI 主端口 x 发布功能控制寄存器（AXI_MPx_ISS_CTL） 130
1.9.14. AXI 从端口 x 功能控制寄存器（AXI_SPx_CTL） 131
1.9.15. AXI 从端口 x AHB 发布功能控制寄存器（AXI_SPx_AHBISS_CTL） 131
1.9.16. AXI 从端口 x 读 QOS 控制寄存器（AXI_SPx_RDQOS_CTL） 132
1.9.17. AXI 从端口 x 写 QOS 控制寄存器（AXI_SPx_WRQOS_CTL） 132
1.9.18. AXI 从端口 x 发布功能控制寄存器（AXI_SPx_ISS_CTL） 133

1.10. 设备电子签名 133
1.10.1. 存储容量信息 133
1.10.2. 设备唯一 ID（96 位） 134

2. RAM ECC 监视器单元（RAMECCMU） 136

2.1. 主要特性 136

2.2. 功能描述 136

2.3. RAMECCMU 寄存器 138
2.3.1. RAMECCMU 全局中断寄存器（RAMECCMU_INT） 138
2.3.2. RAMECCMU 监视器 x 控制寄存器（RAMECCMU_MxCTL） 138
2.3.3. RAMECCMU 监视器 x 状态寄存器（RAMECCMU_MxSTAT） 139
2.3.4. RAMECCMU 监视器 x 故障地址寄存器（RAMECCMU_MxFADDR） 140

2.3.5. RAMECCMU 监视器 x 故障数据低位寄存器（RAMECCMU_MxFDL） 140
2.3.6. RAMECCMU 监视器 x 故障数据高位寄存器（RAMECCMU_MxFDH） 141
2.3.7. RAMECCMU 监视器 x 故障 ECC 错误代码寄存器（RAMECCMU_MxFECODE） 141
3. 闪存控制器（FMC） 142
3.1. 简介 142
3.2. 主要特性 142
3.3. 功能说明 142
3.3.1. 闪存结构 142
3.3.2. 读操作 143
3.3.3. FMC_CTL/FMC_OBCTL 寄存器解锁 144
3.3.4. 扇区擦除 144
3.3.5. 整片擦除 145
3.3.6. 主存储闪存块编程 149
3.3.7. 选项字节 150
3.3.8. 扇区擦除/编程保护 153
3.3.9. 安全保护 154
3.3.10. DCRP 区域 155
3.3.11. 安全用户区域 157
3.3.12. 安全模式 158
3.3.13. 基础安全服务 159
3.3.14. 错误描述 160
3.3.15. FMC 中断 162
3.4. FMC 寄存器 163
3.4.1. 解锁寄存器（FMC_KEY） 163
3.4.2. 选项字节操作解锁寄存器（FMC_OBKEY） 163
3.4.3. 控制寄存器（FMC_CTL） 163
3.4.4. 状态寄存器（FMC_STAT） 165
3.4.5. 地址寄存器（FMC_ADDR） 166
3.4.6. 选项字节控制寄存器（FMC_OBCTL） 167
3.4.7. 选项字节状态寄存器 0（FMC_OBSTAT0_EFT） 167
3.4.8. 选项字节状态寄存器 0（FMC_OBSTAT0_MDF） 169
3.4.9. DCRP 地址寄存器（FMC_DCRPADDR_EFT） 170
3.4.10. DCRP 地址寄存器（FMC_DCRPADDR_MDF） 171
3.4.11. 安全用户区域地址寄存器（FMC_SCRADDR_EFT） 172
3.4.12. 安全用户区域地址寄存器（FMC_SCRADDR_MDF） 173
3.4.13. 擦除/编程保护寄存器（FMC_WP_EFT） 174
3.4.14. 擦除/编程保护寄存器（FMC_WP_MDF） 174
3.4.15. 引导装载地址寄存器（FMC_BTADDR_EFT） 175
3.4.16. 引导装载地址寄存器（FMC_BTADDR_MDF） 175
3.4.17. 选项字节状态寄存器 1（FMC_OBSTAT1_EFT） 176
3.4.18. 选项字节状态寄存器 1（FMC_OBSTAT1_MDF） 177
3.4.19. NO-RTDEC 区域寄存器（FMC_NODEC） 178

3.4.20. AES 初始向量寄存器 x（FMC_AESIVx_EFT）（x = 0…2） 178
3.4.21. AES 初始向量寄存器 x（FMC_AESIVx_MDF）（x = 0…2） 179
3.4.22. 产品 ID 寄存器 x（FMC_PIDx）（x = 0，1） 179
4. 熔丝（EFUSE） 181
4.1. 简介 181
4.2. 主要特性 181
4.3. 功能说明 181
4.3.1. 模块框图 181
4.3.2. 熔丝内容简介 182
4.3.3. 读操作 184
4.3.4. 写操作 185
4.3.5. AES 秘钥 CRC 功能 185
4.3.6. EFUSE 中断 185
4.4. EFUSE 寄存器 187
4.4.1. 控制寄存器（EFUSE_CTL） 187
4.4.2. 地址寄存器（EFUSE_ADDR） 188
4.4.3. 状态寄存器（EFUSE_STAT） 188
4.4.4. 状态标志清除寄存器（EFUSE_STATC） 189
4.4.5. 用户控制寄存器（EFUSE_USER_CTL） 190
4.4.6. MCU 保留寄存器（EFUSE_MCU_RSV） 192
4.4.7. 调试秘钥寄存器（EFUSE_DPx）（x = 0，1） 194
4.4.8. 固件 AES 秘钥寄存器 （EFUSE_AES_KEYx）（x = 0…3） 195
4.4.9. 用户数据寄存器 （EFUSE_USER_DATAx）（x = 0…3） 195
5. 电源管理单元（PMU） 197
5.1. 简介 197
5.2. 主要特征 197
5.3. 功能说明 197
5.3.1. 备份域 198
5.3.2. V_{DD} / V_{DDA} 电源域 200
5.3.3. V_{CORE} 电源域 204
5.3.4. USB 电源域 206
5.3.5. 省电模式 207
5.4. PMU 寄存器 209
5.4.1. 控制寄存器 0（PMU_CTL0） 209
5.4.2. 电源控制和状态寄存器（PMU_CS） 210
5.4.3. 控制寄存器 1（PMU_CTL1） 212
5.4.4. 控制寄存器 2（PMU_CTL2） 213
5.4.5. 控制寄存器 3（PMU_CTL3） 215
5.4.6. 参数寄存器（PMU_PAR） 215
6. 复位和时钟单元（RCU） 217

6.1. 复位控制单元（RCTL） 217
6.1.1. 简介 217
6.1.2. 功能说明 217
6.2. 时钟控制单元（CCTL） 218
6.2.1. 简介(适用于 GD32H737_757_759) 218
6.2.2. 简介(适用于 GD32H75E) 221
6.2.3. 主要特征 224
6.2.4. 功能说明 224
6.3. RCU 寄存器（适用于 GD32H737_757_759） 230
6.3.1. 控制寄存器（RCU_CTL） 230
6.3.2. PLL0 寄存器（RCU_PLL0） 232
6.3.3. 时钟配置寄存器 0（RCU_CFG0） 233
6.3.4. 时钟中断寄存器（RCU_INT） 235
6.3.5. AHB1 复位寄存器（RCU_AHB1RST） 239
6.3.6. AHB2 复位寄存器（RCU_AHB2RST） 240
6.3.7. AHB3 复位寄存器（RCU_AHB3RST） 241
6.3.8. AHB4 复位寄存器（RCU_AHB4RST） 243
6.3.9. APB1 复位寄存器（RCU_APB1RST） 244
6.3.10. APB2 复位寄存器（RCU_APB2RST） 248
6.3.11. APB3 复位寄存器（RCU_APB3RST） 251
6.3.12. APB4 复位寄存器（RCU_APB4RST） 251
6.3.13. AHB1 使能寄存器（RCU_AHB1EN） 252
6.3.14. AHB2 使能寄存器（RCU_AHB2EN） 254
6.3.15. AHB3 使能寄存器（RCU_AHB3EN） 256
6.3.16. AHB4 使能寄存器（RCU_AHB4EN） 257
6.3.17. APB1 使能寄存器（RCU_APB1EN） 259
6.3.18. APB2 使能寄存器（RCU_APB2EN） 262
6.3.19. APB3 使能寄存器（RCU_APB3EN） 265
6.3.20. APB4 使能寄存器（RCU_APB4EN） 266
6.3.21. AHB1 睡眠模式使能寄存器（RCU_AHB1SPEN） 267
6.3.22. AHB2 睡眠模式使能寄存器（RCU_AHB2SPEN） 269
6.3.23. AHB3 睡眠模式使能寄存器（RCU_AHB3SPEN） 270
6.3.24. AHB4 睡眠模式使能寄存器（RCU_AHB4SPEN） 272
6.3.25. APB1 睡眠模式使能寄存器（RCU_APB1SPEN） 274
6.3.26. APB2 睡眠模式使能寄存器（RCU_APB2SPEN） 277
6.3.27. APB3 睡眠模式使能寄存器（RCU_APB3SPEN） 280
6.3.28. APB4 睡眠模式使能寄存器（RCU_APB4SPEN） 281
6.3.29. 备份域控制寄存器（RCU_BDCTL） 281
6.3.30. 复位源/时钟寄存器（RCU_RSTSCK） 283
6.3.31. PLL 时钟附加控制寄存器（RCU_PLLADDCTL） 285
6.3.32. PLL1 寄存器（RCU_PLL1） 287
6.3.33. PLL2 寄存器（RCU_PLL2） 289
6.3.34. 时钟配置寄存器 1（RCU_CFG1） 290
6.3.35. 时钟配置寄存器 2（RCU_CFG2） 293

6.3.36. 时钟配置寄存器 3（RCU_CFG3） 295
6.3.37. PLL 控制寄存器（RCU_PLLALL） 297
6.3.38. PLL0 小数配置寄存器（RCU_PLL0FRA） 298
6.3.39. PLL1 小数配置寄存器（RCU_PLL1FRA） 299
6.3.40. PLL2 小数配置寄存器（RCU_PLL2FRA） 299
6.3.41. 附加时钟控制寄存器 0（RCU_ADDCTL0） 300
6.3.42. 附加时钟控制寄存器 1（RCU_ADDCTL1） 301
6.3.43. 附加时钟中断寄存器（RCU_ADDINT） 303
6.3.44. 时钟配置寄存器 4（RCU_CFG4） 304
6.3.45. USB 时钟控制寄存器（RCU_USBCLKCTL） 305
6.3.46. PLLUSB 时钟配置寄存器（RCU_PLLUSBCFG） 307
6.3.47. APB2 附加复位寄存器（RCU_ADDAPB2RST） 308
6.3.48. APB2 附加使能寄存器（RCU_ADDAPB2EN） 309
6.3.49. APB2 附加睡眠模式使能寄存器（RCU_ADDAPB2SPEN） 310
6.3.50. 时钟配置寄存器 5（RCU_CFG5） 310

6.1. RCU 寄存器（适用于 GD32H75E） 313
6.1.1. 控制寄存器（RCU_CTL） 313
6.1.2. PLL0 寄存器（RCU_PLL0） 315
6.1.3. 时钟配置寄存器 0（RCU_CFG0） 316
6.1.4. 时钟中断寄存器（RCU_INT） 318
6.1.5. AHB1 复位寄存器（RCU_AHB1RST） 322
6.1.6. AHB2 复位寄存器（RCU_AHB2RST） 323
6.1.7. AHB3 复位寄存器（RCU_AHB3RST） 324
6.1.8. AHB4 复位寄存器（RCU_AHB4RST） 325
6.1.9. APB1 复位寄存器（RCU_APB1RST） 326
6.1.10. APB2 复位寄存器（RCU_APB2RST） 329
6.1.11. APB3 复位寄存器（RCU_APB3RST） 332
6.1.12. APB4 复位寄存器（RCU_APB4RST） 333
6.1.13. AHB1 使能寄存器（RCU_AHB1EN） 333
6.1.14. AHB2 使能寄存器（RCU_AHB2EN） 335
6.1.15. AHB3 使能寄存器（RCU_AHB3EN） 336
6.1.16. AHB4 使能寄存器（RCU_AHB4EN） 337
6.1.17. APB1 使能寄存器（RCU_APB1EN） 338
6.1.18. APB2 使能寄存器（RCU_APB2EN） 341
6.1.19. APB3 使能寄存器（RCU_APB3EN） 344
6.1.20. APB4 使能寄存器（RCU_APB4EN） 345
6.1.21. AHB1 睡眠模式使能寄存器（RCU_AHB1SPEN） 346
6.1.22. AHB2 睡眠模式使能寄存器（RCU_AHB2SPEN） 347
6.1.23. AHB3 睡眠模式使能寄存器（RCU_AHB3SPEN） 348
6.1.24. AHB4 睡眠模式使能寄存器（RCU_AHB4SPEN） 349
6.1.25. APB1 睡眠模式使能寄存器（RCU_APB1SPEN） 351
6.1.26. APB2 睡眠模式使能寄存器（RCU_APB2SPEN） 354
6.1.27. APB3 睡眠模式使能寄存器（RCU_APB3SPEN） 357
6.1.28. APB4 睡眠模式使能寄存器（RCU_APB4SPEN） 357

6.1.29. 备份域控制寄存器（RCU_BDCTL） 358
6.1.30. 复位源/时钟寄存器（RCU_RSTSCK） 360
6.1.31. PLL 时钟附加控制寄存器（RCU_PLLADDCTL） 361
6.1.32. PLL1 寄存器（RCU_PLL1） 364
6.1.33. PLL2 寄存器（RCU_PLL2） 365
6.1.34. 时钟配置寄存器 1（RCU_CFG1） 367
6.1.35. 时钟配置寄存器 2（RCU_CFG2） 369
6.1.36. 时钟配置寄存器 3（RCU_CFG3） 371
6.1.37. PLL 控制寄存器（RCU_PLLALL） 372
6.1.38. PLL0 小数配置寄存器（RCU_PLL0FRA） 374
6.1.39. PLL1 小数配置寄存器（RCU_PLL1FRA） 374
6.1.40. PLL2 小数配置寄存器（RCU_PLL2FRA） 375
6.1.41. 附加时钟控制寄存器 0（RCU_ADDCTL0） 376
6.1.42. 附加时钟控制寄存器 1（RCU_ADDCTL1） 377
6.1.43. 附加时钟中断寄存器（RCU_ADDINT） 378
6.1.44. 时钟配置寄存器 4（RCU_CFG4） 380
6.1.45. USB 时钟控制寄存器（RCU_USBCLKCTL） 380
6.1.46. PLLUSB 时钟配置寄存器（RCU_PLLUSBCFG） 382
6.1.47. APB2 附加复位寄存器（RCU_ADDAPB2RST） 384
6.1.48. APB2 附加使能寄存器（RCU_ADDAPB2EN） 385
6.1.49. APB2 附加睡眠模式使能寄存器（RCU_ADDAPB2SPEN） 385
6.1.50. 时钟配置寄存器 5（RCU_CFG5） 386

7. 时钟校准控制器（CTC） 389

7.1. 简介 389

7.2. 主要特性 389

7.3. 功能描述 389

7.3.1. REF 同步脉冲发生器 390
7.3.2. CTC 校准计数器 390
7.3.3. 频率评估和自动校准过程 391
7.3.4. 软件编程指南 392

7.4. CTC 寄存器 393

7.4.1. 控制寄存器 0（CTC_CTL0） 393
7.4.2. 控制寄存器 1（CTC_CTL1） 394
7.4.3. 状态寄存器（CTC_STAT） 395
7.4.4. 中断清除寄存器（CTC_INTC） 397

8. 中断/事件控制器（EXTI） 398

8.1. 简介 398

8.2. 主要特征 398

8.3. 功能说明 398

8.4. 外部中断及事件（EXTI）框图 409

8.5. 外部中断及事件功能概述 409
8.6. EXTI 寄存器 413
8.6.1. 中断使能寄存器 0（EXTI_INTEN0） 413
8.6.2. 事件使能寄存器 0（EXTI_EVEN0） 413
8.6.3. 上升沿触发使能寄存器 0（EXTI_RTEN0） 413
8.6.4. 下降沿触发使能寄存器 0（EXTI_FTEN0） 414
8.6.5. 软件中断事件寄存器 0（EXTI_SWIEV0） 414
8.6.6. 挂起寄存器 0（EXTI_PD0） 415
8.6.7. 中断使能寄存器 1（EXTI_INTEN1） 415
8.6.8. 事件使能寄存器 1（EXTI_EVEN1） 416
8.6.9. 上升沿触发使能寄存器 1（EXTI_RTEN1） 416
8.6.10. 下降沿触发使能寄存器 1（EXTI_FTEN1） 416
8.6.11. 软件中断事件寄存器 1（EXTI_SWIEV1） 417
8.6.12. 挂起寄存器 1（EXTI_PD1） 417
9. 触发选择控制器（TRIGSEL） 419
9.1. 简介 419
9.2. 主要特征 419
9.3. 功能说明 419
9.4. 内部连接 420
9.5. TRIGSEL 寄存器 430
9.5.1. EXTOUT 触发选择寄存器 0（TRIGSEL_EXTOUT_0） 430
9.5.2. EXTOUT 触发选择寄存器 1（TRIGSEL_EXTOUT_1） 430
9.5.3. EXTOUT 触发选择寄存器 2（TRIGSEL_EXTOUT_2） 431
9.5.4. EXTOUT 触发选择寄存器 3（TRIGSEL_EXTOUT_3） 432
9.5.5. ADC0 触发选择寄存器（TRIGSEL_ADC0） 432
9.5.6. ADC1 触发选择寄存器（TRIGSEL_ADC1） 433
9.5.7. ADC2 触发选择寄存器（TRIGSEL_ADC2） 433
9.5.8. DAC0_OUT0 触发选择寄存器（TRIGSEL_DAC0OUT0） 434
9.5.9. DAC0_OUT1 触发选择寄存器（TRIGSEL_DAC0OUT1） 435
9.5.10. TIMER0_BRKIN 触发选择寄存器（TRIGSEL_TIMER0BRKIN） 435
9.5.11. TIMER7_BRKIN 触发选择寄存器（TRIGSEL_TIMER7BRKIN） 436
9.5.12. TIMER14_BRKIN 触发选择寄存器（TRIGSEL_TIMER14BRKIN） 437
9.5.13. TIMER15_BRKIN 触发选择寄存器（TRIGSEL_TIMER15BRKIN） 437
9.5.14. TIMER16_BRKIN 触发选择寄存器（TRIGSEL_TIMER16BRKIN） 438
9.5.15. TIMER40_BRKIN 触发选择寄存器（TRIGSEL_TIMER40BRKIN） 438
9.5.16. TIMER41_BRKIN 触发选择寄存器（TRIGSEL_TIMER41BRKIN） 439
9.5.17. TIMER42_BRKIN 触发选择寄存器（TRIGSEL_TIMER42BRKIN） 440
9.5.18. TIMER43_BRKIN 触发选择寄存器（TRIGSEL_TIMER43BRKIN） 440
9.5.19. TIMER44_BRKIN 触发选择寄存器（TRIGSEL_TIMER44BRKIN） 441
9.5.20. CAN0 触发选择寄存器（TRIGSEL_CAN0） 441
9.5.21. CAN1 触发选择寄存器（TRIGSEL_CAN1） 442
9.5.22. CAN2 触发选择寄存器（TRIGSEL_CAN2） 442

9.5.23. LPDTS 触发选择寄存器（TRIGSEL_LPDTS） 443
9.5.24. TIMER0_ETI 触发选择寄存器（TRIGSEL_TIMER0ETI） 444
9.5.25. TIMER1_ETI 触发选择寄存器（TRIGSEL_TIMER1ETI） 444
9.5.26. TIMER2_ETI 触发选择寄存器（TRIGSEL_TIMER2ETI） 445
9.5.27. TIMER3_ETI 触发选择寄存器（TRIGSEL_TIMER3ETI） 445
9.5.28. TIMER4_ETI 触发选择寄存器（TRIGSEL_TIMER4ETI） 446
9.5.29. TIMER7_ETI 触发选择寄存器（TRIGSEL_TIMER7ETI） 446
9.5.30. TIMER22_ETI 触发选择寄存器（TRIGSEL_TIMER22ETI） 447
9.5.31. TIMER23_ETI 触发选择寄存器（TRIGSEL_TIMER23ETI） 448
9.5.32. TIMER30_ETI 触发选择寄存器（TRIGSEL_TIMER30ETI） 448
9.5.33. TIMER31_ETI 触发选择寄存器（TRIGSEL_TIMER31ETI） 449
9.5.34. EDOUT 触发选择寄存器（TRIGSEL_EDOUT） 449
9.5.35. HPDF 触发选择寄存器（TRIGSEL_HPDF） 450
9.5.36. TIMER0_ITI14 触发选择寄存器（TRIGSEL_TIMER0ITI14） 450
9.5.37. TIMER1_ITI14 触发选择寄存器（TRIGSEL_TIMER1ITI14） 451
9.5.38. TIMER2_ITI14 触发选择寄存器（TRIGSEL_TIMER2ITI14） 452
9.5.39. TIMER3_ITI14 触发选择寄存器（TRIGSEL_TIMER3ITI14） 452
9.5.40. TIMER4_ITI14 触发选择寄存器（TRIGSEL_TIMER4ITI14） 453
9.5.41. TIMER7_ITI14 触发选择寄存器（TRIGSEL_TIMER7ITI14） 453
9.5.42. TIMER14_ITI14 触发选择寄存器（TRIGSEL_TIMER14ITI14） 454
9.5.43. TIMER22_ITI14 触发选择寄存器（TRIGSEL_TIMER22ITI14） 454
9.5.44. TIMER23_ITI14 触发选择寄存器（TRIGSEL_TIMER23ITI14） 455
9.5.45. TIMER30_ITI14 触发选择寄存器（TRIGSEL_TIMER30ITI14） 456
9.5.46. TIMER31_ITI14 触发选择寄存器（TRIGSEL_TIMER31ITI14） 456
9.5.47. TIMER40_ITI14 触发选择寄存器（TRIGSEL_TIMER40ITI14） 457
9.5.48. TIMER41_ITI14 触发选择寄存器（TRIGSEL_TIMER41ITI14） 457
9.5.49. TIMER42_ITI14 触发选择寄存器（TRIGSEL_TIMER42ITI14） 458
9.5.50. TIMER43_ITI14 触发选择寄存器（TRIGSEL_TIMER43ITI14） 458
9.5.51. TIMER44_ITI14 触发选择寄存器（TRIGSEL_TIMER44ITI14） 459
10. 通用和备用输入/输出接口（GPIO 和 AFIO） 460
10.1. 简介 460
10.2. 主要特征 460
10.3. 功能说明 460
10.3.1. GPIO 引脚配置 461
10.3.2. 外部中断及事件 462
10.3.3. 备用功能（AF） 462
10.3.4. 附加功能 462
10.3.5. 输入配置 462
10.3.6. 输出配置 463
10.3.7. 模拟配置 463
10.3.8. 备用功能（AF）配置 464
10.3.9. GPIO 锁定功能 465
10.3.10. GPIO 单周期输出翻转功能 465

10.3.11. I/O 补偿单元 465
10.3.12. ADC 的模拟配置 465
10.3.13. 输入滤波 466
10.4. GPIO 寄存器 469
10.4.1. 端口控制寄存器（GPIOx_CTL，x=A...H，J，K） 469
10.4.2. 端口输出模式寄存器（GPIOx_OMODE，x=A...H，J，K） 471
10.4.3. 端口输出速度寄存器（GPIOx_OSPD，x=A…H，J，K） 472
10.4.4. 端口上拉/下拉寄存器（GPIOx_PUD，x=A...H，J，K） 474
10.4.5. 端口输入状态寄存器（GPIOx_ISTAT，x=A…H，J，K） 476
10.4.6. 端口输出控制寄存器（GPIOx_OCTL，x=A...H，J，K） 476
10.4.7. 端口位操作寄存器（GPIOx_BOP，x=A...H，J，K） 477
10.4.8. 端口配置锁定寄存器（GPIOx_LOCK，x=A...H，J，K） 477
10.4.9. 备用功能选择寄存器 0（GPIOx_AFSEL0，x=A...H，J，K） 478
10.4.10. 备用功能选择寄存器 1（GPIOx_AFSEL1，x=A...H，J，K） 479
10.4.11. 位清除寄存器（GPIOx_BC，x=A...H，J，K） 480
10.4.12. 端口位翻转寄存器（GPIOx_TG，x=A…H，J，K） 481
10.4.13. 输入滤波寄存器（GPIOx_IFL，x=A…H，J，K） 481
10.4.14. 输入滤波类型寄存器（GPIOx_IFTP，x=A...H，J，K） 482
11. 循环冗余校验管理单元（CRC） 484
11.1. 简介 484
11.2. 主要特性 484
11.3. 功能描述 485
11.4. CRC 寄存器 486
11.4.1. 数据寄存器（CRC_DATA） 486
11.4.2. 独立数据寄存器（CRC_FDATA） 486
11.4.3. 控制寄存器（CRC_CTL） 487
11.4.4. 初值寄存器（CRC_IDATA） 487
11.4.5. 多项式寄存器（CRC_POLY） 488
12. 真随机数生成器（TRNG） 489
12.1. 简介 489
12.2. 主要特征 489
12.3. 功能说明 489
12.3.1. 线性反馈移位寄存器（LFSR） 490
12.3.2. 后处理 490
12.3.3. 训练单元 490
12.3.4. 输出 FIFO 490
12.3.5. 健康测试 491
12.3.6. NIST 模式状态 492
12.3.7. 操作流程 492
12.3.8. 错误标志 493

12.3.9. 低功耗配置 493
12.4. TRNG 寄存器 494
12.4.1. 控制寄存器（TRNG_CTL） 494
12.4.2. 状态寄存器（TRNG_STAT） 496
12.4.3. 数据寄存器（TRNG_DATA） 497
12.4.4. 健康测试配置寄存器（TRNG_HTCFG） 497
13. 加密处理器（CAU） 498
13.1. 简介 498
13.2. 主要特征 498
13.3. CAU 数据类型和初始化向量 499
13.3.1. 数据类型 499
13.3.2. 初始化向量 500
13.4. 加密处理器流程 500
13.4.1. DES / TDES 加密处理流程 501
13.4.2. AES 加密处理流程 505
13.5. 操作模式 512
13.6. CAU DMA 接口 513
13.7. CAU 中断 513
13.8. CAU 挂起模式 514
13.9. CAU 寄存器 516
13.9.1. 控制寄存器（CAU_CTL） 516
13.9.2. 状态寄存器 0 (CAU_STAT0) 518
13.9.3. 数据输入寄存器 (CAU_DI) 518
13.9.4. 数据输出寄存器 (CAU_DO) 519
13.9.5. DMA 使能寄存器 (CAU_DMAEN) 519
13.9.6. 中断使能寄存器 (CAU_INTEN) 520
13.9.7. 状态寄存器 1 (CAU_STAT1) 520
13.9.8. 中断标志寄存器 (CAU_INTF) 521
13.9.9. 密钥寄存器 (CAU_KEY0..3(H/L)) 521
13.9.10. 初始化向量寄存器 (CAU_IV0..1(H/L)) 524
13.9.11. GCM 或 CCM 模式上下文交换寄存器 x (CAU_GCMCCMCTXSx) (x=0..7) 525
13.9.12. GCM 模式上下文交换寄存器 x (CAU_GCMCTXSx) (x=0..7) 526
14. 哈希处理器（HAU） 527
14.1. 简介 527
14.2. 主要特征 527
14.3. 数据类型 527
14.4. HAU 内核 529
14.4.1. 自动数据填充 529

14.4.2. 摘要计算 530
14.4.3. 哈希模式 530
14.4.4. HMAC 模式 531
14.5. HAU 挂起模式 531
14.5.1. 通过 CPU 加载数据 531
14.5.2. 通过 DMA 加载数据 532
14.6. HAU 中断 532
14.6.1. 输入 FIFO 中断 532
14.6.2. 计算完成中断 532
14.7. HAU 寄存器 533
14.7.1. 控制寄存器（HAU_CTL） 533
14.7.2. 数据输入寄存器（HAU_DI） 534
14.7.3. 配置寄存器 (HAU_CFG) 535
14.7.4. 数据输出寄存器 (HAU_DO0..7) 535
14.7.5. 中断使能寄存器 (HAU_INTEN) 538
14.7.6. 状态与标志寄存器 (HAU_STAT) 538
14.7.7. 上下文交换寄存器 x (HAU_CTXSx) (x=0...53) 539
15. 三角函数加速器（TMU） 540
15.1. 简介 540
15.2. 主要特性 540
15.3. 结构框图 540
15.4. 功能描述 541
15.4.1. 数据格式和配置 541
15.4.2. 模式配置 541
15.4.3. TMU 精度 549
15.4.4. TMU 运算挂起 550
15.4.5. 零开销 550
15.4.6. 中断和 DMA 551
15.5. TMU 寄存器 551
15.5.1. 控制和状态寄存器（TMU_CS） 551
15.5.2. 输入数据寄存器（TMU_IDATA） 553
15.5.3. 输出数据寄存器（TMU_ODATA） 554
16. 直接存储器访问控制器（DMA） 555
16.1. 简介 555
16.2. 主要特征 555
16.3. 功能说明 556
16.3.1. 结构框图 556
16.3.2. 外设握手 557
16.3.3. 数据处理 558

16.3.4. 地址生成 562
16.3.5. 循环模式 563
16.3.6. 存储切换模式 563
16.3.7. 传输操作 563
16.3.8. 传输完成 564
16.3.9. 通道配置 565
16.4. 中断 566
16.4.1. 标志 566
16.4.2. 异常 567
16.4.3. 错误 567
16.4.4. DMA 请求映射 569
16.5. DMA 寄存器 570
16.5.1. 中断标志位寄存器 0（DMA_INTF0） 570
16.5.2. 中断标志位寄存器 1（DMA_INTF1） 571
16.5.3. 中断标志位清除寄存器 0（DMA_INTC0） 572
16.5.4. 中断标志位清除寄存器 1（DMA_INTC1） 572
16.5.5. 通道 x 控制寄存器（DMA_CHxCTL） 573
16.5.6. 通道 x 计数寄存器（DMA_CHxCNT） 576
16.5.7. 通道 x 外设基地址寄存器（DMA_CHxPADDR） 577
16.5.8. 通道 x 存储器 0 基地址寄存器（DMA_CHxM0ADDR） 577
16.5.9. 通道 x 存储器 1 基地址寄存器（DMA_CHxM1ADDR） 578
16.5.10. 通道 x FIFO 控制寄存器（DMA_CHxFCTL） 578
17. 主机直接存储器访问控制器（MDMA） 580
17.1. 简介 580
17.2. 主要特征 580
17.3. 功能说明 581
17.3.1. 数据处理 583
17.3.2. 地址生成 585
17.3.3. 传输模式 586
17.3.4. 传输状态 588
17.3.5. MDMA 错误和中断 589
17.4. MDMA 寄存器 591
17.4.1. 全局中断标志寄存器（MDMA_GINTF） 591
17.4.2. 通道 x 状态寄存器 0（MDMA_CHxSTAT0） 591
17.4.3. 通道 x 状态清除寄存器（MDMA_CHxSTATC） 592
17.4.4. 通道 x 状态寄存器 1（MDMA_CHxSTAT1） 593
17.4.5. 通道 x 控制寄存器 0（MDMA_CHxCTL0） 594
17.4.6. 通道 x 配置寄存器（MDMA_CHxCFG） 596
17.4.7. 通道 x 块传输配置寄存器（MDMA_CHxBTCFG） 599
17.4.8. 通道 x 源地址寄存器（MDMA_CHxSADDR） 600
17.4.9. 通道 x 目的地址寄存器（MDMA_CHxDADDR） 600
17.4.10. 通道 x 多块地址更新寄存器（MDMA_CHxMBADDRU） 600

17.4.11. 通道 x 链路地址寄存器（MDMA_CHxLADDR） 601
17.4.12. 通道 x 控制寄存器 1（MDMA_CHxCTL1） 602
17.4.13. 通道 x 掩码地址寄存器（MDMA_CHxMADDR） 602
17.4.14. 通道 x 掩码数据寄存器（MDMA_CHxMDATA） 603
18. DMA 请求多路复用器（DMAMUX） 604
18.1. 简介 604
18.2. 主要特征 604
18.3. 结构框图 605
18.4. 信号描述 605
18.5. 功能说明 606
18.5.1. DMAMUX 请求路由器 606
18.5.2. DMAMUX 请求生成器 608
18.5.3. 通道配置 609
18.5.4. 中断 609
18.5.5. DMAMUX 映射 610
18.6. DMAMUX 寄存器 618
18.6.1. 请求路由通道 x 配置寄存器（DMAMUX_RM_CHxCFG） 618
18.6.2. 请求路由通道中断标志位寄存器（DMAMUX_RM_INTF） 619
18.6.3. 请求路由通道中断标志位清除寄存器（DMAMUX_RM_INTC） 619
18.6.4. 请求生成通道 x 配置寄存器（DMAMUX_RG_CHxCFG） 620
18.6.5. 请求生成通道中断标志位寄存器（DMAMUX_RG_INTF） 620
18.6.6. 请求生成通道中断标志位清除寄存器（DMAMUX_RG_INTC） 621
19. 调试（DBG） 622
19.1. 简介 622
19.2. JTAG/SW 功能描述 622
19.2.1. 切换 JTAG / SW 接口 622
19.2.2. 引脚分配 622
19.2.3. JTAG 623
19.2.4. 调试复位 624
19.2.5. JEDEC-106 ID code 625
19.3. 调试保持功能描述 625
19.3.1. 低功耗模式调试支持 625
19.3.2. TIMER, I2C, RTC, WWDGT, FWDGT 和 CAN 外设调试支持 625
19.4. DBG 寄存器 626
19.4.1. ID 寄存器（DBG_ID） 626
19.4.2. 控制寄存器 0（DBG_CTL0） 626
19.4.3. 控制寄存器 1（DBG_CTL1） 627
19.4.4. 控制寄存器 2（DBG_CTL2） 628
19.4.5. 控制寄存器 3（DBG_CTL3） 630
19.4.6. 控制寄存器 4（DBG_CTL4） 631

20. 模数转换器（ADC） 633
20.1. 简介 633
20.2. 主要特征 633
20.3. 引脚和内部信号 634
20.4. 功能描述 635
20.4.1. 前置校准功能 635
20.4.2. 双时钟域架构 636
20.4.3. ADCON 使能 636
20.4.4. 单端和差分输入通道 636
20.4.5. 常规序列 637
20.4.6. 运行模式 637
20.4.7. 转换结果阈值监测功能 640
20.4.8. 数据存储模式 640
20.4.9. 采样时间配置 641
20.4.10. 外部触发配置 641
20.4.11. DMA 请求 641
20.4.12. 溢出检测 642
20.4.13. ADC 内部通道 642
20.4.14. 电池电压检测电路 643
20.4.15. 使用 HPDF 管理转换结果 643
20.4.16. 可编程分辨率（DRES） 644
20.4.17. 片上硬件过采样 644
20.5. ADC 同步模式 646
20.5.1. 独立模式 647
20.5.2. 常规并行模式 647
20.5.3. 常规跟随模式 648
20.5.4. 在 ADC 同步模式中使用 DMA 648
20.6. 中断 649
20.7. ADC 寄存器 650
20.7.1. 状态寄存器（ADC_STAT） 650
20.7.2. 控制寄存器 0（ADC_CTL0） 651
20.7.3. 控制寄存器 1（ADC_CTL1） 653
20.7.4. 看门狗高阈值寄存器 0（ADC_WDHT0） 655
20.7.5. 看门狗低阈值寄存器 0（ADC_WDLT0） 656
20.7.6. 常规序列寄存器 0（ADC_RSQ0） 656
20.7.7. 常规序列寄存器 1（ADC_RSQ1） 657
20.7.8. 常规序列寄存器 2（ADC_RSQ2） 658
20.7.9. 常规序列寄存器 3（ADC_RSQ3） 659
20.7.10. 常规序列寄存器 4（ADC_RSQ4） 660
20.7.11. 常规序列寄存器 5（ADC_RSQ5） 661
20.7.12. 常规序列寄存器 6（ADC_RSQ6） 662
20.7.13. 常规序列寄存器 7（ADC_RSQ7） 663

20.7.14. 常规序列寄存器 8（ADC_RSQ8） 664
20.7.15. 常规数据寄存器（ADC_RDATA） 665
20.7.16. 过采样控制寄存器（ADC_OVSAMPCTL） 665
20.7.17. 看门狗 1 通道选择寄存器（ADC_WD1SR） 666
20.7.18. 看门狗 2 通道选择寄存器（ADC_WD2SR） 667
20.7.19. 看门狗 1 高阈值寄存器（ADC_WDHT1） 668
20.7.20. 看门狗 1 低阈值寄存器（ADC_WDLT1） 668
20.7.21. 看门狗 2 高阈值寄存器（ADC_WDHT2） 668
20.7.22. 看门狗 2 低阈值寄存器（ADC_WDLT2） 669
20.7.23. 差分模式控制寄存器（ADC_DIFCTL） 669
20.7.24. 摘要状态寄存器（ADC_SSTAT） 670
20.7.25. 同步控制寄存器（ADC_SYNCCTL） 671
20.7.26. 同步常规数据寄存器 0（ADC_SYNCDATA0） 673
20.7.27. 同步常规数据寄存器 1（ADC_SYNCDATA1） 673

21. 数模转换器（DAC） 674

21.1. 简介 674

21.2. 主要特性 674

21.3. 功能描述 675

21.3.1. DAC 使能 675
21.3.2. DAC 输出缓冲 676
21.3.3. DAC 数据配置 676
21.3.4. DAC 触发 676
21.3.5. DAC 转换 676
21.3.6. DAC 噪声波 676
21.3.7. DAC 输出电压 677
21.3.8. DMA 请求 678
21.3.9. DAC 并发转换 678
21.3.10. DAC 输出缓冲区校准 678
21.3.11. DAC 模式 679
21.3.12. DAC 低功耗模式 680

21.4. DAC 寄存器 682

21.4.1. DACx 控制寄存器 (DAC_CTL0) 682
21.4.2. DACx 软件触发寄存器 (DAC_SWT) 684
21.4.3. DAC_OUT0 12 位右对齐数据保持寄存器 (DAC_OUT0_R12DH) 685
21.4.4. DAC_OUT0 12 位左对齐数据保持寄存器 (DAC_OUT0_L12DH) 685
21.4.5. DAC_OUT0 8 位右对齐数据保持寄存器 (DAC_OUT0_R8DH) 686
21.4.6. DAC_OUT1 12 位右对齐数据保持寄存器 (DAC_OUT1_R12DH) 686
21.4.7. DAC_OUT1 12 位左对齐数据保持寄存器 (DAC_OUT1_L12DH) 687
21.4.8. DAC_OUT1 8 位右对齐数据保持寄存器 (DAC_OUT1_R8DH) 687
21.4.9. DAC 并发模式 12 位右对齐数据保持寄存器 (DACC_R12DH) 687
21.4.10. DAC 并发模式 12 位左对齐数据保持寄存器 (DACC_L12DH) 688
21.4.11. DAC 并发模式 8 位右对齐数据保持寄存器 (DACC_R8DH) 688

21.4.12. DAC_OUT0 数据输出寄存器 (DAC_OUT0_DO) 689
21.4.13. DAC_OUT1 数据输出寄存器 (DAC_OUT1_DO) 689
21.4.14. DAC 状态寄存器 0 (DAC_STAT0) 690
21.4.15. DAC 校准寄存器 (DAC_CALR) 691
21.4.16. DAC 模式寄存器 (DAC_MDCR) 691
21.4.17. DAC 采样保持模式采样时间寄存器 0 (DAC_SKSTR0) 692
21.4.18. DAC 采样保持模式采样时间寄存器 1 (DAC_SKSTR1) 693
21.4.19. DAC 采样保持模式保持时间寄存器 (DAC_SKKTR) 693
21.4.20. DAC 采样保持模式刷新时间寄存器 (DAC_SKRTR) 693

22. 看门狗定时器（WDGT） 695

22.1. 独立看门狗定时器（FWDGT） 695
22.1.1. 简介 695
22.1.2. 主要特征 695
22.1.3. 功能说明 695
22.1.4. FWDGT 寄存器 698

22.2. 窗口看门狗定时器（WWDGT） 701
22.2.1. 简介 701
22.2.2. 主要特征 701
22.2.3. 功能说明 701
22.2.4. WWDGT 寄存器 703

23. 实时时钟（RTC） 705

23.1. 简介 705

23.2. 主要特征 705

23.3. 功能描述 706
23.3.1. 结构框图 706
23.3.2. 时钟源和预分频 707
23.3.3. 影子寄存器 707
23.3.4. 位域可屏蔽可配置的闹钟 707
23.3.5. 可配置周期的自动唤醒定时器 708
23.3.6. RTC 初始化和配置 708
23.3.7. 读取日历 709
23.3.8. RTC 复位 710
23.3.9. RTC 移位功能 711
23.3.10. RTC 参考时钟检测 711
23.3.11. RTC 数字平滑校准 712
23.3.12. 时间戳功能 713
23.3.13. 侵入检测 713
23.3.14. 校准时钟输出 715
23.3.15. 闹钟输出 715
23.3.16. RTC 引脚配置 715
23.3.17. RTC 省电模式管理 716
23.3.18. RTC 中断 716

23.4. RTC 寄存器 717
23.4.1. 时间寄存器（RTC_TIME） 717
23.4.2. 日期寄存器（RTC_DATE） 717
23.4.3. 控制寄存器（RTC_CTL） 718
23.4.4. 状态寄存器（RTC_STAT） 721
23.4.5. 预分频寄存器（RTC_PSC） 722
23.4.6. 唤醒定时器寄存器（RTC_WUT） 723
23.4.7. 闹钟 0 时间日期寄存器（RTC_ALRM0TD） 724
23.4.8. 闹钟 1 时间日期寄存器（RTC_ALRM1TD） 725
23.4.9. 写保护钥匙寄存器（RTC_WPK） 726
23.4.10. 亚秒寄存器（RTC_SS） 726
23.4.11. 移位控制寄存器（RTC_SHIFTCTL） 727
23.4.12. 时间戳时间寄存器（RTC_TTS） 727
23.4.13. 时间戳日期寄存器（RTC_DTS） 728
23.4.14. 时间戳亚秒寄存器（RTC_SSTS） 729
23.4.15. 高精度频率补偿寄存器（RTC_HRFC） 729
23.4.16. 侵入寄存器（RTC_TAMP） 730
23.4.17. 闹钟 0 亚秒寄存器（RTC_ALRM0SS） 732
23.4.18. 闹钟 1 亚秒寄存器（RTC_ALRM1SS） 732
23.4.19. 配置寄存器（RTC_CFG） 733
23.4.20. 备份寄存器（RTC_BKPx）（x=0..31） 734

24. 定时器（TIMER） 735

24.1. 高级定时器（TIMERx, x=0, 7） 737
24.1.1. 简介 737
24.1.2. 主要特性 737
24.1.3. 结构框图 737
24.1.4. 功能描述 738
24.1.5. TIMERx 寄存器（x=0,7） 777

24.2. 通用定时器 L0（TIMERx, x=1,2,3,4,22,23,30,31） 839
24.2.1. 简介 839
24.2.2. 主要特性 839
24.2.3. 结构框图 840
24.2.4. 功能描述 840
24.2.5. TIMERx 寄存器（x=1/2/3/4/22/23/30/31） 864

24.3. 通用定时器 L3（TIMERx, x=14,40,41,42,43,44） 893
24.3.1. 简介 893
24.3.2. 主要特性 893
24.3.3. 结构框图 894
24.3.4. 功能描述 894
24.3.5. TIMERx 寄存器(x=14/40/41/42/43/44) 917

24.4. 通用定时器 L4（TIMERx, x=15,16） 947
24.4.1. 简介 947

24.4.2. 主要特性 947
24.4.3. 结构框图 948
24.4.4. 功能描述 948
24.4.5. TIMERx 寄存器（x=15/16） 964
24.5. 基本定时器（TIMERx, x=5,6,50,51） 988
24.5.1. 简介 988
24.5.2. 主要特性 988
24.5.3. 结构框图 988
24.5.4. 功能描述 988
24.5.5. TIMERx 寄存器（x=5/6/50/51） 992
25. 通用同步异步收发器（USART） 999
25.1. 简介 999
25.2. 主要特征 999
25.3. 功能说明 1000
25.3.1. USART 帧格式 1001
25.3.2. 波特率发生 1002
25.3.3. USART 发送器 1002
25.3.4. USART 接收器 1003
25.3.5. DMA 方式访问数据缓冲区 1005
25.3.6. 硬件流控制 1006
25.3.7. 多处理器通信 1007
25.3.8. LIN 模式 1008
25.3.9. 同步通信模式 1009
25.3.10. 串行红外（IrDA SIR）编解码功能模块 1010
25.3.11. 半双工通信模式 1011
25.3.12. 智能卡（ISO7816-3）模式 1011
25.3.13. ModBus 通信 1012
25.3.14. 接收 FIFO 1013
25.3.15. 发送 FIFO 1013
25.3.16. 从 Deepsleep 模式唤醒 1014
25.3.17. USART 中断 1014
25.4. USART 寄存器 1016
25.4.1. USART 控制寄存器 0（USART_CTL0） 1016
25.4.2. USART 控制寄存器 1（USART_CTL1） 1019
25.4.3. USART 控制寄存器 2（USART_CTL2） 1021
25.4.4. USART 波特率寄存器（USART_BAUD） 1024
25.4.5. USART 保护时间和预分频器寄存器（USART_GP） 1025
25.4.6. USART 接收超时寄存器（USART_RT） 1026
25.4.7. USART 请求寄存器（USART_CMD） 1026
25.4.8. USART 状态寄存器（USART_STAT） 1027
25.4.9. USART 中断标志清除寄存器（USART_INTC） 1031
25.4.10. USART 数据接收寄存器（USART_RDATA） 1032

25.4.11. USART 数据发送寄存器（USART_TDATA） 1033
25.4.12. USART 兼容性控制寄存器（USART_CHC） 1033
25.4.13. USART FIFO 控制和状态寄存器（USART_FCS） 1034
26. 内部集成电路总线接口（I2C） 1038
26.1. 简介 1038
26.2. 主要特征 1038
26.3. 功能说明 1038
26.3.1. 时钟要求 1039
26.3.2. I2C 通讯流程 1040
26.3.3. 噪声滤波器 1042
26.3.4. I2C 时序配置 1042
26.3.5. I2C 复位 1044
26.3.6. 数据传输 1045
26.3.7. I2C 从机模式 1046
26.3.8. I2C 主机模式 1051
26.3.9. SMBus 支持 1056
26.3.10. SMBus 模式 1058
26.3.11. 从省电模式唤醒 1060
26.3.12. DMA 模式下数据传输 1060
26.3.13. I2C 错误和中断 1060
26.3.14. I2C 调试模式 1061
26.4. I2C 寄存器 1062
26.4.1. 控制寄存器 0（I2C_CTL0） 1062
26.4.2. 控制寄存器 1（I2C_CTL1） 1064
26.4.3. 从机地址寄存器 0（I2C_SADDR0） 1066
26.4.4. 从机地址寄存器 1（I2C_SADDR1） 1067
26.4.5. 时序寄存器（I2C_TIMING） 1067
26.4.6. 超时寄存器（I2C_TIMEOUT） 1068
26.4.7. 状态寄存器（I2C_STAT） 1069
26.4.8. 状态清除寄存器（I2C_STATC） 1072
26.4.9. PEC 寄存器（I2C_PEC） 1073
26.4.10. 接收数据寄存器（I2C_RDATA） 1073
26.4.11. 发送数据寄存器（I2C_TDATA） 1074
26.4.12. 控制寄存器 2（I2C_CTL2） 1074
27. 串行外设接口/片上音频接口（SPI/I2S） 1075
27.1. 简介 1075
27.2. 主要特性 1075
27.2.1. SPI 主要特性 1075
27.2.2. I2S 主要特性 1075
27.3. SPI 功能说明 1076
27.3.1. SPI 结构框图 1076

27.3.2. SPI 信号线描述 1076
27.3.3. SPI 时序和数据帧格式 1078
27.3.4. SPI 时钟延迟模式 1079
27.3.5. RxFIFO 和 TxFIFO 1081
27.3.6. NSS 功能 1083
27.3.7. SPI 运行模式 1085
27.3.8. DMA 功能 1093
27.3.9. CRC 功能 1093
27.3.10. SPI 中断 1094
27.4. I2S 功能说明 1097
27.4.1. I2S 结构框图 1097
27.4.2. I2S 信号线描述 1097
27.4.3. I2S 音频标准 1097
27.4.4. I2S 时钟 1104
27.4.5. RxFIFO 和 TxFIFO 1105
27.4.6. 运行 1105
27.4.7. DMA 功能 1107
27.4.8. I2S 中断 1107
27.5. SPI / I2S 寄存器 1109
27.5.1. 控制寄存器 0（SPI_CTL0） 1109
27.5.2. 控制寄存器 1（SPI_CTL1） 1111
27.5.3. 配置寄存器 0（SPI_CFG0） 1112
27.5.4. 配置寄存器 1（SPI_CFG1） 1114
27.5.5. 中断寄存器（SPI_INT） 1116
27.5.6. 状态寄存器（SPI_STAT） 1117
27.5.7. 中断/状态标志清除寄存器（SPI_STATC） 1119
27.5.8. 数据发送寄存器（SPI_TDATA） 1120
27.5.9. 数据接收寄存器（SPI_RDATA） 1121
27.5.10. CRC 多项式寄存器（SPI_CRCPOLY） 1121
27.5.11. 发送 CRC 寄存器（SPI_TCRC） 1122
27.5.12. 接收 CRC 寄存器（SPI_RCRC） 1122
27.5.13. 下溢数据寄存器（SPI_URDATA） 1123
27.5.14. I2S 控制寄存器（SPI_I2SCTL） 1123
27.5.15. 四线 SPI 控制寄存器（SPI_QCTL） 1125
27.5.16. 接收时钟延迟寄存器（SPI_RXDLYCK） 1126
28. OSPI I/O 管理器（OSPIM） 1127
28.1. 简介 1127
28.2. 主要特征 1127
28.3. 功能说明 1127
28.3.1. OSPIM 结构框图 1127
28.3.2. OSPIM 矩阵 1128
28.4. OSPIM 寄存器 1128

28.4.1. 端口配置寄存器（OSPIM_PCFGx）（x = 0, 1） 1128
29. 八线 SPI 接口（OSPI） 1130
29.1. 简介 1130
29.2. 主要特征 1130
29.3. OSPI 功能描述 1130
29.3.1. OSPI 结构框图 1130
29.3.2. OSPI 常规命令模式 1133
29.4. 操作模式 1136
29.4.1. 间接模式 1136
29.4.2. 状态轮询模式 1137
29.4.3. 内存映射模式 1137
29.5. OSPI 配置 1138
29.5.1. OSPI 系统配置 1138
29.5.2. OSPI 器件配置 1138
29.5.3. OSPI 常规命令模式配置 1138
29.6. OSPI 延迟数据采样 1139
29.7. 繁忙状态 1139
29.8. 错误管理 1139
29.9. OSPI 中断 1140
29.10. OSPI 寄存器 1141
29.10.1. 控制寄存器（OSPI_CTL） 1141
29.10.2. 设备配置寄存器 0（OSPI_DCFG0） 1142
29.10.3. 设备配置寄存器 1（OSPI_DCFG1） 1143
29.10.4. 状态寄存器（OSPI_STAT） 1144
29.10.5. 状态清除寄存器（OSPI_STATC） 1145
29.10.6. 数据长度寄存器（OSPI_DTLEN） 1146
29.10.7. 地址寄存器（OSPI_ADDR） 1146
29.10.8. 数据寄存器（OSPI_DATA） 1147
29.10.9. 状态屏蔽寄存器（OSPI_STATMK） 1147
29.10.10. 状态匹配寄存器（OSPI_STATMATCH） 1148
29.10.11. 间隔寄存器（OSPI_INTERVAL） 1148
29.10.12. 传输配置寄存器（OSPI_TCFG） 1148
29.10.13. 时序配置寄存器（OSPI_TIMCFG） 1151
29.10.14. 指令寄存器（OSPI_INS） 1151
29.10.15. 交替字节寄存器（OSPI_ALTE） 1152
29.10.16. 回卷传输配置寄存器（OSPI_WPTCFG） 1152
29.10.17. 回卷时序配置寄存器（OSPI_WPTIMCFG） 1154
29.10.18. 回卷指令寄存器（OSPI_WPINS） 1155
29.10.19. 回卷交替字节寄存器（OSPI_WPALTE） 1155
29.10.20. 写入传输配置寄存器（OSPI_WTCFG） 1156

29.10.21. 写入时序配置寄存器（OSPI_WTIMCFG） 1158
29.10.22. 写入指令寄存器（OSPI_WINS） 1159
29.10.23. 写入交替字节寄存器（OSPI_WALTE） 1159
30. 时钟相位延迟模块（CPDM）（适用于 GD32H737_757_759） 1160
30.1. 简介 1160
30.2. 主要特征 1160
30.3. 功能说明 1160
30.3.1. 概述 1161
30.3.2. 操作流程 1161
30.4. CPDM 寄存器 1163
30.4.1. 控制寄存器（CPDM_CTL） 1163
30.4.2. 配置寄存器（CPDM_CFG） 1163
31. 数字摄像头接口（DCI）（适用于 GD32H737_757_759） 1165
31.1. 简介 1165
31.2. 主要特性 1165
31.3. 结构框图 1165
31.4. 信号描述 1166
31.5. 功能描述 1166
31.5.1. DCI 硬件同步模式 1166
31.5.2. 内嵌码同步模式 1167
31.5.3. CCIR656 模式 1167
31.5.4. 用快照或连续捕获模式捕获数据 1169
31.5.5. 窗口功能 1169
31.5.6. 像素格式，数据填充和 DMA 接口 1170
31.6. 状态、错误和中断 1170
31.7. DCI 寄存器 1172
31.7.1. 控制寄存器（DCI_CTL） 1172
31.7.2. 状态寄存器 0（DCI_STAT0） 1173
31.7.3. 状态寄存器 1（DCI_STAT1） 1174
31.7.4. 中断使能寄存器（DCI_INTEN） 1175
31.7.5. 中断标志寄存器（DCI_INTF） 1176
31.7.6. 中断标志清除寄存器（DCI_INTC） 1177
31.7.7. 同步码寄存器（DCI_SC） 1178
31.7.8. 同步码屏蔽寄存器（DCI_SCUMSK） 1178
31.7.9. 剪裁窗口开始位置寄存器（DCI_CWSPOS） 1179
31.7.10. 剪裁窗口大小寄存器（DCI_CWSZ） 1179
31.7.11. 数据寄存器（DCI_DATA） 1180
32. TFT-LCD 接口（TLI）（适用于 GD32H737_757_759） 1181
32.1. 简介 1181

32.2. 主要特点 1181
32.3. 结构框图 1181
32.4. 信号线描述 1182
32.5. 功能描述 1182
32.5.1. LCD 显示时序 1182
32.5.2. 像素 DMA 功能 1183
32.5.3. 像素格式 1183
32.5.4. 层窗口和混合功能 1184
32.5.5. Layer 配置重载 1185
32.5.6. 抖动 1185
32.6. 中断 1185
32.7. TLI 寄存器 1187
32.7.1. 同步脉冲宽度寄存器（TLI_SPSZ） 1187
32.7.2. 后沿宽度寄存器（TLI_BPSZ） 1187
32.7.3. 有效宽度寄存器（TLI_ASZ） 1188
32.7.4. 总宽度寄存器（TLI_TSZ） 1188
32.7.5. 控制寄存器（TLI_CTL） 1189
32.7.6. 重载层配置寄存器（TLI_RL） 1190
32.7.7. 背景色配置寄存器（TLI_BGC） 1190
32.7.8. 中断使能寄存器（TLI_INTEN） 1191
32.7.9. 中断标志寄存器（TLI_INTF） 1192
32.7.10. 中断标志清除寄存器（TLI_INTC） 1192
32.7.11. 行标记寄存器（TLI_LM） 1193
32.7.12. 当前像素位置寄存器（TLI_CPPOS） 1193
32.7.13. 状态寄存器（TLI_STAT） 1194
32.7.14. 第 x 层控制寄存器（TLI_LxCTL）（x = 0, 1） 1194
32.7.15. 第 x 层水平位置参数寄存器（TLI_LxHPOS）（x = 0, 1） 1195
32.7.16. 第 x 层垂直位置参数寄存器（TLI_LxVPOS）（x = 0, 1） 1195
32.7.17. 第 x 层色键值寄存器（TLI_LxCKEY）（x = 0, 1） 1196
32.7.18. 第 x 层像素格式寄存器（TLI_LxPPF）（x = 0, 1） 1196
32.7.19. 第 x 层恒定 Alpha 寄存器（TLI_LxSA）（x = 0, 1） 1197
32.7.20. 第 x 层默认颜色寄存器（TLI_LxDC）（x = 0, 1） 1197
32.7.21. 第 x 层混合寄存器（TLI_LxBLEND）（x = 0, 1） 1198
32.7.22. 第 x 层帧基地址寄存器（TLI_LxFBADDR）（x = 0, 1） 1199
32.7.23. 第 x 层行长度寄存器（TLI_LxFLLEN）（x = 0, 1） 1199
32.7.24. 第 x 层总行数寄存器（TLI_LxFTLN）（x = 0, 1） 1200
32.7.25. 第 x 层颜色查找表寄存器（TLI_LxLUT）（x = 0, 1） 1200
33. S/P 数字音频接口接收器（RSPDIF）（适用于 GD32H737_757_759） 1201
33.1. 简介 1201
33.2. 主要特征 1201
33.3. 功能说明 1201

33.3.1. RSPDIF 结构框图 1201
33.3.2. S/PDIF 协议 1202
33.3.3. RSPDIF 1205
33.3.4. RSPDIF 同步过程 1207
33.3.5. RSPDIF 状态机 1209
33.3.6. RSPDIF 数据接收管理 1212
33.3.7. RSPDIF 时钟管理 1214
33.3.8. DMA 功能 1216
33.3.9. 状态、错误和中断 1216
33.4. RSPDIF 寄存器 1221
33.4.1. 控制寄存器（RSPDIF_CTL） 1221
33.4.2. 中断使能寄存器（RSPDIF_INTEN） 1223
33.4.3. 状态寄存器（RSPDIF_STAT） 1224
33.4.4. 状态清除寄存器（RSPDIF_STATC） 1226
33.4.5. 接收数据寄存器（RSPDIF_DATA） 1227
33.4.6. 接收通道状态寄存器（RSPDIF_CHSTAT） 1229
33.4.7. 接收阈值寄存器（RSPDIF_DTH） 1230
34. 串行音频接口（SAI）（适用于 GD32H737_757_759） 1231
34.1. 简介 1231
34.2. 主要特征 1231
34.3. 功能描述 1232
34.3.1. 模块框图 1232
34.3.2. 时钟分频器 1233
34.3.3. 操作模式 1234
34.3.4. 同步模式 1234
34.3.5. 帧配置 1235
34.3.6. Slot 配置 1237
34.3.7. 数据配置 1239
34.3.8. 同步 FIFO 1240
34.3.9. PDM 接口 1240
34.3.10. AC’97 链路控制器 1247
34.3.11. SPDIF 输出 1248
34.3.12. 立体声/单声道 1250
34.3.13. 静音 1250
34.3.14. 压缩扩展器 1251
34.3.15. 输出驱动 1253
34.3.16. IO 管理 1253
34.3.17. DMA 接口 1254
34.3.18. 使能/失能 1254
34.3.19. 错误标志位 1254
34.3.20. 中断 1256
34.4. SAI 寄存器 1258

34.4.1. 同步配置寄存器（SAI_SYNCFG） 1258
34.4.2. 子模块 x 配置寄存器 0（SAI_BxCFG0）（x = 0, 1） 1258
34.4.3. 子模块 x 配置寄存器 1（SAI_BxCFG1）（x = 0, 1） 1261
34.4.4. 子模块 x 帧配置寄存器（SAI_BxFCFG）（x = 0, 1） 1263
34.4.5. 子模块 x slot 配置寄存器（SAI_BxSCFG）（x = 0, 1） 1264
34.4.6. 子模块 x 中断使能寄存器（SAI_BxINTEN）（x = 0, 1） 1265
34.4.7. 子模块 x 状态寄存器（SAI_BxSTAT）（x = 0, 1） 1266
34.4.8. 子模块 x 中断标志清除寄存器（SAI_BxINTC）（x = 0, 1） 1268
34.4.9. 子模块 x 数据寄存器（SAI_BxDATA）（x = 0, 1） 1269
34.4.10. PDM 控制寄存器（SAI_PDM） 1270
34.4.11. PDM 配置寄存器（SAI_PDMCFG） 1270

35. 图像处理加速器（IPA）（适用于 GD32H737_757_759） 1273

35.1. 简介 1273

35.2. 主要特性 1273

35.3. 结构框图 1274

35.4. 信号描述 1274

35.5. 功能概述 1274

35.5.1. 转换操作 1275
35.5.2. 前景层和背景层 LUT 1276
35.5.3. 前景层和背景层像素通道扩展（PCE） 1276
35.5.4. 前景通道缩放 1279
35.5.5. 混合 1281
35.5.6. 目标像素通道压缩（PCC） 1282
35.5.7. 旋转 1283
35.5.8. 内部定时器 1283
35.5.9. 行标记 1284
35.5.10. 传输流 1284
35.5.11. 配置 1284

35.6. 中断 1288

35.7. IPA 寄存器 1291

35.7.1. 控制寄存器（IPA_CTL） 1291
35.7.2. 中断状态寄存器（IPA_INTF） 1292
35.7.3. 中断标志清除寄存器（IPA_INTC） 1293
35.7.4. 前景层存储区基地址寄存器（IPA_FMADDR） 1294
35.7.5. 前景层行偏移寄存器（IPA_FLOFF） 1295
35.7.6. 背景层存储区基地址寄存器（IPA_BMADDR） 1295
35.7.7. 背景层行偏移寄存器（IPA_BLOFF） 1296
35.7.8. 前景层像素控制寄存器（IPA_FPCTL） 1296
35.7.9. 前景层像素值寄存器（IPA_FPV） 1298
35.7.10. 背景层像素控制寄存器（IPA_BPCTL） 1299
35.7.11. 背景层像素值寄存器（IPA_BPV） 1300

35.7.12. 前景层 LUT 存储区基地址寄存器（IPA_FLMADDR） 1301
35.7.13. 背景层 LUT 存储区基地址寄存器（IPA_BLMADDR） 1301
35.7.14. 目标像素控制寄存器（IPA_DPCTL） 1302
35.7.15. 目标像素值寄存器（IPA_DPV） 1303
35.7.16. 目标存储区基地址寄存器（IPA_DMADDR） 1305
35.7.17. 目标行偏移寄存器（IPA_DLOFF） 1306
35.7.18. 图像大小寄存器（IPA_IMS） 1306
35.7.19. 行标记寄存器（IPA_LM） 1307
35.7.20. 内部定时器控制寄存器（IPA_ITCTL） 1307
35.7.21. 双线性缩放控制寄存器（IPA_BSCTL） 1308
35.7.22. 缩放目标图像大小寄存器（IPA_DIMS） 1309
35.7.23. 前景层偶数帧/UV 存储区基地址寄存器（IPA_EF_UV_MADDR） 1309
35.7.24. 色彩空间转换系数配置寄存器 0（IPA_CSCC_CFG0） 1310
35.7.25. 色彩空间转换系数配置寄存器 1（IPA_CSCC_CFG1） 1310
35.7.26. 色彩空间转换系数配置寄存器 2（IPA_CSCC_CFG2） 1311

36. SDIO 接口（SDIO）（适用于 GD32H737_757_759） 1312

36.1. 简介 1312

36.2. 主要特征 1312

36.3. SDIO 功能说明 1313

36.3.1. SDIO 总线拓扑 1313
36.3.2. SDIO 操作模式 1315
36.3.3. SDIO 框图 1315
36.3.4. SDIO 引脚和内部信号 1316
36.3.5. SDIO 介绍 1316
36.3.6. SDIO 适配器 1318
36.3.7. AHB 从接口 1322
36.3.8. AHB 主接口 1322
36.3.9. MDMA 请求 1323
36.3.10. AHB 总线与 SDIO_CLK 时钟的关系 1324
36.3.11. 硬件流控制 1325

36.4. 卡功能描述 1325

36.4.1. 卡寄存器 1325
36.4.2. 命令 1326
36.4.3. 响应 1335
36.4.4. 数据包格式 1339
36.4.5. 卡的两种状态 1341

36.5. 编程序列 1347

36.5.1. 卡识别 1347
36.5.2. 引导操作 1348
36.5.3. 无数据命令 1350
36.5.4. 单个数据块或多个数据块写 1350
36.5.5. 单个数据块或多个数据块读 1351

36.5.6. 数据流写和数据流读（仅适用于 e•MMC） 1352
36.5.7. 擦除 1353
36.5.8. 总线宽度选择 1354
36.5.9. 保护管理 1354
36.5.10. 卡上锁/解锁操作 1355
36.5.11. 睡眠 1356
36.5.12. CMD12 发送时序 1357
36.6. 特定操作 1359
36.6.1. UHS-I 电压切换 1359
36.7. SDIO 寄存器 1361
36.7.1. 电源控制寄存器（SDIO_PWRCTL） 1361
36.7.2. 时钟控制寄存器（SDIO_CLKCTL） 1362
36.7.3. 命令参数寄存器（SDIO_CMDAGMT） 1363
36.7.4. 命令控制寄存器（SDIO_CMDCTL） 1364
36.7.5. 命令索引响应寄存器（SDIO_RSPCMDIDX） 1366
36.7.6. 响应寄存器（SDIO_RESPx x = 0..3） 1366
36.7.7. 数据超时寄存器（SDIO_DATATO） 1367
36.7.8. 数据长度寄存器（SDIO_DATALEN） 1367
36.7.9. 数据控制寄存器（SDIO_DATACTL） 1368
36.7.10. 数据计数寄存器（SDIO_DATACNT） 1369
36.7.11. 状态寄存器（SDIO_STAT） 1370
36.7.12. 中断清除寄存器（SDIO_INTC） 1372
36.7.13. 中断使能寄存器（SDIO_INTEN） 1373
36.7.14. ACK 超时寄存器（SDIO_ACKTO） 1375
36.7.15. FIFO 数据寄存器（SDIO_FIFO） 1376
36.7.16. 内部 DMA（IDMA）控制寄存器（SDIO_IDMACTL） 1376
36.7.17. 内部 DMA（IDMA）缓冲大小寄存器（SDIO_IDMASIZE） 1377
36.7.18. IDMA 缓冲区 0 基地址寄存器（SDIO_IDMAADDR0） 1377
36.7.19. IDMA 缓冲区 1 基地址寄存器（SDIO_IDMAADDR1） 1378
37. 管理数据输入/输出接口（MDIO）（适用于 GD32H737_757_759） 1379
37.1. 简介 1379
37.2. 主要特性 1379
37.3. 引脚和内部信号 1379
37.4. 功能描述 1380
37.4.1. 帧结构 1380
37.4.2. 典型使用流程 1382
37.5. MDIO 寄存器 1384
37.5.1. 控制寄存器（MDIO_CTL） 1384
37.5.2. 接收帧信息寄存器（MDIO_RFRM） 1384
37.5.3. 数据接收寄存器（MDIO_RDATA） 1385
37.5.4. 地址接收寄存器（MDIO_RADDR） 1385

37.5.5. 数据发送寄存器（MDIO_TDATA） 1385
37.5.6. 配置寄存器（MDIO_CFG） 1386
37.5.7. 状态寄存器（MDIO_STAT） 1386
37.5.8. 中断使能寄存器 (MDIO_INTEN) 1388
37.5.9. 引脚数值寄存器（MDIO_PIN） 1389
37.5.10. 超时寄存器（MDIO_TO） 1389
38. 外部存储器控制器（EXMC） 1390
38.1. 简介 1390
38.2. 主要特性 1390
38.3. 功能描述 1390
38.3.1. 结构框图 1390
38.3.2. 总线接口 1391
38.3.3. AXI 错误 1391
38.3.4. EXMC 访问基本规范 1392
38.3.5. 外部设备地址映射 1392
38.3.6. NOR/PSRAM 控制器 1396
38.3.7. NAND flash 控制器 1414
38.3.8. SDRAM 控制器 1417
38.4. EXMC 寄存器 1428
38.4.1. NOR/PSRAM 控制器寄存器 1428
38.4.2. NAND Flash 控制器寄存器 1432
38.4.3. SDRAM 控制器寄存器 1437
39. VREF 1444
39.1. 简介 1444
39.2. 主要特性 1444
39.3. 功能描述 1444
39.4. VREF 寄存器 1446
39.4.1. 控制状态寄存器（VREF_CS） 1446
39.4.2. 校准寄存器（VREF_CALIB） 1446
40. 低功耗数字温度传感器（LPDTS） 1448
40.1. 简介 1448
40.2. 主要特征 1448
40.3. 模块框图 1448
40.4. 功能说明 1449
40.4.1. LPDTS 内部信号 1449
40.4.2. 操作模式 1449
40.4.3. 温度测量原理 1449
40.4.4. 采样时间 1450
40.4.5. 触发源 1451

40.4.6. 开关控制 1452
40.4.7. LPDTS 低功耗模式 1452
40.4.8. LPDTS 中断 1452
40.5. LPDTS 寄存器 1453
40.5.1. LPDTS 配置寄存器（LPDTS_CFG） 1453
40.5.2. T0 传感器数据寄存器（LPDTS_SDATA） 1453
40.5.3. 斜率数据寄存器（LPDTS_RDATA） 1454
40.5.4. 中断阈值寄存器（LPDTS_IT） 1455
40.5.5. 温度值寄存器（LPDTS_DATA） 1455
40.5.6. 温度传感器状态寄存器（LPDTS_STAT） 1455
40.5.7. 中断使能寄存器（LPDTS_INTEN） 1457
40.5.8. 中断标志清除寄存器（LPDTS_INTC） 1458
40.5.9. 选择寄存器（LPDTS_OP） 1458
41. 编码器分频输出控制器（EDOUT） 1460
41.1. 简介 1460
41.2. 主要特征 1460
41.3. 功能说明 1460
41.4. Z 相输出模式 1461
41.5. 操作指导 1461
41.5.1. EDOUT 初始化 1462
41.5.2. EDOUT 更新处理 1462
41.5.3. EDOUT 工作案例 1463
41.6. EDOUT 寄存器 1464
41.6.1. 控制寄存器（EDOUT_CTL） 1464
41.6.2. 使能寄存器（EDOUT_ENABLE） 1464
41.6.3. 位置寄存器（EDOUT_LOC） 1465
41.6.4. 输出计数器寄存器（EDOUT_OCNT） 1465
41.6.5. 位置计数寄存器（EDOUT_LCNT） 1466
41.6.6. Z 相配置寄存器（EDOUT_ZCR） 1466
42. 控制器局域网络（CAN） 1468
42.1. 简介 1468
42.2. 主要特征 1468
42.3. 功能说明 1469
42.3.1. 邮箱描述符 1470
42.3.2. 接收 FIFO 描述符 1474
42.3.3. 通信模式 1479
42.3.4. 省电模式 1480
42.3.5. 数据发送 1481
42.3.6. 数据接收 1485
42.3.7. 在虚拟联网模式下的数据接收 1491

42.3.8. CAN FD 操作 1493
42.3.9. 错误和状态 1496
42.3.10. 通信参数 1498
42.3.11. 中断 1500
42.4. 典型的 CAN 配置流程示例 1501
42.5. CAN 寄存器 1503
42.5.1. 控制寄存器 0（CAN_CTL0） 1503
42.5.2. 控制寄存器 1（CAN_CTL1） 1505
42.5.3. 计数器寄存器（CAN_TIMER） 1507
42.5.4. 接收邮箱公有过滤寄存器（CAN_RMPUBF） 1507
42.5.5. 错误寄存器 0（CAN_ERR0） 1508
42.5.6. 错误寄存器 1（CAN_ERR1） 1508
42.5.7. 中断使能寄存器（CAN_INTEN） 1511
42.5.8. 状态寄存器（CAN_STAT） 1512
42.5.9. 控制寄存器 2（CAN_CTL2） 1513
42.5.10. 常规帧 CRC 寄存器（CAN_CRCC） 1515
42.5.11. 接收 FIFO 共有过滤寄存器（CAN_RFIFOPUBF） 1515
42.5.12. 接收 FIFO 标识符过滤元素匹配序号寄存器（CAN_RFIFOIFMN） 1516
42.5.13. 位时间寄存器（CAN_BT） 1516
42.5.14. 接收 FIFO/邮箱私有过滤 x 寄存器（CAN_RFIFOMPFx）（x=0..31） 1517
42.5.15. 虚拟联网模式控制寄存器 0（CAN_PN_CTL0） 1518
42.5.16. 虚拟联网模式超时寄存器（CAN_PN_TO） 1519
42.5.17. 虚拟联网模式状态寄存器（CAN_PN_STAT） 1519
42.5.18. 虚拟联网模式期望标识符 0 寄存器（CAN_PN_EID0） 1520
42.5.19. 虚拟联网模式期望 DLC 寄存器（CAN_PN_EDLC） 1521
42.5.20. 虚拟联网模式期望数据低字 0 寄存器（CAN_PN_EDL0） 1521
42.5.21. 虚拟联网模式期望数据低字 1 寄存器（CAN_PN_EDL1） 1522
42.5.22. 虚拟联网模式标识符过滤器 / 期望标识符 1 寄存器（CAN_PN_IFEID1） 1522
42.5.23. 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） 1523
42.5.24. 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） 1524
42.5.25. 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） 1524
42.5.26. 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） 1525
42.5.27. 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） 1526
42.5.28. 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） 1526
42.5.29. FD 控制寄存器（CAN_FDCTL） 1526
42.5.30. FD 位时间寄存器（CAN_FDBT） 1528
42.5.31. 常规帧和 FD 帧 CRC 寄存器（CAN_CRCCFD） 1528
43. 以太网（ENET）（适用于 GD32H737_757_759） 1530
43.1. 简介 1530
43.2. 主要特性 1530
43.2.1. 模块框图 1531
43.2.2. MAC 802.3 以太网数据包描述 1532

43.2.3. 以太网信号描述 1532
43.3. 功能描述 1534
43.3.1. 接口配置 1534
43.3.2. MAC 功能简介 1538
43.3.3. DMA 控制器描述 1546
43.3.4. MAC 统计计数器：MSC 1567
43.3.5. 唤醒管理：WUM 1568
43.3.6. 精确时间协议：PTP 1570
43.3.7. 典型的以太网配置流程示例 1574
43.3.8. 以太网中断 1575
43.4. ENET 寄存器 1576
43.4.1. MAC 配置寄存器（ENET_MAC_CFG） 1577
43.4.2. MAC 帧过滤器寄存器（ENET_MAC_FRMF） 1579
43.4.3. MAC hash 列表高寄存器（ENET_MAC_HLH） 1580
43.4.4. MAC hash 列表低寄存器（ENET_MAC_HLL） 1581
43.4.5. MAC PHY 控制寄存器（ENET_MAC_PHY_CTL） 1581
43.4.6. MAC PHY 数据寄存器（ENET_MAC_PHY_DATA） 1582
43.4.7. MAC 流控寄存器（ENET_MAC_FCTL） 1583
43.4.8. MAC VLAN 标签寄存器（ENET_MAC_VLT） 1584
43.4.9. MAC 远程唤醒帧过滤器寄存器（ENET_MAC_RWFF） 1585
43.4.10. MAC 唤醒管理寄存器（ENET_MAC_WUM） 1585
43.4.11. MAC 调试寄存器（ENET_MAC_DBG） 1586
43.4.12. MAC 中断状态寄存器（ENET_MAC_INTF） 1588
43.4.13. MAC 中断屏蔽寄存器（ENET_MAC_INTMSK） 1589
43.4.14. MAC 地址 0 高寄存器（ENET_MAC_ADDR0H） 1589
43.4.15. MAC 地址 0 低寄存器（ENET_MAC_ADDR0L） 1590
43.4.16. MAC 地址 1 高寄存器（ENET_MAC_ADDR1H） 1590
43.4.17. MAC 地址 1 低寄存器（ENET_MAC_ADDR1L） 1591
43.4.18. MAC 地址 2 高寄存器（ENET_ MAC_ADDR2H） 1591
43.4.19. MAC 地址 2 低寄存器（ENET_MAC_ADDR2L） 1592
43.4.20. MAC 地址 3 高寄存器（ENET_MAC_ADDR3H） 1593
43.4.21. MAC 地址 3 低寄存器（ENET_MAC_ADDR3L） 1593
43.4.22. MAC 流控阈值寄存器（ENET_MAC_FCTH） 1594
43.4.23. MSC 控制寄存（ENET_MSC_CTL） 1595
43.4.24. MSC 接收中断状态寄存器（ENET_MSC_RINTF） 1596
43.4.25. MSC 发送中断状态寄存器（ENET_MSC_TINTF） 1596
43.4.26. MSC 接收中断屏蔽寄存器（ENET_MSC_RINTMSK） 1597
43.4.27. MSC 发送中断屏蔽寄存器（ENET_MSC_TINTMSK） 1598
43.4.28. MSC 1 次冲突后发送”好”帧的计数器寄存器（ENET_MSC_SCCNT） 1598
43.4.29. MSC 1 次以上冲突后发送”好”帧的计数器寄存器（ENET_MSC_MSCCNT） 1599
43.4.30. MSC 发送”好”帧计数器寄存器（ENET_MSC_TGFCNT） 1599
43.4.31. MSC CRC 错误接收帧计数器寄存器（ENET_MSC_RFCECNT） 1600
43.4.32. MSC 对齐错误接收帧计数器寄存器（ENET_MSC_RFAECNT） 1600
43.4.33. MSC“好”单播帧接收帧计数器寄存器（ENET_MSC_RGUFCNT） 1600

43.4.34. PTP 时间戳控制寄存器（ENET_PTP_TSCTL） 1601
43.4.35. PTP 亚秒递增寄存器（ENET_PTP_SSINC） 1603
43.4.36. PTP 时间戳高寄存器（ENET_PTP_TSH） 1604
43.4.37. PTP 时间戳低寄存器（ENET_PTP_TSL） 1604
43.4.38. PTP 时间戳高更新寄存器（ENET_PTP_TSUH） 1605
43.4.39. PTP 时间戳低更新寄存器（ENET_PTP_TSUL） 1605
43.4.40. PTP 时间戳加数寄存器（ENET_PTP_TSADDEND） 1606
43.4.41. PTP 期望时间高寄存器（ENET_PTP_ETH） 1606
43.4.42. PTP 期望时间低寄存器（ENET_PTP_ETL） 1606
43.4.43. PTP 时间戳标志寄存器（ENET_PTP_TSF） 1607
43.4.44. PTP PPS 控制寄存器（ENET_PTP_PPSCTL） 1607
43.4.45. DMA 总线控制寄存器（ENET_DMA_BCTL） 1608
43.4.46. DMA 发送查询使能寄存器（ENET_DMA_TPEN） 1610
43.4.47. DMA 接收查询使能寄存器（ENET_DMA_RPEN） 1610
43.4.48. DMA 接收描述符列表地址寄存器（ENET_DMA_RDTADDR） 1611
43.4.49. DMA 发送描述符列表地址寄存器（ENET_DMA_TDTADDR） 1611
43.4.50. DMA 状态寄存器（ENET_DMA_STAT） 1612
43.4.51. DMA 控制寄存器（ENET_DMA_CTL） 1615
43.4.52. DMA 中断使能寄存器（ENET_DMA_INTEN） 1617
43.4.53. DMA 丢失帧和缓存溢出计数器寄存器（ENET_DMA_MFBOCNT） 1619
43.4.54. DMA 接收状态看门狗计数器寄存器（ENET_DMA_RSWDC） 1620
43.4.55. DMA 当前发送描述符地址寄存器（ENET_DMA_CTDADDR） 1620
43.4.56. DMA 当前接收描述符地址寄存器（ENET_DMA_CRDADDR） 1621
43.4.57. DMA 当前发送缓存地址寄存器（ENET_DMA_CTBADDR） 1621
43.4.58. DMA 当前接收缓存地址寄存器（ENET_DMA_CRBADDR） 1622

44. 比较器（CMP） 1623

44.1. 简介 1623

44.2. 主要特征 1623

44.3. 功能描述 1623

44.3.1. 比较器时钟 1624
44.3.2. 比较器的 I/O 配置 1624
44.3.3. 比较器供电模式 1626
44.3.4. 比较器窗口模式 1626
44.3.5. 比较器迟滞 1626
44.3.6. 比较器寄存器写保护 1626
44.3.7. 比较器输出消隐 1626
44.3.8. 电压定标器功能 1627
44.3.9. 比较器中断 1627

44.4. 比较器寄存器 1628

44.4.1. 状态寄存器（CMP_STAT） 1628
44.4.2. 中断标志位清除寄存器（CMP_IFC） 1629
44.4.3. 备用选择寄存器（CMP_SR） 1629

44.4.4. CMP0 控制状态寄存器（CMP0_CS） 1630
44.4.5. CMP1 控制状态寄存器（CMP1_CS） 1632
45. 高性能数字滤波器（HPDF） 1635
45.1. 简介 1635
45.2. 主要特性 1635
45.3. 功能描述 1635
45.3.1. HPDF 结构框图 1635
45.3.2. HPDF 开关控制 1637
45.3.3. HPDF 时钟 1637
45.3.4. 复用串行数据通道 1638
45.3.5. 并行数据输入 1644
45.3.6. 规则组转换 1646
45.3.7. 注入组转换 1647
45.3.8. 数字滤波器 1648
45.3.9. 积分器 1648
45.3.10. 阈值监视器 1649
45.3.11. 故障监视器 1651
45.3.12. 极值监视器 1651
45.3.13. 数据单元 1651
45.3.14. HPDF 中断 1652
45.4. HPDF 寄存器 1654
45.4.1. HPDF 通道 x 寄存器（x = 0~7） 1654
45.4.2. HPDF 滤波器 y 寄存器（y = 0~3） 1659
46. 实时解密（RTDEC）（适用于 GD32H737_757_759） 1672
46.1. 简介 1672
46.2. 主要特性 1672
46.3. 功能说明 1672
46.3.1. 结构框图 1673
46.3.2. RTDEC 实时解密介绍 1673
46.3.3. 计数器模式下使用 AES 解密 1674
46.3.4. RTDEC 配置 1675
46.3.5. RTDEC 错误管理 1677
46.4. RTDEC 中断 1677
46.5. RTDEC 寄存器 1679
46.5.1. 区域 x 配置寄存器（RTDEC_AREx_CFG） 1679
46.5.2. 区域 x 起始地址寄存器（RTDEC_AREx_SADDR） 1680
46.5.3. 区域 x 结束地址寄存器（RTDEC_AREx_EADDR） 1680
46.5.4. 区域 x 随机数寄存器 0（RTDEC_AREx_NONCE0） 1681
46.5.5. 区域 x 随机数寄存器 1（RTDEC_AREx_NONCE1） 1681
46.5.6. 区域 x 秘钥寄存器 0（RTDEC_AREx_KEY0） 1682

46.5.7. 区域 x 秘钥寄存器 1（RTDEC_AREx_KEY1） 1682
46.5.8. 区域 x 秘钥寄存器 2（RTDEC_AREx_KEY2） 1683
46.5.9. 区域 x 秘钥寄存器 3（RTDEC_AREx_KEY3） 1683
46.5.10. 中断标志寄存器（RTDEC_INTF） 1683
46.5.11. 中断标志清除寄存器（RTDEC_INTC） 1684
46.5.12. 中断使能寄存器（RTDEC_INTEN） 1685
47. 滤波算法加速器（FAC） 1687
47.1. 简介 1687
47.2. 主要特征 1687
47.3. 功能描述 1687
47.3.1. 基本描述 1687
47.3.2. 本地存储和缓冲区 1688
47.3.3. 输入缓冲区 1688
47.3.4. 输出缓冲区 1690
47.3.5. 初始化函数 1692
47.3.6. 滤波器函数 1692
47.3.7. 定点数据格式 1694
47.3.8. 浮点数据格式 1694
47.3.9. FIR 滤波器 1695
47.3.10. IIR 滤波器 1696
47.4. FAC 寄存器 1697
47.4.1. FAC X0 缓冲区配置寄存器（FAC_X0BCFG） 1697
47.4.2. FAC X1 缓冲区配置寄存器（FAC_X1BCFG） 1697
47.4.3. FAC Y 缓冲区配置寄存器（FAC_YBCFG） 1698
47.4.4. FAC 参数配置寄存器（FAC_PARACFG） 1698
47.4.5. FAC 控制寄存器（FAC_CTL） 1699
47.4.6. FAC 状态寄存器（FAC_STAT） 1701
47.4.7. FAC 写数据寄存器（FAC_WDATA） 1702
47.4.8. FAC 读数据寄存器（FAC_RDATA） 1702
48. 硬件信号量（HWSEM）（适用于 GD32H737_757_759） 1704
48.1. 简介 1704
48.2. 主要特征 1704
48.3. 功能说明 1704
48.3.1. 模块框图 1704
48.3.2. 信号量 x 1704
48.3.3. 锁定信号量 1705
48.3.4. 解锁信号量 1706
48.3.5. 解锁所有信号量 1706
48.3.6. 中断 1706
48.4. HWSEM 寄存器 1709

48.4.1. 控制寄存器（HWSEM_CTLx）（x=0…31） 1709
48.4.2. 读锁定寄存器（HWSEM_RLKx）（x=0...31） 1709
48.4.3. 中断使能寄存器（HWSEM_INTEN） 1710
48.4.4. 中断状态清除寄存器（HWSEM_INTC） 1710
48.4.5. 状态寄存器（HWSEM_STAT） 1711
48.4.6. 中断状态寄存器（HWSEM_INTF） 1711
48.4.7. 解锁寄存器（HWSEM_UNLK） 1712
48.4.8. 键值寄存器（HWSEM_KEY） 1712
49. 通用串行总线高速接口（USBHS） 1714
49.1. 概述 1714
49.2. 主要特性 1714
49.3. 结构框图 1715
49.4. 信号线描述 1715
49.5. 功能描述 1715
49.5.1. USBHS PHY 选择、时钟及工作模式 1715
49.5.2. USB 主机功能 1719
49.5.3. USB 设备功能 1721
49.5.4. OTG 功能概述 1722
49.5.5. 数据 FIFO 1723
49.5.6. DMA 功能 1725
49.5.7. 操作手册 1726
49.6. 中断 1731
49.7. USBHS 寄存器 1733
49.7.1. 全局控制与状态寄存器组 1733
49.7.2. 主机控制和状态寄存器 1761
49.7.3. 设备控制和状态寄存器 1774
49.7.4. 电源和时钟控制寄存器（USBHS_PWRCLKCTL） 1801
50. EtherCAT®从站控制器（ESC）（适用于 GD32H75E） 1803
50.1. 系统和总线架构 1803
50.1.1. 总线架构 1803
50.1.2. 存储器映射 1803
50.1.3. AHB 直接/间接访问 1806
50.1.4. 忙碌状态下的寄存器保护 1807
50.1.5. OPB 传输超时功能 1808
50.1.6. ESC 内核控制器（ESC_CCTL） 1808
50.1.7. ESC 内核控制器寄存器定义 1808
50.1.8. 系统配置控制器（SYSCFG） 1814
50.1.9. 系统配置寄存器定义 1814
50.2. 电源管理单元（PMU） 1817
50.2.1. 简介 1817

50.2.2. 主要特征 1817
50.2.3. 功能说明 1817
50.2.4. 寄存器定义 1821
50.3. 复位和时钟单元（RCU） 1824
50.3.1. 复位控制单元 1824
50.3.2. 时钟控制单元（CCTL） 1825
50.3.3. RCU 寄存器 1826
50.4. 中断控制器（INTC） 1832
50.4.1. 简介 1832
50.4.2. 主要特征 1832
50.4.3. 功能说明 1832
50.4.4. INTC 寄存器 1835
50.5. 通用输入/输出（GPIO） 1839
50.5.1. 概述 1839
50.5.2. 特征 1839
50.5.3. 功能概述 1839
50.5.4. 寄存器定义 1843
50.6. 定时器（TIMER） 1852
50.6.1. 基本定时器 1852
50.6.2. 自由运行计数器 1854
50.7. 总线接口（PDI Wrapper） 1856
50.7.1. SPI/QSPI/OSPI 从机 1856
50.8. 以太网 PHYS 1876
50.8.1. 简介 1876
50.8.2. 主要特征 1876
50.8.3. 功能概述 1877
50.8.4. PHY 寄存器 1879
50.9. EtherCAT 1922
50.9.1. 简介 1922
50.9.2. 主要特征 1922
50.9.3. 功能描述 1924
50.9.4. ESC 寄存器定义 1931
51. 附录 1993
52. 版本历史 1994

图索引

图 1-1. Cortex®-M7 处理器结构框图 60
图 1-2. GD32H73x_75x 器件的系统架构示意图（适用于 GD32H737_757_759） 62
图 1-3. GD32H73x_75x 器件的系统架构示意图（适用于 GD32H75E） 63
图 1-4. 总线矩阵区域 0 64
图 1-5. 总线矩阵区域 1 65
图 1-6. 总线矩阵区域 2 65
图 1-7. AXI SRAM 框图 82
图 1-8. ITCM/DTCM/AXI SRAM 共享的 RAM 框图 82
图 1-9. AXI 互连矩阵的框图 86
图 2-1. RAMECCMU 架构图 136
图 3-1. FMC 框图 142
图 3-2. 扇区擦除操作流程 145
图 3-3. 标准整片擦除操作流程 146
图 3-4. 带清除保护的整片擦除 147
图 3-5. 编程操作流程 149
图 3-6. 标准模式和安全模式内部存储架构 158
图 4-1 熔丝控制器结构框图 181
图 4-2. EFUSE 中断映射图 186
图 5-1. 电源域概览 198
图 5-2. 备用域电压阈值的波形 200
图 5-3. 上电 / 掉电复位波形图 201
图 5-4. BOR 波形图 201
图 5-5. LVD 阈值波形图 202
图 5-6. VAVD 阈值监测波形图 203
图 5-7. 温度阈值监测波形图 203
图 5-8. LDO 供电 V_{CORE} 电源域 204
图 5-9. SMPS 供电 V_{CORE} 电源域 205
图 5-10.旁路 205
图 5-11. VOVD 波形 206
图 5-12. USB 稳压器供电时连接示意图 206
图 5-13. USB 稳压器旁路时连接示意图 206
图 6-1. 系统复位电路 218
图 6-2. 时钟树 219
图 6-3. 时钟树 222
图 6-4. HXTAL 时钟源 225
图 6-5. 旁路模式下 HXTAL 时钟源 225
图 7-1. CTC 简介 390
图 7-2. CTC 校准计数器 391
图 8-1. EXTI 框图 409
图 9-1. TRIGSEL 主要组成示例 420
图 10-1. 标准 I/O 端口位的基本结构 461

图 10-2. 浮空/上拉/下拉输入配置 463
图 10-3. 输出配置 463
图 10-4. 模拟高阻配置 464
图 10-5. 备用功能配置 464
图 10-6. GD32H73x_75x 系列 ADC 的模拟配置 465
图 10-7. 使用采样窗口进行过滤 467
图 10-8. 输入滤波时钟周期 468
图 11-1. CRC 计算单元框图 484
图 12-1. TRNG 模块框图 489
图 13-1. DATAM 不交换/半字交换 499
图 13-2. DATATM 字节交换/位交换 500
图 13-3. CAU 框图 501
图 13-4. DES / TDES ECB 加密 502
图 13-5. DES / TDES ECB 解密 503
图 13-6. DES / TDES CBC 加密 504
图 13-7. DES / TDES CBC 解密 505
图 13-8. AES ECB 加密 506
图 13-9. AES ECB 解密 506
图 13-10. AES CBC 加密 507
图 13-11. AES CBC 解密 508
图 13-12. 计数器块结构 508
图 13-13. AES CTR 加密/解密 509
图 14-1. DATAM 不交换/半字交换 528
图 14-2. DATAM 字节交换/位交换 528
图 14-3. HAU 结构框图 529
图 15-1. TMU 模块结构框图 540
图 16-1. 系统架构 556
图 16-2. 三种传输模式的数据流 557
图 16-3. 握手机制 558
图 16-4. PWIDTH 为‘00’时，数据的打包/解包 561
图 16-5. PWIDTH 为‘01’时，数据的打包/解包 562
图 16-6. PWIDTH 为‘10’时，数据的打包/解包 562
图 16-7. 存储切换模式 563
图 16-8. DMA0 与 DMA1 的系统连接 569
图 17-1. MDMA 系统框图 581
图 17-2. 传输模式之间的联系 582
图 17-3. 字、半字、字节顺序交换 583
图 17-4. 数据填充和对齐（源大于目的） 584
图 17-5. 数据填充和对齐（源小于目的） 584
图 17-6. 数据的打包/解包 585
图 17-7. MDMA 中断逻辑图 590
图 18-1. DMAMUX 结构框图 605
图 18-2. 同步模式 607
图 18-3. 通道事件输出 608

图 19-1. JTAG 模块框图 623
图 20-1. ADC 模块框图 635
图 20-2. 单次运行模式 637
图 20-3. 连续运行模式 638
图 20-4. 扫描运行模式，且连续运行模式失能 639
图 20-5. 扫描运行模式，连续运行模式使能 639
图 20-6. 间断运行模式 639
图 20-7. 14 位数据存储模式 640
图 20-8. 12 位数据存储模式 641
图 20-9. 6 位数据存储模式 641
图 20-10. HFDF 与 ADC 模块握手信号示意图 643
图 20-11. 12 位 ADC 20 位到 16 位的结果截断 645
图 20-12. 12 位 ADC 右移 5 位和取整的数例 645
图 20-13. 14 位 ADC 过采样右移 10 位 645
图 20-14. 数值例子 14 位 ADC 过采样右移 10 位 646
图 20-15. ADC 同步框图 647
图 20-16. 基于 16 个通道的常规并行模式 648
图 20-17. 一个采用连续运行模式通道上的常规跟随模式 648
图 21-1. DAC 结构框图 675
图 21-2. DAC LFSR 算法 677
图 21-3. DAC 三角噪声模式生成的波形 677
图 21-4. 采样时间和刷新时间计算公式 680
图 21-5. DAC 采样保持模式 680
图 22-1. 独立看门狗定时器框图 696
图 22-2. 窗口看门狗定时器框图 701
图 22-3. 窗口看门狗定时器时序图 702
图 23-1. RTC 结构框图 706
图 24-1. 高级定时器结构框图 738
图 24-2. 内部时钟分频为 1 时正常模式下的控制电路 739
图 24-3. 当预分频器的参数从 1 变到 2 时，计数器的时序图 740
图 24-4. 向上计数时序图，PSC=0/2 741
图 24-5. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 741
图 24-6. 向下计数时序图，PSC=0/2 742
图 24-7. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值 743
图 24-8. 中央计数模式计数器时序图 744
图 24-9. 中央计数模式下计数器重复时序图 745
图 24-10. 在向上计数模式下计数器重复时序图 745
图 24-11. 在向下计数模式下计数器重复时序图 746
图 24-12. 通道 0 输入捕获逻辑 747
图 24-13. 多模式通道 0 输入捕获逻辑 747
图 24-14. 输出比较逻辑（当 MCHxMSEL = 2’00 时，x=0,1,2,3） 748
图 24-15. 输出比较逻辑（当 MCHxMSEL = 2’11 时，x=0,1,2,3） 748
图 24-16. 三种输出比较模式 750
图 24-17. EAPWM 时序图 751

图 24-18. CAPWM 时序图 751
图 24-19. 通道 x 输出 PWM（CHxVAL < CHxCOMVAL_ADD） 753
图 24-20. 通道 x 输出 PWM（CHxVAL = CHxCOMVAL_ADD） 753
图 24-21. 通道 x 输出 PWM（CHxVAL > CHxCOMVAL_ADD） 754
图 24-22. 通道 x 输出 PWM（CHxVAL 或 CHxCOMVAL_ADD > CARL） 754
图 24-23. 通道 x 输出 PWM 占空比随着 CHxCOMVAL_ADD 值而改变 755
图 24-24. 复合 PWM 模式下四通道输出 755
图 24-25. 边沿对齐模式下 CHx_O 输出脉冲（CHxOMPSEL≠2'b00） 756
图 24-26. 中央对齐模式下 CHx_O 输出脉冲（CHxOMPSEL≠2'b00） 756
图 24-27. 带死区时间的互补输出 759
图 24-28. BREAK0 中止功能逻辑图 760
图 24-29. BREAK1 中止功能逻辑图 760
图 24-30. 通道响应 BREAK0 中止输入（高电平有效）时，输出信号的行为（IOS=1） 761
图 24-31. BREAK0 和 BREAK1 中止输入有效时通道输出信号的行为 762
图 24-32. BREAK0 的 BRKINx（x=0..2）引脚锁存功能逻辑图 763
图 24-33. 译码器接口模式下计数器运行例子 764
图 24-34. CI0FE0 极性反相的译码器接口模式下的例子 765
图 24-35. 正交译码器信号断线检测框图 765
图 24-36. 非正交译码器模式 0 计数器运行实例（CH1P=0） 766
图 24-37. 非正交译码器模式 1 计数器运行实例（CH0P=0） 766
图 24-38. 霍尔传感器用在 BLDC 电机控制中 767
图 24-39. 两个定时器之间的霍尔传感器时序图 768
图 24-40. 复位模式 769
图 24-41. 暂停模式下的控制电路 769
图 24-42. 事件模式 770
图 24-43. 单脉冲模式，TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60 770
图 24-44. 可延时单脉冲模式（TIMERx_CHxCV=0x00, TIMERx_CAR=0x60） 771
图 24-45. 用定时器 2 的使能信号触发定时器 0 772
图 24-46. 用定时器 2 的更新事件来触发定时器 0 773
图 24-47. 用定时器 2 的使能来选通定时器 0 773
图 24-48. 用定时器 2 的 O0CPRE 信号选通定时器 1 774
图 24-49. 用定时器 2 的 CI0 输入来触发定时器 0 和定时器 2 775
图 24-50. 通用定时器 L0 结构框图 840
图 24-51. 内部时钟分频为 1 时正常模式下的控制电路 841
图 24-52. 当预分频器的参数从 1 变到 2 时，计数器的时序图 842
图 24-53. 向上计数时序图，PSC=0/2 843
图 24-54. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 843
图 24-55. 向下计数时序图，PSC=0/2 844
图 24-56. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值 845
图 24-57. 中央计数模式计数器时序图 846
图 24-58. 输入捕获逻辑 847
图 24-59. 输出比较逻辑（x=0,1,2,3） 848
图 24-60. 三种输出比较模式 849
图 24-61. EAPWM 时序图 850

图 24-62. CAPWM 时序图 850
图 24-63. 通道 x 输出 PWM（CHxVAL < CHxCOMVAL_ADD） 852
图 24-64. 通道 x 输出 PWM（CHxVAL = CHxCOMVAL_ADD） 852
图 24-65. 通道 x 输出 PWM（CHxVAL > CHxCOMVAL_ADD） 853
图 24-66. 通道 x 输出 PWM（CHxVAL 或 CHxCOMVAL_ADD > CARL） 853
图 24-67. 通道 x 输出 PWM 占空比随着 CHxCOMVAL_ADD 值而改变 854
图 24-68. 复合 PWM 模式下四通道输出 854
图 24-69. 边沿对齐模式下 CHx_O 输出脉冲（CHxOMPSEL≠2'b00） 855
图 24-70. 中央对齐模式下 CHx_O 输出脉冲（CHxOMPSEL≠2'b00） 855
图 24-71. 译码器接口模式下计数器运行例子 857
图 24-72. CI0FE0 极性反相的译码器接口模式下的例子 857
图 24-73. 正交译码器信号断线检测框图 858
图 24-74. 非正交译码器模式 0 计数器运行实例（CH1P=0） 858
图 24-75. 非正交译码器模式 1 计数器运行实例（CH0P=0） 859
图 24-76. 复位模式 860
图 24-77. 暂停模式 860
图 24-78. 事件模式 861
图 24-79. 单脉冲模式，TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60 861
图 24-80. 可延时单脉冲模式（TIMERx_CHxCV=0x00, TIMERx_CAR=0x60） 862
图 24-81. 通用定时器 L3 结构框图 894
图 24-82. 内部时钟分频为 1 时正常模式下的控制电路 895
图 24-83. 当预分频器的参数从 1 变到 2 时，计数器的时序图 896
图 24-84. 向上计数时序图，PSC=0/2 897
图 24-85. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 897
图 24-86. 在向上计数模式下计数器重复时序图 898
图 24-87. 通道 0 输入捕获逻辑 899
图 24-88. 多模式通道 0 输入捕获逻辑 899
图 24-89. 输出比较逻辑（当 MCHxMSEL = 2'00 时，x=0） 900
图 24-90. 输出比较逻辑（当 MCHxMSEL = 2'11 时，x=0） 900
图 24-91. 输出比较逻辑（x=1） 901
图 24-92. 三种输出比较模式 902
图 24-93. PWM 时序图 903
图 24-94. 通道 x 输出 PWM（CHxVAL < CHxCOMVAL_ADD） 904
图 24-95. 通道 x 输出 PWM（CHxVAL = CHxCOMVAL_ADD） 905
图 24-96. 通道 x 输出 PWM（CHxVAL > CHxCOMVAL_ADD） 905
图 24-97. 通道 x 输出 PWM（CHxVAL 或 CHxCOMVAL_ADD > CARL） 906
图 24-98. 通道 x 输出 PWM 占空比随着 CHxCOMVAL_ADD 值而改变 906
图 24-99. 边沿对齐模式下 CHx_O 输出脉冲（CHxOMPSEL=2'b01） 907
图 24-100. 带死区时间的互补输出 909
图 24-101. BREAK0 中止功能逻辑图 910
图 24-102. 通道响应 BREAK0 中止输入（高电平有效）时，输出信号的行为（IOS=1） 911
图 24-103. BREAK0 的 BRKIN0 引脚锁存功能逻辑图 912
图 24-104. 复位模式下的控制电路 913
图 24-105. 暂停模式下的控制电路 913

图 24-106. 事件模式下的控制电路 914
图 24-107. 单脉冲模式，TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60 914
图 24-108. 可延时单脉冲模式（TIMERx_CHxCV=0x00, TIMERx_CAR=0x60） 915
图 24-109. 通用定时器 L4 结构框图 948
图 24-110. 内部时钟分频为 1 时正常模式下的控制电路 949
图 24-111. 当预分频器的参数从 1 变到 2 时，计数器的时序图 949
图 24-112. 向上计数时序图，PSC=0/2 950
图 24-113. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 951
图 24-114. 在向上计数模式下计数器重复时序图 952
图 24-115. 通道 0 输入捕获逻辑 952
图 24-116. 多模式通道 0 输入捕获逻辑 953
图 24-117. 输出比较逻辑（当 MCHxMSEL = 2’00 时，x=0） 954
图 24-118. 输出比较逻辑（当 MCHxMSEL = 2’11 时，x=0） 954
图 24-119. 三种输出比较模式 955
图 24-120. PWM 时序图 956
图 24-121. 带死区时间的互补输出 959
图 24-122. BREAK0 中止功能逻辑图 960
图 24-123. 通道响应 BREAK0 中止输入（高电平有效）时，输出信号的行为（IOS=1） 960
图 24-124. BREAK0 的 BRKIN0 引脚锁存功能逻辑图 961
图 24-125. 单脉冲模式，TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60 962
图 24-126. 基本定时器结构框图 988
图 24-127. 内部时钟分频为 1 时正常模式下的控制电路 989
图 24-128. 当预分频器的参数从 1 变到 2 时，计数器的时序图 989
图 24-129. 向上计数时序图，PSC=0/2（TIMERx, x=5,6） 990
图 24-130. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值（TIMERx, x=5,6） 991
图 25-1. USART 模块内部框图 1001
图 25-2. USART 字符帧（8 数据位和 1 停止位） 1001
图 25-3. USART 发送步骤 1003
图 25-4. 过采样方式接收一个数据位（OSB = 0） 1004
图 25-5. 采用 DMA 方式实现 USART 数据发送配置步骤 1005
图 25-6. 采用 DMA 方式实现 USART 数据接收配置步骤 1006
图 25-7. 两个 USART 之间的硬件流控制 1006
图 25-8. 硬件流控制 1007
图 25-9. 空闲状态下检测断开帧 1008
图 25-10. 数据传输过程中检测断开帧 1009
图 25-11. 同步模式下的 USART 示例 1009
图 25-12. 8-bit 格式的 USART 同步通信波形（CLEN = 1） 1009
图 25-13. IrDA SIR ENDEC 模块 1010
图 25-14. IrDA 数据调制 1010
图 25-15. ISO7816-3 数据帧格式 1011
图 25-16. USART 接收 FIFO 结构 1013
图 25-17. USART 发送 FIFO 结构 1014
图 25-18. USART 中断映射框图 1015
图 26-1. I2C 模块框图 1039

图 26-2. 数据有效性 1040
图 26-3. 开始和停止信号 1041
图 26-4. 10 位地址的 I2C 通讯流程（主机发送） 1041
图 26-5. 7 位地址的 I2C 通讯流程（主机发送） 1041
图 26-6. 7 位地址的 I2C 通讯流程（主机接收） 1042
图 26-7. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R = 0） 1042
图 26-8. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R = 1） 1042
图 26-9. 数据保持时间 1043
图 26-10. 数据建立时间 1043
图 26-11. 数据发送 1045
图 26-12. 数据接收 1046
图 26-13. I2C 从机初始化 1048
图 26-14. I2C 从机发送编程模型（SS=0） 1049
图 26-15. I2C 从机发送编程模型（SS=1） 1050
图 26-16. I2C 从机接收编程模型 1051
图 26-17. I2C 主机初始化 1052
图 26-18. I2C 主机发送编程模型（N<=255） 1053
图 26-19. I2C 主机发送编程模型（N>255） 1054
图 26-20. I2C 主机接收编程模型（N<=255） 1055
图 26-21. I2C 主机接收编程模型（N>255） 1056
图 26-22. SMBus 主机发送器和从机接收器通信流程 1059
图 26-23. SMBus 主机接收器和从机发送器通信流程 1060
图 27-1. SPI 结构框图 1076
图 27-2. SPI 常规模式下的时序图 1078
图 27-3. SPI3 / 4 四线模式下的 SPI 时序图（CKPL = 1, CKPH = 1, LF = 0） 1078
图 27-4. SPI 数据帧右对齐示意图 1079
图 27-5. SPI 数据时钟传输路径示意图 1079
图 27-6. SPI 主机接收延迟配置时序图 1080
图 27-7. SPI 从机接收延迟配置时序图 1080
图 27-8. NSS 信号延时时序图（MSSD[3:0] = 0011（3 x Tclk）, MDFD = 0011（3 x Tclk）） 1084
图 27-9. NSS 交错脉冲时序图（MSSD[3:0] = 0011（3 x Tclk）, MDFD = 0011（3 x Tclk）） 1085
图 27-10. 典型的全双工模式连接 1086
图 27-11. 典型的单工模式连接（主机：接收，从机：发送） 1086
图 27-12. 典型的单工模式连接（主机：只发送，从机：接收） 1087
图 27-13. 典型的双向线连接 1087
图 27-14. 主机 TI 模式在不连续发送时的时序图 1089
图 27-15. 主机 TI 模式在连续发送时的时序图 1089
图 27-16. 从机 TI 模式时序图 1090
图 27-17. SPI 四线模式四线写操作时序图 1091
图 27-18. SPI 四线模式四线读操作时序图 1092
图 27-19. I2S 结构框图 1097
图 27-20. I2S 飞利浦标准时序图（DTLEN = 00，CHLEN = 0，CKPL = 0） 1098
图 27-21. I2S 飞利浦标准时序图（DTLEN = 00，CHLEN = 0，CKPL = 1） 1098
图 27-22. I2S 飞利浦标准时序图（DTLEN = 10，CHLEN = 1，CKPL = 0） 1098

图 27-23. I2S 飞利浦标准时序图（DTLEN = 10，CHLEN = 1，CKPL = 1） 1098
图 27-24. I2S 飞利浦标准时序图（DTLEN = 01，CHLEN = 1，CKPL = 0） 1099
图 27-25. I2S 飞利浦标准时序图（DTLEN = 01，CHLEN = 1，CKPL = 1） 1099
图 27-26. I2S 飞利浦标准时序图（DTLEN = 00，CHLEN = 1，CKPL = 0） 1099
图 27-27. I2S 飞利浦标准时序图（DTLEN = 00，CHLEN = 1，CKPL = 1） 1099
图 27-28. MSB 对齐标准时序图（DTLEN = 00，CHLEN = 0，CKPL = 0） 1099
图 27-29. MSB 对齐标准时序图（DTLEN = 00，CHLEN = 0，CKPL = 1） 1099
图 27-30. MSB 对齐标准时序图（DTLEN = 10，CHLEN = 1，CKPL = 0） 1100
图 27-31. MSB 对齐标准时序图（DTLEN = 10，CHLEN = 1，CKPL = 1） 1100
图 27-32. MSB 对齐标准时序图（DTLEN = 01，CHLEN = 1，CKPL = 0） 1100
图 27-33. MSB 对齐标准时序图（DTLEN = 01，CHLEN = 1，CKPL = 1） 1100
图 27-34. MSB 对齐标准时序图（DTLEN = 00，CHLEN = 1，CKPL = 0） 1100
图 27-35. MSB 对齐标准时序图（DTLEN = 00，CHLEN = 1，CKPL = 1） 1100
图 27-36. LSB 对齐标准时序图（DTLEN = 01，CHLEN = 1，CKPL = 0） 1101
图 27-37. LSB 对齐标准时序图（DTLEN = 01，CHLEN = 1，CKPL = 1） 1101
图 27-38. LSB 对齐标准时序图（DTLEN = 00，CHLEN = 1，CKPL = 0） 1101
图 27-39. LSB 对齐标准时序图（DTLEN = 00，CHLEN = 1，CKPL = 1） 1101
图 27-40. PCM 标准短帧同步模式时序图（DTLEN = 00，CHLEN = 0，CKPL = 0） 1101
图 27-41. PCM 标准短帧同步模式时序图（DTLEN = 00，CHLEN = 0，CKPL = 1） 1102
图 27-42. PCM 标准短帧同步模式时序图（DTLEN = 10，CHLEN = 1，CKPL = 0） 1102
图 27-43. PCM 标准短帧同步模式时序图（DTLEN = 10，CHLEN = 1，CKPL = 1） 1102
图 27-44. PCM 标准短帧同步模式时序图（DTLEN = 01，CHLEN = 1，CKPL = 0） 1102
图 27-45. PCM 标准短帧同步模式时序图（DTLEN = 01，CHLEN = 1，CKPL = 1） 1102
图 27-46. PCM 标准短帧同步模式时序图（DTLEN = 00，CHLEN = 1，CKPL = 0） 1102
图 27-47. PCM 标准短帧同步模式时序图（DTLEN = 00，CHLEN = 1，CKPL = 1） 1103
图 27-48. PCM 标准长帧同步模式时序图（DTLEN = 00，CHLEN = 0，CKPL = 0） 1103
图 27-49. PCM 标准长帧同步模式时序图（DTLEN = 00，CHLEN = 0，CKPL = 1） 1103
图 27-50. PCM 标准长帧同步模式时序图（DTLEN = 10，CHLEN = 1，CKPL = 0） 1103
图 27-51. PCM 标准长帧同步模式时序图（DTLEN = 10，CHLEN = 1，CKPL = 1） 1103
图 27-52. PCM 标准长帧同步模式时序图（DTLEN = 01，CHLEN = 1，CKPL = 0） 1103
图 27-53. PCM 标准长帧同步模式时序图（DTLEN = 01，CHLEN = 1，CKPL = 1） 1104
图 27-54. PCM 标准长帧同步模式时序图（DTLEN = 00，CHLEN = 1，CKPL = 0） 1104
图 27-55. PCM 标准长帧同步模式时序图（DTLEN = 00，CHLEN = 1，CKPL = 1） 1104
图 27-56. I2S 时钟生成结构框图 1104
图 29-1. OSPI 八线通信模式结构框图 1132
图 29-2. OSPI 四线通信模式结构框图 1133
图 29-3. 八线模式 OSPI 命令格式 1134
图 29-4. 四线模式 OSPI 命令格式 1134
图 29-5. CSN 和 SCK 的行为 1136
图 30-1. CPDM 模块结构框图 1160
图 30-2. CPDM 延迟线长度配置流程图 1162
图 30-3. CPDM 输出时钟相位配置流程图 1162
图 31-1. DCI 模块示意图 1165
图 31-2. 硬件同步模式 1166

图 31-3. 硬件同步模式之 JPEG 格式 1167
图 32-1. TLI 模块框图 1181
图 32-2. 显示时序图 1183
图 32-3. 混合过程框图 1185
图 33-1. RSPDIF 模块框图 1202
图 33-2. S/PDIF 块及子帧格式 1203
图 33-3. 信息位编码示例 1204
图 33-4. S/PDIF 报头 1204
图 33-5. 噪声滤波及上升\下降边沿检测 1205
图 33-6. TWCNT 波形 1206
图 33-7. 同步流程图 1208
图 33-8. 同步过程时序 1209
图 33-9. RSPDIF 状态机 1210
图 33-10. RSPDIF_DATA 寄存器格式 1213
图 33-11. 通道/用户数据格式 1213
图 33-12. RSPDIF 上溢错误（RXSTEOMEN = 0 且 RXDF[1:0] = 2’b0x） 1218
图 33-13. RSPDIF 上溢错误（RXSTEOMEN = 0 且 RXDF[1:0] = 2’b10） 1218
图 33-14. RSPDIF 上溢错误（RXSTEOMEN = 1 且 RXDF[1:0] = 2’b0x） 1219
图 33-15. RSPDIF 上溢错误（RXSTEOMEN = 1 且 RXDF[1:0] = 2’b10） 1219
图 34-1. 模块框图 1232
图 34-2. 时钟分频逻辑 1233
图 34-3. FS 有效宽度 1236
图 34-4. FS 极性 1236
图 34-5. FS 功能 1236
图 34-6. Slot 激活 1237
图 34-7. 当 FSFUNC=0 时，slot 分布 1238
图 34-8. 当 FSFUNC=1 时，slot 分布 1238
图 34-9. Slot 部分的规定 1238
图 34-10. 偏移区的处理 1239
图 34-11. SD 输出管理 1239
图 34-12. 数据配置 1240
图 34-13. PDM 典型连接和时序图 1241
图 34-14. PDM 数据处理示意图 1242
图 34-15. PDM 数据传输的启动过程 1242
图 34-16. 八麦克风，不同 slot 宽度下，BxDATA 寄存器数据格式 1244
图 34-17. 四麦克风，不同 slot 宽度下，BxDATA 寄存器数据格式 1245
图 34-18. 双麦克风，不同 slot 宽度下，BxDATA 寄存器数据格式 1245
图 34-19. AC’97 的 slot 划分 1247
图 34-20. AC’97 TAG 定义 1247
图 34-21. SPDIF 数据格式 1249
图 34-22. 静音帧有效 1250
图 34-23. 压缩扩展数据通路 1251
图 34-24. 帧同步提前检测示意图 1255
图 34-25. 帧同步滞后检测示意图 1256

图 35-1. IPA 模块框图 1274
图 35-2. 从‘RGB888’到‘ARGB8888’像素格式扩展 1278
图 35-3. 从‘RGB565’到‘ARGB8888’像素格式扩展 1278
图 35-4. 从‘ARGB1555’或‘ARGB4444’到‘ARGB8888’像素格式扩展 1278
图 35-5. 抽取滤波器的采样方法 1280
图 35-6. 双线性缩放图 1281
图 35-7. 像素压缩 1283
图 35-8. 内部定时器操作 1284
图 35-9. IPA 的系统连接 1290
图 36-1. SDIO “无响应” 和 “无数据” 操作 1313
图 36-2. SDIO 多块读操作 1314
图 36-3. SDIO 多块写操作 1314
图 36-4. SDIO 数据流读操作 1314
图 36-5. SDIO 数据流写操作 1314
图 36-6. SDIO 框图 1316
图 36-7. CRU 单元 1322
图 36-8 硬件流控制时序 1325
图 36-9. 命令标记格式 1326
图 36-10. 响应令牌格式 1336
图 36-11. 1 位数据总线宽度 1339
图 36-12. 4 位数据总线宽度 1339
图 36-13. 8 位数据总线宽度 1339
图 36-14. 4 位 DDR 数据包格式 1340
图 36-15. 8 位 DDR 数据包格式 1340
图 36-16. 引导操作时序 1349
图 36-17. 备用的引导操作时序 1349
图 36-18. CMD5 时序 1357
图 36-19. CMD12 影响流操作的时序 1359
图 36-20. CMD11 电压切换时序 1360
图 37-1. CFP 管理接口结构 1380
图 37-2. MDIO 框图 1380
图 37-3. MDIO 帧格式 1381
图 37-4. MDIO 从机通讯流程 1383
图 38-1. 系统架构 1391
图 38-2. EXMC Bank 划分 1392
图 38-3. Bank0 地址映射 1393
图 38-4. NAND 地址映射 1394
图 38-5. Bank2 通用空间 1395
图 38-6. SDRAM 地址映射 1396
图 38-7. 模式 1 读时序 1399
图 38-8. 模式 1 写时序 1400
图 38-9. 模式 A 读时序 1401
图 38-10. 模式 A 写时序 1401
图 38-11. 模式 2/B 读时序 1403

图 38-12. 模式 2 写时序 1403
图 38-13. 模式 B 写时序 1403
图 38-14. 模式 C 读时序 1405
图 38-15. 模式 C 写时序 1405
图 38-16. 模式 D 读时序 1406
图 38-17. 模式 D 写时序 1407
图 38-18. 复用模式读时序 1408
图 38-19. 复用模式写时序 1408
图 38-20. 异步等待有效时的读时序 1410
图 38-21. 异步等待有效时的写时序 1410
图 38-22. 同步复用突发传输读时序 1412
图 38-23. 同步复用突发传输写时序 1413
图 38-24. NAND flash 通用存储空间操作时序 1416
图 38-25. NCE 敏感 NAND Flash 访问时序 1417
图 38-26. SDRAM 系统架构 1419
图 38-27. 突发读操作 1422
图 38-28. 数据采样时钟延迟模块 1422
图 38-29. 突发写操作 1423
图 38-30. FIFO 未命中时的读访问（BRSTRD=1，CL=2，SDCLK=2，PIPED=2） 1424
图 38-31. FIFO 命中时的读访问（BRSTRD=1） 1424
图 38-32. 跨边界读操作 1425
图 38-33. 跨边界写操作 1425
图 38-34. 自刷新模式进入和退出的处理 1426
图 38-35. 掉电模式进入和退出的处理 1426
图 39-1. VREF 连接 1444
图 40-1. LPDTS 模块框图 1448
图 40-2. 测量方式 1450
图 41-1. EDOUT 模块结构框图 1460
图 41-2. ABZ 相输出波形 1461
图 41-3. EDOUT 设置案例和 AB 相及 Z 相输出波形 1463
图 42-1. CAN 模块结构框图 1469
图 42-2. 传输延迟 1496
图 42-3. CAN 位时间 1499
图 43-1. 以太网模块框图 1531
图 43-2. MAC/带标签的 MAC 帧格式 1532
图 43-3. 站点管理接口信号 1534
图 43-4. 媒体独立接口（MII）信号线 1536
图 43-5. 精简媒体独立接口（RMII）信号线 1537
图 43-6. 描述符的环结构和链结构 1547
图 43-7. 常规发送描述符 1551
图 43-8. 增强发送描述符 1557
图 43-9. 常规接收描述符 1560
图 43-10. 增强接收描述符 1565
图 43-11. 唤醒帧过滤器寄存器 1570

图 43-12. 系统时钟精细校准方法 1571
图 43-13. MAC 控制器中断示意图 1576
图 43-14. 以太网中断示意图 1576
图 43-15. 远程唤醒帧过滤器寄存器 1585
图 44-1. 比较器框图 1624
图 44-2. 比较器迟滞 1626
图 44-3. 比较器的输出消隐 1627
图 45-1. HPDF 结构框图 1636
图 45-2. SPI 数据传输时序图 1639
图 45-3. 曼切斯特数据传输时序图 1639
图 45-4. 曼切斯特同步时序图 1640
图 45-5. 时钟丢失检测时序图 1642
图 45-6. 通道引脚重定向 1643
图 45-7. HPDF 模块外部输入数据处理流程 1651
图 45-8. HPDF 中断逻辑图 1653
图 46-1. RTDEC 结构框图 1673
图 46-2. CTR 下的 AES-128 解密流 1675
图 46-3. RTDEC 中断框图 1678
图 47-1. FAC 结构框图 1687
图 47-2. 输入缓冲区 1689
图 47-3. 循环输入缓冲区 1689
图 47-4. 循环输入缓冲区区域 1690
图 47-5. 循环输出缓冲区 1691
图 47-6. 循环输出缓冲区区域 1691
图 47-7. FIR 滤波器结构图 1693
图 47-8. IIR 滤波器结构图 1693
图 48-1. HWSEM 框图 1704
图 48-2. HWSEM 状态机 1705
图 48-3. 第一次尝试锁定操作时成功 1707
图 48-4. 第二次尝试锁定操作时成功 1707
图 48-5. 第二次尝试锁定操作时失败 1707
图 48-6. HWSEM 中断逻辑图 1708
图 49-1. USBHS 结构框图 1715
图 49-2. 在主机或设备模式下连接示意图 1717
图 49-3. OTG 模式下使用内部嵌入式 PHY 连接示意图 1718
图 49-4. 使用外部 ULPI PHY 的连接示意图 1718
图 49-5. 主机端口状态转移图 1719
图 49-6. 主机模式 FIFO 空间 1724
图 49-7. 主机模式 FIFO 访问寄存器映射表 1724
图 49-8. 设备模式 FIFO 空间 1725
图 49-9. 设备模式 FIFO 访问寄存器映射表 1725
图 50-1. ESC 的总线架构 1803
图 50-2. PME 中断待处理 1818
图 50-3. 时钟树 1825

图 50-4. HXTAL 时钟源 1826
图 50-5. 中断框图 1833
图 50-6. 端口连接 PHYS 1841
图 50-7. 模拟配置的基本结构 1842
图 50-8. 备用功能配置的基本结构 1842
图 50-9. 基本定时器结构框图 1852
图 50-10. 向下计数时序图 1853
图 50-11. 自由运行计数器结构框图 1854
图 50-12. PDI Wrapper 框图 1856
图 50-13. SPI 的 block 图 1857
图 50-14. 使能 QSPI 1861
图 50-15. 使能 OSPI 1862
图 50-16. SPI 模式复位 SPI 1862
图 50-17. QSPI 模式复位 QSPI 1863
图 50-18. OSPI 模式复位 OSPI 1863
图 50-19. SPI 读取 1864
图 50-20. QSPI 读取 1864
图 50-21. OSPI 读取 1865
图 50-22. SPI 双输出读取 1866
图 50-23. SPI 四线输出读取 1867
图 50-24. SPI 双 I/O 读取 1868
图 50-25. SPI 四线 I/O 读取 1869
图 50-26. SPI 写入 1870
图 50-27. QSPI 写入 1870
图 50-28. OSPI 写入 1871
图 50-29. SPI 双数据写入 1872
图 50-30. SPI 四线数据写入 1873
图 50-31. SPI 双地址/数据写入 1874
图 50-32. SPI 四路地址/数据写入 1875
图 50-33. PHY 功能模块图 1876
图 50-34. LED 连接图 1878
图 50-35.100Base-TX LPI 1879
图 50-36. EtherCAT 系统框图 1923
图 50-37. EtherCAT 状态机 1928
图 50-38. EEPROM 布局 1929
图 50-39. PDI 中断屏蔽和中断信号 1930
图 50-40. EtherCAT 中断屏蔽 1931

表索引

表 1-1. 互联矩阵的互联关系 60
表 1-2. GD32H73x_75x 系列器件的存储器映射表（适用于 GD32H737_757_759） 66
表 1-3. GD32H73x_75x 系列器件的存储器映射表（适用于 GD32H75E） 73
表 1-4. ITCM/DTCM/AXI SRAM 的配置 83
表 1-5. 引导模式选择 84
表 1-6. 引导模式详细描述（适用于 GD32H737_757_759） 84
表 1-7. 引导模式详细描述（适用于 GD32H75E） 85
表 1-8. ASIB 配置 87
表 1-9. AMIB 配置 87
表 2-1. Region 0 的 RAMECC 监视器单元 x（x=0..4） 136
表 2-2. Region 1 的 RAMECC 监视器单元 x（x=0..2） 137
表 3-1. GD32H73x_75x 闪存基地址和构成 143
表 3-2. 选项字节 150
表 3-3. 扇区保护 WP 位 153
表 3-4. SPC 保护等级配置 155
表 3-5. DCRP 区域配置 156
表 3-6. 安全用户区域配置 157
表 3-7. 函数 resetAndInitializeSecureAreas 159
表 3-8. 函数 exitSecureArea 160
表 3-9. FMC 中断请求 162
表 4-1. 求系统参数 183
表 4-2. EFUSE 中断请求 186
表 5-1 供电模式 205
表 5-2. 节电模式总结 208
表 6-1. 时钟输出 0 的时钟源选择 228
表 6-2. 时钟输出 1 的时钟源选择 229
表 8-1. Cortex®-M7 中的 NVIC 异常类型 398
表 8-2. 中断向量表（适用于 GD32H737_757_759） 399
表 8-3. 中断向量表（适用于 GD32H75E） 404
表 8-4. EXTI 触发源（适用于 GD32H737_757_759） 410
表 8-5. EXTI 触发源（适用于 GD32H75E） 411
表 9-1. 触发输入位域选择 420
表 9-2. TRIGSEL 输入输出映射关系 424
表 10-1. GPIO 配置表 460
表 12-1. ALGO 配置 491
表 15-1. 输入数据配置 541
表 15-2. 输出数据配置 541
表 15-3. TMU 模式配置 542
表 15-4. 模式 0 描述 542
表 15-5. 模式 1 描述 543
表 15-6. 模式 2 描述 544

表 15-7. 模式 3 描述 545
表 15-8. 模式 4 描述 546
表 15-9. 模式 5 描述 546
表 15-10. 模式 6 描述 547
表 15-11. 模式 7 描述 547
表 15-12. 模式 8 描述 548
表 15-13. 模式 8 推荐的缩放因子 548
表 15-14. 模式 9 描述 549
表 15-15. 模式 9 推荐的缩放因子 549
表 15-16. 不同迭代次数下的精度 549
表 16-1. 传输模式 556
表 16-2. CNT 配置 559
表 16-3. FIFO 计数器临界值配置 560
表 16-4. DMA 中断事件 566
表 17-1. 传输模式 581
表 17-2. MDMA 硬件请求源 582
表 17-3. 源和目标地址生成配置 586
表 17-4. 源和目的地址更新方式 587
表 17-5. 寄存器加载地址 588
表 17-6. MDMA 错误标志 589
表 17-7. MDMA 中断事件 589
表 18-1. 中断事件 609
表 18-2. DMAMUX 请求路由输入信号映射 610
表 18-3. 触发输入信号映射 615
表 18-4. 同步输入信号映射 616
表 19-1. 引脚分配 622
表 20-1. ADC 内部输入信号 634
表 20-2. ADC 输入引脚定义 634
表 20-3. ADC 差分通道引脚匹配 636
表 20-4. ADC0/ADC1/ADC2 常规通道的触发源 641
表 20-5. ADC0 和 ADC1 不同分辨率对应的 t_{CONV} 时间 644
表 20-6. ADC2 不同分辨率对应的 t_{CONV} 时间 644
表 20-7. 12 位 ADC 部分举例 N 和 M 的最大输出值（灰色部分表示截断） 646
表 20-8. ADC 同步模式表 647
表 21-1. DAC 引脚 675
表 21-2. DAC 触发与输出 675
表 21-3. DAC 外部触发 676
表 22-1. 独立看门狗定时器在 32kHz (IRC32K)时的最小/最大超时周期 696
表 22-2. 在 150MHz (f_{PCLK3})时的最大/最小超时值 702
表 23-1. RTC 引脚配置和功能 715
表 23-2. 省电模式管理 716
表 23-3. RTC 中断控制 716
表 24-1. 定时器（TIMERx）分为 5 种类型 735
表 24-2. 高级定时器通道介绍 738

表 24-3 复合 PWM 脉冲宽度 752
表 24-4. 由参数控制的互补输出表（MCHxMSEL=2'b11） 757
表 24-5. 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） 761
表 24-6. 中止功能锁存/释放条件 763
表 24-7. 不同译码器模式下的计数方向 764
表 24-8. 非正交译码器模式 1 的计数情况 766
表 24-9. 从模式例子列表 768
表 24-10. 复合 PWM 脉冲宽度 851
表 24-11. 不同译码器模式下的计数方向 856
表 24-12. 非正交译码器模式 1 的计数情况 859
表 24-13. 从模式列表和举例（通用定时器 L0） 859
表 24-14. 复合 PWM 脉冲宽度 903
表 24-15. 由参数控制的互补输出表（MCHxMSEL=2'b11） 908
表 24-16. 中止功能锁存/释放条件 912
表 24-17. 从模式例子列表 912
表 24-18. 由参数控制的互补输出表（MCHxMSEL=2'b11） 957
表 24-19. 中止功能锁存/释放条件 961
表 25-1. USART 重要引脚描述 1000
表 25-2. 停止位配置 1002
表 25-3. USART 中断请求 1014
表 26-1. I2C 总线术语说明（参考飞利浦 I2C 规范） 1039
表 26-2. 数据建立时间和数据保持时间 1044
表 26-3. 可关闭通信模式 1046
表 26-4. I2C 错误标志 1060
表 26-5. I2C 中断事件 1061
表 27-1. SPI 信号描述 1076
表 27-2. SPI 四线信号描述 1077
表 27-3. MISO / MISO 信号交换描述 1077
表 27-4. SPIX FIFO 最大存储数据帧数量 1081
表 27-5. 从机模式 NSS 功能 1083
表 27-6. 主机模式 NSS 功能 1084
表 27-7. SPI 运行模式 1085
表 27-8. SPI 中断请求 1096
表 27-9. I2S 比特率计算公式 1104
表 27-10. 音频采样频率计算公式 1105
表 27-11. I2SX FIFO 最大存储数据帧数量 1105
表 27-12. 各种运行模式下 I2S 接口信号的方向 1106
表 27-13. I2S 中断 1109
表 28-1. OSPIM 矩阵映射 1128
表 29-1. OSPI 信号线描述 1130
表 29-2. SPI 中断请求 1140
表 31-1. DCI 引脚 1166
表 31-2. 典型的单行数据组成 1167
表 31-3. SAV 和 EAV 编码 1168

表 31-4. 图像格式定义 1168
表 31-5. 逐行模式的一般情况....1168
表 31-6. 字节填充模式下的存储视图....1170
表 31-7. 半字填充模式下的存储视图....1170
表 31-8. 状态/错误标志 1170
表 32-1. TLI 提供的显示接口的引脚 1182
表 32-2. 八种像素格式 1184
表 32-3. 状态标志....1186
表 32-4. 错误标志....1186
表 33-1. 子帧格式.... 1203
表 33-2. 报头模式.... 1204
表 33-3. RSPDIF 时间间隔 1205
表 33-4. 转换编码器编码规则.... 1206
表 33-5. 临界值计算 1207
表 33-6. 报头的转换序列.... 1207
表 33-7. 双相解码器解码规则.... 1207
表 33-8. RSPDIF 状态.... 1209
表 33-9. 不同 RSPDIF 状态下 RSPDIF_CTL 比特位访问特性 1211
表 33-10. 混合数据和控制流 1214
表 33-11. 最小 rspdif_ck 频率与音频采样率 1215
表 33-12. 符号时钟 spdif_symbol_ck 生成条件.... 1215
表 33-13. RSPDIF 中断事件 1219
表 34-1. 常用的音频采用率 1233
表 34-2. 外部同步配置 1235
表 34-3. FIFO 请求的产生条件 1240
表 34-4. 不同配置下，获取麦克风数据需要读取寄存器次数 1243
表 34-5. TDM 配置表 1246
表 34-6. AC’97 发送 slot 定义 1248
表 34-7. AC’97 接收 slot 定义 1248
表 34-8. SOPD 模式.... 1249
表 34-9. 校验位奇数 1249
表 34-10. 静音帧输出值 1251
表 34-11. A-law 编码.... 1252
表 34-12. A-law 解码 1252
表 34-13. Mu-law 编码.... 1252
表 34-14. Mu-law 解码.... 1253
表 34-15. 中断控制.... 1256
表 35-1. IPA 信号描述.... 1274
表 35-2. IPA 转换模式.... 1275
表 35-3. 前景层和背景层 CLUT 像素格式.... 1276
表 35-4. 前景层和背景层像素格式 1277
表 35-5. YUV 和 YCbCr 模式的期望系数.... 1279
表 35-6. Alpha 通道值调制 1279
表 35-7. 目标像素格式 1282

表 35-8. IPA 中断事件 1288
表 36-1. 两个 SDIO 1312
表 36-2. SDIO 适用 SD & SD/IO 卡的操作模式 1315
表 36-3. SDIO 适用 e•MMC 卡的操作模式 1315
表 36-4. SDIO 内部输入/输出信号 1316
表 36-5. SDIO 引脚介绍 1316
表 36-6. SDIO 命令和数据输出的相位选择 1317
表 36-7. SDIO 与 MDMA 的连接 1324
表 36-8. AHB 和 e•MMC 时钟频率关系 1324
表 36-9. AHB 和 SD/SD I/O 卡时钟频率关系 1324
表 36-10. 命令格式 1327
表 36-11. 卡命令类 （CCCs） 1327
表 36-12. 基本命令（class 0） 1329
表 36-13. 面向块的读命令（class 2） 1331
表 36-14. 流读取命令（class 1）和流写入命令（class 3） 1332
表 36-15. 面向块的写命令（class 4） 1332
表 36-16. 擦除命令（class 5） 1333
表 36-17. 面向块的写保护命令（class 6） 1333
表 36-18. 锁卡命令（class 7） 1333
表 36-19. 特定应用命令（class 8） 1334
表 36-20. I/O 模式命令（class 9） 1334
表 36-21. 切换功能命令（class 10） 1335
表 36-22. R1 响应 1336
表 36-23. R2 响应 1337
表 36-24. R3 响应 1337
表 36-25. R4 响应（e•MMC） 1337
表 36-26. R4 响应（SD I/O） 1337
表 36-27. R5 响应（MMC） 1338
表 36-28. R5 响应（SD I/O） 1338
表 36-29. R6 响应 1338
表 36-30. R7 响应 1338
表 36-31. 卡状态 1341
表 36-32. SD 状态 1343
表 36-33. 移动性能字段 1345
表 36-34. AU_SIZE 字段 1345
表 36-35. 最大 AU 大小 1345
表 36-36. 擦除大小字段 1346
表 36-37. 擦除超时字段 1346
表 36-38. 擦除偏移字段 1346
表 36-39. 上锁/解锁数据结构 1355
表 36-40. CMD12 的用法 1358
表 36-41. 不同响应类型对应的 SDIO_RESPx 寄存器 1367
表 37-1. MDIO 引脚定义 1379
表 37-2. 不同帧类型具体描述(1) 1381

表 37-3. 操作码 1382
表 38-1. EXMC bank 映射 1393
表 38-2. SDRAM 地址映射 1396
表 38-3. NOR Flash 接口信号描述 1396
表 38-4. PSRAM 非复用接口信号描述 1397
表 38-5. EXMC Bank0 支持的访问模式 1397
表 38-6. NOR/PSRAM 控制时序参数 1398
表 38-7. EXMC 时序模型 1398
表 38-8. 模式 1 相关寄存器配置 1400
表 38-9. 模式 A 相关寄存器配置 1401
表 38-10. 模式 2/B 相关寄存器配置 1403
表 38-11. 模式 C 相关寄存器配置 1405
表 38-12. 模式 D 相关寄存器配置 1407
表 38-13. 复用模式相关寄存器配置 1408
表 38-14. 同步复用模式读时序配置 1412
表 38-15. 同步复用模式写时序配置 1413
表 38-16. 8 位/16 位 NAND 接口信号描述 1414
表 38-17. EXMC Bank2 支持的访问模式 1414
表 38-18. NAND flash 可编程参数 1415
表 38-19. SDRAM 命令真值表 1419
表 38-20. SDRAM IO 口定义 1420
表 39-1. VREF 模式 1445
表 40-1. LPDTS 信号 1449
表 40-2. 采样时间设置 1450
表 40-3. 触发源设置 1451
表 40-4. 低功耗描述 1452
表 40-5. 低功耗模式下温度传感器行为 1452
表 42-1. 64 字节数据的邮箱描述符 1470
表 42-2. DLC 表示的数据字节长度 1471
表 42-3. 接收邮箱 CODE 1472
表 42-4. 发送邮箱 CODE 1472
表 42-5. 邮箱数目 1474
表 42-6. 接收 FIFO 描述符 1474
表 42-7. 当本地优先级禁用时的邮箱仲裁值（32 位） 1483
表 42-8. 当本地优先级使能时的邮箱仲裁值（35 位） 1483
表 42-9. 接收邮箱匹配 1489
表 42-10. Rx FIFO 匹配 1490
表 42-11. 中断事件 1501
表 42-12. Rx FIFO 标识符过滤表元素数目 1514
表 43-1. 以太网信号（MII） 1532
表 43-2. 以太网信号（RMII） 1533
表 43-3. 时钟范围 1535
表 43-4. 接收接口信号编码 1536
表 43-5. 目标地址过滤器结果列表 1541

表 43-6. 源地址过滤器结果列表 1542
表 43-7. 接收描述符 0 错误状态描述，仅适用于常规描述符（DFM=0） 1563
表 43-8. 支持的 PTP 时间戳及其寄存器配置 1603
表 44-1. CMP 的输入和输出总结 1625
表 45-1. HPDF 引脚定义 1636
表 45-2. HPDF 断路连接 1636
表 45-3. SPI 接口时钟配置 1638
表 45-4. 并行数据封装模式 1645
表 45-5. 注入组的触发信号 1647
表 45-6. Sinc^x 滤波的最大输出分辨率与过采样滤的关系 1648
表 45-7. 积分器的最大输出分辨率与 IOR、SFOR、SFO 之间的关系 1649
表 45-8. 阈值监视器工作模式特点 1649
表 45-9. 最大输出速率 1652
表 45-10. HPDF 中断事件 1653
表 46-1. RTDEC 中断请求 1678
表 47-1. IEEE 32 位单精度浮点格式 1694
表 48-1. 中断事件 1708
表 49-1. USBHS 信号线描述 1715
表 49-2. USBHS 支持速度列表 1716
表 49-3. USBHS 全局中断 1732
表 50-1. ESC 存储映射表 1804
表 50-2. 有效数据的对齐 1807
表 50-3. 节能模式总结 1819
表 50-4. GPIO 配置表 1839
表 50-5. 4 线模式 1857
表 50-6. 6 线模式 1857
表 50-7. 4 线模式下的 OSPI 1857
表 50-8. SPI 指令 1859
表 50-9. QSPI 指令 1860
表 50-10. OSPI 指令 1860
表 50-11. EtherCAT 的 PDI 1924
表 50-12. 通过写入确认 PDI 寄存器功能的受影响寄存器 1925
表 51-1. 寄存器功能位访问属性 1993
表 51-2. 术语 1993
表 52-1. 版本历史 1994

1. 系统及存储器架构

GD32H73x_75x系列器件是基于Arm® Cortex®-M7处理器的32位通用微控制器。Arm® Cortex®-M7处理器包括64位AMBA4 AXI、32位AHB外设（AHBP）端口、32位AHB从设备端口，用于外部主设备访问内存，以及用于CoreSight调试组件的APB接口。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®-M7 处理器

Arm® Cortex®-M7处理器是一种高效、高性能的嵌入式处理器，具有低中断延迟、低调试成本以及与现有Cortex-M配置文件处理器的向后兼容性。处理器有一个有序的超标量流水线，因为有多个内存接口，这意味着许多指令可以双重发布，包括加载/加载和加载/存储指令对。Cortex®-M7是一款高性能处理器，具有带分支预测的6级超标量流水线和可选的FPU，能够进行单精度和可选的双精度运算。指令和数据总线已比以前的32位总线扩大到64位宽。

处理器支持的接口包括：

- 64位AXI4接口
- 32位AHB主机接口
- 32位AHB从机接口
- 64位指令TCM接口
- 2x32位数据TCM接口

处理器包含以下外部接口：

- AHBP接口
- AHBS接口
- AHBD接口
- 外部专用外围总线
- ATB接口
- TCM接口
- 交叉触发接口
- MBIST接口
- AXIM接口

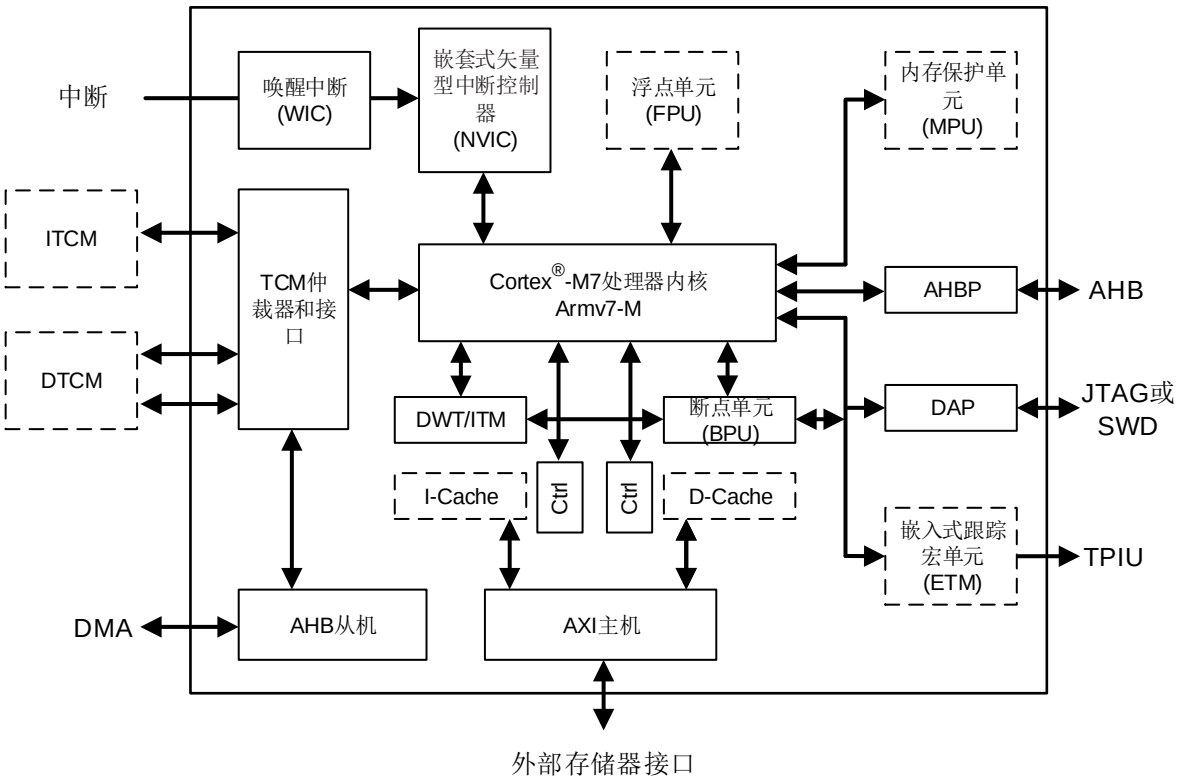
Cortex®-M7处理器基于ARMv7架构，并且支持一个强大且可扩展的指令集，包括通用数据处理I/O控制任务、增强的数据处理位域操作、DSP(数字信号处理)和浮点运算指令。下面列出由Cortex®-M7提供的一些系统外设：

- 嵌套式向量型中断控制器（NVIC）
- 闪存地址重载及断点单元（FPB）
- 数据观测点及跟踪单元（DWT）
- 指令跟踪宏单元（ITM）
- JTAG或SWD调试接口
- 跟踪端口接口单元（TPIU）
- 内存保护单元（MPU）

- 双精度浮点运算单元（FPU）
- 加载存储单元（LSU）
- 数据处理单元（DPU）
- 预取单元（PFU）

*图1-1. Cortex®-M7处理器结构框图*显示了Cortex®-M7处理器结构框图。欲了解更多信息，请参阅Arm® Cortex®-M7技术参考手册。

图 1-1. Cortex®-M7 处理器结构框图



1.2. 系统架构

互连矩阵包括一个AXI总线矩阵和两个AHB总线矩阵，可实现系统中多个主设备和从设备之间的并行访问路径。互连矩阵的互连关系如下所示。在*表1-1. 互联矩阵的互联关系*中，“1”表示对应的主机可以通过互连矩阵访问对应的从机，空白的单元格表示对应的主机不能通过互连矩阵访问对应的从机。

表 1-1. 互联矩阵的互联关系

主机接口 / 从机接口	AXIM	AHBP	ITCM	DTCM	MDMA AXI	MDMA AHBS	DMA0 MEM	DMA0 PERIPH	DMA1 MEM	DMA1 PERIPH	IPA	TLI	SDIO0	SDIO1	ENET	USBHS0	USBHS1
ITCM			1			1											
DTCM				1		1											

主机接口 / 从机接口	AXIM	AHBP	ITCM	DTCM	MDMA AXI	MDMA AHBS	DMA0 MEM	DMA0 PERIPH	DMA1 MEM	DMA1 PERIPH	IPA	TLI	SDIO0	SDIO1	ENET	USBHS0	USBHS1
FMC	1				1		1	1	1	1	1	1	1	1	1	1	1
AXI SRAM	1				1		1	1	1	1	1	1	1	1	1	1	1
共享 RAM (ITCM/DTCM/AXI)	1				1		1	1	1	1	1	1	1	1	1	1	1
SRAM0	1				1		1	1	1	1	1			1	1	1	1
SRAM1	1				1		1	1	1	1	1			1	1	1	1
Backup RAM	1				1		1	1	1	1				1			1
AHB1	1	1			1		1	1	1	1	1			1			
AHB2		1					1	1	1	1				1			
AHB3	1				1												
AHB4	1				1		1	1	1	1				1			1
APB1		1					1	1	1	1				1			
APB2		1					1	1	1	1				1			
APB3	1				1												
APB4	1				1		1	1	1	1				1			1
EXMC	1				1		1	1	1	1	1	1	1	1	1	1	1
OSPI	1				1		1	1	1	1	1	1	1	1	1	1	1

GD32H73x_75x器件的系统架构如*图1-2. GD32H73x_75x器件的系统架构示意图（适用于GD32H737_757_759）*和*图1-3. GD32H73x_75x器件的系统架构示意图（适用于GD32H75E）*所示。工作频率与电源供电电压相关，具体请参考数据手册。

图 1-2. GD32H73x_75x 器件的系统架构示意图（适用于 GD32H737_757_759）

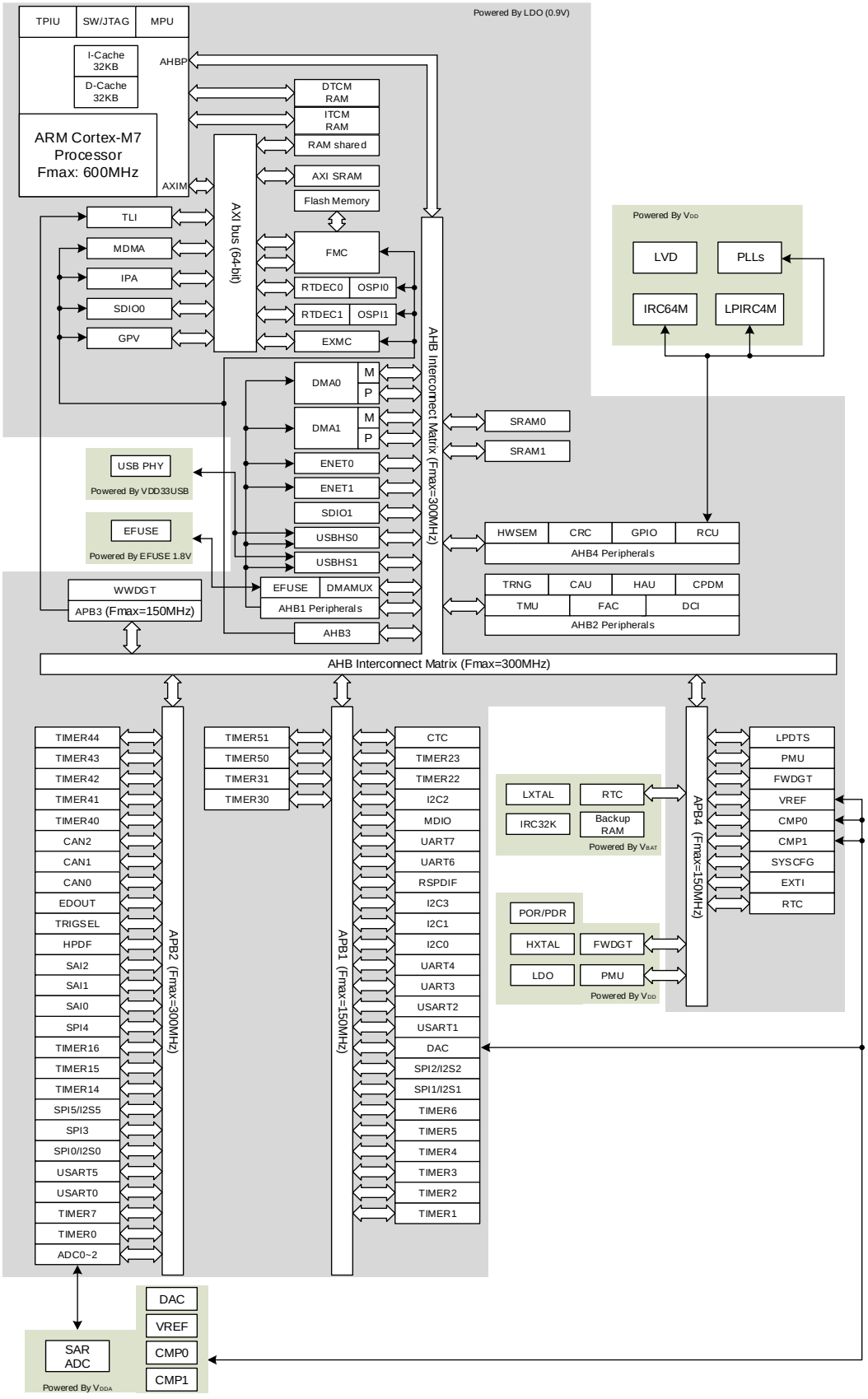
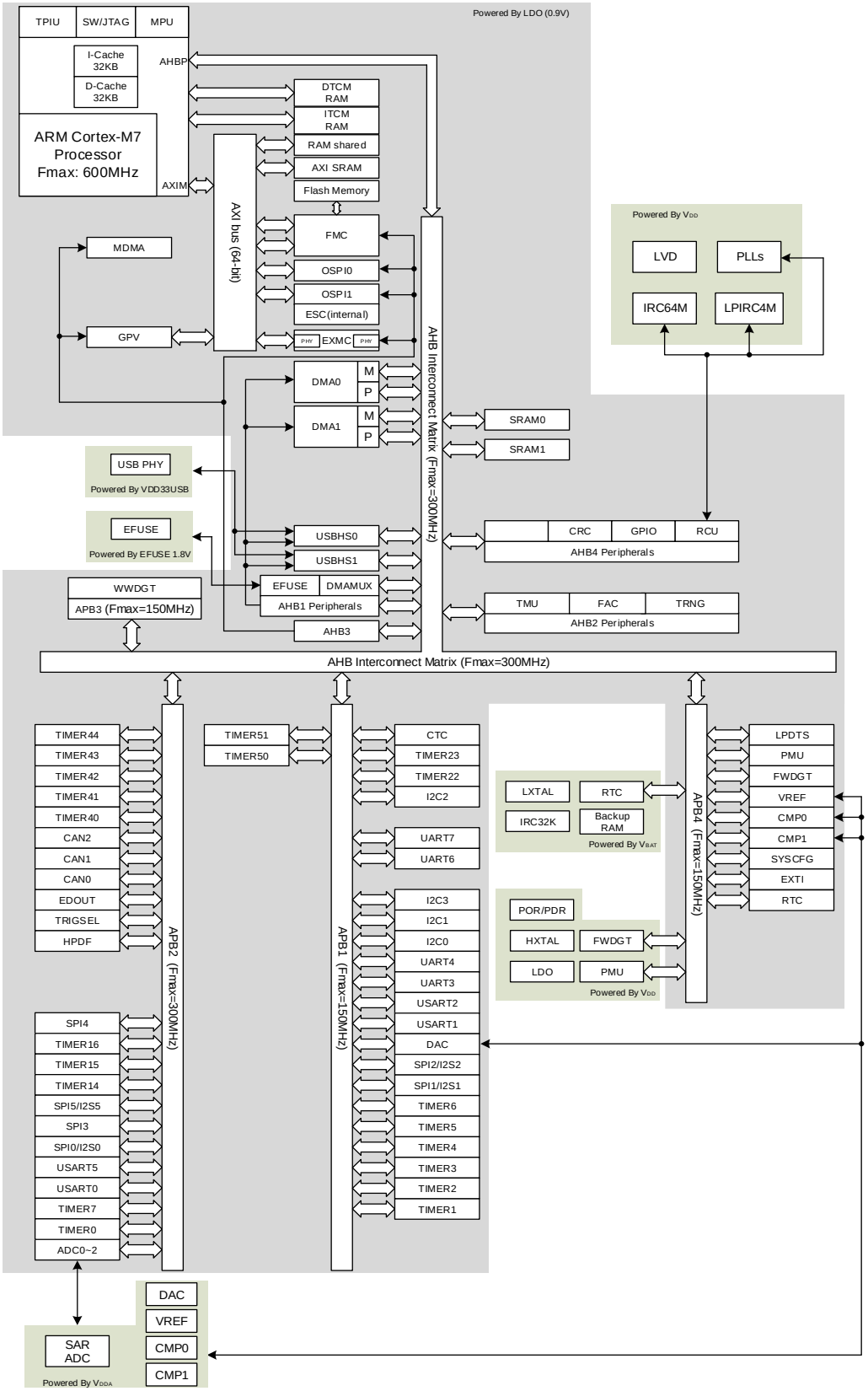


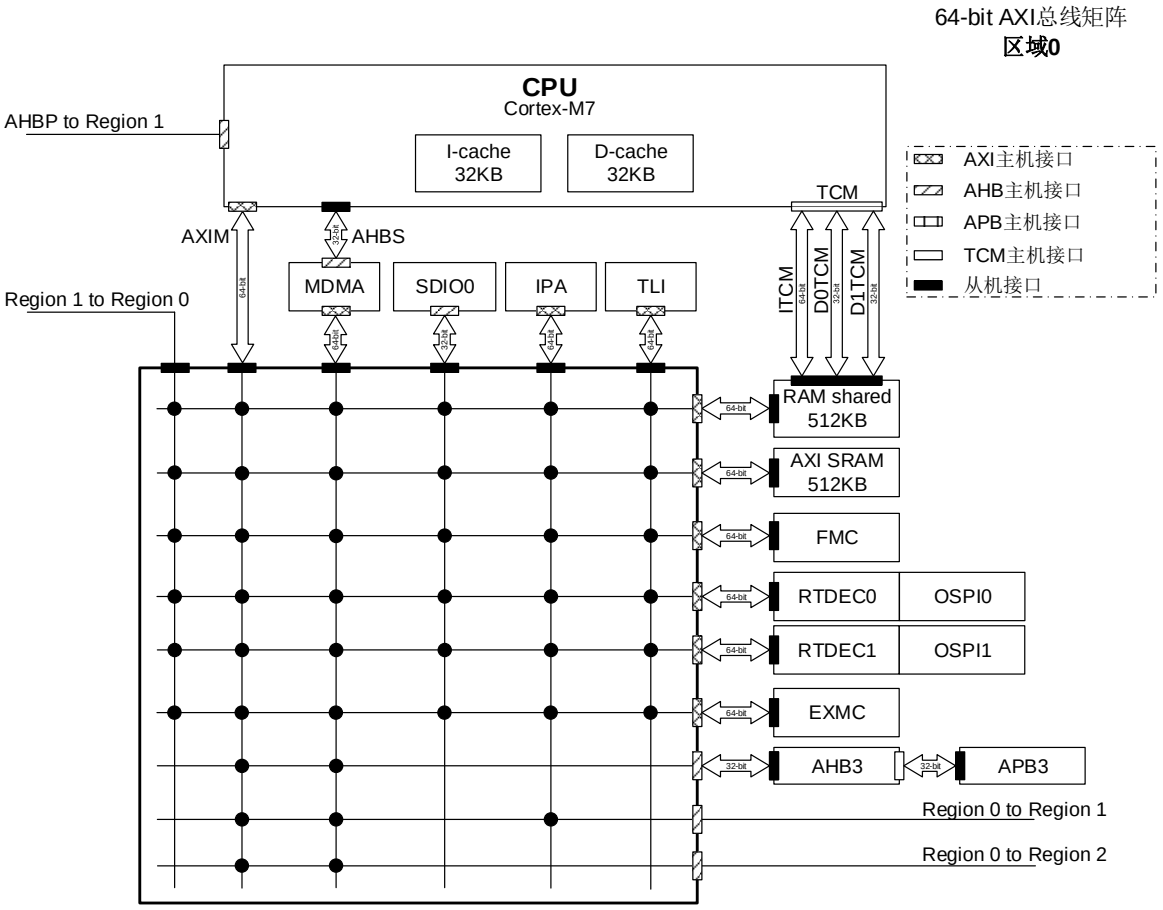
图 1-3. GD32H73x_75x 器件的系统架构示意图（适用于 GD32H75E）



1.2.1. 总线矩阵区域 0

64位AXI总线矩阵区域0如*图1-4. 总线矩阵区域0*所示。

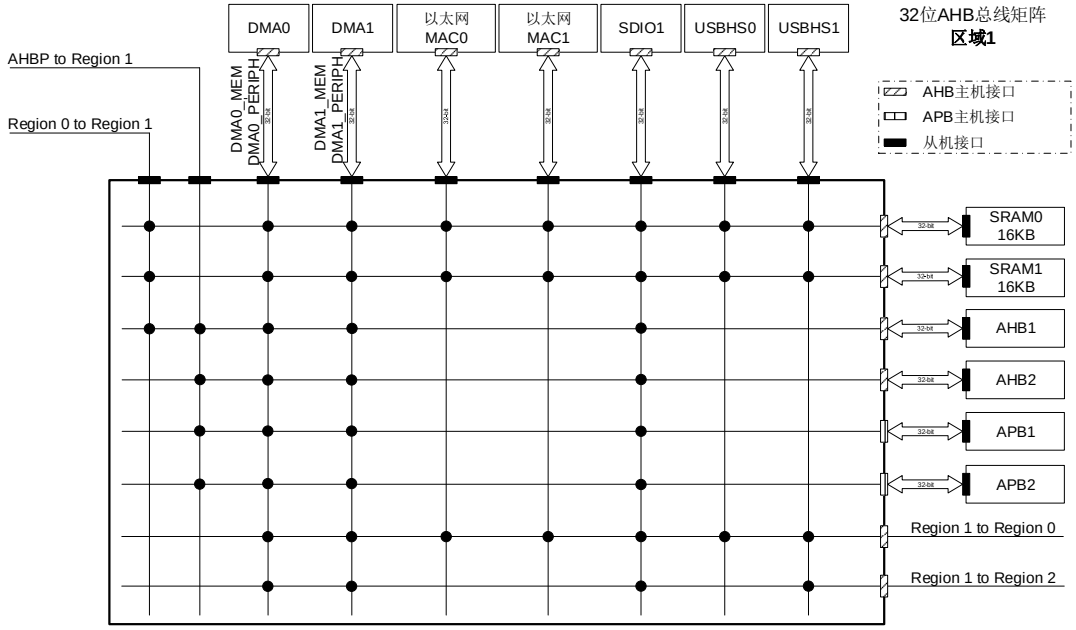
图 1-4. 总线矩阵区域 0



1.2.2. 总线矩阵区域 1

32位AHB总线矩阵区域1如*图1-5. 总线矩阵区域1*所示。

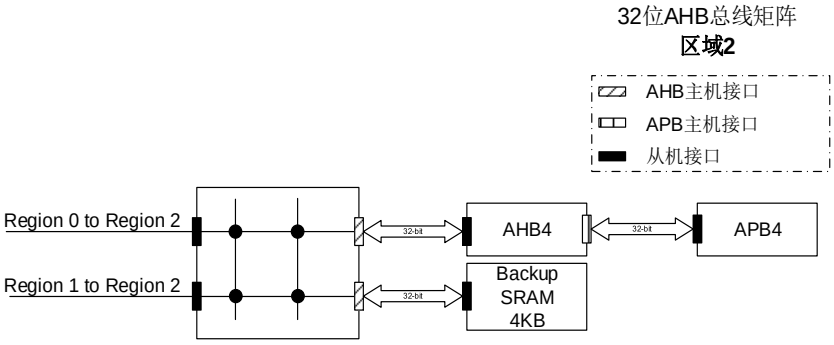
图 1-5. 总线矩阵区域 1



1.2.3. 总线矩阵区域 2

32位AHB总线矩阵区域2如*图1-6. 总线矩阵区域2*所示。

图 1-6. 总线矩阵区域 2



如上图所示，总线矩阵有3个区域（Region），包括I-Cache、D-Cache、TCM、MDMA、DMA0、DMA1、IPA、TLI、SDIO0、SDIO1、ENET、USBHS0和USBHS1。Region 0中的AXI总线矩阵和Region 1和Region 2中的AHB总线矩阵为多主到多从的并发访问提供保证和仲裁。仲裁在Region 0中采用带QoS功能的循环调度算法，Region 1和Region 2采用循环调度算法。ITCM和DTCM通过TCM总线直接连接到Cortex®-M7内核，ITCM和DTCM的访问为零等待。ENET是以太网。TLI是TFT LCD接口。USBHS是高速USB。IPA是图像处理加速器。

1.3. 存储器映射

Arm® Cortex®-M7处理器采用哈佛结构，可以使用相互独立的总线来读取指令和加载/存储数据。*表1-2. GD32H73x_75x系列器件的存储器映射表（适用于GD32H737_757_759）*和*表1-3. GD32H73x_75x系列器件的存储器映射表（适用于GD32H75E）*显示了GD32H73x_75x

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. GD32H73x_75x 系列器件的存储器映射表（适用于 GD32H737_757_759）

预先定义的地址空间	总线	地址范围	外设
外部 RAM		0xD000 0000 - 0xDFFF FFFF	EXMC - SDRAM 设备 1
		0xC000 0000 - 0xCFFF FFFF	EXMC – SDRAM 设备 0 (EXMC Bank 0 Region 0-3)
		0xA000 1000 - 0xBFFF FFFF	保留
		0xA000 0000 - 0xA000 0FFF	保留
		0x9000 0000 - 0x9FFF FFFF	OSPI0
		0x8000 0000 - 0x8FFF FFFF	EXMC - NAND
		0x7000 0000 - 0x7FFF FFFF	OSPI1
		0x6000 0000 - 0x6FFF FFFF	EXMC - NOR/PSRAM/SRAM
外设	AHB4	0x5802 7000 - 0x5FFF FFFF	保留
		0x5802 6400 - 0x5802 67FF	HWSEM
		0x5802 6000 - 0x5802 63FF	保留
		0x5802 5000 - 0x5802 5FFF	保留
		0x5802 4C00 - 0x5802 4FFF	CRC
		0x5802 4800 - 0x5802 4BFF	保留
		0x5802 4400 - 0x5802 47FF	RCU
		0x5802 2C00 - 0x5802 43FF	保留
		0x5802 2800 - 0x5802 2BFF	GPIOK
		0x5802 2400 - 0x5802 27FF	GPIOJ
		0x5802 2000 - 0x5802 23FF	保留
		0x5802 1C00 - 0x5802 1FFF	GPIOH
		0x5802 1800 - 0x5802 1BFF	GPIOG
		0x5802 1400 - 0x5802 17FF	GPIOF
		0x5802 1000 - 0x5802 13FF	GPIOE
		0x5802 0C00 - 0x5802 0FFF	GPIOD
		0x5802 0800 - 0x5802 0BFF	GPIOC
		0x5802 0400 - 0x5802 07FF	GPIOB
		0x5802 0000 - 0x5802 03FF	GPIOA
		0x5801 0000 - 0x5801 FFFF	保留
	APB4	0x5800 7400 - 0x5800 FFFF	保留
		0x5800 7000 - 0x5800 73FF	保留
		0x5800 6C00 - 0x5800 6FFF	保留
		0x5800 6800 - 0x5800 6BFF	LPDTS
		0x5800 5800 - 0x5800 67FF	PMU
		0x5800 5400 - 0x5800 57FF	保留
		0x5800 4C00 - 0x5800 53FF	保留
		0x5800 4800 - 0x5800 4BFF	FWDGT

预先定义的地址空间	总线	地址范围	外设
		0x5800 4000 - 0x5800 43FF	RTC
		0x5800 3C00 - 0x5800 3FFF	VREF
		0x5800 3800 - 0x5800 3BFF	CMP0 - CMP1
		0x5800 3400 - 0x5800 37FF	保留
		0x5800 3000 - 0x5800 33FF	保留
		0x5800 2C00 - 0x5800 2FFF	保留
		0x5800 2800 - 0x5800 2BFF	保留
		0x5800 2400 - 0x5800 27FF	保留
		0x5800 2000 - 0x5800 23FF	保留
		0x5800 1C00 - 0x5800 1FFF	保留
		0x5800 1400 - 0x5800 17FF	保留
		0x5800 0800 - 0x5800 13FF	保留
		0x5800 0400 - 0x5800 07FF	SYSCFG
		0x5800 0000 - 0x5800 03FF	EXTI
	AHB3	0x5200 C000 - 0x57FF FFFF	保留
		0x5200 BC00 - 0x5200 BFFF	RTDEC1
		0x5200 B800 - 0x5200 BBFF	RTDEC0
		0x5200 B400 - 0x5200 B7FF	OSPIM
		0x5200 B000 - 0x5200 B3FF	保留
		0x5200 A000 - 0x5200 AFFF	OSPI1
		0x5200 9400 - 0x5200 9FFF	保留
		0x5200 9000 - 0x5200 93FF	RAMECCMU Region 0
		0x5200 8000 - 0x5200 8FFF	CPDM(SDIO0)
		0x5200 7000 - 0x5200 7FFF	SDIO0
		0x5200 6000 - 0x5200 6FFF	保留
		0x5200 5000 - 0x5200 5FFF	OSPI0
		0x5200 4000 - 0x5200 4FFF	EXMC
		0x5200 3400 - 0x5200 3FFF	保留
		0x5200 3000 - 0x5200 33FF	保留
		0x5200 2000 - 0x5200 2FFF	FMC
		0x5200 1000 - 0x5200 1FFF	IPA
		0x5200 0000 - 0x5200 0FFF	MDMA
		0x5110 0000 - 0x511F FFFF	保留
		0x5100 0000 - 0x510F FFFF	AXI 互联矩阵
	APB3	0x5006 1000 - 0x50FF FFFF	保留
		0x5006 0C00 - 0x5006 0FFF	保留
		0x5006 0800 - 0x5006 0BFF	保留
		0x5006 0400 - 0x5006 07FF	保留
		0x5006 0000 - 0x5006 03FF	保留
		0x5005 0400 - 0x5005 FFFF	保留
		0x5005 0000 - 0x5005 03FF	保留

预先定义的地址空间	总线	地址范围	外设
		0x5004 0000 - 0x5004 FFFF	保留
		0x5000 0000 - 0x5003 FFFF	保留
		0x5000 3000 - 0x5000 3FFF	WWDGT
		0x5000 2000 - 0x5000 2FFF	保留
		0x5000 1000 - 0x5000 1FFF	TLI
		0x5000 0000 - 0x5000 0FFF	保留
	AHB2	0x4802 5000 - 0x4FFF FFFF	保留(AHB2)
		0x4802 4800 - 0x4802 4FFF	FAC
		0x4802 4400 - 0x4802 47FF	TMU
		0x4802 4000 - 0x4802 43FF	保留
		0x4802 3000 - 0x4802 3FFF	RAMECCMU Region 1
		0x4802 2C00 - 0x4802 2FFF	保留(AHB2)
		0x4802 2800 - 0x4802 2BFF	CPDM(SDIO1)
		0x4802 2400 - 0x4802 27FF	SDIO1
		0x4802 1C00 - 0x4802 23FF	保留(AHB2)
		0x4802 1800 - 0x4802 1BFF	TRNG
		0x4802 1400 - 0x4802 17FF	HAU
		0x4802 1000 - 0x4802 13FF	CAU
		0x4802 0400 - 0x4802 0FFF	保留(AHB2)
		0x4802 0000 - 0x4802 03FF	DCI
		0x4800 1800 - 0x4801 FFFF	保留(AHB2)
		0x4800 1400 - 0x4800 17FF	保留
		0x4800 1000 - 0x4800 13FF	保留
		0x4800 0C00 - 0x4800 0FFF	保留
		0x4800 0800 - 0x4800 0BFF	保留
		0x4800 0400 - 0x4800 07FF	保留
		0x4800 0000 - 0x4800 03FF	保留
	AHB1	0x400C 0000 - 0x47FF FFFF	保留(AHB1)
		0x4008 0000 - 0x400B FFFF	USBHS1
		0x4004 0000 - 0x4007 FFFF	USBHS0
		0x4003 8C00 - 0x4003 FFFF	保留
		0x4003 8400 - 0x4003 8BFF	保留
		0x4003 8000 - 0x4003 83FF	保留
		0x4003 3000 - 0x4003 7FFF	保留
		0x4003 0000 - 0x4003 2FFF	保留
		0x4002 C000 - 0x4002 FFFF	保留
		0x4002 BC00 - 0x4002 BFFF	ENET1
		0x4002 B000 - 0x4002 BBFF	
		0x4002 A000 - 0x4002 AFFF	
		0x4002 8000 - 0x4002 9FFF	ENET0
		0x4002 6800 - 0x4002 7FFF	保留

预先定义的地址空间	总线	地址范围	外设
		0x4002 6400 - 0x4002 67FF	保留
		0x4002 6000 - 0x4002 63FF	保留
		0x4002 5000 - 0x4002 5FFF	保留
		0x4002 4000 - 0x4002 4FFF	保留
		0x4002 3C00 - 0x4002 3FFF	保留
		0x4002 3800 - 0x4002 3BFF	保留
		0x4002 3400 - 0x4002 37FF	保留
		0x4002 3000 - 0x4002 33FF	保留
		0x4002 2C00 - 0x4002 2FFF	保留
		0x4002 2800 - 0x4002 2BFF	EFUSE
		0x4002 2400 - 0x4002 27FF	保留
		0x4002 2000 - 0x4002 23FF	保留
		0x4002 1C00 - 0x4002 1FFF	保留
		0x4002 1800 - 0x4002 1BFF	保留
		0x4002 1400 - 0x4002 17FF	保留
		0x4002 1000 - 0x4002 13FF	保留
		0x4002 0C00 - 0x4002 0FFF	保留
		0x4002 0800 - 0x4002 0BFF	DMAMUX
		0x4002 0400 - 0x4002 07FF	DMA1
		0x4002 0000 - 0x4002 03FF	DMA0
	APB2	0x4001 F400 - 0x4001 FFFF	保留
		0x4001 F000 - 0x4001 F3FF	TIMER44
		0x4001 DC00 - 0x4001 DFFF	TIMER43
		0x4001 D800 - 0x4001 DBFF	TIMER42
		0x4001 D400 - 0x4001 D7FF	TIMER41
		0x4001 D000 - 0x4001 D3FF	TIMER40
		0x4001 C000 - 0x4001 CFFF	CAN2(4KB)
		0x4001 B000 - 0x4001 BFFF	CAN1(4KB)
		0x4001 A000 - 0x4001 AFFF	CAN0(4KB)
		0x4001 8C00 - 0x4001 9FFF	保留
		0x4001 8800 - 0x4001 8BFF	EDOUT
		0x4001 8400 - 0x4001 87FF	TRIGSEL
		0x4001 8000 - 0x4001 83FF	保留(APB2)
		0x4001 7C00 - 0x4001 7FFF	保留
		0x4001 7800 - 0x4001 7BFF	保留
		0x4001 7400 - 0x4001 77FF	保留
		0x4001 7000 - 0x4001 73FF	HPDF
		0x4001 6C00 - 0x4001 6FFF	保留
		0x4001 6800 - 0x4001 6BFF	保留
		0x4001 6400 - 0x4001 67FF	保留
		0x4001 6000 - 0x4001 63FF	SAI2

预先定义的地址空间	总线	地址范围	外设
		0x4001 5C00 - 0x4001 5FFF	SAI1
		0x4001 5800 - 0x4001 5BFF	SAI0
		0x4001 5400 - 0x4001 57FF	保留
		0x4001 5000 - 0x4001 53FF	SPI4
		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	SPI5/I2S5
		0x4001 3400 - 0x4001 37FF	SPI3
		0x4001 3000 - 0x4001 33FF	SPI0/I2S0
		0x4001 2C00 - 0x4001 2FFF	ADC2
		0x4001 2800 - 0x4001 2BFF	ADC1
		0x4001 2400 - 0x4001 27FF	ADC0
		0x4001 2000 - 0x4001 23FF	保留
		0x4001 1C00 - 0x4001 1FFF	保留
		0x4001 1800 - 0x4001 1BFF	保留
		0x4001 1400 - 0x4001 17FF	USART5
		0x4001 1000 - 0x4001 13FF	USART0
		0x4001 0C00 - 0x4001 0FFF	保留
		0x4001 0800 - 0x4001 0BFF	保留
		0x4001 0400 - 0x4001 07FF	TIMER7
		0x4001 0000 - 0x4001 03FF	TIMER0
	APB1	0x4000 F800 - 0x4000 FFFF	保留
		0x4000 F400 - 0x4000 F7FF	TIMER51
		0x4000 F000 - 0x4000 F3FF	TIMER50
		0x4000 EC00 - 0x4000 EFFF	TIMER31
		0x4000 E800 - 0x4000 EBFF	TIMER30
		0x4000 E400 - 0x4000 E7FF	TIMER23
		0x4000 E000 - 0x4000 E3FF	TIMER22
		0x4000 DC00 - 0x4000 DFFF	保留
		0x4000 D800 - 0x4000 DBFF	保留
		0x4000 D400 - 0x4000 D7FF	保留
		0x4000 D000 - 0x4000 D3FF	保留
		0x4000 CC00 - 0x4000 CFFF	保留
		0x4000 C800 - 0x4000 CBFF	保留
		0x4000 C400 - 0x4000 C7FF	保留
		0x4000 C000 - 0x4000 C3FF	I2C2
		0x4000 9800 - 0x4000 BFFF	保留
		0x4000 9400 - 0x4000 97FF	MDIO

预先定义的地址空间	总线	地址范围	外设
		0x4000 8800 - 0x4000 93FF	保留
		0x4000 8400 - 0x4000 87FF	CTC
		0x4000 8000 - 0x4000 83FF	保留
		0x4000 7C00 - 0x4000 7FFF	UART7
		0x4000 7800 - 0x4000 7BFF	UART6
		0x4000 7400 - 0x4000 77FF	DAC
		0x4000 7000 - 0x4000 73FF	保留
		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	RSPDIF
		0x4000 3C00 - 0x4000 3FFF	SPI2/I2S2
		0x4000 3800 - 0x4000 3BFF	SPI1/I2S1
		0x4000 3400 - 0x4000 37FF	保留
		0x4000 3000 - 0x4000 33FF	保留
		0x4000 2C00 - 0x4000 2FFF	保留
		0x4000 2800 - 0x4000 2BFF	保留
		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
		0x4000 0400 - 0x4000 07FF	TIMER2
		0x4000 0000 - 0x4000 03FF	TIMER1
SRAM		0x3880 1000 - 0x3FFF FFFF	保留
		0x3880 0000 - 0x3880 0FFF	Backup SRAM
		0x3000 8000 - 0x387F FFFF	保留
		0x3000 4000 - 0x3000 7FFF	SRAM1(16KB)
		0x3000 0000 - 0x3000 3FFF	SRAM0(16KB)
		0x2410 0000 - 0x2FFF FFFF	保留

预先定义的地址空间	总线	地址范围	外设
		0x2408 0000 - 0x240F FFFF	共享 RAM(512KB) (ITCM/DTCM/AXI SRAM)
		0x2400 0000 - 0x2407 FFFF	AXI SRAM(512KB)
		0x2008 0000 - 0x23FF FFFF	保留
		0x2007 0000 - 0x2007 FFFF	DTCM RAM（来自共享 RAM）
		0x2006 0000 - 0x2006 FFFF	
		0x2003 0000 - 0x2005 FFFF	
		0x2002 0000 - 0x2002 FFFF	
		0x2001 C000 - 0x2001 FFFF	
		0x2001 8000 - 0x2001 BFFF	
		0x2001 0000 - 0x2001 7FFF	
		0x2000 D000 - 0x2000 FFFF	
		0x2000 C000 - 0x2000 CFFF	
		0x2000 8000 - 0x2000 BFFF	
		0x2000 5000 - 0x2000 7FFF	
		0x2000 2000 - 0x2000 4FFF	
		0x2000 1000 - 0x2000 1FFF	
		0x2000 0000 - 0x2000 0FFF	
代码		0x1FFF FC10 - 0x1FFF FFFF	保留
		0x1FFF FC00 - 0x1FFF FC0F	保留
		0x1FFF F818 - 0x1FFF BFFF	保留
		0x1FFF F800 - 0x1FFF F817	保留
		0x1FFF F000 - 0x1FFF F7FF	保留
		0x1FFF EC00 - 0x1FFF EFFF	保留
		0x1FFF C010 - 0x1FFF EBFF	保留
		0x1FFF C000 - 0x1FFF C00F	保留
		0x1FFF B000 - 0x1FFF BFFF	保留
		0x1FFF 8000 - 0x1FFF AFFF	保留
		0x1FFF 7A10 - 0x1FFF 7FFF	保留
		0x1FFF 7800 - 0x1FFF 7A0F	保留
		0x1FFF 7400 - 0x1FFF 77FF	保留
		0x1FFF 7000 - 0x1FFF 73FF	保留
		0x1FFF 0000 - 0x1FFF 6FFF	保留
		0x1FFE C010 - 0x1FFE FFFF	保留
		0x1FFE C000 - 0x1FFE C00F	保留
		0x1FF6 0000 - 0x1FFE BFFF	保留
		0x1FF4 0000 - 0x1FF5 FFFF	保留
		0x1FF1 0000 - 0x1FF3 FFFF	保留
		0x1FF0 0000 - 0x1FF0 FFFF	系统存储区
		0x1002 0000 - 0x1FEF FFFF	保留

预先定义的地址空间	总线	地址范围	外设
		0x1001 0000 - 0x1001 FFFF	保留
		0x1000 0000 - 0x1000 FFFF	保留
		0x0A00 D000 - 0x0FFF FFFF	保留
		0x0A00 C000 - 0x0A00 CFFF	保留
		0x0A00 8000 - 0x0A00 BFFF	保留
		0x0A00 0000 - 0x0A00 7FFF	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
		0x0840 0000 - 0x087F FFFF	保留
		0x083C 0000 - 0x083F FFFF	保留
		0x0830 0000 - 0x083B FFFF	Flash 存储器
		0x0810 0000 - 0x082F FFFF	
		0x0808 0000 - 0x080F FFFF	
		0x0806 0000 - 0x0807 FFFF	
		0x0802 0000 - 0x0805 FFFF	
		0x0801 0000 - 0x0801 FFFF	
		0x0800 0000 - 0x0800 FFFF	
		0x0030 0000 - 0x07FF FFFF	保留
		0x0010 0000 - 0x002F FFFF	保留
		0x0008 0000 - 0x000F FFFF	保留
		0x0002 6000 - 0x0007 FFFF	ITCM RAM（来自共享 RAM）
		0x0002 0000 - 0x0002 5FFF	
		0x0001 0000 - 0x0001 FFFF	
		0x0000 0000 - 0x0000 FFFF	

表 1-3. GD32H73x_75x 系列器件的存储器映射表（适用于 GD32H75E）

预先定义的地址空间	总线	地址范围	外设
外部 RAM		0xD000 0000 - 0xDFFF FFFF	EXMC - SDRAM 设备 1
		0xC000 0000 - 0xCFFF FFFF	EXMC – SDRAM 设备 0 (EXMC Bank 0 Region 0-3)
		0xA000 1000 - 0xBFFF FFFF	保留
		0xA000 0000 - 0xA000 0FFF	保留
		0x9000 0000 - 0x9FFF FFFF	OSPI0
		0x8000 0000 - 0x8FFF FFFF	EXMC - NAND
		0x7000 0000 - 0x7FFF FFFF	OSPI1
		0x6000 0000 - 0x6FFF FFFF	EXMC - NOR/PSRAM/SRAM
外设	AHB4	0x5802 7000 - 0x5FFF FFFF	保留

预先定义的地址空间	总线	地址范围	外设
		0x5802 6400 - 0x5802 67FF	保留
		0x5802 6000 - 0x5802 63FF	保留
		0x5802 5000 - 0x5802 5FFF	保留
		0x5802 4C00 - 0x5802 4FFF	CRC
		0x5802 4800 - 0x5802 4BFF	保留
		0x5802 4400 - 0x5802 47FF	RCU
		0x5802 2C00 - 0x5802 43FF	保留
		0x5802 2800 - 0x5802 2BFF	GPIOK
		0x5802 2400 - 0x5802 27FF	GPIOJ
		0x5802 2000 - 0x5802 23FF	保留
		0x5802 1C00 - 0x5802 1FFF	GPIOH
		0x5802 1800 - 0x5802 1BFF	GPIOG
		0x5802 1400 - 0x5802 17FF	GPIOF
		0x5802 1000 - 0x5802 13FF	GPIOE
		0x5802 0C00 - 0x5802 0FFF	GPIOD
		0x5802 0800 - 0x5802 0BFF	GPIOC
		0x5802 0400 - 0x5802 07FF	GPIOB
		0x5802 0000 - 0x5802 03FF	GPIOA
		0x5801 0000 - 0x5801 FFFF	保留
	APB4	0x5800 7400 - 0x5800 FFFF	保留
		0x5800 7000 - 0x5800 73FF	保留
		0x5800 6C00 - 0x5800 6FFF	保留
		0x5800 6800 - 0x5800 6BFF	LPDTS
		0x5800 5800 - 0x5800 67FF	PMU
		0x5800 5400 - 0x5800 57FF	保留
		0x5800 4C00 - 0x5800 53FF	保留
		0x5800 4800 - 0x5800 4BFF	FWDGT
		0x5800 4000 - 0x5800 43FF	RTC
		0x5800 3C00 - 0x5800 3FFF	VREF
		0x5800 3800 - 0x5800 3BFF	CMP0 - CMP1
		0x5800 3400 - 0x5800 37FF	保留
		0x5800 3000 - 0x5800 33FF	保留
		0x5800 2C00 - 0x5800 2FFF	保留
		0x5800 2800 - 0x5800 2BFF	保留
		0x5800 2400 - 0x5800 27FF	保留
		0x5800 2000 - 0x5800 23FF	保留
		0x5800 1C00 - 0x5800 1FFF	保留
		0x5800 1400 - 0x5800 17FF	保留
		0x5800 0800 - 0x5800 13FF	保留
		0x5800 0400 - 0x5800 07FF	SYSCFG
		0x5800 0000 - 0x5800 03FF	EXTI

预先定义的地址空间	总线	地址范围	外设
	AHB3	0x5200 C000 - 0x57FF FFFF	保留
		0x5200 BC00 - 0x5200 BFFF	保留
		0x5200 B800 - 0x5200 BBFF	保留
		0x5200 B400 - 0x5200 B7FF	OSPIM
		0x5200 B000 - 0x5200 B3FF	保留
		0x5200 A000 - 0x5200 AFFF	OSPI1(内部)
		0x5200 9400 - 0x5200 9FFF	保留
		0x5200 9000 - 0x5200 93FF	RAMECCMU Region 0
		0x5200 8000 - 0x5200 8FFF	保留
		0x5200 7000 - 0x5200 7FFF	保留
		0x5200 6000 - 0x5200 6FFF	保留
		0x5200 5000 - 0x5200 5FFF	OSPI0
		0x5200 4000 - 0x5200 4FFF	EXMC
		0x5200 3400 - 0x5200 3FFF	保留
		0x5200 3000 - 0x5200 33FF	保留
		0x5200 2000 - 0x5200 2FFF	FMC
		0x5200 1000 - 0x5200 1FFF	保留
		0x5200 0000 - 0x5200 0FFF	MDMA
		0x5110 0000 - 0x511F FFFF	保留
		0x5100 0000 - 0x510F FFFF	AXI 互联矩阵
	APB3	0x5006 1000 - 0x50FF FFFF	保留
		0x5006 0C00 - 0x5006 0FFF	保留
		0x5006 0800 - 0x5006 0BFF	保留
		0x5006 0400 - 0x5006 07FF	保留
		0x5006 0000 - 0x5006 03FF	保留
		0x5005 0400 - 0x5005 FFFF	保留
		0x5005 0000 - 0x5005 03FF	保留
		0x5004 0000 - 0x5004 FFFF	保留
		0x5000 0000 - 0x5003 FFFF	保留
		0x5000 3000 - 0x5000 3FFF	WWDGT
		0x5000 2000 - 0x5000 2FFF	保留
		0x5000 1000 - 0x5000 1FFF	保留
		0x5000 0000 - 0x5000 0FFF	保留
	AHB2	0x4802 5000 - 0x4FFF FFFF	保留(AHB2)
		0x4802 4800 - 0x4802 4FFF	FAC
		0x4802 4400 - 0x4802 47FF	TMU
		0x4802 4000 - 0x4802 43FF	保留
		0x4802 3000 - 0x4802 3FFF	RAMECCMU Region 1
		0x4802 2C00 - 0x4802 2FFF	保留(AHB2)
		0x4802 2800 - 0x4802 2BFF	保留
		0x4802 2400 - 0x4802 27FF	保留

预先定义的地址空间	总线	地址范围	外设
		0x4802 1C00 - 0x4802 23FF	保留(AHB2)
		0x4802 1800 - 0x4802 1BFF	TRNG
		0x4802 1400 - 0x4802 17FF	保留
		0x4802 1000 - 0x4802 13FF	保留
		0x4802 0400 - 0x4802 0FFF	保留(AHB2)
		0x4802 0000 - 0x4802 03FF	保留
		0x4800 1800 - 0x4801 FFFF	保留(AHB2)
		0x4800 1400 - 0x4800 17FF	保留
		0x4800 1000 - 0x4800 13FF	保留
		0x4800 0C00 - 0x4800 0FFF	保留
		0x4800 0800 - 0x4800 0BFF	保留
		0x4800 0400 - 0x4800 07FF	保留
		0x4800 0000 - 0x4800 03FF	保留
	AHB1	0x400C 0000 - 0x47FF FFFF	保留(AHB1)
		0x4008 0000 - 0x400B FFFF	USBHS1
		0x4004 0000 - 0x4007 FFFF	USBHS0
		0x4003 8C00 - 0x4003 FFFF	保留
		0x4003 8400 - 0x4003 8BFF	保留
		0x4003 8000 - 0x4003 83FF	保留
		0x4003 3000 - 0x4003 7FFF	保留
		0x4003 0000 - 0x4003 2FFF	保留
		0x4002 C000 - 0x4002 FFFF	保留
		0x4002 BC00 - 0x4002 BFFF	
		0x4002 B000 - 0x4002 BBFF	保留
		0x4002 A000 - 0x4002 AFFF	
		0x4002 8000 - 0x4002 9FFF	保留
		0x4002 6800 - 0x4002 7FFF	保留
		0x4002 6400 - 0x4002 67FF	保留
		0x4002 6000 - 0x4002 63FF	保留
		0x4002 5000 - 0x4002 5FFF	保留
		0x4002 4000 - 0x4002 4FFF	保留
		0x4002 3C00 - 0x4002 3FFF	保留
		0x4002 3800 - 0x4002 3BFF	保留
		0x4002 3400 - 0x4002 37FF	保留
		0x4002 3000 - 0x4002 33FF	保留
		0x4002 2C00 - 0x4002 2FFF	保留
		0x4002 2800 - 0x4002 2BFF	EFUSE
		0x4002 2400 - 0x4002 27FF	保留
		0x4002 2000 - 0x4002 23FF	保留
		0x4002 1C00 - 0x4002 1FFF	保留
		0x4002 1800 - 0x4002 1BFF	保留

预先定义的地址空间	总线	地址范围	外设
		0x4002 1400 - 0x4002 17FF	保留
		0x4002 1000 - 0x4002 13FF	保留
		0x4002 0C00 - 0x4002 0FFF	保留
		0x4002 0800 - 0x4002 0BFF	DMAMUX
		0x4002 0400 - 0x4002 07FF	DMA1
		0x4002 0000 - 0x4002 03FF	DMA0
	APB2	0x4001 F400 - 0x4001 FFFF	保留
		0x4001 F000 - 0x4001 F3FF	TIMER44
		0x4001 DC00 - 0x4001 DFFF	TIMER43
		0x4001 D800 - 0x4001 DBFF	TIMER42
		0x4001 D400 - 0x4001 D7FF	TIMER41
		0x4001 D000 - 0x4001 D3FF	TIMER40
		0x4001 C000 - 0x4001 CFFF	CAN2(4KB)
		0x4001 B000 - 0x4001 BFFF	CAN1(4KB)
		0x4001 A000 - 0x4001 AFFF	CAN0(4KB)
		0x4001 8C00 - 0x4001 9FFF	保留
		0x4001 8800 - 0x4001 8BFF	EDOUT
		0x4001 8400 - 0x4001 87FF	TRIGSEL
		0x4001 8000 - 0x4001 83FF	保留(APB2)
		0x4001 7C00 - 0x4001 7FFF	保留
		0x4001 7800 - 0x4001 7BFF	保留
		0x4001 7400 - 0x4001 77FF	保留
		0x4001 7000 - 0x4001 73FF	HPDF
		0x4001 6C00 - 0x4001 6FFF	保留
		0x4001 6800 - 0x4001 6BFF	保留
		0x4001 6400 - 0x4001 67FF	保留
		0x4001 6000 - 0x4001 63FF	保留
		0x4001 5C00 - 0x4001 5FFF	保留
		0x4001 5800 - 0x4001 5BFF	保留
		0x4001 5400 - 0x4001 57FF	保留
		0x4001 5000 - 0x4001 53FF	SPI4
		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	SPI5/I2S5
		0x4001 3400 - 0x4001 37FF	SPI3
		0x4001 3000 - 0x4001 33FF	SPI0/I2S0
		0x4001 2C00 - 0x4001 2FFF	ADC2
		0x4001 2800 - 0x4001 2BFF	ADC1

预先定义的地址空间	总线	地址范围	外设
		0x4001 2400 - 0x4001 27FF	ADC0
		0x4001 2000 - 0x4001 23FF	保留
		0x4001 1C00 - 0x4001 1FFF	保留
		0x4001 1800 - 0x4001 1BFF	保留
		0x4001 1400 - 0x4001 17FF	USART5
		0x4001 1000 - 0x4001 13FF	USART0
		0x4001 0C00 - 0x4001 0FFF	保留
		0x4001 0800 - 0x4001 0BFF	保留
		0x4001 0400 - 0x4001 07FF	TIMER7
		0x4001 0000 - 0x4001 03FF	TIMER0
	APB1	0x4000 F800 - 0x4000 FFFF	保留
		0x4000 F400 - 0x4000 F7FF	TIMER51
		0x4000 F000 - 0x4000 F3FF	TIMER50
		0x4000 EC00 - 0x4000 EFFF	保留
		0x4000 E800 - 0x4000 EBFF	保留
		0x4000 E400 - 0x4000 E7FF	TIMER23
		0x4000 E000 - 0x4000 E3FF	TIMER22
		0x4000 DC00 - 0x4000 DFFF	保留
		0x4000 D800 - 0x4000 DBFF	保留
		0x4000 D400 - 0x4000 D7FF	保留
		0x4000 D000 - 0x4000 D3FF	保留
		0x4000 CC00 - 0x4000 CFFF	保留
		0x4000 C800 - 0x4000 CBFF	保留
		0x4000 C400 - 0x4000 C7FF	保留
		0x4000 C000 - 0x4000 C3FF	I2C2
		0x4000 9800 - 0x4000 BFFF	保留
		0x4000 9400 - 0x4000 97FF	保留
		0x4000 8800 - 0x4000 93FF	保留
		0x4000 8400 - 0x4000 87FF	CTC
		0x4000 8000 - 0x4000 83FF	保留
		0x4000 7C00 - 0x4000 7FFF	UART7
		0x4000 7800 - 0x4000 7BFF	UART6
		0x4000 7400 - 0x4000 77FF	DAC
		0x4000 7000 - 0x4000 73FF	保留
		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

预先定义的地址空间	总线	地址范围	外设
		0x4000 5000 - 0x4000 53FF	UART4
		0x4000 4C00 - 0x4000 4FFF	UART3
		0x4000 4800 - 0x4000 4BFF	USART2
		0x4000 4400 - 0x4000 47FF	USART1
		0x4000 4000 - 0x4000 43FF	保留
		0x4000 3C00 - 0x4000 3FFF	SPI2/I2S2
		0x4000 3800 - 0x4000 3BFF	SPI1/I2S1
		0x4000 3400 - 0x4000 37FF	保留
		0x4000 3000 - 0x4000 33FF	保留
		0x4000 2C00 - 0x4000 2FFF	保留
		0x4000 2800 - 0x4000 2BFF	保留
		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
		0x4000 0400 - 0x4000 07FF	TIMER2
		0x4000 0000 - 0x4000 03FF	TIMER1
SRAM		0x3880 1000 - 0x3FFF FFFF	保留
		0x3880 0000 - 0x3880 0FFF	Backup SRAM
		0x3000 8000 - 0x387F FFFF	保留
		0x3000 4000 - 0x3000 7FFF	SRAM1(16KB)
		0x3000 0000 - 0x3000 3FFF	SRAM0(16KB)
		0x2410 0000 - 0x2FFF FFFF	保留
		0x2408 0000 - 0x240F FFFF	共享 RAM(512KB) (ITCM/DTCM/AXI SRAM)
		0x2400 0000 - 0x2407 FFFF	AXI SRAM(512KB)
		0x2008 0000 - 0x23FF FFFF	保留
		0x2007 0000 - 0x2007 FFFF	DTCM RAM（来自共享 RAM）
		0x2006 0000 - 0x2006 FFFF	
		0x2003 0000 - 0x2005 FFFF	
		0x2002 0000 - 0x2002 FFFF	
		0x2001 C000 - 0x2001 FFFF	
		0x2001 8000 - 0x2001 BFFF	
		0x2001 0000 - 0x2001 7FFF	
		0x2000 D000 - 0x2000 FFFF	
		0x2000 C000 - 0x2000 CFFF	

预先定义的地址空间	总线	地址范围	外设
		0x2000 8000 - 0x2000 BFFF	
		0x2000 5000 - 0x2000 7FFF	
		0x2000 2000 - 0x2000 4FFF	
		0x2000 1000 - 0x2000 1FFF	
		0x2000 0000 - 0x2000 0FFF	
代码		0x1FFF FC10 - 0x1FFF FFFF	保留
		0x1FFF FC00 - 0x1FFF FC0F	保留
		0x1FFF F818 - 0x1FFF BFFF	保留
		0x1FFF F800 - 0x1FFF F817	保留
		0x1FFF F000 - 0x1FFF F7FF	保留
		0x1FFF EC00 - 0x1FFF EFFF	保留
		0x1FFF C010 - 0x1FFF EBFF	保留
		0x1FFF C000 - 0x1FFF C00F	保留
		0x1FFF B000 - 0x1FFF BFFF	保留
		0x1FFF 8000 - 0x1FFF AFFF	保留
		0x1FFF 7A10 - 0x1FFF 7FFF	保留
		0x1FFF 7800 - 0x1FFF 7A0F	保留
		0x1FFF 7400 - 0x1FFF 77FF	保留
		0x1FFF 7000 - 0x1FFF 73FF	保留
		0x1FFF 0000 - 0x1FFF 6FFF	保留
		0x1FFE C010 - 0x1FFE FFFF	保留
		0x1FFE C000 - 0x1FFE C00F	保留
		0x1FF6 0000 - 0x1FFE BFFF	保留
		0x1FF4 0000 - 0x1FF5 FFFF	保留
		0x1FF1 0000 - 0x1FF3 FFFF	保留
		0x1FF0 0000 - 0x1FF0 FFFF	系统存储区
		0x1002 0000 - 0x1FEF FFFF	保留
		0x1001 0000 - 0x1001 FFFF	保留
		0x1000 0000 - 0x1000 FFFF	保留
		0x0A00 D000 - 0x0FFF FFFF	保留
		0x0A00 C000 - 0x0A00 CFFF	保留
		0x0A00 8000 - 0x0A00 BFFF	保留
		0x0A00 0000 - 0x0A00 7FFF	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
		0x0840 0000 - 0x087F FFFF	保留
		0x083C 0000 - 0x083F FFFF	保留
		0x0830 0000 - 0x083B FFFF	Flash 存储器
		0x0810 0000 - 0x082F FFFF	

预先定义的地址空间	总线	地址范围	外设
		0x0808 0000 - 0x080F FFFF	
		0x0806 0000 - 0x0807 FFFF	
		0x0802 0000 - 0x0805 FFFF	
		0x0801 0000 - 0x0801 FFFF	
		0x0800 0000 - 0x0800 FFFF	
		0x0030 0000 - 0x07FF FFFF	保留
		0x0010 0000 - 0x002F FFFF	保留
		0x0008 0000 - 0x000F FFFF	保留
		0x0002 6000 - 0x0007 FFFF	ITCM RAM（来自共享 RAM）
		0x0002 0000 - 0x0002 5FFF	
		0x0001 0000 - 0x0001 FFFF	
		0x0000 0000 - 0x0000 FFFF	

1.3.1. 片上 SRAM 存储器

GD32H73x_75x系列的设备包含高达512KB的片上SRAM（AXI SRAM）、4KB的备份SRAM和高达512KB的ITCM/DTCM/AXI SRAM共享RAM。所有AHB SRAM都支持字节、半字（16位）和字（32位）访问。片上SRAM（AXI SRAM）支持字节、半字（16位）、字（32位）和双字（64位）访问。几乎所有AHB主机都可以访问SRAM0和SRAM1。备份SRAM（BKPRAM）在备份域中实现，即使V_{DD}电源关闭，它也可以保留其内容。

ITCM/DTCM SRAM访问频率

如果SYSCFG_SRAMCFG1寄存器的TCM_WAITSTATE位为0，可以以最高f_{TWW}频率访问对应的TCM SRAM。TCM_WAITSTATE位在系统复位后为0。f_{TWW}数值详见芯片数据手册。

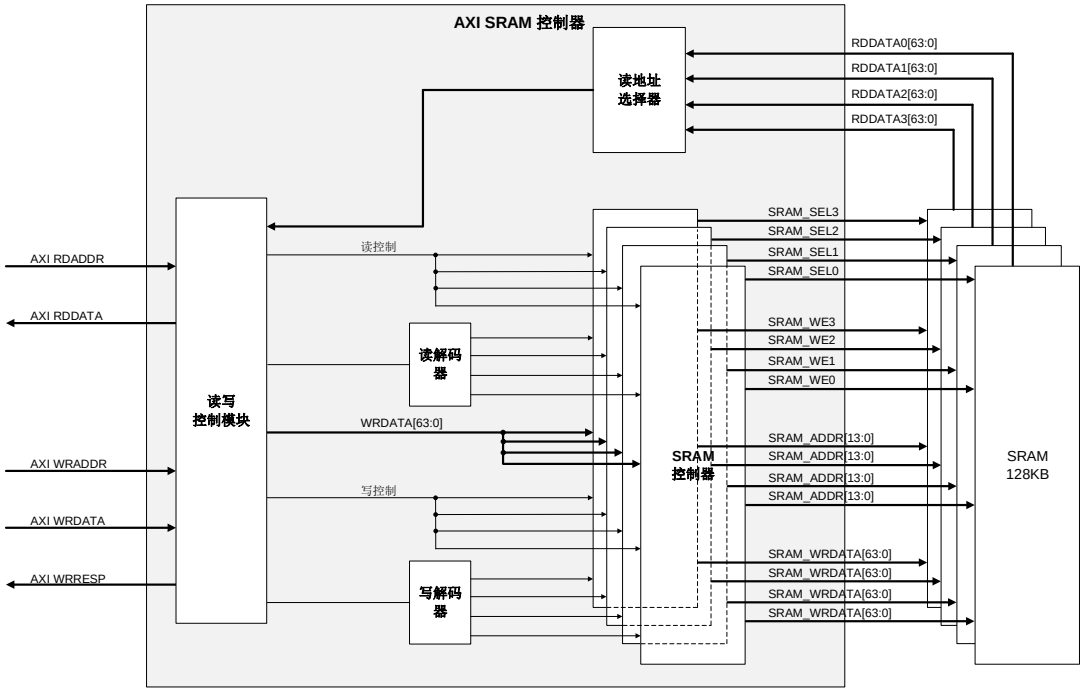
ITCM/DTCM SRAM ECC功能

如果将TCM ECC功能(ITCMECCEN / DTCM0ECCEN / DTCM1ECCEN)打开，可以增加健壮性。

AXI SRAM

片上SRAM（AXI SRAM）控制器由读写控制模块、MUX、解码器和SRAM控制器组成。每个SRAM控制器都有一个带有轮询机制的仲裁器。框图如*图1-7. AXI SRAM框图*所示。

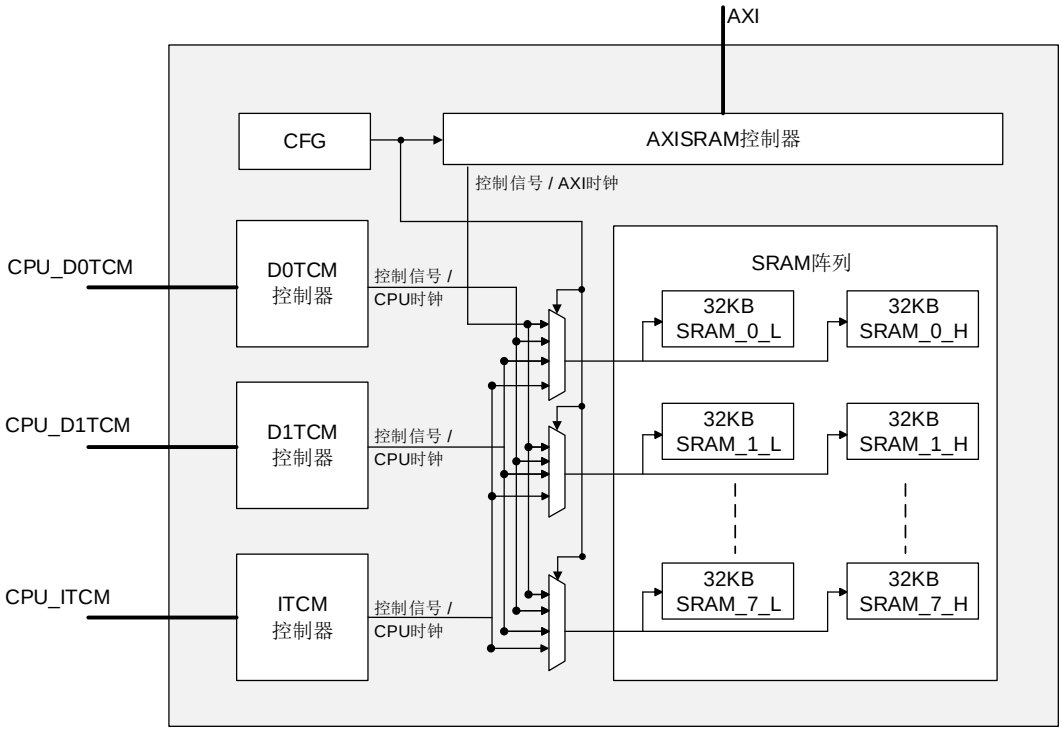
图 1-7. AXI SRAM 框图



ITCM/DTCM/AXI SRAM共享的RAM

512KB的RAM可由ITCM或DTCM或AXI SRAM使用，可通过选项字节状态寄存器1寄存器中的ITCM_SZ_SHRRAM和DTCM_SZ_SHRRAM位进行配置，如*表1-4. ITCM/DTCM/AXI SRAM的配置*所述。框图如*图1-8. ITCM/DTCM/AXI SRAM共享的RAM框图*所示。

图 1-8. ITCM/DTCM/AXI SRAM 共享的 RAM 框图



ITCM，DTCM和共享SRAM（AXI SRAM）总容量为512KB，其中ITCM的配置优先，DTCM配

置次之，剩下是共享SRAM部分。

表 1-4. ITCM/DTCM/AXI SRAM 的配置

ITCM_SZ_SHRRAM[3:0]	DTCM_SZ_SHRRAM[3:0]	ITCM容量(KB)	DTCM容量(KB)	AXI SRAM容量(KB)
0000	0000	0	0	512
0000	0111	0	64	448
0000	1000	0	128	384
0000	1001	0	256	256
0000	1010	0	512	0
0111	0000	64	0	448
0111	0111	64	64	384
0111	1000	64	128	320
0111	1001	64	256	192
1000	0000	128	0	384
1000	0111	128	64	320
1000	1000	128	128	256
1000	1001	128	256	128
1001	0000	256	0	256
1001	0111	256	64	192
1001	1000	256	128	128
1001	1001	256	256	0
1010	0000	512	0	0

1.3.2. 片上 FLASH 存储器概述

GD32H73x_75x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3840KB主FLASH存储器；
- 高达64KB引导装载程序(boot loader)信息块存储器；
- 器件配置的选项字节。

更多详细说明请参考*闪存控制器（FMC）*章节。

1.4. 引导配置

GD32H73x_75x 设备提供不同的引导源，可通过 Arm® Cortex®-M7 核心寄存器（FMC_BTADDR_MDF）的引导地址中的引导引脚和引导地址0/1[15:0]进行选择。详情见*表 1-5. 引导模式选择*、*表1-6. 引导模式详细描述（适用于GD32H737_757_759）*和*表1-7. 引导模式详细描述（适用于GD32H75E）*。BOOT引脚的电平状态会在复位后的第四个CK_SYS(系统时钟)的上升沿进行锁存。用户可自行选择所需要的引导源，通过设置上电复位和系统复位后的BOOT的引脚电平。一旦这个引脚电平被采样，它们可以被释放并用于其他用途。

BOOT_ADDR0[15:0]和BOOT_ADDR1[15:0]地址允许将引导内存地址配置为0x0000 0000到0x9000 0000之间的任何地址。引导模式可由SYSCFG_USERCFG寄存器的

BOOT_MODE[2:0]位域中获取。

表 1-5. 引导模式选择

引导源地址	启动模式选择引脚
	BOOT
引导地址高位：由BOOT_ADDR0[15:0]定义 引导地址低位：0x0000	0
引导地址高位：由BOOT_ADDR1[15:0]定义 引导地址低位：0x0000	1

表 1-6. 引导模式详细描述（适用于 GD32H737_757_759）

SCR	SPC[7:0]	BOOT_ADDRESS（在BOOT_ADDRx(x = 0,1)配置）	BOOT_MODE	启动地址
1	x	XXXX	SECURITY BOOT	ROM
0	安全保护等级高	0x0800_0000~max user flash	USER BOOT	BOOT_ADDRESS
		其他地址	USER BOOT	0x0800_0000
	安全保护等级低	0x2408_0000~ max RAM shared(ITCM/DTCM/AXI)	SRAM BOOT (RAM shared)	BOOT_ADDRESS
		0x2400_0000~ max AXI SRAM	SRAM BOOT (AXI SRAM)	BOOT_ADDRESS
		0x2000_0000	SRAM BOOT(DTCM)	0x2000_0000
		0x0800_0000~max user flash	USER BOOT	BOOT_ADDRESS
		0x0000_0000	SRAM BOOT(ITCM)	0x0000_0000
		0x1FF0_0000	SYSTEM BOOT	BootLoader
		其他地址	USER BOOT	0x0800_0000(BOOT Pin = 0)
			SYSTEM BOOT	BootLoader (BOOT Pin = 1)
	无保护状态	0x9000_0000	USER BOOT	OSPI0
		0x7000_0000	USER BOOT	OSPI1
		0x2408_0000~ max RAM shared(ITCM/DTCM/AXI)	SRAM BOOT(RAM shared)	BOOT_ADDRESS
		0x2400_0000~ max AXI SRAM	SRAM BOOT(AXI SRAM)	BOOT_ADDRESS
		0x2000_0000	SRAM BOOT(DTCM)	0x2000_0000
		0x0800_0000~max user flash	USER BOOT	BOOT_ADDRESS
		0x0000_0000	SRAM BOOT(ITCM)	0x0000_0000
		0x1FF0_0000	SYSTEM BOOT	BootLoader
		其他地址	USER BOOT	0x0800_0000 (BOOT Pin = 0)
			SYSTEM BOOT	BootLoader

SCR	SPC[7:0]	BOOT_ADDRESS（在BOOT_ADDRx(x = 0,1)配置）	BOOT_MODE	启动地址
				(BOOT Pin = 1)

表 1-7. 引导模式详细描述（适用于 GD32H75E）

SCR	SPC[7:0]	BOOT_ADDRESS（在BOOT_ADDRx(x = 0,1)配置）	BOOT_MODE	启动地址
1	x	XXXX	SECURITY BOOT	ROM
0	安全保护等级高	0x9000_0000	USER BOOT	OSPI0
		0x7000_0000	USER BOOT	OSPI1
		0x0800_0000~max user flash	USER BOOT	BOOT_ADDRESS
		其他地址	USER BOOT	0x0800_0000
	无保护状态/安全保护等级低	0x9000_0000	USER BOOT	OSPI0
		0x7000_0000	USER BOOT	OSPI1
		0x2408_000~max RAM shared (ITCM/DTCM/AXI)	SRAM BOOT (RAM shared)	BOOT_ADDRESS
		0x2400_0000~max AXI SRAM	SRAM BOOT (AXI SRAM)	BOOT_ADDRESS
		0x2000_0000	SRAM BOOT(DTCM)	0x2000_0000
		0x0800_0000~max user flash	USER BOOT	BOOT_ADDRESS
		0x0000_0000	SRAM BOOT(ITCM)	0x0000_0000
		0x1FF0_0000	SYSTEM BOOT	BootLoader
		其他地址	USER BOOT	0x0800_0000 (BOOT Pin = 0)
			SYSTEM BOOT	BootLoader (BOOT Pin = 1)

嵌入式的Bootloader存放在系统存储空间，用于对FLASH存储器进行重新编程。在GD32H73x_75x设备中，Bootloader可以通过USART端口、USBHS端口、SDIO端口和外界交互。具体情况见产品数据手册。

1.5. 系统配置控制器（SYSCFG）

系统配置控制器的主要功能如下：

- 模拟开关配置管理
- 配置I2C Fm+
- 以太网PHY接口选择
- 管理外部中断线与GPIO的连接
- 管理I/O补偿单元
- 管理BOR复位电平

- 管理定时器中止输入锁定

1.6. 定时器中止输入锁定

当内部SRAM/FMC ECC错误、LVD或CPU内核锁定发生时，此功能允许禁用TIMER输出。有关详细信息，请参阅*锁定控制寄存器（SYSCFG_LKCTL）*。

1.7. AXI 互连矩阵（AXIIM）

AXI 互连矩阵是基于 Arm® CoreLink™ NIC-400 网络互连。它有 6 个发起者端口或 ASIB（AMBA 从接口块）和 8 个目标端口或 AMIB（AMBA 主接口块）。

1.7.1. 主要特性

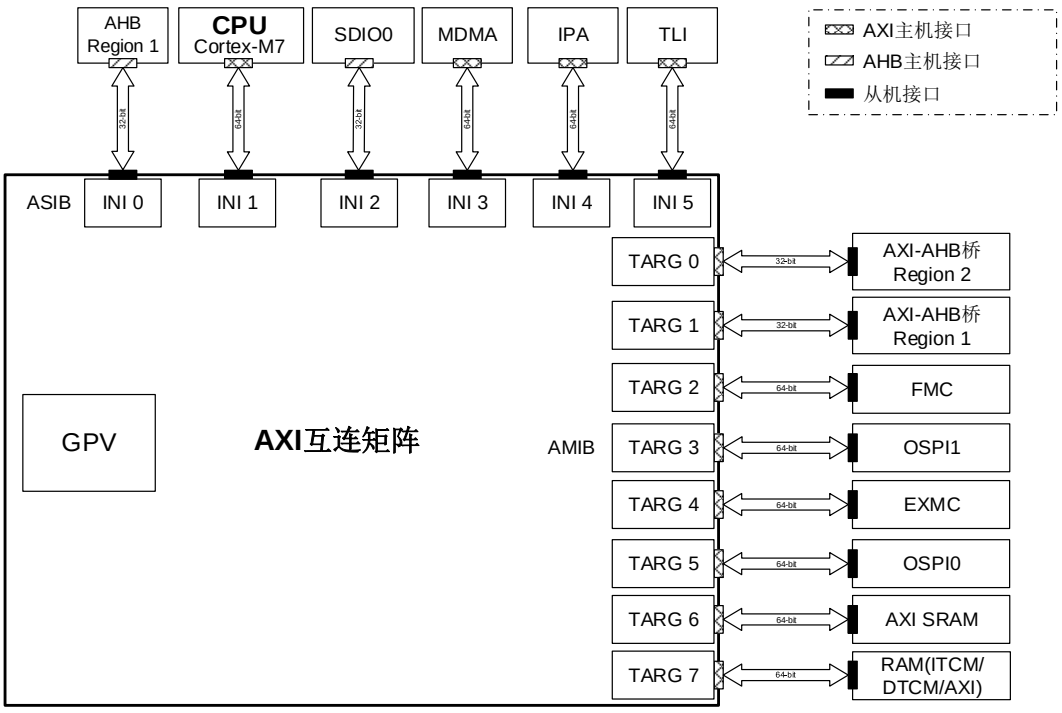
AXI 互连矩阵的主要功能如下：

- 具有6个ASIB和8个AMIB的64位AXI总线开关矩阵
- 分布式全局编程视图（GPV）
- 可编程服务质量（QoS）

1.7.2. 功能说明

AXI互连矩阵的框图如*图1-9. AXI互连矩阵的框图*所示。

图 1-9. AXI 互连矩阵的框图



ASIB和AMIB的配置如*表1-8. ASIB配置*和*表1-9. AMIB配置*所示。

表 1-8. ASIB 配置

ASIB	协议	总线宽度	读发布	写发布	主机接口
INI 0	AHB-lite	32	1	1	AHB Region 1
INI 1	AXI4	64	7	32	CPU Cortex-M7
INI 2	AHB-lite	32	1	1	SDIO0
INI 3	AXI4	64	4	1	MDMA
INI 4	AXI4	64	3	1	IPA
INI 5	AXI4	64	1	1	TLI

表 1-9. AMIB 配置

AMIB	协议	总线宽度	读验收	写验收	总体验收	从机接口
TARG 0	AXI4	32	1	1	1	AHB3外设和 Region 2
TARG 1	AXI4	32	1	1	1	Region 1
TARG 2	AXI4	64	3	2	5	FMC
TARG 3	AXI4	64	2	1	3	OSPI1
TARG 4	AXI4	64	3	3	6	EXMC
TARG 5	AXI4	64	2	1	3	OSPI0
TARG 6	AXI4	64	2	2	2	AXI SRAM AXI
TARG 7	AXI4	64	2	2	2	RAM(ITCM/DTCM/ AXI SRAM)

服务质量（QoS）

使用QoS为ASIB和AMIB提供可配置的QoS选项，为连接的AMBA主机提供读写请求调节和可编程QoS。当不同的ASIB尝试访问AMIB时，AXI互连矩阵使用基于优先级的仲裁。ASIB具有可配置的读写通道优先级。优先级范围为0x0~0xF，0为最低优先级。详情请参考***AXI从端口x读QOS控制寄存器（AXI_SPx_RDQOS_CTL）***和***AXI从端口x写QOS控制寄存器（AXI_SPx_WRQOS_CTL）***。当不同ASIB的优先级相同时，使用最近最少使用（LRU）优先级方案。

全局编程视图（GPV）

全局编程视图（GPV）用于可配置的整个互连，以便任何主接口或离散配置从接口都可以访问它。有关更多信息，请参阅Arm® CoreLink™ QoS-400网络互连高级服务质量和Arm® CoreLink™ NIC-400网络互连技术参考手册的补充。

1.8. 系统配置寄存器（SYSCFG）

SYSCFG基地址：0x5800 0400

1.8.1. 外设模式配置寄存器（SYSCFG_PMCFG）

地址偏移：0x004
复位值：0x0F00 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				PC3SWON	PC2SWON	PA1SWON	PA0SWON	ENET0_PHY_SEL	ENET1_PHY_SEL	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								PB9FMPEN	PB8FMPEN	PB7FMPEN	PB6FMPEN	I2C3FMPEN	I2C2FMPEN	I2C1FMPEN	I2C0FMPEN
								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	PC3SWON	PC3 开关打开 该位控制 PC3 和 PC3_C（双引脚）之间的模拟开关，决定引脚是通过模拟开关相连接还是分离。 0：关闭模拟开关 1：打开模拟开关（引脚分离）
26	PC2SWON	PC2 开关打开 该位控制 PC2 和 PC2_C（双引脚）之间的模拟开关，决定引脚是通过模拟开关相连接还是分离。 0：关闭模拟开关 1：打开模拟开关（引脚分离）
25	PA1SWON	PA1 开关打开 该位控制 PA1 和 PA1_C（双引脚）之间的模拟开关，决定引脚是通过模拟开关相连接还是分离。 0：关闭模拟开关 1：打开模拟开关（引脚分离）
24	PA0SWON	PA0 开关打开 该位控制 PA0 和 PA0_C（双引脚）之间的模拟开关，决定引脚是通过模拟开关相连接还是分离。 0：关闭模拟开关 1：打开模拟开关（引脚分离）
23	ENET0_PHY_SEL	以太网 0 PHY 接口选择

		该位选择以太网 PHY 接口。 0：选择 MII 1：选择 RMII
22	ENET1_PHY_SEL	以太网 1 PHY 接口选择 该位选择以太网 PHY 接口。 0：选择 MII 1：选择 RMII
21:8	保留	必须保持复位值。
7	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
6	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
5	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
4	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
3	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
2	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
1	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
0	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

1.8.2. EXTI 源选择寄存器 0（SYSCFG_EXTISS0）

地址偏移：0x008

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	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 0111：PH3 引脚
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 0111：PH2 引脚 1010：PK2 引脚
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 0111：PH1 引脚 1010：PK1 引脚
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 0111：PH0 引脚 1010：PK0 引脚

1.8.3. EXTI 源选择寄存器 1（SYSCFG_EXTISS1）

地址偏移：0x00C

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI7_SS[3:0]	EXTI 7 源选择 0000：PA7 引脚 0001：PB7 引脚 0010：PC7 引脚 0011：PD7 引脚 0100：PE7 引脚 0101：PF7 引脚 0110：PG7 引脚 0111：PH7 引脚
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 0111：PH6 引脚
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 0111：PH5 引脚
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 0111：PH4 引脚

1.8.4. EXTI 源选择寄存器 2（SYSCFG_EXTISS2）

地址偏移：0x010

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	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：保留 0001：PB11 引脚

		0010：PC11 引脚 0011：PD11 引脚 0100：PE11 引脚 0101：PF11 引脚 0110：PG11 引脚 0111：PH11 引脚 1001：PJ11 引脚
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 0111：PH10 引脚 1001：PJ10 引脚
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 0111：PH9 引脚 1001：PJ9 引脚
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 0111：PH8 引脚 1001：PJ8 引脚

1.8.5. EXTI 源选择寄存器 3（SYSCFG_EXTISS3）

地址偏移：0x014

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	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 0110：PG15 引脚 0111：PH15 引脚
11:8	EXTI14_SS[3:0]	EXTI 14 源选择 0000：PA14 引脚 0001：PB14 引脚 0010：PC14 引脚 0011：PD14 引脚 0100：PE14 引脚 0101：PF14 引脚 0110：PG14 引脚 0111：PH14 引脚
7:4	EXTI13_SS[3:0]	EXTI 13 源选择 0000：PA13 引脚 0001：PB13 引脚 0010：PC13 引脚 0011：PD13 引脚 0100：PE13 引脚 0101：PF13 引脚 0110：PG13 引脚 0111：PH13 引脚
3:0	EXTI12_SS[3:0]	EXTI 12 源选择 0000：保留 0001：PB12 引脚 0010：PC12 引脚 0011：PD12 引脚

0100：PE12 引脚
0101：PF12 引脚
0110：PG12 引脚
0111：PH12 引脚

1.8.6. 锁定控制寄存器（SYSCFG_LKCTL）

地址偏移：0x018
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AXIRAM_LOCK	ITCM_LOCK	DTCM_LOCK	SRAM0_LOCK	SRAM1_LOCK	保留			BKPRAM_LOCK	CPU_LOCK	保留			LVD_LOCK	保留	
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	AXIRAM_LOCK	Region 0 AXI-SRAM ECC 双比特错误锁定位 该位由软件置位，仅由复位系统清零。 0：Region 0 AXI-SRAM ECC 双比特错误信号从 TIMER0/7/14/15/16 中止输入端断开 1：Region 0 AXI-SRAM ECC 双比特错误信号与 TIMER0/7/14/15/16 中止输入端连接
14	ITCM_LOCK	Region 0 ITCM-RAM ECC 双比特错误锁定位 该位由软件置位，仅由复位系统清零。 0：Region 0 ITCM-RAM ECC 双比特错误信号从 TIMER0/7/14/15/16 中止输入端断开 1：Region 0 ITCM-RAM ECC 双比特错误信号与 TIMER0/7/14/15/16 中止输入端连接
13	DTCM_LOCK	Region 0 DTCM ECC 双比特错误锁定位 该位由软件置位，仅由复位系统清零。 0：Region 0 DTCM ECC 双比特错误信号从 TIMER0/7/14/15/16 中止输入端断开 1：Region 0 DTCM ECC 双比特错误信号与 TIMER0/7/14/15/16 中止输入端连接
12	SRAM0_LOCK	Region 1 SRAM0 ECC 双比特错误锁定位 该位由软件置位，仅由复位系统清零。 0：Region 1 SRAM0 ECC 双比特错误信号从 TIMER0/7/14/15/16 中止输入端断开 1：Region 1 SRAM0 ECC 双比特错误信号与 TIMER0/7/14/15/16 中止输入端连接
11	SRAM1_LOCK	Region 1 SRAM1 ECC 双比特错误锁定位

		该位由软件置位，仅由复位系统清零。 0：Region 1 SRAM1 ECC 双比特错误信号从 TIMER0/7/14/15/16 中止输入端断开 1：Region 1 SRAM1 ECC 双比特错误信号与 TIMER0/7/14/15/16 中止输入端连接
10:8	保留	必须保持复位值。
7	BKPRAM_LOCK	Region 2 备份 SRAM ECC 双比特错误锁定位 该位由软件置位，仅由复位系统清零。 0：Region 2 备份 SRAM ECC 双比特错误信号从 TIMER0/7/14/15/16 中止输入端断开 1：Region 2 备份 SRAM ECC 双比特错误信号与 TIMER0/7/14/15/16 中止输入端连接
6	CPU_LOCK	CPU 锁定位 该位由软件置位，仅由复位系统清零。 0：CPU 锁定信号从 TIMER0/7/14/15/16 中止输入端断开 1：CPU 锁定信号与 TIMER0/7/14/15/16 中止输入端连接
5:3	保留	必须保持复位值。
2	LVD_LOCK	LVD 锁定位 该位由软件置位，仅由复位系统清零。 0：LVD 信号从 TIMER0/7/14/15/16 中止输入端断开 1：LVD 信号与 TIMER0/7/14/15/16 中止输入端连接
1:0	保留	必须保持复位值。

1.8.7. I/O 补偿控制寄存器（SYSCFG_CPSCTL）

地址偏移：0x020
复位值：0x00X0 0000

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							CPS_RDY	保留							CPS_EN
							r								rw

位/位域	名称	描述
31:24	保留	必须保持复位值。
23	IOLV	I/O低电压状态 0：产品在2.5V以上工作 1：产品在2.5V以下工作

22:17	保留	必须保持复位值。
16	IOSPDOP	I/O 速度优化，低电压下高速 该位由软件写入，用于在产品电压较低时优化 I/O 速度。 仅当产品电源电压低于 2.5 V 时，才必须使用该位。当 V_{DD} 高于 2.5 V 时，设置该位可能具有破坏性。 0：无 I/O 速度优化 1：I/O 速度优化
15:9	保留	必须保持复位值。
8	CPS_RDY	I/O 补偿单元是否准备好，该位只读。 0：I/O 补偿单元没有准备好 1：I/O 补偿单元准备好
7:1	保留	必须保持复位值。
0	CPS_EN	I/O 补偿单元使能 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

1.8.8. I/O 补偿单元代码配置寄存器（SYSCFG_CPSCCCFG）

地址偏移：0x028
复位值：0x0000 0088

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	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								PCPSCC[3:0]				NCPSCC[3:0]			
								rw				rw			

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:4	PCPSCC[3:0]	PMOS 补偿单元代码 这些位定义了 PMOS 晶体管的 I/O 补偿单元代码。
3:0	NCPSCC[3:0]	NMOS 补偿单元代码 这些位定义了 NMOS 晶体管的 I/O 补偿单元代码。

1.8.9. TIMER 输入选择寄存器 0（SYSCFG_TIMERCISEL0）

地址偏移：0x034
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIMER0_CI3_SEL[3:0]				TIMER0_CI2_SEL[3:0]				TIMER0_CI1_SEL[3:0]				TIMER0_CI0_SEL[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER7_CI3_SEL[3:0]				TIMER7_CI2_SEL[3:0]				TIMER7_CI1_SEL[3:0]				TIMER7_CI0_SEL[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:28	TIMER0_CI3_SEL[3:0]	TIMER0_CI3 输入选择 这些位选择 TIMER 输入源。 0000：TIMER0_CH3 输入 其他：保留
27:24	TIMER0_CI2_SEL[3:0]	TIMER0_CI2 输入选择 这些位选择 TIMER 输入源。 0000：TIMER0_CH2 输入 其他：保留
23:20	TIMER0_CI1_SEL[3:0]	TIMER0_CI1 输入选择 这些位选择 TIMER 输入源。 0000：TIMER0_CH1 输入 其他：保留
19:16	TIMER0_CI0_SEL[3:0]	TIMER0_CI0 输入选择 这些位选择 TIMER 输入源。 0000：TIMER0_CH0 输入 0001：CMP0 输出 其他：保留
15:12	TIMER7_CI3_SEL[3:0]	TIMER7_CI3 输入选择 这些位选择 TIMER 输入源。 0000：TIMER7_CH3 输入 其他：保留
11:8	TIMER7_CI2_SEL[3:0]	TIMER7_CI2 输入选择 这些位选择 TIMER 输入源。 0000：TIMER7_CH2 输入 其他：保留
7:4	TIMER7_CI1_SEL[3:0]	TIMER7_CI1 输入选择 这些位选择 TIMER 输入源。 0000：TIMER7_CH1 输入 其他：保留
3:0	TIMER7_CI0_SEL[3:0]	TIMER7_CI0 输入选择 这些位选择 TIMER 输入源。 0000：TIMER7_CH0 输入

0001：CMP1 输出
其他：保留

1.8.10. TIMER 输入选择寄存器 1（SYSCFG_TIMERCISEL1）

地址偏移：0x038
复位值：0x0000 0000

该寄存器只能按字（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]	TIMER1_CI3 输入选择 这些位选择 TIMER 输入源。 0000：TIMER1_CH3 输入 0001：CMP0 输出 0010：CMP1 输出 0011：CMP0 输出 OR CMP1 输出 其他：保留
27:24	TIMER1_CI2_SEL[3:0]	TIMER1_CI2 输入选择 这些位选择 TIMER 输入源。 0000：TIMER1_CH2 输入 其他：保留
23:20	TIMER1_CI1_SEL[3:0]	TIMER1_CI1 输入选择 这些位选择 TIMER 输入源。 0000：TIMER1_CH1 输入 其他：保留
19:16	TIMER1_CI0_SEL[3:0]	TIMER1_CI0 输入选择 这些位选择 TIMER 输入源。 0000：TIMER1_CH0 输入 其他：保留
15:12	TIMER2_CI3_SEL[3:0]	TIMER2_CI3 输入选择 这些位选择 TIMER 输入源。 0000：TIMER2_CH3 输入 其他：保留
11:8	TIMER2_CI2_SEL[3:0]	TIMER2_CI2 输入选择 这些位选择 TIMER 输入源。

0000：TIMER2_CH2 输入
其他：保留

7:4	TIMER2_CI1_SEL[3:0]	TIMER2_CI1 输入选择 这些位选择 TIMER 输入源。 0000：TIMER2_CH1 输入 其他：保留
3:0	TIMER2_CI0_SEL[3:0]	TIMER2_CI0 输入选择 这些位选择 TIMER 输入源。 0000：TIMER2_CH0 输入 0001：CMP0 输出 0010：CMP1 输出 0011：CMP0 输出或 CMP1 输出 其他：保留

1.8.11. TIMER 输入选择寄存器 2（SYSCFG_TIMERCISEL2）

地址偏移：0x03C
复位值：0x0000 0000

该寄存器只能按字（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]	TIMER3_CI3 输入选择 这些位选择 TIMER 输入源。 0000：TIMER3_CH3 输入 其他：保留
27:24	TIMER3_CI2_SEL[3:0]	TIMER3_CI2 输入选择 这些位选择 TIMER 输入源。 0000：TIMER3_CH2 输入 其他：保留
23:20	TIMER3_CI1_SEL[3:0]	TIMER3_CI1 输入选择 这些位选择 TIMER 输入源。 0000：TIMER3_CH1 输入 其他：保留
19:16	TIMER3_CI0_SEL[3:0]	TIMER3_CI0 输入选择 这些位选择 TIMER 输入源。

		0000：TIMER3_CH0 输入 其他：保留
15:12	TIMER4_CI3_SEL[3:0]	TIMER4_CI3 输入选择 这些位选择 TIMER 输入源。 0000：TIMER4_CH3 输入 其他：保留
11:8	TIMER4_CI2_SEL[3:0]	TIMER4_CI2 输入选择 这些位选择 TIMER 输入源。 0000：TIMER4_CH2 输入 其他：保留
7:4	TIMER4_CI1_SEL[3:0]	TIMER4_CI1 输入选择 这些位选择 TIMER 输入源。 0000：TIMER4_CH1 输入 其他：保留
3:0	TIMER4_CI0_SEL[3:0]	TIMER4_CI0 输入选择 这些位选择 TIMER 输入源。 0000：TIMER4_CH0 输入 其他：保留

1.8.12. TIMER 输入选择寄存器 3（SYSCFG_TIMERCISEL3）

地址偏移：0x040
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIMER22_CI3_SEL[3:0]				TIMER22_CI2_SEL[3:0]				TIMER22_CI1_SEL[3:0]				TIMER22_CI0_SEL[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER23_CI3_SEL[3:0]				TIMER23_CI2_SEL[3:0]				TIMER23_CI1_SEL[3:0]				TIMER23_CI0_SEL[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:28	TIMER22_CI3_SEL[3:0]	TIMER22_CI3 输入选择 这些位选择 TIMER 输入源。 0000：TIMER22_CH3 输入 0001：CMP0 输出 0010：CMP1 输出 0011：CMP0 输出或 CMP1 输出 其他：保留
27:24	TIMER22_CI2_SEL[3:0]	TIMER22_CI2 输入选择 这些位选择 TIMER 输入源。

0000：TIMER22_CH2 输入
其他：保留

23:20	TIMER22_CI1_SEL[3:0]	TIMER22_CI1 输入选择 这些位选择 TIMER 输入源。 0000：TIMER22_CH1 输入 其他：保留
19:16	TIMER22_CI0_SEL[3:0]	TIMER22_CI0 输入选择 这些位选择 TIMER 输入源。 0000：TIMER22_CH0 输入 其他：保留
15:12	TIMER23_CI3_SEL[3:0]	TIMER23_CI3 输入选择 这些位选择 TIMER 输入源。 0000：TIMER23_CH3 输入 其他：保留
11:8	TIMER23_CI2_SEL[3:0]	TIMER23_CI2 输入选择 这些位选择 TIMER 输入源。 0000：TIMER23_CH2 输入 其他：保留
7:4	TIMER23_CI1_SEL[3:0]	TIMER23_CI1 输入选择 这些位选择 TIMER 输入源。 0000：TIMER23_CH1 输入 其他：保留
3:0	TIMER23_CI0_SEL[3:0]	TIMER23_CI0 输入选择 这些位选择 TIMER 输入源。 0000：TIMER23_CH0 输入 其他：保留

1.8.13. TIMER 输入选择寄存器 4（SYSCFG_TIMERCISEL4）

地址偏移：0x044
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIMER30_CI3_SEL[3:0]				TIMER30_CI2_SEL[3:0]				TIMER30_CI1_SEL[3:0]				TIMER30_CI0_SEL[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER31_CI3_SEL[3:0]				TIMER31_CI2_SEL[3:0]				TIMER31_CI1_SEL[3:0]				TIMER31_CI0_SEL[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:28	TIMER30_CI3_SEL[3	TIMER30_CI3 输入选择

	:0]	这些位选择 TIMER 输入源。 0000：TIMER30_CH3 输入 其他：保留
27:24	TIMER30_CI2_SEL[3 :0]	TIMER30_CI2 输入选择 这些位选择 TIMER 输入源。 0000：TIMER30_CH2 输入 其他：保留
23:20	TIMER30_CI1_SEL[3 :0]	TIMER30_CI1 输入选择 这些位选择 TIMER 输入源。 0000：TIMER30_CH1 输入 其他：保留
19:16	TIMER30_CI0_SEL[3 :0]	TIMER30_CI0 输入选择 这些位选择 TIMER 输入源。 0000：TIMER30_CH0 输入 0001：CMP0 输出 0010：CMP1 输出 0011：CMP0 输出或 CMP1 输出 其他：保留
15:12	TIMER31_CI3_SEL[3 :0]	TIMER31_CI3 输入选择 这些位选择 TIMER 输入源。 0000：TIMER31_CH3 输入 其他：保留
11:8	TIMER31_CI2_SEL[3 :0]	TIMER31_CI2 输入选择 这些位选择 TIMER 输入源。 0000：TIMER31_CH2 输入 其他：保留
7:4	TIMER31_CI1_SEL[3 :0]	TIMER31_CI1 输入选择 这些位选择 TIMER 输入源。 0000：TIMER31_CH1 输入 其他：保留
3:0	TIMER31_CI0_SEL[3 :0]	TIMER31_CI0 输入选择 这些位选择 TIMER 输入源。 0000：TIMER31_CH0 输入 0001：CMP0 输出 0010：CMP1 输出 0011：CMP0 输出或 CMP1 输出 其他：保留

1.8.14. TIMER 输入选择寄存器 5（SYSCFG_TIMERCISEL5）

地址偏移：0x048

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIMER42_CI1_SEL[3:0]				TIMER42_CI0_SEL[3:0]				TIMER41_CI1_SEL[3:0]				TIMER41_CI0_SEL[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER40_CI1_SEL[3:0]				TIMER40_CI0_SEL[3:0]				TIMER14_CI1_SEL[3:0]				TIMER14_CI0_SEL[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:28	TIMER42_CI1_SEL[3:0]	TIMER42_CI1 输入选择 这些位选择 TIMER 输入源。 0000：TIMER42_CH1 输入 0001：TIMER4_CH1 输入 0010：TIMER22_CH1 输入 0011：TIMER23_CH1 输入 其他：保留
27:24	TIMER42_CI0_SEL[3:0]	TIMER42_CI0 输入选择 这些位选择 TIMER 输入源。 0000：TIMER42_CH0 输入 0001：TIMER4_CH0 输入 0010：TIMER22_CH0 输入 0011：TIMER23_CH0 输入 0100：LXTAL 0101：LPIRC4M 0110：CKOUT1 其他：保留
23:20	TIMER41_CI1_SEL[3:0]	TIMER41_CI1 输入选择 这些位选择 TIMER 输入源。 0000：TIMER41_CH1 输入 0001：TIMER3_CH1 输入 0010：TIMER4_CH1 输入 0011：TIMER22_CH1 输入 其他：保留
19:16	TIMER41_CI0_SEL[3:0]	TIMER41_CI0 输入选择 这些位选择 TIMER 输入源。 0000：TIMER41_CH0 输入 0001：TIMER3_CH0 输入 0010：TIMER4_CH0 输入 0011：TIMER22_CH0 输入 0100：LXTAL 0101：LPIRC4M

0110：CKOUT1
其他：保留

15:12	TIMER40_CI1_SEL[3:0]	TIMER40_CI1 输入选择 这些位选择 TIMER 输入源。 0000：TIMER40_CH1 输入 0001：TIMER2_CH1 输入 0010：TIMER3_CH1 输入 0011：TIMER4_CH1 输入 其他：保留
11:8	TIMER40_CI0_SEL[3:0]	TIMER40_CI0 输入选择 这些位选择 TIMER 输入源。 0000：TIMER40_CH0 输入 0001：TIMER2_CH0 输入 0010：TIMER3_CH0 输入 0011：TIMER4_CH0 输入 0100：LXTAL 0101：LPIRC4M 0110：CKOUT1 其他：保留
7:4	TIMER14_CI1_SEL[3:0]	TIMER14_CI1 输入选择 这些位选择 TIMER 输入源。 0000：TIMER14_CH1 输入 0001：TIMER1_CH1 输入 0010：TIMER2_CH1 输入 0011：TIMER3_CH1 输入 其他：保留
3:0	TIMER14_CI0_SEL[3:0]	TIMER14_CI0 输入选择 这些位选择 TIMER 输入源。 0000：TIMER14_CH0 输入 0001：TIMER1_CH0 输入 0010：TIMER2_CH0 输入 0011：TIMER3_CH0 输入 0100：LXTAL 0101：LPIRC4M 0110：CKOUT1 其他：保留

1.8.15. TIMER 输入选择寄存器 6（SYSCFG_TIMERCISEL6）

地址偏移：0x04C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								TIMER44_CI1_SEL[3:0]				TIMER44_CI0_SEL[3:0]			
								rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER43_CI1_SEL[3:0]				TIMER43_CI0_SEL[3:0]				TIMER16_CI0_SEL[3:0]				TIMER15_CI0_SEL[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	TIMER44_CI1_SEL[3:0]	TIMER44_CI1 输入选择 这些位选择 TIMER 输入源。 0000：TIMER44_CH1 输入 0001：TIMER23_CH1 输入 0010：TIMER30_CH1 输入 0011：TIMER31_CH1 输入 其他：保留
19:16	TIMER44_CI0_SEL[3:0]	TIMER44_CI0 输入选择 这些位选择 TIMER 输入源。 0000：TIMER44_CH0 输入 0001：TIMER23_CH0 输入 0010：TIMER30_CH0 输入 0011：TIMER31_CH0 输入 0100：LXTAL 0101：LPIRC4M 0110：CKOUT1 其他：保留
15:12	TIMER43_CI1_SEL[3:0]	TIMER43_CI1 输入选择 这些位选择 TIMER 输入源。 0000：TIMER43_CH1 输入 0001：TIMER22_CH1 输入 0010：TIMER23_CH1 输入 0011：TIMER30_CH1 输入 其他：保留
11:8	TIMER43_CI0_SEL[3:0]	TIMER43_CI0 输入选择 这些位选择 TIMER 输入源。 0000：TIMER43_CH0 输入 0001：TIMER22_CH0 输入 0010：TIMER23_CH0 输入 0011：TIMER30_CH0 输入 0100：LXTAL 0101：LPIRC4M 0110：CKOUT1

其他：保留

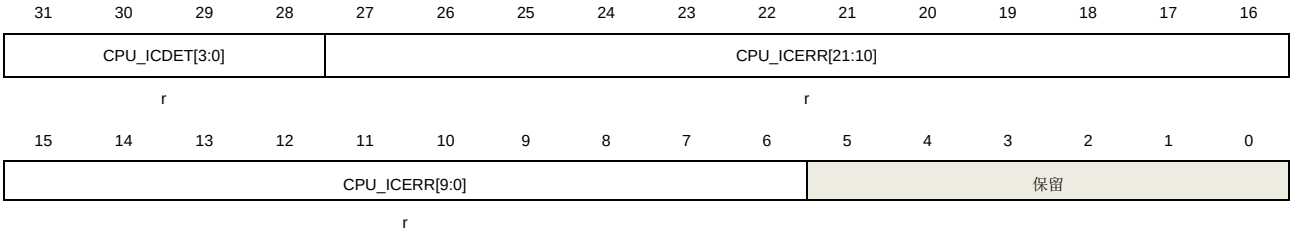
7:4	TIMER16_CI0_SEL[3:0]	TIMER16_CI0 输入选择 这些位选择 TIMER 输入源。 0000：TIMER16_CH0 输入 0001: RSPDIF symbol_ck 0010：CK_HXTAL / RTCDIV 0011：CKOUT0 其他：保留
3:0	TIMER15_CI0_SEL[3:0]	TIMER15_CI0 输入选择 这些位选择 TIMER 输入源。 0000：TIMER15_CH0 输入 0001：IRC32K 0010：LXTAL 0011：WKUP_IT 其他：保留

1.8.16. CPU ICACHE 错误状态寄存器（SYSCFG_CPUICAC）

地址偏移：0x054

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



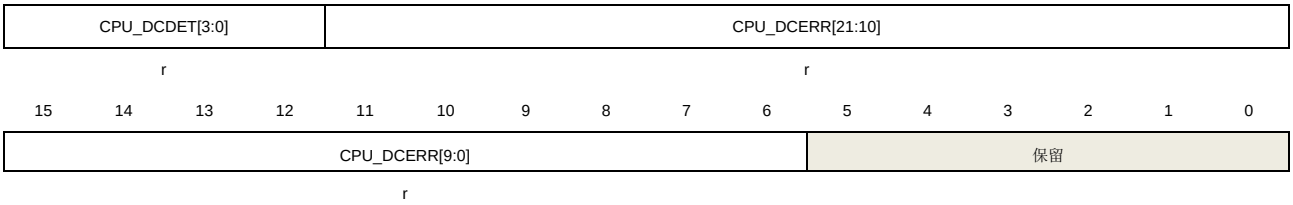
位/位域	名称	描述
31:28	CPU_ICDET[3:0]	ICACHE 错误检测信息 这些位由 CPU 提供，用于指示 ICACHE 错误检测信息。
27:6	CPU_ICERR[21:0]	ICACHE 错误库信息 这些位由 CPU 提供，用于指示 ICACHE 错误库信息。
5:0	保留	必须保持复位值。

1.8.17. CPU DCACHE 错误状态寄存器（SYSCFG_CPUDCAC）

地址偏移：0x058

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

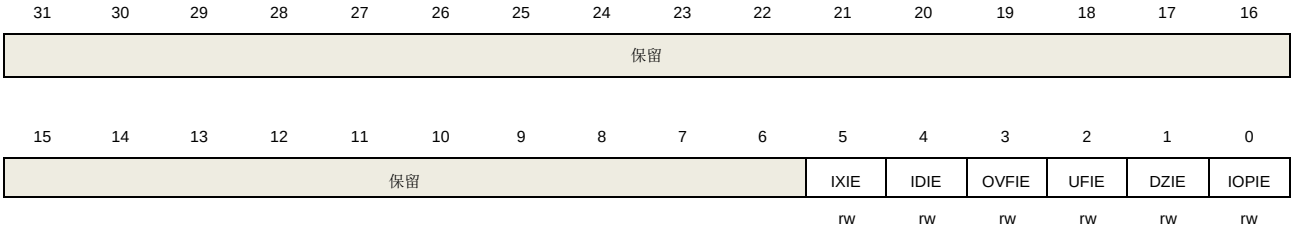


位/位域	名称	描述
31:28	CPU_DCDET[3:0]	DCACHE 错误检测信息 这些位由 CPU 提供，用于指示 DCACHE 错误检测信息。
27:6	CPU_DCERR[21:0]	DCACHE 错误库信息 这些位由 CPU 提供，用于指示 DCACHE 错误库信息。
5:0	保留	必须保持复位值。

1.8.18. FPU 中断使能寄存器（SYSCFG_FPUINTEN）

地址偏移：0x05C
复位值：0x0000 001F

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	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
4	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
3	OVFIE	溢出中断使能位 0：溢出中断禁能 1：溢出中断使能
2	UFIE	下溢中断使能位 0：下溢中断禁能 1：下溢中断使能
1	DZIE	除0中断使能位

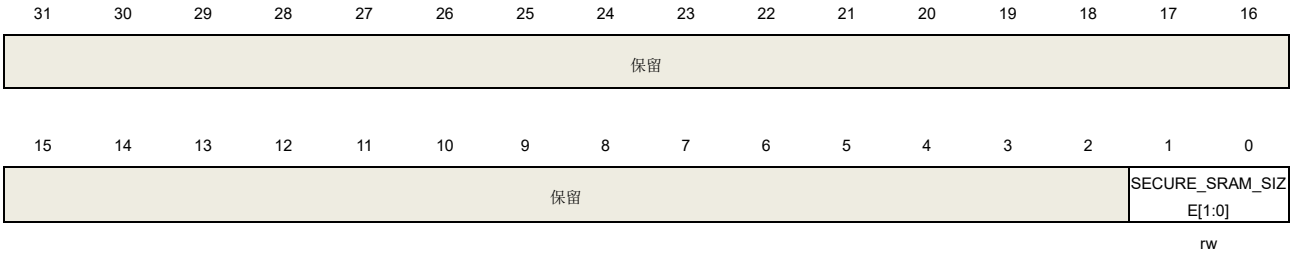
0：除0中断禁能
1：除0中断使能

0	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

1.8.19. SRAM 配置寄存器 0（SYSCFG_SRAMCFG0）

地址偏移：0x64
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

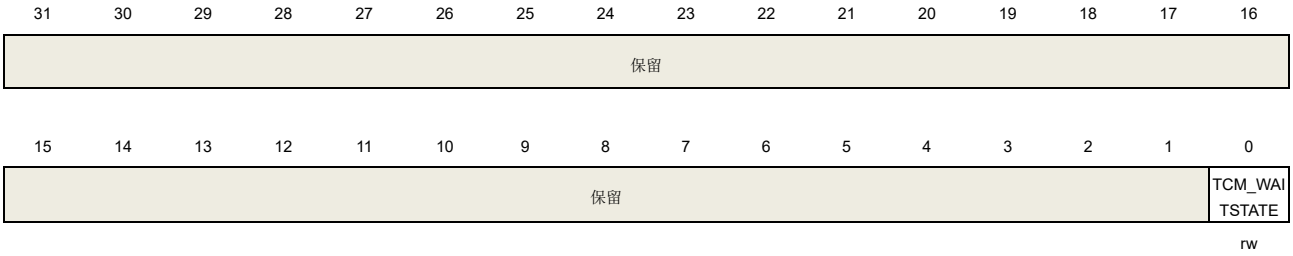


位/位域	名称	描述
31:2	保留	必须保持复位值。
1:0	SECURE_SRAM_SIZE[1:0]	安全SRAM的大小。 这些位由软件设置，仅通过电源复位清除。 00：0 KB 01：32 KB 10：64 KB 11：128 KB

1.8.20. SRAM 配置寄存器 1（SYSCFG_SRAMCFG1）

地址偏移：0x68
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	TCM_WAITSTATE	TCM等待周期配置 该位由软件设置和清除。只能通过系统复位来复位。 该位用于在ITCM / D0TCM / D1TCM访问中插入等待周期。 **注意**：当系统频率高于f_{TWW}时，该位需要被置1。 0：无等待周期 1：插入等待周期

1.8.21. TIMERx 配置寄存器 0（SYSCFG_TIMERxCFG0, x=0, 7）

地址偏移：0x100 for TIMER0
地址偏移：0x13C for TIMER7
复位值：0x0000 0000

TSCFG0[4:0]，TSCFG1[4:0]..TSCFG9[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：滤波后的通道 2 输入（CI2FE2） 01010：滤波后的通道 3 输入（CI3FE3） 01011：滤波后的多模式通道 0 输入（MCI0FEM0） 01100：滤波后的多模式通道 1 输入（MCI1FEM1） 01101：滤波后的多模式通道 2 输入（MCI2FEM2） 01110：滤波后的多模式通道 3 输入（MCI3FEM3）

01111：保留
10000：保留
10001：内部触发输入 12（ITI12）
10010：内部触发输入 13（ITI13）
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，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：滤波后的通道 2 输入（CI2FE2） 01010：滤波后的通道 3 输入（CI3FE3） 01011：滤波后的多模式通道 0 输入（MCI0FEM0） 01100：滤波后的多模式通道 1 输入（MCI1FEM1） 01101：滤波后的多模式通道 2 输入（MCI2FEM2） 01110：滤波后的多模式通道 3 输入（MCI3FEM3） 01111：保留 10000：保留 10001：内部触发输入 12（ITI12） 10010：内部触发输入 13（ITI13） 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：滤波后的外部触发输入（ETIFP） 01001：滤波后的通道 2 输入（CI2FE2） 01010：滤波后的通道 3 输入（CI3FE3） 01011：滤波后的多模式通道 0 输入（MCI0FEM0）

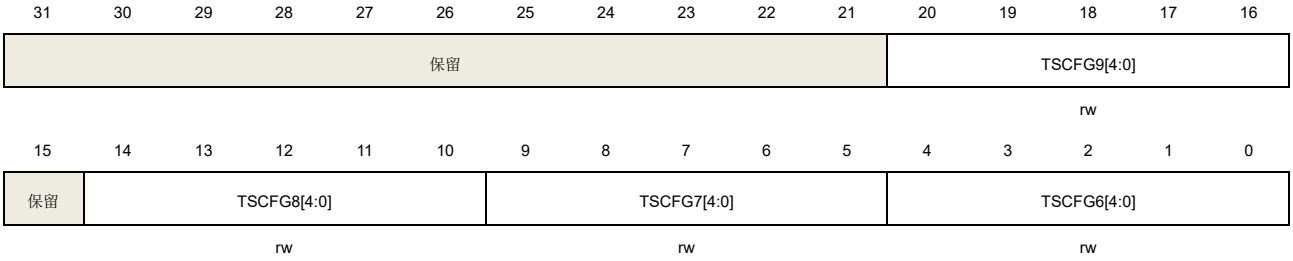
		01100：滤波后的多模式通道 1 输入（MCI1FEM1） 01101：滤波后的多模式通道 2 输入（MCI2FEM2） 01110：滤波后的多模式通道 3 输入（MCI3FEM3） 01111：保留 10000：保留 10001：内部触发输入 12（ITI12） 10010：内部触发输入 13（ITI13） 10011：内部触发输入 14（ITI14） 其他：保留
15	保留	必须保持复位值。
14:10	TSCFG2[4:0]	编码器模式 2 配置 00000：编码器模式 2 禁能 其他：根据另一个信号的输入电平，计数器在 CI0FE0 和 CI1FE1 的边沿向上/下计数。
9:5	TSCFG1[4:0]	编码器模式 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 的边沿向上/下计数。
4:0	TSCFG0[4:0]	编码器模式 0 配置 00000：编码器模式 0 禁能 其他：根据 CI1FE1 的电平，计数器在 CI0FE0 的边沿向上/下计数。

1.8.22. TIMERx 配置寄存器 1（SYSCFG_TIMERxCFG1, x=0, 7）

地址偏移：0x104 for TIMER0
地址偏移：0x140 for TIMER7
复位值：0x0000 0000

TSCFG0[4:0]，TSCFG1[4:0]..TSCFG9[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。



位/位域	名称	描述
31:21	保留	必须保持复位值。
20:16	TSCFG9[4:0]	非正交译码器模式 1 配置 00000：非正交译码器模式 1 禁能

		其他：CI0 作为计数脉冲，CH0P 用于选择计数边沿，CI1 作为计数方向选择信号。
15	保留	必须保持复位值。
14:10	TSCFG8[4:0]	非正交译码器模式 0 配置 00000：非正交译码器模式 0 禁能 其他：CI0 作为计数脉冲，CI1 作为计数选择信号。CH1P=0 时，只有当 CI1 输入信号为高电平时，计数器才会在 CI0 输入信号的上升沿向上计数；CH1P=1 时，只有当 CI1 输入信号为低电平时，计数器才会在 CI0 输入信号的上升沿向上计数。
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和启动计数器，并且更新影子寄存器。 00000：复位+事件模式配置禁能 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：滤波后的通道 2 输入（CI2FE2） 01010：滤波后的通道 3 输入（CI3FE3） 01011：滤波后的多模式通道 0 输入（MCI0FEM0） 01100：滤波后的多模式通道 1 输入（MCI1FEM1） 01101：滤波后的多模式通道 2 输入（MCI2FEM2） 01110：滤波后的多模式通道 3 输入（MCI3FEM3） 01111：保留 10000：保留 10001：内部触发输入 12（ITI12） 10010：内部触发输入 13（ITI13） 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 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：滤波后的通道 2 输入（CI2FE2）

01010：滤波后的通道 3 输入（CI3FE3）
01011：滤波后的多模式通道 0 输入（MCI0FEM0）
01100：滤波后的多模式通道 1 输入（MCI1FEM1）
01101：滤波后的多模式通道 2 输入（MCI2FEM2）
01110：滤波后的多模式通道 3 输入（MCI3FEM3）
01111：保留
10000：保留
10001：内部触发输入 12（ITI12）
10010：内部触发输入 13（ITI13）
10011：内部触发输入 14（ITI14）
其他：保留

1.8.23. TIMERx 配置寄存器 2（SYSCFG_TIMERxCFG2, x=0, 7）

地址偏移：0x108 for TIMER0
地址偏移：0x144 for TIMER7
复位值：0x0000 0000

TSCFG0[4:0]，TSCFG1[4:0]..TSCFG9[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															

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：内部触发输入 0（ITI0） 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：保留 01010：保留 01011：保留 01100：保留

01101：保留
01110：保留
01111：保留
10000：保留
10001：内部触发输入 12（ITI12）
10010：内部触发输入 13（ITI13）
10011：内部触发输入 14（ITI14）
其他：保留
注意：使用 TSCFG15[4:0]时，需保证 TSCFGy[4:0]（y=0..9）为零，否则 ITS 触发源需与 TSCFGy[4:0]选择的通道输入源保持一致。

15:0	保留	必须保持复位值。

1.8.24. TIMERx 配置寄存器 0（SYSCFG_TIMERxCFG0, x=1, 2, 3, 4, 22, 23, 30, 31）

地址偏移：0x10C for TIMER1
地址偏移：0x118 for TIMER2
地址偏移：0x124 for TIMER3
地址偏移：0x130 for TIMER4
地址偏移：0x154 for TIMER22
地址偏移：0x160 for TIMER23
地址偏移：0x16C for TIMER30
地址偏移：0x178 for TIMER31
复位值：0x0000 0000

TSCFG0[4:0]，TSCFG1[4:0]..TSCFG9[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：保留
01100：内部触发输入 7（ITI7）
01101：保留
01110：内部触发输入 9（ITI9）
01111：内部触发输入 10（ITI10）
10000：内部触发输入 11（ITI11）
10001：内部触发输入 12（ITI12）
10010：内部触发输入 13（ITI13）
10011：内部触发输入 14（ITI14）
其他：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，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：保留
01100：内部触发输入 7（ITI7）
01101：保留
01110：内部触发输入 9（ITI9）
01111：内部触发输入 10（ITI10）
10000：内部触发输入 11（ITI11）
10001：内部触发输入 12（ITI12）
10010：内部触发输入 13（ITI13）
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：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：保留 01100：内部触发输入 7（ITI7） 01101：保留 01110：内部触发输入 9（ITI9） 01111：内部触发输入 10（ITI10） 10000：内部触发输入 11（ITI11） 10001：内部触发输入 12（ITI12） 10010：内部触发输入 13（ITI13） 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]	编码器模式 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 的边沿向上/下计数。
9:5	TSCFG1[4:0]	编码器模式 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 的边沿向上/下计数。
4:0	TSCFG0[4:0]	编码器模式 0 配置 00000：编码器模式 0 禁能 其他：根据 CI1FE1 的电平，计数器在 CI0FE0 的边沿向上/下计数。

1.8.25. TIMERx 配置寄存器 1（SYSCFG_TIMERxCFG1, x=1, 2, 3, 4, 22, 23, 30, 31）

地址偏移：0x110 for TIMER1
地址偏移：0x11C for TIMER2
地址偏移：0x128 for TIMER3
地址偏移：0x134 for TIMER4
地址偏移：0x158 for TIMER22
地址偏移：0x164 for TIMER23
地址偏移：0x170 for TIMER30
地址偏移：0x17C for TIMER31
复位值：0x0000 0000

TSCFG0[4:0]，TSCFG1[4:0]..TSCFG9[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											TSCFG9[4:0]				
											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
31:21	保留	必须保持复位值。
20:16	TSCFG9[4:0]	非正交译码器模式 1 配置 00000：非正交译码器模式 1 禁能 其他：CI0作为计数脉冲，CH0P用于选择计数边沿，CI1作为计数方向选择信号。
15	保留	必须保持复位值。
14:10	TSCFG8[4:0]	非正交译码器模式 0 配置 00000：非正交译码器模式 0 禁能 其他：CI0作为计数脉冲，CI1作为计数选择信号。CH1P=0时，只有当CI1输入信号为高电平时，计数器才会在CI0输入信号的上升沿向上计数；CH1P=1时，只有当CI1输入信号为低电平时，计数器才会在CI0输入信号的上升沿向上计数。
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和启动计数器，并且更新影子寄存器。 00000：复位+事件模式配置禁能 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：保留 01100：内部触发输入 7（ITI7） 01101：保留 01110：内部触发输入 9（ITI9） 01111：内部触发输入 10（ITI10） 10000：内部触发输入 11（ITI11） 10001：内部触发输入 12（ITI12） 10010：内部触发输入 13（ITI13）

		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 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：保留 01100：内部触发输入 7（ITI7） 01101：保留 01110：内部触发输入 9（ITI9） 01111：内部触发输入 10（ITI10） 10000：内部触发输入 11（ITI11） 10001：内部触发输入 12（ITI12） 10010：内部触发输入 13（ITI13） 10011：内部触发输入 14（ITI14） 其他：保留

1.8.26. TIMERx 配置寄存器 2（SYSCFG_TIMERxCFG2, x=1, 2, 3, 4, 22, 23, 30, 31）

地址偏移：0x114 for TIMER1
地址偏移：0x120 for TIMER2
地址偏移：0x12C for TIMER3
地址偏移：0x138 for TIMER4
地址偏移：0x15C for TIMER22
地址偏移：0x168 for TIMER23
地址偏移：0x174 for TIMER30
地址偏移：0x180 for TIMER31
复位值：0x0000 0000

TSCFG0[4:0]，TSCFG1[4:0]..TSCFG9[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															

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：保留 01100：内部触发输入 7（ITI7） 01101：保留 01110：内部触发输入 9（ITI9） 01111：内部触发输入 10（ITI10） 10000：内部触发输入 11（ITI11） 10001：内部触发输入 12（ITI12） 10010：内部触发输入 13（ITI13） 10011：内部触发输入 14（ITI14） 其他：保留 **注意：**使用 TSCFG15[4:0]时，需保证 TSCFGy[4:0]（y=0..9）为零，否则 ITS 触发源需与 TSCFGy[4:0]选择的通道输入源保持一致。
15:0	保留	必须保持复位值。

1.8.27. TIMERx 配置寄存器 0（SYSCFG_TIMERxCFG0, x=14, 40, 41, 42, 43, 44）

地址偏移：0x148 for TIMER14
地址偏移：0x184 for TIMER40
地址偏移：0x190 for TIMER41
地址偏移：0x19C for TIMER42
地址偏移：0x1A8 for TIMER43
地址偏移：0x1B4 for TIMER44
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	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：保留 01010：保留 01011：滤波后的多模式通道 0 输入（MCI0FEM0） 10011：内部触发输入 14（ITI14） 其他：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，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：保留 01010：保留 01011：滤波后的多模式通道 0 输入（MCI0FEM0） 10011：内部触发输入 14（ITI14） 其他：保留

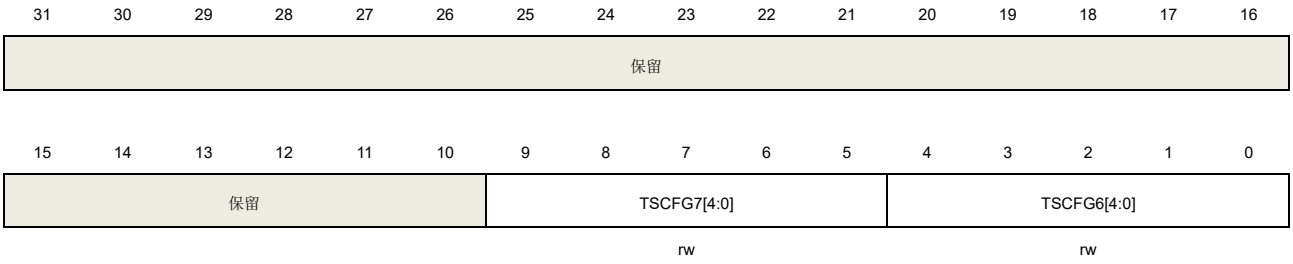
20:16	TSCFG3[4:0]	复位模式配置 选中的触发输入的上升沿重新初始化计数器，并且更新影子寄存器。 00000：复位模式禁能 00001：内部触发输入 0（ITI0） 00010：内部触发输入 1（ITI1） 00011：内部触发输入 2（ITI2） 00100：内部触发输入 3（ITI3） 00101：CI0 的边沿标志位（CI0F_ED） 00110：滤波后的通道 0 输入（CI0FE0） 00111：滤波后的通道 1 输入（CI1FE1） 01000：保留 01001：保留 01010：保留 01011：滤波后的多模式通道 0 输入（MCI0FEM0） 10011：内部触发输入 14（ITI14） 其他：保留
15:0	保留	必须保持复位值。

1.8.28. TIMERx 配置寄存器 1（SYSCFG_TIMERxCFG1, x=14, 40, 41, 42, 43, 44）

地址偏移：0x14C for TIMER14
地址偏移：0x188 for TIMER40
地址偏移：0x194 for TIMER41
地址偏移：0x1A0 for TIMER42
地址偏移：0x1AC for TIMER43
地址偏移：0x1B8 for TIMER44
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。



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 选中的触发输入的上升沿重新初始化和启动计数器，并且更新影子寄存器。 00000：复位+事件模式配置禁能

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：保留
01010：保留
01011：滤波后的多模式通道 0 输入（MCI0FEM0）
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 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：保留 01010：保留 01011：滤波后的多模式通道 0 输入（MCI0FEM0） 10011：内部触发输入 14（ITI14） 其他：保留

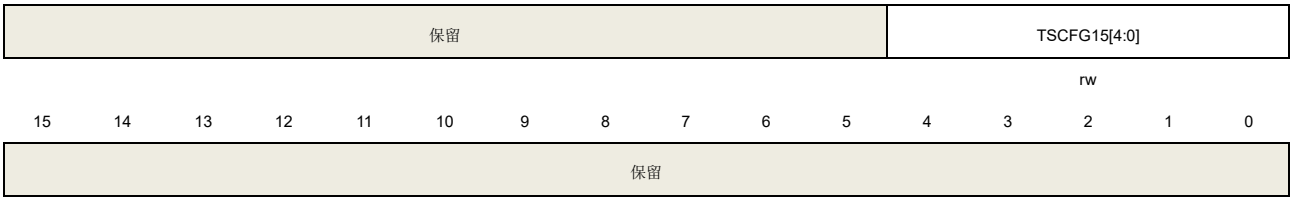
1.8.29. TIMERx 配置寄存器 2（SYSCFG_TIMERxCFG2, x=14, 40, 41, 42, 43, 44）

地址偏移：0x150 for TIMER14
地址偏移：0x18C for TIMER40
地址偏移：0x198 for TIMER41
地址偏移：0x1A4 for TIMER42
地址偏移：0x1B0 for TIMER43
地址偏移：0x1BC for TIMER44
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



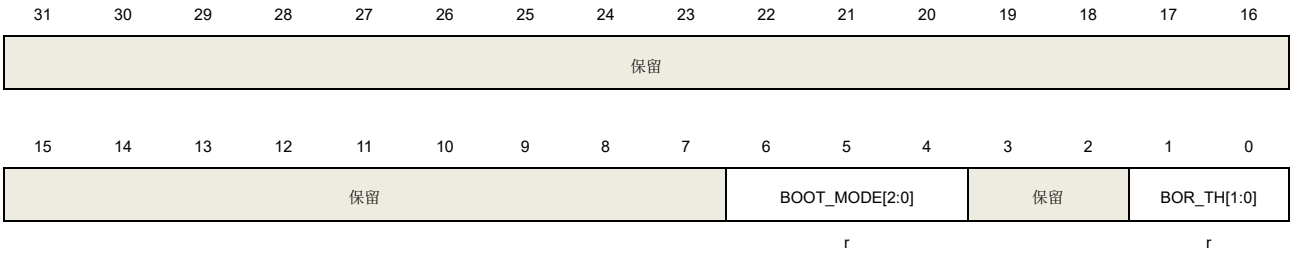
位/位域	名称	描述
31:21	保留	必须保持复位值。
20:16	TSCFG15[4:0]	内部触发输入源配置 00000：保留 00001：内部触发输入 0（ITI0） 00010：内部触发输入 1（ITI1） 00011：内部触发输入 2（ITI2） 00100：内部触发输入 3（ITI3） 00101：CI0 的边沿标志位（CI0F_ED） 00110：保留 00111：保留 01000：保留 01001：保留 01010：保留 10010：保留 10011：内部触发输入 14（ITI14） 其他：保留 **注意：**使用 TSCFG15[4:0]时，需保证 TSCFGy[4:0]（y=0..7）为零，否则 ITS 触发源需与 TSCFGy[4:0]选择的通道输入源保持一致。
15:0	保留	必须保持复位值。

1.8.30. 用户配置寄存器（SYSCFG_USERCFG）

地址偏移：0x300

复位值：0x0000 00XX

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	BOOT_MODE[2:0]	引导模式 这些位指示引导模式。 000：BOOT SRAM(ITCM/DTCM/共享 RAM/AXI SRAM)启动 001：BOOT 安全启动 010：BOOT_SYS (BootLoader) 011：BOOT_USER (用户 flash OSPI0/1) 其他：保留
3:2	保留	必须保持复位值。
1:0	BOR_TH[1:0]	欠压复位（BOR）阈值状态位 00：无 BOR 功能 01：BOR 阈值 1 10：BOR 阈值 2 11：BOR 阈值 3

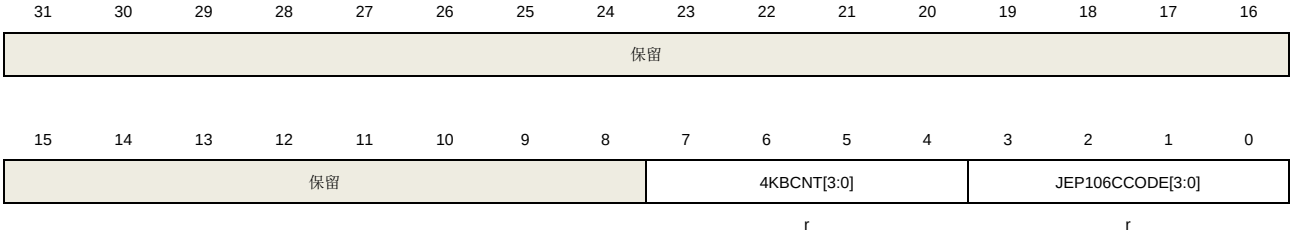
1.9. AXI 互联寄存器

AXI互联基地址：0x5100 0000

1.9.1. AXI 外设 ID4 寄存器（AXI_PERIPH_ID4）

地址偏移：0x1FD0
复位值：0x0000 0004

该寄存器只能按字（32位）访问。

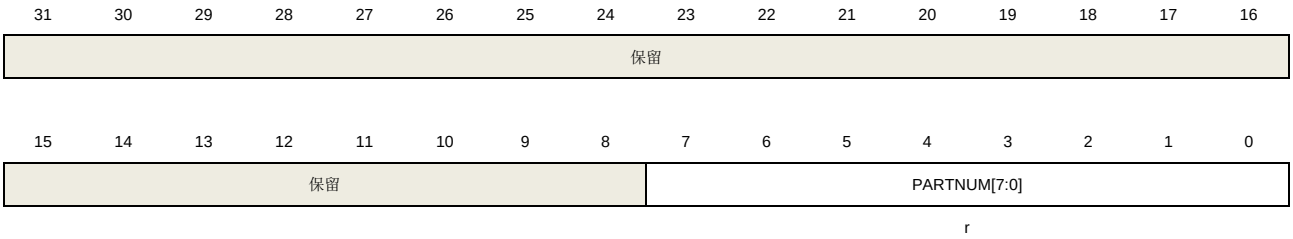


位/位域	名称	描述
31:8	保留	必须保持复位值。
7:4	4KBCNT[3:0]	4KB 大小
3:0	JEP106CCODE[3:0]	JEP106 连续代码

1.9.2. AXI 外设 ID0 寄存器（AXI_PERIPH_ID0）

地址偏移：0x1FE0
复位值：0x0000 0000

该寄存器只能按字（32位）访问。



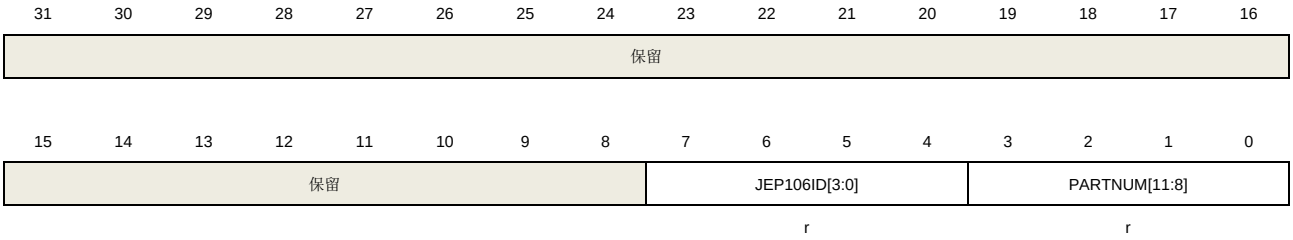
位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	PARTNUM[7:0]	产品编号[7:0]

1.9.3. AXI 外设 ID1 寄存器（AXI_PERIPH_ID1）

地址偏移：0x1FE4

复位值：0x0000 00B4

该寄存器只能按字（32位）访问。



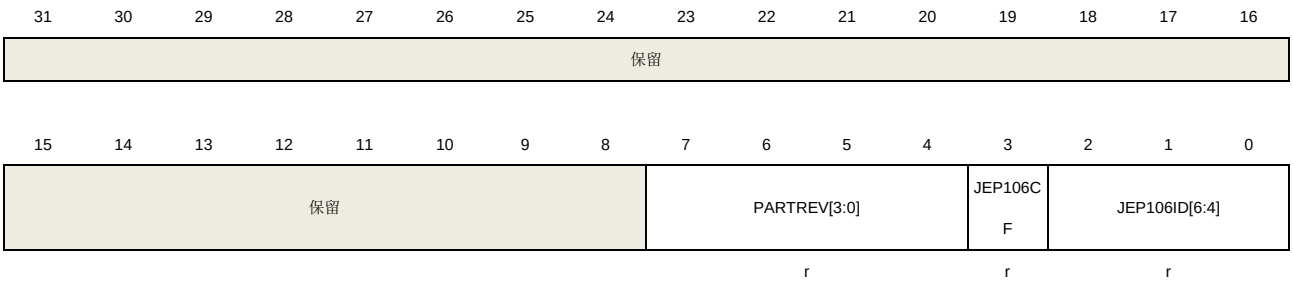
位/位域	名称	描述
31:8	保留	必须保持复位值。
7:4	JEP106ID[3:0]	JEP106 识别位[3:0]
3:0	PARTNUM[11:8]	产品编号[11:8]

1.9.4. AXI 外设 ID2 寄存器（AXI_PERIPH_ID2）

地址偏移：0x1FE8

复位值：0x0000 002B

该寄存器只能按字（32位）访问。

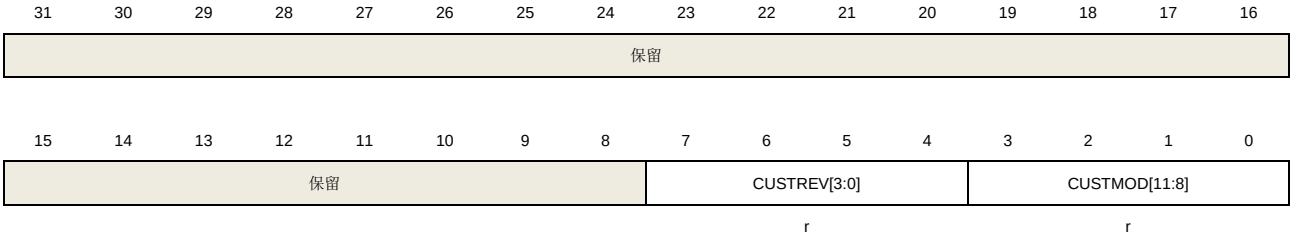


位/位域	名称	描述
31:8	保留	必须保持复位值。
7:4	PARTREV[3:0]	版本编号
3	JEP106CF	JEP106 代码标志
2:0	JEP106ID[6:4]	产品编号[6:4]

1.9.5. AXI 外设寄存器 ID3 寄存器（AXI_PERIPH_ID3）

地址偏移：0x1FEC
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

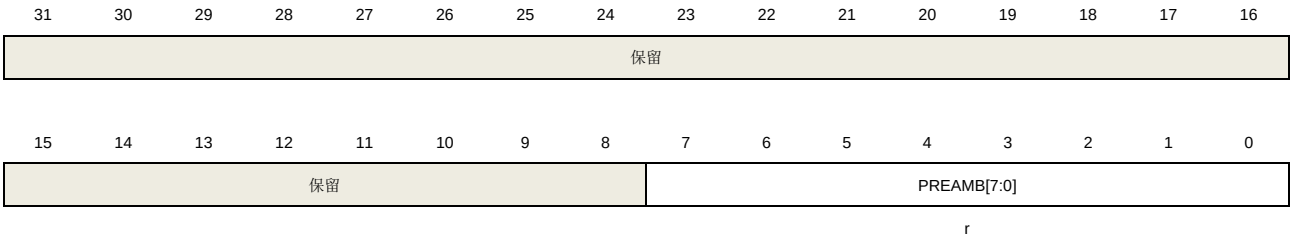


位/位域	名称	描述
31:8	保留	必须保持复位值。
7:4	CUSTREV[3:0]	客户版本
3:0	CUSTMOD[11:8]	客户修改

1.9.6. AXI 组件 ID0 寄存器（AXI_COMP_ID0）

地址偏移：0x1FF0
复位值：0x0000 000D

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	PREAMB[7:0]	报头位[7:0]

1.9.7. AXI 组件 ID1 寄存器（AXI_COMP_ID1）

地址偏移：0x1FF4
复位值：0x0000 00F0

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	CLASS[3:0]	组件类
3:0	PARTNUM[11:8]	报头位[11:8]

1.9.8. AXI 组件 ID2 寄存器（AXI_COMP_ID2）

地址偏移：0x1FF8
复位值：0x0000 0005

该寄存器只能按字（32位）访问。

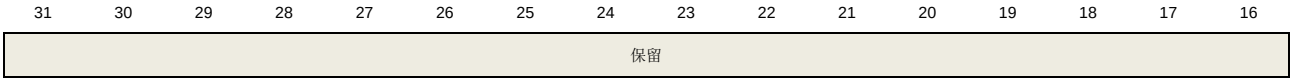


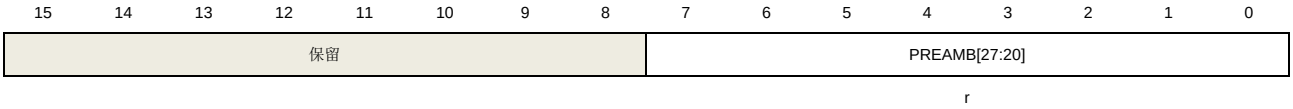
位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	PREAMB[19:12]	报头位[19:12]

1.9.9. AXI 组件 ID3 寄存器（AXI_COMP_ID3）

地址偏移：0x1FFC
复位值：0x0000 00B1

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	PREAMB[27:20]	报头位[27:20]

1.9.10. AXI 主端口 x 总线矩阵发布功能控制寄存器（AXI_MPxBM_ISS_CTL）

地址偏移：0x2008 + 0x1000 * x, x = 0 to 7
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

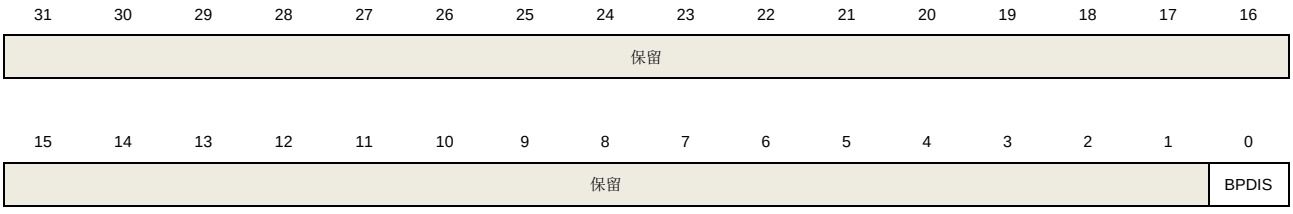


位/位域	名称	描述
31:2	保留	必须保持复位值。
1	WR_ISSOV	覆盖目标写发布函数 0：正常发布功能 1：将总线矩阵写发布能力置为 1
0	RD_ISSOV	覆盖目标读发布函数 0：正常发布功能 1：将总线矩阵读发布能力置为 1

1.9.11. AXI 主端口 x 总线矩阵功能控制寄存器（AXI_MPxBM_CTL）

地址偏移：0x2024 + 0x1000 * x, x = 0, 1, 6和7
复位值：0x0000 0000

该寄存器只能按字（32位）访问。



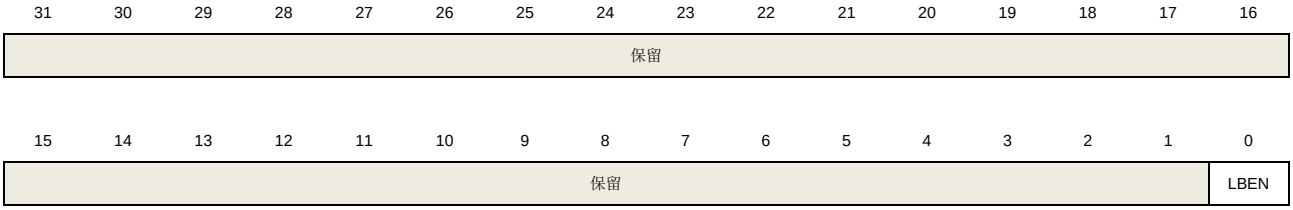
rw

位/位域	名称	描述
31:1	保留	必须保持复位值。
0	BPDIS	禁能节拍封装配置 0：正常工作 1：禁能节拍封装功能

1.9.12. AXI 主端口 x 长突发功能控制寄存器（AXI_MPx_LB_CTL）

地址偏移：0x202C + 0x1000 * x, x = 0和1
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
31:1	保留	必须保持复位值。
0	LBEN	控制长突发功能 0：禁能 ASIB 输出长突发 1：使能 ASIB 输出长突发

1.9.13. AXI 主端口 x 发布功能控制寄存器（AXI_MPx_ISS_CTL）

地址偏移：0x2108 + 0x1000 * x, x = 0, 1, 2, 7
复位值：0x0000 0000

该寄存器只能按字（32位）访问。



rw rw

位/位域	名称	描述
31:2	保留	必须保持复位值。

1	WR_ISSOV	覆盖 AMIB 写发布功能 0：正常发布功能 1：将 AMIB 写发布能力强制置为 1
0	RD_ISSOV	覆盖 AMIB 读发布功能 0：正常发布功能 1：将 AMIB 发布能力强制置为 1

1.9.14. AXI 从端口 x 功能控制寄存器（AXI_SPx_CTL）

地址偏移：0x42024 + 0x1000 * x, x = 0和2
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	TRANSALT	事务更改配置 0：正常操作 1：在允许的情况下事务保持不变

1.9.15. AXI 从端口 x AHB 发布功能控制寄存器（AXI_SPx_AHBISS_CTL）

地址偏移：0x42028 + 0x1000 * x, x = 0和2
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	RD_AHB_ISSOV	将 AHB Lite 读事务转换为单拍 AXI 事务函数 0：禁能覆盖 1：使能覆盖
0	WR_AHB_ISSOV	将 AHB Lite 写事务转换为单拍 AXI 事务函数 0：禁能覆盖 1：使能覆盖

1.9.16. AXI 从端口 x 读 QOS 控制寄存器（AXI_SPx_RDQOS_CTL）

地址偏移：0x42100 + 0x1000 * x, x = 0 to 5
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	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												RDQOS[3:0]			
												rw			

位/位域	名称	描述
31:4	保留	必须保持复位值。
3:0	RDQOS[3:0]	读通道 QoS 配置 0000：最低优先级 … 1111：最高优先级

1.9.17. AXI 从端口 x 写 QOS 控制寄存器（AXI_SPx_WRQOS_CTL）

地址偏移：0x42104 + 0x1000 * x, x = 0 to 5
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	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												WRQOS[3:0]			
												rw			

位/位域	名称	描述
31:4	保留	必须保持复位值。
3:0	WRQOS[3:0]	写通道 QoS 配置

0000：最低优先级

…

1111：最高优先级

1.9.18. AXI 从端口 x 发布功能控制寄存器（AXI_SPx_ISS_CTL）

地址偏移：0x42108 + 0x1000 * x, x = 0 to 5

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	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														WR_ISSOV	RD_ISSOV
														rw	rw

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	WR_ISSOV	覆盖 ASIB 写发布函数 0：正常发布功能 1：将 ASIB 写发布能力强制置为 1
0	RD_ISSOV	覆盖 ASIB 读发布函数 0：正常发布功能 1：将总线矩阵写发布能力强制置为 1

1.10. 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。

1.10.1. 存储容量信息

基地址：0x1FF0 F7E0

该值是原厂设定的，不能由用户修改。

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

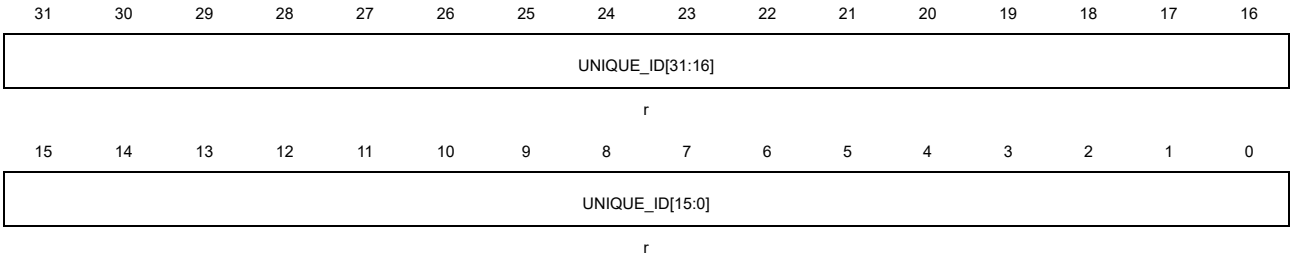
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRAM_DENSITY[15:0]															

r

位/位域	名称	描述
31:16	FLASH_DENSITY [15:0]	Flash存储器容量 该值表明芯片的片上FLASH存储器容量，以Kbytes为单位。 例如：0x0020表示32 Kbytes。
15:0	SRAM_DENSITY [15:0]	SRAM存储器容量 该值表明芯片的片上SRAM容量，以Kbytes为单位。 例如：0x0008表示8 Kbytes。

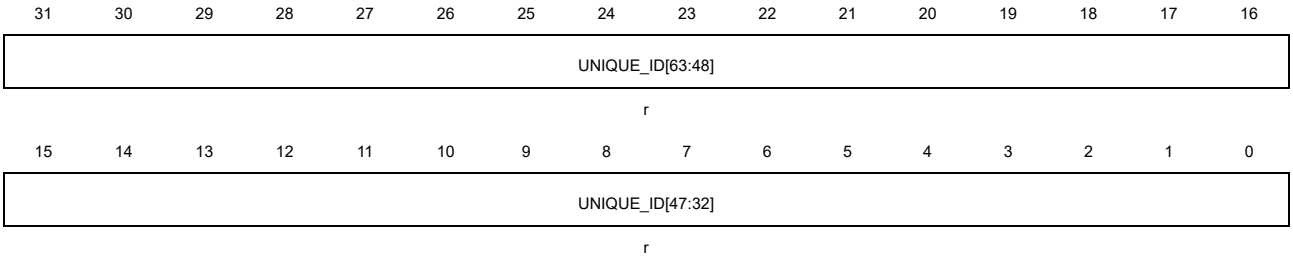
1.10.2. 设备唯一 ID（96 位）

基地址：0x1FF0 F7E8
该值是原厂设定的，不能由用户修改。



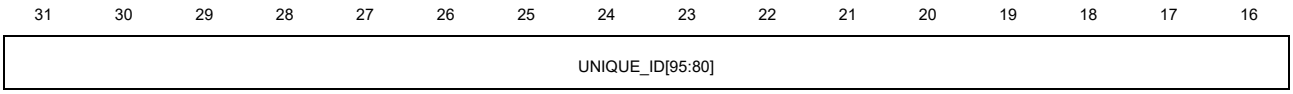
位/位域	名称	描述
31:0	UNIQUE_ID[31:0]	设备唯一ID

基地址：0x1FF0 F7EC
该值是原厂设定的，不能由用户修改。



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

基地址：0x1FF0 F7F0
该值是原厂设定的，不能由用户修改。



r

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

r

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

2. RAM ECC 监视器单元（RAMECCMU）

GD32H73x_75x 设备在 Region 0 和 Region 1 中分别具有两个 RAM ECC 监视器单元（RAMECCMU）。它提供了一种方法来验证应用程序的 ECC 状态，并在发生错误时执行错误处理。

2.1. 主要特性

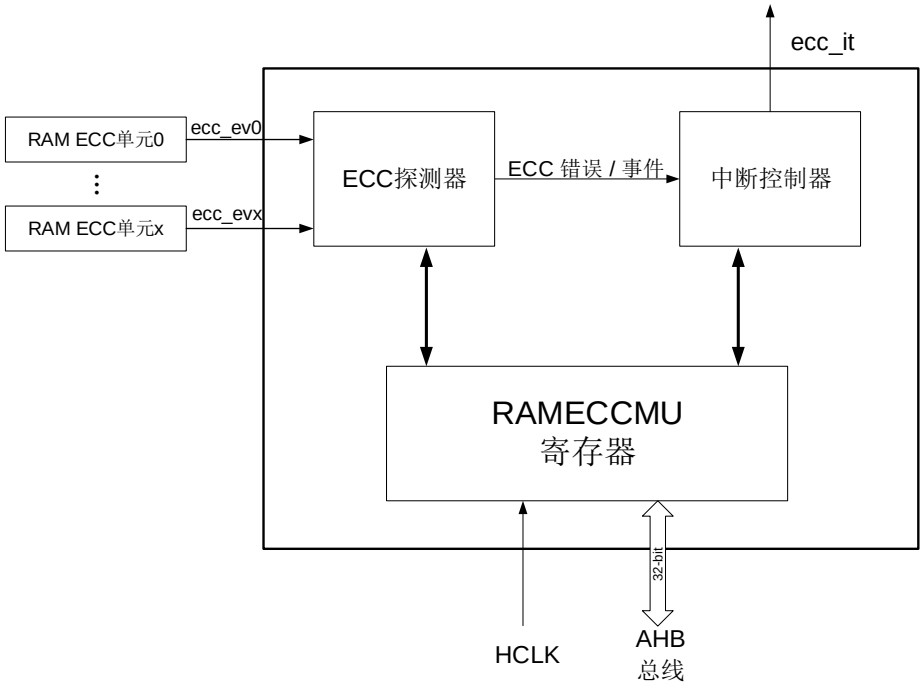
RAMECCMU 的主要特性如下：

- 每个Region都有RAM ECC监视器
- RAM故障地址/数据识别

2.2. 功能描述

GD32H73x_75x 具有两个 RAMECC 监视器单元，分别安装在 Region 0 的 AHB3 和 Region 1 的 AHB2 上。RAMECCMU 的块架构如*图 2-1. RAMECCMU 架构图*所示。

图 2-1. RAMECCMU 架构图



GD32H73x_75x 系列的两个 RAMECC 监视器单元的描述如*表 2-1. Region 0 的 RAMECC 监视器单元 x（x=0..4）*和*表 2-2. Region 1 的 RAMECC 监视器单元 x（x=0..2）*所示。

表 2-1. Region 0 的 RAMECC 监视器单元 x（x=0..4）

RAMECC监视器单元编号	RAMECC监视器状态
0	AXI SRAM ECC
1	ITCM-RAM ECC

RAMECC监视器单元编号	RAMECC监视器状态
2	DTCM-RAM ECC(D0TCM)
3	DTCM-RAM ECC(D1TCM)
4	RAM(ITCM/DTCM/AXI SRAM) ECC

表 2-2. Region 1 的 RAMECC 监视器单元 x（x=0..2）

RAMECC监视器单元编号	RAMECC监视器状态
0	SRAM0 ECC
1	SRAM1 ECC
2	Backup RAM(BKPSRAM) ECC

2.3. RAMECCMU 寄存器

RAMECCMU Region 0基地址：0x5200 9000
RAMECCMU Region 1基地址：0x4802 3000

2.3.1. RAMECCMU 全局中断寄存器（RAMECCMU_INT）

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												GEDERRBWIE	GEDERRIE	GESERRIE	GEIE
												rw	rw	rw	rw

位/位域	名称	描述
31:4	保留	必须保持复位值。
3	GEDERRBWIE	全局 ECC 双比特错误字节写中断使能 0：无中断产生 1：对 RAM 进行字节写操作期间发生 ECC 双比特错误检测时产生中断
2	GEDERRIE	全局 ECC 双比特错误中断使能 0：无中断产生 1：从 RAM 读操作期间发生 ECC 双比特错误检测时产生中断
1	GESERRIE	全局 ECC 单比特错误中断使能 0：无中断产生 1：从 RAM 读操作期间发生 ECC 单比特错误时产生中断
0	GEIE	全局 ECC 中断使能 0：无中断产生 1：发生 GEDERRBWIE、GEDERRIE 或 GESERRIE 错误之一时产生中断

2.3.2. RAMECCMU 监视器 x 控制寄存器（RAMECCMU_MxCTL）

地址偏移：0x20 * (x+1), (x是ECC监视器编号，对于Region 0，x=0..4，而对于Region 1,x=0..2)
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留										ECCERRLATEN	ECCDERRBWIE	ECCDERRIE	ECCSERRIE	保留	
										rw	rw	rw	rw		

位/位域	名称	描述
31:6	保留	必须保持复位值。
5	ECCERRLATEN	ECC 错误锁存使能 0：当 ECC 错误发生时，没有错误上下文被锁存到各自的寄存器中 1：当 ECC 错误发生时，错误上下文被锁存到各自的寄存器中
4	ECCDERRBWIE	ECC 双比特错误字节写中断使能 0：无中断产生 1：对 RAM 的字节写操作发生 ECC 双比特错误时产生中断
3	ECCDERRIE	ECC 双比特错误中断使能 0：无中断产生 1：从 RAM 读操作发生 ECC 双比特错误时产生中断
2	ECCSERRIE	ECC 单比特错误中断使能 0：无中断产生 1：从 RAM 读操作时发生 ECC 单比特错误时产生中断
1:0	保留	必须保持复位值。

2.3.3. RAMECCMU 监视器 x 状态寄存器（RAMECCMU_MxSTAT）

地址偏移：0x24 + 0x20 * x, (x是ECC监视器编号，对于Region 0，x=0..4，而对于Region 1,x=0..2)
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													ECCDERRBWDF	ECCDERRDF	ECCSERRDCF
													rc_w0	rc_w0	rc_w0

位/位域	名称	描述
31:3	保留	必须保持复位值。
2	ECCDERRBWDF	字节写入时 ECC 双比特错误检测标志 该位由硬件置 1，软件写 0 清除。 0：当 ECCDERRDF 为 1 时，表示在读时检测到双比特错误

		1：在非对齐写时检测到双比特错误
1	ECCDERRDF	ECC 双比特错误检测标志 该位由硬件置 1，软件写 0 清除。 0：未检测到错误 1：检测到错误
0	ECCSERRDCF	ECC 单比特错误检测和纠正标志 该位由硬件置 1，软件写 0 清除。 0：无错误检测和纠正 1：错误被检测和纠正

2.3.4. RAMECCMU 监视器 x 故障地址寄存器（RAMECCMU_MxFADDR）

地址偏移：0x28 + 0x20 * x, (x是ECC监视器编号，对于Region 0，x=0..4，而对于Region 1,x=0..2)

复位值：0x2400 0000（AXI SRAM）
0x0000 0000（ITCM）
0x2000 0000（D0TCM）
0x2000 0004（D1TCM）
0x2408 0000（ITCM/DTCM/AXI SRAM 共享 RAM）
0x3000 0000（SRAM0）
0x3000 4000（SRAM1）
0x3880 0000（BKPSRAM）

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:0	ECCFADDR[31:0]	ECC 错误故障地址 该寄存器包含错误发生时 ECC 错误生成的地址。

2.3.5. RAMECCMU 监视器 x 故障数据低位寄存器（RAMECCMU_MxFDL）

地址偏移：0x2C + 0x20 * x, (x是ECC监视器编号，对于Region 0，x=0..4，而对于Region 1,x=0..2)

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:0	ECCFDL[31:0]	ECC 故障数据低位 该寄存器包含发生错误时由 ECC 错误产生的数据的 LSB 或 32 位 SRAM 的完整存储器字内容。

2.3.6. RAMECCMU 监视器 x 故障数据高位寄存器（RAMECCMU_MxFDH）

地址偏移：0x30 + 0x20 * x, (x是ECC监视器编号，对于Region 0，x=0..4，而对于Region 1,x=0..2)
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:0	ECCFDH[31:0]	ECC 故障数据高位（64-bit） 该寄存器包含发生错误时由 ECC 错误产生的数据的 MSB。

2.3.7. RAMECCMU 监视器 x 故障 ECC 错误代码寄存器（RAMECCMU_MxFECODE）

地址偏移：0x34 + 0x20 * x, (x是ECC监视器编号，对于Region 0，x=0..4，而对于Region 1,x=0..2)
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:0	ECCFECODE[31:0]	ECC 故障错误代码 该寄存器包含发生位错误的索引和 ECC 代码。

3. 闪存控制器（FMC）

3.1. 简介

闪存控制器（FMC），提供了片上闪存需要的所有功能。包括扇区擦除，整片擦除，以及编程操作等。

3.2. 主要特性

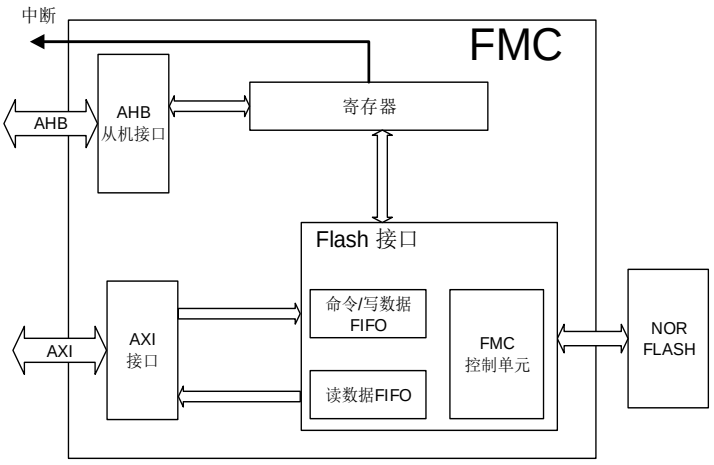
- 高达3840KB字节的片上闪存可用于存储指令或数据；
- 支持64位双字、32位整字、16位半字或字节读操作；
- 支持64位双字、32位整字编程，扇区擦除和整片擦除操作；
- 选项字节会在每次系统复位时装载到选项字节控制寄存器；
- 具有安全保护状态，可阻止对代码或数据的非法读访问；
- 具有擦除和编程保护状态，可阻止意外写操作；
- 具有仅执行的专用代码读保护（DCRP）区域；
- 具有仅能在安全模式下访问的安全用户区域；

3.3. 功能说明

3.3.1. 闪存结构

FMC 支持用以访问代码或数据的 64 位 AXI 接口以及用以访问寄存器的 32 位 AHB 从机接口。*图 3-1. FMC 框图*显示了 FMC 的架构。

图 3-1. FMC 框图



FMC的AXI接口可以同时支持读写操作。FMC的AXI从机接口支持以下访问类型：

- 支持数据宽度为1、2、4、8字节的单一读操作；
- 支持数据宽度为1、2、4、8字节的增量突发读操作，突发传输数据长度最多可达128字节。

- 支持数据宽度为8字节的单一、2拍、4拍、8拍的回卷突发读操作；
- 支持数据宽度为4字节、8字节的单一编程操作；
- 支持数据宽度为8字节的最多32拍的突发编程操作，且4拍的突发编程操作数据传输效率最高；
- 使用数据宽度为8字节的非单拍的突发编程操作时，则只能使用MDMA而不能使用DMA，否则会产生不可预期的错误；
- 所有编程地址必须与数据宽度对齐；
- 在一次突发读/写操作中，地址不能跨越4K边界；
- 以上描述以外的其他所有操作将产生总线错误，且读操作将返回零，编程操作将被忽略。

AHB从机接口支持32位整字访问。

闪存包括3840KB字节主闪存，分为960个扇区，扇区大小为4KB，和64KB用于引导加载程序的信息块。主存储闪存的每个扇区都可以单独擦除。

闪存结构细节见*表3-1. GD32H73x_75x闪存基地址和构成*。

表 3-1. GD32H73x_75x 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	扇区0	0x0800 0000 - 0x0800 0FFF	4KB
	扇区1	0x0800 1000 - 0x0800 1FFF	4KB
	扇区2	0x0800 2000 - 0x0800 2FFF	4KB
	.	.	.
	.	.	.
	.	.	.
	扇区959	0x083B F000 - 0x083B FFFF	4KB
信息块	引导装载程序	0x1FF0 0000 - 0x1FF0 FFFF	64KB

3.3.2. 读操作

闪存可以像普通存储空间一样直接寻址访问。对闪存取指令和取数据是通过AXI接口访问的。

FMC 内部 RTDEC 功能

FMC内部RTDEC功能是指从闪存中读取数据时，可以根据EFUSE模块中配置的AES秘钥进行实时解密数据（写入闪存的数据已经加密）。当EFUSE_USER_CTL寄存器中的AESEN位置1时，开启即时解密功能。这是通过硬件即时实现的，不能通过软件实现。AES秘钥由EFUSE_AES_KEY寄存器来设置。初始向量AES_IV[127:0] = AESIV[95:0] || 12'b0 || 读地址[23:4]。其中，用户可以通过FMC_AESIVx_MDF寄存器来设置初始向量中的高96位（即AESIV[95:0]）。且96位的AESIV[95:0]按照AESIV2 || AESIV1 || AESIV0的顺序组成。当需要修改初始向量时，用户需要依次修改FMC_ASIV0_MDF、FMC_ASIV1_MDF、FMC_ASIV2_MDF寄存器，在写入FMC_AESIV2_MDF寄存器后，FMC_ASIV0/1/2_MDF寄存器中的值都将被更新至AES初始向量区域中。

修改AESIV[95:0]的操作步骤如下：

1. 确保FMC_CTL寄存器不处于锁定状态；

2. 等待FMC_STAT寄存器的BUSY位变为0来确保没有闪存操作在进行，否则等待该操作完成；
3. 依次将初始向量AES_IV[32:63]写入FMC_AESIV0_MDF，将AES_IV[64:95]写入FMC_ASIV1_MDF，将AES_IV[96:127]写入FMC_AESIV2_MDF寄存器；
4. 一旦FMC_AESIV2_MDF寄存器被写入后，FMC_AESIV0/1/2_MDF寄存器中的值都将自动被更新至AES初始向量区域中；
5. 通过检查FMC_STAT寄存器的BUSY位是否清0，来确定编程指令执行完毕；
6. 启动一次系统复位以加载初始向量，使之生效；
7. 如有需要，读取FMC_AESIVx_EFT中的值验证是否修改成功。

当修改完成后，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。

NO-RTDEC 功能

即使将EFUSE_USER_CTL寄存器中AESEN位置1，也可以通过FMC_NODEC寄存器配置区域不使用RTDEC功能。

3.3.3. 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，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。

3.3.4. 扇区擦除

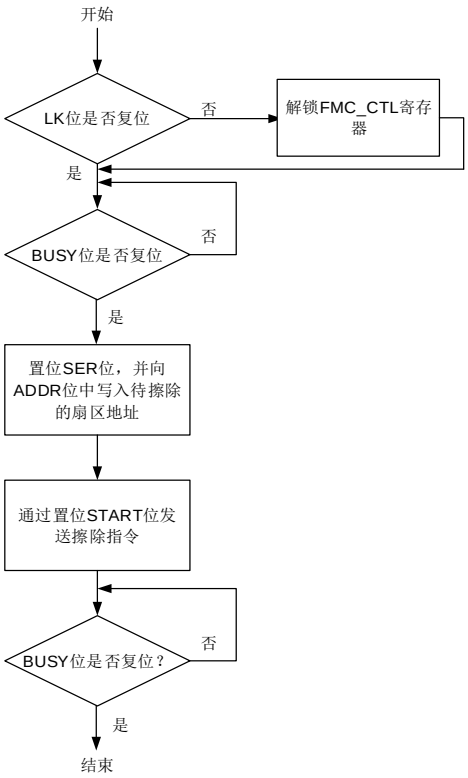
FMC的扇区擦除功能使得主存储闪存的扇区内容初始化为高电平。每一扇区都可以被独立擦除，而不影响其他扇区内容。FMC扇区擦除操作步骤如下：

1. 确保FMC_CTL寄存器不处于锁定状态；
2. 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；
3. 置位FMC_CTL寄存器的SER位；
4. 将待擦除扇区的绝对地址（0x08XX XXXX）写到 FMC_ADDR 寄存器；
5. 通过将FMC_CTL寄存器的START位置1来发送扇区擦除命令到FMC
6. 通过检查FMC_STAT寄存器的BUSY位是否清0，来确定擦除指令执行完毕；
7. 如果需要，使用读操作验证该扇区是否擦除成功。

如果对包含了DCRP、安全用户区域、擦除/编程保护区域的扇区进行扇区擦除操作，该操作将会被中止，且FMC_STAT寄存器的WPERR位将置位。如果同时请求了整片擦除和扇区擦除，整片擦除将会替代扇区擦除操作。

当扇区擦除成功执行，FMC_STAT寄存器的ENDF位将置位。若FMC_CTL寄存器的ENDIE位被置1，则FMC将触发一个中断。需要注意的是，用户需确保写入的是正确的擦除目标扇区地址。否则当待擦除目标扇区被用来取指令或访问数据时，软件将会跑飞。该情况下，FMC不会提供任何出错通知。另一方面，对擦除/编程保护的扇区进行扇区擦除操作将无效。如果FMC_CTL寄存器的WPERRIE位被置位，该操作将触发擦除/编程保护错误中断。中断服务程序可通过检测FMC_STAT寄存器的WPERR位来判断该中断是否发生。*图3-2. 扇区擦除操作流程*显示了扇区擦除操作流程。

图 3-2. 扇区擦除操作流程



注意：在编程、擦除尤其是全片擦除时，应尽量避免异常掉电或复位，否则可能会产生不可预知的后果。如果有掉电和复位风险，应避免使用带清除保护的整片擦除。

注意：在修改选项字节时，严禁掉电或复位，否则会损坏闪存数据，产生不可预知的后果，可能会让芯片强制进入安全模式。

注意：在编程和擦除主存储闪存块中的数据时，避免中途发生掉电或复位，否则可能会损坏当前扇区的数据，在下次上电后，需要擦除损坏的扇区之后才能继续使用。

3.3.5. 整片擦除

标准整片擦除

FMC提供了标准整片擦除功能，可以初始化主存储闪存块中，除包含安全或保护区域的扇区以外的所有其他扇区的内容。FMC标准整片擦除操作步骤如下：

1. 确保FMC_CTL寄存器不处于锁定状态；
2. 等待FMC_STAT寄存器的BUSY位变为0来确保没有闪存操作在进行，否则等待该操作完

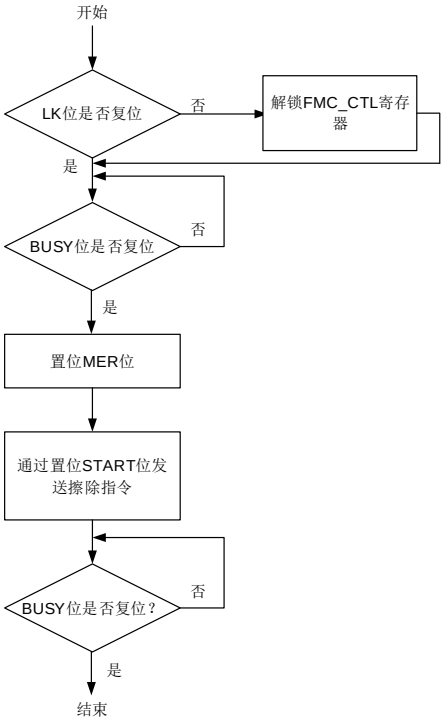
成；

3. 置位FMC_CTL寄存器的MER位；
4. 通过将FMC_CTL寄存器的START位置1来发送扇区擦除命令到FMC
5. 通过检查FMC_STAT寄存器的BUSY位是否清0，来确定擦除指令执行完毕；
6. 如果需要，使用读操作验证是否擦除成功。

如果同时请求整片擦除和扇区擦除，则整片擦除将替代扇区擦除操作。

当标准整片擦除成功执行，FMC_STAT寄存器的ENDF位置位。若FMC_CTL寄存器的ENDIE位被置1，FMC将触发一个中断。由于除安全访问或被保护的区域外的其他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寄存器来实现标准整片擦除操作。*图3-3. 标准整片擦除操作流程*显示了标准整片擦除操作流程。

图 3-3. 标准整片擦除操作流程



注意：在编程、擦除尤其是全片擦除时，应尽量避免异常掉电或复位，否则可能会产生不可预知的后果。如果有掉电和复位风险，应避免使用带清除保护的整片擦除。

注意：在修改选项字节时，严禁掉电或复位，否则会损坏闪存数据，产生不可预知的后果，可能会让芯片强制进入安全模式。

注意：在编程和擦除主存储闪存块中的数据时，避免中途发生掉电或复位，否则可能会损坏当前扇区的数据，在下次上电后，需要擦除损坏的扇区之后才能继续使用。

带清除保护的整片擦除

FMC提供了带清除保护的整片擦除功能，可以初始化主存储闪存块中的所有内容（包括那些包含了安全或保护区域的扇区）。FMC带清除保护的整片擦除操作步骤如下：

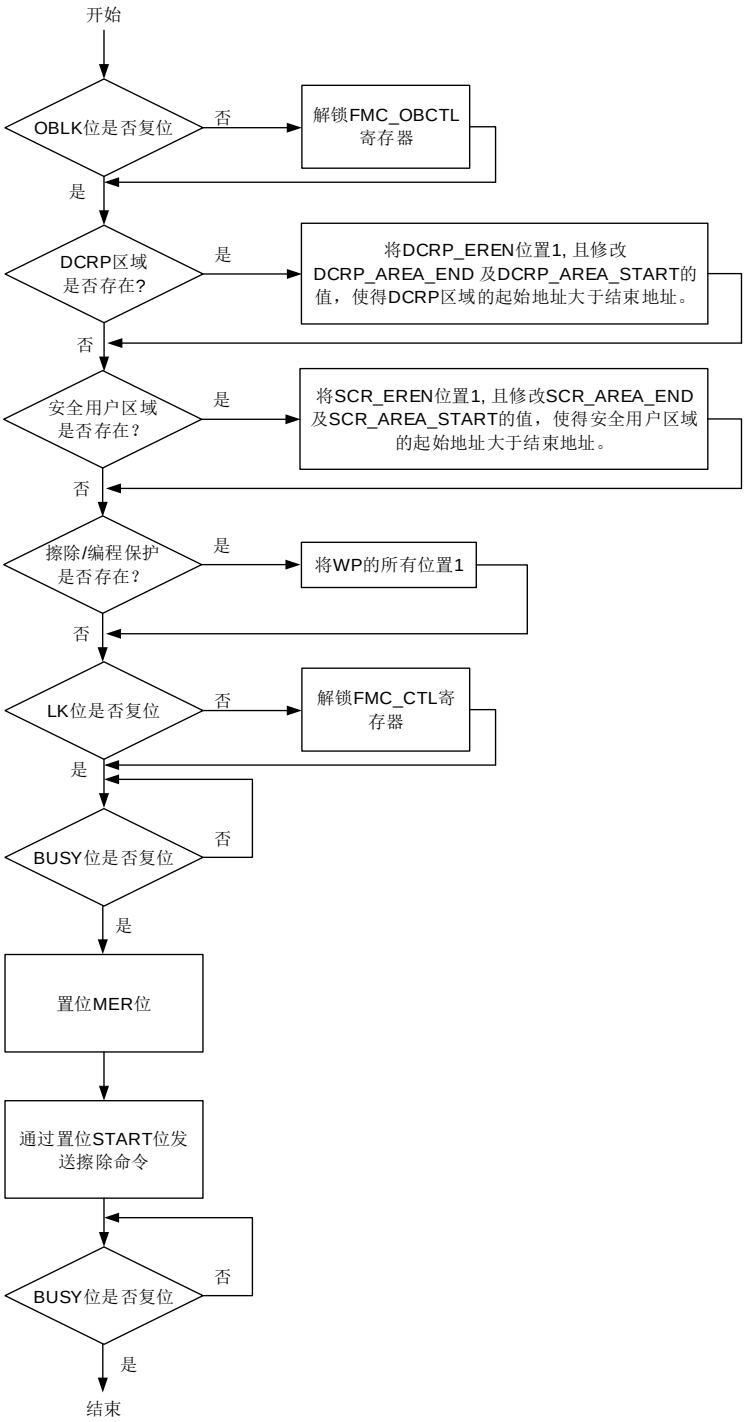
1. 确保FMC_OBCTL寄存器不处于锁定状态；
2. 如果存在DCRP保护区，将FMC_DCRPADDR_MDF或FMC_DCRPADDR_EFT寄存器中的DCRP_EREN置位。并将满足DCRP_AREA_END < DCRP_AREA_START的值写到FMC_DCRPADDR_MDF寄存器中，从而使DCRP结束地址小于DCRP起始地址；
3. 如果存在安全访问保护区，将FMC_SCRADDR_MDF或FMC_SCRADDR_EFT寄存器中的SCR_EREN置位。并将满足SCR_AREA_END < SCR_AREA_START的值写到FMC_SCRADDR_MDF寄存器中，从而使安全用户区域结束地址小于安全用户区域起始地址；
4. 将FMC_WP_MDF寄存器中的所有WP位都置1，从而失能所有扇区的擦除/编程保护功能；
5. 确保FMC_CTL寄存器不处于锁定状态；
6. 等待FMC_STAT寄存器的BUSY位变为0来确保没有闪存操作在进行，否则等待该操作完成；
7. 置位FMC_CTL寄存器的MER位；
8. 通过将FMC_CTL寄存器的START位置1来发送扇区擦除命令到FMC；
9. 通过检查FMC_STAT寄存器的BUSY位是否清0，来确定擦除指令执行完毕；此时，带清除保护的整片擦除操作将擦除了整个主存储闪存区，包括包含了DCRP及安全访问数据的扇区，且选项字节的修改将自动执行，从而禁用所有保护。
10. 如果需要，使用读操作验证是否擦除成功。

注意：（1）以上步骤中，除上述提到的选项字节外，其他选项字节请勿修改。（2）只有步骤2、3、4中的条件都满足，才会执行带清除保护的整片擦除。如果有任一步骤不满足，都会只执行标准整片擦除，且不会报错。

如果同时请求整片擦除和扇区擦除，则整片擦除将替代扇区擦除操作。

当带清除保护的整片擦除成功执行，FMC_STAT寄存器的ENDF位置位。若FMC_CTL寄存器的ENDIE位被置1，FMC将触发一个中断。由于所有的闪存数据（包括那些安全区域、保护区域）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寄存器来实现带清除保护的整片擦除操作。*图3-4. 带清除保护的整片擦除*显示了标准整片擦除操作流程。

图 3-4. 带清除保护的整片擦除



注意：在编程、擦除尤其是全片擦除时，应尽量避免异常掉电或复位，否则可能会产生不可预知的后果。如果有掉电和复位风险，应避免使用带清除保护的整片擦除。

注意：在修改选项字节时，严禁掉电或复位，否则会损坏闪存数据，产生不可预知的后果，可能会让芯片强制进入安全模式。

注意：在编程和擦除主存储闪存块中的数据时，避免中途发生掉电或复位，否则可能会损坏当前扇区的数据，在下次上电后，需要擦除损坏的扇区之后才能继续使用。

3.3.6. 主存储闪存块编程

FMC通过AXI接口提供了一个64位双字/32位整字编程功能，用来修改主存储闪存块内容。FMC闪存编程操作步骤如下：

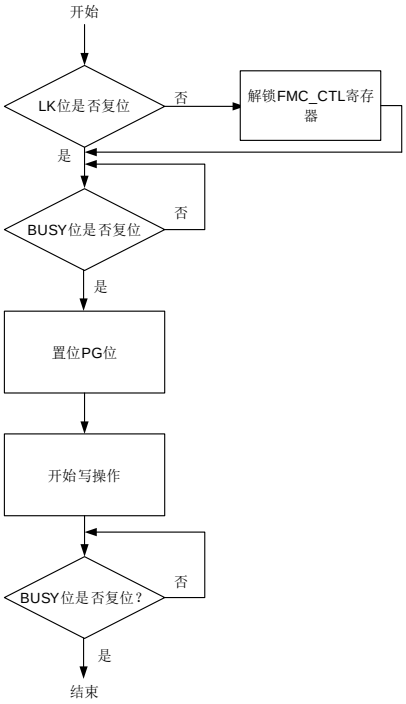
1. 确保FMC_CTL寄存器不处于锁定状态；
2. 等待FMC_STAT寄存器的BUSY位变为0来确保没有闪存操作在进行，否则等待该操作完成；
3. 置位FMC_CTL寄存器的PG位；
4. 将要写的数据写到目的绝对地址（0x08XX XXXX）；
5. 通过检查FMC_STAT寄存器的BUSY位是否清0，来确定编程指令执行完毕；
6. 如果需要，使用读操作验证是否编程成功。

当主存储块编程成功执行，FMC_STAT寄存器的ENDF位置位。若FMC_CTL寄存器的ENDIE位被置1，FMC将触发一个中断。需要注意的是，PG位必须在64位双字/32位整字编程开始前进行置位，否则FMC_STAT寄存器中的PGSERR位会被置位，若FMC_CTL寄存器的PGSERRIE位被置1，FMC将触发一个中断。此外，向被保护擦除/编程扇区进行的编程操作会被忽略，同时FMC_STAT寄存器中的WPERR位被置位，若FMC_CTL寄存器的WPERRIE位被置1，FMC将触发一个中断。在中断服务程序中，可以检查FMC_STAT寄存器的PGSERR位和WPERR位来判断哪一种错误发生了。*图3-5. 编程操作流程*显示了编程操作流程。

用户可以通过在编程操作前，将在FMC_CTL寄存器中的PGCHEN位置1，来检查要写进数据的区域是否全部为0xFF。如果该区域并非全部为0xFF，则FMC_STAT寄存器中的PGSERR将被设置。

当突发编程操作时，PGCHEN为1，FMC将根据突发数据宽度来检查闪存数据。如果数据并非全部为0xFF，则该节拍编程失败，并将FMC_STAT寄存器中的PGSERR置位。但其他节拍可以正常编程。

图 3-5. 编程操作流程



注意：在编程、擦除尤其是全片擦除时，应尽量避免异常掉电或复位，否则可能会产生不可预知的后果。如果有掉电和复位风险，应避免使用带清除保护的整片擦除。

注意：在修改选项字节时，严禁掉电或复位，否则会损坏闪存数据，产生不可预知的后果，可能会让芯片强制进入安全模式。

注意：在编程和擦除主存储闪存块中的数据时，避免中途发生掉电或复位，否则可能会损坏当前扇区的数据，在下次上电后，需要擦除损坏的扇区之后才能继续使用。

3.3.7. 选项字节

选项字节说明

FMC提供两组选项字节寄存器：

■ “_EFT”寄存器组（只读）

该寄存器组包含选项字节的生效值。在系统复位或从深度睡眠模式初始唤醒后，该选项字节组的值都会被自动重加载。

■ “_MDF”寄存器组（可读写）

该寄存器组包含选项字节的修改值。当用户想要修改选项字节时，需要往该寄存器组里写入修改值。

每次系统复位后，闪存的选项字节被重加载到“_EFT”寄存器后，选项字节字节生效。选项字节详情见*表3-2. 选项字节*。选项字节最终根据应用程序的要求来配置

表 3-2. 选项字节

名称	出厂值	寄存器映射
[29]: IOSPDOPEN	0x0	FMC_OBSTAT0_EFT / FMC_OBSTAT0_MDF
[24]: DTCM1ECCEN	0x1	
[23]: DTCM0ECCEN	0x1	
[22]: ITCMECCEN	0x1	
[21]: SCR	0x0	
[18]: FWDGSPD_STDBY	0x1	
[17]: FWDGSPD_DPSLP	0x1	
[15:8]: SPC[7:0]	0xAA	
[7]: nRST_STDBY	0x1	
[6]: nRST_DPSLP	0x1	
[4]: nWDG_HW	0x1	
[3:2]: BOR_TH[1:0]	0x0	
[31]: DCRP_EREN	0x0	FMC_DCRPADDR_EFT / FMC_DCRPADDR_MDF
[26:16]: DCRP_AREA_END[10:0]	0x0	
[10:0]: DCRP_AREA_START[10:0]	0x0FF	
[31]: SCR_EREN	0x0	FMC_SCRADDR_EFT / FMC_SCRADDR_MDF
[26:16]: SCR_AREA_END[10:0]	0x0	
[10:0]: SCR_AREA_START[10:0]	0x0FF	
[21:0]: WP[21:0]	0x3FFFFFFF	FMC_WP_EFT / FMC_WP_MDF
[31:16]: BOOT_ADDR1[15:0]	0x1FF0	FMC_BTADDR_EFT / FMC_BTADDR_MDF
[15:0]: BOOT_ADDR0[15:0]	0x0800	
[31:16]: DATA[15:0]	0x0	FMC_OBSTAT1_EFT / FMC_OBSTAT1_MDF
[7:4]: DTCM_SZ_SHRRAM[3:0]	0x8	
[3:0]: ITCM_SZ_SHRRAM[3:0]	0x7	

选项字节修改

修改选项字节的操作步骤如下：

1. 确保FMC_OBCTL寄存器不处于锁定状态；
2. 等待FMC_STAT寄存器的BUSY位变为0来确保没有闪存操作在进行，否则等待该操作完成；
3. 在“_MDF”寄存器组（FMC_XXX_MDF）中，写入想要修改的选项字节寄存器值；
4. 通过将FMC_OBCTL寄存器的OBSTART位置1来发送选项字节编程命令到FMC；
5. 通过检查FMC_STAT寄存器的BUSY位是否清0，来确定编程指令执行完毕；
6. 启动一次系统复位以加载选项字节，使之生效；
7. 如有需要，读取“_EFT”寄存器组（FMC_XXX_EFT）中的值验证是否修改成功。

注意：“XXX”包括OBSTAT0、DCRPADDR、SCRADDR、WP、BTADDR或OBSTAT1。

当FMC_OBCTL寄存器中的OBLK为1时，禁止修改“_MDF”寄存器。当OBSTART位被设置为1时，FMC会将生效值（_EFT）与修改值（_MDF）进行比较，以检查是否需要修改选项字节。

注意：在修改选项字节时，严禁掉电或复位，否则会损坏闪存数据，产生不可预知的后果，可能会让芯片强制进入安全模式。

注意：在编程和擦除主存储闪存块中的数据时，避免中途发生掉电或复位，否则可能会损坏当前扇区的数据，在下次上电后，需要擦除损坏的扇区之后才能继续使用。

选项字节修改规则

一些选项字节修改必须遵守特定的规则。如果不满足以下规则，OBMERR位将置位并停止选项字节的修改操作。如果满足以下规则，FMC开始将修改选项字节并更新_EFT选项字节寄存器的值。

- 安全保护等级（SPC）

当SPC设置为保护等级高时，所有选项字节不允许任何修改。此时，如果用户应用程序试图降低安全保护等级，将会产生选项字节修改错误（OBMERR），并忽略所有修改操作。

- 擦除/编程保护（WP）

该选项字节配置了扇区擦除/编程保护属性。当SPC不是保护等级高时，该选项字节的修改不受任何限制。

- DCRP区域大小（DCRP_AREA_START和DCRP_AREA_END）

增大DCRP区域没有任何限制，但必须包含原先的DCRP区域，即DCRP的起始地址（DCRP_AREA_START）不能变大，结束地址（DCRP_AREA_END）不能变小。

在移除DCRP区域时，必须在（FMC_DCRPADDR_EFT或FMC_DCRPADDR_MDF寄存器中）DCRP_EREN位为1的同时，去请求SPC降级或带清除保护的整片擦除操作。

在减小DCRP区域时，必须在（FMC_DCRPADDR_EFT或FMC_DCRPADDR_MDF寄存器中）DCRP_EREN位为1的同时，去请求SPC降级。

注意：在通过SPC降级去移除或减小DCRP区域时，若（FMC_DCRPADDR_EFT及FMC_DCRPADDR_MDF寄存器中）DCRP_EREN位全为0，将会产生选项字节修改错误（OBMERR）。

- DCRP_EREN

SPC保护等级低到无保护状态的降级或带清除保护的整片擦除时，若该选项字节置1，DCRP区域的内容将被擦除。否则将被保留。

将DCRP_EREN位置1没有任何限制。但只有在将DCRP_EREN位清0的同时，去请求SPC降级或带清除保护的整片擦除操作，清0操作才能完成。否则会产生选项字节修改错误（OBMERR）。

- 安全模式（SCR）

如果不存在有效的DCRP区域或安全用户区域，该位可以自由清0。否则，SCR清0的唯一方式是：在DCRP_EREN位（FMC_DCRPADDR_EFT或FMC_DCRPADDR_MDF寄存器中）为1，且SCR_EREN位（FMC_SCRADDR_EFT或FMC_SCRADDR_MDF寄存器中）为1时，去执行SPC降级。否则，将会产生选项字节修改错误（OBMERR）。

注意：在执行SPC降级以禁用SCR选项字节时，需编程使DCRP区域和安全用户区域的起始地址都大于其结束地址。

- 安全用户区域大小（SCR_AREA_START和SCR_AREA_END）

在用户安全应用程序或设备上运行的GD安全库可以无限制地修改安全用户区域的大小。

对于用户非安全应用程序，在移除安全用户区域时，必须在（FMC_SCRADDR_EFT或FMC_SCRADDR_MDF寄存器中）SCR_EREN位为1的同时，去执行SPC降级或执行带清除保护的整片擦除操作。

注意：在通过SPC降级去移除安全用户区域时，若（FMC_SCRADDR_EFT及FMC_SCRADDR_MDF寄存器中）SCR_EREN位全为0时，去请求SPC降级，将会产生选项字节修改错误（OBMERR）。

- SCR_EREN

SPC降级或带清除保护的整片擦除时，若该选项字节置1，安全用户区域的内容将被擦除。否则将被保留。

SCR_EREN位置1没有任何限制。但只有在将SCR_EREN位清0的同时，去请求SPC降级或带清除保护的整片擦除操作，清0操作才能完成，否则会产生选项字节修改错误（OBMERR）。

3.3.8. 扇区擦除/编程保护

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。注意，当MER位被置位或扇区地址无效时，进行扇区擦除时WPERR仍会被置位。配置选项字节的WP[21:0]某位为0可以单独使能某几扇区的保护功能。

表 3-3. 扇区保护 WP 位

WP位	扇区保护
WP[0]	扇区 0~扇区 15
WP[1]	扇区 16~扇区 31
. . .	. . .
WP[14]	扇区 224~扇区 239
WP[15]	扇区 240~扇区 255
WP[16]	扇区 256~扇区 383
WP[17]	扇区 384~扇区 511
. . .	. . .
WP[20]	扇区 768~扇区 895
WP[21]	扇区 896~扇区 959

注意：对于WP[x] (x=0…15)，1位对应16个扇区，范围为扇区(x*16)~扇区(x*16+15)；

对于WP[x]（x=16…20），1位对应128个扇区，范围为扇区(x*128 - 1792)~扇区(x*128 - 1665)；

对于WP[21]，1位对应64个扇区，范围为扇区896~扇区959；

擦除/编程保护扇区不能被擦除或编程。因此，如果有扇区受擦除/编程保护，则除非执行SPC保护等级低到无保护状态的降级或带清除保护的整片擦除操作，否则无法执行整片擦除。

当SPC为保护等级高时，WP[21:0]选项字节无法更改，否则该选项字节可以自由修改。

注意：DCRP区域和安全用户区域受擦除/编程保护。

3.3.9. 安全保护

FMC提供了一个安全保护功能来阻止非法读取闪存。并且安全保护是全局的，不仅可以保护闪存，还可以保护其他安全区域。其他安全区域包括备份SRAM（BKPSRAM）、RTC备份寄存器和受实时解密（RTDEC）保护的加密区域（RTDEC仅适用于GD32H737_757_759）。安全保护等级划分三等。

无保护状态

当熔丝中的SPC_H及SPC_L位均为0，且将SPC选项字节设置为0xAA，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。其他安全区域也允许被访问。

保护等级低

当熔丝中的SPC_H为0时，只要将SPC_L位设置为1或将SPC选项字节设置为除0xAA或0xCC外的任何值，就能激活低安全保护等级。主存储闪存块仅能被用户代码访问。在调试模式或者从SRAM中启动或者从boot loader模式启动时，这些模式下对主存储块或其他安全区域（如备份SRAM）的操作都被禁止。无论是在调试模式或者从SRAM中启动，还是从bootloader模式启动，如果对主存储块执行一次读操作，将会产生一个读保护错误（RPERR）。在调试模式或者从SRAM中启动时或者从boot loader模式启动，如果对主存储块执行一次编程/擦除操作，FMC_STAT寄存器中的WPERR位会被置位。在低安全保护等级下，对于选项字节的所有操作都被允许。如果通过设置SPC字节为0xAA进入无保护状态，主存储闪存块将执行一次整片擦除操作。

从保护等级低到无保护状态的降级将导致以下擦除事件：

- 主存储闪存块中不包含DCRP/安全访问/WP区域的扇区，都将会被擦除；
- 擦除/编程保护的扇区将会被擦除；
- DCRP区域：如果FMC_DCRPADDR_EFT和FMC_DCRPADDR_MDF寄存器中的DCRP_EREN位均为0的情况，属于DCRP区域的扇区将不会被擦除。否则，无论是否和安全用户区域有重叠，DCRP区域将会被擦除；
- 安全用户区域：如果FMC_SCRADDR_EFT和FMC_SCRADDR_MDF寄存器中的SCR_EREN位均为0的情况，属于安全用户区域的扇区将不会被擦除。否则，无论是否和DCRP区域有重叠，安全用户区域将会被擦除；
- 如果FMC_DCRPADDR_EFT或FMC_DCRPADDR_MDF寄存器中的DCRP_EREN位至少有一个为1，并且FMC_SCRADDR_EFT或FMC_SCRADDR_MDF寄存器中的SCR_EREN位至少有一个为1，这时候整个主存储块的所有内容都将会被擦除；

- 其他的安全区域（如备份SRAM）将会被擦除。

注意：用户只有通过修改受保护的扇区的边界的方式，才能移除DCRP或安全用户区域。

保护等级高

当熔丝中的SPC_H为1或设置SPC选项字节为0xCC，激活高安全保护等级。当编程选择该保护等级时，调试模式，从SRAM中启动，或者从boot loader启动都被禁止。主存储闪存块可由用户代码的所有操作进行访问。SPC字节禁止再次编程。所以，如果高保护等级被激活，将不能再降回到低保护等级或无保护等级。在高保护等级下，无法更改选项字节。

在保护等级高下，debug端口将永久禁用。因此，无法对设置了保护等级高的设备进行调试分析。如果在调试器在仍处于连接状态时设置了保护等级高，应当进行一次上电复位操作。

用户可以参考*表3-4. SPC保护等级配置*来配置SPC保护等级。

表 3-4. SPC 保护等级配置

EFUSE_USER_CTL 寄存器		FMC_OBSTAT0_MDF 寄存器	FMC_OBSTAT0_EFT 寄存器	SPC等级
SPC_H	SPC_L	SPC[7:0]	SPC[7:0]	
1	0 或 1	任意值	= 0xCC	保护等级高
0 或 1	0 或 1	0xCC	= 0xCC	保护等级高
0	1	除 0xCC 外任意值	= 0xFF	保护等级低
0	0	除0xAA、0xCC外任意值	= FMC_OBSTAT0_MDF 寄存器中 SPC[7:0]值	保护等级低
0	0	0xAA	= 0xAA	无保护

注意：（1）如果熔丝中设置的SPC保护等级和FMC中设置的SPC保护等级不一样，实际生效的SPC等级取两者中SPC保护等级高的那组设置。且FMC_OBSTAT0_EFT寄存器中的SPC[7:0]位反映的是实际生效的SPC保护等级。（2）如果熔丝中的SPC_L位置位，将禁止SPC保护等级低到无保护状态的降级。否则FMC_STAT寄存器的OBMERR位将会被置位。

3.3.10. DCRP 区域

在主存储闪存块，FMC可以定义仅执行区域，仅允许来自系统的指令，但不允许数据访问。

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。

可以通过设置粒度为4KB字节的DCRP_AREA_END[10:0]和DCRP_AREA_START[10:0]选项字节来定义一个DCRP区域。实际DCRP区域大小由以下公式定义：

- [(DCRP_AREA_END - DCRP_AREA_START) + 1] x 4Kbytes.

例如，要在扇区0到扇区15上设置DCRP区域，选项字节应设置如下：

- DCRP_AREA_START[10:0] = 0x000.
- DCRP_AREA_END[10:0] = 0x00F.

因此，DCRP区域大小为：

- [(DCRP_AREA_END - DCRP_AREA_START) + 1] x 4Kbytes = 64 Kbytes.

DCRP区域最小可设置为8KB字节，最大可设置为整个主存储闪存块（将DCRP起始地址与结束地址设置为相等）。

在该区域执行代码时，将忽略调试事件。只有CPU可以访问它，且只使用指令获取任务。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并产生一个读保护错误（RPERR），写操作将被忽略并产生一个编程/擦除保护错误（WPERR）。

DCRP区域受擦除保护，无法擦除该区域内的扇区。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或执行带清除保护的整片擦除操作，否则无法执行整片擦除。

只有CPU可以修改DCRP区域大小设置和DCRP_EREN位。若DCRP区域存在，如果在SPC保护等级低到无保护状态的降级期间，除了在FMC_DCRPADDR_EFT和FMC_DCRPADDR_MDF寄存器中DCRP_EREN位均为0的情况下，不会擦除DCRP区域的内容，其他情况下都将擦除该区域的内容。

除选项字节外，也可以通过修改熔丝中的MCU保留段对DCRP区域进行配置，其粒度为32KB字节。

如果熔丝中的MCU保留段的DCRPLK位为1，且熔丝中的DCRP_AREA_END[7:0]、DCRP_AREA_START[7:0]位不全为0，则DCRP区域大小由熔丝MCU保留段中的DCRP_AREA_END[7:0]、DCRP_AREA_START[7:0]位定义。否则，DCRP区域大小由DCRP_AREA_END[10:0]、DCRP_AREA_START[10:0]选项字节决定。更多细节可以参考*表 3-5. DCRP区域配置*。

表 3-5. DCRP 区域配置

选项字节中 DCRP区域设置	熔丝中 DCRP区域设置	熔丝中 DCRPLK位设置	生效的 DCRP区域
区域 1	起始/结束地址都为 0	0	区域 1(2)
	起始/结束地址都为 0	1	区域 1(2)
	区域 2(1)	0	区域 1(2)
	区域 2(1)	1	区域 2

注意：（1）区域2的起始及结束地址不全为0；（2）如果DCRP区域是由熔丝决定，那么FMC_DCRPADDR_MDF及FMC_DCRPADDR_EFT寄存器中的DCRP_AREA_START[10:0]的值为{熔丝中DCRP_AREA_START[7:0],3'b0}，DCRP_AREA_END[10:0]的值为{ 熔丝中DCRP_AREA_END[7:0] ，3'b111 }。

如果DCRP区域由熔丝定义，FMC_DCRPADDR_MDF寄存器中的DCRP_AREA_END[10:0]位或DCRP_AREA_START[10:0]位将不能被修改，即DCRP区域将会固定不变。带清除保护的整片擦除也将不能执行。

注意：（1）熔丝的DCRP配置优先级高于闪存的选项字节，所以在产品中，如果采用选项字节配置DCRP区域时，应该保持熔丝中的DCRP区域的起始/结束地址都设置为0，并将熔丝中DCRPLK位置1，否则可能会导致DCRP区域有漏洞。（2）如果用户有安全性考虑，请使用熔丝对DCRP区域进行配置，否则可能会有安全隐患。

3.3.11. 安全用户区域

在主存储闪存块，FMC可以定义安全用户区域。只有在CPU执行安全应用程序时，才能访问此区域。安全用户区域可以将安全用户代码与应用程序非安全代码隔离。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。

可以通过设置粒度为4KB字节的SCR_AREA_END[10:0]和SCR_AREA_START[10:0]选项字节来定义一个安全用户区域。实际安全用户区域大小由以下公式定义：

- [(SCR_AREA_END[10:0] - SCR_AREA_START[10:0]) + 1] x 4Kbytes.

例如，要在扇区0到扇区15上设置安全用户区域，选项字节应设置如下：

- SCR_AREA_START[10:0] = 0x000.
- SCR_AREA_END[10:0] = 0x00F.

因此，安全用户区域大小为：

- [(SCR_AREA_END[10:0] - SCR_AREA_START[10:0]) + 1] x 4Kbytes = 64 Kbytes.

SCR_AREA_END[10:0]及SCR_AREA_START[10:0]选项字节能由CPU通过运行安全库或应用程序安全代码的方式进行修改。但对于非安全应用程序，只能通过执行SPC降级或带清除保护的整片擦除的方式才能修改（清零）。

安全用户区域区域最小可设置为8KB字节，最大可设置为整个主存储闪存块（将安全用户区域起始地址与结束地址设置为相等）。

在该区域执行代码时，将忽略调试事件。只有通过CPU执行GD安全库或用户安全应用程序来访问它。在所有其他情况下，访问安全用户区域是非法的。例如，读操作将返回0并产生一个读安全错误（RSERR），写操作将被忽略并产生一个编程/擦除保护错误（WPERR）。

安全用户区域受到擦除保护，无法擦除该区域内的扇区。除非应用程序从安全用户区域去执行，否则不可能擦除位于该区域的扇区。如果定义了有效的安全用户区域，则除非执行SPC降级或带清除保护的整片擦除操作，否则无法执行整片擦除。

只有CPU可以修改安全用户区域大小设置和SCR_EREN位。若安全用户区域存在，如果在SPC保护等级低到无保护状态的降级期间，除了在FMC_SCRADDR_EFT和FMC_SCRADDR_MDF寄存器中SCR_EREN位均为0的情况下，不会擦除安全用户区域的内容，其他情况下都将擦除该区域的内容。

除选项字节外，也可以通过修改熔丝中的用户控制段对安全用户区域进行配置，其粒度为32KB字节。

如果熔丝中的用户控制段的SCRLK位为1，且熔丝中的SCR_AREA_END[7:0]、SCR_AREA_START[7:0]位不全为0，则安全用户区域大小由熔丝用户控制段中的SCR_AREA_END[7:0]、SCR_AREA_START[7:0]位定义。否则，安全用户区域大小由SCR_AREA_END[10:0]、SCR_AREA_START[10:0]选项字节决定。更多细节可以参考*表3-6. 安全用户区域配置*。

表 3-6. 安全用户区域配置

选项字节中 安全用户区域设置	熔丝中 安全用户区域设置	熔丝中 SCRLK位设置	生效的 安全用户区域
区域 1	起始/结束地址都为 0	0	区域 1(2)
	起始/结束地址都为 0	1	区域 1(2)
	区域 2(1)	0	区域 1(2)
	区域 2(1)	1	区域 2

注意：（1）区域2的起始及结束地址不全为0；（2）如果安全用户区域是由熔丝决定，那么FMC_SCRADDR_MDF及FMC_SCRADDR_EFT寄存器中的SCR_AREA_START[10:0]的值为{ 熔丝中SCR_AREA_START[7:0] ，3'b0 }，SCR_AREA_END[10:0]的值为{ 熔丝中SCR_AREA_END[7:0] ，3'b111 }。

如果安全用户区域由熔丝定义，FMC_SCRADDR_MDF寄存器中的SCR_AREA_END[10:0]位和SCR_AREA_START[10:0]位将不能被修改，即安全用户区域将会固定不变。带清除保护的整片擦除也将不能执行。

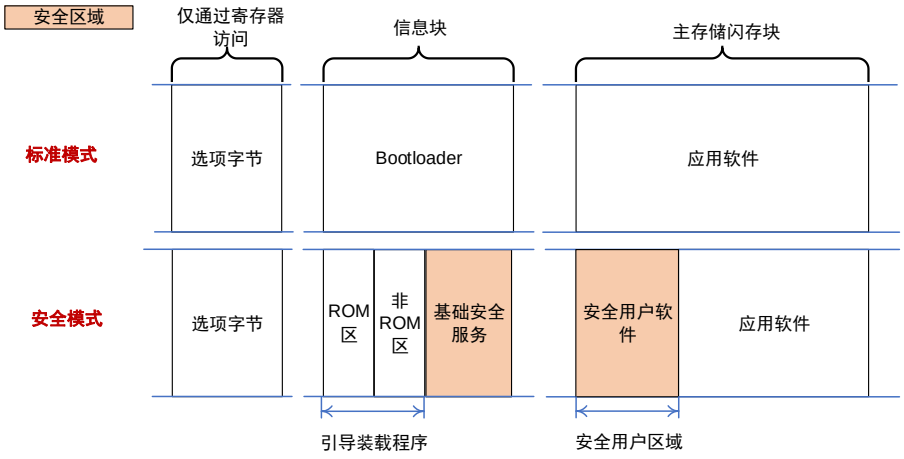
注意：（1）熔丝的安全配置优先级高于闪存的选项字节，所以在产品中，如果采用选项字节配置安全用户区域时，应该保持熔丝中的安全用户区域的起始/结束地址都设置为0，并将熔丝中SCRLK位置1，否则可能会导致安全区域有漏洞。（2）如果用户有安全性考虑，请在熔丝中对安全用户区域进行配置，否则会有安全隐患。

3.3.12. 安全模式

对某些敏感程序设置安全保护是有必要的，以避免潜在的恶意软件攻击。例如，预授权固件更新软件需要高度的保护，因为它处理其他进程无法检索的机密数据（如加密秘钥）。

MCU提供了具有限制访问特性的安全存储区。在该区域里能够构建，可以在任何用户应用程序之前执行的安全服务。只有在设备配置为安全模式时，才能访问这些安全区域及其包含的软件。*图3-6. 标准模式和安全模式内部存储架构*显示了该区域的详细信息。

图 3-6. 标准模式和安全模式内部存储架构



其中，安全用户区域在复位后被访问一次，且区域内代码执行完成后将被隐藏。基础安全服务是配置安全服务的 Gigadevice 软件。安全用户软件位于安全用户区域，复位后执行一次。安全用户软件可以实现安全启动和预授权固件更新。

安全模式和安全用户区域可以通过选项字节或熔丝配置。用户可以通过基础安全服务去设置选项字节中的安全用户区域，从而把安全代码和数据配置在安全用户区域。

在安全模式下，安全固件存储在信息块中，以支持引导。安全用户软件是存储在主存储闪存块中的安全应用代码、数据或算法。

如果没有配置安全用户区域，则安全引导装载程序将跳转至由 FMC_BTADDR_EFT 寄存器中的 BOOT_ADDR0[15:0]选项字节决定的引导地址。

在安全模式下，MCU 无论引导配置如何（BOOT 引脚及 BOOT 地址），都会强制从安全 ROM 区启动。

只要选项字节或熔丝中的SCR位有一个位被置为了1，安全模式都将被启用，此时FMC_OBSTAT0_EFT寄存器的SCR位为1。且启用后需要系统复位来激活安全模式。

当选项字节的SCR位为1而熔丝中的SCR位不为1时，可以返回标准模式，若要返回标准模式，在清除SCR选项位之前或在清除SCR选项位的同时，需要移除安全用户区域和DCRP区域。具体操作详见相关选项字节的修改规则。

注意：如果用户有安全性考虑，请在熔丝中对安全模式进行配置，否则会有安全隐患。

3.3.13. 基础安全服务

基础安全服务提供了安全区域设置函数和安全区域退出函数。

- 安全区域设置服务

安全区域设置函数是由Gigadevice提供用来执行安全区域初始化的函数，在标准模式下运行时，可以通过直接调用函数（resetAndInitializeSecureAreas）来设置安全用户区域，而其他基础安全服务不允许访问。

函数*表3-7. 函数resetAndInitializeSecureAreas*见下表:

表 3-7. 函数 resetAndInitializeSecureAreas

函数名称	resetAndInitializeSecureAreas
函数原型	void resetAndInitializeSecureAreas(BSS_secure_area_struct area);
功能描述	根据SCR_AREA_START和SCR_AREA_END选项字节来配置安全用户区域范围。
先决条件	-
被调用函数	-
输入参数{in}	
area	安全用户区域起始及结束地址
输出参数{out}	
-	-
返回值	
-	-

注意：在函数完成后,将产生系统复位。此函数仅在首次设置安全用户区域时才可使用。用户应该保证目标安全区域内有正确的安全程序，使其能够退出到标准程序，否则会造成芯片报废。

- 安全区域退出服务

Gigadevice提供了一个跳转到用户应用程序的函数（exitSecureArea）。它允许安全地关闭安全用户区域，以保证安全区域的内容不再被访问。

函数secure_area_exit见下表:

表 3-8. 函数 exitSecureArea

函数名称	exitSecureArea
函数原型	void exitSecureArea(unsigned int vectors, unsigned int jtagState);
功能描述	从安全用户区域退出并跳转到主用户应用程序
先决条件	-
被调用函数	-
输入参数{in}	
vectors	退出后要跳转的应用程序向量的地址
输入参数{in}	
jtag_state	退出安全用户区域后的JTAG的状态
BSS_EXIT_SCR_JTAG_ENABLE	退出安全用户区域后使能JTAG
BSS_EXIT_SCR_JTAG_DISABLE	退出安全用户区域后失能JTAG
输出参数{out}	
-	-
返回值	
-	-

注意：在函数完成后,将不会产生系统复位。出于安全考虑，用户在安全用户区域调用此函数前应禁用缓存。

3.3.14. 错误描述

擦除/编程保护错误（WPERR）

以下擦除操作将被拒绝，并置位FMC_STAT寄存器中WPERR位：

- 擦除有效的DCRP区域内的扇区；
- 擦除有效的安全用户区域内的扇区（除非应用程序是从有效的安全用户区域执行的）；
- 擦除有效的擦除/编程保护扇区；
- 擦除主存储闪存块外的区域。

以下编程操作将被忽略，并置位FMC_STAT寄存器中WPERR位：

- SPC保护等级低，且在调试模式或从SRAM中启动或从boot loader启动时，去编程主存储闪存块；
- 未使用GD安全库去编程有效的DCRP区域内的扇区；
- 未使用用户安全代码或GD安全库去编程有效的安全用户区域内的扇区；
- 编程有效的擦除/编程保护扇区；
- 编程主存储闪存块外的区域。

如果FMC_CTL寄存器中的WPERRIE位设置为1，则在WPERR置位时生成中断。软件可以通过写1来清除它。

编程顺序错误（PGSERR）

以下操作将会置位FMC_STAT寄存器中PGSERR位，且中止当前编程的操作：

- 当请求编程操作前，没有将FMC_CTL寄存器中的编程启用位（PG）置位时；
- 在编程操作之前，当PGCHEN位置位，但被编程的区域不全为0时。

注意：当突发编程操作时，PGCHEN为1，FMC将根据突发数据宽度来检查闪存数据。如果数据并非全部为1，则该节拍编程失败，并将FMC_STAT寄存器中的PGSERR置位。但其他节拍可以正常编程。

如果将FMC_CTL寄存器中的PGSERRIE位设置为1，则在触发PGSERR标志时产生中断。软件可以通过写1来清除它。

读保护错误（RPERR）

当尝试对SPC、DCRP保护的区域进行读操作时，FMC_STAT寄存器中RPERR将会置位，且中止当前的读操作，但应用程序可以请求新的读取操作。

如果将FMC_CTL寄存器中的RPERRIE位设置为1，则在触发RPERR标志时产生中断。软件可以通过写1来清除它。

读安全错误（RSERR）

当尝试对一个安全用户区域进行读操作时，FMC_STAT寄存器中RSERR将会置位，且中止当前的读操作。

如果将FMC_CTL寄存器中的RSERRIE位设置为1，则在触发RSERR标志时产生中断。软件可以通过写1来清除它。

选项字节修改错误（OBMERR）

当选项字节修改操作期间发生错误时，FMC_STAT寄存器中OBMERR将会置位，且中止当前的操作。

如果将FMC_CTL寄存器中的OBMERRIE位设置为1，则在触发OBMERR标志时产生中断。软件可以通过写1来清除它。

硬件故障错误

下列操作将会产生总线错误：

- SPC保护等级低，且在调试模式或从SRAM中启动或从boot loader启动时，去访问主存储闪存块；
- 没有正确访问权限却访问安全用户区域；
- 访问地址超出范围；
- FMC_CTL/FMC_OBCTL寄存器解锁时写入秘钥的顺序不对。

3.3.15. FMC 中断

FMC的中断事件和标志位如*表3-9. FMC中断请求*所示。

表 3-9. FMC 中断请求

中断标志	描述	清除方式	中断使能位
ENDF	操作结束	向FMC_STAT寄存器对应位写1	ENDIE
WPERR	擦除/编程保护错误		WPERRIE
PGSERR	编程顺序错误		PGSERRIE
RPERR	读保护错误		RPERRIE
RSERR	读安全错误		RSERRIE
OBMERR	选项字节修改错误		OBMERRIE

3.4. FMC 寄存器

FMC基地址：0x5200 2000

3.4.1. 解锁寄存器（FMC_KEY）

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

3.4.2. 选项字节操作解锁寄存器（FMC_OBKEY）

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

位/位域	名称	描述
31:0	OBKEY [31:0]	这些位仅能被软件写 写解锁值到OBKEY[31:0]解锁FMC_OBCTL寄存器。

3.4.3. 控制寄存器（FMC_CTL）

地址偏移：0x0C
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							RSERRIE	RPERRIE	保留				PGSERRIE	WPERRIE	ENDIE
							rw	rw					rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								START	保留		PGCHEN	MER	SER	PG	LK
								rw			rw	rw	rw	rw	rs

位/位域	名称	描述
31:25	保留	必须保持复位值。
24	RSERRIE	读安全错误中断使能位 当 LK 设置为 0 时，该位才能被软件置 1 和清 0。 0：失能读安全错误中断 1：使能读安全错误中断
23	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
22:19	保留	必须保持复位值。
18	PGSERRIE	编程顺序错误中断使能位 当 LK 设置为 0 时，该位才能被软件置 1 和清 0。 0：失能编程顺序错误中断 1：使能编程顺序错误中断
17	WPERRIE	擦除/删除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：失能擦除/删除保护错误中断 1：使能擦除/删除保护错误中断
16	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
15:8	保留	必须保持复位值。
7	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，发送擦除命令到 FMC。 当 BUSY 位被清 0 时，此位由硬件清 0。
6:5	保留	必须保持复位值。
4	PGCHEN	编程区域检查使能位 当 LK 设置为 0 时，该位才能被软件置 1 和清 0。 0：编程前不去检查编程区域数据是否为全 0xFF 1：编程前去检查编程区域数据是否为全 0xFF

		若该位置 1，且编程区域数据不全为 0xFF 时，PGSERR 将会置位。且该编程操作无效。
3	MER	整片擦除命令位 当 LK 设置为 0 时，该位才能被软件置 1 和清 0。 0：无作用 1：整片擦除命令 如果同时请求整片擦除和扇区擦除，则整片擦除将替代扇区擦除操作。
2	SER	扇区擦除命令位 当 LK 设置为 0 时，该位才能被软件置 1 和清 0。 0：无作用 1：扇区擦除命令 如果同时请求整片擦除和扇区擦除，则整片擦除将替代扇区擦除操作。
1	PG	主存储闪存块编程命令位 当 LK 设置为 0 时，该位才能被软件置 1 和清 0。 0：无作用 1：主存储闪存块编程命令
0	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。

注意：当相应闪存操作完成后，该寄存器需处于复位状态

3.4.4. 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	OBMERR	保留					RSERR	RPERR	保留				PGSERR	WPERR	ENDF
	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
保留															BUSY
															r

位/位域	名称	描述
31	保留	必须保持复位值。
30	OBMERR	选项字节修改错误标志位 该位由硬件置位，软件写 1 清 0。 0：未发生选项字节修改错误 1：发生选项字节修改错误
29:25	保留	必须保持复位值。
24	RSERR	读安全错误标志位

		该位由硬件置位，软件写 1 清 0。 0：未发生读安全错误 1：发生读安全错误
23	RPERR	读保护错误标志位 该位由硬件置位，软件写 1 清 0。 0：未发生读保护错误 1：发生读保护错误
22:19	保留	必须保持复位值。
18	PGSERR	编程顺序错误标志位 该位由硬件置位，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
17	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
16	ENDF	操作结束标志位 当操作执行成功，此位被硬件置 1。软件写 1 清 0。
15:1	保留	必须保持复位值。
0	BUSY	闪存忙标志位 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。

3.4.5. 地址寄存器（FMC_ADDR）

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:0	ADDR[31:0]	闪存擦除地址 该位通过软件设置。 ADDR 位是闪存擦除命令的地址。

3.4.6. 选项字节控制寄存器（FMC_OBCTL）

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	OBMERRIE	保留													
	rw														

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

位/位域	名称	描述
31	保留	必须保持复位值。
30	OBMERRIE	选项字节修改错误中断使能位 当 OBLK 设置为 0 时，该位才能被软件置 1 和清 0。 0：失能选项字节修改错误中断 1：使能选项字节修改错误中断
29:2	保留	必须保持复位值。
1	OBSTART	发送选项字节命令到 FMC 仅当 OBLK 设置为 0 时，该位才能由软件置 1。当 BUSY 位清 0 时由硬件清除该位。
0	OBLK	FMC_OBCTL 锁定位 当往 FMC_OBKEY 寄存器写值顺序正确时，该位由硬件清 0。软件置 1。

3.4.7. 选项字节状态寄存器 0（FMC_OBSTAT0_EFT）

地址偏移：0x1C
复位值：0xXXXX XXXX，出厂值为0x01C6 AAD0

该寄存器是相应选项位的生效值。复位后装载选项字节中的值。

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		IOSPDOPEN	保留				DTCM1ECCEN	DTCM0ECCEN	ITCMECCEN	SCR	保留		FWDGSPD_STDBY	FWDGSPD_DPSLP	保留
		r					r	r	r	r			r	r	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPC[7:0]								nRST_STDBY	nRST _DPSLP	保留	nWDG_HW	BOR_TH[1:0]		保留	
r								r	r		r	r			

位/位域	名称	描述

31:30	保留	必须保持复位值。
29	IOSPDOPEN	低电压下的 I/O 速度优化功能的允许使能状态位 0：芯片工作电压大于 2.5V，因此 I/O 速度优化不被允许 1：芯片工作电压低于 2.5V，因此 I/O 速度优化被允许
28:25	保留	必须保持复位值。
24	DTCM1ECCEN	DTCM1 的 ECC 功能使能状态位 0：失能 DTCM1 的 ECC 功能 1：使能 DTCM1 的 ECC 功能
23	DTCM0ECCEN	DTCM0 的 ECC 功能使能状态位 0：失能 DTCM0 的 ECC 功能 1：使能 DTCM0 的 ECC 功能
22	ITCMECCEN	ITCM 的 ECC 功能使能状态位 0：失能 ITCM 的 ECC 功能 1：使能 ITCM 的 ECC 功能
21	SCR	安全模式使能状态位 0：失能安全模式 1：使能安全模式
20:19	保留	必须保持复位值。
18	FWDGSPD_STDBY	待机模式下独立看门狗（FWDG）暂停选项状态位 0：在待机状态下暂停独立看门狗 1：在待机状态下运行独立看门狗
17	FWDGSPD_DPSLP	深度睡眠模式下独立看门狗暂停选项状态位 0：在深度睡眠状态下暂停独立看门狗 1：在深度睡眠状态下运行独立看门狗
16	保留	必须保持复位值。
15:8	SPC[7:0]	安全保护等级状态值 0xAA：无保护状态 0xCC：安全保护等级高 除 0xAA 或 0xCC 之外任何值：安全保护等级低
7	nRST_STDBY	进入待机模式复位选项状态位 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
6	nRST_DPSLP	进入深度睡眠模式复位选项状态位 0：进入深度睡眠模式时产生复位 1：进入深度睡眠模式时不产生复位
5	保留	必须保持复位值。
4	nWDG_HW	看门狗控制状态位

		0：硬件控制看门狗 1：软件控制看门狗
3:2	BOR_TH[1:0]	欠压复位（BOR）阈值状态位 00：无 BOR 功能 01：BOR 阈值 1 10：BOR 阈值 2 11：BOR 阈值 3
1:0	保留	必须保持复位值。

3.4.8. 选项字节状态寄存器 0（FMC_OBSTAT0_MDF）

地址偏移：0x20
复位值：0xXXXX XXXX

该寄存器是相应选项位的修改值。系统复位后的值是相应选项位的生效值。

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		IOSPDOPEN	保留				DTCM1ECCEN	DTCM0ECCEN	ITCMECCEN	SCR	保留		FWDGSP D_STDBY	FWDGSP D_DPSLP	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
SPC[7:0]								nRST_STDBY	nRST_DPSLP	保留	nWDG_HW	BOR_TH[1:0]		保留	
			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	IOSPDOPEN	低电压下的 I/O 速度优化功能的允许使能配置位 0：芯片工作电压大于 2.5V，因此 I/O 速度优化不被允许 1：芯片工作电压低于 2.5V，因此 I/O 速度优化被允许
28:25	保留	必须保持复位值。
24	DTCM1ECCEN	DTCM1 的 ECC 功能使能配置位 0：失能 DTCM1 的 ECC 功能 1：使能 DTCM1 的 ECC 功能
23	DTCM0ECCEN	DTCM0 的 ECC 功能使能配置位 0：失能 DTCM0 的 ECC 功能 1：使能 DTCM0 的 ECC 功能
22	ITCMECCEN	ITCM 的 ECC 功能使能配置位 0：失能 ITCM 的 ECC 功能 1：使能 ITCM 的 ECC 功能

21	SCR	安全模式使能配置位 0：失能安全模式 1：使能安全模式
20:19	保留	必须保持复位值。
18	FWDGSPD_STDBY	待机模式下独立看门狗暂停选项配置位 0：在待机状态下暂停独立看门狗 1：在待机状态下运行独立看门狗
17	FWDGSPD_DPSLP	深度睡眠模式下独立看门狗暂停选项配置位 0：在深度睡眠状态下暂停独立看门狗 1：在深度睡眠状态下运行独立看门狗
16	保留	必须保持复位值。
15:8	SPC[7:0]	安全保护等级配置值 0xAA：无保护状态 0xCC：安全保护等级高 除 0xAA 或 0xCC 之外任何值：安全保护等级低
7	nRST_STDBY	进入待机模式复位选项配置位 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
6	nRST_DPSLP	进入深度睡眠模式复位选项配置位 0：进入深度睡眠模式时产生复位 1：进入深度睡眠模式时不产生复位
5	保留	必须保持复位值。
4	nWDG_HW	看门狗控制配置位 0：硬件控制看门狗 1：软件控制看门狗
3:2	BOR_TH[1:0]	BOR 阈值配置位 00：无 BOR 功能 01：BOR 阈值 1 10：BOR 阈值 2 11：BOR 阈值 3
1:0	保留	必须保持复位值。

3.4.9. DCRP 地址寄存器（FMC_DCRPADDR_EFT）

地址偏移：0x28
复位值：0xXXXX 0XXX，出厂值为0x0000 00FF

该寄存器是相应选项位的生效值。复位后装载选项字节中的值。

该寄存器只能按字（32位）访问。

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

位/位域	名称	描述
31	DCRP_EREN	DCRP 区域擦除使能状态位 0：DCRP 不被擦除 1：当 SPC 降级或执行带清除保护的整片擦除操作时，DCRP 区域被擦除。
30:27	保留	必须保持复位值。
26:16	DCRP_AREA_END[10:0]	DCRP 区域结束地址状态位 该位域包含了 DCRP 区域的最后的 4K 字节块。 区域最后一个字节地址= (DCRP_AREA_END[10:0] + 1) * 4096 – 1 + 0x0800_0000 如果 DCRP_AREA_END[10:0]等于 DCRP_AREA_START[10:0]，整个主存储闪存块都是 DCRP 区域。 如果 DCRP_AREA_END[10:0]小于 DCRP_AREA_START[10:0]，DCRP 区域为空。
15:11	保留	必须保持复位值。
10:0	DCRP_AREA_START[10:0]	DCRP 区域起始地址状态位 该位域包含了 DCRP 区域的起始的 4K 字节块。 区域第一个字节地址= DCRP_AREA_START[10:0] * 4096 + 0x0800_0000 如果 DCRP_AREA_END 等于 DCRP_AREA_START，整个主存储闪存块都是 DCRP 区域。 如果 DCRP_AREA_END 小于 DCRP_AREA_START，DCRP 区域为空。

3.4.10. DCRP 地址寄存器（FMC_DCRPADDR_MDF）

地址偏移：0x2C
复位值：0xXXXX 0XXX

该寄存器是相应选项位的修改值。系统复位后的值是相应选项位的生效值。

该寄存器只能按字（32位）访问。

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

位/位域	名称	描述
31	DCRP_EREN	DCRP 区域擦除使能选项配置位 0：DCRP 不被擦除 1：当 SPC 降级或执行带清除保护的整片擦除操作时，DCRP 被擦除。
30:27	保留	必须保持复位值。
26:16	DCRP_AREA_END[10:0]	DCRP 区域结束地址配置位 该位域包含了 DCRP 区域的最后的 4K 字节块。 区域最后一个字节地址= (DCRP_AREA_END[10:0] + 1) * 4096 – 1 + 0x0800_0000 如果 DCRP_AREA_END[10:0]等于 DCRP_AREA_START[10:0]，整个主存储闪存块都是 DCRP 区域。 如果 DCRP_AREA_END[10:0]小于 DCRP_AREA_START[10:0]，DCRP 区域为空。
15:11	保留	必须保持复位值。
10:0	DCRP_AREA_START[10:0]	DCRP 区域起始地址配置位 该位域包含了 DCRP 区域的起始的 4K 字节块。 区域第一个字节地址= DCRP_AREA_START[10:0] * 4096 + 0x0800_0000 如果 DCRP_AREA_END[10:0]等于 DCRP_AREA_START[10:0]，整个主存储闪存块都是 DCRP 区域。 如果 DCRP_AREA_END[10:0]小于 DCRP_AREA_START[10:0]，DCRP 区域为空。

3.4.11. 安全用户区域地址寄存器（FMC_SCRADDR_EFT）

地址偏移：0x30
复位值：0xXXXX 0XXX，出厂值为0x0000 00FF

该寄存器是相应选项位的生效值。复位后装载选项字节中的值。

该寄存器只能按字（32位）访问。

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

位/位域	名称	描述
31	SCR_EREN	安全用户区域擦除使能选项状态位 0：安全用户区域不被擦除 1：当 SPC 降级或执行带清除保护的整片擦除操作时，安全用户区域被擦除。
30:27	保留	必须保持复位值。
26:16	SCR_AREA_END[10:0]	安全用户区域结束地址状态位 该位域包含了安全用户区域的最后的 4K 字节块。

		区域最后一个字节地址= (SCR_AREA_END[10:0] + 1) * 4096 – 1 + 0x0800_0000 如果 SCR_AREA_END[10:0]等于 SCR_AREA_START[10:0]，整个主存储闪存块都是安全用户区域。 如果 SCR_AREA_END[10:0]小于 SCR_AREA_START[10:0]，安全用户区域为空。
15:11	保留	必须保持复位值。
10:0	SCR_AREA_START[10:0]	安全用户区域起始地址状态位 该位域包含了安全用户区域的起始的 4K 字节块。 区域第一个字节地址= SCR_AREA_START[10:0] * 4096 + 0x0800_0000 如果 SCR_AREA_END[10:0]等于 SCR_AREA_START[10:0]，整个主存储闪存块都是安全用户区域。如果 SCR_AREA_END[10:0]小于 SCR_AREA_START[10:0]，安全用户区域为空。

3.4.12. 安全用户区域地址寄存器（FMC_SCRADDR_MDF）

地址偏移：0x34

复位值：0xXXXX 0XXX

该寄存器是相应选项位的修改值。系统复位后的值是相应选项位的生效值。

该寄存器只能按字（32位）访问。

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

位/位域	名称	描述
31	SCR_EREN	安全用户区域擦除使能选项配置位 0：安全用户区域不被擦除 1：当 SPC 降级或执行带清除保护的整片擦除操作时，安全用户区域被擦除。
30:27	保留	必须保持复位值。
26:16	SCR_AREA_END[10:0]	安全用户区域结束地址配置位 该位域包含了安全用户区域的最后的 4K 字节块。 区域最后一个字节地址= (SCR_AREA_END[10:0] + 1) * 4096 – 1 + 0x0800_0000 如果 SCR_AREA_END 等于 SCR_AREA_START，整个主存储闪存块都是安全用户区域。 如果 SCR_AREA_END 小于 SCR_AREA_START，安全用户区域为空。
15:11	保留	必须保持复位值。
10:0	SCR_AREA_START[10:0]	安全用户区域起始地址配置位 该位域包含了安全用户区域的起始的 4K 字节块。

区域第一个字节地址= SCR_AREA_START[10:0] * 4096 + 0x0800_0000
如果 SCR_AREA_END[10:0]等于 SCR_AREA_START[10:0]，整个主存储闪存块都是安全用户区域。
如果 SCR_AREA_END[10:0]小于 SCR_AREA_START[10:0]，安全用户区域为空。

3.4.13. 擦除/编程保护寄存器（FMC_WP_EFT）

地址偏移：0x38
复位值：0xXXXX XXXX，出厂值为0x3FFF FFFF

该寄存器是相应选项位的生效值。复位后装载选项字节中的值。

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:22	保留	必须保持复位值。
21:0	WP[21:0]	扇区擦除/编程保护选项状态位 对于 WP[21]，每一位反映了对应 64 个扇区的擦除/编程保护状态。 0：对应的 64 个扇区受擦除/编程保护 1：对应的 64 个扇区不受擦除/编程保护 对于 WP[20:16]，每一位反映了对应 128 个扇区的擦除/编程保护状态。 0：对应的 128 个扇区受擦除/编程保护 1：对应的 128 个扇区不受擦除/编程保护 对于 WP[15:0]，每一位反映了对应 16 个扇区的擦除/编程保护状态。 0：对应的 16 个扇区受擦除/编程保护 1：对应的 16 个扇区不受擦除/编程保护

3.4.14. 擦除/编程保护寄存器（FMC_WP_MDF）

地址偏移：0x3C
复位值：0xXXXX XXXX

该寄存器是相应选项位的修改值。系统复位后的值是相应选项位的生效值。

该寄存器只能按字（32位）访问。

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

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

WP[15:0]
rw

位/位域	名称	描述
31:22	保留	必须保持复位值。
21:0	WP[21:0]	扇区擦除/编程保护选项配置位 对于 WP[21]，该位可以将对应 64 个扇区设置为擦除/编程保护。 0：将对应的 64 个扇区设置为受擦除/编程保护 1：将对应的 64 个扇区设置为不受擦除/编程保护 对于 WP[20:16]，每一位可以将对应 128 个扇区设置为擦除/编程保护。 0：将对应的 128 个扇区设置为受擦除/编程保护 1：将对应的 128 个扇区设置为不受擦除/编程保护 对于 WP[15:0]，每一位可以将对应 16 个扇区设置为擦除/编程保护。 0：将对应的 16 个扇区设置为受擦除/编程保护 1：将对应的 16 个扇区设置为不受擦除/编程保护

3.4.15. 引导装载地址寄存器（FMC_BTADDR_EFT）

地址偏移：0x40
复位值：0xXXXX XXXX，出厂值为0x1FF0 0800

该寄存器是相应选项位的生效值。复位后装载选项字节中的值。

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BOOT_ADDR1[15:0]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOOT_ADDR0[15:0]															
r															

位/位域	名称	描述
31:16	BOOT_ADDR1[15:0]	引导装载地址 1 状态位 如果 BOOT 管脚拉高，引导装载地址的高 16 位为该字域。
15:0	BOOT_ADDR0[15:0]	引导装载地址 0 状态位 如果 BOOT 管脚拉低，引导装载地址的高 16 位为该字域。

3.4.16. 引导装载地址寄存器（FMC_BTADDR_MDF）

地址偏移：0x44
复位值：0xXXXX XXXX

该寄存器是相应选项位的修改值。系统复位后的值是相应选项位的生效值。

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BOOT_ADDR1[15:0]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOOT_ADDR0[15:0]															
rw															

位/位域	名称	描述
31:16	BOOT_ADDR1[15:0]	引导装载地址 1 配置位. 如果 BOOT 管脚拉高，引导装载地址的高 16 位为该字域。
15:0	BOOT_ADDR0[15:0]	引导装载地址 0 配置位. 如果 BOOT 管脚拉低，引导装载地址的高 16 位为该字域。

3.4.17. 选项字节状态寄存器 1（FMC_OBSTAT1_EFT）

地址偏移：0x50

复位值：0xXXXX 0XXX，出厂值为0x0000 0087

该寄存器是相应选项位的生效值。复位后装载选项字节中的值。

该寄存器只能按字（32位）访问。

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

位/位域	名称	描述
31:16	DATA[15:0]	用户定义选项字节状态位
15:8	保留	必须保持复位值。
7:4	DTCM_SZ_SHRRAM[3:0]	共享 RAM 中的 DTCM 大小状态位 DTCM + ITCM 大小不能超过 512KB 字节 0000：0 字节 DTCM 0001~0110：保留 0111：64-KB DTCM 1000：128-KB DTCM 1001：256-KB DTCM 1010：512-KB DTCM 1011~1111：保留
3:0	ITCM_SZ_SHRRAM[3:0]	共享 RAM 中的 ITCM 大小状态位 DTCM + ITCM 大小不能超过 512KB 字节 0000：0 字节 ITCM

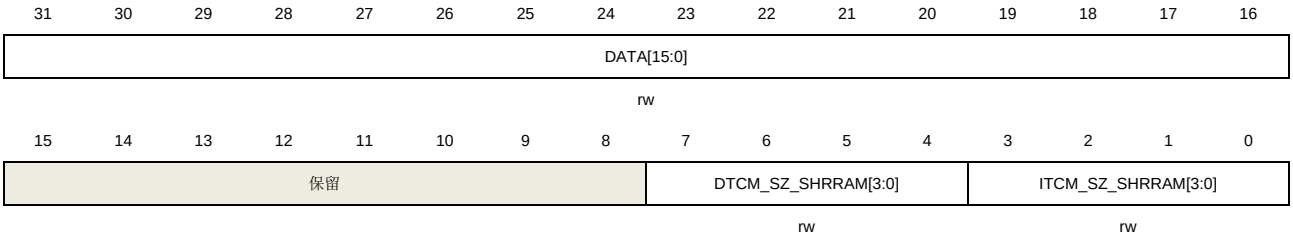
0001~0110：保留
0111：64-KB ITCM
1000：128-KB ITCM
1001：256-KB ITCM
1010：512-KB ITCM
1011~1111：保留

3.4.18. 选项字节状态寄存器 1（FMC_OBSTAT1_MDF）

地址偏移：0x54
复位值：0xXXXX 0XXX

该寄存器是相应选项位的修改值。系统复位后的值是相应选项位的生效值。

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:16	DATA[15:0]	用户定义选项字节配置位
15:8	保留	必须保持复位值。
7:4	DTCM_SZ_SHRRAM[3:0]	共享 RAM 中的 DTCM 大小配置位 DTCM + ITCM 大小不能超过 512KB 字节 0000：0 字节 DTCM 0001~0110：保留 0111：64-KB DTCM 1000：128-KB DTCM 1001：256-KB DTCM 1010：512-KB DTCM 1011~1111：保留
3:0	ITCM_SZ_SHRRAM[3:0]	共享 RAM 中的 ITCM 大小配置位 DTCM + ITCM 大小不能超过 512KB 字节 0000：0 字节 ITCM 0001~0110：保留 0111：64-KB ITCM 1000：128-KB ITCM 1001：256-KB ITCM 1010：512-KB ITCM 1011~1111：保留

3.4.19. NO-RTDEC 区域寄存器（FMC_NODEC）

地址偏移：0x60
复位值：0x0000 00FF

当LK位设置为0时，该寄存器才能被访问，只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留					NODEC_AREA_END[10:0]										
					rw										

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

位/位域	名称	描述
31:27	保留	必须保持复位值。
26:16	NODEC_AREA_END[10:0]	NO-RTDEC 区域结束地址 该位域包含了 NO-RTDEC 区域的最后的 4K 字节块。 区域最后一字节地址= NODEC_AREA_END[10:0] * 4096 – 1 + 0x0800_0000。 如果 NODEC_AREA_END[10:0]等于 NODEC_AREA_START[10:0]，整个主存储闪存块在读操作时都不解密。 如果 DCRP_AREA_END[10:0]小于 DCRP_AREA_START[10:0]，整个主存储闪存块在读操作时都解密。
15:11	保留	必须保持复位值。
10:0	NODEC_AREA_START[10:0]	NO-RTDEC 区域起始地址 该位域包含了 NO-RTDEC 区域的起始的 4K 字节块。 最后一字节地址= NODEC_AREA_ START [10:0] * 4096 + 0x0800_0000。 如果 NODEC_AREA_END[10:0]等于 NODEC_AREA_START[10:0]，整个主存储闪存块在读操作时都不解密。 如果 DCRP_AREA_END[10:0]小于 DCRP_AREA_START[10:0]，整个主存储闪存块在读操作时都解密。

3.4.20. AES 初始向量寄存器 x（FMC_AESIVx_EFT）（x = 0…2）

地址偏移：0x68 + 0x4 * x
复位值：0xXXXX XXXX

该寄存器是AES初始向量高96位的生效值。AES初始向量不是选项字节，而是存放在非易失性AES IV存储区内，复位后从该区域中装载。

该寄存器只能按字（32位）访问。

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

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

AESIV[15:0]

r

位/位域	名称	描述
31:0	AESIV[31:0]	AES 初始向量状态位 128 位的 AES 初始向量 AES_IV[127:0] = AESIV[95:0] || 12'b0 || 读地址[23:4]。其中，96 位的 AESIV[95:0]按照 AESIV2 || AESIV1 || AESIV0 的顺序组成。

3.4.21. AES 初始向量寄存器 x（FMC_AESIVx_MDF）（x = 0…2）

地址偏移：0x74 + 0x4 * x
复位值：0x0000 0000

该寄存器是AES初始向量高96位的修改值。

当LK位设置为0时，该寄存器才能被访问，只能按字（32位）访问。

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

位/位域	名称	描述
31:0	AESIV[31:0]	AES 初始向量配置位 128 位的 AES 初始向量 AES_IV[127:0] = AESIV[95:0] || 12'b0 || 读地址[23:4]。其中，96 位的 AESIV[95:0]按照 AESIV2 || AESIV1 || AESIV0 的顺序组成。 在初始向量写入 FMC_AESIV2_MDF 寄存器后，FMC_ASIV0/1/2_MDF 寄存器中的值都将被更新至 AES 初始向量区域中，且 BUSY 位自动置 1。当更新完成后，BUSY 位自动清 0。 **注意：**在写入 FMC_AESIV2_MDF 之前，用户需要确保没有编程、擦除或选项字节修改操作在进行。否则，FMC_AESIV2_MDF 寄存器无法被写入，更新操作也不会进行。

3.4.22. 产品 ID 寄存器 x（FMC_PIDx）（x = 0，1）

地址偏移：0x100 + 0x4 * x
复位值：0xXXXX XXXX

该寄存器只能按字（32位）访问。

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

r

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

4. 熔丝（EFUSE）

4.1. 简介

熔丝（EFUSE）作为一种非易失性存储单元存储了一些必需的系统参数。作为非易失性存储单元，熔丝的每一个比特位一旦从 0 被改写为 1，就无法恢复为 0。

4.2. 主要特性

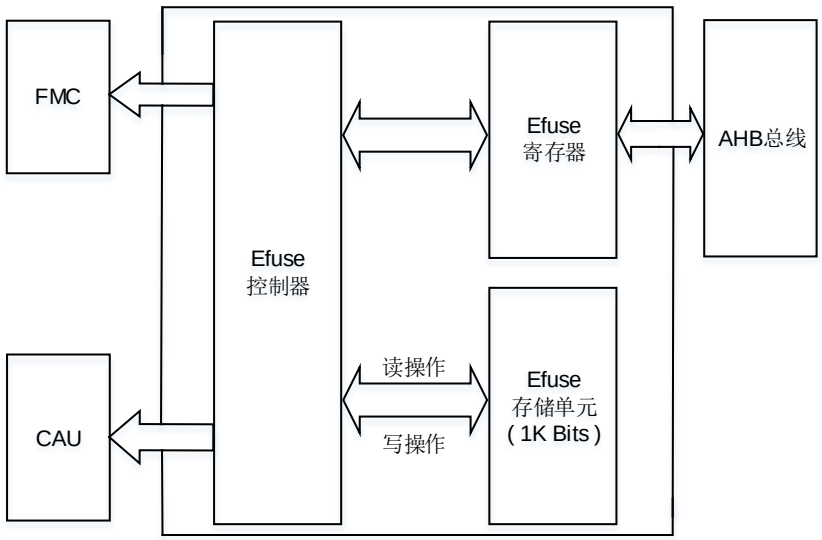
- 熔丝的存储单元大小为32*32比特；
- Double-bit冗余备份机制；
- 熔丝中的所有比特位不支持回退；
- 熔丝内容只能通过相应的寄存器读取；
- 熔丝的每个比特位只能被编程一次，软件必须避免比特位被重复编程；
- 编程操作电压：1.71~1.98V；
- 读操作电压：0.72~1.05V。

4.3. 功能说明

4.3.1. 模块框图

熔丝控制器实现了熔丝的读写操作逻辑，其中熔丝模块的存储单元共计 1K 比特。

图 4-1 熔丝控制器结构框图



熔丝采用了 double-bit 冗余备份机制，前 512 比特数据和后 512 比特数据互相备份，从而有效保证数据的正确性。当编程熔丝的第 n 比特时，熔丝控制器会同时编程第 n 及 n+512 比特。当读取熔丝的第 n 比特时，熔丝控制器会同时读取第 n 及 n+512 比特，并将读取的结果进行

或运算，最终将运算结果返回至参数寄存器中。以上过程全部由芯片内部进行处理，用户操作时只需访问前 512 比特即可，后 512 比特数据用户不可访问。

4.3.2. 熔丝内容简介

熔丝存储单元中存储了 5 个系统参数，不同的系统参数具有不同的位宽，并且每个系统参数都有各自的读写保护属性。

写保护属性如下所示：

- 用户控制段：
  存储在熔丝内的参数可以被多次修改，但一旦某一位被编程为1，硬件将自动禁止该位被再次编程。只有在SCRLK位为0时，用户才能修改寄存器高16位的内容。只有在UCLK位为0时，用户才能修改寄存器低16位的内容。但修改后的寄存器值不会被存储到熔丝内，除非成功执行了熔丝*写操作*。在执行完熔丝写操作后，用户需要电源复位（JTAGNSW及NDBG[1:0]位）或系统复位（除JTAGNSW及NDBG[1:0]位外其他位）才能将存储在熔丝内的参数加载到对应的寄存器中，且修改过的用户控制段只有在电源复位（JTAGNSW及NDBG[1:0]位）或系统复位（其他位）后才生效。

- MCU保留段：
  存储在熔丝内的参数可以被多次修改，但一旦某一位被编程为1，用户必须通过软件的方式去禁止该位被再次编程（当熔丝写操作时，将熔丝中已经被编程为1的位所对应的寄存器中的位清0）。用户可以修改寄存器里的内容，但修改后的寄存器值不会被存储到熔丝内，除非成功执行了熔丝*写操作*。在执行完熔丝写操作后，用户需要系统复位或熔丝*读操作*才能将存储在熔丝内的参数加载到对应的寄存器中，但是，其中修改后的DCRP_AREA_END[7:0]、DCRP_AREA_START[7:0]、DCRPLK、AESNCAU（AESNCAU仅适用于GD32H737_757_759）位只有在系统复位后才生效。

- 调试秘钥：
  存储在熔丝内的参数可以被多次修改，但一旦某一位被编程为1，用户必须通过软件的方式去禁止该位被再次编程（当熔丝写操作时，将熔丝中已经被编程为1的位所对应的寄存器中的位清0）。只有在DPLK位为0时，用户才能修改寄存器的内容，但修改后的寄存器值不会被存储到熔丝内，除非成功执行了熔丝*写操作*。在执行完熔丝写操作后，用户需要系统复位或熔丝*读操作*才能将存储在熔丝内的参数加载到对应的寄存器中，且修改过的调试秘钥只有在系统复位后才生效。

- AES秘钥：
  存储在熔丝内的参数可以被多次修改，但一旦某一位被编程为1，用户必须通过软件的方式去禁止该位被再次编程（当熔丝写操作时，将熔丝中已经被编程为1的位所对应的寄存器中的位清0）。只有在AESEN位为0时，用户才能修改寄存器的内容，但修改后的寄存器值不会被存储到熔丝内，除非成功执行了熔丝*写操作*。在执行完熔丝写操作后，用户需要系统复位，因为被修改的AES秘钥只有在系统复位后才生效。

- 用户数据段：
  存储在熔丝内的参数可以被多次修改，但一旦某一位被编程为1，用户必须通过软件的方式去禁止该位被再次编程（当熔丝写操作时，将熔丝中已经被编程为1的位所对应的寄存器中的位清0）。只有在UDLK位为0时，用户才能修改寄存器的内容，但修改后的寄存器

值不会被存储到熔丝内，除非成功执行了熔丝*写操作*。在执行完熔丝写操作后，用户需要系统复位或熔丝*读操作*将存储在熔丝内的参数加载到对应的寄存器中。

读保护属性如下所示：

- 用户控制段：
  寄存器可读。在系统复位后，除JTAGNSW, NDBG[1:0]位以外的其他位将恢复为从熔丝中读取的参数值。在电源复位后，JTAGNSW, NDBG[1:0]位的内容将恢复为从熔丝中读取的参数值。

- MCU保留段：
  寄存器可读。在系统复位后，寄存器的内容将恢复为从熔丝中读取的参数值。用户也可以通过配置控制寄存器来执行*读操作*，从熔丝中读取数据。

- 调试秘钥：
  当DPLK位为1，JTAGNSW位为1，且NDBG[1:0]位为2b'01或2b'11时，寄存器不可读。其他情况下寄存器可读。在系统复位后，寄存器的内容将恢复为从熔丝中读取的参数值。用户也可以通过配置控制寄存器来执行*读操作*，从熔丝中读取数据。

- AES秘钥：
  寄存器不可读。但用户可以通过AES秘钥CRC功能来验证已写入的AES秘钥的正确性。
  注意：用户必须将完整的16字节AES秘钥连续写入EFUSE_AES_KEYx寄存器，以确保CRC功能可以检查所有AES秘钥的内容。

- 用户数据段：

  寄存器可读。在系统复位后，寄存器的内容将恢复为从熔丝中读取的参数值。用户也可以通过配置控制寄存器来执行*读操作*，从熔丝中读取数据。

*表 4-1. 求系统参数*显示了熔丝中存储的系统参数详情。

表 4-1. 求系统参数

名称	位宽/字节	起始地址	写保护属性	读保护属性	描述
用户控制段	4B	10'd0	熔丝中参数可多次修改，但每个比特位不可回退、不可再编程。硬件将禁止已被编程过的位被再次编程。	系统复位后寄存器中对应位的内容将恢复为从熔丝中读取的参数值。	除 JTAGNSW 位，NDBG[1:0]位以外的其他位。详细内容请参考*用户控制寄存器（EFUSE_USER_CTL）*
				电源复位后寄存器中对应位的内容将恢复为从熔丝中读取的参数值。	JTAGNSW 位及 NDBG[1:0]位。详细内容请参考*用户控制寄存器（EFUSE_USER_CTL）*
MCU保留段	4B	10'd32	熔丝中参数可多次修改，但每个比特位不可回退、不可再编程。软件需要禁止已被编程过的位被再次编程。	系统复位后寄存器中对应位的内容将恢复为从熔丝中读取的参数值。且用户可以配置控制寄存器来执行读操作，从熔丝中读取数据。	MCU 保留段。详细内容请参考*MCU 保留寄存器（EFUSE_MCU_RSV）*
调试秘钥	8B	10'd64			当 JTAGNSW = 1，且 NDBG[1:0] = 2b'01 或 2b'11 时，该参数作为调试验证秘钥，用于调试服务。否则，该参数将被用作用户数据使用。详细内

名称	位宽/字节	起始地址	写保护属性	读保护属性	描述
					容请参考*调试秘钥寄存器（EFUSE_DPx）（x = 0，1）*
AES 秘钥	16B	10'd128		该参数用户不可读。但用户可以通过 AES 秘钥 CRC 功能来验证已写入的 AES 秘钥的正确性	加密固件所需的 AES 秘钥。详细内容请参考*固件 AES 秘钥寄存器 （EFUSE_AES_KEYx）（x = 0...3）*
用户数据段	16B	10'd256		系统复位后寄存器中对应位的内容将恢复为从熔丝中读取的参数值。且用户可以配置控制寄存器来执行读操作，从熔丝中读取数据。	用户自定义数据。详细内容请参考*用户数据寄存器 （EFUSE_USER_DATAx）（x = 0...3）*

注意：以上所有系统参数都是由用户设置的。熔丝的 10'd384~10'd511 位由系统使用，用户不可访问。

EFADDR[9:0]应设置为系统参数的起始地址，EFSIZE[4:0]应设置为系统参数的位宽。

读取操作：一次可以读取一个系统参数。禁止同时读取多个系统参数，否则会出现非法访问错误。然而，如果用户仅读取系统参数的一部分，即未完全读取系统参数，则不会发生非法访问错误，但它可能导致系统参数的读出数据不正确。用户应避免上述这种情况。

写入操作：写入操作的范围不能超过单个系统参数的地址范围，也不能同时写入多个系统参数，否则会出现非法访问错误。

4.3.3. 读操作

熔丝中的内容只能通过对应的寄存器来访问。

读取熔丝中系统参数时需要遵循以下操作步骤：

1. 将EFUSE_STAT寄存器中的RDIF位清零，并确保没有出现非法访问错误；
2. 将EFUSE_CTL寄存器中的EFRW位清零；
3. 在EFUSE_ADDR寄存器中填入需要读取的熔丝地址及大小；
4. 将EFUSE_CTL寄存器中EFSTR位置1；
5. 等待EFUSE_STAT寄存器中的RDIF位置位；
6. 读取对应的寄存器值。

当读取操作成功后，EFUSE_STAT 寄存器中的 RDIF 位会置位，如果 EFUSE_CTL 中的 RDIE 位置位，熔丝控制器会产生一个完成中断。

注意：熔丝对浪涌电流十分敏感，会影响读操作的结果。在掉电和上电的过程中，严禁对熔丝进行读操作，否则会导致无法预测的后果。

4.3.4. 写操作

熔丝中的内容只能通过对应的寄存器来写入，熔丝的写操作步骤如下：

1. 将EFUSE_STAT寄存器中的PGIF位清零，并确保没有出现非法访问错误；
2. 将EFUSE_CTL寄存器中的EFRW位置1；
3. 在EFUSE_ADDR寄存器中填入需要写入的熔丝地址及大小；
4. 在对应的寄存器中写入数据；
5. 将EFUSE_CTL寄存器中的EFSTR位置1；
6. 等待EFUSE_STAT寄存器中的PGIF位置位。

注意：如果对应的参数寄存器中的数据是全 0，在 EFSTR 位被设置为 1 后，熔丝将不会被执行写操作，PGIF 位将自动置 1。

当写操作完成后，EFUSE_STAT 寄存器中的 PGIF 位会置位，如果 EFUSE_CTL 中的 PGIE 位置位，熔丝控制器会产生一个中断。另外需要注意的是，数据写入的寄存器所对应的熔丝地址以及数据大小应与 EFUSE_ADDR 寄存器中的地址和大小相吻合，否则 IAERRIF 位将置位，如果 IAERRIE 位置位，熔丝控制器会产生一个中断。

注意：熔丝对浪涌电流十分敏感，会影响写操作的结果。在掉电和上电的过程中，严禁对熔丝进行写操作，否则会导致无法预测的后果。

4.3.5. AES 秘钥 CRC 功能

本模块中 CRC 计算采用标准 CRC-8-CCITT 算法，CRC 算法用于验证 EFUSE_AES_KEYx 寄存器中的值或熔丝中存储的 AES 秘钥值。

将 16 字节的 AES 秘钥连续写入偏移地址为 0x24、0x28、0x2C 和 0x30 的 EFUSE_AES_KEYx 寄存器后，硬件 CRC 模块将根据 EFUSE_AES_KEYx 寄存器中的 AES 秘钥值自动计算出相应的 CRC 校验码，并将计算结果存储到 EFUSE_CTL 寄存器中的 AES_KEY_CRC 位域中。此时，用户可以将硬件计算的 CRC 校验码与用户软件计算的软件 CRC 校验码进行比较。如果软件和硬件计算的校验码相同，则表明写入 EFUSE_AES_KEYx 寄存器的 16 字节的 AES 秘钥正确，否则写入寄存器的 AES 秘钥错误。

当系统复位后读取存储在熔丝中的 AES 秘钥时，硬件 CRC 模块将根据存储在熔丝中的 AES 秘钥值自动计算出相应的 AES 秘钥的 CRC 校验码，并将计算结果加载到 EFUSE_CTL 寄存器中的 AES_KEY_CRC 位域。此时，用户可以将硬件计算的 CRC 校验码与用户软件计算的 CRC 校验码进行比较。如果软件和硬件计算的校验码相同，则表明用户希望写入的 16 字节 AES 秘钥已成功正确写入熔丝，否则写入熔丝的秘钥错误。

注意：CRC 计算结果是在写完偏移地址 0x30 的 EFUSE_AES_KEY3 寄存器或系统复位读取熔丝完成后生成的。

4.3.6. EFUSE 中断

以下操作将会产生非法访问错误（IAERRIF）：

- 读写熔丝中参数时地址越界；

- UCLK 位为 1 且生效时，去写 EFUSE_USER_CTL 的低 16 位；
- SCRLK 位为 1 且生效时，去写 EFUSE_USER_CTL 的高 16 位；
- MCURSVLK 位为 1 且生效时，去写 EFUSE_MCU_RSV 的低 16 位；
- DCRPLK 位为 1 且生效时，去写 EFUSE_MCU_RSV 的高 16 位
- DPLK 位为 1 且生效时，去写 EFUSE_DPx；
- 当 DPLK 位为 1，JTAGNSW 位为 1，且 NDBG[1:0]位为 2b'01 或 2b'11 时，去读 EFUSE_DPx；
- AESEN 位为 1 且生效时，去写 EFUSE_AES_KEYx；
- UDLK 位为 1 且生效时，去写 EFUSE_USER_DATAx；
- 使用熔丝读操作去读取熔丝中的用户控制段；
- 使用熔丝读操作去读取熔丝中的 AES 秘钥；

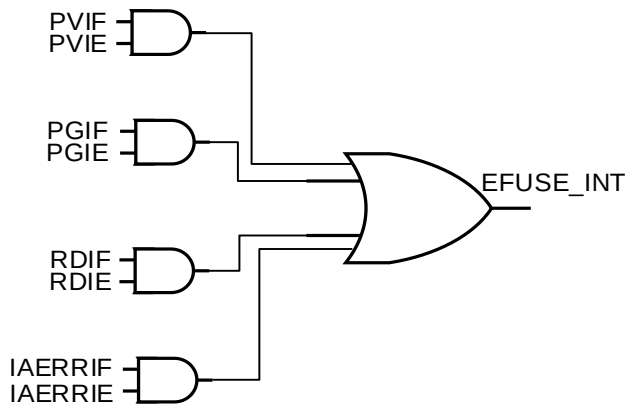
熔丝中断事件和标志如*表 4-2. EFUSE 中断请求*所示：

表 4-2. EFUSE 中断请求

中断事件	事件标志	使能控制位
编程电压设置错误中断	PVIF	PVIE
写操作完成	PGIF	PGIE
读操作完成	RDIF	RDIE
非法访问错误中断	IAERRIF	IAERRIE

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。因此在任何时候熔丝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 4-2. EFUSE 中断映射图



4.4. EFUSE 寄存器

EFUSE 基地址：0x4002 2800

4.4.1. 控制寄存器（EFUSE_CTL）

地址偏移：0x00
复位值：0x7E00 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AES_KEY_CRC								保留				PVIE	RDIE	PGIE	IAERRIE
r												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MPVEN	保留													EFRW	EFSTR
rw														rw	rw

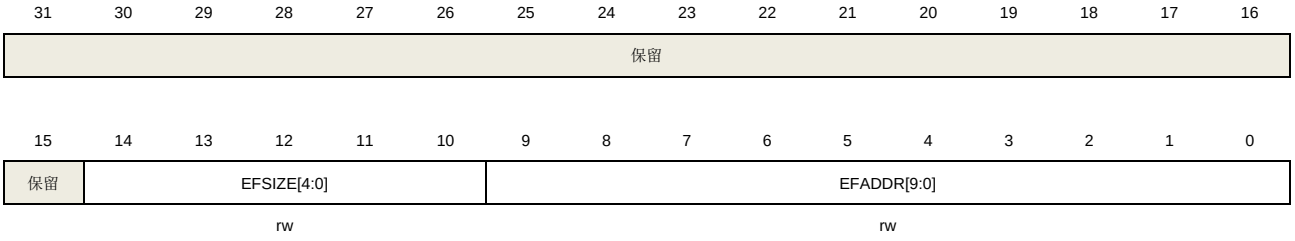
位/位域	名称	描述
31:24	AES_KEY_CRC	AES秘钥的8位CRC计算结果值 该位域用于验证EFUSE_AES_KEYx寄存器中的值或熔丝中存储的AES秘钥值。 CRC计算是使用标准CRC-8-CCITT算法X8+X2+X+1的8位校验和。 如果AESEN为0，则有两种情况将计算AES秘钥的CRC值，并将CRC计算结果存储到此位字段中： （1）将16字节的AES秘钥连续写入偏移地址为0x24、0x28、0x2C和0x30的EFUSE_AES_KEYx寄存器。CRC计算结果将在写入EFUSE_AES_KEY3寄存器（偏移地址0x30）后生成。 （2）系统复位后，由MCU从熔丝中自动读出AES值。CRC计算结果将在系统复位完成从熔丝中读出全部AES值后生成。
23:20	保留	必须保持复位值。
19	PVIE	编程电压设置错误中断使能位 0：失能编程电压设置错误中断 1：使能编程电压设置错误中断
18	RDIE	读操作完成中断使能位 0：失能读操作完成中断 1：使能读操作完成中断
17	PGIE	写操作完成中断使能位 0：失能写操作完成中断 1：使能写操作完成中断
16	IAERRIE	非法访问错误中断使能位 0：失能非法访问错误中断 1：使能非法访问错误中断

		当EFUSE_CTL寄存器中的EFSTR位为1时，该位不可写。
15	MPVEN	监控编程电压功能使能位 0：失能监控编程电压功能 1：使能监控编程电压功能 当EFUSE_CTL寄存器中的EFSTR位为1时，该位不可写。
14:2	保留	必须保持复位值。
1	EFRW	熔丝读写操作选择位 0：读熔丝内容 1：写熔丝内容 当EFUSE_CTL寄存器中的EFSTR位为1时，该位不可写。
0	EFSTR	发送熔丝读/写操作命令位 该位由软件置1，硬件清0 0：无影响 1：开始读/写操作

4.4.2. 地址寄存器（EFUSE_ADDR）

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
31:15	保留	必须保持复位值。
14:10	EFSIZE[4:0]	读/写熔丝数据大小 数据大小的单位是字节。 当EFUSE_CTL寄存器中的EFSTR位为1时，该位域不可写。
9:0	EFADDR[9:0]	读/写熔丝数据起始地址 EFADDR[9]必须设置为0，因为用户无法访问地址超过512的位的数据，否则EFUSE_STAT寄存器中的IAERRIF位将会置位。 当EFUSE_CTL寄存器中的EFSTR位为1时，该位域不可写。

4.4.3. 状态寄存器（EFUSE_STAT）

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											LDO_RDY	PVIF	RDIF	PGIF	IAERRIF
											r	r	r	r	r

位/位域	名称	描述
31:5	保留	必须保持复位值。
4	LDO_RDY	熔丝LDO准备完成信号 0：LDO未准备完成 1: LDO准备完成 **注意：**不论LDO旁路模式是否使能，该信号都有效。 该位在编程开始前由硬件自动置1，在编程完成后由硬件自动清0。
3	PVIF	编程电压设置错误标志位 0：编程电压设置在正确范围内 1：编程电压未设置在正确范围内
2	RDIF	读操作完成标志位 0：读操作未完成 1：读操作完成
1	PGIF	写操作完成标志位 0：写操作未完成 1：写操作完成
0	IAERRIF	非法访问错误标志位 0：未发生非法访问错误（越界或访问锁定参数） 1：发生非法访问错误（越界或访问锁定参数）

4.4.4. 状态标志清除寄存器（EFUSE_STATC）

地址偏移：0x10

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												PVIC	RDIC	PGIC	IAERRIC
												rc_w1	rc_w1	rc_w1	rc_w1

位/位域	名称	描述
31:4	保留	必须保持复位值。
3	PVIC	编程电压设置错误中断标志清除位 0：无影响 1：清除编程电压设置错误中断标志位
2	RDIC	读操作完成中断标志清除位 0：无影响 1：清除读操作完成中断标志位
1	PGIC	写操作完成中断标志清除位 0：无影响 1：清除写操作完成中断标志位
0	IAERRIC	非法访问错误中断标志清除位 0：无影响 1：清除非法访问错误中断标志位

4.4.5. 用户控制寄存器（EFUSE_USER_CTL）

地址偏移：0x14
复位值：0xXXXX XXXX，复位后装载熔丝存储单元中的值。

寄存器可读。只有当 SCRLK 位为 0 时，用户才能写入该寄存器的高 16 位。只有当 UCLK 位为 0 时，用户才能写入该寄存器的低位 16 位。但除非成功执行熔丝写操作，否则该寄存器中所有位的修改将不会存至熔丝中。

该寄存器只能按字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		SCR	SPC_H	SPC_L	JTAGNSW	NDBG		保留			UDLK	AESEN	UCLK	SCRLK	DPLK
		rw	rw	rw	rw	rw					rw	rw	rw	rw	rw

位/位域	名称	描述
31:24	SCR_AREA_END[7:0]	安全用户区域结束地址 该位域出厂值为0。 该位域包含了安全用户区域的最后的32K字节块。 安全用户区域在熔丝中以32K字节的粒度定义。 区域最后一个字节地址= (SCR_AREA_END[7:0] + 1) * 32768 - 1 + 0x0800_0000 如果SCR_AREA_END[7:0]及SCR_AREA_START[7:0]都为0，则安全用户区域未定义。 如果SCR_AREA_END[7:0]等于SCR_AREA_START[7:0]且不为0，整个主存储闪存块都是安全用户区域。

		如果SCR_AREA_END[7:0]小于SCR_AREA_START[7:0]，安全用户区域为空。 安全用户区域地址配置详见*表3-6. 安全用户区域配置*。
23:16	SCR_AREA_START[7:0]	安全用户区域起始地址 该位域出厂值为0。 该位域包含了安全用户区域的起始的32K字节块。 安全用户区域在熔丝中以32K字节的粒度定义。 区域最后一个字节地址= SCR_AREA_END[7:0] * 32768 - 1 + 0x0800_0000 如果SCR_AREA_END[7:0]及SCR_AREA_START[7:0]都为0，则安全用户区域未定义。 如果SCR_AREA_END[7:0]等于SCR_AREA_START[7:0]且不为0，整个主存储闪存块都是安全用户区域。 如果SCR_AREA_END[7:0]小于SCR_AREA_START[7:0]，安全用户区域为空。 安全用户区域地址配置详见*表3-6. 安全用户区域配置*。
15:14	保留	必须保持复位值。
13	SCR	安全模式使能 该位出厂值为0。 0：失能安全模式。 1：使能安全模式。 **注意：**只要该位或选项字节中的SCR位为1，安全模式就将启用。
12	SPC_H	将安全保护等级配置为保护等级高 该位出厂值为0。 如果熔丝中的SPC_H和SPC_L位都为1，则SPC为保护等级高。 安全保护等级配置详见*表3-4. SPC保护等级配置*。
11	SPC_L	将安全保护配置为保护等级低 该位出厂值为0。 **注意：**如果熔丝中的SPC_L设置为1，则禁止SPC保护等级低到无保护状态的降级。如果熔丝中的SPC_H和SPC_L位都为1，则SPC为保护等级高。 安全保护等级配置详见*表3-4. SPC保护等级配置*。
10	JTAGNSW	SW或JTAG调试器选择 该位出厂值为0。 0: SW 1: JTAG **注意：**当NDBG[1:0]选择为无调试功能时，JTAGNSW位无效，调试功能关闭。
9:8	NDBG[1:0]	调试权限设置 该位出厂值为0。 00：普通JTAG（仅在JTAGNSW为1时有效，否则为SW调试） 01：安全JTAG（仅在JTAGNSW为1时有效，否则为SW调试） 10~11：无调试功能（无论JTAGNSW取值，调试功能都关闭）
7:5	保留	必须保持复位值。
4	UDLK	EFUSE_USER_DATAx寄存器锁定位

		该位出厂值为0。 0：解锁EFUSE_USER_DATAx寄存器，寄存器内容可以被修改 1：锁定EFUSE_USER_DATAx寄存器，寄存器内容不可以被修改
3	AESEN	EFUSE_AES_KEYx寄存器锁定及AES加解密功能使能位 该位出厂值为0。 0：失能AES加解密功能，EFUSE_AES_KEYx寄存器可以写数据 1：使能AES加解密功能并锁定EFUSE_AES_KEYx寄存器，寄存器内容不可改写
2	UCLK	EFUSE_USER_CTL寄存器低16位锁定位 该位出厂值为0。 0：解锁EFUSE_USER_CTL寄存器中的低16位，寄存器低16位可以被修改 1：锁定EFUSE_USER_CTL寄存器中的低16位，寄存器低16位不可被修改 UCLK位置1后，EFUSE_USER_CTL寄存器中的其他锁定位将无法进行修改，用户需要对该位谨慎操作。 **注意：**当UCLK位为1时，如果想要修改熔丝中用户控制段的高16位，起始地址必须设置为10'd16（此时EFSIZE[4:0] = 1或2）或10'd24（此时EFSIZE[4:0]只能设置为1）。否则会产生非法访问错误。
1	SCRLK	安全区域地址锁存位 该位出厂值为0。 0：解锁EFUSE_USER_CTL寄存器中的高16位，寄存器高16位可以被修改 1：锁定EFUSE_USER_CTL寄存器中的高16位，寄存器高16位不可被修改 **注意：**当SCRLK位为1时，如果想要修改熔丝中用户控制段的低16位，起始地址必须设置为10'd0（此时EFSIZE[4:0] = 1或2）或10'd8（此时EFSIZE[4:0]只能设置为1）。否则会产生非法访问错误。
0	DPLK	EFUSE_DPx寄存器锁定位 该位出厂值为0。 0：解锁EFUSE_DPx寄存器，寄存器内容可读可写。 1：锁定EFUSE_DPx寄存器，寄存器内容不可写。该位为1时，只有当JTAGNSW位为1，且NDBG[1:0]位为2b'01或2b'11时，寄存器不可读。其他情况下，寄存器可读。

4.4.6. MCU 保留寄存器（EFUSE_MCU_RSV）

地址偏移：0x18
复位值：0xXXXX XXXX，复位后装载熔丝存储单元中的值。

寄存器可读。只有当DCRPLK位为0时，用户才能写入该寄存器的高16位。只有当MCURSVLK位为0时，用户才能写入该寄存器的低位16位。但除非成功执行熔丝写操作，否则该寄存器中所有位的修改将不会存至熔丝中。

该寄存器只能按字（32位）访问。

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

MCU_RSV[5:0]	DCRPLK	MCURSVLK	VFIMG	DISLFI	保留	AESNCAU
rw	rw	rw	rw	rw		rw

位/位域	名称	描述
31:24	DCRP_AREA_END[7:0]	DCRP区域结束地址 该位域出厂值为0。 该位域包含了DCRP区域的最后的32K字节块。 DCRP区域在熔丝中以32K字节的粒度定义。 区域最后一个字节地址= (DCRP_AREA_END[7:0] + 1) * 32768 - 1 + 0x0800_0000 如果DCRP_AREA_END[7:0]及DCRP_AREA_START[7:0]都为0，则DCRP区域未定义。 如果DCRP_AREA_END[7:0]等于DCRP_AREA_START[7:0]且不为0，整个主存储闪存块都是DCRP区域。 如果DCRP_AREA_END[7:0]小于DCRP_AREA_START[7:0]，DCRP区域为空。 DCRP地址配置详见*表3-5. DCRP区域配置*。
23:16	DCRP_AREA_START[7:0]	DCRP区域起始地址 该位域出厂值为0。 该位域包含了DCRP区域的起始的32K字节块。 DCRP区域在熔丝中以32K字节的粒度定义。 如果DCRP_AREA_END[7:0]及DCRP_AREA_START[7:0]都为0，则DCRP区域未定义。 如果SCR_AREA_END[7:0]等于SCR_AREA_START[7:0]且不为0，整个主存储闪存块都是DCRP区域。 如果SCR_AREA_END[7:0]小于SCR_AREA_START[7:0]，DCRP访问区域为空。 DCRP地址配置详见*表3-5. DCRP区域配置*。
15:10	MCU_RSV[5:0]	熔丝MCU保留数据 该位域出厂值为0。
9	DCRPLK	DCRP区域地址锁存位 该位出厂值为0。 0：解锁EFUSE_MCU_RSV寄存器中的高16位，寄存器高16位可以被修改 1：锁定EFUSE_MCU_RSV寄存器中的高16位，寄存器高16位不可被修改 **注意：**当DCRPLK位为1时，如果想要修改熔丝中MCU保留段的低16位，起始地址必须设置为10'd32（此时EFSIZE[4:0] = 1或2）或10'd40（此时EFSIZE[4:0]只能设置为1）。否则会产生非法访问错误。
8	MCURSVLK	EFUSE_MCU_RSV寄存器低16位锁定位 该位出厂值为0。 0：解锁EFUSE_MCU_RSV寄存器中的低16位，寄存器低16位可以被修改 1：锁定EFUSE_MCU_RSV寄存器中的低16位，寄存器低16位不可被修改 MCURSVLK位置1后，EFUSE_MCU_RSV寄存器中的其他锁定位将无法进行修改，用户需要对该位谨慎操作。

		注意：当MCURSVLK位为1时，如果想要修改熔丝中MCU保留段的高16位，起始地址必须设置为10'd48（此时EFSIZE[4:0]= 1或2）或10'd56（此时EFSIZE[4:0]需设置为1）。否则会产生非法访问错误。
7	VFIMG	验证固件镜像使能位 该位出厂值为0。 0：失能固件镜像认证功能 1：使能固件镜像认证功能
6	DISLFI	授权固件安装功能设置 该位出厂值为0。 0: 使能授权固件安装 1: 失能授权固件安装
5:1	保留	必须保持复位值。
0	AESNCAU	用于CAU的AES秘钥配置位（AESNCAU仅适用于GD32H737_757_759） 该位出厂值为0。 0：AES秘钥用于CAU 1：AES秘钥不用于CAU

4.4.7. 调试秘钥寄存器（EFUSE_DPx）（x = 0，1）

地址偏移：0x1C + 0x4 * x
复位值：0xXXXX XXXX，复位后装载熔丝存储单元中的值。

当 JTAGNSW = 1，且 NDBG[1:0] = 2b’01 或 2b’11 时，该参数作为调试验证秘钥，用于调试服务。否则，该参数将被用作用户数据使用。

作为调试密钥时：仅当 DPLK 位为 0 时，寄存器才可读。仅当 DPLK 位为 0 时，寄存器可写。但除非成功执行熔丝写操作，否则该寄存器中所有位的修改将不会存至熔丝中。

作为用户数据时：无论 DPLK 位为 0 或 1，寄存器都可读。仅当 DPLK 位为 0 时，寄存器可写。但除非成功执行熔丝写操作，否则该寄存器中所有位的修改将不会存至熔丝中。

该寄存器只能按字（32 位）访问。

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

位/位域	名称	描述
31:0	DP[31:0]	熔丝中调试秘钥字段值 该位域出厂值为0。

4.4.8. 固件 AES 秘钥寄存器 （EFUSE_AES_KEYx）（x = 0…3）

地址偏移：0x24 + 0x4 * x
复位值：0xXXXX XXXX，复位后装载熔丝存储单元中的值。

寄存器不可读。只有当 AESEN 位为 0 时，寄存器可写。但除非成功执行熔丝写操作，否则该寄存器中所有位的修改将不会存至熔丝中。

该寄存器只能按字（32 位）访问。

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

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

位/位域	名称	描述
31:0	AESKEY[31:0]	熔丝中AES秘钥字段值 该位域出厂值为0。 用户必须将完整的16字节AESKEY[127:0]连续写入EFUSE_AES_KEYx寄存器。 该寄存器只能按照字（32位）写访问，每个寄存器中的4个字节是按照由低字节到高字节的顺序存储的（即AESKEY的低字节对应寄存器的低位）。 同时，AESKEY[31:0]写入EFUSE_AES_KEY0寄存器（偏移地址0x24）， AESKEY[63:32]写入EFUSE_AES_KEY1寄存器（偏移地址0x28）， AESKEY[95:64]写入EFUSE_AES_KEY2寄存器（偏移地址0x2C）， AESKEY[127:96]写入EFUSE_AES_KEY3寄存器（偏移地址0x30）。 CRC计算结果将在写入EFUSE_AES_KEY3寄存器（偏移地址0x30）后生成。

4.4.9. 用户数据寄存器 （EFUSE_USER_DATAx）（x = 0…3）

地址偏移：0x34 + 0x4 * x
复位值：0xXXXX XXXX，复位后装载熔丝存储单元中的值。

寄存器可读。只有当 UDLK 位为 0 时，寄存器可写。但修改后的寄存器值不会存储在熔丝中，除非成功执行熔丝写操作。

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
USERDATA[31:16]															
rw															

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

位/位域	名称	描述

31:0	USERDATA[31:0]	熔丝中用户自定义数据字段值 该位域出厂值为0。

5. 电源管理单元（PMU）

5.1. 简介

功耗设计是 GD32H73x_75x 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 5-1. 电源域概览*所示，GD32H73x_75x 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。在嵌入的 LDO 和低功率开关电源降压稳压器（SMPS 降压稳压器），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 V_{BAT} 引脚，此时备份域由 V_{BAT} 引脚（电池）供电。外设供电调节 USB 的调节器。

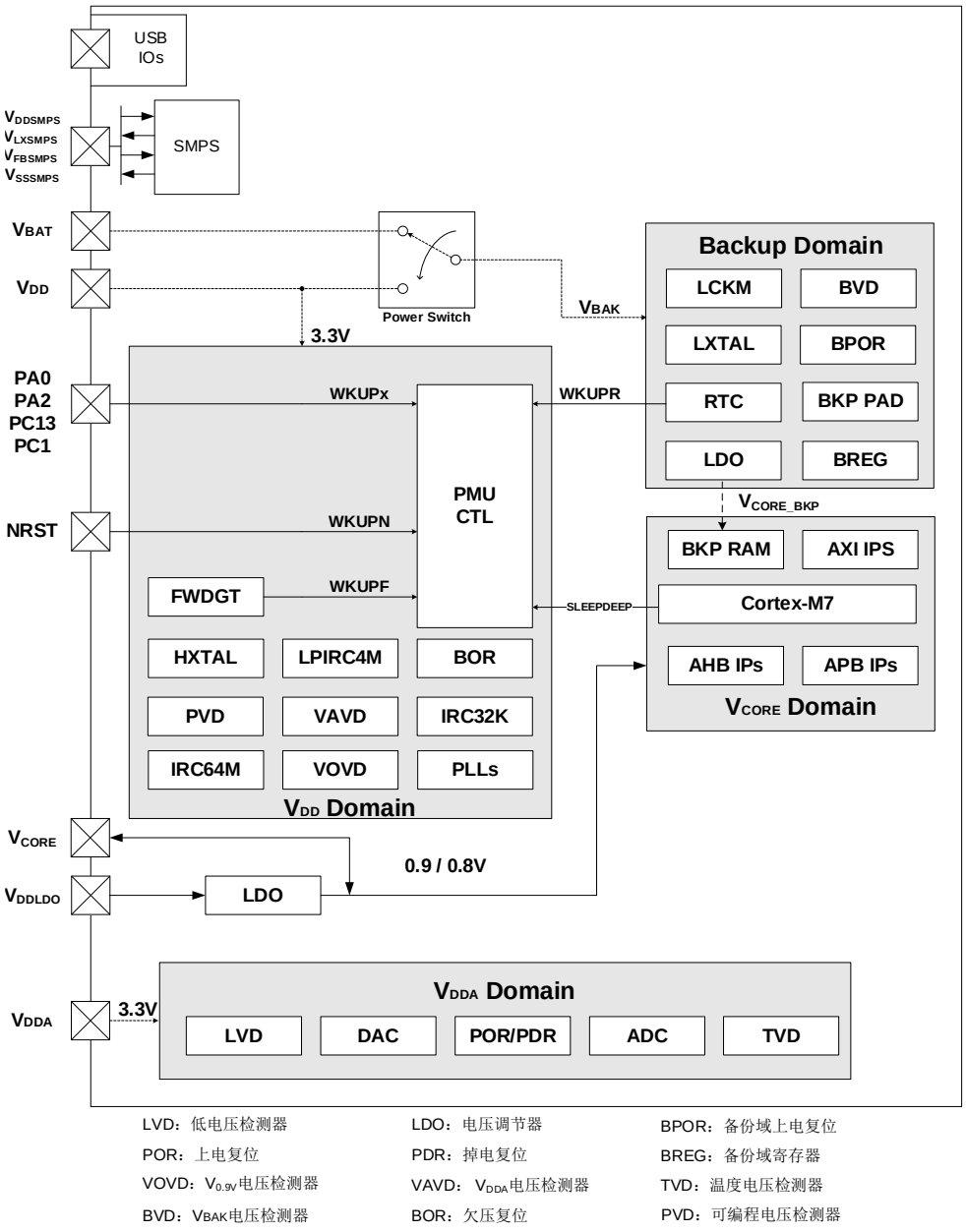
5.2. 主要特征

- 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V_{CORE}电源域提供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。
- LDO输出电压用于节约能耗。
- USB电源调节器。
- 供电监控：POR / PDR监控、BOR监控、LVD监控、V_{DDA}电压检测和监控（VAVD）、V_{BAK}阈值监测、温度阈值监测。
- V_{BAT}电池充电管理，工作模式管理，电压输出控制，低功耗模式管理。
- 低功率开关电源降压稳压器（SMPS降压稳压器）。
- 支持cpu 进入deepsleep或者sleep 模式的状态输出。

5.3. 功能说明

*图 5-1. 电源域概览*提供了 PMU 及相关电源域的内部结构框图。

图5-1. 电源域概览



注意：SMPS 供电不是在所有的设备上都支持，具体描述可参考数据手册。

5.3.1. 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）、BVD（V_{BAK} 电压检测器）、LCKM（LXTAL 时钟监视器）、BPOR（备份域上电复位）和 BREG，以及 PC13 至 PC15 共 3 个 BKP PAD。为了确保备份域中寄存器的内容及 RTC 正常工作，当 V_{DD} 关闭时，V_{BAT} 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 V_{BAT} 引脚通过 100nF 的外部陶瓷去耦电容连接到 V_{DD} 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RTCDIV[5:0]（位于RCU_CFG0寄存器中）位域控制的高速外部晶体振荡器（HXTAL）时钟分频。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®-M7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）。

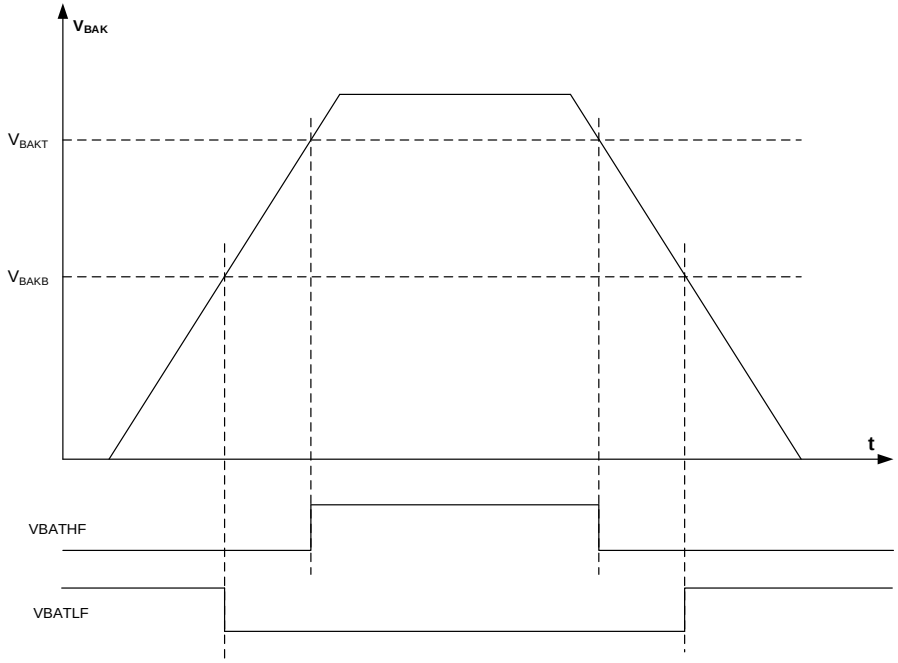
V_{DD} 可以通过一个内部电阻给外部电池充电。通过配置 PMU_CTL2 寄存器中 VCRSEL 位，可以选择内部电阻 5K 欧姆或 1.5K 欧姆用于外部 V_{BAT} 电池充电。将 PMU_CTL2 寄存器中 VCEN 位置 1 可以使能 V_{BAT} 电池充电。在 BKP only 模式，V_{BAT} 电池充电不可用。

注意：在 BKP only 模式下，V_{DD} 掉电，备份域由 V_{BAT} 引脚供电。

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} 或低于 V_{BAKB}，则标志位 VBATHF / VBATLF 将设置，该功能仅在置位 BKPVSEN 位时可用。*图 5-2. 备用域电压阈值的波形*，显示了备用域电压阈值监测。

图5-2. 备用域电压阈值的波形



5.3.2. VDD / VDDA 电源域

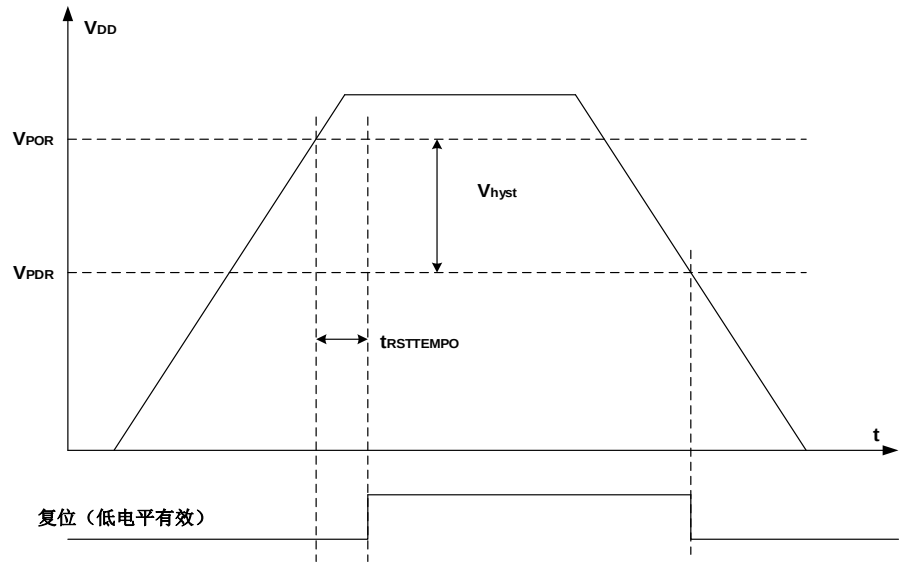
VDD / VDDA 域包括 VDD 域和 VDDA 域两部分。VDD 域包括 HXTAL（高速外部晶体振荡器）、FWDGT（独立看门狗定时器）、BOR（欠压复位）、LPIRC4M（内部 4MHz RC 振荡器）、IRC64M（内部 64M RC 振荡器）、IRC32K（内部 32KHz RC 振荡器）、PVD（可编程电压检测器）、PLLs（锁相环）、VOVD（VCORE 电压检测器）、VAVD（VDDA 电压检测器）和除 PC13、PC14 和 PC15 之外的所有 PAD 等等。VDDA 域包括 ADC / DAC（AD / DA 转换器）、LVD（低电压检测器）、TVD（温度电压检测器）、POR / PDR（上电 / 掉电复位）等等。

VDD 域

为 VCORE 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（VCORE 全供电状态和低功耗状态）、深度睡眠模式和待机模式（关闭状态）。

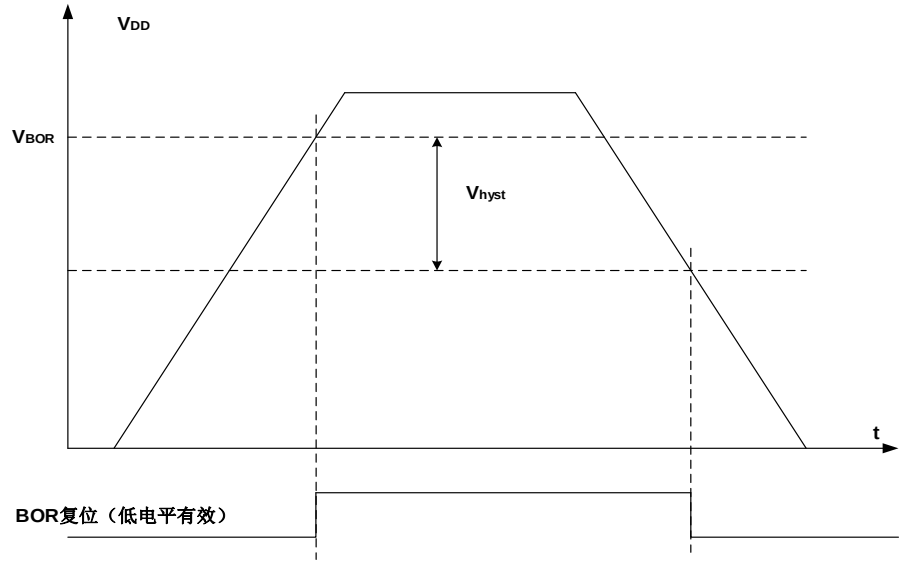
POR / PDR（上电 / 掉电复位）电路检测 VDD 并在电压低于特定阈值时产生电源复位信号复位除备份域之外的整个芯片。*图 5-3. 上电 / 掉电复位波形图*显示了供电电压和电源复位信号之间的关系。VPOR 表示上电复位的阈值电压，VPDR 表示掉电复位的阈值电压。迟滞电压 Vhyst 值可以参考芯片数据手册。

图5-3. 上电 / 掉电复位波形图



BOR 电路检测 VDD 并在电压低于选项字节的 BOR_TH 定义的阈值且该阈值不为 0b00（BOR_TH=0b00，BOR 功能关闭）时产生电源复位信号复位除备份域之外的整个芯片。不管选项字节 BOR_TH 的值是否为 0b00，POR / PDR（上电 / 掉电复位）电路会一直处于检测状态。*图 5-4. 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 Vhyst 值可以参考芯片数据手册。

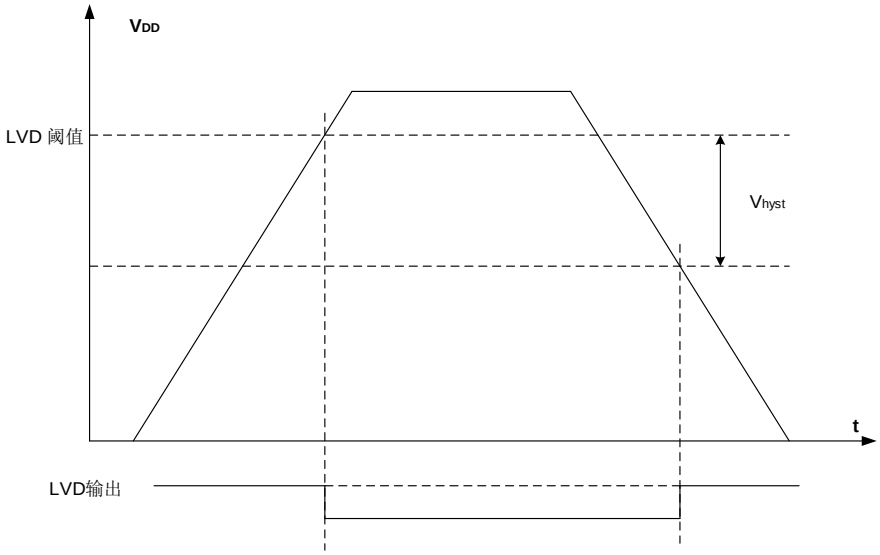
图5-4. BOR波形图



V_{DDA} 域

LVD 的功能是检测 V_{DD} 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 5-5. LVD 阈值波形图*显示了 V_{DD} 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} 值可以参考芯片数据手册。

图 5-5. LVD 阈值波形图



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

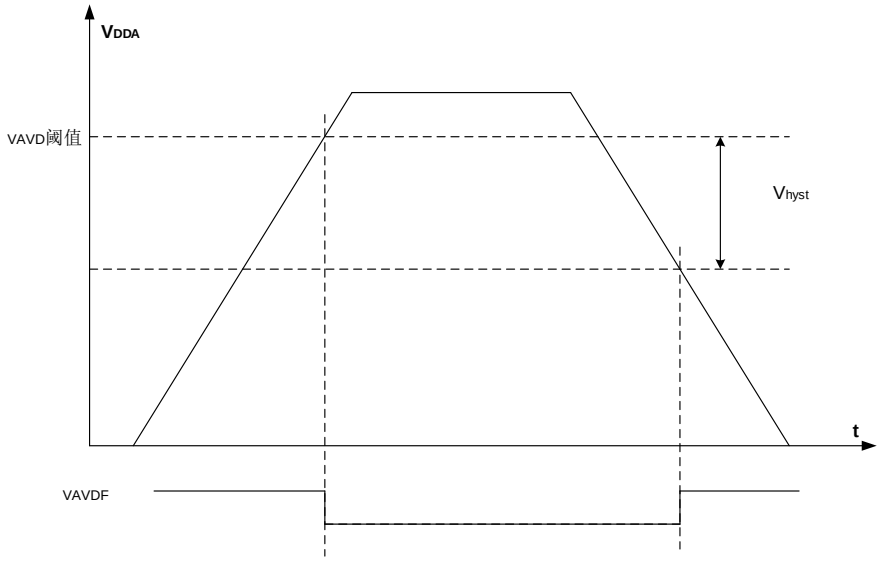
为提高 ADC 和 DAC 的精度，可将独立的外部参考电压连接至 ADC / DAC 引脚 VREFP / VREFN。根据不同的封装，VREFP 可被连接至 V_{DDA} 引脚，或者外部参考电压，外部参考电压的范围请参考*表 20-2. ADC 输入引脚定义*和*表 21-1. DAC 引脚*。VREFN 须被连接至 V_{SSA} 引脚。VREFP 引脚仅存在于不小于 100-pin 的封装上，而在更少引脚的封装上不存在，因其内部已经连接至 V_{DDA}，VREFN 内部则直接连接至 V_{SSA}。

V_{DDA} 阈值电压监测

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 16。如果通过 EXTI 寄存器使能，可以产生一个中断。

*图 5-6. VAVD 阈值监测波形图*显示了 VAVD 门限与 VAVDF 之间的关系。

图 5-6. VAVD 阈值监测波形图

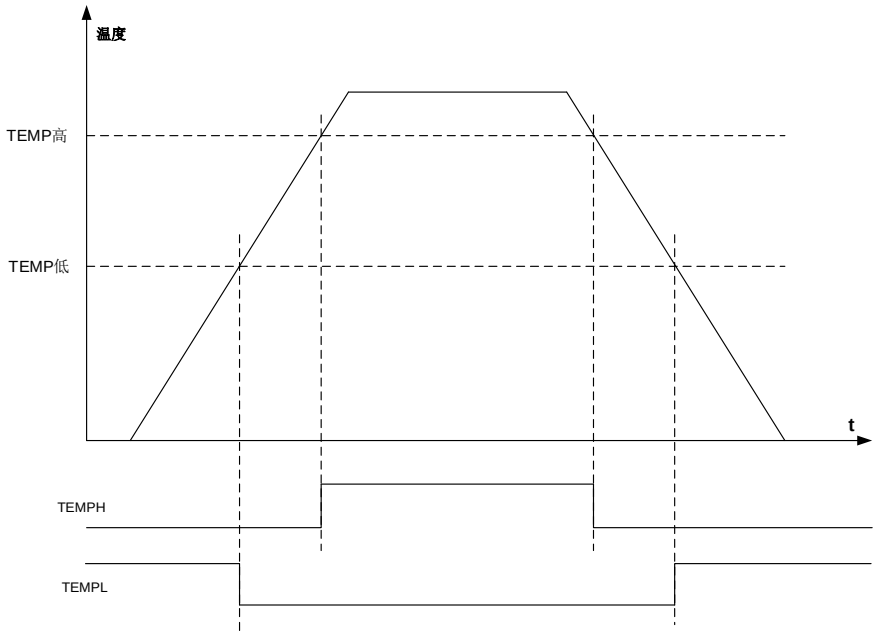


温度电压阈值监测

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 TEMPH 和 TEMPL 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 VBTMEN 位置位，温度阈值监测才有效。

TEMPH 和 TEMPL 唤醒中断可用于 RTC 触发信号

图 5-7. 温度阈值监测波形图



5.3.3. V_{CORE} 电源域

主要功能包括 Cortex®-M7 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。

V_{CORE} 电源域供电

使用 SMPS 降压稳压器和 LDO，可以设置 V_{CORE} 电源域的供电电源。不同配置可提供四种有效的 V_{CORE} 电源域供电模式。

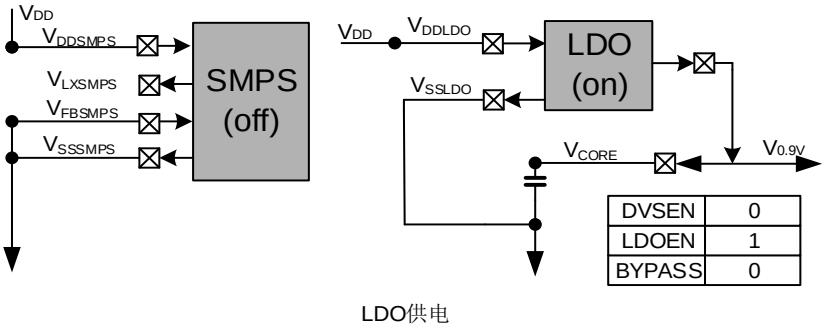
- 无配置的供电模式（默认供电模式）

复位后，DVSEN 位为 0b1。此时，SMPS 降压稳压器打开，工作在正常模式下，工作电压为 1.0V，SMPS 降压稳压器可以为 LDO 供电；LDOEN 位为 0b1，LDO 为开启状态，并为 V_{CORE} 电源域供电，供电电压由 LDOVS[2:0]位域控制；BYPASS 位为 0b0，V_{CORE} 电源域不由 V_{CORE} 供电（V_{CORE} 供电是直接外部供电）。

- LDO 供电模式

进入该供电模式的配置方式为：DVSEN 位为 0b0，此时 SMPS 降压稳压器为关闭状态；LDOEN 位为 0b1，LDO 为开启状态，并为 V_{CORE} 电源域供电，供电电压由 LDOVS[2:0]位域控制，LDO 的工作模式与系统的低功耗模式一致；BYPASS 位为 0b0，V_{CORE} 电源域不由 V_{CORE} 供电（V_{CORE} 供电是直接外部供电）。*图 5-8. LDO 供电 VCORE 电源域*显示了这种供电方式。

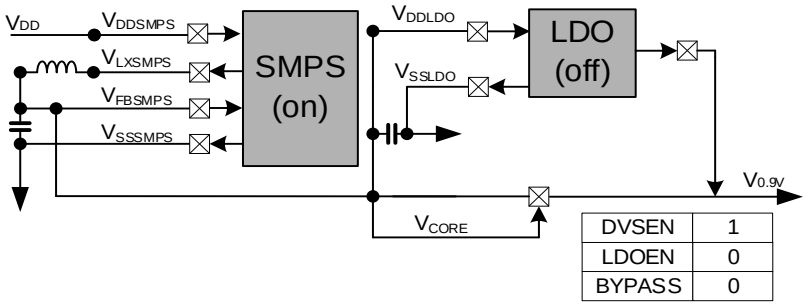
图 5-8. LDO 供电 V_{CORE} 电源域



- SMPS 供电模式

进入该供电模式的配置方式为：DVSEN 位为 0b1，此时 SMPS 降压稳压器为开启状态，并为 V_{CORE} 电源域供电，供电电压由 LDOVS[2:0]位域控制，SMPS 的工作模式与系统的低功耗模式一致；LDOEN 位为 0b0，LDO 为关闭状态；BYPASS 位为 0b0，V_{CORE} 电源域不由 V_{CORE} 供电（V_{CORE} 供电是直接外部供电）。*图 5-9. SMPS 供电 VCORE 电源域*显示了这种供电方式。

图 5-9. SMPS 供电 V_{CORE} 电源域

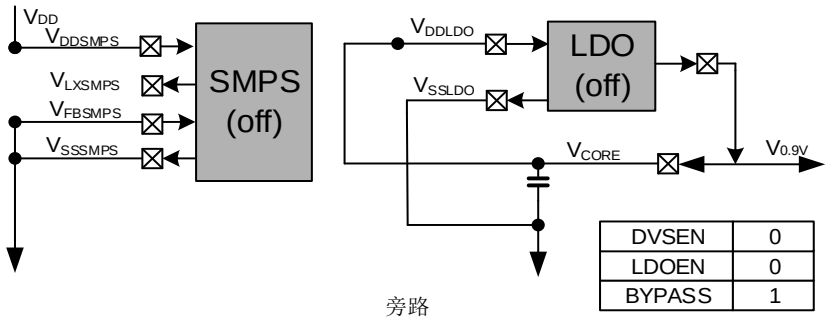


SMPS供电

■ 旁路模式

进入该供电模式的配置方式为：DVSEN 位为 0b0，此时 SMPS 降压稳压器为关闭状态；LDOEN 位为 0b0，LDO 为关闭状态；BYPASS 位为 0b1，V_{CORE} 电源域由 V_{CORE} 供电（V_{CORE} 供电是直接外部供电）。*图 5-10.旁路*显示了这种供电方式

图 5-10.旁路



旁路

表 5-1 供电模式

模式	供电配置	DVSEN	LDOEN	LDOVS	BYPASS
0	无配置的供电模式（默认供电模式）	1	1	0b010	0
1	LDO 供电模式	0	1	0b000-101	0
2	SMPS 供电模式	1	0	0b000-0b101	0
6	旁路模式	0	0	x	1

注意：除上述有效组合外，其它 DVSEN、LDOEN、BYPASS 位或位值的配置组合均无效。V_{CORE} 电源域的电源状态在复位后保持不变（无配置的供电模式）。

注意：最大工作频率与供电电压有关，具体请参考数据手册。

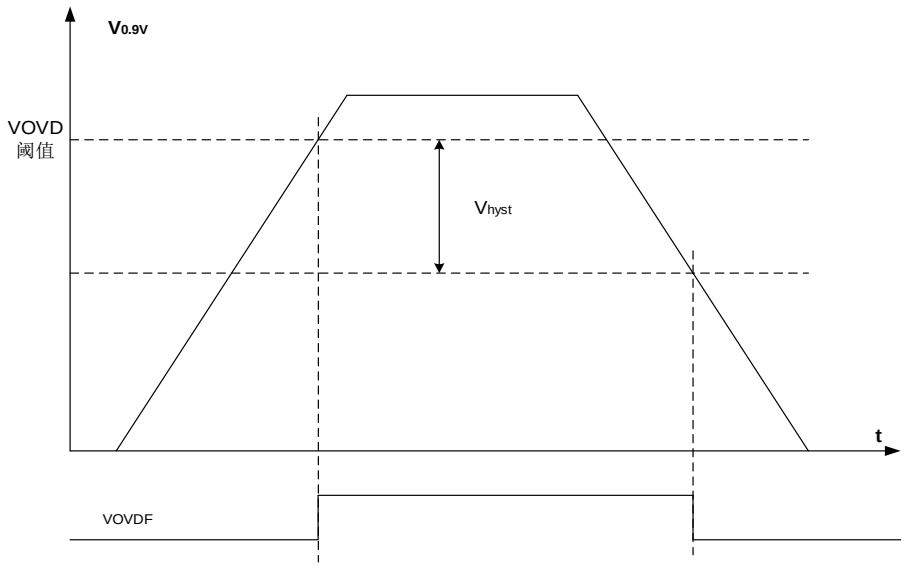
注意：SMPS 供电不是在所有的设备上都支持，具体描述可参考数据手册。

注意：使用旁路模式时，具体请参考 **AN225 GD32H7xx 电源旁路模式使用指南**。

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} 电源域出现过压，VOVDF 将置位。

图 5-11. VOVD 波形



5.3.4. USB 电源域

GD32H73x_75x系列USB内部集成了一个稳压器，用户可以选择使能该稳压器，将$V_{DD50USB}$引脚连接5 V电源为USB模块提供电源，如*图5-12. USB稳压器供电时连接示意图*所示；或旁路该稳压器，将$V_{DD33USB}$引脚连接3.3 V电源为USB模块提供电源，如*图5-13. USB稳压器旁路时连接示意图*所示。

图 5-12. USB 稳压器供电时连接示意图

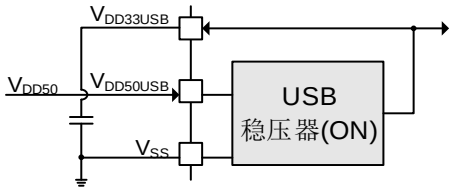
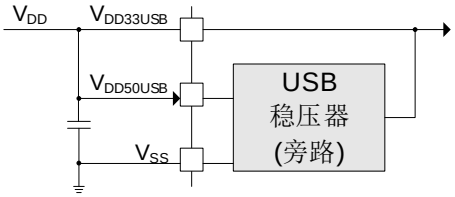


图 5-13. USB 稳压器旁路时连接示意图



注意：当 $V_{DD} \geq 3V$ 时，USB 电源可使用 USB 稳压器供电模式或旁路模式；当 $V_{DD} < 3V$ 时，USB 电源只能使用 USB 稳压器供电模式。

5.3.5. 省电模式

系统复位或电源复位后，GD32H73x_75x 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 和 PCLK2），关闭未使用的外设的时钟或者通过 PMU_CTL3 寄存器的 LDOVS[2:0]位配置 LDO 的输出电压，LDOVS[2:0]只有在 PLL 未使能的时候才能配置。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。

在系统复位、上电复位或从待机模式唤醒产生复位后，MCU 进入普通运行模式，对所有时钟无影响，LDO 工作在 V_{CORE} 模式。

睡眠模式

睡眠模式与 Cortex®-M7 的 SLEEPING 模式相对应。在睡眠模式下，仅关闭 Cortex®-M7 的时钟。如需进入睡眠模式，只要清除 Cortex®-M7 系统控制寄存器中的 SLEEPDEEP 位，并执行一条 WFI 或一条 SEV 指令与两条 WFE 指令。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 SEV 和 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®-M7 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®-M7 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®-M7 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，LPIRC4M、IRC64M、HXTAL 及 PLLs 也全部被禁用。进入深度睡眠模式之前，先将 Cortex®-M7 系统控制寄存器的 SLEEPDEEP 位置 1，再将 PMU_CTL0 寄存器的 LPMOD 位配置为 0b1，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®-M7 技术手册）。

注意：如果上电或者从待机模式唤醒，在进入深度睡眠模式之前需要等待一段时间。

待机模式

待机模式是基于 Cortex®-M7 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 LPIRC4M、IRC64M、HXTAL 和 PLLs 也会被关闭。进入待机模式前，先将 Cortex®-M7 系统控制寄存器的 SLEEPDEEP 位置 1，再将 PMU_CTL0 寄存器的 LPMOD 位域配置为 0b1，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（闹钟和侵入检测），FWDGT 复位，LCKMD，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®-M7 将从 0x00000000 地址开始执行指令代码。

表5-2. 节电模式总结

模式	描述	LDO 状态	进入指令	唤醒	唤醒后模式	唤醒延时
睡眠	仅关闭 CPU 时钟	LDO 开启	SLEEPDEEP = 0，在运行模式下执行 WFI 或 SEV 和 WFE	若通过 WFI 进入，则任何中断均可唤醒； 若通过 SEV 和 WFE 进入，则任何事件（或 SEVONPEND=1 时的中断）均可唤醒	普通运行模式	-
深度睡眠	1、关闭 V_{CORE} 电源域的所有时钟 2、关闭 LPIRC4M、IRC64M、HXTAL 和 PLLs	LDO 开启	SLEEPDEEP = 1，LPMOD = 0，执行 WFI 或 WFE	若通过 WFI 进入，来自 EXTI 的任何中断可唤醒；若通过 WFE 进入，来自 EXTI 的任何事件（或 SEVONPEND=1 时的中断）可唤醒	普通运行模式	LPIRC4M / IRC64M（由 DSPWUSSEL 确认）唤醒时间+ Flash 唤醒时间
待机	1、关闭 V_{CORE} 电源域的所有时钟 2、关闭 LPIRC4M、IRC64M、HXTAL 和 PLLs	LDO 关闭	SLEEPDEEP = 1，LPMOD = 1，执行 WFI 或 WFE	1、NRST 引脚 2、WKUP 引脚 3、FWDGT 复位 4、RTC（闹钟和侵入检测） 5、LCKMD	普通运行模式	IRC64M 唤醒时间+LDO 唤醒时间+ Flash 唤醒时间
BKP only	V_{DD} 域 / V_{CORE} 域全部掉电	LDO 关闭	V_{DD} 关闭	V_{DD} 开启	普通运行模式	V_{DD} 上电序列

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

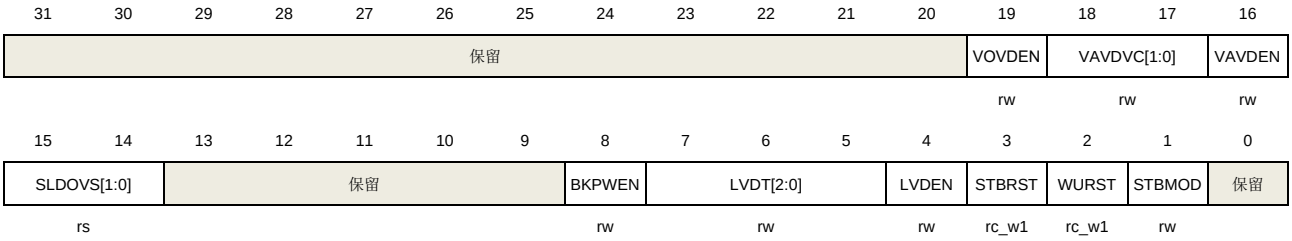
5.4. PMU 寄存器

PMU 基地址：0x5800 5800

5.4.1. 控制寄存器 0（PMU_CTL0）

地址偏移：0x00
复位值：0x0000 8000（从待机模式唤醒后复位）

该寄存器可以按字（32位）访问。



位/位域	名称	描述
31:20	保留	必须保持复位值。
19	VOVDEN	V_{CORE}外设电压检测器使能位 该位由软件置位和清除。 0：失能V_{CORE}外设电压检测器 1：使能V_{CORE}外设电压检测器
18:17	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1.7V 01：配置V_{DDA}模拟电压检测器电压等级为2.1V 10：配置V_{DDA}模拟电压检测器电压等级为2.5V 11：配置V_{DDA}模拟电压检测器电压等级为2.8V
16	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
15:14	SLDOVS[1:0]	Deep-sleep模式电压选择 这些位控制Deep-sleep模式时V_{CORE}电压值，以便在性能和功耗之间实现最佳的平衡。 00：SLDOVS设置电压为0.6V 01：SLDOVS设置电压为0.7V 10：SLDOVS设置电压为0.8V（默认） 11：SLDOVS设置电压为0.9V
13:9	保留	必须保持复位值。

8	BKPWEN	备份域写使能 0：禁止对备份域寄存器的写访问。 1：允许对备份域寄存器的写访问。 复位之后，任何对备份域寄存器的写访问都将被禁止。如需对备份域寄存器做写访问，需先将该位置 1。
7:5	LVDT[2:0]	低电压检测器阈值 000: 2.1V 001: 2.3V 010: 2.4V 011: 2.6V 100: 2.7V 101: 2.9V 110: 3.0V 111：PB7 输入模拟电压（与 0.8V 进行比较） **注意：**测量值请参考数据手册。
4	LVDEN	低电压检测器使能 0：关闭低电压检测器。 1：开启低电压检测器 **注意：**当 SYSCFG_LKCTL 寄存器里的 LVD_LOCK 位被置 1 时，LVDEN 和 LVDT[2:0] 仅可读。
3	STBRST	待机标志复位 0：无影响 1：复位待机标志 读该位，始终返回 0。
2	WURST	唤醒标志复位 0：无影响 1：复位唤醒标志 读该位，始终返回 0。
1	STBMOD	待机模式选择 0：当 Cortex®-M7 进入 SLEEPDEEP 模式时，MCU 进入 Deep-sleep 模式 1：当 Cortex®-M7 进入 SLEEPDEEP 模式时，MCU 进入待机模式
0	保留	必须保持复位值。

5.4.2. 电源控制和状态寄存器（PMU_CS）

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

该寄存器可以按字（32位）访问。

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

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

位/位域	名称	描述
31:21	保留	必须保持复位值。
20	VOVDF	V_{CORE}外设电压监控器标志位 该位由硬件置位和清除。 仅在VOVDEN置位的时候有效 0：V_{CORE}低于VOVD阈值（1.15V） 1：V_{CORE} 等于或高于 VOVD 阈值（1.15V）
19:17	保留	必须保持复位值。
16	VAVDF	V_{DDA} 模拟电压检测器标志位 该位由硬件置位和清除。 仅在VAVDEN置位的时候有效 0：VDDA等于或者高于VAVD阈值，阈值由VAVDVC[1:0]配置 1：VDDA 小于 VAVD 阈值，阈值由 VAVDVC[1:0]配置
15:14	保留	必须保持复位值。
13	WUPEN5	WKUP 引脚 5（PC1）唤醒使能 0：关闭 WKUP 引脚 5 唤醒功能。 1：开启 WKUP 引脚 5 唤醒功能。 如果 WUPEN5 在进入省电模式之前置 1，WKUP 引脚 5 的上升沿会将系统从省电模式唤醒。由于 WKUP 引脚 5 为高电平有效，WKUP 引脚 5 内部被配置为输入下拉模式。当在输入已经为高的时候置位该控制位，将会触发一个唤醒事件。
12	保留	必须保持复位值。
11	WUPEN3	WKUP 引脚 3（PC13）唤醒使能 0：禁能 WKUP 引脚 3 唤醒功能。 1：使能 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	保留	必须保持复位值。
9	WUPEN1	WKUP 引脚 1（PA2）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。
7:3	保留	必须保持复位值。
2	LVDF	低电压状态标志 0：低电压事件没出现（V_{DD} 高于设定的 LVD 阈值）。 1：低电压事件出现（V_{DD} 等于或低于 LVD 阈值）。 **注意：**LVD 功能在待机模式被禁用。
1	STBF	待机标志 0：设备没进入过待机模式。 1：设备曾进入过待机模式。 该位只能由 POR / PDR 或通过置位 PMU_CTL0 寄存器的 STBRST 位来清零。
0	WUF	唤醒标志 0：没有收到唤醒事件。 1：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。

5.4.3. 控制寄存器 1（PMU_CTL1）

地址偏移：0x08
复位值：0x0000 0000（从待机模式唤醒后复位）

该寄存器可以按字（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
								r	r	r	r				r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											VBTMEN	保留			BKPVSEN
											rw				rw

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 **注意：**V_{BAT} 和温度监测器只有在 BKPVSRF 置位时才有效
3:1	保留	必须保持复位值。
0	BKPVSEN	备份域电压稳压器使能位 该位由软件置位和清除。 置位后将使能备份域稳压器（能够在待机模式和电池供电模式时保持RAM内容）。 没有置位是备份域电压稳压器将失能，RAM能够在普通运行模式和Deep-sleep模式时保持内容，但无法在待机模式和V_{BAT}模式时保持内容。 如果使能，需要在BKPVSRF置位后才能写入数据到SRAM，这样才能在待机模式和电池供电模式下保持SRAM内容。 0：失能备份域电压稳压器 1：使能备份域电压稳压器

5.4.4. 控制寄存器 2（PMU_CTL2）

地址偏移：0x10
复位值：0x0000 0046（从待机模式唤醒后复位）

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					USB33RF	USBSEN	VUSB33DEN	保留							DVSRF

						r	rw	rw							r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						VCRSEL	VCEN	保留					DVSEN	LDOEN	BYPASS
						rw	rw						rw	rw	rw

位/位域	名称	描述
31:27	保留	必须保持复位值。
26	USB33RF	USB供电电压就绪标志位 0：USB33供电电压未就绪 1：USB33 供电电压已就绪
25	USBSEN	USB电压稳压器使能位 该位由软件件置位和清除。 0：失能USB电压稳压器 1：使能 USB 电压稳压器
24	VUSB33DEN	$V_{DD33USB}$电压监控器使能位 该位由软件件置位和清除。 0：失能$V_{DD33USB}$电压监控器 1：使能$V_{DD33USB}$电压监控器
23:17	保留	必须保持复位值。
16	DVSRF	降压稳压器就绪标志位 该位由硬件置位用于指示供电来源降压稳压器的外部供电是否就绪。 0：外部供电未就绪 1：外部供电已就绪
15:10	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。
7:3	保留	必须保持复位值。
2	DVSEN	降压稳压器使能位 该位由软件置位和清除。 0：失能降压稳压器 1：使能降压稳压器
1	LDOEN	LDO使能位 该位由软件置位和清除。 0：LDO失能

1：LDO使能

0	BYPASS	电源管理单元旁路控制位 该位由软件置位和清除。 0：电源管理单元正常工作 1：电源管理单元旁路，电压检测依然有效

5.4.5. 控制寄存器 3（PMU_CTL3）

地址偏移：0x14

复位值：0x0000 2000

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															VOVRF
															r

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

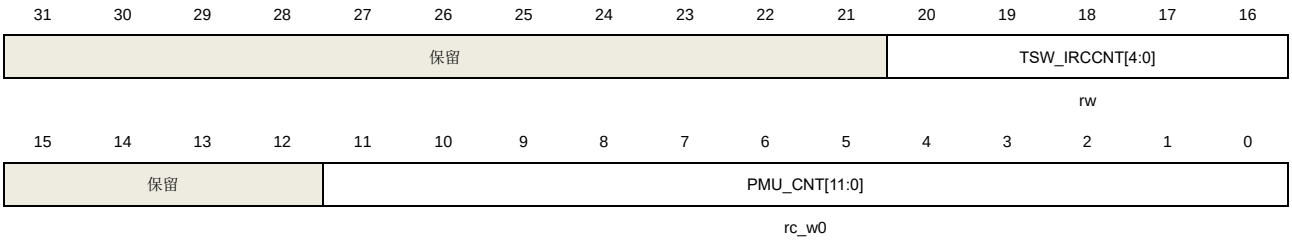
位/位域	名称	描述
31:17	保留	必须保持复位值。
16	VOVRF	V_{CORE} 电源电压就绪标志位 该位由硬件置位用于指示 V_{CORE} 供电电压是否就绪。 0：V_{CORE}供电电压未就绪 1：V_{CORE} 供电电压已就绪
15	保留	必须保持复位值。
14:12	LDOVS[2:0]	选择 LDO 输出 这些位控制 V_{CORE} 电压水平，不同的电压等级和系统时钟频率会使得 MCU 具有不同的性能，当准备降低 MCU 性能时应当先降低系统时钟频率，再改变 LDO 输出电压值，与之相反在准备提升 MCU 性能时应当先改变 LDO 输出电压值，再提升系统时钟频率。 000：LDO 输出 0.8V 电压 001：LDO 输出 0.85V 电压 010：LDO 输出 0.9V 电压（默认） 011：LDO 输出 0.95V 电压 100：LDO 输出 0.975V 电压 101：LDO输出1V电压 其它：保留。
11:0	保留	必须保持复位值。

5.4.6. 参数寄存器（PMU_PAR）

地址偏移：0x18

复位值：0x000A0000

该寄存器可以按字（32位）访问。



位/位域	名称	描述
31:21	保留	必须保持复位值。
20:16	TSW_IRCCNT[4:0]	当进入 Deep-sleep，切换到 LPIRC4M / IRC64M（由 DSPWUSSEL 确认）时钟。等待时钟计数后设置深度睡眠状态。默认值为 10 个时钟。
15:12	保留	必须保持复位值。
11:0	PMU_CNT[11:0]	退出深度睡眠模式时等待时间配置位 在退出深度睡眠模式，开启系统时钟之前，推荐等待 5us~50us。

6. 复位和时钟单元（RCU）

6.1. 复位控制单元（RCTL）

6.1.1. 简介

GD32H73x_75x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。

6.1.2. 功能说明

电源复位

当以下事件中之一发生时，产生电源复位：1、上电/掉电复位（POR / PDR 复位）2、欠压复位（BOR复位）3、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0.9V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

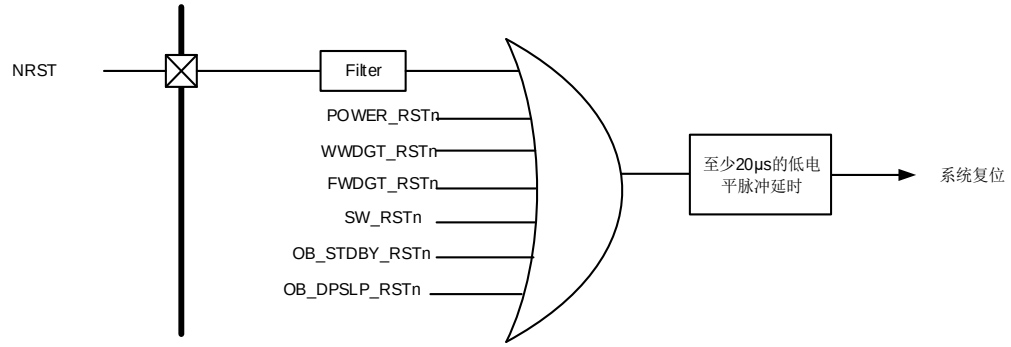
当发生以下任一事件时，产生一个系统复位：

- 电源复位（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®-M7的中断应用和复位控制寄存器中的SYSRESETREQ位置‘1’（SW_RSTn）；
- 用户选择字节寄存器nRST_STDBY位设置为0，并且进入待机模式时将产生复位（OB_STDBY_RSTn）；
- 用户选择字节寄存器nRST_DPSLP设置为0，并且进入深度睡眠模式时将产生复位（OB_DPSLP_RSTn）。

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

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

图 6-1. 系统复位电路



备份域复位

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

注意：当备份域复位时，BKPSRAM域不会复位。

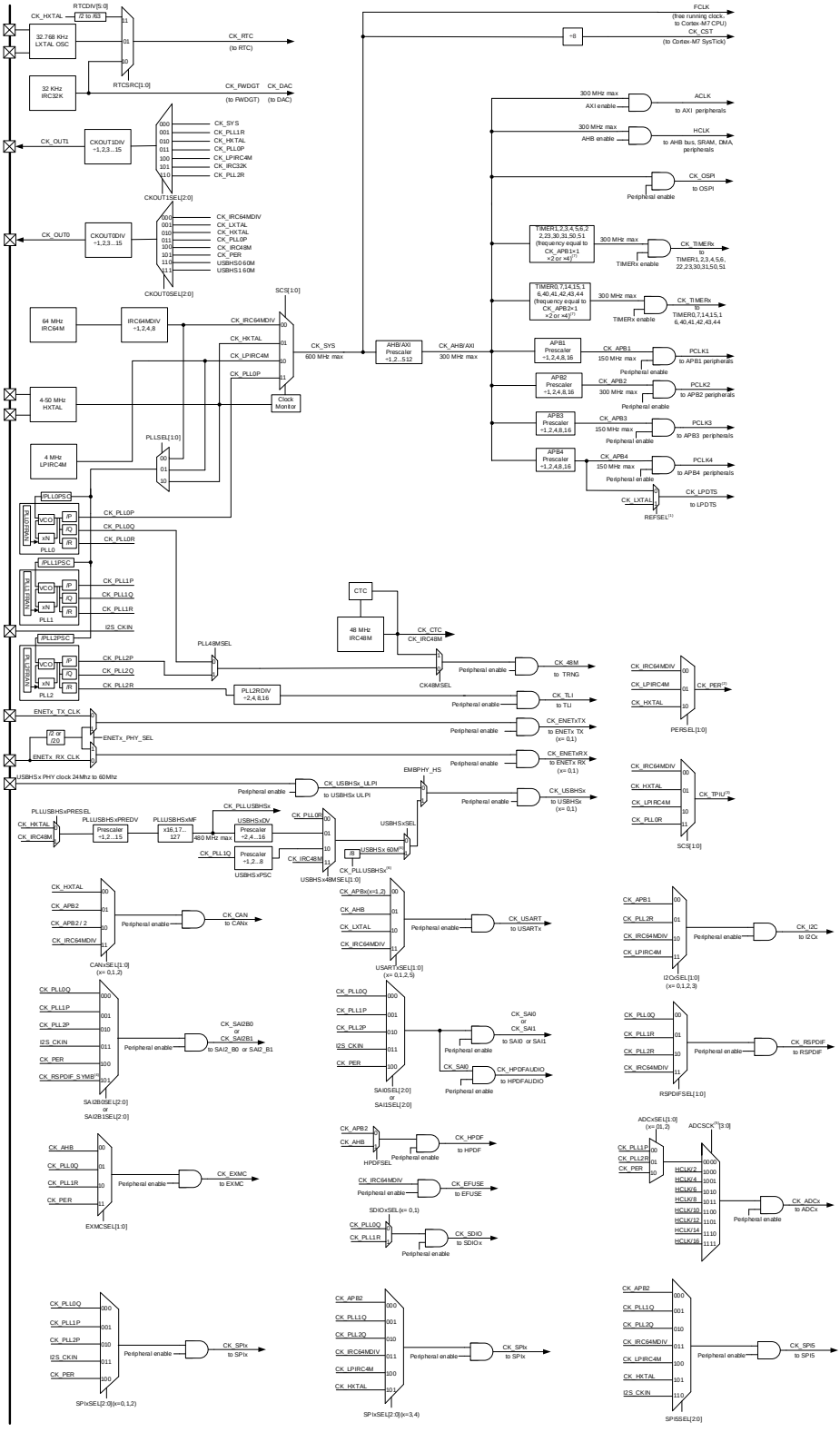
6.2. 时钟控制单元（CCTL）

6.2.1. 简介(适用于 GD32H737_757_759)

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64M RC振荡器时钟（IRC64M）、一个内部48M RC 振荡器时钟（IRC48M）、一个外部高速晶体振荡器时钟（HXTAL）、一个内部32K RC振荡器时钟（IRC32K）、一个外部低速晶体振荡器时钟（LXTAL）、一个低功耗内部4M RC振荡器时钟（LPIRC4M）、五个锁相环（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。

AXI、AHB、APB和Cortex®-M7时钟都源自系统时钟（CK_SYS），系统时钟的时钟源可以选择IRC64M、HXTAL、LPIRC4M或PLL0。系统时钟的最大运行时钟频率可以达到600MHz。

图 6-2. 时钟树



（1）REFSEL为LPDTS参考时钟选择位，可选择CK_PCLK4或CK_LXTAL参考*图40-1. LPDTS 模块框图*。

（2）CK_PER为外设时钟，该时钟可以为CK_IRC64MDIV，CK_LPIRC4M或CK_HXTAL。

（3）CK_TPIU为跟踪端口接口单元（TPIU）时钟，该时钟可以为CK_IRC64MDIV, CK_LPIRC4M，CK_HXTAL或CK_PLL0R。

（4）CK_RSPDIF_SYMB为RSPDIF符号时钟，参考*图33-1. RSPDIF模块框图*。

（5）ADCSCK为ADC同步时钟选择位，参考*同步控制寄存器（ADC_SYNCCTL）*。

（6）USBHSx 60M为USBHSx内部PHY 60M输入时钟源，参考*内部嵌入式PHY*。当选择USBHSx 60M作为USBHSx时钟时，CK_PLLUSBHSx应配置为480Mhz。

（7）TIMER时钟频率，参考RCU_CFG1寄存器中的TIMERSEL位域。

预分频器可以配置 AXI、AHB、APB4、APB3、APB2 和 APB1 域的时钟频率。AXI / AHB 和 APB4 / APB3 / APB2 / APB1 域的最高时率分别为 300 MHz / 300 MHz / 150 MHz / 150 MHz / 300 MHz / 150 MHz。RCU 通过系统时钟（CK_SYS）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，可选择上述时钟或系统时钟（CK_SYS）时钟作为 SysTick 时钟。

ADC时钟由PLL1P、PLL2R、CK_PER或由AHB时钟经2、4、6、8、10、12、14、16分频获得。

TIMER时钟由AHB时钟分频获得，它的频率可以等于CK_APBx、CK_APBx的两倍或CK_APBx的四倍。详细信息请参考RCU_CFG1寄存器的TIMERSEL位。

TRNG的时钟由CK_48M时钟提供。通过配置RCU_ADDCTL0寄存器的CK48MSEL及PLL48MSEL位可以选择PLL0Q时钟、PLL2P时钟或IRC48M时钟作为CK48M的时钟源。TRNG支持时钟动态切换。

USBHS ULPI的时钟可以选择由外部ULPI PHY时钟或RCU_USBCLKCTL寄存器中USBxHSSEL位定义的时钟提供。

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

通过设置时钟配置寄存器RCU_CFG1的USARTxSEL位，USART时钟可以选择由CK_APBx、CK_AHB、CK_LXTAL或CK_IRC64MDIV时钟提供。USART支持时钟动态切换。

通过设置时钟配置寄存器RCU_CFG0或RCU_CFG3的I2CxSEL位，I2C时钟可以选择由CK_APB1、CK_PLL2R、CK_IRC64MDIV或CK_LPIRC4M时钟提供。I2C支持时钟动态切换。

通过设置时钟配置寄存器RCU_CFG5的SPIxSEL（x = 0，1，2）位，SPI0（I2S0）、SPI1（I2S1）和SPI2（I2S2）时钟可以选择由CK_PLL0Q，CK_PLL1P，CK_PLL2P，I2S_CKIN或CK_PER时钟提供。SPI0（I2S0）、SPI1（I2S1）和SPI2（I2S2）支持时钟动态切换。

通过设置时钟配置寄存器RCU_CFG5的SPIxSEL（x = 3，4）位，SPI3和SPI4时钟可以选择由CK_APB2，CK_PLL1Q，CK_PLL2Q，CK_IRC64MDIV，CK_LPIRC4M或CK_HXTAL时钟提供。SPI3和SPI4支持时钟动态切换。

通过设置时钟配置寄存器RCU_CFG5的SPI5SEL位，SPI5（I2S5）时钟可以选择由CK_APB2，CK_PLL1Q，CK_PLL2Q，CK_IRC64MDIV，CK_LPIRC4M，CK_HXTAL或I2S_CKIN时钟提供。SPI5支持时钟动态切换。

OSPI时钟由CK_AHB时钟提供。

LPDTS时钟可以选择由CK_APB4或CK_LXTAL时钟提供。

通过设置时钟配置寄存器RCU_CFG1的CANxSEL位，CAN时钟可以选择由CK_HXTAL、CK_APB2、CK_APB2 / 2或CK_IRC64MDIV时钟提供。CAN支持时钟动态切换。

通过设置时钟配置寄存器RCU_CFG2的RSPDIFSEL位，RSPDIF时钟可以选择由CK_PLL0Q、

CK_PLL1R、CK_PLL2R或CK_IRC64MDIV时钟提供。RSPDIF支持时钟动态切换。

通过设置时钟配置寄存器RCU_CFG2的SAI2B0SEL或SAI2B1SEL位，SAI2时钟可以选择由CK_PLL0Q、CK_PLL1P、CK_PLL2P、I2S_CKIN、CK_PER或CK_RSPDIF_SYMB时钟提供。SAI2支持时钟动态切换。

通过设置时钟配置寄存器RCU_CFG2的SAI0SEL或SAI1SEL位，SAI0和SAI1时钟可以选择由CK_PLL0Q、CK_PLL1P、CK_PLL2P、I2S_CKIN或CK_PER时钟提供。SAI0和SAI1支持时钟动态切换。

通过设置时钟配置寄存器RCU_CFG1的HPDFSEL位，HPDF时钟可以选择由CK_AHB或CK_APB2时钟提供。HPDF支持时钟动态切换。

通过设置时钟配置寄存器RCU_CFG2的SAI0SEL位，HPDF_AUDIO时钟可以选择由CK_PLL0Q，CK_PLL1P，CK_PLL2P，I2S_CKIN或CK_PER时钟提供。

通过设置时钟配置寄存器RCU_CFG4的EXMCSEL位，EXMC时钟可以选择由CK_AHB，CK_PLL0Q，CK_PLL1R或CK_PER时钟提供。EXMC支持时钟动态切换。

通过设置时钟配置寄存器RCU_CFG4或RCU_CFG4的SDIOxSEL位，SDIO时钟可以选择由CK_PLL0Q或CK_PLL1R时钟提供。SDIO支持时钟动态切换。

EFUSE时钟由CK_IRC64MDIV时钟提供。

通过设置时钟配置寄存器RCU_CFG1的PLL2RDIV位，TLI时钟可以选择由PLL2R时钟的2、4、8、16分频提供。

通过配置SYSCFG_PMCFG寄存器的ENET0_PHY_SEL或ENET1_PHY_SEL位，以太网TX/RX时钟可以选择由外部引脚（ENETx_TX_CLK / ENETx_RX_CLK）输入时钟提供。

通过配置RCU_BDCTL寄存器的RTCSRC位，RTC时钟可以选择由LXTAL时钟、IRC32K时钟或HXTAL时钟的2-63（由RCU_CFG0寄存器的RTCDIV位域值决定）分频提供。RTC时钟选择HXTAL时钟的分频作为时钟源后，当0.9V内核电压域掉电时，时钟将停止。RTC时钟选择IRC32K时钟作为时钟源后，当V_{DD}掉电时，时钟将停止。RTC时钟选择LXTAL时钟作为时钟源后，当V_{DD}和V_{BAT}都掉电时，时钟将停止。

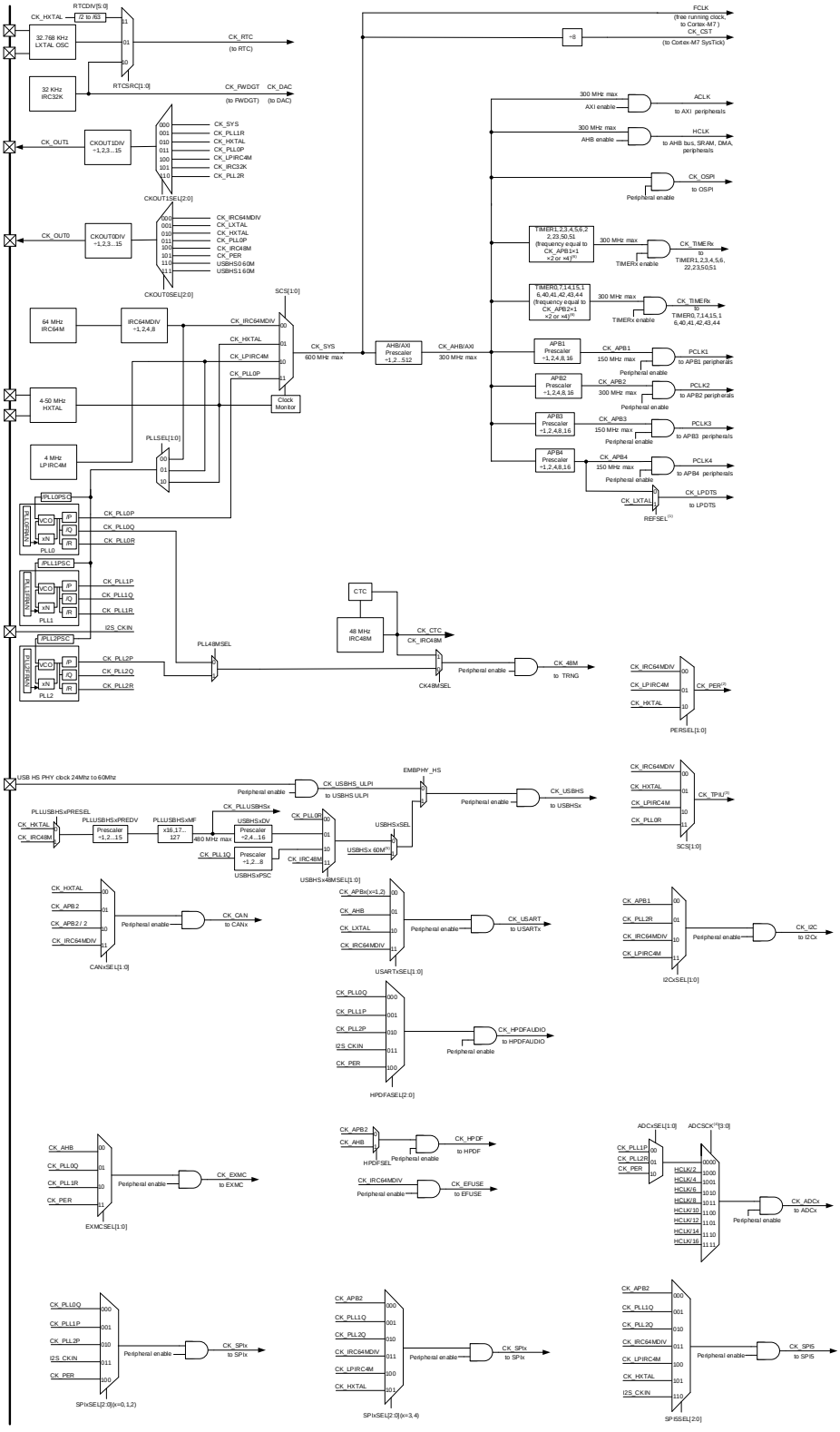
当FWDG启动时，FWDG时钟被强制选择由IRC32K时钟作为时钟源。

6.2.2. 简介(适用于 GD32H75E)

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64M RC振荡器时钟（IRC64M）、一个内部48M RC 振荡器时钟（IRC48M）、一个外部高速晶体振荡器时钟（HXTAL）、一个内部32K RC振荡器时钟（IRC32K）、一个外部低速晶体振荡器时钟（LXTAL）、一个低功耗内部4M RC振荡器时钟（LPIRC4M）、五个锁相环（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。

AXI、AHB、APB和Cortex®-M7时钟都源自系统时钟（CK_SYS），系统时钟的时钟源可以选择IRC64M、HXTAL、LPIRC4M或PLL0。系统时钟的最大运行时钟频率可以达到600MHz。

图 6-3. 时钟树



（1）REFSEL为LPDTS参考时钟选择位，可选择CK_PCLK4或CK_LXTAL参考*图40-1. LPDTS 模块框图*。

（2）CK_PER为外设时钟，该时钟可以为CK_IRC64MDIV，CK_LPIRC4M或CK_HXTAL。

（3）CK_TPIU为跟踪端口接口单元（TPIU）时钟，该时钟可以为CK_IRC64MDIV, CK_LPIRC4M，CK_HXTAL或CK_PLL0R。

（4）ADCSCK为ADC同步时钟选择位，参考*同步控制寄存器（ADC_SYNCCTL）*。

（5）USBHSx 60M为USBHSx内部PHY 60M输入时钟源，参考*内部嵌入式PHY*。当选择USBHSx 60M作为USBHSx时钟时，CK_PLLUSBHSx应配置为480Mhz。

（6）TIMER时钟频率，参考RCU_CFG1寄存器中的TIMERSEL位域。

预分频器可以配置 AXI、AHB、APB4、APB3、APB2 和 APB1 域的时钟频率。AXI / AHB 和 APB4 / APB3 / APB2 / APB1 域的最高时率分别为 300 MHz / 300 MHz / 150 MHz / 150 MHz / 300 MHz / 150 MHz。RCU 通过系统时钟（CK_SYS）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，可选择上述时钟或系统时钟（CK_SYS）时钟作为 SysTick 时钟。

ADC时钟由PLL1P、PLL2R、CK_PER或由AHB时钟经2、4、6、8、10、12、14、16分频获得。

TIMER时钟由AHB时钟分频获得，它的频率可以等于CK_APBx、CK_APBx的两倍或CK_APBx的四倍。详细信息请参考RCU_CFG1寄存器的TIMERSEL位。

TRNG的时钟由CK_48M时钟提供。通过配置RCU_ADDCTL0寄存器的CK48MSEL及PLL48MSEL位可以选择PLL0Q时钟、PLL2P时钟或IRC48M时钟作为CK48M的时钟源。TRNG支持时钟动态切换。

USBHS ULPI的时钟可以选择由外部ULPI PHY时钟或RCU_USBCLKCTL寄存器中USBxHSSEL位定义的时钟提供。

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

通过设置时钟配置寄存器RCU_CFG1的USARTxSEL位，USART时钟可以选择由CK_APBx、CK_AHB、CK_LXTAL或CK_IRC64MDIV时钟提供。USART支持时钟动态切换。

通过设置时钟配置寄存器RCU_CFG0或RCU_CFG3的I2CxSEL位，I2C时钟可以选择由CK_APB1、CK_PLL2R、CK_IRC64MDIV或CK_LPIRC4M时钟提供。I2C支持时钟动态切换。

通过设置时钟配置寄存器RCU_CFG5的SPIxSEL（x = 0，1，2）位，SPI0（I2S0）、SPI1（I2S1）和SPI2（I2S2）时钟可以选择由CK_PLL0Q，CK_PLL1P，CK_PLL2P，I2S_CKIN或CK_PER时钟提供。SPI0（I2S0）、SPI1（I2S1）和SPI2（I2S2）支持时钟动态切换。

通过设置时钟配置寄存器RCU_CFG5的SPIxSEL（x = 3，4）位，SPI3和SPI4时钟可以选择由CK_APB2，CK_PLL1Q，CK_PLL2Q，CK_IRC64MDIV，CK_LPIRC4M或CK_HXTAL时钟提供。SPI3和SPI4支持时钟动态切换。

通过设置时钟配置寄存器RCU_CFG5的SPI5SEL位，SPI5（I2S5）时钟可以选择由CK_APB2，CK_PLL1Q，CK_PLL2Q，CK_IRC64MDIV，CK_LPIRC4M，CK_HXTAL或I2S_CKIN时钟提供。SPI5支持时钟动态切换。

OSPI时钟由CK_AHB时钟提供。

LPDTS时钟可以选择由CK_APB4或CK_LXTAL时钟提供。

通过设置时钟配置寄存器RCU_CFG1的CANxSEL位，CAN时钟可以选择由CK_HXTAL、CK_APB2、CK_APB2 / 2或CK_IRC64MDIV时钟提供。CAN支持时钟动态切换。

通过设置时钟配置寄存器RCU_CFG1的HPDFSEL位，HPDF时钟可以选择由CK_AHB或CK_APB2时钟提供。HPDF支持时钟动态切换。

通过设置时钟配置寄存器RCU_CFG2的HPDFASEL位，HPDF_AUDIO时钟可以选择由CK_PLL0Q，CK_PLL1P，CK_PLL2P，I2S_CKIN或CK_PER时钟提供。

通过设置时钟配置寄存器RCU_CFG4的EXMCSEL位，EXMC时钟可以选择由CK_AHB，CK_PLL0Q，CK_PLL1R或CK_PER时钟提供。EXMC支持时钟动态切换。

EFUSE时钟由CK_IRC64MDIV时钟提供。

通过配置RCU_BDCTL寄存器的RTCSRC位，RTC时钟可以选择由LXTAL时钟、IRC32K时钟或HXTAL时钟的2-63（由RCU_CFG0寄存器的RTCDIV位域值决定）分频提供。RTC时钟选择HXTAL时钟的分频作为时钟源后，当0.9V内核电压域掉电时，时钟将停止。RTC时钟选择IRC32K时钟作为时钟源后，当V_{DD}掉电时，时钟将停止。RTC时钟选择LXTAL时钟作为时钟源后，当V_{DD}和V_{BAT}都掉电时，时钟将停止。

当FWDG启动时，FWDG时钟被强制选择由IRC32K时钟作为时钟源。

6.2.3. 主要特征

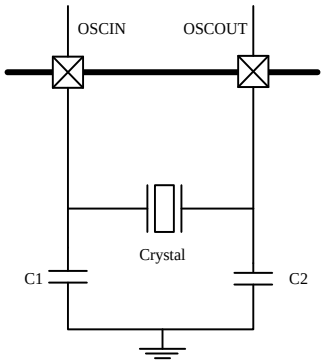
- 4到50 MHz外部高速晶体振荡器（HXTAL）；
- 内部64 MHz RC振荡器（IRC64M）；
- 内部48 MHz RC振荡器（IRC48M）；
- 32,768 Hz外部低速晶体振荡器 （LXTAL）；
- 内部32 KHz RC振荡器（IRC32K）；
- 低功耗内部4M RC振荡器（LPIRC4M）；
- PLL时钟源可选HXTAL、LPIRC4M或IRC64M；
- PLLs支持整数和小数倍频因子；
- PLLs小数倍频因子可在运行时修改；
- 外设时钟支持动态切换；
- HXTAL时钟监视器；
- LXTAL时钟监视器。

6.2.4. 功能说明

外部高速晶体振荡器时钟（HXTAL）

4到50MHz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与靠近两个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。

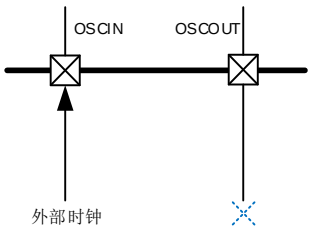
图 6-4. HXTAL 时钟源



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*图6-5. 旁路模式下HXTAL时钟源*。此时，CK_HXTAL等于驱动OSCIN管脚的外部时钟。

图 6-5. 旁路模式下 HXTAL 时钟源



内部 64M RC 振荡器时钟（IRC64M）

内部64MHz RC振荡器时钟，简称IRC64M时钟，拥有64MHz的固定频率，设备上电后CPU默认选择其作为系统时钟源。通过配置RCU_ADDCTL1寄存器中的IRC64MDIV[1:0]位域，CK_IRC64MDIV可提供8、16、32或64MHz时钟输出。IRC64M 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。IRC64M RC振荡器可以通过设置控制寄存器（RCU_CTL）中的IRC64MEN位被启动和关闭。控制寄存器RCU_CTL中的IRC64MSTB位用来指示IRC64M内部RC振荡器是否稳定。IRC64M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IRC64MSTBIE被置‘1’，在IRC64M稳定以后，将产生一个中断。IRC64M时钟也可用作系统时钟源或PLL输入时钟。

工厂会校准IRC64M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PLL0P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CK_IRC64MDIV或CK_LPIRC4M时钟作为系统或内核时钟。

内部 48M RC 振荡器时钟（IRC48M）

内部48MHz RC振荡器时钟，简称IRC48M时钟，拥有48MHz的固定频率，当使用USBHS / TRNG模块时，IRC48M振荡器在不需要任何外部器件的条件下为用户提供了一种成本更低的时钟源选择。IRC48M RC振荡器可以通过设置RCU_ADDCTL0寄存器中的IRC48MEN位被启动和关闭。RCU_ADDCTL0寄存器中的IRC48MSTB位用来指示内部48MHz RC振荡器是否稳定。如果RCU_ADDINT寄存器中的相应中断使能位IRC48MSTBIE被置‘1’，在IRC48M稳定以后，将产生一个中断。IRC48M时钟可作为USBHS / TRNG模块时钟。

工厂会校准IRC48M时钟频率的精度，但是它的精度仍然不够精准。因为USBHS模块需要的时钟频率必须满足48MHz±1%。CTC单元提供了一种硬件自动执行动态调整的功能将IRC48M时钟调整到需要的频率。

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，它的时钟频率大约32 kHz，为独立看门狗和实时时钟电路提供时钟。IRC32K提供低成本的时钟源，因为不需要外部器件。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。

低功耗内部 4M RC 振荡器时钟（LPIRC4M）

低功耗内部4MHz RC振荡器时钟，简称LPIRC4M时钟，拥有4MHz的固定频率，可以用作系统输入时钟或PLL输入时钟。LPIRC4M 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。LPIRC4M RC振荡器可以通过设置附加控制寄存器1（RCU_ADDCTL1）中的LPIRC4MEN位被启动和关闭。中断寄存器RCU_INT中的LPIRC4MSTB位用来指示内部LPIRC4M RC振荡器是否稳定。如果中断寄存器RCU_INT中的相应中断使能位LPIRC4MSTBIE被置‘1’，在LPIRC4M稳定以后，将产生一个中断。

工厂会校准LPIRC4M时钟频率的精度。复位后，校准值将会被加载到RCU_ADDCTL1寄存器中的LPIRC4MCALIB位域。

如果HXTAL或者PLL0P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CK_IRC64MDIV或CK_LPIRC4M时钟作为系统或内核时钟。

锁相环（PLL）

存在五个内部锁相环，PLL0、PLL1、PLL2、PLLUSBHS0和PLLUSBHS1。PLL0、PLL1和PLL2支持整数和小数倍频因子且小数倍频因子可在运行时修改。另外，PLL0、PLL1和PLL2可分别产生P / Q / R时钟输出。PLL0P时钟可作为系统时钟（不超过600MHz）。

对于每一个PLL，当RCU_PLLxFRA寄存器中的PLLxFRAEN位为‘1’且PLLxFRAN值不为‘0’时，PLLx处在小数模式，例如：

$$CK_PLL0VCO = CK_PLL0VCOSRC * \left(PLL0N + \frac{PLL0FRAN}{2^{13}}\right) \quad (6\text{-}1)$$

否则，PLLx处于整数模式，例如：

$$CK_PLL0VCO = CK_PLL0VCOSRC * PLL0N \quad (6\text{-}2)$$

PLL0可以通过设置RCU_CTL寄存器中的PLL0EN位被启动和关闭。RCU_CTL寄存器中的PLL0STB位用来指示PLL0时钟是否稳定。如果RCU_INT寄存器中的相应中断使能位PLL0STBIE被置‘1’，在PLL0稳定以后，将产生一个中断。

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

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

PLLUSBHS0可以通过设置RCU_ADDCTL1寄存器中的PLLUSBHS0EN位被启动和关闭。RCU_ADDCTL1寄存器中的PLLUSBHS0STB位用来指示PLLUSBHS0时钟是否稳定。如果RCU_ADDINT寄存器中的相应中断使能位PLLUSBHS0STBIE被置‘1’，在PLLUSBHS0稳定以后，将产生一个中断。

PLLUSBHS1可以通过设置RCU_ADDCTL1寄存器中的PLLUSBHS1EN位被启动和关闭。RCU_ADDCTL1寄存器中的PLLUSBHS1STB位用来指示PLLUSBHS1时钟是否稳定。如果RCU_ADDINT寄存器中的相应中断使能位PLLUSBHS1STBIE被置‘1’，在PLLUSBHS1稳定以后，将产生一个中断。

当进入Deepsleep/Standby模式或者HXTAL监视器检测到时钟阻塞时（HXTAL作为锁相环的输入时钟），这三路PLL将被关闭。

外设时钟动态切换

如果外设有两个以上的时钟源选择，则该外设可以在运行时动态地切换至另一个开启的时钟源。否则，该外设时钟将无法切换。对于 GD32H737_757_759 产品，TRNG / USART / I2C / SPI / RSPDIF / SAI / SDIO / EXMC / CAN / HPDF 外设支持时钟动态切换。对于 GD32H75E 产品，TRNG / USART / I2C / SPI / EXMC / CAN / HPDF 外设支持时钟动态切换。

系统时钟（CK_SYS）选择

系统复位后，IRC64M时钟默认作为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、LPIRC4M或CK_PLL0P。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PLL0P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®-M7的不可屏蔽中断NMI相连。如果HXTAL被选作系统或PLL0的时钟源，HXTAL故障将促使选择IRC64M为系统时钟源且PLL0将被自动禁止。如果HXTAL被选作PLLs的时钟源，HXTAL故障将促使该PLL被自动禁止。

LXTAL 时钟监视器（LCKM）

设置时钟控制寄存器RCU_BDCTL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启动延迟完毕后使能。

LXTAL上的时钟监视器在除V_{BAT}以外的所有模式下工作。如果在外部32 kHz振荡器上检测到故障，可以向CPU发送中断。

然后，软件必须禁用LCKMEN位，停止有缺陷的32 kHz振荡器，并更改RTC时钟源，或采取任何必要的措施来保护应用程序。

当LCKMEN启用时，一个4位加一个计数器将在IRC32K域工作。如果LXTAL时钟卡在0/1错误或减慢约20KHz，计数器将溢出。将发现LXTAL时钟故障。一旦监测到LXTAL故障，中断寄存器RCU_INT中的LXTAL时钟阻塞中断标志位LCKMIF将被置‘1’，产生LXTAL故障事件。

时钟输出功能

时钟输出功能输出从32kHz到600MHz的时钟。通过设置时钟配置寄存器0（RCU_CFG2）中的CK_OUT0时钟源选择位域CKOUT0SEL能够选择不同的时钟信号。相应的GPIO引脚应该被配置成备用功能I/O（AFIO）模式来输出选择的时钟信号。CK_OUT1时钟输出源选择通过设置时钟配置寄存器RCU_CFG2中的CKOUT1SEL位域实现。

表 6-1. 时钟输出 0 的时钟源选择

时钟输出 0 的时钟源选择位域	时钟源
000	CK_IRC64MDIV
001	CK_LXTAL
010	CK_HXTAL
011	CK_PLL0P
100	CK_IRC48M
101	CK_PER
110	USBHS0 60M
111	USBHS1 60M

表 6-2. 时钟输出 1 的时钟源选择

时钟输出 1 的时钟源选择位域	时钟源
000	CK_SYS
001	CK_PLL1R
010	CK_HXTAL
011	CK_PLL0P
100	CK_LPIRC4M
101	CK_IRC32K
110	CK_PLL2R

通过配置RCU_CFG2寄存器的CKOUT0DIV位域，可以将CK_OUT0输出时钟的频率按比例分频，进而降低CK_OUT0的输出频率。

通过配置RCU_CFG0寄存器的CKOUT1DIV位域，可以将CK_OUT1输出时钟的频率按比例分频，进而降低CK_OUT1的输出频率。

RTC 时钟测量

RTC时钟的三种时钟源：LXTAL、IRC32K和HXTAL时钟的2-63分频（通过配置RCU_CFG0寄存器的RTCDIV位域），可以通过TIMER模块测量频率。用户可以根据计算得到的时钟频率调整RTC和独立看门狗计数器。详细信息请参考***SYSCFG_TIMERCISEL6***寄存器的TIMER15_CI0_SEL位与TIMER16_CI0_SEL位。

6.3. RCU 寄存器（适用于 GD32H737_757_759）

RCU 基地址：0x5802 4400

6.3.1. 控制寄存器（RCU_CTL）

地址偏移：0x00
复位值：0xC000 8040

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
IRC64MSTB	IRC64MEN	PLL2STB	PLL2EN	PLL1STB	PLL1EN	PLL0STB	PLL0EN	保留				CKMEN	HXTALBPS	HXTALSTB	HXTALEN
r	rw	r	rw	r	rw	r	rw					rw	rw	r	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRC64MCALIB[8:0]									IRC64MADJ[6:0]						
r									rw						

位/位域	名称	描述
31	IRC64MSTB	内部64MHz RC振荡器稳定标志位 硬件置‘1’来指示IRC64M振荡器时钟是否稳定待用 0：IRC64M振荡器未稳定 1：IRC64M 振荡器已稳定
30	IRC64MEN	内部64MHz RC振荡器使能 软件置位或复位，如果IRC64M时钟作为系统时钟时，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IRC64M振荡器。 0：内部64 MHz RC振荡器被关闭 1：内部 64 MHz RC 振荡器被打开
29	PLL2STB	PLL2 时钟稳定标志位 硬件置1来表示PLL2输出时钟是否稳定待用 0：PLL2未稳定 1：PLL2 已稳定
28	PLL2EN	PLL2 使能 软件置位或复位，当进入深度睡眠或待机模式时由硬件复位 0：PLL2被关闭 1：PLL2 被打开
27	PLL1STB	PLL1 时钟稳定标志位 硬件置1来表示PLL1输出时钟是否稳定待用 0：PLL1未稳定 1：PLL1 已稳定
26	PLL1EN	PLL1 使能

		软件置位或复位，当进入深度睡眠或待机模式时由硬件复位 0：PLL1被关闭 1：PLL1 被打开
25	PLL0STB	PLL0 时钟稳定标志位 硬件置1来表示PLL0输出时钟是否稳定待用 0：PLL0未稳定 1：PLL0 已稳定
24	PLL0EN	PLL0 使能 软件置位或复位，当PLL0时钟作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：PLL0被关闭 1：PLL0 被打开
23:20	保留	必须保持复位值。
19	CKMEN	HXTAL时钟监视器使能 0：禁止高速4 ~ 50 MHz晶体振荡器（HXTAL）时钟监视器 1：使能高速4 ~ 50 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IRC64M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IRC64MEN的状态，自动使能IRC64M时钟。
18	HXTALBPS	高速晶体振荡器（HXTAL）时钟旁路模式使能 只有在HXTAL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PLL0P时钟作为系统时钟时，其作为PLL0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 ~ 50 MHz晶体振荡器被关闭 1：高速 4 ~ 50 MHz 晶体振荡器被打开
15:7	IRC64MCALIB[8:0]	内部64MHz RC振荡器校准值 上电时自动加载这些位
6:0	IRC64MADJ[6:0]	内部 64MHz RC 振荡器时钟调整值 这些位由软件置位，最终调整值为 IRC64MADJ[6:0]位域的当前值加上

IRC64MCALIB[8:0]位域的值。最终调整值应该调整 IRC64M 到 64 MHz ± 1%

6.3.2. PLL0 寄存器（RCU_PLL0）

地址偏移：0x04

复位值：0x0100 2020

配置PLL0时钟可参考下列公式：

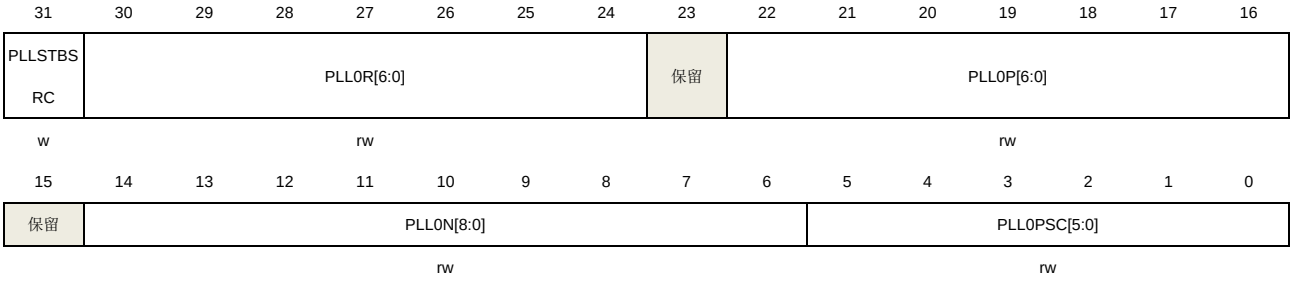
CK_PLL0VCOSRC = CK_PLL0SRC / PLL0PSC

CK_PLL0VCO = CK_PLL0VCOSRC × （PLL0N + PLL0FRAN / 2^{13}）

CK_PLL0R = CK_PLL0VCO / PLL0R

CK_PLL0P = CK_PLL0VCO / PLL0P

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	PLLSTBSRC	PLLs 稳定信号源 0：模拟信号 1：数字信号
30:24	PLL0R[6:0]	PLL0R 输出频率的分频系数（PLL0 VCO 时钟作为输入） 当 PLL0 被关闭时由软件置位或清零。这些位域用做将 PLL0 VCO 时钟（CK_PLL0VCO）分频生成 PLL0R 输出时钟（CK_PLL0R）。RCU_PLL0 寄存器的 PLL0N 位域对 CK_PLL0VCO 时钟进行了描述。 0000000：CK_PLL0R = CK_PLL0VCO 0000001：CK_PLL0R = CK_PLL0VCO / 2 0000010：CK_PLL0R = CK_PLL0VCO / 3. 0000011：CK_PLL0R = CK_PLL0VCO / 4 0000100：CK_PLL0R = CK_PLL0VCO / 5 … 1111111：CK_PLL0R = CK_PLL0VCO / 128
23	保留	必须保持复位值。
22:16	PLL0P[6:0]	PLL0P 输出频率分频系数（PLL0 VCO 时钟作为输入） 当 PLL0 被关闭时由软件置位或清零。这些位域用做将 PLL0 VCO 时钟（CK_PLL0VCO）分频生成 PLL0P 输出时钟（CK_PLL0P）。CK_PLL0P 时钟可以被用作系统时钟（不超过 600MHz）。 RCU_PLL0 寄存器的 PLL0N 位域对 CK_PLL0VCO 时钟进行了描述。

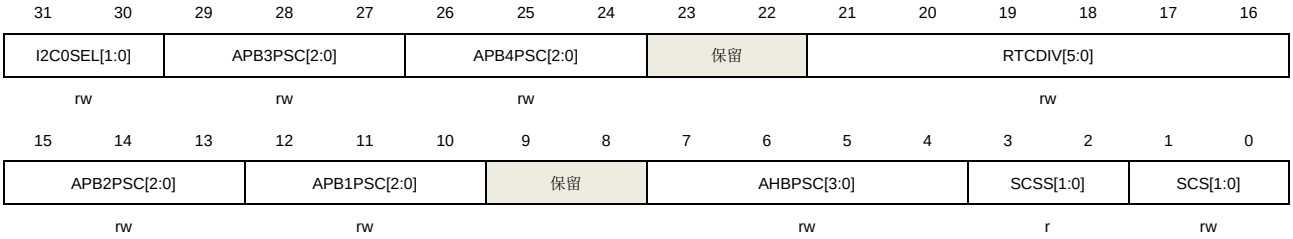
0000000：CK_PLL0P = CK_PLL0VCO
0000001：CK_PLL0P = CK_PLL0VCO / 2
0000010：CK_PLL0P = CK_PLL0VCO / 3
0000011：CK_PLL0P = CK_PLL0VCO / 4
0000100：CK_PLL0P = CK_PLL0VCO / 5
…
1111111：CK_PLL0P = CK_PLL0VCO / 128

15	保留	必须保持复位值。
14:6	PLL0N[8:0]	PLL0 VCO 时钟倍频因子 当 PLL0 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 PLL0 VCO 源时钟（CK_PLL0VCOSRC）倍频生成 PLL0 VCO 输出时钟（CK_PLL0VCO）。RCU_PLL0 寄存器的 PLL0PSC 位域对 CK_PLL0VCOSRC 时钟进行了描述。 **注意**：CK_PLL0VCO 时钟频率范围必须在 150MHz 到 836MHz 之间 PLL0N 的值必须满足： 当 PLL0 小数锁存禁能时，PLL0N 的值必须满足：9 ≤ PLL0N ≤ 512 当 PLL0 小数锁存使能时，PLL0N 的值必须满足：12 ≤ PLL0N ≤ 508 000000000：保留 … 000000111：保留 000001000：PLL0N = 9 … 001000000：PLL0N = 65 001000001：PLL0N = 66 … 111111111：PLL0N = 512
5:0	PLL0PSC[5:0]	PLL0 VCO 源时钟分频器 当 PLL0 被关闭时由软件置位或清零。这些位域用做将 PLL0 源时钟（CK_PLL0SRC）分频生成 PLL0 VCO 源时钟（CK_PLL0VCOSRC）。RCU_PLLALL 寄存器的 PLLSEL 位对 CK_PLL0SRC 时钟进行了描述。 VCO 源时钟频率范围必须在 1MHz 到 16MHz 之间 000000：保留 000001：CK_PLL0SRC 000010：CK_PLL0SRC / 2 000011：CK_PLL0SRC / 3 … 111111：CK_PLL0SRC / 63

6.3.3. 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。



位/位域	名称	描述
31:30	I2C0SEL[1:0]	I2C0时钟源选择 由软件置位或复位，控制I2C0时钟源 00：选择 CK_APB1 时钟作为 I2C0 源时钟 01：选择 CK_PLL2R 时钟作为 I2C0 源时钟 10：选择 CK_IRC64MDIV 时钟作为 I2C0 源时钟 11：选择CK_LPIRC4M时钟作为I2C0源时钟
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：选择 CK_AHB 时钟不分频 100：选择 CK_AHB 时钟 2 分频 101：选择 CK_AHB 时钟 4 分频 110：选择 CK_AHB 时钟 8 分频 111：选择 CK_AHB 时钟 16 分频
26:24	APB4PSC[2:0]	APB4 预分频选择 由软件置位或清零，控制 APB4 时钟分频因子 0xx：选择 CK_AHB 时钟不分频 100：选择 CK_AHB 时钟 2 分频 101：选择 CK_AHB 时钟 4 分频 110：选择 CK_AHB 时钟 8 分频 111：选择 CK_AHB 时钟 16 分频
23:22	保留	必须保持复位值。
21:16	RTCDIV[5:0]	RTC 时钟分频系数 由软件置位或清零。这些位用作将 HXTAL 时钟分频生成 RTC 时钟（不超过 1MHz） 000000：无时钟 000001：无时钟 000010：CK_HXTAL / 2 000011：CK_HXTAL / 3 … 111111：CK_HXTAL / 63
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：选择 CK_AHB 时钟不分频 100：选择 CK_AHB 时钟 2 分频

		101：选择 CK_AHB 时钟 4 分频 110：选择 CK_AHB 时钟 8 分频 111：选择 CK_AHB 时钟 16 分频
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：选择 CK_AHB 时钟不分频 100：选择 CK_AHB 时钟 2 分频 101：选择 CK_AHB 时钟 4 分频 110：选择 CK_AHB 时钟 8 分频 111：选择 CK_AHB 时钟 16 分频
9:8	保留	必须保持复位值。
7:4	AHBPSC[3:0]	AHB / AXI 预分频选择 由软件置位或清零，控制 AHB / AXI 时钟分频因子. 0xxx：选择 CK_SYS 时钟不分频 1000：选择 CK_SYS 时钟 2 分频 1001：选择 CK_SYS 时钟 4 分频 1010：选择 CK_SYS 时钟 8 分频 1011：选择 CK_SYS 时钟 16 分频 1100：选择 CK_SYS 时钟 64 分频 1101：选择 CK_SYS 时钟 128 分频 1110：选择 CK_SYS 时钟 256 分频 1111：选择 CK_SYS 时钟 512 分频
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_IRC64MDIV 时钟作为 CK_SYS 时钟源 01：选择 CK_HXTAL 时钟作为 CK_SYS 时钟源 10：选择 CK_LPIRC4M 时钟作为 CK_SYS 时钟源 11：选择 CK_PLL0P 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CK_IRC64MDIV作为系统时钟。 00：选择 CK_IRC64MDIV 时钟作为 CK_SYS 时钟源 01：选择 CK_HXTAL 时钟作为 CK_SYS 时钟源 10：选择 CK_LPIRC4M 时钟作为 CK_SYS 时钟源 11：选择 CK_PLL0P 时钟作为 CK_SYS 时钟源

6.3.4. 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	LPIRC4MSTBIC	LPIRC4MSTBIE	LPIRC4MSTBIF	CKMIC	PLL2STBIC	PLL1STBIC	PLL0STBIC	HXTALSTBIC	IRC64MSTBIC	LXTALSTBIC	IRC32KSTBIC
			w	r	w	rw	r	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	PLL2STBIE	PLL1STBIE	PLL0STBIE	HXTALSTBIE	IRC64MSTBIE	LXTALSTBIE	IRC32KSTBIE	CKMIF	PLL2STBIF	PLL1STBIF	PLL0STBIF	HXTALSTBIF	IRC64MSTBIF	LXTALSTBIF	IRC32KSTBIF
	rw	rw	rw	rw	rw	rw	rw	r	r	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	LPIRC4MSTBIC	LPIRC4M 时钟稳定中断清零 软件写 1 复位 LPIRC4MSTBIF 标志位 0：不复位 LPIRC4MSTBIF 标志位 1：复位 LPIRC4MSTBIF 标志位
25	LPIRC4MSTBIE	LPIRC4M 时钟稳定中断使能 软件置位和复位来使能/禁止LPIRC4M时钟稳定中断 0：禁止LPIRC4M时钟稳定中断 1：使能 LPIRC4M 时钟稳定中断
24	LPIRC4MSTBIF	LPIRC4M 时钟稳定中断标志位 当内部64MHz RC振荡器时钟稳定且LPIRC4MSTBIE位被置1时由硬件置1 软件置位 LPIRC4MSTBIC 位时清除该位 0：无LPIRC4M时钟稳定中断产生 1：产生 LPIRC4M 时钟稳定中断
23	CKMIC	HXTAL时钟阻塞中断清零 软件写 1 复位 CKMIF 标志位. 0：不复位 CKMIF 标志位 1：复位 CKMIF 标志位
22	PLL2STBIC	PLL2 时钟稳定中断清零 软件写 1 复位 PLL2STBIF 标志位 0：不复位 PLL2STBIF 标志位

		1：复位 PLL2STBIF 标志位
21	PLL1STBIC	PLL1 时钟稳定中断清零 软件写 1 复位 PLL1STBIF 标志位 0：不复位 PLL1STBIF 标志位 1：复位 PLL1STBIF 标志位
20	PLL0STBIC	PLL0 时钟稳定中断清零 软件写 1 复位 PLL0STBIF 标志位 0：不复位 PLL0STBIF 标志位 1：复位 PLL0STBIF 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	IRC64MSTBIC	IRC64M 时钟稳定中断清零 软件写 1 复位 IRC64MSTBIF 标志位 0：不复位 IRC64MSTBIF 标志位 1：复位 IRC64MSTBIF 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	保留	必须保持复位值。
14	PLL2STBIE	PLL2 时钟稳定中断使能 软件置位和复位来使能/禁止PLL2时钟稳定中断 0：禁止 PLL2 时钟稳定中断 1：使能 PLL2 时钟稳定中断
13	PLL1STBIE	PLL1 时钟稳定中断使能 软件置位和复位来使能/禁止PLL1时钟稳定中断 0：禁止 PLL1 时钟稳定中断 1：使能 PLL1 时钟稳定中断
12	PLL0STBIE	PLL0 时钟稳定中断使能 软件置位和复位来使能/禁止PLL0时钟稳定中断 0：禁止 PLL0 时钟稳定中断 1：使能 PLL0 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	IRC64MSTBIE	IRC64M 时钟稳定中断使能 软件置位和复位来使能/禁止IRC64M时钟稳定中断 0：禁止IRC64M时钟稳定中断 1：使能 IRC64M 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	PLL2STBIF	PLL2 时钟稳定中断标志位 当PLL2时钟稳定且PLL2STBIE位被置1时由硬件置1 软件置位 PLL2STBIC 位时清除该位 0：无PLL2时钟稳定中断产生 1：产生 PLL2 时钟稳定中断
5	PLL1STBIF	PLL1 时钟稳定中断标志位 当PLL1时钟稳定且PLL1STBIE位被置1时由硬件置1 软件置位 PLL1STBIC 位时清除该位 0：无PLL1时钟稳定中断产生 1：产生 PLL1 时钟稳定中断
4	PLL0STBIF	PLL0 时钟稳定中断标志位 当PLL0时钟稳定且PLL0STBIE位被置1时由硬件置1 软件置位 PLL0STBIC 位时清除该位 0：无PLL0时钟稳定中断产生 1：产生 PLL0 时钟稳定中断
3	HXTALSTBIF	HXTAL 时钟稳定中断标志位 当高速4 ~ 50 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	IRC64MSTBIF	IRC64M 时钟稳定中断标志位 当内部64MHz RC振荡器时钟稳定且IRC64MSTBIE位被置1时由硬件置1 软件置位 IRC64MSTBIC 位时清除该位 0：无IRC64M时钟稳定中断产生 1：产生 IRC64M 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

6.3.5. AHB1 复位寄存器（RCU_AHB1RST）

地址偏移：0x10

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		USBHS1RST	保留			ENET0RST	保留	DMAMUXRST	DMA1RST	DMA0RST	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	USBHS0RST	保留													ENET1RST
	rw														rw

位/位域	名称	描述
31:30	保留	必须保持复位值。
29	USBHS1RST	USBHS1 复位 由软件置位或复位 0：无作用 1：复位 USBHS1
28:26	保留	必须保持复位值。
25	ENET0RST	Ethernet0 复位 由软件置位或复位 0：无作用

		1：复位 ENET0
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
20:15	保留	必须保持复位值。
14	USBHS0RST	USBHS0 复位 由软件置位或复位 0：无作用 1：复位 USBHS0
13:1	保留	必须保持复位值。
0	ENET1RST	Ethernet1 复位 由软件置位或复位 0：无作用 1：复位 ENET1

6.3.6. AHB2 复位寄存器（RCU_AHB2RST）

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
保留								TMURST	TRNGRST	保留	HAURST	CAURST	SDIO1RST	FACRST	DCIRST
								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	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	保留	必须保持复位值。
4	HAURST	HAU 复位 由软件置位或复位 0：无作用 1：复位 HAU
3	CAURST	CAU 复位 由软件置位或复位 0：无作用 1：复位 CAU
2	SDIO1RST	SDIO1 复位 由软件置位或复位 0：无作用 1：复位 SDIO1
1	FACRST	FAC 复位 由软件置位或复位 0：无作用 1：复位 FAC
0	DCIRST	DCI 复位 由软件置位或复位 0：无作用 1：复位 DCI

6.3.7. AHB3 复位寄存器（RCU_AHB3RST）

地址偏移：0x18

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						RTDEC1RST	RTDEC0RST	保留	OSPI1RST	OSPI0RST	OSPIMRST	MDMARST	SDIO0RST	IPARST	EXMCRST
						rw	rw		rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:10	保留	必须保持复位值。
9	RTDEC1RST	RTDEC1 复位 由软件置位或复位 0：无作用 1：复位 RTDEC1
8	RTDEC0RST	RTDEC0 复位 由软件置位或复位 0：无作用 1：复位 RTDEC0
7	保留	必须保持复位值。
6	OSPI1RST	OSPI1 复位 由软件置位或复位 0：无作用 1：复位 OSPI1
5	OSPI0RST	OSPI0 复位 由软件置位或复位 0：无作用 1：复位 OSPI0
4	OSPIMRST	OSPIM 复位 由软件置位或复位 0：无作用 1：复位 OSPIM
3	MDMARST	MDMA 复位 由软件置位或复位 0：无作用 1：复位 MDMA
2	SDIO0RST	SDIO0 复位 由软件置位或复位 0：无作用 1：复位 SDIO0
1	IPARST	IPA 复位 由软件置位或复位 0：无作用

1：复位 IPA

0 EXMCRST EXMC 复位
由软件置位或复位
0：无作用
1：复位 EXMC

6.3.8. AHB4 复位寄存器（RCU_AHB4RST）

地址偏移：0x1C
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
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
HWSEMRST	CRCRST	保留				PKRST	PJRST	PHRST	PGRST	PFRST	PERST	PDRST	PCRST	PBRST	PARST

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	HWSEMRST	HWSEM 复位 由软件置位或复位 0：无作用 1：复位 HWSEM
14	CRCRST	CRC 复位 由软件置位或复位 0：无作用 1：复位 CRC
13:10	保留	必须保持复位值。
9	PKRST	GPIO 端口 K 复位 由软件置位或复位 0：无作用 1：复位 GPIO 端口 K
8	PJRST	GPIO 端口 J 复位 由软件置位或复位 0：无作用 1：复位 GPIO 端口 J
7	PHRST	GPIO 端口 H 复位 由软件置位或复位

0：无作用
1：复位 GPIO 端口 H

6 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

5 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

4 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

3 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

2 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

1 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

0 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

6.3.9. 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
UART7RST	UART6RST	DACRST	DACHOLDRST	CTCRST	保留		I2C3RST	I2C2RST	I2C1RST	I2C0RST	UART4RST	UART3RST	USART2RST	USART1RST	MDIORST
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
SPI2RST	SPI1RST	RSPDIFRST	保留	TIMER51RST	TIMER50RST	TIMER31RST	TIMER30RST	TIMER23RST	TIMER22RST	TIMER6RST	TIMER5RST	TIMER4RST	TIMER3RST	TIMER2RST	TIMER1RST
rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	UART7RST	UART7 复位 由软件置位或复位 0：无作用 1：复位 UART7
30	UART6RST	UART6 复位 由软件置位或复位 0：无作用 1：复位 UART6
29	DACRST	DAC 复位 由软件置位或复位 0：无作用 1：复位 DAC
28	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
27	CTCRST	CTC 复位 由软件置位或复位 0：无作用 1：复位 CTC
26:25	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	MDIORST	MDIO 复位 由软件置位或复位 0：无作用 1：复位 MDIO
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	RSPDIFRST	RSPDIF 复位 由软件置位或复位 0：无作用 1：复位 RSPDIF
12	保留	必须保持复位值。
11	TIMER51RST	TIMER51 复位 由软件置位或复位

		0：无作用 1：复位 TIMER51
10	TIMER50RST	TIMER50 复位 由软件置位或复位 0：无作用 1：复位 TIMER50
9	TIMER31RST	TIMER31 复位 由软件置位或复位 0：无作用 1：复位 TIMER31
8	TIMER30RST	TIMER30 复位 由软件置位或复位 0：无作用 1：复位 TIMER30
7	TIMER23RST	TIMER23 复位 由软件置位或复位 0：无作用 1：复位 TIMER23
6	TIMER6RST	TIMER6 复位 由软件置位或复位 0：无作用 1：复位 TIMER6
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

6.3.10. 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
TRIGSELRST	EDOUTRST	TIMER44RST	TIMER43RST	TIMER42RST	TIMER41RST	TIMER40RST	SAI2RST	SAI1RST	SAI0RST	SPI5RST	SPI4RST	HPDFRST	TIMER16RST	TIMER15RST	TIMER14RST
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		SPI3RST	SPI0RST	保留	ADC2RST	ADC1RST	ADC0RST	保留		USART5RST	USART0RST	保留		TIMER7RST	TIMER0RST
		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	EDOUTRST	EDOUT 复位 由软件置位或复位 0：无作用 1：复位 EDOUT
29	TIMER44RST	TIMER44 复位 由软件置位或复位 0：无作用 1：复位 TIMER44
28	TIMER43RST	TIMER43 复位 由软件置位或复位 0：无作用 1：复位 TIMER43
27	TIMER42RST	TIMER42 复位 由软件置位或复位

0：无作用
1：复位 TIMER42

26	TIMER41RST	TIMER41 复位 由软件置位或复位 0：无作用 1：复位 TIMER41
25	TIMER40RST	TIMER40 复位 由软件置位或复位 0：无作用 1：复位 TIMER40
24	SAI2RST	SAI2 复位 由软件置位或复位 0：无作用 1：复位 SAI2
23	SAI1RST	SAI1 复位 由软件置位或复位 0：无作用 1：复位 SAI1
22	SAI0RST	SAI0 复位 由软件置位或复位 0：无作用 1：复位 SAI0
21	SPI5RST	SPI5 复位 由软件置位或复位 0：无作用 1：复位 SPI5
20	SPI4RST	SPI4 复位 由软件置位或复位 0：无作用 1：复位 SPI4
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
15:14	保留	必须保持复位值。
13	SPI3RST	SPI3 复位 由软件置位或复位 0：无作用 1：复位 SPI3
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	ADC2RST	ADC2 复位 由软件置位或复位 0：无作用 1：复位所有 ADC2
9	ADC1RST	ADC1 复位 由软件置位或复位 0：无作用 1：复位所有 ADC1
8	ADC0RST	ADC0 复位 由软件置位或复位 0：无作用 1：复位所有 ADC0
7:6	保留	必须保持复位值。
5	USART5RST	USART5 复位 由软件置位或复位 0：无作用 1：复位 USART5
4	USART0RST	USART0 复位 由软件置位或复位 0：无作用 1：复位 USART0

3:2	保留	必须保持复位值。
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

6.3.11. 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															

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

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	WWDGTRST	WWDGT 复位 由软件置位或复位 0：无作用 1：复位 WWDGT
0	TLIRST	TLI 复位 由软件置位或复位 0：无作用 1：复位 TLI

6.3.12. APB4 复位寄存器（RCU_APB4RST）

地址偏移：0x2C
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											PMURST	LPDTSRST	VREFRST	CMPRST	SYSCFGRST
											rw	rw	rw	rw	rw

位/位域	名称	描述
31:5	保留	必须保持复位值。
4	PMURST	PMU 复位 由软件置位或复位 0：无作用 1：复位 PMU
3	LPDTSRST	LPDTS 复位 由软件置位或复位 0：无作用 1：复位 LPDTS
2	VREFRST	VREF 复位 由软件置位或复位 0：无作用 1：复位 VREF
1	CMPRST	CMP 复位 由软件置位或复位 0：无作用 1：复位 CMP
0	SYSCFGRST	SYSCFG 复位 由软件置位或复位 0：无作用 1：复位 SYSCFG

6.3.13. AHB1 使能寄存器（RCU_AHB1EN）

地址偏移：0x30

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	USBHS1ULPIEN	USBHS1EN	ENET0PTPEN	ENET0RXEN	ENET0TXEN	ENET0EN	保留	DMAMUXEN	DMA1EN	DMA0EN	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
USBHS0ULPIEN	USBHS0EN	保留										ENET1PTPEN	ENET1RXEN	ENET1TXEN	ENET1EN

rw rw rw rw rw rw

位/位域	名称	描述
31	保留	必须保持复位值.
30	USBHS1ULPIEN	USBHS1 ULPI 时钟使能 由软件置位或复位 0：关闭 USBHS1 ULPI 时钟 1：开启 USBHS1 ULPI 时钟
29	USBH1SEN	USBHS1 时钟使能 由软件置位或复位 0：关闭 USBHS1 时钟 1：开启 USBHS1 时钟
28	ENET0PTPEN	以太网 0 PTP 时钟使能 由软件置位或复位 0：关闭以太网 0 PTP 时钟 1：开启以太网 0 PTP 时钟
27	ENET0RXEN	以太网 0 RX 时钟使能 由软件置位或复位 0：关闭以太网 0 RX 时钟 1：开启以太网 0 RX 时钟
26	ENET0TXEN	以太网 0 TX 时钟使能 由软件置位或复位 0：关闭以太网 0 TX 时钟 1：开启以太网 0 TX 时钟
25	ENE0TEN	以太网 0 时钟使能 由软件置位或复位 0：关闭以太网 0 时钟 1：开启以太网 0 时钟
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	USBHS0ULPIEN	USBHS0 ULPI 时钟使能 由软件置位或复位 0：关闭 USBHS0 ULPI 时钟 1：开启 USBHS0 ULPI 时钟
14	USBH0SEN	USBHS0 时钟使能 由软件置位或复位 0：关闭 USBHS0 时钟 1：开启 USBHS0 时钟
13:4	保留	必须保持复位值。
3	ENET1PTPEN	以太网 1 PTP 时钟使能 由软件置位或复位 0：关闭以太网 1 PTP 时钟 1：开启以太网 1 PTP 时钟
2	ENET1RXEN	以太网 1 RX 时钟使能 由软件置位或复位 0：关闭以太网 1 RX 时钟 1：开启以太网 1 RX 时钟
1	ENET1TXEN	以太网 1 TX 时钟使能 由软件置位或复位 0：关闭以太网 1 TX 时钟 1：开启以太网 1 TX 时钟
0	ENE1TEN	以太网 1 时钟使能 由软件置位或复位 0：关闭以太网 1 时钟 1：开启以太网 1 时钟

6.3.14. AHB2 使能寄存器（RCU_AHB2EN）

地址偏移：0x34
复位值：0x0000 0100

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							RAMECC	TMUEN	TRNGEN	保留	HAUEN	CAUEN	SDIO1EN	FACEN	DCIEN

	MU1EN								
	rw	rw	rw		rw	rw	rw	rw	rw

位/位域	名称	描述
31:9	保留	必须保持复位值。
8	RAMECCMU1EN	RAMECCMU1 时钟使能 由软件置位或复位 0：关闭 RAMECCMU1 时钟 1：开启 RAMECCMU1 时钟
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	保留	必须保持复位值。
4	HAUEN	HAU 时钟使能 由软件置位或复位 0：关闭 HAU 时钟 1：开启 HAU 时钟
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	SDIO1EN	SDIO1 时钟使能 由软件置位或复位 0：关闭 SDIO1 时钟 1：开启 SDIO1 时钟
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	DCIEN	DCI 时钟使能 由软件置位或复位 0：关闭 DCI 时钟 1：开启 DCI 时钟

6.3.15. AHB3 使能寄存器（RCU_AHB3EN）

地址偏移：0x38

复位值：0x0000 8400

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
CPUEN	保留				RAMECCMU0EN	RTDEC1EN	RTDEC0EN	保留	OSPI1EN	OSPI0EN	OSPIMEN	MDMAEN	SDIO0EN	IPAEN	EXMCEN
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	CPUEN	CPU 时钟使能 由软件置位或复位 0：关闭 CPU 时钟 1：开启 CPU 时钟
14:11	保留	必须保持复位值。
10	RAMECCMU0EN	RAMECCMU0 时钟使能 由软件置位或复位 0：关闭 RAMECCMU0 时钟 1：开启 RAMECCMU0 时钟
9	RTDEC1EN	RTDEC1 时钟使能 由软件置位或复位 0：关闭 RTDEC1 时钟 1：开启 RTDEC1 时钟
8	RTDEC0EN	RTDEC0 时钟使能 由软件置位或复位 0：关闭 RTDEC0 时钟 1：开启 RTDEC0 时钟
7	保留	必须保持复位值。
6	OSPI1EN	OSPI1 时钟使能 由软件置位或复位 0：关闭 OSPI1 时钟 1：开启 OSPI1 时钟
5	OSPI0EN	OSPI0 时钟使能 由软件置位或复位 0：关闭 OSPI0 时钟

1：开启 OSPI0 时钟

4	OSPIMEN	OSPIM 时钟使能 由软件置位或复位 0：关闭 OSPIM 时钟 1：开启 OSPIM 时钟
3	MDMAEN	MDMA 时钟使能 由软件置位或复位 0：关闭 MDMA 时钟 1：开启 MDMA 时钟
2	SDIO0EN	SDIO0 时钟使能 由软件置位或复位 0：关闭 SDIO0 时钟 1：开启 SDIO0 时钟
1	IPAEN	IPAEN 时钟使能 由软件置位或复位 0：关闭 IPAEN 时钟 1：开启 IPAEN 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

6.3.16. AHB4 使能寄存器（RCU_AHB4EN）

地址偏移：0x3C
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															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HWSEMEN	CRCEN	BKPSRAMEN	保留			PKEN	PJEN	PHEN	PGEN	PFEN	PEEN	PDEN	PCEN	PBEN	PAEN
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	HWSEMEN	HWSEM 时钟使能 由软件置位或复位 0：关闭 HWSEM 时钟

		1：开启 HWSEM 时钟
14	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
13	BKPSRAMEN	BKPSRAM 时钟使能 由软件置位或复位 0：关闭 BKPSRAM 时钟 1：开启 BKPSRAM 时钟
12:10	保留	必须保持复位值。
9	PKEN	GPIO 端口 K 时钟使能 由软件置位或复位 0：关闭 GPIO 端口 K 时钟 1：开启 GPIO 端口 K 时钟
8	PJEN	GPIO 端口 J 时钟使能 由软件置位或复位 0：关闭 GPIO 端口 J 时钟 1：开启 GPIO 端口 J 时钟
7	PHEN	GPIO 端口 H 时钟使能 由软件置位或复位 0：关闭 GPIO 端口 H 时钟 1：开启 GPIO 端口 H 时钟
6	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
5	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
4	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
3	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
2	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

1 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

0 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

6.3.17. 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
UART7EN	UART6EN	DACEN	DACHOLDEN	CTCEN	保留		I2C3EN	I2C2EN	I2C1EN	I2C0EN	UART4EN	UART3EN	USART2EN	USART1EN	MDIOEN
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
SPI2EN	SPI1EN	RSPDIFEN	保留	TIMER51EN	TIMER50EN	TIMER31EN	TIMER30EN	TIMER23EN	TIMER22EN	TIMER6EN	TIMER5EN	TIMER4EN	TIMER3EN	TIMER2EN	TIMER1EN
rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	UART7EN	UART7 时钟使能 由软件置位或复位 0：关闭 UART7 时钟 1：开启 UART7 时钟
30	UART6EN	UART6 时钟使能 由软件置位或复位 0：关闭 UART6 时钟 1：开启 UART6 时钟
29	DACEN	DAC 时钟使能 由软件置位或复位 0：关闭 DAC 时钟 1：开启 DAC 时钟
28	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
27	CTCEN	CTC 保持时钟使能 由软件置位或复位 0：关闭 CTC 保持时钟 1：开启 CTC 保持时钟
26:25	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	MDIOEN	MDIO 时钟使能 由软件置位或复位 0：关闭 MDIO 时钟 1：开启 MDIO 时钟
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	RSPDIFEN	RSPDIF 时钟使能 由软件置位或复位 0：关闭 RSPDIF 时钟 1：开启 RSPDIF 时钟
12	保留	必须保持复位值。
11	TIMER51EN	TIMER51 时钟使能 由软件置位或复位 0：关闭 TIMER51 时钟 1：开启 TIMER51 时钟
10	TIMER51EN	TIMER51 时钟使能 由软件置位或复位 0：关闭 TIMER51 时钟 1：开启 TIMER51 时钟
9	TIMER31EN	TIMER31 时钟使能 由软件置位或复位 0：关闭 TIMER31 时钟 1：开启 TIMER31 时钟
8	TIMER30EN	TIMER30 时钟使能 由软件置位或复位 0：关闭 TIMER30 时钟 1：开启 TIMER30 时钟
7	TIMER23EN	TIMER23 时钟使能 由软件置位或复位 0：关闭 TIMER23 时钟 1：开启 TIMER23 时钟
6	TIMER22EN	TIMER22 时钟使能 由软件置位或复位

0：关闭 TIMER22 时钟
1：开启 TIMER22 时钟

5	TIMER6EN	TIMER6 时钟使能 由软件置位或复位 0：关闭 TIMER6 时钟 1：开启 TIMER6 时钟
4	TIMER5EN	TIMER5 时钟使能 由软件置位或复位 0：关闭 TIMER5 时钟 1：开启 TIMER5 时钟
3	TIMER4EN	TIMER4 时钟使能 由软件置位或复位 0：关闭 TIMER4 时钟 1：开启 TIMER4 时钟
2	TIMER3EN	TIMER3 时钟使能 由软件置位或复位 0：关闭 TIMER3 时钟 1：开启 TIMER3 时钟
1	TIMER2EN	TIMER2 时钟使能 由软件置位或复位 0：关闭 TIMER2 时钟 1：开启 TIMER2 时钟
0	TIMER1EN	TIMER1 时钟使能 由软件置位或复位 0：关闭 TIMER1 时钟 1：开启 TIMER1 时钟

6.3.18. 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	EDOUTE N	TIMER44 EN	TIMER43 EN	TIMER42 EN	TIMER41 EN	TIMER40 EN	SAI2EN	SAI1EN	SAI0EN	SPI5EN	SPI4EN	HPDFEN	TIMER16 EN	TIMER15 EN	TIMER14 EN
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		SPI3EN	SPI0EN	保留	ADC2EN	ADC1EN	ADC0EN	保留		USART5 EN	USART0 EN	保留		TIMER7E N	TIMER0E N
		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	EDOUTEN	EDOUT 时钟使能 由软件置位或复位 0：关闭 EDOUT 时钟 1：开启 EDOUT 时钟
29	TIMER44EN	TIMER44 时钟使能 由软件置位或复位 0：关闭 TIMER44 时钟 1：开启 TIMER44 时钟
28	TIMER43EN	TIMER43 时钟使能 由软件置位或复位 0：关闭 TIMER43 时钟 1：开启 TIMER43 时钟
27	TIMER42EN	TIMER42 时钟使能 由软件置位或复位 0：关闭 TIMER42 时钟 1：开启 TIMER42 时钟
26	TIMER41EN	TIMER41 时钟使能 由软件置位或复位 0：关闭 TIMER41 时钟 1：开启 TIMER41 时钟
25	TIMER40EN	TIMER40 时钟使能 由软件置位或复位 0：关闭 TIMER40 时钟 1：开启 TIMER40 时钟
24	SAI2EN	SAI2 时钟使能 由软件置位或复位 0：关闭 SAI2 时钟 1：开启 SAI2 时钟
23	SAI1EN	SAI1 时钟使能 由软件置位或复位 0：关闭 SAI1 时钟 1：开启 SAI1 时钟
22	SAI0EN	SAI0 时钟使能 由软件置位或复位 0：关闭 SAI0 时钟

		1：开启 SAI0 时钟
21	SPI5EN	SPI5 时钟使能 由软件置位或复位 0：关闭 SPI5 时钟 1：开启 SPI5 时钟
20	SPI4EN	SPI4 时钟使能 由软件置位或复位 0：关闭 SPI4 时钟 1：开启 SPI4 时钟
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	TIMER15 时钟使能 由软件置位或复位 0：关闭 TIMER15 时钟 1：开启 TIMER15 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
15:14	保留	必须保持复位值。
13	SPI3EN	SPI3 时钟使能 由软件置位或复位 0：关闭 SPI3 时钟 1：开启 SPI3 时钟
12	SPI0EN	SPI0 时钟使能 由软件置位或复位 0：关闭 SPI0 时钟 1：开启 SPI0 时钟
11	保留	必须保持复位值。
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
7:6	保留	必须保持复位值。
5	USART5EN	USART5 时钟使能 由软件置位或复位 0：关闭 USART5 时钟 1：开启 USART5 时钟
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
3:2	保留	必须保持复位值。
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

6.3.19. 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。

31	30	29	28	27	26	25	24	23	22	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														WWDGTEN	TLIEN

															rw	rw

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

6.3.20. APB4 使能寄存器（RCU_APB4EN）

地址偏移：0x4C
复位值：0x0000 0010

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											PMUEN	LPDTSEN	VREFEN	CMPEN	SYSCFGEN
											rw	rw	rw	rw	rw

位/位域	名称	描述
31:5	保留	必须保持复位值
4	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
3	LPDTSEN	LPDTS 时钟使能 由软件置位或复位 0：关闭 LPDTS 时钟 1：开启 LPDTS 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	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
0	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

6.3.21. AHB1 睡眠模式使能寄存器（RCU_AHB1SPEN）

地址偏移：0x50

复位值：0x7EE3 C00F

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	USBHS1ULPISPEN	USBHS1SPEN	ENET0PTPSPEN	ENET0RXSPEN	ENET0TXSPEN	ENET0SPEN	保留	DMAMUXSPEN	DMA1SPEN	DMA0SPEN	保留			SRAM1SPEN	SRAM0SPEN
	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
USBHS0ULPISPEN	USBHS0SPEN	保留										ENET1PTPSPEN	ENET1RXSPEN	ENET1TXSPEN	ENET1SPEN
rw	rw											rw	rw	rw	rw

位/位域	名称	描述
31	保留	必须保持复位值。
30	USBHS1ULPISPEN	在睡眠模式下 USBHS1 ULPI 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 USBHS1 ULPI 时钟 1：在睡眠模式下开启 USBHS1 ULPI 时钟
29	USBHS1SPEN	在睡眠模式下 USBHS1 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 USBHS1 时钟 1：在睡眠模式下开启 USBHS1 时钟
28	ENET0PTPSPEN	在睡眠模式下以太网 0 PTP 时钟使能 由软件置位或复位 0：在睡眠模式下关闭以太网 0 PTP 时钟 1：在睡眠模式下开启以太网 0 PTP 时钟
27	ENET0RXSPEN	在睡眠模式下以太网 0 RX 时钟使能 由软件置位或复位

0：在睡眠模式下关闭以太网 0 RX 时钟
1：在睡眠模式下开启以太网 0 RX 时钟

26	ENET0TXSPEN	在睡眠模式下以太网 0 TX 时钟使能 由软件置位或复位 0：在睡眠模式下关闭以太网 0 TX 时钟 1：在睡眠模式下开启以太网 0 TX 时钟
25	ENET0SPEN	在睡眠模式下以太网 0 时钟使能 由软件置位或复位 0：在睡眠模式下关闭以太网 0 时钟 1：在睡眠模式下开启以太网 0 时钟
24	保留	必须保持复位值。
23	DMAMUXSPEN	在睡眠模式下 DMAMUX 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 DMAMUX 时钟 1：在睡眠模式下开启 DMAMUX 时钟
22	DMA1SPEN	在睡眠模式下 DMA1 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 DMA1 时钟 1：在睡眠模式下开启 DMA1 时钟
21	DMA0SPEN	在睡眠模式下 DMA0 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 DMA0 时钟 1：在睡眠模式下开启 DMA0 时钟
20:18	保留	必须保持复位值。
17	SRAM1SPEN	在睡眠模式下 SRAM1 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 SRAM1 时钟 1：在睡眠模式下开启 SRAM1 时钟
16	SRAM0SPEN	在睡眠模式下 SRAM0 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 SRAM0 时钟 1：在睡眠模式下开启 SRAM0 时钟
15	USBHS0ULPISPEN	在睡眠模式下 USBHS0 ULPI 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 USBHS0 ULPI 时钟 1：在睡眠模式下开启 USBHS0 ULPI 时钟
14	USBHS0SPEN	在睡眠模式下 USBHS0 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 USBHS0 时钟

1：在睡眠模式下开启 USBHS0 时钟

13:4	保留	必须保持复位值。
3	ENET1PTPSPEN	在睡眠模式下以太网 1 PTP 时钟使能 由软件置位或复位 0：在睡眠模式下关闭以太网 1 PTP 时钟 1：在睡眠模式下开启以太网 1 PTP 时钟
2	ENET1RXSPEN	在睡眠模式下以太网 1 RX 时钟使能 由软件置位或复位 0：在睡眠模式下关闭以太网 1 RX 时钟 1：在睡眠模式下开启以太网 1 RX 时钟
1	ENET1TXSPEN	在睡眠模式下以太网 1 TX 时钟使能 由软件置位或复位 0：在睡眠模式下关闭以太网 1 TX 时钟 1：在睡眠模式下开启以太网 1 TX 时钟
0	ENET1SPEN	在睡眠模式下以太网 1 时钟使能 由软件置位或复位 0：在睡眠模式下关闭以太网 1 时钟 1：在睡眠模式下开启以太网 1 时钟

6.3.22. AHB2 睡眠模式使能寄存器（RCU_AHB2SPEN）

地址偏移：0x54

复位值：0x0000 01DF

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							RAMECCMU1SPEN	TMUSPEN	TRNGSPEN	保留	HAUSPEN	CAUSPEN	SDIO1SPEN	FACSPEN	DCISPEN
							rw	rw	rw		rw	rw	rw	rw	rw

位/位域	名称	描述
31:9	保留	必须保持复位值。
8	RAMECCMU1SPEN	在睡眠模式下 RAMECCMU1 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 RAMECCMU1 时钟 1：在睡眠模式下开启 RAMECCMU1 时钟
7	TMUSPEN	在睡眠模式下 TMU 时钟使能 由软件置位或复位

0：在睡眠模式下关闭 TMU 时钟
1：在睡眠模式下开启 TMU 时钟

6	TRNGSPEN	在睡眠模式下 TRNG 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TRNG 时钟 1：在睡眠模式下开启 TRNG 时钟
5	保留	必须保持复位值。
4	HAUSPEN	在睡眠模式下 HAU 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 HAU 时钟 1：在睡眠模式下开启 HAU 时钟
3	CAUSPEN	在睡眠模式下 CAU 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 CAU 时钟 1：在睡眠模式下开启 CAU 时钟
2	SDIO1SPEN	在睡眠模式下 SDIO1 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 SDIO1 时钟 1：在睡眠模式下开启 SDIO1 时钟
1	FACSPEN	在睡眠模式下 FAC 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 FAC 时钟 1：在睡眠模式下开启 FAC 时钟
0	DCISPEN	在睡眠模式下 DCI 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 DCI 时钟 1：在睡眠模式下开启 DCI 时钟

6.3.23. AHB3 睡眠模式使能寄存器（RCU_AHB3SPEN）

地址偏移：0x58
复位值：0x0000 C77F

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
FMCSPEN	AXISRAMSPEN	保留			RAMECCMU0SPEN	RTDEC1SPEN	RTDEC0SPEN	保留	OSPI1SPEN	OSPI0SPEN	OSPIMSPEN	MDMASPEN	SDIO0SPEN	IPASPEN	EXMCSPEN
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	FMCSPEN	在睡眠模式下 FMC 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 FMC 时钟 1：在睡眠模式下开启 FMC 时钟
14	AXISRAMSPEN	在睡眠模式下 AXI SRAM 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 AXI SRAM 时钟 1：在睡眠模式下开启 AXI SRAM 时钟
13:11	保留	必须保持复位值。
10	RAMECCMU0SPEN	在睡眠模式下 RAMECCMU0 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 RAMECCMU0 时钟 1：在睡眠模式下开启 RAMECCMU0 时钟
9	RTDEC1SPEN	在睡眠模式下 RTDEC1 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 RTDEC1 时钟 1：在睡眠模式下开启 RTDEC1 时钟
8	RTDEC0SPEN	在睡眠模式下 RTDEC0 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 RTDEC0 时钟 1：在睡眠模式下开启 RTDEC0 时钟
7	保留	必须保持复位值。
6	OSPI1SPEN	在睡眠模式下 OSPI1 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 OSPI1 时钟 1：在睡眠模式下开启 OSPI1 时钟
5	OSPI0SPEN	在睡眠模式下 OSPI0 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 OSPI0 时钟 1：在睡眠模式下开启 OSPI0 时钟
4	OSPIMSPEN	在睡眠模式下 OSPIM 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 OSPIM 时钟 1：在睡眠模式下开启 OSPIM 时钟
3	MDMASPEN	在睡眠模式下 MDMA 时钟使能 由软件置位或复位

0：在睡眠模式下关闭 MDMA 时钟
1：在睡眠模式下开启 MDMA 时钟

2	SDIO0SPEN	在睡眠模式下 SDIO0 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 SDIO0 时钟 1：在睡眠模式下开启 SDIO0 时钟
1	IPASPEN	在睡眠模式下 IPA 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 IPA 时钟 1：在睡眠模式下开启 IPA 时钟
0	EXMCSPEN	在睡眠模式下 EXMC 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 EXMC 时钟 1：在睡眠模式下开启 EXMC 时钟

6.3.24. AHB4 睡眠模式使能寄存器（RCU_AHB4SPEN）

地址偏移：0x5C
复位值：0x0000 63FF

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	CRCSPEN	BKPSRAMSPEN	保留			PKSPEN	PJSPEN	PHSPEN	PGSPEN	PFSPEN	PESPEN	PDSPEN	PCSPEN	PBSPEN	PASPEN
	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	CRCSPEN	在睡眠模式下 CRC 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 CRC 时钟 1：在睡眠模式下开启 CRC 时钟
13	BKPSRAMSPEN	在睡眠模式下备份域 SRAM 时钟使能 由软件置位或复位 0：在睡眠模式下关闭备份域 SRAM 时钟 1：在睡眠模式下开启备份域 SRAM 时钟
12:10	保留	必须保持复位值。
9	PKSPEN	在睡眠模式下 GPIO 端口 K 时钟使能

由软件置位或复位
0：在睡眠模式下关闭 GPIO 端口 K 时钟
1：在睡眠模式下开启 GPIO 端口 K 时钟

8 PJSPEN 在睡眠模式下 GPIO 端口 J 时钟使能
由软件置位或复位
0：在睡眠模式下关闭 GPIO 端口 J 时钟
1：在睡眠模式下开启 GPIO 端口 J 时钟

7 PHSPEN 在睡眠模式下 GPIO 端口 H 时钟使能
由软件置位或复位
0：在睡眠模式下关闭 GPIO 端口 H 时钟
1：在睡眠模式下开启 GPIO 端口 H 时钟

6 PGSPEN 在睡眠模式下 GPIO 端口 G 时钟使能
由软件置位或复位
0：在睡眠模式下关闭 GPIO 端口 G 时钟
1：在睡眠模式下开启 GPIO 端口 G 时钟

5 PFSPEN 在睡眠模式下 GPIO 端口 F 时钟使能
由软件置位或复位
0：在睡眠模式下关闭 GPIO 端口 F 时钟
1：在睡眠模式下开启 GPIO 端口 F 时钟

4 PEEN 在睡眠模式下 GPIO 端口 E 时钟使能
由软件置位或复位
0：在睡眠模式下关闭 GPIO 端口 E 时钟
1：在睡眠模式下开启 GPIO 端口 E 时钟

3 PDSPEN 在睡眠模式下 GPIO 端口 D 时钟使能
由软件置位或复位
0：在睡眠模式下关闭 GPIO 端口 D 时钟
1：在睡眠模式下开启 GPIO 端口 D 时钟

2 PCSPEN 在睡眠模式下 GPIO 端口 C 时钟使能
由软件置位或复位
0：在睡眠模式下关闭 GPIO 端口 C 时钟
1：在睡眠模式下开启 GPIO 端口 C 时钟

1 PBSPEN 在睡眠模式下 GPIO 端口 B 时钟使能
由软件置位或复位
0：在睡眠模式下关闭 GPIO 端口 B 时钟
1：在睡眠模式下开启 GPIO 端口 B 时钟

0 PASPEN 在睡眠模式下 GPIO 端口 A 时钟使能
由软件置位或复位
0：在睡眠模式下关闭 GPIO 端口 A 时钟
1：在睡眠模式下开启 GPIO 端口 A 时钟

6.3.25. APB1 睡眠模式使能寄存器（RCU_APB1SPEN）

地址偏移：0x60

复位值：0xF9FF EFFF

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
UART7SPEN	UART6SPEN	DACSPEN	DACHOLDSPEN	CTCSPEN	保留		I2C3SPEN	I2C2SPEN	I2C1SPEN	I2C0SPEN	UART4SPEN	UART3SPEN	USART2SPEN	USART1SPEN	MDIOSPEN
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
SPI2SPEN	SPI1SPEN	RSPDIFSPEN	保留	TIMER51SPEN	TIMER50SPEN	TIMER31SPEN	TIMER30SPEN	TIMER23SPEN	TIMER22SPEN	TIMER6SPEN	TIMER5SPEN	TIMER4SPEN	TIMER3SPEN	TIMER2SPEN	TIMER1SPEN
rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	UART7SPEN	在睡眠模式下 UART7 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 UART7 时钟 1：在睡眠模式下开启 UART7 时钟
30	UART6SPEN	在睡眠模式下 UART6 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 UART6 时钟 1：在睡眠模式下开启 UART6 时钟
29	DACSPEN	在睡眠模式下 DAC 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 DAC 时钟 1：在睡眠模式下开启 DAC 时钟
28	DACHOLDSPEN	在睡眠模式下 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
27	CTCSPEN	在睡眠模式下 CTC 保持时钟使能 由软件置位或复位 0：在睡眠模式下关闭 CTC 保持时钟 1：在睡眠模式下开启 CTC 保持时钟
26:25	保留	必须保持复位值。
24	I2C3SPEN	在睡眠模式下 I2C3 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 I2C3 时钟 1：在睡眠模式下开启 I2C3 时钟

23	I2C2SPEN	在睡眠模式下 I2C2 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 I2C2 时钟 1：在睡眠模式下开启 I2C2 时钟
22	I2C1SPEN	在睡眠模式下 I2C1 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 I2C1 时钟 1：在睡眠模式下开启 I2C1 时钟
21	I2C0SPEN	在睡眠模式下 I2C0 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 I2C0 时钟 1：在睡眠模式下开启 I2C0 时钟
20	UART4SPEN	在睡眠模式下 UART4 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 UART4 时钟 1：在睡眠模式下开启 UART4 时钟
19	UART3SPEN	在睡眠模式下 UART3 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 UART3 时钟 1：在睡眠模式下开启 UART3 时钟
18	USART2SPEN	在睡眠模式下 USART2 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 USART2 时钟 1：在睡眠模式下开启 USART2 时钟
17	USART1SPEN	在睡眠模式下 USART1 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 USART1 时钟 1：在睡眠模式下开启 USART1 时钟
16	MDIOSPEN	在睡眠模式下 MDIO 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 MDIO 时钟 1：在睡眠模式下开启 MDIO 时钟
15	SPI2SPEN	在睡眠模式下 SPI2 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 SPI2 时钟 1：在睡眠模式下开启 SPI2 时钟
14	SPI1SPEN	在睡眠模式下 SPI1 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 SPI1 时钟

		1：在睡眠模式下开启 SPI1 时钟
13	RSPDIFSPEN	在睡眠模式下 RSPDIF 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 RSPDIF 时钟 1：在睡眠模式下开启 RSPDIF 时钟
12	保留	必须保持复位值。
11	TIMER51SPEN	在睡眠模式下 TIMER51 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER51 时钟 1：在睡眠模式下开启 TIMER51 时钟
10	TIMER50SPEN	在睡眠模式下 TIMER50 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER50 时钟 1：在睡眠模式下开启 TIMER50 时钟
9	TIMER31SPEN	在睡眠模式下 TIMER31 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER31 时钟 1：在睡眠模式下开启 TIMER31 时钟
8	TIMER30SPEN	在睡眠模式下 TIMER30 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER30 时钟 1：在睡眠模式下开启 TIMER30 时钟
7	TIMER23SPEN	在睡眠模式下 TIMER23 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER23 时钟 1：在睡眠模式下开启 TIMER23 时钟
6	TIMER22SPEN	在睡眠模式下 TIMER22 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER22 时钟 1：在睡眠模式下开启 TIMER22 时钟
5	TIMER6SPEN	在睡眠模式下 TIMER6 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER6 时钟 1：在睡眠模式下开启 TIMER6 时钟
4	TIMER5SPEN	在睡眠模式下 TIMER5 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER5 时钟 1：在睡眠模式下开启 TIMER5 时钟
3	TIMER4SPEN	在睡眠模式下 TIMER4 时钟使能

由软件置位或复位
0：在睡眠模式下关闭 TIMER4 时钟
1：在睡眠模式下开启 TIMER4 时钟

2	TIMER3SPEN	在睡眠模式下 TIMER3 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER3 时钟 1：在睡眠模式下开启 TIMER3 时钟
1	TIMER2SPEN	在睡眠模式下 TIMER2 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER2 时钟 1：在睡眠模式下开启 TIMER2 时钟
0	TIMER1SPEN	在睡眠模式下 TIMER1 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER1 时钟 1：在睡眠模式下开启 TIMER1 时钟

6.3.26. APB2 睡眠模式使能寄存器（RCU_APB2SPEN）

地址偏移：0x64
复位值：0xFFFF 3733

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
TRIGSELSPEN	EDOUTSPEN	TIMER44SPEN	TIMER43SPEN	TIMER42SPEN	TIMER41SPEN	TIMER40SPEN	SAI2SPEN	SAI1SPEN	SAI0SPEN	SPI5SPEN	SPI4SPEN	HPDFSPEN	TIMER16SPEN	TIMER15SPEN	TIMER14SPEN
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		SPI3SPEN	SPI0SPEN	保留	ADC2SPEN	ADC1SPEN	ADC0SPEN	保留		USART5SPEN	USART0SPEN	保留		TIMER7SPEN	TIMER0SPEN
		rw	rw		rw	rw	rw			rw	rw			rw	rw

位/位域	名称	描述
31	TRIGSELSPEN	在睡眠模式下 TRIGSEL 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TRIGSEL 时钟 1：在睡眠模式下开启 TRIGSEL 时钟
30	EDOUTSPEN	在睡眠模式下 EDOUT 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 EDOUT 时钟 1：在睡眠模式下开启 EDOUT 时钟
29	TIMER44SPEN	在睡眠模式下 TIMER44 时钟使能 由软件置位或复位

0：在睡眠模式下关闭 TIMER44 时钟
1：在睡眠模式下开启 TIMER44 时钟

28	TIMER43SPEN	在睡眠模式下 TIMER43 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER43 时钟 1：在睡眠模式下开启 TIMER43 时钟
27	TIMER42SPEN	在睡眠模式下 TIMER42 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER42 时钟 1：在睡眠模式下开启 TIMER42 时钟
26	TIMER41SPEN	在睡眠模式下 TIMER41 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER41 时钟 1：在睡眠模式下开启 TIMER41 时钟
25	TIMER40SPEN	在睡眠模式下 TIMER40 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER40 时钟 1：在睡眠模式下开启 TIMER40 时钟
24	SAI2SPEN	在睡眠模式下 SAI2 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 SAI2 时钟 1：在睡眠模式下开启 SAI2 时钟
23	SAI1SPEN	在睡眠模式下 SAI1 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 SAI1 时钟 1：在睡眠模式下开启 SAI1 时钟
22	SAI0SPEN	在睡眠模式下 SAI0 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 SAI0 时钟 1：在睡眠模式下开启 SAI0 时钟
21	SPI5SPEN	在睡眠模式下 SPI5 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 SPI5 时钟 1：在睡眠模式下开启 SPI5 时钟
20	SPI4SPEN	在睡眠模式下 SPI4 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 SPI4 时钟 1：在睡眠模式下开启 SPI4 时钟
19	HPDFSPEN	在睡眠模式下 HPDF 时钟使能

		由软件置位或复位 0：在睡眠模式下关闭 HPDF 时钟 1：在睡眠模式下开启 HPDF 时钟
18	TIMER16SPEN	在睡眠模式下 TIMER16 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER16 时钟 1：在睡眠模式下开启 TIMER16 时钟
17	TIMER15SPEN	在睡眠模式下 TIMER15 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER15 时钟 1：在睡眠模式下开启 TIMER15 时钟
16	TIMER14SPEN	在睡眠模式下 TIMER14 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER14 时钟 1：在睡眠模式下开启 TIMER14 时钟
15:14	保留	必须保持复位值。
13	SPI3SPEN	在睡眠模式下 SPI3 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 SPI3 时钟 1：在睡眠模式下开启 SPI3 时钟
12	SPI0SPEN	在睡眠模式下 SPI0 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 SPI0 时钟 1：在睡眠模式下开启 SPI0 时钟
11	保留	必须保持复位值。
10	ADC2SPEN	在睡眠模式下 ADC2 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 ADC2 时钟 1：在睡眠模式下开启 ADC2 时钟
9	ADC1SPEN	在睡眠模式下 ADC1 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 ADC1 时钟 1：在睡眠模式下开启 ADC1 时钟
8	ADC0SPEN	在睡眠模式下 ADC0 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 ADC0 时钟 1：在睡眠模式下开启 ADC0 时钟
7:6	保留	必须保持复位值。

5	USART5SPEN	在睡眠模式下 USART5 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 USART5 时钟 1：在睡眠模式下开启 USART5 时钟
4	USART0SPEN	在睡眠模式下 USART0 时钟使能 由软件置位或复位 0：关闭在睡眠模式下 USART0 时钟 1：开启在睡眠模式下 USART0 时钟
3:2	保留	必须保持复位值。
1	TIMER7SPEN	在睡眠模式下 TIMER7 时钟使能 由软件置位或复位 0：关闭在睡眠模式下 TIMER7 时钟 1：开启在睡眠模式下 TIMER7 时钟
0	TIMER0SPEN	在睡眠模式下 TIMER0 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER0 时钟 1：在睡眠模式下开启 TIMER0 时钟

6.3.27. APB3 睡眠模式使能寄存器（RCU_APB3SPEN）

地址偏移：0x68

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。



位/位域	名称	描述
31:2	保留	必须保持复位值。
1	WWDGTSPEN	在睡眠模式下 WWDGT 时钟使能 由软件置位或复位 0：关闭在睡眠模式下 WWDGT 时钟 1：开启在睡眠模式下 WWDGT 时钟
0	TLISPEN	在睡眠模式下 TLI 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TLI 时钟

1：在睡眠模式下开启 TLI 时钟

6.3.28. APB4 睡眠模式使能寄存器（RCU_APB4SPEN）

地址偏移：0x6C

复位值：0x0000 001F

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											PMUSPEN	LPDTSSPEN	VREFSPEN	CMPSPEN	SYSCFGSPEN
											rw	rw	rw	rw	rw

位/位域	名称	描述
31:5	保留	必须保持复位值。
4	PMUSPEN	在睡眠模式下 PMU 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 PMU 时钟 1：在睡眠模式下开启 PMU 时钟
3	LPDTSSPEN	在睡眠模式下 LPDTS 时钟使能 由软件置位或复位 0：关闭在睡眠模式下 LPDTS 时钟 1：开启在睡眠模式下 LPDTS 时钟
2	VREFSPEN	在睡眠模式下 VREF 时钟使能 由软件置位或复位 0：关闭在睡眠模式下 VREF 时钟 1：开启在睡眠模式下 VREF 时钟
1	CMPSPEN	在睡眠模式下 CMP 时钟使能 由软件置位或复位 0：关闭在睡眠模式下 CMP 时钟 1：开启在睡眠模式下 CMP 时钟
0	SYSCFGSPEN	在睡眠模式下 SYSCFG 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 SYSCFG 时钟 1：在睡眠模式下开启 SYSCFG 时钟

6.3.29. 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															BKPRST
															rw

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

位/位域	名称	描述
31:17	保留	必须保持复位值。
16	BKPRST	备份域复位 由软件置位或复位 0：无作用 1：复位备份域
15	RTCEN	RTC 时钟使能 由软件置位或复位 0：关闭 RTC 时钟 1：开启 RTC 时钟
14:10	保留	必须保持复位值。
9:8	RTCSRC[1:0]	RTC 时钟源选择 由软件置位或清零来控制RTC的时钟源。一旦RTC的时钟源选择后，除了将备份域复位否则时钟源不能被改变。 00：没有时钟 01：选择 CK_LXTAL 时钟作为 RTC 的时钟源 10：选择 CK_IRC32K 时钟作为 RTC 的时钟源 11：选择 CK_HXTAL / RTCDIV 时钟作为 RTC 的时钟源，请参考 RCU_CFG0 寄存器的 RTCDIV 位域。
7	保留	必须保持复位值。
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时钟监视器

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。

4:3	LXTALDRI[1:0]	LXTAL 驱动能力 由软件置位或复位。当备份域复位时将复位该值 00：弱驱动能力 01：中低驱动能力 10：中高驱动能力 11：强驱动能力 **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

6.3.30. 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	BOR RSTF	RSTFC	保留							
r	r	r	r	r	r	r	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														IRC32K STB	IRC32KE N
														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
23:2	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

6.3.31. PLL 时钟附加控制寄存器（RCU_PLLADDCTL）

地址偏移：0x80

复位值：0xFF81 0101

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
PLL2PEN	PLL2REN	PLL2QEN	PLL1PEN	PLL1REN	PLL1QEN	PLL0PEN	PLL0REN	PLL0QEN	PLL2Q[6:0]						
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	PLL1Q[6:0]							保留	PLL0Q[6:0]						
	rw								rw						

位/位域	名称	描述
31	PLL2PEN	PLL2P 分频器输出使能 由软件置位或复位。只有在 PLL2EN 位为 0 时 PLL2PEN 位才可写。 0：禁止 CK_PLL2P 输出 1：使能 CK_PLL2P 输出
30	PLL2REN	PLL2R 分频器输出使能 由软件置位或复位。只有在 PLL2EN 位为 0 时 PLL2REN 位才可写。 0：禁止 CK_PLL2R 输出 1：使能 CK_PLL2R 输出
29	PLL2QEN	PLL2Q 分频器输出使能 由软件置位或复位。只有在 PLL2EN 位为 0 时 PLL2QEN 位才可写。 0：禁止 CK_PLL2Q 输出 1：使能 CK_PLL2Q 输出
28	PLL1PEN	PLL1P 分频器输出使能 由软件置位或复位。只有在 PLL1EN 位为 0 时 PLL1PEN 位才可写。 0：禁止 CK_PLL1P 输出 1：使能 CK_PLL1P 输出
27	PLL1REN	PLL1R 分频器输出使能 由软件置位或复位。只有在 PLL1EN 位为 0 时 PLL1REN 位才可写。 0：禁止 CK_PLL1R 输出

		1：使能 CK_PLL1R 输出
26	PLL1QEN	PLL1Q 分频器输出使能 由软件置位或复位。只有在 PLL1EN 位为 0 时 PLL1QEN 位才可写。 0：禁止 CK_PLL1Q 输出 1：使能 CK_PLL1Q 输出
25	PLL0PEN	PLL0P 分频器输出使能 由软件置位或复位。只有在 PLL0EN 位为 0 时 PLL0PEN 位才可写。 0：禁止 CK_PLL0P 输出 1：使能 CK_PLL0P 输出
24	PLL0REN	PLL0R 分频器输出使能 由软件置位或复位。只有在 PLL0EN 位为 0 时 PLL0REN 位才可写。 0：禁止 CK_PLL0R 输出 1：使能 CK_PLL0R 输出
23	PLL0QEN	PLL0Q 分频器输出使能 由软件置位或复位。只有在 PLL0EN 位为 0 时 PLL0QEN 位才可写。 0：禁止 CK_PLL0Q 输出 1：使能 CK_PLL0Q 输出
22:16	PLL2Q[6:0]	PLL2Q 输出频率的分频系数（PLL2 VCO 时钟作为输入） 当 PLL2 被关闭时由软件置位或清零。这些位域用做将 PLL2 VCO 时钟（CK_PLL2VCO）分频生成 PLL2Q 输出时钟（CK_PLL2Q）。RCU_PLL2 寄存器的 PLL2N 位域对 CK_PLL2VCO 时钟进行了描述。 0000000：CK_PLL2Q = CK_PLL2VCO 0000001：CK_PLL2Q = CK_PLL2VCO / 2 0000010：CK_PLL2Q = CK_ PLL2VCO / 3. 0000011：CK_PLL2Q = CK_ PLL2VCO / 4 0000100：CK_PLL2Q = CK_ PLL2VCO / 5 … 1111111：CK_PLL2Q = CK_ PLL2VCO / 128
15	保留	必须保持复位值。
14:8	PLL1Q[6:0]	PLL1Q 输出频率的分频系数（PLL1 VCO 时钟作为输入） 当 PLL1 被关闭时由软件置位或清零。这些位域用做将 PLL1 VCO 时钟（CK_PLL1VCO）分频生成 PLL1Q 输出时钟（CK_PLL1Q）。RCU_PLL1 寄存器的 PLL1N 位域对 CK_PLL1VCO 时钟进行了描述。 0000000：CK_PLL1Q = CK_PLL1VCO 0000001：CK_ PLL1Q = CK_PLL1VCO / 2 0000010：CK_PLL1Q = CK_PLL1VCO / 3. 0000011：CK_PLL1Q = CK_PLL1VCO / 4 0000100：CK_PLL1Q = CK_PLL1VCO / 5 … 1111111：CK_PLL1Q = CK_PLL1VCO / 128

7	保留	必须保持复位值。
6:0	PLL0Q[6:0]	PLL0Q 输出频率的分频系数（PLL0 VCO 时钟作为输入） 当 PLL0 被关闭时由软件置位或清零。这些位域用做将 PLL0 VCO 时钟（CK_PLL0VCO）分频生成 PLL0Q 输出时钟（CK_PLL0Q）。CK_PLL0Q 输出可用于 USBHS（48M）、TRNG（48M）、SDIO。RCU_PLL0 寄存器的 PLL0N 位域对 CK_PLL0VCO 时钟进行了描述。 0000000：CK_PLL0Q = CK_PLL0VCO 0000001：CK_PLL0Q = CK_PLL0VCO / 2 0000010：CK_PLL0Q = CK_PLL0VCO / 3 0000011：CK_PLL0Q = CK_PLL0VCO / 4 0000100：CK_PLL0Q = CK_PLL0VCO / 5 … 1111111：CK_PLL0Q = CK_PLL0VCO / 128

6.3.32. PLL1 寄存器（RCU_PLL1）

地址偏移：0x84

复位值：0x0101 2020

配置PLL1时钟可参考下列公式：

CK_PLL1VCOSRC = CK_PLL1SRC / PLL1PSC

CK_PLL1VCO = CK_PLL1VCOSRC × （PLL1N + PLL1FRAN / 2^{13}）

CK_PLL1P = CK_ PLL1VCO / PLL1P

CK_PLL1R = CK_PLL1VCO / PLL1R

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	PLL1R[6:0]							保留	PLL1P[6:0]						
	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PLL1N[8:0]									PLL1PSC[5:0]					
	rw									rw					

位/位域	名称	描述
31	保留	必须保持复位值。
30:24	PLL1R[6:0]	PLL1R 输出频率的分频系数（PLL1 VCO 时钟作为输入） 当 PLL1 被关闭时由软件置位或清零。这些位域用做将 PLL1 VCO 时钟（CK_PLL1VCO）分频生成 PLL1R 输出时钟（CK_PLL1R）。RCU_PLL1 寄存器的 PLL1N 位域对 CK_PLL1VCO 时钟进行了描述。 0000000：CK_PLL1R = CK_PLL1VCO 0000001：CK_PLL1R = CK_PLL1VCO / 2 0000010：CK_PLL1R = CK_PLL1VCO / 3. 0000011：CK_PLL1R = CK_PLL1VCO / 4 0000100：CK_PLL1R = CK_PLL1VCO / 5

		… 1111111：CK_PLL1R = CK_PLL1VCO / 128
23	保留	必须保持复位值。
22:16	PLL1P[6:0]	PLL1P 输出频率的分频系数（PLL1 VCO 时钟作为输入） 当 PLL1 被关闭时由软件置位或清零。这些位域用做将 PLL1 VCO 时钟（CK_PLL1VCO）分频生成 PLL1P 输出时钟（CK_PLL1P）。RCU_PLL1 寄存器的 PLL1N 位域对 CK_PLL1VCO 时钟进行了描述。 0000000：CK_PLL1P = CK_PLL1VCO 0000001：CK_PLL1P = CK_PLL1VCO / 2 0000010：CK_PLL1P = CK_PLL1VCO / 3 0000011：CK_PLL1P = CK_PLL1VCO / 4 0000100：CK_PLL1P = CK_PLL1VCO / 5 … 1111111：CK_PLL1P = CK_PLL1VCO / 128
15	保留	必须保持复位值。
14:6	PLL1N[8:0]	PLL1 VCO 时钟倍频因子 当 PLL1 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 PLL1 VCO 源时钟（CK_PLL1VCOSRC）倍频生成 PLL1 VCO 输出时钟（CK_PLL1VCO）。RCU_PLL1 寄存器的 PLL1PSC 位域对 CK_PLL1VCOSRC 时钟进行了描述。 **注意：**CK_PLL1VCO 时钟频率范围必须在 150MHz 到 836MHz 之间 PLL1N 的值必须满足： 当 PLL1 小数锁存禁能时，PLL1N 的值必须满足：9 ≤ PLL1N ≤ 512 当 PLL1 小数锁存使能时，PLL1N 的值必须满足：12 ≤ PLL1N ≤ 508 000000000：保留 … 000000111：保留 000001000：PLL1N = 9 … 001000000：PLL1N = 65 001000001：PLL1N = 66 … 111111111：PLL1N = 512
5:0	PLL1PSC[5:0]	PLL1 VCO 源时钟分频器 当 PLL1 被关闭时由软件置位或清零。这些位域用做将 PLL1 源时钟（CK_PLL1SRC）分频生成 PLL1 VCO 源时钟（CK_PLL1VCOSRC）。RCU_PLLALL 寄存器的 PLL1SEL 位对 CK_PLL1SRC 时钟进行了描述。 VCO 源时钟频率范围必须在 1MHz 到 16MHz 之间 000000：保留 000001：CK_PLL1SRC 000010：CK_PLL1SRC / 2 000011：CK_PLL1SRC / 3

…

111111：CK_PLL1SRC / 63

6.3.33. PLL2 寄存器（RCU_PLL2）

地址偏移：0x88

复位值：0x0101 2020

配置PLL2时钟可参考下列公式：

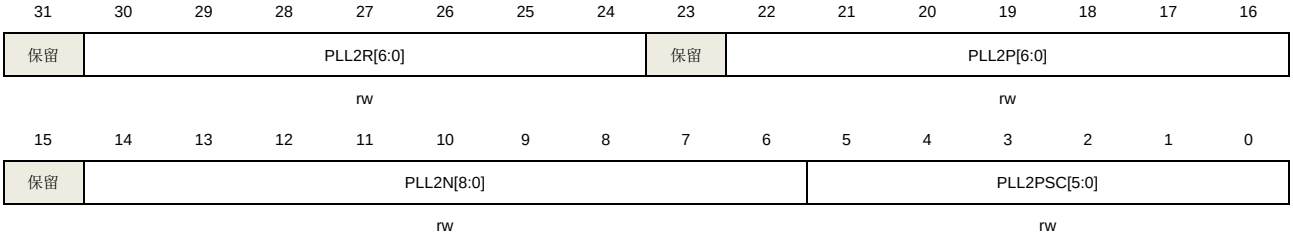
CK_PLL2VCOSRC = CK_PLL2SRC / PLL2PSC

CK_PLL2VCO = CK_ PLL2VCOSRC × （PLL2N + PLL2FRAN / 2^{13}）

CK_PLL2P = CK_ PLL2VCO / PLL2P

CK_PLL2R = CK_ PLL2VCO / PLL2R

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	保留	必须保持复位值。
30:24	PLL2R[6:0]	PLL2R 输出频率的分频系数（PLL2 VCO 时钟作为输入） 当 PLL2 被关闭时由软件置位或清零。这些位域用做将 PLL2 VCO 时钟（CK_PLL2VCO）分频生成 PLL2R 输出时钟（CK_PLL2R）。RCU_PLL2 寄存器的 PLL2N 位域对 CK_PLL2VCO 时钟进行了描述。 0000000：CK_PLL2R = CK_PLL2VCO 0000001：CK_PLL2R = CK_PLL2VCO / 2 0000010：CK_PLL2R = CK_PLL2VCO / 3 0000011：CK_PLL2R = CK_PLL2VCO / 4 0000100：CK_PLL2R = CK_PLL2VCO / 5 … 1111111：CK_PLL2R = CK_PLL2VCO / 128
23	保留	必须保持复位值。
22:16	PLL2P[6:0]	PLL2P 输出频率的分频系数（PLL2 VCO 时钟作为输入） 当 PLL2 被关闭时由软件置位或清零。这些位域用做将 PLL2 VCO 时钟（CK_PLL2VCO）分频生成 PLL2P 输出时钟（CK_PLL2P）。RCU_PLL2 寄存器的 PLL2N 位域对 CK_PLL2VCO 时钟进行了描述。 0000000：CK_PLL2P = CK_PLL2VCO 0000001：CK_PLL2P = CK_PLL2VCO / 2 0000010：CK_PLL2P = CK_PLL2VCO / 3

0000011：CK_PLL2P = CK_PLL2VCO / 4
0000100：CK_PLL2R = CK_PLL2VCO / 5
…
1111111：CK_PLL2R = CK_PLL2VCO / 128

15	保留	必须保持复位值。
14:6	PLL2N[8:0]	PLL2 VCO 时钟倍频因子 当 PLL2 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 PLL2 VCO 源时钟（CK_PLL2VCOSRC）倍频生成 PLL2 VCO 输出时钟（CK_PLL2VCO）。RCU_PLL2 寄存器的 PLL2PSC 位域对 CK_PLL2VCOSRC 时钟进行了描述。 **注意：**CK_PLL2VCO 时钟频率范围必须在 150MHz 到 836MHz 之间 PLL2N 的值必须满足： 当 PLL2 小数锁存禁能时，PLL2N 的值必须满足：9 ≤ PLL2N ≤ 512 当 PLL2 小数锁存使能时，PLL2N 的值必须满足：12 ≤ PLL2N ≤ 508 000000000：保留 … 000000111：保留 000001000：PLL2N = 9. … 001000000：PLL2N = 65 001000001：PLL2N = 66 … 111111111：PLL2N = 512
5:0	PLL2PSC[5:0]	PLL2 VCO 源时钟分频器 当 PLL2 被关闭时由软件置位或清零。这些位域用做将 PLL2 源时钟（CK_PLL2SRC）分频生成 PLL2 VCO 源时钟（CK_PLL2VCOSRC）。RCU_PLLALL 寄存器的 PLL2SEL 位对 CK_PLL2SRC 时钟进行了描述。 VCO 源时钟频率范围必须在 1MHz 到 16MHz 之间 000000：保留 000001：CK_PLL2SRC 000010：CK_PLL2SRC / 2 000011：CK_PLL2SRC / 3 … 111111：CK_PLL2SRC / 63

6.3.34. 时钟配置寄存器 1（RCU_CFG1）

地址偏移：0x8C
复位值：0x0000 3F00

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

HPDFSEL	保留	TIMERSEL	USART5SEL[1:0]	USART2SEL[1:0]	USART1SEL[1:0]	PLL2RDIV[1:0]
rw		rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PERSEL[1:0]		CAN2SEL[1:0]		CAN1SEL[1:0]		CAN0SEL[1:0]		保留		RSPDIFSEL[1:0]		保留		USART0SEL[1:0]	
rw		rw		rw		rw				rw				rw	

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：选择 CK_APB2 时钟作为 HPDF 源时钟 1：选择 CK_AHB 时钟作为 HPDF 源时钟
30:25	保留	必须保持复位值。
24	TIMERSEL	TIMER 时钟源选择 由软件置位或复位 该位定义了所有定时器的时钟源选择 0：如果 RCU_CFG0 寄存器的 APB1PSC/APB2PSC 位域的值为 0b0xx（CK_APBx = CK_AHB）或 0b100（CK_APBx = CK_AHB/2），定时器时钟等于 CK_AHB（CK_TIMERx = CK_AHB），否则定时器时钟等于 APB 时钟的两倍（在 APB1 域的定时器：CK_TIMERx = 2 x CK_APB1，在 APB2 域的定时器：CK_TIMERx = 2 x CK_APB2）。 1：如果 RCU_CFG0 寄存器的 APB1PSC/APB2PSC 位域的值为 0b0xx（CK_APBx = CK_AHB），0b100（CK_APBx = CK_AHB/2），或 0b101（CK_APBx = CK_AHB/4），定时器时钟等于 CK_AHB（CK_TIMERx = CK_AHB）。否则定时器时钟等于 APB 时钟的四倍（在 APB1 域的定时器：CK_TIMERx = 4 x CK_APB1；在 APB2 域的定时器：CK_TIMERx = 4 x CK_APB2）。
23:22	USART5SEL[1:0]	USART5时钟源选择 由软件置位或复位，控制USART5时钟源 00：选择 CK_APB2 时钟作为 USART5 源时钟 01：选择 CK_AHB 时钟作为 USART5 源时钟 10：选择 CK_LXTAL 时钟作为 USART5 源时钟 11：选择CK_IRC64MDIV时钟作为USART5源时钟
21:20	USART2SEL[1:0]	USART2时钟源选择 由软件置位或复位，控制USART2时钟源 00：选择 CK_APB1 时钟作为 USART2 源时钟 01：选择 CK_AHB 时钟作为 USART2 源时钟 10：选择 CK_LXTAL 时钟作为 USART2 源时钟 11：选择CK_IRC64MDIV时钟作为USART2源时钟
19:18	USART1SEL[1:0]	USART1时钟源选择 由软件置位或复位，控制USART1时钟源 00：选择 CK_APB1 时钟作为 USART1 源时钟 01：选择 CK_AHB 时钟作为 USART1 源时钟

10：选择 CK_LXTAL 时钟作为 USART1 源时钟
11：选择CK_IRC64MDIV时钟作为USART1源时钟

17:16	PLL2RDIV[1:0]	PLL2R 时钟的分频因子 当 PLL2 时钟被关闭时由软件置位或复位。该位用于生成 TLI 模块的时钟源。 00：CK_PLL2R / 2 01：CK_PLL2R / 4 10：CK_PLL2R / 8 11：CK_PLL2R / 16
15:14	PERSEL[1:0]	CK_PER时钟源选择 由软件置位或复位，控制CK_PER时钟源 00：选择 CK_IRC64MDIV 时钟作为 CK_PER 源时钟 01：选择 CK_LPIRC4M 时钟作为 CK_PER 源时钟 10：选择 CK_HXTAL 时钟作为 CK_PER 源时钟 11：保留
13:12	CAN2SEL[1:0]	CAN2时钟源选择 由软件置位或复位，控制CAN2时钟源 00：选择 CK_HXTAL 时钟作为 CAN2 源时钟 01：选择 CK_APB2 时钟作为 CAN2 源时钟 10：选择 CK_APB2 / 2 时钟作为 CAN2 源时钟 11：选择CK_IRC64MDIV时钟作为CAN2源时钟
11:10	CAN1SEL[1:0]	CAN1时钟源选择 由软件置位或复位，控制CAN1时钟源 00：选择 CK_HXTAL 时钟作为 CAN1 源时钟 01：选择 CK_APB2 时钟作为 CAN1 源时钟 10：选择 CK_APB2 / 2 时钟作为 CAN1 源时钟 11：选择CK_IRC64MDIV时钟作为CAN1源时钟
9:8	CAN0SEL[1:0]	CAN0时钟源选择 由软件置位或复位，控制CAN0时钟源 00：选择 CK_HXTAL 时钟作为 CAN0 源时钟 01：选择 CK_APB2 时钟作为 CAN0 源时钟 10：选择 CK_APB2 / 2 时钟作为 CAN0 源时钟 11：选择CK_IRC64MDIV时钟作为CAN0源时钟
7:6	保留	必须保持复位值。
5:4	RSPDIFSEL[1:0]	RSPDIF时钟源选择 由软件置位或复位，控制RSPDIF时钟源 00：选择 CK_PLL0Q 时钟作为 RSPDIF 源时钟 01：选择 CK_PLL1R 时钟作为 RSPDIF 源时钟 10：选择 CK_PLL2R 时钟作为 RSPDIF 源时钟 11：选择CK_IRC64MDIV时钟作为RSPDIF源时钟

3:2	保留	必须保持复位值。
1:0	USART0SEL[1:0]	USART0时钟源选择 由软件置位或复位，控制USART0时钟源 00：选择 CK_APB2 时钟作为 USART0 源时钟 01：选择 CK_AHB 时钟作为 USART0 源时钟 10：选择 CK_LXTAL 时钟作为 USART0 源时钟 11：选择CK_IRC64MDIV时钟作为USART0源时钟

6.3.35. 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	SAI2B1SEL[2:0]			保留	SAI2B0SEL[2:0]			保留	SAI1SEL[2:0]			保留	SAI0SEL[2:0]		
	rw				rw				rw				rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CKOUT1SEL[2:0]			CKOUT1DIV[3:0]				保留	CKOUT0SEL[2:0]			CKOUT0DIV[3:0]			
	rw			rw					rw			rw			

位/位域	名称	描述
31	保留	必须保持复位值。
30:28	SAI2B1SEL[2:0]	SAI2块1时钟源选择 由软件置位或复位，控制SAI2块1时钟源 000：选择 CK_PLL0Q 时钟作为 SAI2 块 1 源时钟 001：选择 CK_PLL1P 时钟作为 SAI2 块 1 源时钟 010：选择 CK_PLL2P 时钟作为 SAI2 块 1 源时钟 011：选择 I2S_CKIN 时钟作为 SAI2 块 1 源时钟 100：选择 CK_PER 时钟作为 SAI2 块 1 源时钟 101：选择 CK_RSPDIF_SYMB 时钟作为 SAI2 块 1 源时钟 11x：保留
27	保留	必须保持复位值
26:24	SAI2B0SEL[2:0]	SAI2块0时钟源选择 由软件置位或复位，控制SAI2块0时钟源 000：选择 CK_PLL0Q 时钟作为 SAI2 块 0 源时钟 001：选择 CK_PLL1P 时钟作为 SAI2 块 0 源时钟 010：选择 CK_PLL2P 时钟作为 SAI2 块 0 源时钟 011：选择 I2S_CKIN 时钟作为 SAI2 块 0 源时钟 100：选择 CK_PER 时钟作为 SAI2 块 0 源时钟 101：选择 CK_RSPDIF_SYMB 时钟作为 SAI2 块 0 源时钟 11x：保留

23	保留	必须保持复位值。
22:20	SAI1SEL[2:0]	SAI1时钟源选择 由软件置位或复位，控制SAI1时钟源 000：选择 CK_PLL0Q 时钟作为 SAI1 源时钟 001：选择 CK_PLL1P 时钟作为 SAI1 源时钟 010：选择 CK_PLL2P 时钟作为 SAI1 源时钟 011：选择 I2S_CKIN 时钟作为 SAI1 源时钟 100：选择 CK_PER 时钟作为 SAI1 源时钟 其它：保留
19	保留	必须保持复位值。
18:16	SAI0SEL[2:0]	SAI0 / HPDF audio时钟源选择 由软件置位或复位，控制SAI0时钟源 000：选择 CK_PLL0Q 时钟作为 SAI0 / HPDF audio 源时钟 001：选择 CK_PLL1P 时钟作为 SAI0 / HPDF audio 源时钟 010：选择 CK_PLL2P 时钟作为 SAI0 / HPDF audio 源时钟 011：选择 I2S_CKIN 时钟作为 SAI0 / HPDF audio 源时钟 100：选择 CK_PER 时钟作为 SAI0 / HPDF audio 源时钟 其它：保留
15	保留	必须保持复位值。
14:12	CKOUT1SEL[2:0]	CKOUT1时钟源选择 由软件置位或清零 000：选择系统时钟 001：选择CK_PLL1R时钟 010：选择高速晶体振荡器时钟（HXTAL） 011：选择CK_PLL0P时钟 100：选择CK_LPIRC4M 时钟 101：选择CK_IRC32K时钟 110：选择CK_PLL2R时钟 111：保留 **注意：**对该位域的配置可能会造成对CK_OUT1的干扰，强烈建议仅在复位后但在使能HXTAL和PLLs之前来配置这些位。
11:8	CKOUT1DIV[3:0]	CK_OUT1分频器，来降低CK_OUT1频率 CK_OUT1时钟源的选择参考 RCU_CFG2寄存器的14:12位 0000：保留 0001：CK_OUT1不分频 0010：CK_OUT1被2分频 0011：CK_OUT1被3分频 0100：CK_OUT1被4分频 … 1111：CK_OUT1被15分频 **注意：**对该位域的配置可能会造成对CK_OUT1的干扰，强烈建议仅在复位后但在

使能HXTAL和PLLs之前来配置这些位。

7	保留	必须保持复位值。
6:4	CKOUT0SEL[2:0]	CKOUT0时钟源选择 由软件置位或清零 000：选择CK_IRC64MDIV时钟 001：选择CK_LXTAL时钟 010：选择高速晶体振荡器时钟（HXTAL） 011：选择CK_PLL0P时钟 100：选择CK_IRC48M时钟 101：选择CK_PER时钟 110：选择USBHS0 60M时钟 111：选择USBHS1 60M时钟 **注意：**对该位域的配置可能会造成对CK_OUT0的干扰，强烈建议仅在复位后但在使能HXTAL和PLLs之前来配置这些位。
3:0	CKOUT0DIV[3:0]	CK_OUT0分频器，来降低CK_OUT0频率 CK_OUT0时钟源的选择参考RCU_CFG2寄存器的6:4位 0000：保留 0001：CK_OUT1不分频 0010：CK_OUT1被2分频 0011：CK_OUT1被3分频 0100：CK_OUT1被4分频 … 1111：CK_OUT1被15分频 **注意：**对该位域的配置可能会造成对CK_OUT0的干扰，强烈建议仅在复位后但在使能HXTAL和PLLs之前来配置这些位。

6.3.36. 时钟配置寄存器 3（RCU_CFG3）

地址偏移：0x94

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		ADC2SEL[1:0]		ADC01SEL[1:0]		保留	DSPWUSSEL	保留							
		rw		rw			rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			SDIO1SEL	保留						I2C3SEL[1:0]		I2C2SEL[1:0]		I2C1SEL[1:0]	
			rw							rw		rw		rw	

位/位域	名称	描述
31:30	保留	必须保持复位值。

29:28	ADC2SEL[1:0]	ADC2时钟源选择 由软件置位或复位，控制ADC2时钟源 00：选择 CK_PLL1P 时钟作为 ADC2 源时钟 01：选择 CK_PLL2R 时钟作为 ADC2 源时钟 10：选择 CK_PER 时钟作为 ADC2 源时钟 11：保留
27:26	ADC01SEL[1:0]	ADC0与ADC1时钟源选择 由软件置位或复位，控制ADC0与ADC1时钟源 00：选择 CK_PLL1P 时钟作为 ADC0 与 ADC1 源时钟 01：选择 CK_PLL2R 时钟作为 ADC0 与 ADC1 源时钟 10：选择 CK_PER 时钟作为 ADC0 与 ADC1 源时钟 11：保留
25	保留	必须保持复位值。
24	DSPWUSSEL	唤醒深度睡眠的系统时钟源选择 由软件置位或复位，控制从深度睡眠唤醒的系统时钟源 该位也用于控制 HXTAL 阻塞时的系统时钟 0：选择 CK_IRC64MDIV 时钟作为从深度睡眠唤醒的系统时钟源 1：选择 CK_LPIRC4M 时钟作为内核从深度睡眠唤醒的系统时钟源 **注意：**如果 DSPWUSSEL = ‘1’且外设时钟源选择 CK_IRC64MDIV，当系统通过此外设从深度睡眠模式唤醒后，如果此时关闭该外设的唤醒功能，将导致 IRC64M 时钟关闭，此时外设将没有时钟驱动。这种情况下，用户需要重新置位 RCU_CTL 寄存器中的 IRC64MEN 位，再次打开 IRC64M 时钟。当 CKMEN 位置位且系统时钟为 CK_HXTAL 或者将系统时钟切换到 HXTAL 时，该位不能被修改。
23:13	保留	必须保持复位值。
12	SDIO1SEL	SDIO1时钟源选择 由软件置位或复位，控制SDIO1时钟源 0：选择 CK_PLL0Q 时钟作为 SDIO1 源时钟 1：选择 CK_PLL1R 时钟作为 SDIO1 源时钟
11:6	保留	必须保持复位值。
5:4	I2C3SEL[1:0]	I2C3时钟源选择 由软件置位或复位，控制I2C3时钟源 00：选择 CK_APB1 时钟作为 I2C3 源时钟 01：选择 CK_PLL2R 时钟作为 I2C3 源时钟 10：选择 CK_IRC64MDIV 时钟作为 I2C3 源时钟 11：选择CK_LPIRC4M时钟作为I2C3源时钟
3:2	I2C2SEL[1:0]	I2C2时钟源选择 由软件置位或复位，控制I2C2时钟源 00：选择 CK_APB1 时钟作为 I2C2 源时钟 01：选择 CK_PLL2R 时钟作为 I2C2 源时钟 10：选择 CK_IRC64MDIV 时钟作为 I2C2 源时钟

11：选择CK_LPIRC4M时钟作为I2C2源时钟

1:0	I2C1SEL[1:0]	I2C1时钟源选择 由软件置位或复位，控制I2C1时钟源 00：选择 CK_APB1 时钟作为 I2C1 源时钟 01：选择 CK_PLL2R 时钟作为 I2C1 源时钟 10：选择 CK_IRC64MDIV 时钟作为 I2C1 源时钟 11：选择CK_LPIRC4M时钟作为I2C1源时钟

6.3.37. PLL 控制寄存器（RCU_PLLALL）

地址偏移：0x98
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														PLLSEL[1:0]	
														rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					PLL2VCOSEL	PLL2RNG[1:0]		保留	PLL1VCOSEL	PLL1RNG[1:0]		保留	PLL0VCOSEL	PLL0RNG[1:0]	
					rw	rw			rw	rw			rw	rw	

位/位域	名称	描述
31:18	保留	必须保持复位值
17:16	PLLSEL[1:0]	PLLs时钟源选择 由软件置位或复位，控制PLLs时钟源 00：选择 CK_IRC64MDIV 时钟作为 PLL、PLL1、PLL2 源时钟 01：选择 CK_LPIRC4M 时钟作为 PLL、PLL1、PLL2 源时钟 10：选择 CK_HXTAL 时钟作为 PLL、PLL1、PLL2 源时钟 11：无时钟作为PLL、PLL1、PLL2源时钟
15:11	保留	必须保持复位值
10	PLL2VCOSEL	PLL2 VCO 范围选择 当 PLL2 被关闭时由软件置位或清零 0：选择宽范围（192 - 836MHz），PLL2 输入时钟频率范围应为 2-16MHz 1：选择窄范围（150 - 420MHz），PLL2 输入时钟频率范围应为 1-2MHz
9:8	PLL2RNG[1:0]	PLL2 输入时钟范围 当 PLL2 被关闭时由软件置位或清零 00：PLL2 输入时钟频率范围 1 - 2MHz 01：PLL2 输入时钟频率范围 2 - 4MHz 10：PLL2 输入时钟频率范围 4 - 8MHz

		11：PLL2 输入时钟频率范围 8 - 16MHz
7	保留	必须保持复位值。
6	PLL1VCOSEL	PLL1 VCO 范围选择 当PLL被关闭时由软件置位或清零 0：选择宽范围（192 - 836MHz），PLL1 输入时钟频率范围应为 2-16MHz 1：选择窄范围（150 - 420MHz），PLL1 输入时钟频率范围应为 1-2MHz
5:4	PLL1RNG[1:0]	PLL1 输入时钟范围 当 PLL1 被关闭时由软件置位或清零 00：PLL1 输入时钟频率范围 1 - 2MHz 01：PLL1 输入时钟频率范围 2 - 4MHz 10：PLL1 输入时钟频率范围 4 - 8MHz 11：PLL1 输入时钟频率范围 8 - 16MHz
3	保留	必须保持复位值。
2	PLL0VCOSEL	PLL0 VCO 范围选择 当PLL0被关闭时由软件置位或清零 0：选择宽范围（192 - 836MHz），PLL0 输入时钟频率范围应为 2-16MHz 1：选择窄范围（150 - 420MHz），PLL0 输入时钟频率范围应为 1-2MHz
1:0	PLL0RNG[1:0]	PLL0 输入时钟范围 当 PLL0 被关闭时由软件置位或清零 00：PLL0 输入时钟频率范围 1 - 2MHz 01：PLL0 输入时钟频率范围 2 - 4MHz 10：PLL0 输入时钟频率范围 4 - 8MHz 11：PLL0 输入时钟频率范围 8 - 16MHz

6.3.38. PLL0 小数配置寄存器（RCU_PLL0FRA）

地址偏移：0x9C

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15	PLL0FRAEN	PLL0小数锁存使能 由软件置位或复位，用于将PLL0FRAN的内容锁存到 Sigma-Delta调制器。当

		PLL0FRAEN从0切换到1，PLL0FRAEN的值将被转移到调制器中。
14:13	保留	必须保持复位值
12:0	PLL0FRAN[12:0]	PLL0 VCO倍频因子的小数部分 由软件置位或复位，用于控制PLL0 VCO倍频因子的小数部分。该位域可以动态修改从而对PLL0 VCO进行微调。 必须配置该值使 PLL0 VCO 输出频率为如下范围： 当 PLL0VCOSEL 为 0 时，范围为 192 - 836MHz 当 PLL0VCOSEL 为 1 时，范围为 150 - 420MHz

6.3.39. PLL1 小数配置寄存器（RCU_PLL1FRA）

地址偏移：0xA0
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15	PLL1FRAEN	PLL1小数锁存使能 由软件置位或复位，用于将PLL1FRAN的内容锁存到 Sigma-Delta调制器。当PLL1FRAEN从0切换到1，PLL1FRAEN的值将被转移到调制器中。
14:13	保留	必须保持复位值
12:0	PLL1FRAN[12:0]	PLL1 VCO倍频因子的小数部分 由软件置位或复位，用于控制PLL1 VCO倍频因子的小数部分。该位域可以动态修改从而对PLL1 VCO进行微调。 必须配置该值使 PLL1 VCO 输出频率为如下范围： 当 PLL1VCOSEL 为 0 时，范围为 192 - 836MHz 当 PLL1VCOSEL 为 1 时，范围为 150 - 420MHz

6.3.40. PLL2 小数配置寄存器（RCU_PLL2FRA）

地址偏移：0xA4
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



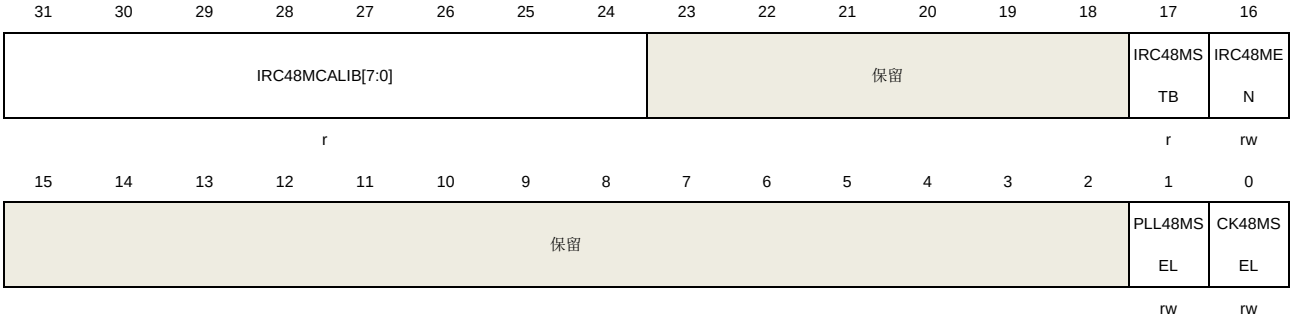
位/位域	名称	描述
31:16	保留	必须保持复位值。
15	PLL2FRAEN	PLL2小数锁存使能 由软件置位或复位，用于将PLL2FRAN的内容锁存到 Sigma-Delta调制器。当PLL2FRAEN从0切换到1，PLL2FRAEN的值将被转移到调制器中。
14:13	保留	必须保持复位值
12:0	PLL2FRAN[12:0]	PLL2 VCO倍频因子的小数部分 由软件置位或复位，用于控制PLL2 VCO倍频因子的小数部分。该位域可以动态修改从而对PLL2 VCO进行微调。 必须配置该值使 PLL2 VCO 输出频率为如下范围： 当 PLL2VCOSEL 为 0 时，范围为 192 - 836MHz 当 PLL2VCOSEL 为 1 时，范围为 150 - 420MHz

6.3.41. 附加时钟控制寄存器 0（RCU_ADDCTL0）

地址偏移：0xC0

复位值：0x8000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域	名称	描述
31:24	IRC48MCALIB[7:0]	内部 48MHz RC 振荡器校准值寄存器 上电时自动加载这些位
23:18	保留	必须保持复位值。
17	IRC48MSTB	内部 48MHz RC 振荡器时钟稳定标志位 硬件置‘1’来指示IRC48M振荡器时钟是否稳定待用

0：IRC48M未稳定
1：IRC48M 已稳定

16	IRC48MEN	内部 48MHz RC 振荡器使能 由软件置位和复位。当进入深度睡眠或待机模式后由硬件复位 0：关闭IRC48M时钟 1：打开 IRC48M 时钟
15:2	保留	必须保持复位值。
1	PLL48MSEL	PLL48M时钟源选择 由软件置位和复位。该位用于选择CK_PLL0Q时钟或CK_PLL2P时钟作为PLL48M的时钟源 0：选择CK_PLL0Q时钟 1：选择 CK_PLL2P 时钟
0	CK48MSEL	48MHz时钟源选择 由软件置位和复位。该位用于选择IRC48M时钟或PLL48M时钟作为CK48M时钟源。CK48M 时钟为TRNG/SDIO/USBHS模块提供时钟。RCU_ADDCTL0寄存器的PLL48MSEL位对PLL48M时钟进行了描述。 0：不选择IRC48M时钟（通过PLL48MSEL位选择使用CK_PLL0Q时钟或CK_PLL2P时钟） 1：选择 IRC48M 时钟

6.3.42. 附加时钟控制寄存器 1（RCU_ADDCTL1）

地址偏移：0xC4
复位值：0x0000 7080

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
PLLUSBHS1STB	PLLUSBHS1EN	PLLUSBHS0STB	PLLUSBHS0EN	保留							LPIRC4MDSPEN	保留		IRC64MDIV[1:0]	
r	rw	r	rw								rw			rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPIRC4MCALIB[7:0]								LPIRC4MADJ[5:0]						LPIRC4MSTB	LPIRC4MEN
r								rw						r	rw

位/位域	名称	描述
31	PLLUSBHS1STB	PLLUSBHS1 时钟稳定标志位 硬件置1来表示PLLUSBHS1输出时钟是否稳定待用 0：PLLUSBHS1未稳定 1：PLLUSBHS1 已稳定
30	PLLUSBHS1EN	PLLUSBHS1时钟使能 软件置位或复位，当进入深度睡眠或待机模式时由硬件复位

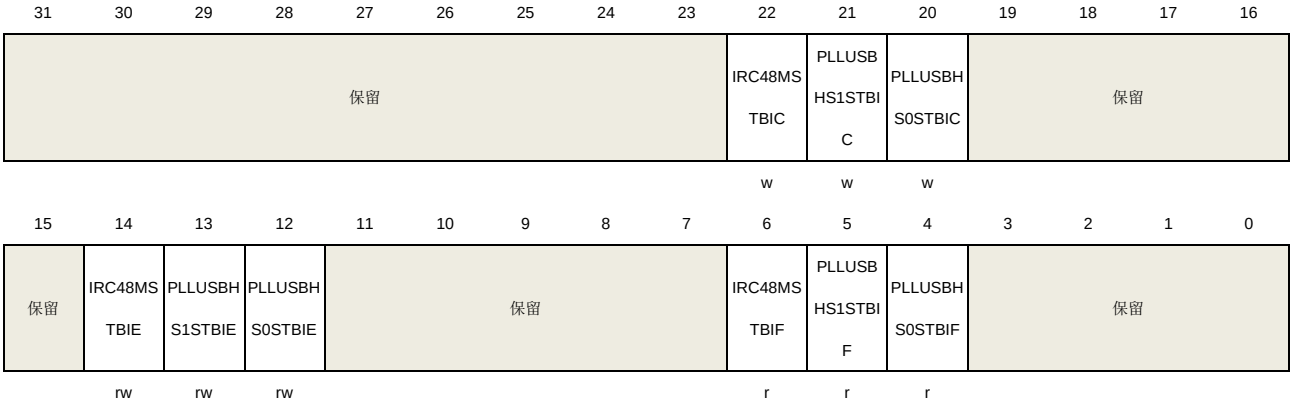
		0：PLLUSBHS1被关闭 1：PLLUSBHS1 被打开
29	PLLUSBHS0STB	PLLUSBHS1 时钟稳定标志位 硬件置1来表示PLLUSBHS1输出时钟是否稳定待用 0：PLLUSBHS0未稳定 1：PLLUSBHS0 已稳定
28	PLLUSBHS0EN	PLLUSBHS0时钟使能 软件置位或复位，当进入深度睡眠或待机模式时由硬件复位 0：PLLUSBHS0被关闭 1：PLLUSBHS0 被打开
27:21	保留	必须保持复位值。
20	LPIRC4MDSPEN	在深度睡眠模式下LPIRC4M时钟使能 由软件置位和复位。在深度睡眠模式下，可强制LPIRC4M作为部分外设的时钟。该位与LPIRC4M是否使能无关。 0：无作用 1：强制 LPIRC4M 在深度睡眠模式下运行
19:18	保留	必须保持复位值。
17:16	IRC64MDIV[1:0]	IRC64M时钟分频 由软件置位和复位，当系统时钟选择CK_IRC64MDIV或IRC64MEN位置1时无法写入。 00：CK_IRC64MDIV = CK_IRC64M / 1 01：CK_IRC64MDIV = CK_IRC64M / 2 10：CK_IRC64MDIV = CK_IRC64M / 4 11：CK_IRC64MDIV = CK_IRC64M / 8
15:8	LPIRC4MCALIB[7:0]	LPIRC4M时钟校准值 上电时自动加载这些位，校准信号步长位0.4%
7:2	LPIRC4MADJ[5:0]	LPIRC4M 时钟调整值 这些位由软件置位，最终调整值为LPIRC4MADJ[7:0]位域的当前值加上LPIRC4MCALIB[8:0]位域的值。最终调整值应该调整LPIRC4M到4 MHz ± 1%
1	LPIRC4MSTB	LPIRC4M时钟稳定标志位 硬件置‘1’来指示LPIRC4M时钟是否稳定待用 0：LPIRC4M振荡器未稳定 1：LPIRC4M 振荡器已稳定
0	LPIRC4MEN	LPIRC4M RC振荡器使能 软件置位或复位，如果LPIRC4M时钟作为系统时钟时，该位不能被复位。如果DSPWUSSEL为1，当从深度睡眠或待机模式返回，该位由硬件置1来启动LPIRC4M振荡器。 0：LPIRC4M RC振荡器被关闭 1：LPIRC4M RC 振荡器被打开

6.3.43. 附加时钟中断寄存器（RCU_ADDINT）

地址偏移：0xCC

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域	名称	描述
31:23	保留	必须保持复位值。
22	IRC48MSTBIC	内部 48 MHz RC 振荡器稳定中断清零 软件写 1 复位 IRC48MSTBIF 标志位 0：不复位 IRC48MSTBIF 标志位 1：复位 IRC48MSTBIF 标志位
21	PLLUSBHS1STBIC	USBHS1 内部 PLL 时钟稳定中断清零 软件写 1 复位 PLLUSBHS1STBIF 标志位 0：不复位 PLLUSBHS1STBIF 标志位 1：复位 PLLUSBHS1STBIF 标志位
20	PLLUSBHS0STBIC	USBHS0 内部 PLL 时钟稳定中断清零 软件写 1 复位 PLLUSBHS0STBIF 标志位 0：不复位 PLLUSBHS0STBIF 标志位 1：复位 PLLUSBHS0STBIF 标志位
19:15	保留	必须保持复位值。
14	IRC48MSTBIE	内部 48 MHz RC 振荡器稳定中断使能 由软件置位和复位来使能/禁止 IRC48M 时钟稳定中断 0：禁止IRC48M时钟稳定中断 1：使能 IRC48M 时钟稳定中断
13	PLLUSBHS1STBIE	USBHS1 内部 PLL 时钟稳定中断使能 由软件置位和复位来使能/禁止 USBHS1 内部 PLL 时钟稳定中断 0：禁止USBHS1内部PLL时钟稳定中断 1：使能 USBHS1 内部 PLL 时钟稳定中断
12	PLLUSBHS0STBIE	USBHS0 内部 PLL 时钟稳定中断使能

由软件置位和复位来使能/禁止 USBHS0 内部 PLL 时钟稳定中断
0：禁止USBHS0内部PLL时钟稳定中断
1：使能 USBHS0 内部 PLL 时钟稳定中断

11:7	保留	必须保持复位值。
6	IRC48MSTBIF	IRC48M 时钟稳定中断标志位 当内部48 MHz RC振荡器时钟稳定且IRC48MSTBIE位被置1时由硬件置1 软件置位 IRC48MSTBIC 位时清除该位 0：无IRC48M时钟稳定中断产生 1：产生 IRC48M 时钟稳定中断
5	PLLUSBHS1STBIF	USBHS1 内部 PLL 时钟稳定中断标志位 当USBHS1内部PLL时钟稳定且PLLUSBHS1STBIE位被置1时由硬件置1 软件置位 PLLUSBHS1STBIC 位时清除该位 0：无USBHS1内部PLL时钟稳定中断产生 1：产生 USBHS1 内部 PLL 时钟稳定中断
4	PLLUSBHS0STBIF	USBHS0 内部 PLL 时钟稳定中断标志位 当USBHS0内部PLL时钟稳定且PLLUSBHS0STBIE位被置1时由硬件置1 软件置位 PLLUSBHS0STBIC 位时清除该位 0：无USBHS0内部PLL时钟稳定中断产生 1：产生 USBHS0 内部 PLL 时钟稳定中断
3:0	保留	必须保持复位值。

6.3.44. 时钟配置寄存器 4（RCU_CFG4）

地址偏移：0xD0
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域	名称	描述
31:10	保留	必须保持复位值。
9:8	EXMCSEL[1:0]	EXMC时钟源选择 由软件置位或复位，控制EXMC时钟源 00：CK_EXMC = CK_AHB 01：CK_EXMC = CK_PLL0Q

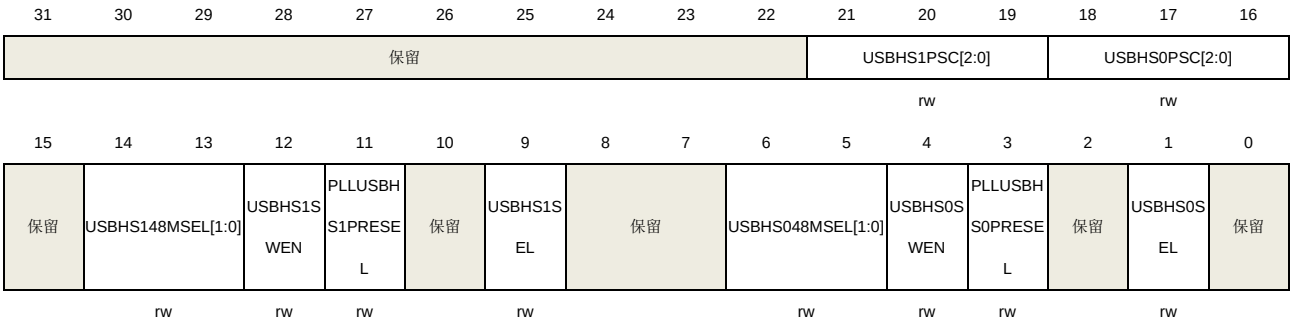
10：CK_EXMC = CK_PLL1R
11：CK_EXMC = CK_PER

7:1	保留	必须保持复位值。
0	SDIO0SEL	SDIO0时钟源选择 由软件置位或复位，控制SDIO0时钟源 0：选择 CK_PLL0Q 时钟作为 SDIO0 源时钟 1：选择CK_PLL1R时钟作为SDIO0源时钟

6.3.45. USB 时钟控制寄存器（RCU_USBCLKCTL）

地址偏移：0xD4
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域	名称	描述
31:22	保留	必须保持复位值。
21:19	USBHS1PSC[2:0]	USBHS1的时钟分频系数 由软件置位或清零。USBHS1的时钟必须为48MHz，当USBHS1时钟使能的时候，这些位无法修改。 000：CK_USBHS1 = CK_PLL1Q / 1 001：CK_USBHS1 = CK_PLL1Q / 2 010：CK_USBHS1 = CK_PLL1Q / 3 011：CK_USBHS1 = CK_PLL1Q / 4 100：CK_USBHS1 = CK_PLL1Q / 5 101：CK_USBHS1 = CK_PLL1Q / 6 110：CK_USBHS1 = CK_PLL1Q / 7 111：CK_USBHS1 = CK_PLL1Q / 8
18:16	USBHS0PSC[2:0]	USBHS0的时钟分频系数 由软件置位或清零。USBHS0的时钟必须为48MHz，当USBHS0时钟使能的时候，这些位无法修改。 000：CK_USBHS0 = CK_PLL1Q / 1 001：CK_USBHS0 = CK_PLL1Q / 2 010：CK_USBHS0 = CK_PLL1Q / 3

		011：CK_USBHS0 = CK_PLL1Q / 4 100：CK_USBHS0 = CK_PLL1Q / 5 101：CK_USBHS0 = CK_PLL1Q / 6 110：CK_USBHS0 = CK_PLL1Q / 7 111：CK_USBHS0 = CK_PLL1Q / 8
15	保留	必须保持复位值。
14:13	USBHS148MSEL[1:0]	USBHS1 48M时钟源选择 由软件置位或复位，控制USBHS1 48M时钟源 00：选择 CK_PLL0R 时钟作为 USBHS1 48M 源时钟 01：选择 CK_PLLUSBHS1/USBHS1DV 作为 USBHS1 48M 源时钟 10：选择 CK_PLL1Q/USBHS1PSC 作为 USBHS1 48M 源时钟 11：选择CK_IRC48M时钟作为USBHS1 48M源时钟
12	USBHS1SWEN	USBHS1 时钟源选择使能 0：通过 USBHS1 模块硬件选择 USBHS1 时钟 1：使用 USBHS1SEL 选择 USBHS1 时钟
11	PLLUSBHS1PRESEL	PLLUSBHS1 时钟源预选择 由软件置位和复位选择 PLLUSBHS1 时钟源。 0：选择 CK_HXTAL 时钟作为 PLLUSBHS1 时钟源 1：选择 CK_IRC48M 时钟作为 PLLUSBHS1 时钟源
10	保留	必须保持复位值。
9	USBHS1SEL	USBHS1 时钟源选择 由软件置位和复位。 0：选择 48M 时钟作为 USBHS1 时钟源 1：选择 60M 时钟作为 USBHS1 时钟源
8:7	保留	必须保持复位值。
6:5	USBHS048MSEL[1:0]	USBHS0 48M时钟源选择 由软件置位或复位，控制USBHS0 48M时钟源 00：选择 CK_PLL0R 时钟作为 USBHS0 48M 源时钟 01：选择 CK_PLLUSBHS0/USBHS0DV 作为 USBHS0 48M 源时钟 10：选择 CK_PLL1Q/USBHS0PSC 作为 USBHS0 48M 源时钟 11：选择CK_IRC48M时钟作为USBHS0 48M源时钟
4	USBHS0SWEN	USBHS0 时钟源选择使能 0：通过 USBHS0 模块硬件选择 USBHS0 时钟 1：使用 USBHS0SEL 选择 USBHS0 时钟
3	PLLUSBHS0PRESEL	PLLUSBHS0 时钟源预选择 由软件置位和复位选择 PLLUSBHS0 时钟源。 0：选择 CK_HXTAL 时钟作为 PLLUSBHS0 时钟源 1：选择 CK_IRC48M 时钟作为 PLLUSBHS0 时钟源

2	保留	必须保持复位值。
1	USBHS0SEL	USBHS0 时钟源选择 由软件置位和复位。 0：选择 48M 时钟作为 USBHS0 时钟源 1：选择 60M 时钟作为 USBHS0 时钟源
0	保留	必须保持复位值。

6.3.46. PLLUSB 时钟配置寄存器（RCU_PLLUSBCFG）

地址偏移：0xD8
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	PLLUSBHS1MF[6:0]							保留	USBHS1DV[2:0]			PLLUSBHS1PREDV[3:0]			
	rw								rw			rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PLLUSBHS0MF[6:0]							保留	USBHS0DV[2:0]			PLLUSBHS0PREDV[3:0]			
	rw								rw			rw			

位/位域	名称	描述
31	保留	必须保持复位值。
30:24	PLLUSBHS1MF[6:0]	PLLUSBHS1 时钟倍频因子 0000000：保留 0000001：保留 … 0001111：保留 0010000：PLLUSBHS1MF输入源时钟16倍频 0010001：PLLUSBHS1MF输入源时钟17倍频 0010010：PLLUSBHS1MF输入源时钟18倍频 0010011：PLLUSBHS1MF输入源时钟19倍频 … 1111111：PLLUSBHS1MF输入源时钟127倍频 **注意：**PLLUSBHS1 输出时钟频率不能超过 480MHz
23	保留	必须保持复位值。
22:20	USBHS1DV[2:0]	USBHS1 时钟分频因子 由软件置位或清零。 000：USBHS1DV 输入源时钟 2 分频 001：USBHS1DV 输入源时钟 4 分频 010：USBHS1DV 输入源时钟 6 分频 …

		111：USBHS1DV 输入源时钟 16 分频
19:16	PLLUSBHS1PREDV[3:0]	PLLUSBHS1PREDV 的时钟分频系数 由软件置位和复位。 0000：保留 0001：PLLUSBHS1PREDV 输入源时钟未分频 0010：PLLUSBHS1PREDV 输入源时钟 2 分频 … 1111：PLLUSBHS1PREDV 输入源时钟 15 分频
15	保留	必须保持复位值。
14:8	PLLUSBHS0MF[6:0]	PLLUSBHS0 时钟倍频因子 0000000：保留 0000001：保留 … 0001111：保留 0010000：PLLUSBHS0MF输入源时钟16倍频 0010001：PLLUSBHS0MF输入源时钟17倍频 0010010：PLLUSBHS0MF输入源时钟18倍频 0010011：PLLUSBHS0MF输入源时钟19倍频 … 1111111：PLLUSBHS0MF输入源时钟127倍频 **注意：**PLLUSBHS0输出时钟频率不能超过480MHz
7	保留	必须保持复位值。
6:4	USBHS0DV[2:0]	USBHS0 时钟分频因子 由软件置位或清零。 000：USBHS0DV 输入源时钟 2 分频 001：USBHS0DV 输入源时钟 4 分频 010：USBHS0DV 输入源时钟 6 分频 … 111：USBHS0DV 输入源时钟 16 分频
3:0	PLLUSBHS0PREDV[3:0]	PLLUSBHS0PREDV 的时钟分频系数 由软件置位和复位。 0000：保留 0001：PLLUSBHS0PREDV 输入源时钟未分频 0010：PLLUSBHS0PREDV 输入源时钟 2 分频 … 1111：PLLUSBHS0PREDV 输入源时钟 15 分频

6.3.47. APB2 附加复位寄存器（RCU_ADDAPB2RST）

地址偏移：0xE0

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													CAN2RST	CAN1RST	CAN0RST
													rw	rw	rw

位/位域	名称	描述
31:3	保留	必须保持复位值。
2	CAN2RST	CAN2 复位 由软件置位或复位 0：无作用 1：复位 CAN2
1	CAN1RST	CAN1 复位 由软件置位或复位 0：无作用 1：复位 CAN1
0	CAN0RST	CAN0 复位 由软件置位或复位 0：无作用 1：复位 CAN0

6.3.48. APB2 附加使能寄存器（RCU_ADDAPB2EN）

地址偏移：0xE4

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													CAN2EN	CAN1EN	CAN0EN
													rw	rw	rw

位/位域	名称	描述
31:3	保留	必须保持复位值。
2	CAN2EN	CAN2 时钟使能 由软件置位或复位 0：关闭 CNA2 时钟

1：开启 CAN2 时钟

1	CAN1EN	CAN1 时钟使能 由软件置位或复位 0：关闭 CNA1 时钟 1：开启 CAN1 时钟
0	CAN0EN	CAN0 时钟使能 由软件置位或复位 0：关闭 CNA0 时钟 1：开启 CAN0 时钟

6.3.49. APB2 附加睡眠模式使能寄存器（RCU_ADDAPB2SPEN）

地址偏移：0xE8

复位值：0x0000 0007

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													CAN2SPEN	CAN1SPEN	CAN0SPEN
													rw	rw	rw

位/位域	名称	描述
31:3	保留	必须保持复位值。
2	CAN2SPEN	睡眠模式下 CAN2 时钟使能 由软件置位或复位 0：睡眠模式下关闭 CAN2 时钟 1：睡眠模式下开启 CAN2 时钟
1	CAN1SPEN	睡眠模式下 CAN1 时钟使能 由软件置位或复位 0：睡眠模式下关闭 CAN1 时钟 1：睡眠模式下开启 CAN1 时钟
0	CAN0SPEN	睡眠模式下 CAN0 时钟使能 由软件置位或复位 0：睡眠模式下关闭 CAN0 时钟 1：睡眠模式下开启 CAN0 时钟

6.3.50. 时钟配置寄存器 5（RCU_CFG5）

地址偏移：0xF0

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									SPI5SEL[2:0]			保留	SPI4SEL[2:0]		
										rw				rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	SPI3SEL[2:0]			保留	SPI2SEL[2:0]			保留	SPI1SEL[2:0]			保留	SPI0SEL[2:0]		
	rw				rw				rw				rw		

位/位域	名称	描述
31:23	保留	必须保持复位值。
22:20	SPI5SEL[2:0]	SPI5 / I2S5时钟源选择 由软件置位或复位，控制SPI5 / I2S5时钟源 000：选择CK_APB2时钟作为SPI5 / I2S5源时钟 001：选择CK_PLL1Q时钟作为SPI5 / I2S5源时钟 010：选择CK_PLL2Q时钟作为SPI5 / I2S5源时钟 011：选择CK_IRC64MDIV时钟作为SPI5 / I2S5源时钟 100：选择CK_LPIRC4M时钟作为SPI5 / I2S5源时钟 101：选择CK_HXTAL时钟作为SPI5 / I2S5源时钟 110：选择I2S_CKIN时钟作为SPI5 / I2S5源时钟 111：保留
19	保留	必须保持复位值。
18:16	SPI4SEL[2:0]	SPI4时钟源选择 由软件置位或复位，控制SPI4时钟源 000：选择CK_APB2时钟作为SPI4源时钟 001：选择CK_PLL1Q时钟作为SPI4源时钟 010：选择CK_PLL2Q时钟作为SPI4源时钟 011：选择CK_IRC64MDIV时钟作为SPI4源时钟 100：选择CK_LPIRC4M时钟作为SPI4源时钟 101：选择CK_HXTAL时钟作为SPI4源时钟 其它：保留
15	保留	必须保持复位值。
14:12	SPI3SEL[2:0]	SPI3时钟源选择 由软件置位或复位，控制SPI3时钟源 000：选择CK_APB2时钟作为SPI3源时钟 001：选择CK_PLL1Q时钟作为SPI3源时钟 010：选择CK_PLL2Q时钟作为SPI3源时钟 011：选择CK_IRC64MDIV时钟作为SPI3源时钟 100：选择CK_LPIRC4M时钟作为SPI3源时钟 101：选择CK_HXTAL时钟作为SPI3源时钟 其它：保留

11	保留	必须保持复位值。
10:8	SPI2SEL[2:0]	SPI2 / I2S2时钟源选择 由软件置位或复位，控制SPI2 / I2S2时钟源 000：选择CK_PLL0Q时钟作为SPI2 / I2S2源时钟 001：选择CK_PLL1P时钟作为SPI2 / I2S2源时钟 010：选择CK_PLL2P时钟作为SPI2 / I2S2源时钟 011：选择I2S_CKIN时钟作为SPI2 / I2S2源时钟 100：选择CK_PER时钟作为SPI2 / I2S2源时钟 其它：保留
7	保留	必须保持复位值。
6:4	SPI1SEL[2:0]	SPI1 / I2S1时钟源选择 由软件置位或复位，控制SPI1 / I2S1时钟源 000：选择CK_PLL0Q时钟作为SPI1 / I2S1源时钟 001：选择CK_PLL1P时钟作为SPI1 / I2S1源时钟 010：选择CK_PLL2P时钟作为SPI1 / I2S1源时钟 011：选择I2S_CKIN时钟作为SPI1 / I2S1源时钟 100：选择CK_PER时钟作为SPI1 / I2S1源时钟 其它：保留
3	保留	必须保持复位值。
2:0	SPI0SEL[2:0]	SPI0 / I2S0时钟源选择 由软件置位或复位，控制SPI0 / I2S0时钟源 000：选择CK_PLL0Q时钟作为SPI0 / I2S0源时钟 001：选择CK_PLL1P时钟作为SPI0 / I2S0源时钟 010：选择CK_PLL2P时钟作为SPI0 / I2S0源时钟 011：选择I2S_CKIN时钟作为SPI0 / I2S0源时钟 100：选择CK_PER时钟作为SPI0 / I2S0源时钟 其它：保留

6.1. RCU 寄存器（适用于 GD32H75E）

RCU 基地址：0x5802 4400

6.1.1. 控制寄存器（RCU_CTL）

地址偏移：0x00

复位值：0xC000 8040

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
IRC64MSTB	IRC64MEN	PLL2STB	PLL2EN	PLL1STB	PLL1EN	PLL0STB	PLL0EN	保留				CKMEN	HXTALBPS	HXTALSTB	HXTALEN
r	rw	r	rw	r	rw	r	rw					rw	rw	r	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRC64MCALIB[8:0]								IRC64MADJ[6:0]							
r								rw							

位/位域	名称	描述
31	IRC64MSTB	内部64MHz RC振荡器稳定标志位 硬件置‘1’来指示IRC64M振荡器时钟是否稳定待用 0：IRC64M振荡器未稳定 1：IRC64M 振荡器已稳定
30	IRC64MEN	内部64MHz RC振荡器使能 软件置位或复位，如果IRC64M时钟作为系统时钟时，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IRC64M振荡器。 0：内部64 MHz RC振荡器被关闭 1：内部 64 MHz RC 振荡器被打开
29	PLL2STB	PLL2 时钟稳定标志位 硬件置1来表示PLL2输出时钟是否稳定待用 0：PLL2未稳定 1：PLL2 已稳定
28	PLL2EN	PLL2 使能 软件置位或复位，当进入深度睡眠或待机模式时由硬件复位 0：PLL2被关闭 1：PLL2 被打开
27	PLL1STB	PLL1 时钟稳定标志位 硬件置1来表示PLL1输出时钟是否稳定待用 0：PLL1未稳定 1：PLL1 已稳定
26	PLL1EN	PLL1 使能

		软件置位或复位，当进入深度睡眠或待机模式时由硬件复位 0：PLL1被关闭 1：PLL1 被打开
25	PLL0STB	PLL0 时钟稳定标志位 硬件置1来表示PLL0输出时钟是否稳定待用 0：PLL0未稳定 1：PLL0 已稳定
24	PLL0EN	PLL0 使能 软件置位或复位，当PLL0时钟作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：PLL0被关闭 1：PLL0 被打开
23:20	保留	必须保持复位值。
19	CKMEN	HXTAL时钟监视器使能 0：禁止高速4 ~ 50 MHz晶体振荡器（HXTAL）时钟监视器 1：使能高速4 ~ 50 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IRC64M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IRC64MEN的状态，自动使能IRC64M时钟。
18	HXTALBPS	高速晶体振荡器（HXTAL）时钟旁路模式使能 只有在HXTAL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PLL0P时钟作为系统时钟时，其作为PLL0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 ~ 50 MHz晶体振荡器被关闭 1：高速 4 ~ 50 MHz 晶体振荡器被打开
15:7	IRC64MCALIB[8:0]	内部64MHz RC振荡器校准值 上电时自动加载这些位
6:0	IRC64MADJ[6:0]	内部 64MHz RC 振荡器时钟调整值 这些位由软件置位，最终调整值为 IRC64MADJ[6:0]位域的当前值加上

IRC64MCALIB[8:0]位域的值。最终调整值应该调整 IRC64M 到 64 MHz ± 1%

6.1.2. PLL0 寄存器（RCU_PLL0）

地址偏移：0x04

复位值：0x0100 2020

配置PLL0时钟可参考下列公式：

CK_PLL0VCOSRC = CK_PLL0SRC / PLL0PSC

CK_PLL0VCO = CK_PLL0VCOSRC × （PLL0N + PLL0FRAN / 2^{13}）

CK_PLL0R = CK_PLL0VCO / PLL0R

CK_PLL0P = CK_PLL0VCO / PLL0P

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
PLLSTBSRC	PLL0R[6:0]							保留	PLL0P[6:0]						
w	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PLL0N[8:0]									PLL0PSC[5:0]					
	rw									rw					

位/位域	名称	描述
31	PLLSTBSRC	PLLs 稳定信号源 0：模拟信号 1：数字信号
30:24	PLL0R[6:0]	PLL0R 输出频率的分频系数（PLL0 VCO 时钟作为输入） 当 PLL0 被关闭时由软件置位或清零。这些位域用做将 PLL0 VCO 时钟（CK_PLL0VCO）分频生成 PLL0R 输出时钟（CK_PLL0R）。RCU_PLL0 寄存器的 PLL0N 位域对 CK_PLL0VCO 时钟进行了描述。 0000000：CK_PLL0R = CK_PLL0VCO 0000001：CK_PLL0R = CK_PLL0VCO / 2 0000010：CK_PLL0R = CK_PLL0VCO / 3. 0000011：CK_PLL0R = CK_PLL0VCO / 4 0000100：CK_PLL0R = CK_PLL0VCO / 5 … 1111111：CK_PLL0R = CK_PLL0VCO / 128
23	保留	必须保持复位值。
22:16	PLL0P[6:0]	PLL0P 输出频率分频系数（PLL0 VCO 时钟作为输入） 当 PLL0 被关闭时由软件置位或清零。这些位域用做将 PLL0 VCO 时钟（CK_PLL0VCO）分频生成 PLL0P 输出时钟（CK_PLL0P）。CK_PLL0P 时钟可以被用作系统时钟（不超过 600MHz）。 RCU_PLL0 寄存器的 PLL0N 位域对 CK_PLL0VCO 时钟进行了描述。

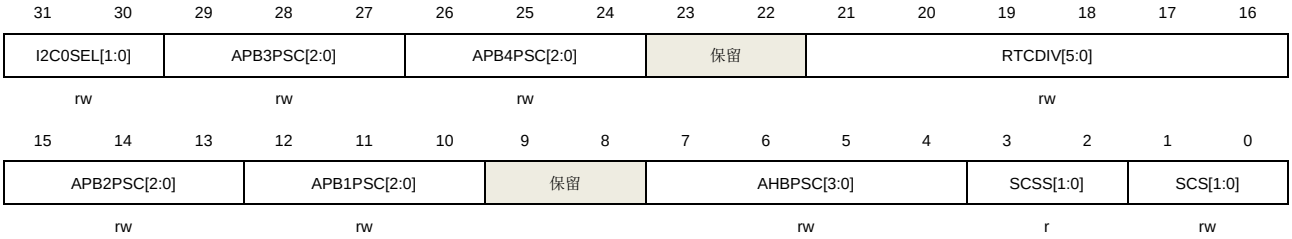
0000000：CK_PLL0P = CK_PLL0VCO
0000001：CK_PLL0P = CK_PLL0VCO / 2
0000010：CK_PLL0P = CK_PLL0VCO / 3
0000011：CK_PLL0P = CK_PLL0VCO / 4
0000100：CK_PLL0P = CK_PLL0VCO / 5
…
1111111：CK_PLL0P = CK_PLL0VCO / 128

15	保留	必须保持复位值。
14:6	PLL0N[8:0]	PLL0 VCO 时钟倍频因子 当 PLL0 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 PLL0 VCO 源时钟（CK_PLL0VCOSRC）倍频生成 PLL0 VCO 输出时钟（CK_PLL0VCO）。RCU_PLL0 寄存器的 PLL0PSC 位域对 CK_PLL0VCOSRC 时钟进行了描述。 **注意：**CK_PLL0VCO 时钟频率范围必须在 150MHz 到 836MHz 之间 PLL0N 的值必须满足： 当 PLL0 小数锁存禁能时，PLL0N 的值必须满足：9 ≤ PLL0N ≤ 512 当 PLL0 小数锁存使能时，PLL0N 的值必须满足：12 ≤ PLL0N ≤ 508 000000000：保留 … 000000111：保留 000001000：PLL0N = 9 … 001000000：PLL0N = 65 001000001：PLL0N = 66 … 111111111：PLL0N = 512
5:0	PLL0PSC[5:0]	PLL0 VCO 源时钟分频器 当 PLL0 被关闭时由软件置位或清零。这些位域用做将 PLL0 源时钟（CK_PLL0SRC）分频生成 PLL0 VCO 源时钟（CK_PLL0VCOSRC）。RCU_PLLALL 寄存器的 PLLSEL 位对 CK_PLL0SRC 时钟进行了描述。 VCO 源时钟频率范围必须在 1MHz 到 16MHz 之间 000000：保留 000001：CK_PLL0SRC 000010：CK_PLL0SRC / 2 000011：CK_PLL0SRC / 3 … 111111：CK_PLL0SRC / 63

6.1.3. 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。



位/位域	名称	描述
31:30	I2C0SEL[1:0]	I2C0时钟源选择 由软件置位或复位，控制I2C0时钟源 00：选择 CK_APB1 时钟作为 I2C0 源时钟 01：选择 CK_PLL2R 时钟作为 I2C0 源时钟 10：选择 CK_IRC64MDIV 时钟作为 I2C0 源时钟 11：选择CK_LPIRC4M时钟作为I2C0源时钟
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：选择 CK_AHB 时钟不分频 100：选择 CK_AHB 时钟 2 分频 101：选择 CK_AHB 时钟 4 分频 110：选择 CK_AHB 时钟 8 分频 111：选择 CK_AHB 时钟 16 分频
26:24	APB4PSC[2:0]	APB4 预分频选择 由软件置位或清零，控制 APB4 时钟分频因子 0xx：选择 CK_AHB 时钟不分频 100：选择 CK_AHB 时钟 2 分频 101：选择 CK_AHB 时钟 4 分频 110：选择 CK_AHB 时钟 8 分频 111：选择 CK_AHB 时钟 16 分频
23:22	保留	必须保持复位值。
21:16	RTCDIV[5:0]	RTC 时钟分频系数 由软件置位或清零。这些位用作将 HXTAL 时钟分频生成 RTC 时钟（不超过 1MHz） 000000：无时钟 000001：无时钟 000010：CK_HXTAL / 2 000011：CK_HXTAL / 3 … 111111：CK_HXTAL / 63
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：选择 CK_AHB 时钟不分频 100：选择 CK_AHB 时钟 2 分频

		101：选择 CK_AHB 时钟 4 分频 110：选择 CK_AHB 时钟 8 分频 111：选择 CK_AHB 时钟 16 分频
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：选择 CK_AHB 时钟不分频 100：选择 CK_AHB 时钟 2 分频 101：选择 CK_AHB 时钟 4 分频 110：选择 CK_AHB 时钟 8 分频 111：选择 CK_AHB 时钟 16 分频
9:8	保留	必须保持复位值。
7:4	AHBPSC[3:0]	AHB / AXI 预分频选择 由软件置位或清零，控制 AHB / AXI 时钟分频因子. 0xxx：选择 CK_SYS 时钟不分频 1000：选择 CK_SYS 时钟 2 分频 1001：选择 CK_SYS 时钟 4 分频 1010：选择 CK_SYS 时钟 8 分频 1011：选择 CK_SYS 时钟 16 分频 1100：选择 CK_SYS 时钟 64 分频 1101：选择 CK_SYS 时钟 128 分频 1110：选择 CK_SYS 时钟 256 分频 1111：选择 CK_SYS 时钟 512 分频
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_IRC64MDIV 时钟作为 CK_SYS 时钟源 01：选择 CK_HXTAL 时钟作为 CK_SYS 时钟源 10：选择 CK_LPIRC4M 时钟作为 CK_SYS 时钟源 11：选择 CK_PLL0P 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CK_IRC64MDIV作为系统时钟。 00：选择 CK_IRC64MDIV 时钟作为 CK_SYS 时钟源 01：选择 CK_HXTAL 时钟作为 CK_SYS 时钟源 10：选择 CK_LPIRC4M 时钟作为 CK_SYS 时钟源 11：选择 CK_PLL0P 时钟作为 CK_SYS 时钟源

6.1.4. 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	LPIRC4MSTBIC	LPIRC4MSTBIE	LPIRC4MSTBIF	CKMIC	PLL2STBIC	PLL1STBIC	PLL0STBIC	HXTALSTBIC	IRC64MSTBIC	LXTALSTBIC	IRC32KSTBIC
			w	r	w	rw	r	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	PLL2STBIE	PLL1STBIE	PLL0STBIE	HXTALSTBIE	IRC64MSTBIE	LXTALSTBIE	IRC32KSTBIE	CKMIF	PLL2STBIF	PLL1STBIF	PLL0STBIF	HXTALSTBIF	IRC64MSTBIF	LXTALSTBIF	IRC32KSTBIF
	rw	rw	rw	rw	rw	rw	rw	r	r	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	LPIRC4MSTBIC	LPIRC4M 时钟稳定中断清零 软件写 1 复位 LPIRC4MSTBIF 标志位 0：不复位 LPIRC4MSTBIF 标志位 1：复位 LPIRC4MSTBIF 标志位
25	LPIRC4MSTBIE	LPIRC4M 时钟稳定中断使能 软件置位和复位来使能/禁止LPIRC4M时钟稳定中断 0：禁止LPIRC4M时钟稳定中断 1：使能 LPIRC4M 时钟稳定中断
24	LPIRC4MSTBIF	LPIRC4M 时钟稳定中断标志位 当内部64MHz RC振荡器时钟稳定且LPIRC4MSTBIE位被置1时由硬件置1 软件置位 LPIRC4MSTBIC 位时清除该位 0：无LPIRC4M时钟稳定中断产生 1：产生 LPIRC4M 时钟稳定中断
23	CKMIC	HXTAL时钟阻塞中断清零 软件写 1 复位 CKMIF 标志位. 0：不复位 CKMIF 标志位 1：复位 CKMIF 标志位
22	PLL2STBIC	PLL2 时钟稳定中断清零 软件写 1 复位 PLL2STBIF 标志位 0：不复位 PLL2STBIF 标志位

		1：复位 PLL2STBIF 标志位
21	PLL1STBIC	PLL1 时钟稳定中断清零 软件写 1 复位 PLL1STBIF 标志位 0：不复位 PLL1STBIF 标志位 1：复位 PLL1STBIF 标志位
20	PLL0STBIC	PLL0 时钟稳定中断清零 软件写 1 复位 PLL0STBIF 标志位 0：不复位 PLL0STBIF 标志位 1：复位 PLL0STBIF 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	IRC64MSTBIC	IRC64M 时钟稳定中断清零 软件写 1 复位 IRC64MSTBIF 标志位 0：不复位 IRC64MSTBIF 标志位 1：复位 IRC64MSTBIF 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	保留	必须保持复位值。
14	PLL2STBIE	PLL2 时钟稳定中断使能 软件置位和复位来使能/禁止PLL2时钟稳定中断 0：禁止 PLL2 时钟稳定中断 1：使能 PLL2 时钟稳定中断
13	PLL1STBIE	PLL1 时钟稳定中断使能 软件置位和复位来使能/禁止PLL1时钟稳定中断 0：禁止 PLL1 时钟稳定中断 1：使能 PLL1 时钟稳定中断
12	PLL0STBIE	PLL0 时钟稳定中断使能 软件置位和复位来使能/禁止PLL0时钟稳定中断 0：禁止 PLL0 时钟稳定中断 1：使能 PLL0 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	IRC64MSTBIE	IRC64M 时钟稳定中断使能 软件置位和复位来使能/禁止IRC64M时钟稳定中断 0：禁止IRC64M时钟稳定中断 1：使能 IRC64M 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	PLL2STBIF	PLL2 时钟稳定中断标志位 当PLL2时钟稳定且PLL2STBIE位被置1时由硬件置1 软件置位 PLL2STBIC 位时清除该位 0：无PLL2时钟稳定中断产生 1：产生 PLL2 时钟稳定中断
5	PLL1STBIF	PLL1 时钟稳定中断标志位 当PLL1时钟稳定且PLL1STBIE位被置1时由硬件置1 软件置位 PLL1STBIC 位时清除该位 0：无PLL1时钟稳定中断产生 1：产生 PLL1 时钟稳定中断
4	PLL0STBIF	PLL0 时钟稳定中断标志位 当PLL0时钟稳定且PLL0STBIE位被置1时由硬件置1 软件置位 PLL0STBIC 位时清除该位 0：无PLL0时钟稳定中断产生 1：产生 PLL0 时钟稳定中断
3	HXTALSTBIF	HXTAL 时钟稳定中断标志位 当高速4 ~ 50 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	IRC64MSTBIF	IRC64M 时钟稳定中断标志位 当内部64MHz RC振荡器时钟稳定且IRC64MSTBIE位被置1时由硬件置1 软件置位 IRC64MSTBIC 位时清除该位 0：无IRC64M时钟稳定中断产生 1：产生 IRC64M 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

6.1.5. AHB1 复位寄存器（RCU_AHB1RST）

地址偏移：0x10

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		USBHS1RST	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
保留	USBHS0RST	保留													
	rw														

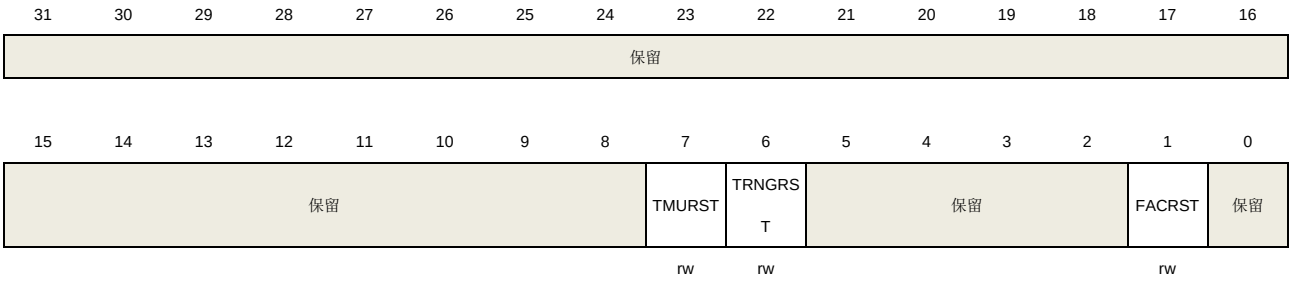
位/位域	名称	描述
31:30	保留	必须保持复位值。
29	USBHS1RST	USBHS1 复位 由软件置位或复位 0：无作用 1：复位 USBHS1
28:24	保留	必须保持复位值。
23	DMAMUXRST	DMAMUX 复位 由软件置位或复位 0：无作用

		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
20:15	保留	必须保持复位值。
14	USBHS0RST	USBHS0 复位 由软件置位或复位 0：无作用 1：复位 USBHS0
13:0	保留	必须保持复位值。

6.1.6. AHB2 复位寄存器（RCU_AHB2RST）

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。



位/位域	名称	描述
31:8	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

5:2	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。

6.1.7. AHB3 复位寄存器（RCU_AHB3RST）

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。



位/位域	名称	描述
31:7	保留	必须保持复位值。
6	OSPI1RST	OSPI1 复位 由软件置位或复位 0：无作用 1：复位 OSPI1
5	OSPI0RST	OSPI0 复位 由软件置位或复位 0：无作用 1：复位 OSPI0
4	OSPIMRST	OSPIM 复位 由软件置位或复位 0：无作用 1：复位 OSPIM
3	MDMARST	MDMA 复位 由软件置位或复位 0：无作用 1：复位 MDMA
2:1	保留	必须保持复位值。

0	EXMCRST	EXMC 复位 由软件置位或复位 0：无作用 1：复位 EXMC

6.1.8. AHB4 复位寄存器（RCU_AHB4RST）

地址偏移：0x1C
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CRCRST	保留				PKRST	PJRST	PHRST	PGRST	PFRST	PERST	PDRST	PCRST	PBRST	PARST
	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	CRCRST	CRC 复位 由软件置位或复位 0：无作用 1：复位 CRC
13:10	保留	必须保持复位值。
9	PKRST	GPIO 端口 K 复位 由软件置位或复位 0：无作用 1：复位 GPIO 端口 K
8	PJRST	GPIO 端口 J 复位 由软件置位或复位 0：无作用 1：复位 GPIO 端口 J
7	PHRST	GPIO 端口 H 复位 由软件置位或复位 0：无作用 1：复位 GPIO 端口 H
6	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

5	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
4	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
3	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
2	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
1	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
0	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

6.1.9. 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
UART7RST	UART6RST	DACRST	DACHOLDRST	CTCRST	保留		I2C3RST	I2C2RST	I2C1RST	I2C0RST	UART4RST	UART3RST	USART2RST	USART1RST	保留
rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI2RST	SPI1RST	保留		TIMER51RST	TIMER50RST	保留		TIMER23RST	TIMER22RST	TIMER6RST	TIMER5RST	TIMER4RST	TIMER3RST	TIMER2RST	TIMER1RST
rw	rw			rw	rw			rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述

31	UART7RST	UART7 复位 由软件置位或复位 0：无作用 1：复位 UART7
30	UART6RST	UART6 复位 由软件置位或复位 0：无作用 1：复位 UART6
29	DACRST	DAC 复位 由软件置位或复位 0：无作用 1：复位 DAC
28	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
27	CTCRST	CTC 复位 由软件置位或复位 0：无作用 1：复位 CTC
26:25	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	TIMER51RST	TIMER51 复位 由软件置位或复位 0：无作用 1：复位 TIMER51
10	TIMER50RST	TIMER50 复位 由软件置位或复位 0：无作用 1：复位 TIMER50
9:8	保留	必须保持复位值。
7	TIMER23RST	TIMER23 复位 由软件置位或复位 0：无作用 1：复位 TIMER23
6	TIMER6RST	TIMER6 复位

由软件置位或复位
0：无作用
1：复位 TIMER6

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

6.1.10. 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
TRIGSELRST	EDOUTRST	TIMER44RST	TIMER43RST	TIMER42RST	TIMER41RST	TIMER40RST	保留			SPI5RST	SPI4RST	HPDFRST	TIMER16RST	TIMER15RST	TIMER14RST
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		SPI3RST	SPI0RST	保留	ADC2RST	ADC1RST	ADC0RST	保留		USART5RST	USART0RST	保留		TIMER7RST	TIMER0RST

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	EDOUTRST	EDOUT 复位 由软件置位或复位 0：无作用 1：复位 EDOUT
29	TIMER44RST	TIMER44 复位 由软件置位或复位 0：无作用 1：复位 TIMER44
28	TIMER43RST	TIMER43 复位 由软件置位或复位 0：无作用 1：复位 TIMER43
27	TIMER42RST	TIMER42 复位 由软件置位或复位 0：无作用 1：复位 TIMER42
26	TIMER41RST	TIMER41 复位 由软件置位或复位 0：无作用 1：复位 TIMER41
25	TIMER40RST	TIMER40 复位 由软件置位或复位 0：无作用 1：复位 TIMER40
24:22	保留	必须保持复位值。
21	SPI5RST	SPI5 复位 由软件置位或复位 0：无作用 1：复位 SPI5
20	SPI4RST	SPI4 复位 由软件置位或复位 0：无作用

		1：复位 SPI4
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
15:14	保留	必须保持复位值。
13	SPI3RST	SPI3 复位 由软件置位或复位 0：无作用 1：复位 SPI3
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	ADC2RST	ADC2 复位 由软件置位或复位 0：无作用 1：复位所有 ADC2
9	ADC1RST	ADC1 复位 由软件置位或复位 0：无作用 1：复位所有 ADC1
8	ADC0RST	ADC0 复位 由软件置位或复位 0：无作用

		1：复位所有 ADC0
7:6	保留	必须保持复位值。
5	USART5RST	USART5 复位 由软件置位或复位 0：无作用 1：复位 USART5
4	USART0RST	USART0 复位 由软件置位或复位 0：无作用 1：复位 USART0
3:2	保留	必须保持复位值。
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

6.1.11. 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															

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

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	WWDGTRST	WWDGT 复位 由软件置位或复位 0：无作用 1：复位 WWDGT

0	保留	必须保持复位值。

6.1.12. APB4 复位寄存器（RCU_APB4RST）

地址偏移：0x2C

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											PMURST	LPDTSRST	VREFRST	CMPRST	SYSCFGRST
											rw	rw	rw	rw	rw

位/位域	名称	描述
31:5	保留	必须保持复位值。
4	PMURST	PMU 复位 由软件置位或复位 0：无作用 1：复位 PMU
3	LPDTSRST	LPDTS 复位 由软件置位或复位 0：无作用 1：复位 LPDTS
2	VREFRST	VREF 复位 由软件置位或复位 0：无作用 1：复位 VREF
1	CMPRST	CMP 复位 由软件置位或复位 0：无作用 1：复位 CMP
0	SYSCFGRST	SYSCFG 复位 由软件置位或复位 0：无作用 1：复位 SYSCFG

6.1.13. AHB1 使能寄存器（RCU_AHB1EN）

地址偏移：0x30

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	USBHS1ULPIEN	USBHS1EN	保留					DMAMUXEN	DMA1EN	DMA0EN	保留				
	rw	rw						rw	rw	rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBHS0ULPIEN	USBHS0EN	保留													
rw	rw														

位/位域	名称	描述
31	保留	必须保持复位值。
30	USBHS1ULPIEN	USBHS1 ULPI 时钟使能 由软件置位或复位 0：关闭 USBHS1 ULPI 时钟 1：开启 USBHS1 ULPI 时钟
29	USBH1SEN	USBHS1 时钟使能 由软件置位或复位 0：关闭 USBHS1 时钟 1：开启 USBHS1 时钟
28:24	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	USBHS0ULPIEN	USBHS0 ULPI 时钟使能 由软件置位或复位 0：关闭 USBHS0 ULPI 时钟 1：开启 USBHS0 ULPI 时钟

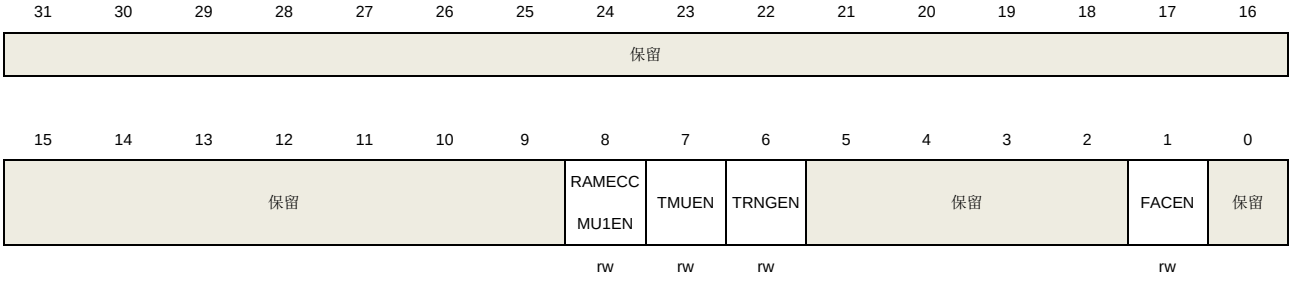
14	USBH0SEN	USBHS0 时钟使能 由软件置位或复位 0：关闭 USBHS0 时钟 1：开启 USBHS0 时钟
13:0	保留	必须保持复位值。

6.1.14. AHB2 使能寄存器（RCU_AHB2EN）

地址偏移：0x34

复位值：0x0000 0100

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
31:9	保留	必须保持复位值。
8	RAMECCMU1EN	RAMECCMU1 时钟使能 由软件置位或复位 0：关闭 RAMECCMU1 时钟 1：开启 RAMECCMU1 时钟
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
5:2	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。

6.1.15. AHB3 使能寄存器（RCU_AHB3EN）

地址偏移：0x38

复位值：0x0000 8400

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
CPUEN	保留				RAMECCMU0EN	保留			OSPI1EN	OSPI0EN	OSPIMEN	MDMAEN	保留		EXMCEN
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	CPUEN	CPU 时钟使能 由软件置位或复位 0：关闭 CPU 时钟 1：开启 CPU 时钟
14:11	保留	必须保持复位值。
10	RAMECCMU0EN	RAMECCMU0 时钟使能 由软件置位或复位 0：关闭 RAMECCMU0 时钟 1：开启 RAMECCMU0 时钟
9:7	保留	必须保持复位值。
6	OSPI1EN	OSPI1 时钟使能 由软件置位或复位 0：关闭 OSPI1 时钟 1：开启 OSPI1 时钟
5	OSPI0EN	OSPI0 时钟使能 由软件置位或复位 0：关闭 OSPI0 时钟 1：开启 OSPI0 时钟
4	OSPIMEN	OSPIM 时钟使能 由软件置位或复位 0：关闭 OSPIM 时钟 1：开启 OSPIM 时钟
3	MDMAEN	MDMA 时钟使能 由软件置位或复位 0：关闭 MDMA 时钟

		1：开启 MDMA 时钟
2:1	保留	必须保持复位值。
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

6.1.16. AHB4 使能寄存器（RCU_AHB4EN）

地址偏移：0x3C
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	CRCEN	BKPSRAMEN	保留			PKEN	PJEN	PHEN	PGEN	PFEN	PEEN	PDEN	PCEN	PBEN	PAEN
	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	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
13	BKPSRAMEN	BKPSRAM 时钟使能 由软件置位或复位 0：关闭 BKPSRAM 时钟 1：开启 BKPSRAM 时钟
12:10	保留	必须保持复位值。
9	PKEN	GPIO 端口 K 时钟使能 由软件置位或复位 0：关闭 GPIO 端口 K 时钟 1：开启 GPIO 端口 K 时钟
8	PJEN	GPIO 端口 J 时钟使能 由软件置位或复位 0：关闭 GPIO 端口 J 时钟 1：开启 GPIO 端口 J 时钟

7	PHEN	GPIO 端口 H 时钟使能 由软件置位或复位 0：关闭 GPIO 端口 H 时钟 1：开启 GPIO 端口 H 时钟
6	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
5	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
4	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
3	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
2	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
1	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
0	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

6.1.17. 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

UART7EN	UART6EN	DACEN	DACHOLDEN	CTCEN	保留	I2C3EN	I2C2EN	I2C1EN	I2C0EN	UART4EN	UART3EN	USART2EN	USART1EN	保留
rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SPI2EN	SPI1EN	保留	TIMER51EN	TIMER50EN	保留	TIMER23EN	TIMER22EN	TIMER6EN	TIMER5EN	TIMER4EN	TIMER3EN	TIMER2EN	TIMER1EN
rw	rw		rw	rw		rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	UART7EN	UART7 时钟使能 由软件置位或复位 0：关闭 UART7 时钟 1：开启 UART7 时钟
30	UART6EN	UART6 时钟使能 由软件置位或复位 0：关闭 UART6 时钟 1：开启 UART6 时钟
29	DACEN	DAC 时钟使能 由软件置位或复位 0：关闭 DAC 时钟 1：开启 DAC 时钟
28	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
27	CTCEN	CTC 保持时钟使能 由软件置位或复位 0：关闭 CTC 保持时钟 1：开启 CTC 保持时钟
26:25	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	TIMER51EN	TIMER51 时钟使能 由软件置位或复位 0：关闭 TIMER51 时钟 1：开启 TIMER51 时钟
10	TIMER51EN	TIMER51 时钟使能 由软件置位或复位 0：关闭 TIMER51 时钟

		1：开启 TIMER51 时钟
9:8	保留	必须保持复位值。
7	TIMER23EN	TIMER23 时钟使能 由软件置位或复位 0：关闭 TIMER23 时钟 1：开启 TIMER23 时钟
6	TIMER22EN	TIMER22 时钟使能 由软件置位或复位 0：关闭 TIMER22 时钟 1：开启 TIMER22 时钟
5	TIMER6EN	TIMER6 时钟使能 由软件置位或复位 0：关闭 TIMER6 时钟 1：开启 TIMER6 时钟
4	TIMER5EN	TIMER5 时钟使能 由软件置位或复位 0：关闭 TIMER5 时钟 1：开启 TIMER5 时钟
3	TIMER4EN	TIMER4 时钟使能 由软件置位或复位 0：关闭 TIMER4 时钟 1：开启 TIMER4 时钟
2	TIMER3EN	TIMER3 时钟使能 由软件置位或复位 0：关闭 TIMER3 时钟 1：开启 TIMER3 时钟
1	TIMER2EN	TIMER2 时钟使能 由软件置位或复位 0：关闭 TIMER2 时钟 1：开启 TIMER2 时钟
0	TIMER1EN	TIMER1 时钟使能 由软件置位或复位 0：关闭 TIMER1 时钟 1：开启 TIMER1 时钟

6.1.18. 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
TRIGSELEN	EDOUTEN	TIMER44EN	TIMER43EN	TIMER42EN	TIMER41EN	TIMER40EN	保留			SPI5EN	SPI4EN	HPDFEN	TIMER16EN	TIMER15EN	TIMER14EN
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		SPI3EN	SPI0EN	保留	ADC2EN	ADC1EN	ADC0EN	保留		USART5EN	USART0EN	保留		TIMER7EN	TIMER0EN
		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	EDOUTEN	EDOUT 时钟使能 由软件置位或复位 0：关闭 EDOUT 时钟 1：开启 EDOUT 时钟
29	TIMER44EN	TIMER44 时钟使能 由软件置位或复位 0：关闭 TIMER44 时钟 1：开启 TIMER44 时钟
28	TIMER43EN	TIMER43 时钟使能 由软件置位或复位 0：关闭 TIMER43 时钟 1：开启 TIMER43 时钟
27	TIMER42EN	TIMER42 时钟使能 由软件置位或复位 0：关闭 TIMER42 时钟 1：开启 TIMER42 时钟
26	TIMER41EN	TIMER41 时钟使能 由软件置位或复位 0：关闭 TIMER41 时钟 1：开启 TIMER41 时钟
25	TIMER40EN	TIMER40 时钟使能 由软件置位或复位 0：关闭 TIMER40 时钟 1：开启 TIMER40 时钟
24:20	保留	必须保持复位值。

21	SPI5EN	SPI5 时钟使能 由软件置位或复位 0：关闭 SPI5 时钟 1：开启 SPI5 时钟
20	SPI4EN	SPI4 时钟使能 由软件置位或复位 0：关闭 SPI4 时钟 1：开启 SPI4 时钟
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	TIMER15 时钟使能 由软件置位或复位 0：关闭 TIMER15 时钟 1：开启 TIMER15 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
15:14	保留	必须保持复位值。
13	SPI3EN	SPI3 时钟使能 由软件置位或复位 0：关闭 SPI3 时钟 1：开启 SPI3 时钟
12	SPI0EN	SPI0 时钟使能 由软件置位或复位 0：关闭 SPI0 时钟 1：开启 SPI0 时钟
11	保留	必须保持复位值。
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
7:6	保留	必须保持复位值。
5	USART5EN	USART5 时钟使能 由软件置位或复位 0：关闭 USART5 时钟 1：开启 USART5 时钟
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
3:2	保留	必须保持复位值。
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

6.1.19. 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。

31	30	29	28	27	26	25	24	23	22	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														WWDGTEN	保留
														rw	

位/位域	名称	描述

31:2	保留	必须保持复位值。
1	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
0	保留	必须保持复位值。

6.1.20. APB4 使能寄存器（RCU_APB4EN）

地址偏移：0x4C
复位值：0x0000 0010

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											PMUEN	LPDTSEN	VREFEN	CMPEN	SYSCFGEN
											rw	rw	rw	rw	rw

位/位域	名称	描述
31:5	保留	必须保持复位值
4	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
3	LPDTSEN	LPDTS 时钟使能 由软件置位或复位 0：关闭 LPDTS 时钟 1：开启 LPDTS 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	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
0	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

6.1.21. AHB1 睡眠模式使能寄存器（RCU_AHB1SPEN）

地址偏移：0x50
复位值：0x7EE3 C00F

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	USBHS1ULPISPEN	USBHS1SPEN	保留					DMAMUXSPEN	DMA1SPEN	DMA0SPEN	保留			SRAM1SPEN	SRAM0SPEN
	rw	rw						rw	rw	rw				rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBHS0ULPISPEN	USBHS0SPEN	保留													
rw	rw														

位/位域	名称	描述
31	保留	必须保持复位值。
30	USBHS1ULPISPEN	在睡眠模式下 USBHS1 ULPI 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 USBHS1 ULPI 时钟 1：在睡眠模式下开启 USBHS1 ULPI 时钟
29	USBHS1SPEN	在睡眠模式下 USBHS1 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 USBHS1 时钟 1：在睡眠模式下开启 USBHS1 时钟
28:24	保留	必须保持复位值。
23	DMAMUXSPEN	在睡眠模式下 DMAMUX 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 DMAMUX 时钟 1：在睡眠模式下开启 DMAMUX 时钟
22	DMA1SPEN	在睡眠模式下 DMA1 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 DMA1 时钟 1：在睡眠模式下开启 DMA1 时钟
21	DMA0SPEN	在睡眠模式下 DMA0 时钟使能 由软件置位或复位

0：在睡眠模式下关闭 DMA0 时钟
1：在睡眠模式下开启 DMA0 时钟

20:18	保留	必须保持复位值。
17	SRAM1SPEN	在睡眠模式下 SRAM1 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 SRAM1 时钟 1：在睡眠模式下开启 SRAM1 时钟
16	SRAM0SPEN	在睡眠模式下 SRAM0 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 SRAM0 时钟 1：在睡眠模式下开启 SRAM0 时钟
15	USBHS0ULPISPEN	在睡眠模式下 USBHS0 ULPI 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 USBHS0 ULPI 时钟 1：在睡眠模式下开启 USBHS0 ULPI 时钟
14	USBHS0SPEN	在睡眠模式下 USBHS0 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 USBHS0 时钟 1：在睡眠模式下开启 USBHS0 时钟
13:0	保留	必须保持复位值。

6.1.22. AHB2 睡眠模式使能寄存器（RCU_AHB2SPEN）

地址偏移：0x54
复位值：0x0000 01DF

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							RAMECCMU1SPEN	TMUSPEN	TRNGSPEN	保留				FACSPEN	保留
							rw	rw	rw					rw	

位/位域	名称	描述
31:9	保留	必须保持复位值。
8	RAMECCMU1SPEN	在睡眠模式下 RAMECCMU1 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 RAMECCMU1 时钟

		1：在睡眠模式下开启 RAMECCMU1 时钟
7	TMUSPEN	在睡眠模式下 TMU 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TMU 时钟 1：在睡眠模式下开启 TMU 时钟
6	TRNGSPEN	在睡眠模式下 TRNG 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TRNG 时钟 1：在睡眠模式下开启 TRNG 时钟
5:2	保留	必须保持复位值。
1	FACSPEN	在睡眠模式下 FAC 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 FAC 时钟 1：在睡眠模式下开启 FAC 时钟
0	保留	必须保持复位值。

6.1.23. AHB3 睡眠模式使能寄存器（RCU_AHB3SPEN）

地址偏移：0x58

复位值：0x0000 C77F

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															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FMCSPEN	AXISRAMSPEN	保留			RAMECCMU0SPEN	保留			OSPI1SPEN	OSPI0SPEN	OSPIMSPEN	MDMASPEN	保留		EXMCSPEN
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	FMCSPEN	在睡眠模式下 FMC 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 FMC 时钟 1：在睡眠模式下开启 FMC 时钟
14	AXISRAMSPEN	在睡眠模式下 AXI SRAM 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 AXI SRAM 时钟 1：在睡眠模式下开启 AXI SRAM 时钟

13:11	保留	必须保持复位值。
10	RAMECCMU0SPEN	在睡眠模式下 RAMECCMU0 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 RAMECCMU0 时钟 1：在睡眠模式下开启 RAMECCMU0 时钟
9:7	保留	必须保持复位值。
6	OSPI1SPEN	在睡眠模式下 OSPI1 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 OSPI1 时钟 1：在睡眠模式下开启 OSPI1 时钟
5	OSPI0SPEN	在睡眠模式下 OSPI0 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 OSPI0 时钟 1：在睡眠模式下开启 OSPI0 时钟
4	OSPIMSPEN	在睡眠模式下 OSPIM 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 OSPIM 时钟 1：在睡眠模式下开启 OSPIM 时钟
3	MDMASPEN	在睡眠模式下 MDMA 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 MDMA 时钟 1：在睡眠模式下开启 MDMA 时钟
2:1	保留	必须保持复位值。
0	EXMCSPEN	在睡眠模式下 EXMC 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 EXMC 时钟 1：在睡眠模式下开启 EXMC 时钟

6.1.24. AHB4 睡眠模式使能寄存器（RCU_AHB4SPEN）

地址偏移：0x5C

复位值：0x0000 63FF

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	CRCSPEN	BKPSRAMSPEN	保留			PKSPEN	PJSPEN	PHSPEN	PGSPEN	PFSPEN	PESPEN	PDSPEN	PCSPEN	PBSPEN	PASPEN

rw rw rw rw rw rw rw rw rw rw rw rw rw rw

位/位域	名称	描述
31:15	保留	必须保持复位值。
14	CRCSPEN	在睡眠模式下 CRC 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 CRC 时钟 1：在睡眠模式下开启 CRC 时钟
13	BKPSRAMSPEN	在睡眠模式下备份域 SRAM 时钟使能 由软件置位或复位 0：在睡眠模式下关闭备份域 SRAM 时钟 1：在睡眠模式下开启备份域 SRAM 时钟
12:10	保留	必须保持复位值。
9	PKSPEN	在睡眠模式下 GPIO 端口 K 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 GPIO 端口 K 时钟 1：在睡眠模式下开启 GPIO 端口 K 时钟
8	PJSPEN	在睡眠模式下 GPIO 端口 J 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 GPIO 端口 J 时钟 1：在睡眠模式下开启 GPIO 端口 J 时钟
7	PHSPEN	在睡眠模式下 GPIO 端口 H 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 GPIO 端口 H 时钟 1：在睡眠模式下开启 GPIO 端口 H 时钟
6	PGSPEN	在睡眠模式下 GPIO 端口 G 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 GPIO 端口 G 时钟 1：在睡眠模式下开启 GPIO 端口 G 时钟
5	PFSPEN	在睡眠模式下 GPIO 端口 F 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 GPIO 端口 F 时钟 1：在睡眠模式下开启 GPIO 端口 F 时钟
4	PEEN	在睡眠模式下 GPIO 端口 E 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 GPIO 端口 E 时钟 1：在睡眠模式下开启 GPIO 端口 E 时钟
3	PDSPEN	在睡眠模式下 GPIO 端口 D 时钟使能 由软件置位或复位

0：在睡眠模式下关闭 GPIO 端口 D 时钟
1：在睡眠模式下开启 GPIO 端口 D 时钟

2 PCSPEN 在睡眠模式下 GPIO 端口 C 时钟使能
由软件置位或复位
0：在睡眠模式下关闭 GPIO 端口 C 时钟
1：在睡眠模式下开启 GPIO 端口 C 时钟

1 PBSPEN 在睡眠模式下 GPIO 端口 B 时钟使能
由软件置位或复位
0：在睡眠模式下关闭 GPIO 端口 B 时钟
1：在睡眠模式下开启 GPIO 端口 B 时钟

0 PASPEN 在睡眠模式下 GPIO 端口 A 时钟使能
由软件置位或复位
0：在睡眠模式下关闭 GPIO 端口 A 时钟
1：在睡眠模式下开启 GPIO 端口 A 时钟

6.1.25. APB1 睡眠模式使能寄存器（RCU_APB1SPEN）

地址偏移：0x60
复位值：0xF9FF EFFF

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
UART7SPEN	UART6SPEN	DACSPEN	DACHOLDSPEN	CTCSPEN	保留		I2C3SPEN	I2C2SPEN	I2C1SPEN	I2C0SPEN	UART4SPEN	UART3SPEN	USART2SPEN	USART1SPEN	保留
rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI2SPEN	SPI1SPEN	保留		TIMER51SPEN	TIMER50SPEN	保留		TIMER23SPEN	TIMER22SPEN	TIMER6SPEN	TIMER5SPEN	TIMER4SPEN	TIMER3SPEN	TIMER2SPEN	TIMER1SPEN
rw	rw			rw	rw			rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	UART7SPEN	在睡眠模式下 UART7 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 UART7 时钟 1：在睡眠模式下开启 UART7 时钟
30	UART6SPEN	在睡眠模式下 UART6 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 UART6 时钟 1：在睡眠模式下开启 UART6 时钟
29	DACSPEN	在睡眠模式下 DAC 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 DAC 时钟

		1：在睡眠模式下开启 DAC 时钟
28	DACHOLDSPEN	在睡眠模式下 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
27	CTCSPEN	在睡眠模式下 CTC 保持时钟使能 由软件置位或复位 0：在睡眠模式下关闭 CTC 保持时钟 1：在睡眠模式下开启 CTC 保持时钟
26:25	保留	必须保持复位值。
24	I2C3SPEN	在睡眠模式下 I2C3 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 I2C3 时钟 1：在睡眠模式下开启 I2C3 时钟
23	I2C2SPEN	在睡眠模式下 I2C2 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 I2C2 时钟 1：在睡眠模式下开启 I2C2 时钟
22	I2C1SPEN	在睡眠模式下 I2C1 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 I2C1 时钟 1：在睡眠模式下开启 I2C1 时钟
21	I2C0SPEN	在睡眠模式下 I2C0 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 I2C0 时钟 1：在睡眠模式下开启 I2C0 时钟
20	UART4SPEN	在睡眠模式下 UART4 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 UART4 时钟 1：在睡眠模式下开启 UART4 时钟
19	UART3SPEN	在睡眠模式下 UART3 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 UART3 时钟 1：在睡眠模式下开启 UART3 时钟
18	USART2SPEN	在睡眠模式下 USART2 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 USART2 时钟 1：在睡眠模式下开启 USART2 时钟
17	USART1SPEN	在睡眠模式下 USART1 时钟使能

		由软件置位或复位 0：在睡眠模式下关闭 USART1 时钟 1：在睡眠模式下开启 USART1 时钟
16	保留	必须保持复位值。
15	SPI2SPEN	在睡眠模式下 SPI2 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 SPI2 时钟 1：在睡眠模式下开启 SPI2 时钟
14	SPI1SPEN	在睡眠模式下 SPI1 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 SPI1 时钟 1：在睡眠模式下开启 SPI1 时钟
13:12	保留	必须保持复位值。
11	TIMER51SPEN	在睡眠模式下 TIMER51 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER51 时钟 1：在睡眠模式下开启 TIMER51 时钟
10	TIMER50SPEN	在睡眠模式下 TIMER50 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER50 时钟 1：在睡眠模式下开启 TIMER50 时钟
9:8	保留	必须保持复位值。
7	TIMER23SPEN	在睡眠模式下 TIMER23 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER23 时钟 1：在睡眠模式下开启 TIMER23 时钟
6	TIMER22SPEN	在睡眠模式下 TIMER22 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER22 时钟 1：在睡眠模式下开启 TIMER22 时钟
5	TIMER6SPEN	在睡眠模式下 TIMER6 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER6 时钟 1：在睡眠模式下开启 TIMER6 时钟
4	TIMER5SPEN	在睡眠模式下 TIMER5 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER5 时钟 1：在睡眠模式下开启 TIMER5 时钟

3	TIMER4SPEN	在睡眠模式下 TIMER4 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER4 时钟 1：在睡眠模式下开启 TIMER4 时钟
2	TIMER3SPEN	在睡眠模式下 TIMER3 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER3 时钟 1：在睡眠模式下开启 TIMER3 时钟
1	TIMER2SPEN	在睡眠模式下 TIMER2 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER2 时钟 1：在睡眠模式下开启 TIMER2 时钟
0	TIMER1SPEN	在睡眠模式下 TIMER1 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER1 时钟 1：在睡眠模式下开启 TIMER1 时钟

6.1.26. APB2 睡眠模式使能寄存器（RCU_APB2SPEN）

地址偏移：0x64

复位值：0xFFFF 3733

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
TRIGSELSPEN	EDOUTSPEN	TIMER44SPEN	TIMER43SPEN	TIMER42SPEN	TIMER41SPEN	TIMER40SPEN	保留			SPI5SPEN	SPI4SPEN	HPDFSPEN	TIMER16SPEN	TIMER15SPEN	TIMER14SPEN
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		SPI3SPEN	SPI0SPEN	保留	ADC2SPEN	ADC1SPEN	ADC0SPEN	保留		USART5SPEN	USART0SPEN	保留		TIMER7SPEN	TIMER0SPEN
		rw	rw		rw	rw	rw			rw	rw			rw	rw

位/位域	名称	描述
31	TRIGSELSPEN	在睡眠模式下 TRIGSEL 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TRIGSEL 时钟 1：在睡眠模式下开启 TRIGSEL 时钟
30	EDOUTSPEN	在睡眠模式下 EDOUT 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 EDOUT 时钟 1：在睡眠模式下开启 EDOUT 时钟
29	TIMER44SPEN	在睡眠模式下 TIMER44 时钟使能

		由软件置位或复位 0：在睡眠模式下关闭 TIMER44 时钟 1：在睡眠模式下开启 TIMER44 时钟
28	TIMER43SPEN	在睡眠模式下 TIMER43 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER43 时钟 1：在睡眠模式下开启 TIMER43 时钟
27	TIMER42SPEN	在睡眠模式下 TIMER42 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER42 时钟 1：在睡眠模式下开启 TIMER42 时钟
26	TIMER41SPEN	在睡眠模式下 TIMER41 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER41 时钟 1：在睡眠模式下开启 TIMER41 时钟
25	TIMER40SPEN	在睡眠模式下 TIMER40 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER40 时钟 1：在睡眠模式下开启 TIMER40 时钟
24:22	保留	必须保持复位值。
21	SPI5SPEN	在睡眠模式下 SPI5 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 SPI5 时钟 1：在睡眠模式下开启 SPI5 时钟
20	SPI4SPEN	在睡眠模式下 SPI4 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 SPI4 时钟 1：在睡眠模式下开启 SPI4 时钟
19	HPDFSPEN	在睡眠模式下 HPDF 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 HPDF 时钟 1：在睡眠模式下开启 HPDF 时钟
18	TIMER16SPEN	在睡眠模式下 TIMER16 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER16 时钟 1：在睡眠模式下开启 TIMER16 时钟
17	TIMER15SPEN	在睡眠模式下 TIMER15 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER15 时钟

		1：在睡眠模式下开启 TIMER15 时钟
16	TIMER14SPEN	在睡眠模式下 TIMER14 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER14 时钟 1：在睡眠模式下开启 TIMER14 时钟
15:14	保留	必须保持复位值。
13	SPI3SPEN	在睡眠模式下 SPI3 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 SPI3 时钟 1：在睡眠模式下开启 SPI3 时钟
12	SPI0SPEN	在睡眠模式下 SPI0 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 SPI0 时钟 1：在睡眠模式下开启 SPI0 时钟
11	保留	必须保持复位值。
10	ADC2SPEN	在睡眠模式下 ADC2 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 ADC2 时钟 1：在睡眠模式下开启 ADC2 时钟
9	ADC1SPEN	在睡眠模式下 ADC1 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 ADC1 时钟 1：在睡眠模式下开启 ADC1 时钟
8	ADC0SPEN	在睡眠模式下 ADC0 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 ADC0 时钟 1：在睡眠模式下开启 ADC0 时钟
7:6	保留	必须保持复位值。
5	USART5SPEN	在睡眠模式下 USART5 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 USART5 时钟 1：在睡眠模式下开启 USART5 时钟
4	USART0SPEN	在睡眠模式下 USART0 时钟使能 由软件置位或复位 0：关闭在睡眠模式下 USART0 时钟 1：开启在睡眠模式下 USART0 时钟
3:2	保留	必须保持复位值。
1	TIMER7SPEN	在睡眠模式下 TIMER7 时钟使能

由软件置位或复位
0：关闭在睡眠模式下 TIMER7 时钟
1：开启在睡眠模式下 TIMER7 时钟

0	TIMER0SPEN	在睡眠模式下 TIMER0 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 TIMER0 时钟 1：在睡眠模式下开启 TIMER0 时钟

6.1.27. APB3 睡眠模式使能寄存器（RCU_APB3SPEN）

地址偏移：0x68
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														WWDGTSPEN	保留
														rw	

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	WWDGTSPEN	在睡眠模式下 WWDGT 时钟使能 由软件置位或复位 0：关闭在睡眠模式下 WWDGT 时钟 1：开启在睡眠模式下 WWDGT 时钟
0	保留	必须保持复位值。

6.1.28. APB4 睡眠模式使能寄存器（RCU_APB4SPEN）

地址偏移：0x6C
复位值：0x0000 001F

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											PMUSPEN	LPDTSSPEN	VREFSPEN	CMPSPEN	SYSCFGSPEN
											rw	rw	rw	rw	rw

位/位域	名称	描述
31:5	保留	必须保持复位值。
4	PMUSPEN	在睡眠模式下 PMU 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 PMU 时钟 1：在睡眠模式下开启 PMU 时钟
3	LPDTSSPEN	在睡眠模式下 LPDTS 时钟使能 由软件置位或复位 0：关闭在睡眠模式下 LPDTS 时钟 1：开启在睡眠模式下 LPDTS 时钟
2	VREFSPEN	在睡眠模式下 VREF 时钟使能 由软件置位或复位 0：关闭在睡眠模式下 VREF 时钟 1：开启在睡眠模式下 VREF 时钟
1	CMPSPEN	在睡眠模式下 CMP 时钟使能 由软件置位或复位 0：关闭在睡眠模式下 CMP 时钟 1：开启在睡眠模式下 CMP 时钟
0	SYSCFGSPEN	在睡眠模式下 SYSCFG 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 SYSCFG 时钟 1：在睡眠模式下开启 SYSCFG 时钟

6.1.29. 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															BKPRST
															rw

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

位/位域	名称	描述
31:17	保留	必须保持复位值。
16	BKPRST	备份域复位 由软件置位或复位 0：无作用 1：复位备份域
15	RTCEN	RTC 时钟使能 由软件置位或复位 0：关闭 RTC 时钟 1：开启 RTC 时钟
14:10	保留	必须保持复位值。
9:8	RTCSRC[1:0]	RTC 时钟源选择 由软件置位或清零来控制RTC的时钟源。一旦RTC的时钟源选择后，除了将备份域复位否则时钟源不能被改变。 00：没有时钟 01：选择 CK_LXTAL 时钟作为 RTC 的时钟源 10：选择 CK_IRC32K 时钟作为 RTC 的时钟源 11：选择 CK_HXTAL / RTCDIV 时钟作为 RTC 的时钟源，请参考 RCU_CFG0 寄存器的 RTCDIV 位域。
7	保留	必须保持复位值。
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时钟监视器 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。
4:3	LXTALDRI[1:0]	LXTAL 驱动能力 由软件置位或复位。当备份域复位时将复位该值 00：弱驱动能力 01：中低驱动能力 10：中高驱动能力 11：强驱动能力 **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

6.1.30. 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
LPRSTF	WWDGTRSTF	FWDGTRSTF	SWRSTF	PORRSTF	EPRSTF	BORRSTF	RSTFC	保留							
r	r	r	r	r	r	r	rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														IRC32KSTB	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
23:2	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

6.1.31. PLL 时钟附加控制寄存器（RCU_PLLADDCTL）

地址偏移：0x80

复位值：0xFF81 0101

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
PLL2PEN	PLL2REN	PLL2QEN	PLL1PEN	PLL1REN	PLL1QEN	PLL0PEN	PLL0REN	PLL0QEN	PLL2Q[6:0]						
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	PLL1Q[6:0]							保留	PLL0Q[6:0]						
	rw								rw						

位/位域	名称	描述
31	PLL2PEN	PLL2P 分频器输出使能 由软件置位或复位。只有在 PLL2EN 位为 0 时 PLL2PEN 位才可写。 0：禁止 CK_PLL2P 输出 1：使能 CK_PLL2P 输出
30	PLL2REN	PLL2R 分频器输出使能 由软件置位或复位。只有在 PLL2EN 位为 0 时 PLL2REN 位才可写。 0：禁止 CK_PLL2R 输出 1：使能 CK_PLL2R 输出
29	PLL2QEN	PLL2Q 分频器输出使能 由软件置位或复位。只有在 PLL2EN 位为 0 时 PLL2QEN 位才可写。 0：禁止 CK_PLL2Q 输出 1：使能 CK_PLL2Q 输出
28	PLL1PEN	PLL1P 分频器输出使能 由软件置位或复位。只有在 PLL1EN 位为 0 时 PLL1PEN 位才可写。 0：禁止 CK_PLL1P 输出 1：使能 CK_PLL1P 输出
27	PLL1REN	PLL1R 分频器输出使能 由软件置位或复位。只有在 PLL1EN 位为 0 时 PLL1REN 位才可写。 0：禁止 CK_PLL1R 输出 1：使能 CK_PLL1R 输出
26	PLL1QEN	PLL1Q 分频器输出使能 由软件置位或复位。只有在 PLL1EN 位为 0 时 PLL1QEN 位才可写。 0：禁止 CK_PLL1Q 输出 1：使能 CK_PLL1Q 输出
25	PLL0PEN	PLL0P 分频器输出使能 由软件置位或复位。只有在 PLL0EN 位为 0 时 PLL0PEN 位才可写。 0：禁止 CK_PLL0P 输出 1：使能 CK_PLL0P 输出
24	PLL0REN	PLL0R 分频器输出使能 由软件置位或复位。只有在 PLL0EN 位为 0 时 PLL0REN 位才可写。 0：禁止 CK_PLL0R 输出

		1：使能 CK_PLL0R 输出
23	PLL0QEN	PLL0Q 分频器输出使能 由软件置位或复位。只有在 PLL0EN 位为 0 时 PLL0QEN 位才可写。 0：禁止 CK_PLL0Q 输出 1：使能 CK_PLL0Q 输出
22:16	PLL2Q[6:0]	PLL2Q 输出频率的分频系数（PLL2 VCO 时钟作为输入） 当 PLL2 被关闭时由软件置位或清零。这些位域用做将 PLL2 VCO 时钟（CK_PLL2VCO）分频生成 PLL2Q 输出时钟（CK_PLL2Q）。RCU_PLL2 寄存器的 PLL2N 位域对 CK_PLL2VCO 时钟进行了描述。 0000000：CK_PLL2Q = CK_PLL2VCO 0000001：CK_PLL2Q = CK_PLL2VCO / 2 0000010：CK_PLL2Q = CK_ PLL2VCO / 3 0000011：CK_PLL2Q = CK_ PLL2VCO / 4 0000100：CK_PLL2Q = CK_ PLL2VCO / 5 … 1111111：CK_PLL2Q = CK_ PLL2VCO / 128
15	保留	必须保持复位值。
14:8	PLL1Q[6:0]	PLL1Q 输出频率的分频系数（PLL1 VCO 时钟作为输入） 当 PLL1 被关闭时由软件置位或清零。这些位域用做将 PLL1 VCO 时钟（CK_PLL1VCO）分频生成 PLL1Q 输出时钟（CK_PLL1Q）。RCU_PLL1 寄存器的 PLL1N 位域对 CK_PLL1VCO 时钟进行了描述。 0000000：CK_PLL1Q = CK_PLL1VCO 0000001：CK_ PLL1Q = CK_PLL1VCO / 2 0000010：CK_PLL1Q = CK_PLL1VCO / 3 0000011：CK_PLL1Q = CK_PLL1VCO / 4 0000100：CK_PLL1Q = CK_PLL1VCO / 5 … 1111111：CK_PLL1Q = CK_PLL1VCO / 128
7	保留	必须保持复位值。
6:0	PLL0Q[6:0]	PLL0Q 输出频率的分频系数（PLL0 VCO 时钟作为输入） 当 PLL0 被关闭时由软件置位或清零。这些位域用做将 PLL0 VCO 时钟（CK_PLL0VCO）分频生成 PLL0Q 输出时钟（CK_PLL0Q）。CK_PLL0Q 输出可用于 USBHS（48M）、TRNG（48M）。RCU_PLL0 寄存器的 PLL0N 位域对 CK_PLL0VCO 时钟进行了描述。 0000000：CK_PLL0Q = CK_PLL0VCO 0000001：CK_PLL0Q = CK_PLL0VCO / 2 0000010：CK_PLL0Q = CK_PLL0VCO / 3 0000011：CK_PLL0Q = CK_PLL0VCO / 4 0000100：CK_PLL0Q = CK_PLL0VCO / 5 … 1111111：CK_PLL0Q = CK_PLL0VCO / 128

6.1.32. PLL1 寄存器（RCU_PLL1）

地址偏移：0x84

复位值：0x0101 2020

配置PLL1时钟可参考下列公式：

CK_PLL1VCOSRC = CK_PLL1SRC / PLL1PSC

CK_PLL1VCO = CK_PLL1VCOSRC × （PLL1N + PLL1FRAN / 2^{13}）

CK_PLL1P = CK_ PLL1VCO / PLL1P

CK_PLL1R = CK_PLL1VCO / PLL1R

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	PLL1R[6:0]							保留	PLL1P[6:0]						
	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PLL1N[8:0]									PLL1PSC[5:0]					
	rw									rw					

位/位域	名称	描述
31	保留	必须保持复位值。
30:24	PLL1R[6:0]	PLL1R 输出频率的分频系数（PLL1 VCO 时钟作为输入） 当 PLL1 被关闭时由软件置位或清零。这些位域用做将 PLL1 VCO 时钟（CK_PLL1VCO）分频生成 PLL1R 输出时钟（CK_PLL1R）。RCU_PLL1 寄存器的 PLL1N 位域对 CK_PLL1VCO 时钟进行了描述。 0000000：CK_PLL1R = CK_PLL1VCO 0000001：CK_PLL1R = CK_PLL1VCO / 2 0000010：CK_PLL1R = CK_PLL1VCO / 3 0000011：CK_PLL1R = CK_PLL1VCO / 4 0000100：CK_PLL1R = CK_PLL1VCO / 5 … 1111111：CK_PLL1R = CK_PLL1VCO / 128
23	保留	必须保持复位值。
22:16	PLL1P[6:0]	PLL1P 输出频率的分频系数（PLL1 VCO 时钟作为输入） 当 PLL1 被关闭时由软件置位或清零。这些位域用做将 PLL1 VCO 时钟（CK_PLL1VCO）分频生成 PLL1P 输出时钟（CK_PLL1P）。RCU_PLL1 寄存器的 PLL1N 位域对 CK_PLL1VCO 时钟进行了描述。 0000000：CK_PLL1P = CK_PLL1VCO 0000001：CK_PLL1P = CK_PLL1VCO / 2 0000010：CK_PLL1P = CK_PLL1VCO / 3 0000011：CK_PLL1P = CK_PLL1VCO / 4 0000100：CK_PLL1P = CK_PLL1VCO / 5 …

		1111111：CK_PLL1P = CK_PLL1VCO / 128
15	保留	必须保持复位值。
14:6	PLL1N[8:0]	PLL1 VCO 时钟倍频因子 当 PLL1 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 PLL1 VCO 源时钟（CK_PLL1VCOSRC）倍频生成 PLL1 VCO 输出时钟（CK_PLL1VCO）。RCU_PLL1 寄存器的 PLL1PSC 位域对 CK_PLL1VCOSRC 时钟进行了描述。 **注意：**CK_PLL1VCO 时钟频率范围必须在 150MHz 到 836MHz 之间 PLL1N 的值必须满足：参考 PLL0N 取值描述。 000000000：保留 … 000000111：保留 000001000：PLL1N = 9 … 001000000：PLL1N = 65 001000001：PLL1N = 66 … 111111111：PLL1N = 512
5:0	PLL1PSC[5:0]	PLL1 VCO 源时钟分频器 当 PLL1 被关闭时由软件置位或清零。这些位域用做将 PLL1 源时钟（CK_PLL1SRC）分频生成 PLL1 VCO 源时钟（CK_PLL1VCOSRC）。RCU_PLLALL 寄存器的 PLL1SEL 位对 CK_PLL1SRC 时钟进行了描述。 VCO 源时钟频率范围必须在 1MHz 到 16MHz 之间 000000：保留 000001：CK_PLL1SRC 000010：CK_PLL1SRC / 2 000011：CK_PLL1SRC / 3 … 111111：CK_PLL1SRC / 63

6.1.33. PLL2 寄存器（RCU_PLL2）

地址偏移：0x88
复位值：0x0101 2020

配置PLL2时钟可参考下列公式：
CK_PLL2VCOSRC = CK_PLL2SRC / PLL2PSC
CK_PLL2VCO = CK_ PLL2VCOSRC × （PLL2N + PLL2FRAN / 2^{13}）
CK_PLL2P = CK_ PLL2VCO / PLL2P
CK_PLL2R = CK_ PLL2VCO / PLL2R

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	PLL2R[6:0]	保留	PLL2P[6:0]
	rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PLL2N[8:0]								PLL2PSC[5:0]						
	rw								rw						

位/位域	名称	描述
31	保留	必须保持复位值。
30:24	PLL2R[6:0]	PLL2R 输出频率的分频系数（PLL2 VCO 时钟作为输入） 当 PLL2 被关闭时由软件置位或清零。这些位域用做将 PLL2 VCO 时钟（CK_PLL2VCO）分频生成 PLL2R 输出时钟（CK_PLL2R）。RCU_PLL2 寄存器的 PLL2N 位域对 CK_PLL2VCO 时钟进行了描述。 0000000：CK_PLL2R = CK_PLL2VCO 0000001：CK_PLL2R = CK_PLL2VCO / 2 0000010：CK_PLL2R = CK_PLL2VCO / 3 0000011：CK_PLL2R = CK_PLL2VCO / 4 0000100：CK_PLL2R = CK_PLL2VCO / 5 … 1111111：CK_PLL2R = CK_PLL2VCO / 128
23	保留	必须保持复位值。
22:16	PLL2P[6:0]	PLL2P 输出频率的分频系数（PLL2 VCO 时钟作为输入） 当 PLL2 被关闭时由软件置位或清零。这些位域用做将 PLL2 VCO 时钟（CK_PLL2VCO）分频生成 PLL2P 输出时钟（CK_PLL2P）。RCU_PLL2 寄存器的 PLL2N 位域对 CK_PLL2VCO 时钟进行了描述。 0000000：CK_PLL2P = CK_PLL2VCO 0000001：CK_PLL2P = CK_PLL2VCO / 2 0000010：CK_PLL2P = CK_PLL2VCO / 3 0000011：CK_PLL2P = CK_PLL2VCO / 4 0000100：CK_PLL2R = CK_PLL2VCO / 5 … 1111111：CK_PLL2R = CK_PLL2VCO / 128
15	保留	必须保持复位值。
14:6	PLL2N[8:0]	PLL2 VCO 时钟倍频因子 当 PLL2 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 PLL2 VCO 源时钟（CK_PLL2VCOSRC）倍频生成 PLL2 VCO 输出时钟（CK_PLL2VCO）。RCU_PLL2 寄存器的 PLL2PSC 位域对 CK_PLL2VCOSRC 时钟进行了描述。 **注意：**CK_PLL2VCO 时钟频率范围必须在 150MHz 到 836MHz 之间 PLL2N 的值必须满足：参考 PLL0N 取值描述。 000000000：保留 … 000000111：保留

000001000：PLL2N = 9
…
001000000：PLL2N = 65
001000001：PLL2N = 66
…
111111111：PLL2N = 512

5:0	PLL2PSC[5:0]	PLL2 VCO 源时钟分频器 当 PLL2 被关闭时由软件置位或清零。这些位域用做将 PLL2 源时钟（CK_PLL2SRC）分频生成 PLL2 VCO 源时钟（CK_PLL2VCOSRC）。RCU_PLLALL 寄存器的 PLL2SEL 位对 CK_PLL2SRC 时钟进行了描述。 VCO 源时钟频率范围必须在 1MHz 到 16MHz 之间 000000：保留 000001：CK_PLL2SRC 000010：CK_PLL2SRC / 2 000011：CK_PLL2SRC / 3 … 111111：CK_PLL2SRC / 63

6.1.34. 时钟配置寄存器 1（RCU_CFG1）

地址偏移：0x8C
复位值：0x0000 3F00

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
HPDFSEL		保留					TIMERSEL	USART5SEL[1:0]		USART2SEL[1:0]		USART1SEL[1:0]		PLL2RDIV[1:0]	
rw							rw	rw		rw		rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PERSEL[1:0]		CAN2SEL[1:0]		CAN1SEL[1:0]		CAN0SEL[1:0]		保留						USART0SEL[1:0]	
rw		rw		rw		rw								rw	

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：选择 CK_APB2 时钟作为 HPDF 源时钟 1：选择 CK_AHB 时钟作为 HPDF 源时钟
30:25	保留	必须保持复位值。
24	TIMERSEL	TIMER 时钟源选择 由软件置位或复位 该位定义了所有定时器的时钟源选择 0：如果 RCU_CFG0 寄存器的 APB1PSC/APB2PSC 位域的值为 0b0xx（CK_APBx = CK_AHB）或 0b100（CK_APBx = CK_AHB/2），定时器时钟等于 CK_AHB（CK_TIMERx = CK_AHB），否则定时器时钟等于 APB 时钟的两倍（在

APB1 域的定时器：CK_TIMERx = 2 x CK_APB1，在 APB2 域的定时器：CK_TIMERx = 2 x CK_APB2）。
1：如果 RCU_CFG0 寄存器的 APB1PSC/APB2PSC 位域的值为 0b0xx（CK_APBx = CK_AHB），0b100（CK_APBx = CK_AHB/2），或 0b101（CK_APBx = CK_AHB/4），定时器时钟等于 CK_AHB（CK_TIMERx = CK_AHB）。否则定时器时钟等于 APB 时钟的四倍（在 APB1 域的定时器：CK_TIMERx = 4 x CK_APB1；在 APB2 域的定时器：CK_TIMERx = 4 x CK_APB2）。

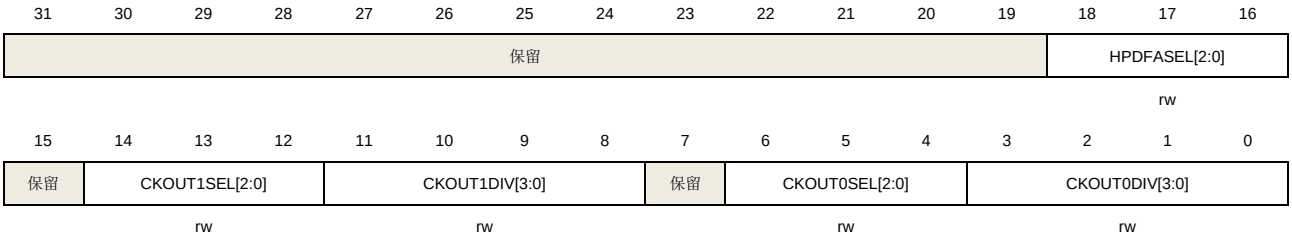
23:22	USART5SEL[1:0]	USART5时钟源选择 由软件置位或复位，控制USART5时钟源 00：选择 CK_APB2 时钟作为 USART5 源时钟 01：选择 CK_AHB 时钟作为 USART5 源时钟 10：选择 CK_LXTAL 时钟作为 USART5 源时钟 11：选择CK_IRC64MDIV时钟作为USART5源时钟
21:20	USART2SEL[1:0]	USART2时钟源选择 由软件置位或复位，控制USART2时钟源 00：选择 CK_APB1 时钟作为 USART2 源时钟 01：选择 CK_AHB 时钟作为 USART2 源时钟 10：选择 CK_LXTAL 时钟作为 USART2 源时钟 11：选择CK_IRC64MDIV时钟作为USART2源时钟
19:18	USART1SEL[1:0]	USART1时钟源选择 由软件置位或复位，控制USART1时钟源 00：选择 CK_APB1 时钟作为 USART1 源时钟 01：选择 CK_AHB 时钟作为 USART1 源时钟 10：选择 CK_LXTAL 时钟作为 USART1 源时钟 11：选择CK_IRC64MDIV时钟作为USART1源时钟
17:16	PLL2RDIV[1:0]	PLL2R 时钟的分频因子 当 PLL2 时钟被关闭时由软件置位或复位。 00：CK_PLL2R / 2 01：CK_PLL2R / 4 10：CK_PLL2R / 8 11：CK_PLL2R / 16
15:14	PERSEL[1:0]	CK_PER时钟源选择 由软件置位或复位，控制CK_PER时钟源 00：选择 CK_IRC64MDIV 时钟作为 CK_PER 源时钟 01：选择 CK_LPIRC4M 时钟作为 CK_PER 源时钟 10：选择 CK_HXTAL 时钟作为 CK_PER 源时钟 11：保留
13:12	CAN2SEL[1:0]	CAN2时钟源选择 由软件置位或复位，控制CAN2时钟源 00：选择 CK_HXTAL 时钟作为 CAN2 源时钟

		01：选择 CK_APB2 时钟作为 CAN2 源时钟 10：选择 CK_APB2 / 2 时钟作为 CAN2 源时钟 11：选择CK_IRC64MDIV时钟作为CAN2源时钟
11:10	CAN1SEL[1:0]	CAN1时钟源选择 由软件置位或复位，控制CAN1时钟源 00：选择 CK_HXTAL 时钟作为 CAN1 源时钟 01：选择 CK_APB2 时钟作为 CAN1 源时钟 10：选择 CK_APB2 / 2 时钟作为 CAN1 源时钟 11：选择CK_IRC64MDIV时钟作为CAN1源时钟
9:8	CAN0SEL[1:0]	CAN0时钟源选择 由软件置位或复位，控制CAN0时钟源 00：选择 CK_HXTAL 时钟作为 CAN0 源时钟 01：选择 CK_APB2 时钟作为 CAN0 源时钟 10：选择 CK_APB2 / 2 时钟作为 CAN0 源时钟 11：选择CK_IRC64MDIV时钟作为CAN0源时钟
7:2	保留	必须保持复位值。
1:0	USART0SEL[1:0]	USART0时钟源选择 由软件置位或复位，控制USART0时钟源 00：选择 CK_APB2 时钟作为 USART0 源时钟 01：选择 CK_AHB 时钟作为 USART0 源时钟 10：选择 CK_LXTAL 时钟作为 USART0 源时钟 11：选择CK_IRC64MDIV时钟作为USART0源时钟

6.1.35. 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。



位/位域	名称	描述
31:15	保留	必须保持复位值。
18:16	HPDFASEL[2:0]	HPDF音频时钟源选择 由软件置位或复位，控制HPDF音频时钟源 000：选择CK_PLL0Q时钟作为HPDF音频源时钟 001：选择CK_PLL1P时钟作为HPDF音频源时钟

		010：选择CK_PLL2P时钟作为HPDF音频源时钟 011：选择I2S_CKIN时钟作为HPDF音频源时钟 100：选择CK_PER时钟作为HPDF音频源时钟 其它：保留
14:12	CKOUT1SEL[2:0]	CKOUT1时钟源选择 由软件置位或清零 000：选择系统时钟 001：选择CK_PLL1R时钟 010：选择高速晶体振荡器时钟（HXTAL） 011：选择CK_PLL0P时钟 100：选择CK_LPIRC4M 时钟 101：选择CK_IRC32K时钟 110：选择CK_PLL2R时钟 111：保留 **注意：**对该位域的配置可能会造成对CK_OUT1的干扰，强烈建议仅在复位后但在使能HXTAL和PLLs之前来配置这些位。
11:8	CKOUT1DIV[3:0]	CK_OUT1分频器，来降低CK_OUT1频率 CK_OUT1时钟源的选择参考 RCU_CFG2寄存器的14:12位 0000：保留 0001：CK_OUT1不分频 0010：CK_OUT1被2分频 0011：CK_OUT1被3分频 0100：CK_OUT1被4分频 … 1111：CK_OUT1被15分频 **注意：**对该位域的配置可能会造成对CK_OUT1的干扰，强烈建议仅在复位后但在使能HXTAL和PLLs之前来配置这些位。
7	保留	必须保持复位值。
6:4	CKOUT0SEL[2:0]	CKOUT0时钟源选择 由软件置位或清零 000：选择CK_IRC64MDIV时钟 001：选择CK_LXTAL时钟 010：选择高速晶体振荡器时钟（HXTAL） 011：选择CK_PLL0P时钟 100：选择CK_IRC48M时钟 101：选择CK_PER时钟 110：选择USBHS0 60M时钟 111：选择USBHS1 60M时钟 **注意：**对该位域的配置可能会造成对CK_OUT0的干扰，强烈建议仅在复位后但在使能HXTAL和PLLs之前来配置这些位。

3:0	CKOUT0DIV[3:0]	CK_OUT0分频器，来降低CK_OUT0频率 CK_OUT0时钟源的选择参考RCU_CFG2寄存器的6:4位 0000：保留 0001：CK_OUT1不分频 0010：CK_OUT1被2分频 0011：CK_OUT1被3分频 0100：CK_OUT1被4分频 … 1111：CK_OUT1被15分频 **注意：**对该位域的配置可能会造成对CK_OUT0的干扰，强烈建议仅在复位后但在使能HXTAL和PLLs之前来配置这些位。

6.1.36. 时钟配置寄存器 3（RCU_CFG3）

地址偏移：0x94

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		ADC2SEL[1:0]		ADC01SEL[1:0]		保留	DSPWUSSEL	保留							
		rw		rw			rw								

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

位/位域	名称	描述
31:30	保留	必须保持复位值。
29:28	ADC2SEL[1:0]	ADC2时钟源选择 由软件置位或复位，控制ADC2时钟源 00：选择 CK_PLL1P 时钟作为 ADC2 源时钟 01：选择 CK_PLL2R 时钟作为 ADC2 源时钟 10：选择 CK_PER 时钟作为 ADC2 源时钟 11：保留
27:26	ADC01SEL[1:0]	ADC0与ADC1时钟源选择 由软件置位或复位，控制ADC0与ADC1时钟源 00：选择 CK_PLL1P 时钟作为 ADC0 与 ADC1 源时钟 01：选择 CK_PLL2R 时钟作为 ADC0 与 ADC1 源时钟 10：选择 CK_PER 时钟作为 ADC0 与 ADC1 源时钟 11：保留
25	保留	必须保持复位值。
24	DSPWUSSEL	唤醒深度睡眠的系统时钟源选择

由软件置位或复位，控制从深度睡眠唤醒的系统时钟源
该位也用于控制 HXTAL 阻塞时的系统时钟
0：选择 CK_IRC64MDIV 时钟作为从深度睡眠唤醒的系统时钟源
1：选择 CK_LPIRC4M 时钟作为内核从深度睡眠唤醒的系统时钟源
注意：如果 DSPWUSSEL = ‘1’且外设时钟源选择 CK_IRC64MDIV，当系统通过此外设从深度睡眠模式唤醒后，如果此时关闭该外设的唤醒功能，将导致 IRC64M 时钟关闭，此时外设将没有时钟驱动。这种情况下，用户需要重新置位 RCU_CTL 寄存器中的 IRC64MEN 位，再次打开 IRC64M 时钟。当 CKMEN 位置位且系统时钟为 CK_HXTAL 或者将系统时钟切换到 HXTAL 时，该位不能被修改。

23:6	保留	必须保持复位值。
5:4	I2C3SEL[1:0]	I2C3时钟源选择 由软件置位或复位，控制I2C3时钟源 00：选择 CK_APB1 时钟作为 I2C3 源时钟 01：选择 CK_PLL2R 时钟作为 I2C3 源时钟 10：选择 CK_IRC64MDIV 时钟作为 I2C3 源时钟 11：选择CK_LPIRC4M时钟作为I2C3源时钟
3:2	I2C2SEL[1:0]	I2C2时钟源选择 由软件置位或复位，控制I2C2时钟源 00：选择 CK_APB1 时钟作为 I2C2 源时钟 01：选择 CK_PLL2R 时钟作为 I2C2 源时钟 10：选择 CK_IRC64MDIV 时钟作为 I2C2 源时钟 11：选择CK_LPIRC4M时钟作为I2C2源时钟
1:0	I2C1SEL[1:0]	I2C1时钟源选择 由软件置位或复位，控制I2C1时钟源 00：选择 CK_APB1 时钟作为 I2C1 源时钟 01：选择 CK_PLL2R 时钟作为 I2C1 源时钟 10：选择 CK_IRC64MDIV 时钟作为 I2C1 源时钟 11：选择CK_LPIRC4M时钟作为I2C1源时钟

6.1.37. PLL 控制寄存器（RCU_PLLALL）

地址偏移：0x98
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域	名称	描述
31:18	保留	必须保持复位值
17:16	PLLSEL[1:0]	PLLs时钟源选择 由软件置位或复位，控制PLLs时钟源 00：选择 CK_IRC64MDIV 时钟作为 PLL、PLL1、PLL2 源时钟 01：选择 CK_LPIRC4M 时钟作为 PLL、PLL1、PLL2 源时钟 10：选择 CK_HXTAL 时钟作为 PLL、PLL1、PLL2 源时钟 11：无时钟作为PLL、PLL1、PLL2源时钟
15:11	保留	必须保持复位值
10	PLL2VCOSEL	PLL2 VCO 范围选择 当PLL2被关闭时由软件置位或清零 0：选择宽范围（192 - 836MHz），PLL2 输入时钟频率范围应为 2-16MHz 1：选择窄范围（150 - 420MHz），PLL2 输入时钟频率范围应为 1-2MHz
9:8	PLL2RNG[1:0]	PLL2 输入时钟范围 当 PLL2 被关闭时由软件置位或清零 00：PLL2 输入时钟频率范围 1 - 2MHz 01：PLL2 输入时钟频率范围 2 - 4MHz 10：PLL2 输入时钟频率范围 4 - 8MHz 11：PLL2 输入时钟频率范围 8 - 16MHz
7	保留	必须保持复位值。
6	PLL1VCOSEL	PLL1 VCO 范围选择 当PLL被关闭时由软件置位或清零 0：选择宽范围（192 - 836MHz），PLL1 输入时钟频率范围应为 2-16MHz 1：选择窄范围（150 - 420MHz），PLL1 输入时钟频率范围应为 1-2MHz
5:4	PLL1RNG[1:0]	PLL1 输入时钟范围 当 PLL1 被关闭时由软件置位或清零 00：PLL1 输入时钟频率范围 1 - 2MHz 01：PLL1 输入时钟频率范围 2 - 4MHz 10：PLL1 输入时钟频率范围 4 - 8MHz 11：PLL1 输入时钟频率范围 8 - 16MHz
3	保留	必须保持复位值。
2	PLL0VCOSEL	PLL0 VCO 范围选择 当PLL0被关闭时由软件置位或清零 0：选择宽范围（192 - 836MHz），PLL0 输入时钟频率范围应为 2-16MHz 1：选择窄范围（150 - 420MHz），PLL0 输入时钟频率范围应为 1-2MHz
1:0	PLL0RNG[1:0]	PLL0 输入时钟范围 当 PLL0 被关闭时由软件置位或清零 00：PLL0 输入时钟频率范围 1 - 2MHz 01：PLL0 输入时钟频率范围 2 - 4MHz 10：PLL0 输入时钟频率范围 4 - 8MHz

11：PLL0 输入时钟频率范围 8 - 16MHz

6.1.38. PLL0 小数配置寄存器（RCU_PLL0FRA）

地址偏移：0x9C

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15	PLL0FRAEN	PLL0小数锁存使能 由软件置位或复位，用于将PLL0FRAN的内容锁存到 Sigma-Delta调制器。当PLL0FRAEN从0切换到1，PLL0FRAEN的值将被转移到调制器中。
14:13	保留	必须保持复位值
12:0	PLL0FRAN[12:0]	PLL0 VCO倍频因子的小数部分 由软件置位或复位，用于控制PLL0 VCO倍频因子的小数部分。该位域可以动态修改从而对PLL0 VCO进行微调。 必须配置该值使 PLL0 VCO 输出频率为如下范围： 当 PLL0VCOSEL 为 0 时，范围为 192 - 836MHz 当 PLL0VCOSEL 为 1 时，范围为 150 - 420MHz

6.1.39. PLL1 小数配置寄存器（RCU_PLL1FRA）

地址偏移：0xA0

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15	PLL1FRAEN	PLL1小数锁存使能 由软件置位或复位，用于将PLL1FRAN的内容锁存到 Sigma-Delta调制器。当PLL1FRAEN从0切换到1，PLL1FRAEN的值将被转移到调制器中。
14:13	保留	必须保持复位值
12:0	PLL1FRAN[12:0]	PLL1 VCO倍频因子的小数部分 由软件置位或复位，用于控制PLL1 VCO倍频因子的小数部分。该位域可以动态修改从而对PLL1 VCO进行微调。 必须配置该值使 PLL1 VCO 输出频率为如下范围： 当 PLL1VCOSEL 为 0 时，范围为 192 - 836MHz 当 PLL1VCOSEL 为 1 时，范围为 150 - 420MHz

6.1.40. PLL2 小数配置寄存器（RCU_PLL2FRA）

地址偏移：0xA4

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15	PLL2FRAEN	PLL2小数锁存使能 由软件置位或复位，用于将PLL2FRAN的内容锁存到 Sigma-Delta调制器。当PLL2FRAEN从0切换到1，PLL2FRAEN的值将被转移到调制器中。
14:13	保留	必须保持复位值
12:0	PLL2FRAN[12:0]	PLL2 VCO倍频因子的小数部分 由软件置位或复位，用于控制PLL2 VCO倍频因子的小数部分。该位域可以动态修改从而对PLL2 VCO进行微调。 必须配置该值使 PLL2 VCO 输出频率为如下范围： 当 PLL2VCOSEL 为 0 时，范围为 192 - 836MHz 当 PLL2VCOSEL 为 1 时，范围为 150 - 420MHz

6.1.41. 附加时钟控制寄存器 0（RCU_ADDCTL0）

地址偏移：0xC0

复位值：0x8000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRC48MCALIB[7:0]								保留						IRC48MSTB	IRC48MEN
r														r	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														PLL48MSEL	CK48MSEL
														rw	rw

位/位域	名称	描述
31:24	IRC48MCALIB[7:0]	内部 48MHz RC 振荡器校准值寄存器 上电时自动加载这些位
23:18	保留	必须保持复位值。
17	IRC48MSTB	内部 48MHz RC 振荡器时钟稳定标志位 硬件置‘1’来指示IRC48M振荡器时钟是否稳定待用 0：IRC48M未稳定 1：IRC48M 已稳定
16	IRC48MEN	内部 48MHz RC 振荡器使能 由软件置位和复位。当进入深度睡眠或待机模式后由硬件复位 0：关闭IRC48M时钟 1：打开 IRC48M 时钟
15:2	保留	必须保持复位值。
1	PLL48MSEL	PLL48M时钟源选择 由软件置位和复位。该位用于选择CK_PLL0Q时钟或CK_PLL2P时钟作为PLL48M的时钟源 0：选择CK_PLL0Q时钟 1：选择 CK_PLL2P 时钟
0	CK48MSEL	48MHz时钟源选择 由软件置位和复位。该位用于选择IRC48M时钟或PLL48M时钟作为CK48M时钟源。CK48M时钟为TRNG/USBHS模块提供时钟。RCU_ADDCTL0寄存器的PLL48MSEL位对PLL48M时钟进行了描述。 0：不选择IRC48M时钟（通过PLL48MSEL位选择使用CK_PLL0Q时钟或CK_PLL2P时钟） 1：选择 IRC48M 时钟

6.1.42. 附加时钟控制寄存器 1（RCU_ADDCTL1）

地址偏移：0xC4

复位值：0x0000 7080

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
PLLUSBHS1STB	PLLUSBHS1EN	PLLUSBHS0STB	PLLUSBHS0EN	保留							LPIRC4MDSPEN	保留		IRC64MDIV[1:0]	
r	rw	r	rw								rw			rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPIRC4MCALIB[7:0]								LPIRC4MADJ[5:0]						LPIRC4MSTB	LPIRC4MEN
r								rw						r	rw

位/位域	名称	描述
31	PLLUSBHS1STB	PLLUSBHS1 时钟稳定标志位 硬件置1来表示PLLUSBHS1输出时钟是否稳定待用 0：PLLUSBHS1未稳定 1：PLLUSBHS1 已稳定
30	PLLUSBHS1EN	PLLUSBHS1时钟使能 软件置位或复位，当进入深度睡眠或待机模式时由硬件复位 0：PLLUSBHS1被关闭 1：PLLUSBHS1 被打开
29	PLLUSBHS0STB	PLLUSBHS1 时钟稳定标志位 硬件置1来表示PLLUSBHS1输出时钟是否稳定待用 0：PLLUSBHS0未稳定 1：PLLUSBHS0 已稳定
28	PLLUSBHS0EN	PLLUSBHS0时钟使能 软件置位或复位，当进入深度睡眠或待机模式时由硬件复位 0：PLLUSBHS0被关闭 1：PLLUSBHS0 被打开
27:21	保留	必须保持复位值。
20	LPIRC4MDSPEN	在深度睡眠模式下LPIRC4M时钟使能 由软件置位和复位。在深度睡眠模式下，可强制LPIRC4M作为部分外设的时钟。该位与LPIRC4M是否使能无关。 0：无作用 1：强制 LPIRC4M 在深度睡眠模式下运行
19:18	保留	必须保持复位值。
17:16	IRC64MDIV[1:0]	IRC64M时钟分频 由软件置位和复位，当系统时钟选择CK_IRC64MDIV或IRC16MEN位置1时无法写

入。
00：CK_IRC64MDIV = CK_IRC64M / 1
01：CK_IRC64MDIV = CK_IRC64M / 2
10：CK_IRC64MDIV = CK_IRC64M / 4
11：CK_IRC64MDIV = CK_IRC64M / 8

15:8	LPIRC4MCALIB[7:0]	LPIRC4M时钟校准值 上电时自动加载这些位，校准信号步长位0.4%
7:2	LPIRC4MADJ[5:0]	LPIRC4M 时钟调整值 这些位由软件置位，最终调整值为LPIRC4MADJ[7:0]位域的当前值加上LPIRC4MCALIB[8:0]位域的值。最终调整值应该调整LPIRC4M到4 MHz ± 1%
1	LPIRC4MSTB	LPIRC4M时钟稳定标志位 硬件置‘1’来指示LPIRC4M时钟是否稳定待用 0：LPIRC4M振荡器未稳定 1：LPIRC4M 振荡器已稳定
0	LPIRC4MEN	LPIRC4M RC振荡器使能 软件置位或复位，如果LPIRC4M时钟作为系统时钟时，该位不能被复位。如果DSPWUSSEL为1，当从深度睡眠或待机模式返回，该位由硬件置1来启动LPIRC4M振荡器。 0：LPIRC4M RC振荡器被关闭 1：LPIRC4M RC 振荡器被打开

6.1.43. 附加时钟中断寄存器（RCU_ADDINT）

地址偏移：0xCC
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留									IRC48MSTBIC	PLLUSBHS1STBIC	PLLUSBHS0STBIC	保留			
									w	w	w				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	IRC48MSTBIE	PLLUSBHS1STBIE	PLLUSBHS0STBIE	保留					IRC48MSTBIF	PLLUSBHS1STBIF	PLLUSBHS0STBIF	保留			
	rw	rw	rw						r	r	r				

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

		0：不复位 IRC48MSTBIF 标志位 1：复位 IRC48MSTBIF 标志位
21	PLLUSBHS1STBIC	USBHS1 内部 PLL 时钟稳定中断清零 软件写 1 复位 PLLUSBHS1STBIF 标志位 0：不复位 PLLUSBHS1STBIF 标志位 1：复位 PLLUSBHS1STBIF 标志位
20	PLLUSBHS0STBIC	USBHS0 内部 PLL 时钟稳定中断清零 软件写 1 复位 PLLUSBHS0STBIF 标志位 0：不复位 PLLUSBHS0STBIF 标志位 1：复位 PLLUSBHS0STBIF 标志位
19:15	保留	必须保持复位值。
14	IRC48MSTBIE	内部 48 MHz RC 振荡器稳定中断使能 由软件置位和复位来使能/禁止 IRC48M 时钟稳定中断 0：禁止IRC48M时钟稳定中断 1：使能 IRC48M 时钟稳定中断
13	PLLUSBHS1STBIE	USBHS1 内部 PLL 时钟稳定中断使能 由软件置位和复位来使能/禁止 USBHS1 内部 PLL 时钟稳定中断 0：禁止USBHS1内部PLL时钟稳定中断 1：使能 USBHS1 内部 PLL 时钟稳定中断
12	PLLUSBHS0STBIE	USBHS0 内部 PLL 时钟稳定中断使能 由软件置位和复位来使能/禁止 USBHS0 内部 PLL 时钟稳定中断 0：禁止USBHS0内部PLL时钟稳定中断 1：使能 USBHS0 内部 PLL 时钟稳定中断
11:7	保留	必须保持复位值。
6	IRC48MSTBIF	IRC48M 时钟稳定中断标志位 当内部48 MHz RC振荡器时钟稳定且IRC48MSTBIE位被置1时由硬件置1 软件置位 IRC48MSTBIC 位时清除该位 0：无IRC48M时钟稳定中断产生 1：产生 IRC48M 时钟稳定中断
5	PLLUSBHS1STBIF	USBHS1 内部 PLL 时钟稳定中断标志位 当USBHS1内部PLL时钟稳定且PLLUSBHS1STBIE位被置1时由硬件置1 软件置位 PLLUSBHS1STBIC 位时清除该位 0：无USBHS1内部PLL时钟稳定中断产生 1：产生 USBHS1 内部 PLL 时钟稳定中断
4	PLLUSBHS0STBIF	USBHS0 内部 PLL 时钟稳定中断标志位 当USBHS0内部PLL时钟稳定且PLLUSBHS0STBIE位被置1时由硬件置1 软件置位 PLLUSBHS0STBIC 位时清除该位 0：无USBHS0内部PLL时钟稳定中断产生 1：产生 USBHS0 内部 PLL 时钟稳定中断

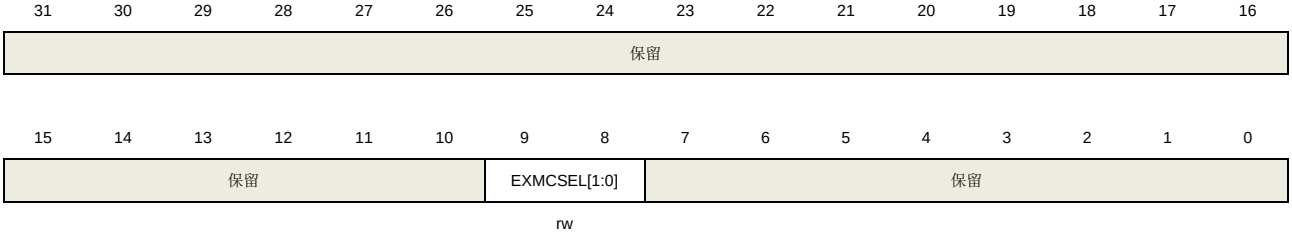
3:0	保留	必须保持复位值。

6.1.44. 时钟配置寄存器 4（RCU_CFG4）

地址偏移：0xD0

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



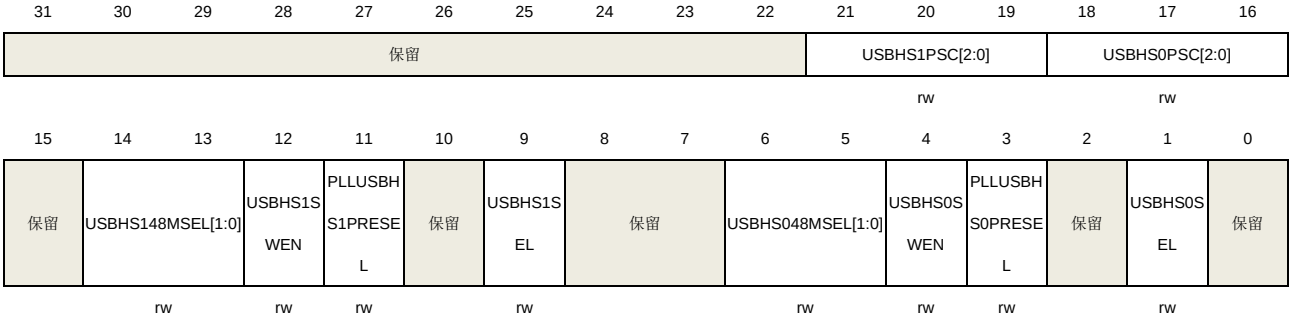
位/位域	名称	描述
31:10	保留	必须保持复位值。
9:8	EXMCSEL[1:0]	EXMC时钟源选择 由软件置位或复位，控制EXMC时钟源 00：CK_EXMC = CK_AHB 01：CK_EXMC = CK_PLL0Q 10：CK_EXMC = CK_PLL1R 11：CK_EXMC = CK_PER
7:0	保留	必须保持复位值。

6.1.45. USB 时钟控制寄存器（RCU_USBCLKCTL）

地址偏移：0xD4

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域	名称	描述
31:22	保留	必须保持复位值。

21:19	USBHS1PSC[2:0]	USBHS1的时钟分频系数 由软件置位或清零。USBHS1的时钟必须为48MHz，当USBHS1时钟使能的时候，这些位无法修改。 000：CK_USBHS1 = CK_PLL1Q / 1 001：CK_USBHS1 = CK_PLL1Q / 2 010：CK_USBHS1 = CK_PLL1Q / 3 011：CK_USBHS1 = CK_PLL1Q / 4 100：CK_USBHS1 = CK_PLL1Q / 5 101：CK_USBHS1 = CK_PLL1Q / 6 110：CK_USBHS1 = CK_PLL1Q / 7 111：CK_USBHS1 = CK_PLL1Q / 8
18:16	USBHS0PSC[2:0]	USBHS0的时钟分频系数 由软件置位或清零。USBHS0的时钟必须为48MHz，当USBHS0时钟使能的时候，这些位无法修改。 000：CK_USBHS0 = CK_PLL1Q / 1 001：CK_USBHS0 = CK_PLL1Q / 2 010：CK_USBHS0 = CK_PLL1Q / 3 011：CK_USBHS0 = CK_PLL1Q / 4 100：CK_USBHS0 = CK_PLL1Q / 5 101：CK_USBHS0 = CK_PLL1Q / 6 110：CK_USBHS0 = CK_PLL1Q / 7 111：CK_USBHS0 = CK_PLL1Q / 8
15	保留	必须保持复位值。
14:13	USBHS148MSEL[1:0]	USBHS1 48M时钟源选择 由软件置位或复位，控制USBHS1 48M时钟源 00：选择 CK_PLL0R 时钟作为 USBHS1 48M 源时钟 01：选择 CK_PLLUSBHS1/USBHS1DV 作为 USBHS1 48M 源时钟 10：选择 CK_PLL1Q/USBHS1PSC 作为 USBHS1 48M 源时钟 11：选择CK_IRC48M时钟作为USBHS1 48M源时钟
12	USBHS1SWEN	USBHS1 时钟源选择使能 0：通过 USBHS1 模块硬件选择 USBHS1 时钟 1：使用 USBHS1SEL 选择 USBHS1 时钟
11	PLLUSBHS1PRESEL	PLLUSBHS1 时钟源预选择 由软件置位和复位选择 PLLUSBHS1 时钟源。 0：选择 CK_HXTAL 时钟作为 PLLUSBHS1 时钟源 1：选择 CK_IRC48M 时钟作为 PLLUSBHS1 时钟源
10	保留	必须保持复位值。
9	USBHS1SEL	USBHS1 时钟源选择 由软件置位和复位。 0：选择 48M 时钟作为 USBHS1 时钟源

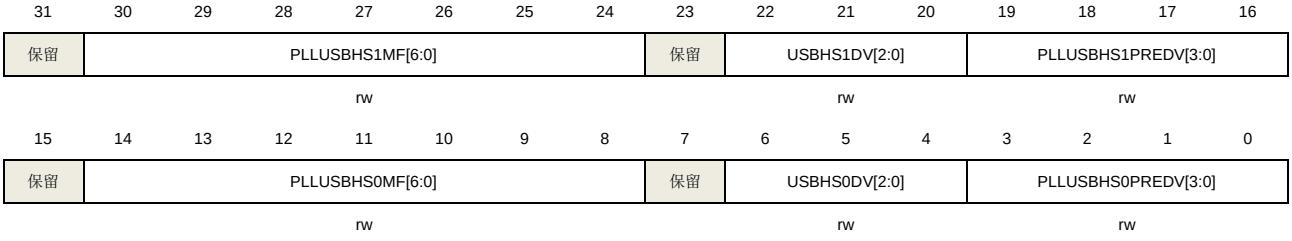
		1：选择 60M 时钟作为 USBHS1 时钟源
8:7	保留	必须保持复位值。
6:5	USBHS048MSEL[1:0]	USBHS0 48M时钟源选择 由软件置位或复位，控制USBHS0 48M时钟源 00：选择 CK_PLL0R 时钟作为 USBHS0 48M 源时钟 01：选择 CK_PLLUSBHS0/USBHS0DV 作为 USBHS0 48M 源时钟 10：选择 CK_PLL1Q/USBHS0PSC 作为 USBHS0 48M 源时钟 11：选择CK_IRC48M时钟作为USBHS0 48M源时钟
4	USBHS0SWEN	USBHS0 时钟源选择使能 0：通过 USBHS0 模块硬件选择 USBHS0 时钟 1：使用 USBHS0SEL 选择 USBHS0 时钟
3	PLLUSBHS0PRESEL	PLLUSBHS0 时钟源预选择 由软件置位和复位选择 PLLUSBHS0 时钟源。 0：选择 CK_HXTAL 时钟作为 PLLUSBHS0 时钟源 1：选择 CK_IRC48M 时钟作为 PLLUSBHS0 时钟源
2	保留	必须保持复位值。
1	USBHS0SEL	USBHS0 时钟源选择 由软件置位和复位。 0：选择 48M 时钟作为 USBHS0 时钟源 1：选择 60M 时钟作为 USBHS0 时钟源
0	保留	必须保持复位值。

6.1.46. PLLUSB 时钟配置寄存器（RCU_PLLUSBCFG）

地址偏移：0xD8

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	保留	必须保持复位值。
30:24	PLLUSBHS1MF[6:0]	PLLUSBHS1 时钟倍频因子 0000000：保留 0000001：保留

		… 0001111：保留 0010000：PLLUSBHS1MF输入源时钟16倍频 0010001：PLLUSBHS1MF输入源时钟17倍频 0010010：PLLUSBHS1MF输入源时钟18倍频 0010011：PLLUSBHS1MF输入源时钟19倍频 … 1111111：PLLUSBHS1MF输入源时钟127倍频 **注意：**PLLUSBHS1 输出时钟频率不能超过 480MHz
23	保留	必须保持复位值。
22:20	USBHS1DV[2:0]	USBHS1 时钟分频因子 由软件置位或清零。 000：USBHS1DV 输入源时钟 2 分频 001：USBHS1DV 输入源时钟 4 分频 010：USBHS1DV 输入源时钟 6 分频 … 111：USBHS1DV 输入源时钟 16 分频
19:16	PLLUSBHS1PREDV[3:0]	PLLUSBHS1PREDV 的时钟分频系数 由软件置位和复位。 0000：保留 0001：PLLUSBHS1PREDV 输入源时钟未分频 0010：PLLUSBHS1PREDV 输入源时钟 2 分频 … 1111：PLLUSBHS1PREDV 输入源时钟 15 分频
15	保留	必须保持复位值。
14:8	PLLUSBHS0MF[6:0]	PLLUSBHS0 时钟倍频因子 0000000：保留 0000001：保留 … 0001111：保留 0010000：PLLUSBHS0MF输入源时钟16倍频 0010001：PLLUSBHS0MF输入源时钟17倍频 0010010：PLLUSBHS0MF输入源时钟18倍频 0010011：PLLUSBHS0MF输入源时钟19倍频 … 1111111：PLLUSBHS0MF输入源时钟127倍频 **注意：**PLLUSBHS0输出时钟频率不能超过480MHz
7	保留	必须保持复位值。
6:4	USBHS0DV[2:0]	USBHS0 时钟分频因子 由软件置位或清零。 000：USBHS0DV 输入源时钟 2 分频

001：USBHS0DV 输入源时钟 4 分频
010：USBHS0DV 输入源时钟 6 分频
…
111：USBHS0DV 输入源时钟 16 分频

3:0	PLLUSBHS0PREDV[3:0]	PLLUSBHS0PREDV 的时钟分频系数 由软件置位和复位。 0000：保留 0001：PLLUSBHS0PREDV 输入源时钟未分频 0010：PLLUSBHS0PREDV 输入源时钟 2 分频 … 1111：PLLUSBHS0PREDV 输入源时钟 15 分频

6.1.47. APB2 附加复位寄存器（RCU_ADDAPB2RST）

地址偏移：0xE0
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													CAN2RST	CAN1RST	CAN0RST
													rw	rw	rw

位/位域	名称	描述
31:3	保留	必须保持复位值。
2	CAN2RST	CAN2 复位 由软件置位或复位 0：无作用 1：复位 CAN2
1	CAN1RST	CAN1 复位 由软件置位或复位 0：无作用 1：复位 CAN1
0	CAN0RST	CAN0 复位 由软件置位或复位 0：无作用 1：复位 CAN0

6.1.48. APB2 附加使能寄存器（RCU_ADDAPB2EN）

地址偏移：0xE4

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													CAN2EN	CAN1EN	CAN0EN
													rw	rw	rw

位/位域	名称	描述
31:3	保留	必须保持复位值。
2	CAN2EN	CAN2 时钟使能 由软件置位或复位 0：关闭 CNA2 时钟 1：开启 CAN2 时钟
1	CAN1EN	CAN1 时钟使能 由软件置位或复位 0：关闭 CNA1 时钟 1：开启 CAN1 时钟
0	CAN0EN	CAN0 时钟使能 由软件置位或复位 0：关闭 CNA0 时钟 1：开启 CAN0 时钟

6.1.49. APB2 附加睡眠模式使能寄存器（RCU_ADDAPB2SPEN）

地址偏移：0xE8

复位值：0x0000 0007

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													CAN2SPEN	CAN1SPEN	CAN0SPEN
													rw	rw	rw

位/位域	名称	描述

31:3	保留	必须保持复位值。
2	CAN2SPEN	睡眠模式下 CAN2 时钟使能 由软件置位或复位 0：睡眠模式下关闭 CAN2 时钟 1：睡眠模式下开启 CAN2 时钟
1	CAN1SPEN	睡眠模式下 CAN1 时钟使能 由软件置位或复位 0：睡眠模式下关闭 CAN1 时钟 1：睡眠模式下开启 CAN1 时钟
0	CAN0SPEN	睡眠模式下 CAN0 时钟使能 由软件置位或复位 0：睡眠模式下关闭 CAN0 时钟 1：睡眠模式下开启 CAN0 时钟

6.1.50. 时钟配置寄存器 5（RCU_CFG5）

地址偏移：0xF0

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									SPI5SEL[2:0]			保留	SPI4SEL[2:0]		
										rw				rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	SPI3SEL[2:0]			保留	SPI2SEL[2:0]			保留	SPI1SEL[2:0]			保留	SPI0SEL[2:0]		
	rw				rw				rw				rw		

位/位域	名称	描述
31:23	保留	必须保持复位值。
22:20	SPI5SEL[2:0]	SPI5 / I2S5时钟源选择 由软件置位或复位，控制SPI5 / I2S5时钟源 000：选择CK_APB2时钟作为SPI5 / I2S5源时钟 001：选择CK_PLL1Q时钟作为SPI5 / I2S5源时钟 010：选择CK_PLL2Q时钟作为SPI5 / I2S5源时钟 011：选择CK_IRC64MDIV时钟作为SPI5 / I2S5源时钟 100：选择CK_LPIRC4M时钟作为SPI5 / I2S5源时钟 101：选择CK_HXTAL时钟作为SPI5 / I2S5源时钟 110：选择I2S_CKIN时钟作为SPI5 / I2S5源时钟 111：保留
19	保留	必须保持复位值。
18:16	SPI4SEL[2:0]	SPI4时钟源选择

		由软件置位或复位，控制SPI4时钟源 000：选择CK_APB2时钟作为SPI4源时钟 001：选择CK_PLL1Q时钟作为SPI4源时钟 010：选择CK_PLL2Q时钟作为SPI4源时钟 011：选择CK_IRC64MDIV时钟作为SPI4源时钟 100：选择CK_LPIRC4M时钟作为SPI4源时钟 101：选择CK_HXTAL时钟作为SPI4源时钟 其它：保留
15	保留	必须保持复位值。
14:12	SPI3SEL[2:0]	SPI3时钟源选择 由软件置位或复位，控制SPI3时钟源 000：选择CK_APB2时钟作为SPI3源时钟 001：选择CK_PLL1Q时钟作为SPI3源时钟 010：选择CK_PLL2Q时钟作为SPI3源时钟 011：选择CK_IRC64MDIV时钟作为SPI3源时钟 100：选择CK_LPIRC4M时钟作为SPI3源时钟 101：选择CK_HXTAL时钟作为SPI3源时钟 其它：保留
11	保留	必须保持复位值。
10:8	SPI2SEL[2:0]	SPI2 / I2S2时钟源选择 由软件置位或复位，控制SPI2 / I2S2时钟源 000：选择CK_PLL0Q时钟作为SPI2 / I2S2源时钟 001：选择CK_PLL1P时钟作为SPI2 / I2S2源时钟 010：选择CK_PLL2P时钟作为SPI2 / I2S2源时钟 011：选择I2S_CKIN时钟作为SPI2 / I2S2源时钟 100：选择CK_PER时钟作为SPI2 / I2S2源时钟 其它：保留
7	保留	必须保持复位值。
6:4	SPI1SEL[2:0]	SPI1 / I2S1时钟源选择 由软件置位或复位，控制SPI1 / I2S1时钟源 000：选择CK_PLL0Q时钟作为SPI1 / I2S1源时钟 001：选择CK_PLL1P时钟作为SPI1 / I2S1源时钟 010：选择CK_PLL2P时钟作为SPI1 / I2S1源时钟 011：选择I2S_CKIN时钟作为SPI1 / I2S1源时钟 100：选择CK_PER时钟作为SPI1 / I2S1源时钟 其它：保留
3	保留	必须保持复位值。
2:0	SPI0SEL[2:0]	SPI0 / I2S0时钟源选择 由软件置位或复位，控制SPI0 / I2S0时钟源 000：选择CK_PLL0Q时钟作为SPI0 / I2S0源时钟

001：选择CK_PLL1P时钟作为SPI0 / I2S0源时钟
010：选择CK_PLL2P时钟作为SPI0 / I2S0源时钟
011：选择I2S_CKIN时钟作为SPI0 / I2S0源时钟
100：选择CK_PER时钟作为SPI0 / I2S0源时钟
其它：保留

7. 时钟校准控制器（CTC）

7.1. 简介

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

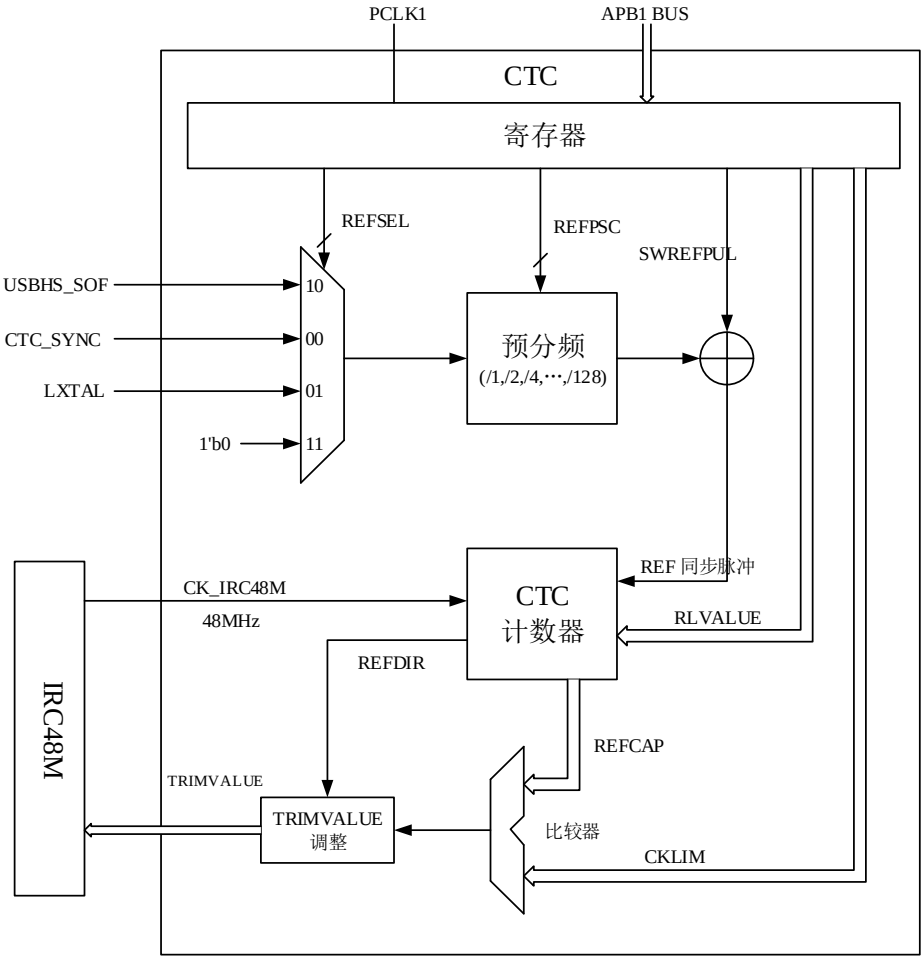
7.2. 主要特性

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

7.3. 功能描述

CTC模块的内部结构图如*图7-1. CTC简介*。

图 7-1. CTC 简介



7.3.1. REF 同步脉冲发生器

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

然后，可以通过设置CTC_CTL1寄存器中的REFPOL位来配置参考信号源同步时的信号极性，通过设置CTC_CTL1寄存器中的REFPSC位来产生一个合适的同步时钟频率信号。

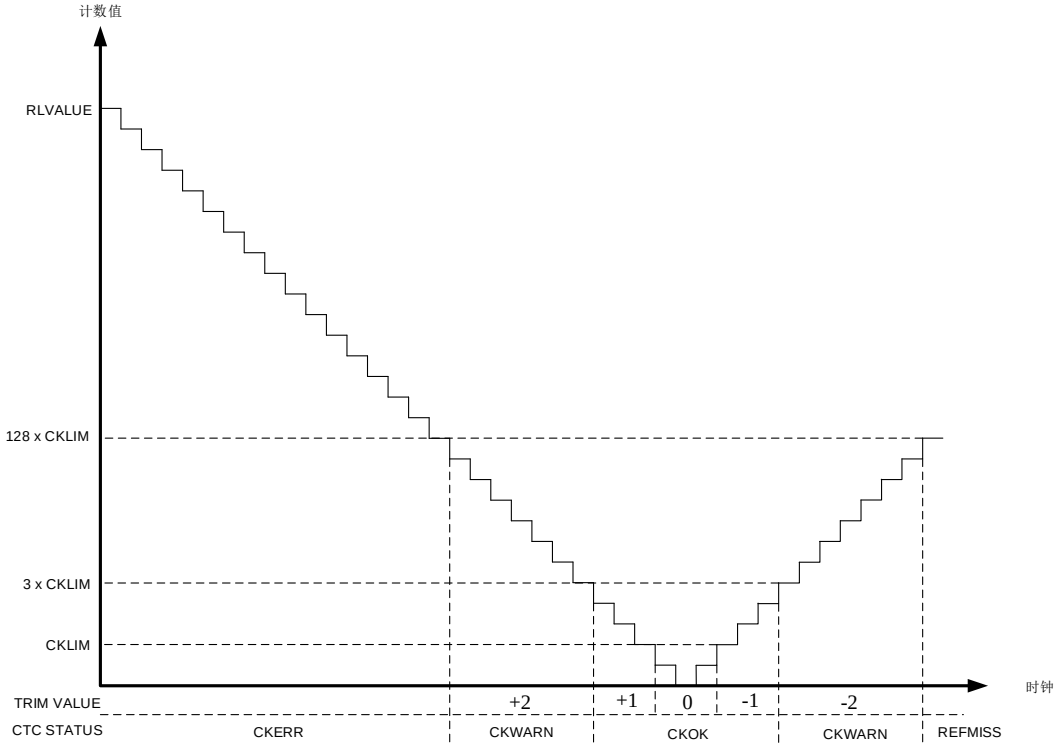
如果需要使用软件参考脉冲信号，则需要设置CTC_CTL0寄存器中的SWREFPUL位为1。软件参考脉冲信号与外部参考脉冲信号最后进行逻辑或操作。

7.3.2. CTC 校准计数器

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

存器中的REFCAP位，同时，当前计数器的计数方向被存入CTC_STAT寄存器中的REFDIR位。详细内容如*图7-2. CTC校准计数器*所示。

图 7-2. CTC 校准计数器



7.3.3. 频率评估和自动校准过程

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

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

- Counter < CKLIM时，检测到REF同步脉冲信号；

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

- CKLIM ≤ Counter < 3 x CKLIM时，检测到REF同步脉冲信号；

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

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

- 3 x CKLIM ≤ Counter < 128 x CKLIM时，检测到REF同步脉冲信号；

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

- Counter ≥ 128 x CKLIM，计数器在向下计数过程中，检测到REF同步脉冲信号；

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

- Counter = 128 x CKLIM，计数器在向上计数过程中；

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

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

7.3.4. 软件编程指南

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

$$RLVALUE=(F_{clock} \div F_{REF})-1 \quad (7\text{-}1)$$

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

$$CKLIM=(F_{clock} \div F_{REF}) \times 0.12\% \div 2 \quad (7\text{-}2)$$

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

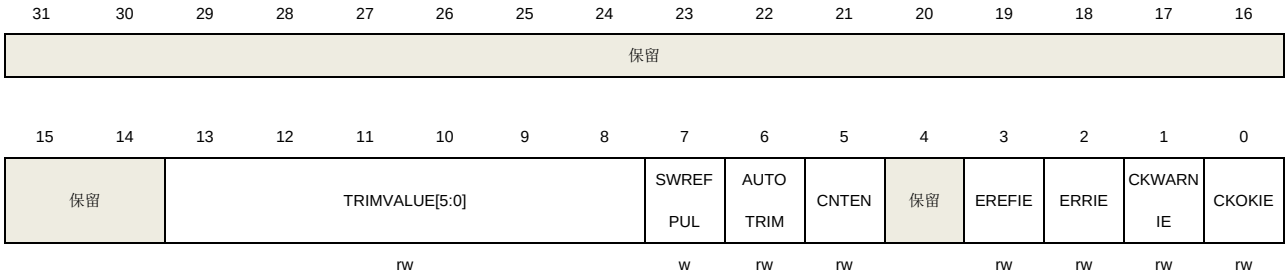
7.4. CTC 寄存器

CTC 基地址：0x4000 8400

7.4.1. 控制寄存器 0（CTC_CTL0）

地址偏移：0x00
复位值：0x0000 2000

该寄存器只能按字（32 位）访问。



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

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

7.4.2. 控制寄存器 1（CTC_CTL1）

地址偏移：0x04

复位值：0x2022 BB7F

该寄存器只能按字（32位）访问。当CNTEN为1时，不能修改该寄存器的值。

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

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

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

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

7.4.3. 状态寄存器（CTC_STAT）

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REFDIR	保留				TRIM ERR	REF MISS	CKERR	保留				EREFIF	ERRIF	CKWARN IF	CKOK IF
r					r	r	r					r	r	r	r

位/位域	名称	描述
31:16	REFCAP[15:0]	CTC 计数器捕获值 当检测到一个同步参考脉冲信号时，CTC 校准计数器中的计数值被存入到 REFCAP 位中。
15	REFDIR	CTC 校准时钟计数方向 当检测到一个同步参考脉冲信号时，CTC 校准计数器的计数方向被存入 REFDIR 位中。 0：向上计数 1：向下计数
14:11	保留	必须保持复位值。
10	TRIMERR	校准值错误位

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

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

7.4.4. 中断清除寄存器（CTC_INTC）

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



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

8. 中断/事件控制器（EXTI）

8.1. 简介

Cortex®-M7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®-M7技术参考手册》。

EXTI（中断/事件控制器）包括38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。

8.2. 主要特征

- Cortex®-M7系统异常；
- GD32H737_757_759支持217个可屏蔽外设中断，GD32H75E为215个；
- 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38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。

8.3. 功能说明

Arm Cortex®-M7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®-M7中的NVIC异常类型。

表 8-1. Cortex®-M7 中的 NVIC 异常类型

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

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

SysTick 校准值固定为 1000，SysTick 时钟源可选择为 CK_SYS 或 CK_SYS/8。通过对 SYST_RVR 寄存器进行配置，从而为系统提供 1ms 时基。当 SysTick 时钟源为 CK_SYS（CK_SYS=a MHz）时，SYST_RVR 寄存器值设置为（a*1000-1）。当 SysTick 时钟源为 CK_SYS/8 时，SYST_RVR 寄存器值设置为（a/8*1000-1）。

表 8-2. 中断向量表（适用于 GD32H737_757_759）

中断编号	向量编号	外设中断描述	向量地址
IRQ 0	16	窗口看门狗中断	0x0000_0040
IRQ 1	17	连接到 EXTI 线的 VAVD/LVD/VOVD 中断	0x0000_0044
IRQ 2	18	连接到 EXTI 线的 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
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

中断编号	向量编号	外设中断描述	向量地址
IRQ 25	41	TIMER0 更新中断	0x0000_00A4
IRQ 26	42	TIMER0 触发和换相中断	0x0000_00A8
IRQ 27	43	TIMER0 捕获比较中断	0x0000_00AC
IRQ 28	44	TIMER1 全局中断	0x0000_00B0
IRQ 29	45	TIMER2 全局中断	0x0000_00B4
IRQ 30	46	TIMER3 全局中断	0x0000_00B8
IRQ 31	47	I2C0 事件中断	0x0000_00BC
IRQ 32	48	I2C0 错误中断	0x0000_00C0
IRQ 33	49	I2C1 事件中断	0x0000_00C4
IRQ 34	50	I2C1 错误中断	0x0000_00C8
IRQ 35	51	SPI0 全局中断	0x0000_00CC
IRQ 36	52	SPI1 全局中断	0x0000_00D0
IRQ 37	53	USART0 全局和唤醒中断	0x0000_00D4
IRQ 38	54	USART1 全局和唤醒中断	0x0000_00D8
IRQ 39	55	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 线的 RTC 闹钟中断	0x0000_00E4
IRQ 42	58	保留	0x0000_00E8
IRQ 43	59	TIMER7 中止中断	0x0000_00EC
IRQ 44	60	TIMER7 更新中断	0x0000_00F0
IRQ 45	61	TIMER7 触发和换相中断	0x0000_00F4
IRQ 46	62	TIMER7 捕获比较中断	0x0000_00F8
IRQ 47	63	DMA0 通道 7 全局中断	0x0000_00FC
IRQ 48	64	EXMC 全局中断	0x0000_0100
IRQ 49	65	SDIO0 全局中断	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 DAC 下溢错误中断	0x0000_0118
IRQ 55	71	TIMER6 全局中断	0x0000_011C
IRQ 56	72	DMA1 通道 0 全局中断	0x0000_0120
IRQ 57	73	DMA1 通道 1 全局中断	0x0000_0124
IRQ 58	74	DMA1 通道 2 全局中断	0x0000_0128
IRQ 59	75	DMA1 通道 3 全局中断	0x0000_012C
IRQ 60	76	DMA1 通道 4 全局中断	0x0000_0130
IRQ 61	77	以太网 0 全局中断	0x0000_0134
IRQ 62	78	连接到 EXTI 线的以太网 0 唤醒中断	0x0000_0138
IRQ 63-67	79-83	保留	0x0000_013C- 0x0000_014C
IRQ 68	84	DMA1 通道 5 全局中断	0x0000_0150

中断编号	向量编号	外设中断描述	向量地址
IRQ 69	85	DMA1 通道 6 全局中断	0x0000_0154
IRQ 70	86	DMA1 通道 7 全局中断	0x0000_0158
IRQ 71	87	USART5 全局和唤醒中断	0x0000_015C
IRQ 72	88	I2C2 事件中断	0x0000_0160
IRQ 73	89	I2C2 错误中断	0x0000_0164
IRQ 74	90	USBHS0 端点 1 输出中断	0x0000_0168
IRQ 75	91	USBHS0 端点 1 输入中断	0x0000_016C
IRQ 76	92	连接到 EXTI 线的 USBHS0 唤醒中断	0x0000_0170
IRQ 77	93	USBHS0 全局中断	0x0000_0174
IRQ 78	94	DCI 全局中断	0x0000_0178
IRQ 79	95	CAU 全局中断	0x0000_017C
IRQ 80	96	HAU 和 TRNG 全局中断	0x0000_0180
IRQ 81	97	FPU 全局中断	0x0000_0184
IRQ 82	98	UART6 全局中断	0x0000_0188
IRQ 83	99	UART7 全局中断	0x0000_018C
IRQ 84	100	SPI3 全局中断	0x0000_0190
IRQ 85	101	SPI4 全局中断	0x0000_0194
IRQ 86	102	SPI5 全局中断	0x0000_0198
IRQ 87	103	SAI0 全局中断	0x0000_019C
IRQ 88	104	TLI 全局中断	0x0000_01A0
IRQ 89	105	TLI 错误中断	0x0000_01A4
IRQ 90	106	IPA 全局中断	0x0000_01A8
IRQ 91	107	SAI1 全局中断	0x0000_01AC
IRQ 92	108	OSPI0 全局中断	0x0000_01B0
IRQ 93-94	109-110	保留	0x0000_01B4-0x0000_01B8
IRQ 95	111	I2C3 事件中断	0x0000_01BC
IRQ 96	112	I2C3 错误中断	0x0000_01C0
IRQ 97	113	RSPDIF 全局中断	0x0000_01C4
IRQ 98-101	114-117	保留	0x0000_01C8-0x0000_01D4
IRQ 102	118	DMAMUX 上溢中断	0x0000_01D8
IRQ 103-109	119-125	保留	0x0000_01DC-0x0000_01F4
IRQ 110	126	HPDF 全局中断 0	0x0000_01F8
IRQ 111	127	HPDF 全局中断 1	0x0000_01FC
IRQ 112	128	HPDF 全局中断 2	0x0000_0200
IRQ 113	129	HPDF 全局中断 3	0x0000_0204
IRQ 114	130	SAI2 全局中断	0x0000_0208
IRQ 115	131	保留	0x0000_020C
IRQ 116	132	TIMER14 全局中断	0x0000_0210
IRQ 117	133	TIMER15 全局中断	0x0000_0214

中断编号	向量编号	外设中断描述	向量地址
IRQ 118	134	TIMER16 全局中断	0x0000_0218
IRQ 119	135	保留	0x0000_021C
IRQ 120	136	MDIO 全局中断	0x0000_0220
IRQ 121	137	保留	0x0000_0224
IRQ 122	138	MDMA 全局中断	0x0000_0228
IRQ 123	139	保留	0x0000_022C
IRQ 124	140	SDIO1 全局中断	0x0000_0230
IRQ 125	141	HWSEM 全局中断	0x0000_0234
IRQ 126	142	保留	0x0000_0238
IRQ 127	143	ADC2 全局中断	0x0000_023C
IRQ 128-136	144-152	保留	0x0000_0240-0x0000_0260
IRQ 137	153	CMP0 和 CMP1 全局中断 连接到 EXTI 线的 CMP0 和 CMP1 中断	0x0000_0264
IRQ 138-143	154-159	保留	0x0000_0268-0x0000_027C
IRQ 144	160	CTC 中断	0x0000_0280
IRQ 145	161	RAMECCMU 全局中断	0x0000_0284
IRQ 146-149	162-165	保留	0x0000_0288-0x0000_0294
IRQ 150	166	OSPI1 全局中断	0x0000_0298
IRQ 151	167	RTDEC0 全局中断	0x0000_029C
IRQ 152	168	RTDEC1 全局中断	0x0000_02A0
IRQ 153	169	FAC 全局中断	0x0000_02A4
IRQ 154	170	TMU 全局中断	0x0000_02A8
IRQ 155-160	171-176	保留	0x0000_02AC-0x0000_02C0
IRQ 161	177	TIMER22 全局中断	0x0000_02C4
IRQ 162	178	TIMER23 全局中断	0x0000_02C8
IRQ 163	179	TIMER30 全局中断	0x0000_02CC
IRQ 164	180	TIMER31 全局中断	0x0000_02D0
IRQ 165	181	TIMER40 全局中断	0x0000_02D4
IRQ 166	182	TIMER41 全局中断	0x0000_02D8
IRQ 167	183	TIMER42 全局中断	0x0000_02DC
IRQ 168	184	TIMER43 全局中断	0x0000_02E0
IRQ 169	185	TIMER44 全局中断	0x0000_02E4
IRQ 170	186	TIMER50 全局中断	0x0000_02E8
IRQ 171	187	TIMER51 全局中断	0x0000_02EC
IRQ 172	188	USBHS1 端点 1 输出中断	0x0000_02F0
IRQ 173	189	USBHS1 端点 1 输入中断	0x0000_02F4
IRQ 174	190	连接到 EXTI 线的 USBHS1 唤醒中断	0x0000_02F8
IRQ 175	191	USBHS1 全局中断	0x0000_02FC

中断编号	向量编号	外设中断描述	向量地址
IRQ 176	192	以太网 1 全局中断	0x0000_0300
IRQ 177	193	连接到 EXTI 线的以太网 1 唤醒中断	0x0000_0304
IRQ 178	194	保留	0x0000_0308
IRQ 179	195	连接到 EXTI 线的 CAN0 唤醒中断	0x0000_030C
IRQ 180	196	CAN0 消息缓冲区中断	0x0000_0310
IRQ 181	197	CAN0 总线关闭/总线关闭完成中断	0x0000_0314
IRQ 182	198	CAN0 错误中断	0x0000_0318
IRQ 183	199	CAN0 快速传输错误中断	0x0000_031C
IRQ 184	200	CAN0 发送警告中断	0x0000_0320
IRQ 185	201	CAN0 接收警告中断	0x0000_0324
IRQ 186	202	连接到 EXTI 线的 CAN1 唤醒中断	0x0000_0328
IRQ 187	203	CAN1 消息缓冲区中断	0x0000_032C
IRQ 188	204	CAN1 总线关闭/总线关闭完成中断	0x0000_0330
IRQ 189	205	CAN1 错误中断	0x0000_0334
IRQ 190	206	CAN1 快速传输错误中断	0x0000_0338
IRQ 191	207	CAN1 发送警告中断	0x0000_033C
IRQ 192	208	CAN1 接收警告中断	0x0000_0340
IRQ 193	209	连接到 EXTI 线的 CAN2 唤醒中断	0x0000_0344
IRQ 194	210	CAN2 消息缓冲区中断	0x0000_0348
IRQ 195	211	CAN2 总线关闭/总线关闭完成中断	0x0000_034C
IRQ 196	212	CAN2 错误中断	0x0000_0350
IRQ 197	213	CAN2 快速传输错误中断	0x0000_0354
IRQ 198	214	CAN2 发送警告中断	0x0000_0358
IRQ 199	215	CAN2 接收警告中断	0x0000_035C
IRQ 200	216	EFUSE 全局中断	0x0000_0360
IRQ 201	217	连接到 EXTI 线的 I2C0 唤醒中断	0x0000_0364
IRQ 202	218	连接到 EXTI 线的 I2C1 唤醒中断	0x0000_0368
IRQ 203	219	连接到 EXTI 线的 I2C2 唤醒中断	0x0000_036C
IRQ 204	220	连接到 EXTI 线的 I2C3 唤醒中断	0x0000_0370
IRQ 205	221	LPDTS 中断	0x0000_0374
IRQ 206	222	连接到 EXTI 线的 LPDTS 唤醒中断	0x0000_0378
IRQ 207	223	TIMER0 译码器检测中断	0x0000_037C
IRQ 208	224	TIMER7 译码器检测中断	0x0000_0380
IRQ 209	225	TIMER1 译码器检测中断	0x0000_0384
IRQ 210	226	TIMER2 译码器检测中断	0x0000_0388
IRQ 211	227	TIMER3 译码器检测中断	0x0000_038C
IRQ 212	228	TIMER4 译码器检测中断	0x0000_0390
IRQ 213	229	TIMER22 译码器检测中断	0x0000_0394
IRQ 214	230	TIMER23 译码器检测中断	0x0000_0398
IRQ 215	231	TIMER30 译码器检测中断	0x0000_039C
IRQ 216	232	TIMER31 译码器检测中断	0x0000_03A0

表 8-3. 中断向量表（适用于 GD32H75E）

中断编号	向量编号	外设中断描述	向量地址
IRQ 0	16	窗口看门狗中断	0x0000_0040
IRQ 1	17	连接到 EXTI 线的 VAVD/LVD/VOVD 中断	0x0000_0044
IRQ 2	18	连接到 EXTI 线的 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
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 触发和换相中断	0x0000_00A8
IRQ 27	43	TIMER0 捕获比较中断	0x0000_00AC
IRQ 28	44	TIMER1 全局中断	0x0000_00B0
IRQ 29	45	TIMER2 全局中断	0x0000_00B4
IRQ 30	46	TIMER3 全局中断	0x0000_00B8
IRQ 31	47	I2C0 事件中断	0x0000_00BC
IRQ 32	48	I2C0 错误中断	0x0000_00C0
IRQ 33	49	I2C1 事件中断	0x0000_00C4
IRQ 34	50	I2C1 错误中断	0x0000_00C8
IRQ 35	51	SPI0 全局中断	0x0000_00CC
IRQ 36	52	SPI1 全局中断	0x0000_00D0
IRQ 37	53	USART0 全局和唤醒中断	0x0000_00D4
IRQ 38	54	USART1 全局和唤醒中断	0x0000_00D8
IRQ 39	55	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 线的 RTC 闹钟中断	0x0000_00E4

中断编号	向量编号	外设中断描述	向量地址
IRQ 42	58	保留	0x0000_00E8
IRQ 43	59	TIMER7 中止中断	0x0000_00EC
IRQ 44	60	TIMER7 更新中断	0x0000_00F0
IRQ 45	61	TIMER7 触发和换相中断	0x0000_00F4
IRQ 46	62	TIMER7 捕获比较中断	0x0000_00F8
IRQ 47	63	DMA0 通道 7 全局中断	0x0000_00FC
IRQ 48	64	EXMC 全局中断	0x0000_0100
IRQ 49	65	保留	0x0000_0104
IRQ 50	66	TIMER4 全局中断	0x0000_0108
IRQ 51	67	SPI2 全局中断	0x0000_010C
IRQ 52	68	UART3 全局中断	0x0000_0110
IRQ 53	69	UART4 全局中断	0x0000_0114
IRQ 54	70	TIMER5 全局中断 DAC 下溢错误中断	0x0000_0118
IRQ 55	71	TIMER6 全局中断	0x0000_011C
IRQ 56	72	DMA1 通道 0 全局中断	0x0000_0120
IRQ 57	73	DMA1 通道 1 全局中断	0x0000_0124
IRQ 58	74	DMA1 通道 2 全局中断	0x0000_0128
IRQ 59	75	DMA1 通道 3 全局中断	0x0000_012C
IRQ 60	76	DMA1 通道 4 全局中断	0x0000_0130
IRQ 61-67	77-83	保留	0x0000_0134- 0x0000_014C
IRQ 68	84	DMA1 通道 5 全局中断	0x0000_0150
IRQ 69	85	DMA1 通道 6 全局中断	0x0000_0154
IRQ 70	86	DMA1 通道 7 全局中断	0x0000_0158
IRQ 71	87	USART5 全局和唤醒中断	0x0000_015C
IRQ 72	88	I2C2 事件中断	0x0000_0160
IRQ 73	89	I2C2 错误中断	0x0000_0164
IRQ 74	90	USBHS0 端点 1 输出中断	0x0000_0168
IRQ 75	91	USBHS0 端点 1 输入中断	0x0000_016C
IRQ 76	92	连接到 EXTI 线的 USBHS0 唤醒中断	0x0000_0170
IRQ 77	93	USBHS0 全局中断	0x0000_0174
IRQ 78-79	94-95	保留	0x0000_0178- 0x0000_017C
IRQ 80	96	TRNG 全局中断	0x0000_0180
IRQ 81	97	FPU 全局中断	0x0000_0184
IRQ 82	98	UART6 全局中断	0x0000_0188
IRQ 83	99	UART7 全局中断	0x0000_018C
IRQ 84	100	SPI3 全局中断	0x0000_0190
IRQ 85	101	SPI4 全局中断	0x0000_0194
IRQ 86	102	SPI5 全局中断	0x0000_0198

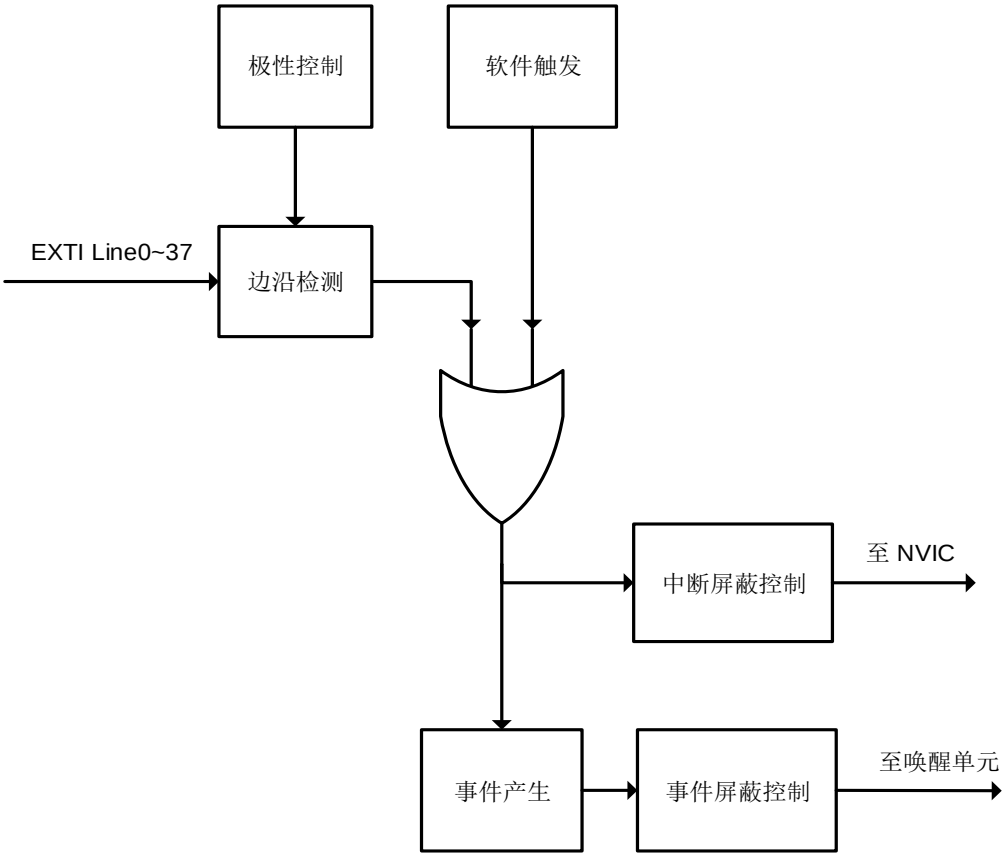
中断编号	向量编号	外设中断描述	向量地址
IRQ 87-91	103-107	保留	0x0000_019C-0x0000_01AC
IRQ 92	108	OSPI0 全局中断	0x0000_01B0
IRQ 93-94	109-110	保留	0x0000_01B4-0x0000_01B8
IRQ 95	111	I2C3 事件中断	0x0000_01BC
IRQ 96	112	I2C3 错误中断	0x0000_01C0
IRQ 97-101	113-117	保留	0x0000_01C4-0x0000_01D4
IRQ 102	118	DMAMUX 上溢中断	0x0000_01D8
IRQ 103-109	119-125	保留	0x0000_01DC-0x0000_01F4
IRQ 110	126	HPDF 全局中断 0	0x0000_01F8
IRQ 111	127	HPDF 全局中断 1	0x0000_01FC
IRQ 112	128	HPDF 全局中断 2	0x0000_0200
IRQ 113	129	HPDF 全局中断 3	0x0000_0204
IRQ 114-115	130-131	保留	0x0000_0208-0x0000_020C
IRQ 116	132	TIMER14 全局中断	0x0000_0210
IRQ 117	133	TIMER15 全局中断	0x0000_0214
IRQ 118	134	TIMER16 全局中断	0x0000_0218
IRQ 119-121	135-137	保留	0x0000_021C-0x0000_0224
IRQ 122	138	MDMA 全局中断	0x0000_0228
IRQ 123-126	139-142	保留	0x0000_022C-0x0000_0238
IRQ 127	143	ADC2 全局中断	0x0000_023C
IRQ 128-136	144-152	保留	0x0000_0240-0x0000_0260
IRQ 137	153	CMP0 和 CMP1 全局中断 连接到 EXTI 线的 CMP0 和 CMP1 中断	0x0000_0264
IRQ 138-143	154-159	保留	0x0000_0268-0x0000_027C
IRQ 144	160	CTC 中断	0x0000_0280
IRQ 145	161	RAMECCMU 全局中断	0x0000_0284
IRQ 146-149	162-165	保留	0x0000_0288-0x0000_0294
IRQ 150	166	OSPI1 全局中断	0x0000_0298
IRQ 151-152	167-168	保留	0x0000_029C-0x0000_02A0
IRQ 153	169	FAC 全局中断	0x0000_02A4
IRQ 154	170	TMU 全局中断	0x0000_02A8

中断编号	向量编号	外设中断描述	向量地址
IRQ 155-160	171-176	保留	0x0000_02AC-0x0000_02C0
IRQ 161	177	TIMER22 全局中断	0x0000_02C4
IRQ 162	178	TIMER23 全局中断	0x0000_02C8
IRQ 163-164	179-180	保留	0x0000_02CC-0x0000_02D0
IRQ 165	181	TIMER40 全局中断	0x0000_02D4
IRQ 166	182	TIMER41 全局中断	0x0000_02D8
IRQ 167	183	TIMER42 全局中断	0x0000_02DC
IRQ 168	184	TIMER43 全局中断	0x0000_02E0
IRQ 169	185	TIMER44 全局中断	0x0000_02E4
IRQ 170	186	TIMER50 全局中断	0x0000_02E8
IRQ 171	187	TIMER51 全局中断	0x0000_02EC
IRQ 172	188	USBHS1 端点 1 输出中断	0x0000_02F0
IRQ 173	189	USBHS1 端点 1 输入中断	0x0000_02F4
IRQ 174	190	连接到 EXTI 线的 USBHS1 唤醒中断	0x0000_02F8
IRQ 175	191	USBHS1 全局中断	0x0000_02FC
IRQ 176-178	192-194	保留	0x0000_0300-0x0000_0308
IRQ 179	195	连接到 EXTI 线的 CAN0 唤醒中断	0x0000_030C
IRQ 180	196	CAN0 消息缓冲区中断	0x0000_0310
IRQ 181	197	CAN0 总线关闭/总线关闭完成中断	0x0000_0314
IRQ 182	198	CAN0 错误中断	0x0000_0318
IRQ 183	199	CAN0 快速传输错误中断	0x0000_031C
IRQ 184	200	CAN0 发送警告中断	0x0000_0320
IRQ 185	201	CAN0 接收警告中断	0x0000_0324
IRQ 186	202	连接到 EXTI 线的 CAN1 唤醒中断	0x0000_0328
IRQ 187	203	CAN1 消息缓冲区中断	0x0000_032C
IRQ 188	204	CAN1 总线关闭/总线关闭完成中断	0x0000_0330
IRQ 189	205	CAN1 错误中断	0x0000_0334
IRQ 190	206	CAN1 快速传输错误中断	0x0000_0338
IRQ 191	207	CAN1 发送警告中断	0x0000_033C
IRQ 192	208	CAN1 接收警告中断	0x0000_0340
IRQ 193	209	连接到 EXTI 线的 CAN2 唤醒中断	0x0000_0344
IRQ 194	210	CAN2 消息缓冲区中断	0x0000_0348
IRQ 195	211	CAN2 总线关闭/总线关闭完成中断	0x0000_034C
IRQ 196	212	CAN2 错误中断	0x0000_0350
IRQ 197	213	CAN2 快速传输错误中断	0x0000_0354
IRQ 198	214	CAN2 发送警告中断	0x0000_0358
IRQ 199	215	CAN2 接收警告中断	0x0000_035C
IRQ 200	216	EFUSE 全局中断	0x0000_0360
IRQ 201	217	连接到 EXTI 线的 I2C0 唤醒中断	0x0000_0364

中断编号	向量编号	外设中断描述	向量地址
IRQ 202	218	连接到 EXTI 线的 I2C1 唤醒中断	0x0000_0368
IRQ 203	219	连接到 EXTI 线的 I2C2 唤醒中断	0x0000_036C
IRQ 204	220	连接到 EXTI 线的 I2C3 唤醒中断	0x0000_0370
IRQ 205	221	LPDTS 中断	0x0000_0374
IRQ 206	222	连接到 EXTI 线的 LPDTS 唤醒中断	0x0000_0378
IRQ 207	223	TIMER0 译码器检测中断	0x0000_037C
IRQ 208	224	TIMER7 译码器检测中断	0x0000_0380
IRQ 209	225	TIMER1 译码器检测中断	0x0000_0384
IRQ 210	226	TIMER2 译码器检测中断	0x0000_0388
IRQ 211	227	TIMER3 译码器检测中断	0x0000_038C
IRQ 212	228	TIMER4 译码器检测中断	0x0000_0390
IRQ 213	229	TIMER22 译码器检测中断	0x0000_0394
IRQ 214	230	TIMER23 译码器检测中断	0x0000_0398

8.4. 外部中断及事件（EXTI）框图

图 8-1. EXTI 框图



8.5. 外部中断及事件功能概述

EXTI 包含多达 38 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 22 根线，具体细节参考*表 8-4. EXTI 触发源（适用于 GD32H737_757_759）*和*表 8-5. EXTI 触发源（适用于 GD32H75E）*。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*系统配置寄存器（SYSCFG）*。

除了中断，EXTI 还可以向处理器提供事件信号。Cortex®-M7 内核完全支持等待中断（WFI），等待事件（WFE）和发送事件（SEV）指令。芯片内部有一个唤醒中断控制器（WIC），用户可以放心的让处理器和 NVIC 进入功耗极低的省电模式，由 WIC 来识别中断和事件以及判断优先级。当某些预期的事件发生时，例如一个特定的 I/O 管脚电平翻转或者 RTC 闹钟动作，EXTI 能唤醒处理器及整个系统。

硬件触发

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

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

软件触发

按照如下步骤软件也可以触发 EXTI 中断或事件：

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

表 8-4. EXTI 触发源（适用于 GD32H737_757_759）

EXTI 线编号	触发源
0	PA0 / PB0 / PC0 / PD0 / PE0 / PF0 / PG0 / PH0 / PK0
1	PA1 / PB1 / PC1 / PD1 / PE1 / PF1 / PG1 / PH1 / PK1
2	PA2 / PB2 / PC2 / PD2 / PE2 / PF2 / PG2 / PH2 / PK2
3	PA3 / PB3 / PC3 / PD3 / PE3 / PF3 / PG3 / PH3
4	PA4 / PB4 / PC4 / PD4 / PE4 / PF4 / PG4 / PH4
5	PA5 / PB5 / PC5 / PD5 / PE5 / PF5 / PG5 / PH5
6	PA6 / PB6 / PC6 / PD6 / PE6 / PF6 / PG6 / PH6
7	PA7 / PB7 / PC7 / PD7 / PE7 / PF7 / PG7 / PH7
8	PA8 / PB8 / PC8 / PD8 / PE8 / PF8 / PG8 / PH8 / PJ8
9	PA9 / PB9 / PC9 / PD9 / PE9 / PF9 / PG9 / PH9 / PJ9
10	PA10 / PB10 / PC10 / PD10 / PE10 / PF10 / PG10 / PH10 / PJ10
11	PB11 / PC11 / PD11 / PE11 / PF11 / PG11 / PH11 / PJ11
12	PB12 / PC12 / PD12 / PE12 / PF12 / PG12 / PH12
13	PA13 / PB13 / PC13 / PD13 / PE13 / PF13 / PG13 / PH13
14	PA14 / PB14 / PC14 / PD14 / PE14 / PF14 / PG14 / PH14
15	PA15 / PB15 / PC15 / PD15 / PE15 / PF15 / PG15 / PH15
16	VAVD，LVD 和 VOVD
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

EXTI 线编号	触发源
22	以太网 1 唤醒
23	以太网 0 唤醒
24	CAN0 唤醒
25	CAN1 唤醒
26	CAN2 唤醒
27	USART0 唤醒
28	USART1 唤醒
29	USART2 唤醒
30	USART5 唤醒
31	USBHS0 唤醒
32	USBHS1 唤醒
33	I2C0 唤醒
34	I2C1 唤醒
35	I2C2 唤醒
36	I2C3 唤醒
37	LPDTS 唤醒

表 8-5. EXTI 触发源（适用于 GD32H75E）

EXTI 线编号	触发源
0	PA0 / PB0 / PC0 / PD0 / PE0 / PH0
1	PA1 / PB1 / PC1 / PD1 / PE1 / PH1
2	PA2 / PB2 / PC2 / PD2 / PE2 / PH2
3	PA3 / PB3 / PC3 / PD3 / PE3 / PH3
4	PA4 / PB4 / PC4 / PD4 / PE4 / PH4
5	PA5 / PB5 / PC5 / PD5 / PE5 / PF5 / PH5
6	PA6 / PB6 / PC6 / PD6 / PE6 / PF6 / PG6 / PH6
7	PA7 / PB7 / PC7 / PD7 / PE7 / PF7 / PG7 / PH7
8	PA8 / PB8 / PC8 / PD8 / PE8 / PF8 / PG8 / PH8
9	PA9 / PB9 / PC9 / PD9 / PE9 / PF9 / PG9 / PH9
10	PA10 / PB10 / PC10 / PD10 / PE10 / PF10 / PH10
11	PA11 / PB11 / PC11 / PD11 / PE11 / PF11 / PH11
12	PA12 / PB12 / PC12 / PD12 / PE12 / PF12 / PH12
13	PA13 / PB13 / PC13 / PD13 / PE13 / PF13 / PG13 / PH13
14	PA14 / PB14 / PC14 / PD14 / PE14 / PF14 / PG14
15	PA15 / PB15 / PC15 / PD15 / PE15 / PF15 / PG15
16	VAVD，LVD 和 VOVD
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	保留

EXTI 线编号	触发源
23	保留
24	CAN0 唤醒
25	CAN1 唤醒
26	CAN2 唤醒
27	USART0 唤醒
28	USART1 唤醒
29	USART2 唤醒
30	USART5 唤醒
31	USBHS0 唤醒
32	USBHS1 唤醒
33	I2C0 唤醒
34	I2C1 唤醒
35	I2C2 唤醒
36	I2C3 唤醒
37	LPDTS 唤醒

8.6. EXTI 寄存器

EXTI 基地址：0x5800 0000

8.6.1. 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

8.6.2. 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

8.6.3. 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	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
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）

8.6.4. 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	FTEN22	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）

8.6.5. 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

8.6.6. 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	PD22	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。

8.6.7. 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										INTEN37	INTEN36	INTEN35	INTEN34	INTEN33	INTEN32
										rw	rw	rw	rw	rw	rw

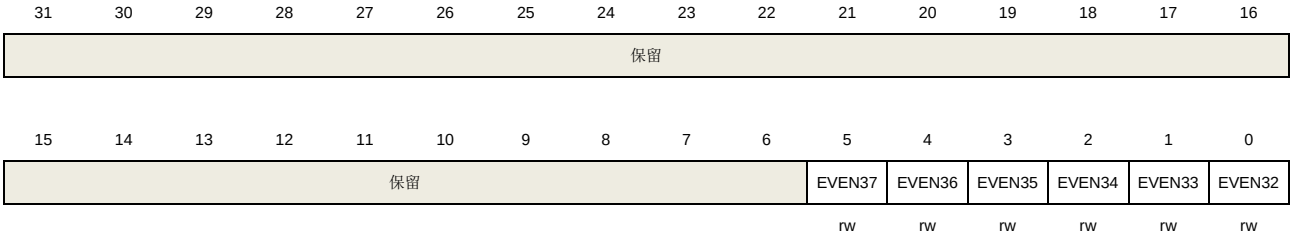
位/位域	名称	描述
31:6	保留	必须保持复位值。
5:0	INTENx	中断使能位 x（x = 32…37） 0：第 x 线中断被禁用 1：第 x 线中断被使能

8.6.8. 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。



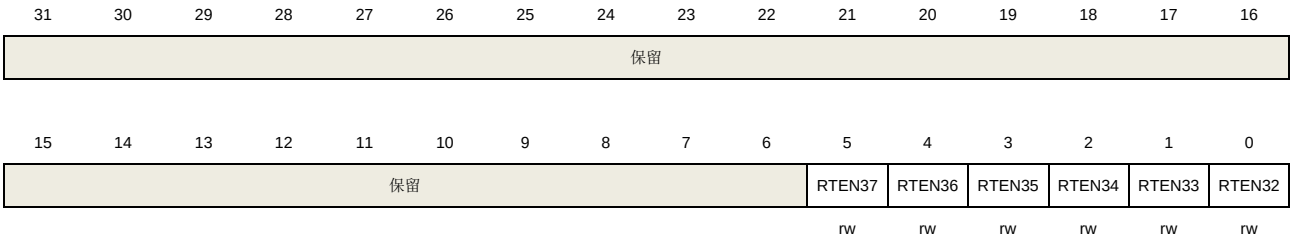
位/位域	名称	描述
31:6	保留	必须保持复位值。
5:0	EVENx	事件使能位 x（x = 32…37） 0：第 x 线事件被禁用 1：第 x 线事件被使能

8.6.9. 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。



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…37） 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）

8.6.10. 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										FTEN37	FTEN36	FTEN35	FTEN34	FTEN33	FTEN32
										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…37） 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）

8.6.11. 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										SWIEV37	SWIEV36	SWIEV35	SWIEV34	SWIEV33	SWIEV32
										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…37） 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

8.6.12. 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										PD37	PD36	PD35	PD34	PD33	PD32

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…37） 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。

9. 触发选择控制器（TRIGSEL）

9.1. 简介

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

使用TRIGSEL，有多达150个触发输入信号可选。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。

9.2. 主要特征

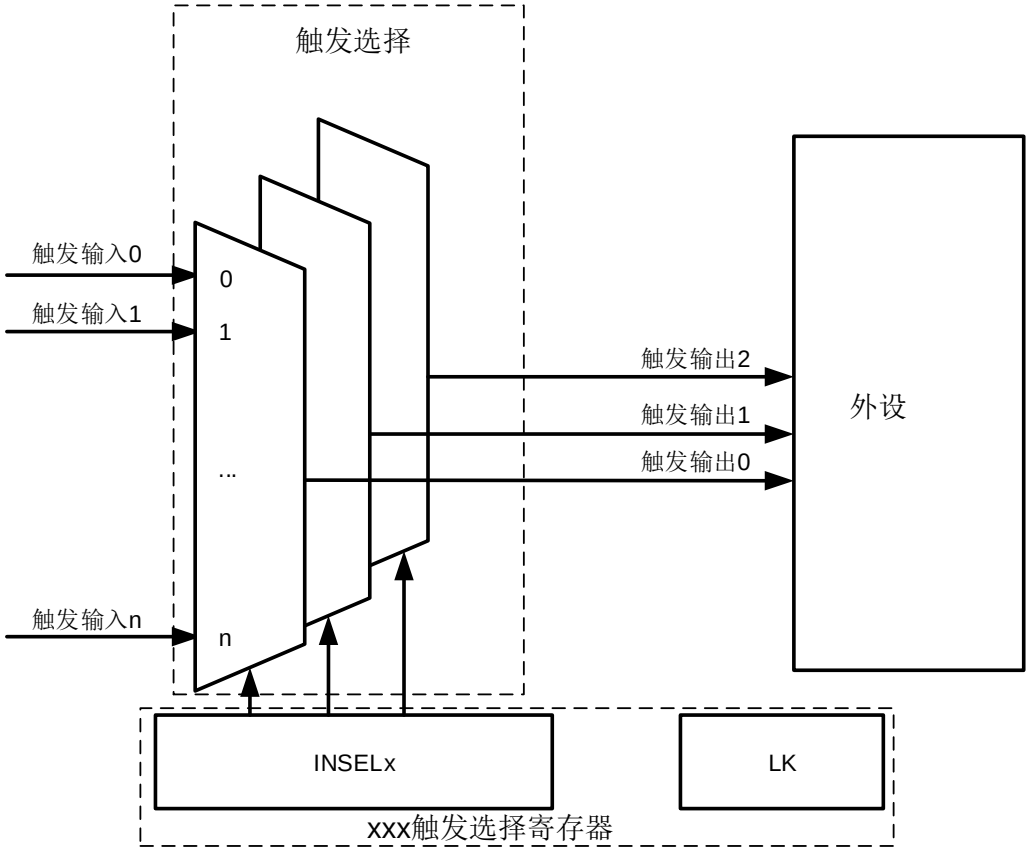
- 支持多达150个触发输入信号；
- 每个外设都有专用的触发信号选择寄存器；
- 触发选择控制器的输入信号可来源于外部输输入或外设输出；
- 触发选择控制器的输出信号可输出到外部输出或者到外设输入。

9.3. 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9-1. TRIGSEL主要组成示例*显示了TRIGSEL的主要组成结构。

图 9-1. TRIGSEL 主要组成示例



9.4. 内部连接

TRIGSEL 允许软件方式为外设选择触发输入。*表 9-1. 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 9-1. 触发输入位域选择

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

位域名称	位域值	触发输入选择
	0x0e	TRIGSEL_IN12
	0x0f	TRIGSEL_IN13
	0x10	LXTAL_TRG
	0x11	TIMER0_TRGO0
	0x12	TIMER0_TRGO1
	0x13	TIMER0_CH0
	0x14	TIMER0_CH1
	0x15	TIMER0_CH2
	0x16	TIMER0_CH3
	0x17	TIMER0_MCH0
	0x18	TIMER0_MCH1
	0x19	TIMER0_MCH2
	0x1a	TIMER0_MCH3
	0x1b~0x20	保留
	0x21	TIMER0_BRKIN0
	0x22	TIMER0_BRKIN1
	0x23	TIMER0_BRKIN2
	0x24	TIMER0_ETI
	0x25	TIMER1_TRGO0
	0x26	TIMER1_CH0
	0x27	TIMER1_CH1
	0x28	TIMER1_CH2
	0x29	TIMER1_CH3
	0x2a	TIMER1_ETI
	0x2b	TIMER2_TRGO0
	0x2c	TIMER2_CH0
	0x2d	TIMER2_CH1
	0x2e	TIMER2_CH2
	0x2f	TIMER2_CH3
	0x30	TIMER2_ETI
	0x31	TIMER3_TRGO0
	0x32	TIMER3_CH0
	0x33	TIMER3_CH1
	0x34	TIMER3_CH2
	0x35	TIMER3_CH3
	0x36	TIMER3_ETI
	0x37	TIMER4_TRGO0
	0x38	TIMER4_CH0
	0x39	TIMER4_CH1
	0x3a	TIMER4_CH2
	0x3b	TIMER4_CH3
	0x3c	TIMER4_ETI

位域名称	位域值	触发输入选择
	0x3d	TIMER5_TRGO0
	0x3e	TIMER6_TRGO0
	0x3f	TIMER7_TRGO0
	0x40	TIMER7_TRGO1
	0x41	TIMER7_CH0
	0x42	TIMER7_CH1
	0x43	TIMER7_CH2
	0x44	TIMER7_CH3
	0x45	TIMER7_MCH0
	0x46	TIMER7_MCH1
	0x47	TIMER7_MCH2
	0x48	TIMER7_MCH3
	0x49~0x4e	保留
	0x4f	TIMER7_BRKIN0
	0x50	TIMER7_BRKIN1
	0x51	TIMER7_BRKIN2
	0x52	TIMER7_ETI
	0x53	TIMER14_TRGO0
	0x54	TIMER14_CH0
	0x55	TIMER14_CH1
	0x56	TIMER14_MCH0
	0x57~0x58	保留
	0x59	TIMER14_BRKIN0
	0x5a	TIMER15_CH0
	0x5b	TIMER15_MCH0
	0x5c~0x5d	保留
	0x5e	TIMER15_BRKIN0
	0x5f	TIMER16_CH0
	0x60	TIMER16_MCH0
	0x61~0x62	保留
	0x63	TIMER16_BRKIN0
	0x64	TIMER22_TRGO0
	0x65	TIMER22_CH0
	0x66	TIMER22_CH1
	0x67	TIMER22_CH2
	0x68	TIMER22_CH3
	0x69	TIMER22_ETI
	0x6a	TIMER23_TRGO0
	0x6b	TIMER23_CH0
	0x6c	TIMER23_CH1
	0x6d	TIMER23_CH2
	0x6e	TIMER23_CH3

位域名称	位域值	触发输入选择
	0x6f	TIMER23_ETI
	0x70	TIMER30_TRGO0
	0x71	TIMER30_CH0
	0x72	TIMER30_CH1
	0x73	TIMER30_CH2
	0x74	TIMER30_CH3
	0x75	TIMER30_ETI
	0x76	TIMER31_TRGO0
	0x77	TIMER31_CH0
	0x78	TIMER31_CH1
	0x79	TIMER31_CH2
	0x7a	TIMER31_CH3
	0x7b	TIMER31_ETI
	0x7c	TIMER40_TRGO0
	0x7d	TIMER40_CH0
	0x7e	TIMER40_CH1
	0x7f	TIMER40_MCH0
	0x80~0x81	保留
	0x82	TIMER40_BRKIN0
	0x83	TIMER41_TRGO0
	0x84	TIMER41_CH0
	0x85	TIMER41_CH1
	0x86	TIMER41_MCH0
	0x87~0x88	保留
	0x89	TIMER41_BRKIN0
	0x8a	TIMER42_TRGO0
	0x8b	TIMER42_CH0
	0x8c	TIMER42_CH1
	0x8d	TIMER42_MCH0
	0x8e~0x8f	保留
	0x90	TIMER42_BRKIN0
	0x91	TIMER43_TRGO0
	0x92	TIMER43_CH0
	0x93	TIMER43_CH1
	0x94	TIMER43_MCH0
	0x95~0x96	保留
	0x97	TIMER43_BRKIN0
	0x98	TIMER44_TRGO0
	0x99	TIMER44_CH0
	0x9a	TIMER44_CH1
	0x9b	TIMER44_MCH0
	0x9c~0x9d	保留

位域名称	位域值	触发输入选择
	0x9e	TIMER44_BRKIN0
	0x9f	TIMER50_TRGO0
	0xa0	TIMER51_TRGO0
	0xa1	RTC_Alarm
	0xa2	RTC_TPTS
	0xa3	ADC0_WD0_OUT
	0xa4	ADC0_WD1_OUT
	0xa5	ADC0_WD2_OUT
	0xa6	ADC1_WD0_OUT
	0xa7	ADC1_WD1_OUT
	0xa8	ADC1_WD2_OUT
	0xa9	ADC2_WD0_OUT
	0xaa	ADC2_WD1_OUT
	0xab	ADC2_WD2_OUT
	0xac	CMP0_OUT
	0xad	CMP1_OUT
	0xae	SAI0_FS0
	0xaf	SAI0_FS1
	0xb0	SAI2_FS0
	0xb1	SAI2_FS1
	0xb2~0xff	保留

如*表 9-2. 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 9-2. 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	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

触发源	触发选择	TRIGSEL 寄存器	TRIGSEL 输出	外设
TRIGSEL_IN10 TRIGSEL_IN11 TRIGSEL_IN12 TRIGSEL_IN13 LXTAL_TRG TIMER0_TRGO0 TIMER0_TRGO1 TIMER0_CH0 TIMER0_CH1 TIMER0_CH2 TIMER0_CH3 TIMER0_MCH0 TIMER0_MCH1 TIMER0_MCH2 TIMER0_MCH3 TIMER0_BRKIN0 TIMER0_BRKIN1 TIMER0_BRKIN2 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		TRIGSEL_EXTOUT_3	Output0 Output1	TRIGSEL_OUT6 TRIGSEL_OUT7
		TRIGSEL_ADC0	Output0	ADC0_ROUTRG
		TRIGSEL_ADC1	Output0	ADC1_ROUTRG
		TRIGSEL_ADC2	Output0	ADC2_ROUTRG
		TRIGSEL_DAC0OUT0	Output0	DAC0_OUT0_EXTRG
		TRIGSEL_DAC0OUT1	Output0	DAC0_OUT1_EXTRG
		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

触发源	触发选择	TRIGSEL 寄存器	TRIGSEL 输出	外设
TIMER4_CH2 TIMER4_CH3 TIMER4_ETI TIMER5_TRGO0 TIMER6_TRGO0 TIMER7_TRGO0 TIMER7_TRGO1 TIMER7_CH0 TIMER7_CH1 TIMER7_CH2 TIMER7_CH3 TIMER7_MCH0 TIMER7_MCH1 TIMER7_MCH2 TIMER7_MCH3 TIMER7_BRKIN0 TIMER7_BRKIN1 TIMER7_BRKIN2 TIMER7_ETI TIMER14_TRGO0 TIMER14_CH0 TIMER14_CH1 TIMER14_MCH0 TIMER14_BRKIN0 TIMER15_CH0 TIMER15_MCH0 TIMER15_BRKIN0 TIMER16_CH0 TIMER16_MCH0 TIMER16_BRKIN0 TIMER22_TRGO0 TIMER22_CH0 TIMER22_CH1 TIMER22_CH2 TIMER22_CH3 TIMER22_ETI TIMER23_TRGO0 TIMER23_CH0 TIMER23_CH1 TIMER23_CH2		TRIGSEL_TIMER16BRKIN	Output0	TIMER16_BRKIN0
		TRIGSEL_TIMER40BRKIN	Output0	TIMER40_BRKIN0
		TRIGSEL_TIMER41BRKIN	Output0	TIMER41_BRKIN0
		TRIGSEL_TIMER42BRKIN	Output0	TIMER42_BRKIN0
		TRIGSEL_TIMER43BRKIN	Output0	TIMER43_BRKIN0
		TRIGSEL_TIMER44BRKIN	Output0	TIMER44_BRKIN0
		TRIGSEL_CAN0	Output0	CAN0_EX_TIME_TICK
		TRIGSEL_CAN1	Output0	CAN1_EX_TIME_TICK
		TRIGSEL_CAN2	Output0	CAN2_EX_TIME_TICK
		TRIGSEL_LPDTS	Output0	LPDTS_TRG

触发源	触发选择	TRIGSEL 寄存器	TRIGSEL 输出	外设
TIMER23_CH3 TIMER23_ETI TIMER30_TRGO0 TIMER30_CH0 TIMER30_CH1 TIMER30_CH2 TIMER30_CH3 TIMER30_ETI TIMER31_TRGO0 TIMER31_CH0 TIMER31_CH1 TIMER31_CH2 TIMER31_CH3 TIMER31_ETI TIMER40_TRGO0 TIMER40_CH0 TIMER40_CH1 TIMER40_MCH0 TIMER40_BRKIN0 TIMER41_TRGO0 TIMER41_CH0 TIMER41_CH1 TIMER41_MCH0 TIMER41_BRKIN0 TIMER42_TRGO0 TIMER42_CH0 TIMER42_CH1 TIMER42_MCH0 TIMER42_BRKIN0 TIMER43_TRGO0 TIMER43_CH0 TIMER43_CH1 TIMER43_MCH0 TIMER43_BRKIN0 TIMER44_TRGO0 TIMER44_CH0 TIMER44_CH1 TIMER44_MCH0 TIMER44_BRKIN0 TIMER50_TRGO0		TRIGSEL_EDOUT	Output0	EDOUT_TRG
		TRIGSEL_HPDF	Output0	HPDF_ITRG
		TRIGSEL_TIMER0ETI	Output0	TIMER0_ETI
		TRIGSEL_TIMER1ETI	Output0	TIMER1_ETI
		TRIGSEL_TIMER2ETI	Output0	TIMER2_ETI
		TRIGSEL_TIMER3ETI	Output0	TIMER3_ETI
		TRIGSEL_TIMER4ETI	Output0	TIMER4_ETI
		TRIGSEL_TIMER7ETI	Output0	TIMER7_ETI
		TRIGSEL_TIMER22ETI	Output0	TIMER22_ETI
		TRIGSEL_TIMER23ETI	Output0	TIMER23_ETI

触发源	触发选择	TRIGSEL 寄存器	TRIGSEL 输出	外设
TIMER51_TRGO0 RTC_Alarm RTC_TPTS 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 CMP0_OUT CMP1_OUT SAI0_FS0 SAI0_FS1 SAI2_FS0 SAI2_FS1		TRIGSEL_TIMER30ETI	Output0	TIMER30_ETI
		TRIGSEL_TIMER31ETI	Output0	TIMER31_ETI
		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_TIMER22ITI14	Output0	TIMER22_ITI14

触发源	触发选择	TRIGSEL 寄存器	TRIGSEL 输出	外设
		TRIGSEL_TIMER23ITI14	Output0	TIMER23_ITI14
		TRIGSEL_TIMER30ITI14	Output0	TIMER30_ITI14
		TRIGSEL_TIMER31ITI14	Output0	TIMER31_ITI14
		TRIGSEL_TIMER40ITI14	Output0	TIMER40_ITI14
		TRIGSEL_TIMER41ITI14	Output0	TIMER41_ITI14
		TRIGSEL_TIMER42ITI14	Output0	TIMER42_ITI14
		TRIGSEL_TIMER43ITI14	Output0	TIMER43_ITI14
		TRIGSEL_TIMER44ITI14	Output0	TIMER44_ITI14

注意：TIMERx_BRKINx只能选择TIMERx_BRKINx作为触发源。TIMERx_ITI14不能选择CMP_OUT、LXTAL_TRG、TIMERx_BRKINx和它们自己的信号作为触发源。除了TIMERx_BRKINx，其他输出都可以选择所有输入作为触发源。当选择了非法触发输入时，其输出将被强制选择为0。

触发输入选择 INSELx[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，TRIGSEL 触发输入选择为高电平。

9.5. TRIGSEL 寄存器

TRIGSEL 基地址：0x4001 8400

9.5.1. 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。

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_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 9-1. 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 9-1. 触发输入位域选择*。

9.5.2. 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。

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_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 9-1. 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 9-1. 触发输入位域选择*。

9.5.3. 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。

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 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 寄存器可读可写 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 9-1. 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 9-1. 触发输入位域选择*。

9.5.4. EXTOUT 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。

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 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 寄存器可读可写 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 9-1. 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 9-1. 触发输入位域选择*。

9.5.5. ADC0 触发选择寄存器（TRIGSEL_ADC0）

地址偏移：0x10

复位值：0x0000 1113

该寄存器只能按字（32位）访问。

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

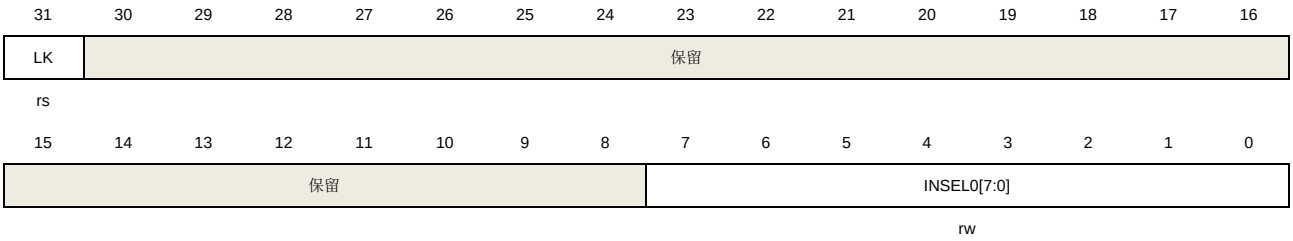
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								INSEL0[7:0]							
								rw							

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_ADC0将不能被修改。 0：TRIGSEL_ADC0 寄存器可读可写 1：TRIGSEL_ADC0 寄存器只读
30:16	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 9-1. 触发输入位域选择*。

9.5.6. ADC1 触发选择寄存器（TRIGSEL_ADC1）

地址偏移：0x14
复位值：0x0000 1113

该寄存器只能按字（32位）访问。

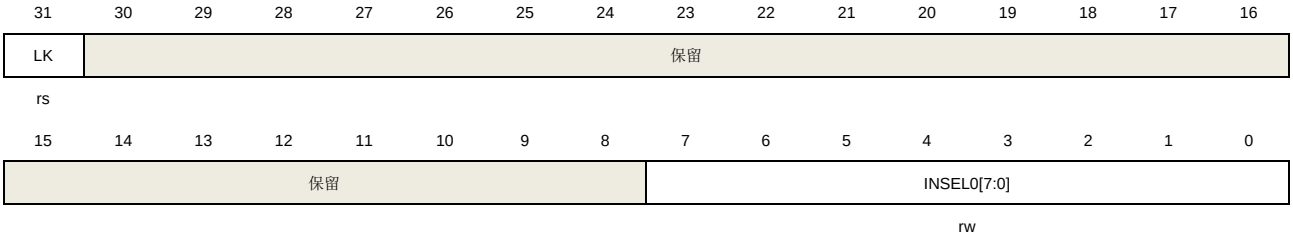


位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_ADC1将不能被修改。 0：TRIGSEL_ADC1 寄存器可读可写 1：TRIGSEL_ADC1 寄存器只读
30:16	保留	必须保持复位值。
15:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连到接输出 0 的触发输入信号，输出 0 作为 ADC1_ROUTRG（ADC1 常规序列）的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*。

9.5.7. ADC2 触发选择寄存器（TRIGSEL_ADC2）

地址偏移：0x18
复位值：0x0000 1113

该寄存器只能按字（32位）访问。

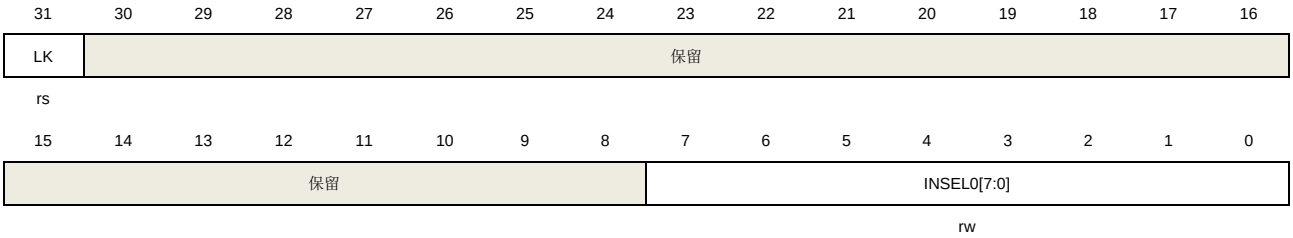


位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_ADC2将不能被修改。 0：TRIGSEL_ADC2 寄存器可读可写 1：TRIGSEL_ADC2 寄存器只读
30:16	保留	必须保持复位值。
15:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连到接输出 0 的触发输入信号，输出 0 作为 ADC2_ROUTRG（ADC2 常规序列）的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*。

9.5.8. DAC0_OUT0 触发选择寄存器（TRIGSEL_DAC0OUT0）

地址偏移：0x1C
复位值：0x0000 0025

该寄存器只能按位（32位）访问。



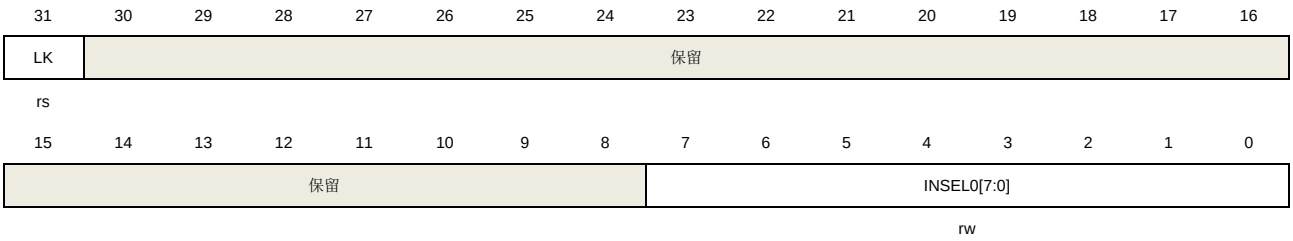
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_DAC0OUT0将不能被修改。 0：TRIGSEL_DAC0OUT0 寄存器可读可写 1：TRIGSEL_DAC0OUT0 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择

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 DAC0_OUT0_EXTRIG（DAC0_OUT0 外部触发）的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*

9.5.9. DAC0_OUT1 触发选择寄存器（TRIGSEL_DAC0OUT1）

地址偏移：0x20
复位值：0x0000 0025

该寄存器只能按位（32位）访问。

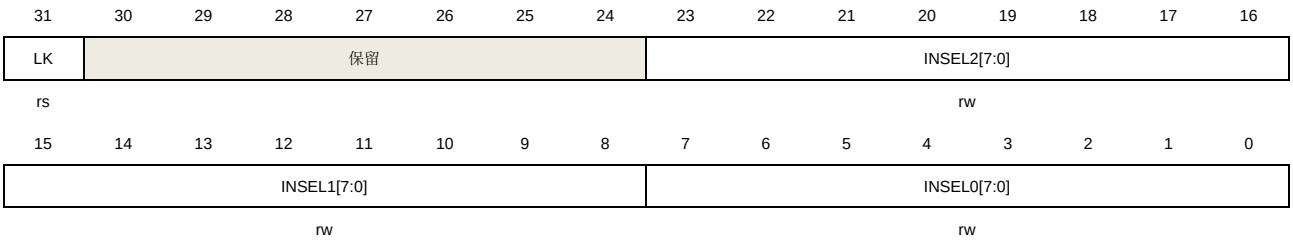


位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_DAC0OUT1将不能被修改。 0：TRIGSEL_DAC0OUT1 寄存器可读可写 1：TRIGSEL_DAC0OUT1 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 DAC0_OUT1_EXTRIG（DAC0_OUT1 外部触发）的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*。

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

地址偏移：0x24
复位值：0x0023 2221

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_TIMER0BRKIN将不能被修改。 0：TRIGSEL_TIMER0BRKIN 寄存器可读可写 1：TRIGSEL_TIMER0BRKIN 寄存器只读
30:24	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 TIMER0_BRINK2 的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*。
15:8	INSEL1[7:0]	触发输出 1 输入源选择 这些位用来选择连接到输出 1 的触发输入信号，输出 1 作为 TIMER0_BRINK1 的触发源。关于具体配置请参考*表 9-1. 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 TIMER0_BRINK0 的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*。

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

地址偏移：0x28
复位值：0x0051 504F

该寄存器只能按字（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_TIMER7BRKIN将不能被修改。 0：TRIGSEL_TIMER7BRKIN 寄存器可读可写 1：TRIGSEL_TIMER7BRKIN 寄存器只读
30:24	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 TIMER7_BRKIN2 的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*。
15:8	INSEL1[7:0]	触发输出 1 的输入源选择

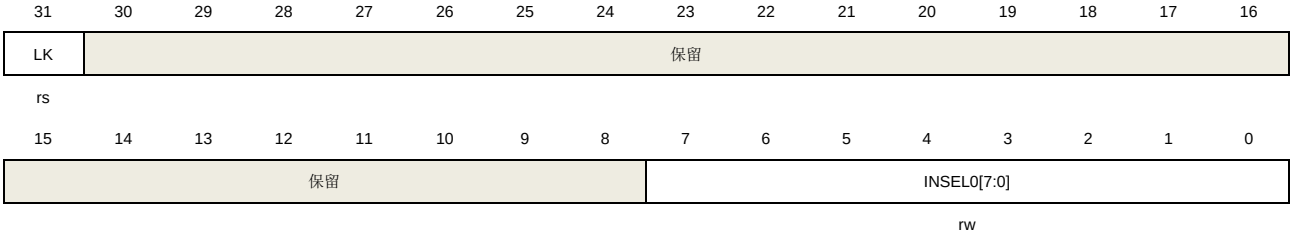
		这些位用来选择连接到输出 1 的触发输入信号，输出 1 作为 TIMER7_BRKIN1 的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER7_BRKIN0 的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*。

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

地址偏移：0x2C

复位值：0x0000 0059

该寄存器只能按字（32位）访问。



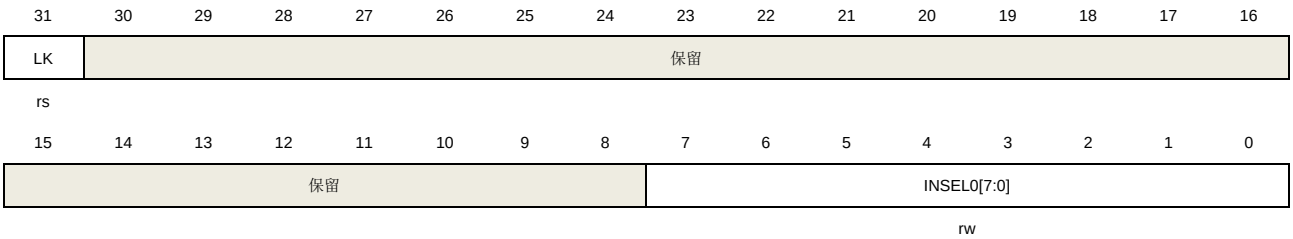
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER14BRKIN将不能被修改。 0：TRIGSEL_TIMER14BRKIN 寄存器可读可写 1：TRIGSEL_TIMER14BRKIN 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER14_BRKIN0 的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*

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

地址偏移：0x30

复位值：0x0000 005E

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_TIMER15BRKIN将不能被修改。 0：TRIGSEL_TIMER15BRKIN 寄存器可读可写 1：TRIGSEL_TIMER15BRKIN 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 TIMER15_BRKIN0 的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*。

9.5.14. TIMER16_BRKIN 触发选择寄存器（TRIGSEL_TIMER16BRKIN）

地址偏移：0x34
复位值：0x0000 0063

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_TIMER16BRKIN将不能被修改。 0：TRIGSEL_TIMER16BRKIN 寄存器可读可写 1：TRIGSEL_TIMER16BRKIN 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 TIMER16_BRKIN0 的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*。

9.5.15. TIMER40_BRKIN 触发选择寄存器（TRIGSEL_TIMER40BRKIN）

地址偏移：0x38
复位值：0x0000 0082

该寄存器只能按字（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_TIMER40BRKIN将不能被修改。 0：TRIGSEL_TIMER40BRKIN 寄存器可读可写 1：TRIGSEL_TIMER40BRKIN 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 TIMER40_BRKIN0 的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*

9.5.16. TIMER41_BRKIN 触发选择寄存器（TRIGSEL_TIMER41BRKIN）

地址偏移：0x3C

复位值：0x0000 0089

该寄存器只能按字（32位）访问。

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

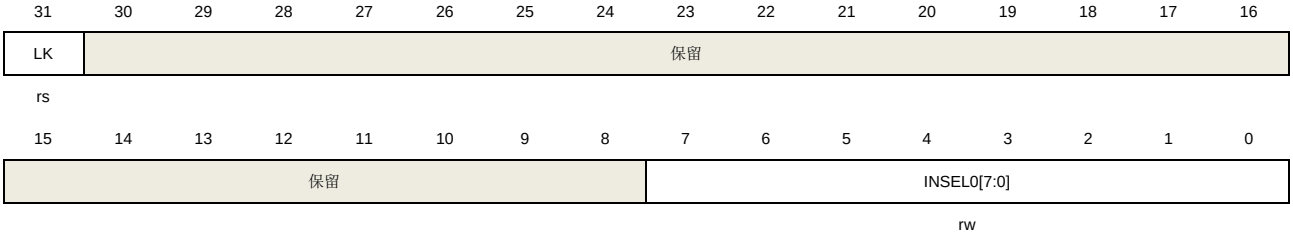
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								INSEL0[7:0]							
								rw							

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_TIMER41BRKIN将不能被修改。 0：TRIGSEL_TIMER41BRKIN 寄存器可读可写 1：TRIGSEL_TIMER41BRKIN 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 TIMER41_BRKIN0 的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*。

9.5.17. TIMER42_BRKIN 触发选择寄存器（TRIGSEL_TIMER42BRKIN）

地址偏移：0x40
复位值：0x0000 0090

该寄存器只能按字（32位）访问。

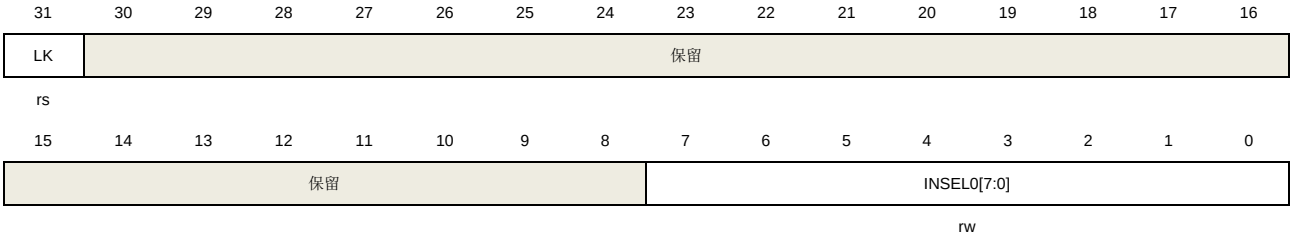


位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER42BRKIN将不能被修改。 0：TRIGSEL_TIMER42BRKIN 寄存器可读可写 1：TRIGSEL_TIMER42BRKIN 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER42_BRKIN0 的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*。

9.5.18. TIMER43_BRKIN 触发选择寄存器（TRIGSEL_TIMER43BRKIN）

地址偏移：0x44
复位值：0x0000 0097

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_TIMER43BRKIN将不能被修改。 0：TRIGSEL_TIMER43BRKIN 寄存器可读可写 1：TRIGSEL_TIMER43BRKIN 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 TIMER43_BRKIN0 的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*。

9.5.19. TIMER44_BRKIN 触发选择寄存器（TRIGSEL_TIMER44BRKIN）

地址偏移：0x48
复位值：0x0000 009e

该寄存器只能按字（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_TIMER44BRKIN将不能被修改。 0：TRIGSEL_TIMER44BRKIN 寄存器可读可写 1：TRIGSEL_TIMER44BRKIN 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 TIMER44_BRKIN0 的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*。

9.5.20. CAN0 触发选择寄存器（TRIGSEL_CAN0）

地址偏移：0x4C
复位值：0x0000 003d

该寄存器只能按字（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_CAN0将不能被修改。 0：TRIGSEL_CAN0 寄存器可读可写 1：TRIGSEL_CAN0 寄存器只读
30:8	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 9-1. 触发输入位域选择*。

9.5.21. CAN1 触发选择寄存器（TRIGSEL_CAN1）

地址偏移：0x50
复位值：0x0000 003d

该寄存器只能按字（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 寄存器可读可写 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 9-1. 触发输入位域选择*。

9.5.22. CAN2 触发选择寄存器（TRIGSEL_CAN2）

地址偏移：0x54
复位值：0x0000 003d

该寄存器只能按字（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_CAN2将不能被修改。 0：TRIGSEL_CAN2 寄存器可读可写 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 9-1. 触发输入位域选择*。

9.5.23. LPDTS 触发选择寄存器（TRIGSEL_LPDTS）

地址偏移：0x58
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

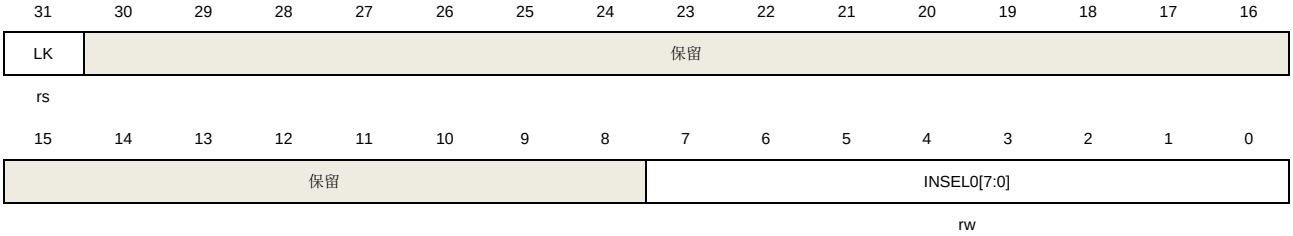
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								INSEL0[7:0]							
								rw							

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_LPDTS将不能被修改。 0：TRIGSEL_LPDTS 寄存器可读可写 1：TRIGSEL_LPDTS 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的触发输入信号，输出 0 作为 LPDTS_TRG 的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*。

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

地址偏移：0x5C
复位值：0x0000 0024

该寄存器只能按字（32位）访问。

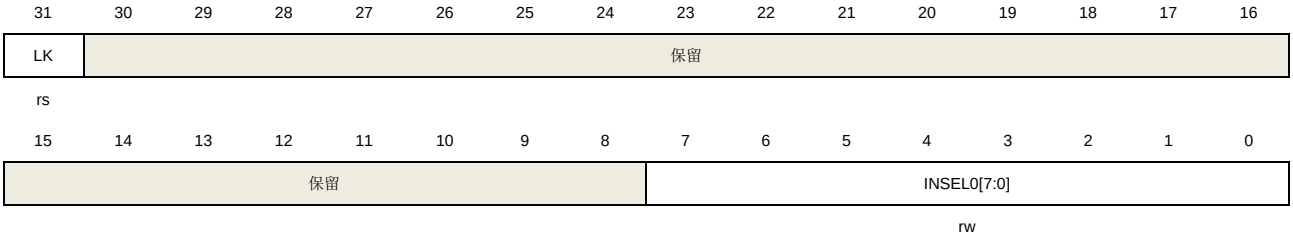


位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER0ETI将不能被修改。 0：TRIGSEL_TIMER0ETI 寄存器可读可写 1：TRIGSEL_TIMER0ETI 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出的触发输入信号，输出 0 作为 TIMER0_ETI 的触发源。 关于具体配置请参考*表 9-1. 触发输入位域选择*。

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

地址偏移：0x60
复位值：0x0000 002a

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 寄存器可读可写 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 9-1. 触发输入位域选择*。

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

地址偏移：0x64

复位值：0x0000 0030

该寄存器只能按字（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_TIMER2ETI将不能被修改。 0：TRIGSEL_TIMER2ETI 寄存器可读可写 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 9-1. 触发输入位域选择*。

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

地址偏移：0x68

复位值：0x0000 0036

该寄存器只能按字（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_TIMER3ETI将不能被修改。 0：TRIGSEL_TIMER3ETI 寄存器可读可写 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 9-1. 触发输入位域选择*。

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

地址偏移：0x6C
复位值：0x0000 003c

该寄存器只能按字（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_TIMER4ETI将不能被修改。 0：TRIGSEL_TIMER4ETI 寄存器可读可写 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 9-1. 触发输入位域选择*。

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

地址偏移：0x70
复位值：0x0000 0052

该寄存器只能按字（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_TIMER7ETI将不能被修改。 0：TRIGSEL_TIMER7ETI 寄存器可读可写 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 9-1. 触发输入位域选择*。

9.5.30. TIMER22_ETI 触发选择寄存器（TRIGSEL_TIMER22ETI）

地址偏移：0x74

复位值：0x0000 0069

该寄存器只能按字（32位）访问。

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

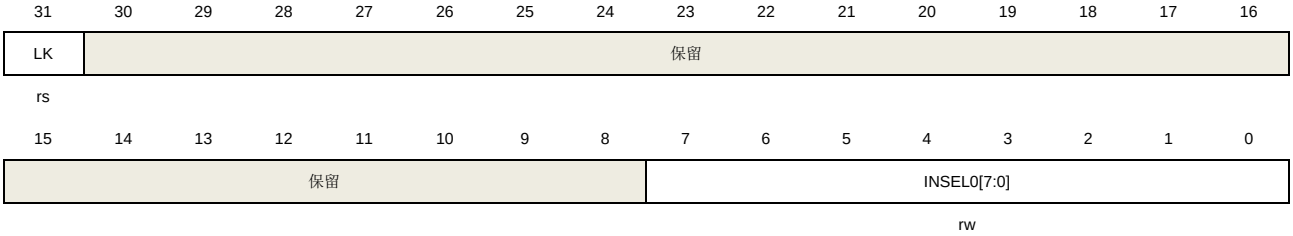
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								INSEL0[7:0]							
								rw							

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_TIMER22ETI将不能被修改。 0：TRIGSEL_TIMER22ETI 寄存器可读可写 1：TRIGSEL_TIMER22ETI 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的触发输入信号，输出 0 作为 TIMER22_ETI 的触发源。 关于具体配置请参考*表 9-1. 触发输入位域选择*。

9.5.31. TIMER23_ETI 触发选择寄存器（TRIGSEL_TIMER23ETI）

地址偏移：0x78
复位值：0x0000 006f

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_TIMER23ETI将不能被修改。 0：TRIGSEL_TIMER23ETI 寄存器可读可写 1：TRIGSEL_TIMER23ETI 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的触发输入信号，输出 0 作为 TIMER23_ETI 的触发源。 关于具体配置请参考*表 9-1. 触发输入位域选择*

9.5.32. TIMER30_ETI 触发选择寄存器（TRIGSEL_TIMER30ETI）

地址偏移：0x7C
复位值：0x0000 0075

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_TIMER30ETI将不能被修改。 0：TRIGSEL_TIMER30ETI 寄存器可读可写 1：TRIGSEL_TIMER30ETI 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的触发输入信号，输出 0 作为 TIMER30_ETI 的触发源。 关于具体配置请参考*表 9-1. 触发输入位域选择*。

9.5.33. TIMER31_ETI 触发选择寄存器（TRIGSEL_TIMER31ETI）

地址偏移：0x80
复位值：0x0000 007b

该寄存器只能按字（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_TIMER31ETI将不能被修改。 0：TRIGSEL_TIMER31ETI 寄存器可读可写 1：TRIGSEL_TIMER31ETI 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的触发输入信号，输出 0 作为 TIMER31_ETI 的触发源。 关于具体配置请参考*表 9-1. 触发输入位域选择*。

9.5.34. EDOUT 触发选择寄存器（TRIGSEL_EDOUT）

地址偏移：0x84
复位值：0x0000 0000

该寄存器只能按字（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_EDOUT将不能被修改。 0：TRIGSEL_EDOUT 寄存器可读可写 1：TRIGSEL_EDOUT 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的触发输入信号，输出 0 作为 EDOUT_TRG 的触发源。 关于具体配置请参考*表 9-1. 触发输入位域选择*。

9.5.35. HPDF 触发选择寄存器（TRIGSEL_HPDF）

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							

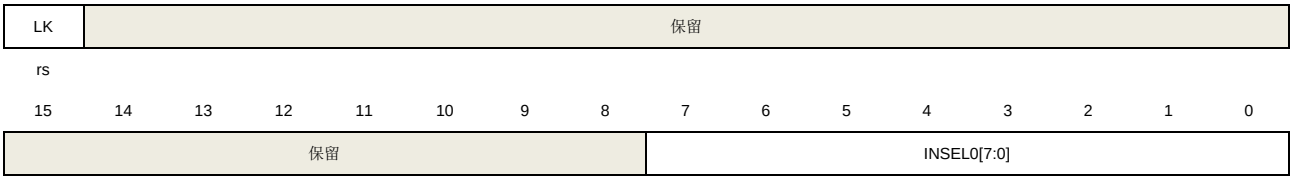
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_HPDF将不能被修改。 0：TRIGSEL_HPDF 寄存器可读可写 1：TRIGSEL_HPDF 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出的触发输入信号，输出 0 作为 HPDF_ITRG 的触发源。 关于具体配置请参考*表 9-1. 触发输入位域选择*。

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

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



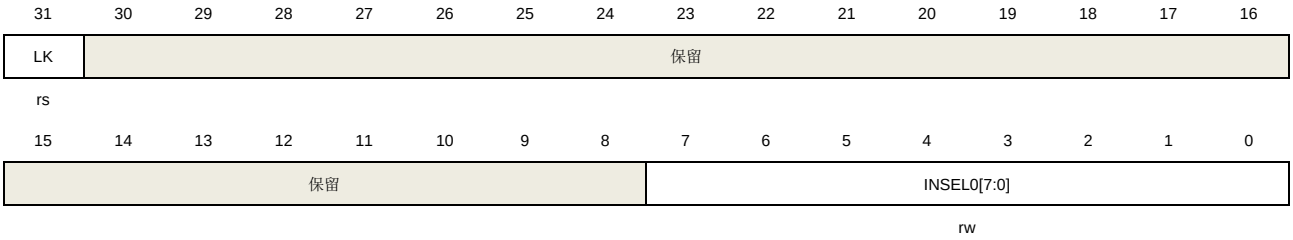
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER0ITI14将不能被修改。 0：TRIGSEL_TIMER0ITI14 寄存器可读可写 1：TRIGSEL_TIMER0ITI14 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出的触发输入信号，输出 0 作为 TIMER0_ITI14 的触发源。 关于具体配置请参考*表 9-1. 触发输入位域选择*。

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

地址偏移：0x90

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

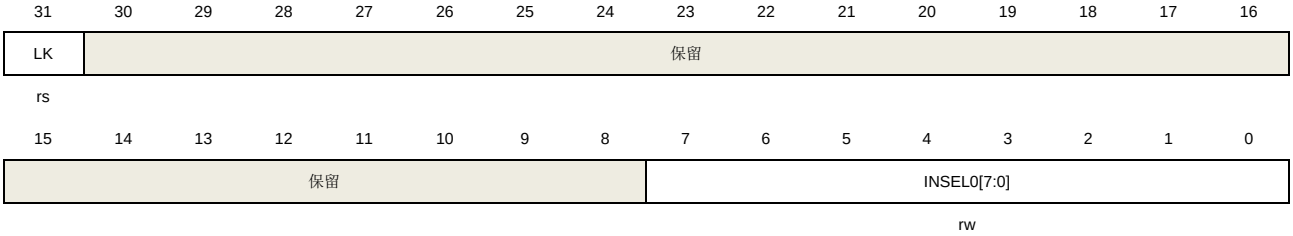


位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER1ITI14将不能被修改。 0：TRIGSEL_TIMER1ITI14 寄存器可读可写 1：TRIGSEL_TIMER1ITI14 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出的触发输入信号，输出 0 作为 TIMER1_ITI14 的触发源。 关于具体配置请参考*表 9-1. 触发输入位域选择*。

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

地址偏移：0x94
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

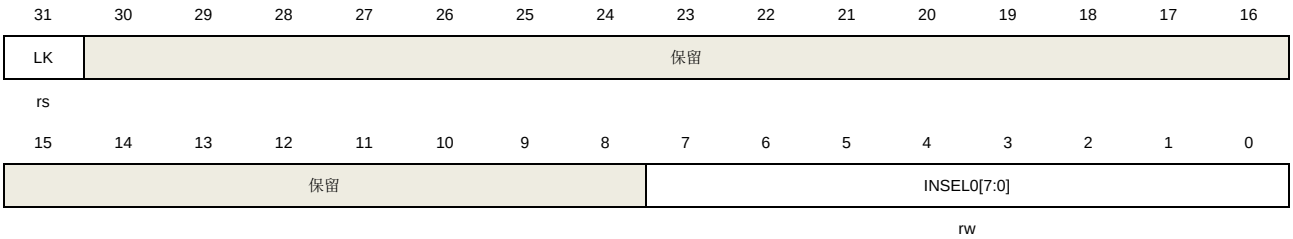


位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER2ITI14将不能被修改。 0：TRIGSEL_TIMER2ITI14 寄存器可读可写 1：TRIGSEL_TIMER2ITI14 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出的触发输入信号，输出 0 作为 TIMER2_ITI14 的触发源。 关于具体配置请参考*表 9-1. 触发输入位域选择*。

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

地址偏移：0x98
复位值：0x0000 0000

该寄存器只能按字（32位）访问。



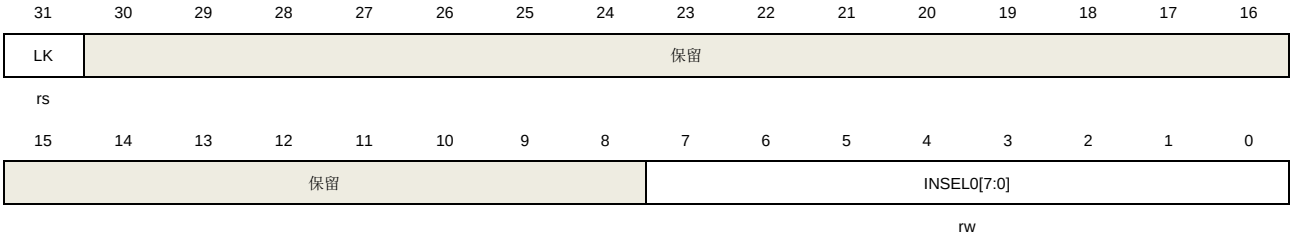
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER3ITI14将不能被修改。 0：TRIGSEL_TIMER3ITI14 寄存器可读可写 1：TRIGSEL_TIMER3ITI14 寄存器只读

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 9-1. 触发输入位域选择*。

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

地址偏移：0x9C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

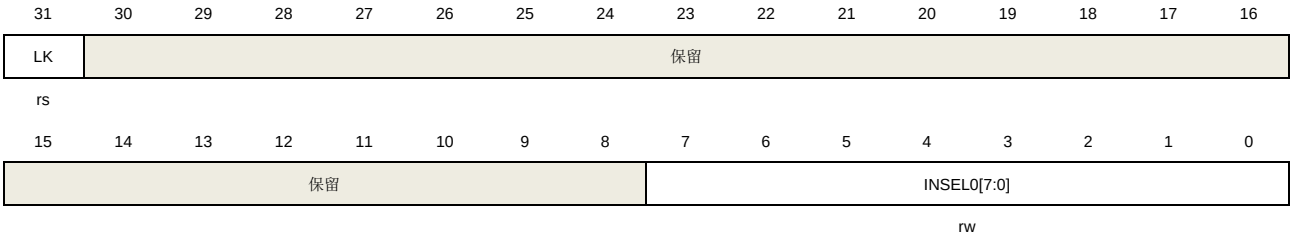


位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER4ITI14将不能被修改。 0：TRIGSEL_TIMER4ITI14 寄存器可读可写 1：TRIGSEL_TIMER4ITI14 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出的触发输入信号，输出 0 作为 TIMER4_ITI14 的触发源。 关于具体配置请参考*表 9-1. 触发输入位域选择*。

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

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	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 寄存器可读可写 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 9-1. 触发输入位域选择*。

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

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
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							

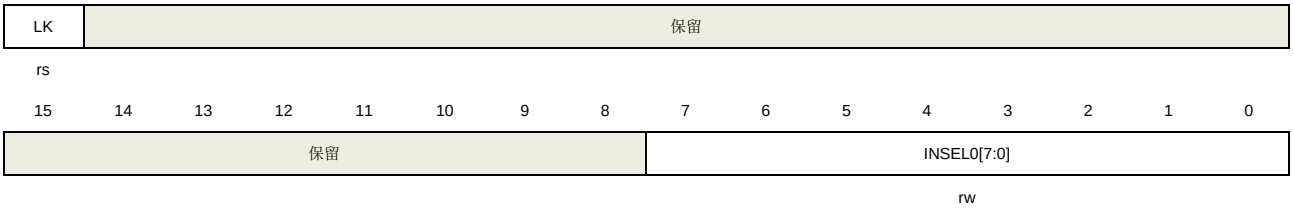
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER14ITI14将不能被修改。 0：TRIGSEL_TIMER14ITI14 寄存器可读可写 1：TRIGSEL_TIMER14ITI14 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出的触发输入信号，输出 0 作为 TIMER14_ITI14 的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*。

9.5.43. TIMER22_ITI14 触发选择寄存器（TRIGSEL_TIMER22ITI14）

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

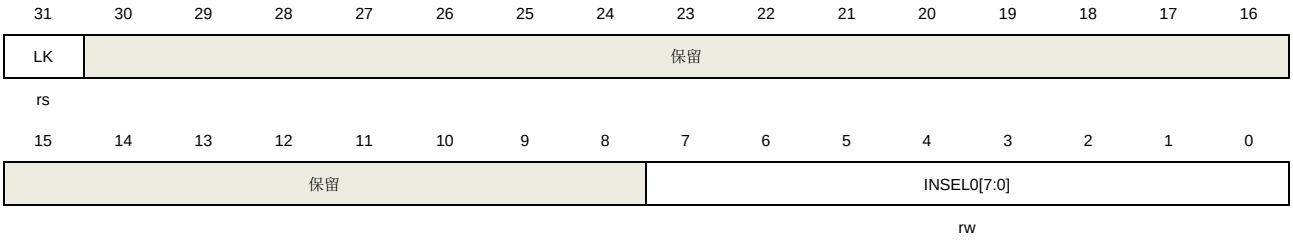


位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER22ITI14将不能被修改。 0：TRIGSEL_TIMER22ITI14 寄存器可读可写 1：TRIGSEL_TIMER22ITI14 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出的触发输入信号，输出 0 作为 TIMER22_ITI14 的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*。

9.5.44. TIMER23_ITI14 触发选择寄存器（TRIGSEL_TIMER23ITI14）

地址偏移：0xAC
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

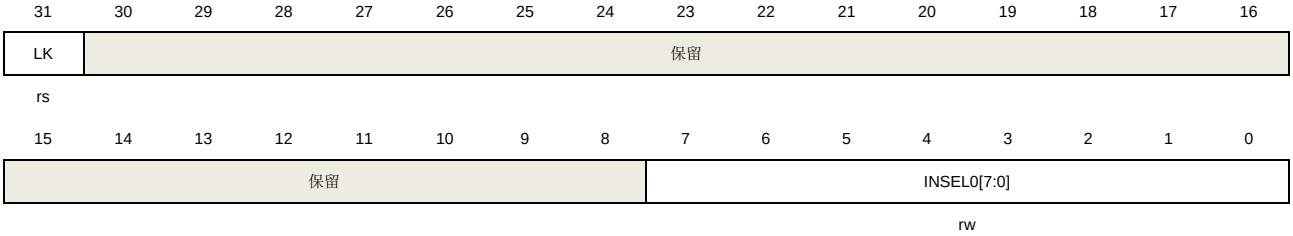


位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER23ITI14将不能被修改。 0：TRIGSEL_TIMER23ITI14 寄存器可读可写 1：TRIGSEL_TIMER23ITI14 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出的触发输入信号，输出 0 作为 TIMER23_ITI14 的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*

9.5.45. TIMER30_ITI14 触发选择寄存器（TRIGSEL_TIMER30ITI14）

地址偏移：0xB0
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_TIMER30ITI14将不能被修改。 0：TRIGSEL_TIMER30ITI14 寄存器可读可写 1：TRIGSEL_TIMER30ITI14 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的触发输入信号，输出 0 作为 TIMER30_ITI14 的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*。

9.5.46. TIMER31_ITI14 触发选择寄存器（TRIGSEL_TIMER31ITI14）

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	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_TIMER31ITI14将不能被修改。 0：TRIGSEL_TIMER31ITI14 寄存器可读可写 1：TRIGSEL_TIMER31ITI14 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的触发输入信号，输出 0 作为 TIMER31_ITI14 的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*。

9.5.47. TIMER40_ITI14 触发选择寄存器（TRIGSEL_TIMER40ITI14）

地址偏移：0xB8
复位值：0x0000 0000

该寄存器只能按字（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_TIMER40ITI14将不能被修改。 0：TRIGSEL_TIMER40ITI14 寄存器可读可写 1：TRIGSEL_TIMER40ITI14 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的触发输入信号，输出 0 作为 TIMER40_ITI14 的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*。

9.5.48. TIMER41_ITI14 触发选择寄存器（TRIGSEL_TIMER41ITI14）

地址偏移：0xBC
复位值：0x0000 0000

该寄存器只能按字（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_TIMER41ITI14将不能被修改。 0：TRIGSEL_TIMER41ITI14 寄存器可读可写 1：TRIGSEL_TIMER41ITI14 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的触发输入信号，输出 0 作为 TIMER41_ITI14 的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*。

9.5.49. TIMER42_ITI14 触发选择寄存器（TRIGSEL_TIMER42ITI14）

地址偏移：0xC0
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

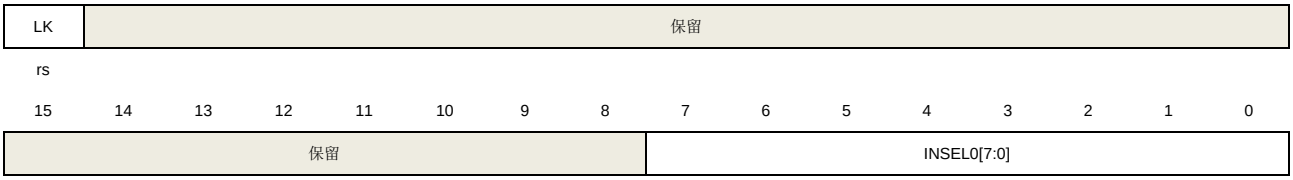
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER42ITI14将不能被修改。 0：TRIGSEL_TIMER42ITI14 寄存器可读可写 1：TRIGSEL_TIMER42ITI14 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出的触发输入信号，输出 0 作为 TIMER42_ITI14 的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*。

9.5.50. TIMER43_ITI14 触发选择寄存器（TRIGSEL_TIMER43ITI14）

地址偏移：0xC4
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

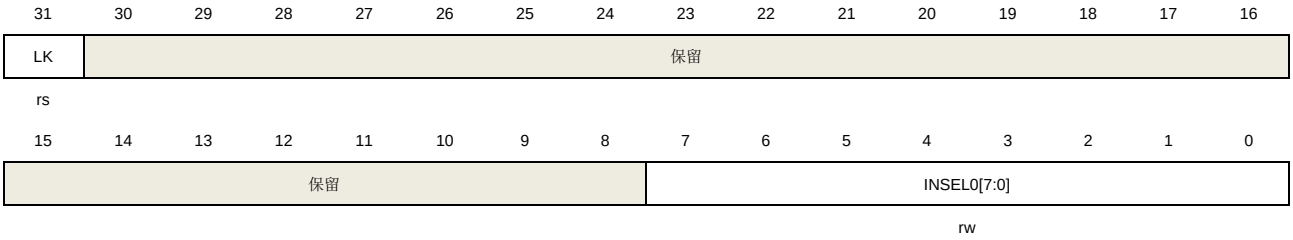


位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER43ITI14将不能被修改。 0：TRIGSEL_TIMER43ITI14 寄存器可读可写 1：TRIGSEL_TIMER43ITI14 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出的触发输入信号，输出 0 作为 TIMER43_ITI14 的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*。

9.5.51. TIMER44_ITI14 触发选择寄存器（TRIGSEL_TIMER44ITI14）

地址偏移：0xC8
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_TIMER44ITI14将不能被修改。 0：TRIGSEL_TIMER44ITI14 寄存器可读可写 1：TRIGSEL_TIMER44ITI14 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的触发输入信号，输出 0 作为 TIMER44_ITI14 的触发源。关于具体配置请参考*表 9-1. 触发输入位域选择*。

10. 通用和备用输入/输出接口（GPIO 和 AFIO）

10.1. 简介

GD32H73x_75x 系列最多可支持 131 个通用 I/O 引脚（GPIO），分别为 PA0～PA10，PA13～PA15，PB0～PB13，PC0～PC15，PD0～PD15，PE0～PE15，PF0～PF15，PG0～PG15，PH0 ～ PH15，PJ8 ～ PJ11，PK0 ～ PK2。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。

10.2. 主要特征

- 输入/输出方向控制；
- 施密特触发输入功能使能控制；
- 每个引脚都具有弱上拉/下拉功能；
- 推挽/开漏输出使能控制；
- 置位/复位输出使能；
- 可编程的边沿触发外部中断-由 EXTI 寄存器配置；
- 模拟输入/输出配置；
- 备用功能输入/输出配置；
- 端口锁定配置；
- 单周期输出翻转功能。

10.3. 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。

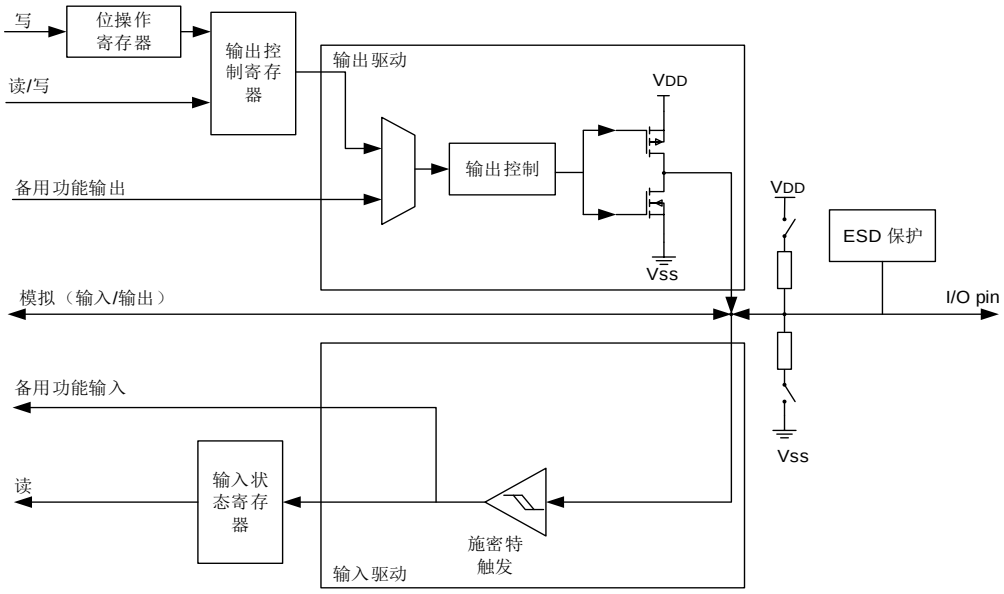
表 10-1. GPIO 配置表

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

PAD TYPE			CTLy	OMy	PUDy
		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10-1. 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 10-1. 标准 I/O 端口位的基本结构



10.3.1. GPIO 引脚配置

在复位期间或复位之后，备用功能并未激活，除端口 A/B 外所有 GPIO 端口都被配置成模拟模式，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

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。

10.3.2. 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。

10.3.3. 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。

10.3.4. 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。

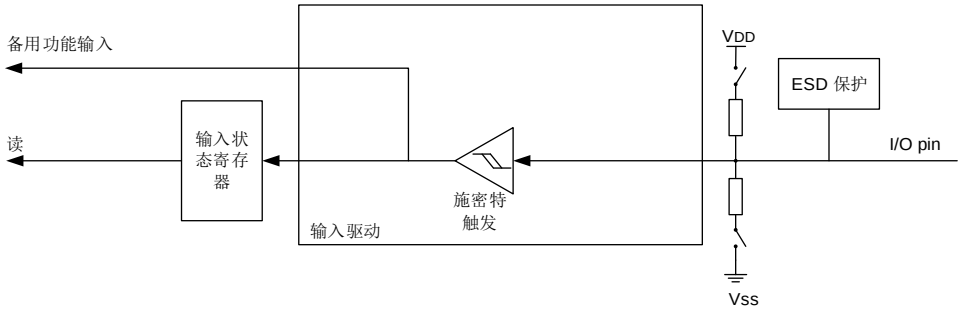
10.3.5. 输入配置

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10-2. 浮空/上拉/下拉输入配置*是I/O引脚的输入配置。

图 10-2. 浮空/上拉/下拉输入配置



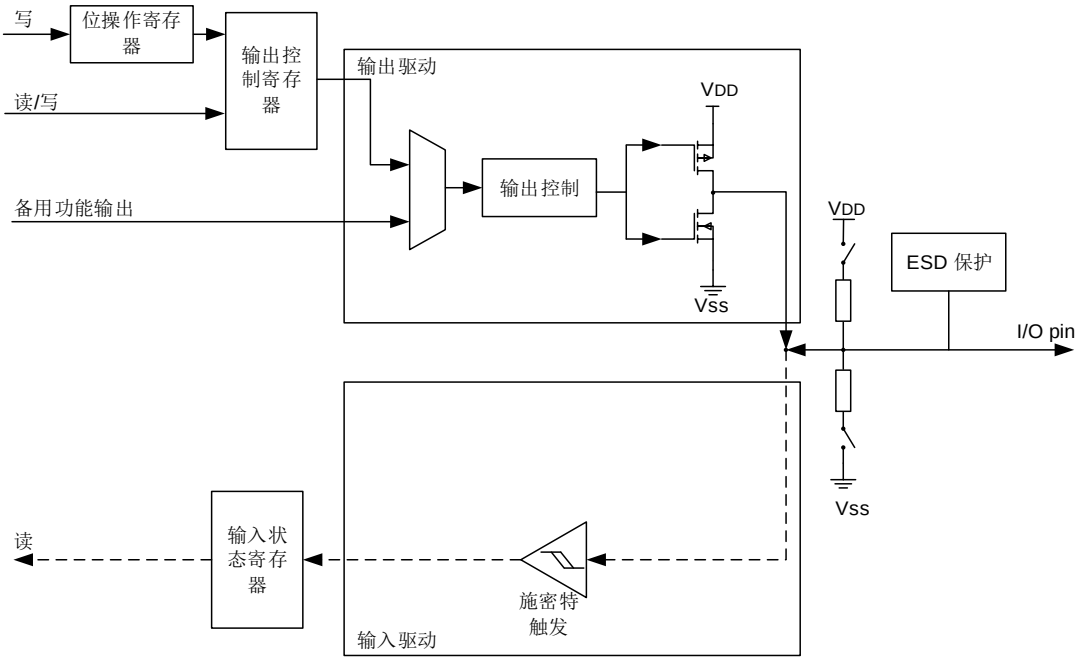
10.3.6. 输出配置

当GPIO配置为输出时：

- 施密特触发输入使能；
- 可选择的弱上拉和下拉电阻；
- 开漏模式：输出控制寄存器设置为“0”时，相应引脚输出低电平；输出控制寄存器设置为“1”，相应管脚处于高阻状态；
- 推挽模式：输出控制寄存器设置为“0”时，相应引脚输出低电平；输出控制寄存器设置为“1”，相应引脚输出高电平；
- 在推挽模式下，对端口输出控制寄存器的读访问将返回上次写入的值；
- 在开漏模式下，对端口输入状态寄存器的读访问将返回I/O的状态。

*图 10-3. 输出配置*是 I/O 端口的输出配置。

图 10-3. 输出配置



10.3.7. 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10-4. 模拟高阻配置*是I/O端口的模拟高阻配置。

图 10-4. 模拟高阻配置



10.3.8. 备用功能（AF）配置

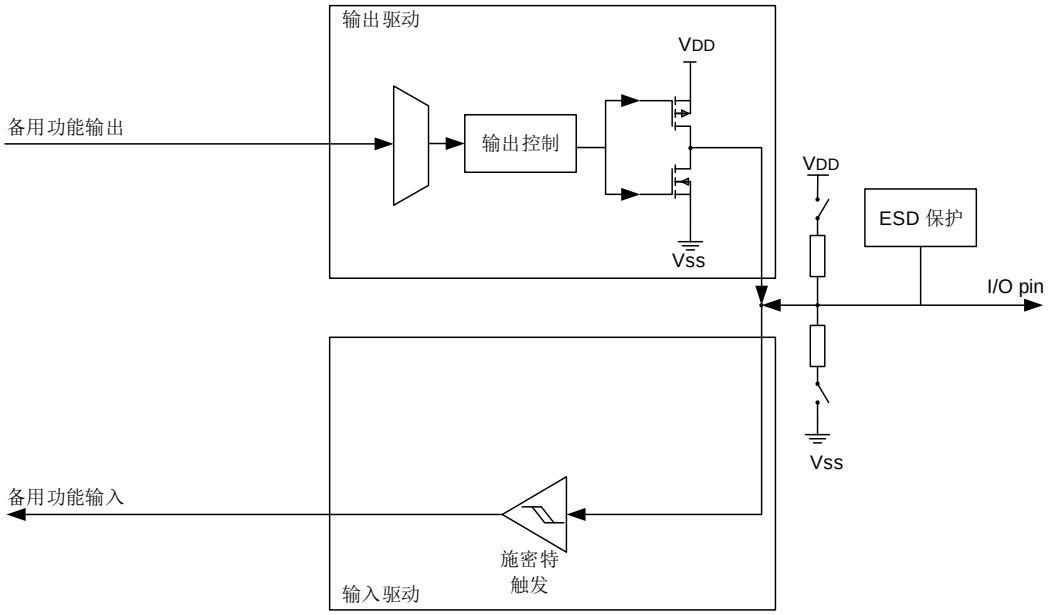
为了适应不同的器件封装，GPIO端口支持软件配置将一些备用功能应用到其他引脚上。

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10-5. 备用功能配置*是I/O端口备用功能配置图。

图 10-5. 备用功能配置



10.3.9. 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。

10.3.10. 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。

10.3.11. 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。

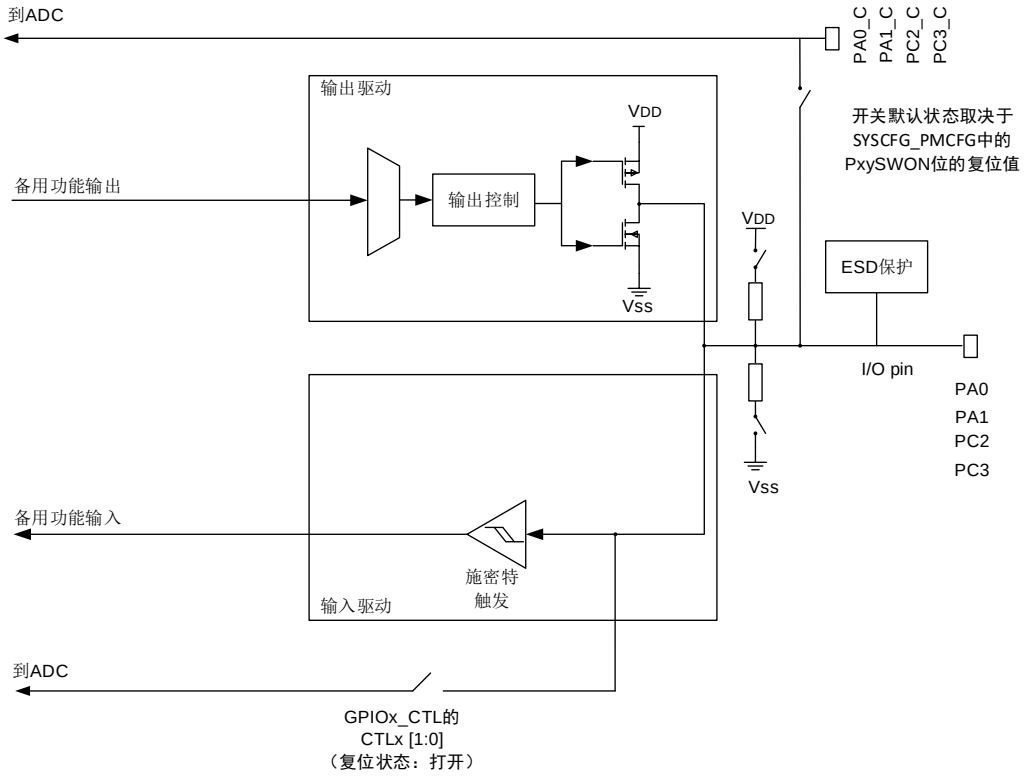
该单位为当前温度、环境等条件下提供最佳的补偿码，当SYSCFG_CPSCTL的CPS_RDY位置位时，可以读取存储在该区域中的补偿码。用户也可以通过软件编程SYSCFG_CPSCCCFG寄存器来配置补偿码。

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。这里的两个电压范围都是指芯片的VDD电压范围。当VDD为1.62~2.0V时，因为VDD电压比较低，IO速度比较慢，配置IOSPDOP为1，可以使得IO速度得到改善。当VDD为2.7V~3.6V时，VDD电压足够高，IO速度比较快，此时无需设置IOSPDOP。

10.3.12. ADC 的模拟配置

对于GD32H73x_75x系列，引脚PA0_C、PA1_C、PC2_C和PC3_C直接连接到ADC模拟输入端（如*图10-6. GD32H73x_75x系列ADC的模拟配置*）：Pxy_C和Pxy引脚之间通过模拟开关直接相连（详见system寄存器）。

图10-6. GD32H73x_75x系列ADC的模拟配置



10.3.13. 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。

注意：GPIOx_ISTAT寄存器的状态为GPIO滤波前信号，输出给EXTI的信号为GPIO滤波后信号。

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。

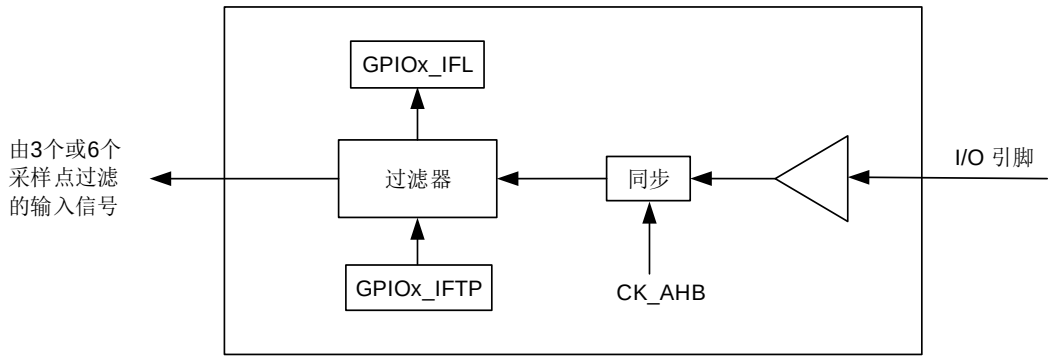
仅同步到 CK_AHB

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10-7. 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10-8. 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10-8. 输入滤波时钟周期*所示的例子，输入滤波配置如下：

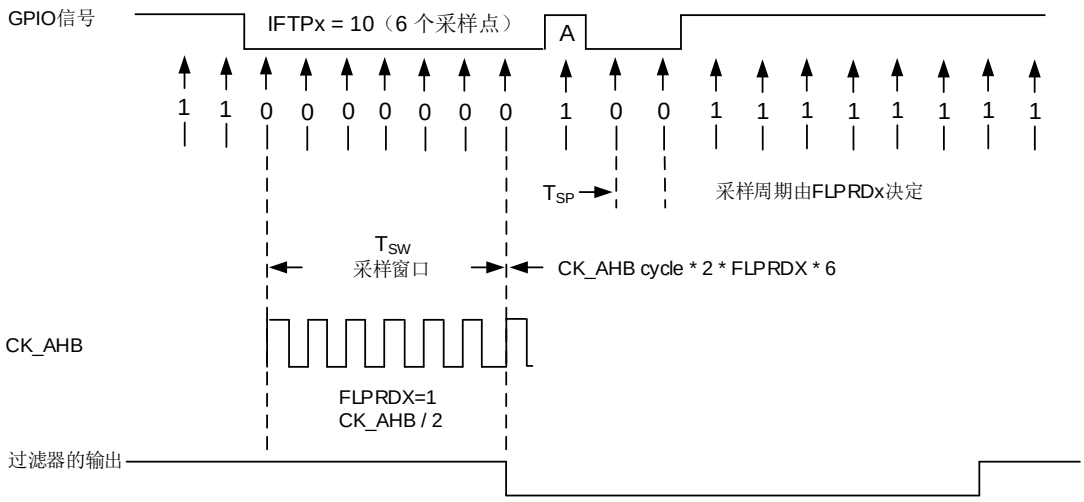
- GPIOx_IFTP寄存器的IFTP0[1:0]=10，表示有6个采样点；

- GPIOx_IFL寄存器的FLPRD0=1，采样周期$T_{SP}=2\times FLPRD0\times T_{CK_AHB}=2\times T_{CK_AHB}$；

配置结果如下：

- 采样宽度为：$T_{SW}=6\times T_{SP}=6\times 2\times FLPRD0\times T_{CK_AHB}=6\times 2\times T_{CK_AHB}$；
- 如果T_{CK_AHB}=10ns，则采样窗口的持续时间为：
  $T_{SP}=2\times T_{CK_AHB}=2\times 10ns=20ns$
  $T_{SW}=6\times 2\times T_{CK_AHB}=6\times 20ns=120ns$
- 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10-8. 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10-8. 输入滤波时钟周期



10.4. GPIO 寄存器

GPIOA基地址：0x5802 0000

GPIOB基地址：0x5802 0400

GPIOC基地址：0x5802 0800

GPIOD基地址：0x5802 0C00

GPIOE基地址：0x5802 1000

GPIOF基地址：0x5802 1400

GPIOG基地址：0x5802 1800

GPIOH基地址：0x5802 1C00

GPIOJ基地址：0x5802 2400

GPIOK基地址：0x5802 2800

10.4.1. 端口控制寄存器（GPIOx_CTL，x=A...H，J，K）

地址偏移：0x00
复位值：端口 A 0xABFF FFFF；端口 B 0xFFFF FE8F；其他端口 0xFFFF FFFF

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）写，仅能按字（32 位）读。

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）

10.4.2. 端口输出模式寄存器（GPIOx_OMODE，x=A...H，J，K）

地址偏移：0x04

复位值：0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）写，仅能按字（32 位）读。

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
OM15	OM14	OM13	OM12	OM11	OM10	OM9	OM8	OM7	OM6	OM5	OM4	OM3	OM2	OM1	OM0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

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

10.4.3. 端口输出速度寄存器（GPIOx_OSPD，x=A…H，J，K）

地址偏移：0x08

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

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）写，仅能按字（32 位）读。

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

10.4.4. 端口上拉/下拉寄存器（GPIOx_PUD，x=A...H，J，K）

地址偏移：0x0C

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

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）写，仅能按字（32 位）读。

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

10.4.5. 端口输入状态寄存器（GPIOx_ISTAT，x=A…H，J，K）

地址偏移：0x10
复位值：0x0000 XXXX

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）写，仅能按字（32 位）读。

31	30	29	28	27	26	25	24	23	22	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

10.4.6. 端口输出控制寄存器（GPIOx_OCTL，x=A...H，J，K）

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）写，仅能按字（32 位）读。

31	30	29	28	27	26	25	24	23	22	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

10.4.7. 端口位操作寄存器（GPIOx_BOP，x=A...H，J，K）

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）写，仅能按字（32 位）读。

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

10.4.8. 端口配置锁定寄存器（GPIOx_LOCK，x=A...H，J，K）

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

10.4.9. 备用功能选择寄存器 0（GPIOx_AFSEL0，x=A...H，J，K）

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）写，仅能按字（32 位）读。

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

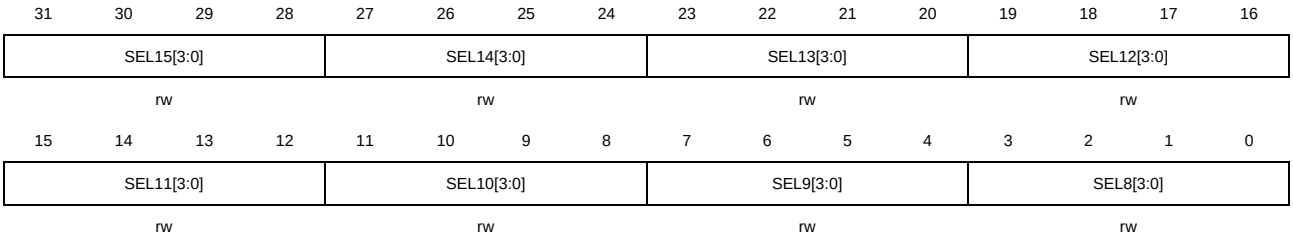
3:0	SEL0[3:0]	Pin 0选择备用功能 该位由软件置位和清除。 0000：选择AF0功能（复位值） 0001：选择AF1功能 0010：选择AF2功能 0011：选择AF3功能 0100：选择AF4功能 0101：选择AF5功能 0110：选择AF6功能 0111：选择AF7功能 1000：选择AF8功能 1001：选择AF9功能 1010：选择AF10功能 1011：选择AF11功能 1100：选择AF12功能 1101：选择AF13功能 1110：选择AF14功能 1111：选择AF15功能

10.4.10. 备用功能选择寄存器 1（GPIOx_AFSEL1，x=A...H，J，K）

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）写，仅能按字（32 位）读。



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 1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：选择AF10功能 1011：选择AF11功能 1100：选择AF12功能 1101：选择AF13功能 1110：选择AF14功能 1111：选择AF15功能

10.4.11. 位清除寄存器（GPIOx_BC，x=A...H，J，K）

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）写，仅能按字（32 位）读。

31	30	29	28	27	26	25	24	23	22	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

10.4.12. 端口位翻转寄存器（GPIOx_TG，x=A…H，J，K）

地址偏移：0x2C

复位值：0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）写，仅能按字（32 位）读。

31	30	29	28	27	26	25	24	23	22	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

10.4.13. 输入滤波寄存器（GPIOx_IFL，x=A…H，J，K）

地址偏移：0x30

复位值：0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）写，仅能按字（32 位）读。

31	30	29	28	27	26	25	24	23	22	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

10.4.14. 输入滤波类型寄存器（GPIOx_IFTP，x=A...H，J，K）

地址偏移：0x34

复位值：0xFFFF FFFF

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）写，仅能按字（32 位）读。

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）

11. 循环冗余校验管理单元（CRC）

11.1. 简介

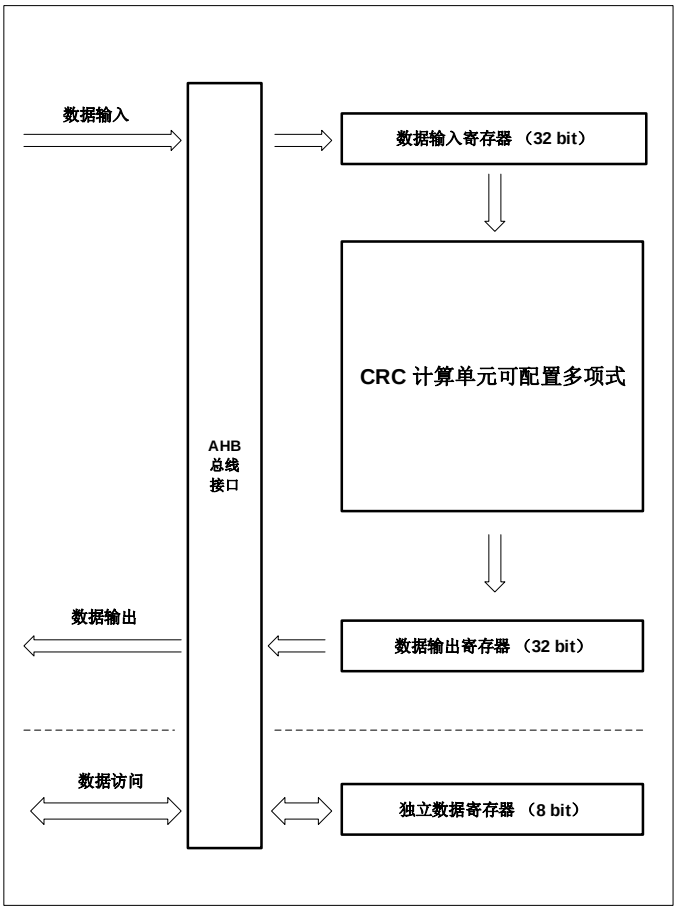
循环冗余校验码是一种用在数字网络和存储设备上的差错校验码，可以校验原始数据的偶然误差。

CRC 计算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。

11.2. 主要特性

- 支持7 / 8 / 16 / 32位数据输入；
- 对于7（8）/ 16 / 32位的输入数据长度，计算周期分别为1 / 2 / 4个AHB时钟周期；
- 用户可以配置多项式及多项式长度；
- CRC复位后，用户可以配置计算初值；
- 配有与计算无关的独立8位寄存器，可以供其他任何外设使用。

图 11-1. CRC 计算单元框图



11.3. 功能描述

- 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。

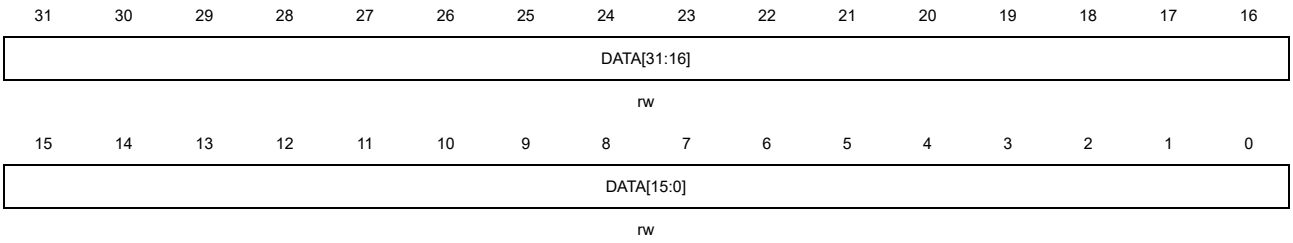
11.4. CRC 寄存器

CRC基地址：0x4002 3000

11.4.1. 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。

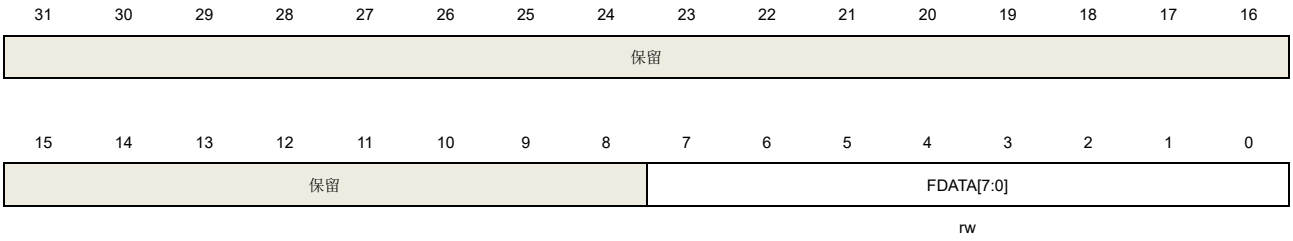


位/位域	名称	描述
31:0	DATA[31:0]	CRC 计算结果位 软件可读可写。 该寄存器用于接收待计算的新数据，直接将其写入即可。刚写入的数据不能被读出来因为读取该寄存器得到的是上次 CRC 计算的结果。

11.4.2. 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。



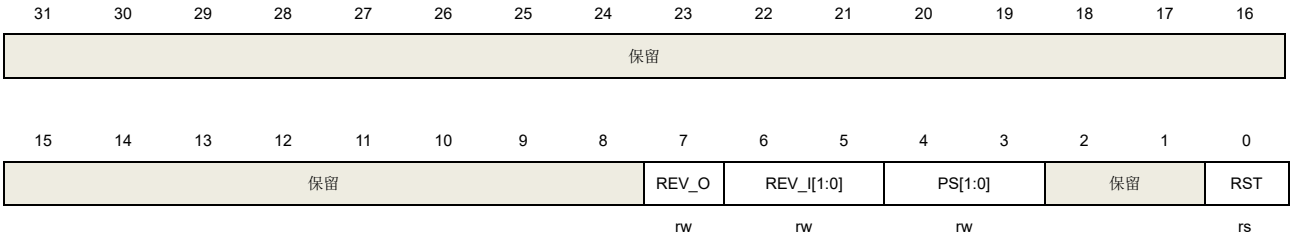
位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	FDATA[7:0]	独立数据寄存器位 软件可读可写。 这些位与 CRC 计算无关。该字节能被任何其他外设用于其他任何目的。该字节不受 CRC_CTL 寄存器的影响。

11.4.3. 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。



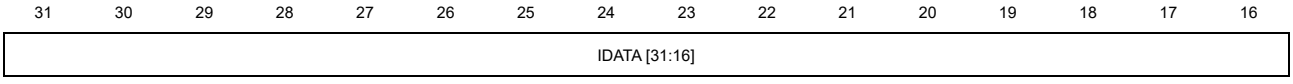
位/位域	名称	描述
31:8	保留	必须保持复位值。
7	REV_O	按位顺序翻转输出数据功能 0：输出数据不翻转 1：输出数据按位顺序翻转
6:5	REV_I[1:0]	翻转输入数据功能 0：输入数据不翻转 1：输入数据按字节翻转 2：输入数据按半字翻转 3：输入数据按字翻转
4:3	PS[1:0]	多项式长度 0：32 位 1：16（POLY[15:0]用于计数）位 2：8（POLY[7:0]用于计数）位 3：7（POLY[6:0]用于计数）位
2:1	保留	必须保持复位值。
0	RST	软件可读写 该位用来复位 CRC_DATA 寄存器。 置位时，CRC_DATA 寄存器的值将自动初始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

11.4.4. 初值寄存器（CRC_IDATA）

地址偏移：0x10

复位值：0xFFFF FFFF

该寄存器只能按字（32 位）访问



rw

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

rw

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

11.4.5. 多项式寄存器（CRC_POLY）

地址偏移：0x14

复位值：0x04C1 1DB7

该寄存器只能按字（32 位）访问

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

rw

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

rw

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

12. 真随机数生成器（TRNG）

12.1. 简介

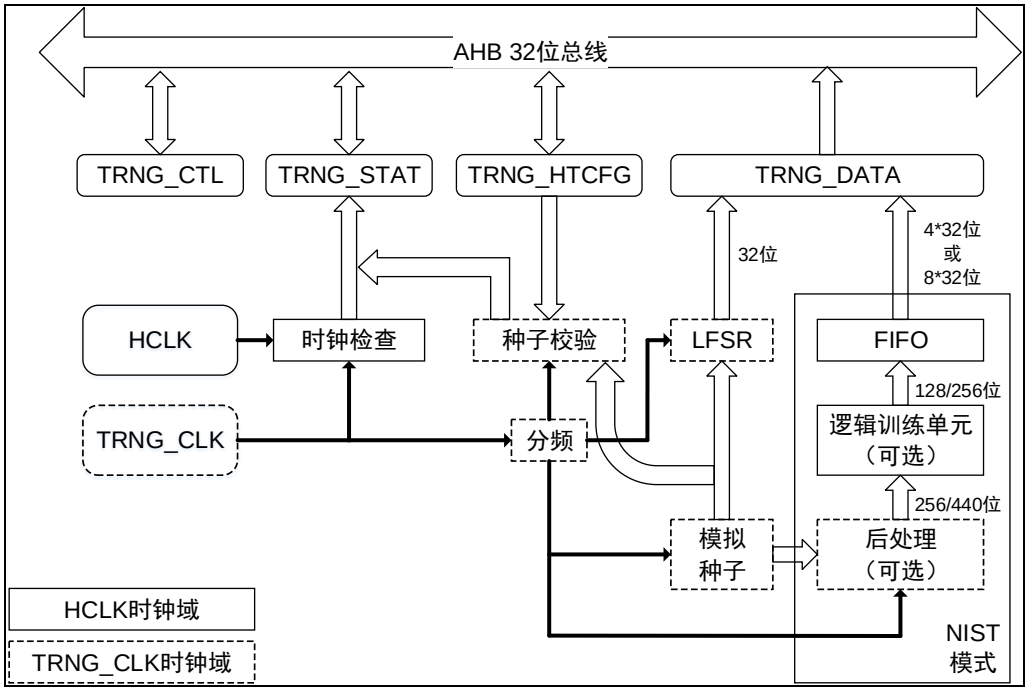
真随机数发生器模块（TRNG）能够通过连续模拟噪声生成一个 32 位的随机数值，该模块遵循 NIST SP800-90B 标准。

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4 * 32位或8 * 32位的随机数；
- 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. 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，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质量。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，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（参考 RCU 章节描述）驱动，因此随机数质量仅与 TRNG_CLK 时钟有关，与 HCLK 频率无关。

12.3.1. 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质量。TRNG 每次产生一个 32 位随机数。

12.3.2. 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。

12.3.3. 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，用以增加输出固定长度比特流的质量。NIST SP800-90B 的目标确保输出是满熵的。

通过配置 TRNG_CTL 寄存器中的 ALGO 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。

12.3.4. 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

12.3.5. 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 APT_TH 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 REP_TH 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 ERR_STA 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。

12.3.6. 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 RT_EN 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。

12.3.7. 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、ERR_STA、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 ERR_STA 位置位；
- 产生一次时钟错误，且 TRNG_STAT 寄存器中的 CEIF 和 ERR_STA 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。

12.3.8. 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。

12.3.9. 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。

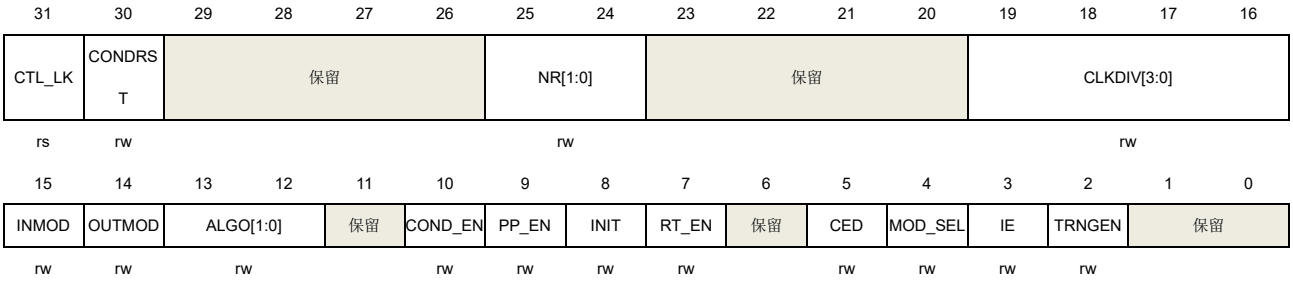
12.4. TRNG 寄存器

TRNG 基地址：0x4802 1800

12.4.1. 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。



位/位域	名称	描述
31	CTL_LK	TRNG_CTL 寄存器锁定位 该位仅能在模块复位时清 0 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。 注意：该 bit 由 1 到 0 期间需要至少保持 3 个分频后的 TRNG_CLK 时钟信号以确保配置数据同步
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

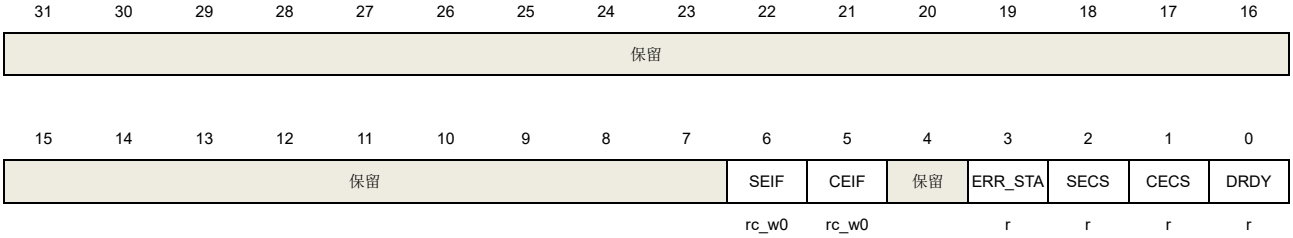
		0：向训练单元输入 256 比特 1：向训练单元输入 440 比特
14	OUTMOD	随机数输出模式选择 0：从训练单元输出 128 比特 1：从训练单元输出 256 比特
13:12	ALGO[1:0]	训练单元算法选择 00：sha1 算法 01：md5 算法 10：sha224 算法 11：sha256 算法
11	保留	必须保持复位值。
10	COND_EN	启动训练单元 0：失能训练单元 1：使能训练单元
9	PP_EN	启动后处理功能 0：失能后处理功能 1：使能后处理功能
8	INIT	使能训练单元时初始化哈希算法 0：不初始化哈希算法 1：初始化哈希算法
7	RT_EN	替换测试使能位 0：失能替换测试 1：使能替换测试
6	保留	必须保持复位值。
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	MOD_SEL	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 ERR_STA 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。

12.4.2. 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 被检测到或自适应比例测试期间超出 APT_TH 阈值则此位将置 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	ERR_STA	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 被检测到或自适应比例测试期间超出 APT_TH 阈值时，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

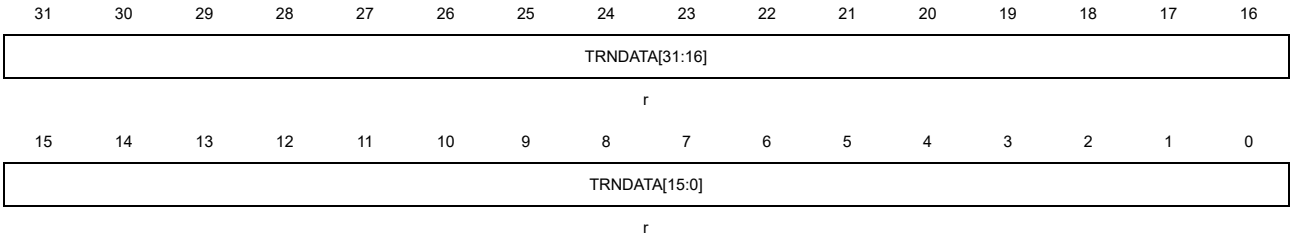
12.4.3. 数据寄存器（TRNG_DATA）

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。



位/位域	名称	描述
31:0	TRNDATA[31:0]	32 位随机数据

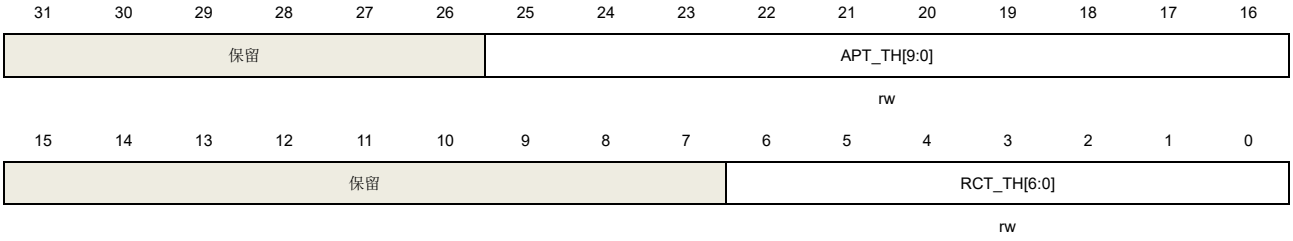
12.4.4. 健康测试配置寄存器（TRNG_HTCFG）

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。



位/位域	名称	描述
31:26	保留	必须保持复位值。
25:16	APT_TH[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	RCT_TH[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

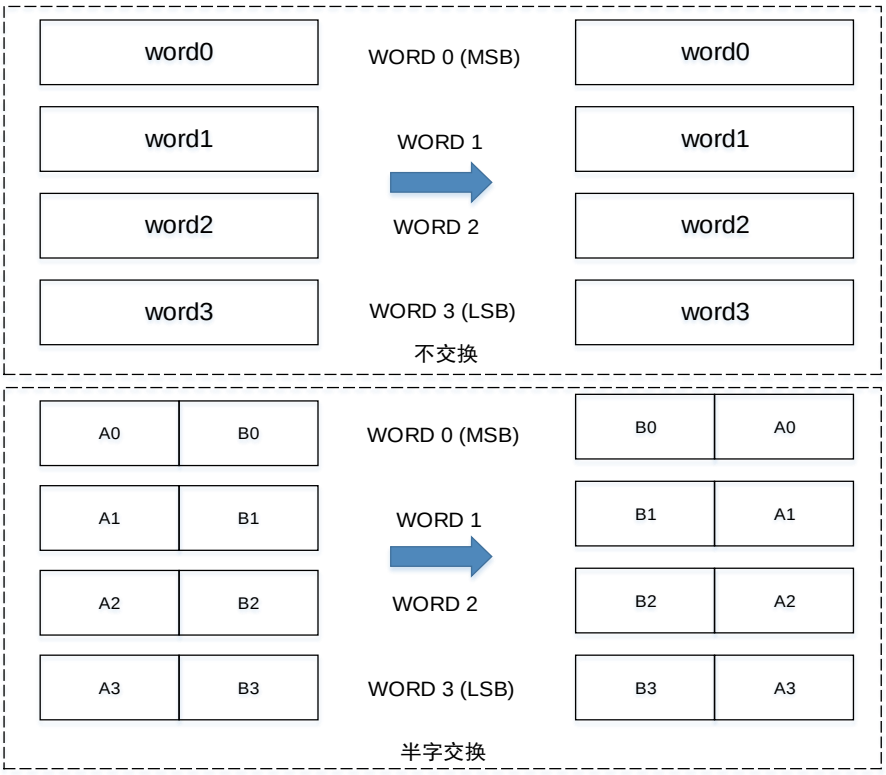


图 13-2. DATATM 字节交换/位交换



13.3.2. 初始化向量

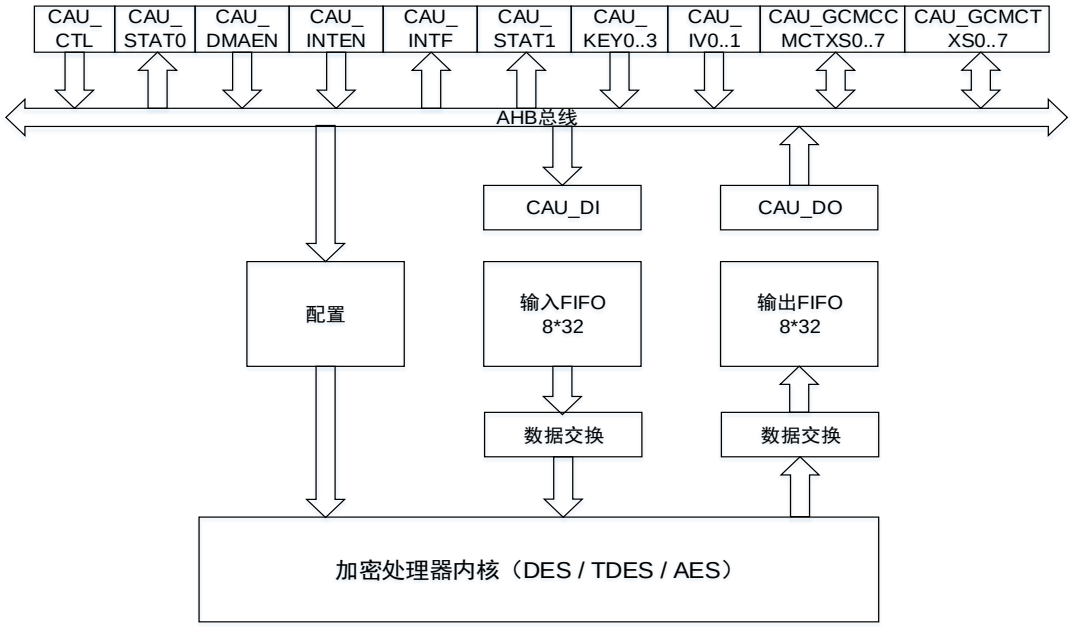
初始化向量用于在CBC、CTR、GCM、GMAC、CCM、CFB和OFB模式下与数据块进行异或。初始化向量与明文或密码数据无关，而且它们不受DATAM值的影响。注意初始化向量寄存器CAU_IV0..1（H/L）只有在BUSY位（CAU_STAT0寄存器位4）为0时才能被修改，否则写操作都是无效的。

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 / TDES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）。

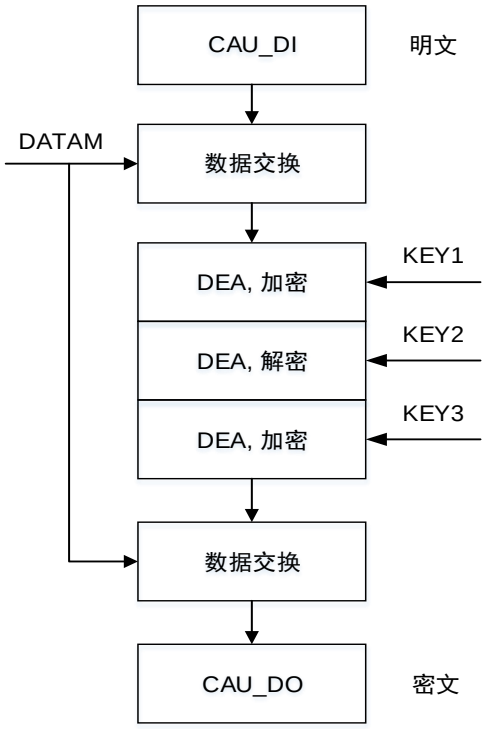
FIPS PUB 46-3（以及ANSI X9.52-1998）对DES / TDES中密钥的使用进行了详尽的解释，在本手册中不进行赘述。

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*。

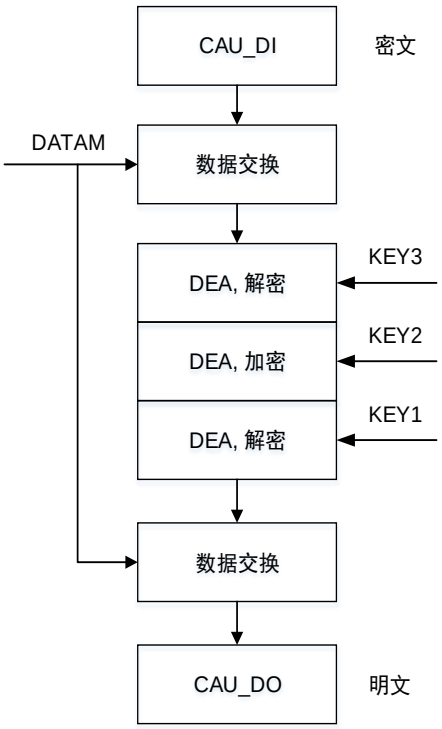
图 13-4. DES / TDES ECB 加密



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*。

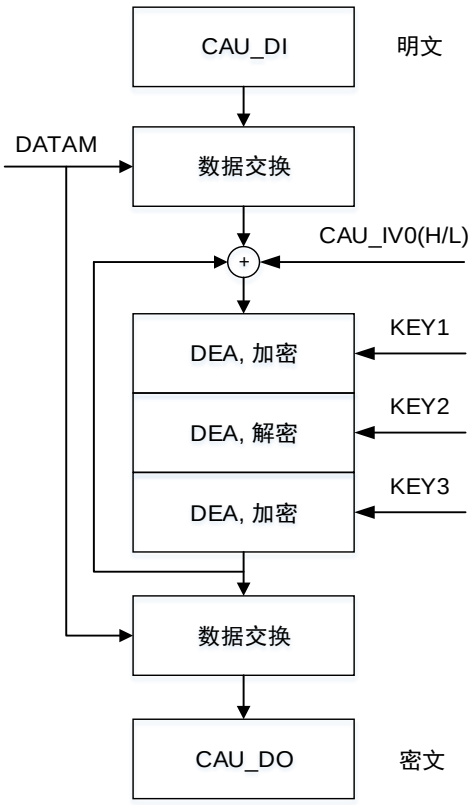
图 13-5. DES / TDES ECB 解密



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*。

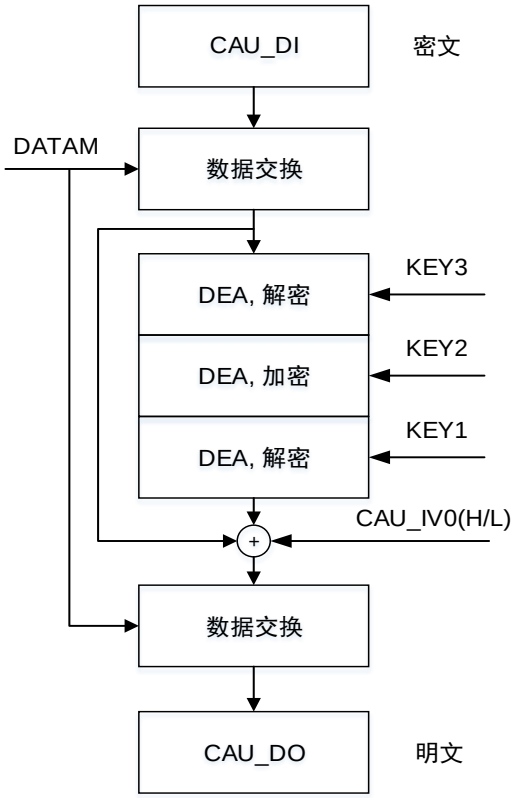
图 13-6. DES / TDES CBC 加密



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。

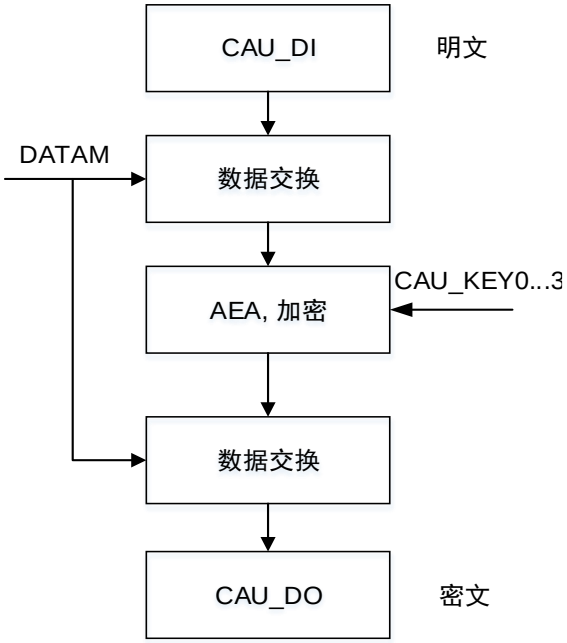
使用128位密钥时AES密钥为[KEY3 KEY2]，使用192位密钥时AES密钥为[KEY3 KEY2 KEY1]，使用256位密钥时AES密钥为[KEY3 KEY2 KEY1 KEY0]。

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*。

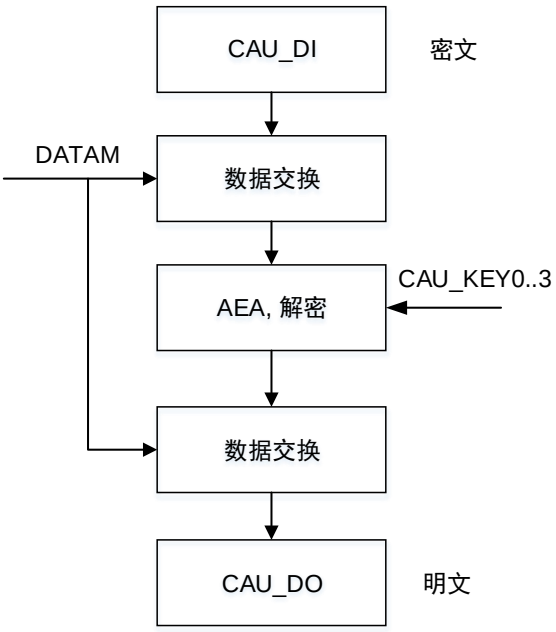
图 13-8. AES ECB 加密



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*。

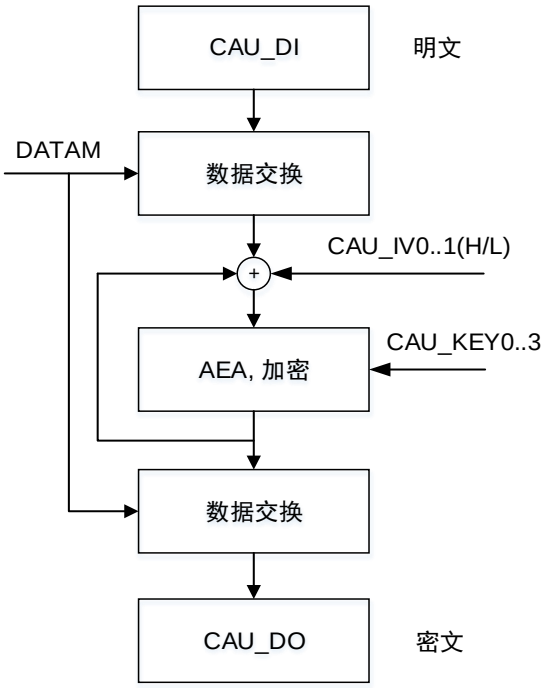
图 13-9. AES ECB 解密



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*。

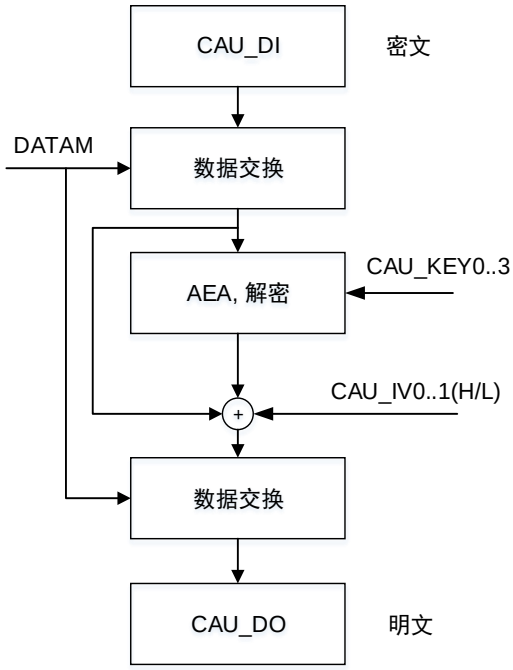
图 13-10. AES CBC 加密



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



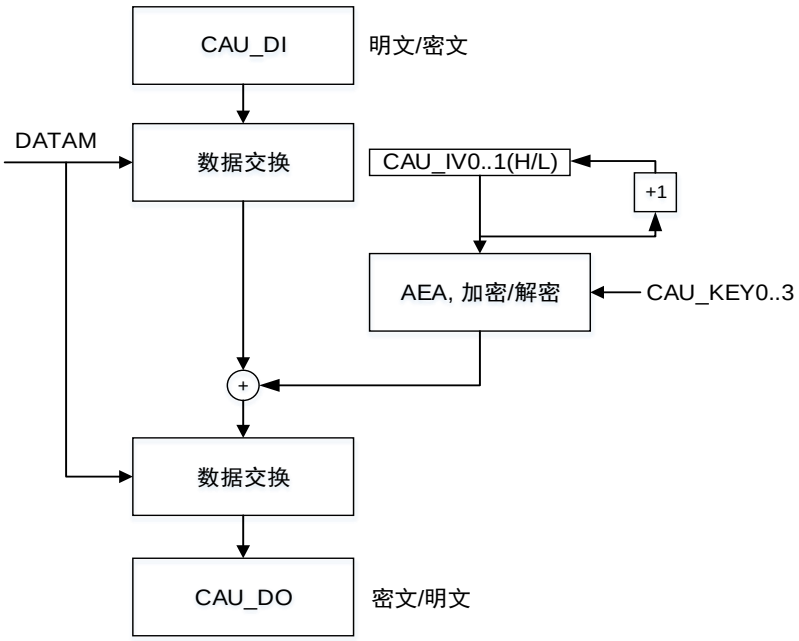
AES 计数器（CTR）模式

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，需首先准备好密钥。

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. 如果用户需要使用EFUSE中的AES密钥，跳转到第2步，否则跳转到第5步；
2. 将AES密钥按顺序（小端存储）写入到EFUSE_AES_KEYx寄存器中，检查写入的AES密钥的CRC校验值正确后将AES密钥烧写到EFUSE中；
3. 将EFUSE_USER_CTL寄存器中的AESEN位置1并将值烧写到EFUSE中；
4. 确保EFUSE_MCU_RSV寄存器中的AESNCAU位为0；
5. 将CAU_CTL寄存器的CAUEN位清零，以禁用CAU；
6. 如果使用来自于CAU模块寄存器中的AES密钥则跳过该步骤。如果使用的AES密钥来自于EFUSE，请确保EFUSE中写入了正确的AES密钥且CAU_CTL寄存器中的KEY_SEL位被置1；
7. 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，如果使用了来自于EFUSE中的AES密钥则跳过该步骤；
8. 将CAU_CTL寄存器中的KEY_SEL位清零，然后根据算法配置CAU_KEY0..3(H/L)寄存器，如果使用了来自于EFUSE中的AES密钥则跳过该步骤；
9. 设置CAU_CTL寄存器的DATAM位，配置数据交换类型；
10. 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）；
11. 设置CAU_CTL寄存器的CAUDIR位为0，配置为加密操作；
12. 设置CAU_IV0..1(H / L)寄存器，配置初始化向量；
13. 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；
14. 设置CAU_CTL寄存器的CAUEN位为1，使能CAU；
15. 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；
16. 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；
17. 重复步骤15和步骤16，直到所有的数据块都完成加密。

注意：当用户使用EFUSE中的AES密钥时（KEY_SEL置位），AES算法模式会被强制配置为AES-CTR模式（ALGM[3:0]位域被强制配置为0b0110），且AES密钥长度强制为128位。

解密

1. 将CAU_CTL寄存器的CAUEN位清零，以禁用CAU；
2. 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；
3. 将CAU_CTL寄存器中的KEY_SEL位清零，然后根据算法配置CAU_KEY0..3(H / L)寄存器，若使用了来自于EFUSE中的AES密钥，则跳过该步骤；

4. 设置CAU_CTL寄存器的DATAM位，配置数据交换类型；
5. 设置CAU_CTL寄存器的ALGM[3:0]位为“0111”，配置准备密钥用于解密；
6. 设置CAU_CTL寄存器的CAUEN位为1，使能CAU；
7. 等待BUSY位和CAUEN位为0，确保解密用的密钥已准备好；
8. 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）；
9. 设置CAU_CTL寄存器的CAUDIR位为1，配置为解密操作；
10. 设置CAU_IV0..1(H / L)寄存器，配置初始化向量；
11. 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；
12. 设置CAU_CTL寄存器的CAUEN位为1，使能CAU；
13. 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；
14. 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；
15. 重复步骤13和步骤14，直到所有的数据块都完成解密。

注意：当AESEN置位，AESNCAU为0且KEY_SEL被配置为1时，AES密钥长度强制为128位，且ALGM[3:0]位域被强制配置为0b0110。

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解密。需要注意的是，只有在倒数第二个数据块加解密完成后，才可以对NBPILB[3:0]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。

CAU DMA接口支持单数据传输和突发传输用以确保即使被传输数据的长度不是突发传输宽度的整数倍也能被正确传输。需要注意的是配置DMA控制器突发传输的长度应该不超过四个字以确保数据不会丢失。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。

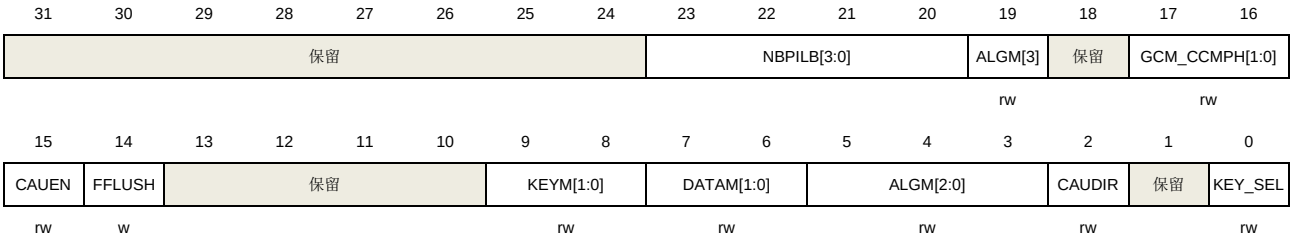
13.9. CAU 寄存器

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。



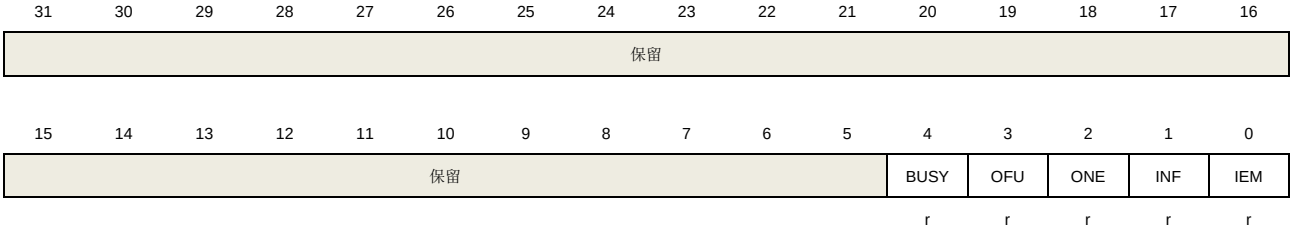
位/位域	名称	描述
31:24	保留	必须保持复位值。
23:20	NBPILB[3:0]	最后一个非 128 比特整数倍数据块的填充字节数 0000：所有数据有效（无填充） 0001：一个填充字节 … 1111：15 个填充字节
19	ALGM[3]	加密/解密算法模式位 3
18	保留	必须保持复位值。
17:16	GCM_CCMPH[1:0]	GCM CCM 阶段 00：准备阶段 01：AAD 阶段 10：加密解密阶段 11：标签阶段
15	CAUEN	加密处理器使能 0：加密处理器禁用 1：加密处理器使能 **注意**：当准备密钥（ALGM=0111b）完成后，CAUEN 位将硬件自动清零。
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	保留	必须保持复位值。
0	KEY_SEL	密钥来源选择 0：使用来自于 CAU_KEY0..3(H/L)寄存器的密钥 1：使用来自于 EFUSE 的密钥

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。

31	30	29	28	27	26	25	24	23	22	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														DMAOEN	DMAIEN
														rw	rw

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。

31	30	29	28	27	26	25	24	23	22	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														OINTEN	IINTEN
														rw	rw

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。

31	30	29	28	27	26	25	24	23	22	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														OSTA	ISTA

r r

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 to 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 to 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
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或GMAC模式下使用。

14. 哈希处理器（HAU）

14.1. 简介

哈希处理器应用于信息安全。支持应用于多种场合的安全哈希算法（SHA-1，SHA-224和SHA-256），消息摘要算法（MD5）和哈希运算消息认证码（HMAC）。对长达（2^{64}-1）位的消息，哈希处理器计算消息摘要长度对应于SHA-1，SHA-224，SHA-256，和MD5算法分别为160位，224位，256位，128位。而在HMAC算法中，SHA-1、SHA-224、SHA-256或MD5将作为哈希函数被调用两次，来产生验证消息。

哈希处理器完全兼容下列标准：

- 联邦信息处理标准出版物180-4（FIPS PUB 180-4）；
- 安全散列标准规范（SHA-1，SHA-224，SHA256）；
- 互联网工程任务组征求意见文档编号1321（IETF RFC 1321）规范（MD5）。

14.2. 主要特征

- 32位AHB从外设；
- 高性能的哈希算法运算；
- 小端数据表示；
- 支持多种数据交换类型，包括32位字不交换，半字交换，字节交换和位交换；
- 可自动填充来适应模数为512位（16×32位）消息摘要的计算；
- 支持DMA模式的数据流传输；
- 哈希/HMAC计算挂起模式。

14.3. 数据类型

哈希处理器每次接收32位字，但每次计算处理一个512位块。对每个输入字，在送入哈希内核之前都会根据数据类型进行位/字节/半字/不交换。同样在数据输出之前也要进行相同的数据交换。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。SHA-1，SHA-224，SHA-256的计算均为大端模式。

*图14-1. DATAM 不交换/半字交换*和*图14-2. DATAM 字节交换/位交换*介绍了在不同数据类型下的数据交换。

图 14-1. DATAM 不交换/半字交换

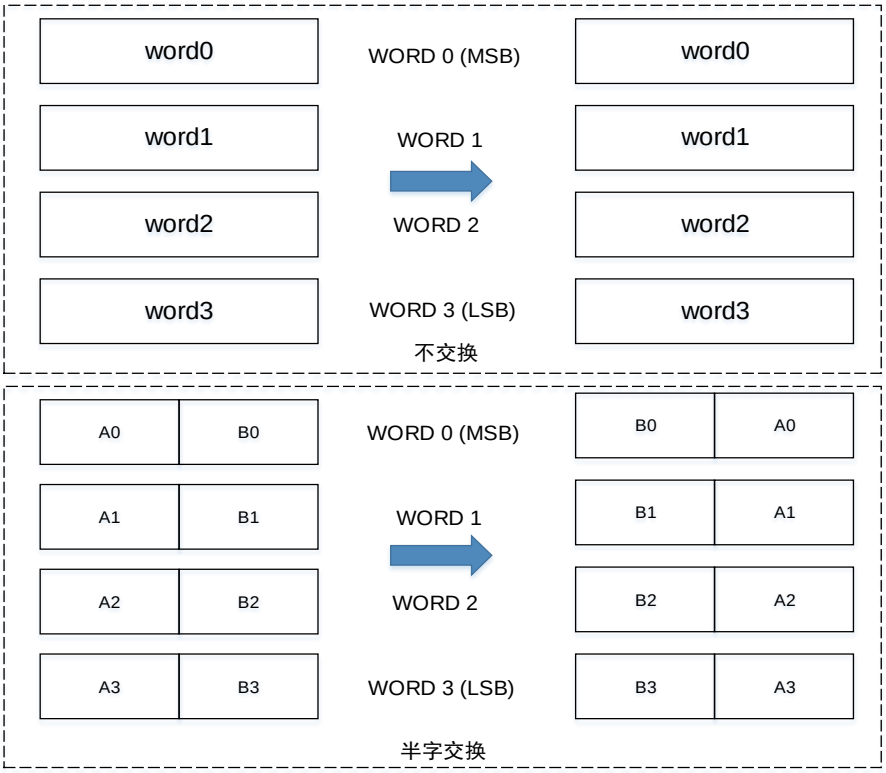
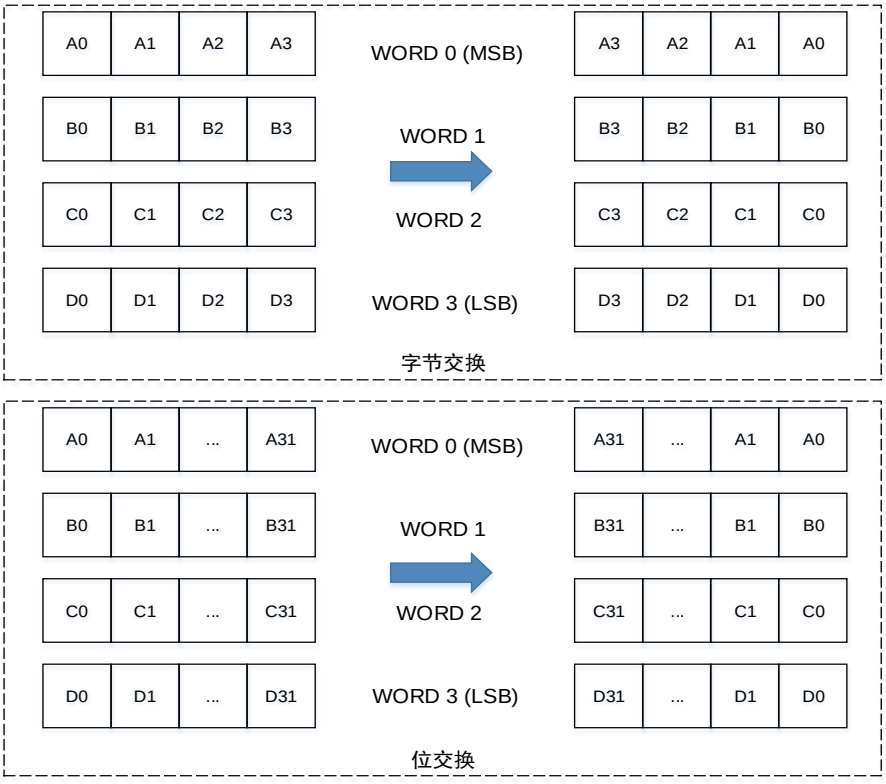


图 14-2. DATAM 字节交换/位交换

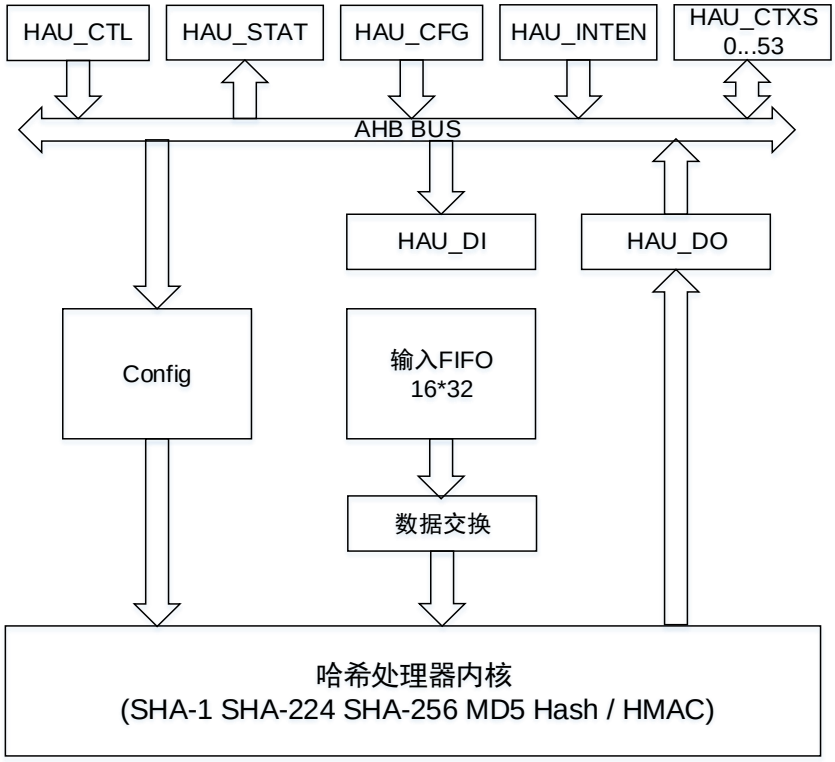


14.4. HAU 内核

哈希处理器使用安全哈希算法对输入消息进行信息压缩计算。对长度为（2^{64}-1）位的消息摘要计算结果的长度对应于SHA-1，SHA-224，SHA-256，和MD5算法分别为160位，224位，256位，128位。哈希处理器可用于生成和验证消息签名，并由于摘要远远小于消息的大小而具有更高的效率。

要由哈希处理器处理的消息应视为位串。消息长度为消息的位数。哈希处理器可以确保信息的安全，因为根据某个给定消息摘要来寻找原对应的消息在计算层面是无法实现的，而在原输入消息上任何的改动都将导致生成完全不同的消息摘要。

图 14-3. HAU 结构框图



14.4.1. 自动数据填充

为确保输入HAU内核的数据为512位的整数倍，需要对输入消息进行填充。消息填充操作由在原始消息的结尾添加一个1，后跟几个0和一个64位整数，填充物（0）将消息填充到整个512位的前448位，实现生成一个长度为512的填充消息块。

消息填充完成后，通过配置HAU_CFG寄存器的VBL位域来设置上面所述的64位整数。设置HAU_CFG寄存器的CALEN位为1，开始计算上个数据块的摘要。

数据填充示例：输入消息为“HAU”，对应的ASCII码16进制表示为

484155

接着根据消息的有效位长度，设置HAU_CFG寄存器的VBL位域为24。接着在位串的第24位处

添加一个”1”，随后填充数个“0”使位串模数为448，十六进制结果如下所示：

48415580 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000

之后，添加64位整数到已填充的输入消息后，该64位整数十六进制值为18，则最后的结果应该是：

48415580 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000018

14.4.2. 摘要计算

数据填充完成之后，通过DMA或CPU每次将512位的数据块送入HAU内核，HAU对每个数据块进行计算。在HAU内核开始计算之前，外设需要知道HAU_DI寄存器是否包含消息的最后一位。这可以从输入FIFO的状态和HAU_DI寄存器来确认。

通过DMA传输数据：

数据块传输的状态将自动通过DMA控制器发送的信息来解释。当HAU_CFG寄存器的CALEN位置1时，将自动开始进行数据填充和摘要计算。

注意：如果消息是个大文件并需要多个DMA传输，则应将MDS位置1。另外在传输之前需要设置VBL位域。在DMA的传输完成之后硬件不会自动将CALEN位置1，以便可以接收新的DMA传输。在最后的DMA传输期间，需要将MDS位清零，从而在最后一个块传输结束时硬件自动将CALEN位置1。

若消息不需要多个DMA传输，则将MDS置0即可，这样在一个DMA传输完成之后就会硬件自动置位CALEN位。同样的，在DMA传输之前也需要先设置VBL位域。

通过CPU传输数据：

- 当HAU_DI寄存器中写入下一个数据块的第一个字时，将开始计算当前数据块的摘要；
- 将HAU_CFG寄存器中CALEN位置1，将开始最后一个数据块的摘要计算。

14.4.3. 哈希模式

将HAU_CTL寄存器的HMS位设为0，选择为哈希模式。则当HAU_CTL寄存器的START位为1时，将根据ALGM位域的配置选择SHA-1，SHA-224，SHA-256或MD5算法进行计算。

当从HAU_DI寄存器和接收FIFO中接收到512位的消息块时，将根据DMA和CALEN位状态开始摘要的计算。

最终的计算结果可以从HAU_DO0...7寄存器中读取。

14.4.4. HMAC 模式

HMAC模式根据用户所选的密钥来进行消息验证。更多关于HMAC规范的信息请参阅“HMAC：密钥散列消息认证，H. Krawczyk，M. Bellare，R. Canetti，1997年2月”。

HMAC算法表示如下：

HMAC(input) = HASH[((key | opad) XOR 0x5c) | HASH(((key | ipad) XOR 0x36) | input)]

其中“ipad”和“opad”用于将密钥用数个“0”进行填充扩展到512位，“|”为连接符。

HMAC模式需要四个不同阶段：

1. 将HAU_CTL寄存器的HMS位置1，并根据期望的算法设置ALGM位域。若密钥“key”长度超过64个字节，则还需配置HAU_CTL寄存器的KLM位。之后，将START位置位以启动HAU内核；
2. 密钥“key”作为输入消息来进行哈希模式下的计算；
3. 当输入了最后一个字并开始计算之后，HAU生成新的密钥“key”作为内部哈希密钥；
4. 在第一次的哈希计算后，HAU内核开始接收用于外部哈希函数的密钥，通常外部的哈希函数使用与内部哈希密钥相同的新的密钥“key”。当输入了密钥的最后一个字，则开始进行计算，计算结果可从HAU_DO寄存器中读取。

14.5. HAU 挂起模式

HAU可以暂时挂起哈希或HMAC操作，从而先执行优先级更高的任务，在处理完优先级更高的任务后，再完成被挂起数据块的哈希或HMAC操作。

挂起任务前，必须将被挂起任务的上下文从寄存器保存到存储器，恢复任务时，再从存储器恢复到HAU寄存器。

以下说明由CPU或DMA传输数据时，按照下列的步骤来完成被挂起HAU任务的处理。

14.5.1. 通过 CPU 加载数据

1. 停止当前数据处理与传输。等待BUSY位为0，若NWIF[3:0]的值大于0，则需等待DIF位置位（若NWIF[3:0]的值等于0，则不等待DIF位置位）。只有在当前未处理任何块时才能保存上下文；
2. 保存当前配置。将HAU_INTEN，HAU_CFG，HAU_CTL，HAU_CTXS0到HAU_CTXS37（如果正在进行HMAC操作，则HAU_CTXS0到HAU_CTXS53）寄存器的内容保存到存储器中；
3. 配置并处理新消息；
4. 恢复之前的配置环境。将存储器中保存的值写入HAU_INTEN，HAU_CFG，HAU_CTL寄存器中；
5. 恢复消息的计算。置位HAU_CTL寄存器的START位来初始化重新开始新消息的摘要计算；
6. 恢复之前的内核状态。将存储器中保存的值写入HAU_CTXS0到HAU_CTXS37寄存器中

（如果涉及HMAC操作，则HAU_CTXS0到HAU_CTXS53）；

7. 从之前挂起的地方继续处理。

14.5.2. 通过 DMA 加载数据

1. 等待BUSY位为0，此时若HAU_STAT寄存器的CCF位置位，则不需要后续的上下文交换，否则再等待BUSY位为1；
2. 停止当前数据传输。禁能DMA的数据传输，再将HAU_CTL寄存器中的DMAE位清零以禁能DMA请求；
3. 保存当前配置。等待BUSY位为0，此时若HAU_STAT寄存器的CCF位置位，则不需要后续的上下文交换，否则将HAU_INTEN，HAU_CFG，HAU_CTL，HAU_CTXS0 到 HAU_CTXS37（如果正在进行HMAC操作，则HAU_CTXS0 到 HAU_CTXS53）寄存器的内容保存到存储器中；
4. 配置并处理新消息；
5. 恢复之前的环境配置，将存储器中保存的值写入HAU_INTEN，HAU_CFG，HAU_CTL寄存器中；
6. 恢复DMA通道传输。重新配置DMA通道以继续数据传输；
7. 恢复消息的计算。置位HAU_CTL寄存器的START位来初始化重新开始新消息的摘要计算。
8. 恢复之前内核的状态。将存储器中保存的值写入HAU_CTXS0到HAU_CTXS37寄存器中（如果涉及HMAC操作，则HAU_CTXS0 到 HAU_CTXS53）；
9. 置位HAU_CTL寄存器的位DMAE，从之前挂起的地方继续处理。

注意：如果HAU_CTL寄存器的位NWIF[3:0]为0，则说明上下文交换发生在在两个块之间，上一个数据块已完全处理，并且下一个块还未推入到输入FIFO，那么此时不需要保存和恢复HAU_CTXS22到HAU_CTXS37寄存器。

14.6. HAU 中断

HAU具有两个独立的中断源，并在HAU_STAT有相应的状态位。这两个状态位用于指示输入FIFO的状态，以及摘要的计算是否完成。

HAU_INTEN寄存器为中断使能寄存器。将相应位置1可以使能中断。

14.6.1. 输入 FIFO 中断

当输入FIFO中的数据处理已完成时，输入FIFO标志位DIF置位。如果置位DIIE位使能了输入FIFO中断，则当输入FIFO标志位DIF置位时会发生输入FIFO中断。

14.6.2. 计算完成中断

当摘要计算完成时，计算完成标志位CCF将置位。如果置位CCIE位使能了计算完成中断，则当计算完成标志位CCF置位时会发生计算完成中断。

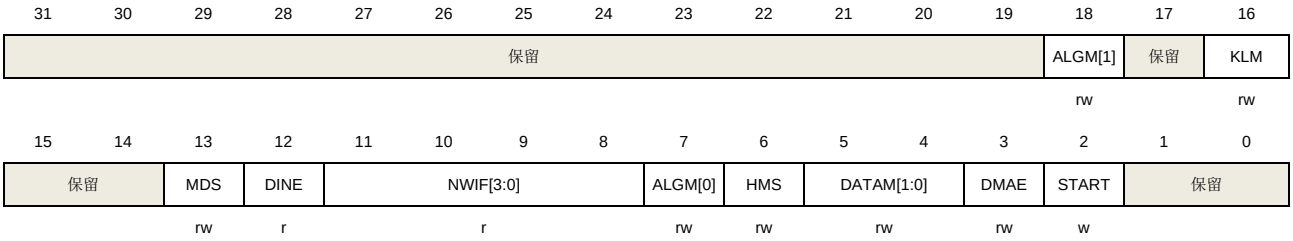
14.7. HAU 寄存器

HAU基地址：0x4802 1400

14.7.1. 控制寄存器（HAU_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
31:19	保留	必须保持复位值。
18	ALGM[1]	算法选择位1
17	保留	必须保持复位值。
16	KLM	密钥长度模式 0：密钥长度 ≤ 64 字节 1：密钥长度 > 64 字节 **注意**：必须在非计算期间修改该位
15:14	保留	必须保持复位值。
13	MDS	多DMA选择 如果哈希消息为大型文件需要多个DMA传输时，将此位置1 0：仅需要单次DMA传输，在DMA传输结束时硬件自动将CALEN位置1 1：需要多次DMA传输，在DMA传输结束时硬件不自动将CALEN位置1
12	DINE	DI寄存器非空 0：DI寄存器空 1：DI寄存器非空 **注意**：当START位或CALEN位为1时此位会清零
11:8	NWIF[3:0]	输入FIFO中的字数 **注意**：当START位置位时，或开始进行摘要计算时（CALEN位置位，或者DMA传输结束），该位域清零
7	ALGM[0]	算法选择位0 该位和CTL寄存器的位18用于选择SHA-1，SHA-224，SHA256或MD5 算法： 00：选择SHA-1算法

		01：选择MD5算法 10：选择SHA224算法 11：选择SHA256算法
6	HMS	HAU模式选择，必须在非计算期间修改该位 0：选择HASH模式 1：选择HMAC模式。如果密钥长度大于64字节，则还需配置KLM位。
5:4	DATAM[1:0]	数据交换类型 定义输入到HAU_DI寄存器中的数据格式 00：不交换，写入到HAU_DI寄存器的数据将直接送入FIFO，不进行交换 01：半字交换。写入到HAU_DI寄存器的数据在送入FIFO前，需要进行半字交换 10：字节交换。写入到HAU_DI寄存器的数据在送入FIFO前，需要进行字节交换 11：位交换。写入到HAU_DI寄存器的数据在送入FIFO前，需要进行位交换
3	DMAE	DMA使能 0：禁止DMA传输 1：使能DMA传输 **注意：**1.当DMA传输消息的最后一个数据时，将由硬件清零该位。当START置位时，不会清零该位。 2.如果DMA正在传输数据，将该位写入0不会中止当前的传输，而直到当前传输结束或START位置为1之后，才会禁止传输。
2	START	开始摘要计算 0：没有影响 1：开始新消息的摘要计算 **注意：**读取该位将始终返回0
1:0	保留	必须保持复位值。

14.7.2. 数据输入寄存器（HAU_DI）

地址偏移：0x04
复位值：0x0000 0000

该数据输入寄存器用于将512位的数据块送入输入FIFO进行处理。当正在进行摘要计算时，所有对该寄存器的新的写访问将被延迟，直到计算完成。

该寄存器只能按字(32位)访问。

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

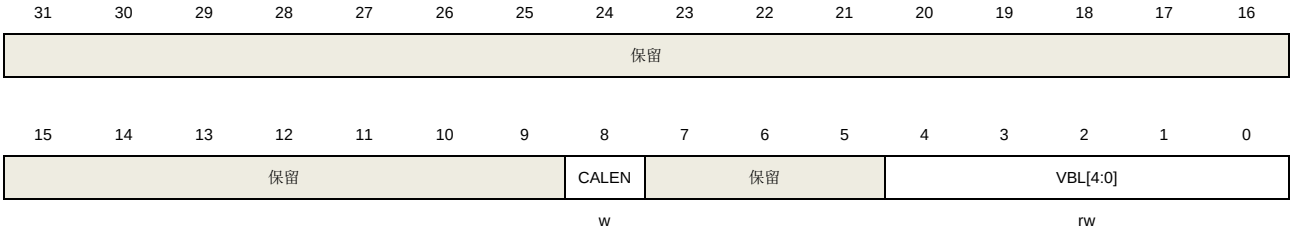
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DI[15:0]															
rw															

位/位域	名称	描述
31:0	DI[31:0]	消息数据输入 当数据写入这些寄存器时，寄存器中当前的内容被推入输入FIFO中同时更新为新的值。当读寄存器时，返回寄存器的当前内容。

14.7.3. 配置寄存器 (HAU_CFG)

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
31:9	保留	必须保持复位值。
8	CALEN	使能摘要计算 0：不计算 1：先使用VBL位域对数据进行数据填充，然后开始计算最终消息摘要 **注意：**读该位将返回0
7:5	保留	必须保持复位值。
4:0	VBL[4:0]	消息的最后一个字中的有效位数 0x00：对于写入HAU_DI寄存器的最后一个数据，所有32位（在数据交换后）均有效。 0x01：对于写入HAU_DI寄存器的最后一个数据，仅位[31]（在数据交换后）有效。 0x02：对于写入HAU_DI寄存器的最后一个数据，仅位[31:30]（在数据交换后）有效。 0x03：对于写入HAU_DI寄存器的最后一个数据，仅位[31:29]（在数据交换后）有效。 … 0x1F：对于写入HAU_DI寄存器的最后一个数据，仅位[31:1]（在数据交换后）有效。 **注意：**必须在置位CALEN位之前配置该位。

14.7.4. 数据输出寄存器 (HAU_DO0..7)

数据输出寄存器为只读寄存器，用于从输出FIFO中接收计算结果。置位START位将复位该寄存器。当正在进行摘要计算时，所有对该寄存器的新的读访问将被延迟，直到计算完成。

在SHA-1模式中，使用HAU_DO0…4

在MD5模式中，使用HAU_DO0…3

在SHA-224模式中，使用HAU_DO0…6

在SHA-256模式中，使用HAU_DO0…7

HAU_DO0

地址偏移：0x0C和0x310
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO0[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO0[15:0]															
r															

HAU_DO1

地址偏移：0x10和0x314
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO1[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO1[15:0]															
r															

HAU_DO2

地址偏移：0x14和0x318
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO2[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO2[15:0]															
r															

HAU_DO3

地址偏移：0x18和0x31C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

DO3[31:16]
r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO3[15:0]															
r															

HAU_DO4

地址偏移：0x1C和0x320

复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO4[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO4[15:0]															
r															

HAU_DO5

地址偏移：0x324

复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO5[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO5[15:0]															
r															

HAU_DO6

地址偏移：0x328

复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO6[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO6[15:0]															
r															

HAU_DO7

地址偏移：0x32C

复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO7[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO7[15:0]															
r															

位/位域	名称	描述
31:0	DO0..7[31:0]	消息摘要结果

14.7.5. 中断使能寄存器 (HAU_INTEN)

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														CCIE	DIIE
														rw	rw

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	CCIE	计算完成中断使能 0：禁止计算完成中断 1：使能计算完成中断
0	DIIE	数据输入中断使能 0：禁止数据输入中断 1：使能数据输入中断

14.7.6. 状态与标志寄存器 (HAU_STAT)

地址偏移：0x24

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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												BUSY	DMAS	CCF	DIF
												r	r	rc_w0	rc_w0

位/位域	名称	描述
31:4	保留	必须保持复位值。
3	BUSY	忙标志位 0：未处理任何块 1：正在处理某个数据块
2	DMAS	DMA状态标志 0：DMA接口被禁用(DMAE=0)并且未在进行任何传输 1：DMA接口被使能(DMAE=1)并且未在进行任何传输
1	CCF	计算完成状态标志 0：计算未完成 1：所有消息摘要计算完成
0	DIF	数据输入状态标志 0：有一个字数据写入数据输入寄存器 1：完成一个字数据的初步处理（只有在输入FIFO中的数据才会被处理）

14.7.7. 上下文交换寄存器 x (HAU_CTXSx) (x=0...53)

地址偏移：0xF8 + 0x04 × x, (x = 0...53)
复位值：0x0000 0002（当x=0），0x0000 0000（当x = 1…53）

该寄存器只能按字(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]	HAU处理器完整的内部状态信息。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。

15. 三角函数加速器（TMU）

15.1. 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，输入和输出数据符合 q1.31 或者 q1.15 格式。

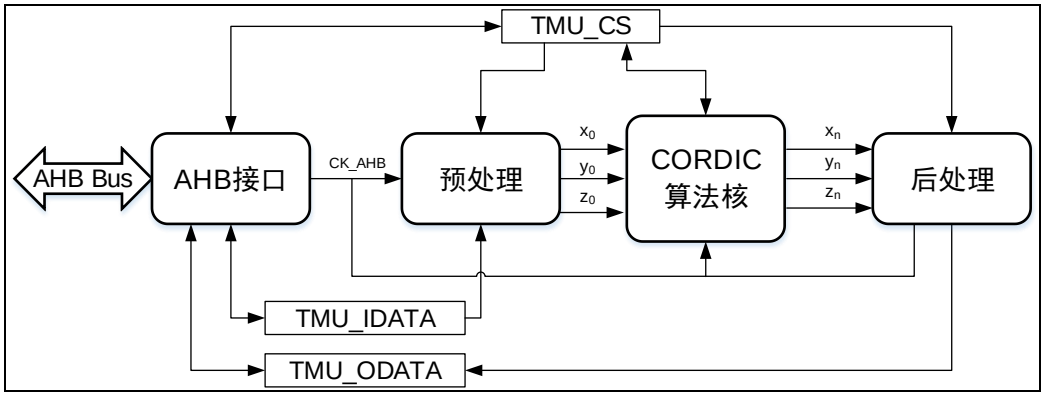
15.2. 主要特性

- 10 种函数；
- 中断和 DMA 请求；
- 定点格式可配置；
- 精度编程；
- 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。

15.3. 结构框图

图 15-1. TMU 模块结构框图 TMU 模块内部结构细节。

图 15-1. 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内容是 q1.31 或者 q1.15 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内容是 q1.31 或者 q1.15 格式。

15.4. 功能描述

15.4.1. 数据格式和配置

TMU 模块的输入数据和输出数据是定点有符号整型格式（q1.31 和 q1.15 格式）。

Q1.31 格式第 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]。

TMU_CS 寄存器的 IWITH 位用来配置输入数据的定点格式。有的模式（例如，模式 0）需要两个输入数据，有的模式只需要一个输入数据，TMU_CS 寄存器的 INUM 位用来配置输入数据的数量。详细配置参考*表 15-1. 输入数据配置*。

注意：当输入数据配置为 q1.15 格式，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 15-1. 输入数据配置

IWIDTH 位	INUM 位	定点格式	写 TMU_IDATA 寄存器
0	0	q1.31	写一次
0	1	q1.31	连续写两次
1	0	q1.15	写一次
1	1	q1.15	不可用

TMU_CS 寄存器的 OWITH 位用来配置输出数据的定点格式。有的模式（例如，模式 0）有两个输出数据，有的模式只有一个输出数据，TMU_CS 寄存器的 ONUM 位用来配置输出数据的数量。详细配置参考*表 15-2. 输出数据配置*。

注意：当输出数据配置为 q1.15 格式，则只需要读一次 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 15-2. 输出数据配置

OWIDTH 位	ONUM 位	定点格式	读 TMU_ODATA 寄存器
0	0	q1.31	读一次
0	1	q1.31	连续读两次
1	0	q1.15	读一次
1	1	q1.15	不可用

15.4.2. 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 15-3. TMU 模式配置*。由于输入和输出数据都是 q1.31 或者 q1.15 格式，所以有些模

式需要对实际输入参数进行缩放。TMU_CS 寄存器的 FACTOR[2:0]位域用来配置缩放因子。

表 15-3. 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	tan2 (y,x)	$\sqrt{x^2+y^2}$	圆周系统，向量模式
模式 3	x	y	$\sqrt{x^2+y^2}$	tan2 (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。有两个输入和两个输出，参考*表 15-4. 模式 0 描述*。

表 15-4. 模式 0 描述

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 或者 q1.15 格式写入 TMU_IDATA 寄存器
第二个输入数据	m ∈ [0,1)	当0 ≤ m < 1时，按照 q1.31 或者 q1.15 格式写入 TMU_IDATA 寄存器。当m ≥ 1时，软件缩小m到[0,1)范围内，按照 q1.31 或者 q1.15 格式写入 TMU_IDATA 寄存器。
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	$m * \sin(\theta)$,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。
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，缩放比例是自行选择的。

例如，计算$100 * \cos\left(\frac{\pi}{2}\right)$，默认输入输出配置为 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 格式为 0x4000。
2. 软件处理模长m。$\frac{100}{128}=0.78125$，q1.15 格式为 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$y_2=\frac{100}{128}*\sin\left(\frac{\pi}{2}\right)$。输出数据是 q1.15 格式。
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。有两个输入和两个输出，参考*表 15-5. 模式 1 描述*。

表 15-5. 模式 1 描述

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 或者 q1.15 格式写入 TMU_IDATA 寄存器
第二个输入数据	m ∈ [0,1)	当0 ≤ m < 1时，按照 q1.31 或者 q1.15 格式写入 TMU_IDATA 寄存器。当m ≥ 1时，软件缩小m到[0,1)范围内，按照 q1.31 或者 q1.15 格式写入 TMU_IDATA 寄存器。
第一个输出数据	$m*\sin(\theta)\in$[-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	$m*\cos(\theta)\in$[-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。
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，缩放比例是自行选择的。

例如，计算$100*\sin\left(\frac{\pi}{2}\right)$，默认输入输出配置为 q1.15 格式。可以按照以下步骤进行：

1. 软件处理角度$\frac{\pi}{2}$。$\frac{\frac{\pi}{2}}{\pi}=0.5$，q1.15 格式为0x4000。
2. 软件处理模长m。$\frac{100}{128}=0.78125$，q1.15 格式为 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$y_2=\frac{100}{128}*\cos\left(\frac{\pi}{2}\right)$。输出数据是 q1.15 格式。
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函数。有两个输入和一个输出，参考*表 15-6. 模式 2 描述*。

表 15-6. 模式 2 描述

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：

1. x和y只要有一个超出范围[-1,1)，需要同时对x和y进行同比例缩放，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。
2. 当$\sqrt{x^2+y^2}\geq1$时，模长 m都只能饱和到定点格式的最大值（$1-2^{-15}$或者$1-2^{-31}$）。对x和y进行同比例缩放前，要考虑缩放因子的大小，尽量避免出现模长饱和的情况。

例如，计算θ=atan(5,80)，默认输入输出配置为 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，得(0.0390625,0.625)，q1.15 表达形式为(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（此处角度为 q1.15 格式），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=atan(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。有两个输入和两个输出，参考*表 15-7. 模式 3 描述*。

表 15-7. 模式 3 描述

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。
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：

1. x和y只要有一个超出范围[-1,1)，需要同时对x和y进行同比例缩放，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。
2. 当$\sqrt{x^2+y^2}$≥1时，模长 m都只能饱和到定点格式的最大值（$1-2^{-15}$或者$1-2^{-31}$）。对x和y进行同比例缩放前，要考虑缩放因子的大小，尽量避免出现模长饱和的情况。

例如，计算$\sqrt{5^2+80^2}$，默认输入输出配置为 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，得(0.0390625,0.625)，q1.15 表达形式为(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（此处角度为 q1.15 格式）。
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。有一个输入和一个输出，参考*表 15-8. 模式 4 描述*。

表 15-8. 模式 4 描述

参数	范围	描述
输入数据	$\frac{x}{2^f} \in [-1,1)$	如果x∈[-1,1]，软件不需要处理，缩放因子 FACTOR[2:0] =3'b000。如果x超出[-1,1]范围，软件进行缩放，缩放后要保证$-1 \le x*2^{-f} < 1$，把f写入缩放因子 FACTOR[2:0]位域，把缩放后的数据$\frac{x}{2^f}$以 q1.15 或者 q1.31 写入 TMU_IDATA 寄存器。
输出数据	$\frac{\theta}{2^f} \in [-1,1)$	[-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

例如，计算$\tan^{-1}(100)$，默认输入输出配置为 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 表达形式为 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$\frac{\theta}{2^f}$，输出数据为 q1.15 格式。
5. 结果处理。输出数据$\frac{\theta}{2^f}$需要乘以π和128以得到真实弧度。

模式 5: cosh (x)

该模式用来计算cosh (x)函数。有一个输入和两个输出，参考*表 15-9. 模式 5 描述*。

表 15-9. 模式 5 描述

参数	范围	描述
输入数据	$\frac{x}{2} \in [-0.559,0.559]$	x∈[-1.118,1.118]，软件将x除以 2，然后以 q1.15 或者 q1.31 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：缩放因子 FACTOR[2:0]只能配置为 3'b001。

例如，计算cosh (1.0)，默认输入输出配置为 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)，得0.5，q1.15 表达形式为0x4000。
2. 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$y_1=\frac{\sinh(1.0)}{2}$。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。有一个输入和两个输出，参考*表 15-10. 模式 6 描述*。

表 15-10. 模式 6 描述

参数	范围	描述
输入数据	$\frac{x}{2}\in$[-0.559,0.559]	x∈[-1.118,1.118]，软件将x除以 2，然后以 q1.15 或者 q1.31 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：缩放因子 FACTOR[2:0]只能配置为 3'b001。

例如，计算sinh (1.0)，默认输入输出配置为 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)，得0.5，q1.15 表达形式为0x4000。
2. 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 (1.0)和双曲余弦cosh (1.0)。

模式 7: $\tanh^{-1}$ (x)

该模式用来计算$\tanh^{-1}$ (x)函数。有一个输入和一个输出，参考*表 15-11. 模式 7 描述*。

表 15-11. 模式 7 描述

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 或者 q1.31 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)。

参数	范围	描述
缩放因子 FACTOR[2:0]	f=1	FACTOR[2:0]配置为 3'b001

注意：缩放因子 FACTOR[2:0]只能配置为 3'b001。

例如，计算$\tanh^{-1}(0.5)$，默认输入输出配置为 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)，得0.25，q1.15 表达形式为0x2000。
2. 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$y_1=\frac{\tanh^{-1}(0.5)}{2}$。输出数据是 q1.15 格式。
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。有一个输入和一个输出，参考*表 15-12. 模式 8 描述*。

表 15-12. 模式 8 描述

参数	范围	描述
输入数据	$\frac{x}{2^f}\in[0.0535,0.875]$	x∈[0.107,9.35]。软件进行缩放处理，保证$\frac{x}{2^f}<(1-\frac{1}{2^f})$，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	$\frac{\ln(x)}{2^{(f+1)}}\in[-0.558,0.137]$	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

例如，计算ln (8)，默认输入输出配置为 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），得0.5，q1.15 表达形式为0x4000。
2. 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$y_1=\frac{\ln(x)}{2^{(4+1)}}$。输出数据是 q1.15 格式。
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 15-13. 模式 8 推荐的缩放因子*中的缩放因子。

表 15-13. 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
$0.107 \le x < 1$	3'b001	[0.0535,0.5)
$1 \le x < 3$	3'b010	[0.25,0.75)
$3 \le x < 7$	3'b011	[0.375,0.875)

输入x范围	缩放因子 FACTOR[2:0]	输入数据范围
$7 \le x < 9.35$	3'b100	[0.4375,0.584)

模式 9: $\sqrt{x}$

该模式用来计算$\sqrt{x}$函数。有一个输入和一个输出，参考*表 15-14. 模式 9 描述*。

表 15-14. 模式 9 描述

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	$\frac{\sqrt{x}}{2^f} \in [0.04,1]$	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

例如，计算$\sqrt{2}$，默认输入输出配置为 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），得0.5，q1.15 表达形式为0x4000。
2. 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$y_1=\frac{\sqrt{2}}{2^2}$。输出数据是 q1.15 格式。
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 15-15. 模式 9 推荐的缩放因子*中的缩放因子。

表 15-15. 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)

15.4.3. TMU 精度

表 15-16. 不同迭代次数下的精度

模式	迭代次数	计算周期数	最大残差 (1)	
			q1.31 格式	q1.15 格式
模式 0，模式 1，模式(2) 2，模式(2) 3，模式(4) 4	4	1	2^{-3}	2^{-3}
	8	2	2^{-7}	2^{-7}
	12	3	2^{-11}	2^{-11}
	16	4	2^{-15}	2^{-15}
	20	5	2^{-18}	2^{-16}

模式	迭代次数	计算周期数	最大残差 (1)	
			q1.31 格式	q1.15 格式
	24	6	2^{-19}	2^{-16}
模式 5，模式 6，模式 7，模式(3) 8	4	1	2^{-2}	2^{-2}
	8	2	2^{-6}	2^{-6}
	12	3	2^{-10}	2^{-10}
	16	4	2^{-13}	2^{-13}
	20	5	2^{-17}	2^{-15}
	24	6	2^{-18}	2^{-15}
模式(4) 9	4	1	2^{-7}	2^{-7}
	8	2	2^{-14}	2^{-14}
	12	3	2^{-19}	2^{-15}

1. 最大剩余误差是在给定次数的迭代后，与在双精度浮点中执行的相同计算相比，剩余的最大误差。可能会产生额外的舍入误差，对于 q15 格式最多为 2^{-16}，对于 q31 格式最多为 2^{-20}。
2. 当坐标(x,y)靠近(0,0)时，精度会急剧下降。
3. FACTOR[2:0] = 1。如果使用更高的比例因子，则可实现的精度会成比例地降低。
4. FACTOR[2:0] = 0。如果使用更高的比例因子，则可实现的精度会成比例地降低。

15.4.4. 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。

15.4.5. 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。

15.4.6. 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。

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。

15.5. TMU 寄存器

TMU 基地址：0x4001 0000

15.5.1. 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	保留								IWIDTH	OWIDTH	INUM	ONUM	WDEN	RDEN	RIE
r									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					FACTOR[2:0]			保留				MODE[3:0]			
					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。
30:23	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。
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。
20	INUM	写TMU_IDATA寄存器的次数 0：一次 32-bit 写操作。一次 32-bit 写 TMU_IDATA 操作可以启动一次 TMU 运算。 1：两次 32-bit 写操作。两次连续的 32-bit 写 TMU_IDATA 操作可以启动一次 TMU 运算。 **注意**：当输入数据格式为q1.15（IWIDTH=1）并且TMU模式只需要一个输入数据（INUM=0），TMU_IDATA的高半字不使用。
19	ONUM	写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OWIDTH=1（输出数据格式为q1.15）,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。
15:11	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：

1. 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。
2. 输入数据（TMU_IDATA）和输出数据（TMU_ODATA）是 q1.31 或者 q1.15 格式的。

7:4	保留	必须保持复位值。
3:0	MODE[3:0]	TMU模式 0000：模式0, $m * cos(\theta)$ 0001：模式1, $m * sin(\theta)$ 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

15.5.2. 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 15-1. 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。

15.5.3. 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15-2. 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。

16. 直接存储器访问控制器（DMA）

16.1. 简介

DMA 控制器提供了一种硬件的方式在外设和存储器之间或者存储器和存储器之间传输数据，而无需 MCU 的介入，避免了 MCU 多次进入中断进行大规模的数据拷贝，最终提高整体的系统性能。

每个 DMA 控制器包含了两个 AHB 总线接口和 8 个 4 字深度的 FIFO，使 DMA 可以高效的传输数据。DMA 控制器共有 16 个通道（DMA0，DMA1 控制器分别有 8 个通道），每个通道可以被分配给一个或多个特定的外设用于存储器访问请求管理。两个内置的总线仲裁器用来处理 DMA 请求的优先级问题。

DMA 控制器与 Cortex-M7 内核都是通过系统总线来处理数据，引入仲裁机制来处理它们之间的竞争关系。当 MCU 和 DMA 指定相同的外设的时候，MCU 将会在特定的总线周期挂起。总线矩阵使用了轮询的算法保证 MCU 至少占用了一半的带宽。

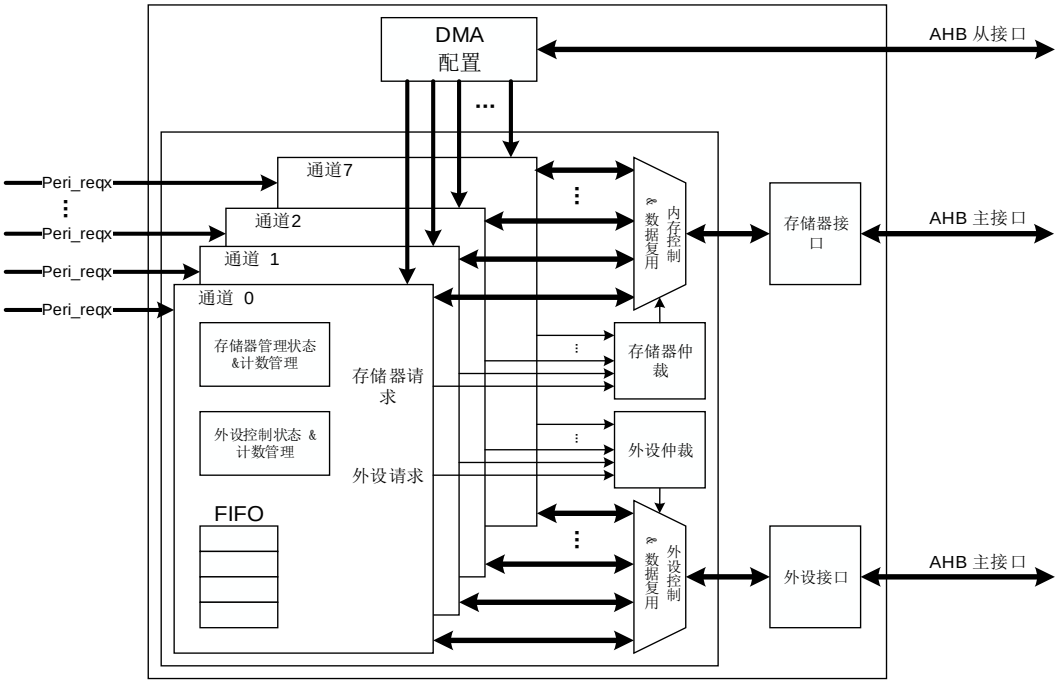
16.2. 主要特征

- 两个 AHB 主机接口传输数据，一个 AHB 从机接口配置 DMA。
- 16 个通道（DMA0 有 8 个通道，DMA1 有 8 个通道），并且每个通道都可配置。
- 存储器和外设支持单一传输，4 拍、8 拍和 16 拍增量突发传输。
- 当外设和存储器传输数据时，支持缓冲区切换。
- 支持软件优先级（低、中、高、超高）和硬件优先级（通道号越低，优先级越高）。
- 存储器和外设的数据传输宽度可配置：字节，半字，字。
- 存储器和外设的数据传输支持固定寻址和增量式寻址。
- 支持循环传输模式。
- 支持三种传输方式：
  - 存储器到外设；
  - 外设到存储器；
  - 存储器到存储器。
- 支持单数据传输和多数据传输模式，FIFO 深度最大为 4 个字：
  - 多数据传输模式：在存储器数据宽度和外设数据宽度不同的时候，自动打包/解包数据；
  - 单数据传输模式：当且仅当 FIFO 空的时候从源地址读取数据，存进 FIFO，然后把 FIFO 的数据写到目标地址。
- 每个通道有 5 种类型的事件标志和独立的中断。
- 支持中断的使能和清除。

16.3. 功能说明

16.3.1. 结构框图

图 16-1. 系统架构



如*图 16-1. 系统架构*所示，DMA 控制器由 4 部分组成：

- AHB 从接口配置 DMA；
- 通过两个 AHB 主接口分别为访问存储器和访问外设提供数据传输功能；
- 两个内置仲裁器用于管理同时发出的多个外设请求；
- 通道数据管理，以控制数据打包/解包和计数。

DMA 控制器在没有 CPU 参与的情况下从一个地址向另一个地址传输数据，它支持多种数据宽度，突发类型，地址生成算法，优先级和传输模式，可以灵活的配置 DMA 寄存器相应位以满足应用的需求。所有的 DMA 寄存器都可以通过 AHB 从机接口进行 32 位的操作。

支持外设到存储器、存储器到外设以及存储器到存储器三种传输模式，具体模式选择通过 DMA_CHxCTL 寄存器中的 TM 位域决定，如*表 16-1. 传输模式*所示。

表 16-1. 传输模式

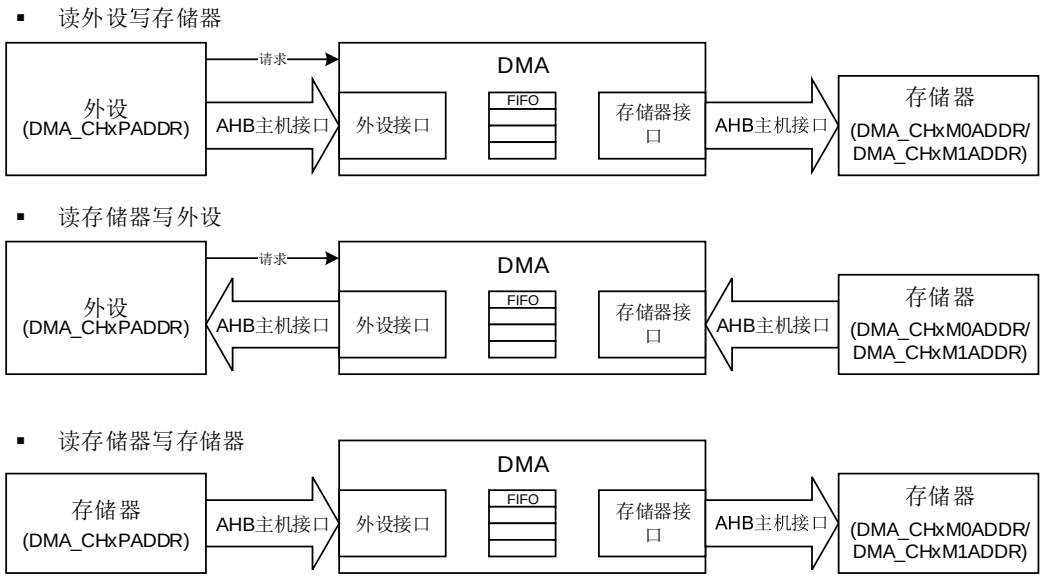
传输模式	TM[1:0]	源地址	目的地址
外设到存储器	00	DMA_CHxPADDR	DMA_CHxM0ADDR/ DMA_CHxM1ADDR
存储器到外设	01	DMA_CHxM0ADDR/ DMA_CHxM1ADDR	DMA_CHxPADDR
存储器到存储器	10	DMA_CHxPADDR	DMA_CHxM0ADDR/ DMA_CHxM1ADDR

注意：

1. 寄存器 DMA_CHxCTL 的 MBS 位选择 DMA_CHxM0ADDR 或者 DMA_CHxM1ADDR 作为存储器地址。详细请参考*存储切换模式*。

2. 寄存器 DMA_CHxCTL 的 TM 位域禁止配置成‘0b11’，否则通道将会自动关闭。

图 16-2. 三种传输模式的数据流



如*图 16-2. 三种传输模式的数据流*所示，DMA 控制器的两个 AHB 主机接口分别对应存储器和外设的数据访问。

- 存储器到外设：通过 AHB 存储器主机接口从存储器读取数据，通过 AHB 外设主机接口向外设写入数据；
- 外设到存储器：通过 AHB 外设主机接口从外设读取数据，通过 AHB 存储器主机接口向存储器写入数据；
- 存储器到存储器：通过 AHB 外设主机接口从存储器读取数据，通过 AHB 存储器主机接口向存储器写入数据。

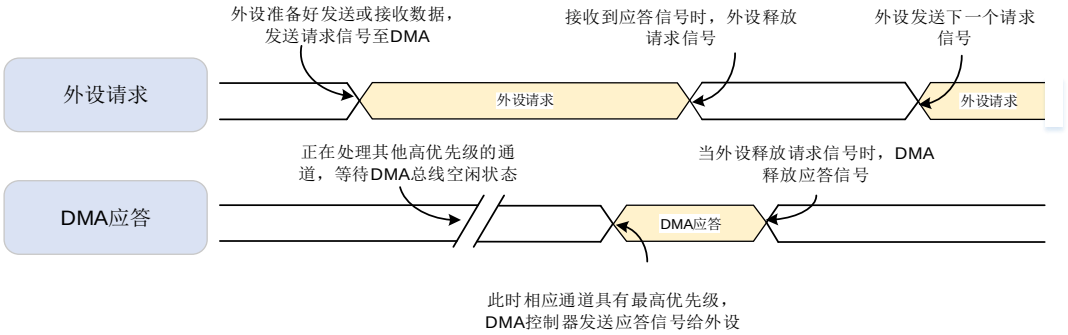
16.3.2. 外设握手

为了保证良好的组织和高效的数据传输，DMA 控制器中引入了 DMA 和外设之间的握手机制，包括请求信号和应答信号：

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

如*图 16-3. 握手机制*详细描述了 DMA 控制器与外设之间的握手机制。

图 16-3. 握手机制



16.3.3. 数据处理

仲裁

每个 DMA 控制器有两个分别对应于外设和存储器的仲裁器。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。优先级规则如下：

- 软件优先级：分为4级，低，中，高和超高。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。例如，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。

传输宽度，突发传输和计数

传输宽度

寄存器 DMA_CHxCTL 的 PWIDTH 和 MWIDTH 位域决定了外设和存储器的数据传输宽度。DMA 控制器支持 8 位，16 位和 32 位的传输宽度。在多数据传输模式中，如果 PWIDTH 和 MWIDTH 不相等，DMA 会自动的打包/解包数据来进行完整、正确的数据传输。在单数据传输模式中，MWIDTH 在通道使能以后，会被硬件强制设置与 PWIDTH 相等。

突发传输类型

寄存器 DMA_CHxCTL 的 PBURST 和 MBURST 位域决定了外设和存储器的突发传输方式。DMA 控制器的外设和存储器接口均支持单一传输，4 拍，8 拍，16 拍的增量突发传输。对于单数据传输模式，当使能通道后，PBURST 和 MBURST 会立即被硬件强制设为 0，仅支持单一传输。

在外设到存储器或者存储器到外设传输模式中，如果 PBURST 不为 0，在每次外设请求之后，DMA 控制器会根据 PBURST 的值进行 4 拍，8 拍，16 拍的增量突发传输。如果剩余的数据不够一次突发传输，剩余的数据将会进行单一传输。

AMBA 协议指定突发传输不能超过 1KB 的地址边界，否则将会产生传输错误并响应至主机。对于每个 DMA 的外设和存储器，当突发传输超过 1KB 的地址边界，硬件会自动的把 4 拍，8

拍，16 拍（由 PBURST 和 MBURST 决定）的突发传输拆分为单一传输。

传输计数

寄存器 DMA_CHxCNT 的 CNT 位域决定了待传输数据的数量，在使能 DMA 通道之前，待传输数据的数量必须完成配置。在传输过程中，CNT 表示剩余待传输的数据数量。

CNT 位的大小与外设的数据传输宽度有关，数据传输总量的字节数等于 CNT 乘以外设数据传输宽度。例如，如果 PWIDTH 的值设置为‘10’，则传输的数据总量的字节数等于 CNT*4。即使是外设到存储器或者存储器到存储器，CNT 的值在外设的每次单一传输或者在突发模式（存储器到存储器中源存储器）中每个节拍传输完成后都会减 1。

CNT 值的配置需要满足下列要求：

1. 如果关闭循环模式（清除寄存器 DMA_CHxCTL 的 CMEN 位），DMA_CHxCTL 中 CNT 值的配置应该满足*表 16-2. CNT 配置*的要求。传输的数据总量的字节数必须是存储器数据传输宽度的整数倍，以保证完整的单次存储器传输。

注意：如果 PBURST 和 MBURST 都不是‘00’，传输的数据总量不需要是存储器和外设的突发传输数据的整数倍。对于不满足一次突发传输的剩余数据，硬件会自动的拆分成多个单一传输。

表 16-2. CNT 配置

外设位宽	存储器位宽	CNT 值
8 位	16 位	2 的倍数
8 位	32 位	4 的倍数
16 位	32 位	2 的倍数
其他情况		任意值

2. 如果开启循环模式（置位寄存器 DMA_CHxCTL 的 CMEN 位），传输的数据总量必须保证同时是存储器突发传输数据总量和外设突发传输数据总量的整数倍，否则将不能保证数据的正确性。

- $CNT / PBURST_beats$ 必须是整数；
- $(CNT \times PWIDTH_bytes) / (MBURST_beats \times MWIDTH_bytes)$ 必须是整数。
  - PWIDTH_bytes 是外设的数据传输宽度的字节数。8 位是 1，16 位是 2，32 位是 4。
  - PBURST_beats 是外设突发传输的节拍数，单一传输是 1，INCR4（4 拍增量突发传输）是 4，INCR8（8 拍增量突发传输）是 8，INCR16（16 拍增量突发传输）是 16。
  - MWIDTH_bytes 是存储器的数据传输宽度的字节数。8 位是 1，16 位是 2，32 位是 4。
  - MBURST_beats 是存储器突发传输的节拍数，单一传输是 1，4 拍增量突发传输是 4，8 拍增量突发传输是 8，16 拍增量突发传输是 16。

例如：

1. 如果PWIDTH是16位，PBURST是INCR4，MWIDTH是8位，MBURST是INCR16，则CNT/4与(CNT×2)/(1×16)必须是整数，所以CNT必须是8的整数倍。
2. 如果PWIDTH是8位，PBURST是INCR16，MWIDTH是16位，MBURST是INCR4，则CNT/16与(CNT×1)/(2×4)必须是整数，所以CNT必须是16的倍数。

注意：如果通过将寄存器 DMA_CHxCTL 的 SBMEN 位置位使能了存储切换模式，循环模式会被硬件强制打开，所以也必须满足上述要求。

FIFO

DMA 控制器的每个通道都有一个 4 字深度的 FIFO 用于缓冲数据，从源地址读取的数据会先暂时保存在 FIFO 中，再传输到目的地址。根据 FIFO 的配置，DMA 控制器支持两种数据处理模式：单数据传输模式和多数据传输模式。在存储器到存储器模式下，DMA 控制器仅支持多数据传输模式。

多数据传输模式

通过将寄存器 DMA_CHxFCTL 的 MDMEN 位置 1 来开启多数据传输模式。

在这个模式中，当 FIFO 有足够的空间时，DMA 控制器响应源端的请求，将从源地址读取的数据压入 FIFO。如果目的端是外设，当 FIFO 内的数据量满足外设的一次突发传输时，DMA 会响应外设的请求。如果目标端是存储器，寄存器 DMA_CHxFCTL 的 FCCV 位设置的 FIFO 临界值决定 DMA 控制器何时进行将 FIFO 中的数据写入存储器，当 FIFO 计数器达到配置的临界值时，FIFO 中的所有数据会被写入目标存储器地址。

为了保证正确的数据传输，FIFO 的临界值（DMA_CHxFCTL 的 FCCV 位）必须配置为存储器一次突发传输数据量的整数倍。这样才能保证 FIFO 中有足够的数据可以完成存储器突发传输。FIFO 计数器的临界值的设置与存储器数据传输宽度和存储器突发传输类型有关，具体见*表 16-3. FIFO 计数器临界值配置*。

表 16-3. FIFO 计数器临界值配置

MWIDTH	MBURST	FIFO 计数器临界值			
		1 字	2 字	3 字	4 字
8 位	单一	4 次单一传输	8 次单一传输	12 次单一传输	16 次单一传输
	INCR4	1 次突发传输	2 次突发传输	3 次突发传输	4 次突发传输
	INCR8	错误	1 次突发传输	错误	2 次突发传输
	INCR16	错误	错误	错误	1 次突发传输
16 位	单一	2 次单一传输	4 次单一传输	6 次单一传输	8 次单一传输
	INCR4	错误	1 次突发传输	错误	2 次突发传输
	INCR8	错误	错误	错误	1 次突发传输
	INCR16	错误	错误	错误	错误
32 位	单一	1 次单一传输	2 次单一传输	3 次单一传输	4 次单一传输
	INCR4	错误	错误	错误	1 次突发传输
	INCR8	错误	错误	错误	错误
	INCR16	错误	错误	错误	错误

注意：当传输模式是外设到存储器时，如果 PBURST_beats × PWIDTH_bytes = 16，FIFO 计数器临界值不能设置成‘10’。如果设置成‘10’，当接收到外设请求时，DMA 控制器启动外设突发传输并填满 FIFO，然后 DMA 会向存储器中写入 3 个字的数据（这个是由 FIFO 的临界值决定），同时 FIFO 中剩余一个字的数据。这时当新的外设请求来临时，FIFO 中没有足够的空间进行下一次的外设突发传输，同时 FIFO 中的数据没有达到 FIFO 临界值也不会进行存储器突发传输，通道数据传输将被冻结。

单数据传输模式

通过将寄存器 DMA_CHxFCTL 的 MDMEN 位清 0 来开启单数据传输模式。在这个模式中，DMA 控制器一次只能传输一个数据，FIFO 计数器临界值的配置（寄存器 DMA_CHxFCTL 的 FCCV 位域）没有意义。

在单数据传输模式中，当且仅当 FIFO 空的时候，DMA 会响应源端的请求，不管源传输宽度是多少，从源地址读取数据进入 FIFO。当 FIFO 非空时，DMA 响应目的端的请求，将 FIFO 中的数据写入目的地址。

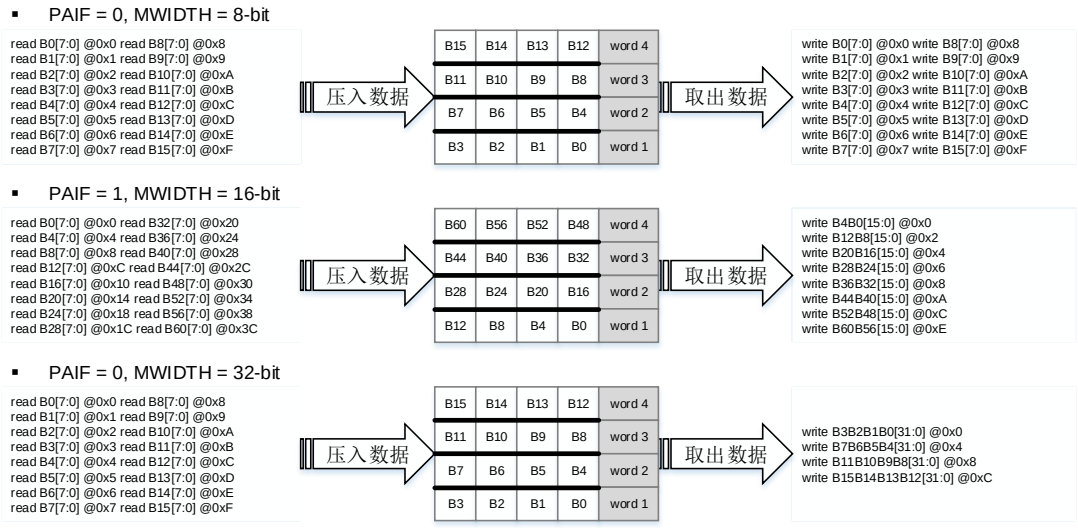
打包/解包

在单数据传输模式中，MWIDTH 会被硬件强制设置与 PWIDTH 相等，无需使用数据的打包/解包功能。

在多数据传输模式中，MWIDTH 与 PWIDTH 相互独立，配置更为灵活。当 MWIDTH 与 PWIDTH 不相等时，DMA 的读写传输宽度不同，DMA 会自动对数据进行打包/解包操作。在 DMA 传输过程中，存储器和外设都只支持小端操作。

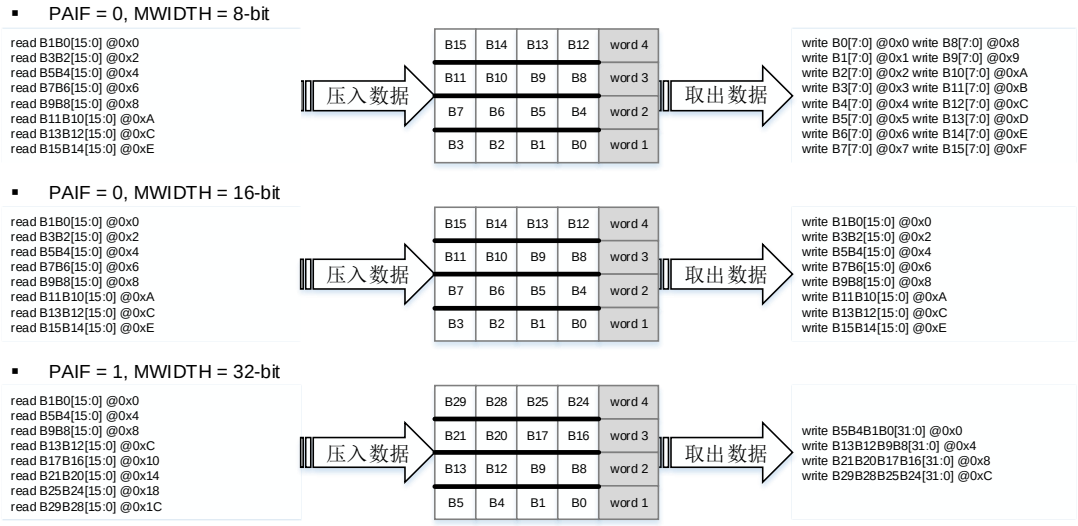
假设当 CNT 被设置为 16，PWIDTH 为‘00’，PNAGA 和 MNAGA 被置 1。对于不同的 MWIDTH，DMA 的传输操作如*图 16-4. PWIDTH 为‘00’时，数据的打包/解包*所示。

图 16-4. PWIDTH 为‘00’时，数据的打包/解包



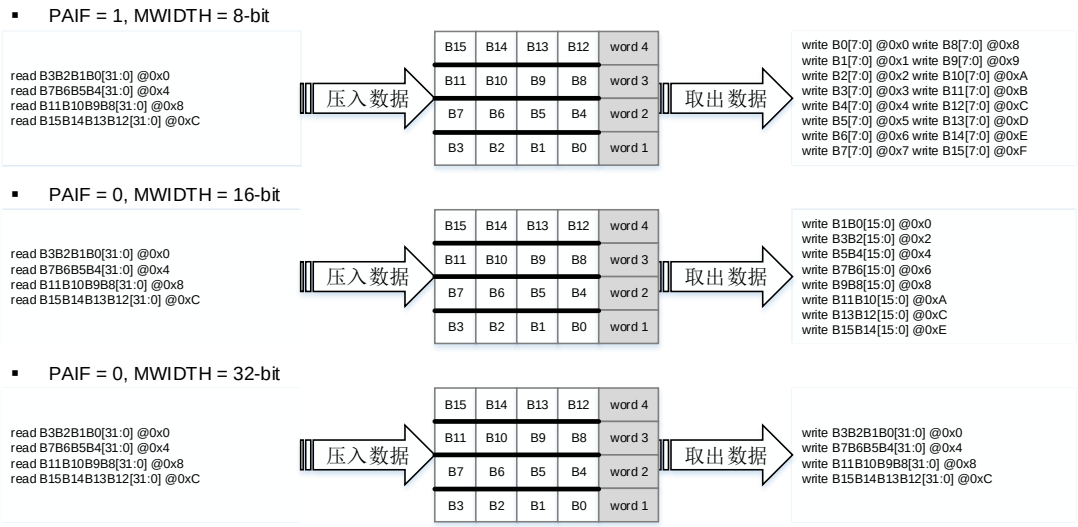
假设当 CNT 被设置为 8，PWIDTH 为‘01’，PNAGA 和 MNAGA 被置 1。对于不同的 WIDTH，DMA 的传输操作如*图 16-5. PWIDTH 为‘01’时，数据的打包/解包*所示。

图 16-5. PWIDTH 为‘01’时，数据的打包/解包



当 CNT 被设置为 4，PWIDTH 为‘10’，PNAGA 和 MNAGA 被置 1。对于不同的 MWIDTH，DMA 的传输操作如*图 16-6. PWIDTH 为‘10’时，数据的打包/解包*所示。

图 16-6. PWIDTH 为‘10’时，数据的打包/解包



16.3.4. 地址生成

存储器和外设都独立的支持两种地址生成算法：固定模式和增量模式。寄存器 DMA_CHxCTL 的 PNAGA 和 MNAGA 位用来设置存储器和外设的下一次传输的地址生成算法。

在固定模式中，下一次传输地址一直固定为初始化的基地址（DMA_CHxPADDR，DMA_CHxM0ADDR，DMA_CHxM1ADDR）。

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。在多数据传输模式中，若寄存器 DMA_CHxCTL 的 PBURST 配置为‘00’，当寄存器 DMA_CHxCTL 的 PAIF 位置 1 使能时，外设下一次传输的地址增量被固定为 4，与外设的数据传输宽度无关。PAIF 与存储器地址生成无关。

注意：若DMA_CHxCTL寄存器中的PAIF位使能，寄存器DMA_CHxPADDR中配置的外设基地

址必须4字节对齐。

16.3.5. 循环模式

循环模式用来处理连续的外设请求。可以通过配置寄存器 DMA_CHxCTL 的 CMEN 位来使能/禁能循环模式。循环模式只在 DMA 作为传输控制器时有效。

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，直到出现传输错误或者 DMA_CHxCTL 寄存器中 CHEN 位被清 0。

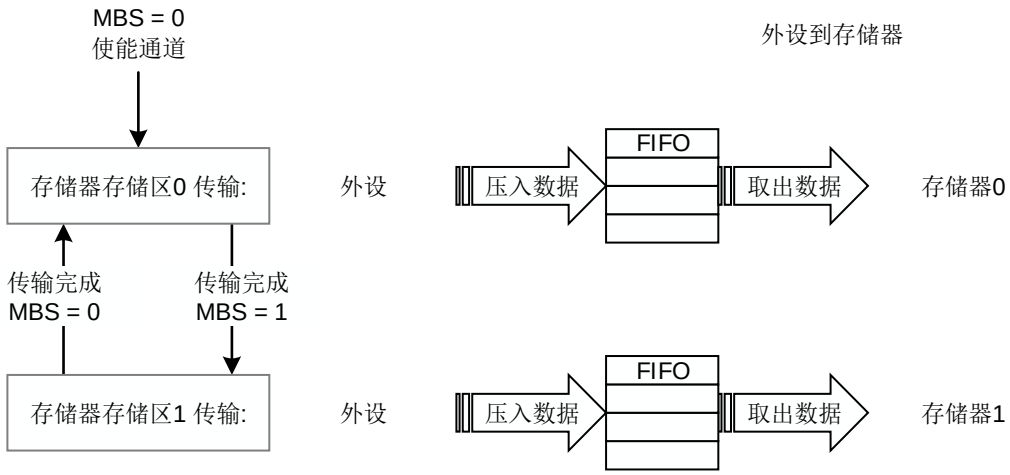
16.3.6. 存储切换模式

与循环模式类似，存储切换模式也是用来处理连续的外设请求。可以通过配置寄存器 DMA_CHxCTL 的 SBMEN 位来使能/禁能存储切换模式。若使能了存储切换模式，在通道使能后，硬件会自动打开循环模式。存储切换模式只能应用于外设到存储器或存储器到外设的数据传输，在存储器到存储器模式中，存储切换模式在通道使能后立即被禁止使用。

存储切换模式支持两个存储器缓冲区，两个存储器基地址可以分别在寄存器 DMA_CHxM0ADDR 和 DMA_CHxM1ADDR 中配置。在存储切换模式中，每次 DMA 传输完成后，存储器指针指向另一个存储器缓冲区。在 DMA 传输过程中，没有被 DMA 占用的缓冲区可以被其他的 AHB 主机接口操作，且即使通道使能，其基地址可以改变。

在通道使能之前，软件可以通过设定寄存器 DMA_CHxCTL 的 MBS 位来指定第一次数据传输 DMA 使用的缓冲区。MBS 可视为 DMA 在传输过程中访问的当前内存缓冲区的标志，它会在每次传输完成后自动在‘0’，‘1’之间切换，DMA 存储切换模式操作如*图 16-7. 存储切换模式*所示。

图 16-7. 存储切换模式



16.3.7. 传输操作

数据传输支持三种操作方式：外设到存储器，存储器到外设，和存储器到存储器。存储器和外设都可以配置为源端和目的端。

存储器端数据传输

- 外设到存储器模式：
  - 单数据传输模式，当 FIFO 非空时，DMA 启动存储器数据传输，写数据到相应的存储器地址中；
  - 多数据传输模式，当 FIFO 计数器达到临界值时，DMA 启动单一或突发数据传输，把 FIFO 的数据全部写入存储器中。
- 存储器到外设模式：
  - 单数据传输模式，当通道使能时 DMA 会立刻进行存储器数据传输，读取数据到 FIFO。数据传输过程中，当且仅当 FIFO 为空时，DMA 控制器就会进行存储器读取操作；
  - 多数据传输模式，当通道使能后，不论是否有外设请求，DMA 都会进行单一或突发数据传输填满 FIFO。在数据传输过程中，当 FIFO 有足够的空间进行一次单一或突发传输时，DMA 控制器就会进行存储器读取操作。
- 存储器到存储器模式：当 FIFO 计数器到达临界值，DMA 进行单一或突发传输把 FIFO 的数据全部写入存储器中。

外设端数据传输

- 外设到存储器模式：当 DMA 收到外设请求且 FIFO 有足够的空间进行数据传输，DMA 启动外设数据传输从外设读取数据写入 FIFO；
- 存储器到外设模式：当 DMA 收到外设请求且 FIFO 有足够的数据进行数据传输，DMA 启动外设数据传输从 FIFO 读取数据写入外设。

16.3.8. 传输完成

DMA 传输由硬件自动完成，DMA_INTF0 寄存器或 DMA_INTF1 寄存器中 FTFIFx 位在以下情况下会被置 1：

- 传输完成；
- 软件清除；
- 传输错误。

传输完成

当 DMA 使能以后，数据会在外设和存储器之间传输。当事先配置的数据量传输完成后，DMA 传输结束，寄存器 DMA_CHxCTL 的 CHEN 位自动清零。

- 外设到存储器模式：当 CNT 减数到 0 且 FIFO 中的数据完全写入到存储器中，传输完成。
- 存储器到外设模式：当 CNT 减数到 0 时传输完成。
- 存储器到存储器模式：当 CNT 减数到 0 且 FIFO 中的数据完全写入到存储器中，传输完成。

软件清除

可以通过将寄存器 DMA_CHxCTL 的 CHEN 位清 0 停止 DMA 传输。在软件清 0 操作之后，若 CHEN 仍然为 1，代表存储器或者外设仍然处在传输状态，或者 FIFO 中还有剩余的数据没有传输完成。

- 外设到存储器：软件清 0 操作后，当前的单次或突发传输完成后，DMA 的外设传输将会停止。为了保证从外设读取的数据完全被写入存储器中，存储器在 FIFO 非空的状态下仍然会进行数据传输，直到 FIFO 中的数据完全被写入存储器中。若 FIFO 中剩余的数据量不满足一次存储器突发传输，这些数据将会被拆分成单一传输。如果 FIFO 总剩余的数据量小于存储器传输宽度，这个数据会被高位补 0，写入存储器中。此时读取 CNT 的值可以计算出存储器中的有效数据量。在 FIFO 中的数据传输完毕之后，CHEN 会被硬件自动清 0，寄存器 DMA_INTF0 或 DMA_INTF1 相应通道标志位 FTFIFx 会被置 1；
- 存储器到外设：软件清 0 操作后，当前的存储器和外设传输完成以后，DMA 传输将会停止。CHEN 会被硬件自动清 0，寄存器 DMA_INTF0 或 DMA_INTF1 相应通道标志位 FTFIFx 会被置 1；
- 存储器到存储器：与外设到存储器相同，其中源端存储器的传输通过外设端口来实现。

传输错误

三种类型的错误会关闭 DMA 传输：

- FIFO 错误：当检测到 FIFO 配置错误，通道会立即关闭且不会开始任何传输。这种情况下，FTFIFx 不会被置 1。更多 FIFO 错误的信息，请参考***错误***;
- 总线错误：当存储器或者外设端口试图访问超出范围的地址时，DMA 控制器会检测到总线错误，通道立即停止传输且 FTFIFx 不会被置 1。如果错误是由外设端口引起，CNT 仍会进行一次减 1 操作。更多总线错误的信息，请参考***错误***;
- 寄存器访问错误：在存储切换模式下，当对 DMA 正在访问的存储器的基地址寄存器进行写操作时，DMA 控制器会检测到寄存器访问错误。发生这个错误后，DMA 控制器的操作与软件清 0 时相同。更多寄存器访问错误的信息，请参考***错误***。

16.3.9. 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 0 或等待 DMA 传输完成。当 CHEN 为 0 时，请按照下列步骤配置 DMA。
2. 清除寄存器 DMA_INTF0 或 DMA_INTF1 相应通道标志位 FTFIFx，否则无法使能 DMA。
3. 配置寄存器 DMA_CHxCTL 的 TM[1:0]位选择数据传输方式。
4. 在寄存器 DMA_CHxCTL 中配置存储器和外设突发类型，目标存储器（存储器 0 或存储器 1），存储切换模式，通道优先级，存储器和外设的传输宽度，存储器和外设的地址生成算法，循环模式。
5. 如果寄存器 DMA_CHxFCTL 中多数据传输方式使能，需要配置 FCCV[1:0]位域以设置 FIFO 计数器临界值。
6. 在寄存器 DMA_CHxCTL 中配置传输完成中断，半传输完成中断，传输错误中断，单数据传输方式异常中断的使能位。在寄存器 DMA_CHxFCTL 中配置 FIFO 错误和异常中断的使能位。
7. 在寄存器 DMA_CHxPADDR 中配置外设基地址。
8. 如果使用存储切换模式，在寄存器 DMA_CHxM0ADDR 和 DMA_CHxM1ADDR 中配置两个存储器基地址。如果只使用一个存储器，寄存器 DMA_CHxCTL 的 MBS 位决定配置 DMA_CHxM0ADDR 或者 DMA_CHxM1ADDR。
9. 在寄存器 DMA_CHxCNT 中配置数据传输总量。

10. 寄存器 DMA_CHxCTL 的 CHEN 位置 1，使能 DMA 通道。

如果要继续被暂停的DMA传输，建议遵循以下步骤进行操作：

1. 读取 CHEN 位，确定 DMA 的挂起操作已经完成。当 CHEN 为 0 时，DMA 处于空闲状态，可以重新配置 DMA 以继续挂起 DMA 传输。
2. 清除寄存器 DMA_INTF0 或 DMA_INTF1 相应通道标志位 FTFIFx，否则 DMA 通道可能无法再使能。
3. 读取寄存器 DMA_CHxCNT 计算出已经发送的数据量与剩余待发的数据量。
4. 在寄存器 DMA_CHxPADDR 中更新外设地址指针。
5. 在寄存器 DMA_CHxM0ADDR 或 DMA_CHxM1ADDR 中更新存储器地址指针。
6. 在寄存器 DMA_CHxCNT 中配置剩余待发的数据总量。
7. 寄存器 DMA_CHxCTL 的 CHEN 位置 1，重新启动 DMA 通道。

16.4. 中断

每个 DMA 通道都有专有的中断，包括 5 个中断事件，传输完成中断，半传输完成中断，传输错误中断，单数据传输模式异常中断，FIFO 错误和异常中断。任何一个中断事件都可以引发 DMA 中断。

寄存器 DMA_INTF0 或 DMA_INTF1 包含每个中断事件的标志位，寄存器 DMA_INTC0 或 DMA_INTC1 包含每个中断事件的标志清除位，寄存器 DMA_CHxCTL 和 DMA_CHxFCTL 包含每个中断事件的使能位，具体如*表 16-4. DMA 中断事件*所示。

表 16-4. DMA 中断事件

中断事件	标志位	使能位	清除位
	DMA_INTF0 或 DMA_INTF1	DMA_CHxCTL 或 DMA_CHxFCTL	DMA_INTC0 或 DMA_INTC1
传输完成	FTFIF	FTFIE	FTFIFC
半传输完成	HTFIF	HTFIE	HTFIFC
传输错误	TAEIF	TAEIE	TAEIFC
单数据模式异常	SDEIF	SDEIE	SDEIFC
FIFO 错误与异常	FEEIF	FEEIE	FEEIFC

这5个事件可以分为3种类型：

- 标志：传输完成标志和半传输完成标志；
- 异常：单数据传输模式异常和 FIFO 异常；
- 错误：传输错误和 FIFO 错误。

发生异常事件时，正在进行的 DMA 传输不会被停止，仍将继续传输。发生错误事件时，正在进行的 DMA 传输会被停止。这三种类型的事件在进行详细描述。

16.4.1. 标志

两种标志事件，传输完成事件和半传输完成事件。

发生以下情况时，传输完成标志位将会被置 1：

- CNT[15:0]计数到 0；
- 在数据传输完成之前，通过软件清 0 的方式停止数据传输，当前的存储器和外设数据传输完成后，如果是外设到存储器或存储器到存储器传输模式，还需满足 FIFO 中的数据全部写入存储器中，传输完成；
- 在数据传输完成之前，由于寄存器访问错误导致停止数据传输，当前的存储器和外设数据传输完成后，如果是外设到存储器或存储器到存储器传输模式，还需满足 FIFO 中的数据全部写入存储器，传输完成。

当传输完成标志位置 1，且传输完成中断使能时，DMA 控制器产生传输完成中断。

当 DMA 作为传输控制器且 CNT[15:0]减数计数达到初始值的一半时，半传输完成标志位会被置 1。当外设作为传输控制器时，DMA 无法得知是否已经传输一半的数据流，此时半传输完成标志仍为 0。

当半传输完成标志位置 1，且半传输完成中断使能时，DMA 控制器产生半传输完成中断。

16.4.2. 异常

两种异常事件，单数据传输模式异常和 FIFO 异常。异常对于 DMA 传输无影响。

单数据传输模式异常

这个异常只有在使能单数据传输模式且传输方式为外设到存储器时才会发生。当 FIFO 非空时，如果外设请求数据传输，DMA 在响应外设请求以后，会有多个数据存储在 FIFO 中，这可能会对存储器后续的数据处理造成影响，此时单数据传输模式异常标志位 SDEIFx 置 1。

当单数据传输模式异常标志位置 1，且单数据传输模式异常中断使能，将产生一个中断。

FIFO 异常

这个异常只有数据在外设和存储器之间传输才会发生，当 FIFO 发生上溢或下溢时，FIFO 异常标志位置 1。

当传输模式为外设到存储器时，如果外设请求有效，但 FIFO 的剩余空间不满足单一或突发外设传输，FIFO 发生上溢。直到 FIFO 有足够空间时，DMA 控制器才会响应此次外设请求，该异常不会影响到数据传输的正确性。

当传输模式为存储器到外设时，如果外设请求有效，但 FIFO 中的数据不够完成单一或突发外设传输，FIFO 发生下溢。直到 FIFO 有足够数据时，DMA 控制器才会响应此次外设请求，该异常不会影响到数据传输的正确性。

当 FIFO 异常标志位置 1，且 FIFO 异常中断使能时，将产生一个中断。

16.4.3. 错误

在数据传输过程中，会发生 FIFO 错误和传输错误（包含寄存器访问错误和总线错误），此时数据传输会被中止。

FIFO 错误

对于一次 DMA 操作，当启用多数据模式时，内存传输宽度和内存突发类型对应的 FIFO 计数器临界值的正确和错误配置如*表 16-3. FIFO 计数器临界值配置*。

错误的配置会引发 FIFO 错误，此时，通道会立即关闭，并不启动任何传输。

当 FIFO 错误标志位置 1，且 FIFO 错误和异常中断使能时，将产生一个中断。

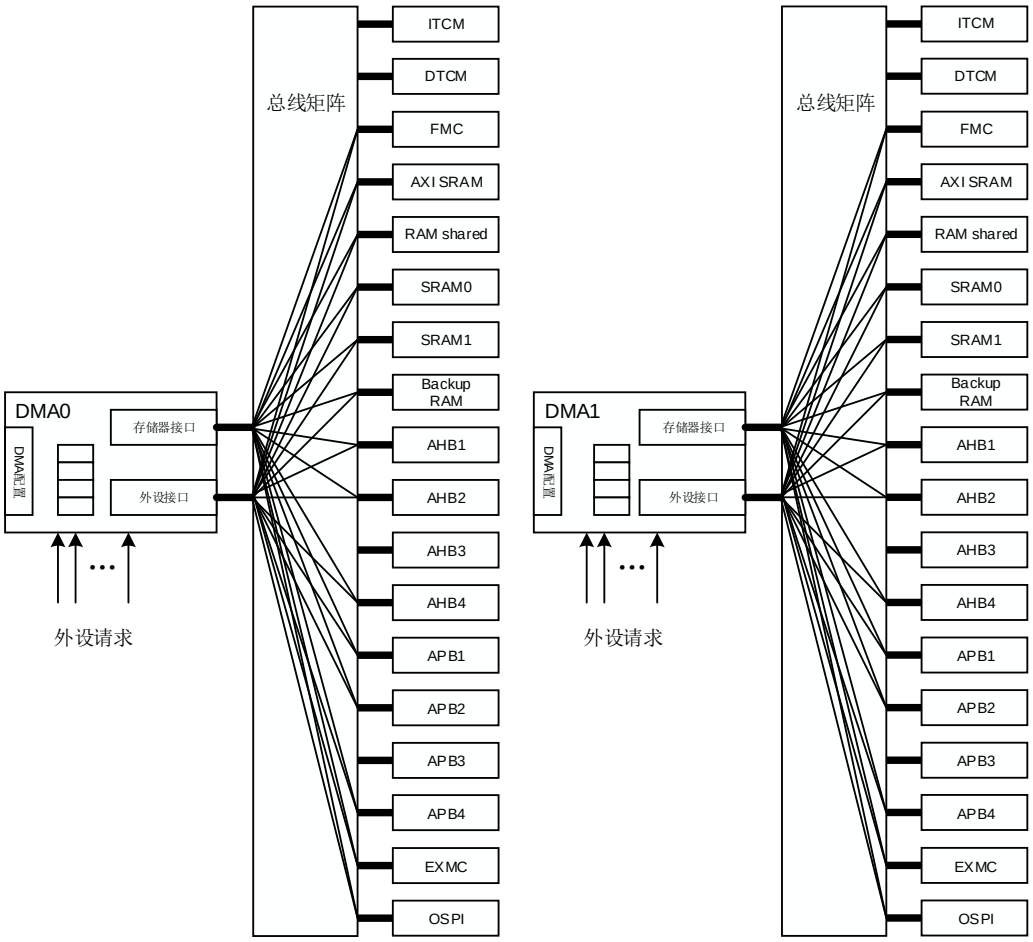
寄存器访问错误

只有在存储切换模式下才会发生寄存器访问错误。如果软件对 DMA 正在使用的存储器的基地址寄存器进行写操作，将会发生寄存器访问错误。举例来说，存储器 0 是 DMA 控制器正在使用的源端或者目的端地址，如果软件对 DMA_CHxM0ADDR 寄存器进行写操作，则会产生寄存器访问错误。寄存器访问错误发生后，当前存储器或外设数据传输完成之后，并且有效的 FIFO 数据完全写入存储器中，DMA 会被自动停止。当寄存器访问错误标志位置 1，且寄存器访问错误中断使能时，将产生一个中断。

总线错误

当 DMA 控制器访问的地址超出了允许的范围，会发生总线错误，同时 DMA 通道立即失能。DMA0 和 DMA1 允许访问的地址空间如*图 16-8. DMA0 与 DMA1 的系统连接*所示。当总线错误标志位置 1，且传输访问错误和异常中断的使能位被设置时，将生成一个中断。

图 16-8. DMA0 与 DMA1 的系统连接



16.4.4. 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 18-2. DMAMUX 请求路由输入信号映射*。

16.5. DMA 寄存器

DMA0 基地址：0x4002 0000

DMA1 基地址：0x4002 0400

16.5.1. 中断标志位寄存器 0（DMA_INTF0）

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				FTFIF3	HTFIF3	TAEIF3	SDEIF3	保留	FEEIF3	FTFIF2	HTFIF2	TAEIF2	SDEIF2	保留	FEEIF2
				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
保留				FTFIF1	HTFIF1	TAEIF1	SDEIF1	保留	FEEIF1	FTFIF0	HTFIF0	TAEIF0	SDEIF0	保留	FEEIF0
				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。
27/21/11/5	FTFIFx	通道 x 的传输完成标志位（x=0…3） 硬件置位，软件写 DMA_INTC0 相应位为 1 清零 0：通道 x 传输未完成 1：通道 x 传输完成
26/20/10/4	HTFIFx	通道 x 的半传输完成标志位（x=0…3） 硬件置位，软件写 DMA_INTC0 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
25/19/9/3	TAEIFx	通道 x 的传输错误标志位（x=0…3） 硬件置位，软件写 DMA_INTC0 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
24/18/8/2	SDEIFx	通道 x 的单数据传输模式异常标志位（x=0…3） 硬件置位，软件写 DMA_INTC0 相应位为 1 清零 0：通道 x 未发生单数据传输模式异常 1：通道 x 发生单数据传输模式异常
23/17/7/1	保留	必须保持复位值。
22/16/6/0	FEEIFx	通道 x 的 FIFO 错误与 FIFO 异常标志位（x=0…3） 硬件置位，软件写 DMA_INTC0 相应位为 1 清零 0：通道 x 未发生 FIFO 错误或 FIFO 异常

1：通道 x 发生 FIFO 错误或 FIFO 异常

16.5.2. 中断标志位寄存器 1（DMA_INTF1）

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				FTFIF7	HTFIF7	TAEIF7	SDEIF7	保留	FEEIF7	FTFIF6	HTFIF6	TAEIF6	SDEIF6	保留	FEEIF6
				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
保留				FTFIF5	HTFIF5	TAEIF5	SDEIF5	保留	FEEIF5	FTFIF4	HTFIF4	TAEIF4	SDEIF4	保留	FEEIF4
				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。
27/21/11/5	FTFIFx	通道 x 的传输完成标志位（x=4…7） 硬件置位，软件写 DMA_INTC0 相应位为 1 清零 0：通道 x 传输未完成 1：通道 x 传输完成
26/20/10/4	HTFIFx	通道 x 的半传输完成标志位（x=4…7） 硬件置位，软件写 DMA_INTC0 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
25/19/9/3	TAEIFx	通道 x 的传输错误标志位（x=4…7） 硬件置位，软件写 DMA_INTC0 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
24/18/8/2	SDEIFx	通道 x 的单数据传输模式异常标志位（x=4…7） 硬件置位，软件写 DMA_INTC0 相应位为 1 清零 0：通道 x 未发生单数据传输模式异常 1：通道 x 发生单数据传输模式异常
23/17/7/1	保留	必须保持复位值。
22/16/6/0	FEEIFx	通道 x 的 FIFO 错误与 FIFO 异常标志位（x=4…7） 硬件置位，软件写 DMA_INTC0 相应位为 1 清零 0：通道 x 未发生 FIFO 错误或 FIFO 异常 1：通道 x 发生 FIFO 错误或 FIFO 异常

16.5.3. 中断标志位清除寄存器 0（DMA_INTC0）

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				FTFIFC3	HTFIFC3	TAEIFC3	SDEIFC3	保留	FEEIFC3	FTFIFC2	HTFIFC2	TAEIFC2	SDEIFC2	保留	FEEIFC2
				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
保留				FTFIFC1	HTFIFC1	TAEIFC1	SDEIFC1	保留	FEEIFC1	FTFIFC0	HTFIFC0	TAEIFC0	SDEIFC0	保留	FEEIFC0
				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。
27/21/11/5	FTFIFCx	通道 x 的传输完成标志清除位（x=0…3） 0：无影响 1：清除传输完成标志位
26/20/10/4	HTFIFCx	通道 x 的半传输完成标志清除位（x=0…3） 0：无影响 1：清除半传输完成标志位
25/19/9/3	TAEIFCx	通道 x 的传输错误标志清除位（x=0…3） 0：无影响 1：清除传输错误标志位
24/18/8/2	SDEIFCx	通道 x 的单数据传输模式异常标志清除位（x=0…3） 0：无影响 1：清除单数据传输模式异常标志位
23/17/7/1	保留	必须保持复位值。
22/16/6/0	FEEIFCx	通道 x 的 FIFO 错误与 FIFO 异常标志清除位（x=0…3） 0：无影响 1：清除 FIFO 错误与 FIFO 异常标志位

16.5.4. 中断标志位清除寄存器 1（DMA_INTC1）

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				FTFIFC7	HTFIFC7	TAEIFC7	SDEIFC7	保留	FEEIFC7	FTFIFC6	HTFIFC6	TAEIFC6	SDEIFC6	保留	FEEIFC6
				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

保留	FTFIFC5	HTFIFC5	TAEIFC5	SDEIFC5	保留	FEEIFC5	FTFIFC4	HTFIFC4	TAEIFC4	SDEIFC4	保留	FEEIFC4
	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。
27/21/11/5	FTFIFCx	通道 x 的传输完成标志清除位（x=4…7） 0：无影响 1：清除传输完成标志位
26/20/10/4	HTFIFCx	通道 x 的半传输完成标志清除位（x=4…7） 0：无影响 1：清除半传输完成标志位
25/19/9/3	TAEIFCx	通道 x 的传输错误标志清除位（x=4…7） 0：无影响 1：清除传输错误标志位
24/18/8/2	SDEIFCx	通道 x 的单数据传输模式异常标志清除位（x=4…7） 0：无影响 1：清除单数据传输模式异常标志位
23/17/7/1	保留	必须保持复位值。
22/16/6/0	FEEIFCx	通道 x 的 FIFO 错误与 FIFO 异常标志清除位（x=4…7） 0：无影响 1：清除 FIFO 错误与 FIFO 异常标志位

16.5.5. 通道 x 控制寄存器（DMA_CHxCTL）

x = 0...7，x 为通道编号

地址偏移：0x10 + 0x18 * x

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留							MBURST[1:0]		PBURST[1:0]		保留	MBS	SBMEN	PRIO[1:0]	
							rw		rw			rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAIF	MWIDTH[1:0]		PWIDTH[1:0]		MNAGA	PNAGA	CMEN	TM[1:0]		保留	FTFIE	HTFIE	TAEIE	SDEIE	CHEN
rw	rw		rw		rw	rw	rw	rw			rw	rw	rw	rw	rw

位/位域	名称	名称
31:25	保留	必须保持复位值。
24:23	MBURST[1:0]	存储器突发类型 软件置 1 与清 0。 00：单一传输

		01：INCR4（4 拍增量突发传输） 10：INCR8（8 拍增量突发传输） 11：INCR16（16 拍增量突发传输） CHEN 为 1 时不可写入。 如果寄存器 DMA_CHxFCTL 的 MDMEN 位为 0，在使能通道后（CHEN 置 1），该位域会被硬件强制清零。
22:21	PBURST[1:0]	外设突发类型 软件置 1 与清 0 00：单一传输 01：INCR4（4 拍增量突发传输） 10：INCR8（8 拍增量突发传输） 11：INCR16（16 拍增量突发传输） CHEN 为 1 时不可写入 如果寄存器 DMA_CHxFCTL 的 MDMEN 位为 0，在使能通道后（CHEN 置 1），该位域会被硬件强制清零。
20	保留	必须保持复位值。
19	MBS	存储器缓冲选择 硬件置 1 清 0，软件置 1 清 0 0：存储器 0 作为存储器传输区域 1：存储器 1 作为存储器传输区域 CHEN 为 1 时不可写入 在每次传输完成时，硬件会自动更新该位，以此来表明 DMA 正在使用哪个存储区
18	SBMEN	存储切换模式使能 软件置 1 与清 0 0：关闭存储切换模式 1：打开存储切换模式 CHEN 为 1 时不可写入
17:16	PRIO[1:0]	软件优先级 软件置 1 与清 0 00：低 01：中 10：高 11：超高 CHEN 为 1 时不可写入
15	PAIF	外设地址增量固定 软件置 1 与清 0 0：外设地址增量由 PWIDTH 决定 1：外设地址增量固定为 4 CHEN 为 1 时不可写入 如果 PNAGA 设置为 0，该位无影响 如果寄存器 DMA_CHxFCTL 的 MDMEN 位为‘0’或者 PBURST 不为‘00’，在使能通

		道后（CHEN 置 1），该位域会被硬件强制清零
14:13	MWIDTH[1:0]	存储器传输宽度 软件置 1 与清 0 00：8 位 01：16 位 10：32 位 11：保留 CHEN 为 1 时不可写入 如果寄存器 DMA_CHxFCTL 的 MDMEN 位为‘0’，在使能通道后（CHEN 置 1），该位域会被硬件强制与 PWIDTH 相等
12:11	PWIDTH[1:0]	外设传输宽度 软件置 1 与清 0 00：8 位 01：16 位 10：32 位 11：保留 CHEN 为 1 时不可写入
10	MNAGA	存储器地址生成算法 软件置 1 与清 0 0：固定地址模式 1：增量地址模式 CHEN 为 1 时不可写入
9	PNAGA	外设地址生成算法 软件置 1 与清 0 0：固定地址模式 1：增量地址模式 CHEN 为 1 时不可写入
8	CMEN	循环模式 软件置 1 与清 0 0：关闭循环模式 1：打开循环模式 CHEN 为 1 时不可写入 如果 SBMEN 为‘1’，在使能通道后（CHEN 置 1），该位被自动置 1
7:6	TM[1:0]	传输方式 软件置 1 与清 0 00：读外设写存储器 01：读存储器写外设 10：读存储器写存储器 11：保留 CHEN 为 1 时不可写入

5	保留	必须保持复位值。
4	FTFIE	传输完成中断使能位 软件置 1 与清 0 0：传输完成中断禁止 1：传输完成中断使能
3	HTFIE	半传输完成中断使能位 软件置 1 与清 0 0：半传输完成中断禁止 1：半传输完成中断使能
2	TAEIE	传输错误中断使能位 软件置 1 与清 0 0：传输错误中断禁止 1：传输错误中断使能
1	SDEIE	单数据传输模式异常中断使能位 软件置 1 与清 0 0：单数据传输模式异常中断禁止 1：单数据传输模式异常中断使能
0	CHEN	通道使能 软件置 1，硬件清 0 0：通道禁止 1：通道使能 该位置 1，DMA 传输开始。发生以下情况该位会被自动清 0： 数据传输完成 发生 FIFO 配置错误或者传输错误 软件清 0 操作后，读该位仍为 1 代表还有正在进行的数据传输，软件查询该位可以确定 DMA 通道是否空闲，可以进行新的数据传输。

16.5.6. 通道 x 计数寄存器（DMA_CHxCNT）

x = 0...7，x 为通道编号

地址偏移：0x14 + 0x18 * x

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

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 在使能通道后（CHEN 置 1），该位域不可写。 传输过程中，CNT 代表剩余未发的数据量。外设每传输一次数据，CNT 减 1。如果寄存器 DMA_CHxCTL 的 CMEN 位或 SBMEN 位置 1，在每次传输完成时，CNT 会由硬件自动重新装载。

16.5.7. 通道 x 外设基地址寄存器（DMA_CHxPADDR）

x = 0...7，x 为通道编号

地址偏移：0x18 + 0x18 * 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
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 在使能通道后（CHEN 置 1），该位域不可写。 当 PWIDTH 位‘01’，最低位被忽略，自动半字对齐 当 PWIDTH 位‘10’，最低位两位被忽略，自动字对齐 **注意：**若寄存器 DMA_CHxCTL 的 PAIF 位置 1，该位域必须配置为 4 字节对齐。

16.5.8. 通道 x 存储器 0 基地址寄存器（DMA_CHxM0ADDR）

x = 0...7，x 为通道编号

地址偏移：0x1C + 0x18 * 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
M0ADDR[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M0ADDR[15:0]															
rw															

位/位域	名称	描述
31:0	M0ADDR[31:0]	存储器 0 基地址

若寄存器 DMA_CHxCTL 位 MBS 为 0，该位域定义 DMA 传输过程中存储器的基地址
如果寄存器 DMA_CHxCTL 的 CHEN 位置 1 且 MBS 位为 0 时，该位域不可写
当 MWIDTH 位‘01’，最低位被忽略，自动半字对齐
当 MWIDTH 位‘10’，最低位两位被忽略，自动字对齐

16.5.9. 通道 x 存储器 1 基地址寄存器（DMA_CHxM1ADDR）

x = 0...7，x 为通道编号
地址偏移：0x20 + 0x18 * 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
M1ADDR[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M1ADDR[15:0]															
rw															

位/位域	名称	描述
31:0	M1ADDR[31:0]	存储器 1 基地址 若寄存器 DMA_CHxCTL 位 MBS 为 1，该位域定义 DMA 传输过程中存储器的基地址 如果寄存器 DMA_CHxCTL 的 CHEN 位置 1 且 MBS 为 1 时，该位域不可写 当 MWIDTH 位‘01’，最低位被忽略，自动半字对齐 当 MWIDTH 位‘10’，最低位两位被忽略，自动字对齐

16.5.10. 通道 x FIFO 控制寄存器（DMA_CHxFCTL）

x = 0...7，x 为通道编号
地址偏移：0x24 + 0x18 * x
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								FEEIE	保留	FCNT[2:0]			MDMEN	FCCV[1:0]	
								rw		r			rw	rw	

位/位域	名称	描述
31:8	保留	必须保持复位值。

7	FEEIE	FIFO 错误和异常中断使能位 软件置 1 与清 0 0：FIFO 错误和异常中断禁止 1：FIFO 错误和异常中断使能
6	保留	必须保持复位值。
5:3	FCNT[2:0]	FIFO 计数器 硬件置位和清零 000：FIFO 非空并且数据少于 1 个字 001：FIFO 数据多于 1 个字少于 2 个字 010：FIFO 数据多于 2 个字少于 3 个字 011：FIFO 数据多于 3 个字少于 4 个字 100：FIFO 空 101：FIFO 满 110~111：保留 该位域表明在数据传输过程 FIFO 中的数据量。若 MDMEN 为 0，则该位域无意义。
2	MDMEN	多数据传输模式使能 软件置位与清除 0：关闭多数据传输模式 1：打开多数据传输模式 CHEN 为 1 时不可写入 如果寄存器 DMA_CHxCTL 的 TM 位域为‘10’，在通道使能后，该位由硬件强制置 1
1:0	FCCV[1:0]	FIFO 计数器临界值 软件置位与清除 00：1 个字 01：2 个字 10：3 个字 11：4 个字 CHEN 为 1 时不可写入。 若 MDMEN 为‘0’，该位域无实际意义。

17. 主机直接存储器访问控制器（MDMA）

17.1. 简介

MDMA 控制器提供了一种硬件的方式在外设和存储器之间或者存储器和存储器之间传输数据，而无需 MCU 的介入，避免了 MCU 多次进入中断进行大规模的数据拷贝，最终提高整体的系统性能。

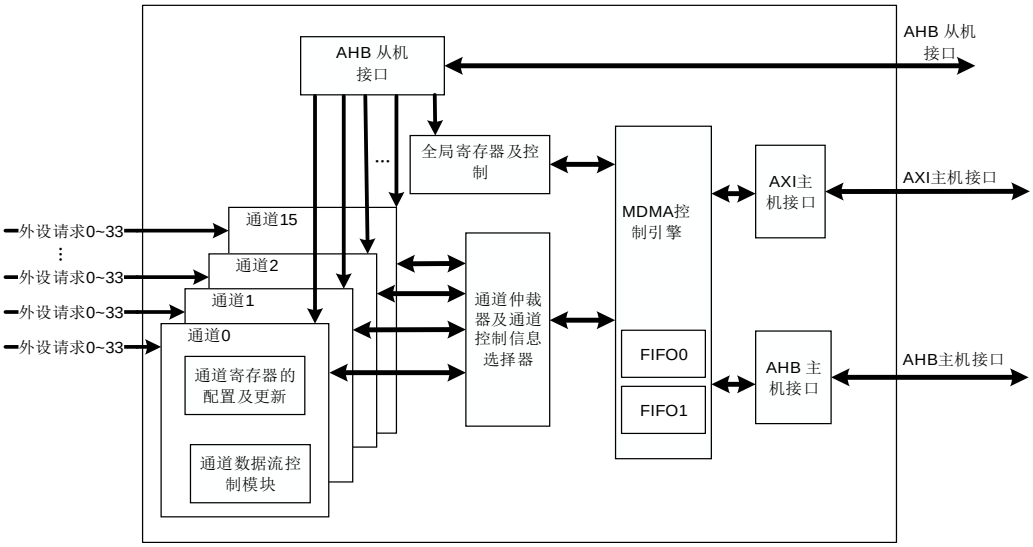
MDMA 控制器包含一个 AXI 总线接口、一个 AHB 总线接口以及两个 16 个双字深度的 FIFO，使 MDMA 可以高效的传输数据。其中 AXI 总线接口用于主存储器和外设寄存器访问（系统访问端口），AHB 总线接口用于 Cortex®-M7 TCM 存储器访问（TCM 访问端口）。MDMA 可以与 DMA 控制器（DMA0 或 DMA1）结合使用。MDMA 最多可提供 16 个通道，每个通道请求均可在任何请求源之间选择。内置总线仲裁器用来处理 MDMA 请求的优先级问题。

17.2. 主要特征

- AXI / AHB 主机接口，AXI 总线接口用于外设与存储器之间的数据传输，AHB 总线接口用于 Cortex®-M7 TCM 存储器的访问。
- 16 个通道，每个通道都支持软件触发且请求均可在任何请求源之间选择。
- 存储器和外设支持单一传输，2 拍，4 拍，8 拍，16 拍，32 拍，64 拍，128 拍增量突发传输。
- 支持软件优先级（低、中、高、超高）和硬件优先级（通道号越低，优先级越高）。
- 源和目标的数据传输宽度可配置：字节，半字，字，双字。
- 源和目标的数据传输支持固定寻址、递增式寻址和递减式寻址。
- 源和目标的数据长度及地址增量可配。
- 支持三种传输方式：
  - 存储器到存储器（软件触发）；
  - 外设到存储器（或存储器映射的外设）；
  - 存储器（或存储器映射的外设）到外设。
- 在源数据宽度和目标数据宽度不同的时候，自动打包/解包数据优化带宽。
- 34 个硬件触发源，所有通道均可连接至任意硬件触发源。
- 两个 16 个双字深度的 FIFO，用于最大化数据带宽和总线使用率。
- AHB 总线接口用于 Cortex®-M7 TCM 存储器访问时，仅当增量和数据大小相等且不大于 32 位时，支持突发访问。当增量和数据大小大于 32 位时，突发访问被禁止。
- 每个通道有 5 种类型的事件标志和独立的中断，支持中断的使能和清除。

17.3. 功能说明

图 17-1. MDMA 系统框图



如*图 17-1. MDMA 系统框图*所示，MDMA 控制器由 4 部分组成：

- AHB 从接口配置 MDMA；
- 一个 AXI 主接口和一个 AHB 主接口进行数据传输；
- 仲裁器进行 MDMA 请求的优先级管理；
- 数据处理和计数。

MDMA 控制器在没有 CPU 参与的情况下从一个地址向另一个地址传输数据，它支持多种数据宽度，突发类型，地址生成算法，优先级和传输模式，可以灵活的配置以满足应用的需求。所有的 MDMA 寄存器都可以通过 AHB 从机接口进行 32 位的操作。

MDMA_CHxCFG 寄存器中 TRIGMOD[1:0]决定了 MDMA 的数据传输模式，如*表 17-1. 传输模式*所示。

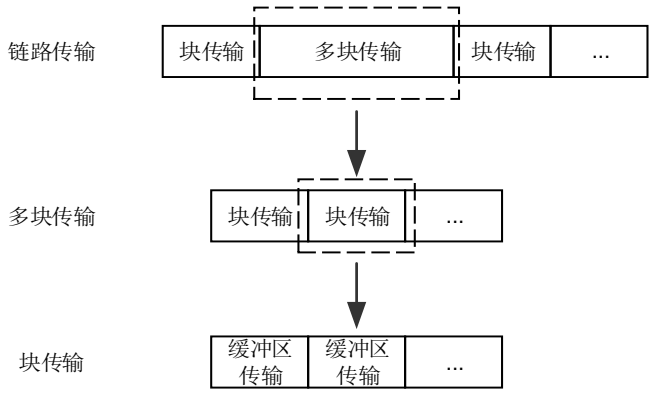
表 17-1. 传输模式

传输模式	TRIGMOD[1:0]
缓冲区传输	00
块传输	01
多块传输	10
链路传输	11

- 缓冲区传输一次最多传输 128 字节；
- 块传输一次最多传输 64KB，传输字节数可以通过 MDMA_CHxBTCFG 寄存器中 TBNUM[16:0]配置，传输过程由硬件自动拆分成多个缓冲区传输；
- 多块传输包含多个块传输，待传输块数可以通过 MDMA_CHxBTCFG 寄存器中 BRNUM[11:0]配置；
- 链路传输可以通过 MDMA_CHxLADDR 寄存器配置链路地址，包含多个块/多块传输。

四种模式之间的联系如*图17-2. 传输模式之间的联系*所示。

图 17-2. 传输模式之间的联系



MDMA 控制器共有 16 个通道，每个通道都支持软件触发且请求均可在如*表 17-2. MDMA 硬件请求源*所示任何请求源之间选择。通过配置 MDMA_CHxCTL1 寄存器中 TRIGSEL[5:0]位域，可以选择通道 x 硬件触发源。

表 17-2. MDMA 硬件请求源

请求源 TRIGSEL[5:0]	来源
0	DMA0_CH0_TRIG
1	DMA0_CH1_TRIG
2	DMA0_CH2_TRIG
3	DMA0_CH3_TRIG
4	DMA0_CH4_TRIG
5	DMA0_CH5_TRIG
6	DMA0_CH6_TRIG
7	DMA0_CH7_TRIG
8	DMA1_CH0_TRIG
9	DMA1_CH1_TRIG
10	DMA1_CH2_TRIG
11	DMA1_CH3_TRIG
12	DMA1_CH4_TRIG
13	DMA1_CH5_TRIG
14	DMA1_CH6_TRIG
15	DMA1_CH7_TRIG
16	TLI_INT
17	保留
18	保留
19	保留
20	保留
21	保留
22	OSPI0_FT
23	OSPI0_TC
24	IPA_CLUT_TRIG

请求源 TRIGSEL[5:0]	来源
25	IPA_TC_TRIG
26	IPA_TWM_TRIG
27	保留
28	保留
29	SDIO0_DATA_END
30	SDIO0_BUF_END
31	SDIO0_CMD_END
32	OSPI1_FT
33	OSPI1_TC

17.3.1. 数据处理

仲裁

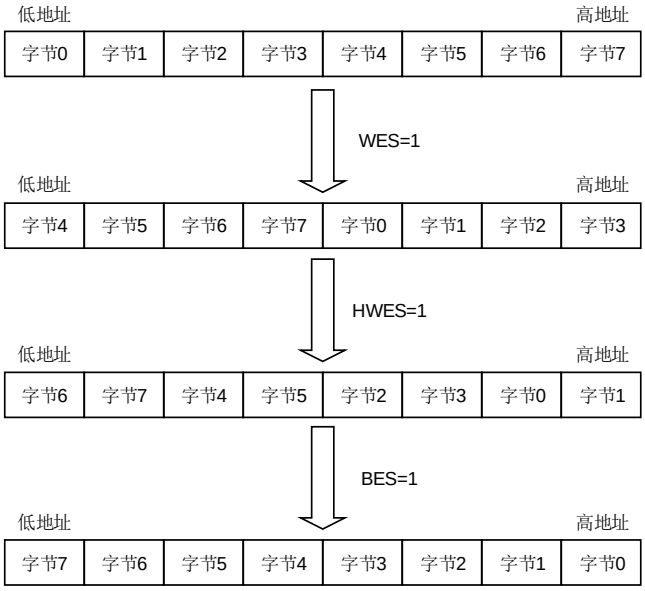
MDMA 通过仲裁器根据通道请求优先级对请求进行管理。当 MDMA 控制器在同一时间接收到多个外设请求时，仲裁器将根据外设请求的优先级来决定响应哪一个外设请求。优先级规则如下：

- 软件优先级：分为4级，包含低，中，高和超高。可以通过寄存器MDMA_CHxCTL0的PRIO[1:0]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。

数据交换模式

通过配置 MDMA_CHxCTL0 寄存器中 WES / HWES / BES 位，可实现对目标数据是否执行字、半字、字节顺序交换操作。交换过程如*图 17-3. 字、半字、字节顺序交换*所示。

图 17-3. 字、半字、字节顺序交换

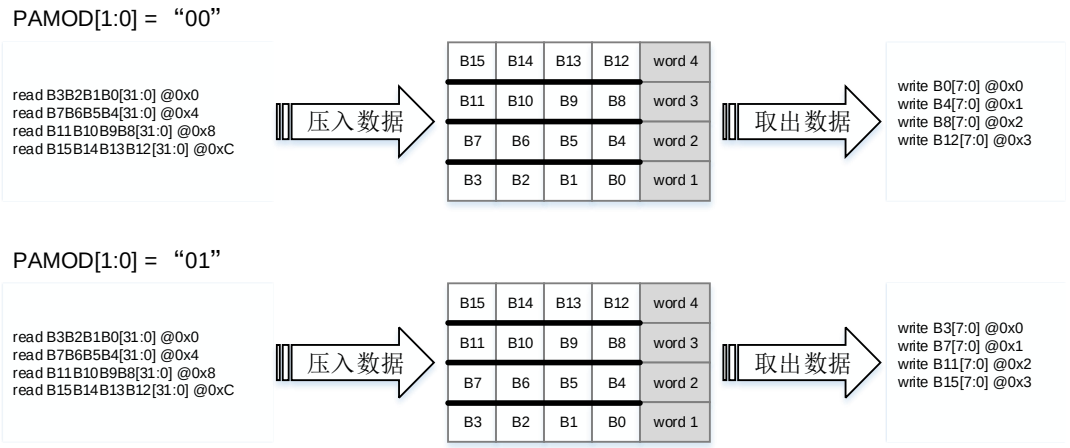


传输宽度

寄存器 MDMA_CHxCFG 的 SWIDTH[1:0]和 DWIDTH[1:0]位域分别决定了源数据宽度和目标数据宽度。MDMA 控制器支持 8 位，16 位，32 位和 64 位的数据宽度。当 PKEN 使能且 SWIDTH[1:0]和 DWIDTH[1:0]宽度不相等，MDMA 会自动打包/解包数据来进行数据传输以优化带宽。当 PKEN 禁能且 SWIDTH[1:0]和 DWIDTH[1:0]宽度不相等时，可以通过配置 MDMA_CHxCFG 寄存器中 PAMOD[1:0]位域选择填充和对齐方式。

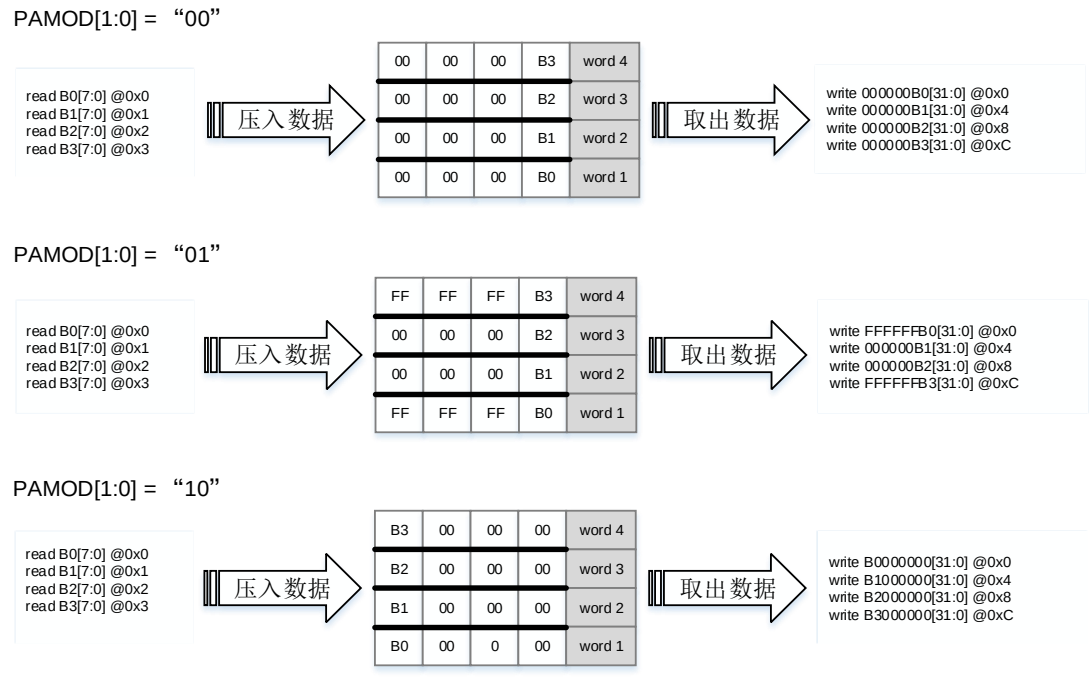
例如，当 SWIDTH[1:0] = 10（32 位），DWIDTH[1:0] = 00（8 位）时，填充和对齐方法如*图 17-4. 数据填充和对齐（源大于目的）*所示。

图 17-4. 数据填充和对齐（源大于目的）



假设 B0 和 B3 的最高位为 1，B1 和 B2 的最高位为 0，当 SWIDTH[1:0] = 00（8 位），DWIDTH[1:0] = 10（32 位）时，填充和对齐方法如*图 17-5. 数据填充和对齐（源小于目的）*所示。

图 17-5. 数据填充和对齐（源小于目的）

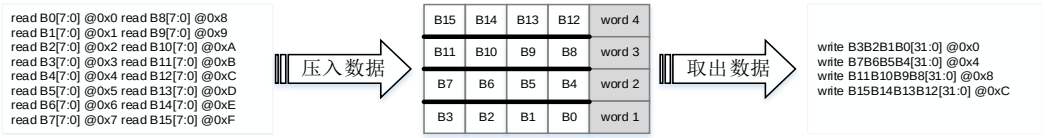


打包/解包

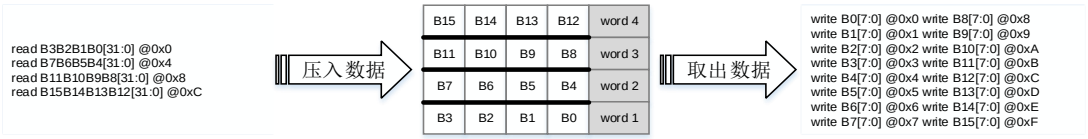
在 MDMA 传输中，源数据大小 SWIDTH 和目标数据大小 DWIDTH 相互独立，配置更为灵活。当 SWIDTH 和 DWIDTH 不相等时，MDMA 的读写传输宽度不同，MDMA 会自动的对数据打包/解包操作。将 MDMA_CHxCFG 寄存器 PKEN 位置 1，源数据将通过打包/解包的方式匹配目标数据大小。在对数据进行打包/解包时，采用小端模式。例如，当 SWIDTH[1:0] = 00，DWIDTH[1:0] = 10 时数据的打包以及 SWIDTH[1:0] = 10，DWIDTH[1:0] = 00 时解包过程如*图 17-6. 数据的打包/解包*所示。

图 17-6. 数据的打包/解包

- SWIDTH[1:0] = 00, DWIDTH[1:0] = 10



- SWIDTH[1:0] = 10, DWIDTH[1:0] = 00



突发传输

寄存器 MDMA_CHxCFG 的 SBURST[2:0]和 DBURST[2:0]位域决定了源和目标的突发传输方式。MDMA 控制器的源和目标均支持单一传输，2 拍，4 拍，8 拍，16 拍，32 拍，64 拍和 128 拍的增量突发传输。对于单数据传输模式，当使能通道后，SBURST[2:0]和 DBURST[2:0]会被强制设为 0。

注意：必须对 SBURST[2:0]和 DBURST[2:0]的值进行编程，以确保突发大小小于传输长度，否则，结果将无法预测。

FIFO

MDMA 控制器提供一个 256 字节大小的缓冲区，该缓冲区被分为两个深度为 16 个双字的 FIFO，并为所有通道共用。FIFO 用于在将源数据写入到目标之前，临时存储这些数据。

FIFO0 用于存储当前缓冲区待传输的数据，当 FIFO0 中数据量满足目标突发，MDMA 将会立即启动写操作。在缓冲区待传输的数据全部读到 FIFO0 时，仲裁器开始对通道优先级进行仲裁，并将下一个缓冲区待传输数据写入 FIFO1。

如果在缓冲区传输过程中出错，导致通道被禁止，则 FIFO0 和 FIFO1 中的数据将被丢弃。

17.3.2. 地址生成

源和目标都独立的支持三种地址生成算法：固定模式、递增模式和递减模式。寄存器 MDMA_CHxCFG 的 DIMOD[1:0]和 SIMOD[1:0]位域分别用于配置目的和源的地址生成算法，如*表 17-3. 源和目标地址生成配置*所示。

表 17-3. 源和目标地址生成配置

SIMOD[1:0]		DIMOD[1:0]	
00	无增量	00	无增量
10	源地址增量为 SIOS	10	目标地址增量为 DIOS
11	源地址减量为 SIOS	11	目标地址减量为 DIOS

在固定模式下，SIMOD[1:0]或 DIMOD[1:0]配置为“00”，源或目的地址一直固定为初始化的基地址（MDMA_CHxSADDR 和 MDMA_CHxDADDR）。

在递增或递减模式下，下一次传输数据的地址是当前地址加/减 1（或者 2，4，8），这个值取决于 MDMA_CHxCFG 寄存器中 SIOS[1:0]或 DIOS[1:0]的配置。

为优化打包操作，可独立编程增量大小和数据大小。

17.3.3. 传输模式

缓冲区传输模式

MDMA 控制器支持单一传输，2 拍，4 拍，8 拍，16 拍，32 拍，64 拍和 128 拍传输。寄存器 MDMA_CHxCFG 的 SBURST[2:0]和 DBURST[2:0]位域决定了源和目标的突发传输方式。缓冲区传输是以单次或突发方式对数据进行传输。MDMA_CHxCFG 寄存器中 SWIDTH[1:0]和 DWIDTH[1:0]用于配置源和目标数据宽度。

当 MDMA 接收到请求时，仲裁器会根据 MDMA 通道请求优先级对其进行管理。如果 MDMA_CHxMADDR 寄存器不为 0，向 MADDR[31:0]指定的地址写入掩码数据时会确认该请求。否则，向发出请求的外设写入或者读取数据会复位请求。如果由目标外设完成请求，则必须将 MDMA_CHxCFG 寄存器中 BWMOD 清零，以避免出现错误的新 MDMA 请求。

如果 MDMA_CHxCFG 寄存器中 TRIGMOD[1:0]为 00，MDMA 会在一个缓冲区的数据传输完成后在同一个通道（如通道 A）上等待另一个请求。

- 如果在通道 A 下一次请求还未发生，其他通道（如通道 B）发生了请求，不管通道 B 的优先级是否高于通道 A，都将响应通道 B 的请求；
- 如果在通道 A 一次缓冲区传输完成后检测到下一次请求，并且同时其他通道（通道 C）发生了请求，仲裁器会根据 MDMA 通道请求优先级对请求事件进行管理。

当缓冲区传输完成时，MDMA_CHxSTAT0 寄存器中 TCF 位将置 1。通过将 MDMA_CHxSTATC 寄存器中 TCFC 位写 1 可以清除 TCF 位。

如果 TRIGMOD[1:0]不为 00，且待传输数据总量大于 128 字节，则当每一次缓冲区传输完成后，仲裁器会根据 MDMA 通道请求优先级对请求事件进行仲裁。如果不存在优先级更高的其他请求，则会继续进行下一次缓冲区传输。如果有优先级更高的其他请求，MDMA 将优先处理优先级别高的请求。

注意：当 TRIGMOD[1:0]为 00，处于缓冲区传输模式时：

- 当BTLEN[6:0]≥TBNUM[15:0]，将传输 TBNUM[15:0]个字节数据。
- 当BTLEN[6:0]<TBNUM[15:0]，将传输（BTLEN[6:0] + 1）个字节数据。

块传输模式

在块传输模式下，块大小由 MDMA_CHxBTCFG 寄存器中 TBNUM[16:0]位域来配置，块中待传输字节数最大为 64KB。当 TBNUM[16:0]计数到 0，块传输完成，MDMA_CHxSTAT0 寄存器中 TCF 位、BTCF 位和 CHTCF 位将置 1。MDMA_CHxCTL0 寄存器中 CHEN 位将被硬件清零，该通道将不继续接受 MDMA 请求。

在多块传输模式下，如果当前块不是最后一块，在当前块传输完成后，硬件将自动重载第一次块传输长度，并根据 MDMA_CHxMBADDRU 寄存器中 DADDRUV 和 SADDRUV 的值以及 MDMA_CHxBTCFG 寄存器中 SADDRUM 位和 DADDRUM 位，计算新的源地址和目标地址，并进行下一次块传输。如果当前块是最后一块，当 TBNUM[16:0]计数到 0，块传输完成，MDMA_CHxSTAT0 寄存器中 TCF 位、BTCF 位、MBTCF 位和 CHTCF 位将置 1。MDMA_CHxCTL0 寄存器中 CHEN 位将被硬件清零，该通道将不继续接受 MDMA 请求。

在链路模式下，如果当前块是单块或者多块的最后一块且 MDMA_CHxLADDR 不为 0，当前块传输完成后，将根据 MDMA_CHxLADDR 寄存器中 LADDR 指定的地址处加载新块配置信息，并开始新的块 / 多块传输。如果当前块是单块或者多块的最后一块且 MDMA_CHxLADDR 为 0，MDMA_CHxSTAT0 寄存器中 TCF 位、MBTCF / BTCF 位和 CHTCF 位将置 1。MDMA_CHxCTL0 寄存器中 CHEN 位将被硬件清零，该通道将不继续接受 MDMA 请求。

当块的大小不是源或目标数据大小的整数倍时，MDMA_CHxSTAT1 寄存器中 BZERR 位将硬件置 1。通过对 MDMA_CHxSTATC 寄存器 ERRC 位写 1 可以清除 BZERR 位。

将 MDMA_CHxSTATC 寄存器中 TCFC 位，BTCFC 位，MBTCF 位和 CHTCFC 位写 1 可以分别清除 TCF 位，BTCF 位，MBTCF 位和 CHTCF 位。

多块传输模式

MDMA_CHxBTCFG 寄存器中 BRNUM[11:0]可配置待传输块数，当 BRNUM[11:0]不为 0 时，多块传输模式被使能。BRNUM[11:0]可配置为 0~4095，当完成一次块传输时，BRNUM 的值减 1，并且下一次块传输的源地址和目标地址会根据 MDMA_CHxBTCFG 寄存器中 SADDRUM 位和 DADDRUM 位配置的源地址和目的地址更新方式更新 MDMA_CHxSADDR 寄存器和 MDMA_CHxDADDR 寄存器的值。源和目的地址更新方式如*表 17-4. 源和目的地址更新方式*所示。MDMA_CHxBTCFG 寄存器中 TBNUM[16:0]将重载第一次块传输时编程的值。当最后一块传输完成时，MDMA_CHxSTAT0 寄存器中 TCF 位、BTCF 位、MBTCF 位和 CHTCF 位将置 1，MDMA_CHxCTL0 寄存器中 CHEN 位将被硬件清零，该通道将不继续接受 MDMA 请求。通过对 MDMA_CHxSTATC 寄存器中 TCFC 位、BTCFC 位、MBTCFC 位和 CHTCFC 位写 1 可以分别将 TCF 位、BTCF 位、MBTCF 位和 CHTCF 位清除。

表 17-4. 源和目的地址更新方式

源/目的地址	更新方式配置	更新后源/目的地址
SADDR	SADDRUM = 0	SADDR = SADDR + SADDRUV
	SADDRUM = 1	SADDR = SADDR - SADDRUV
DADDR	DADDRUM = 0	DADDR = DADDR + DADDRUV
	DADDRUM = 1	DADDR = DADDR - DADDRUV

注意：当 BRNUM[11:0]计数为 0 时，会将最后一个块传输视为单块传输。

链路传输模式

在链路模式下，当多块 / 块传输结束后，当前通道的配置寄存器包括 MDMA_CHxCFG，MDMA_CHxBTCFG，MDMA_CHxSADDR，MDMA_CHxDADDR，MDMA_CHxMBADDRU，MDMA_CHxLADDR，MDMA_CHxCTL1，MDMA_CHxMADDR 和 MDMA_CHxMDATA 将使用 MDMA_CHxLADDR 寄存器中定义的地址 LADDR[31:0]处的数据结构对配置寄存器进行加载。如*表 17-5. 寄存器加载地址*所示。如果 MDMA_CHxCFG 寄存器中 TRIGMOD[1:0] = 11，在进行配置寄存器加载后，通道将接受新的请求或继续传输。

表 17-5. 寄存器加载地址

寄存器	加载地址
MDMA_CHxCFG	LADDR[31:0] + 0x00
MDMA_CHxBTCFG	LADDR[31:0] + 0x04
MDMA_CHxSADDR	LADDR[31:0] + 0x08
MDMA_CHxDADDR	LADDR[31:0] + 0x0C
MDMA_CHxMBADDRU	LADDR[31:0] + 0x10
MDMA_CHxLADDR	LADDR[31:0] + 0x14
MDMA_CHxCTL1	LADDR[31:0] + 0X18
MDMA_CHxMADDR	LADDR[31:0] + 0x20
MDMA_CHxMDATA	LADDR[31:0] + 0x24

如果在对通道配置寄存器加载时，MDMA_CHxCTL1 寄存器中 TRIGSEL[5:0]发生改变，则硬件将自动切换触发源。

注意：在链路传输模式下，MDMA_CHxCFG 寄存器中 SWREQMOD 位和 TRIGMOD[1:0]不能被修改。

17.3.4. 传输状态

传输完成

MDMA_CHxBTCFG 寄存器中 TBNUM[16:0]，BRNUM[11:0]和 MDMA_CHxLADDR 寄存器中 LADDR[31:0]均为 0 时，或者在传输结束前，禁止了通道（CHEN = 0），并且 FIFO 中剩余数据均传输到目标时，通道传输完成后，MDMA_CHxSTAT0 寄存器中 CHTCF 位将置 1。

传输中断

传输中断是指在传输过程中将 MDMA_CHxCTL0 寄存器中 CHEN 清零禁止通道，并且在重新使能通道时不继续上一次的数据传输。在通道禁止后，当 FIFO 中剩余数据均传输到目标时，MDMA_CHxSTAT0 寄存器中 CHTCF 位将置 1。通过 MDMA_CHxBTCFG 寄存器中 TBNUM[16:0]，BRNUM[11:0]可以查看未传输的字节数或块数。

传输暂停

在 MDMA_CHxBTCFG 寄存器中 TBNUM[16:0]计数达到 0 之前，将 MDMA_CHxCTL0 寄存器中 CHEN 清零可以暂停通道传输。当 MDMA_CHxSTAT0 寄存器中 CHTCF 位将置 1 时，

表明 FIFO 中剩余数据已传输完成。如果 MDMA_CHxBTCFG 寄存器，MDMA_CHxSADDR 寄存器以及 MDMA_CHxDADDR 寄存器的值未被软件修改，将 MDMA_CHxSTAT0 寄存器中 CHTCF 位清零并将 CHEN 位重新使能后会继续进行数据传输。

注意：当 TRIGMOD[1:0]为 11 时，建议将下一个节点的数据结构中的 LADDR 字段配置为 0，以暂停通道传输。如果通过清除 MDMA_CHxCTL0 寄存器中的 CHEN 来暂停通道传输，则不能保证结果正确性。

17.3.5. MDMA 错误和中断

MDMA 错误标志如*表 17-6. MDMA 错误标志*所示。

表 17-6. MDMA 错误标志

错误名称	描述
BZERR	块大小错误标志
ASERR	地址和大小错误标志
MDTERR	掩码数据传输错误标志
LDTERR	链路数据传输错误标志
ERR	传输错误标志

当发生下列情况时，传输错误标志（ERR）将置 1：

- MDMA 读或写访问期间发生总线错误；
- 地址对齐的位置与数据的大小不匹配；
- 块大小不是（源和/或目标）数据大小的倍数。

对于每个 MDMA 通道，中断事件有五种类型：通道传输完成，缓冲区传输完成，块传输完成，多块传输完成和传输错误。

寄存器 MDMA_CHxSTAT0 包含每个中断事件的标志位，寄存器 MDMA_CHxSTATC 包含每个中断事件的标志清除位，寄存器 MDMA_CHxCTL0 包含每个中断事件的使能位，如*表 17-7. MDMA 中断事件*所示。

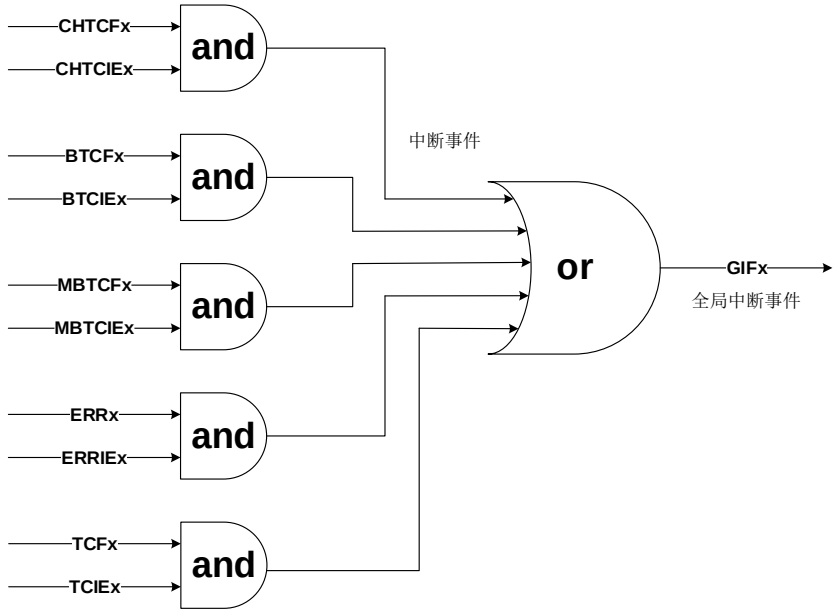
表 17-7. MDMA 中断事件

中断事件	标志位	使能位	清除位
	MDMA_CHxSTAT0	MDMA_CHxCTL0	MDMA_CHxSTATC
通道传输完成	CHTCF	CHTCIE	CHTCFC
缓冲区传输完成	TCF	TCIE	TCFC
块传输完成	BTCF	BTCIE	BTCFC
多块传输完成	MBTCF	MBTCIE	MBTCFC
传输错误	ERR	ERRIE	ERRC

当通道 x 的 BTCF / MBTCF / CHTCF / ERR / TCF 至少有一个标志位置位，并且相应的中断（BTCIE / MBTCIE / CHTCIE / ERRIE / TCIE）已使能，MDMA_GINTF 寄存器中 GIFx 将置 1，如果再 NVIC 中 MDMA 中断已使能，将产生一个中断。

MDMA 中断逻辑如*图 17-7. MDMA 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 17-7. MDMA 中断逻辑图



注意：“x”表示通道数（对应 x=0...15）。

17.4. MDMA 寄存器

MDMA 基地址：0x5200 0000

17.4.1. 全局中断标志寄存器（MDMA_GINTF）

地址偏移：0x00
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															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GIF15	GIF14	GIF13	GIF12	GIF11	GIF10	GIF9	GIF8	GIF7	GIF6	GIF5	GIF4	GIF3	GIF2	GIF1	GIF0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	GIFx	通道 x 全局中断标志（x=0…15） 0：通道 x 标志位（BTCF / MBTCF / CHTCF / ERR / TCF）均未置位，或有标志位置位但其相应的中断未使能。 1：通道 x BTCF / MBTCF / CHTCF / ERR / TCF 至少有一个标志位置位，并且相应的中断（BTCIE / MBTCIE / CHTCIE / ERRIE / TCIE）已使能。

17.4.2. 通道 x 状态寄存器 0（MDMA_CHxSTAT0）

x = 0...15，x 为通道编号
地址偏移：0x40 + 0x40 × x
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															REQAF
															r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											TCF	BTCF	MBTCF	CHTCF	ERR
											r	r	r	r	r

位/位域	名称	描述
31:17	保留	必须保持复位值。
16	REQAF	通道 x 请求激活标志 将 MDMA_CHxCTL0 寄存器中 SWREQ 位置 1，并且使能 CHEN，该位将置 1。当

通道 x 请求完成时，该位由硬件清零。
0：通道 x MDMA 传输未激活。
1：通道 x MDMA 传输激活。

15:5	保留	必须保持复位值。
4	TCF	通道 x 缓冲区传输完成标志 硬件置位，软件写 MDMA_CHxSTATC 相应位为 1 清零。 0：通道 x 缓冲区传输未完成。 1：通道 x 缓冲区传输完成。
3	BTCF	通道 x 块传输完成标志 硬件置位，软件写 MDMA_CHxSTATC 相应位为 1 清零。 0：通道 x 块传输未完成。 1：通道 x 块传输完成。
2	MBTCF	通道 x 多块传输完成标志 硬件置位，软件写 MDMA_CHxSTATC 相应位为 1 清零。 0：通道 x 多块传输未完成。 1：通道 x 多块传输完成。
1	CHTCF	通道 x 通道传输完成标志 硬件置位，软件写 MDMA_CHxSTATC 相应位为 1 清零。 0：通道 x 传输未完成。 1：通道 x 传输完成。 **注意：**当 CHEN 写 0 时，CHTCF 位也将置 1。
0	ERR	通道 x 传输错误标志 硬件置位，软件写 MDMA_CHxSTATC 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。

17.4.3. 通道 x 状态清除寄存器（MDMA_CHxSTATC）

x = 0...15，x 为通道编号
地址偏移：0x44 + 0x40 × x
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											TCFC	BTCFC	MBTCFC	CHTCFC	ERRC
											w	w	w	w	w

位/位域	名称	描述

31:5	保留	必须保持复位值。
4	TCFC	通道 x 缓冲区传输完成标志清零 0：无影响。 1：对该位写 1 清零 MDMA_CHxSTAT0 寄存器中 TCF 位。
3	BTCFC	通道 x 块传输完成标志清零 0：无影响。 1：对该位写 1 清零 MDMA_CHxSTAT0 寄存器中 BTCF 位。
2	MBTCFC	通道 x 多块传输完成标志清零 0：无影响。 1：对该位写 1 清零 MDMA_CHxSTAT0 寄存器中 MBTCF 位。
1	CHTCFC	通道 x 传输完成标志清零 0：无影响。 1：对该位写 1 清零 MDMA_CHxSTAT0 寄存器中 CHTCF 位。
0	ERRC	通道 x 传输错误标志清零 0：无影响。 1：对该位写 1 清零 MDMA_CHxSTAT0 寄存器中 ERR 位。

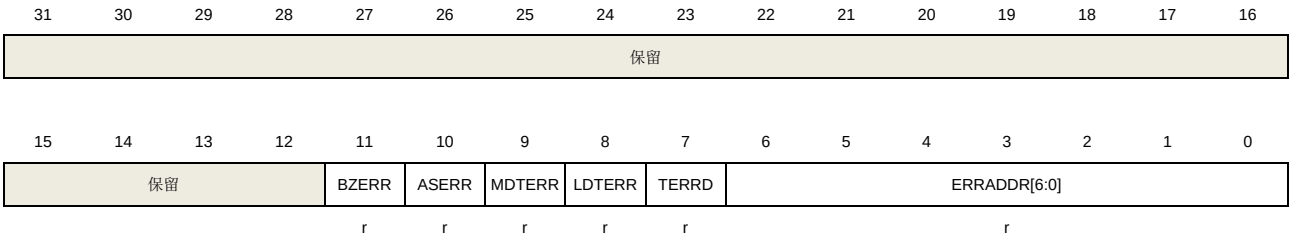
17.4.4. 通道 x 状态寄存器 1（MDMA_CHxSTAT1）

x = 0...15，x 为通道编号

地址偏移：0x48 + 0x40 × x

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
31:12	保留	必须保持复位值。
11	BZERR	块大小错误标志 当块的大小或 BTLEN+1 不是源或目标数据大小的整数倍时，该位由硬件置 1。将 MDMA_CHxSTATC 寄存器 ERRC 位写 1 可清零该位。 0：未发生块大小错误。 1：发生了块大小错误。
10	ASERR	地址和大小错误标志 当地址与数据大小不匹配时，该位由硬件置 1。将 MDMA_CHxSTATC 寄存器 ERRC

位写 1 可清零该位。
0：未发生地址和大小错误。
1：发生了地址和大小错误。

9	MDTERR	掩码数据错误标志 当写入掩码数据产生错误时，该位由硬件置 1。将 MDMA_CHxSTATC 寄存器 ERRC 位写 1 可清零该位。 0：未发生掩码数据错误。 1：发生了掩码数据错误。
8	LDTERR	链路数据错误标志 当读取块链路数据结构时产生错误，该位由硬件置 1。将 MDMA_CHxSTATC 寄存器 ERRC 位写 1 可清零该位。 0：未发生链路数据错误。 1：发生了链路数据错误。
7	TERRD	传输错误方向标志 当通道上传输错误由写访问产生时，该位由硬件置 1。 0：传输错误由读访问产生。 1：传输错误由写访问产生。
6:0	ERRADDR[6:0]	传输错误地址 当传输错误发生时，这些位存储错误地址的低 7 位。绝对错误地址为 ERRADDR + SADDR / DADDR。 **注意：**当重载错误发生时，这些位被忽略。

17.4.5. 通道 x 控制寄存器 0（MDMA_CHxCTL0）

x = 0...15，x 为通道编号
地址偏移：0x4C + 0x40 × x
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															SWREQ
															w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	WES	HWES	BES	保留			SMODEN	PRIO[1:0]		TCIE	BTCIE	MBTCIE	CHTCIE	ERRIE	CHEN
	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	SWREQ	软件请求 当通道使能时，将该位置 1 将激活通道 x 的请求，MDMA_CHxSTAT0 寄存器中 REQAF 位将置 1。

15	保留	必须保持复位值。
14	WES	双字中字的顺序交换 0：双字中不交换字的顺序，保持小端字节顺序。 1：双字中交换字的顺序。 **注意：**如果目标不是双字，则该位忽略。当通道被使能（CHEN=1）时，该位不能被改写。
13	HWES	字中半字的顺序交换 0：字中不交换半字的顺序，保持小端字节顺序。 1：字中交换半字的顺序。 **注意：**如果目标不是字或双字，则该位忽略。当通道被使能（CHEN=1）时，该位不能被改写。
12	BES	半字中字节的顺序交换 0：字中不交换半字的顺序，保持小端字节顺序。 1：字中交换半字的顺序。 **注意：**如果目标不是半字、字或双字，则该位忽略。当通道被使能（CHEN=1）时，该位不能被改写。
11:9	保留	必须保持复位值。
8	SMODEN	安全模式使能 0：安全模式禁能 1：安全模式使能 该位仅在 AHB 从机接口处于安全模式下可写。如果 SMODEN 为 0，当前通道的所有寄存器可写。如果 SMODEN 为 1，当前通道的所有寄存器被写保护。 **注意：**当通道被使能（CHEN=1）时，该位不能被改写。
7:6	PRIO[1:0]	软件优先级 软件置 1 与清零。 00：低 01：中 10：高 11：超高 **注意：**当通道被使能（CHEN=1）时，该位不能被改写。
5	TCIE	缓冲区传输完成中断使能 软件置 1 与清零。 0：缓冲区传输完成中断禁能。 1：缓冲区传输完成中断使能。
4	BTCIE	块传输完成中断使能 软件置 1 与清零。 0：块传输完成中断禁能。 1：块传输完成中断使能。
3	MBTCIE	多块传输完成中断使能 软件置 1 与清零。

0：多块传输完成中断禁能。
1：多块传输完成中断使能。

2	CHTCIE	通道传输完成中断使能 软件置 1 与清零。 0：通道传输完成中断禁能。 1：通道传输完成中断使能。
1	ERRIE	传输错误中断使能 软件置 1 与清零。 0：传输错误中断禁能。 1：传输错误中断使能。
0	CHEN	通道使能 软件置 1 与清零。 0：通道禁能。 1：通道使能。 **注意：**当 MDMA 传输完成、发生 AHB/AXI 总线错误、发生 BZERR 错误或 ASERR 错误时，该位由硬件清零。

17.4.6. 通道 x 配置寄存器（MDMA_CHxCFG）

x = 0...15，x 为通道编号
地址偏移：0x50 + 0x40 × x
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
BWMOD	SWREQMOD	TRIGMOD[1:0]		PAMOD[1:0]		PKEN	BTLEN[6:0]							DBURST[2:1]	
rw	rw	rw		rw		rw	rw							rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBURST[0]	SBURST[2:0]			DIOS[1:0]		SIOS[1:0]		DWIDTH[1:0]		SWIDTH[1:0]		DIMOD[1:0]		SIMOD[1:0]	
rw	rw			rw		rw		rw		rw		rw		rw	

位/位域	名称	描述
31	BWMOD	可缓冲写模式 软件置 1 与清零。 0：不可缓冲写。 1：可缓冲写。 **注意：**当通道被使能（CHEN=1）时，该位不能被改写。
30	SWREQMOD	软件请求模式 软件置 1 与清零。 0：响应软件请求和硬件请求。

1：只响应软件请求。

注意：修改该位将在当前传输完成后生效。

29:28 TRIGMOD[1:0] 触发模式

软件置 1 与清零。

00：软件请求或硬件请求触发缓冲区传输。

01：软件请求或硬件请求触发块传输。

10：软件请求或硬件请求触发多块传输。

11：软件请求或硬件请求触发完整数据传输（如链路模式）。

注意：当通道被使能（CHEN=1）时，该位不能被改写。

27:26 PAMOD[1:0] 填充和对齐模式

软件置 1 与清零。

源数据大小大于目标数据大小		源数据大小小于目标数据大小	
00	右对齐，将源的低字节部分写入目标地址，高字节部分丢弃	00	右对齐，不足的位补 0
01	保留	01	右对齐，符号扩展
10	左对齐，将源的高字节部分写入目标地址，低字节部分丢弃	10	左对齐，不足的位补 0
11	保留	11	保留

注意：当包使能（PKEN=1）或源数据大小等于目标数据大小时，该位域无效。当通道被使能（CHEN=1）时，该位域不能被改写。

25 PKEN 包使能

软件置 1 与清零。

0：根据 PAMOD[1:0]位域配置的方式将源数据写入目标地址。

1：将源数据通过打包/解包方式以匹配目标数据大小。

注意：当通道被使能（CHEN=1）时，该位不能被改写。

24:18 BTLEN[6:0] 缓冲区传输长度

软件置 1 与清零。

单次传输的字节数请参考*缓冲区传输模式*。

注意：BTLEN+1 必须是 DWIDTH 和 SWIDTH 的倍数。

17:15 DBURST[2:0] 目标传输突发类型

软件置 1 与清零。

000：单一传输。

001：2 拍增量突发传输。

010：4 拍增量突发传输。

011：8 拍增量突发传输。

100：16 拍增量突发传输。

101：32 拍增量突发传输。

110：64 拍增量突发传输。

111：128 拍增量突发传输。

		注意：当通道被使能（CHEN=1）时，该位不能被改写。
14:12	SBURST[2:0]	源传输突发类型 软件置 1 与清零。 000：单一传输。 001：2 拍增量突发传输。 010：4 拍增量突发传输。 011：8 拍增量突发传输。 100：16 拍增量突发传输。 101：32 拍增量突发传输。 110：64 拍增量突发传输。 111：128 拍增量突发传输。 **注意：**当通道被使能（CHEN=1）时，该位不能被改写。
11:10	DIOS[1:0]	目标增量偏移 软件置 1 与清零。 00：8 位 01：16 位 10：32 位 11：64 位 **注意：**当通道被使能（CHEN=1）时，该位不能被改写。如果 DIOS 小于 DWIDTH 且 DIMOD 不为 00，结果将不预测。
9:8	SIOS[1:0]	源增量偏移 软件置 1 与清零。 00：8 位 01：16 位 10：32 位 11：64 位 **注意：**当通道被使能（CHEN=1）时，该位不能被改写。如果 SIOS 小于 SWIDTH 且 SIMOD 不为 00，结果将不预测。
7:6	DWIDTH[1:0]	目标数据大小 软件置 1 与清零。 00：8 位 01：16 位 10：32 位 11：64 位 **注意：**当通道被使能（CHEN=1）时，该位不能被改写。
5:4	SWIDTH[1:0]	源数据大小 软件置 1 与清零。 00：8 位 01：16 位 10：32 位 11：64 位

注意：当通道被使能（CHEN=1）时，该位不能被改写。

3:2	DIMOD[1:0]	目标增量模式 软件置 1 与清零。 00：无增量 01：保留 10：目标地址增量为 DIOS 11：目标地址减量为 DIOS **注意：**当通道被使能（CHEN=1）时，该位不能被改写。
1:0	SIMOD[1:0]	源增量模式 软件置 1 与清零。 00：无增量 01：保留 10：源地址增量为 SIOS 11：源地址减量为 SIOS **注意：**当通道被使能（CHEN=1）时，该位不能被改写。

17.4.7. 通道 x 块传输配置寄存器（MDMA_CHxBTCFG）

x = 0...15，x 为通道编号

地址偏移：0x54 + 0x40 × x

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
BRNUM[11:0]												DADDRUM	SADDRUM	保留	TBNUM[16]
rw												rw	rw		rw

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

位/位域	名称	描述
31:20	BRNUM[11:0]	待传输多块块数 **注意：**当通道被使能（CHEN=1）时，该位域不能被改写。
19	DADDRUM	在多块传输模式下，目标地址更新模式 0：DADDR = DADDR + DADDRUV 1：DADDR = DADDR - DADDRUV **注意：**当通道被使能（CHEN=1）时，该位域不能被改写。
18	SADDRUM	在多块传输模式下，源地址更新模式 0：SADDR = SADDR + SADDRUV 1：SADDR = SADDR - SADDRUV **注意：**当通道被使能（CHEN=1）时，该位域不能被改写。

17	保留	必须保持复位值。
16:0	TBNUM[16:0]	块中待传输字节数 当前块待传输的字节数（0-65536）。在多块传输模式下，当块传输完成后，该位将自动重载第一次编程的值。 **注意：**当通道被使能（CHEN=1）时，该位域不能被改写。TBNUM 必须是源和目标数据大小的整数倍。

17.4.8. 通道 x 源地址寄存器（MDMA_CHxSADDR）

x = 0...15，x 为通道编号

地址偏移：0x58 + 0x40 × x

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
SADDR[31:16]															
rw															

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

位/位域	名称	描述
31:0	SADDR[31:0]	源地址

17.4.9. 通道 x 目的地址寄存器（MDMA_CHxDADDR）

x = 0...15，x 为通道编号

地址偏移：0x5C + 0x40 × x

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
DADDR[31:16]															
rw															

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

位/位域	名称	描述
31:0	DADDR[31:0]	目标地址

17.4.10. 通道 x 多块地址更新寄存器（MDMA_CHxMBADDRU）

x = 0...15，x 为通道编号

地址偏移：0x60 + 0x40 × x
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
DADDRUV[11:0]															
rw															

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

位/位域	名称	描述
31:16	DADDRUV[15:0]	目标地址更新值 该位域用于设置块传输结束后，目标地址的增量或减量。为了使 DADDR 与 DWIDTH 对齐，该位域设置的值必须为 DWIDTH 的整数倍。当 BRNUM=0 时，这些位无效。 **注意：**当通道被使能（CHEN=1）时，该位域不能被改写。
15:0	SADDRUV[15:0]	源地址更新值 该位用于设置块传输结束后，源地址的增量或减量。为了使 SADDR 与 SWIDTH 对齐，该位设置的值必须为 SWIDTH 的整数倍。当 BRNUM=0 时，这些位无效。 **注意：**当通道被使能（CHEN=1）时，该位域不能被改写。

17.4.11. 通道 x 链路地址寄存器（MDMA_CHxLADDR）

x = 0...15，x 为通道编号
地址偏移：0x64 + 0x40 × x
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
LADDR[31:16]															
rw															

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

位/位域	名称	描述
31:0	LADDR[31:0]	链路地址 如果该位域的值不为 0，则在块 / 多块传输完成后，当前通道的配置寄存器包括 MDMA_CHxCFG，MDMA_CHxBTCFG，MDMA_CHxSADDR，MDMA_CHxDADDR，MDMA_CHxMBADDRU，MDMA_CHxLADDR，MDMA_CHxCTL1，MDMA_CHxMADDR 和 MDMA_CHxMDATA 将使用 MDMA_CHxLADDR 寄存器中定义的地址 LADDR[31:0]处的数据结构对配置寄存器进行加载。 如果该位域的值为 0，MDMA_CHxSTAT0 寄存器中 CHTCF 位将置 1，且 CHEN 将

由硬件清零。

注意：1、当通道被使能（CHEN=1）时，该位域不能被改写。

2、LADDR[31:0]的值必须是双字对齐。

17.4.12. 通道 x 控制寄存器 1（MDMA_CHxCTL1）

x = 0...15，x 为通道编号

地址偏移：0x68 + 0x40 × x

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														DBSEL	SBSEL
														rw	rw

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

位/位域	名称	描述
31:18	保留	必须保持复位值。
17	DBSEL	目标总线选择 该位用于设置在写操作时，选择通道 x 的目标总线。 0：通道 x 的目标总线是系统总线或 AXI 总线 1：通道 x 的目标总线是 AHB 总线或 TCM **注意：**当通道被使能（CHEN=1）时，该位不能被改写。
16	SBSEL	源总线选择 该位用于设置在读操作时，选择通道 x 的源总线。 0：通道 x 的源总线是系统总线或 AXI 总线 1：通道 x 的源总线是 AHB 总线或 TCM **注意：**当通道被使能（CHEN=1）时，该位不能被改写。
15:6	保留	必须保持复位值。
5:0	TRIGSEL[5:0]	触发选择 该位域用于选择通道 x 的硬件触发源。 如果 SWREQMOD 位为 1，则该位忽略。 **注意：**当通道被使能（CHEN=1）时，该位不能被改写。

17.4.13. 通道 x 掩码地址寄存器（MDMA_CHxMADDR）

x = 0...15，x 为通道编号

地址偏移：0x70 + 0x40 × x

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
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]	掩码地址 当该位域不为 0 时，通过对 MADDR 指定的地址写入 MDMA_CHxMDATA 寄存器中 MDATA 值会确认 DMA 请求。

17.4.14. 通道 x 掩码数据寄存器（MDMA_CHxMDATA）

x = 0...15，x 为通道编号

地址偏移：0x74 + 0x40 × x

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
MDATA[31:16]															
rw															

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

位/位域	名称	描述
31:0	MDATA[31:0]	掩码数据

18. DMA 请求多路复用器（DMAMUX）

18.1. 简介

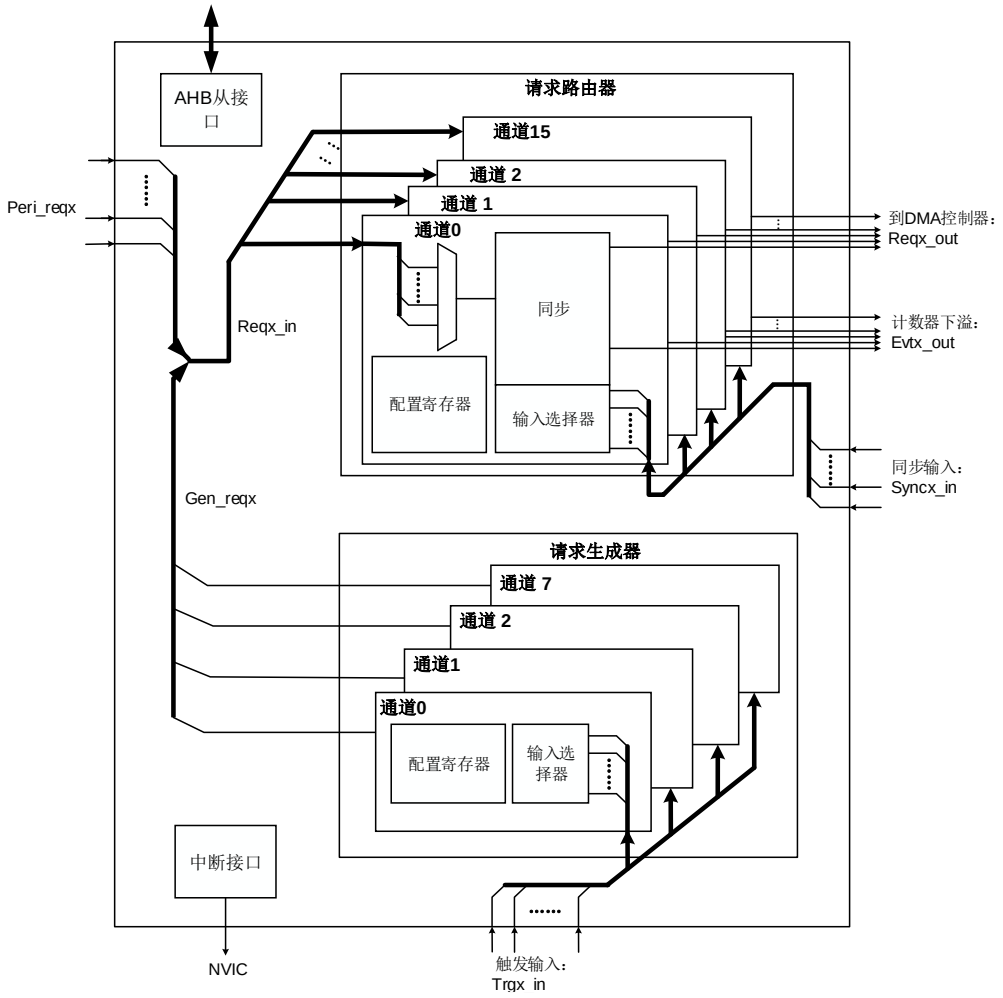
DMAMUX 是 DMA 请求的传输调度器。可编程的 DMA 请求多路复用器 DMAMUX，可在外设和 DMA 控制器之间路由 DMA 请求线路，或者 DMAMUX 也可以将可编程事件连入到输入触发信号上，作为一个 DMAMUX 请求发生器，再由 DMAMUX 请求路由器在 DMAMUX 请求生成器产生的 DMA 请求和 DMA 控制器之间路由 DMA 请求线路。每个 DMAMUX 请求路由通道选择一条唯一的 DMA 请求线路，无条件地或同步地从它的 DMAMUX 同步输入事件。DMA 请求信号会一直挂起，直到 DMA 控制器响应它，并且产生一个 DMA 确认信号，此时相应的 DMA 请求信号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

18.2. 主要特征

- 16 个可配置的 DMAMUX 请求路由输出通道；
- 8 个 DMAMUX 请求生成通道；
- 35 路触发输入信号到 DMAMUX 请求生成器；
- 29 路同步输入信号；
- 每个 DMAMUX 请求生成通道包含一个 DMAMUX 请求触发输入选择器，一个 DMAMUX 请求生成计数器，和一个指示被选中的 DMAMUX 请求触发输入信号的事件溢出标志；
- 每个 DMAMUX 请求路由输出通道包含 189 路外设 DMAMUX 请求输入信号，一个同步输入信号选择器，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，一个 DMAMUX 请求路由计数器，和一个指示被选中的同步输入信号的事件溢出标志。

18.3. 结构框图

图 18-1. DMAMUX 结构框图



18.4. 信号描述

DMAMUX 信号描述如下所示：

- 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。

18.5. 功能说明

如*图 18-1. 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。

18.5.1. DMAMUX 请求路由器

DMAMUX 请求路由器可在外设/ 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 18-2. DMAMUX 请求路由输入信号映射*。一个 DMAMUX 请求路由通道只能与一个 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。

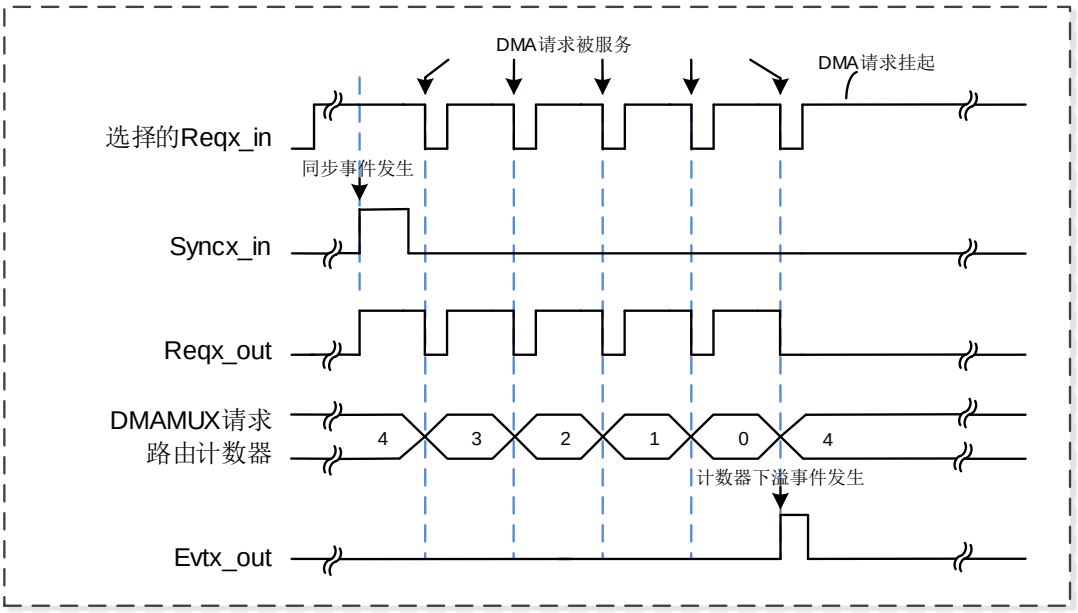
当同步模式使能时

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 18-2. 同步模式*为当 NBR[4:0]=4，SYNCEN=1，EVGEN=1，SYNCP[1:0]=0b01 时的举例。

图 18-2. 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 18-4. 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。

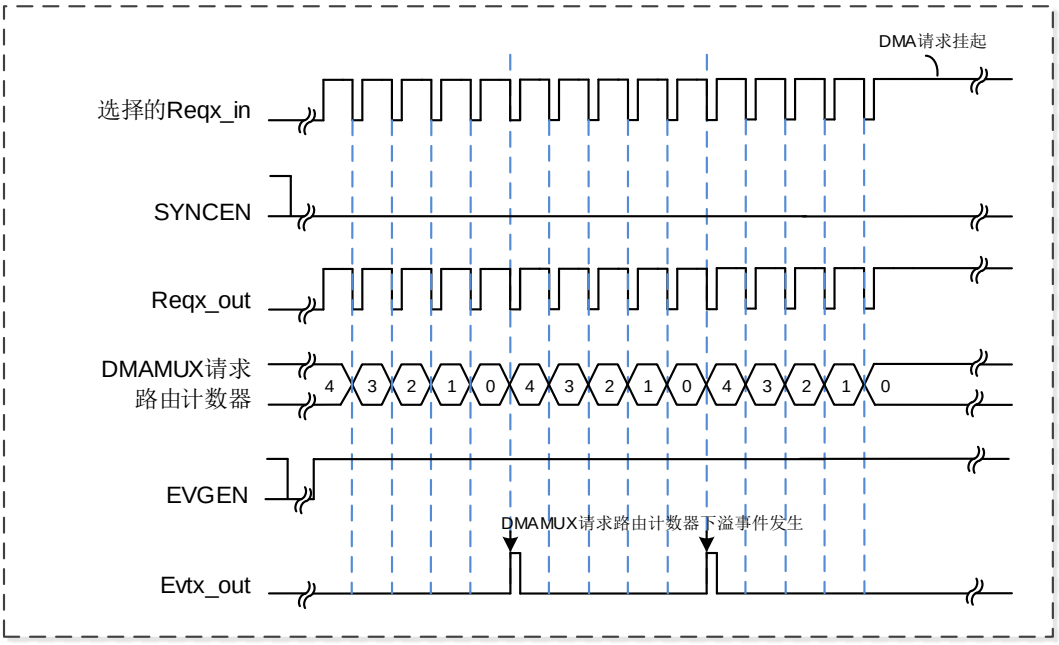
注意：如果同步输入事件发生时，DMAMUX 输入上没有挂起的 DMA 请求，则这个同步输入事件将被忽略，之后如有 DMA 请求被挂起，它将不会被连接到 DMAMUX 请求路由通道 x 的输出，直到发生下一个同步输入事件。

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 18-3. 通道事件输出*为当 NBR[4:0]=4，SYNCEN=0，EVGEN=1 时的举例。

图 18-3. 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。

18.5.2. 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[5:0]位域来配置，参考*表 18-3. 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。

18.5.3. 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。

18.5.4. 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18-1. 中断事件*描述了其对应关系。

表 18-1. 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	DMAMUX_RM_INTF 寄存器的 SOIFx 位	DMAMUX_RM_INTC 寄存器的 SOIFCx 位	DMAMUX_RM_CHxCFG 寄存器的 SOIE 位
DMAMUX 请求生成通道 y 上的触发溢出事件	DMAMUX_RG_INTF 寄存器的 TOIFy 位	DMAMUX_RG_INTC 寄存器的 TOIFCy 位	DMAMUX_RG_CHxCFG 寄存器的 TOIE 位

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。

18.5.5. DMAMUX 映射

DMAMUX 与 DMA0 和 DMA1 配合使用。DMAMUX 的通道 0 到 7 与 DMA0 的通道 0 到 7 相连，DMAMUX 的通道 8 到 15 与 DMA1 的通道 0 到 7 相连。

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 18-2. 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 18-2. 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_req0
2	Gen_req1
3	Gen_req2
4	Gen_req3
5	Gen_req4
6	Gen_req5
7	Gen_req6
8	Gen_req7
9	ADC0
10	ADC1
11	TIMER0_CH0
12	TIMER0_CH1
13	TIMER0_CH2
14	TIMER0_CH3
15	TIMER0_MCH0
16	TIMER0_MCH1
17	TIMER0_MCH2
18	TIMER0_MCH3
19	TIMER0_UP
20	TIMER0_TRG
21	TIMER0_CMT
22	TIMER1_CH0
23	TIMER1_CH1
24	TIMER1_CH2
25	TIMER1_CH3

请求路由通道输入标识MUXID[7:0]	来源
26	TIMER1_UP
27	TIMER1_TRG
28	保留
29	TIMER2_CH0
30	TIMER2_CH1
31	TIMER2_CH2
32	TIMER2_CH3
33	TIMER2_UP
34	保留
35	TIMER2_TRG
36	TIMER3_CH0
37	TIMER3_CH1
38	TIMER3_CH2
39	TIMER3_CH3
40	保留
41	TIMER3_TRG
42	TIMER3_UP
43	I2C0_RX
44	I2C0_TX
45	I2C1_RX
46	I2C1_TX
47	SPI0_RX
48	SPI0_TX
49	SPI1_RX
50	SPI1_TX
51	USART0_RX
52	USART0_TX
53	USART1_RX
54	USART1_TX
55	USART2_RX
56	USART2_TX
57	TIMER7_CH0
58	TIMER7_CH1
59	TIMER7_CH2
60	TIMER7_CH3
61	TIMER7_MCH0
62	TIMER7_MCH1
63	TIMER7_MCH2
64	TIMER7_MCH3
65	TIMER7_UP
66	TIMER7_TRG
67	TIMER7_CMT

请求路由通道输入标识MUXID[7:0]	来源
68	TIMER4_CH0
69	TIMER4_CH1
70	TIMER4_CH2
71	TIMER4_CH3
72	TIMER4_UP
73	保留
74	TIMER4_TRG
75	SPI2_RX
76	SPI2_TX
77	UART3_RX
78	UART3_TX
79	UART4_RX
80	UART4_TX
81	DAC_CH0
82	DAC_CH1
83	TIMER5_UP
84	TIMER6_UP
85	USART5_RX
86	USART5_TX
87	I2C2_RX
88	I2C2_TX
89	DCI
90	CAU_IN
91	CAU_OUT
92	HAU_IN
93	UART6_RX
94	UART6_TX
95	UART7_RX
96	UART7_TX
97	SPI3_RX
98	SPI3_TX
99	SPI4_RX
100	SPI4_TX
101	SAI0_B0
102	SAI0_B1
103	RSPDIF_DATA
104	RSPDIF_CS
105	HPDF_FLT0
106	HPDF_FLT1
107	HPDF_FLT2
108	HPDF_FLT3
109	TIMER14_CH0

请求路由通道输入标识MUXID[7:0]	来源
110	TIMER14_CH1
111	TIMER14_MCH0
112	TIMER14_UP
113	TIMER14_TRG
114	TIMER14_CMT
115	TIMER15_CH0
116	TIMER15_MCH0
117	保留
118	TIMER15_UP
119	TIMER16_CH0
120	TIMER16_MCH0
121	保留
122	TIMER16_UP
123	ADC2
124	FAC_READ
125	FAC_WRITE
126	TMU_READ
127	TMU_WRITE
128	TIMER22_CH0
129	TIMER22_CH1
130	TIMER22_CH2
131	TIMER22_CH3
132	TIMER22_UP
133	保留
134	TIMER22_TRG
135	TIMER23_CH0
136	TIMER23_CH1
137	TIMER23_CH2
138	TIMER23_CH3
139	TIMER23_UP
140	保留
141	TIMER23_TRG
142	TIMER30_CH0
143	TIMER30_CH1
144	TIMER30_CH2
145	TIMER30_CH3
146	TIMER30_UP
147	保留
148	TIMER30_TRG
149	TIMER31_CH0
150	TIMER31_CH1
151	TIMER31_CH2

请求路由通道输入标识MUXID[7:0]	来源
152	TIMER31_CH3
153	保留
154	TIMER31_UP
155	TIMER31_TRG
156	TIMER40_CH0
157	TIMER40_MCH0
158	TIMER40_CMT
159	TIMER40_UP
160	TIMER41_CH0
161	TIMER41_MCH0
162	TIMER41_CMT
163	TIMER41_UP
164	TIMER42_CH0
165	TIMER42_MCH0
166	TIMER42_CMT
167	TIMER42_UP
168	TIMER43_CH0
169	TIMER43_MCH0
170	TIMER43_CMT
171	TIMER43_UP
172	TIMER44_CH0
173	TIMER44_MCH0
174	TIMER44_CMT
175	TIMER44_UP
176	TIMER50_UP
177	TIMER51_UP
178	SAI1_B0
179	SAI1_B1
180	SAI2_B0
181	SAI2_B1
182	SPI5_RX
183	SPI5_TX
184	I2C3_RX
185	I2C3_TX
186	CAN0
187	CAN1
188	CAN2
189	TIMER40_CH1
190	TIMER40_TRG
191	TIMER41_CH1
192	TIMER41_TRG
193	TIMER42_CH1

请求路由通道输入标识MUXID[7:0]	来源
194	TIMER42_TRG
195	TIMER43_CH1
196	TIMER43_TRG
197	TIMER44_CH1
198	TIMER44_TRG

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[5:0]位域来配置，参考*表 18-3. 触发输入信号映射*。

表 18-3. 触发输入信号映射

触发输入标识TID[5:0]	来源
0	Evt0_out
1	Evt1_out
2	Evt2_out
3	Evt3_out
4	Evt4_out
5	Evt5_out
6	Evt6_out
7	EXTI_0
8	EXTI_1
9	EXTI_2
10	EXTI_3
11	EXTI_4
12	EXTI_5
13	EXTI_6
14	EXTI_7
15	EXTI_8
16	EXTI_9
17	EXTI_10
18	EXTI_11
19	EXTI_12
20	EXTI_13
21	EXTI_14
22	EXTI_15
23	RTC_WAKEUP
24	CMP0_OUTPUT
25	CMP1_OUTPUT
26	I2C0_WAKEUP
27	I2C1_WAKEUP
28	I2C2_WAKEUP
29	I2C3_WAKEUP

触发输入标识TID[5:0]	来源
30	I2C0_INT_EVENT
31	I2C1_INT_EVENT
32	I2C2_INT_EVENT
33	I2C3_INT_EVENT
34	ADC2_INT

注意：EXTI_x(x=0…15)仅 EXTI 中断事件发生时会产生 DMA 请求。

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 18-4. 同步输入信号映射*。

表 18-4. 同步输入信号映射

同步输入标识SYNCID[4:0]	来源
0	Evt0_out
1	Evt1_out
2	Evt2_out
3	Evt3_out
4	Evt4_out
5	Evt5_out
6	Evt6_out
7	EXTI_0
8	EXTI_1
9	EXTI_2
10	EXTI_3
11	EXTI_4
12	EXTI_5
13	EXTI_6
14	EXTI_7
15	EXTI_8
16	EXTI_9
17	EXTI_10
18	EXTI_11
19	EXTI_12
20	EXTI_13
21	EXTI_14
22	EXTI_15
23	RTC_WAKEUP
24	CMP0_OUTPUT
25	I2C0_WAKEUP
26	I2C1_WAKEUP
27	I2C2_WAKEUP

同步输入标识SYNCID[4:0]	来源
28	I2C3_WAKEUP

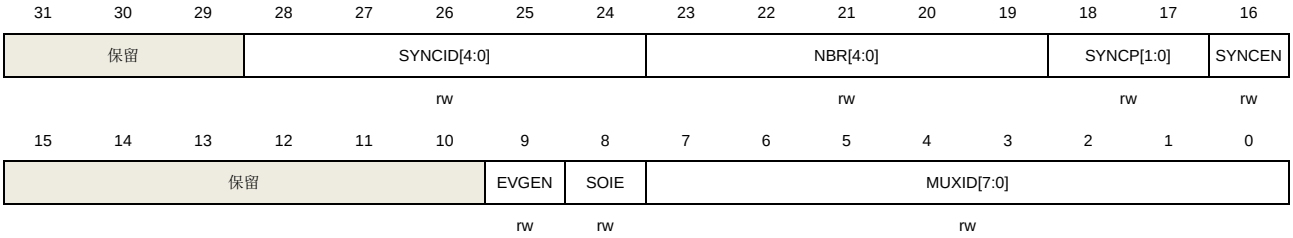
18.6. DMAMUX 寄存器

DMAMUX 基地址：0x4002 0800

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

x = 0...15，其中 x 为通道序号
地址偏移：0x00 + 0x04 * 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。
28:24	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。

18.6.2. 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
SOIF15	SOIF14	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	r	r

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	SOIFx	请求路由通道x的同步溢出事件标志位 如果同步输入事件发生时，DMAMUX请求路由计数器值小于NBR[4:0]，则该位置位。 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。

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

地址偏移：0x84
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOIFC15	SOIFC14	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	w	w

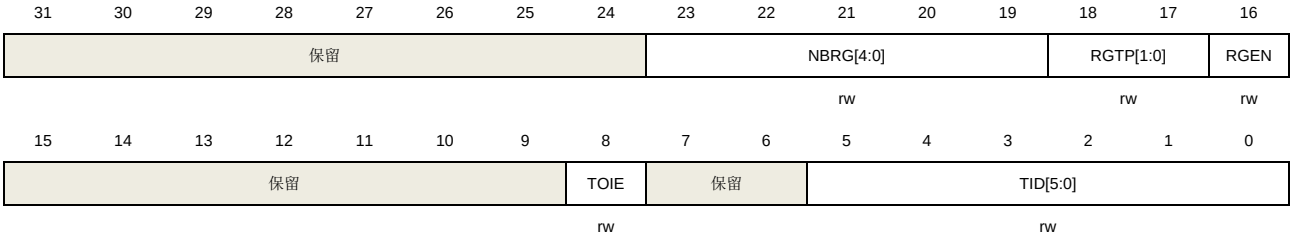
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	SOIFCx	请求路由通道x的同步溢出事件标志清除位 写1可清除相应通道在DMAMUX_RM_INTF寄存器的同步溢出标志SOIFx。

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

x = 0...7，其中 x 为通道序号

地址偏移：0x100 + 0x04 * 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	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
7:6	保留	必须保持复位值。
5:0	TID[5:0]	触发输入标识 选择DMAMUX请求生成通道的触发输入源。

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

地址偏移：0x140
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								TOIF7	TOIF6	TOIF5	TOIF4	TOIF3	TOIF2	TOIF1	TOIF0
								r	r	r	r	r	r	r	r

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	TOIFx	DMAMUX请求生成通道x的触发溢出标志位 如果触发输入事件在DMAMUX请求生成计数器下溢之前发生，则该位置位。 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。

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

地址偏移：0x144
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								TOIFC7	TOIFC6	TOIFC5	TOIFC4	TOIFC3	TOIFC2	TOIFC1	TOIFC0
								w	w	w	w	w	w	w	w

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	TOIFCx	DMAMUX请求生成通道x的触发溢出标志清除位 写1可清除相应通道在DMAMUX_RG_INTF寄存器的触发溢出标志TOIFx。

19. 调试（DBG）

19.1. 简介

GD32H73x_75x 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®-M7 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®-M7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 或者 CAN。

19.2. 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。

19.2.1. 切换 JTAG / SW 接口

默认使用 SWD 调试接口，通过 EFUSE_USER_CTL 寄存器的 JTAGNSW 位实现 JTAG 和 SWD 调试接口的切换。

19.2.2. 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 19-1. 引脚分配

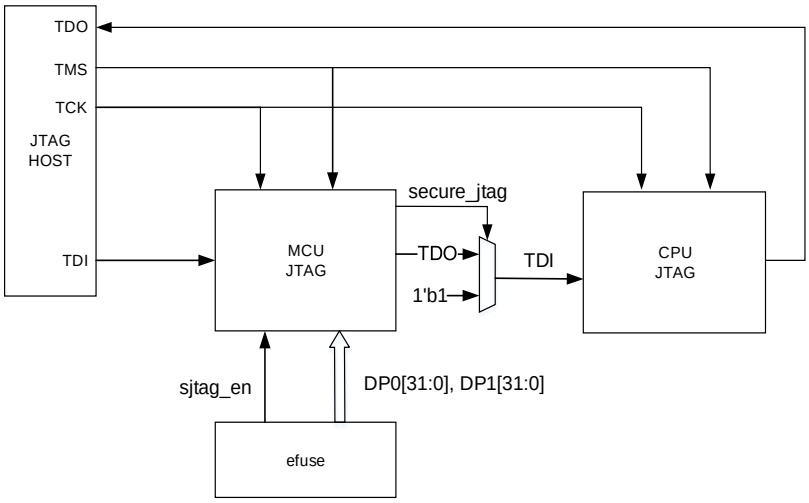
引脚	调试接口
PA15	JTDI
PA14	JTCK/SWCLK
PA13	JTMS/SWDIO
PB4	NJTRST
PB3	JTDO

默认复位后使用五个引脚的 JTAG 调试，用户可以在不使用 NJTRST 引脚情况下正常使用 JTAG 功能，此时 PB4 可以用作普通 GPIO 功能（NJTRST 硬件拉高）。如果切换到 SW 调试

模式，PA15/PB4/PB3 释放作为普通 GPIO 功能。如果 JTAG 和 SW 调试功能都没有使用，这五个引脚都释放作为普通 GPIO 功能。

19.2.3. JTAG

图 19-1. JTAG 模块框图



JTAG 链状结构

Cortex®-M7 内核的 JTAG TAP（CPU JTAG）和边界扫描（BSD）TAP（MCU JTAG）串行连接。边界扫描（BSD）JTAG 的 IR（指令寄存器）是 5 位，而 Cortex®-M7 内核的 JTAG 的 IR（指令寄存器）是 4 位。所以当 JTAG 进行 IR 移位输入时，首先移位 5 位 BYPASS 指令给 BSD JTAG，然后移位 4 位标准指令给 Cortex®-M7 JTAG。当进行数据移位时，数据链只需要额外添加一位，因为 BSD JTAG 已处在 BYPASS 模式。

BSD JTAG ID 代码是 0x000717A3。

安全 JTAG

1. 安全 JTAG 只支持 JTAG，不支持 SW

2. EFUSE 配置

EFUSE 相关位：JTAGNSW, NDBG[1:0], DPx[31:0]（x=0,1）

模式	寄存器配置
No debug	NDBG[1:0] = 2b’10 or 2b’11 JTAGNSW: 不关心 DP0[31:0], DP1[31:0]: 不关心
SW	NDBG[1:0] = 2b’00 or 2b’01 JTAGNSW = 1b’0 DP0[31:0], DP1[31:0]: 不关心

普通JTAG	NDBG[1:0] = 2b'00 JTAGNSW = 1b'1 DP0[31:0], DP1[31:0]: 不关心
安全JTAG	NDBG[1:0] = 2b'01 JTAGNSW = 1b'1 DP0[31:0], DP1[31:0]: 熔丝中调试秘钥字段值

3. 安全 JTAG 的使用

a) 配置 EFUSE 为安全 JTAG：首先配置 JTAG 安全密码 DPx[31:0]（x=0,1），再配置 JTAGNSW=1b'1, NDBG[1:0]= 2b'01。

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 DP1[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。

3. 输入正确密码打开 debug，只限于 SPC_L 及以下，不会打开 ROM、内存安全模式和 SPC_H。

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 DP0[31:0]，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 DP1[31:0]，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，"1"：解密过程发生错误，"0"：解密过程未发生错误。

19.2.4. 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®-M7 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。

19.2.5. JEDEC-106 ID code

Cortex®-M7 集成了 JEDEC-106 ID 代码。位于 ROM 表中，映射地址为 0xE00FD000_0xE00FDFFF。

19.3. 调试保持功能描述

19.3.1. 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保持不变，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保持不变，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 IRC64M 或 LPIRC4M。

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。

19.3.2. TIMER, I2C, RTC, WWDGT, FWDGT 和 CAN 外设调试支持

当内核停止，并且 DBG 控制寄存器 x（DBG_CTLx，x=1，2，3，4）中的相应位置 1。对于不同外设，有不同动作：

对于 TIMER 外设，TIMER 计数器停止并进行调试；

对于 I2C 外设， SMBUS 保持状态并进行调试；

对于 WWDGT 或者 FWDGT 外设， 计数器时钟停止并进行调试；

对于 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。

19.4. DBG 寄存器

DBG 基地址：0xE00E1000

19.4.1. ID 寄存器（DBG_ID）

地址偏移：0x00
只读寄存器

该寄存器只能按字（32 位）访问

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

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

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

19.4.2. 控制寄存器 0（DBG_CTL0）

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32 位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留											TRACECLKEN	TRACE_MODE[1:0]		保留	
											rw	rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													STB_HOLD	DSLP_HOLD	SLP_HOLD
													rw	rw	rw

位/位域	名称	描述
31:21	保留	必须保持复位值。
20	TRACECLKEN	跟踪时钟使能 0: 跟踪时钟失能 1: 跟踪时钟使能。
19:18	TRACE_MODE[1:0]	跟踪引脚分配模式 该位由软件置位和复位 00: 跟踪引脚用于异步模式

		01: 跟踪引脚用于同步模式且数据长度为1 10: 跟踪引脚用于同步模式且数据长度为2 11: 跟踪引脚用于同步模式且数据长度为 4。
17:3	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。

19.4.3. 控制寄存器 1（DBG_CTL1）

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
31:7	保留	必须保持复位值。
6	WWDGT_HOLD	WWDGT 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 WWDGT 计数器时钟，用于调试。
5:0	保留	必须保持复位值。

19.4.4. 控制寄存器 2（DBG_CTL2）

地址偏移：0x3C

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							I2C3_HOLD	I2C2_HOLD	I2C1_HOLD	I2C0_HOLD	保留				
							rw	rw	rw	rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				TIMER51_HOLD	TIMER50_HOLD	TIMER31_HOLD	TIMER30_HOLD	TIMER23_HOLD	TIMER22_HOLD	TIMER6_HOLD	TIMER5_HOLD	TIMER4_HOLD	TIMER3_HOLD	TIMER2_HOLD	TIMER1_HOLD
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:25	保留	必须保持复位值。
24	I2C3_HOLD	I2C3 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 I2C3 的 SMBUS 状态不变，用于调试。
23	I2C2_HOLD	I2C2 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 I2C2 的 SMBUS 状态不变，用于调试。
22	I2C1_HOLD	I2C1 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 I2C1 的 SMBUS 状态不变，用于调试。
21	I2C0_HOLD	I2C0 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 I2C0 的 SMBUS 状态不变，用于调试。
20:12	保留	必须保持复位值。
11	TIMER51_HOLD	TIMER51 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 51 计数器不变，用于调试。
10	TIMER50_HOLD	TIMER50 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 50 计数器不变，用于调试。

9	TIMER31_HOLD	TIMER31 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 31 计数器不变，用于调试。
8	TIMER30_HOLD	TIMER30 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 30 计数器不变，用于调试。
7	TIMER23_HOLD	TIMER 23 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 23 计数器不变，用于调试。
6	TIMER22_HOLD	TIMER 22 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 22 计数器不变，用于调试。
5	TIMER6_HOLD	TIMER 6 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	TIMER 5 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	TIMER 4 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	TIMER 3 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	TIMER 2 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	TIMER 1 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。

19.4.5. 控制寄存器 3（DBG_CTL3）

地址偏移：0x4C

复位值： 0x0000 0000

该寄存器只能按字（32 位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								TIMER44_HOLD	TIMER43_HOLD	TIMER42_HOLD	TIMER41_HOLD	TIMER40_HOLD	TIMER16_HOLD	TIMER15_HOLD	TIMER14_HOLD
								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											CAN2_HOLD	CAN1_HOLD	CAN0_HOLD	TIMER7_HOLD	TIMER0_HOLD
											rw	rw	rw	rw	rw

位/位域	名称	描述
31:24	保留	必须保持复位值。
23	TIMER44_HOLD	TIMER44 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 44 计数器不变，用于调试。
22	TIMER43_HOLD	TIMER43 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 43 计数器不变，用于调试。
21	TIMER42_HOLD	TIMER42 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 42 计数器不变，用于调试。
20	TIMER41_HOLD	TIMER41 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 41 计数器不变，用于调试。
19	TIMER40_HOLD	TIMER40 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 40 计数器不变，用于调试。
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。
15:5	保留	必须保持复位值。
4	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。
3	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。
2	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。
1	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。
0	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。

19.4.6. 控制寄存器 4（DBG_CTL4）

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													FWDGT_HOLD	保留	RTC_HOLD
													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
31:19	保留	必须保持复位值。
18	FWDGT_HOLD	FWDGT 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 FWDGT 计数器时钟，用于调试。
17	保留	必须保持复位值。
16	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。
15:0	保留	必须保持复位值。

20. 模数转换器（ADC）

20.1. 简介

MCU 片上集成了 12/14 位逐次逼近式模数转换器模块（ADC），ADC0 有 20 个外部通道，1 个内部通道（DAC0_OUT0 通道），ADC1 有 18 个外部通道，3 个内部通道（电池电压（V_{BAT}）通道、参考电压输入通道（V_{REFINT}）和 DAC0_OUT1 通道），ADC2 有 17 个外部通道，4 个内部通道（电池电压（V_{BAT}）通道、参考电压输入通道（V_{REFINT}）、内部温度传感通道（V_{SENSE}）和高精度温度传感器通道（V_{SENSE2}））。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（ADC0/1 为 32 位数据寄存器，ADC2 为 16 位数据寄存器）。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。

20.2. 主要特征

- 高性能：
  - ADC采样分辨率：ADC0/1可配置14位、12位、10位或者8位分辨率，ADC2可配置12位、10位、8位或者6位分辨率；
  - ADC0/1采样率：14位分辨率为4 MSPs，12位分辨率为4.5 MSPs，10位分辨率为5.14 MSPs，8位分辨率为6 MSPs。分辨率越低，转换越快；
  - ADC2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：ADC0/1需要1082个ADC时钟周期，ADC2需要46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20个外部模拟输入通道，ADC1有18个外部模拟输入通道，ADC2有17个外部模拟输入通道；
  - 内部温度传感通道（V_{SENSE}）；
  - 内部参考电压输入通道（V_{REFINT}）；
  - 外部监测电池V_{BAT}供电引脚输入通道；
  - 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：
  - ADC0/1为32位的数据寄存器，ADC2为16位数据寄存器；
  - ADC0/1可调整的过采样率，从2x到1024x，ADC2可调整的过采样率，从2x到256x；
  - ADC0/1高达11位的可编程数据移位，ADC2为8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。

20.3. 引脚和内部信号

*图 20-1. ADC 模块框图*给出了 ADC 框图。*表 20-1. ADC 内部输入信号*给出了 ADC 内部信号。*表 20-2. ADC 输入引脚定义*给出了 ADC 引脚说明。

表 20-1. 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

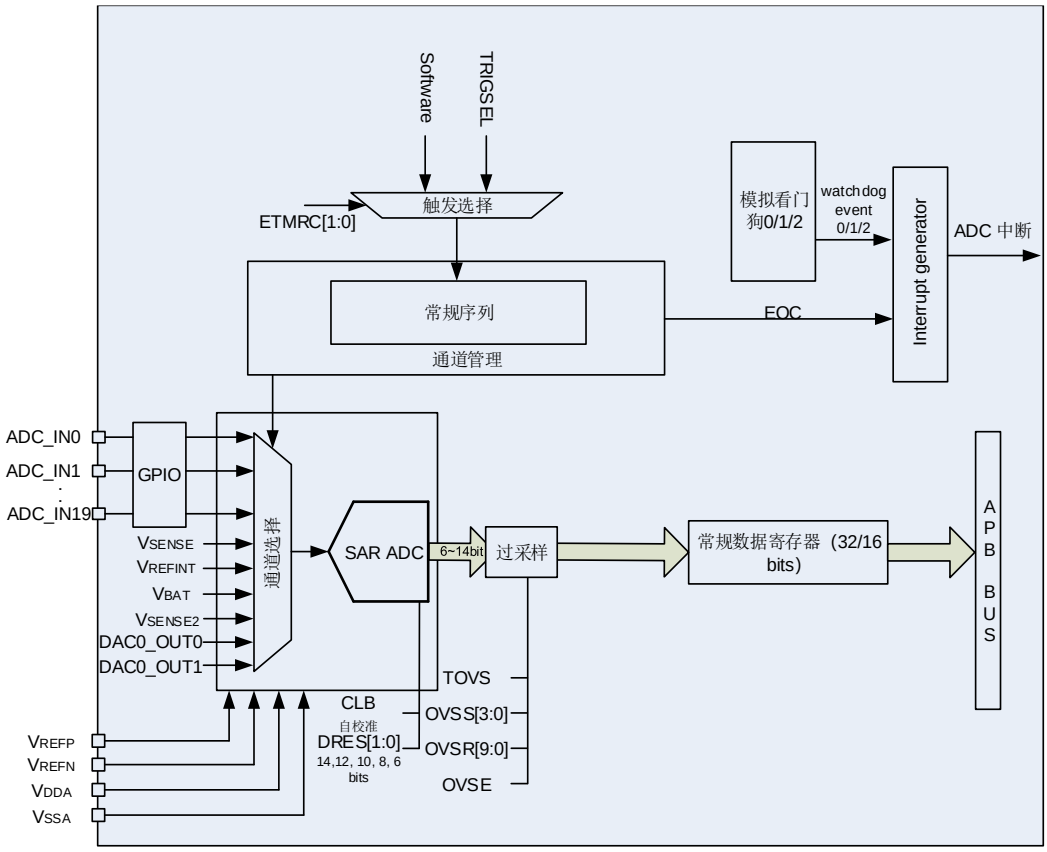
表 20-2. ADC 输入引脚定义

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[19:0]	多达20路外部通道

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

20.4. 功能描述

图 20-1. ADC 模块框图



20.4.1. 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。

校准模式分为失调+失配和失调两种（只针对 ADC0/1），可通过设置 ADC_CTL1 寄存器的 CALMOD 位进行修改，推荐使用失调模式。

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。

20.4.2. 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，ADC0 和 ADC1 最大频率为 72 MHz，ADC2 最大频率为 80 MHz。

CK_ADC 也可以由 CK_PLL1P、CK_PLL2R 或 CK_PER 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 时钟共享 ADC0 时钟，当使用 ADC1 时，必须打开 ADC0 时钟，且只能通过 ADC0 进行时钟分频。

20.4.3. 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。

20.4.4. 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。此时，通道m不能用于单端模式和差分模式，且不能配置转换功能。差分通道引脚分配如*表20-3. ADC差分通道引脚匹配*。

表 20-3. ADC 差分通道引脚匹配

差分通道n编号	ADC0		ADC1		ADC2	
	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_C	PA1_C	PA0_C	PA1_C	PC2_C	PC3_C
1	PA1_C	PA0_C	PA1_C	PA0_C	PC3_C	PC2_C
2	PF11	PF12	PF13	PF14	PF9	PF10
3	PA6	PA7	PA6	PA7	PF7	PF8
4	PC4	PC5	PC4	PC5	PF5	PF6
5	PB1	PB0	PB1	PB0	PF3	PF4
6	PF12	PF11	PF14	PF13	PF10	PF9
7	PA7	PA6	PA7	PA6	PF8	PF7
8	PC5	PC4	PC5	PC4	PF6	PF5
9	PB0	PB1	PB0	PB1	PF4	PF3
10	PC0	PC1	PC0	PC1	PC0	PC1
11	PC1	PC2	PC1	PC2	PC1	PC2
12	PC2	PC3	PC2	PC3	PC2	PC1

13	PC3	PC2	PC3	PC2	PH2	PH3
14	PA2	PA3	PA2	PA3	PH3	PH4
15	PA3	PA2	PA3	PA2	PH4	PH5
16	PA0	PA1	null	null	PH5	PH4
17	PA1	PA0	null	null	null	null
18	PA4	PA5	PA4	PA5	null	null
19	PA5	PA4	PA5	PA4	null	null
20	null	null	null	null	null	null
21	null	null	null	null	null	null

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：

$$V_{INn}-V_{INm} = V_{REFP}*(2*D_{out}/4095-1) \quad (20\text{-}1)$$

20.4.5. 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 21 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。

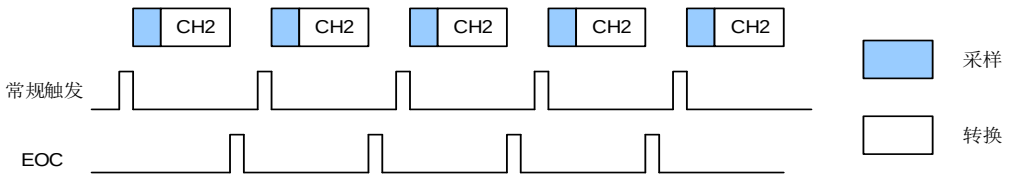
注意：尽管 ADC 支持 21 个通道，但常规序列一次最多转换 16 个通道。

20.4.6. 运行模式

单次运行模式

单次运行模式下，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 20-2. 单次运行模式



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

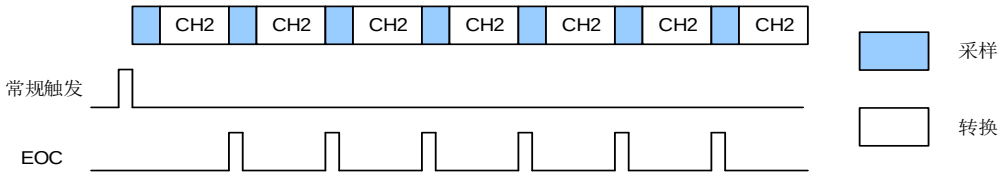
常规序列单次运行模式的软件流程：

1. 确保ADC_CTL0寄存器的DISRC和SM位以及ADC_CTL1寄存器的CTN位为0；
2. 用模拟通道编号来配置RSQ0；
3. 配置ADC_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 20-3. 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。

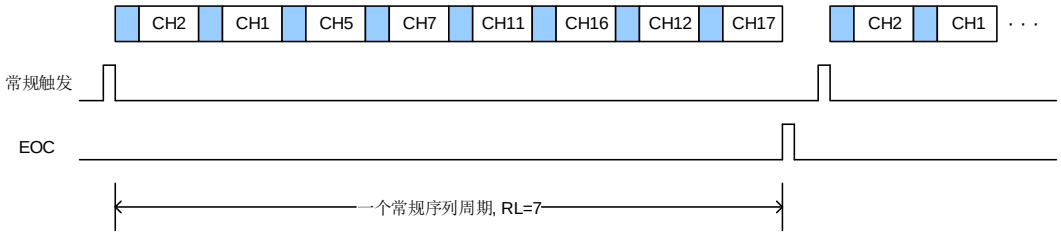
扫描运行模式

扫描运行模式可以通过将 ADC_CTL0 寄存器的 SM 位置 1 来使能。在此模式下，ADC 扫描转换所有被 ADC_RSQ0~ADC_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 位被置 1，将产生中断。当常规序列工作在扫描模式下时，ADC_CTL1 寄存器的 DMA 位必须设置为 1。

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

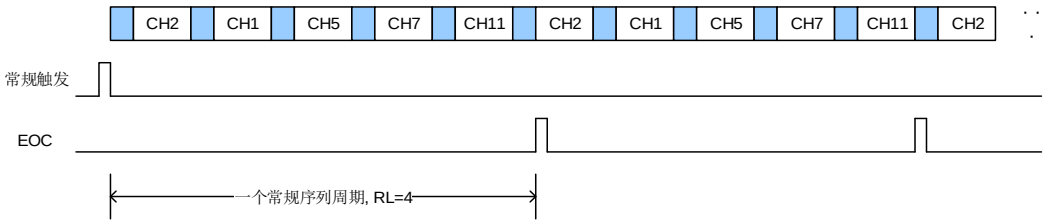
图 20-4. 扫描运行模式，且连续运行模式失能



常规序列扫描运行模式的软件流程：

1. 设置 ADC_CTL0 寄存器的 SM 位和 ADC_CTL1 寄存器的 DMA 位为 1；
2. 配置 ADC_RSQx 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（参考 DMA 模块）；
5. 设置 SWRCST 位，或者给常规序列产生一个 TRIGSEL 触发；

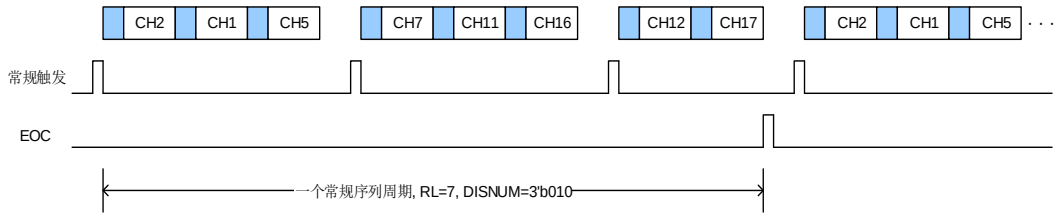
图 20-5. 扫描运行模式，连续运行模式使能



间断运行模式

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 20-6. 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；

20.4.7. 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：对于 ADC0/1，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。

20.4.8. 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 20-7. 14 位数据存储模式

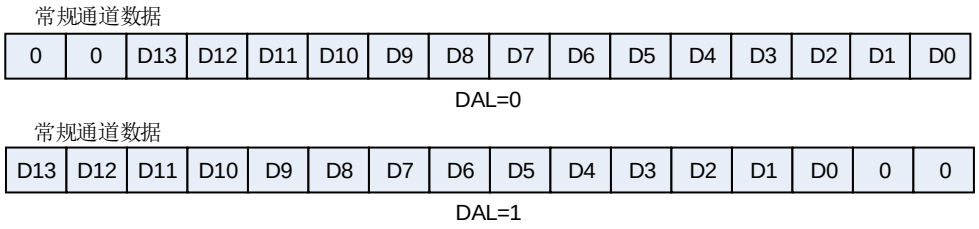
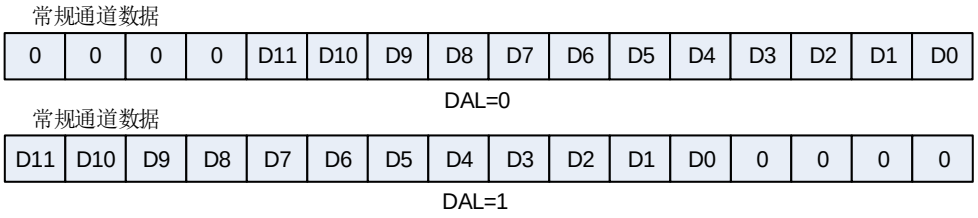
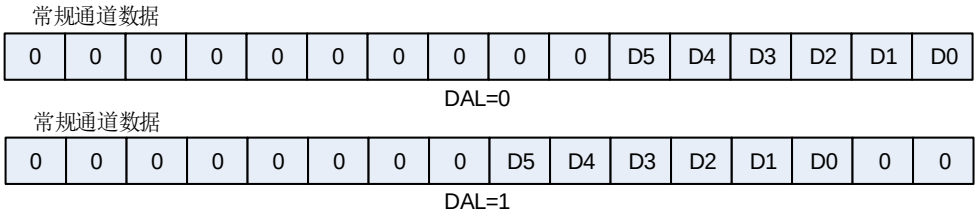


图 20-8. 12 位数据存储模式



6 位分辨率的数据存储模式不同于 14 位/12 位/10 位/8 位分辨率数据存储模式，如*图 20-9. 6 位数据存储模式*。

图 20-9. 6 位数据存储模式



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。

20.4.9. 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。例如，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 3.5 个周期，那么总的转换时间为：“3.5+12.5”个 CK_ADC 周期，即 0.4us。

20.4.10. 外部触发配置

常规通道的转换可通过 TRIGSEL 的上升沿或软件触发。触发源由 ADC_CTL1 寄存器中的 ETMRC[1:0]位控制。

表 20-4. ADC0/ADC1/ADC2 常规通道的触发源

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

20.4.11. 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。

20.4.12. 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。

20.4.13. 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（ADC2_CH18），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_CH20）。将 ADC_CTL1 寄存器的 INREFEN 位置 1 可以使能内部电压参考通道（ADC1_CH17/ADC2_CH19）。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 ADC1_CH17/ADC2_CH19 输入通道。

使用温度传感器：

1. 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；
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：

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

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_CH20）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：

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

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。

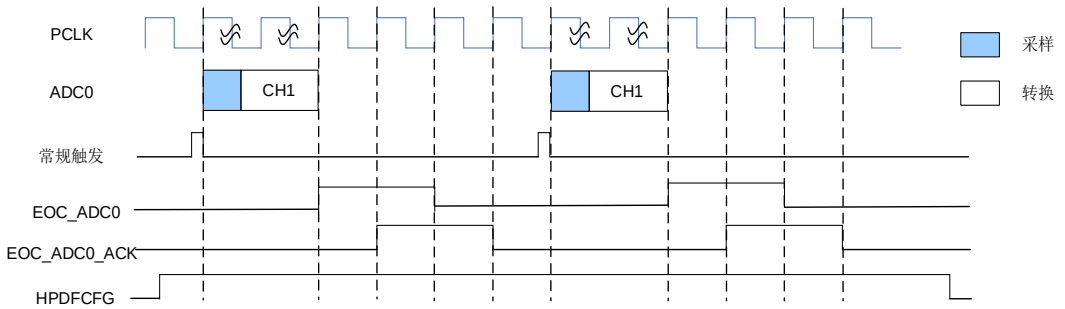
20.4.14. 电池电压检测电路

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（ADC1_IN16/ADC2_IN17）被启用，集成在 V_{BAT} 引脚上的 4 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}/4 连接到 16/17 输入通道中的 ADC1_IN16/ ADC2_IN17。因此，转换后的数字值为 V_{BAT}/4。为了防止不必要的电池能耗，建议仅在需要时启用桥接器。

20.4.15. 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 位必须清除为 0。ADC 将常规序列数据寄存器数据的 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 20-10. HFDF 与 ADC 模块握手信号示意图*所示。

图 20-10. HFDF 与 ADC 模块握手信号示意图



20.4.16. 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、12 及 14 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。较低的分辨率能够减少转换时间。如图*表 20-5. ADC0 和 ADC1 不同分辨率对应的 tCONV 时间*和*表 20-6. ADC2 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 20-5. ADC0 和 ADC1 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}=72MHz	t_{SMPL}(min) (ADC clock cycles)	t_{ADC} (ADC clock cycles)	t_{ADC}(ns) at f_{ADC}=72MHz
14	14.5	201.39 ns	3.5	18	250 ns
12	12.5	173.61 ns	3.5.	16	222.22 ns
10	10.5	145.83 ns	3.5	14	194.44 ns
8	8.5	118.06 ns	3.5	12	166.67 ns

表 20-6. ADC2 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}=80MHz	t_{SMPL}(min) (ADC clock cycles)	t_{ADC} (ADC clock cycles)	t_{ADC}(ns) at f_{ADC}=80MHz
12	12.5	156.25 ns	2.5	15	187.5 ns
10	10.5	121.25 ns	2.5	13	162.5 ns
8	8.5	106.25ns	2.5	11	137.5ns
6	6.5	81.25 ns	2.5	9	112.5ns

20.4.17. 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，在 ADC0 和 ADC1 中最高可达 32 位，在 ADC2 中高达 16 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：

$$Result = \frac{1}{M} * \sum_{n=0}^{N-1} D_{out}(n) \quad (20\text{-}2)$$

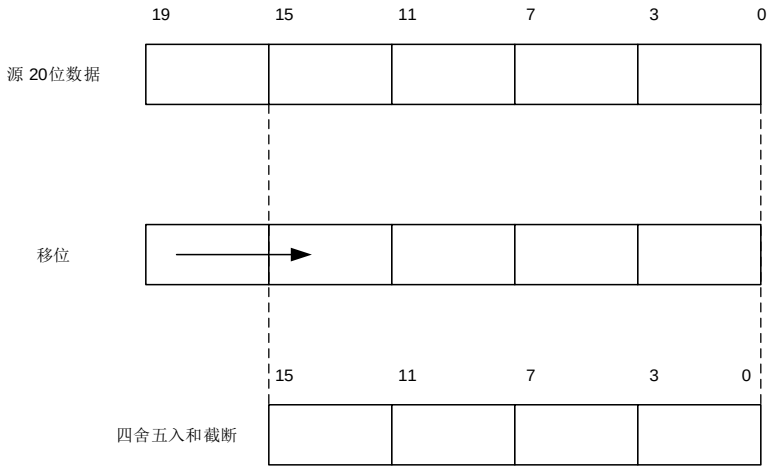
对于 14 位 ADC，片上硬件过采样单元执行两个功能：求和和位右移。过采样率 N 是在 ADC_OVSAMPCTL 寄存器的 OVSR[9:0]位定义，它的取值范围为 2x 到 1024x。除法系数 M 定义一个多达 11 位的右移，它通过 ADC_OVSAMPCTL 寄存器 OVSS[3:0]位进行配置。

对于 14 位 ADC，求和单元能够生成一个多达 24 位（1024 x 14 位）的值，该结果首先右移。然后将数据存储到寄存器中

对于 12 位 ADC，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[7:0]位定义，它的取值范围为 2x 到 256x。除法系数 M 定义一个多达 8 位的右移，它通过 ADC_OVSAMPCTL 寄存器 OVSS[3:0]位进行配置。

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

图 20-11. 12 位 ADC 20 位到 16 位的结果截断



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

*图 20-11. 12 位 ADC 20 位到 16 位的结果截断*描述一个从原始 20 位的累积数值处理成 16 位结果值的例子。

图 20-12. 12 位 ADC 右移 5 位和取整的数例

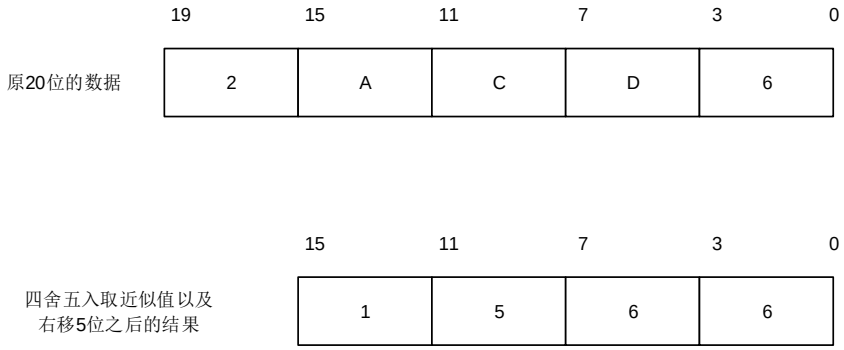


图 20-13. 14 位 ADC 过采样右移 10 位

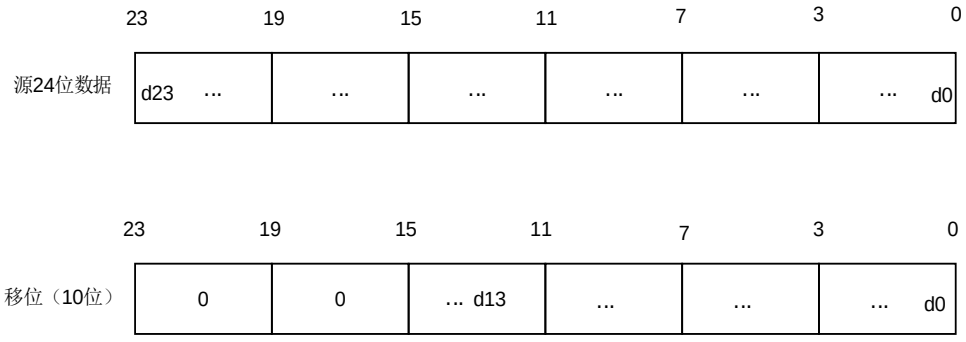


图 20-14. 数值例子 14 位 ADC 过采样右移 10 位

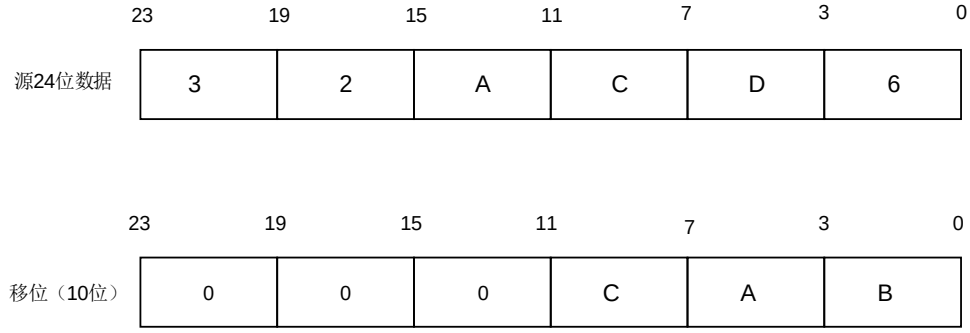


*表 20-7. 12 位 ADC 部分举例 N 和 M 的最大输出值（灰色部分表示截断）*给出了 N 和 M 各种组合的数据格式，初始转换值为 0xFFF。

表 20-7. 12 位 ADC 部分举例 N 和 M 的最大输出值（灰色部分表示截断）

Oversampling ratio	Max Raw data	No-shift OVSS=0000	1-bit shift OVSS=0001	2-bit shift OVSS=0010	3-bit shift OVSS=0011	4-bit shift OVSS=0100	5-bit shift OVSS=0101	6-bit shift OVSS=0110	7-bit shift OVSS=0111	8-bit shift OVSS=1000
2x	0x1FFE	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F	0x003F	0x001F
4x	0x3FFC	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F	0x003F
8x	0x7FF8	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F
16x	0xFFF0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF
32x	0x1FFE0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF
64x	0x3FFC0	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF
128x	0x7FF80	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF
256x	0xFFF00	0xFF00	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF
512x	0x1FFE00	0xFE00	0xFF00	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE
1024x	0x3FFC00	0xFC00	0xFE00	0xFF00	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC

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：

$$N*t_{ADC}=N*(t_{SMPL}+t_{CONV}) \quad (20\text{-}3)$$

20.5. ADC 同步模式

在具有两个 ADC 的设备上，可以使用 ADC 同步模式。

在 ADC 同步模式中，通过 ADC0 的触发器来同步 ADC1 的转换。根据 ADC_SYNCCTL 寄存器的 SYNCM[3:0]位来选择两个 ADC 的并行模式。

在 ADC 同步模式中，当转换配置成外部事件触发时，ADC1 的外部触发必须失能。常规序列通道的转换结果存储在 ADC 同步常规数据寄存器（ADC_SYNCDATA0 或 ADC_SYNCDATA1）中。

ADC 同步模式如*表 20-8. ADC 同步模式表*所示。

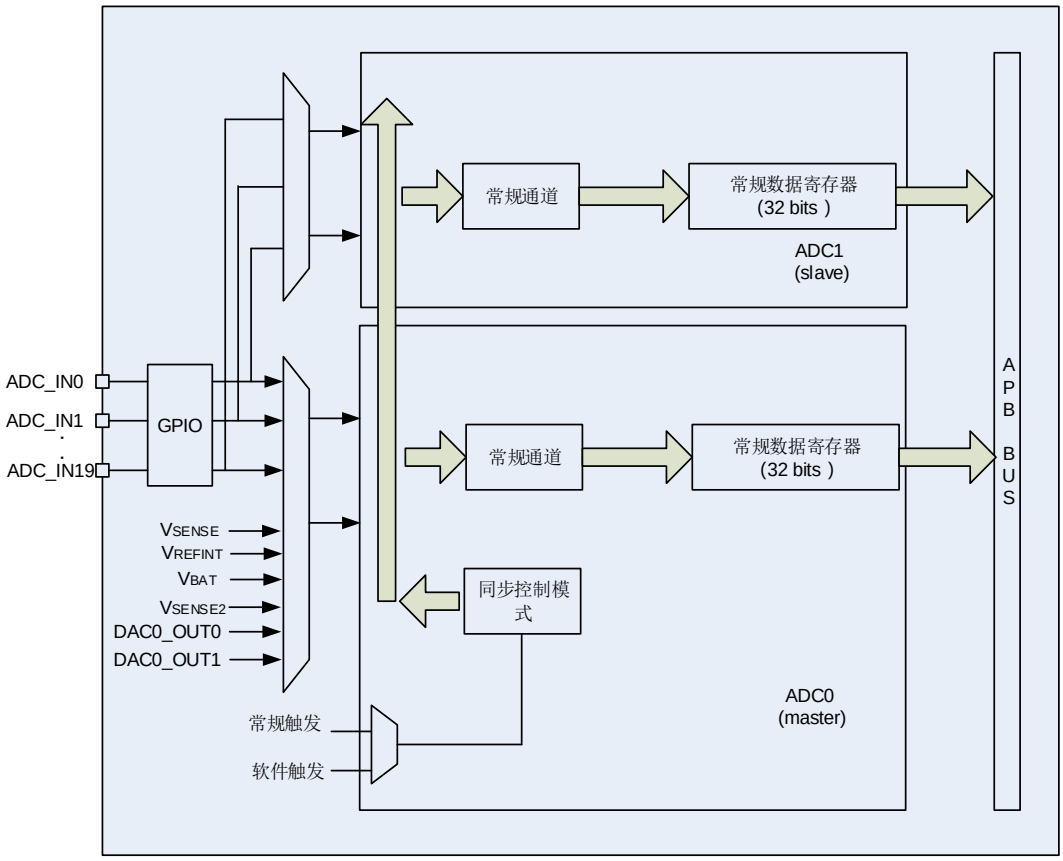
表 20-8. ADC 同步模式表

SYNCM[3: 0]	mode
0000	独立模式
0110	常规并行模式
0111	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 20-15. ADC 同步框图*所示。

图 20-15. ADC 同步框图



20.5.1. 独立模式

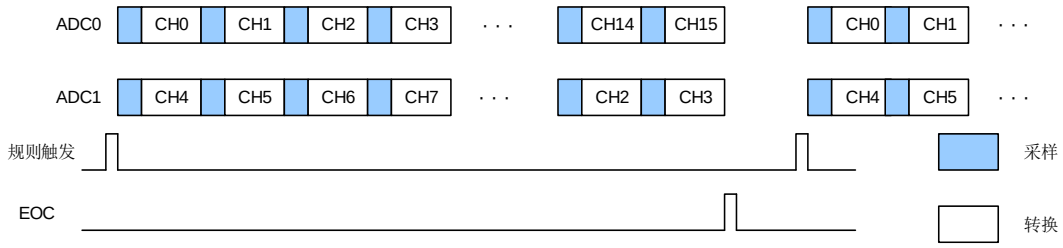
在这种模式下，ADC 同步是忽略的，每个 ADC 都独立工作。

20.5.2. 常规并行模式

设置 ADC_SYNCCTL 寄存器的 SYNCM[3:0]位为 0110，使能常规并行模式。在常规并行模式中，根据 ADC0 中选择的外部触发，所有的 ADC 并行的转换常规序列通道。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）。常规并模式的行为如*图 20-16. 基于 16 个通道的常规并行模式*所示。

图 20-16. 基于 16 个通道的常规并行模式



注意：

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

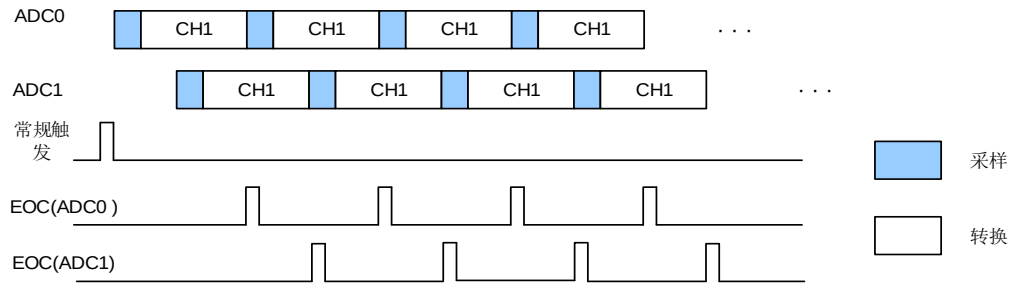
20.5.3. 常规跟随模式

设置 ADC_SYNCCTL 寄存器的 SYNCM[3:0]位为 0111，使能常规跟随模式。在跟随模式中，根据选择的外部触发，ADC0 开始转换常规序列通道。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 开始转换常规序列通道。以上描述中提到的常规序列只能包含一个常规通道。

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，选择的常规序列通道会被连续的转换。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）。跟随模式的行为如*图 20-17. 一个采用连续运行模式通道上的常规跟随模式*所示。

图 20-17. 一个采用连续运行模式通道上的常规跟随模式



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。

20.5.4. 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_SYNCDATA0 or ADC_SYNCDATA1）中，DMA 可以用来传输 ADC_SYNCDATA0 or ADC_SYNCDATA1 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_SYNCDATA1 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]。

ADC 同步 DMA 模式 0 适用于：

- ADC0 和 ADC1 工作在常规并行模式（SYNCM=0110）。

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 的常规序列转换结果中取出。对于每次 DMA 请求，DMA 通道的源地址固定为 ADC_SYNCDATA0 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]}。

ADC 同步 DMA 模式 1 适用于：

- ADC0 和 ADC1 工作在常规并行模式（SYNCM=0110）；
- ADC0 和 ADC1 工作在常规跟随模式（SYNCM=0111）。

20.6. 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映射到中断向量IRQ127。

20.7. ADC 寄存器

ADC0 基地址： 0x4001 2400

ADC1 基地址:：0x4001 2800

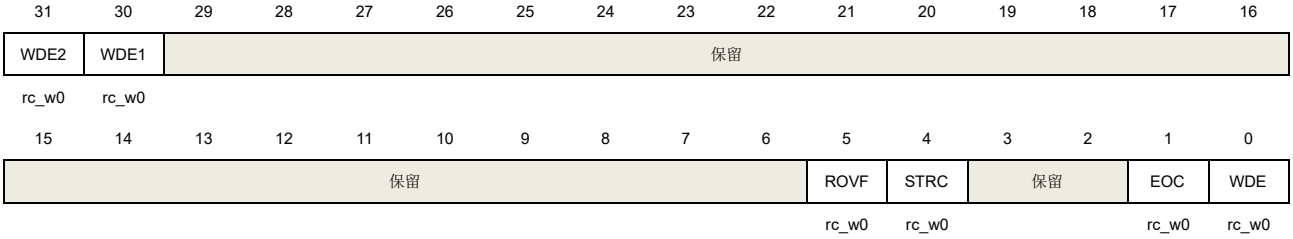
ADC2 基地址:：0x4001 2C00

20.7.1. 状态寄存器（ADC_STAT）

地址偏移：0x00

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31	WDE2	模拟看门狗 2 事件标志 0：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	常规序列数据寄存器溢出 0：常规序列数据寄存器没有溢出 1：常规序列数据寄存器溢出 在单次或多次模式中，当常规序列数据寄存器溢出时，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，则最后的常规序列数据会被丢失。 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	常规序列转换结束标志 0：转换没有结束 1：转换结束 常规序列转换结束时硬件置位，软件写 0 或读 ADC_RDATA 寄存器清除。
0	WDE0	模拟看门狗 0 事件标志 0： 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。

20.7.2. 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：中断禁止 1：中断使能
30	WDE1IE	WDE1 中断使能 0：中断禁止 1：中断使能
29:27	保留	必须保持复位值。
26	ROVFIE	常规序列溢出 ROVF 中断使能 0：ROVF 中断失能 1：ROVF 中断使能
25:24	DRES[1:0]	ADC0/1 数据分辨率 00:14 位 01:12 位 10:10 位 11:8 位

		ADC2 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 在单通道配置 0：模拟看门狗 0 在所有通道有效 1：模拟看门狗 0 在单通道有效
8	SM	扫描模式 0：扫描运行模式禁止 1：扫描运行模式使能
7	保留	必须保持复位值。
6	WDE0IE	WDE0 中断使能 0：中断禁止 1：中断使能
5	EOCIE	EOC 中断使能 0：中断禁止 1：中断使能
4:0	WDCHSEL[4:0]	模拟看门狗通道选择 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
其他值保留。

注意：ADC0 模拟输入通道 20 内部连接至 DAC0_OUT0。ADC1 模拟输入通道 16、通道 17 和通道 20 内部连接至电池、V_{REFINT} 输入、DAC0_OUT1。ADC2 模拟输入通道 17、通道 18、通道 19 和通道 20 内部连接至 V_{BAT}、温度传感器、V_{REFINT} 输入和高精度温度传感器。

20.7.3. 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						
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 20（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	软件触发常规序列启动转换 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（适用于 ADC0/1） 0：校准失调和失配 1：校准失调
26	保留	必须保持复位值。
25	VBATEN	在 ADC2 中，该位可由软件置位或清零。 使能 ADC1 的通道 16（外部电池电压的 1/4） 使能 ADC2 的通道 17（外部电池电压的 1/4） 0：外部电池电压的 1/4 失能 1：外部电池电压的 1/4 使能
24	INREFEN	在 ADC2 中，该位可由软件置位或清零。 使能 ADC1 的通道 17（内部参考电压） 使能 ADC2 的通道 19（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 ADC2 中，该位可由软件置位或清零。 ADC2 通道 18（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
22:13	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：在每个常规序列转换结束时，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，在每个常规序列转换结束时 DMA 机制产生一个 DMA 请求。
8	DMA	DMA 请求使能 0：DMA 请求禁止

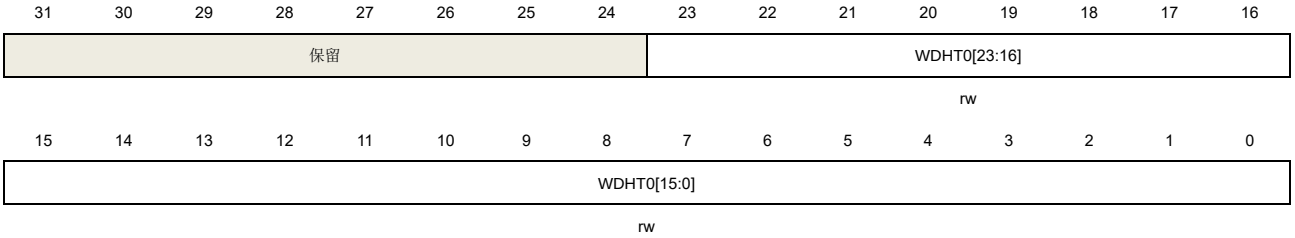
		1：DMA 请求使能
7	保留	必须保持复位值。
6:4	CALNUM[2:0]	校准次数 这些位定义了 ADC 的校准次数。 000：1 次 001：2 次 010：4 次 011：8 次 100：16 次 101：32 次（只针对 12 位 ADC） 其它：保留。
3	RSTCLB	校准复位 在校准寄存器初始化后该位可以软件置位和硬件清零。 0：校准寄存器初始化结束. 1：校准寄存器初始化开始
2	CLB	ADC 校准 0：校准结束 1：校准开始
1	CTN	连续模式 0：禁止连续运行模式 1：使能连续运行模式
0	ADCON	开启 ADC。该位从 0 变成 1 将唤醒 ADC。为了省电，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

20.7.4. 看门狗高阈值寄存器 0（ADC_WDHT0）

地址偏移：0x1C

复位值：0x00FF FFFF

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。

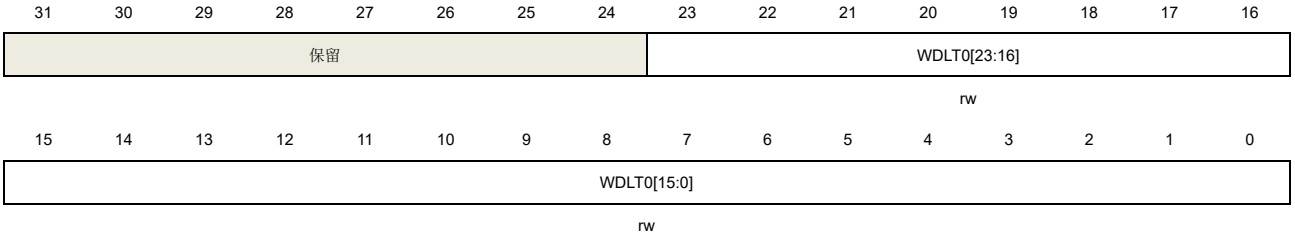
23:0	WDHT0[23:0]	模拟看门狗 0 高侧阈值，对于 ADC0/1 位 WDHT0 [23:0]，对于 ADC2 位 WDHT0 [11:0]。 这些位定义了模拟看门狗 0 的高侧阈值。

20.7.5. 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。



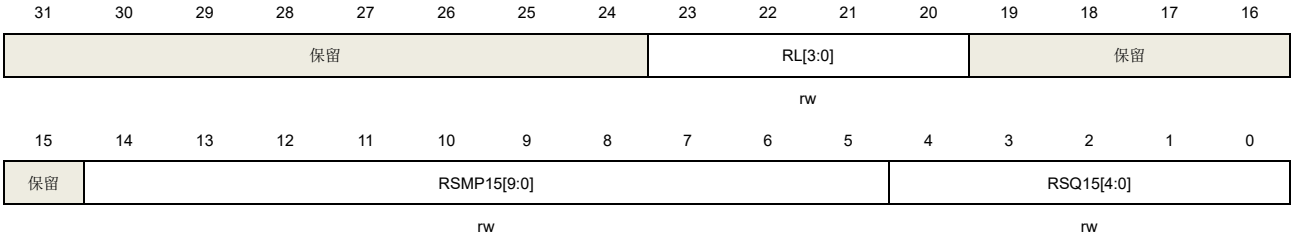
位/位域	名称	描述
31:24	保留	必须保持复位值。
23:0	WDLT0[23:0]	模拟看门狗 0 低侧阈值，对于 ADC0/1 位 WDLT0 [23:0]，对于 ADC2 位 WDLT0 [11:0]。 这些位定义了模拟看门狗 0 的低侧阈值。

20.7.6. 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]	常规通道序列长度 常规通道转换序列中的总的通道数目为 RL[3:0]+1。
19:15	保留	必须保持复位值。
14:5	RSMP15[9:0]	常规通道采样时间

10’d0：ADC0/1 为 3.5 周期，ADC2 为 2.5 周期
10’d1：ADC0/1 为 4.5 周期，ADC2 为 3.5 周期
10’d2：ADC0/1 为 5.5 周期，ADC2 为 4.5 周期
10’d3：ADC0/1 为 6.5 周期，ADC2 为 5.5 周期
10’d4：ADC0/1 为 7.5 周期，ADC2 为 6.5 周期
……
10’d638：ADC0/1 为 641.5 周期，ADC2 为 640.5 周期
10’d639：只有 ADC0/1 为 642.5 周期
……
10’d807：只有 ADC0/1 为 810.5 周期
其余位保留

4:0	RSQ15[4:0]	参考 RSQ0[4:0]描述

20.7.7. 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 10’d0：ADC0/1 为 3.5 周期，ADC2 为 2.5 周期 10’d1：ADC0/1 为 4.5 周期，ADC2 为 3.5 周期 10’d2：ADC0/1 为 5.5 周期，ADC2 为 4.5 周期 10’d3：ADC0/1 为 6.5 周期，ADC2 为 5.5 周期 10’d4：ADC0/1 为 7.5 周期，ADC2 为 6.5 周期 …… 10’d638：ADC0/1 为 641.5 周期，ADC2 为 640.5 周期 10’d639：只有 ADC0/1 为 642.5 周期 …… 10’d807：只有 ADC0/1 为 810.5 周期 其余位保留
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 10’d0：ADC0/1 为 3.5 周期，ADC2 为 2.5 周期 10’d1：ADC0/1 为 4.5 周期，ADC2 为 3.5 周期 10’d2：ADC0/1 为 5.5 周期，ADC2 为 4.5 周期 10’d3：ADC0/1 为 6.5 周期，ADC2 为 5.5 周期 10’d4：ADC0/1 为 7.5 周期，ADC2 为 6.5 周期 …… 10’d638：ADC0/1 为 641.5 周期，ADC2 为 640.5 周期 10’d639：只有 ADC0/1 为 642.5 周期 …… 10’d807：只有 ADC0/1 为 810.5 周期 其余位保留
4:0	RSQ13[4:0]	参考 RSQ0[4:0]描述

20.7.8. 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 10’d0：ADC0/1 为 3.5 周期，ADC2 为 2.5 周期 10’d1：ADC0/1 为 4.5 周期，ADC2 为 3.5 周期 10’d2：ADC0/1 为 5.5 周期，ADC2 为 4.5 周期 10’d3：ADC0/1 为 6.5 周期，ADC2 为 5.5 周期 10’d4：ADC0/1 为 7.5 周期，ADC2 为 6.5 周期 …… 10’d638：ADC0/1 为 641.5 周期，ADC2 为 640.5 周期 10’d639：只有 ADC0/1 为 642.5 周期 …… 10’d807：只有 ADC0/1 为 810.5 周期 其余位保留
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 10’d0：ADC0/1 为 3.5 周期，ADC2 为 2.5 周期 10’d1：ADC0/1 为 4.5 周期，ADC2 为 3.5 周期 10’d2：ADC0/1 为 5.5 周期，ADC2 为 4.5 周期 10’d3：ADC0/1 为 6.5 周期，ADC2 为 5.5 周期 10’d4：ADC0/1 为 7.5 周期，ADC2 为 6.5 周期 …… 10’d638：ADC0/1 为 641.5 周期，ADC2 为 640.5 周期 10’d639：只有 ADC0/1 为 642.5 周期 …… 10’d807：只有 ADC0/1 为 810.5 周期 其余位保留
4:0	RSQ11[4:0]	参考 RSQ0[4:0]描述

20.7.9. 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。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	RSMP10[9:0]										RSQ10[4:0]				
	rw										rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	RSMP9[9:0]										RSQ9[4:0]				
	rw										rw				

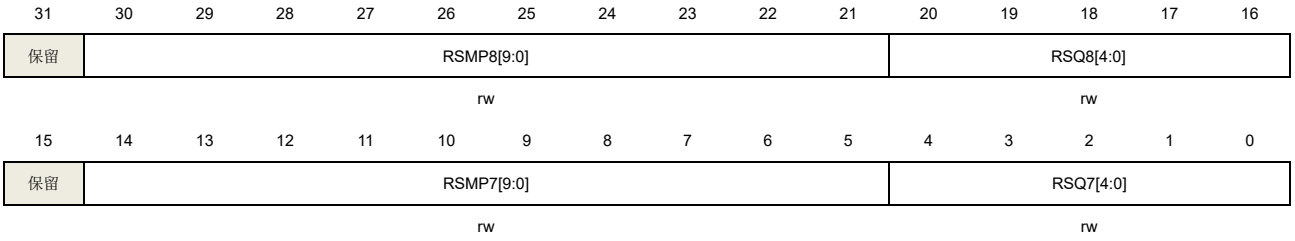
位/位域	名称	描述
31	保留	必须保持复位值。
30:21	RSMP10[9:0]	常规通道采样时间 10’d0：ADC0/1 为 3.5 周期，ADC2 为 2.5 周期 10’d1：ADC0/1 为 4.5 周期，ADC2 为 3.5 周期 10’d2：ADC0/1 为 5.5 周期，ADC2 为 4.5 周期 10’d3：ADC0/1 为 6.5 周期，ADC2 为 5.5 周期 10’d4：ADC0/1 为 7.5 周期，ADC2 为 6.5 周期 …… 10’d638：ADC0/1 为 641.5 周期，ADC2 为 640.5 周期 10’d639：只有 ADC0/1 为 642.5 周期 …… 10’d807：只有 ADC0/1 为 810.5 周期 其余位保留

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 10’d0：ADC0/1 为 3.5 周期，ADC2 为 2.5 周期 10’d1：ADC0/1 为 4.5 周期，ADC2 为 3.5 周期 10’d2：ADC0/1 为 5.5 周期，ADC2 为 4.5 周期 10’d3：ADC0/1 为 6.5 周期，ADC2 为 5.5 周期 10’d4：ADC0/1 为 7.5 周期，ADC2 为 6.5 周期 …… 10’d638：ADC0/1 为 641.5 周期，ADC2 为 640.5 周期 10’d639：只有 ADC0/1 为 642.5 周期 …… 10’d807：只有 ADC0/1 为 810.5 周期 其余位保留
4:0	RSQ9[4:0]	参考 RSQ0[4:0]描述

20.7.10. 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。



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 10’d0：ADC0/1 为 3.5 周期，ADC2 为 2.5 周期 10’d1：ADC0/1 为 4.5 周期，ADC2 为 3.5 周期 10’d2：ADC0/1 为 5.5 周期，ADC2 为 4.5 周期 10’d3：ADC0/1 为 6.5 周期，ADC2 为 5.5 周期 10’d4：ADC0/1 为 7.5 周期，ADC2 为 6.5 周期 …… 10’d638：ADC0/1 为 641.5 周期，ADC2 为 640.5 周期 10’d639：只有 ADC0/1 为 642.5 周期 …… 10’d807：只有 ADC0/1 为 810.5 周期

		其余位保留
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 10’d0：ADC0/1 为 3.5 周期，ADC2 为 2.5 周期 10’d1：ADC0/1 为 4.5 周期，ADC2 为 3.5 周期 10’d2：ADC0/1 为 5.5 周期，ADC2 为 4.5 周期 10’d3：ADC0/1 为 6.5 周期，ADC2 为 5.5 周期 10’d4：ADC0/1 为 7.5 周期，ADC2 为 6.5 周期 …… 10’d638：ADC0/1 为 641.5 周期，ADC2 为 640.5 周期 10’d639：只有 ADC0/1 为 642.5 周期 …… 10’d807：只有 ADC0/1 为 810.5 周期 其余位保留
4:0	RSQ7[4:0]	参考 RSQ0[4:0]描述

20.7.11. 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 10’d0：ADC0/1 为 3.5 周期，ADC2 为 2.5 周期 10’d1：ADC0/1 为 4.5 周期，ADC2 为 3.5 周期 10’d2：ADC0/1 为 5.5 周期，ADC2 为 4.5 周期 10’d3：ADC0/1 为 6.5 周期，ADC2 为 5.5 周期 10’d4：ADC0/1 为 7.5 周期，ADC2 为 6.5 周期 …… 10’d638：ADC0/1 为 641.5 周期，ADC2 为 640.5 周期 10’d639：只有 ADC0/1 为 642.5 周期 ……

		10’d807：只有 ADC0/1 为 810.5 周期 其余位保留
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 10’d0：ADC0/1 为 3.5 周期，ADC2 为 2.5 周期 10’d1：ADC0/1 为 4.5 周期，ADC2 为 3.5 周期 10’d2：ADC0/1 为 5.5 周期，ADC2 为 4.5 周期 10’d3：ADC0/1 为 6.5 周期，ADC2 为 5.5 周期 10’d4：ADC0/1 为 7.5 周期，ADC2 为 6.5 周期 …… 10’d638：ADC0/1 为 641.5 周期，ADC2 为 640.5 周期 10’d639：只有 ADC0/1 为 642.5 周期 …… 10’d807：只有 ADC0/1 为 810.5 周期 其余位保留
4:0	RSQ5[4:0]	参考 RSQ0[4:0]描述

20.7.12. 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 10’d0：ADC0/1 为 3.5 周期，ADC2 为 2.5 周期 10’d1：ADC0/1 为 4.5 周期，ADC2 为 3.5 周期 10’d2：ADC0/1 为 5.5 周期，ADC2 为 4.5 周期 10’d3：ADC0/1 为 6.5 周期，ADC2 为 5.5 周期 10’d4：ADC0/1 为 7.5 周期，ADC2 为 6.5 周期 …… 10’d638：ADC0/1 为 641.5 周期，ADC2 为 640.5 周期 10’d639：只有 ADC0/1 为 642.5 周期

		…… 10’d807：只有 ADC0/1 为 810.5 周期 其余位保留
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 10’d0：ADC0/1 为 3.5 周期，ADC2 为 2.5 周期 10’d1：ADC0/1 为 4.5 周期，ADC2 为 3.5 周期 10’d2：ADC0/1 为 5.5 周期，ADC2 为 4.5 周期 10’d3：ADC0/1 为 6.5 周期，ADC2 为 5.5 周期 10’d4：ADC0/1 为 7.5 周期，ADC2 为 6.5 周期 …… 10’d638：ADC0/1 为 641.5 周期，ADC2 为 640.5 周期 10’d639：只有 ADC0/1 为 642.5 周期 …… 10’d807：只有 ADC0/1 为 810.5 周期 其余位保留
4:0	RSQ3[4:0]	参考 RSQ0[4:0]描述

20.7.13. 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。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	RSMP2[9:0]										RSQ2[4:0]				
	rw										rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	RSMP1[9:0]										RSQ1[4:0]				
	rw										rw				

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 10’d0：ADC0/1 为 3.5 周期，ADC2 为 2.5 周期 10’d1：ADC0/1 为 4.5 周期，ADC2 为 3.5 周期 10’d2：ADC0/1 为 5.5 周期，ADC2 为 4.5 周期 10’d3：ADC0/1 为 6.5 周期，ADC2 为 5.5 周期 10’d4：ADC0/1 为 7.5 周期，ADC2 为 6.5 周期 …… 10’d638：ADC0/1 为 641.5 周期，ADC2 为 640.5 周期

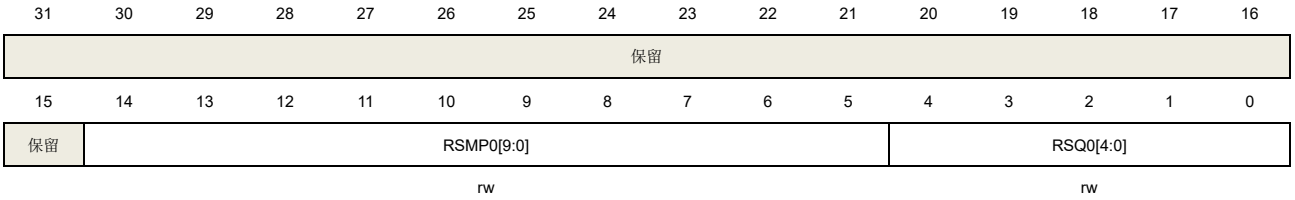
		10’d639：只有 ADC0/1 为 642.5 周期 …… 10’d807：只有 ADC0/1 为 810.5 周期 其余位保留
20:16	RSQ2[4:0]	参考 RSQ0[4:0]描述
15	保留	必须保持复位值。
14:5	RSMP1[9:0]	常规通道采样时间 10’d0：ADC0/1 为 3.5 周期，ADC2 为 2.5 周期 10’d1：ADC0/1 为 4.5 周期，ADC2 为 3.5 周期 10’d2：ADC0/1 为 5.5 周期，ADC2 为 4.5 周期 10’d3：ADC0/1 为 6.5 周期，ADC2 为 5.5 周期 10’d4：ADC0/1 为 7.5 周期，ADC2 为 6.5 周期 …… 10’d638：ADC0/1 为 641.5 周期，ADC2 为 640.5 周期 10’d639：只有 ADC0/1 为 642.5 周期 …… 10’d807：只有 ADC0/1 为 810.5 周期 其余位保留
4:0	RSQ1[4:0]	参考 RSQ0[4:0]描述

20.7.14. 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 10’d0：ADC0/1 为 3.5 周期，ADC2 为 2.5 周期 10’d1：ADC0/1 为 4.5 周期，ADC2 为 3.5 周期 10’d2：ADC0/1 为 5.5 周期，ADC2 为 4.5 周期 10’d3：ADC0/1 为 6.5 周期，ADC2 为 5.5 周期 10’d4：ADC0/1 为 7.5 周期，ADC2 为 6.5 周期 …… 10’d638：ADC0/1 为 641.5 周期，ADC2 为 640.5 周期

10’d639：只有 ADC0/1 为 642.5 周期

……

10’d807：只有 ADC0/1 为 810.5 周期

其余位保留

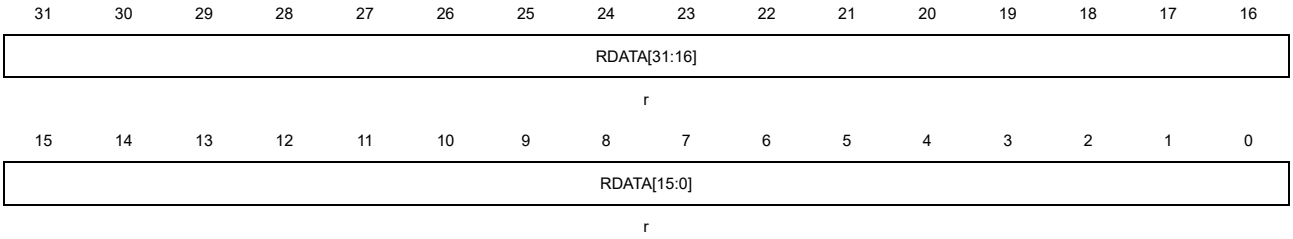
4:0	RSQ0[4:0]	通道编号（0..20）写入这些位来选择常规通道的第 n 个转换的通道。

20.7.15. 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。



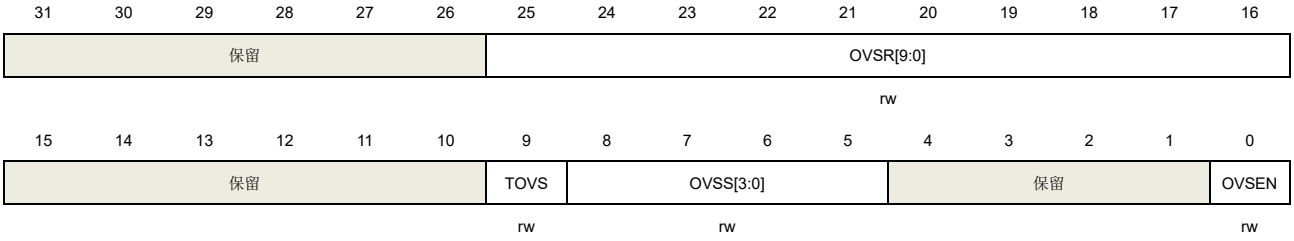
位/位域	名称	描述
31:0	RDATA[31:0]	常规通道数据，对于 ADC0/1，RDATA 为[31:0]，对于 ADC2，RDATA 为[15:0]。 这些位包含了常规通道的转换结果，只读。

20.7.16. 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，ADC0/1 为 1x~1024x，ADC2 为 1x~256x。 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，对于 ADC0/1，OVSS 范围 0000~1111，对于 ADC2，OVSS 范围 0000~1000。 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）。

20.7.17. 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。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留											AWD1CS[20:16]				
											rw				

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

位/位域	名称	描述
31:21	保留	必须保持复位值。
20:0	AWD1CS[20:0]	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[20:0] = 000..0，模拟看门狗1禁能。 **注意：** 1）通过AWD1CS位域配置的模拟看门狗1功能的通道，必须是ADC_RSQn寄存器和ADC_ISQ寄存器中配置的通道； 2）只有在ADC禁能（ADCON=0）时，才能软件写这些位。

20.7.18. 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。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留											AWD2CS[20:16]				
											rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWD2CS[15:0]															
rw															

位/位域	名称	描述
31:21	保留	必须保持复位值。
20:0	AWD2CS[20:0]	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[20:0] = 000..0，模拟看门狗2禁能。 **注意：** 1) 通过AWD2CS位域配置模拟看门狗2功能的通道，必须是ADC_RSQn寄存器和ADC_ISQ寄存器中配置的通道；

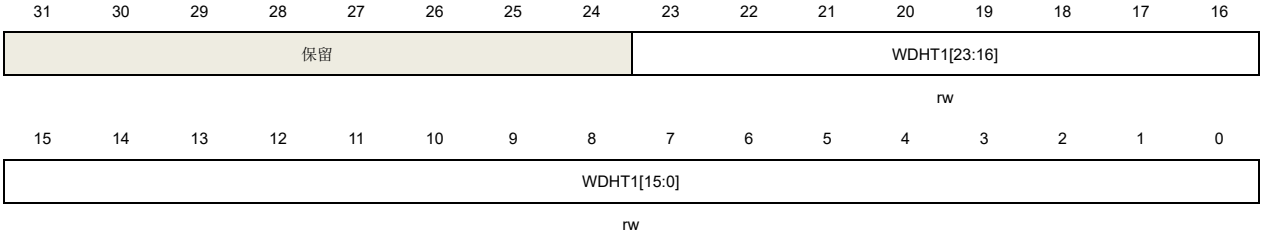
2) 只有在ADC禁能（ADCON=0）时，才能软件写这些位。

20.7.19. 看门狗 1 高阈值寄存器（ADC_WDHT1）

地址偏移：0xA8

复位值：0x00FF FFFF

该寄存器只能按字（32位）访问。



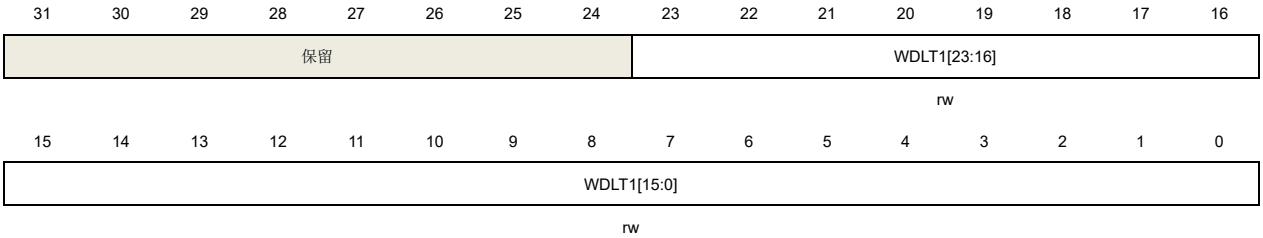
位/位域	名称	描述
31:24	保留	必须保持复位值。
23:0	WDHT1[23:0]	模拟看门狗1高侧阈值，ADC0/1为WDHT1[23:0]，ADC2为WDHT1[7:0]。 这些位定义了模拟看门狗1的高侧阈值。 **注意：**只有在ADC禁能（ADCON=0）时，才能软件写这些位。

20.7.20. 看门狗 1 低阈值寄存器（ADC_WDLT1）

地址偏移：0xA8

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



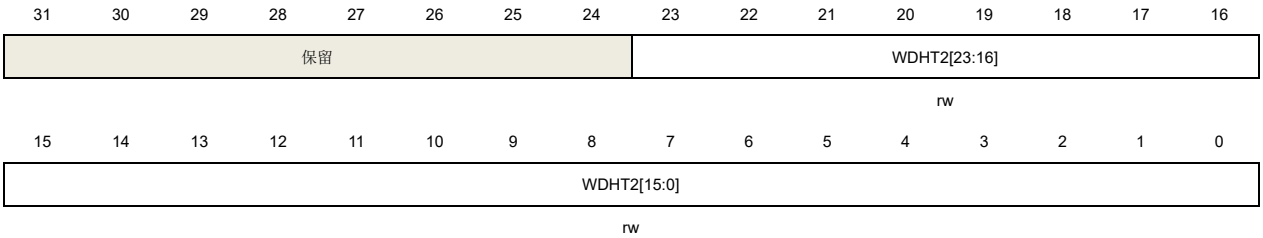
位/位域	名称	描述
31:24	保留	必须保持复位值。
23:0	WDLT1[23:0]	模拟看门狗1低侧阈值，ADC0/1为WDLT1[23:0]，ADC2为WDLT1[7:0]。 这些位定义了模拟看门狗1的低侧阈值。 **注意：**只有在ADC禁能（ADCON=0）时，才能软件写这些位。

20.7.21. 看门狗 2 高阈值寄存器（ADC_WDHT2）

地址偏移：0xB0

复位值：0x00FF FFFF

该寄存器只能按字（32位）访问。



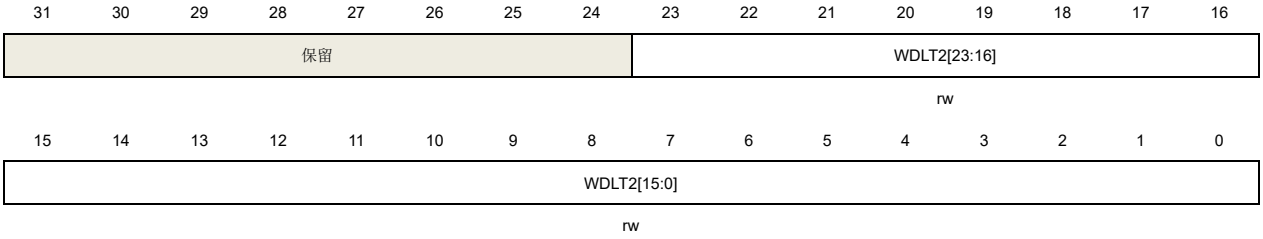
位/位域	名称	描述
31:24	保留	必须保持复位值。
23:0	WDHT2[23:0]	模拟看门狗2高侧阈值，ADC0/1为WDHT2[23:0]，ADC2为WDHT2[7:0]。 这些位定义了模拟看门狗2的高侧阈值。 **注意：**只有在ADC禁能（ADCON=0）时，才能软件写这些位。

20.7.22. 看门狗 2 低阈值寄存器（ADC_WDLT2）

地址偏移：0xB4

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



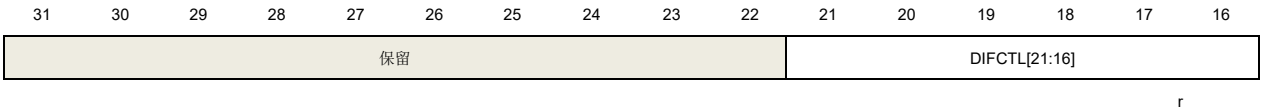
位/位域	名称	描述
31:24	保留	必须保持复位值。
23:0	WDLT2[23:0]	模拟看门狗2低侧阈值，ADC0/1为WDLT2[23:0]，ADC2为WDLT2[7:0]。 这些位定义了模拟看门狗2的低侧阈值。 **注意：**只有在ADC禁能（ADCON=0）时，才能软件写这些位。

20.7.23. 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。



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

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。

20.7.24. 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3个ADC状态的摘要。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								ADC2_ROVF	ADC2_STRC	保留		ADC2_EOC	ADC2_WDE2	ADC2_WDE1	ADC2_WDE0
								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
31:24	保留	必须保持复位值。
23	ADC2_ROVF	该位是 ADC2 的 ROVF 的镜像
22	ADC2_STRC	该位是 ADC2 的 STRC 的镜像
21:20	保留	必须保持复位值。
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

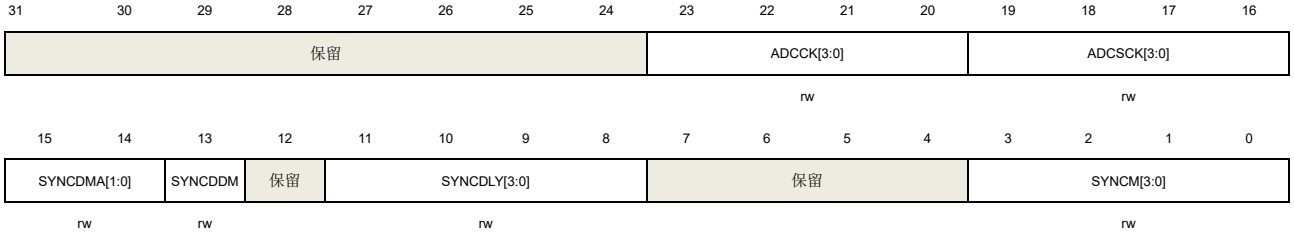
14	ADC1_STRC	该位是 ADC1 的 STRC 的镜像
13:12	保留	必须保持复位值。
11	ADC1_EOC	该位是 ADC1 的 EOC 的镜像
10	ADC1_WDE2	该位是 ADC1 的 WDE2 的镜像
9	ADC1_WDE1	该位是 ADC1 的 WDE1 的镜像
8	ADC1_WDE0	该位是 ADC1 的 WDE0 的镜像
7	ADC0_ROVF	该位是 ADC0 的 ROVF 的镜像
6	ADC0_STRC	该位是 ADC0 的 STRC 的镜像
5:4	保留	必须保持复位值。
3	ADC0_EOC	该位是 ADC0 的 EOC 的镜像
2	ADC0_WDE2	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

20.7.25. 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中不可用。

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	ADCCK[3:0]	ADC 时钟分频 这些位配置所有 ADC 的时钟，可以通过软件设置频率。 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 0000：ADC 同步模式失能。所有的 ADC 都独立工作。 0110：ADC0 和 ADC1 工作在常规并行模式。 0111：ADC0 和 ADC1 工作在常规跟随模式。

其它值保留。

20.7.26. 同步常规数据寄存器 0（ADC_SYNCDATA0）

地址偏移：0x308
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
SYNCDATA1[15:0]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNCDATA0[15:0]															
r															

位/位域	名称	描述
31:16	SYNCDATA1[15:0]	ADC 同步模式中，常规数据 1，且 SYNCDMA[1:0]= 2'b 10。
15:0	SYNCDATA0[15:0]	ADC 同步模式中，常规数据 0，且 SYNCDMA[1:0]= 2'b 10。

20.7.27. 同步常规数据寄存器 1（ADC_SYNCDATA1）

地址偏移：0x30C
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
SYNCDATA[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNCDATA[15:0]															

位/位域	名称	描述
31:0	SYNCDATA[31:0]	当 SYNCDMA[1:0]=2'b01 时，依次从 ADC 的常规数据（主/从）中选择。

21. 数模转换器（DAC）

21.1. 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 的两个通道可以独立或并发工作。

21.2. 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）；
- DACx 双通道并发模式；

*图 21-1. DAC 结构框图*为 DAC 的结构框图，*表 21-1. DAC 引脚*给出了引脚描述。

图 21-1. DAC 结构框图

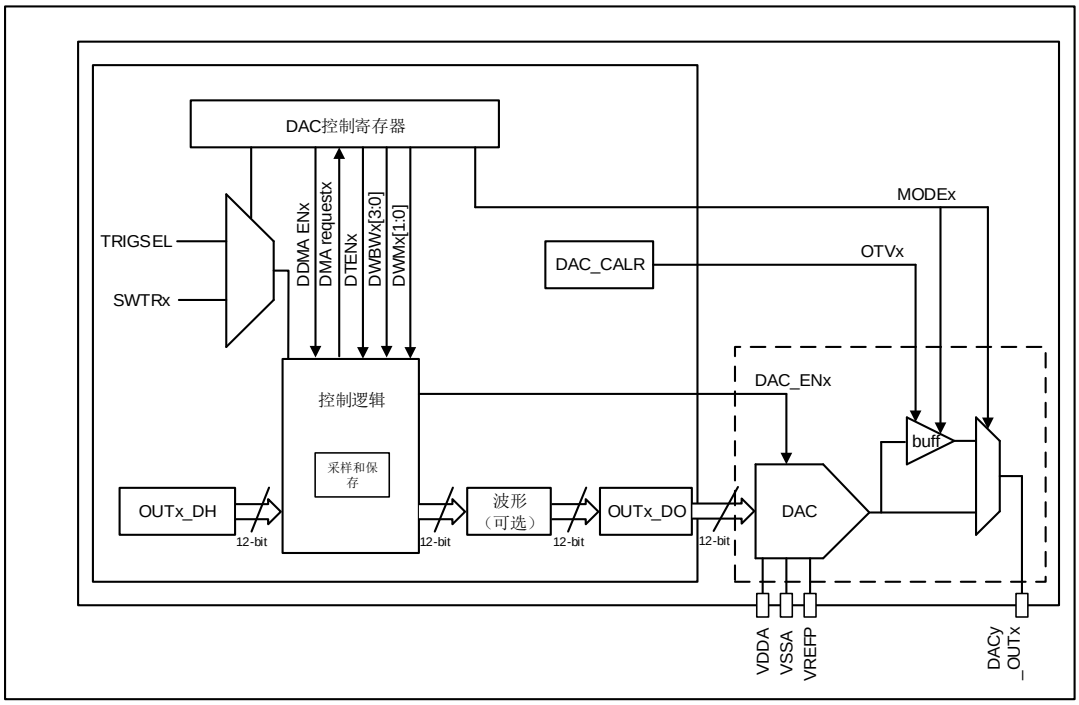


表 21-1. DAC 引脚

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 21-2. DAC 触发与输出

	DAC0	
通道	通道 0	通道 1
DAC 输出 I/O	PA4	PA5
DAC 输出 BUFFER 功能	•	•
TRIGSEL 触发功能	•	
软件触发功能	•	

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

21.3. 功能描述

21.3.1. 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。

21.3.2. 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。

21.3.3. DAC 数据配置

对于 12 位的 DAC 保持数据（OUTx_DH），可以通过对 OUTx_R12DH、OUTx_L12DH 和 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 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。

21.3.4. 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 21-3. DAC 外部触发*所示。

表 21-3. DAC 外部触发

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

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

21.3.5. 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（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（OUTx_DO）。

当 DAC 保持数据（OUTx_DH）加载到 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。

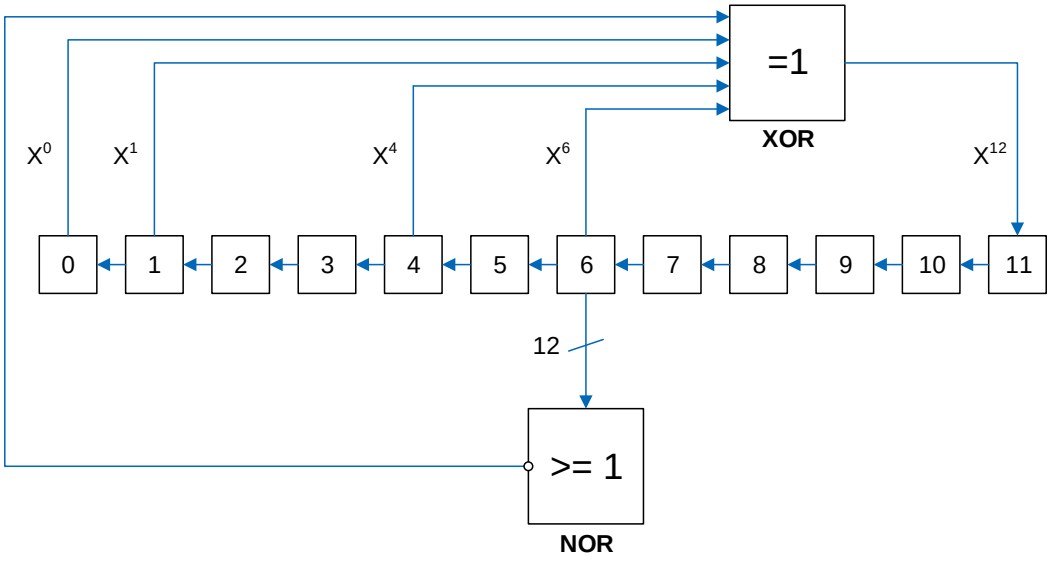
21.3.6. DAC 噪声波

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。

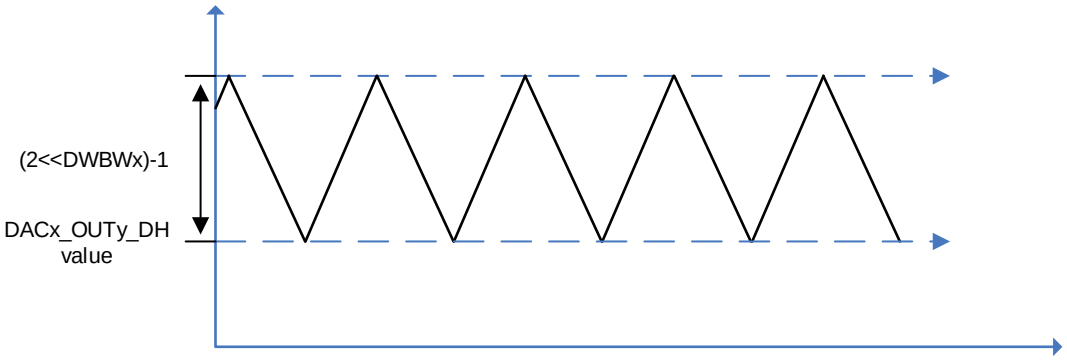
LFSR 噪声模式：在 DAC 控制逻辑中有一个线性反馈移位寄存器（LFSR）。在此模式下，LFSR 的值与 OUTx_DH 值相加后，被写入到 DAC 数据输出寄存器（OUTx_DO）。当配置的 DAC 噪声波位宽小于 12 时，LFSR 的值等于 LFSR 寄存器最低的 DWBWx 位，高位被屏蔽。

图 21-2. 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（OUTx_DO）。三角波幅值的最小值为 0，最大值为(2 << DWBWx) - 1。

图 21-3. DAC 三角噪声模式生成的波形



21.3.7. DAC 输出电压

DAC 引脚上的模拟输出电压取决于下面的等式：

$$V_{DAC_OUT}=V_{REFP}*OUTx_DO/4096 \quad (21\text{-}1)$$

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}。

21.3.8. DMA 请求

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

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

21.3.9. DAC 并发转换

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

有 3 个并发寄存器可用于加载 OUTx_DH 的值，分别是：DACC_R8DH、DACC_R12DH 和 DACC_L12DH 寄存器，配置其中的任意一个寄存器都可实现同时驱动两个 DAC。

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

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

噪声模式和噪声位宽可以根据使用情况配置为相同或不同。

21.3.10. 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：

$$V_{out}=(D/2^{N-1})*G*V_{REFP}+V_{of} \quad (21\text{-}2)$$

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。

21.3.11. DAC 模式

DAC 两个通道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和片上外设。

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和以及寄存器均由 APB1 时钟和 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，从而同步 APB1 时钟和 IRC32K 时钟。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。

时间计算：

以上三个阶段的计算时间均基于LXTAL/IRC32K时钟周期，为了配置足够的采样时间和刷新时间，请参考如下公式：

图 21-4. 采样时间和刷新时间计算公式

输出缓冲区	t_{sample}	$t_{refresh}$	t_{keep}
启用	$t_{wakeup} + R_{BON}*C_{SK}*ln(2^{N+1})$	$t_{wakeup} + R_{BON}*C_{SK}*ln(2*N_{LSB})$	$(V_{REFP}/2^{N})*N_{LSB}*C_{SK}/I_{leak}$
禁用	$t_{wakeup} + R_{BOFF}*C_{SK}*ln(2^{N+1})$	$t_{wakeup} + R_{BOFF}*C_{SK}*ln(2*N_{LSB})$	$(V_{REFP}/2^{N})*N_{LSB}*C_{SK}/I_{leak}$

注意：

(1) 上式中，t_{wakeup}为唤醒时间，是DAC从关闭状态到达输出设定值的时间。输出设定值的充电时间误差为1/2 LSB。

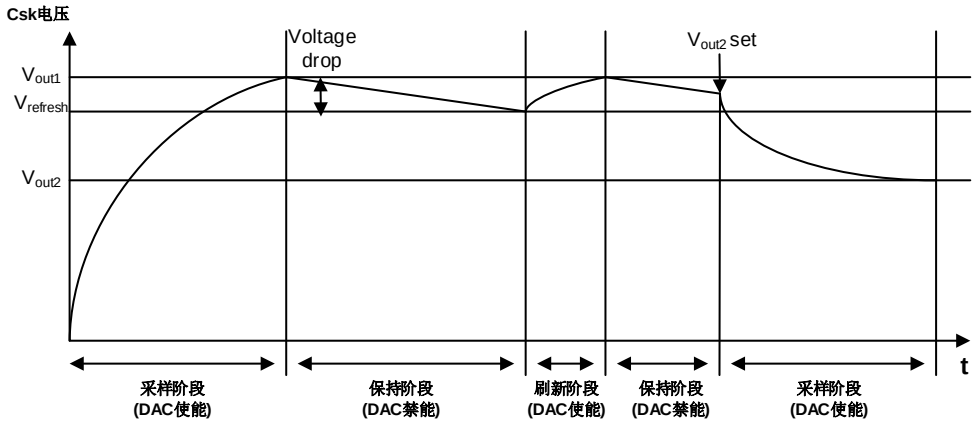
(2) R_{BON}/R_{BOFF}为输出缓冲区启用和禁用时的输出阻抗，C_{SK}为采样和保持电容值（内部或外部），当DAC_MDCR寄存器的MODEx[2:0]设置为3'b111时，内部电容用于为片上外设保持DAC的输出电压。

(3) 保持时间取决于保持阶段电容放电时输出漏电流的容差压降，LSBs的个数N_{LSB}代表电压压降，I_{leak}代表漏电流。

(4) R_{BON}、 R_{BOFF}、 C_{SK}、 t_{wakeup}的值请参考datasheet。

采样保持模式阶段图如下所示。

图 21-5. DAC 采样保持模式



21.3.12. 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。

21.4. DAC 寄存器

DAC0 基地址：0x4000 7400

21.4.1. DACx 控制寄存器 (DAC_CTL0)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	CALEN1	DDUDRIE1	DDMAEN1	DWBW1[3:0]				DWM1[1:0]		保留		DTSEL1[1:0]		DTEN1	DEN1
	rw	rw	rw	rw				rw				rw		rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CALEN0	DDUDRIE0	DDMAEN0	DWBW0[3:0]				DWM0[1:0]		保留		DTSEL0[1:0]		DTEN0	DEN0
	rw	rw	rw	rw				rw				rw		rw	rw

位/位域	名称	描述
31	保留	必须保持复位值。
30	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 1x：三角噪声模式
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：EXTRIG(外部触发来自 TRIGSEL) 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	保留	必须保持复位值。
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 1x：三角噪声模式
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：EXTRIG(外部触发来自 TRIGSEL) 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

21.4.2. DACx 软件触发寄存器 (DAC_SWT)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

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

位/位域	名称	描述
31:2	保留	必须保持复位值
1	SWTR1	DACx_OUT1 软件触发，由硬件清除。 0：软件触发禁能 1：软件触发使能
0	SWTR0	DACx_OUT0 软件触发，由硬件清除。 0：软件触发禁能 1：软件触发使能

21.4.3. DAC_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。

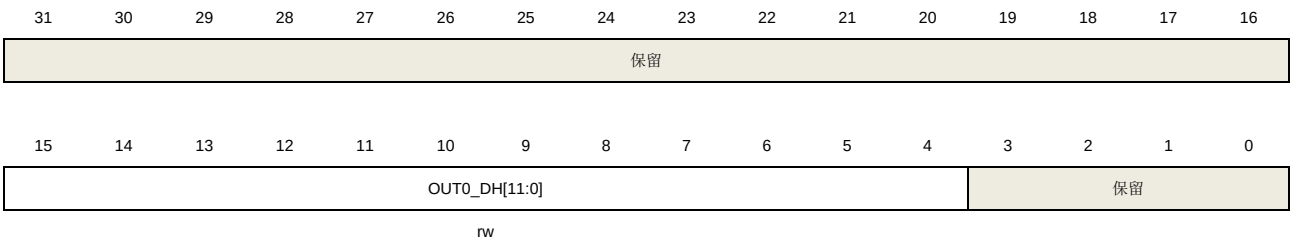


位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	OUT0_DH[11:0]	DACx_OUT0 12 位右对齐数据 这些位指定了将由 DACx_OUT0 转换的数据。

21.4.4. DAC_OUT0 12 位左对齐数据保持寄存器 (DAC_OUT0_L12DH)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



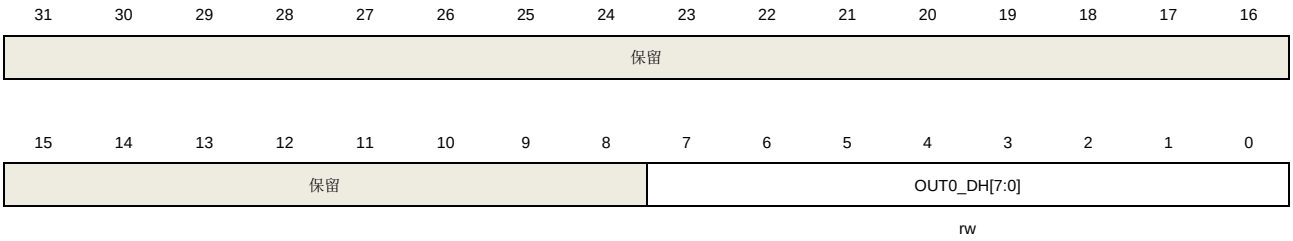
位/位域	名称	描述

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

21.4.5. DAC_OUT0 8 位右对齐数据保持寄存器 (DAC_OUT0_R8DH)

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

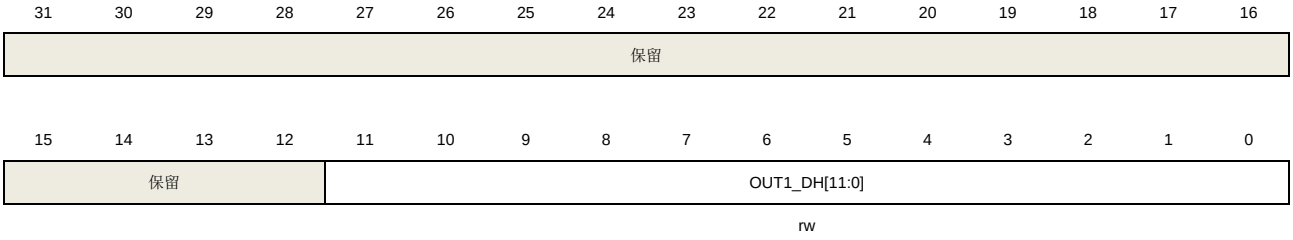


位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	OUT0_DH[7:0]	DACx_OUT0 8 位右对齐数据 这些位指定了将由 DACx_OUT0 转换的数据的最高 8 位有效位。

21.4.6. DAC_OUT1 12 位右对齐数据保持寄存器 (DAC_OUT1_R12DH)

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



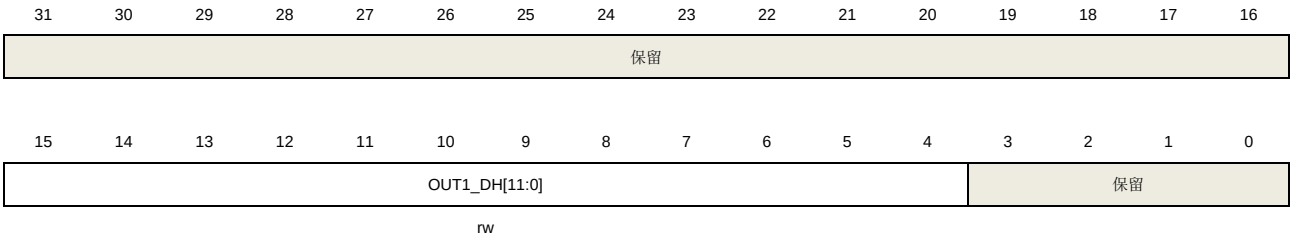
位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	OUT1_DH[11:0]	DACx_OUT1 12 位右对齐数据 这些位指定了将由 DACx_OUT1 转换的数据。

21.4.7. DAC_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

21.4.8. DAC_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。



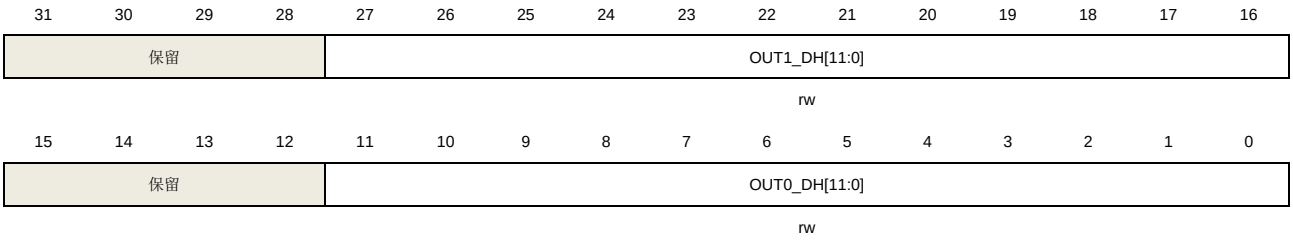
位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	OUT1_DH[7:0]	DACx_OUT1 8 位右对齐数据 这些位指定了将由 DACx_OUT1 转换的数据的 8 位最高有效位。

21.4.9. DAC 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。

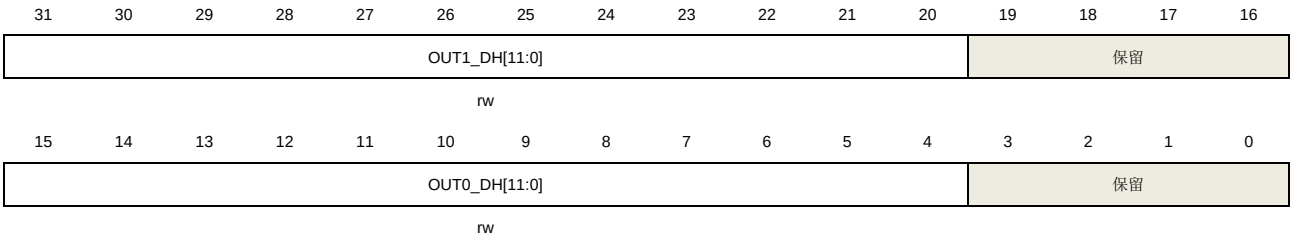


位/位域	名称	描述
31:28	保留	必须保持复位值
27:16	OUT1_DH[11:0]	DACx_OUT1 12 位右对齐数据 这些位指定了将由 DACx_OUT1 转换的数据。
15:12	保留	必须保持复位值
11:0	OUT0_DH[11:0]	DACx_OUT0 12 位右对齐数据 这些位指定了将由 DACx_OUT0 转换的数据。

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

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

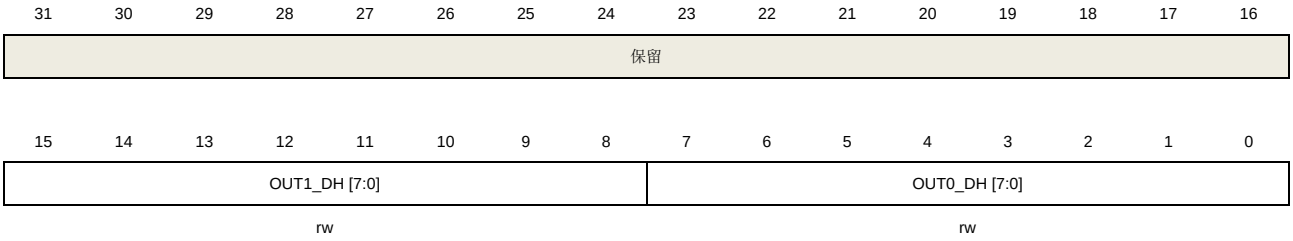


位/位域	名称	描述
31:20	OUT1_DH[11:0]	DACx_OUT1 12 位左对齐数据 这些位指定了将由 DACx_OUT1 转换的数据。
19:16	保留	必须保持复位值
15:4	OUT0_DH[11:0]	DACx_OUT0 12 位左对齐数据 这些位指定了将由 DACx_OUT0 转换的数据。
3:0	保留	必须保持复位值

21.4.11. DAC 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。

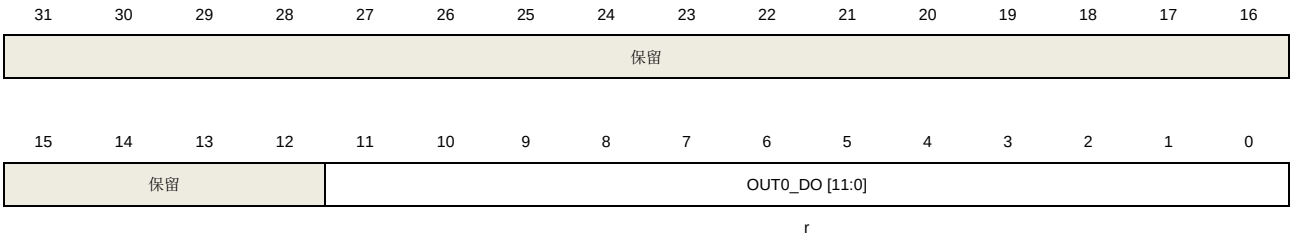


位/位域	名称	描述
31:16	保留	必须保持复位值
15:8	OUT1_DH[7:0]	DACx_OUT1 8 位右对齐数据 这些位指定了将由 DACx_OUT1 转换的数据的 8 位最高有效位。
7:0	OUT0_DH[7:0]	DACx_OUT0 8 位右对齐数据 这些位指定了将由 DACx_OUT0 转换的数据的 8 位最高有效位。

21.4.12. DAC_OUT0 数据输出寄存器 (DAC_OUT0_DO)

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



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

21.4.13. DAC_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。



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。

21.4.14. DAC 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 当使能采样保持模式后，该位由系统设置，当 DACx_SKSTR1 正在执行写操作时，该位置 1；当完成写操作后，硬件清零。 0：DAC_SKSTR1 没有进行写操作 1：DAC_SKSTR1 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 当使能采样保持模式后，该位由系统设置，当 DAC_SKSTR0 正在执行写操作时，该位置 1；当完成写操作后，硬件清零。 0：DAC_SKSTR0 没有进行写操作 1：DAC_SKSTR0 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。

21.4.15. DAC 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				
	rw												rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											OTV0				
													rw		

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。

21.4.16. DAC 模式寄存器 (DAC_MDCR)

地址偏移：0x3C

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													MODE1[2:0]		
														rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													MODE0[2:0]		
														rw	

位/位域	名称	描述

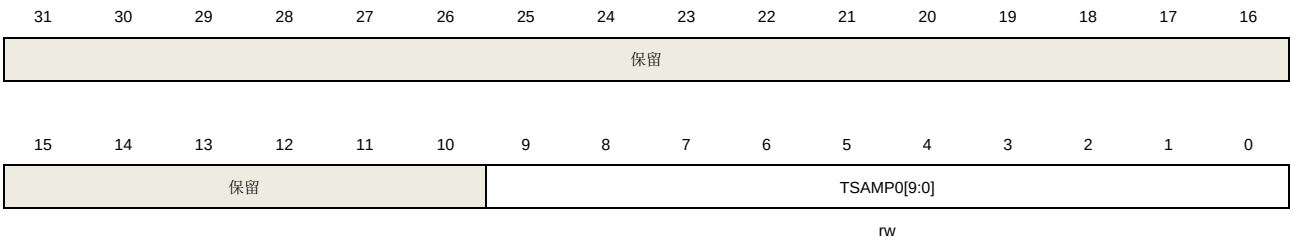
31:19	保留	必须保持复位值。
18:16	MODE1[2:0]	DACx_OUT1 模式 当 DAC_CTL0 寄存器中 DEN1=0 和 CALEN1=0 时才可对这些位进行写操作。当 DAC_CTL0 寄存器中 DEN1=1 或 CALEN1=1，写操作被忽略。 -普通模式下 DACx_OUT1 000:DACx_OUT1 连接到外部引脚，缓冲区启用。 001: DACx_OUT1 连接到外部引脚和片上外设，缓冲区启用。 010:DACx_OUT1 连接到外部引脚，缓冲区禁用。 011: DACx_OUT1 连接到片上外设，缓冲区禁用。 -采样保持模式下 DACx_OUT1 100:DACx_OUT1 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。
15:3	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。

21.4.17. DAC 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。



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。

21.4.18. DAC 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。

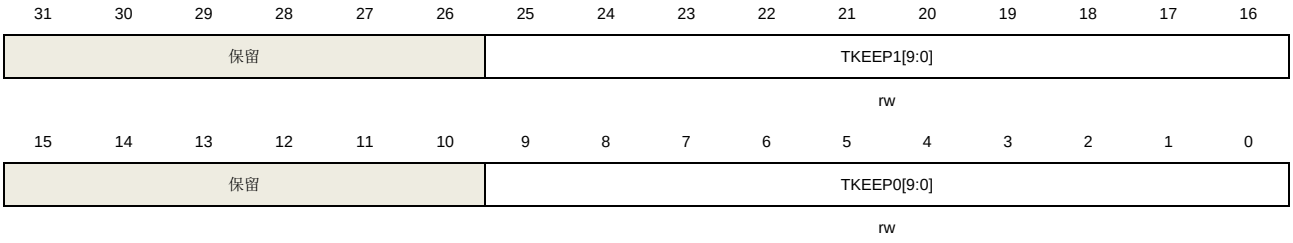


位/位域	名称	描述
31:10	保留	必须保持复位值。
9:0	TSAMP1[9:0]	DACx_OUT1 采样时间。

21.4.19. DAC 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。



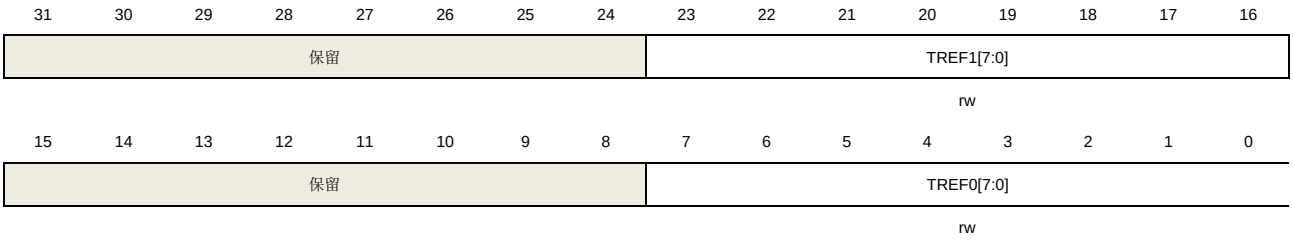
位/位域	名称	描述
31:26	保留	必须保持复位值。
25:16	TKEEP1[9: 0]	DACx_OUT1 保持时间(仅在采样保持模式有效)。
15:10	保留	必须保持复位值。
9:0	TKEEP0[9: 0]	DACx_OUT0 保持时间(仅在采样保持模式有效)。

21.4.20. DAC 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)。

22. 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。

22.1. 独立看门狗定时器（FWDGT）

22.1.1. 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。

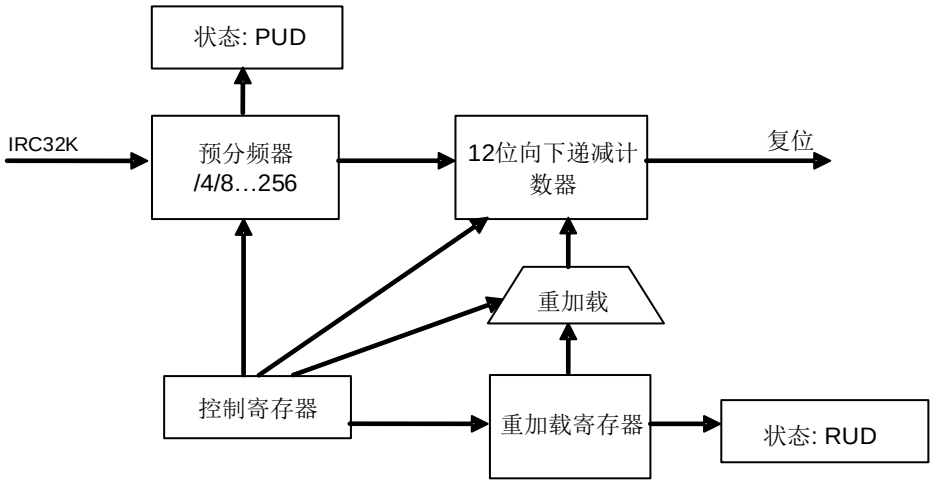
22.1.2. 主要特征

- 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。

22.1.3. 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 22-1. 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 22-1. 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™-M7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 22-1. 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
1/16	010	0.5	2048
1/32	011	1.0	4096
1/64	100	2.0	8192

预分频系数	PSC[2:0] 位	最小超时(ms) RLD [11:0]=0x000	最大超时(ms) RLD [11:0]=0xFFF
1/128	101	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：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。

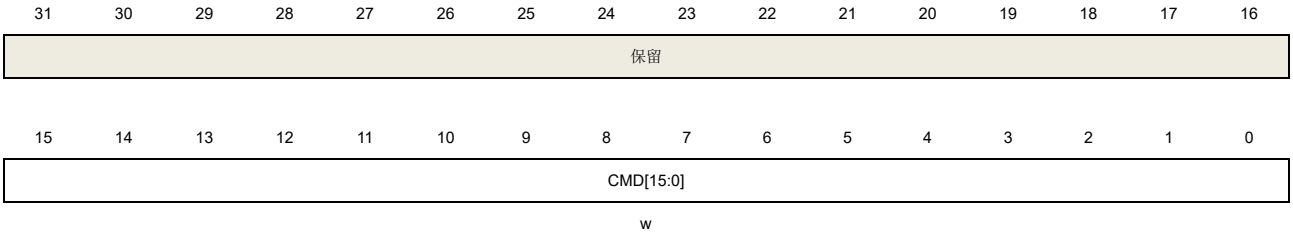
22.1.4. FWDGT 寄存器

FWDGT 基地址：0x5800 4800

控制寄存器（FWDGT_CTL）

地址偏移：0x00
复位值：0x0000 0000

该寄存器可以按字（32 位）访问。

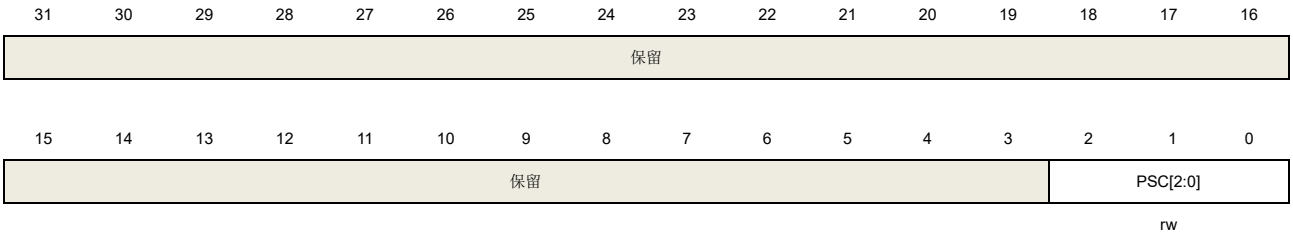


位/位域	名称	说明
31:16	保留	必须保持复位值。
15:0	CMD[15:0]	只可写，写入不同的值来产生不同的功能 0x5555：关闭FWDGT_PSC和FWDGT_RLD的写保护 0xCCCC：开启独立看门狗定时器计数器。计数减到0时产生中断 0xAAAA：重装载计数器

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。



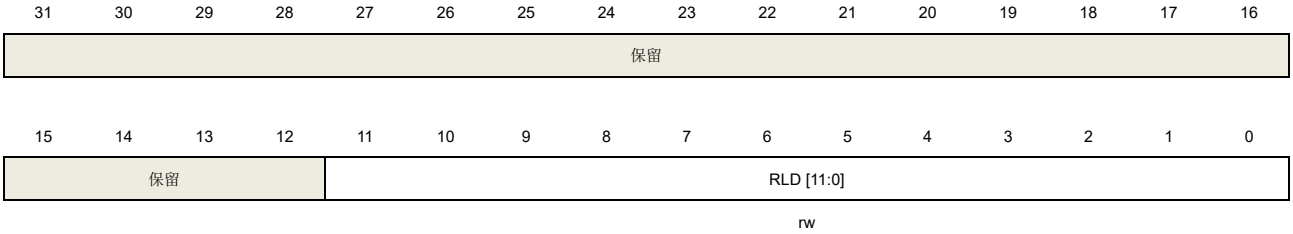
位/位域	名称	说明
31:3	保留	必须保持复位值。
2:0	PSC[2:0]	独立看门狗定时器计时预分频选择。写这些位之前要通过向FWDGT_CTL寄存器写0x5555去除写保护。在改写这个寄存器的过程中，FWDGT_STAT寄存器的PUD位被置1，此时读取此寄存器的值都是无效的。 000：1/4 001：1/8 010：1/16

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。

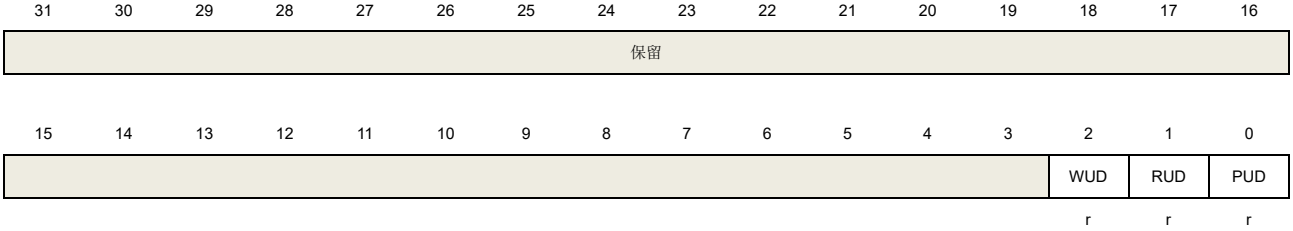


位/位域	名称	说明
31:12	保留	必须保持复位值。
11:0	RLD[11:0]	独立看门狗定时器计数器重装载值，向FWDGT_CTL寄存器写入0xAAAA的时候，这个值会被更新到看门狗定时器计数器中。 这些位有写保护功能。在写这些位之前需向FWDGT_CTL寄存器中写0x5555。在改写这个寄存器的过程中，FWDGT_STAT寄存器的RUD位被置1，从此寄存器中读取的任何值都是无效的。 如果应用需要使用几个重装载值，改变重加载值之前必须等到RUD位被清0。更新了重加载寄存器的值后，在代码持续执行之前不必等待RUD值被清零。

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。

22.2. 窗口看门狗定时器（WWDGT）

22.2.1. 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APB3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。

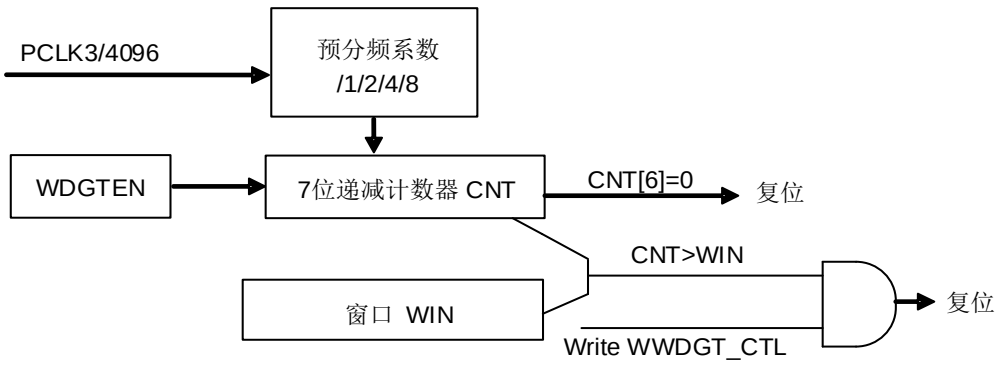
22.2.2. 主要特征

- 可编程的7位独立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。

22.2.3. 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 22-2. 窗口看门狗定时器框图



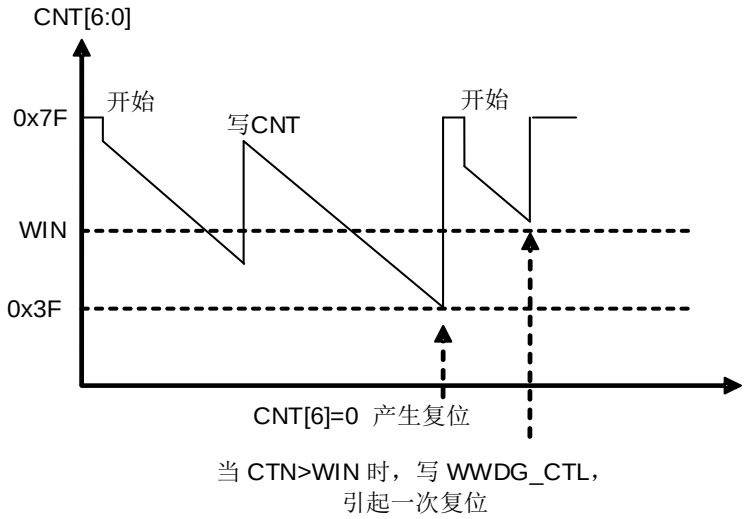
上电复位之后看门狗定时器总是关闭的。软件可以向WWDGT_CTL的WDGTEN写1开启看门狗定时器。窗口看门狗定时器打开后，计数器始终递减计数，计数器配置的值应该大于0x3F，也就是说CNT[6]位应该被置1。CNT[5:0]决定了两次重装载之间的最大间隔时间。计数器的递减速度取决于APB3时钟和预分频器(WWDGT_CFG寄存器的PSC[1:0]位)。

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 22-3. 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：

$$t_{WWDGT}=t_{PCLK3} \times 4096 \times 2^{PSC} \times (CNT[5:0]+1) \quad (ms) \qquad (22\text{-}1)$$

其中：

t_{WWDGT}：窗口看门狗定时器的超时时间

t_{PCLK3}： APB3以ms为单位的时钟周期

t_{WWDGT} 的最大值和最小值请参考*表 22-2. 在 150MHz (fPCLK3)时的最大/最小超时值*。

表 22-2. 在 150MHz (f_{PCLK3})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	27.30 μs	1.75 ms
1/2	01	54.61 μs	3.50 ms
1/4	10	109.22 μs	6.99 ms
1/8	11	218.45 μs	13.98 ms

如果DBG模块中的WWDGT_HOLD位被清0，即使Cortex™-M7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。

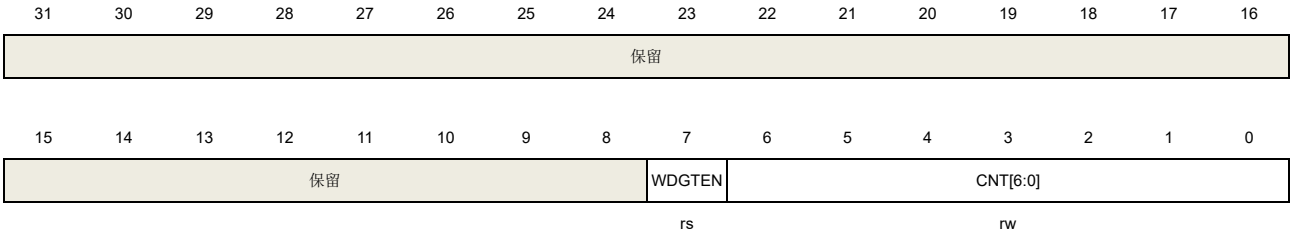
22.2.4. WWDGT 寄存器

WWDGT 基地址：0x5000 3000

控制寄存器（WWDGT_CTL）

地址偏移：0x00
复位值：0x0000 007F

该寄存器可以按字（32 位）访问。

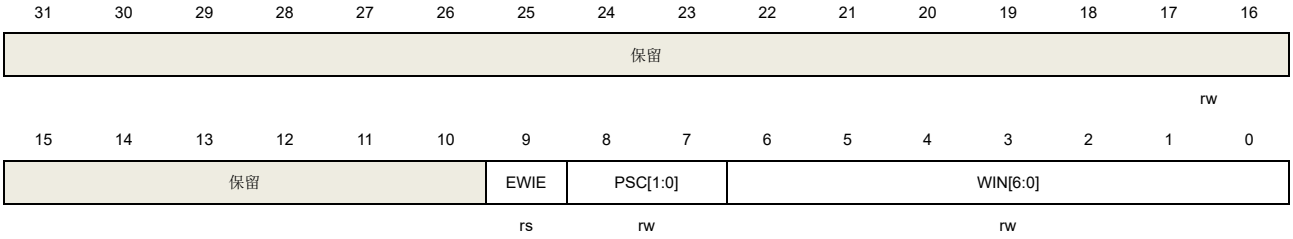


位/位域	名称	说明
31:8	保留	必须保持复位值。
7	WDGTEN	开启窗口看门狗定时器，硬件复位的时候清0，写0无效。 0：关闭窗口看门狗定时器 1：开启窗口看门狗定时器
6:0	CNT[6:0]	看门狗定时器计数器的值。当计数值从0x40降到0x3F时，产生看门狗定时器复位。当计数器值高于窗口值的时候，写计数器可以产生看门狗定时器复位。

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：(PCLK3 / 4096) / 1

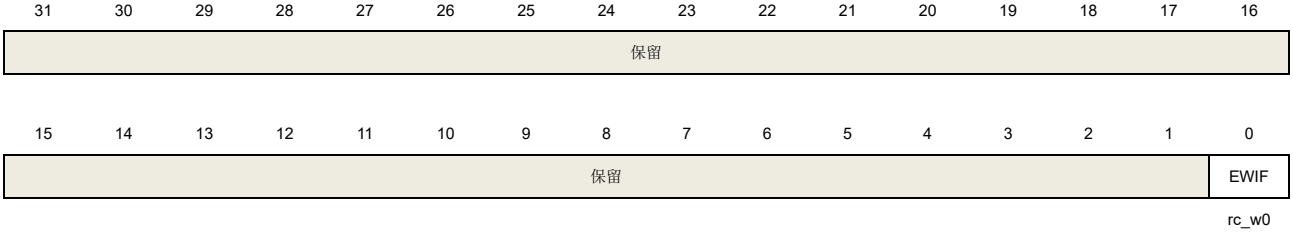
01：(PCLK3 / 4096) / 2
10：(PCLK3 / 4096) / 4
11：(PCLK3 / 4096) / 8

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或达到窗口值之前，更新计数器，即使中断没有被使能（WWDGT_CFG中的EWIE位为0）该位也会被硬件置1。这个bit可以通过写0清零，写1无效。

23. 实时时钟（RTC）

23.1. 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。

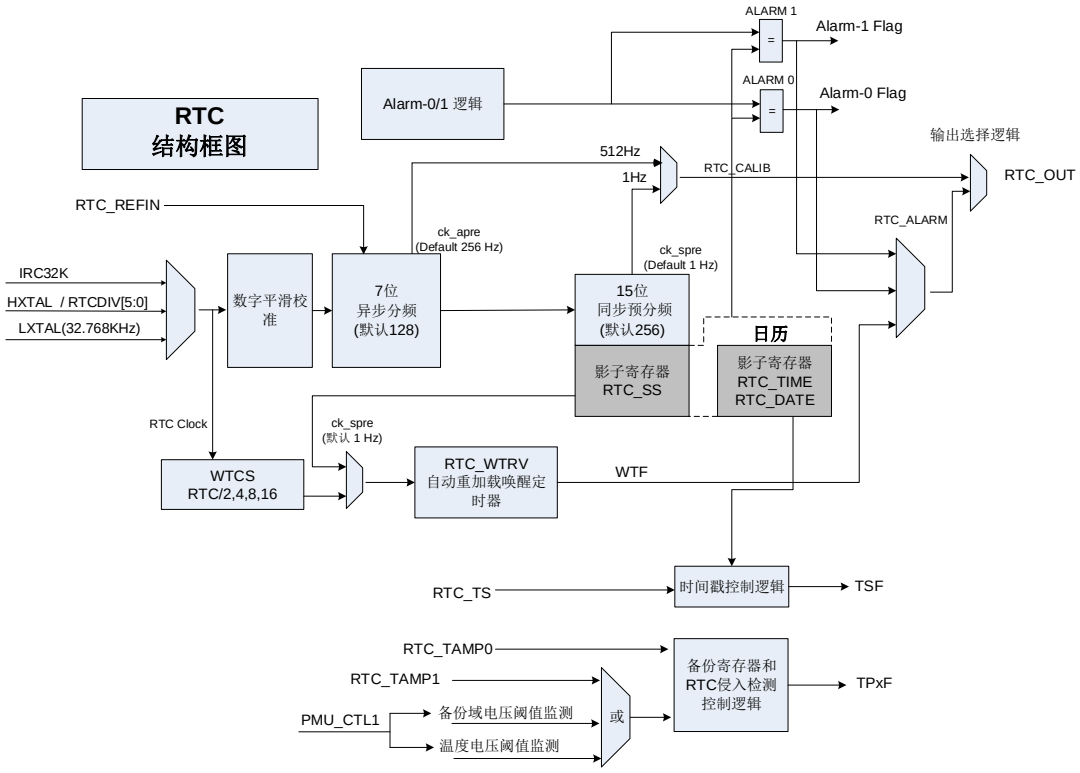
23.2. 主要特征

- 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和RTC_TAMP1）。
- 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。

23.3. 功能描述

23.3.1. 结构框图

图 23-1. 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)；
  - 侵入事件检测 1：RTC_TAMP1(PC1)；
  - 参考时钟输入：RTC_REFIN(PB15 或 PB13)；
  - 侵入事件检测 1 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 1。

23.3.2. 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 RTCDIV[5:0]（由 RCU_CFG0 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：

$$f_{ck_apre} = \frac{f_{rtcclk}}{FACTOR_A + 1} \quad (16\text{-}1)$$

$$f_{ck_spre} = \frac{f_{ck_apre}}{FACTOR_S + 1} = \frac{f_{rtcclk}}{(FACTOR_A + 1)*(FACTOR_S + 1)} \quad (16\text{-}2)$$

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。

23.3.3. 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。

23.3.4. 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。

23.3.5. 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。

23.3.6. 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。

23.3.7. 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。

23.3.8. 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 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。

23.3.9. 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。

23.3.10. 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。

23.3.11. 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：

$$f_{cal}=f_{rtcclk} \times(1+\frac{FREQI\times512-CMSK}{2^{N}+CMSK-FREQI\times512}) \quad (16\text{-}3)$$

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：

$$f_{cal}=f_{rtcclk} \times(1+\frac{256-CMSK}{2^{N}+CMSK-256}) \quad (16\text{-}4)$$

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。

23.3.12. 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。

23.3.13. 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：

1. 默认配置会擦除RTC备份寄存器，BKP SRAM和RTDEC寄存器
2. 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：

1. 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
2. 当 TPTS=1时（入侵事件的时间戳），延迟为3个ck_apre 周期
3. 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。

23.3.14. 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：

$$f_{rtc_calib} = \frac{f_{rtcclk}}{(FACTOR_A+1)\times(FACTOR_S+1)} \quad (16\text{-}5)$$

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。

23.3.15. 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。

23.3.16. 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23-1. RTC引脚配置*

表 23-1. 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 / Shutdown模式下不可用。

23.3.17. RTC 省电模式管理

表 23-2. 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

23.3.18. 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 23-3. RTC 中断控制

中断	事件标志	控制位	退出睡眠模式	退出深度睡眠模式和待机模式
闹钟0	ALRM0F	ALRM0IE	Y	Y(1)
闹钟1	ALRM1F	ALRM1IE	Y	Y(1)
唤醒	WTF	WTIE	Y	Y(1)
时间戳	TSF	TSIE	Y	Y(1)
侵入0	TP0F	TPIE	Y	Y(1)
侵入1	TP1F	TPIE	Y	Y(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。

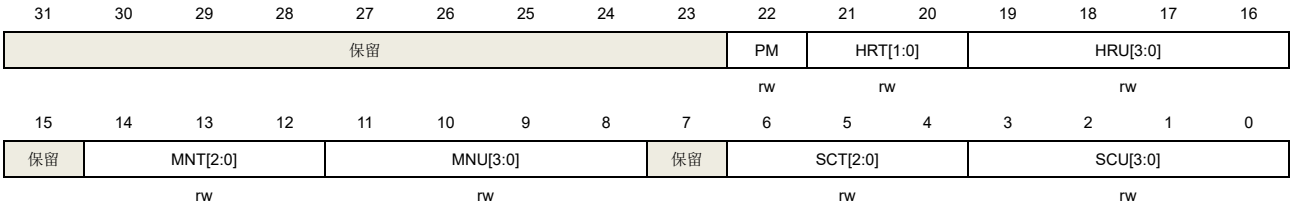
23.4. RTC 寄存器

RTC基地址：0x5800 4000

23.4.1. 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

23.4.2. 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

23.4.3. 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							ITSEN	COEN	OS[1:0]		OPOL	COS	DSM	S1H	A1H
							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
31:25	保留	必须保持复位值。
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}

23.4.4. 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
TP1F	保留	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	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 对 RTC_HRFC 进行软件写操作，该位被硬件置 1。当平滑校准周期完成后，该位被硬件清 0。
15	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。
14	保留	必须保持复位值。
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。
8	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。

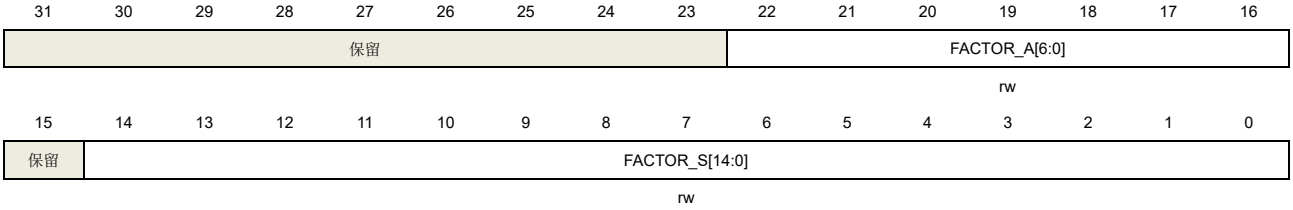
8	ALRM0F	Alarm0 发生标志 当现在的时间/日期与闹钟 0 设置的时间/日期匹配的时候，该位会通过硬件置 1。可以通过向该位软件写 0 来清除。
7	INITM	进入初始化模式 0：自由运行模式 1：进入初始化模式设置时间/日期和预分频，计数器将停止运行
6	INITF	初始化状态标志 该位被硬件置 1，初始化状态时可以设置日历寄存器和预分频器。 0：日历寄存器和预分频器的值不能改变 1：日历寄存器和预分频器的值可以改变
5	RSYNF	寄存器同步标志 每 2 个 RTCCLK 将会由硬件置 1 一次，同时会复制当前日历时间/日期到影子日历寄存器。初始化模式（INITM），移位操作挂起标志（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

23.4.5. 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。

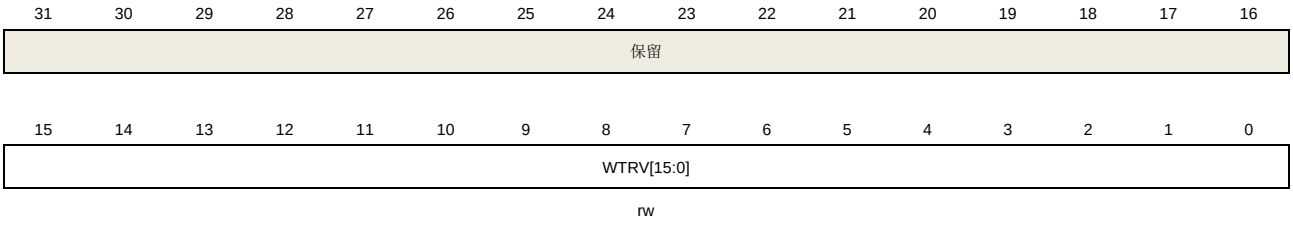


位/位域	名称	描述
31:23	保留	必须保持复位值。
22:16	FACTOR_A[6:0]	异步预分频系数 ck_apre 频率 = RTCCLK 频率/(FACTOR_A+1)
15	保留	必须保持复位值。
14:0	FACTOR_S[14:0]	同步预分频系数 ck_spre 频率 = ck_apre 频率/(FACTOR_S+1)

23.4.6. 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

23.4.7. 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

23.4.8. 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			

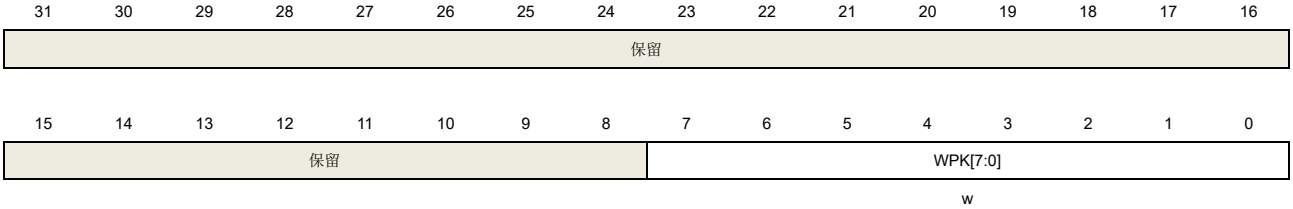
位/位域	名称	描述
31	MSKD	闹钟日期位域屏蔽位 0：不屏蔽日期/天位域 1：屏蔽日期/天位域
30	DOWS	星期选择 0：此时 DAYU[3:0] 代表日期个位值 1：此时 DAYU[3:0] 代表星期几，此时 DAYT[1:0]无意义
29:28	DAYT[1:0]	日期十位值，以 BCD 码格式存储
27:24	DAYU[3:0]	日期个位值或星期天数，以 BCD 码格式存储
23	MSKH	闹钟小时位域屏蔽位 0：不屏蔽小时位域 1：屏蔽小时位域
22	PM	AM/PM 标志 0：AM 或 24 小时制 1：PM
21:20	HRT[1:0]	小时十位值，以 BCD 码形式存储
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

23.4.9. 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。

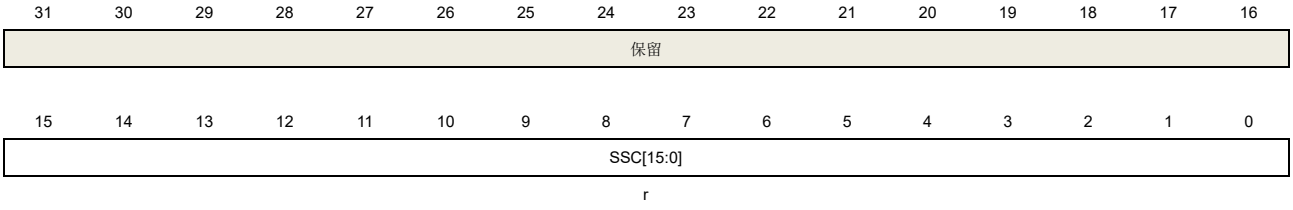


位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	WPK[7:0]	写保护的解锁值

23.4.10. 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)

23.4.11. 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。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
A1S	保留														
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	SFS[14:0]														
	w														

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。

23.4.12. 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。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留									PM	HRT[1:0]		HRU[3:0]			
									r	r		r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	MNT[2:0]			MNU[3:0]				保留	SCT[2:0]			SCU[3:0]			
	r			r					r			r			

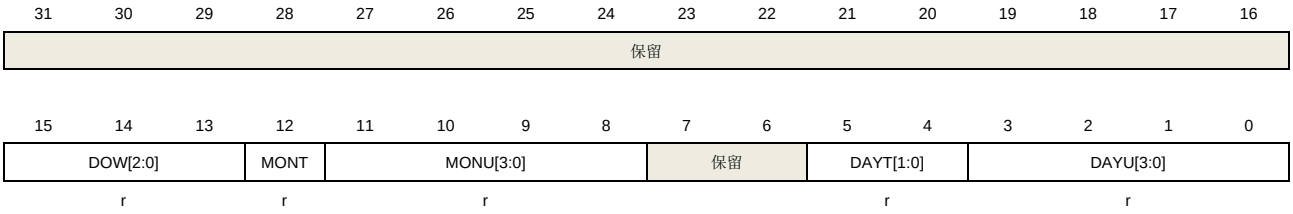
位/位域	名称	描述

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

23.4.13. 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。

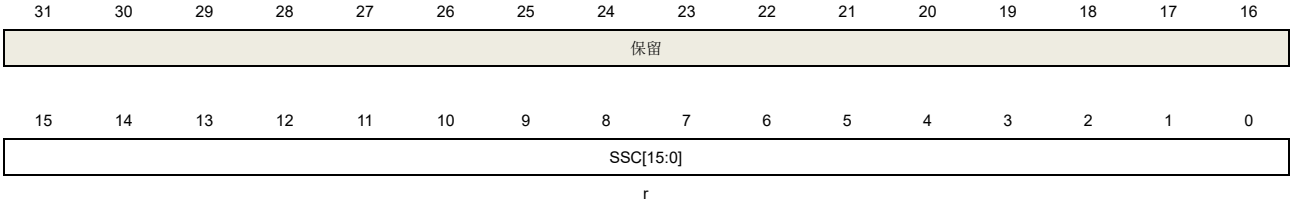


位/位域	名称	描述
31:16	保留	必须保持复位值。
15:13	DOW[2:0]	星期数
12	MONT	月份十位值，以 BCD 码形式存储
11:8	MONU[3:0]	月份个位值，以 BCD 码形式存储
7	保留	必须保持复位值。
6:5	DAYT[1:0]	日期十位值，以 BCD 码形式存储
4:0	DAYU[3:0]	日期个位值，以 BCD 码形式存储

23.4.14. 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。

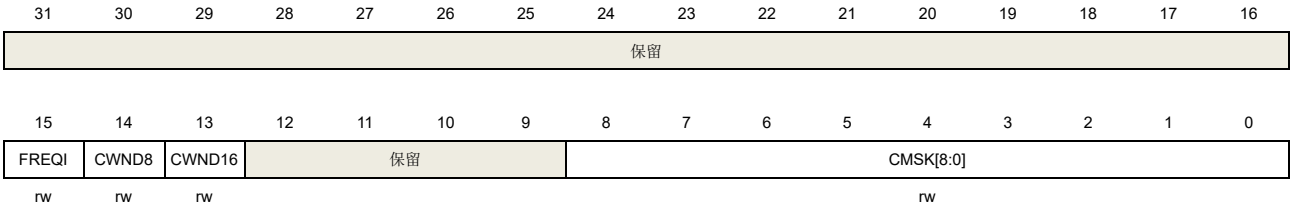


位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	SSC[15:0]	亚秒值 TSF 置 1 时记录当时的同步预分频计数器的值。

23.4.15. 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。



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

23.4.16. 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															

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]			TPTS	TP1EG	TP1EN	保留		TPIE	TP0EG	TP0EN
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	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	TP1EG	TAMP1 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	TP1EN	Tamper1 检测使能位 0：禁用 Tamper1 检测功能 1：启用 Tamper1 检测功能
4:3	保留	必须保持复位值。
2	TPIE	侵入检测中断使能 0：禁用侵入中断 1：启用侵入中断
1	TP0EG	TAMP0 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。

23.4.17. 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。

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

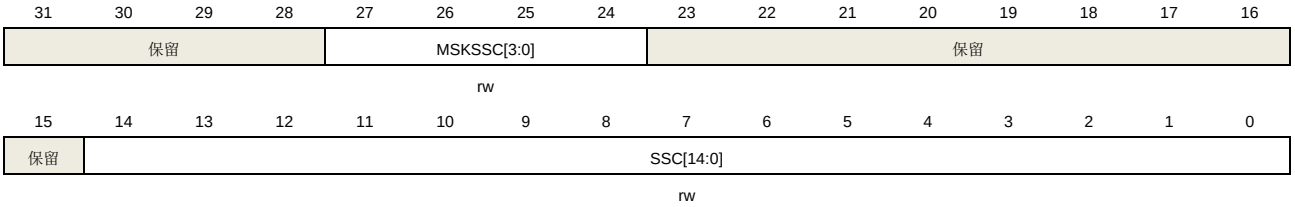
位/位域	名称	描述
31:28	保留	必须保持复位值。
27:24	MSKSSC[3:0]	亚秒位域的屏蔽控制位 0x0：屏蔽闹钟亚秒设置。当所有其他的闹钟位域匹配的时候，闹钟将会在每一秒钟到达的时刻置 1。 0x1：SSC[0]位用于时间匹配，其他位被忽略。 0x2：SSC[1：0]位用于时间匹配，其他位被忽略。 0x3：SSC[2：0]位用于时间匹配，其他位被忽略。 0x4：SSC[3：0]位用于时间匹配，其他位被忽略。 0x5：SSC[4：0]位用于时间匹配，其他位被忽略。 0x6：SSC[5：0]位用于时间匹配，其他位被忽略。 0x7：SSC[6：0]位用于时间匹配，其他位被忽略。 0x8：SSC[7：0]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。

23.4.18. 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。

23.4.19. 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

23.4.20. 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。

24. 定时器（TIMER）

表 24-1. 定时器（TIMERx）分为 5 种类型

定时器		定时器 0/7	定时器 1/2/3/4/22/23/30/31	定时器 14/40/41/42/43/44	定时器 15/16	定时器 5/6/50/51
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（TIMER2/3/30/31） 32 位（TIMER1/4/22/23）	16 位	16 位	32 位（TIMER5/6） 64 位（TIMER50/51）
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		●	●	×	×	×
非正交译码器		●	●	×	×	×
主-从管理		●	●	●	×	×
内部连接		● (1)	● (2)	● (3)	×	TRGO TO DAC
DMA		●	●	●	●	● (4)
Debug 模式		●	●	●	●	●

	TIMERx	ITI0	ITI1	ITI2	ITI3	ITI4	ITI5	ITI6	ITI7	ITI8	ITI9	ITI10	ITI11	ITI12	ITI13	ITI14
(1)	TIMER0	TIMER14_TRGO0	TIMER1_TRGO0	TIMER2_TRGO0	TIMER3_TRGO0	-	-	-	-	-	-	-	-	TIMER22_TRGO0	TIMER23_TRGO0	来自TRIG SEL
	TIMER7	TIMER0_TRGO0	TIMER1_TRGO0	TIMER3_TRGO0	TIMER4_TRGO0	-	-	-	-	-	-	-	-	TIMER22_TRGO0	TIMER23_TRGO0	

	TIMERx	ITI0	ITI1	ITI2	ITI3	ITI4	ITI5	ITI6	ITI7	ITI8	ITI9	ITI10	ITI11	ITI12	ITI13	ITI14
	TIMER1	TIMER0_TRGO0	TIMER7_TRGO0	TIMER2_TRGO0	TIMER3_TRGO0	ETH_PPS	USB0OTG_HS_SOF	-	-	-	-	-	-	TIMER22_TRGO0	TIMER23_TRGO0	
	TIMER2	TIMER0_TRGO0	TIMER1_TRGO0	TIMER14_TRGO0	TIMER3_TRGO0	ETH_PPS	-	-	-	-	-	-	-	TIMER22_TRGO0	TIMER23_TRGO0	
	TIMER3	TIMER0_TRGO0	TIMER1_TRGO0	TIMER2_TRGO0	TIMER7_TRGO0	-	-	-	-	-	-	-	-	TIMER22_TRGO0	TIMER23_TRGO0	
(2)	TIMER4	TIMER0_TRGO0	TIMER7_TRGO0	TIMER2_TRGO0	TIMER3_TRGO0	-	-		USB1OTG_HS_SOF	-	-	-	-	TIMER22_TRGO0	TIMER23_TRGO0	
	TIMER22	TIMER0_TRGO0	TIMER1_TRGO0	TIMER2_TRGO0	TIMER3_TRGO0	TIMER4_TRGO0	TIMER7_TRGO0	-	-	-	TIMER14_TRGO0	TIMER15_CH0	TIMER16_CH0	-	TIMER23_TRGO0	
	TIMER23	TIMER0_TRGO0	TIMER1_TRGO0	TIMER2_TRGO0	TIMER3_TRGO0	TIMER4_TRGO0	TIMER7_TRGO0	-	-	-	TIMER14_TRGO0	TIMER15_CH0	TIMER16_CH0	TIMER22_TRGO0	-	
	TIMER30	TIMER0_TRGO0	TIMER1_TRGO0	TIMER14_TRGO0	TIMER3_TRGO0	ETH1_PPS	USB1OTG_HS_SOF	-	-	-	-	-	-	TIMER22_TRGO0	TIMER23_TRGO0	
	TIMER31	TIMER0_TRGO0	TIMER1_TRGO0	TIMER14_TRGO0	TIMER3_TRGO0	ETH1_PPS	-	-	-	-	-	-	-	TIMER22_TRGO0	TIMER23_TRGO0	
	TIMER14	TIMER0_TRGO0	TIMER2_TRGO0	TIMER15_CH0	TIMER16_CH0	-	-	-	-	-	-	-	-	-	-	
	TIMER40	TIMER0_TRGO0	TIMER2_TRGO0	TIMER15_CH0	TIMER16_CH0	-	-	-	-	-	-	-	-	-	-	
	TIMER41	TIMER0_TRGO0	TIMER2_TRGO0	TIMER15_CH0	TIMER16_CH0	-	-	-	-	-	-	-	-	-	-	
(3)	TIMER42	TIMER0_TRGO0	TIMER2_TRGO0	TIMER15_CH0	TIMER16_CH0	-	-	-	-	-	-	-	-	-	-	
	TIMER43	TIMER0_TRGO0	TIMER2_TRGO0	TIMER15_CH0	TIMER16_CH0	-	-	-	-	-	-	-	-	-	-	
	TIMER44	TIMER0_TRGO0	TIMER2_TRGO0	TIMER15_CH0	TIMER16_CH0	-	-	-	-	-	-	-	-	-	-	
(4)	只有更新事件可以产生DMA请求。但是定时器5/6/50/51中没有DMA配置DMAS。															

24.1. 高级定时器（TIMERx, x=0, 7）

24.1.1. 简介

高级定时器（TIMER0/7）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。

24.1.2. 主要特性

- 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。

24.1.3. 结构框图

*图 24-1. 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 24-2. 高级定时器通道介绍*介绍了通道输入和输出情况。

图 24-1. 高级定时器结构框图

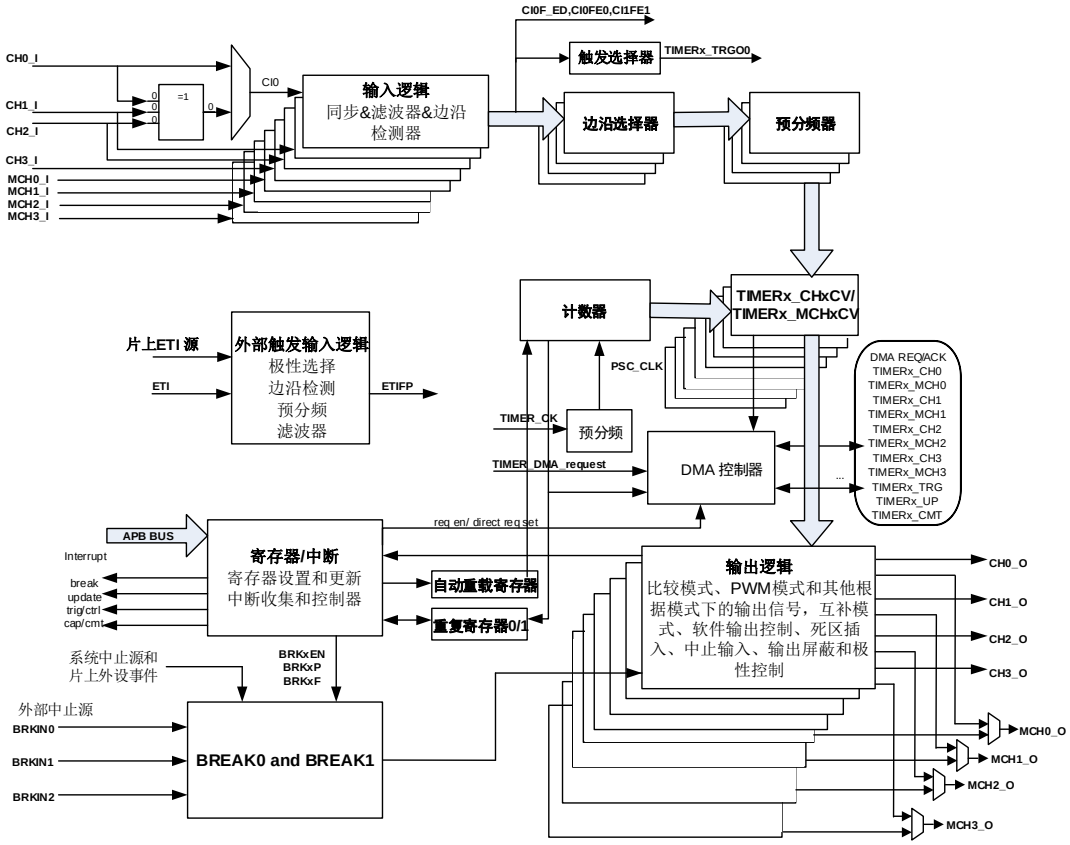


表 24-2. 高级定时器通道介绍

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）		

24.1.4. 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)寄存器中的 TSCFGy[4:0] (y=0..9,15)位域控制的复用时钟源驱动。

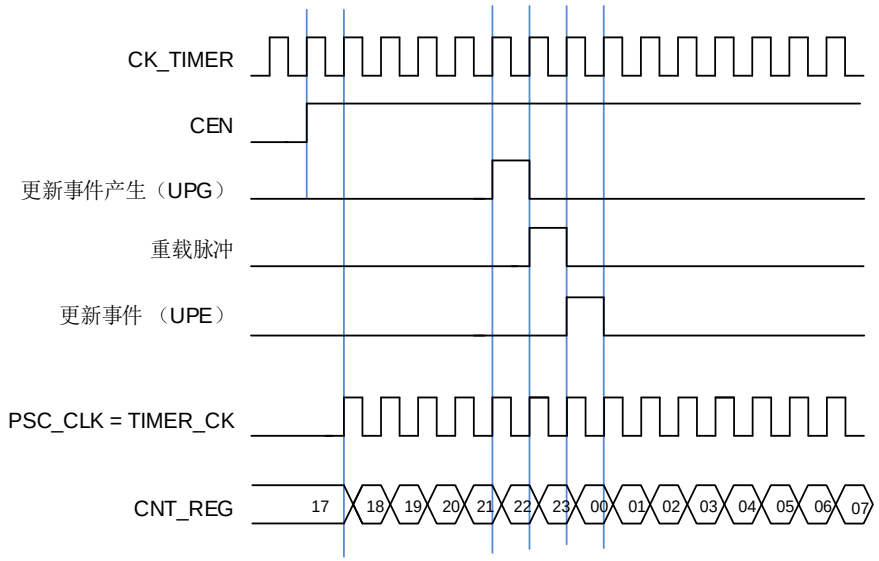
- 当SYSCFG_TIMERxCFG(x=0,7)寄存器中的TSCFGy[4:0]=5'b00000(y=0..9,15)时，定时器选择内部时钟源（连接到RCU模块的CK_TIMER）

如果 SYSCFG_TIMERxCFG(x=0,7)寄存器中的 TSCFGy[4:0] =5'b00000(y=0..9,15)，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)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 24-2. 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 TIMERx_CHn/ TIMERx_MCHn（n=0..3）引脚的每个上升沿或下降沿计数。这种模式可以通过设置 TSCFG6[4:0]为 0x5~0x7 和 0x9~0xE 来选择。

计数器预分频器也可以在内部触发信号 ITI0/1/2/3/12/13/14 的上升沿计数。这种模式可以通过设置 TSCFG6[4:0]为 0x1~0x4，0x11~0x13 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

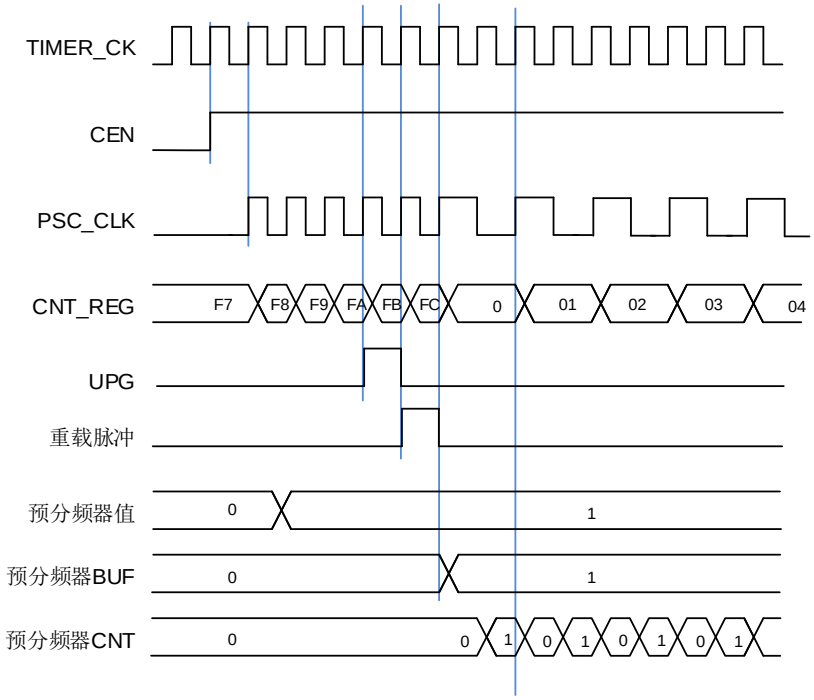
计数器预分频器可以在外部引脚 ETI 的每个上升沿或下降沿计数。这种模式可以通过设置 TIMERx_SMCFG 寄存器中的 SMC1 位为 1 来选择。另一种选择 ETI 信号作为时钟源方式是，设置 TSCFG6[4:0]为 0x8。注意 ETI 信号是通过数字滤波器采样 ETI 引脚得到的。如果选择 ETI 信号为时钟源，触发控制器包括边沿监测电路将在每个 ETI 信号上升沿产生一个时钟脉冲来为计数器预分频器提供时钟。

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）*** 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 24-3. 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24-4. 向上计数时序图,PSC=0/2 和 *图24-5. 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 24-4. 向上计数时序图，PSC=0/2

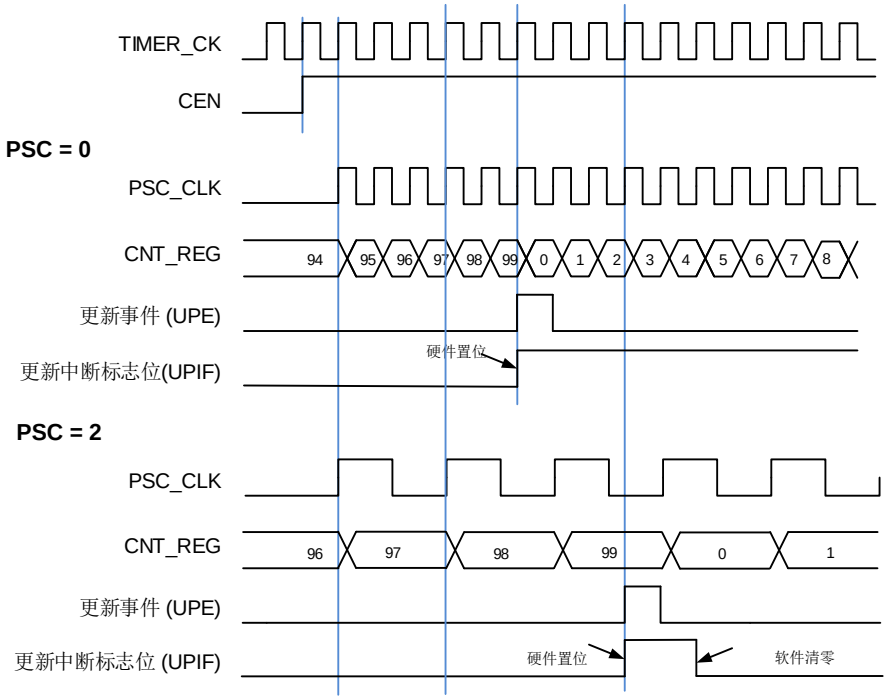
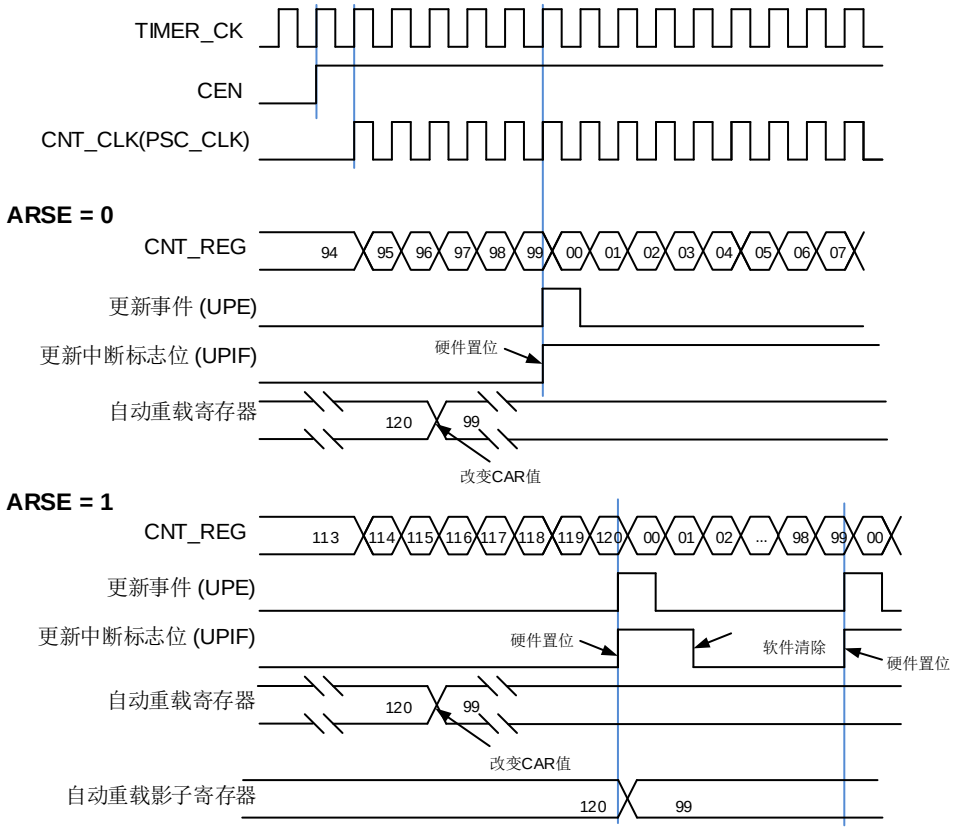


图 24-5. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



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24-6. 向下计数时序图,PSC=0/2 和 *图24-7. 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 24-6. 向下计数时序图，PSC=0/2

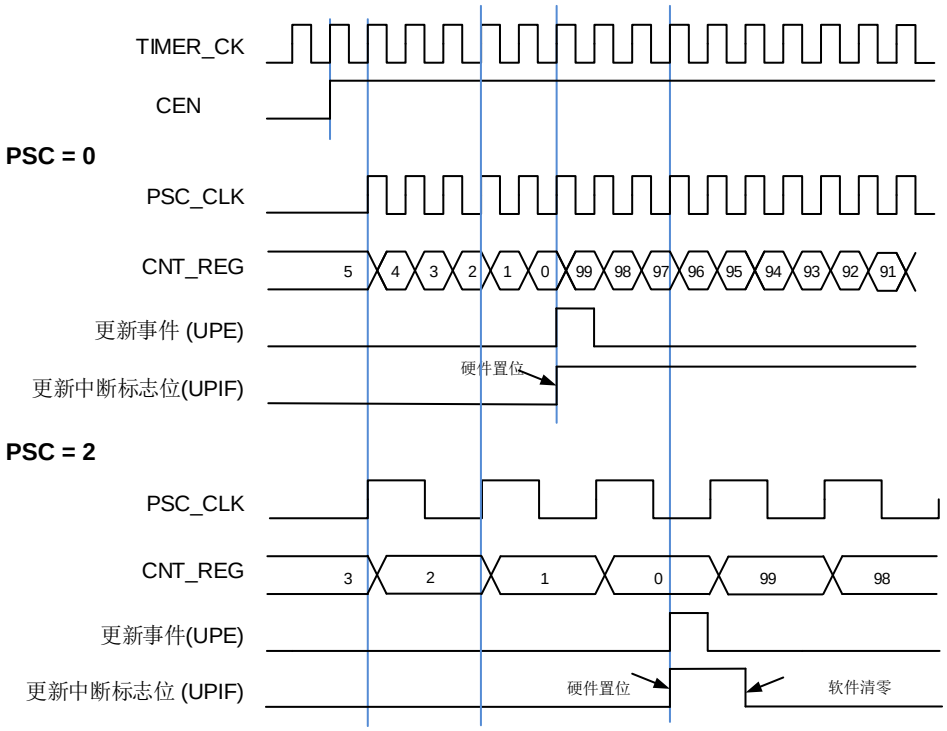
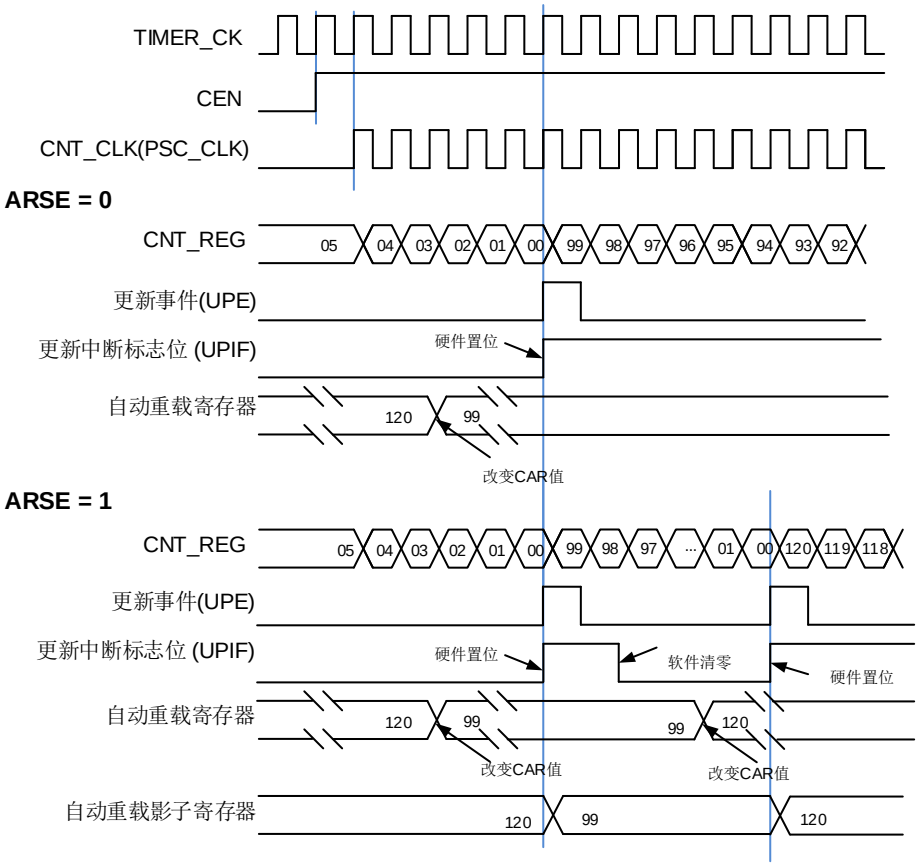


图 24-7. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值



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。

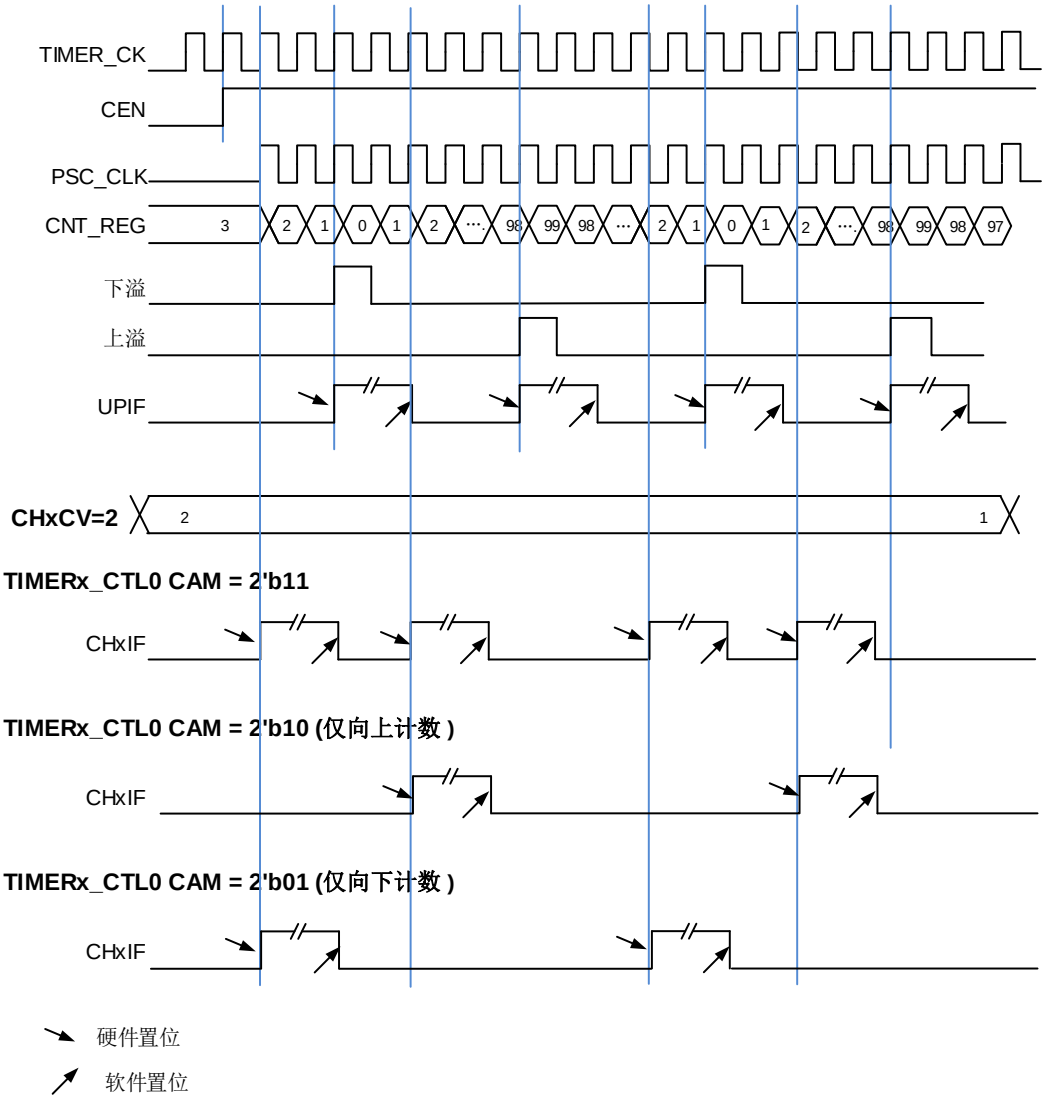
上溢或者下溢时，TIMERx_INTF 寄存器中的 UPIF 位都会被置 1，然而 CHxIF 位置 1 与 TIMERx_CTL0 寄存器中 CAM 的值有关。具体细节参考*图24-8. 中央计数模式计数器时序图*。

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24-8. 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 24-8. 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 24-9. 中央计数模式下计数器重复时序图

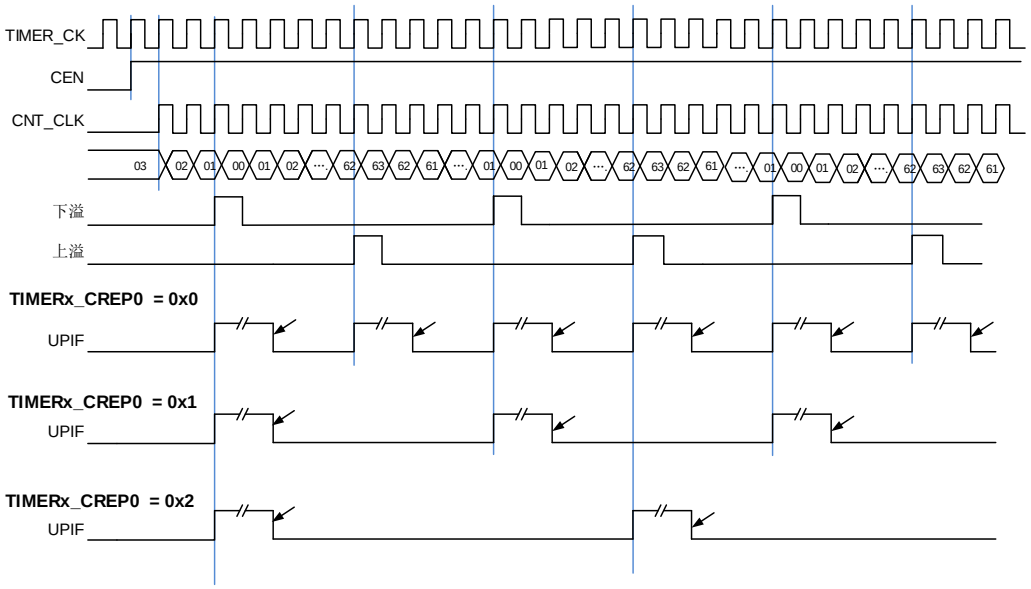


图 24-10. 在向上计数模式下计数器重复时序图

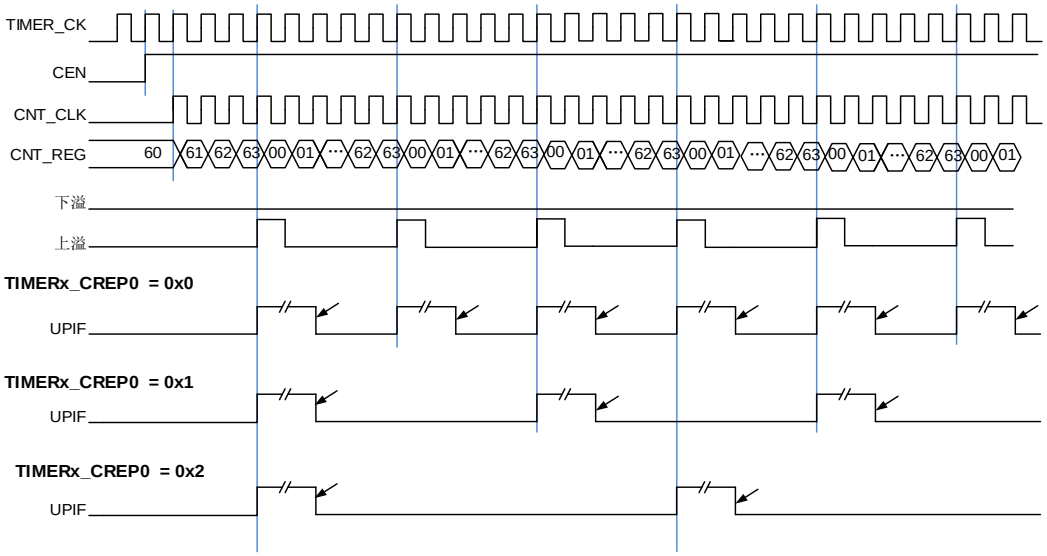
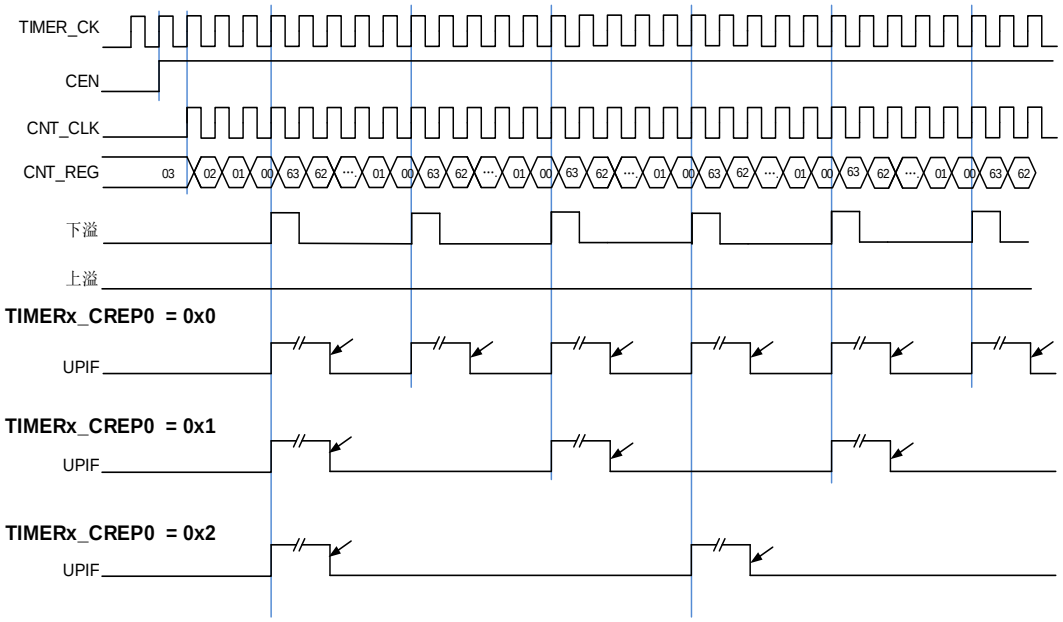


图 24-11. 在向下计数模式下计数器重复时序图



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 24-12. 通道 0 输入捕获逻辑

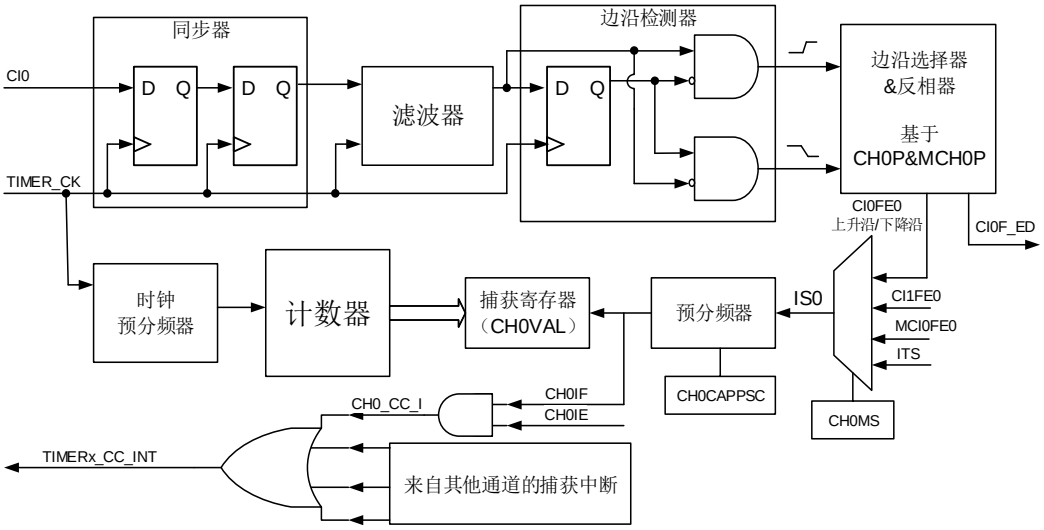
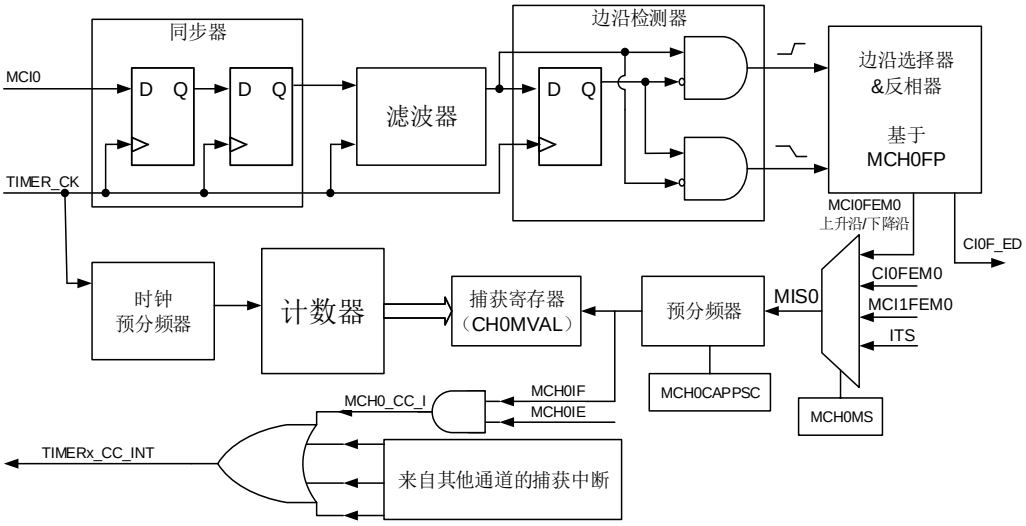


图 24-13. 多模式通道 0 输入捕获逻辑



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 24-14. 输出比较逻辑（当 MCHxMSEL = 2’00 时，x=0,1,2,3）*和*图 24-15. 输出比较逻辑（当 MCHxMSEL = 2’11 时，x=0,1,2,3）*给出了通道的输出比较逻辑。

图 24-14. 输出比较逻辑（当 MCHxMSEL = 2’00 时，x=0,1,2,3）

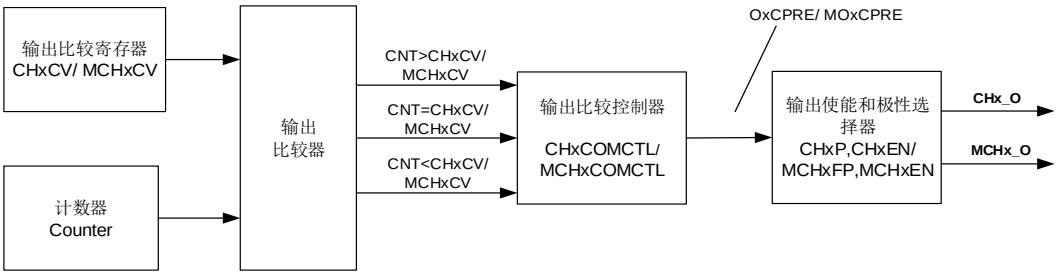
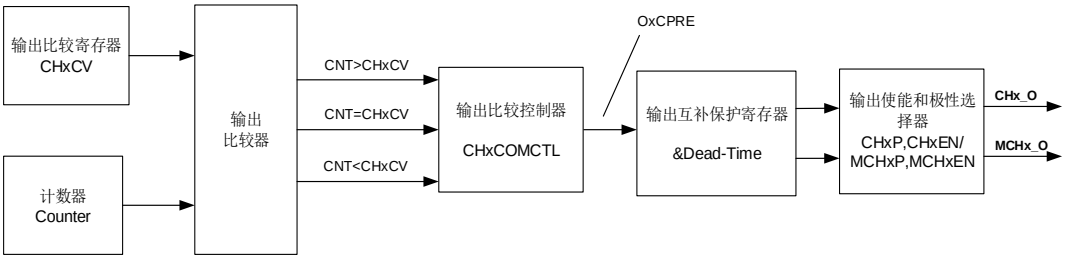


图 24-15. 输出比较逻辑（当 MCHxMSEL = 2’11 时，x=0,1,2,3）



通道输出信号CHx_O/MCHx_O与OxCPRE/ MOxCPRE信号（详情请见*通道输出参考信号*）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24-14. 输出比较逻辑（当MCHxMSEL = 2'00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24-15. 输出比较逻辑（当MCHxMSEL = 2'11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_CCHP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：

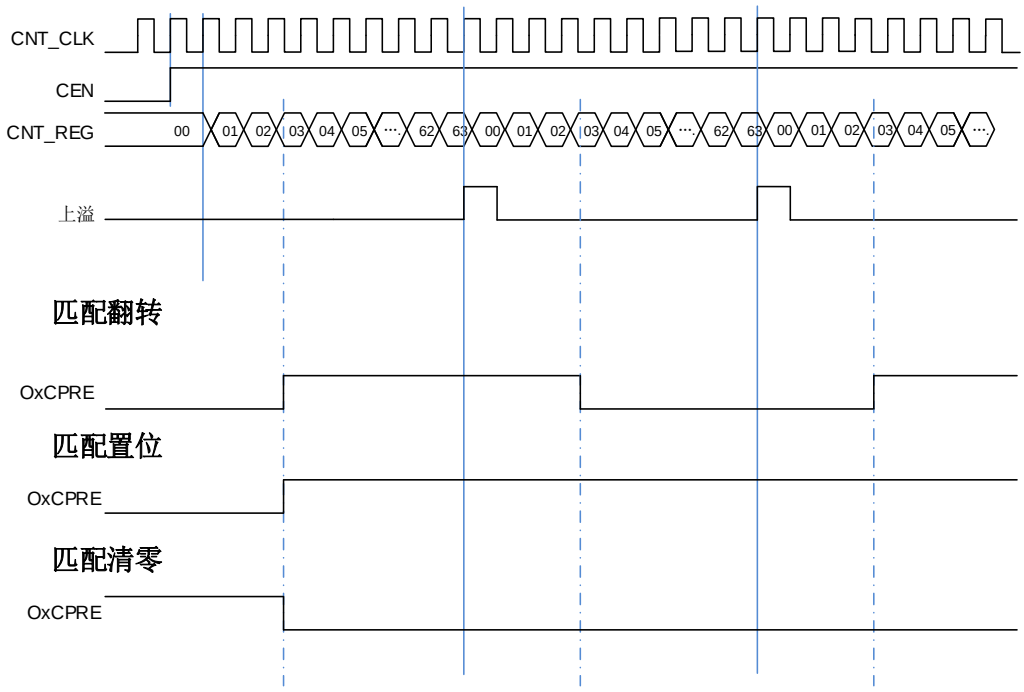
TIMERx_CHxCV/ TIMERx_MCHxCV可以在运行时根据你所期望的波形而改变。

第五步：设置CEN位使能定时器。

*图24-16. 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 24-16. 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24-17. 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24-18. 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 24-17. EAPWM 时序图

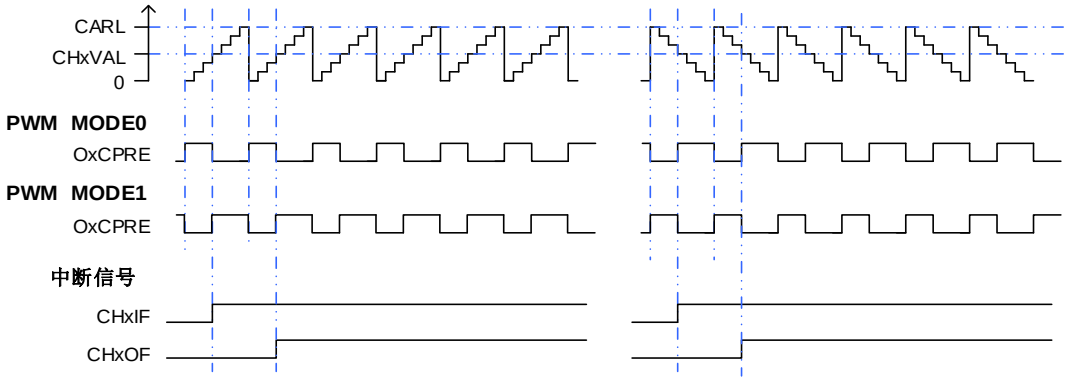
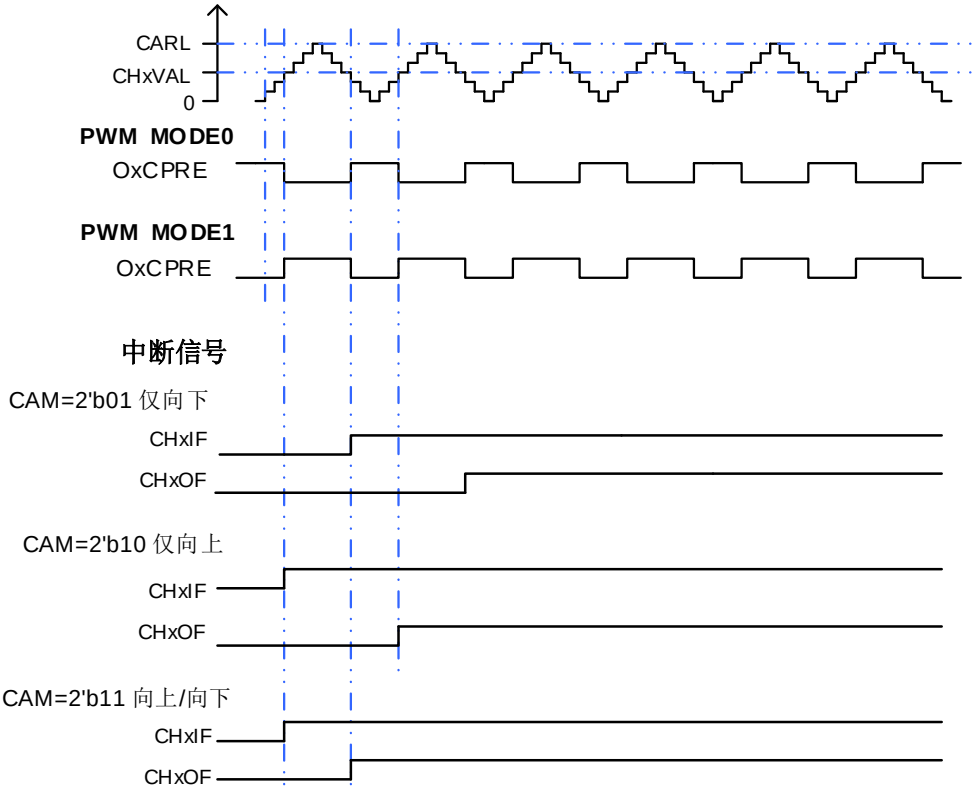


图 24-18. CAPWM 时序图



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 24-3 复合 PWM 脉冲宽度*计算。

表 24-3 复合 PWM 脉冲宽度

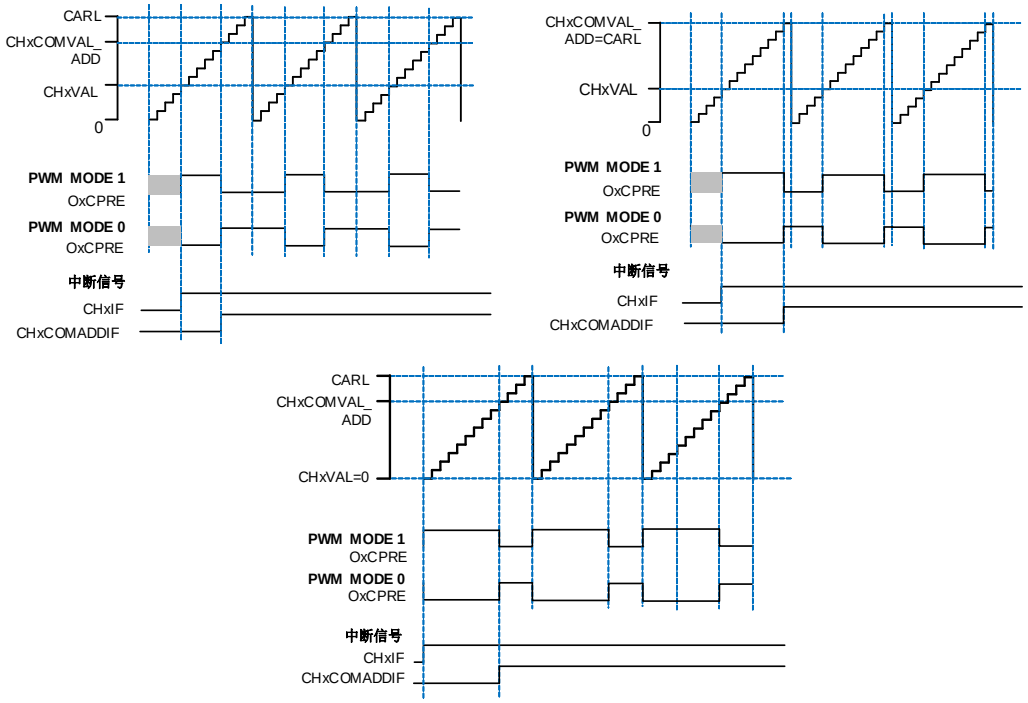
条件	模式	PWM 脉冲宽度
CHxVAL < CHxCOMVAL_ADD ≤ CARL	PWM 模式 0	（CARL + 0x0001） + （CHxVAL – CHxCOMVAL_ADD）
	PWM 模式 1	（CHxCOMVAL_ADD – CHxVAL）
CHxCOMVAL_ADD < CHxVAL ≤ CARL	PWM 模式 0	（CHxVAL - CHxCOMVAL_ADD）
	PWM 模式 1	（CARL + 0x0001） + （CHxCOMVAL_ADD – CHxVAL）
（CHxVAL = CHxCOMVAL_ADD ≤ CARL）或（CHxVAL > CARL > CHxCOMVAL_ADD）	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：

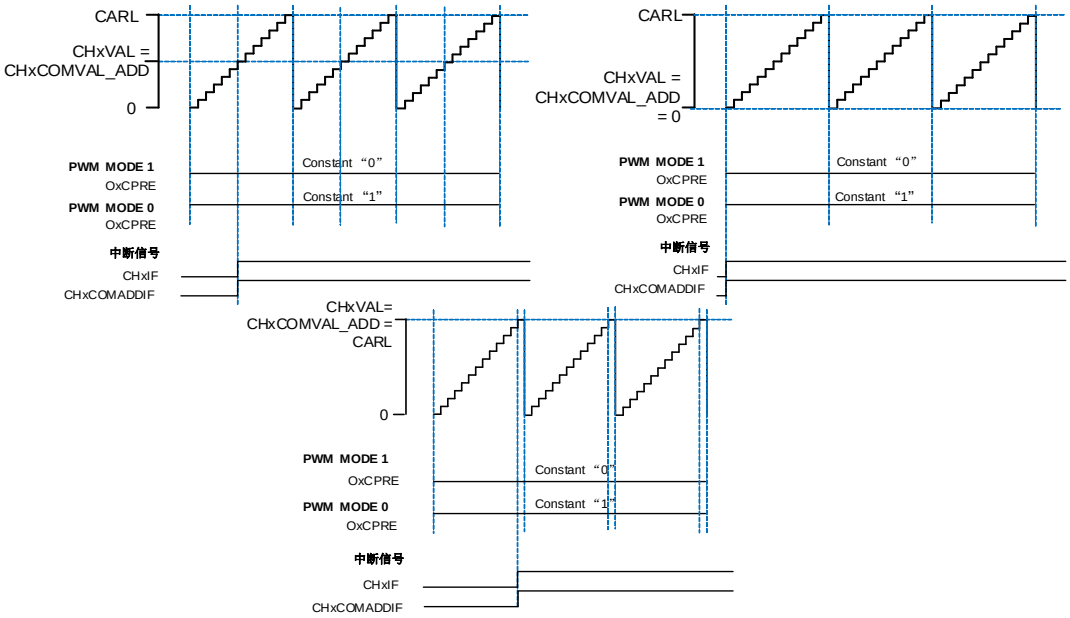
1） CHxVAL < CHxCOMVAL_ADD，CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 24-19. 通道 x 输出 PWM（CHxVAL < CHxCOMVAL_ADD）



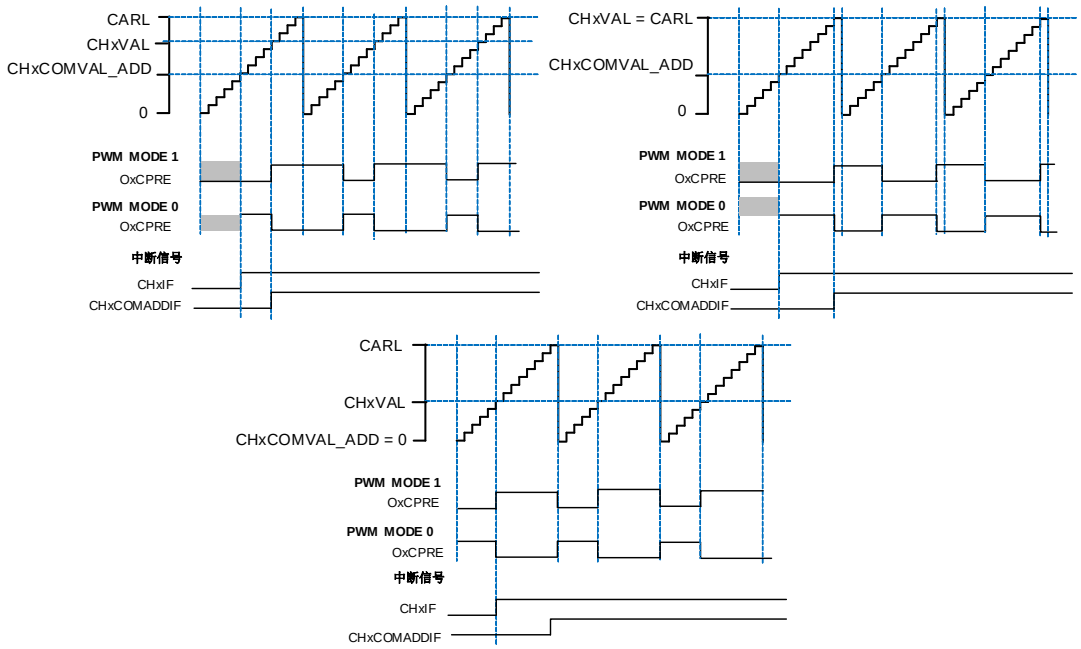
2） CHxVAL = CHxCOMVAL_ADD, CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 24-20. 通道 x 输出 PWM（CHxVAL = CHxCOMVAL_ADD）



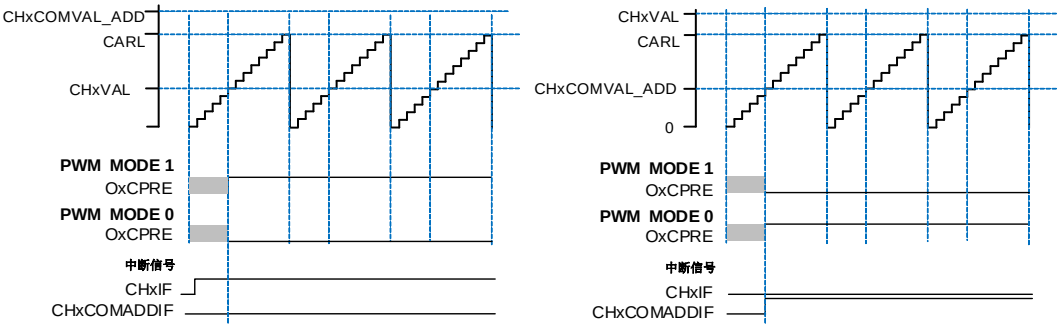
3） CHxVAL > CHxCOMVAL_ADD, CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 24-21. 通道 x 输出 PWM（CHxVAL > CHxCOMVAL_ADD）



4） CHxVAL或CHxCOMVAL_ADD值大于CARL。

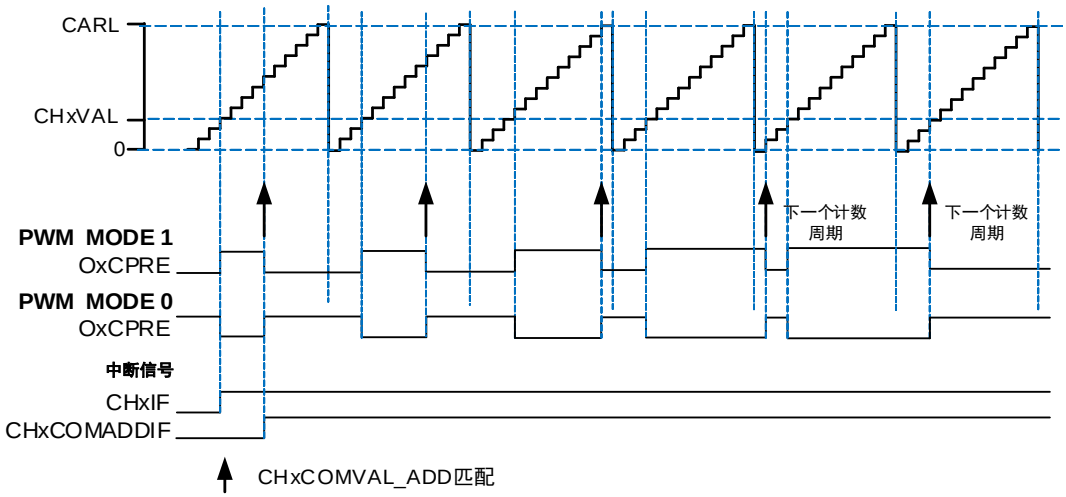
图 24-22. 通道 x 输出 PWM（CHxVAL 或 CHxCOMVAL_ADD > CARL）



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24-23. 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。

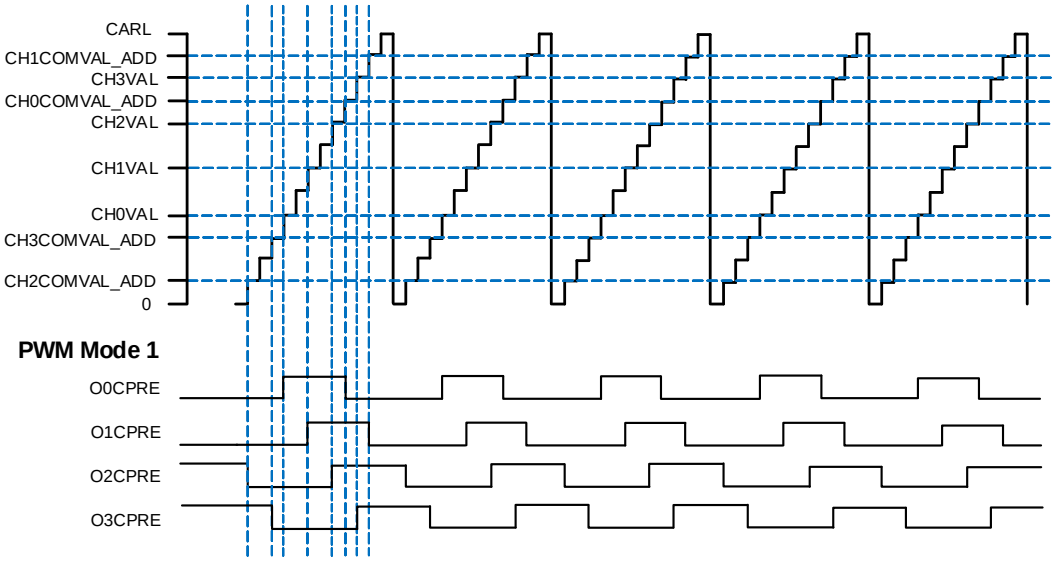
在某些情况下，CHxCOMVAL_ADD的匹配事件可以发生在下一个计数周期（CHxCOMVAL_ADD值在计数器到达CHxVAL值之后被写入，且CHxCOMVAL_ADD值小于或者等于CHxVAL值）。

图 24-23. 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 24-24. 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 24-25. 边沿对齐模式下 CHx_O 输出脉冲（CHxOMPSEL≠2’b00）

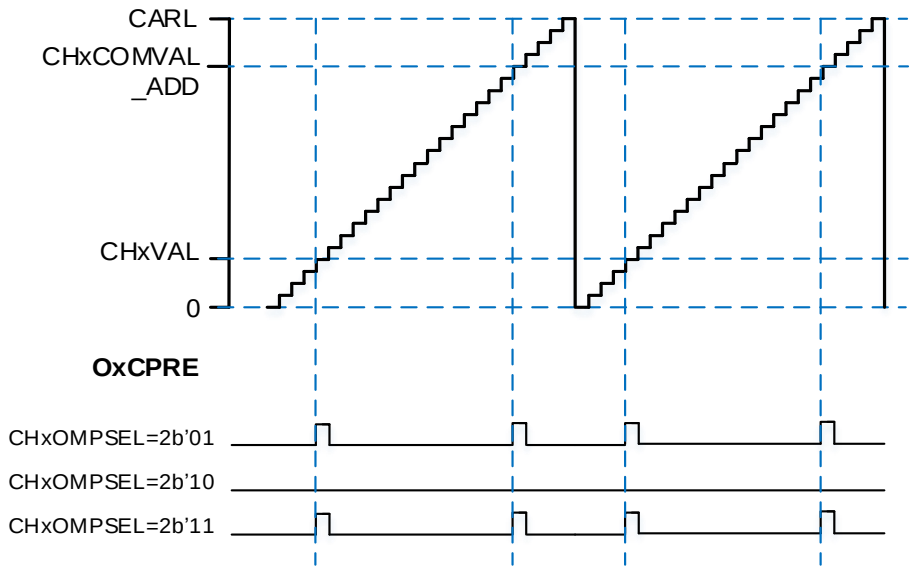
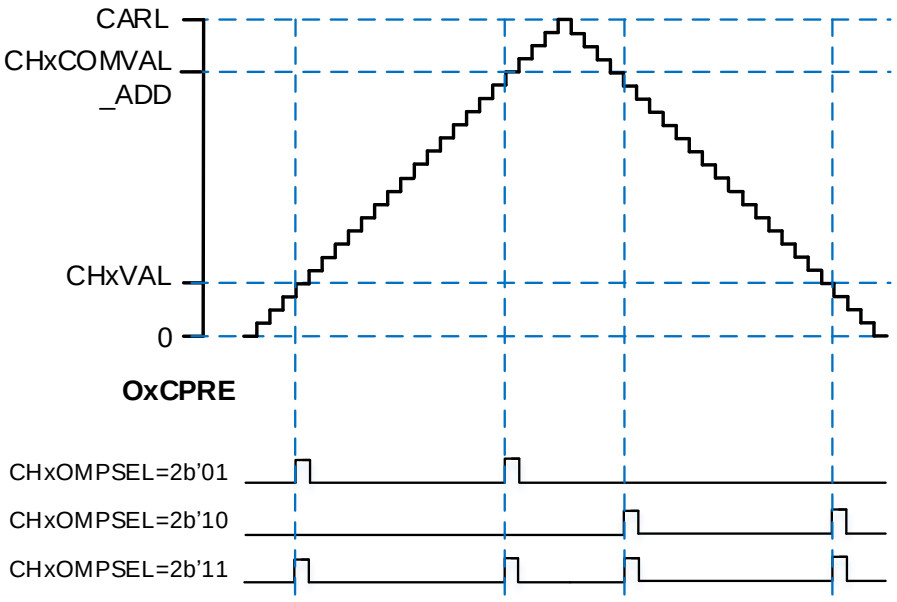


图 24-26. 中央对齐模式下 CHx_O 输出脉冲（CHxOMPSEL≠2’b00）



通道输出参考信号

如*图 24-14. 输出比较逻辑（当 MCHxMSEL = 2’00 时，x=0,1,2,3）*和*图 24-15. 输出比较逻辑（当 MCHxMSEL = 2’11 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。

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_CCHP 寄存器中和 TIMERx_CTL1 寄存器中的 POEN、ROS、IOS、ISOx 和 ISOxN 位（当 CHx_O 和 MCHx_O 具有独立的死区时间和中止功能时，请参考***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 24-4. 由参数控制的互补输出表（MCHxMSEL=2'b11）***。

表 24-4. 由参数控制的互补输出表（MCHxMSEL=2'b11）

互补参数					输出状态	
POEN	ROS	IOS	CHxEN	MCHxEN	CHx_O	MCHx_O
0	0/1	0	0	0	CHx_O / CHx_ON = LOW CHx_O / CHx_ON 输出禁能(1)	
				1	CHx_O/CHx_ON输出关闭状态(2)： 通道先输出无效电平：CHx_O = CHxP，CHx_ON = CHxNP）；如果死区产生时钟未失效，在死区时间之后： CHx_O = ISOx，CHx_ON = ISOxN (3)	
			1	0		
				1		
		1	x	x	CHx_O/CHx_ON输出关闭状态： 通道先输出无效电平：CHx_O = CHxP，CHx_ON = CHxNP）；如果死区产生时钟未失效，在死区时间之后： CHx_O = ISOx，CHx_ON = ISOxN	
1	0	0/1	0	0	CHx_O/MCHx_O = LOW CHx_O/MCHx_O输出禁能	
				1	CHx_O = LOW	MCHx_O=OxCPRE ⊕

互补参数					输出状态	
POEN	ROS	IOS	CHxEN	MCHxEN	CHx_O	MCHx_O
					CHx_O输出禁能	(4)MCHxP MCHx_O输出使能
			1	0	CHx_O=OxCPRE⊕CHxP CHx_O输出使能	MCHx_O = LOW MCHx_O输出禁能
				1	CHx_O=OxCPRE⊕CHxP CHx_O输出使能	MCHx_O=(!OxCPRE)(5)⊕ 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 / CHx_ON 输出与对应引脚断开，对应引脚电平受 GPIO 上下拉配置控制，无上下拉时为悬空高阻态；

（2） 输出关闭状态：CHx_O / CHx_ON 输出无效电平（CHx_O = 0⊕CHxP = CHxP）；

（3） 详情见中止模式章节。

（4） ⊕：异或操作；

（5） (!OxCPRE)：OxCPRE 信号的互补信号。

死区时间插入

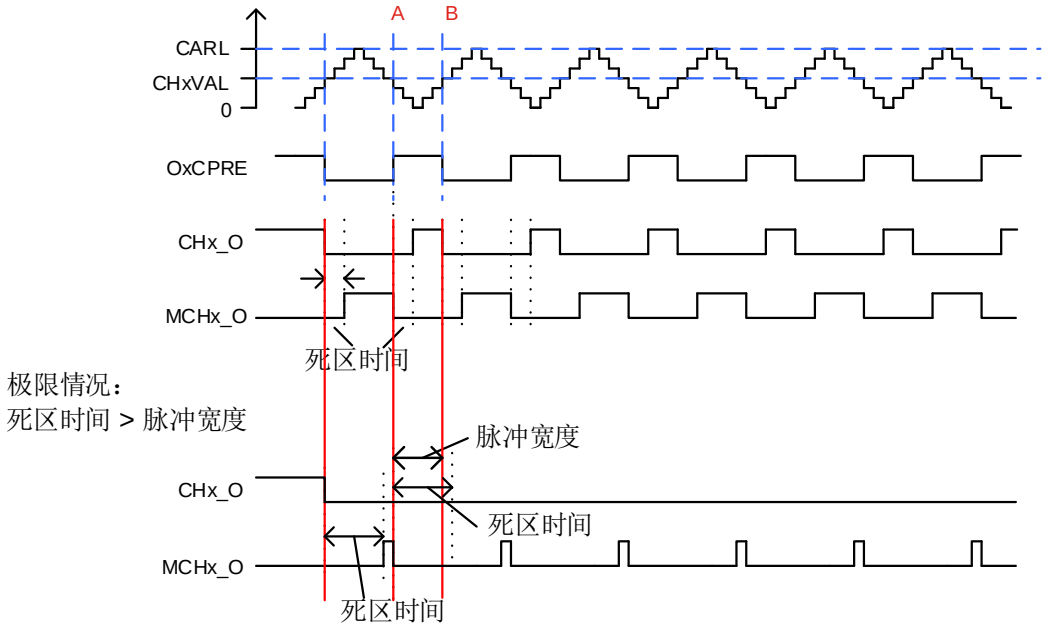
设置 MCHxMSEL=2'b11，CHxEN 和 MCHxEN 为 1'b1，同时设置 POEN=1，就可以使能死区插入功能。DTCFG 位域定义了死区时间，死区时间对所有通道有效。死区时间设置的细节请参考*互补通道保护寄存器（TIMERx_CCHP）*。

死区时间的插入，确保了通道互补的两路信号不会同时有效。

在 PWM0 模式，当通道 x 匹配发生时（TIMERx 计数器=TIMERx_CHxCV），OxCPRE 翻转。在*图 24-27. 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24-27. 带死区时间的互补输出*所示。

图 24-27. 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。

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_CCHP 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_CCHP 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_CCHP 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®-M7锁定事件或SRAM奇偶校验错误事件；
- 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 24-28. BREAK0 中止功能逻辑图***和***图 24-29. 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）***和***TIMER7_BRKIN 触发选择***

寄存器（TRIGSEL_TIMER7BRKIN）。

图 24-28. BREAK0 中止功能逻辑图

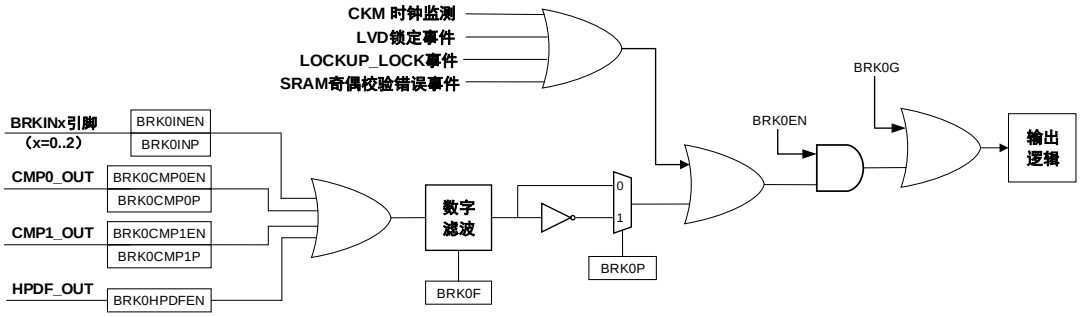
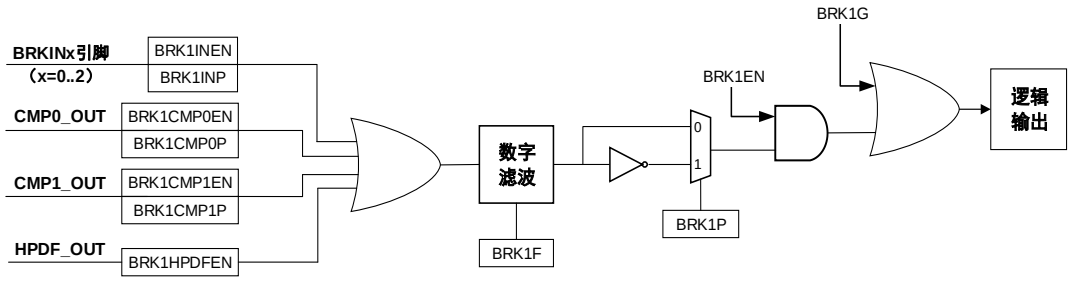


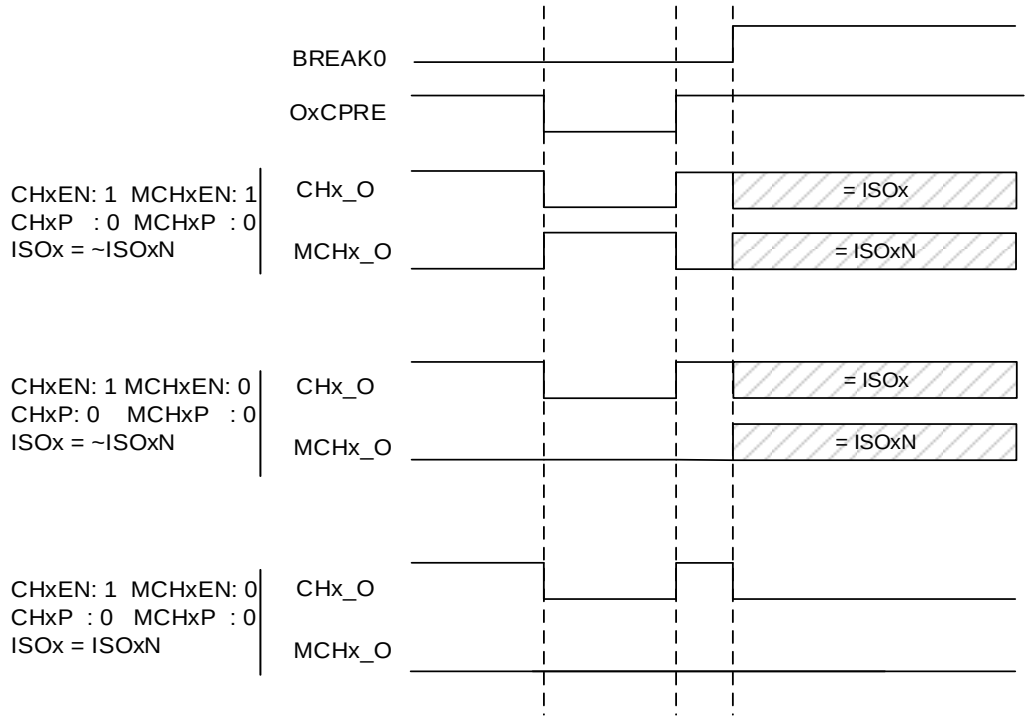
图 24-29. BREAK1 中止功能逻辑图



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 24-30. 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 24-30. 通道响应 BREAK0 中止输入（高电平有效）时，输出信号的行为（IOS=1）



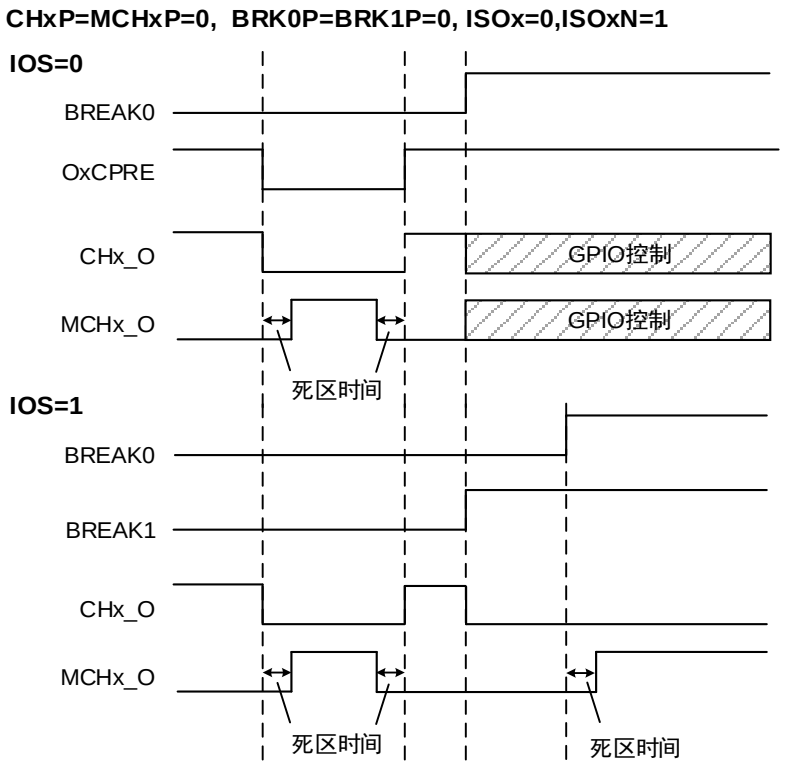
BREAK0的优先级高于BREAK1。只有在IOS=1和ROS=1时，才能使用BREAK1功能。

表 24-5. 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	Output Status	
		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 24-31. 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 BEKENCHx（x = 0..3）位，可实现对每对通道的中止功能进行独立控制。当 BEKENCHx（x = 0..3）位为“0”且发生中止事件时，相应的通道 CHx_O 和 MCHx_O 输出保持不变。

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_CCHP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_CCHP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_CCHP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_CCHP 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 24-6. 中止功能锁存/释放条件

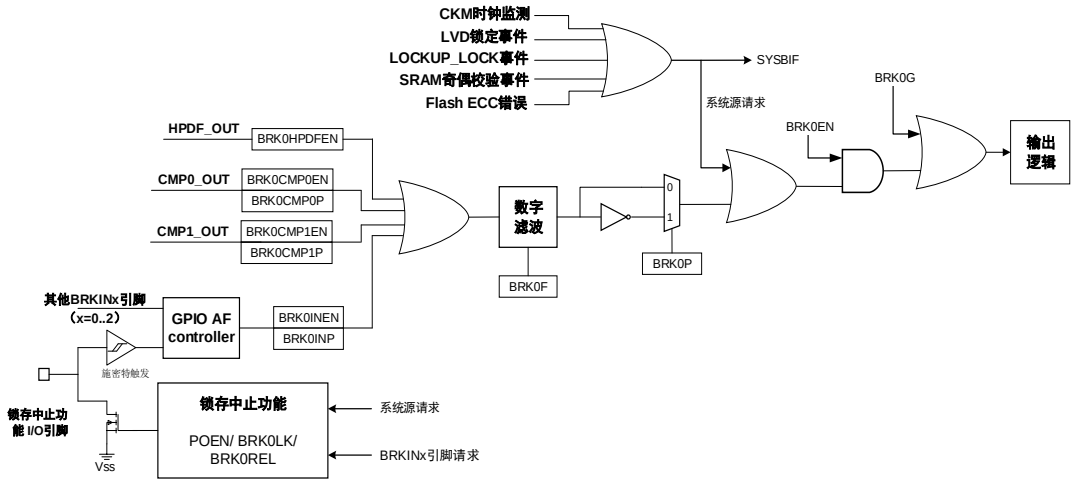
POEN	BRK0LK/ BRK1LK	BRK0REL/ BRK1REL	中止输入引脚状态
0	1	0	锁存
	1	1	释放

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 24-32. 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_CCHP寄存器中的ROS、IOS和DTCFG[7:0]有效；当FCCHPyEN=1时，TIMERx_FCCHPy寄存器中的ROS、IOS和DTCFG[7:0]有效，使能独立的死区时间插入和中止功能。

正交译码器

正交译码器功能使用由TIMERx_CH0和TIMERx_CH1引脚生成的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，或着可以同时有CI0和CI1，通过设置TSCFGy[4:0](y=0..2) != 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24-7. 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 24-7. 不同译码器模式下的计数方向

计数模式	电平	CI0FE0		CI1FE1	
		上升	下降	上升	下降
正交译码器模式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

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 24-33. 译码器接口模式下计数器运行例子

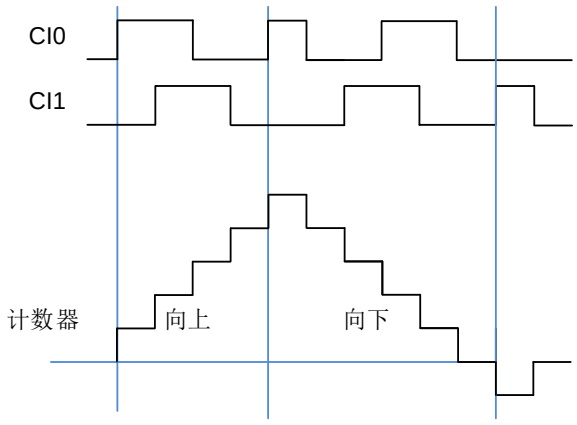
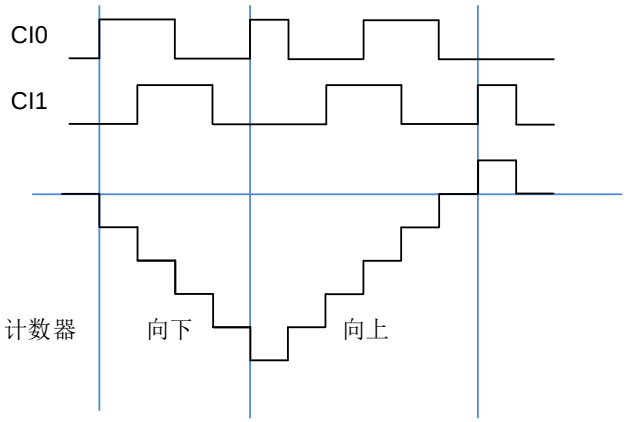


图 24-34. CI0FE0 极性反相的译码器接口模式下的例子



正交译码器信号检测

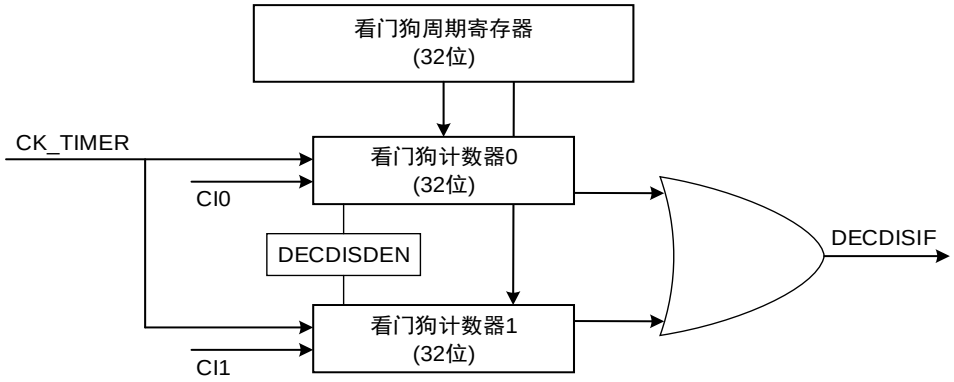
支持两种正交译码器信号检测：信号跳变检测和断线检测。

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24-35. 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 24-35. 正交译码器信号断线检测框图

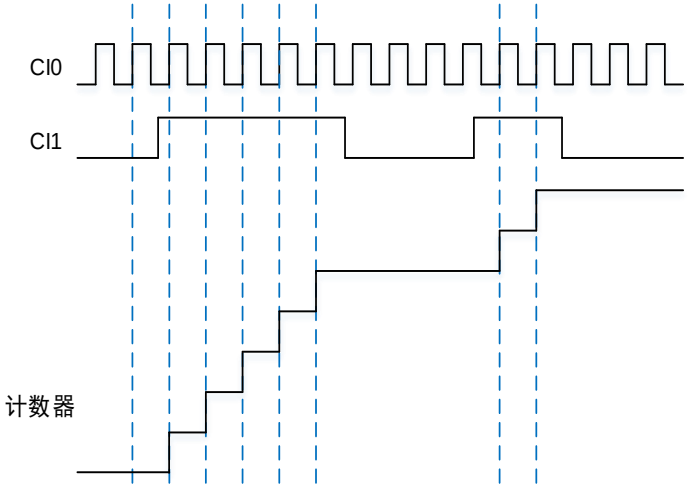


非正交译码

非正交译码器功能有两种模式：非正交译码器模式0和非正交译码器模式1，通过设置TSCFGy[4:0](y=8,9) != 5'b00000来选择。这两种计数模式下的输入源有两个：CI0和CI1。

使用非正交译码器模式0时，CI0作为计数脉冲，CI1作为计数选择信号。CH1P=0时，只有当CI1输入信号为高电平时，计数器才会在CI0输入信号的上升沿向上计数；CH1P=1时，只有当CI1输入信号为低电平时，计数器才会在CI0输入信号的上升沿向上计数。具体细节可见*图24-36. 非正交译码器模式0计数器运行实例（CH1P=0）*.

图 24-36. 非正交译码器模式 0 计数器运行实例（CH1P=0）

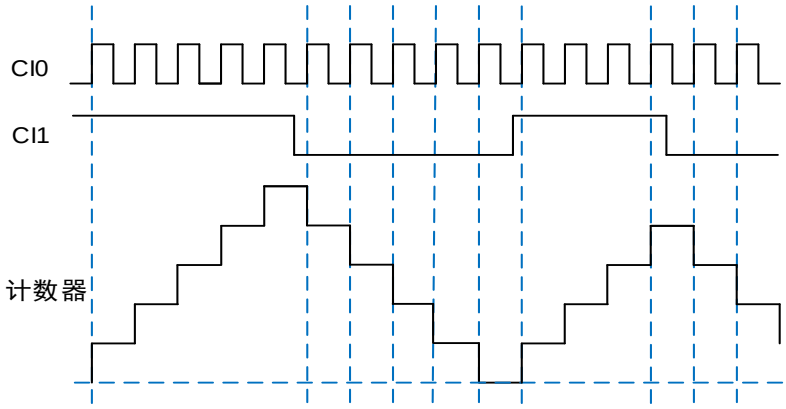


使用非正交译码器模式1时，CI0作为计数脉冲（CH0P用于选择计数边沿）；CI1作为计数方向选择信号。具体计数情况请见*表24-8. 非正交译码器模式1的计数情况*和*图24-37. 非正交译码器模式1计数器运行实例（CH0P=0）*。

表 24-8. 非正交译码器模式 1 的计数情况

CH0P	CI1电平	计数器计数情况
0	CI1为高电平	计数器在CI0输入信号的上升沿向上计数
	CI1为低电平	计数器在CI0输入信号的上升沿向下计数
1	CI1为高电平	计数器在CI0输入信号的下降沿向上计数
	CI1为低电平	计数器在CI0输入信号的下降沿向下计数

图 24-37. 非正交译码器模式 1 计数器运行实例（CH0P=0）



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 24-38. 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 24-38. 霍尔传感器用在 BLDC 电机控制中

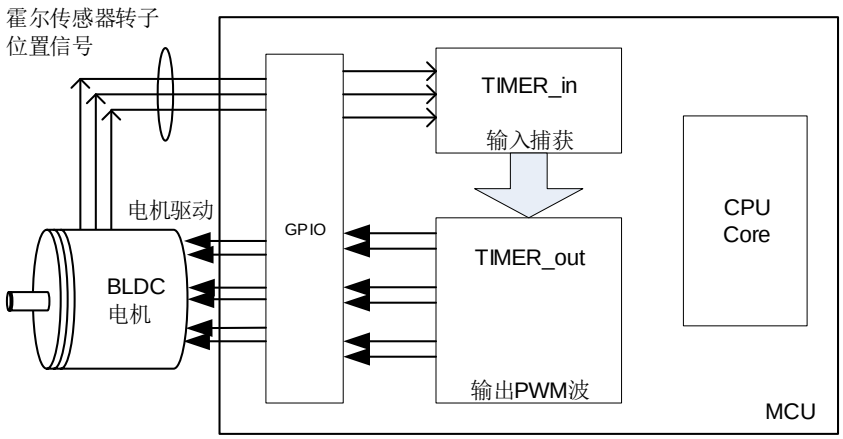
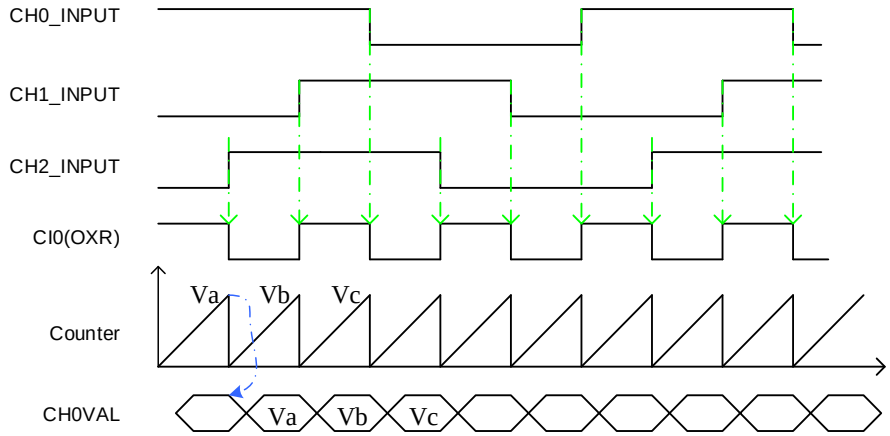
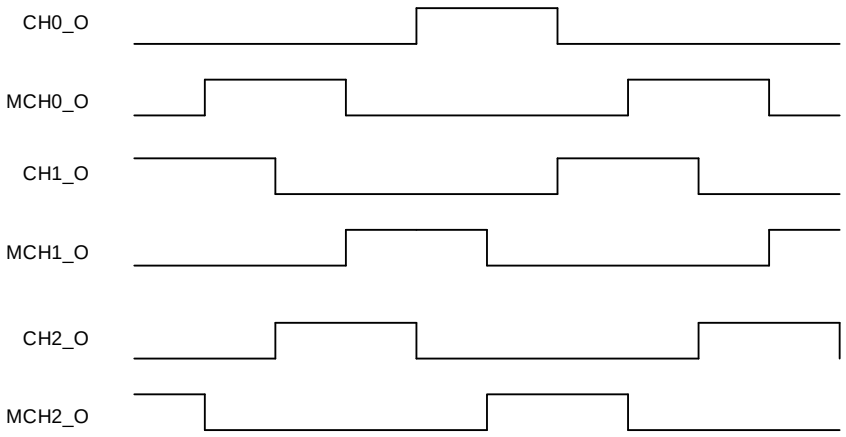


图 24-39. 两个定时器之间的霍尔传感器时序图

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)

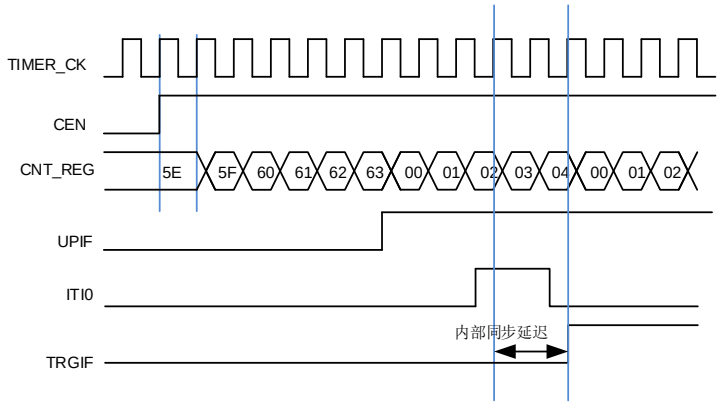
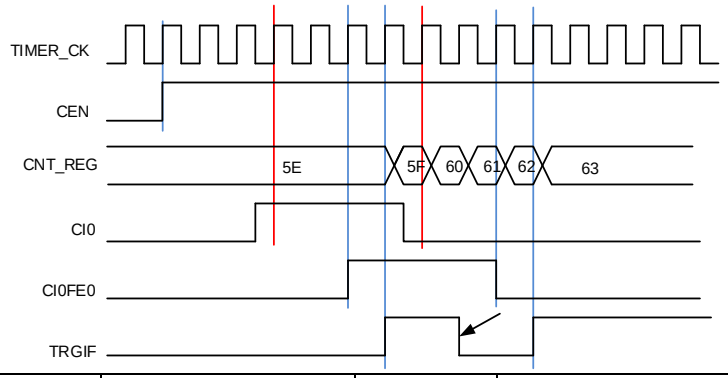


主-从管理

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)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。

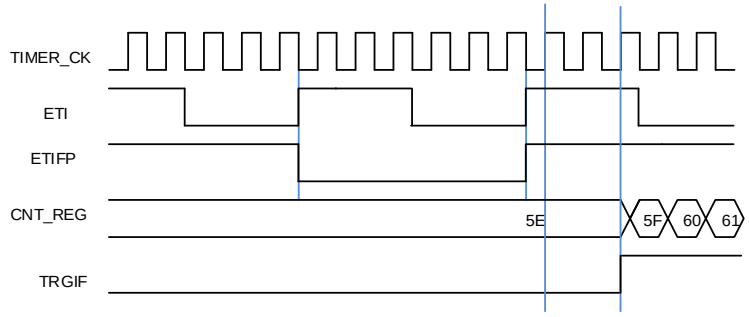
表 24-9. 从模式例子列表

	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 y=7：复位+事件模式	TSCFGy[4:0] 00000：Mode disable 00001：ITI0 00010：ITI1 00011：ITI2 00100：ITI3 00101：CI0F_ED 00110：CI0FE0 00111：CI1FE1	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	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 触发源是ETIFP，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></td><td>01000：ETIFP(1)
01001：CI2FE2
01010：CI3FE3
01011：MCI0FEM0
01100：MCI1FEM1
01101：MCI2FEM2
01110：MCI3FEM3
10001：ITI12
10010：ITI13
10011：ITI14</td><td>极性和反相</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 24-40. 复位模式
 
</td></tr>
<tr><td rowspan="2">例2</td><td>暂停模式
当触发输入为低的时候，计数器暂停计数</td><td>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>
<tr><td colspan="4">图 24-41. 暂停模式下的控制电路
 
</td></tr>
<tr><td>例3</td><td>事件模式
触发输入的上升沿计数器开始计数</td><td>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>
</table>

	模式选择	触发源选择	极性选择	滤波和预分频

图 24-42. 事件模式



(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）*** 模块。

单脉冲模式

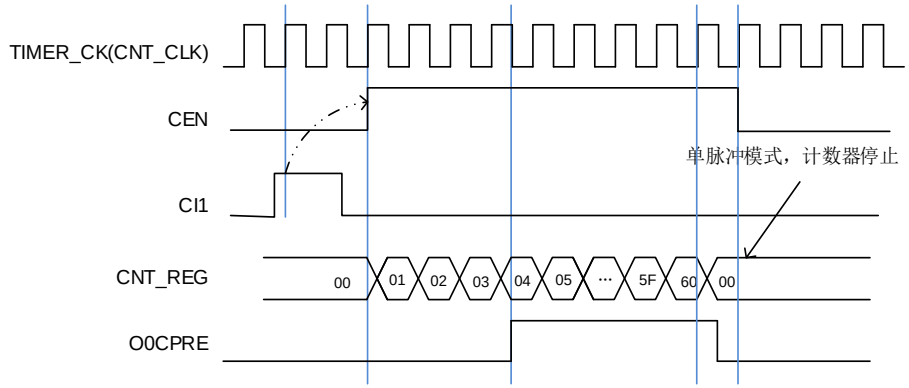
单脉冲模式与重复模式是相反的，设置 TIMERx_CTL0 寄存器的 SPM 位置 1，可使能单脉冲模式。当 SPM 置 1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置 CHxCOMCTL/ MCHxCOMCTL 配置 TIMERx 为 PWM 模式或者比较模式。

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 24-43. 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，通道输出参考信号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)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。

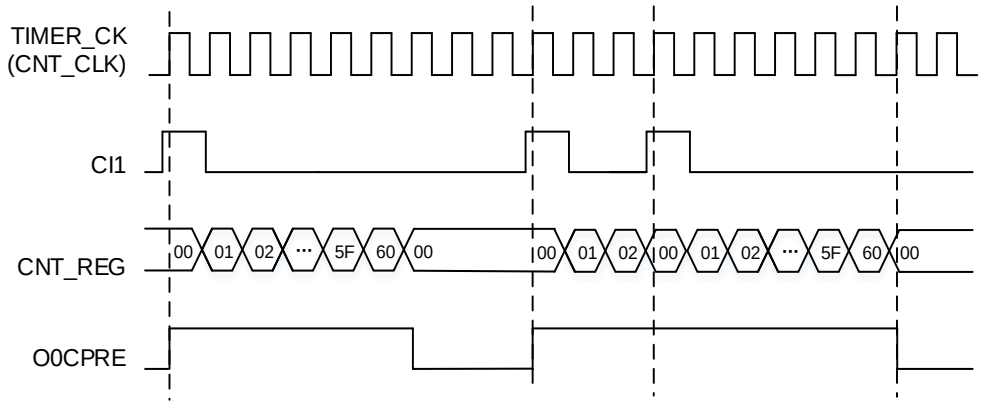
在可延时单脉冲模式1下，OxCPRE/ MOxCPRE的输出情况类似与PWM模式1。在向上计数模式时，OxCPRE/ MOxCPRE先输出无效电平，当外部触发事件发生时，立即输出有效电平，当下一次更新事件发生时，再变成无效电平；在向下计数模式时，OxCPRE/ MOxCPRE先输出有效电平，当外部触发事件发生时，立即输出无效电平，当下一次更新事件发生时，再变成有效电平。

注意：

1） 不能使用中央对齐模式，TIMERx_CTL0 寄存器中的 CAM[1:0] = 2'b00；

2） 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 24-44. 可延时单脉冲模式（TIMERx_CHxCV=0x00, TIMERx_CAR=0x60）



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：

- 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=3'b010）。定时器2在每次计数器溢出产生更新事件时，输出一个周期信号；
2. 配置定时器2周期（TIMER2_CAR寄存器）；
3. 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] = 5b'00011）；
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）。

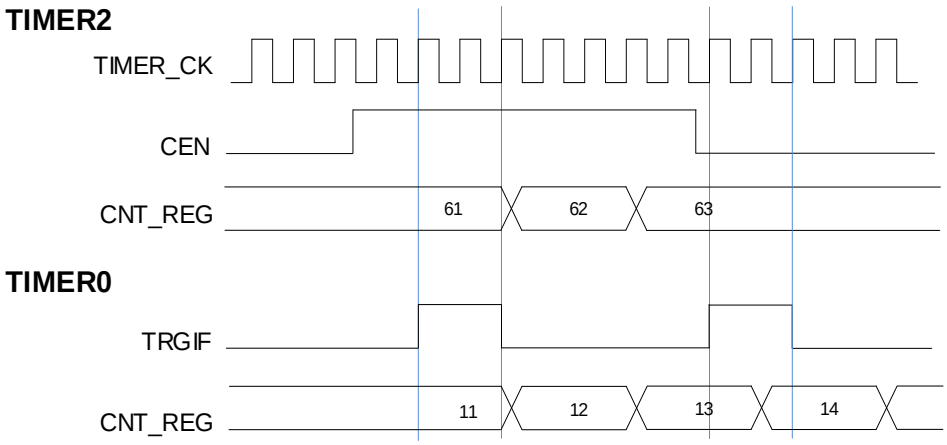
■ 用定时器2的使能/更新信号来启动定时器0

用定时器 2 的使能信号来启动定时器 0，见*图 24-45. 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=3'b001）；
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] = 5b'00011）；
3. 写1到CEN来开启定时器2（TIMER2_CTL0寄存器）。

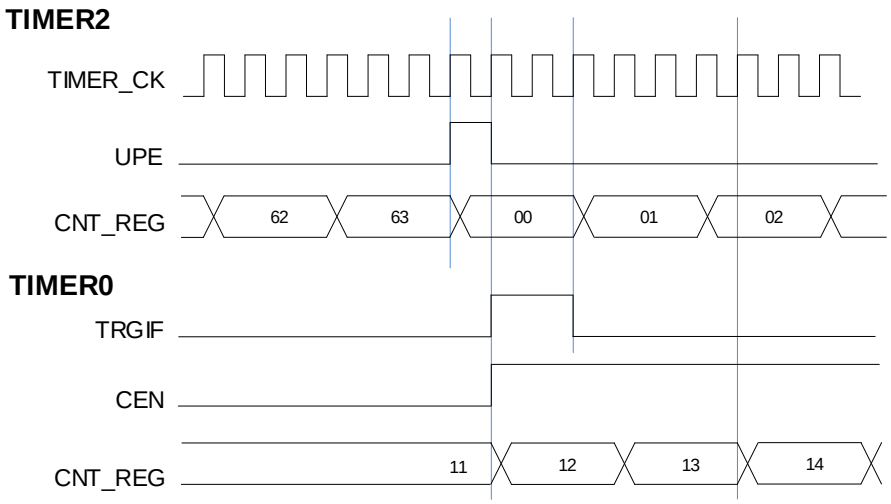
图 24-45. 用定时器 2 的使能信号触发定时器 0



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24-46. 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=3'b010）；
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] = 5b'00011）
4. 写1到CEN来开启定时器2（TIMER2_CTL0寄存器）。

图 24-46. 用定时器 2 的更新事件来触发定时器 0

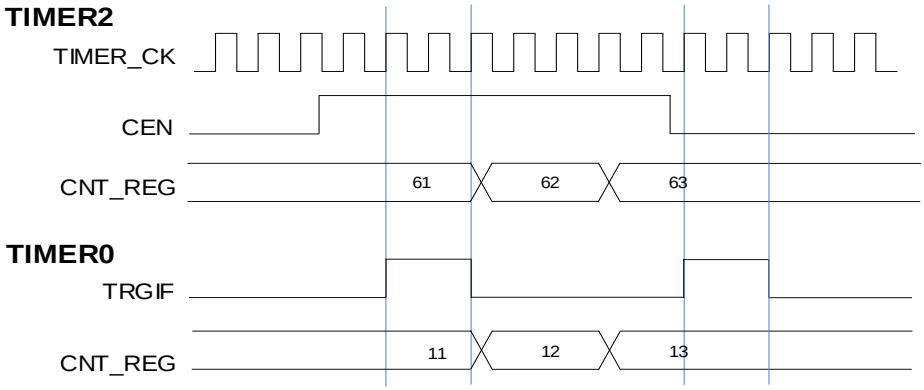


- 使用定时器2的使能/O0CPRE参考信号来使能定时器0计数。

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24-47. 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=3'b001）；
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] = 5b'00011）；
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 24-47. 用定时器 2 的使能来选通定时器 0



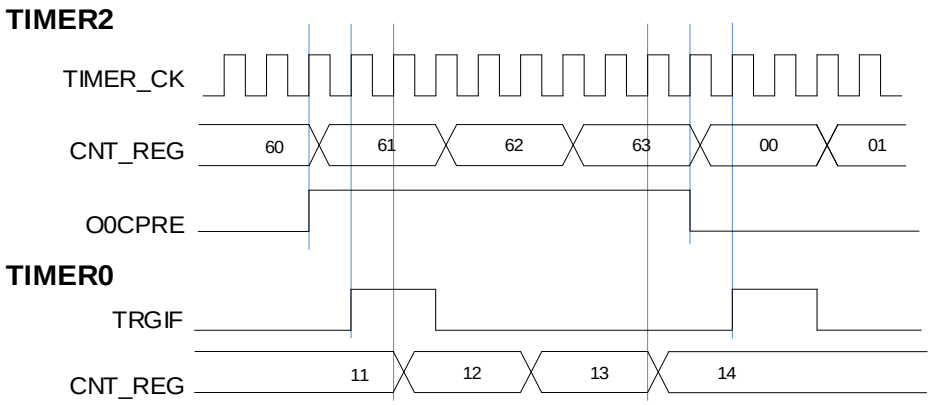
这个例子中，我们也可以使用定时器2的O0CPRE信号代替其使能信号输出作为触发源。步骤如下：

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] = 5b’00011）；
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 24-48. 用定时器 2 的 O0CPRE 信号选通定时器 1



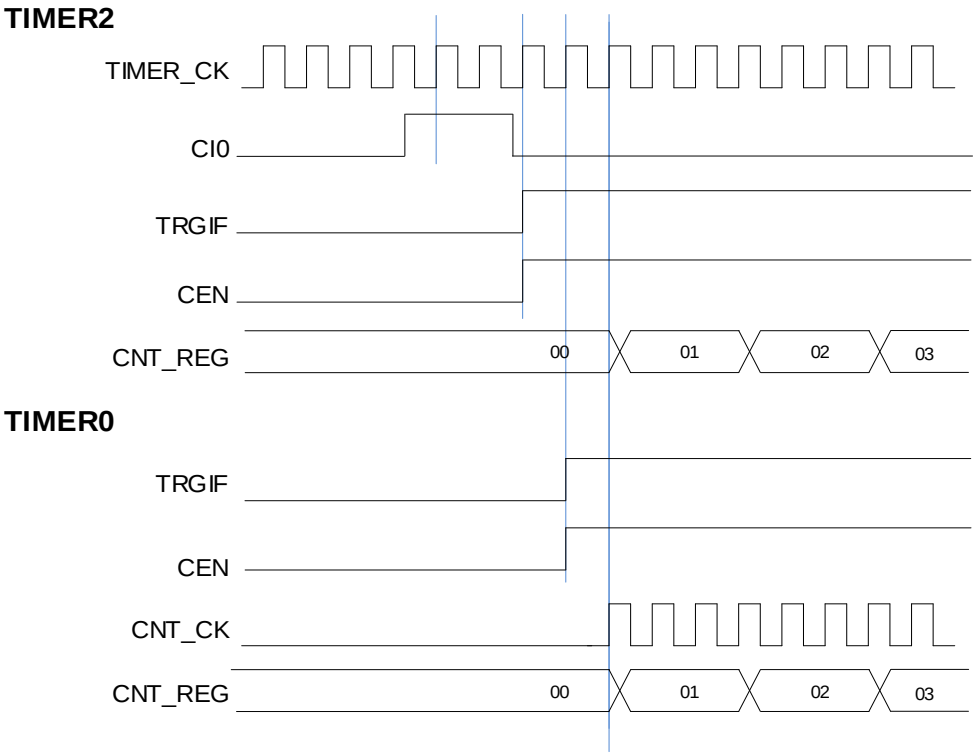
■ 使用一个外部触发来同步两个定时器

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] = 5b’00101）；
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] = 5b’00011）。

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 24-49. 用定时器 2 的 CI0 输入来触发定时器 0 和定时器 2



定时器 DMA 模式

定时器 DMA 模式是指通过 DMA 模块配置定时器的寄存器。有两个跟定时器 DMA 模式相关的寄存器：TIMERx_DMACFG 和 TIMERx_DMATB。当然，必须要使能 DMA 请求，一些内部中断事件可以产生 DMA 请求。当中断事件发生，TIMERx 会给 DMA 发送请求。DMA 配置成 M2P 模式，PADDR 是 TIMERx_DMATB 寄存器地址，DMA 就会访问 TIMERx_DMATB 寄存器。实际上，TIMERx_DMATB 寄存器只是一个缓冲，定时器会将 TIMERx_DMATB 映射到一个内部寄存器，这个内部寄存器由 TIMERx_DMACFG 寄存器中的 DMATA 来指定。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值为 0，表示 1 次传输，定时器的发送 1 个 DMA 请求就可以完成。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值不为 1，例如其值为 3，表示 4 次传输，定时器就需要再多发 3 次 DMA 请求。在这 3 次请求下，DMA 对 TIMERx_DMATB 寄存器的访问会映射到访问定时器的 DMATA+0x4、DMATA+0x8、DMATA+0xC 寄存器。总之，发生一次 DMA 内部中断请求，定时器会连续发送（DMATC+1）次请求。

如果再来1次DMA请求事件，TIMERx将会重复上面的过程。

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®-M7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。

24.1.5. TIMERx 寄存器（x=0,7）

TIMER0基地址：0x4001 0000

TIMER7基地址：0x4001 0400

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				UPIFBUEN	保留	CKDIV[1:0]		ARSE	CAM[1:0]		DIR	SPM	UPS	UPDIS	CEN
				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	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							MMC1[2:0]			保留			

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。
29:23	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_CCHP 寄存器的 PROT [1:0]位为 00 的时候可以被更改。
8	ISO0	通道 0 的空闲状态输出 0：当 POEN 复位，CH0_O 输出低电平 1：当 POEN 复位，CH0_O 输出高电平 如果 MCH0_O 生效，一个死区时间后 CH0_O 输出改变。此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]位为 00 的时候可以被更改。
7	TI0S	通道 0 触发输入选择 0：选择 TIMERx_CH0 引脚作为通道 0 的触发输入 1：选择 TIMERx_CH0，CH1 和 CH2 引脚异或的结果作为通道 0 的触发输入
6:4	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。 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：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。 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 TRGO0。
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															

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						
rw	rw	rw		rw				rw							

位/位域	名称	描述
31:16	保留	必须保持复位值
15	ETP	外部触发极性 该位指定 ETI 信号的极性。 0：ETI 高电平或上升沿有效 1：ETI 低电平或下降沿有效
14	SMC1	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 5b'01000。 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
6:0	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	保留		DECDISIE	DECJIE
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	TRGDEN	CMTDEN	CH3DEN	CH2DEN	CH1DEN	CH0DEN	UPDEN	BRKIE	TRGIE	CMTIE	CH3IE	CH2IE	CH1IE	CH0IE	UPIE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	CH3COMADDIE	通道 3 附加比较中断使能 0：禁止通道 3 附加比较中断 1：使能通道 3 附加比较中断 **注意**：此中断使能位仅用于复合 PWM 模式。
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] = 2b’00 时）。
26	MCH2DEN	多模式通道 2 比较/捕获 DMA 请求使能 0：禁止多模式通道 2 比较/捕获 DMA 请求 1：使能多模式通道 2 比较/捕获 DMA 请求 **注意**：此 DMA 使能位仅用于多模式通道输入和输出独立模式（当 MCH2MSEL[1:0] = 2b’00 时）。
25	MCH1DEN	多模式通道 1 比较/捕获 DMA 请求使能 0：禁止多模式通道 1 比较/捕获 DMA 请求 1：使能多模式通道 1 比较/捕获 DMA 请求 **注意**：此 DMA 使能位仅用于多模式通道输入和输出独立模式（当 MCH1MSEL[1:0] = 2b’00 时）。

24	MCH0DEN	多模式通道 0 比较/捕获 DMA 请求使能 0：禁止多模式通道 0 比较/捕获 DMA 请求 1：使能多模式通道 0 比较/捕获 DMA 请求 **注意**：此 DMA 使能位仅用于多模式通道输入和输出独立模式（当 MCH0MSEL[1:0] = 2b'00 时）。
23	MCH3IE	多模式通道 3 比较/捕获中断使能 0：禁止多模式通道 3 中断 1：使能多模式通道 3 中断 **注意**：此中断使能位仅用于多模式通道输入和输出独立模式（当 MCH3MSEL[1:0] = 2b'00 时）。
22	MCH2IE	多模式通道 2 比较/捕获中断使能 0：禁止多模式通道 2 中断 1：使能多模式通道 2 中断 **注意**：此中断使能位仅用于多模式通道输入和输出独立模式（当 MCH2MSEL[1:0] = 2b'00 时）。
21	MCH1IE	多模式通道 1 比较/捕获中断使能 0：禁止多模式通道 1 中断 1：使能多模式通道 1 中断 **注意**：此中断使能位仅用于多模式通道输入和输出独立模式（当 MCH1MSEL[1:0] = 2b'00 时）。
20	MCH0IE	多模式通道 0 比较/捕获中断使能 0：禁止多模式通道 0 中断 1：使能多模式通道 0 中断 **注意**：此中断使能位仅用于多模式通道输入和输出独立模式（当 MCH0MSEL[1:0] = 2b'00 时）。
19:18	保留	必须保持复位值.
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	保留	必须保持复位值。
14	TRGDEN	触发 DMA 请求使能 0：禁止触发 DMA 请求 1：使能触发 DMA 请求
13	CMTDEN	换相 DMA 更新请求使能

0：禁止换相 DMA 更新请求
1：使能换相 DMA 更新请求

12	CH3DEN	通道 3 比较/捕获 DMA 请求使能 0：禁止通道 3 比较/捕获 DMA 请求 1：使能通道 3 比较/捕获 DMA 请求
11	CH2DEN	通道 2 比较/捕获 DMA 请求使能 0：禁止通道 2 比较/捕获 DMA 请求 1：使能通道 2 比较/捕获 DMA 请求
10	CH1DEN	通道 1 比较/捕获 DMA 请求使能 0：禁止通道 1 比较/捕获 DMA 请求 1：使能通道 1 比较/捕获 DMA 请求
9	CH0DEN	通道 0 比较/捕获 DMA 请求使能 0：禁止通道 0 比较/捕获 DMA 请求 1：使能通道 0 比较/捕获 DMA 请求
8	UPDEN	更新 DMA 请求使能 0：禁止更新 DMA 请求 1：使能更新 DMA 请求
7	BRKIE	中止中断使能 0：禁止中止中断 1：使能中止中断
6	TRGIE	触发中断使能 0：禁止触发中断 1：使能触发中断
5	CMTIE	换相更新中断使能 0：禁止换相更新中断 1：使能换相更新中断
4	CH3IE	通道 3 比较/捕获中断使能 0：禁止通道 3 中断 1：使能通道 3 中断
3	CH2IE	通道 2 比较/捕获中断使能 0：禁止通道 2 中断 1：使能通道 2 中断
2	CH1IE	通道 1 比较/捕获中断使能 0：禁止通道 1 中断 1：使能通道 1 中断
1	CH0IE	通道 0 比较/捕获中断使能 0：禁止通道 0 中断 1：使能通道 0 中断

0	UPIE	更新中断使能 0：禁止更新中断 1：使能更新中断

中断标志寄存器（TIMERx_INTF）

地址偏移：0x10

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3COMADDIF	CH2COMADDIF	CH1COMADDIF	CH0COMADDIF	MCH3OF	MCH2OF	MCH1OF	MCH0OF	MCH3IF	MCH2IF	MCH1IF	MCH0IF	保留		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
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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

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
19:18	保留	必须保持复位值
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。
15:14	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	保留			CH1COMCTL[3]	保留							CH0COMCTL[3]
		保留	保留				保留								保留
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	保留	CH1MS[1:0]		CH0COMCEN	CH0COMCTL[2:0]			CH0COMSEN	保留	CH0MS[1:0]	
CH1CAPFLT[3:0]				CH1CAPPSC[1:0]				CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

输出比较模式：

位/位域	名称	描述
31	CH1MS[2]	通道 1 I/O 模式选择 参考 CH1MS[1:0]描述。
30	CH0MS[2]	通道 0 I/O 模式选择 参考 CH0MS[1:0]描述。
29	CH1COMADDSEN	通道 1 附加输出比较影子寄存器使能 参考 CH0COMADDSEN 描述。
28	CH0COMADDSEN	通道 0 附加输出比较影子寄存器使能 当此位被置 1，TIMERx_CH0COMV_ADD 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 0 附加比较输出影子寄存器 1：使能通道 0 附加比较输出影子寄存器 仅在单脉冲模式下（TIMERx_CTL0 寄存器的 SPM =1），可以在未确认预装载寄存器情况下使用 PWM 模式。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =000 时此位不能被改变。
27:25	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	保留	必须保持复位值
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] = 2b'00 时，TIMERx_CHCTL2 寄存器的 CH1EN 位清 0；当 MCH1MSEL[1:0] = 2b'01 或 2b'11 时，TIMERx_CHCTL2 寄存器的 CH1EN、MCH1EN 位清 0）时，这些位才可以写。 000：通道 1 配置为输出 001：通道 1 配置为输入，IS1 映射在 CI1FE1 上 010：通道 1 配置为输入，IS1 映射在 CI0FE1 上 011：通道 1 配置为输入，IS1 映射在 ITS 上，此模式仅工作在内部触发器输入被选中时（由 SYSCFG_TIMERxCFG2(x=0,7)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] = 2b'11 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_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =000（比较模式）时，此位不能被改变。

3	CH0COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_CH0CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 0 输出/比较影子寄存器 1：使能通道 0 输出/比较影子寄存器 仅在单脉冲模式下（TIMERx_CTL0 寄存器的 SPM =1），可以在未确认预装载寄存器情况下使用 PWM 模式。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =000 时此位不能被改变。
2	保留	必须保持复位值
1:0	CH0MS[1:0]	通道 0 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭（当

MCH0MSEL[1:0] = 2b'00 时，TIMERx_CHCTL2 寄存器的 CH1EN 位清 0；当 MCH0MSEL[1:0] = 2b'01 或 2b'11 时，TIMERx_CHCTL2 寄存器的 CH0EN、MCH0EN 位清 0）时，CH0MS[2:0]才可写。

000：通道 0 配置为输出

001：通道 0 配置为输入，IS0 映射在 CI0FE0 上

010：通道 0 配置为输入，IS0 映射在 CI1FE0 上

011：通道 0 配置为输入，IS0 映射在 ITS 上。此模式仅工作在内部触发输入被选中时（由 SYSCFG_TIMERxCFG2(x=0,7)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	保留			CH3COMCTL[3]	保留							CH2COMCTL[3]
		保留	保留				保留								保留
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	保留	CH3MS[1:0]		CH2COMCEN	CH2COMCTL[2:0]			CH2COMSEN	保留	CH2MS[1:0]	
CH3CAPFLT[3:0]				CH3CAPPSC[1:0]				CH2CAPFLT[3:0]				CH2CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

输出比较模式：

位/位域	名称	描述
31	CH3MS[2]	通道 3 I/O 模式选择 参考 CH3MS[1:0]描述。
30	CH2MS[2]	通道 2 I/O 模式选择 参考 CH2MS[1:0]描述。
29	CH3COMADDSEN	通道 3 附加输出比较影子寄存器使能 参考 CH2COMADDSEN 描述。
28	CH2COMADDSEN	通道 2 附加输出比较影子寄存器使能 当此位被置 1，TIMERx_CH2COMV_ADD 寄存器的影子寄存器使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 2 附加输出/比较影子寄存器 1：使能通道 2 附加输出/比较影子寄存器 仅在单脉冲模式下（TIMERx_CTL0 寄存器的 SPM =1），可以在未确认预装载寄存器情况下使用 PWM 模式。

		当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH2MS =000 时此位不能被改变。
27:25	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	保留	必须保持复位值
9:8	CH3MS[1:0]	通道 3 模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭（当 MCH3MSEL[1:0] = 2b'00 时，TIMERx_CHCTL2 寄存器的 CH3EN 位清 0；当 MCH3MSEL[1:0] = 2b'01 或 2b'11 时，TIMERx_CHCTL2 寄存器的 CH3EN、MCH3EN 位清 0）时，这些位才可以写。 000：通道 3 配置为输出 001：通道 3 配置为输入，IS3 映射在 CI3FE3 上 010：通道 3 配置为输入，IS3 映射在 CI2FE3 上 011：通道 3 配置为输入，IS3 映射在 ITS 上，此模式仅工作在内部触发器输入被选中时（由 SYSCFG_TIMERxCFG2(x=0,7)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] = 2b'11，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。

00100：强制为低。强制 O2CPRE 为低电平

101：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。

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 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_CCHP 寄存器的 PROT [1:0]=11 且 CH2MS =000（比较模式）时此位不能被改变。

3	CH2COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_CH2CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 2 输出/比较影子寄存器 1：使能通道 2 输出/比较影子寄存器 仅在单脉冲模式下（TIMERx_CTL0 寄存器的 SPM =1），可以在未确认预装载寄存器情况下使用 PWM 模式。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH2MS =000 时此位不能被改变。
2	保留	必须保持复位值

1:0	CH2MS[1:0]	通道 2 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭（当 MCH2MSEL[1:0] = 2b'00 时，TIMERx_CHCTL2 寄存器的 CH2EN 位清 0；当 MCH1MSEL[1:0] = 2b'01 或 2b'11 时，TIMERx_CHCTL2 寄存器的 CH2EN、MCH2EN 位清 0）时，这些位才可写。 000：通道 2 配置为输出 001：通道 2 配置为输入，IS2 映射在 CI2FE2 上 010：通道 2 配置为输入，IS2 映射在 CI3FE2 上 011：通道 2 配置为输入，IS2 映射在 ITS 上。此模式仅工作在内部触发输入被选中时（由 SYSCFG_TIMERxCFG2(x=0,7)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
31:16	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] = 2b'11 时，此位定义了多模式通道 0 输出信号 MCH0_O 的极性。 0：多模式通道 0 高电平有效 1：多模式通道 0 低电平有效 当通道 0 配置为输入模式时，此位和 CH0P 联合使用，作为通道 0 的极性选择控制信号。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。
2	MCH0EN	多模式通道 0 捕获/比较使能 当多模式通道 0 配置为输出模式时，将此位置 1 使能 MCH0_O 信号有效。当多模式通道 0 配置为输入模式时，将此位置 1 使能多模式通道 0 上的捕获事件。 0：禁止多模式通道 0 1：使能多模式通道 0
1	CH0P	通道 0 捕获/比较极性 当通道 0 配置为输出模式时，此位定义了输出信号极性。 0：通道 0 高电平有效 1：通道 0 低电平有效 当通道 0 配置为输入模式时，此位定义了通道 0 输入信号的极性。[MCH0P, CH0P] 用于选择通道 0 输入信号信号有效边沿或者捕获极性。 00：把通道 0 输入信号的上升沿作为捕获或者从模式下触发的有效信号，且通道 0 输入信号不会被翻转。 01：把通道 0 输入信号的下降沿作为捕获或者从模式下触发的有效信号，且通道 0 输入信号会被翻转。 10：保留。 11：把通道 0 输入信号的上升沿和下降沿都作为捕获或者从模式下触发的有效信号，且通道 0 输入信号不翻转。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。
0	CH0EN	通道 0 捕获/比较使能 当通道 0 配置为输出模式时，将此位置 1 使能 CH0_O 信号有效。当通道 0 配置为输入模式时，将此位置 1 使能通道 0 上的捕获事件。 0：禁止通道 0 1：使能通道 0

计数器寄存器（TIMERx_CNT）

地址偏移：0x24

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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															

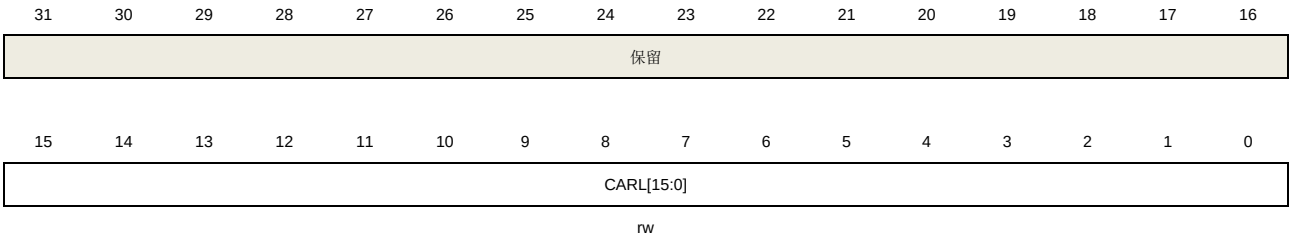
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 PSC 时钟除以（PSC+1），每次当更新事件产生时，PSC 的值被装入当前预分频寄存器。

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。



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

重复计数寄存器 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。



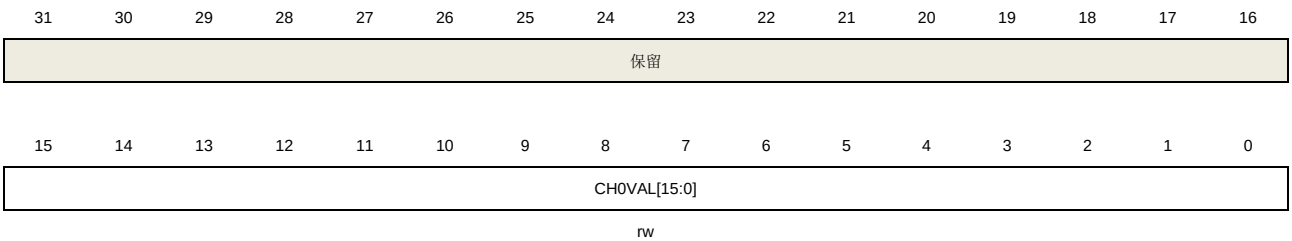
位/位域	名称	描述
31:8	保留	必须保持复位值.
7:0	CREP0[7:0]	重复计数器的值 0 这些位定义了更新事件的产生速率。重复计数器计数值减为 0 时产生更新事件。影子寄存器的更新速率也会受这些位影响（前提是影子寄存器被使能）。 **注意**：当 TIMERx_CFG 寄存器中的 CREPSEL =0 时，使用该位。

通道 0 捕获/比较寄存器（TIMERx_CH0CV）

地址偏移：0x34

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值 当通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 1 捕获/比较寄存器（TIMERx_CH1CV）

地址偏移：0x38

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															
rw															

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

通道 2 捕获/比较寄存器（TIMERx_CH2CV）

地址偏移：0x3C

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH2VAL[15:0]	通道 2 的捕获或比较值

当通道 2 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。
当通道 2 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 3 捕获/比较寄存器（TIMERx_CH3CV）

地址偏移：0x40
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH3VAL[15:0]	通道 3 的捕获或比较值 当通道 3 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 3 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

互补通道保护寄存器（TIMERx_CCHP）

地址偏移：0x44
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		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_CCHP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（高阻态的开漏描述）。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_CCHP 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_CCHP 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_CCHP 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_CCHP 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_CCHP 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_CCHP 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_CCHP 寄存器的 PROT [1:0] =00 时才可修改。

11 ROS
运行模式下“关闭状态”使能
当 POEN 位被置 1（运行模式），此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。参见 ***表 24-4. 由参数控制的互补输出表（MCHxMSEL=2'b11）***。
0：输出“关闭状态”禁能。当 CHxEN 或者 CHxNEN 位被清零，对应通道为输出“禁能状态”。
1：输出“关闭状态”使能。当 CHxEN 或者 CHxNEN 位被清零，对应通道为输出“关闭状态”。
此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。

10 IOS
空闲模式下“关闭状态”使能
当 POEN 位被清 0（空闲模式），此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。参见 ***表 24-4. 由参数控制的互补输出表（MCHxMSEL=2'b11）***。
0：输出“关闭状态”禁能。当 CHxEN 和 CHxNEN 位均被清零，对应通道为输出“禁能状态”。
1：输出“关闭状态”使能。不论 CHxEN 和 CHxNEN 位的值，对应通道为输出“关闭状态”。
此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。

9:8 PROT[1:0]
互补寄存器保护控制
这两位定义了寄存器的写保护特性。
00：禁能保护模式。无写保护。
01：PROT 模式 0。TIMERx_CTL1 寄存器中 ISOx/ISOxN 位，TIMERx_CCHP 寄存器中 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_CCHP 寄存器中 ROS/IOS 位和 TIMERx_FCCHPx（x = 0..3）寄存器中 ROS/IOS 位。
11：PROT 模式 2。除了 PROT 模式 1 下的寄存器写保护外，还有 TIMERx_CHCTLR0/ 1 及 TIMERx_MCHCTL0/ 1 寄存器中 CHxCOMCTL/ CHxCOMSEN/ CHxCOMADDSEN/ MCHxCOMCTL/ MCHxCOMSEN 位（如果相关通道配置为输出模式）写保护。
系统复位后这两位只能被写一次，一旦 TIMERx_CCHP 寄存器被写入，这两位被写

保护。

7:0	DTCFG[7:0]	死区时间控制 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]）xt_{DT}, t_{DT} =t_{DTS}*2. DTCFG [7:5] =3'b 110：DTvalue =（32+DTCFG [4:0]）xt_{DT}, t_{DT}=t_{DTS}*8. DTCFG [7:5] =3'b 111：DTvalue =（32+DTCFG [4:0]）xt_{DT}, t_{DT} =t_{DTS}*16. 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]=00 时才可修改。

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	保留	MCH1MS[1:0]		MCH0COMCEN	MCH0COMCTL[2:0]			MCH0COMSEN	保留	MCH0MS[1:0]	
MCH1CAPFLT[3:0]				MCH1CAPPSC[1:0]				MCH0CAPFLT[3:0]				MCH0CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

输出比较模式：

位/位域	名称	描述
31	MCH1MS[2]	多模式通道 1 I/O 模式选择 参考 MCH1MS[1:0]描述。
30	MCH0MS[2]	多模式通道 0 I/O 模式选择 参考 MCH0MS[1:0]描述。
29:25	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	保留	必须保持复位值
9:8	MCH1MS[1:0]	多模式通道 1 I/O 模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭（TIMERx_CHCTL2 寄存器的 MCH1EN 位清 0）时这些位才可以写。 000：多模式通道 1 配置为输出 001：多模式通道1配置为输入，MIS1映射在MCI1FEM1上 010：多模式通道1配置为输入，MIS1映射在MCI0FEM1上 011：多模式通道 1 配置为输入，MIS1 映射在 ITS 上。此模式仅工作在内部触发器输入被选中时（由 SYSCFG_TIMERxCFG2(x=0,7)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] = 2b’00，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] = 2b’11 时，CH0COMCTL[2:0]位定义了输出准备信号 O0CPRE 的动作，而 O0CPRE 决定了 CH0_O、MCH0_O 的值。O0CPRE 高电平有效，CH0_O、MCH0_O 的有效电平取决于 CH0P、MCH0P 位。 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_CCHP 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_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS=00 时此位不能被改变。
2	保留	必须保持复位值
1:0	MCH0MS[1:0]	多模式通道 0 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭（TIMERx_CHCTL2 寄存器的 MCH0EN 位清 0）时，MCH0MS[2:0]才可写。 000：多模式通道 0 配置为输出 001：多模式通道 0 配置为输入，MIS0 映射在 MCI0FEM0 上 010：多模式通道 0 配置为输入，MIS0 映射在 MCI1FEM0 上 011：多模式通道 0 配置为输入，MIS0 映射在 ITS 上。此模式仅工作在内部触发输入被选中时（由 SYSCFG_TIMERxCFG2(x=0,7)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

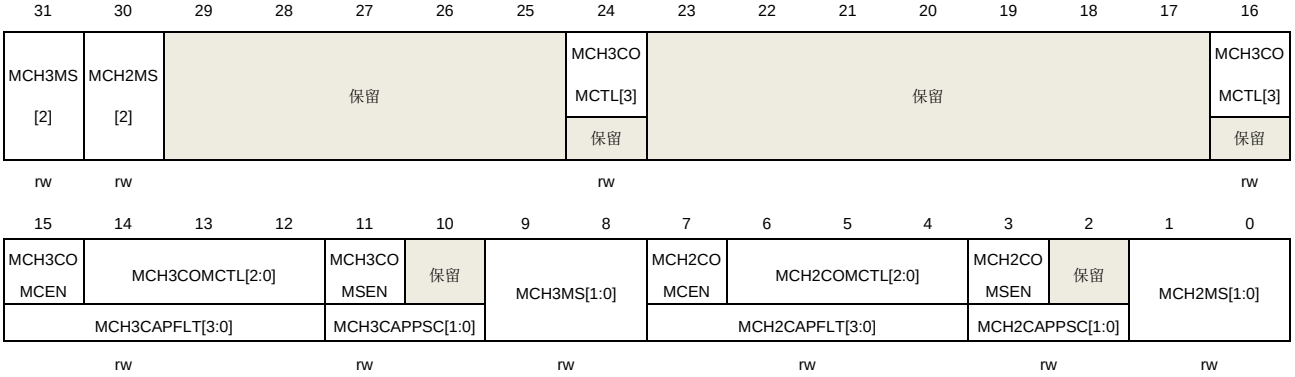
31	MCH1MS[2]	多模式通道 1 I/O 模式选择 参考 MCH1MS[1:0]描述。
30	MCH0MS[2]	多模式通道 0 I/O 模式选择 参考 MCH0MS[1:0]描述。
29:16	保留	必须保持复位值
15:12	MCH1CAPFLT[3:0]	多模式通道 1 输入捕获滤波控制 参见 MCH0CAPFLT 描述。
11:10	MCH1CAPPSC[1:0]	多模式通道 1 输入捕获预分频器 参见 MCH0CAPPSC 描述。
9:8	MCH1MS[1:0]	多模式通道 1 I/O 模式选择 与输出模式相同。
7:4	MCH0CAPFLT[3:0]	通道 0 输入捕获滤波控制 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	保留	必须保持复位值
9:8	MCH3MS[1:0]	多模式通道 3 I/O 模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭（TIMERx_CHCTL2 寄存器的 MCH3EN 位清 0）时，这些位才可以写。 000：多模式通道 3 配置为输出 001：多模式通道 3 配置为输入，MIS3 映射在 MCI3FEM3 上

010：多模式通道 3 配置为输入，MIS3 映射在 MCI2FEM3 上
011：多模式通道 3 配置为输入，MIS3 映射在 ITS 上，此模式仅工作在内部触发器输入被选中时（由 SYSCFG_TIMERxCFG2(x=0,7)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] = 2b'00，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] = 2b'11 时，CH2COMCTL[2:0]位定义了输出准备信号 O2CPRE 的动作，而 O2CPRE 决定了 CH2_O、MCH2_O 的值。O2CPRE 高电平有效，CH2_O、MCH2_O 的有效电平取决于 CH2P、MCH2P 位。 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_CCHP 寄存器的 PROT [1:0]=11 且 MCH2MS =00（比较模式）时此位不能被改变。
3	MCH2COMSEN	多模式通道 2 输出比较影子寄存器使能 当此位被置 1，TIMERx_MCH2CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止多模式通道 2 输出/比较影子寄存器 1：使能多模式通道 2 输出/比较影子寄存器 仅在单脉冲模式下（TIMERx_CTL0 寄存器的 SPM =1），可以在未确认预装载寄存器情况下使用 PWM 模式。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 MCH2MS =00 时此位不能被改变。
2	保留	必须保持复位值
1:0	MCH2MS[1:0]	多模式通道 2 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭（TIMERx_CHCTL2 寄存器的 MCH2EN 位清 0）时，这些位才可写。 000：多模式通道 2 配置为输出 001：多模式通道 2 配置为输入，MIS2 映射在 MCI2FEM2 上 010：多模式通道 2 配置为输入，MIS2 映射在 MCI3FE2M 上 011：多模式通道 2 配置为输入，MIS2 映射在 ITS 上。此模式仅工作在内部触发输入被选中时（由 SYSCFG_TIMERxCFG2(x=0,7)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] = 2b'00，此位定义了输出信号极性。 00：多模式通道 0 高电平有效 01：多模式通道 0 低电平有效 10：保留 11：保留 当通道 0 配置为输入模式时，此位定义了多模式通道 0 输入信号的极性。 MCH0FP[1:0]将选择多模式通道 0 输入信号的有效边沿或者捕获极性。 00：把多模式通道 0 输入信号的上升沿作为捕获或者从模式下触发的有效信号，且多模式通道 0 输入信号不会被翻转。 01：把多模式通道 0 输入信号的下降沿作为捕获或者从模式下触发的有效信号，且多模式通道 0 输入信号会被翻转。 10：保留。 11：把多模式通道 0 输入信号的上升沿或下降沿作为捕获或者从模式下触发的有效信号，并且多模式通道 0 输入信号不会被翻转。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。

多模式通道 0 捕获/比较寄存器（TIMERx_MCH0CV）

地址偏移：0x54

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCH0VAL[15:0]															
rw															

位/位域	名称	描述

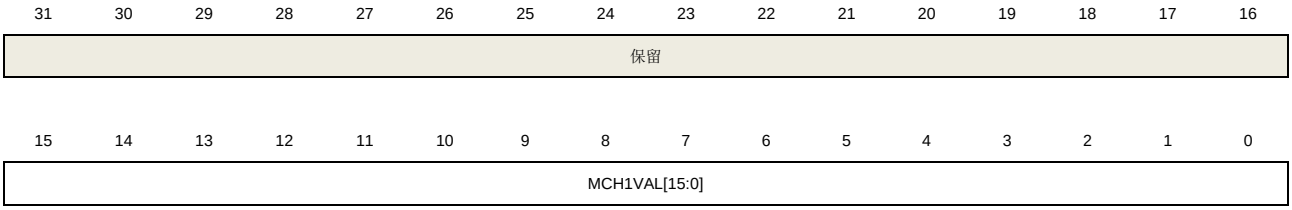
31:16	保留	必须保持复位值
15:0	MCH0VAL[15:0]	多模式通道 0 的捕获或比较值 当多模式通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值，且本寄存器为只读。 当多模式通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

多模式通道 1 捕获/比较寄存器（TIMERx_MCH1CV）

地址偏移：0x58

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



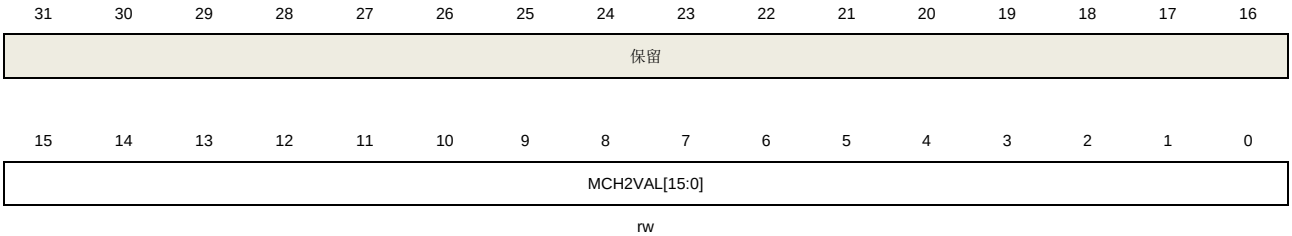
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	MCH1VAL[15:0]	多模式通道 1 的捕获或比较值 当多模式通道 1 配置为输入模式时，这些位决定了上次捕获事件的计数器值，且本寄存器为只读。 当多模式通道 1 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

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。



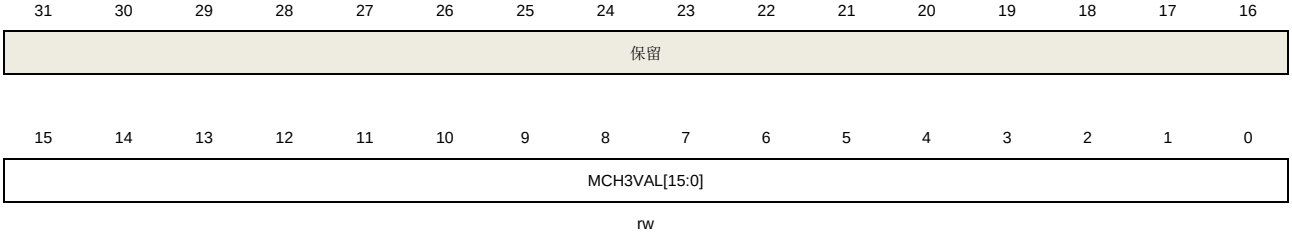
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	MCH2VAL[15:0]	多模式通道 2 的捕获或比较值

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。

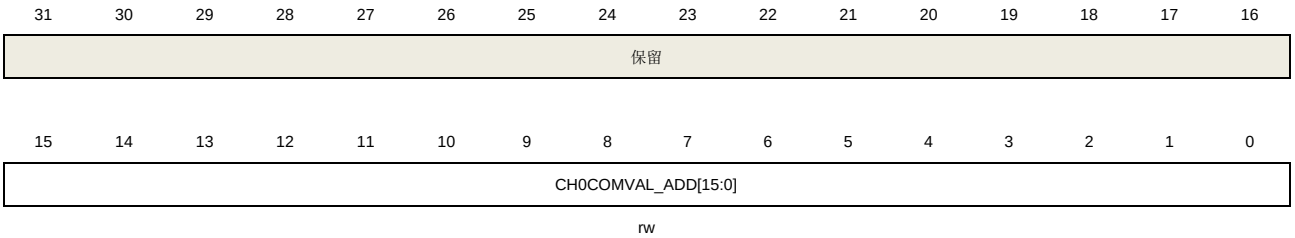


位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	MCH3VAL[15:0]	多模式通道 3 的捕获或比较值 当多模式通道 3 配置为输入模式时，这些位决定了上次捕获事件的计数器值，且本寄存器为只读。 当多模式通道 3 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

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
CH3CPWMEN	CH2CPWMEN	CH1CPWMEN	CH0CPWMEN	MCH3MSEL[1:0]		MCH2MSEL[1:0]		MCH1MSEL[1:0]		MCH0MSEL[1:0]		DECDISDEN	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]		BRKENCH3	BRKENCH2	BRKENCH1	BRKENCH0	DTIENCH3	DTIENCH2	DTIENCH1	DTIENCH0
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
FCCHP0EN								保留							
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				ROS	IOS	保留		DTCFG[7:0]							
				rw	rw			rw							

位/位域	名称	描述
31	FCCHP0EN	独立互补通道寄存器 0 使能位 0：TIMERx_CCHP 寄存器中的 ROS、IOS 和 DTCFG[7:0]有效 1：TIMERx_FCCHP0 寄存器中的 ROS、IOS 和 DTCFG[7:0]有效 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]=00 时才可修改。
30:12	保留	必须保持复位值
11	ROS	运行模式下“关闭状态”使能 当 POEN 位被置 1（运行模式），此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。 0：输出“关闭状态”禁能。当 CH0EN 或者 CH0NEN 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 CH0EN 或者 CH0NEN 位被清零，对应通道为输出

		“关闭状态”。 此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
10	IOS	空闲模式下“关闭状态”使能 当 POEN 位被清 0（空闲模式），此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。 0：输出“关闭状态”禁能。当 CH0EN 和 CH0NEN 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 CH0EN 和 CH0NEN 位的值，对应通道为输出“关闭状态”。 此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
9:8	保留	必须保持复位值
7:0	DTCFG[7:0]	死区时间控制 这些位定义了插入互补输出之间的死区持续时间。DTCFG 值和死区时间的关系如下： DTCFG [7:5] =3’b0xx：DTvalue = DTCFG [7:0]x t_{DT}, t_{DT}=t_{DTS} DTCFG [7:5] =3’b10x：DTvalue =（64+DTCFG [5:0]）xt_{DT}, t_{DT} =t_{DTS}*2 DTCFG [7:5] =3’b110：DTvalue =（32+DTCFG [4:0]）xt_{DT}, t_{DT}=t_{DTS}*8 DTCFG [7:5] =3’b111：DTvalue =（32+DTCFG [4:0]）xt_{DT}, t_{DT} =t_{DTS}*16 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]=00 时才可修改。

独立互补通道保护寄存器 1（TIMERx_FCCHP1）

地址偏移：0x80

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

该寄存器用于配置CH1_O / MCH1_O的输出。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FCCHP1EN	保留														
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				ROS	IOS	保留		DTCFG[7:0]							
				rw	rw			rw							

位/位域	名称	描述
31	FCCHP1EN	独立互补通道寄存器 1 使能位 0：TIMERx_CCHP 寄存器中的 ROS、IOS 和 DTCFG[7:0]有效 1：TIMERx_FCCHP1 寄存器中的 ROS、IOS 和 DTCFG[7:0]有效 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]=00 时才可修改。
30:12	保留	必须保持复位值
11	ROS	运行模式下“关闭状态”使能

		当 POEN 位被置 1（运行模式），此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。 0：输出“关闭状态”禁能。当 CH1EN 或者 CH1NEN 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 CH1EN 或者 CH1NEN 位被清零，对应通道为输出“关闭状态”。 此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
10	IOS	空闲模式下“关闭状态”使能 当 POEN 位被清 0（空闲模式），此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。 0：输出“关闭状态”禁能。当 CH1EN 和 CH1NEN 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 CH1EN 和 CH1NEN 位的值，对应通道为输出“关闭状态”。 此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
9:8	保留	必须保持复位值
7:0	DTCFG[7:0]	死区时间控制 这些位定义了插入互补输出之间的死区持续时间。DTCFG 值和死区时间的关系如下： DTCFG [7:5] =3’b0xx：DTvalue = DTCFG [7:0]x t_{DT}, t_{DT}=t_{DTS}. DTCFG [7:5] =3’b10x：DTvalue =（64+DTCFG [5:0]）xt_{DT}, t_{DT} =t_{DTS}*2. DTCFG [7:5] =3’b110：DTvalue =（32+DTCFG [4:0]）xt_{DT}, t_{DT}=t_{DTS}*8. DTCFG [7:5] =3’b111：DTvalue =（32+DTCFG [4:0]）xt_{DT}, t_{DT} =t_{DTS}*16. 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]=00 时才可修改。

独立互补通道保护寄存器 2（TIMERx_FCCHP2）

地址偏移：0x84

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

该寄存器用于配置CH2_O / MCH2_O的输出。



位/位域	名称	描述
31	FCCHP2EN	独立互补通道寄存器 2 使能位 0：TIMERx_CCHP 寄存器中的 ROS、IOS 和 DTCFG[7:0]有效

		1：TIMERx_FCCHP2 寄存器中的 ROS、IOS 和 DTCFG[7:0]有效 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]=00 时才可修改。
30:12	保留	必须保持复位值
11	ROS	运行模式下“关闭状态”使能 当 POEN 位被置 1（运行模式），此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。 0：输出“关闭状态”禁能。当 CH2EN 或者 CH2NEN 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 CH2EN 或者 CH2NEN 位被清零，对应通道为输出“关闭状态”。 此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
10	IOS	空闲模式下“关闭状态”使能 当 POEN 位被清 0（空闲模式），此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。 0：输出“关闭状态”禁能。当 CH2EN 和 CH2NEN 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 CH2EN 和 CH2NEN 位的值，对应通道为输出“关闭状态”。 此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
9:8	保留	必须保持复位值
7:0	DTCFG[7:0]	死区时间控制 这些位定义了插入互补输出之间的死区持续时间。DTCFG 值和死区时间的关系如下： DTCFG [7:5] =3’b0xx：DTvalue = DTCFG [7:0]x t_{DT}, t_{DT}=t_{DTS}. DTCFG [7:5] =3’b10x：DTvalue =（64+DTCFG [5:0]）xt_{DT}, t_{DT} =t_{DTS}*2. DTCFG [7:5] =3’b110：DTvalue =（32+DTCFG [4:0]）xt_{DT}, t_{DT}=t_{DTS}*8. DTCFG [7:5] =3’b111：DTvalue =（32+DTCFG [4:0]）xt_{DT}, t_{DT} =t_{DTS}*16. 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]=00 时才可修改。

独立互补通道保护寄存器 3（TIMERx_FCCHP3）

地址偏移：0x88

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

该寄存器用于配置CH3_O / MCH3_O的输出。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FCCHP3EN	保留														
rw															

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

rw rw rw

位/位域	名称	描述
31	FCCHP3EN	独立互补通道寄存器 3 使能位 0：TIMERx_CCHP 寄存器中的 ROS、IOS 和 DTCFG[7:0]有效 1：TIMERx_FCCHP3 寄存器中的 ROS、IOS 和 DTCFG[7:0]有效 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]=00 时才可修改。
30:12	保留	必须保持复位值
11	ROS	运行模式下“关闭状态”配置 当 POEN 位被置 1，此位定义了通道（带有互补输出且配置为输出模式）的输出状态。 0：当 POEN 位被置 1，通道输出信号（CH3_O / MCH3_O）被禁止 1：当 POEN 位被置 1，通道输出信号（CH3_O / MCH3_O）被使能，和 TIMER0_CHCTL2 寄存器 CH3EN/ MCH3EN 位有关。 此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
10	IOS	空闲模式下“关闭状态”配置 当 POEN 位被清 0，此位定义了已经配置为输出模式的通道的输出状态。 0：当 POEN 位被清 0，通道输出信号（CH3_O / MCH3_O）被禁止 1：当 POEN 位被清 0，通道输出信号（CH3_O / MCH3_O）被使能，和 TIMERx_CHCTL2 寄存器 CH3EN/ MCH3EN 位有关。 此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
9:8	保留	必须保持复位值
7:0	DTCFG[7:0]	死区时间控制 这些位定义了插入互补输出之间的死区持续时间。DTCFG 值和死区时间的关系如下： DTCFG [7:5] =3’b0xx：DTvalue = DTCFG [7:0]x t_{DT}, t_{DT}=t_{DTS}. DTCFG [7:5] =3’b10x：DTvalue =（64+DTCFG [5:0]）xt_{DT}, t_{DT} =t_{DTS}*2. DTCFG [7:5] =3’b110：DTvalue =（32+DTCFG [4:0]）xt_{DT}, t_{DT}=t_{DTS}*8. DTCFG [7:5] =3’b111：DTvalue =（32+DTCFG [4:0]）xt_{DT}, t_{DT} =t_{DTS}*16. 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]=00 时才可修改。

TIMER0 备用功能控制寄存器 0（TIMER0_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					BRK0CMP1P	BRK0CMP0P	保留						BRK0IN2P	BRK0IN1P	BRK0IN0P
					rw	rw							rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					BRK0CMP1EN	BRK0CMP0EN	BRK0HPDFEN	保留					BRK0IN2EN	BRK0IN1EN	BRK0IN0EN
					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	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_CCHP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_CCHP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_CCHP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_CCHP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_CCHP寄存器的PROT [1:0] =00时才可修改。
15:11	保留	必须保持复位值。
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_CCHP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_CCHP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_CCHP寄存器的PROT [1:0] =00时才可修改。
7:3	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_CCHP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_CCHP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_CCHP寄存器的PROT [1:0] =00时才可修改。

TIMER0 备用功能控制寄存器 1（TIMER0_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					BRK1CMP1P	BRK1CMP0P	保留						BRK1IN2P	BRK1IN1P	BRK1IN0P
					rw	rw							rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					BRK1CMP1EN	BRK1CMP0EN	BRK1HPDFEN	保留					BRK1IN2EN	BRK1IN1EN	BRK1IN0EN

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	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_CCHP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_CCHP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	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_CCHP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_CCHP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_CCHP寄存器的PROT [1:0] =00时才可修改。
15:11	保留	必须保持复位值。

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_CCHP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_CCHP寄存器的PROT [1:0] =00时才可修改。
8	BRK1HPDFEN	BREAK1 HPDF输入（hpdf_break[1]）使能 0：HPDF输入禁能 1：HPDF输入使能 此位只有在TIMERx_CCHP寄存器的PROT [1:0] =00时才可修改。
7:3	保留	必须保持复位值。
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_CCHP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_CCHP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_CCHP寄存器的PROT [1:0] =00时才可修改。

TIMER7 备用功能控制寄存器 0（TIMER7_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					BRK0CMP1P	BRK0CMP0P	保留						BRK0IN2P	BRK0IN1P	BRK0IN0P
					rw	rw							rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					BRK0CMP1EN	BRK0CMP0EN	BRK0HPDFEN	保留					BRK0IN2EN	BRK0IN1EN	BRK0IN0EN
					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	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_CCHP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_CCHP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_CCHP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_CCHP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_CCHP寄存器的PROT [1:0] =00时才可修改。
15:11	保留	必须保持复位值。
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_CCHP寄存器的PROT [1:0] =00时才可修改。
9	BRK0CMP0EN	BREAK1 CMP0输入使能 0：CMP0输入禁能 1：CMP0输入使能 此位只有在TIMERx_CCHP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_CCHP寄存器的PROT [1:0] =00时才可修改。
7:3	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_CCHP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_CCHP寄存器的PROT [1:0] =00时才可修改。
0	BRK0IN0EN	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_CCHP寄存器的PROT [1:0] =00时才可修改。

TIMER7 备用功能控制寄存器 1（TIMER7_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					BRK1CMP1P	BRK1CMP0P	保留						BRK1IN2P	BRK1IN1P	BRK1IN0P
					rw	rw							rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					BRK1CMP1EN	BRK1CMP0EN	BRK1HPDFEN	保留					BRK1IN2EN	BRK1IN1EN	BRK1IN0EN
					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	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_CCHP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_CCHP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	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_CCHP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_CCHP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_CCHP寄存器的PROT [1:0] =00时才可修改。
15:11	保留	必须保持复位值。
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_CCHP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_CCHP寄存器的PROT [1:0] =00时才可修改。
8	BRK1HPDFEN	BREAK1 HPDF输入（hpdf_break[1]）使能 0：HPDF输入禁能 1：HPDF输入使能 此位只有在TIMERx_CCHP寄存器的PROT [1:0] =00时才可修改。
7:3	保留	必须保持复位值。
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_CCHP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_CCHP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_CCHP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。

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’b100101：传输 38 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’b100101：TIMERx_CREP1

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

24.2. 通用定时器 L0（TIMERx, x=1,2,3,4,22,23,30,31）

24.2.1. 简介

通用定时器 L0（TIMER1/2/3/4/22/23/30/31）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。

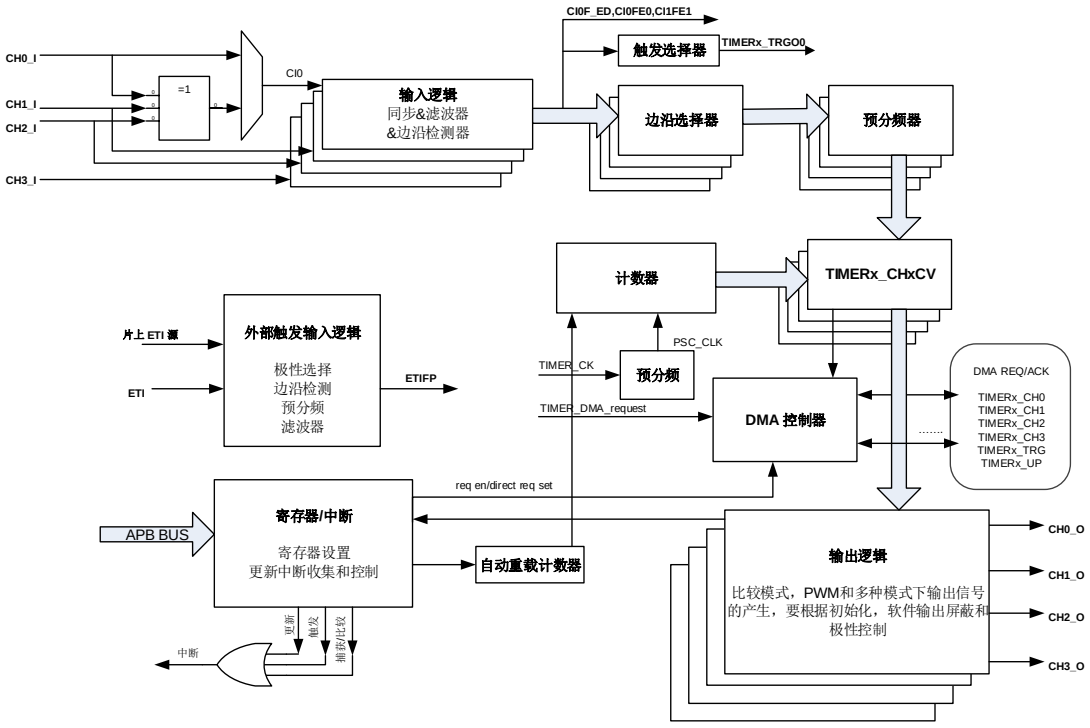
24.2.2. 主要特性

- 总通道数：4；
- 计数器宽度：16位（TIMER2/3/30/31）和32位（TIMER1/4/22/23）；
- 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。

24.2.3. 结构框图

*图 24-50. 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 24-50. 通用定时器 L0 结构框图



24.2.4. 功能描述

时钟源选择

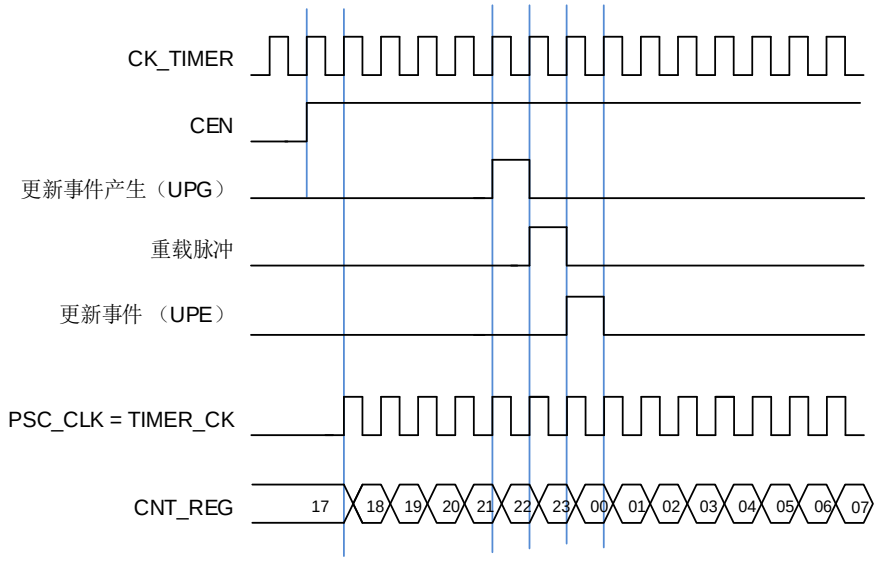
通用定时器 L0 可以由内部时钟源 CK_TIMER 或者由 SYSCFG_TIMERxCFG(x=1..4,22,23,30,31)寄存器中的 TSCFGy[4:0] (y=0..9,15)位域控制的复用时钟源驱动。

- 当 SYSCFG_TIMERxCFG(x=1..4,22,23,30,31) 寄存器中的 TSCFGy[4:0]=5'b00000(y=0..9,15)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..4,22,23,30,31) 寄存器中的 TSCFGy[4:0] =5'b00000(y=0..9,15)，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..4,22,23,30,31)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 24-51. 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 TIMERx_CI0/ TIMERx_CI1 引脚的每个上升沿或下降沿计数。这种模式可以通过设置 TSCFG6[4:0]为 0x5，0x6 或 0x7 来选择。CIx 是 TIMERx_CIx 通过数字滤波器采样后的信号。

计数器预分频器也可以在内部触发信号 ITI0~ITI14 的上升沿计数。这种模式可以通过设置 TSCFG6[4:0]为 0x1~0x4，0x9~0x14 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

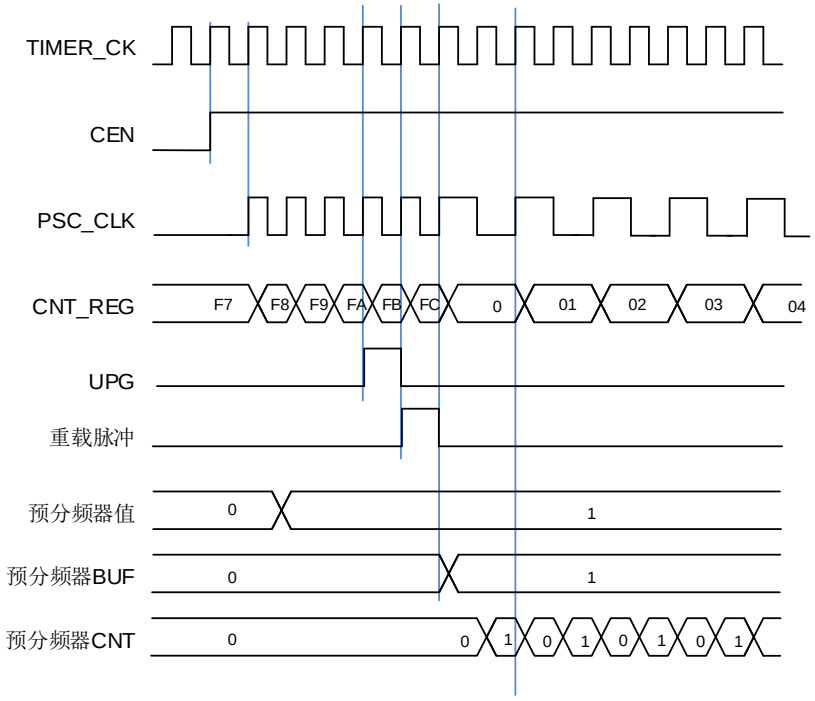
计数器预分频器可以在外部引脚 ETI 的每个上升沿或下降沿计数。这种模式可以通过设置 TIMERx_SMCFG 寄存器中的 SMC1 位为 1 来选择。另一种选择 ETI 信号作为时钟源方式是，设置 TSCFG6[4:0]为 0x8。注意 ETI 信号是通过数字滤波器采样 ETI 引脚得到的。如果选择 ETI 信号为时钟源，触发控制器包括边沿监测电路将在每个 ETI 信号上升沿产生一个时钟脉冲来为计数器预分频器提供时钟。

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 ***TIMER1_ETI 触发选择寄存器（TRIGSEL_TIMER1ETI）***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 24-52. 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 24-53. 向上计数时序图，PSC=0/2 和 *图 24-54. 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 24-53. 向上计数时序图，PSC=0/2

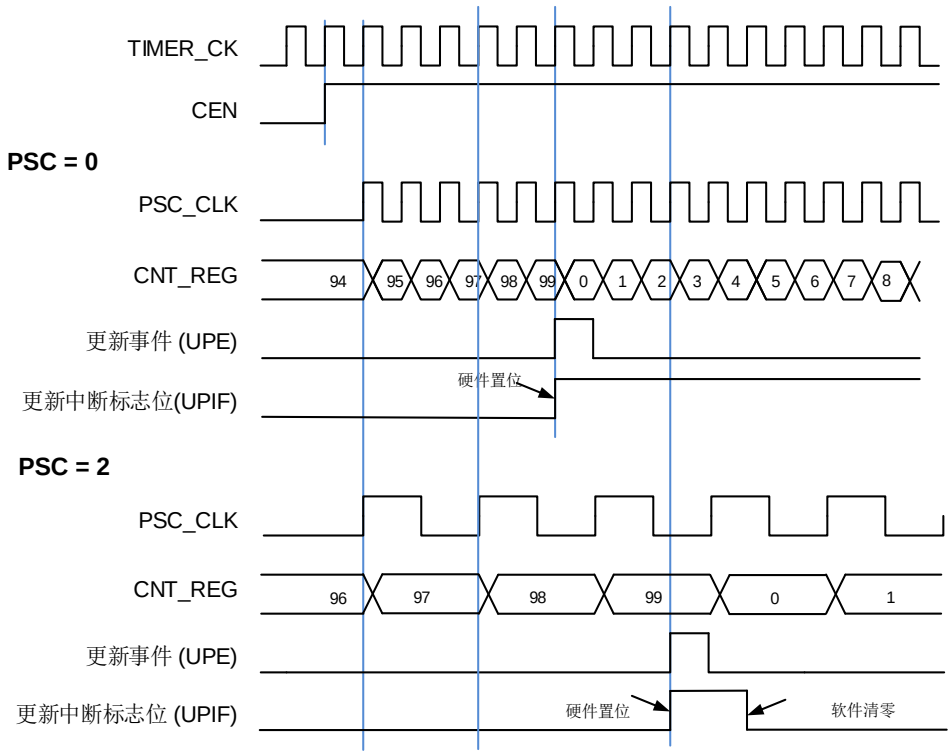
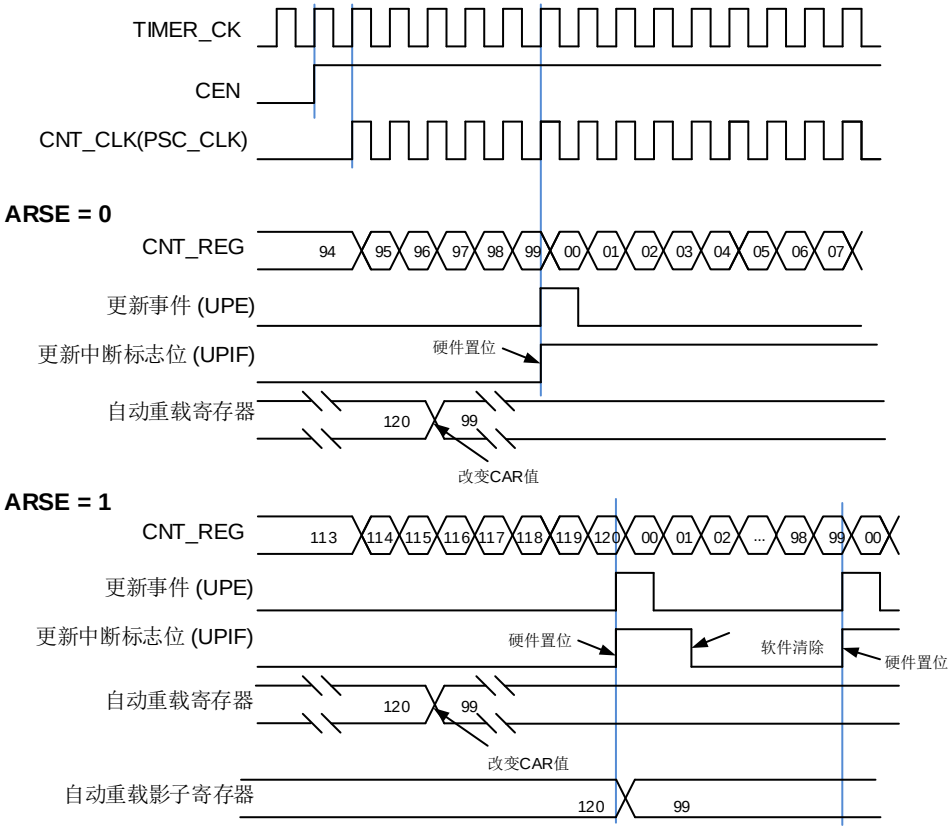


图 24-54. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



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 24-55. 向下计数时序图，PSC=0/2 和 *图 24-56. 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 24-55. 向下计数时序图，PSC=0/2

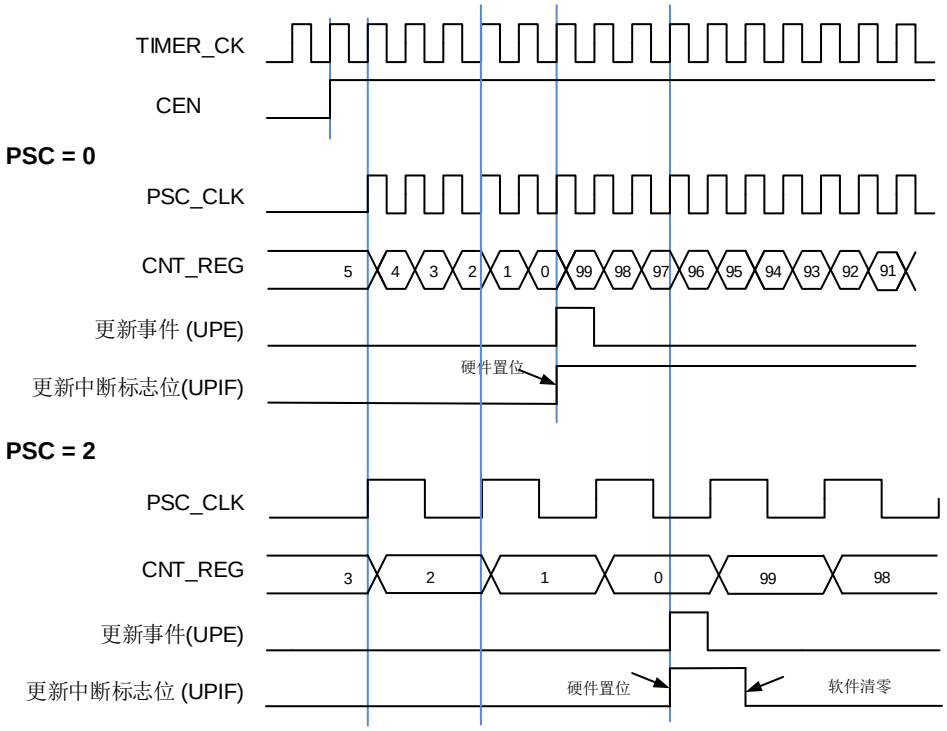
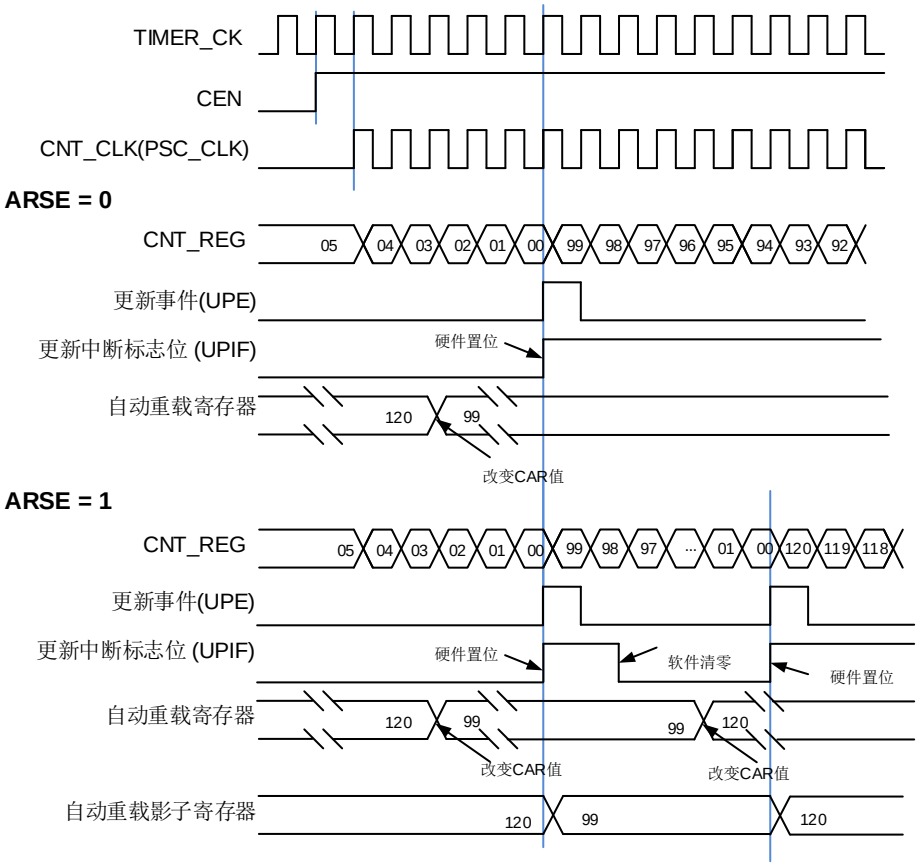


图 24-56. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值



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。

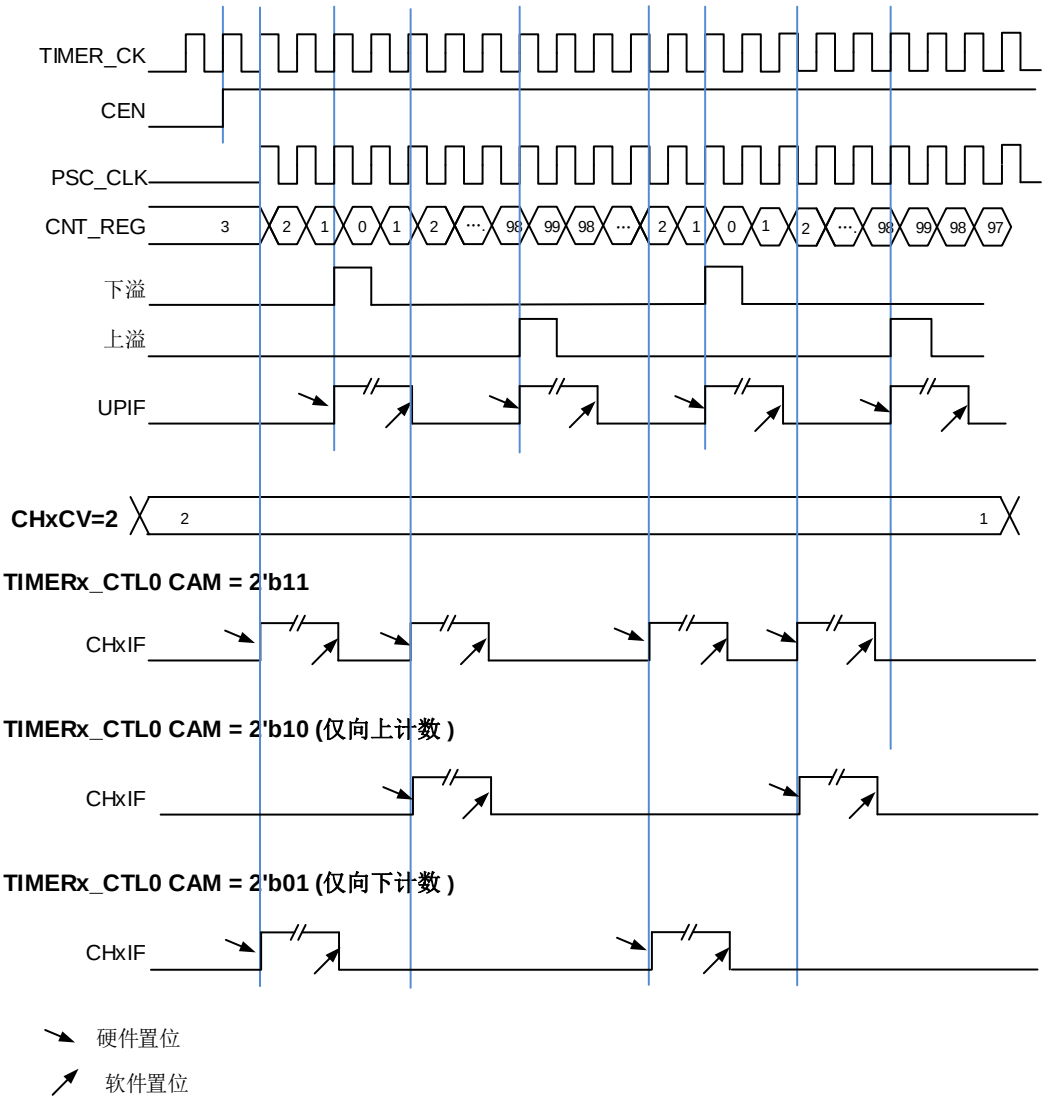
将 TIMERx_SWEVG 寄存器的 UPG 位置 1 可以初始化计数值为 0，并产生一个更新事件，而无需考虑计数器在中央模式下是向上计数还是向下计数。

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 24-57. 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 24-57. 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 24-57. 中央计数模式计数器时序图



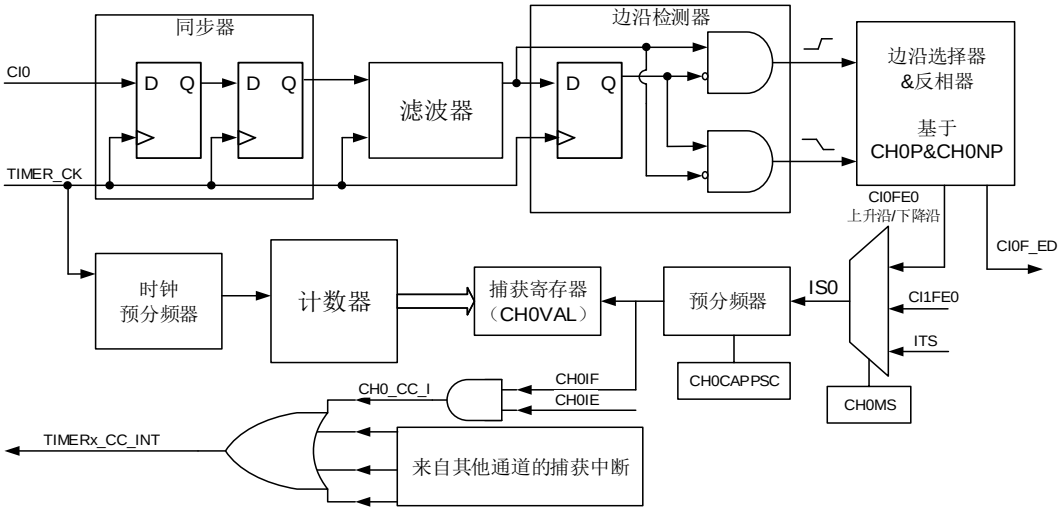
捕获/比较通道

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 24-58. 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 24-59. 输出比较逻辑（x=0,1,2,3）

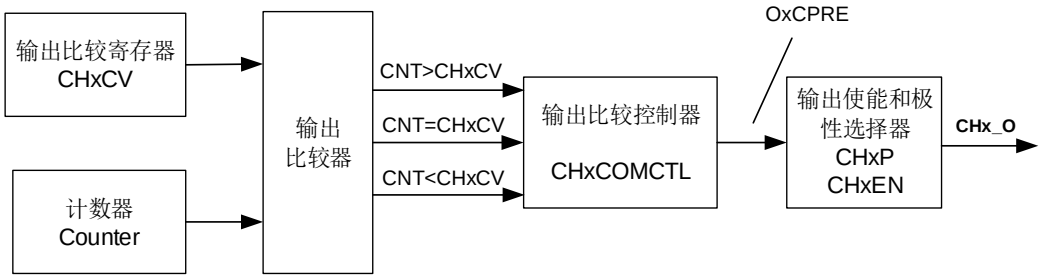


*图24-59. 输出比较逻辑（x=0,1,2,3）*给出了输出比较的逻辑电路。通道输出信号CHx_O与OxCPRE信号（详情请见*通道输出参考信号*）的关系描述如下：OxCPRE信号高电平有效，CHx_O 的输出情况与OxCPRE信号，CHxP位和CHxEN位有关（具体情况请见TIMERx_CHCTL2寄存器中的描述）。例如，当设置CHxP=0（CHx_O高电平有效，与OxCPRE输出极性相同）、CHxEN=1（CHx_O输出使能）时：

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：

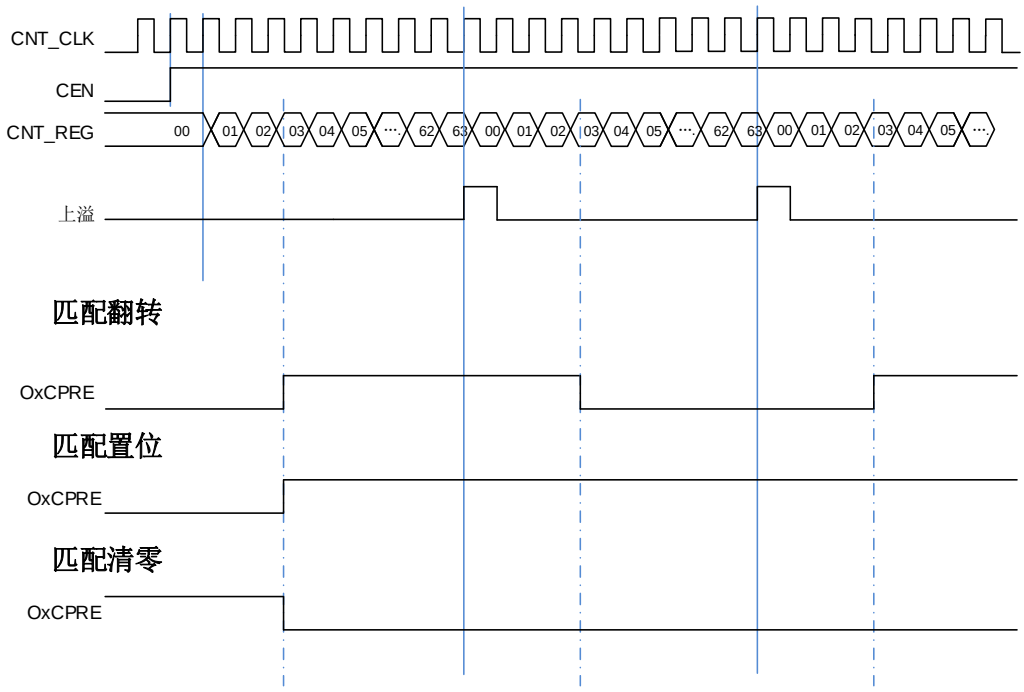
CHxVAL可以在运行时根据你所期望的波形而改变。

第五步：设置CEN位使能定时器。

*图 24-60. 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 24-60. 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 24-61. 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 24-62. 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 24-61. EAPWM 时序图

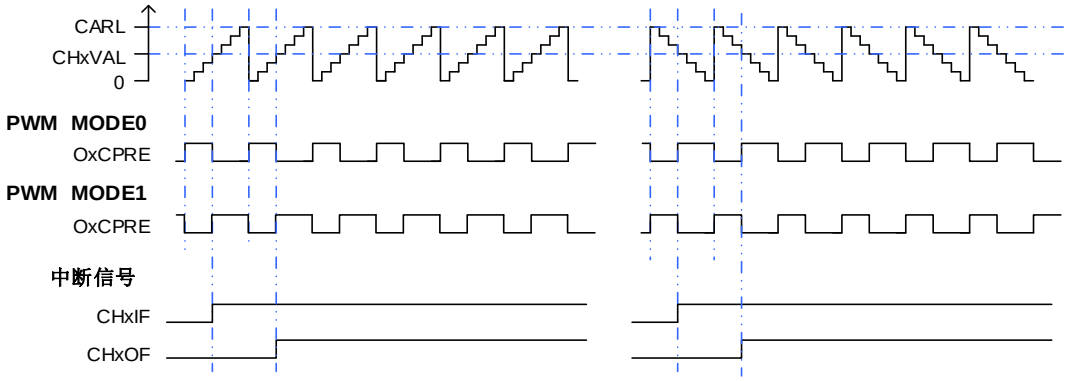
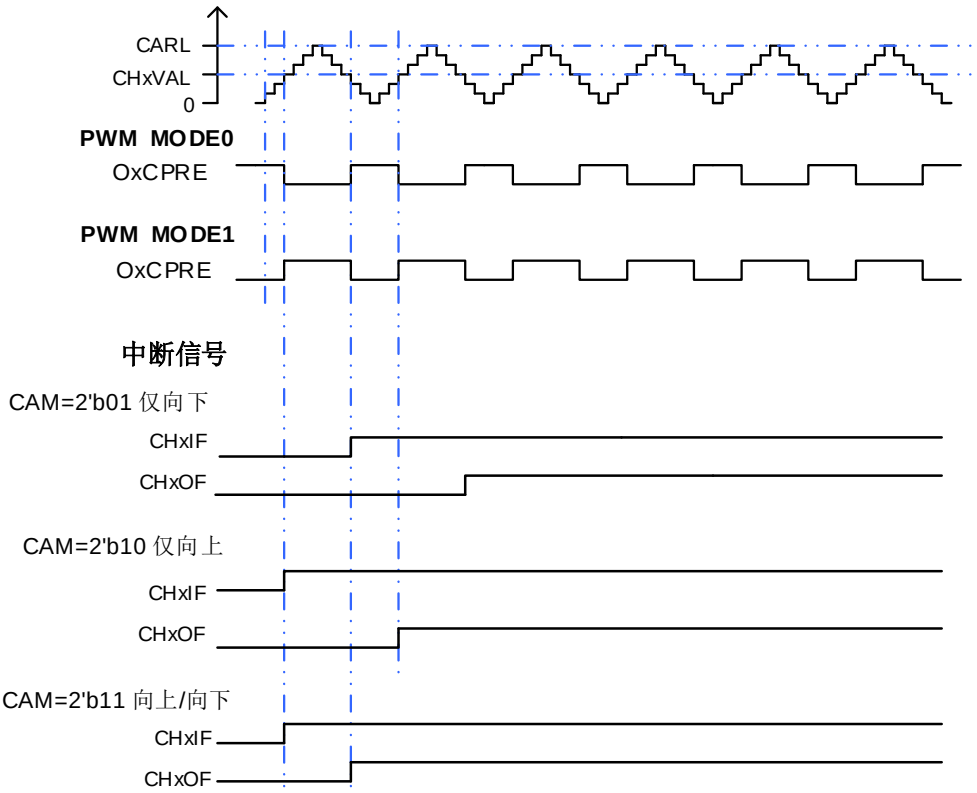


图 24-62. CAPWM 时序图



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 24-10. 复合 PWM 脉冲宽度*计算。

表 24-10. 复合 PWM 脉冲宽度

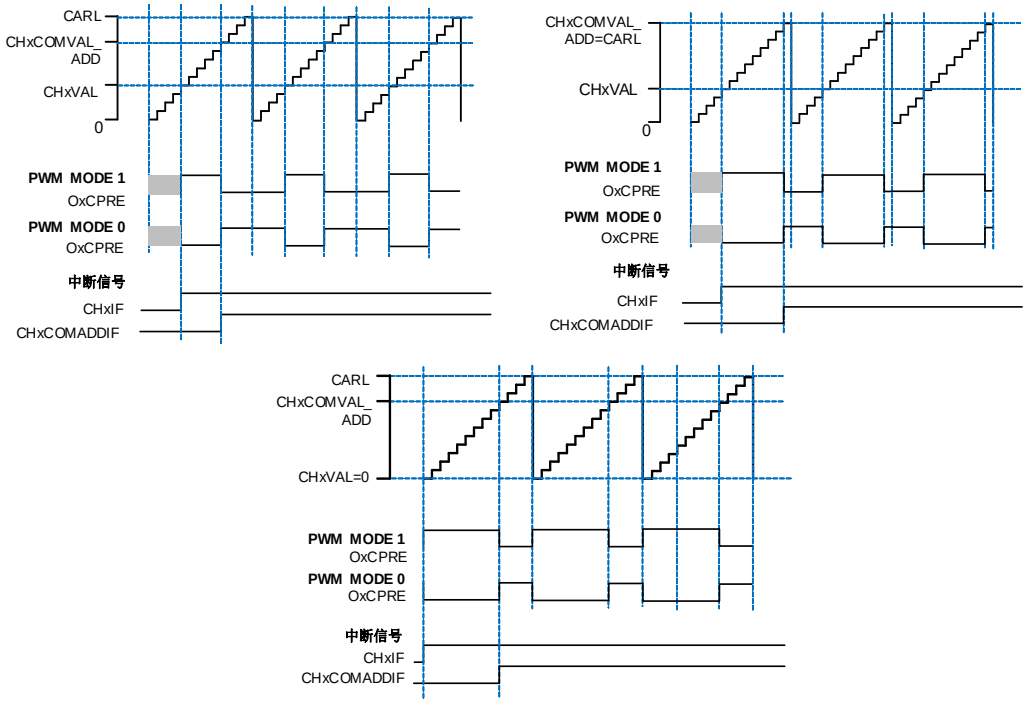
条件	模式	PWM 脉冲宽度
CHxVAL < CHxCOMVAL_ADD ≤ CARL	PWM 模式 0	（CARL + 0x0001） + （CHxVAL – CHxCOMVAL_ADD）
	PWM 模式 1	（CHxCOMVAL_ADD – CHxVAL）
CHxCOMVAL_ADD < CHxVAL ≤ CARL	PWM 模式 0	（CHxVAL - CHxCOMVAL_ADD）
	PWM 模式 1	（CARL + 0x0001） + （CHxCOMVAL_ADD – CHxVAL）
（CHxVAL = CHxCOMVAL_ADD ≤ CARL）或（CHxVAL > CARL > CHxCOMVAL_ADD）	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：

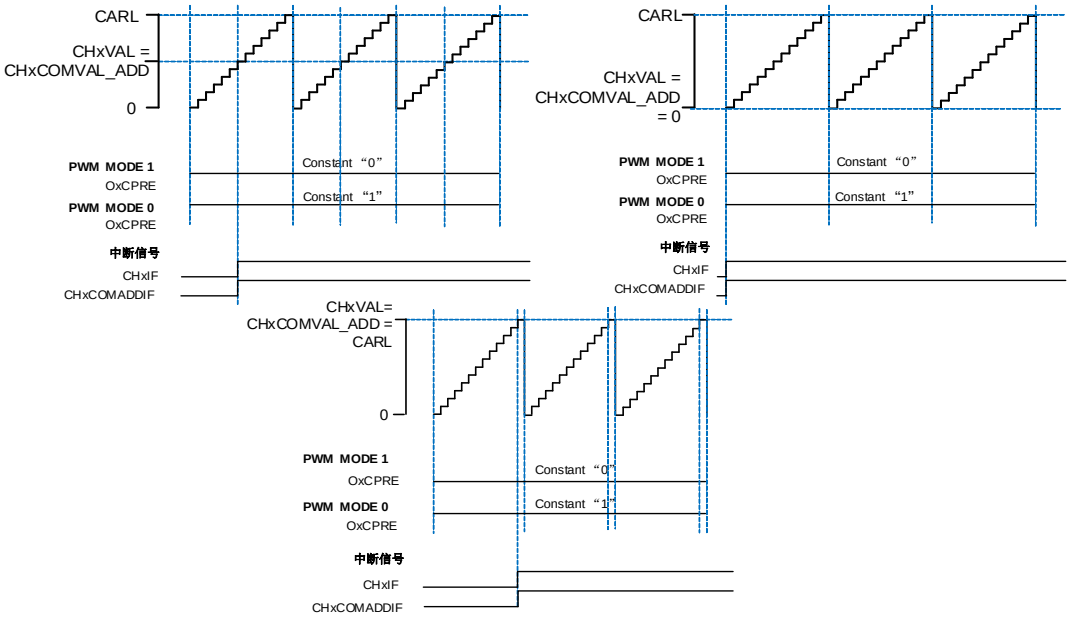
1） CHxVAL < CHxCOMVAL_ADD，CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 24-63. 通道 x 输出 PWM（CHxVAL < CHxCOMVAL_ADD）



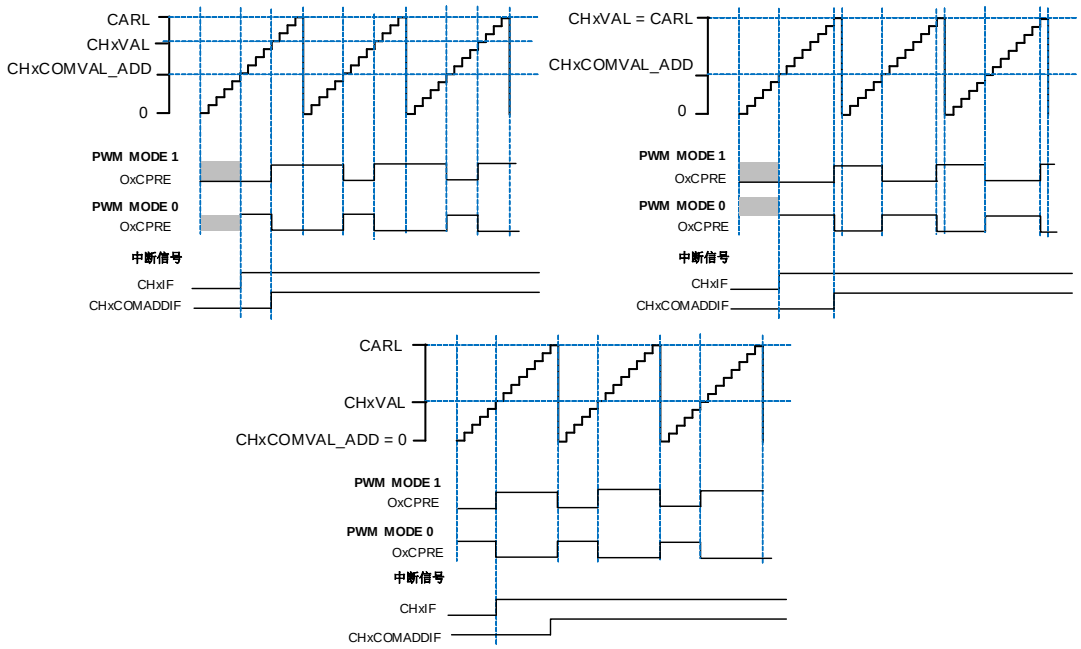
2） CHxVAL = CHxCOMVAL_ADD，CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 24-64. 通道 x 输出 PWM（CHxVAL = CHxCOMVAL_ADD）



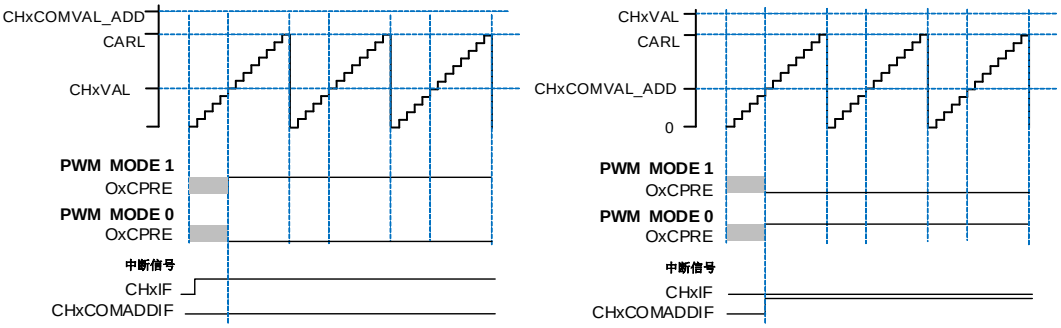
3） CHxVAL > CHxCOMVAL_ADD，CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 24-65. 通道 x 输出 PWM（CHxVAL > CHxCOMVAL_ADD）



4） CHxVAL或CHxCOMVAL_ADD值大于CARL。

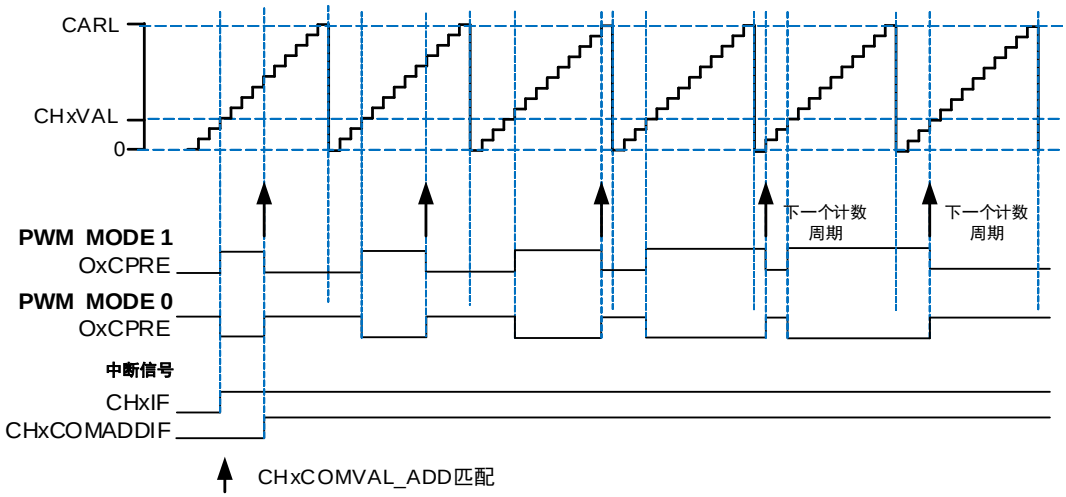
图 24-66. 通道 x 输出 PWM（CHxVAL 或 CHxCOMVAL_ADD > CARL）



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24-67. 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。

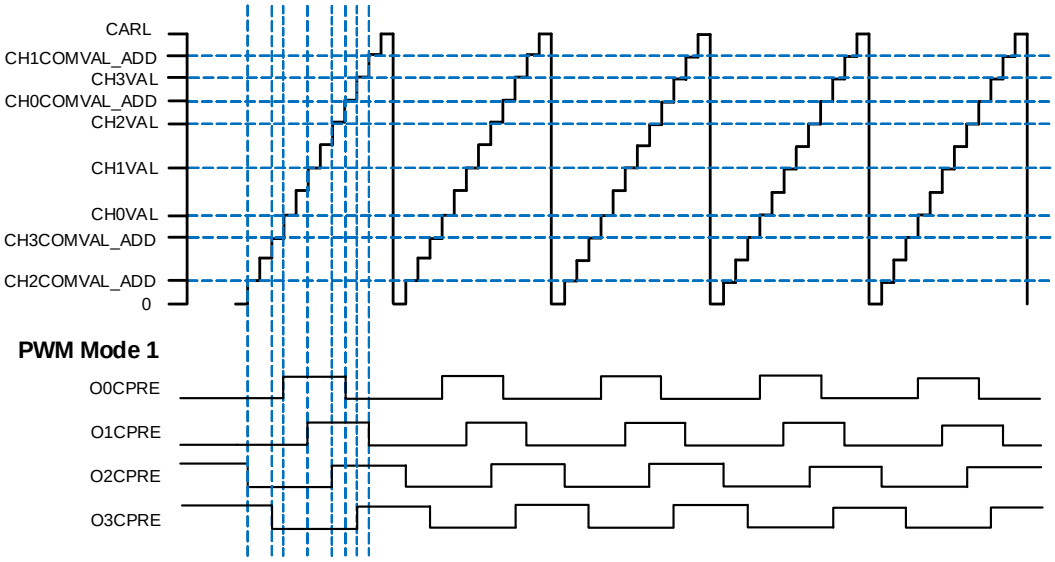
在某些情况下，CHxCOMVAL_ADD的匹配事件可以发生在下一个计数周期（CHxCOMVAL_ADD值在计数器到达CHxVAL值之后被写入，且CHxCOMVAL_ADD值小于或者等于CHxVAL值）。

图 24-67. 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 24-68. 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 24-69. 边沿对齐模式下 CHx_O 输出脉冲（CHxOMPSEL≠2'b00）

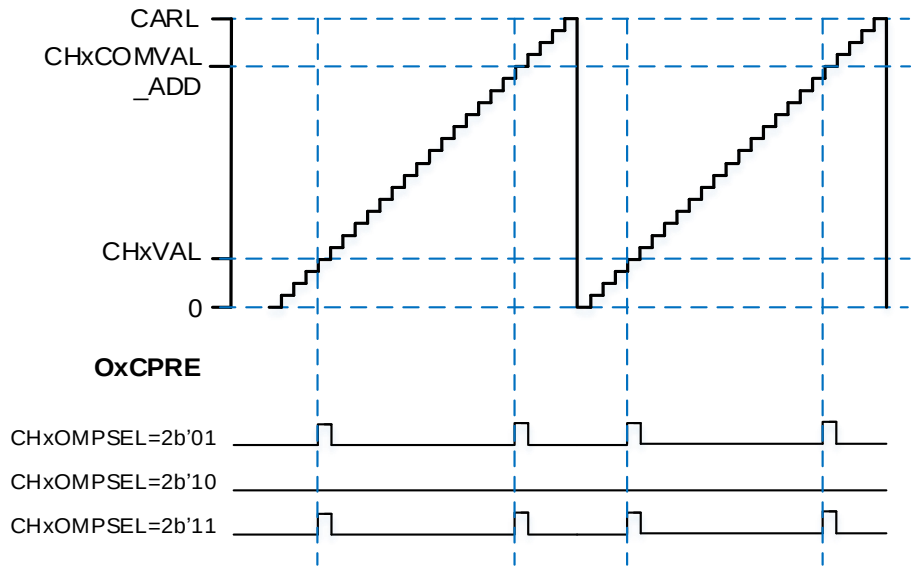
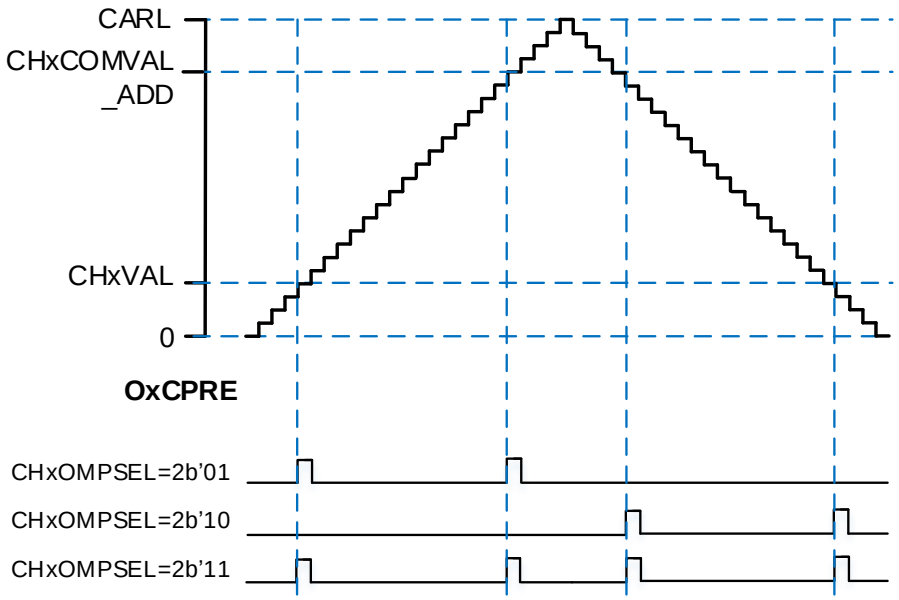


图 24-70. 中央对齐模式下 CHx_O 输出脉冲（CHxOMPSEL≠2'b00）



通道输出参考信号

根据*图 24-59. 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。

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，或着可以同时有 CI0 和 CI1，通过设置 TSCFGy[4:0](y=0..2) != 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 24-11. 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 24-11. 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

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 24-71. 译码器接口模式下计数器运行例子

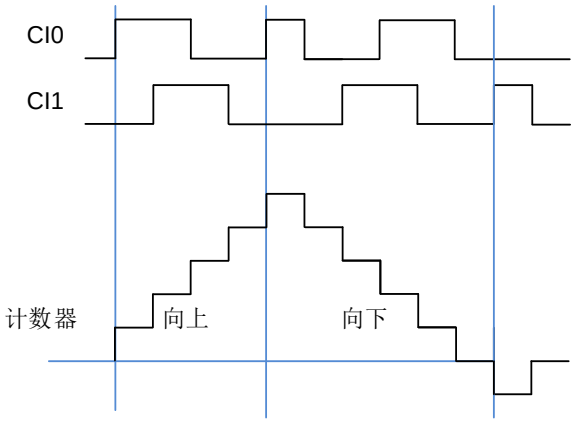
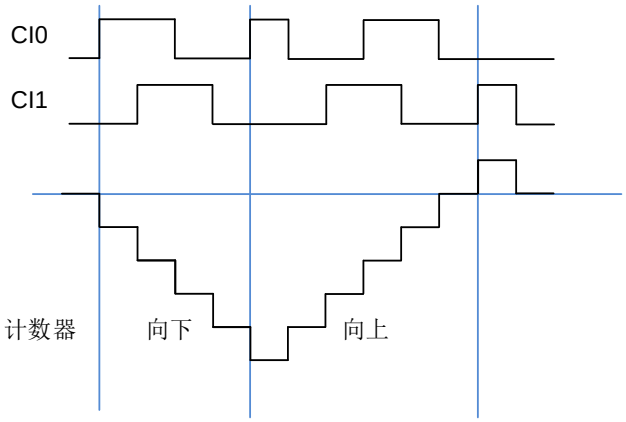


图 24-72. CI0FE0 极性反相的译码器接口模式下的例子



正交译码器信号检测

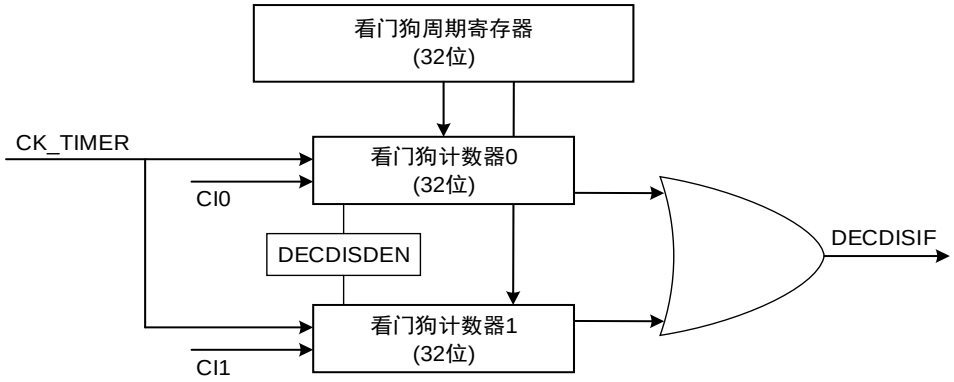
支持两种正交译码器信号检测：信号跳变检测和断线检测。

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24-73. 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 24-73. 正交译码器信号断线检测框图

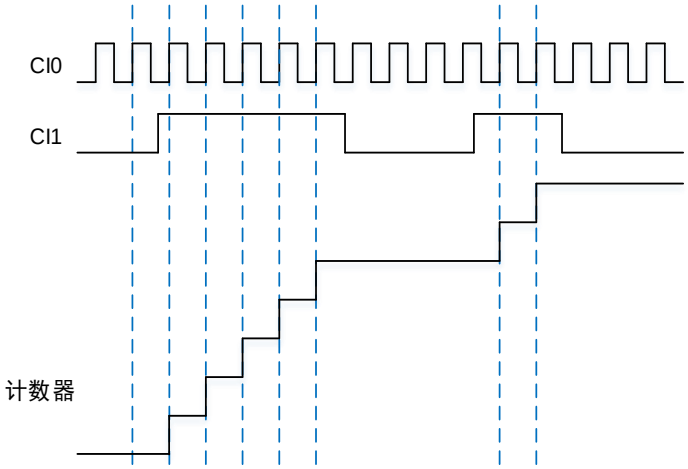


非正交译码

非正交译码器功能有两种模式：非正交译码器模式0和非正交译码器模式1，通过设置TSCFGy[4:0](y=8,9) != 5'b00000来选择。这两种计数模式下的输入源有两个：CI0和CI1。

使用非正交译码器模式0时，CI0作为计数脉冲，CI1作为计数选择信号。CH1P=0时，只有当CI1输入信号为高电平时，计数器才会在CI0输入信号的上升沿向上计数；CH1P=1时，只有当CI1输入信号为低电平时，计数器才会在CI0输入信号的上升沿向上计数。具体细节可见*图24-74. 非正交译码器模式0计数器运行实例（CH1P=0）*。

图 24-74. 非正交译码器模式 0 计数器运行实例（CH1P=0）

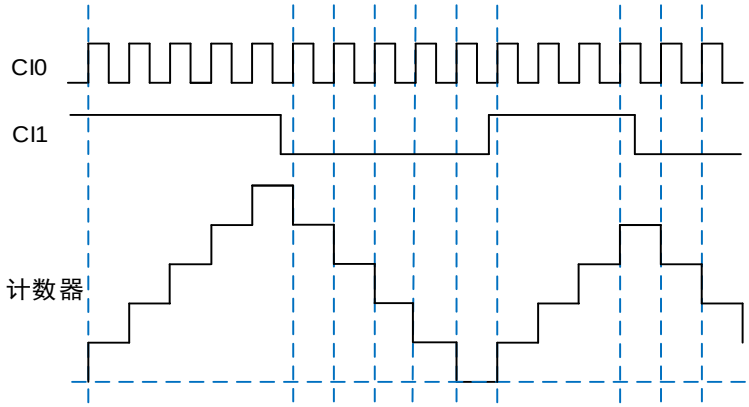


使用非正交译码器模式1时，CI0作为计数脉冲（CH0P用于选择计数边沿）；CI1作为计数方向选择信号。具体计数情况请见*表24-12. 非正交译码器模式1的计数情况*和*图24-75. 非正交译码器模式1计数器运行实例（CH0P=0）*。

表 24-12. 非正交译码器模式 1 的计数情况

CH0P	CI1电平	计数器计数情况
0	CI1为高电平	计数器在CI0输入信号的上升沿向上计数
	CI1为低电平	计数器在CI0输入信号的上升沿向下计数
1	CI1为高电平	计数器在CI0输入信号的下降沿向上计数
	CI1为低电平	计数器在CI0输入信号的下降沿向下计数

图 24-75. 非正交译码器模式 1 计数器运行实例（CH0P=0）



霍尔传感器接口功能

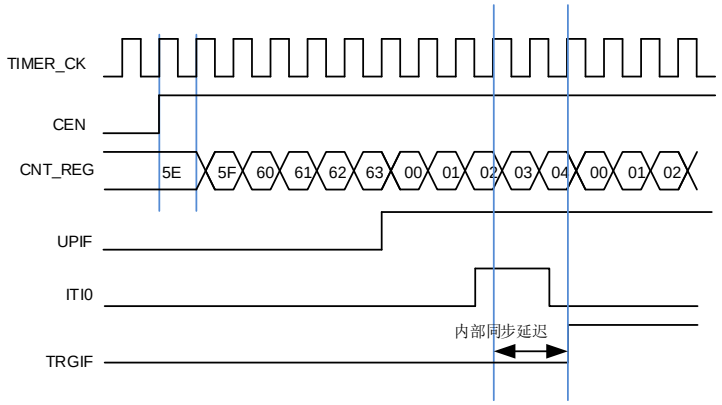
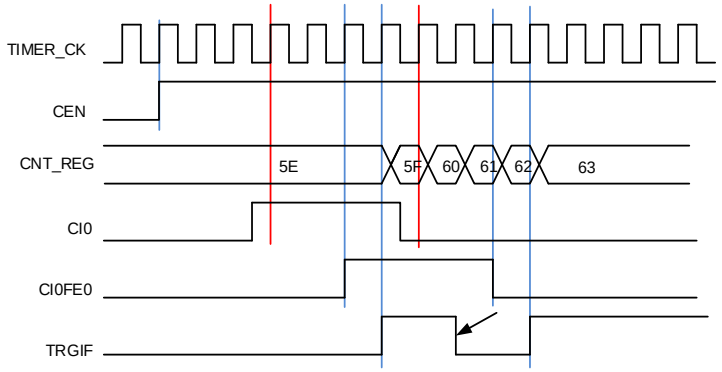
参考*高级定时器(TIMERx,x=0,7)霍尔传感器接口功能*。

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..4,22,23,30,31)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。

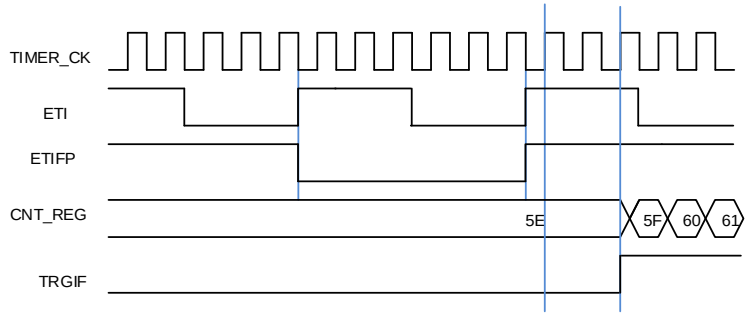
表 24-13. 从模式列表和举例（通用定时器 L0）

	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	TSCFGy[4:0] 00000：Mode disable 00001：ITI0 00010：ITI1 00011：ITI2 00100：ITI3 00101：CI0F_ED 00110：CI0FE0 00111：CI1FE1 01000：ETIFP(1) 01001：ITI4	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

<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></td><td>01010：ITI5
01100：ITI7
01110：ITI9
01111：ITI10
10000：ITI11
10001：ITI12
10010：ITI13
10011：ITI14</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 24-76. 复位模式
 
</td></tr>
<tr><td rowspan="2">例2</td><td>暂停模式
当触发输入为低的时候，计数器暂停计数</td><td>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 colspan="4">图 24-77. 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>
</table>

	模式选择	触发源选择	极性选择	滤波和预分频

图 24-78. 事件模式



(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）***模块。

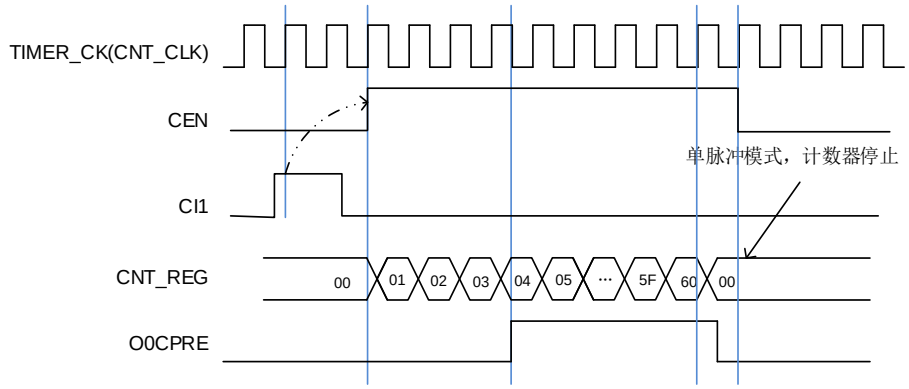
单脉冲模式

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。

图 24-79. 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，通道输出参考信号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..4,22,23,30,31)寄存器中的 TSCFG7[4:0] != 5'b00000；
- 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。

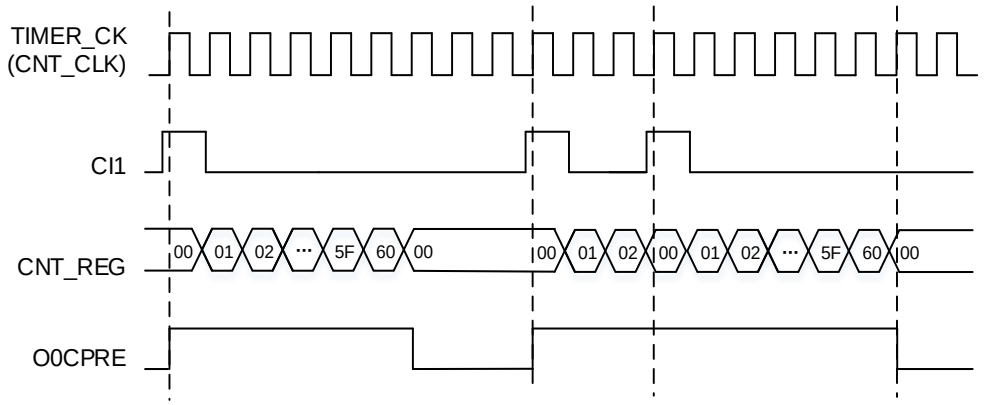
在可延时单脉冲模式1下，OxCPRE的输出情况类似与PWM模式1。在向上计数模式时，OxCPRE先输出无效电平，当外部触发事件发生时，立即输出有效电平，当下一次更新事件发生时，再变成无效电平；在向下计数模式时，OxCPRE先输出有效电平，当外部触发事件发生时，立即输出无效电平，当下一次更新事件发生时，再变成有效电平。

注意：

3） 不能使用中央对齐模式，TIMERx_CTL0 寄存器中的 CAM[1:0]=2'b00；

4） 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 24-80. 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®-M7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。

24.2.5. TIMERx 寄存器（x=1/2/3/4/22/23/30/31）

TIMER1基地址：0x4000 0000

TIMER2基地址：0x4000 0400

TIMER3基地址：0x4000 0800

TIMER4基地址：0x4000 0C00

TIMER22基地址：0x4000 E000

TIMER23基地址：0x4000 E400

TIMER30基地址：0x4000 E800

TIMER31基地址：0x4000 EC00

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				UPIFBUEN	保留	CKDIV[1:0]		ARSE	CAM[1:0]		DIR	SPM	UPS	UPDIS	CEN
				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	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

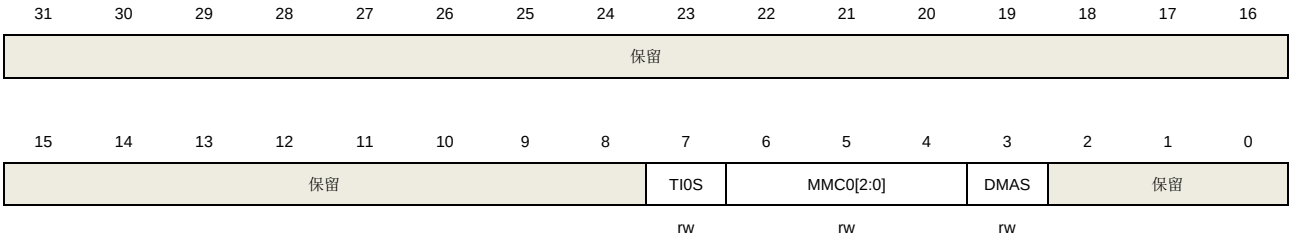
		1：使能 TIMERx_CAR 寄存器的影子寄存器
6:5	CAM[1:0]	计数器对齐模式选择 00：无中央对齐模式（边沿对齐模式）。DIR 位指定了计数方向。 01：中央对齐向下计数置 1 模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0 寄存器中 CHxMS=00），只有在向下计数时，通道的比较中断标志置 1 10：中央对齐向上计数置 1 模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0 寄存器中 CHxMS=00），只有在向上计数时，通道的比较中断标志置 1 11：中央对齐上下计数置 1 模式。计数器在中央计数模式计数，通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
31:8	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 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：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 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 TRGO0
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															

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						
rw	rw	rw		rw				rw							

位/位域	名称	描述
31:16	保留	必须保持复位值
15	ETP	外部触发极性 该位指定 ETI 信号的极性 0：ETI 高电平或上升沿有效 1：ETI 低电平或下降沿有效
14	SMC1	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 5b'01000。 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
6:0	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										DECDISIE	DECJIE
rw	rw	rw	rw											rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TRGDEN	保留	CH3DEN	CH2DEN	CH1DEN	CH0DEN	UPDEN	保留	TRGIE	保留	CH3IE	CH2IE	CH1IE	CH0IE	UPIE
	rw		rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

位/位域	名称	描述
31	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。
27:18	保留	必须保持复位值
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	保留	必须保持复位值
14	TRGDEN	触发 DMA 请求使能 0：禁止触发 DMA 请求 1：使能触发 DMA 请求
13	保留	必须保持复位值.
12	CH3DEN	通道 3 比较/捕获 DMA 请求使能 0：禁止通道 3 比较/捕获 DMA 请求 1：使能通道 3 比较/捕获 DMA 请求
11	CH2DEN	通道 2 比较/捕获 DMA 请求使能 0：禁止通道 2 比较/捕获 DMA 请求 1：使能通道 2 比较/捕获 DMA 请求
10	CH1DEN	通道 1 比较/捕获 DMA 请求使能 0：禁止通道 1 比较/捕获 DMA 请求 1：使能通道 1 比较/捕获 DMA 请求
9	CH0DEN	通道 0 比较/捕获 DMA 请求使能 0：禁止通道 0 比较/捕获 DMA 请求 1：使能通道 0 比较/捕获 DMA 请求
8	UPDEN	更新 DMA 请求使能 0：禁止更新 DMA 请求 1：使能更新 DMA 请求

7	保留	必须保持复位值.
6	TRGIE	触发中断使能 0：禁止触发中断 1：使能触发中断
5	保留	必须保持复位值.
4	CH3IE	通道 3 比较/捕获中断使能 0：禁止通道 3 中断 1：使能通道 3 中断
3	CH2IE	通道 2 比较/捕获中断使能 0：禁止通道 2 中断 1：使能通道 2 中断
2	CH1IE	通道 1 比较/捕获中断使能 0：禁止通道 1 中断 1：使能通道 1 中断
1	CH0IE	通道 0 比较/捕获中断使能 0：禁止通道 0 中断 1：使能通道 0 中断
0	UPIE	更新中断使能 0：禁止更新中断 1：使能更新中断

中断标志寄存器（TIMERx_INTF）

地址偏移：0x10

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3COMADDIF	CH2COMADDIF	CH1COMADDIF	CH0COMADDIF	保留										DECDISIF	DECJIF
rc_w0	rc_w0	rc_w0	rc_w0											rc_w0	rc_w0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			CH3OF	CH2OF	CH1OF	CH0OF	保留		TRGIF	保留	CH3IF	CH2IF	CH1IF	CH0IF	UPIF
			rc_w0	rc_w0	rc_w0	rc_w0			rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位/位域	名称	描述
31	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。
27:18	保留	必须保持复位值
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。
15:13	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]	CH1COM ADDSEN	CH0COM ADDSEN	保留			CH1COM CTL[3]	保留							CH0COM CTL[3]
		保留	保留				保留								保留

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	保留	CH1MS[1:0]		CH0COMCEN	CH0COMCTL[2:0]			CH0COMSEN	保留	CH0MS[1:0]	
CH1CAPFLT[3:0]				CH1CAPPSC[1:0]				CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

输出比较模式：

位/位域	名称	描述
31	CH1MS[2]	通道 1 I/O 模式选择 参考 CH1MS[1:0]描述。
30	CH0MS[2]	通道 0 I/O 模式选择 参考 CH0MS[1:0]描述。
29	CH1COMADDSEN	通道 1 附加输出比较影子寄存器使能 参考 CH0COMADDSEN 描述。
28	CH0COMADDSEN	通道 0 附加输出比较影子寄存器使能 当此位被置 1，TIMERx_CH0COMV_ADD 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 0 附加比较输出影子寄存器 1：使能通道 0 附加比较输出影子寄存器 仅在单脉冲模式下（TIMERx_CTL0 寄存器的 SPM =1），可以在未确认预装载寄存器情况下使用 PWM 模式。
27:25	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	保留	必须保持复位值
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..4,22,23,30,31)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。 1010~1111：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	保留	必须保持复位值
1:0	CH0MS[1:0]	通道 0 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭（TIMERx_CHCTL2 寄存器的 CH0EN 位被清 0）时，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..4,22,23,30,31)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
31:16	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	保留			CH3COMCTL[3]	保留							CH2COMCTL[3]
		保留	保留				保留								保留
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	保留	CH3MS[1:0]		CH2COMCEN	CH2COMCTL[2:0]			CH2COMSEN	保留	CH2MS[1:0]	
CH3CAPFLT[3:0]				CH3CAPPSC[1:0]				CH2CAPFLT[3:0]				CH2CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

输出比较模式：

位/位域	名称	描述
31	CH3MS[2]	通道 3 I/O 模式选择 参考 CH3MS[1:0]描述。
30	CH2MS[2]	通道 2 I/O 模式选择 参考 CH2MS[1:0]描述。
29	CH3COMADDSEN	通道 3 附加输出比较影子寄存器使能

		参考 CH2COMADDSEN 描述。
28	CH2COMADDSEN	通道 2 附加输出比较影子寄存器使能 当此位被置 1，TIMERx_CH2COMV_ADD 寄存器的影子寄存器使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 2 附加输出/比较影子寄存器 1：使能通道 2 附加输出/比较影子寄存器 仅在单脉冲模式下（TIMERx_CTL0 寄存器的 SPM =1)，可以在未确认预装载寄存器情况下使用 PWM 模式。
27:25	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	保留	必须保持复位值
9:8	CH3MS[1:0]	通道 3 模式选择 这些位定义了通道的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..4,22,23,30,31)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。

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	保留	必须保持复位值
1:0	CH2MS[1:0]	通道 2 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭（TIMERx_CHCTL2 寄存器的 CH2EN 位被清 0）时这些位才可写。 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..4,22,23,30,31)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
31:16	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

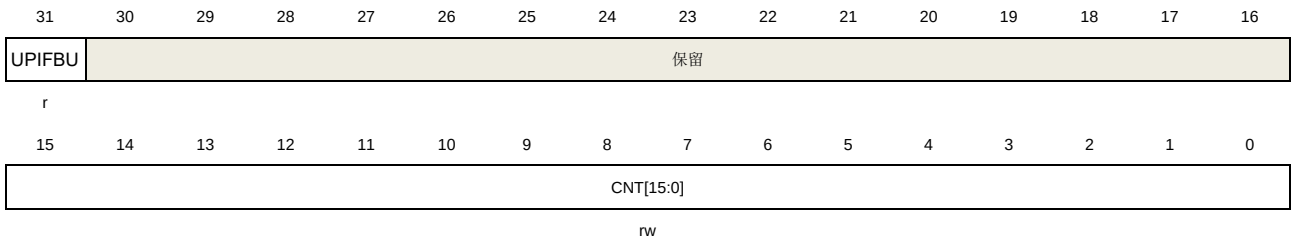
位/位域	名称	描述
31:16	保留	必须保持复位值
15	CH3NP	通道 3 互补输出极性 参考 CH0NP 描述。
14	保留	必须保持复位值。
13	CH3P	通道 3 极性 参考 CH0P 描述。
12	CH3EN	通道 3 使能 参考 CH0EN 描述。
11	CH2NP	通道 2 互补输出极性 参考 CH0NP 描述。
10	保留	必须保持复位值
9	CH2P	通道 2 极性 参考 CH0P 描述。
8	CH2EN	通道 2 使能 参考 CH0EN 描述。
7	CH1NP	通道 1 互补输出极性 参考 CH0NP 描述。
6	保留	必须保持复位值
5	CH1P	通道 1 极性 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= 2,3,30,31）

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字（32位）访问。



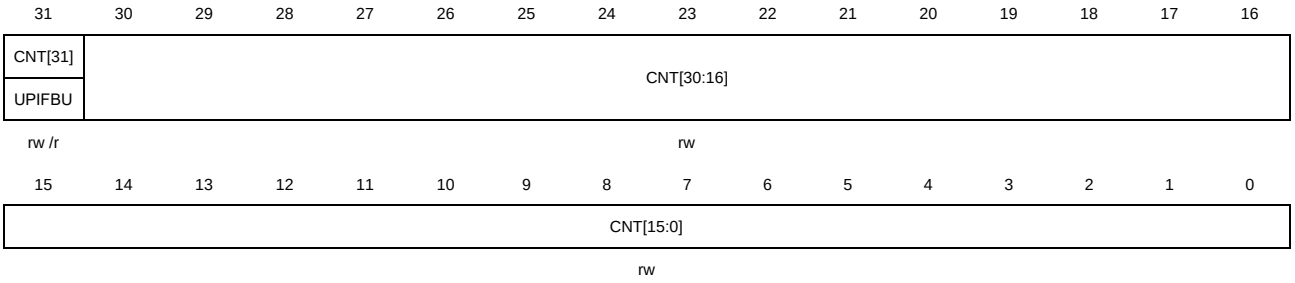
位/位域	名称	描述
31	UPIFBU	UPIF位备份 该位只读，是 TIMERx_INTF 寄存器的 UPIF 位的备份值。当 UPIFBUEN = 1 时，

		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。

计数器寄存器（TIMERx_CNT）（TIMERx,x= 1,4,22,23）

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字（32位）访问。



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。

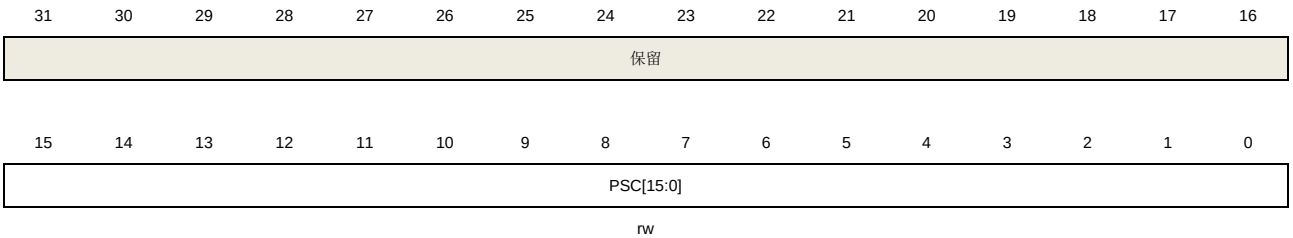
UPIFBUEN = 1:

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。

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：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
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
31:16	CARL[31:16]	计数器自动重载值（bit 16 到 bit 31） 该位域仅用于 TIMER1/ 4/ 22/ 23。
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

通道 0 捕获/比较值寄存器（TIMERx_CH0CV）

地址偏移：0x34
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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
31:16	CH0VAL[31:16]	通道 0 的捕获或比较值（bit 16 到 bit 31） 该位域仅用于 TIMER1/ 4/ 22/ 23。
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值 当通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。

当通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 1 捕获/比较值寄存器（TIMERx_CH1CV）

地址偏移：0x38
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH1VAL[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															
rw															

位/位域	名称	描述
31:16	CH1VAL[31:16]	通道 1 的捕获或比较值（bit 16 到 bit 31） 该位域仅用于 TIMER1/ 4/ 22/ 23。
15:0	CH1VAL[15:0]	通道 1 的捕获或比较值 当通道 1 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 1 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 2 捕获/比较值寄存器（TIMERx_CH2CV）

地址偏移：0x3C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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
31:16	CH2VAL[31:16]	通道 2 的捕获或比较值（bit 16 到 bit 31） 该位域仅用于 TIMER1/ 4/ 22/ 23。
15:0	CH2VAL[15:0]	通道 2 的捕获或比较值 当通道 2 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。

当通道 2 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 3 捕获/比较值寄存器（TIMERx_CH3CV）

地址偏移：0x40
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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
31:16	CH3VAL[31:16]	通道 3 的捕获或比较值（bit 16 到 bit 31） 该位域仅用于 TIMER1/ 4/ 22/ 23。
15:0	CH3VAL[15:0]	通道 3 的捕获或比较值 当通道 3 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 3 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

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（bit 16到bit 31） 该位域仅用于TIMER1/ 4/ 22/ 23。
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
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（bit 16到bit 31） 该位域仅用于TIMER1/ 4/ 22/ 23。
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
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（bit 16到bit 31） 该位域仅用于TIMER1/ 4/ 22/ 23。
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
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（bit 16到bit 31） 该位域仅用于TIMER1/ 4/ 22/ 23。
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：0x0000 0000

该寄存器只能按字（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	保留								DECDISD EN	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

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。

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’b100101：传输 38 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															

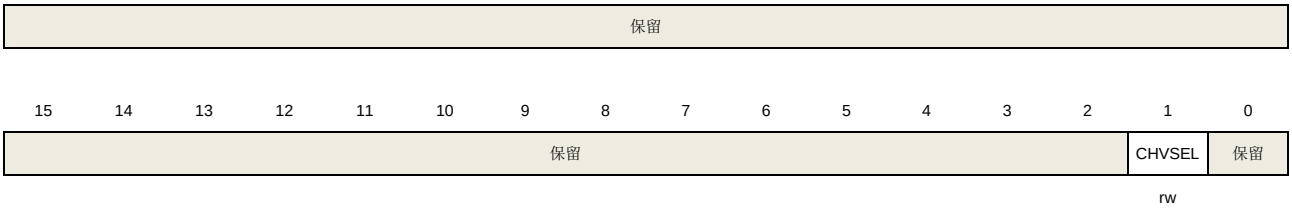
位/位域	名称	描述
31:0	DMATB [31:0]	DMA 发送缓冲 对这个寄存器的读或写，（起始地址+传输次数*4）地址范围内的寄存器会被访问 传输次数由硬件计算，范围为 0 到 DMATC。

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。

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



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

24.3. 通用定时器 L3（TIMERx, x=14,40,41,42,43,44）

24.3.1. 简介

通用定时器 L3（TIMER14/40~44）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。

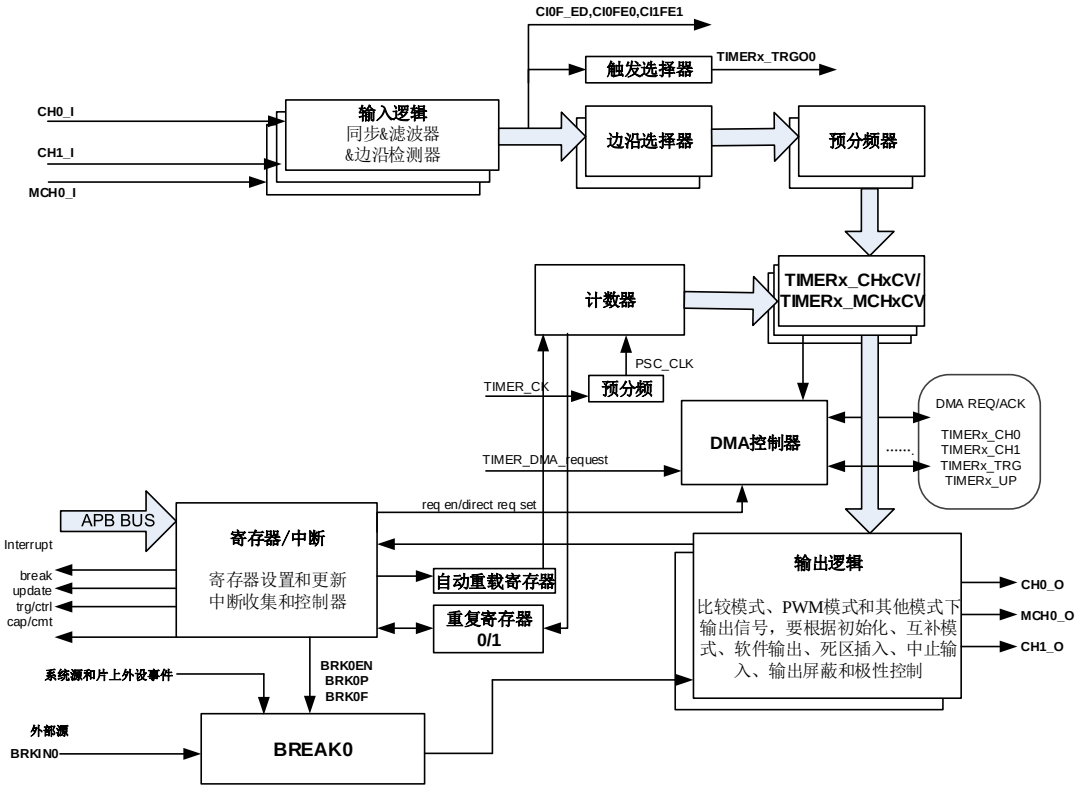
24.3.2. 主要特性

- 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。

24.3.3. 结构框图

*图 24-81. 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 24-81. 通用定时器 L3 结构框图



24.3.4. 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=14,40,41,42,43,44)寄存器中的 TSCFGy[4:0] (y=3..7,15)位域。

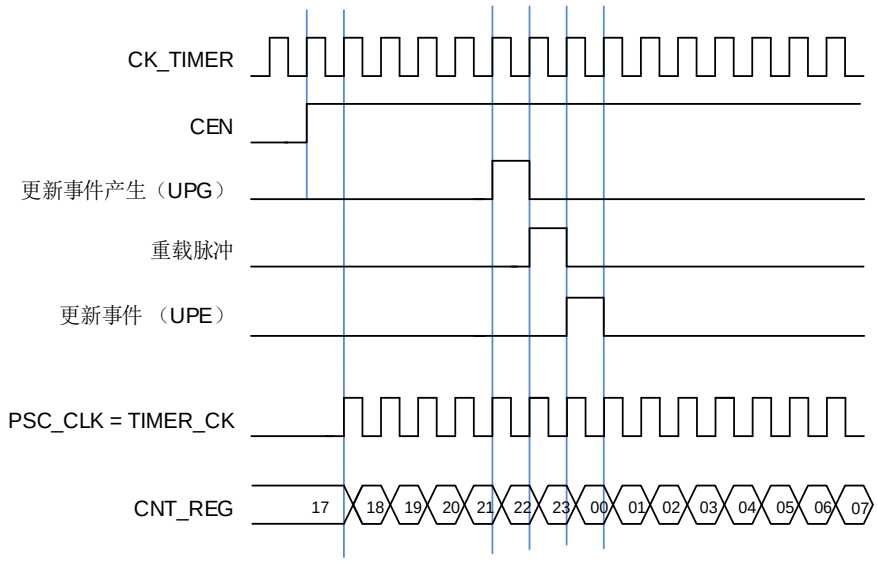
- 当 SYSCFG_TIMERxCFG(x=14,40,41,42,43,44) 寄存器中的 TSCFGy[4:0]=5'b00000(y=3..7,15)，定时器选择内部时钟源（连接到RCU模块的 CK_TIMER）

如果 SYSCFG_TIMERxCFG(x=14,40,41,42,43,44)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=14,40,41,42,43,44)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 24-82. 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

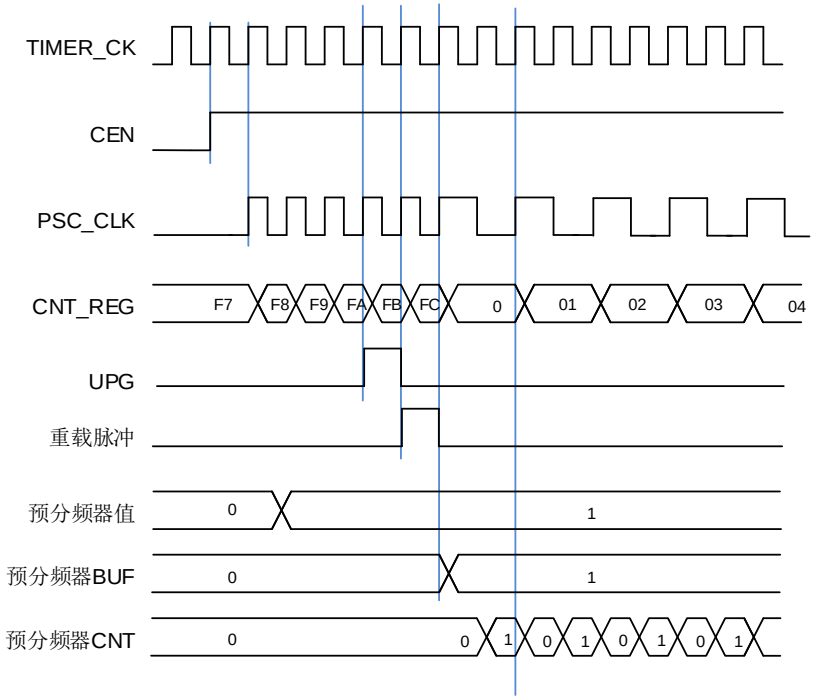
计数器预分频器可以在 TIMERx_CI0/ TIMERx_CI1 引脚的每个上升沿或下降沿计数。这种模式可以通过设置 TSCFG6[4:0]为 0x5~0x7。

计数器预分频器也可以在内部触发信号 ITI0/1/2/3 的上升沿计数。这种模式可以通过设置 TSCFG6[4:0]为 0x1~0x4。

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 24-83. 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 24-84. 向上计数时序图，PSC=0/2 和 *图 24-85. 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 24-84. 向上计数时序图，PSC=0/2

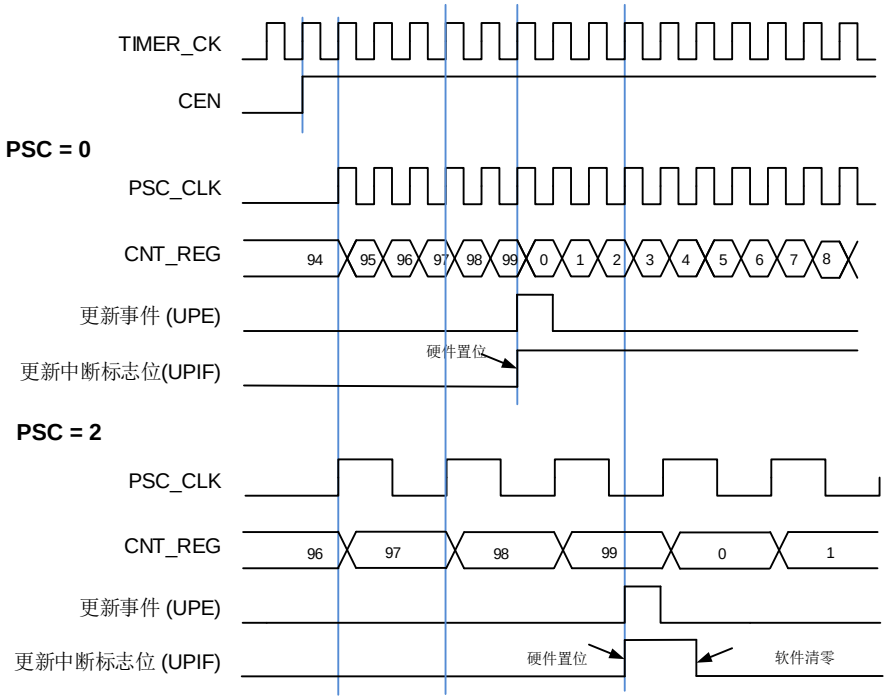
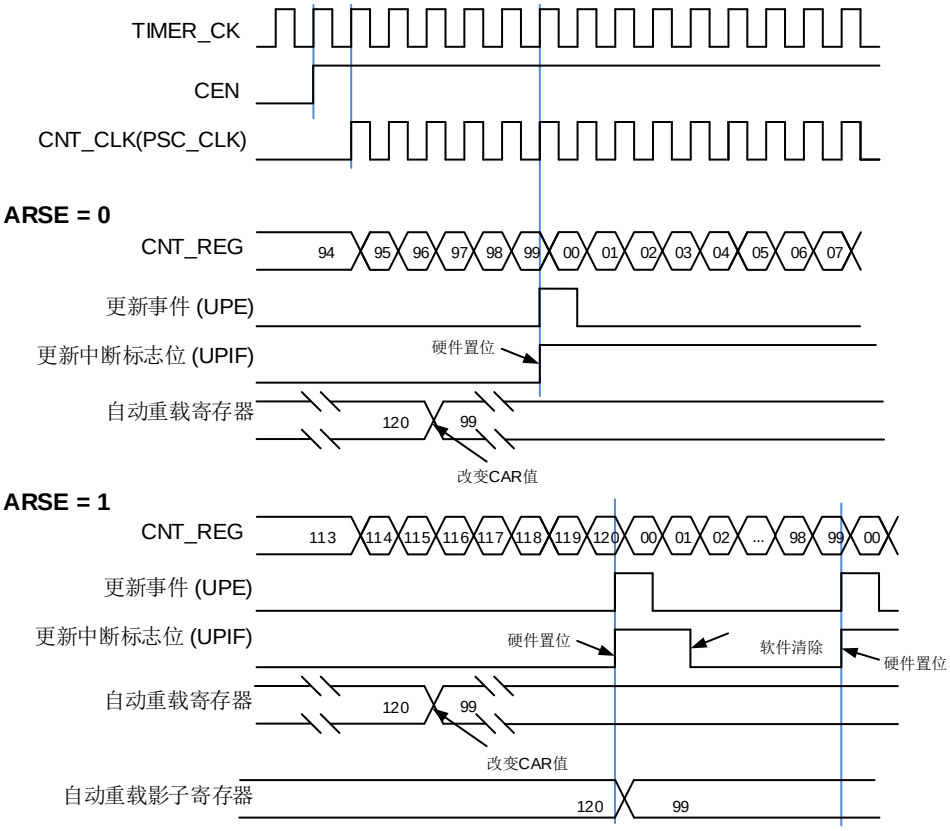


图 24-85. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



重复计数器

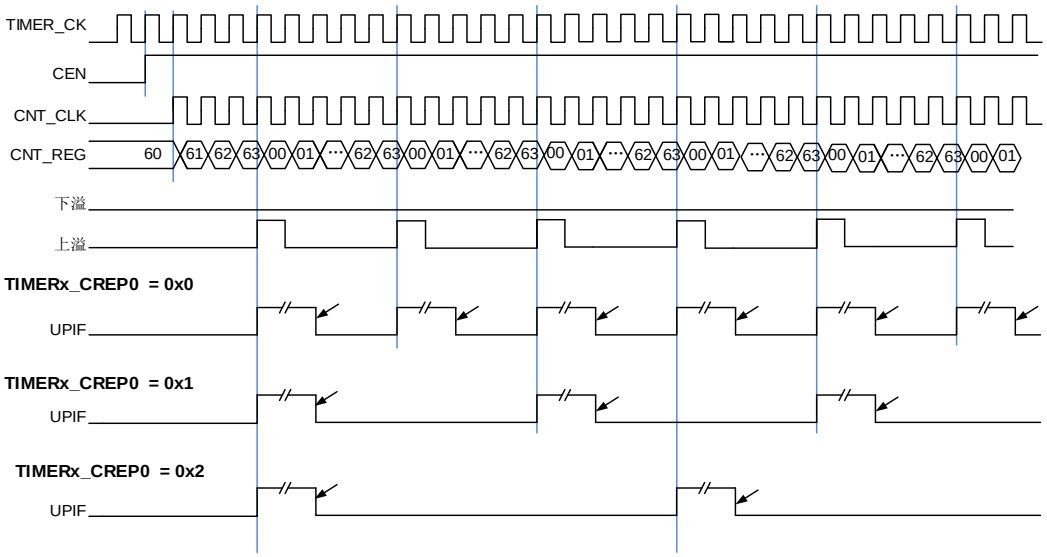
通用L3定时器有两个重复寄存器TIMERx_CREP0/1，可通过配置TIMERx_CFG寄存器中的

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 24-86. 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 24-87. 通道 0 输入捕获逻辑

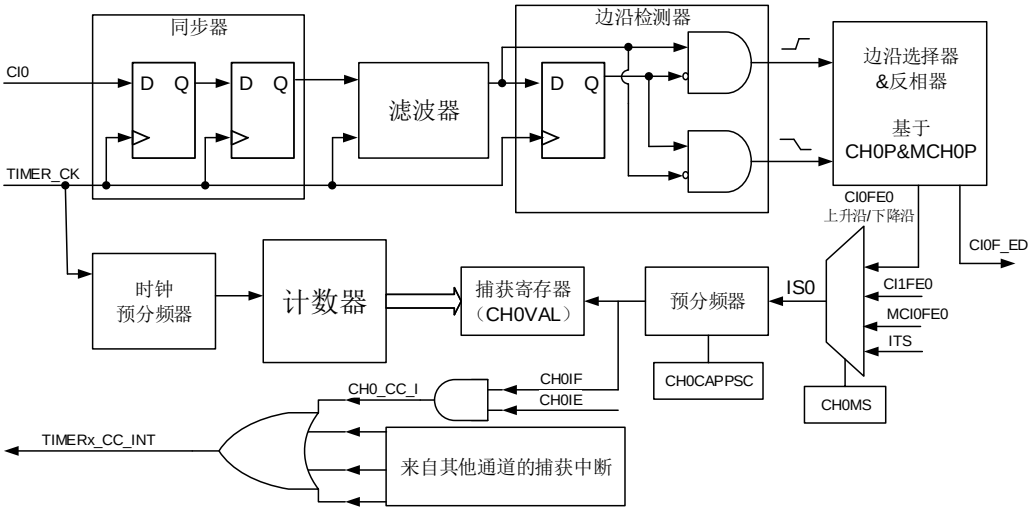
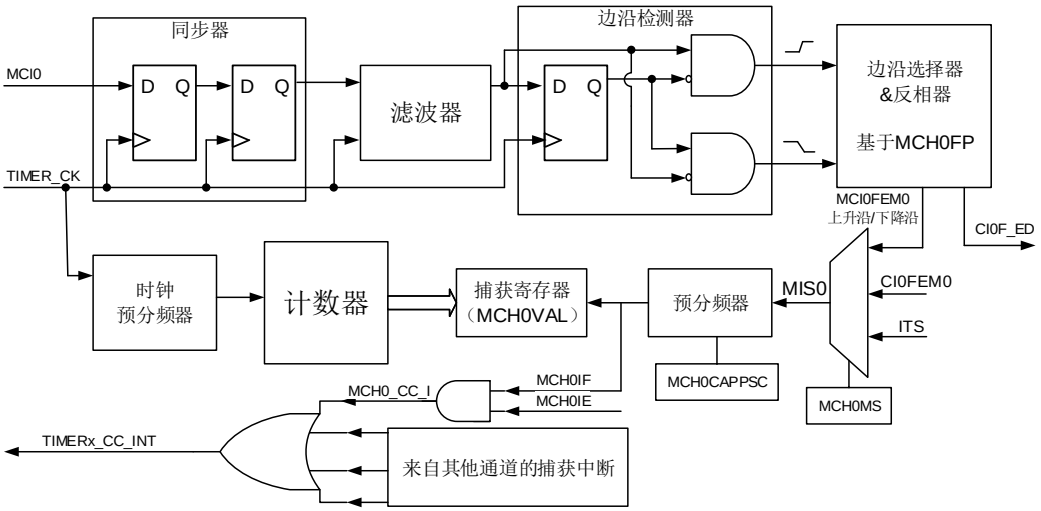


图 24-88. 多模式通道 0 输入捕获逻辑



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 24-89. 输出比较逻辑（当 MCHxMSEL = 2'00 时，x=0），*图 24-90. 输出比较逻辑（当 MCHxMSEL = 2'11 时，x=0）*和*图 24-91. 输出比较逻辑（x=1）*给出了通道的输出比较逻辑。

图 24-89. 输出比较逻辑（当 MCHxMSEL = 2'00 时，x=0）

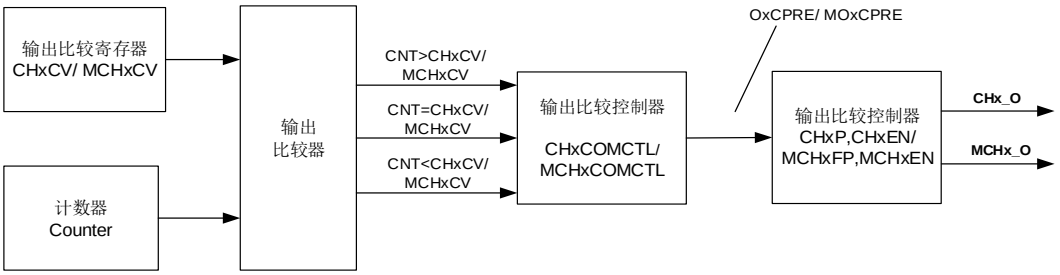


图 24-90. 输出比较逻辑（当 MCHxMSEL = 2'11 时，x=0）

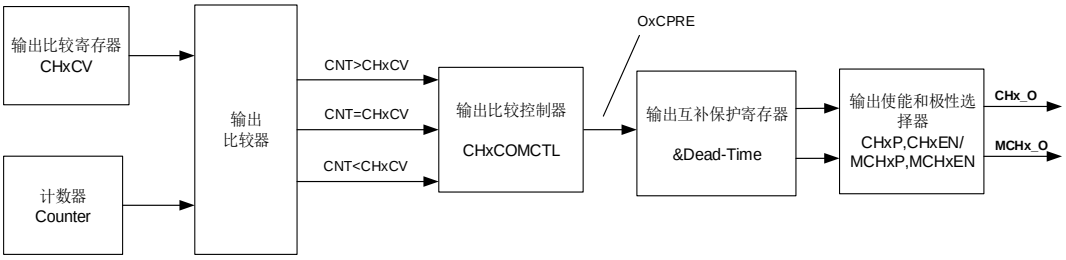
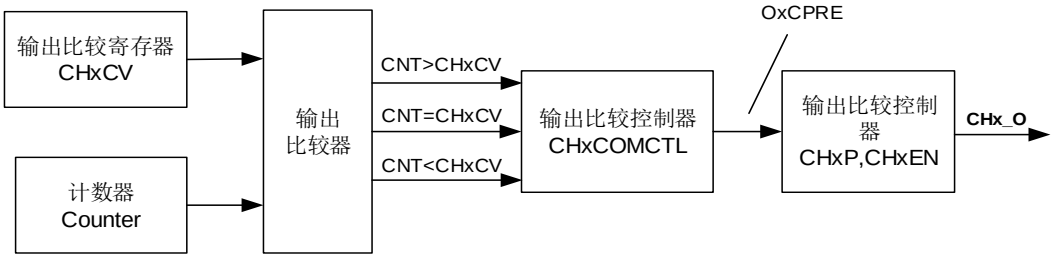


图 24-91. 输出比较逻辑（x=1）



通道输出信号CHx_O/MCHx_O与OxCPRE/ MOxCPRE信号（详情请见*通道输出参考信号*）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24-89. 输出比较逻辑（当MCHxMSEL = 2'00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24-90. 输出比较逻辑（当MCHxMSEL = 2'11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_CCHP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。

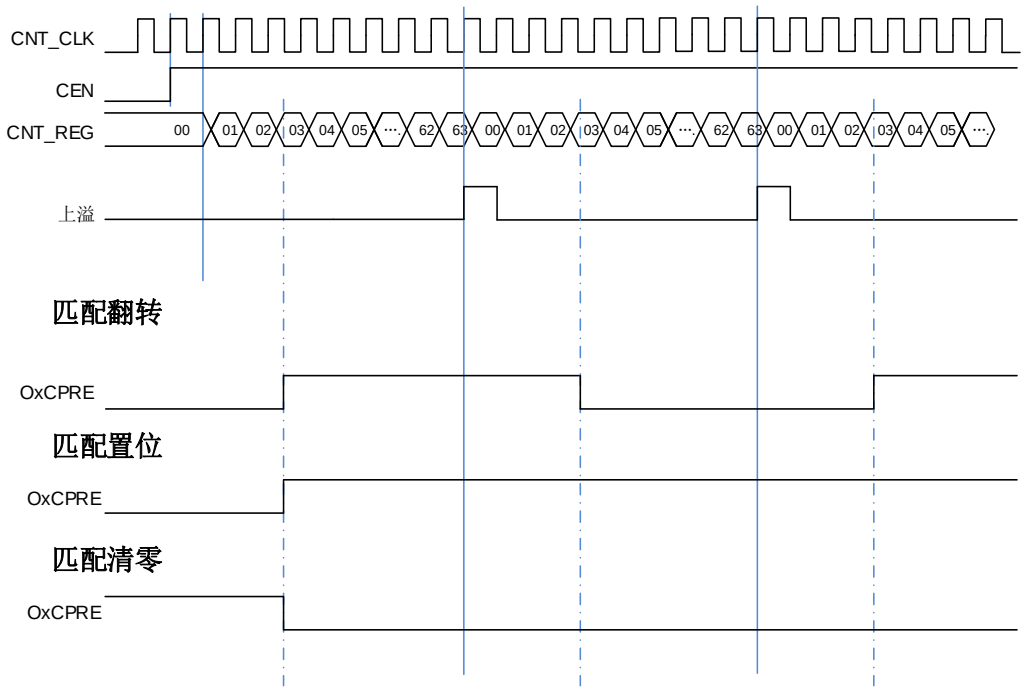
第三步：通过CHxIE/ MCHxIE/ CHxDEN/ MCHxDEN位配置中断/DMA请求使能。

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24-92. 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 24-92. 三种输出比较模式



PWM 模式

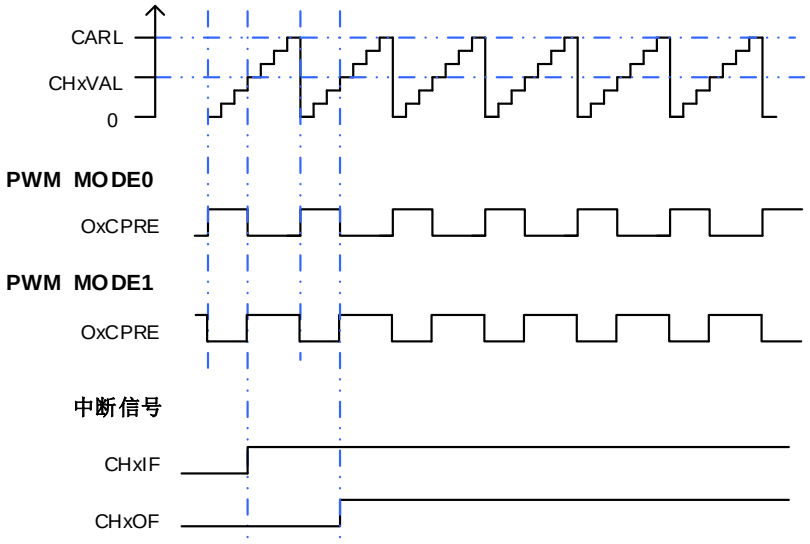
在 PWM 输出模式下（PWM 模式 0 是配置 CHxCOMCTL/ MCHxCOMCTL 为 4’b0110，PWM 模式 1 是配置 CHxCOMCTL/ MCHxCOMCTL 为 4’b0111），通道根据 TIMERx_CAR 寄存器和 TIMERx_CHxCV/ TIMERx_MCHxCV 寄存器的值，输出 PWM 波形。

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 24-93. 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 24-93. PWM 时序图



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 24-14. 复合 PWM 脉冲宽度*计算。

表 24-14. 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	PWM 模式 0（向上计数）或 PWM 模式 1（向下计数）	100%

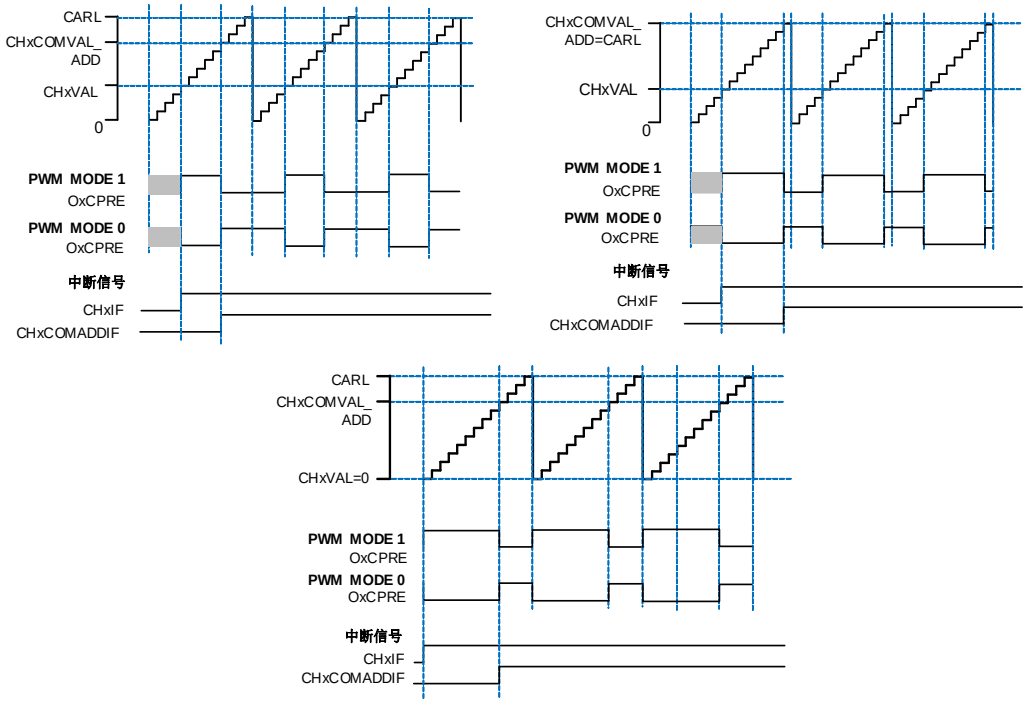
条件	模式	PWM 脉冲宽度
> CHxCOMVAL_ADD）	PWM 模式 0（向下计数） 或 PWM 模式 1（向上计数）	0%
CHxCOMVAL_ADD > CARL > CHxVAL	PWM 模式 0（向上计数） 或 PWM 模式 1（向下计数）	0%
	PWM 模式 0（向下计数） 或 PWM 模式 1（向上计数）	100%
（CHxVAL>CARL）且（CHxCOMVAL_ADD > CARL）	-	CHx_O 输出保持

当计数器计数到CHxVAL，CHxIF位置1且如果CHxIE=1通道x产生中断，如果CHxDEN=1，则产生DMA请求。当计数器计数到CHxCOMVAL_ADD时，CHxCOMADDIF位置1（该中断标志位只在复合PWM模式有效，CHxCPWMEN=1），如果CHxCOMADDIE = 1通道x附加比较中断产生（只有中断产生，没有DMA请求响应）。

根据CHxVAL，CHxCOMVAL_ADD和CARL之间的关系，可以分为四种情况：

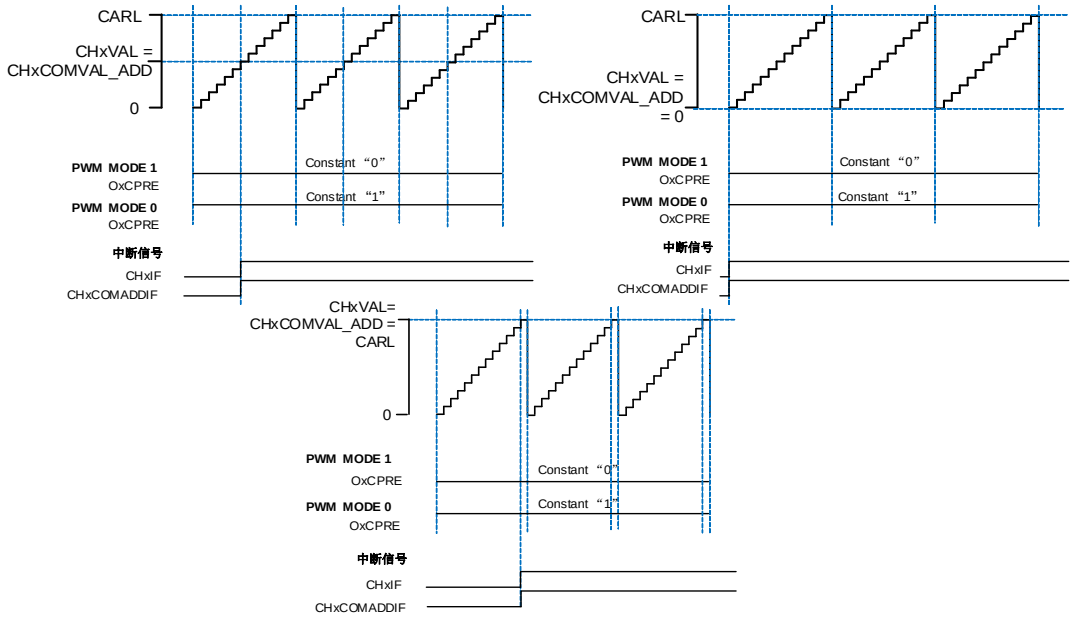
1） CHxVAL < CHxCOMVAL_ADD，CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 24-94. 通道 x 输出 PWM（CHxVAL < CHxCOMVAL_ADD）



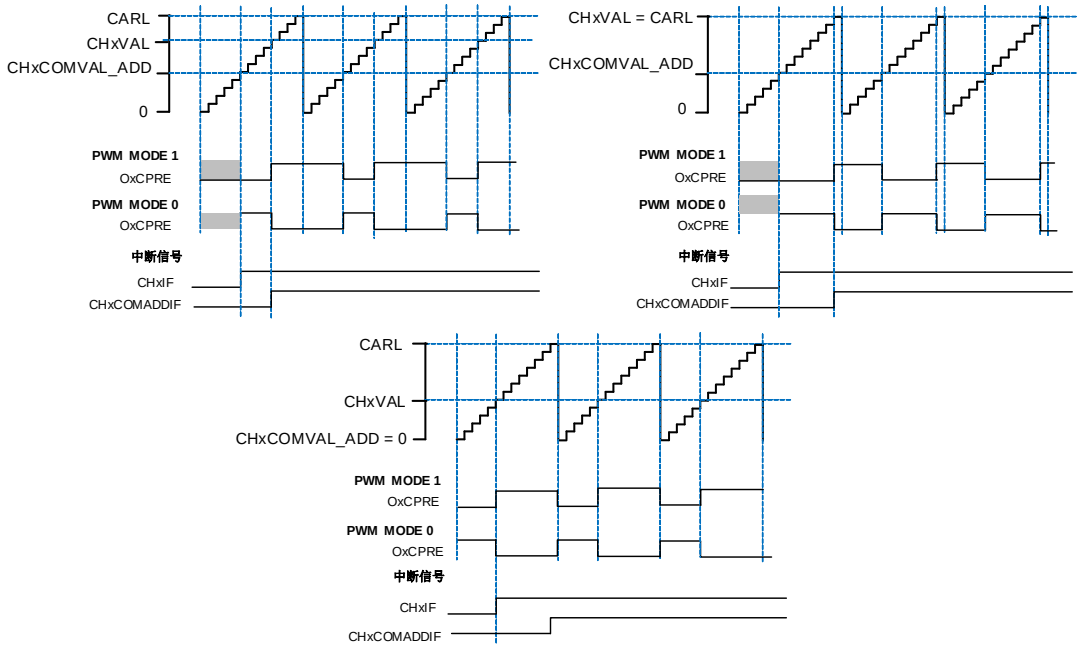
2） CHxVAL = CHxCOMVAL_ADD，CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 24-95. 通道 x 输出 PWM（CHxVAL = CHxCOMVAL_ADD）



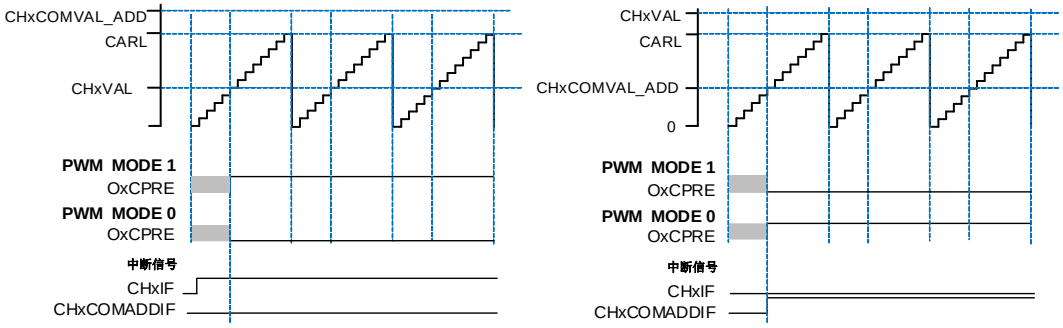
3） CHxVAL > CHxCOMVAL_ADD，CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 24-96. 通道 x 输出 PWM（CHxVAL > CHxCOMVAL_ADD）



4） CHxVAL或CHxCOMVAL_ADD值大于CARL。

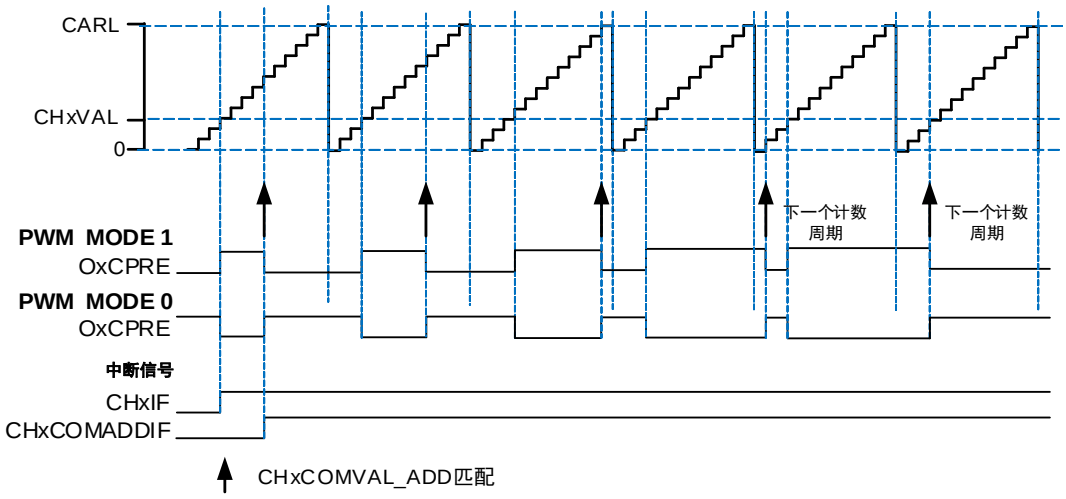
图 24-97. 通道 x 输出 PWM（CHxVAL 或 CHxCOMVAL_ADD > CARL）



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24-98. 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 24-98. 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。

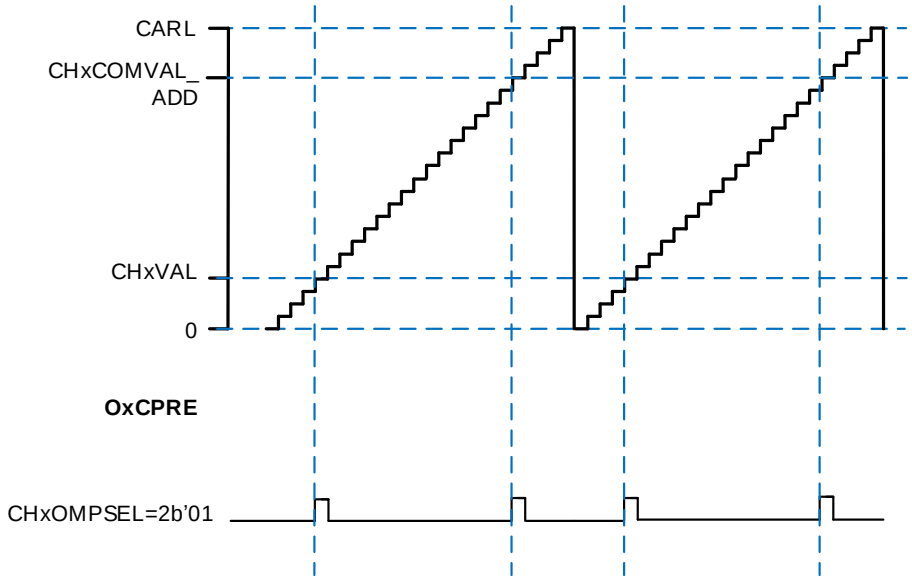
输出匹配脉冲选择

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 24-99. 边沿对齐模式下 CHx_O 输出脉冲（CHxOMPSEL=2'b01）



通道输出参考信号

如*图 24-89. 输出比较逻辑（当 MCHxMSEL = 2'00 时，x=0）*，*图 24-90. 输出比较逻辑（当 MCHxMSEL = 2'11 时，x=0）*和*图 24-91. 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。

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_CCHP 寄存器中和 TIMERx_CTL1 寄存器中的 POEN、ROS、IOS、ISOx 和 ISOxN 位。输出极性由 TIMERx_CHCTL2 寄存器中的 CHxP 和 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 24-15. 由参数控制的互补输出表（MCHxMSEL=2'b11）*。

表 24-15. 由参数控制的互补输出表（MCHxMSEL=2'b11）

互补参数					输出状态	
POEN	POEN	POEN	POEN	POEN	CHx_O	MCHx_O
0	0/1	0	0	0	CHx_O / CHx_ON = LOW CHx_O / CHx_ON 输出禁能(1)	
				1	CHx_O/CHx_ON输出关闭状态(2)： 通道先输出无效电平：CHx_O = CHxP，CHx_ON = CHxNP）；如果死区产生时钟未失效，在死区时间之后： CHx_O = ISOx，CHx_ON = ISOxN (3)	
			1	0		
				1		
		1	x	x	CHx_O/CHx_ON输出关闭状态： 通道先输出无效电平：CHx_O = CHxP，CHx_ON = CHxNP）；如果死区产生时钟未失效，在死区时间之后： CHx_O = ISOx，CHx_ON = ISOxN	
1	0	0/1	0	0	CHx_O/MCHx_O = LOW CHx_O/MCHx_O输出禁能	
				1	CHx_O = LOW CHx_O输出禁能	MCHx_O=OxCPRE⊕(4)MCHxP MCHx_O输出使能
			1	0	CHx_O=OxCPRE⊕CHxP CHx_O输出使能	MCHx_O = LOW MCHx_O输出禁能
				1	CHx_O=OxCPRE⊕CHxP CHx_O输出使能	MCHx_O=(!OxCPRE)(5)⊕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 / CHx_ON 输出与对应引脚断开，对应引脚电平受 GPIO 上下拉配置控制，无上下拉时为悬空高阻态；

（2） 输出关闭状态：CHx_O / CHx_ON 输出无效电平（CHx_O = 0⊕CHxP = CHxP）；

（3） 详情见中止模式章节。

（4） ⊕：异或操作；

（5） (!OxCPRE)：OxCPRE 信号的互补信号。

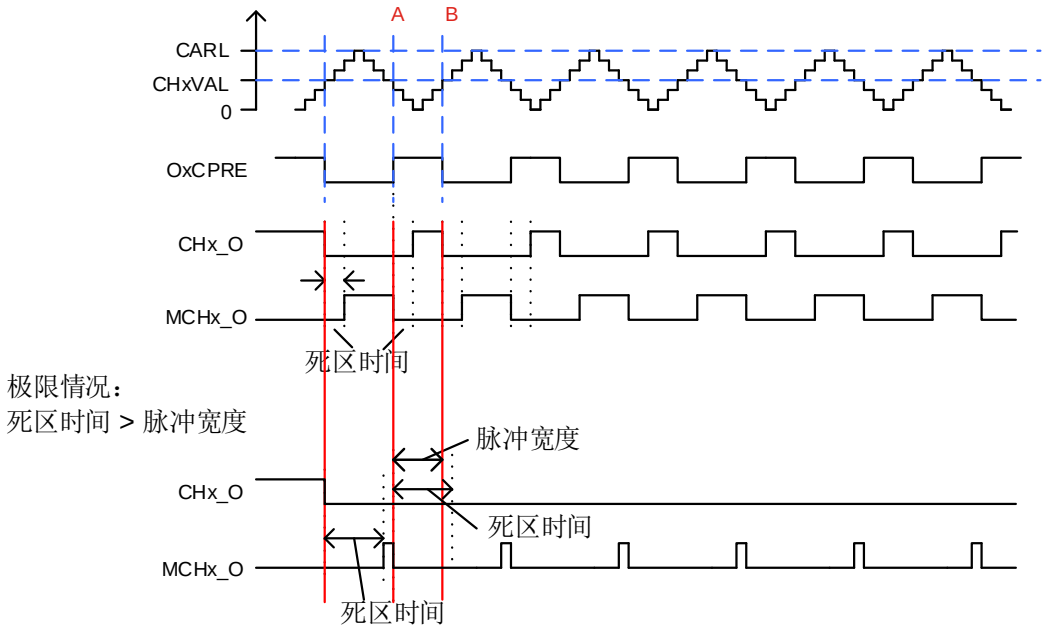
死区时间插入

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（TIMERx_CCHP）*。

死区时间的插入，确保了通道互补的两路信号不会同时有效。

在 PWM0 模式，当通道 x 匹配发生时（TIMERx 计数器=TIMERx_CHxCV），OxCPRE 翻转。在*图 24-100. 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 24-100. 带死区时间的互补输出*所示。

图 24-100. 带死区时间的互补输出



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_CCHP 寄存器中的 BRK0EN 位置 1 来使能。中止输入极性由 TIMERx_CCHP 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_CCHP 寄存器的 POEN、IOS 和 ROS 位，TIMERx_CTL1 寄存器的 ISOx 和 ISOxN 位。

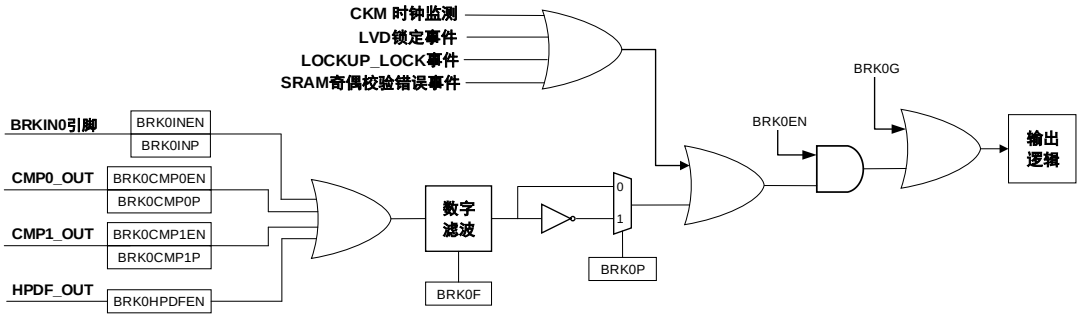
中止事件是所有源逻辑或运算的结果。中止功能可以处理三种类型的事件源：

- 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®-M7锁定事件或SRAM奇偶校验错误事件；
- 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 24-101. 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）*。

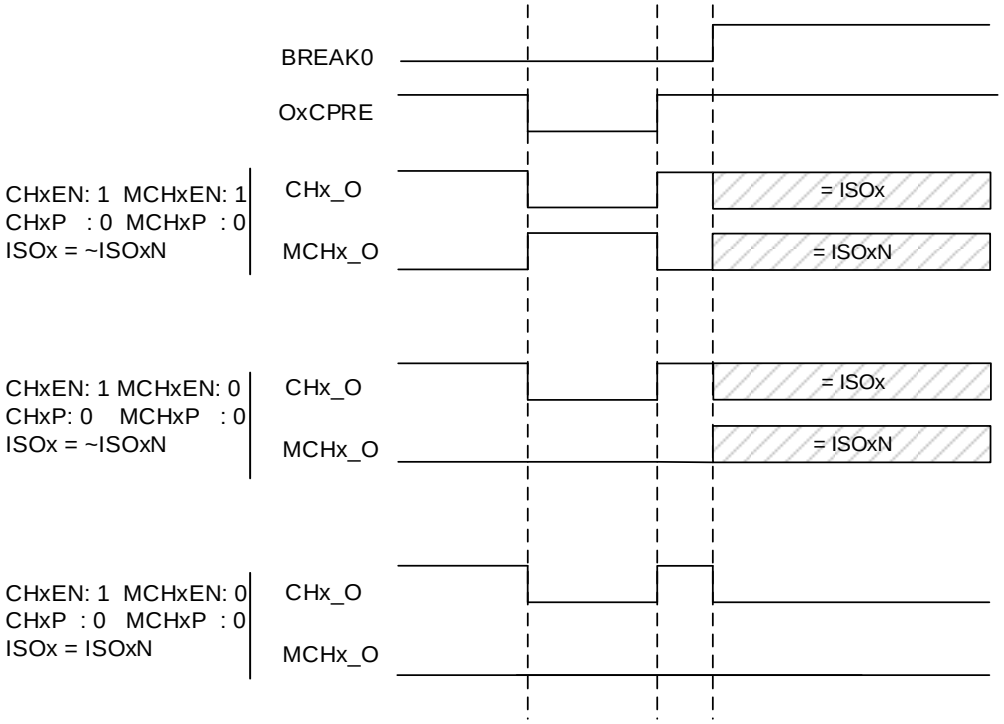
图 24-101. BREAK0 中止功能逻辑图



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 24-102. 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 24-102. 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_CCHP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_CCHP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_CCHP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_CCHP 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 24-16. 中止功能锁存/释放条件

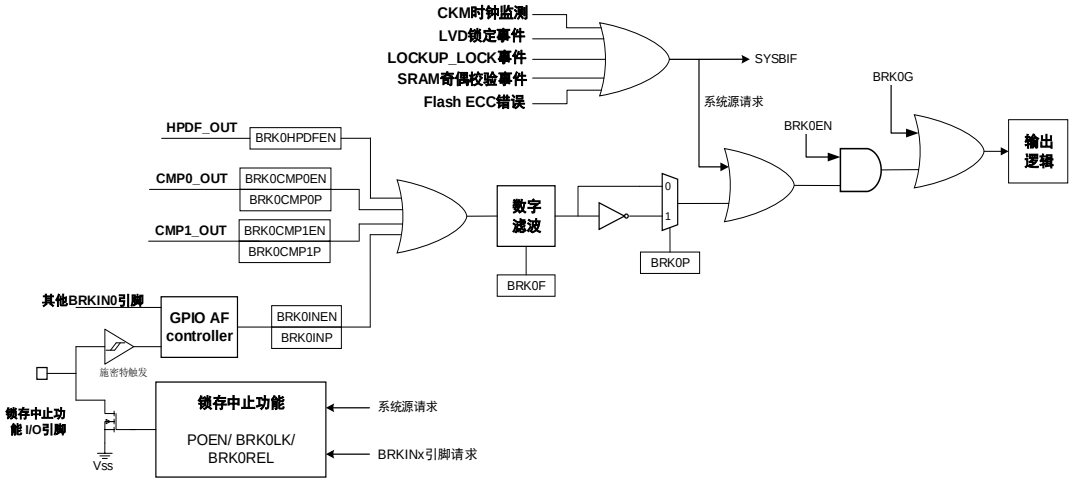
POEN	BRK0LK	BRK0REL	中止输入引脚状态
0	1	0	锁存
	1	1	释放

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 24-103. BREAK0 的 BRKIN0 引脚锁存功能逻辑图

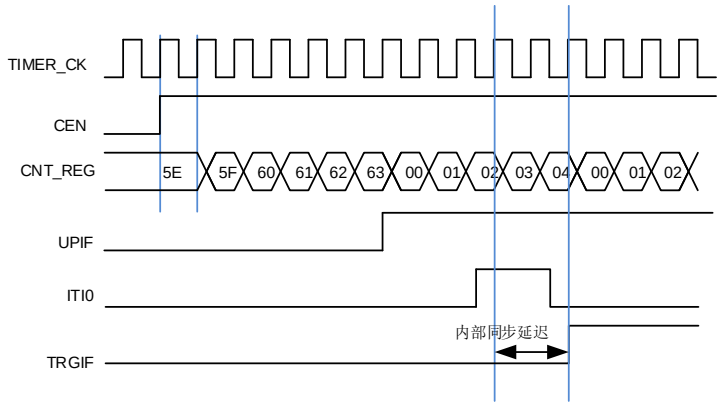
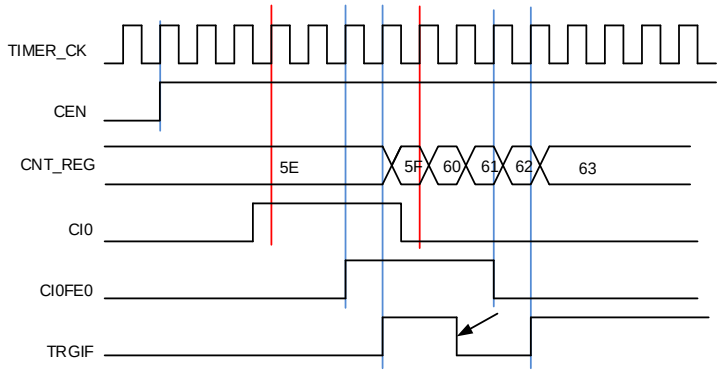


主-从管理

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=14,40,41,42,43,44)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。

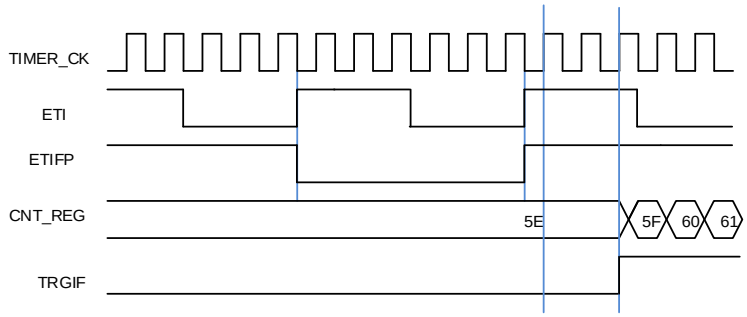
表 24-17. 从模式例子列表

	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 y=7：复位+事件模式	TSCFGy[4:0] 00000：Mode disable 00001：ITI0 00010：ITI1 00011：ITI2 00100：ITI3 00101：CI0F_ED 00110：CI0FE0 00111：CI1FE1 10010：MCI0FEM0	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

	模式选择	触发源选择	极性选择	滤波和预分频
		10011：ITI14		
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
	图 24-104. 复位模式下的控制电路 			
例2	**暂停模式** 当触发输入为低的时候，计数器暂停计数	TSCFG4[4:0] =5'b00110， 选择CI0FE0为触发源	[MCH0P=0,CH0P=0] CI0FE0不反相，在上升沿捕获	在这个例子中滤波被旁路
	图 24-105. 暂停模式下的控制电路 			
例3	**事件模式** 触发输入的上升沿计数器开始计数	TSCFG5[4:0] =5'b01000，选择 ETIFP为触发源	ETP = 0没有极性改变	ETPSC = 1，2分频 ETFC = 0，无滤波

	模式选择	触发源选择	极性选择	滤波和预分频

图 24-106. 事件模式下的控制电路



单脉冲模式

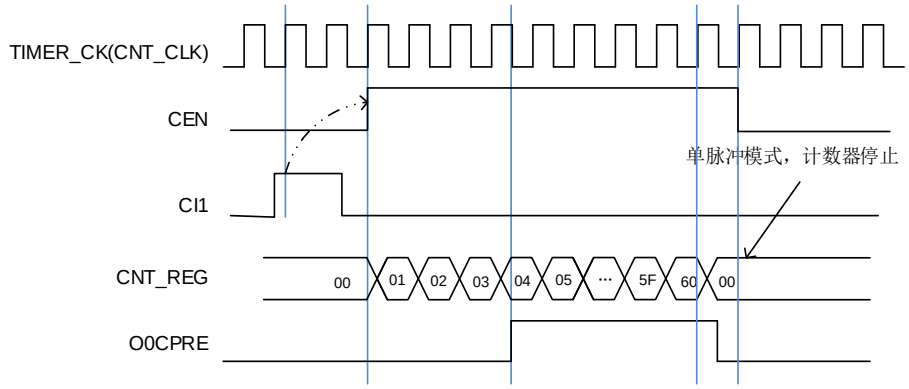
单脉冲模式与重复模式是相反的，设置 TIMERx_CTL0 寄存器的 SPM 位置 1，可使能单脉冲模式。当 SPM 置 1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置 CHxCOMCTL/ MCHxCOMCTL 配置 TIMERx 为 PWM 模式或者比较模式。

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 24-107. 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，通道输出参考信号 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=14,40,41,42,43,44)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。

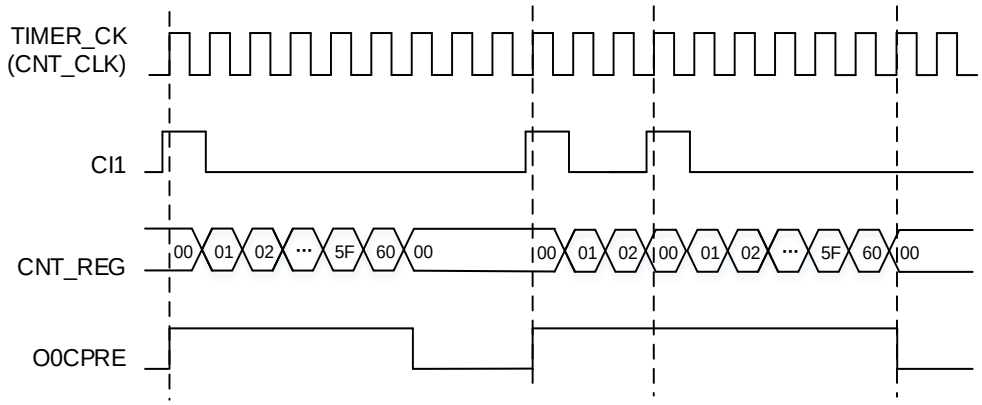
在可延时单脉冲模式1下，OxCPRE/ MOxCPRE的输出情况类似与PWM模式1。在向上计数模式时，OxCPRE/ MOxCPRE先输出无效电平，当外部触发事件发生时，立即输出有效电平，当下一次更新事件发生时，再变成无效电平；在向下计数模式时，OxCPRE/ MOxCPRE先输出有效电平，当外部触发事件发生时，立即输出无效电平，当下一次更新事件发生时，再变成有效电平。

注意：

1） 不能使用中央对齐模式，TIMERx_CTL0 寄存器中的 CAM[1:0] = 2'b00；

2） 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 24-108. 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®-M7 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。

24.3.5. TIMERx 寄存器(x=14/40/41/42/43/44)

TIMER14 基地址：0x4001 4000

TIMER40 基地址：0x4001 D000

TIMER41 基地址：0x4001 D400

TIMER42 基地址：0x4001 D800

TIMER43 基地址：0x4001 DC00

TIMER44 基地址：0x4001 F000

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				UPIFBUEN	保留	CKDIV[1:0]		ARSE	保留			SPM	UPS	UPDIS	CEN
				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	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_CCHP 寄存器的 PROT [1:0]位为 00 的时候可以被更改。
8	ISO0	通道 0 的空闲状态输出 0：当 POEN 复位，CH0_O 输出低电平 1：当 POEN 复位，CH0_O 输出高电平 如果 MCH0_O 生效，一个死区时间后 CH0_O 输出改变。此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]位为 00 的时候可以被更改。
7	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）。

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

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

位/位域	名称	描述
31:8	保留	必须保持复位值
7	MSM	主-从模式 该位被用来同步被选择的定时器同时开始计数。通过 TRIGI 和 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] = 2b'00 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] = 2b'00 时）。
19:15	保留	必须保持复位值.
14	TRGDEN	触发 DMA 请求使能 0：禁止触发 DMA 请求 1：使能触发 DMA 请求
13	CMTDEN	换相 DMA 更新请求使能 0：禁止换相 DMA 更新请求 1：使能换相 DMA 更新请求
12:11	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	保留			CH1COMCTL[3]	保留							CH0COMCTL[3]
		保留	保留				保留								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	CH1COMCTL[2:0]			CH1COMSEN	保留	CH1MS[1:0]		保留	CH0COMCTL[2:0]			CH0COMSEN	保留	CH0MS[1:0]	
CH1CAPFLT[3:0]				CH1CAPPSC[1:0]				CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

输出比较模式：

位/位域	名称	描述
31	CH1MS[2]	通道 1 I/O 模式选择 参考 CH1MS[1:0]描述。
30	CH0MS[2]	通道 0 I/O 模式选择 参考 CH0MS[1:0]描述。
29	CH1COMADDSEN	通道 1 附加输出比较影子寄存器使能 参考 CH0COMADDSEN 描述。
28	CH0COMADDSEN	通道 0 附加输出比较影子寄存器使能 当此位被置 1，TIMERx_CH0COMV_ADD 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：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_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =000 时此位不能被改变。
27:25	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	保留	必须保持复位值
14:12	CH1COMCTL[2:0]	通道 1 输出比较模式 参见 CH0COMCTL 描述
11	CH1COMSEN	通道 1 输出比较影子寄存器使能 参见 CH0COMSEN 描述
10	保留	必须保持复位值
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] = 2b'00 时，TIMERx_CHCTL2 寄存器的 CH1EN 位清 0；当 MCH1MSEL[1:0] = 2b'01 或 2b'11 时，TIMERx_CHCTL2 寄存器的 CH1EN、MCH1EN 位清 0）时，这些位才可以写。 000：通道 1 配置为输出 001：通道 1 配置为输入，IS1 映射在 CI1FE1 上 010：通道 1 配置为输入，IS1 映射在 CI0FE1 上 011：通道 1 配置为输入，IS1 映射在 ITS 上，此模式仅工作在内部触发器输入被选中时（由 SYSCFG_TIMERxCFG2(x=14,40,41,42,43,44)寄存器中的 TSCFG15[4:0]位域选择）。 100~111：保留
7	保留	必须保持复位值
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] = 2b'11 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_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =000（比较模式）时此位不能被改变。

3	CH0COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_CH0CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 0 输出/比较影子寄存器 1：使能通道 0 输出/比较影子寄存器 仅在单脉冲模式下（TIMERx_CTL0 寄存器的 SPM =1），可以在未确认预装载寄存器情况下使用 PWM 模式 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =000 时此位不能被改变。
2	保留	必须保持复位值
1:0	CH0MS[1:0]	通道 0 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭（当 MCH0MSEL[1:0] = 2b’00 时，TIMERx_CHCTL2 寄存器的 CH1EN 位清 0；当 MCH0MSEL[1:0] = 2b’01 或 2b’11 时，TIMERx_CHCTL2 寄存器的 CH0EN、MCH0EN 位清 0）时，CH0MS[2:0]才可写。 000：通道 0 配置为输出 001：通道 0 配置为输入，IS0 映射在 CI0FE0 上

010：通道 0 配置为输入，IS0 映射在 CI1FE0 上
011：通道 0 配置为输入，IS0 映射在 ITS 上。此模式仅工作在内部触发输入被选中时（由 SYSCFG_TIMERxCFG2(x=14,40,41,42,43,44)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
31:16	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

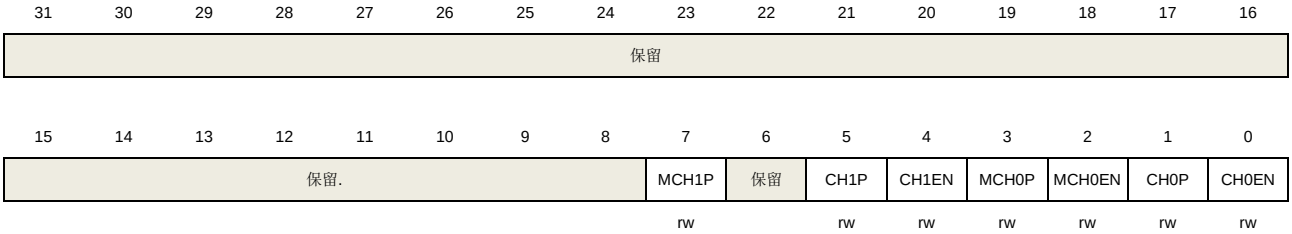
1:0	CH0MS[1:0]	通道 0 模式选择 与输出比较模式相同

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）。



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] = 2b'11 时，此位定义了多模式通道 0 输出信号 MCH0_O 的极性。 0：多模式通道 0 高电平有效 1：多模式通道 0 低电平有效 当通道 0 配置为输入模式时，此位和 CH0P 联合使用，作为通道 0 的极性选择控制信号。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。
2	MCH0EN	多模式通道 0 捕获/比较使能 当多模式通道 0 配置为输出模式时，将此位置 1 使能 MCH0_O 信号有效。当多模式通道 0 配置为输入模式时，将此位置 1 使能多模式通道 0 上的捕获事件。 0：禁止多模式通道 0 1：使能多模式通道 0
1	CH0P	通道 0 捕获/比较极性 当通道 0 配置为输出模式时，此位定义了输出信号极性。

0：通道 0 高电平有效
1：通道 0 低电平有效
当通道 0 配置为输入模式时，此位定义了通道 0 输入信号的极性。[MCH0P, CH0P]用于选择通道 0 输入信号信号有效边沿或者捕获极性。
00：把通道 0 输入信号的上升沿作为捕获或者从模式下触发的有效信号，且通道 0 输入信号不会被翻转。
01：把通道 0 输入信号的下降沿作为捕获或者从模式下触发的有效信号，且通道 0 输入信号会被翻转。
10：保留。
11：把通道 0 输入信号的上升沿和下降沿都作为捕获或者从模式下触发的有效信号，且通道 0 输入信号不翻转。
当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。

0	CH0EN	通道 0 捕获/比较使能 当通道 0 配置为输出模式时，将此位置 1 使能 CH0_O 信号有效。当通道 0 配置为输入模式时，将此位置 1 使能通道 0 上的捕获事件。 0：禁止通道 0 1：使能通道 0

计数器寄存器（TIMERx_CNT）

地址偏移：0x24
复位值：0x0000 0000

该寄存器通过字访问（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。

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）。

31	30	29	28	27	26	25	24	23	22	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
CARL[15:0]															
rw															

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

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）。

31	30	29	28	27	26	25	24	23	22	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							

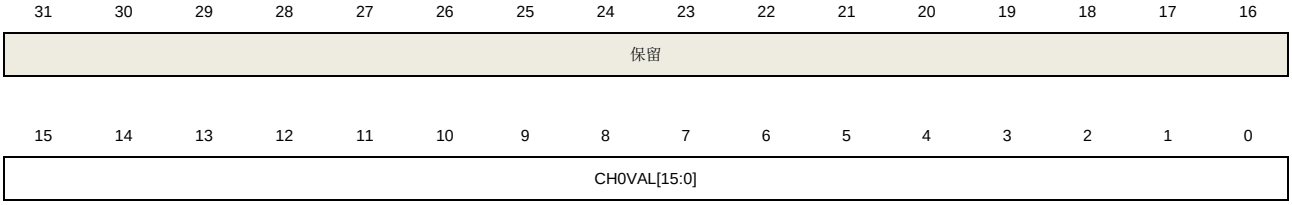
位/位域	名称	描述
31:8	保留	必须保持复位值.
7:0	CREP0[7:0]	重复计数器的值 0 这些位定义了更新事件的产生速率。重复计数器计数值减为 0 时产生更新事件。影子寄存器的更新速率也会受这些位影响（前提是影子寄存器被使能）。 **注意**：当 TIMERx_CFG 寄存器中的 CREPSEL =0 时，使用该位。

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）。



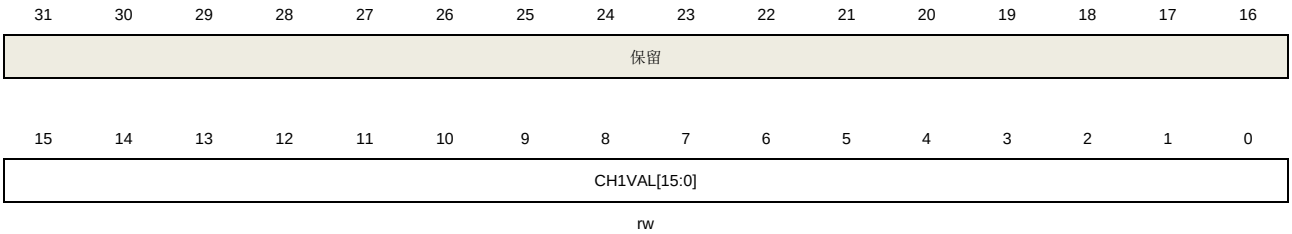
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值 当通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 1 捕获/比较寄存器（TIMERx_CH1CV）

地址偏移：0x38

复位值：0x0000 0000

该寄存器通过字访问（32位）。



位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH1VAL[15:0]	通道 1 的捕获或比较值

当通道 1 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。
当通道 1 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

互补通道保护寄存器（TIMERx_CCHP）

地址偏移：0x44
复位值：0x0000 0000

该寄存器通过字访问（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_CCHP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（高阻态的开漏描述）。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_CCHP 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 CHx_ON 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_CCHP 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_CCHP 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_CCHP 寄存器的 PROT [1:0] =00 时才可修改。

11 ROS
运行模式下“关闭状态”使能
当 POEN 位被置 1（运行模式），此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。参见 ***表 24-15. 由参数控制的互补输出表（MCHxMSEL=2'b11）***。
0：输出“关闭状态”禁能。当 CHxEN 或者 CHxNEN 位被清零，对应通道为输出

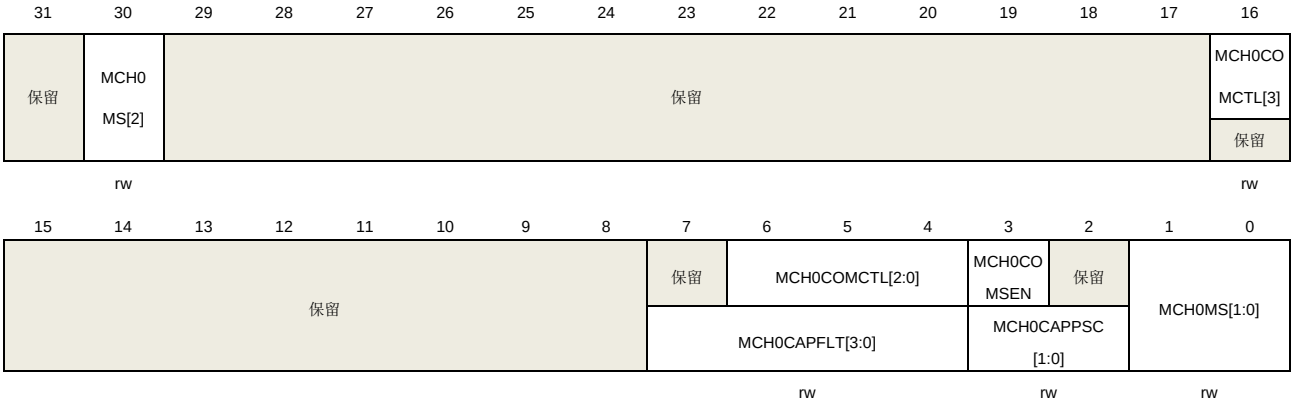
		“禁能状态”。 1：输出“关闭状态”使能。当 CHxEN 或者 CHxNEN 位被清零，对应通道为输出“关闭状态”。 此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
10	IOS	空闲模式下“关闭状态”使能 当 POEN 位被清 0（空闲模式），此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。参见 ***表 24-15. 由参数控制的互补输出表（MCHxMSEL=2’b11）***。 0：输出“关闭状态”禁能。当 CHxEN 和 CHxNEN 位均被清零，对应通道为输出“禁能状态”。 1：输出“关闭状态”使能。不论 CHxEN 和 CHxNEN 位的值，对应通道为输出“关闭状态”。 此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
9:8	PROT[1:0]	互补寄存器保护控制 这两位定义了寄存器的写保护特性。 00：禁能保护模式。无写保护. 01：PROT 模式 0。TIMERx_CTL1 寄存器中 ISOx/ISOxN 位，TIMERx_CCHP 寄存器中 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_CCHP 寄存器中 ROS/IOS 位。 11：PROT 模式 2.。除了 PROT 模式 1 下的寄存器写保护外，还有 TIMERx_CHCTLR0/1 中 CHxCOMCTL/ CHxCOMSEN 位（如果相关通道配置为输出模式）写保护。 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_CCHP 寄存器被写入，这两位被写保护
7:0	DTCFG[7:0]	死区时间控制 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 tDT, tDT=tDTS. DTCFG [7:5] =3’b 10x：DTvalue =（64+DTCFG [5:0]）xtDT, tDT =tDTS*2. DTCFG [7:5] =3’b 110：DTvalue =（32+DTCFG [4:0]）xtDT, tDT=tDTS*8. DTCFG [7:5] =3’b 111：DTvalue =（32+DTCFG [4:0]）xtDT, tDT =tDTS*16. 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]=00 时才可修改。

多模式通道控制寄存器 0（TIMERx_MCHCTL0）

地址偏移：0x48

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



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。
15:7	保留	必须保持复位值。
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] = 2b’00，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] = 2b’11 时，CH0COMCTL[2:0]位定义了输出准备信号 O0CPRE 的动作，而 O0CPRE 决定了 CH0_O、MCH0_O 的值。O0CPRE 高电平有效，CH0_O、MCH0_O 的有效电平取决于 CH0P、MCH0P 位。 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_CCHP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_CCHP寄存器的PROT [1:0]=11且CH0MS=00时此位不能被改变。
2	保留	必须保持复位值。
1:0	MCH0MS[1:0]	多模式通道 0 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭（TIMERx_CHCTL2 寄存器的 MCH0EN 位清 0）时，MCH0MS[2:0]才可写。 000：多模式通道 0 配置为输出 001：多模式通道 0 配置为输入，MIS0 映射在 MCI0FEM0 上 010：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=14,40,41,42,43,44)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] = 2b’00，此位定义了输出信号极性。 00：多模式通道 0 高电平有效 01：多模式通道 0 低电平有效 10：保留 11：保留 当通道 0 配置为输入模式时，此位定义了多模式通道 0 输入信号的极性。 MCH0FP[1:0]将选择多模式通道 0 输入信号的有效边沿或者捕获极性。 00：把多模式通道 0 输入信号的上升沿作为捕获或者从模式下触发的有效信号，且多模式通道 0 输入信号不会被翻转。 01：把多模式通道 0 输入信号的下降沿作为捕获或者从模式下触发的有效信号，且多模式通道 0 输入信号会被翻转。 10：保留。 11：把多模式通道 0 输入信号的上升沿或下降沿作为捕获或者从模式下触发的有效信号，并且多模式通道 0 输入信号不会被翻转。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。

多模式通道 0 捕获/比较寄存器（TIMERx_MCH0CV）

地址偏移：0x54

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCH0VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	MCH0VAL[15:0]	多模式通道 0 的捕获或比较值 当多模式通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值，且本寄存器为只读。 当多模式通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相

应影子寄存器后，影子寄存器值随每次更新事件更新。

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															

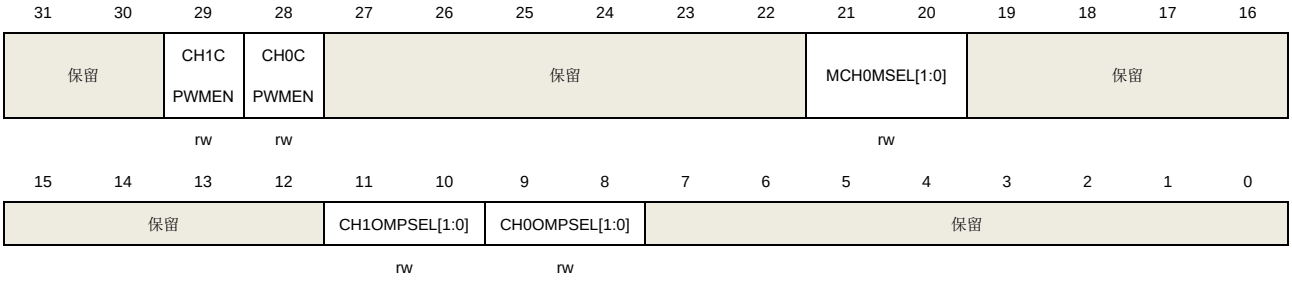
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH1COMVAL_ADD [15:0]	通道1附加比较值 当通道1附加配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。 **注意**：该寄存器仅用于复合PWM模式（当CH0CPWMEN=1时）。

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。



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

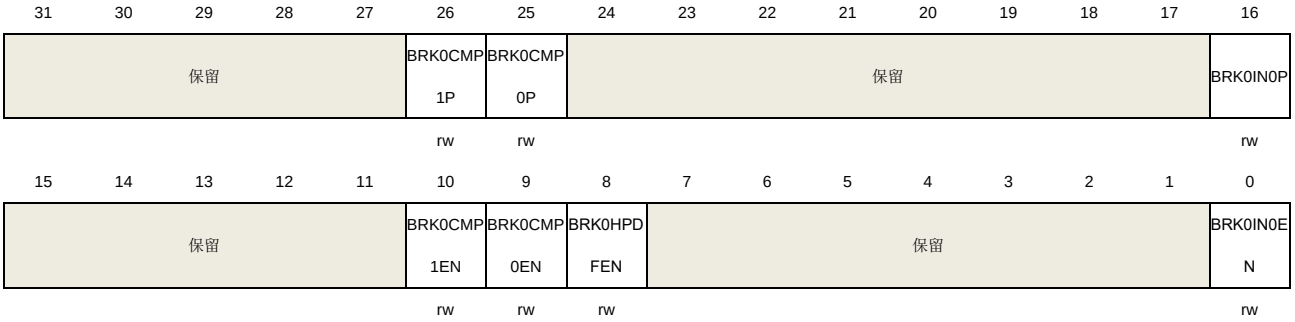
7:0	保留	必须保持复位值

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。



位/位域	名称	描述
31:27	保留	必须保持复位值。
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_CCHP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_CCHP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_CCHP寄存器的PROT [1:0] =00时才可修改。
15:11	保留	必须保持复位值。
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_CCHP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_CCHP寄存器的PROT [1:0] =00时才可修改。
8	BRK0HPDFEN	BREAK0 HPDF输入（hpdf_break[x]，请参考*表45-2. HPDF断路连接*）使能 0：HPDF输入禁能 1：HPDF输入使能 此位只有在TIMERx_CCHP寄存器的PROT [1:0] =00时才可修改。
7:1	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_CCHP寄存器的PROT [1:0] =00时才可修改。

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。

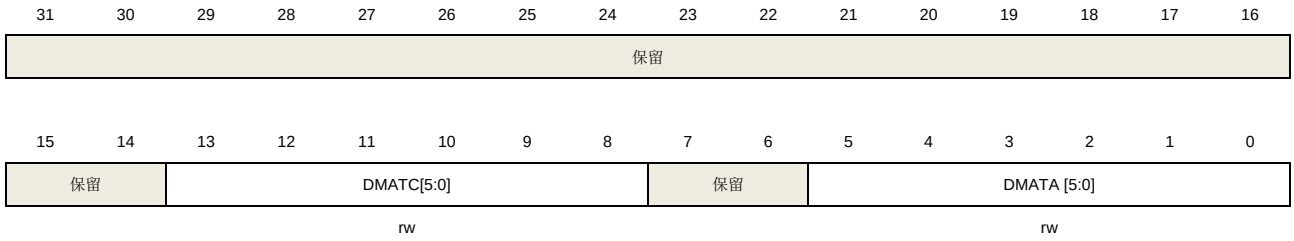
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CREP1[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CREP1[15:0]															
rw															

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。

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。



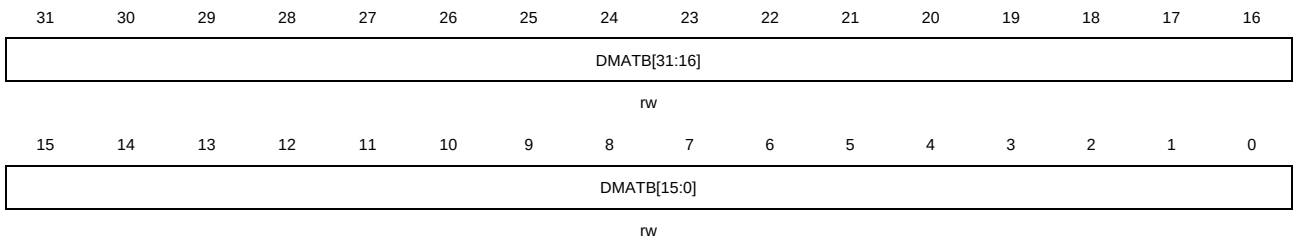
位/位域	名称	描述
31:14	保留	必须保持复位值
13:8	DMATC [5:0]	DMA 传输计数 该位域定义了 DMA 访问（读/写）TIMERx_DMATB 寄存器的次数。 6’b000000：传输1次 6’b000001：传输2次 … 6’b100101：传输 38 次
7:6	保留	必须保持复位值
5:0	DMATA [5:0]	DMA 传输起始地址 该位域定义了 DMA 访问 TIMERx_DMAVB 寄存器的第一个地址。当通过 TIMERx_DMA 第一次访问时，访问的就是该位域指定的地址。第二次访问 TIMERx_DMATB 时，将访问起始地址+0x4。 6’b0_0000：TIMERx_CTL0 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。



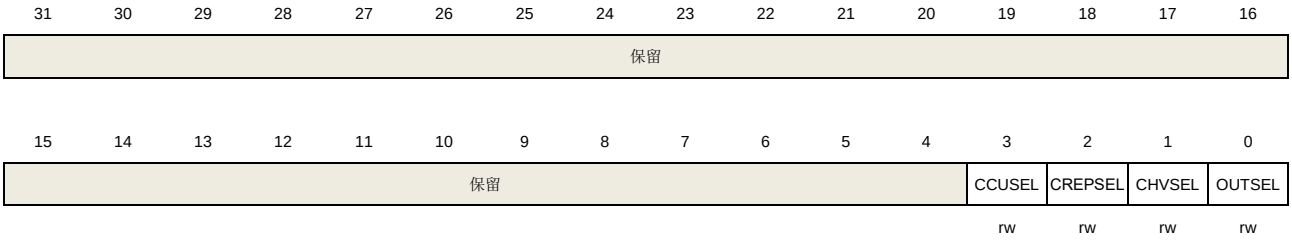
位/位域	名称	描述
31:0	DMATB [31:0]	DMA 发送缓冲 对这个寄存器的读或写，（起始地址+传输次数*4）地址范围内的寄存器会被访问 传输次数由硬件计算，范围为 0 到 DMATC。

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

24.4. 通用定时器 L4（TIMERx, x=15,16）

24.4.1. 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。

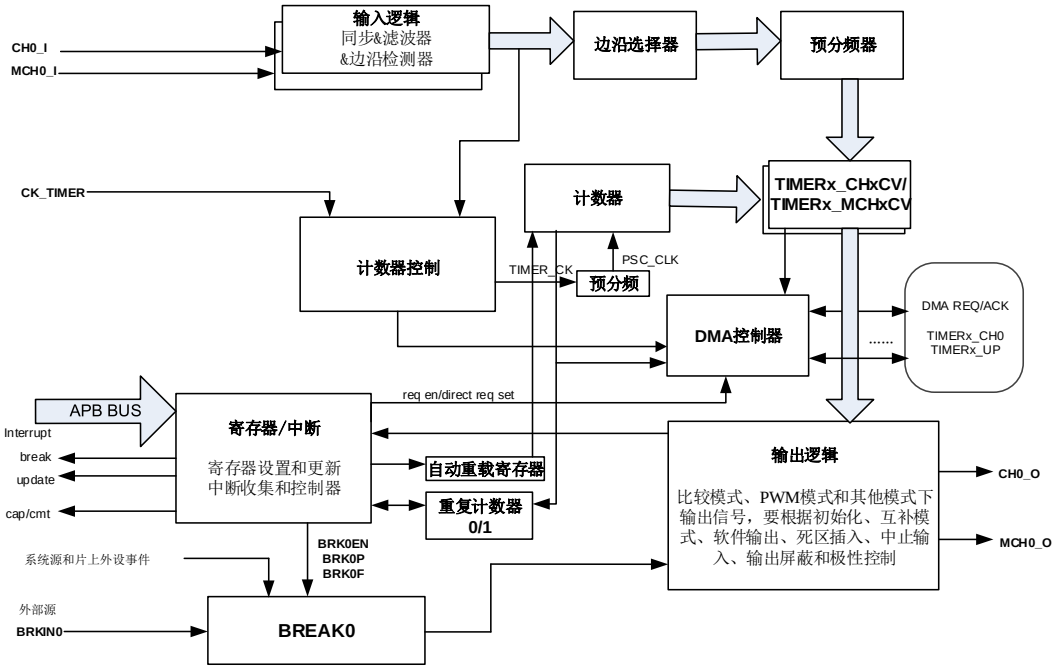
24.4.2. 主要特性

- 总通道数：2；
- 计数器宽度：16位；
- 时钟源可选：内部时钟；
- 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；

24.4.3. 结构框图

*图 24-109. 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 24-109. 通用定时器 L4 结构框图



24.4.4. 功能描述

时钟源选择

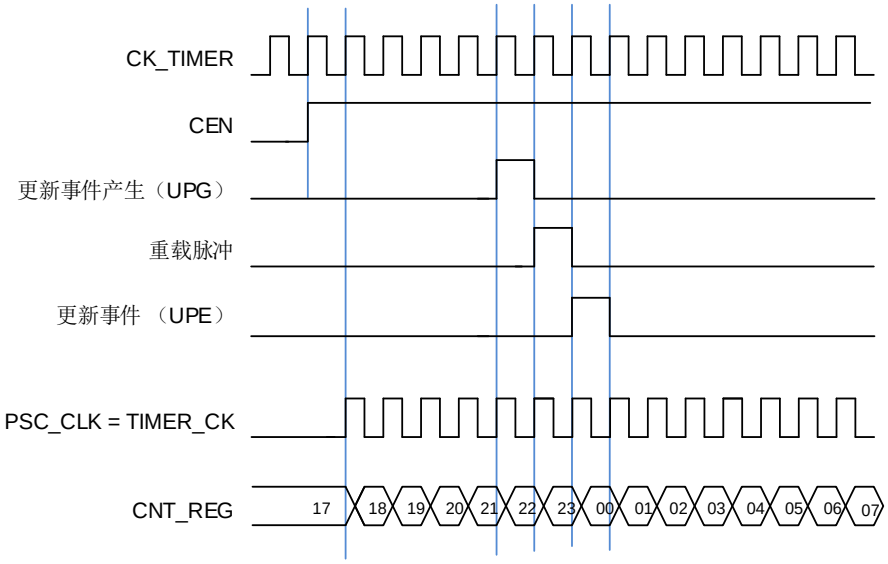
通用定时器 L4 由内部时钟源 TIMER_CK.

- 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。

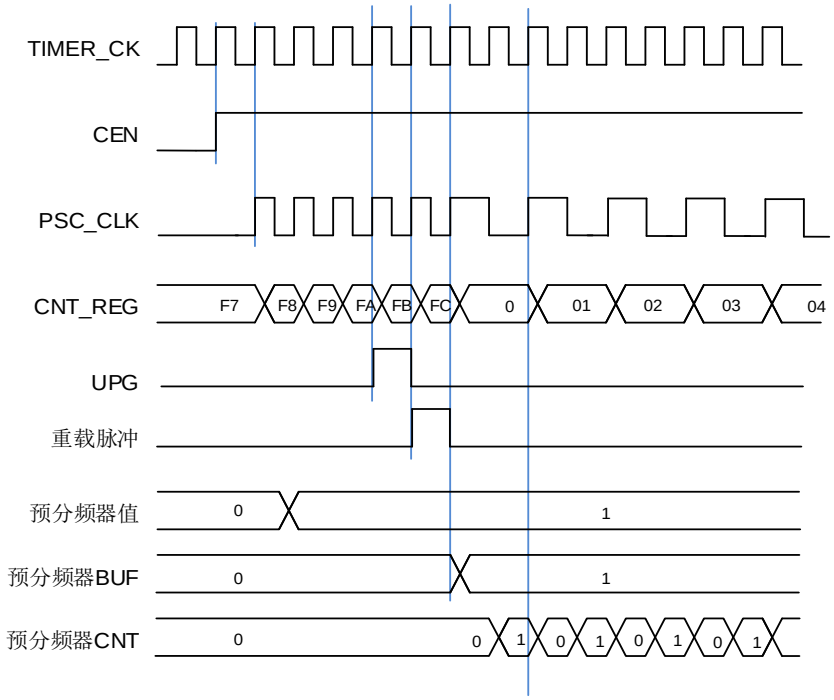
图 24-110. 内部时钟分频为 1 时正常模式下的控制电路



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 24-111. 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 24-112. 向上计数时序图，PSC=0/2 和 *图 24-113. 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 24-112. 向上计数时序图，PSC=0/2

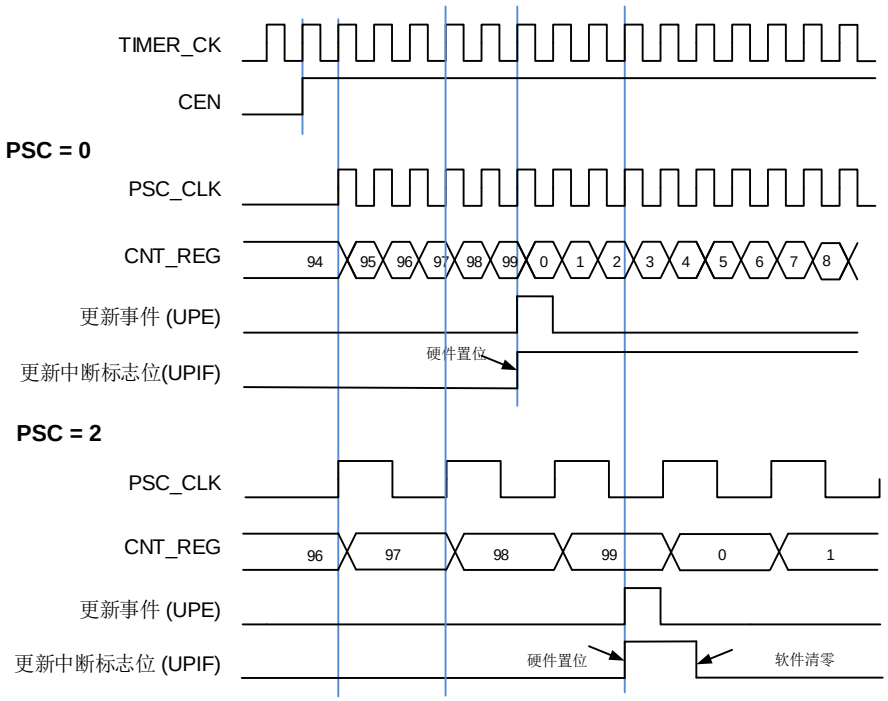
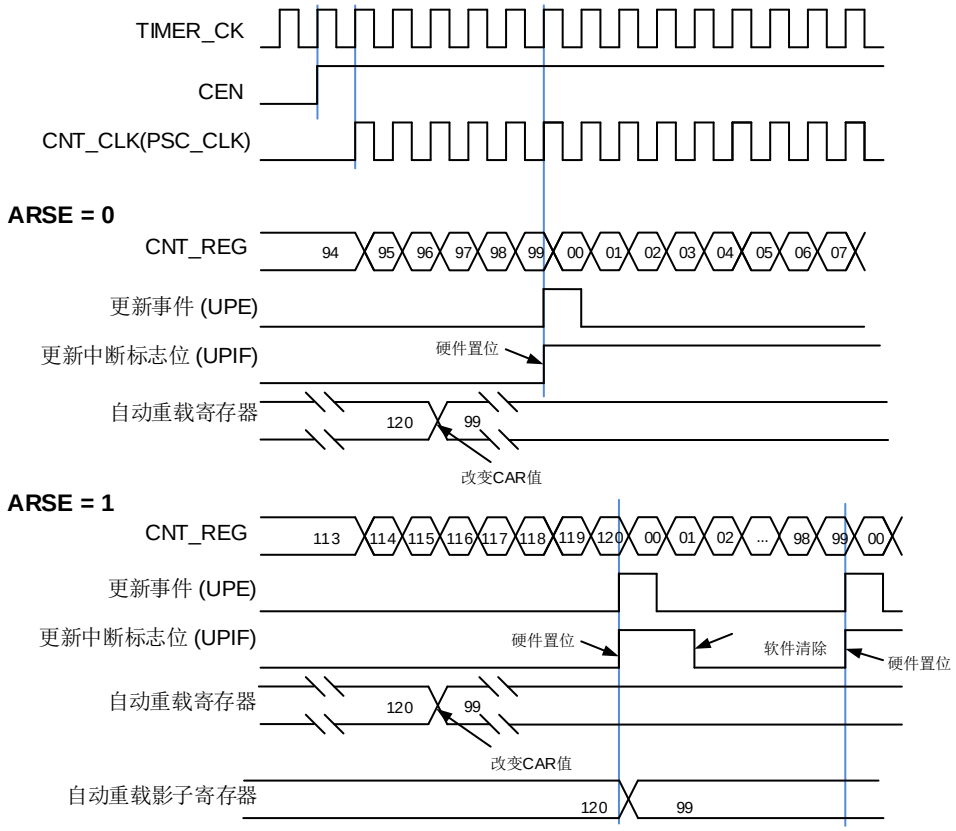


图 24-113. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



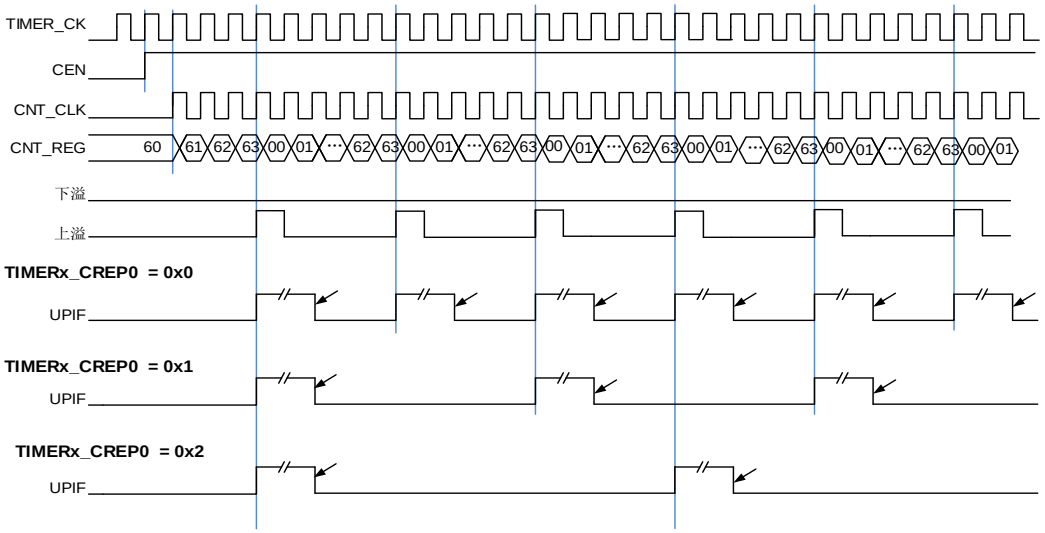
重复计数器

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 24-114. 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 24-115. 通道 0 输入捕获逻辑

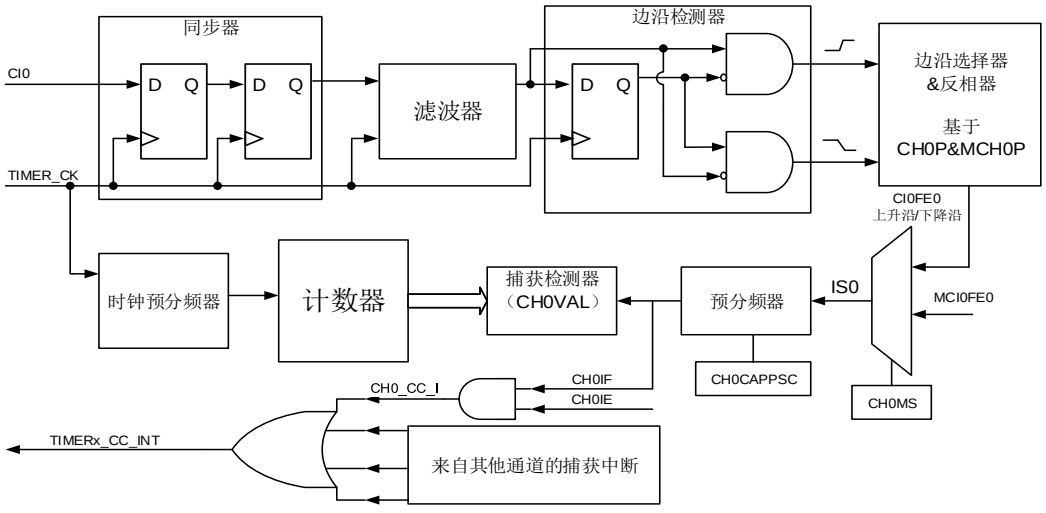
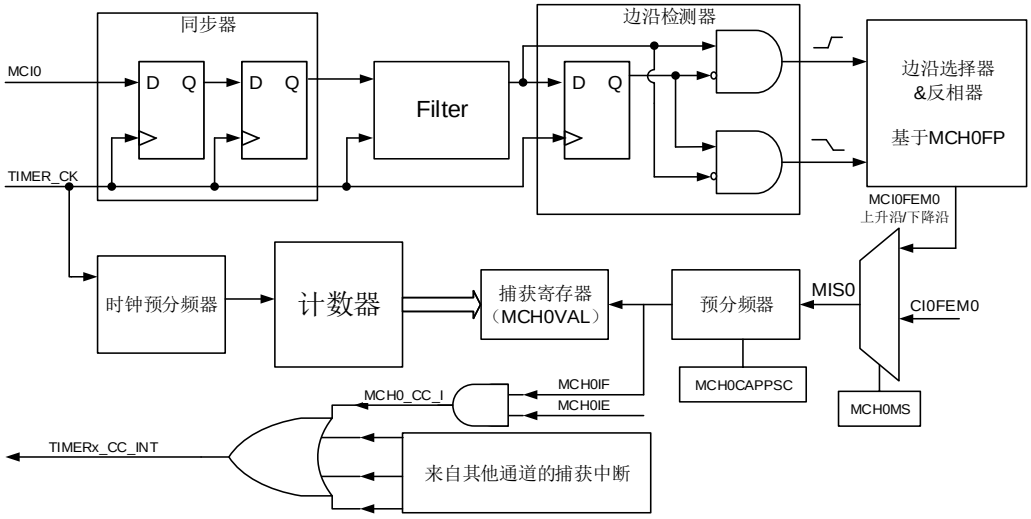


图 24-116. 多模式通道 0 输入捕获逻辑



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 24-117. 输出比较逻辑（当 MCHxMSEL = 2'00 时，x=0）*和*图 24-118. 输出比较逻辑（当 MCHxMSEL = 2'11 时，x=0）*给出了通道的输出比较逻辑。

图 24-117. 输出比较逻辑（当 MCHxMSEL = 2'00 时，x=0）

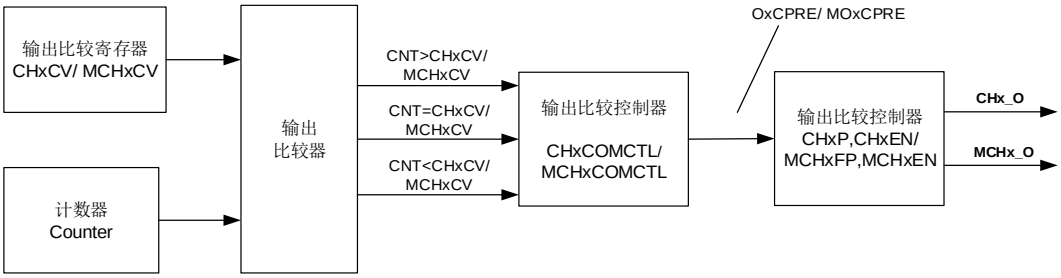
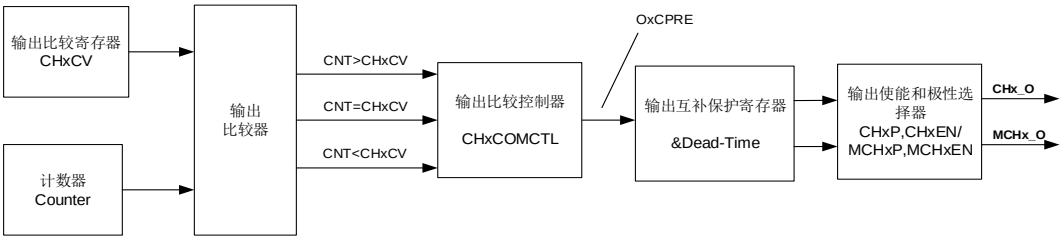


图 24-118. 输出比较逻辑（当 MCHxMSEL = 2'11 时，x=0）



通道输出信号CHx_O/MCHx_O与OxCPRE/ MOxCPRE信号（详情请见*通道输出参考信号*）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24-117. 输出比较逻辑（当MCHxMSEL = 2'00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24-118. 输出比较逻辑（当MCHxMSEL = 2'11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_CCHP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。

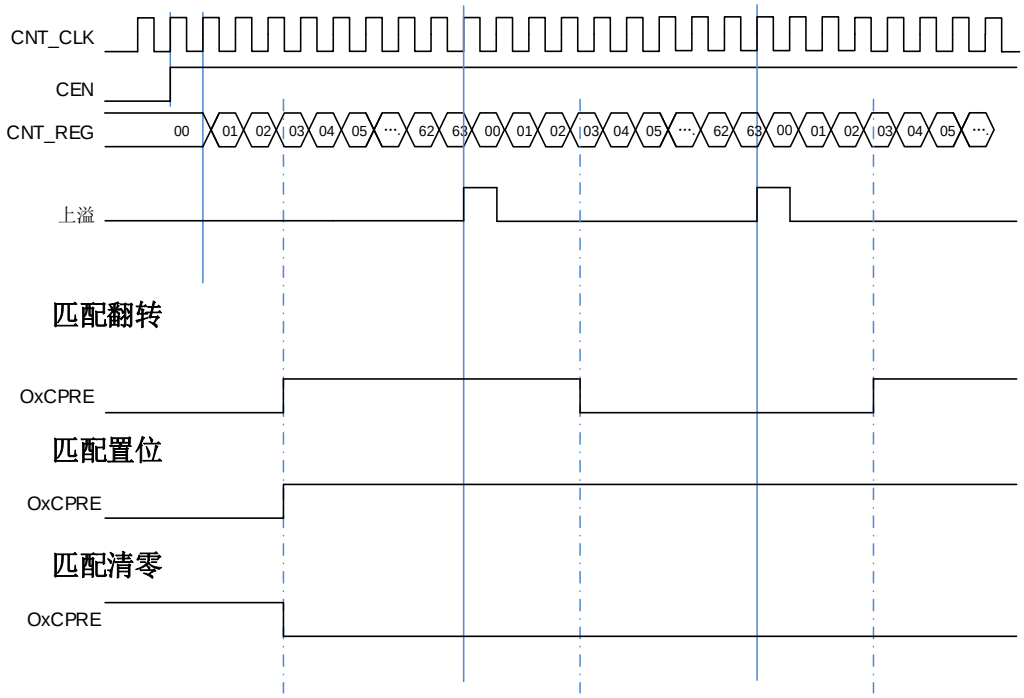
第四步：通过TIMERx_CAR寄存器和TIMERx_CHxCV寄存器配置输出比较时基：

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 24-119. 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 24-119. 三种输出比较模式



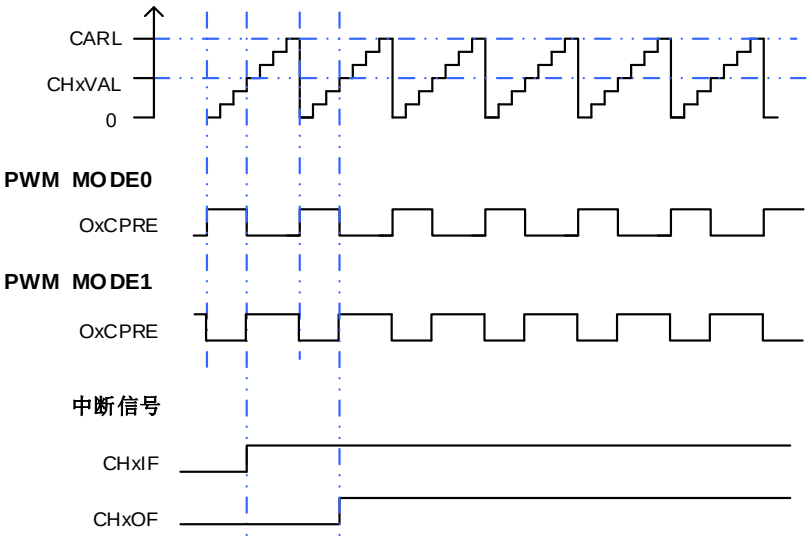
PWM 模式

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 24-120. 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 24-120. PWM 时序图



通道输出参考信号

*图 24-117. 输出比较逻辑（当 MCHxMSEL = 2’00 时，x=0）*和*图 24-118. 输出比较逻辑（当 MCHxMSEL = 2’11 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。

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_CCHP 寄存器中和 TIMERx_CTL1 寄存器中的 POEN、ROS、IOS、ISOx 和 ISOxN 位。输出极性由 TIMERx_CHCTL2 寄存器中的 CHxP 和 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 24-18. 由参数控制的互补输出表（MCHxMSEL=2'b11）*。

表 24-18. 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>POEN</th><th>POEN</th><th>POEN</th><th>POEN</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" rowspan="2">CHx_O / CHx_ON = LOW
CHx_O / CHx_ON 输出禁能(1)</td></tr>
<tr><td>1</td></tr>
<tr><td rowspan="3">1</td><td>0</td><td colspan="2" rowspan="3">CHx_O/CHx_ON输出关闭状态(2)：
通道先输出无效电平：CHx_O = CHxP，CHx_ON = CHxNP）；如果死区产生时钟未失效，在死区时间之后：
CHx_O = ISOx，CHx_ON = ISOxN (3)</td></tr>
<tr><td>1</td><td>1</td></tr>
<tr><td>1</td><td>x</td><td>x</td><td>CHx_O/CHx_ON输出关闭状态：
通道先输出无效电平：CHx_O = CHxP，CHx_ON = CHxNP）；如果死区产生时钟未失效，在死区时间之后：
CHx_O = ISOx，CHx_ON = ISOxN</td></tr>
<tr><td rowspan="4">1</td><td rowspan="4">0</td><td rowspan="4">0/1</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O/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⊕(4)MCHxP
MCHx_O输出使能</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O输出使能</td><td>MCHx_O = LOW
MCHx_O输出禁能</td></tr>
<tr><td>1</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O输出使能</td><td>MCHx_O=(!OxCPRE)(5)⊕MCHxP</td></tr>
</table>

互补参数					输出状态	
POEN	POEN	POEN	POEN	POEN	CHx_O	MCHx_O
						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 / CHx_ON 输出与对应引脚断开，对应引脚电平受 GPIO 上下拉配置控制，无上下拉时为悬空高阻态；

（2） 输出关闭状态：CHx_O / CHx_ON 输出无效电平（CHx_O = 0⊕CHxP = CHxP）；

（3） 详情见中止模式章节。

（4） ⊕：异或操作；

（5） (!OxCPRE)：OxCPRE 信号的互补信号。

死区时间插入

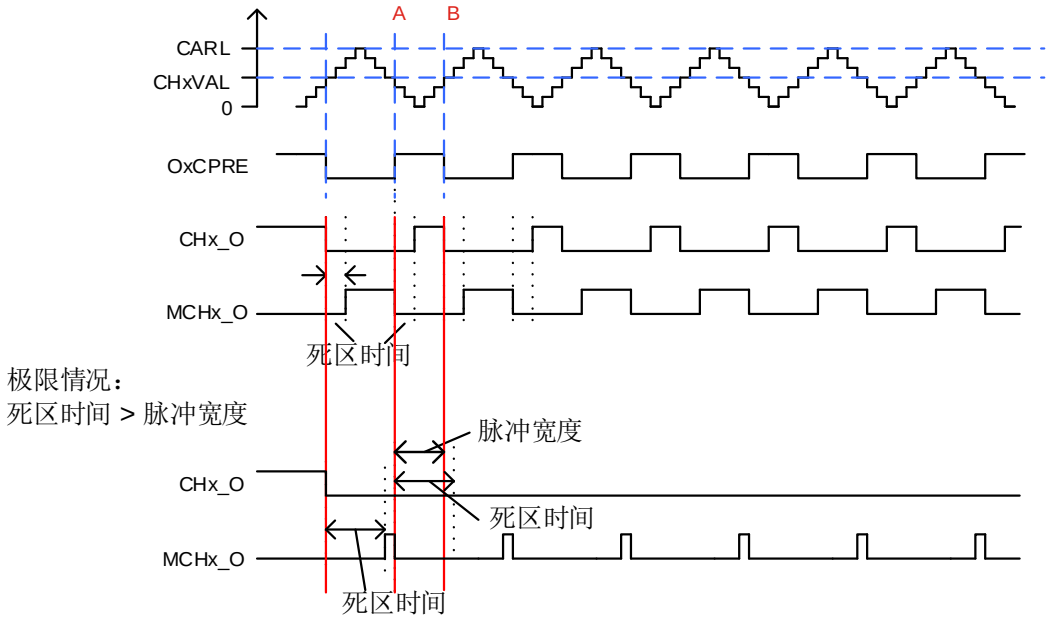
设置 MCHxMSEL=2'b11，CHxEN 和 MCHxEN 为 1'b1，同时设置 POEN=1，就可以使能死区插入功能。DTCFG 位域定义了死区时间，死区时间对所有通道有效。死区时间设置的细节请参考*互补通道保护寄存器（TIMERx_CCHP）*。

死区时间的插入，确保了通道互补的两路信号不会同时有效。

在 PWM0 模式，当通道 x 匹配发生时（TIMERx 计数器=TIMERx_CHxCV），OxCPRE 翻转。在*图 24-121. 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 24-121. 带死区时间的互补输出*所示。

图 24-121. 带死区时间的互补输出



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_CCHP 寄存器中的 BRK0EN 位置 1 来使能。中止输入极性由 TIMERx_CCHP 寄存器中的 BRK0P 位配置，电平有效。

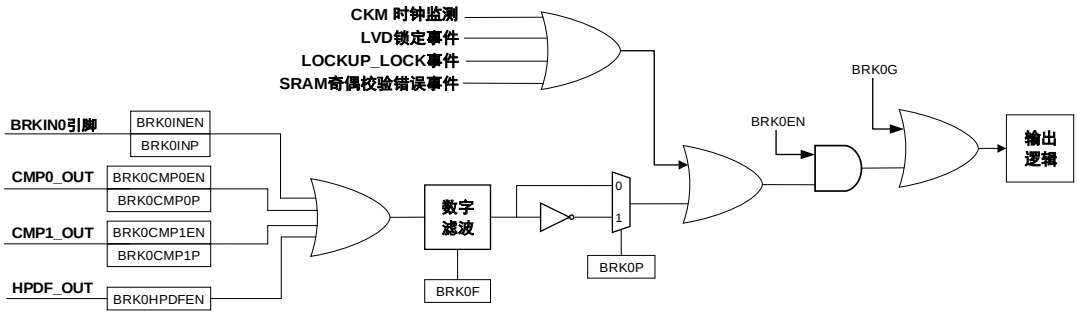
使用 BREAK0 功能时，CHx_O 和 MCHx_O 信号的输出电平由以下位控制：TIMERx_CCHP 寄存器的 POEN、IOS 和 ROS 位，TIMERx_CTL1 寄存器的 ISOx 和 ISOxN 位。

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®-M7锁定事件或SRAM奇偶校验错误事件；
- 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 24-122. 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触发选择寄存器（TRIGSEL_TIMER16BRKIN）*。

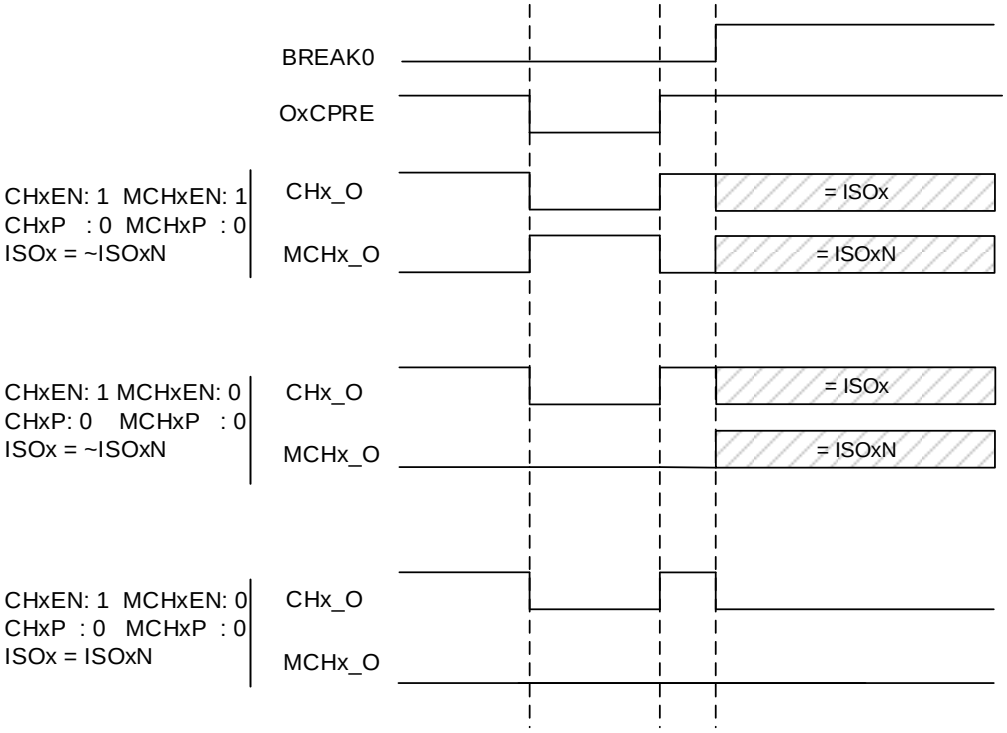
图 24-122. BREAK0 中止功能逻辑图



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 24-123. 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 24-123. 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_CCHP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_CCHP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_CCHP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_CCHP 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 24-19. 中止功能锁存/释放条件

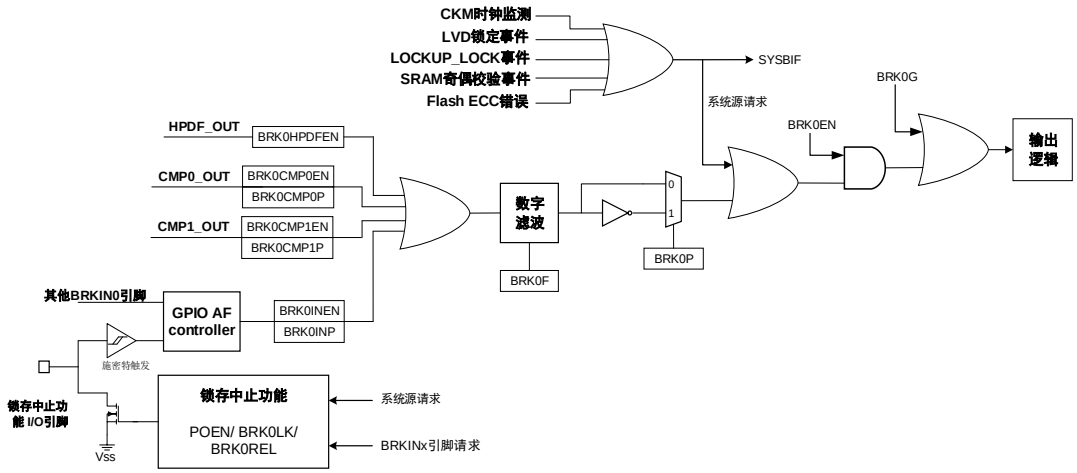
POEN	BRK0LK	BRK0REL	中止输入引脚状态
0	1	0	锁存
	1	1	释放

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 24-124. BREAK0 的 BRKIN0 引脚锁存功能逻辑图



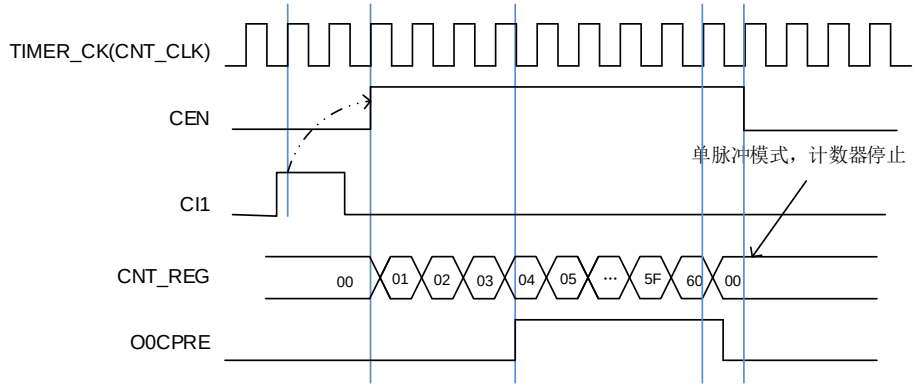
单脉冲模式

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 24-125. 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®-M7 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。

24.4.5. 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）。



位/位域	名称	描述
31:12	保留	必须保持复位值
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_CCHP 寄存器的 PROT [1:0]位为 00 的时候可以被更改。
8	ISO0	通道 0 的空闲状态输出

		0：当 POEN 复位，CH0_O 输出低电平 1：当 POEN 复位，CH0_O 输出高电平 如果 MCH0_O 生效，一个死区时间后 CH0_O 输出改变。此位只有在 TIMERx_CCHP 寄存器的 PROT[1:0]位为 00 的时候可以被更改。
7:4	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] = 2b'00 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] = 2b'00 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
15:7	保留	必须保持复位值
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] = 2b'11 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_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =000（比较模式）时此位不能被改变。
3	CH0COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_CH0CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 0 输出/比较影子寄存器 1：使能通道 0 输出/比较影子寄存器 仅在单脉冲模式下（TIMERx_CTL0 寄存器的 SPM =1），可以在未确认预装载寄存器情况下使用 PWM 模式

		当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =000 时此位不能被改变。
2	保留	必须保持复位值
1:0	CH0MS[1:0]	通道 0 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭（当 MCH0MSEL[1:0] = 2b'00 时，TIMERx_CHCTL2 寄存器的 CH1EN 位清 0；当 MCH0MSEL[1:0] = 2b'01 或 2b'11 时，TIMERx_CHCTL2 寄存器的 CH0EN、MCH0EN 位清 0）时，CH0MS[2:0]才可写。 000：通道 0 配置为输出 001：通道 0 配置为输入，IS0 映射在 CI0FE0 上 010：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] = 2b’11 时，此位定义了多模式通道 0 输出信号 MCH0_O 的极性。 0：多模式通道 0 高电平有效 1：多模式通道 0 低电平有效 当通道 0 配置为输入模式时，此位和 CH0P 联合使用，作为通道 0 的极性选择控制信号。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。
2	MCH0EN	多模式通道 0 捕获/比较使能 当多模式通道 0 配置为输出模式时，将此位置 1 使能 MCH0_O 信号有效。当多模式通道 0 配置为输入模式时，将此位置 1 使能多模式通道 0 上的捕获事件。 0：禁止多模式通道 0 1：使能多模式通道 0
1	CH0P	通道 0 捕获/比较极性 当通道 0 配置为输出模式时，此位定义了输出信号极性。 0：通道 0 高电平有效 1：通道 0 低电平有效 当通道 0 配置为输入模式时，此位定义了通道 0 输入信号的极性。[MCH0P, CH0P] 用于选择通道 0 输入信号信号有效边沿或者捕获极性。 00：把通道 0 输入信号的上升沿作为捕获或者从模式下触发的有效信号，且通道 0

输入信号不会被翻转。
01：把通道 0 输入信号的下降沿作为捕获或者从模式下触发的有效信号，且通道 0 输入信号会被翻转。
10：保留。
11：把通道 0 输入信号的上升沿和下降沿都作为捕获或者从模式下触发的有效信号，且通道 0 输入信号不翻转。
当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。

0	CH0EN	通道 0 捕获/比较使能 当通道 0 配置为输出模式时，将此位置 1 使能 CH0_O 信号有效。当通道 0 配置为输入模式时，将此位置 1 使能通道 0 上的捕获事件。 0：禁止通道 0 1：使能通道 0

计数器寄存器（TIMERx_CNT）

地址偏移：0x24
复位值：0x0000 0000

该寄存器通过字访问（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。

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）。

31	30	29	28	27	26	25	24	23	22	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
CARL[15:0]															

rw

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

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）。

31	30	29	28	27	26	25	24	23	22	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）。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值 当通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

互补通道保护寄存器（TIMERx_CCHP）

地址偏移：0x44

复位值：0x0000 0000

该寄存器通过字访问（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_CCHP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（高阻态的开漏描述）。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_CCHP 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 CHx_ON 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_CCHP 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_CCHP 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_CCHP 寄存器的 PROT [1:0] =00 时才可修改。

11 ROS

运行模式下“关闭状态”使能
当 POEN 位被置 1（运行模式），此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。参见 ***表 24-18. 由参数控制的互补输出表（MCHxMSEL=2'b11）***。
0：输出“关闭状态”禁能。当 CHxEN 或者 CHxNEN 位被清零，对应通道为输出“禁能状态”。
1：输出“关闭状态”使能。当 CHxEN 或者 CHxNEN 位被清零，对应通道为输出“关闭状态”。
此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。

10 IOS

空闲模式下“关闭状态”使能
当 POEN 位被清 0（空闲模式），此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。参见 ***表 24-18. 由参数控制的互补输出表（MCHxMSEL=2'b11）***。
0：输出“关闭状态”禁能。当 CHxEN 和 CHxNEN 位均被清零，对应通道为输出“禁能状态”。
1：输出“关闭状态”使能。不论 CHxEN 和 CHxNEN 位的值，对应通道为输出“关闭状态”。
此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。

9:8 PROT[1:0]

互补寄存器保护控制
这两位定义了寄存器的写保护特性。
00：禁能保护模式。无写保护.
01：PROT 模式 0。TIMERx_CTL1 寄存器中 ISOx/ISOxN 位，TIMERx_CCHP 寄存器中 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_CCHP 寄存器中 ROS/IOS 位。
11：PROT 模式 2.。除了 PROT 模式 1 下的寄存器写保护外，还有 TIMERx_CHCTLR0/1 中 CHxCOMCTL/ CHxCOMSEN 位（如果相关通道配置为输出模式）写保护。
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/

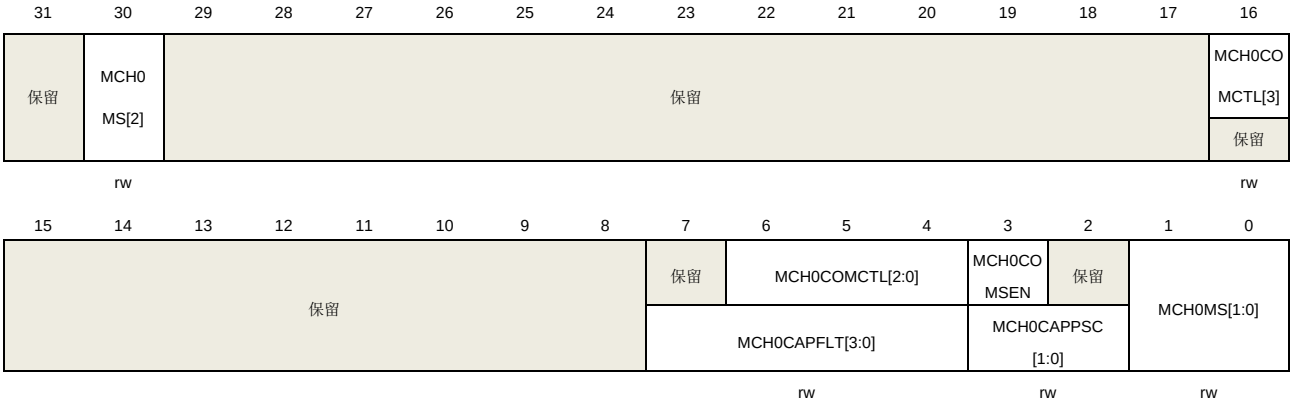
CHxCOMSEN/ CHxCOMADDSEN/ MCHxCOMCTL/ MCHxCOMSEN 位（如果相关通道配置为输出模式）写保护。
系统复位后这两位只能被写一次，一旦 TIMERx_CCHP 寄存器被写入，这两位被写保护

7:0	DTCFG[7:0]	死区时间控制 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 tDT, tDT=tDTS. DTCFG [7:5] =3’b 10x：DTvalue =（64+DTCFG [5:0]）xtDT, tDT =tDTS*2. DTCFG [7:5] =3’b 110：DTvalue =（32+DTCFG [4:0]）xtDT, tDT=tDTS*8. DTCFG [7:5] =3’b 111：DTvalue =（32+DTCFG [4:0]）xtDT, tDT =tDTS*16. 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]=00 时才可修改。

多模式通道控制寄存器 0（TIMERx_MCHCTL0）

地址偏移：0x48
复位值：0x0000 0000

该寄存器只能按字（32位）访问。



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。
15:7	保留	必须保持复位值。
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] = 2b’00，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] = 2b'11 时，CH0COMCTL[2:0]位定义了输出准备信号 O0CPRE 的动作，而 O0CPRE 决定了 CH0_O、MCH0_O 的值。O0CPRE 高电平有效，CH0_O、MCH0_O 的有效电平取决于 CH0P、MCH0P 位。 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_CCHP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_CCHP寄存器的PROT [1:0]=11且CH0MS=00时此位不能被改变。
2	保留	必须保持复位值。
1:0	MCH0MS[1:0]	多模式通道 0 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭（TIMERx_CHCTL2 寄存器的 MCH0EN 位清 0）时，MCH0MS[2:0]才可写。 000：多模式通道 0 配置为输出

001：多模式通道 0 配置为输入，MIS0 映射在 MCI0FEM0 上
010：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：

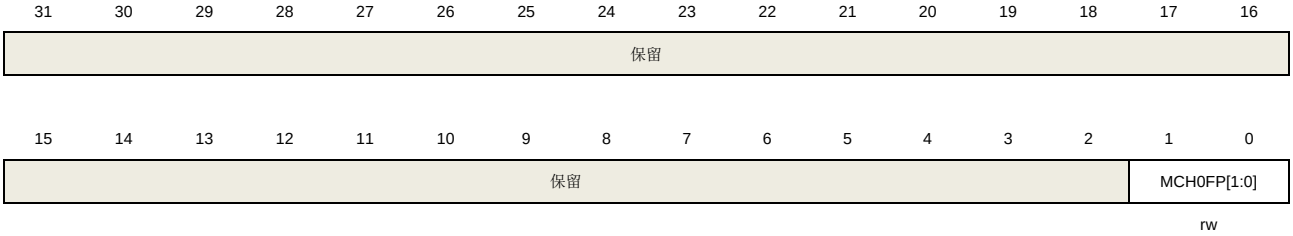
位/位域	名称	描述
31	保留	必须保持复位值
30	MCH0MS[2]	多模式通道 0 I/O 模式选择 参考MCH0MS[1:0]描述。
29:8	保留	必须保持复位值
7:4	MCH0CAPFLT[3:0]	通道 0 输入捕获滤波控制 数字滤波器由一个事件计数器组成，它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 MCI0 输入信号的采样频率和数字滤波器的长度。 0000：无滤波器，f_{SAMP}= f_{DTS}，N=1 0001：f_{SAMP} = f_{CK_TIMER}，N=2 0010：f_{SAMP} = f_{CK_TIMER}，N=4 0011：f_{SAMP} = 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。



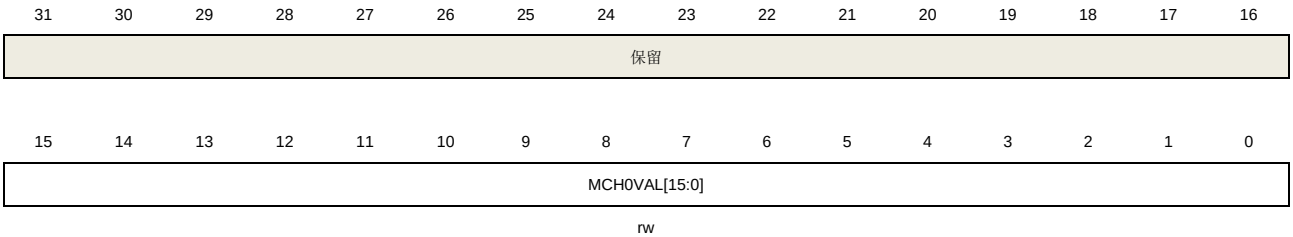
位/位域	名称	描述
31:2	保留	必须保持复位值
1:0	MCH0FP[1:0]	多模式通道 0 捕获/比较独立极性控制 当多模式通道 0 配置为输出模式时，且 MCH0MSEL[1:0] = 2b'00，此位定义了输出信号极性。 00：多模式通道 0 高电平有效 01：多模式通道 0 低电平有效 10：保留 11：保留 当通道 0 配置为输入模式时，此位定义了多模式通道 0 输入信号的极性。 MCH0FP[1:0]将选择多模式通道 0 输入信号的有效边沿或者捕获极性。 00：把多模式通道 0 输入信号的上升沿作为捕获或者从模式下触发的有效信号，且多模式通道 0 输入信号不会被翻转。 01：把多模式通道 0 输入信号的下降沿作为捕获或者从模式下触发的有效信号，且多模式通道 0 输入信号会被翻转。 10：保留。 11：把多模式通道 0 输入信号的上升沿或下降沿作为捕获或者从模式下触发的有效信号，并且多模式通道 0 输入信号不会被翻转。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。

多模式通道 0 捕获/比较寄存器（TIMERx_MCH0CV）

地址偏移：0x54

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



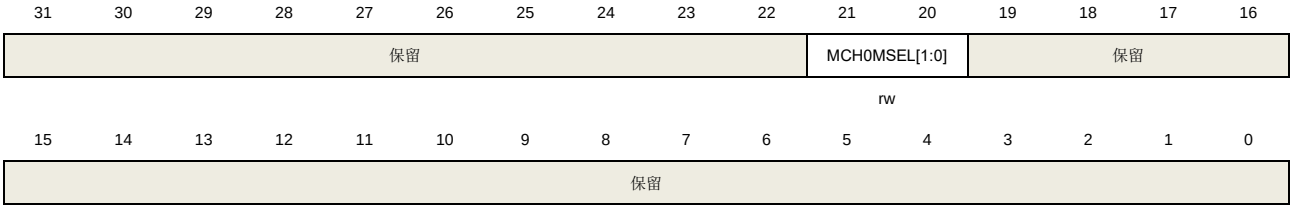
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	MCH0VAL[15:0]	多模式通道 0 的捕获或比较值 当多模式通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值，且本寄存器为只读。 当多模式通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

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。



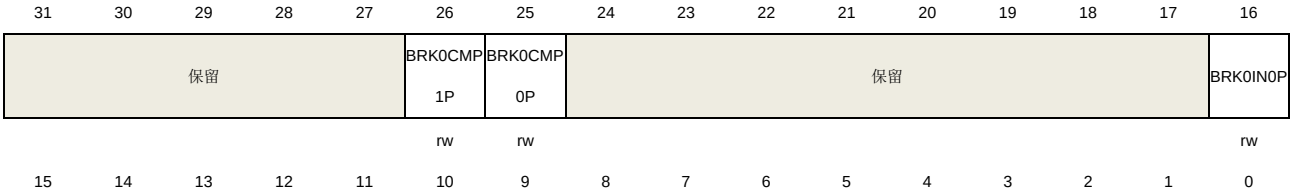
位/位域	名称	描述
31:22	保留	必须保持复位值
21:20	MCH0MSEL[1:0]	多模式通道 0 模式选择 00：独立模式，MCH0 独立于 CH0 01：保留 10：保留 11：互补模式，只有 CH0 可用于输入，MCH0 输出与 CH0 输出互补
19:0	保留	必须保持复位值

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。



保留	BRK0CMP1EN	BRK0CMP0EN	BRK0HPDFEN	保留	BRK0IN0EN
	rw	rw	rw		rw

位/位域	名称	描述
31:27	保留	必须保持复位值。
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_CCHP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_CCHP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_CCHP寄存器的PROT [1:0] =00时才可修改。
15:11	保留	必须保持复位值。
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_CCHP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_CCHP寄存器的PROT [1:0] =00时才可修改。
8	BRK0HPDFEN	BREAK0 HPDF输入（hpdf_break[x]，请参考*表45-2. HPDF断路连接*）使能 0：HPDF输入禁能 1：HPDF输入使能 此位只有在TIMERx_CCHP寄存器的PROT [1:0] =00时才可修改。

7:1	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_CCHP寄存器的PROT [1:0] =00时才可修改。

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。

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’b100101：传输 38 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

24.5. 基本定时器（TIMERx, x=5,6,50,51）

24.5.1. 简介

基本定时器（TIMER5/6/50/51）包含一个无符号 32 位或 64 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。

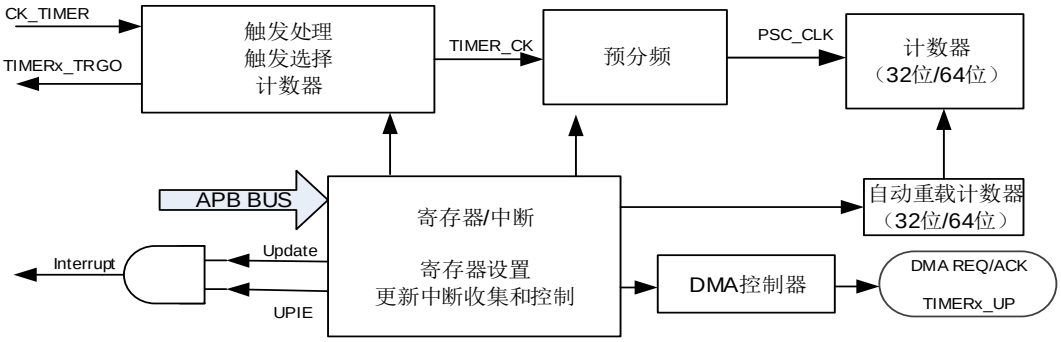
24.5.2. 主要特性

- 计数器宽度：32位（TIMER5/6）、64位（TIMER50/51）
- 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

24.5.3. 结构框图

*图 24-126. 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 24-126. 基本定时器结构框图



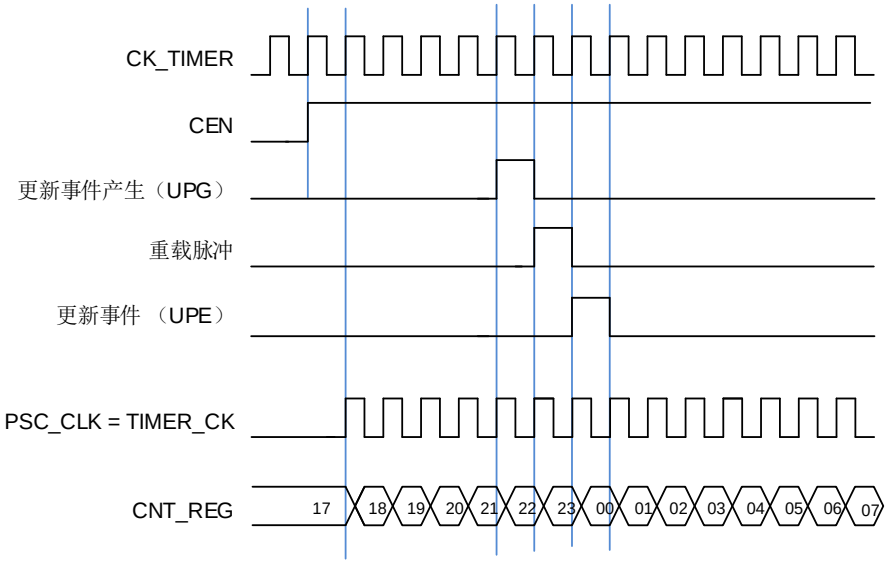
24.5.4. 功能描述

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。

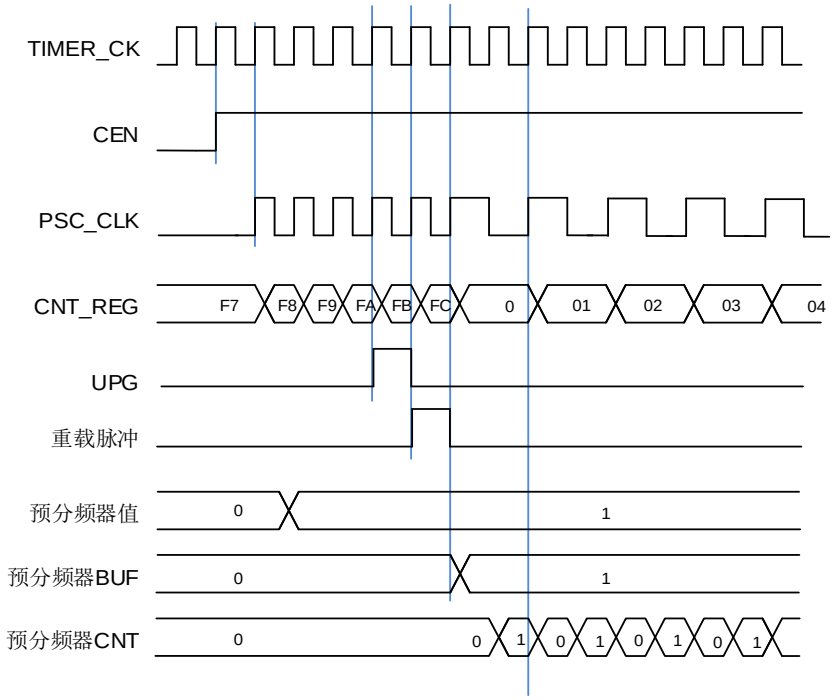
图 24-127. 内部时钟分频为 1 时正常模式下的控制电路



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 24-128. 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/ TIMERx_CARL/ TIMERx_CARH 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（TIMERx, x=5,6）时，计数器在不同预分频因子下的行为。

图 24-129. 向上计数时序图，PSC=0/2（TIMERx, x=5,6）

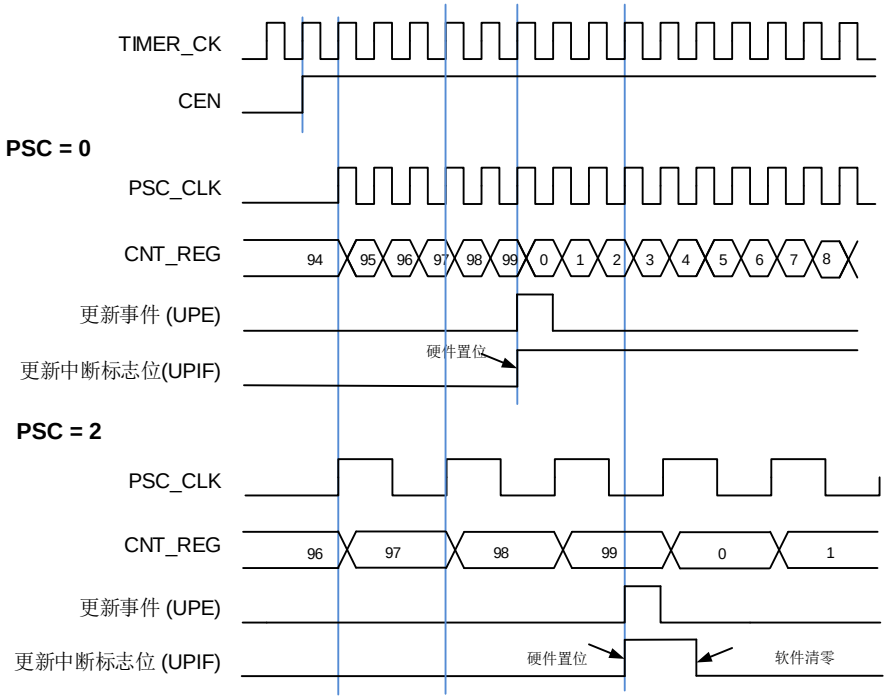
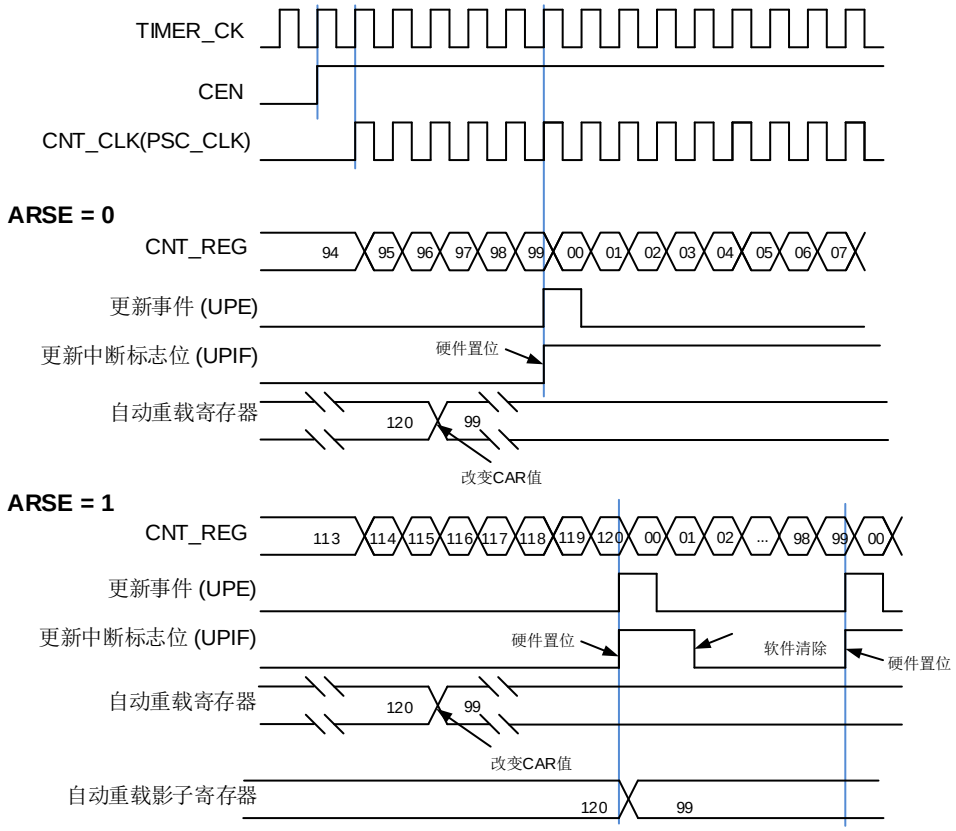


图 24-130. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值（TIMERx, x=5,6）



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®-M7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。

24.5.5. TIMERx 寄存器（x=5/6/50/51）

TIMER5基地址：0x4000 1000

TIMER6基地址：0x4000 1400

TIMER50基地址：0x4000 F000

TIMER51基地址：0x4000 F400

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				UPIFBUEN	保留			ARSE	保留			SPM	UPS	UPDIS	CEN
				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	UPIFBUEN	UPIF位备份使能 0：备份禁能。UPIF位没有备份到TIMERx_CNT/ TIMERx_CNTH寄存器中的UPIFBU位 1：备份使能。UPIF位备份到TIMERx_CNT/ TIMERx_CNTH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/ TIMERx_CARL/ TIMERx_CARH 寄存器的影子寄存器 1：使能 TIMERx_CAR/ TIMERx_CARL/ TIMERx_CARH 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。

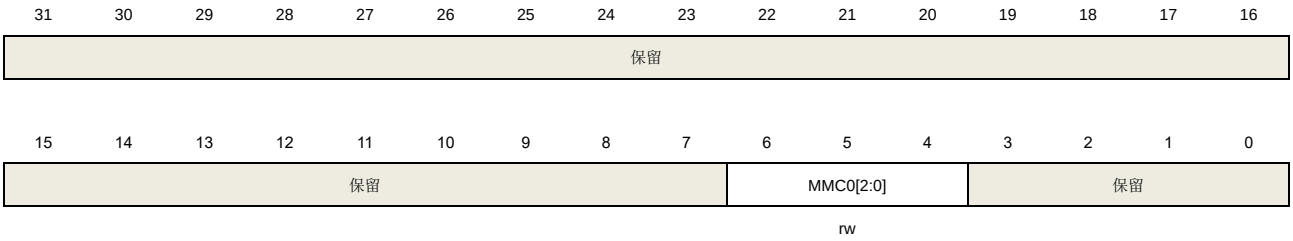
1	UPDIS	禁止更新 该位用来使能或禁能更新事件的产生。 0：更新事件使能。当以下事件之一发生时，更新事件产生，具有缓存的寄存器被装入它们的预装载值： – UPG 位被置 1 – 计数器上溢 – 从模式控制器产生一个更新事件 1：更新事件禁能。带有缓存的寄存器保持原有值，如果 UPG 位被置 1 或者从模式控制器产生一个硬件复位事件，计数器和预分频器被重新初始化。
0	CEN	计数器使能 0：计数器禁能 1：计数器使能 在软件将 CEN 位置 1 后，外部时钟、暂停模式和译码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。

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。

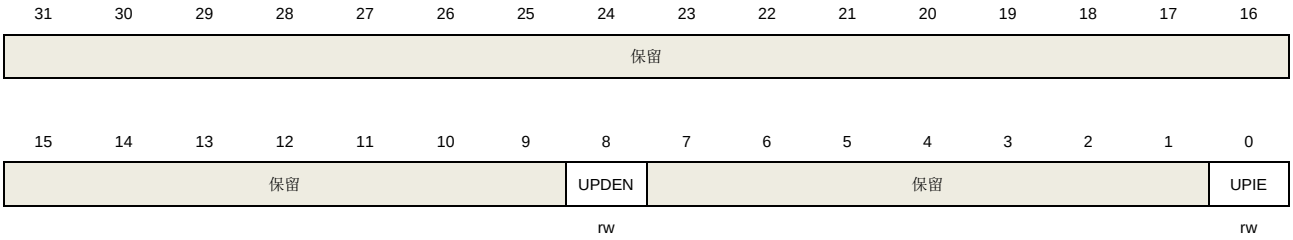
010：更新。主模式控制器选择更新事件作为 TRGO0。
100~111：保留。

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。

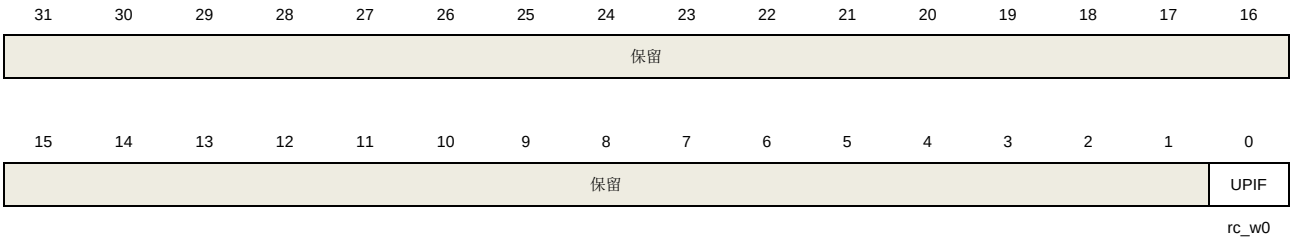


位/位域	名称	描述
31:9	保留	必须保持复位值
8	UPDEN	更新 DMA 请求使能 0：禁止更新 DMA 请求 1：使能更新 DMA 请求
7:1	保留	必须保持复位值
0	UPIE	更新中断使能 0：禁止更新中断 1：使能更新中断

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。



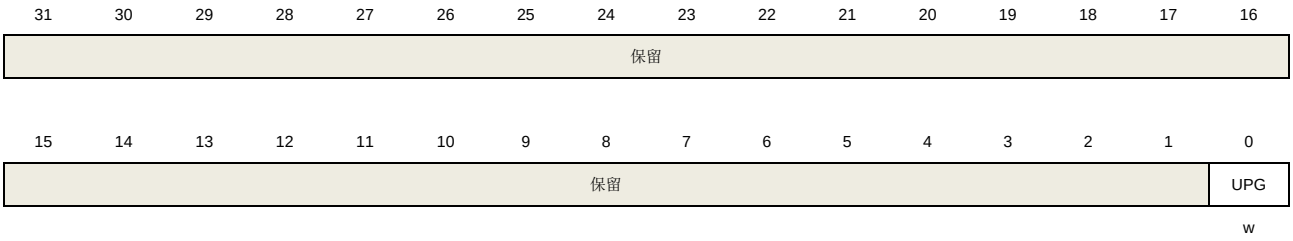
位/位域	名称	描述

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。

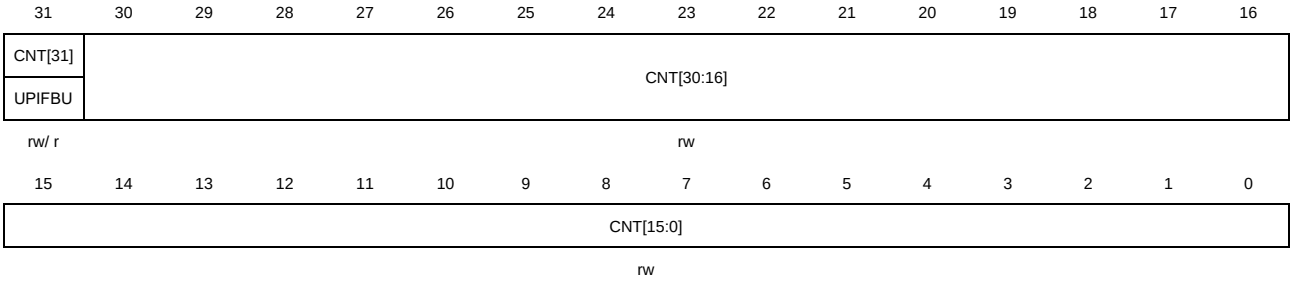


位/位域	名称	描述
31:1	保留	必须保持复位值
0	UPG	更新事件产生 此位由软件置 1，被硬件自动清 0。当此位被置 1 并且向上计数模式，计数器被清 0，预分频计数器将同时被清除。 0：无更新事件产生 1：产生更新事件

计数器寄存器（TIMERx_CNT）（TIMERx, x=5,6）

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字（32位）访问。



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。

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。
30:0	CNT[30:0]	这些位是当前的计数值。写操作能改变计数器值。

计数器低位寄存器（TIMERx_CNTL）（TIMERx, x=50,51）

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
CNTL[31:16]															
rw															

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

位/位域	名称	描述
31:0	CNTL[31:0]	这些位是当前的计数器低位值。写操作能改变计数器值。

预分频寄存器（TIMERx_PSC）

地址偏移：0x28
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 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=5,6）

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

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

计数器自动重载低位寄存器（TIMERx_CARL）（TIMERx, x=50,51）

地址偏移：0x2C

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

位/位域	名称	描述
31:0	CARLL[31:0]	计数器低位自动重载值 这些位定义了计数器的低位自动重载值。

计数器高位寄存器（TIMERx_CNTH）（TIMERx, x=50,51）

地址偏移：0xD0

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNTH[31]	CNTH[63:48]														
UPIFBU															
rw/ r	rw														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CNTH[47:32]

rw

UPIFBUEN = 0:

位/位域	名称	描述
31:0	CNTH[63:32]	这些位是当前的计数器高位值。写操作能改变计数器值。

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。
30:0	CNTH[62:32]	这些位是当前的计数器高位值。写操作能改变计数器值。

计数器自动重载低位寄存器（TIMERx_CARH）（TIMERx, x=50,51）

地址偏移：0xD4

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
CARLH[63:48]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARLH[47:32]															
rw															

位/位域	名称	描述
31:0	CARLH[63:32]	计数器高位自动重载值 这些位定义了计数器的高位自动重载值。

25. 通用同步异步收发器（USART）

25.1. 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_IRC64MDIV）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。

25.2. 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300MHz，过采样为8，最高速度可达37.5MBits/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：接收缓冲区非空（RBNE）、接收FIFO满（RFF）、接收FIFO空（RFE）、接收FIFO已达阈值（RFT）、发送缓冲区空（TBE）、传输已完成（TC）、发送FIFO未满（TFNF）、发送FIFO空（TFE）、发送FIFO已达阈值（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和USART5完全实现上述功能，但是UART3、UART4、UART6和UART7只实现了上面所介绍的部分功能，下面这些功能在UART3、UART4、UART6和UART7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。

25.3. 功能说明

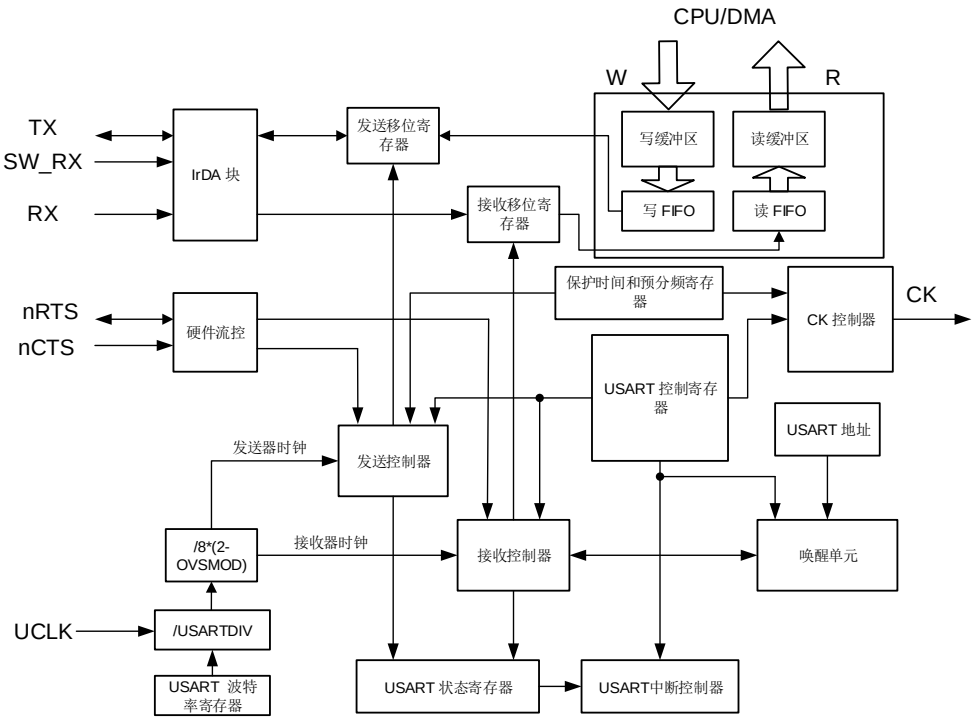
USART 接口通过*表 25-1. USART 重要引脚描述*中主要引脚从外部连接到其他设备。

表 25-1. USART 重要引脚描述

引脚	类型	描述
RX	输入	接收数据

引脚	类型	描述
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

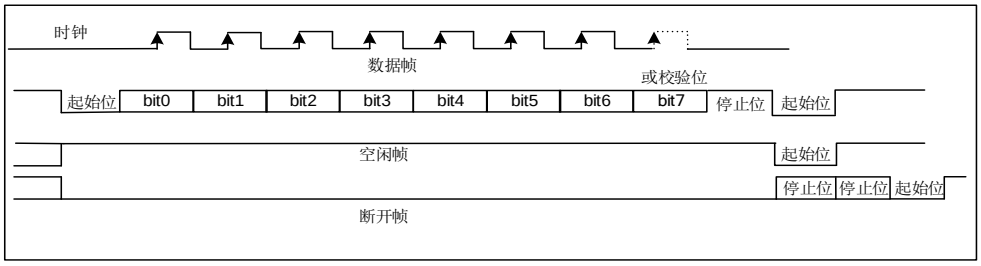
图 25-1. USART 模块内部框图



25.3.1. 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25-2. 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 25-2. 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。

25.3.2. 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：

$$\text{USARTDIV}=\frac{\text{UCLK}}{16\times\text{Baud Rate}} \tag{25-1}$$

如果过采样是8，公式为：

$$\text{USARTDIV}=\frac{\text{UCLK}}{8\times\text{Baud Rate}} \tag{25-2}$$

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。

25.3.3. 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。

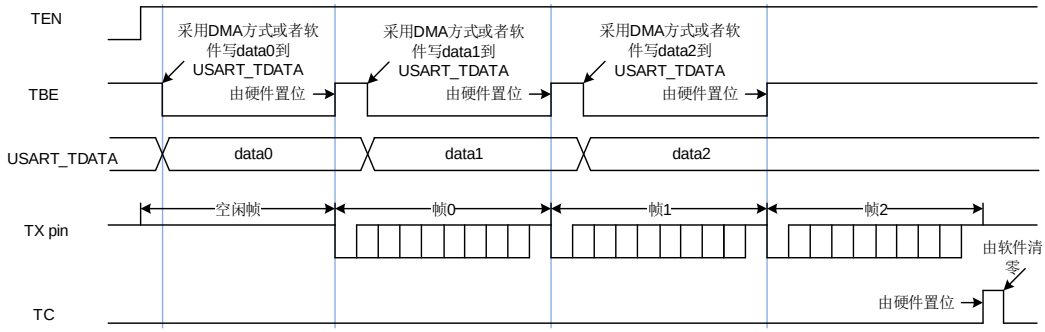
系统上电后，TBE默认为高电平。在USART_STAT寄存器中TBE置位时，数据可以在不覆盖前一个数据的情况下写入USART_TDATA寄存器。当数据写入USART_TDATA寄存器，TBE位将被清0。在数据由USART_TDATA移入移位寄存器后，该位由硬件置1。如果数据在一个发送过程正在进行时被写入USART_TDATA寄存器，它将首先被存入发送缓冲区，在当前发送过程完成时传输到发送移位寄存器中。如果数据在写入USART_TDATA寄存器时，没有发送过程正在进行，TBE位将被清零然后迅速置位，原因是数据被立刻传输到发送移位寄存器。

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 25-3. 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 25-3. 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。

25.3.4. USART 接收器

上电后，按以下步骤使能USART接收器：

1. 写USART_CTL0寄存器的WL位去设置字长；
2. 在USART_CTL1寄存器中写STB[1:0]位来设置停止位的长度；
3. 如果选择了多级缓存通信方式，应该在USART_CTL2寄存器中使能DMA（DENR位）；
4. 在USART_BAUD寄存器中设置波特率；
5. 在USART_CTL0寄存器中置位UEN位，使能USART；
6. 在USART_CTL0中设置REN位。

接收器在使能后若检测到一个有效的起始脉冲便开始接收码流。在接收一个数据帧的过程中会检测噪声错误，奇偶校验错误，帧错误和过载错误。

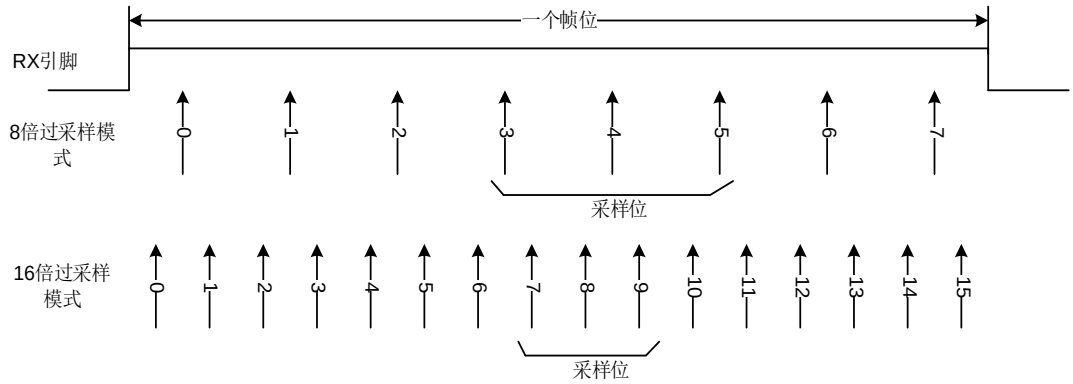
当接收到一个数据帧，USART_STAT寄存器中的RBNE置位，如果设置了USART_CTL0寄存器中相应的中断使能位RBNEIE，将会产生中断。在USART_STAT寄存器中可以观察接收状态标志。

软件可以通过读USART_RDATA寄存器或者DMA方式获取接收到的数据。不管是直接读寄存器还是通过DMA，只要是对USART_RDATA寄存器的一个读操作都可以清除RBNE位。

在接收过程中，需使能REN位，不然当前的数据帧将会丢失。

在默认情况下，接收器通过获取三个采样点的值来估计该位的值。如果是8倍过采样模式，选择第3、4、5个采样点；如果是16倍过采样模式，选择第7、8、9个采样点。如果在3个采样点中有2个或3个为0，该数据位被视为0，否则为1。如果3个采样点中有一个采样点的值与其他两个不同，不管是数据位，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 25-4. 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。

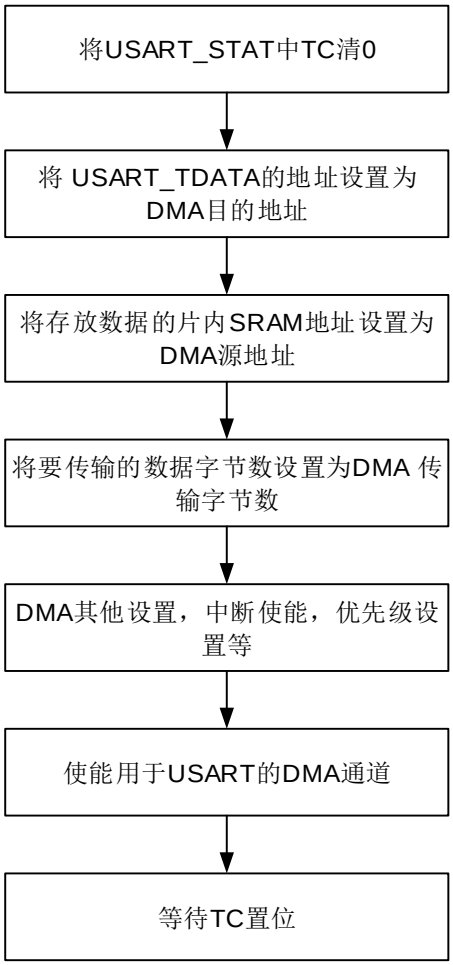
若接收过程中，产生了噪声错误（NERR）、校验错误（PERR）、帧错误（FERR）或溢出错误（ORERR），则NERR、PERR、FERR或ORERR将和RBNE同时置位。如果没有使能DMA，RBNE中断发生时，软件需检查是否有噪声错误、校验错误、帧错误或溢出错误产生。

25.3.5. 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 25-5. 采用 DMA 方式实现 USART 数据发送配置步骤*所示。

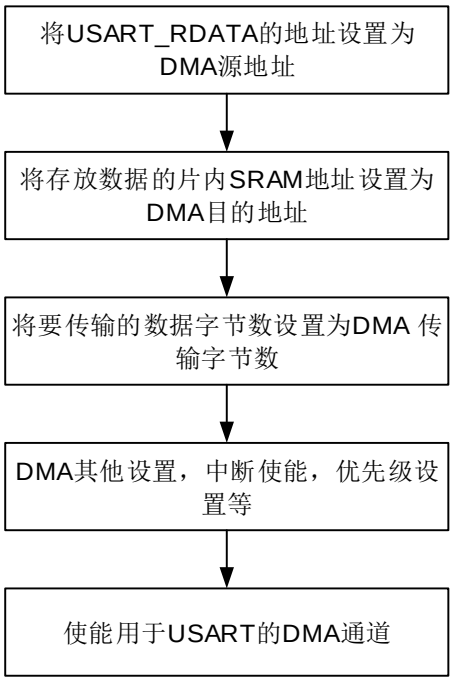
图 25-5. 采用 DMA 方式实现 USART 数据发送配置步骤



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 25-6. 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 25-6. 采用 DMA 方式实现 USART 数据接收配置步骤

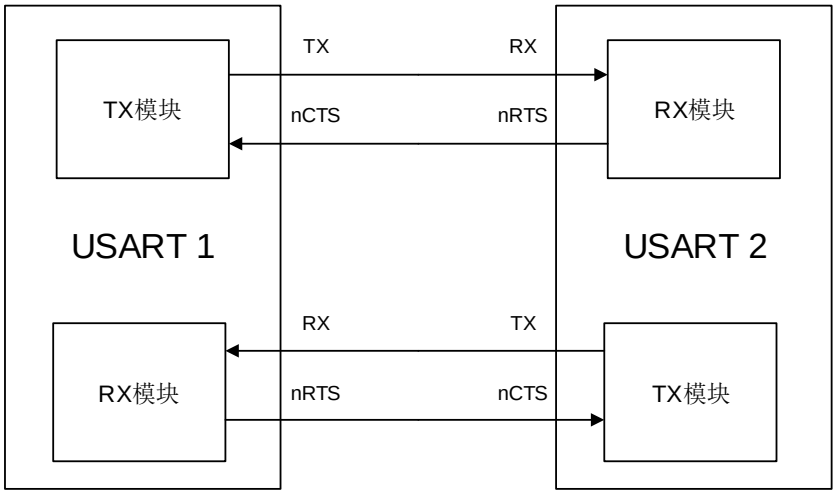


当USART接收到的数据数量达到了DMA传输数据数量，DMA模块将产生传输完成中断。

25.3.6. 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 25-7. 两个 USART 之间的硬件流控制



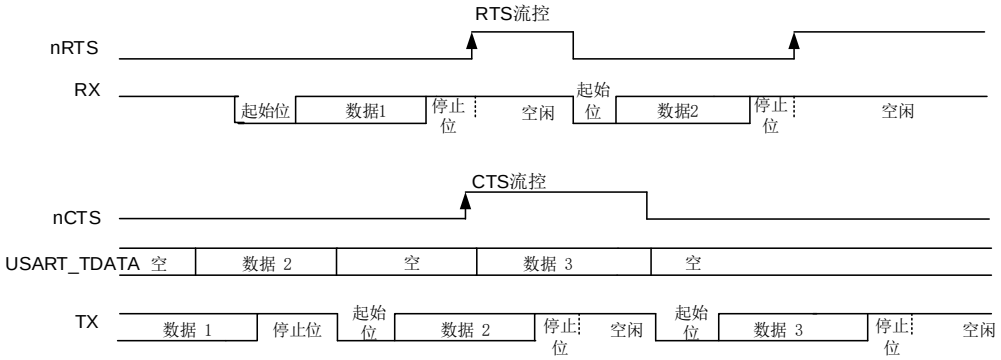
RTS 流控

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 25-8. 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。

25.3.7. 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。

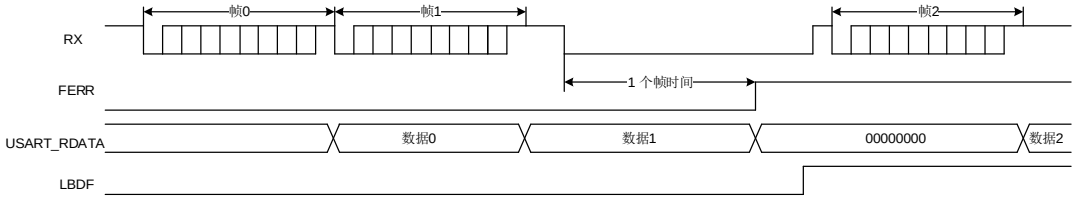
25.3.8. LIN 模式

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。

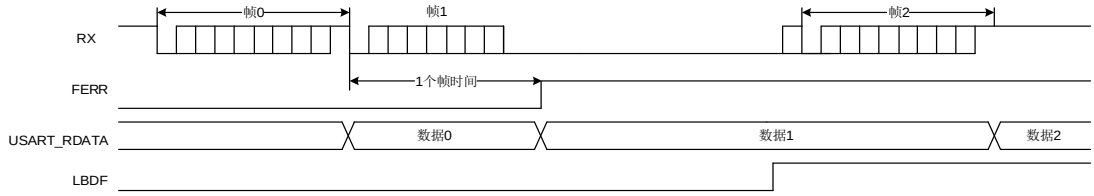
断开检测功能完全独立于普通USART接收器。因此，断开检测可以是在空闲状态下，也可以在数据传输过程中。USART_CTL1寄存器中LBLEN位可以选择断开帧的长度。如果在RX引脚检测到大于或等于与预期的断开帧长度的0（LBLEN = 0时，10个0；LBLEN = 1时，11个0），USART_STAT寄存器中LBDF置位。如果USART_CTL1寄存器中LBDIE被置位，将产生中断。如*图25-9. 空闲状态下检测断开帧*所示，如果断开帧发生在空闲状态下，USART接收器会接收到一个全0数据帧，同时FERR置位。

图 25-9. 空闲状态下检测断开帧



如*图 25-10. 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 25-10. 数据传输过程中检测断开帧



25.3.9. 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 25-11. 同步模式下的 USART 示例

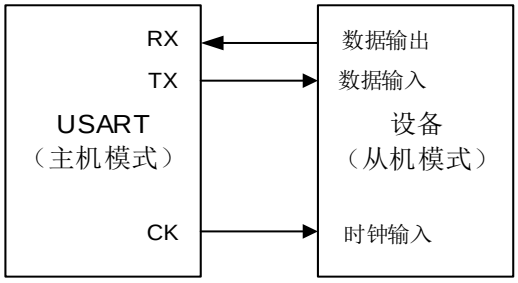
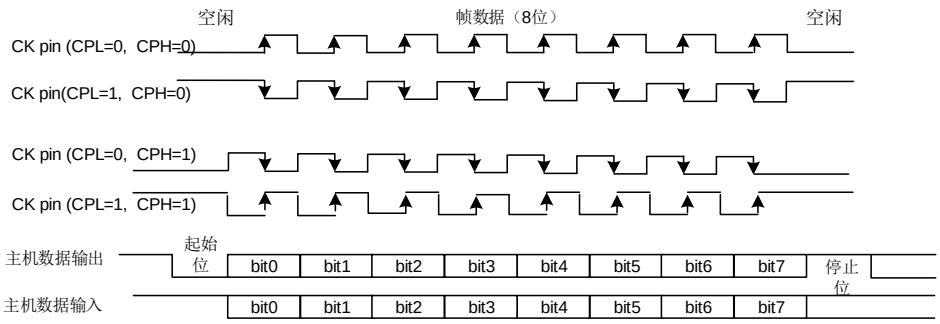


图 25-12. 8-bit 格式的 USART 同步通信波形（CLEN = 1）

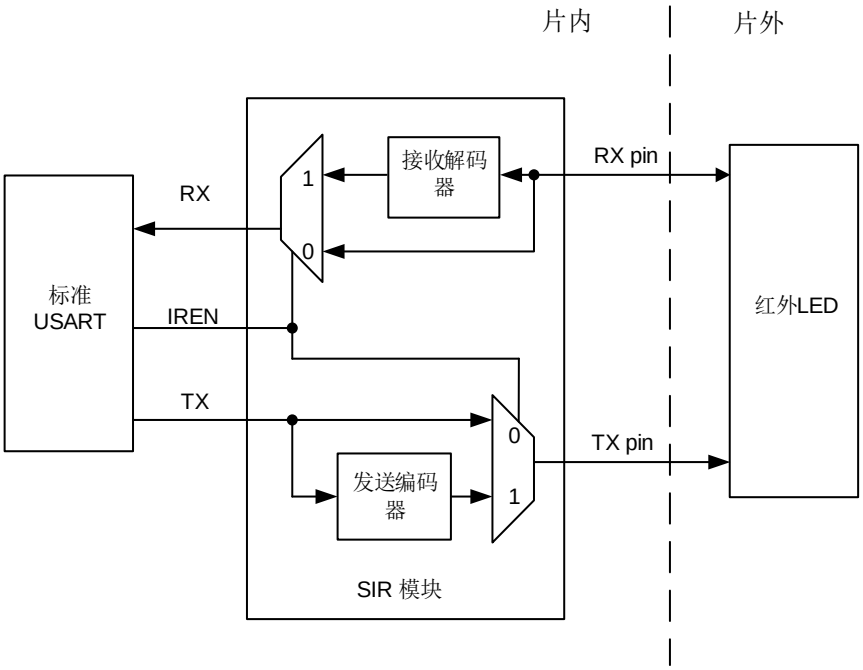


25.3.10. 串行红外（IrDA SIR）编解码功能模块

串行红外编解码功能通过置位USART_CTL2寄存器中IREN使能。在IrDA模式下，USART_CTL1寄存器的LMEN，STB[1:0]，CKEN位和USART_CTL2寄存器的HDEN，SCEN位应被清0。

在IrDA模式下，USART数据帧由SIR发送编码器进行调制，调制后的信号经由红外LED进行发送，经解调后将数据发送至USART接收器。对于编码器而言，波特率应小于115200。

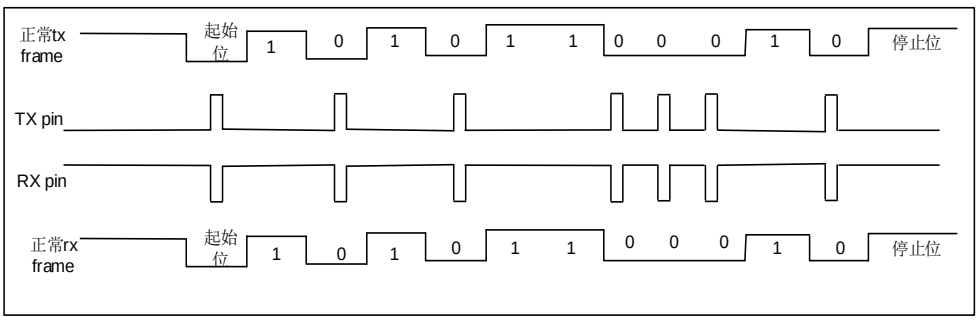
图 25-13. IrDA SIR ENDEC 模块



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 25-14. IrDA 数据调制



将USART_CTL2寄存器中IRLP置位可以使SIR子模块工作在低功耗模式下。发送编码器由PCLK分频得到的低速时钟来驱动。分频系数在USART_GP寄存器中PSC[7:0]位配置，

USART_BAUD寄存器需配置为16*PSC[7:0]。TX引脚脉冲宽度可以为低功耗波特率的3倍。接收解码器工作模式与正常IrDA模式相同。

25.3.11. 半双工通信模式

通过设置USART_CTL2寄存器的HDEN位，可以使能半双工模式。在半双工通信模式下，USART_CTL1寄存器的LMEN，CKEN位和USART_CTL2寄存器的SCEN，IREN位应被清零。

半双工模式下仅用单线通信，TX引脚和RX引脚从内部连接到一起，RX引脚不再使用。TX引脚应被配置为开漏模式，通信冲突由软件处理。当TEN被置位时，在数据寄存器中的数据将会被发送。

25.3.12. 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 25-15. 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。

25.3.13. 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。

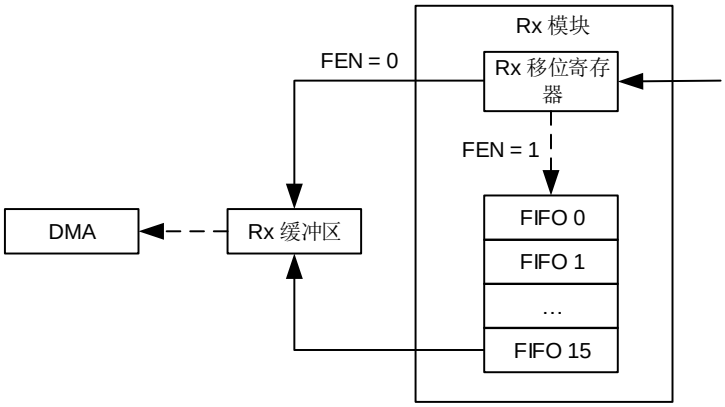
为了检测空闲状态，必须置位USART_CTL1寄存器的RTEN位和USART_CTL0寄存器的RTIE位。USART_RT寄存器必须被设置成与2个字节超时所对应的值。在最后一个停止位被接收后，当接收线在这期间是空闲的，将产生一个中断，通知软件当前块接收已经完成。

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。

25.3.14. 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 25-16. USART 接收 FIFO 结构

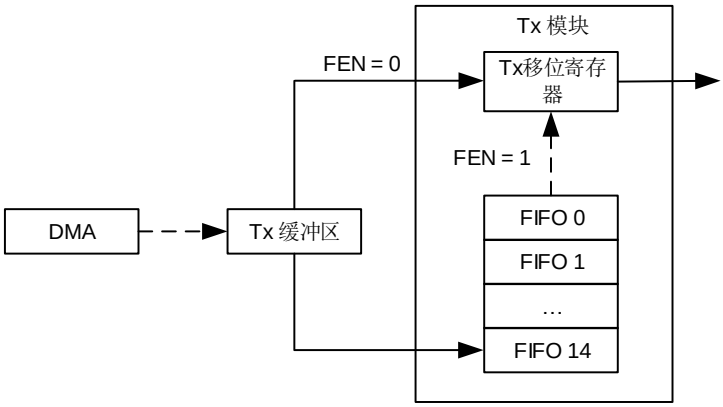


如果软件在响应RBNE中断时读数据接收缓冲区，在响应开始时，RBNEIE位应清0。当所有接收的数据被读出后，RBNEIE位应置位。在读出接收的数据前，PERR，NERR，FERR，EBF都应被清0。

25.3.15. 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可用空间达到TFTCFG[2:0]位域配置的阈值，TFTIF将被置位（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 25-17. USART 发送 FIFO 结构



25.3.16. 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_IRC64MDIV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。

25.3.17. USART 中断

USART 中断事件和标志如*表 25-3. USART 中断请求*所示：

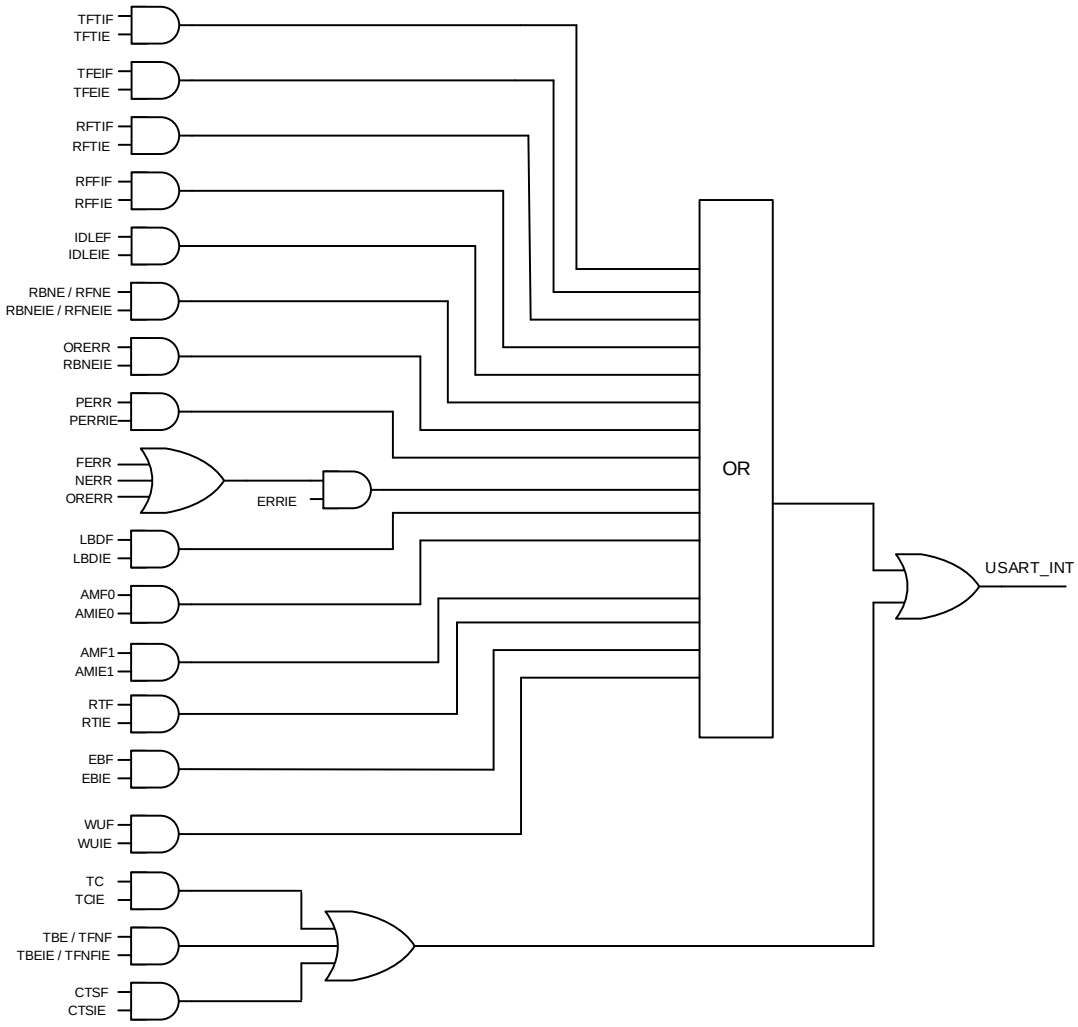
表 25-3. USART 中断请求

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	RFT	RFTIE
发送FIFO空	TFE	TFEIE

中断事件	事件标志	使能控制位
发送FIFO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 25-18. USART 中断映射框图



25.4. USART 寄存器

USART0 基地址：0x4001 1000

USART1 基地址：0x4000 4400

USART2 基地址：0x4000 4800

UART3 基地址：0x4000 4C00

UART4 基地址：0x4000 5000

USART5 基地址：0x4001 1400

UART6 基地址：0x4000 7800

UART7基地址：0x4000 7C00

25.4.1. 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 / UART6 / UART7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 / UART6 / UART7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_IRC64MDIV 或 CK_LXTAL。 在UART3 / UART4 / UART6 / UART7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

25.4.2. 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 / UART6 / UART7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 / UART6 / UART7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 / UART6 / UART7。
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 / UART6 / UART7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 / UART6 / UART7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 / UART6 / UART7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

25.4.3. 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 / UART6 / UART7，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 / UART6 / UART7，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 / UART6 / UART7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

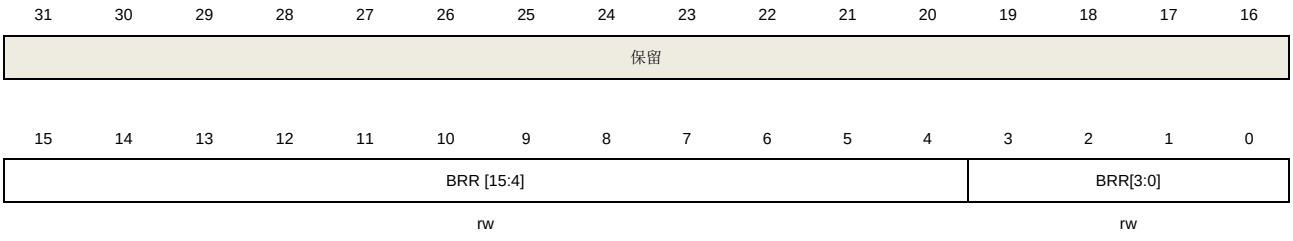
		0：智能卡模式禁用 1：智能卡模式使能 USART被使能（UEN = 1）时，该位域不能被改写。 在UART3 / UART4 / UART6 / UART7中，该位保留。
4	NKEN	智能卡模式NACK使能 0：当出现校验错误时不发送NACK 1：当出现校验错误时发送NACK USART被使能（UEN = 1）时，该位域不能被改写。 在UART3 / UART4 / UART6 / UART7中，该位保留。
3	HDEN	半双工使能 0：禁用半双工模式 1：开启半双工模式 USART被使能（UEN = 1）时，该位域不能被改写。
2	IRLP	IrDA低功耗模式 0：正常模式 1：低功耗模式 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 / UART6 / UART7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。

25.4.4. 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。

25.4.5. 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 / UART6 / UART7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

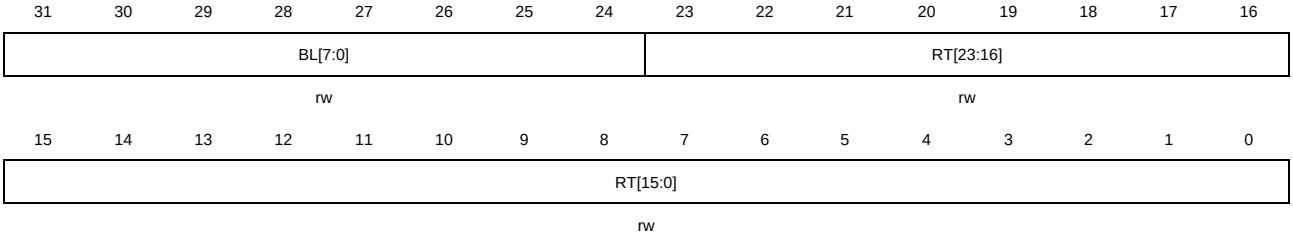
00010：4分频
00011：6分频
...
USART被使能（UEN = 1）时，该位域不能被改写。

25.4.6. USART 接收超时寄存器（USART_RT）

地址偏移：0x14
复位值：0x0000 0000

在UART3 / UART4 / UART6 / UART7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。

25.4.7. 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。

25.4.8. 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

		在UART3 / UART4 / UART6 / UART7中，该位保留。
21	TEA	发送使能通知标志 该位反映了USART核心逻辑的发送使能状态，该位可以通过硬件设置。 0：USART核心发送逻辑禁用 1：USART核心发送逻辑被使能
20	WUF	从深度睡眠模式唤醒标志 0：没有从深度睡眠模式唤醒 1：已从深度睡眠模式唤醒， 如果在USART_CTL2寄存器的WUFIE = 1并且MCU处于深度睡眠模式，将引发一个中断。 当检测到一个唤醒事件时，该位通过硬件置位，这个事件在WUM位域被定义。 向USART_INTC寄存器中的WUC写1，该位被清0。 当UESM被清0时，该位清0。 在UART3 / UART4 / UART6 / UART7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 / UART6 / UART7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 / UART6 / UART7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 / UART6 / UART7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。

25.4.9. 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 / UART6 / UART7中，该位保留。
19:18	保留	必须保持复位值。

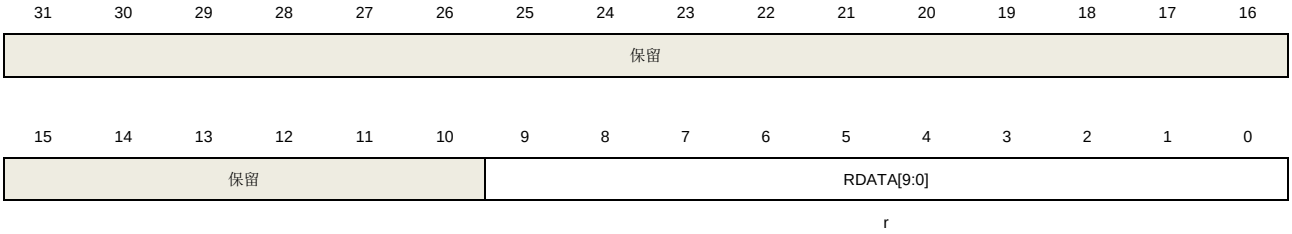
17	AMC0	ADDR0中字符匹配标志清除 向该位写1清除USART_STAT寄存器的AMF0位。
16	AMC1	ADDR1中字符匹配标志清除 向该位写1清除USART_STAT寄存器的AMF1位。
15:13	保留	必须保持复位值。
12	EBC	块结束标志清除 向该位写1清除USART_STAT寄存器的EBF位。 在UART3 / UART4 / UART6 / UART7中，该位保留。
11	RTC	接收超时标志清除 向该位写1清除USART_STAT寄存器的RTF标志。 在UART3 / UART4 / UART6 / UART7中，该位保留。
10	保留	必须保持复位值。
9	CTSC	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 / UART6 / UART7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。

25.4.10. 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。

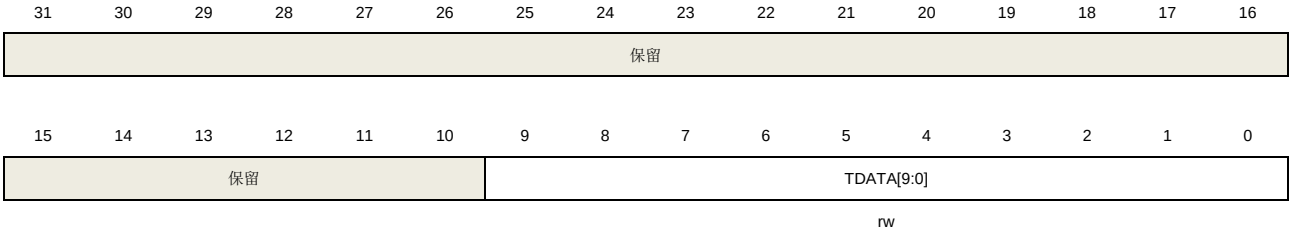


位/位域	名称	描述
31:10	保留	必须保持复位值。
9:0	RDATA[9:0]	接收数据的值 包含接收到的数据字节 如果接收到的数据打开了奇偶校验位（USART_CTL0寄存器的PCEN置1），那么接收到的数据的最高位（第6位、7位、8位或9位，取决于数据的长度）是奇偶校验位。

25.4.11. 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。

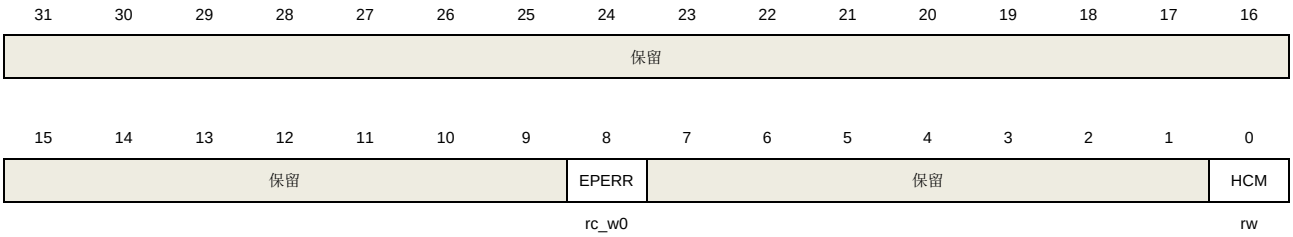


位/位域	名称	描述
31:10	保留	必须保持复位值。
9:0	TDATA[9:0]	发送数据的值 包含发送的数据字节 如果发送到的数据打开了奇偶校验位（USART_CTL0寄存器的PCEN置1），那么发送的数据的最高位（第6位、7位、8位或9位取决于数据的长度）将会被奇偶校验位替代。 只有当USART_STAT寄存器的TBE位被置位时，这个寄存器才可以改写。

25.4.12. 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。



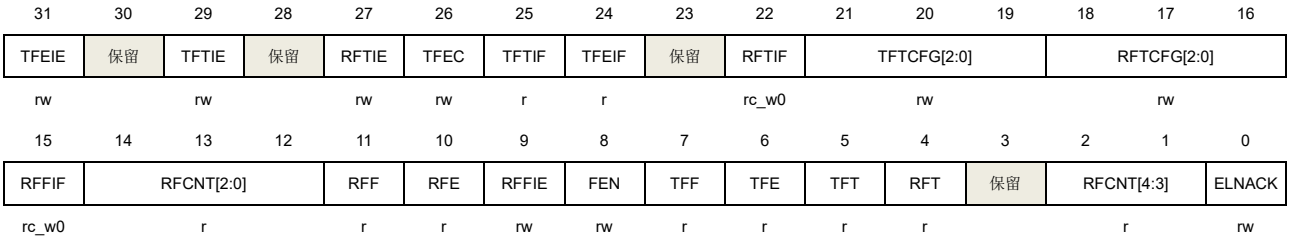
位/位域	名称	描述
31:9	保留	必须保持复位值。
8	EPERR	校验错误超前检测标志。 在RBNE置位前，校验位被检测到时该标志置位。 软件写0可以清除该位。 0：没有检测到校验错误 1：检测到校验错误
7:1	保留	必须保持复位值。
0	HCM	硬件流控制兼容性模式 0：nRTS信号等于RBNE状态寄存器 1：当最后一个数据位（PCE置位时的奇偶位）被采样时，nRTS信号置位

25.4.13. 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。



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 0：发送FIFO可用空间未到达可编程阈值 1：发送FIFO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可用空间到达TFTCFG[2:0]配置的阈值时，由硬件置1。
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可用空间到达可编程阈值 1：发送FIFO可用空间未到达可编程阈值 当发送FIFO可用空间到达TFTCFG[2:0]配置的阈值时，由硬件清0。
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 / UART6 / UART7中，该位保留。

26. 内部集成电路总线接口（I2C）

26.1. 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。

26.2. 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_PMCFG 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。

26.3. 功能说明

I2C 接口的内部结构如*图 26-1. I2C 模块框图*所示。

图 26-1. I2C 模块框图

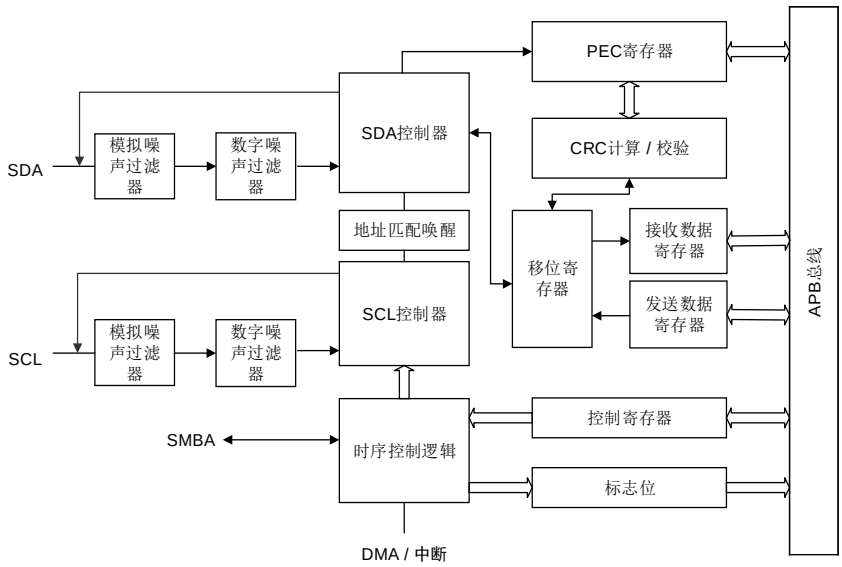


表 26-1. I2C 总线术语说明（参考飞利浦 I2C 规范）

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

26.3.1. 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）可以从以下四个时钟源中选择：

- APB1 时钟 PCLK1（默认值）
- 锁相环（PLL2）
- 内部高速 64M RC 振荡器
- 内部低功耗 4M 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。

26.3.2. I2C 通讯流程

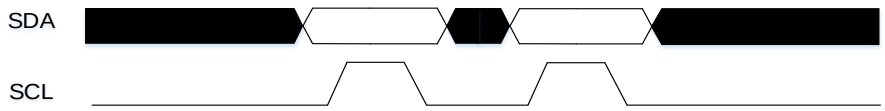
主机和从机都能实现数据收发，因此，I2C 可以实现四种工作模式：

- 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 26-2. 数据有效性*）。每个数据比特传输需要一个时钟脉冲。

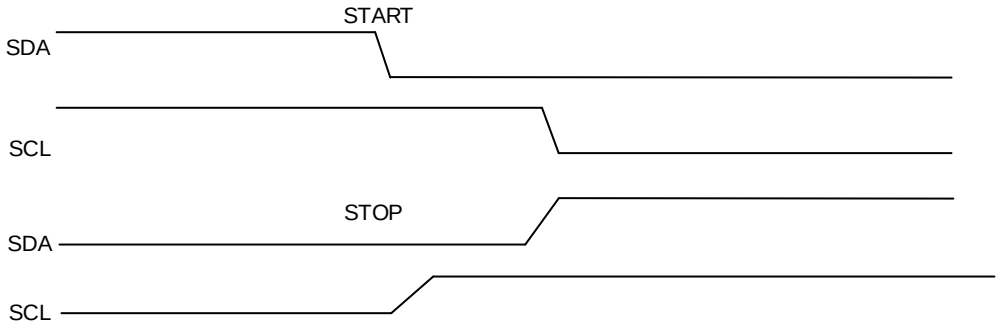
图 26-2. 数据有效性



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 26-3. 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 26-3. 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 26-4. 10 位地址的 I2C 通讯流程（主机发送）

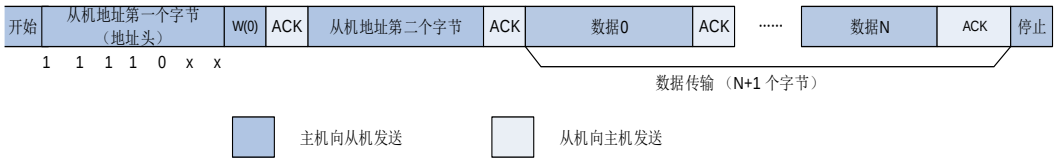


图 26-5. 7 位地址的 I2C 通讯流程（主机发送）

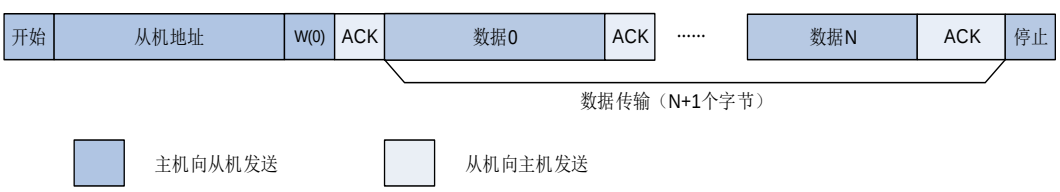
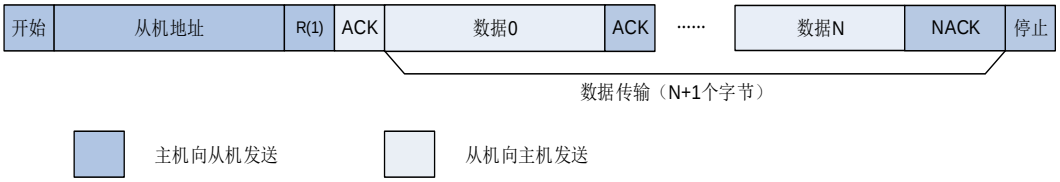


图 26-6. 7 位地址的 I2C 通讯流程（主机接收）



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 26-7. 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 26-8. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R = 1）*所示。

图 26-7. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R = 0）

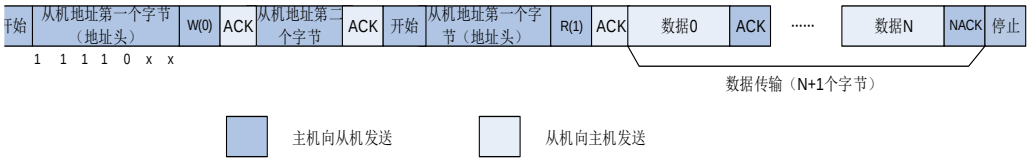
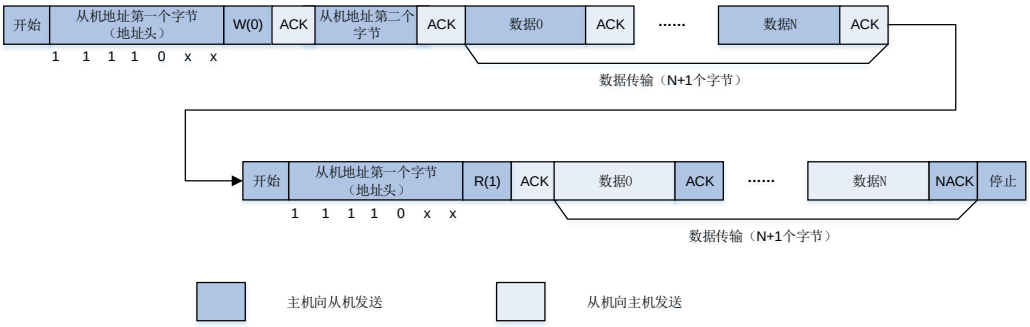


图 26-8. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R = 1）



26.3.3. 噪声滤波器

I2C 外设集成了模拟噪声滤波器和数字噪声滤波器，噪声滤波器可根据实际需要在 I2C 外设启用前进行配置。

将 I2C_CTL0 寄存器中 ANOFF 位置 1 可以禁用模拟噪声滤波器，将 ANOFF 位清 0 时使能模拟噪声滤波器。在快速模式和快速+ 模式下，模拟滤波器需要抑制脉冲宽度高达 130ns 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。

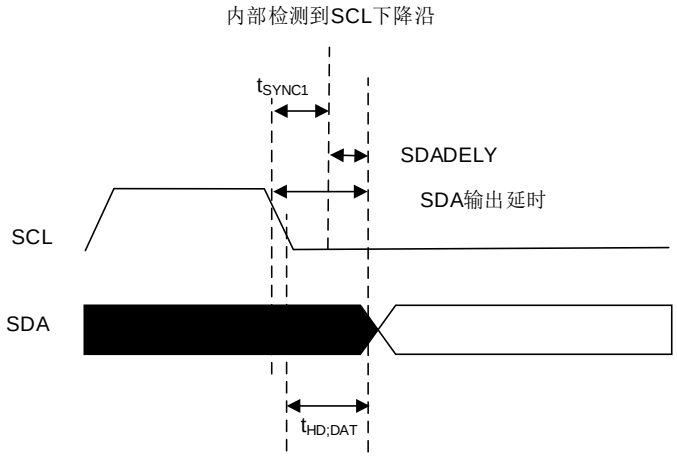
26.3.4. I2C 时序配置

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。

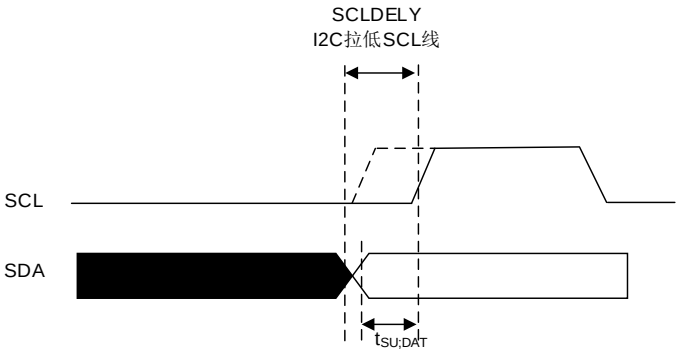
如果数据已经在 I2C_TDATA 寄存器中，在经历 SDADELY 延时后，数据由 SDA 发送，如*图 26-9. 数据保持时间*所示。

图 26-9. 数据保持时间



当数据经过 SDA 发送时，SCLDELY 计数器开启。如图*图 26-10. 数据建立时间*所示。

图 26-10. 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 26-2. 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	

26.3.5. 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：

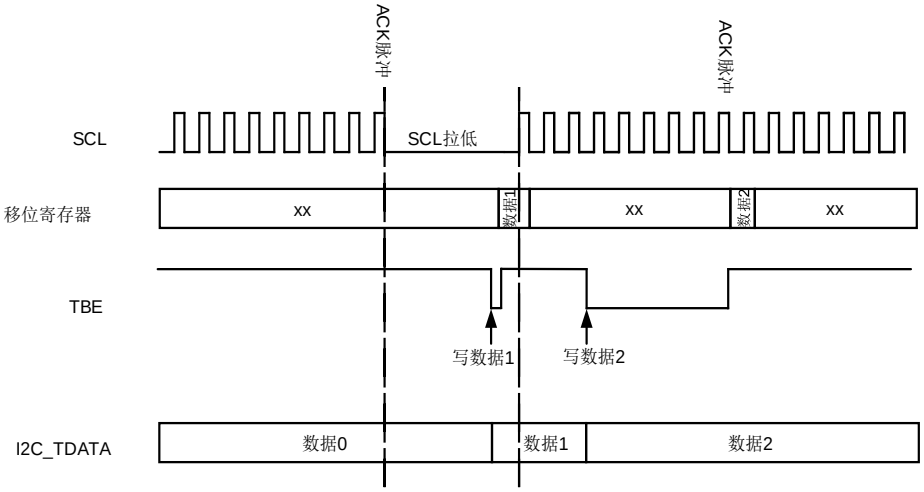
- I2CEN 写 0
- 检查 I2CEN 是否为 0
- I2CEN 写 1

26.3.6. 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。

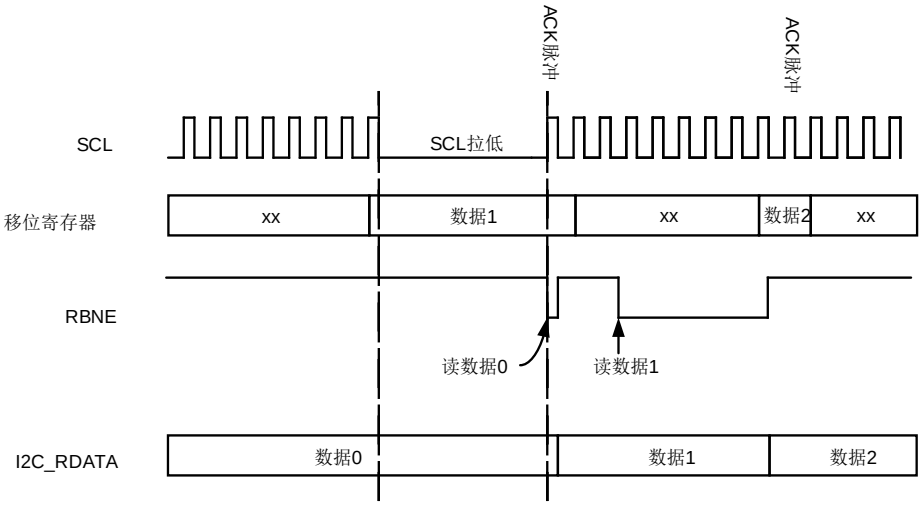
图 26-11. 数据发送



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 26-12. 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 26-3. 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 26-3. 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。

26.3.7. 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：

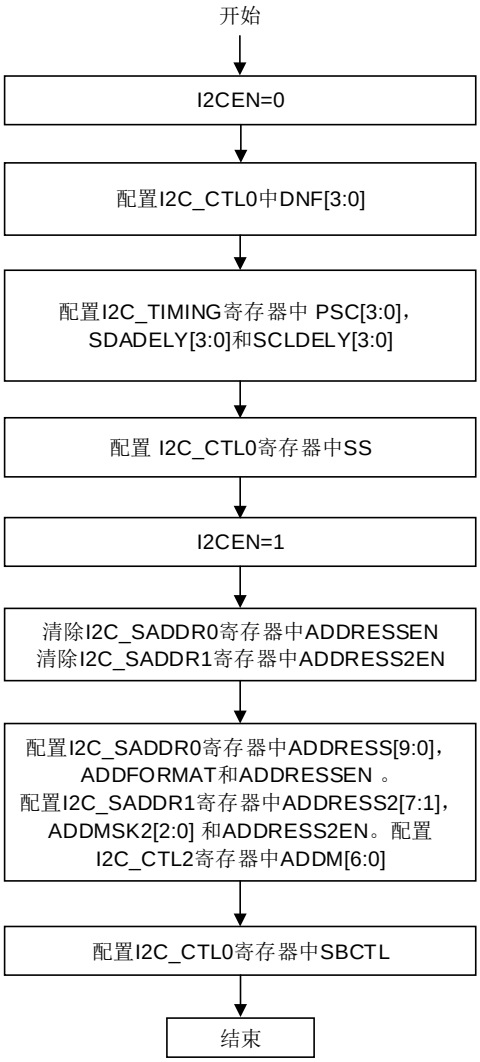
1、 I2CEN = 0

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 26-13. 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。

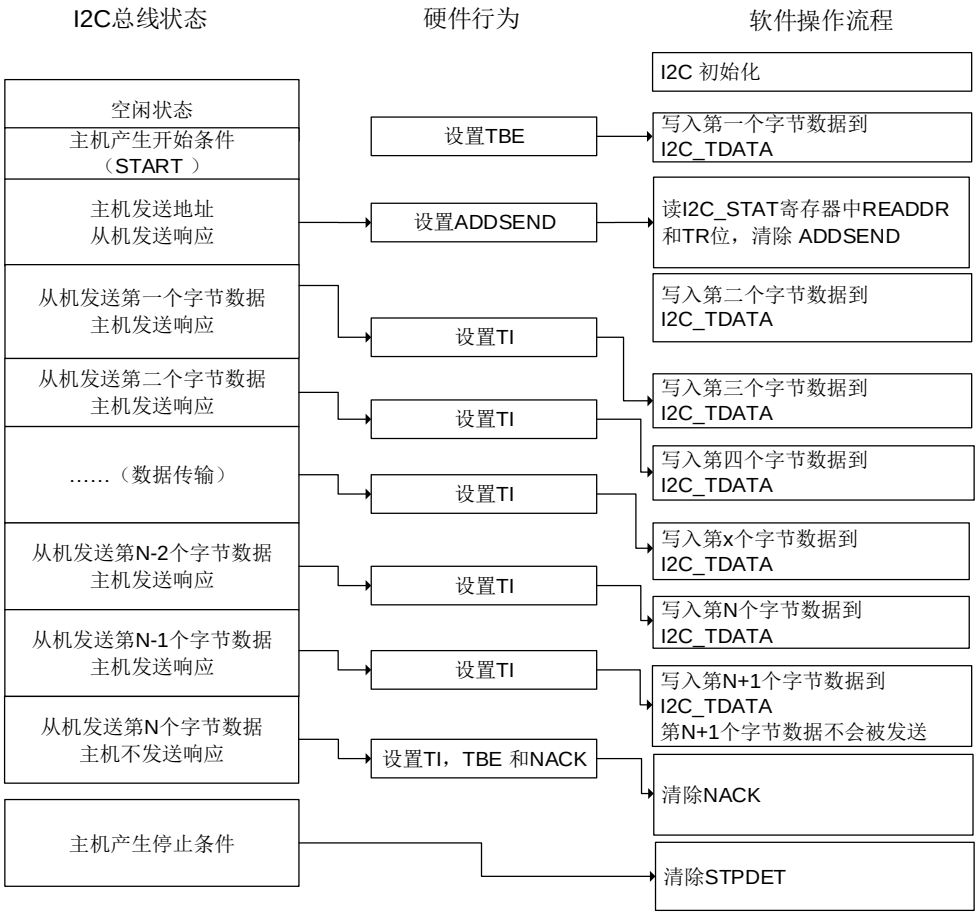
当 SS = 1 时，I2C_STAT 寄存器中 ADDSEND 位置位时 SCL 时钟线不会被拉低。在这种情况下，I2C_TDATA 寄存器中数据不能在 ADDSEND 中断服务程序中清空。因此待发送的第一个字节应该在 ADDSEND 置位之前就被编程到 I2C_TDATA 寄存器。

- 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 26-14. I2C 从机发送编程模型（SS=0）



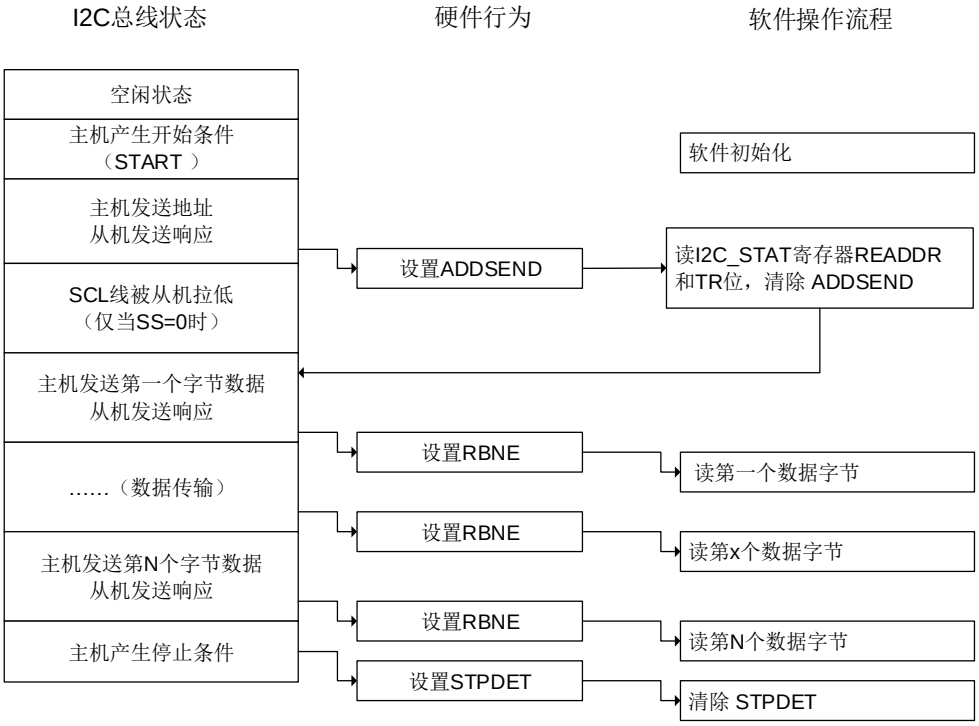
图 26-15. I2C 从机发送编程模型（SS=1）



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 26-16. I2C 从机接收编程模型



26.3.8. 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

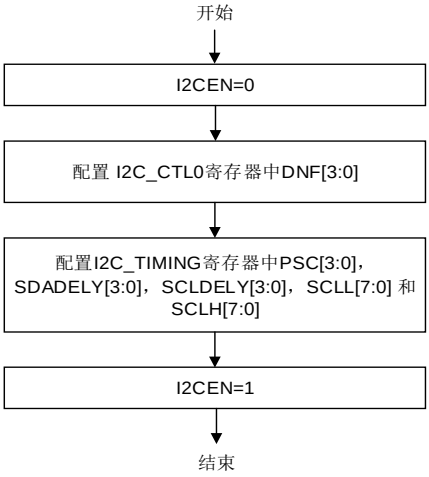
由硬件清零。当从机地址发送完成时，START 位由硬件清零。

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 26-17. 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 26-18. I2C 主机发送编程模型（N<=255）

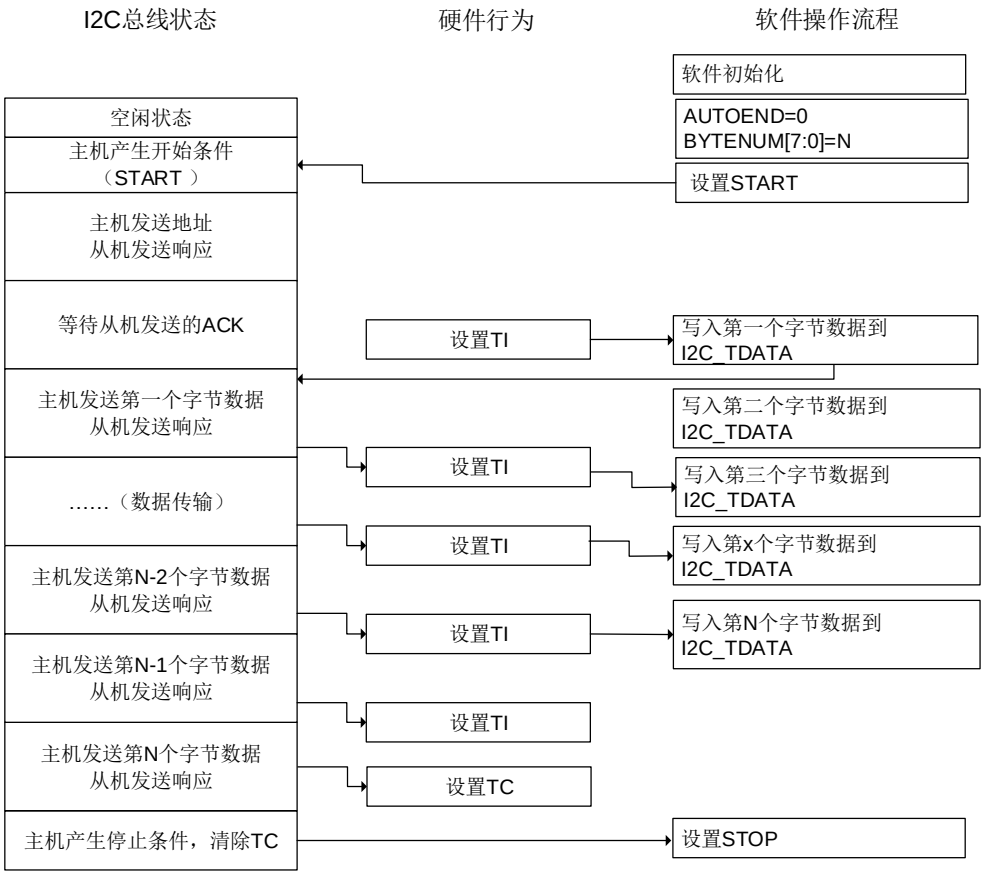
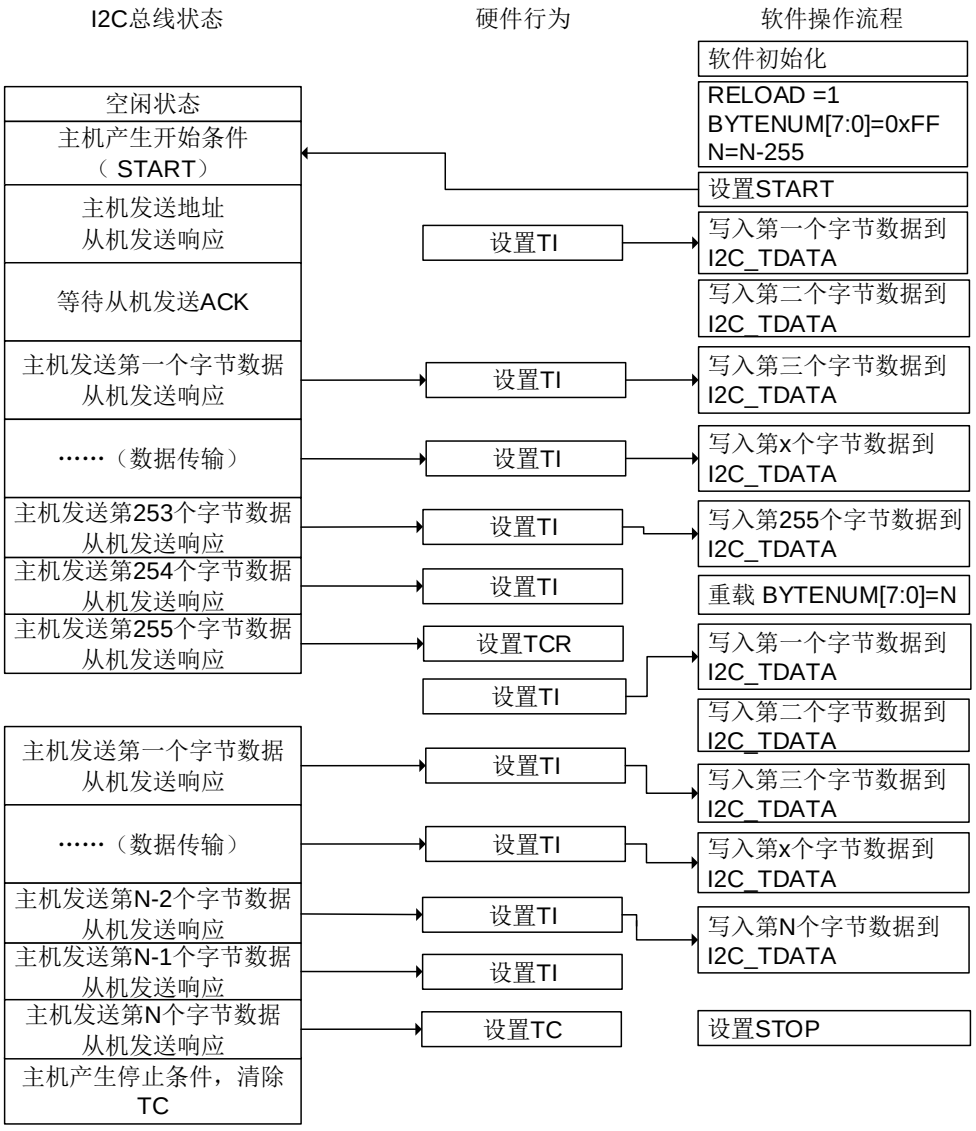


图 26-19. I2C 主机发送编程模型（N>255）



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 26-20. I2C 主机接收编程模型（N<=255）

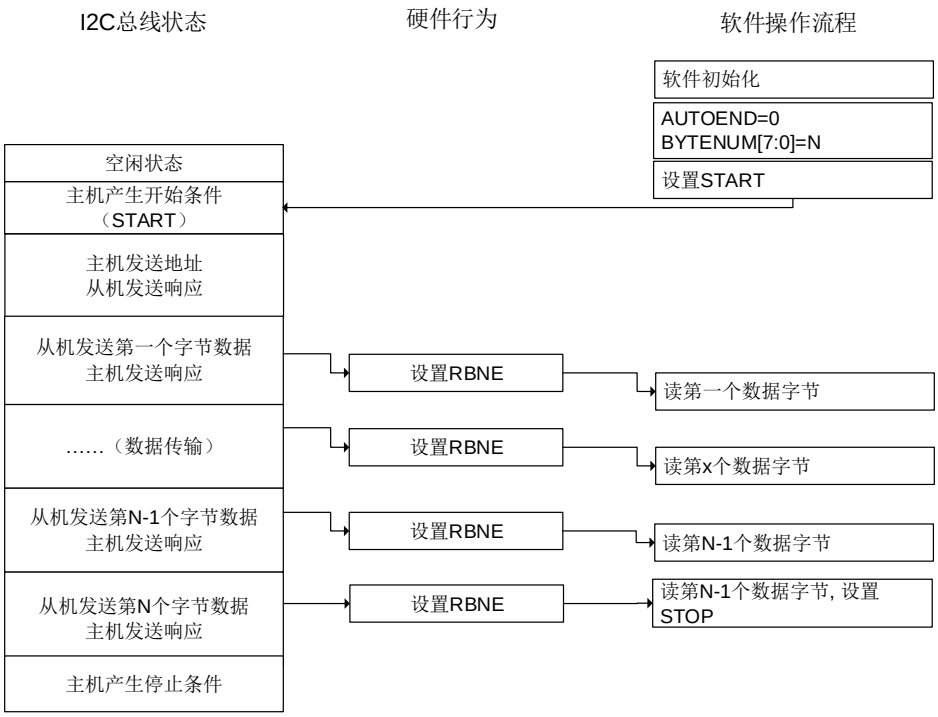


图 26-21. I2C 主机接收编程模型（N>255）



26.3.9. SMBus 支持

系统管理总线（System Management Bus，简写为 SMBus 或 SMB）是一种结构简单的单端双线制总线，可实现轻量级的通信需求。一般来说，SMBus 最常见于计算机主板，主要用于电源传输 ON / OFF 指令的通信。SMBus 是 I2C 的一种衍生总线形式，主要用于计算机主板上的低带宽设备间通信，尤其是与电源相关的芯片，例如笔记本电脑的可充电电池子系统（参见 Smart Battery Data）。

SMBus 协议

SMBus 上每个报文交互都遵从 SMBus 协议中预定义的格式。SMBus 是 I2C 规范中数据传输格式的子集。只要 I2C 设备可通过 SMBus 协议之一进行访问，便视为兼容 SMBus 规范。不符合这些协议的 I2C 设备，将无法被 SMBus 和 ACPI 规范所定义的标准方法访问。

地址解析协议

SMBus 采用了 I2C 硬件以及 I2C 的硬件寻址方式，但在 I2C 的基础上增加了二级软件处理，建立自己独特的系统。比较特别的是 SMBus 规范包含一个地址解析协议，可用于实现动态地址分配。动态识别硬件和软件使得总线设备能够支持热插拔，无需重启系统便能即插即用。总线中的设备将被自动识别并分配唯一地址。这个优点非常有利于实现即插即用的用户界面。在此协议中，系统中的 host 与设备之间有一个重要的区别，即 host 具有分配地址的功能。

SMBus 从机字节控制

SMBus 接收器从机字节控制与 I2C 一样。它允许 ACK 控制每个字节。必须能对接收到的命令或者数据进行 NACK 应答。通过将 I2C_CTL0 寄存器中 SBCTL 位置 1 来使能从机字节控制模式。

主机通知协议

通过将 I2C_CTL0 寄存器 SMBHAEN 位置 1，SMBus 可以支持主机通知协议。在该协议中，从设备作为主机，主设备作为从机，主机将应答 SMBus 主机地址。

超时特性

SMBus 有一种超时特性：假如某个通信耗时太久，便会自动复位设备。这就解释了为什么最小时钟周期为 10kHz——为了防止长时间锁死总线。I2C 在本质上可以视为一个“直流”总线，也就是说当主机正在访问从机的时候，假如从机正在执行一些子程序无法及时响应，从机可以拉住主机的时钟。这样便可以提醒主机：从机正忙，但并不想放弃当前的通信。从机的当前任务结束之后，将可以继续 I2C 通信。I2C 总线协议中并没有限制这个延时的上限，但在 SMBus 系统中，这个时间被限定为 25~35ms。按照 SMBus 协议的假定，如果某个会话耗时太久，就意味着总线出了问题，此时所有设备都应当复位以消除这种（问题）状态。这样就并不允许从设备将时钟拉低太长时间。

将 I2C_TIMEOUT 寄存器中 TOEN 位和 EXTOEN 位置 1 可以使能超时检测。配置定时器必须保证在 SMBus 规范规定的时间最大值之前检测出超时情况。

在 BUSTOA[11:0]中编程的值被用来检查$t_{TIMEOUT}$参数。必须将 TOIDLE 位配置为 0，以检测 SCL 低电平超时。将 I2C_TIMEOUT 寄存器中 TOEN 位置 1 来使能定时器，在 TOEN 置 1 之后，BUSTOA[11:0] 和 TOIDLE 位不能被修改。如果 SCL 低电平时间大于(BUSTOA+1)*2048*t_{I2CCLK}，I2C_STAT 寄存器中 TIMEOUT 位将置 1。

BUSTOA[11:0]为从机校验$t_{LOW:SEXT}$，为主机校验$t_{LOW:MEXT}$。通过将 I2C_TIMEOUT 寄存器中 EXTOEN 位置 1 来使能定时器。在 EXTOEN 置 1 之后，BUSTOB[11:0]不能被修改。如果 SMBus 外设 SCL 拉低时间大于(BUSTOB+1)*2048*t_{I2CCLK}，并且达到了总线空闲检测章节中描述的超时时间间隔，I2C_STAT 寄存器中 TIMEOUT 位将置 1。

报文错误校验

I2C 模块中有一个 PEC 模块，它使用 CRC-8 计算器来执行 I2C 数据的报文校验。一个 PEC 字节（PEC 错误码）附加在每次传输结束。PEC 的计算方式是对所有消息字节（包含地址和读/写位）使用 CRC-8 计算校验和。CRC-8 多项式位x^8+x^2+x+1（CRC-8-ATM HEC 算法，初

始化为 0）。

当 I2C 被禁用时，通过 I2C_CTL0 寄存器中的 PECEN 位置 1 可以使能 PEC。由于 PEC 传输是由 I2C_CTL1 寄存器中 BYTENUM[7:0]管理的，因此在从机模式下必须将 SBCTL 位置 1。当 PECTRANS 置 1，RELOAD 为 0 时，在 BYTENUM[7:0]-1 数据字节后发送 PEC。PEC 在 BYTENUM[7:0]-1 传输完成后发送。当 RELOAD 置 1 时 PECTRANS 无效。

SMBus 警报

SMBus 还有一个额外的共享的中断信号，称为 SMBALERT#。从机上发生事件后，可通过这个信号通知主机来访问从机。主机会处理该中断，并通过报警响应地址，同时访问所有 SMBALERT#设备。如果 SMBALERT#电平被设备拉低，这些设备会应答报警响应地址。当配置为从设备（SMBHAEN = 0）时，通过将 I2C_CTL0 寄存器中 SMBALTEN 置 1 可以将 SMBA 引脚电平拉低。同时也使能了报警响应地址。当配置为主设备（SMBHAEN = 1），且 SMBALTEN 置 1 时，当在 SMBA 引脚检测到下降沿时，I2C_STAT 寄存器中 SMBALT 位将置 1。如果 I2C_CTL0 寄存器中 ERRIE 位置 1，将产生中断。当 SMBALTEN = 0 时，即使外部 SMBA 引脚为低电平，ALERT 线也将被视为高电平。当 SMBALTEN = 0 时，SMBA 引脚可用作标准 GPIO。

总线空闲检测

如果主机检测到时钟信号和数据信号的高电平持续时间大于$t_{HIGH,MAX}$，总线被视为空闲。

该时序参数已考虑到主机已动态添加至总线，但可能还未检测到 SMBCLK 或 SMBDAT 线上的状态转换的情况。在这种情况下，为了保证当前没有数据传输正在进行，主机必须等待足够长的时间。

要使能t_{IDLE}检查，必须将 BUSTOA[11:0]编程为定时器重载值，以获取t_{IDLE}参数。必须将 TIDLE 位置 1，以检测 SCL 和 SDA 高电平超时。然后通过将 I2C_TIMEOUT 寄存器中的 TOEN 位置 1 来使能定时器。TOEN 置 1 后，BUSTOA[11:0]和 TIDLE 不能被修改。如果 SCL 和 SDA 的高电平持续时间都大于(BUSTOA+1)*4*t_{I2CCLK}，I2C_STAT 寄存器中 TIMEOUT 位将置位。

SMBus 从机模式

SMBus 接收器必须能够对接收到的命令和数据进行 NACK 应答。对于从机模式下的 ACK 控制，通过将 I2C_CTL0 寄存器中 SBCTL 位置 1 可以使能从机字节控制模式。

必要时应使能特定的 SMBus 地址。通过将 I2C_CTL0 寄存器中 SMBDAEN 置 1 可以使能 SMBus 设备默认地址（0b1100 001）。通过将 I2C_CTL0 寄存器中 SMBHAEN 置 1 可以使能 SMBus 主机地址（0b0001 000）。通过将 I2C_CTL0 寄存器中 SMBALTEN 置 1 可以使能报警响应地址（0b0001 100）。

26.3.10. 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。

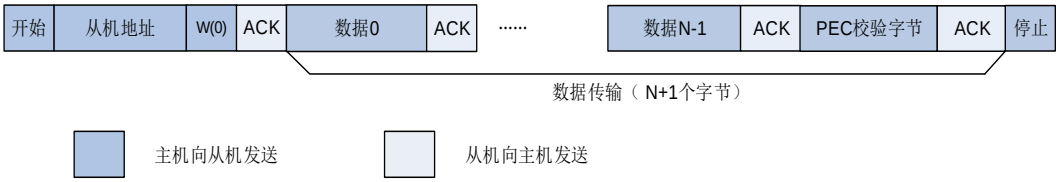
SMBus 作为从机接收器时，为了在数据发送完成时进行 PEC 校验，SBCTL 位必须置 1。要对每个字节进行 ACK 控制，必须通过将 RELOAD 位置 1 来使能 RELOAD 模式。如果要校验 PEC 字节，必须将 RELOAD 位清零同时将 PECTRANS 置 1。在 BYTENUM-1 个字节接收完成后，接收的下一个字节将与 I2C_PEC 寄存器中的数据进行比较。如果校验值不匹配，将自动产生 NACK 信号；如果校验值匹配将自动产生 ACK 信号，将忽略 NACKEN 位的值。当接收到 PEC 字节时，PEC 字节会存到 I2C_RDATA 寄存器中，RBNE 位将置 1。如果 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 26-22. 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

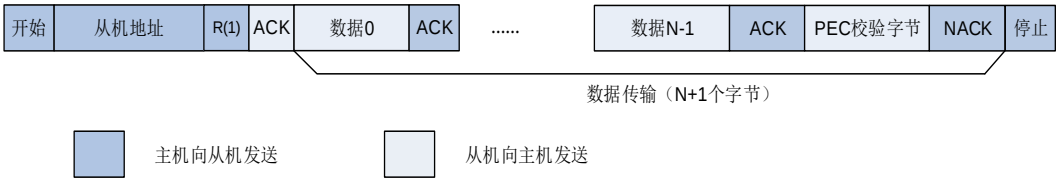
如果 SMBus 主机需要在数据传输完成后接收 PEC 字节，可以使能自动结束模式。在 START 信号发送之前，必须将 PECTRANS 位置 1，且配置好从机地址。在接收 BYTENUM-1 数据之后，接收的下一个字节将自动与 I2C_PEC 寄存器中的数据进行比较。在停止信号发送之前，接收 PEC 字节之后会给出 NACK 响应。

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 26-23. SMBus 主机接收器和从机发送器通信流程



26.3.11. 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 IRC64M。在深度睡眠模式下，IRC64M 关闭。当 I2C 检测到 START 信号时，IRC64M 打开，I2C 会将 SCL 拉低直到 IRC64M 被唤醒。在接收地址期间，IRC64M 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，IRC64M 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，IRC64M 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）。

26.3.12. 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。

26.3.13. I2C 错误和中断

I2C 错误标志如*表 26-4. I2C 错误标志*所示。

表 26-4. 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 26-5. I2C 中断事件*所示。

表 26-5. 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	

26.3.14. I2C 调试模式

当为控制器进入调试模式（Cortex®-M7 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。

26.4. 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

26.4.1. 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。

26.4.2. 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，该位域不能被修改，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。

26.4.3. 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。

26.4.4. 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。

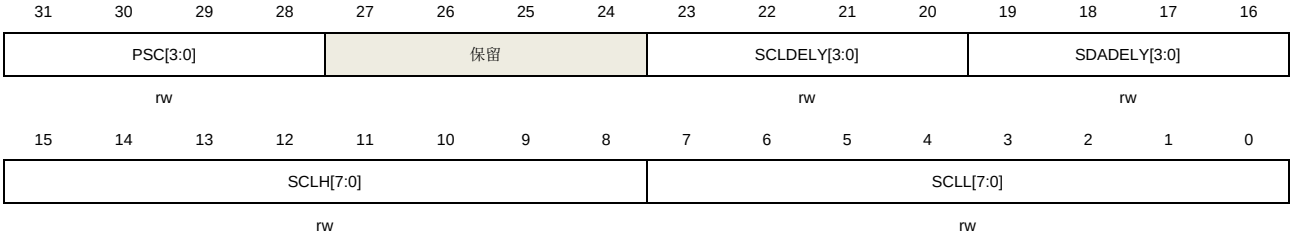
26.4.5. 时序寄存器（I2C_TIMING）

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。



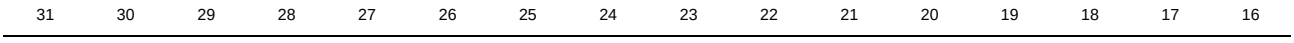
位/位域	名称	描述
31:28	PSC[3:0]	时序预分频 为了生成用于数据建立和数据保持的计数器的时钟周期t_{PSC}，这些位用于配置 I2CCLK 时钟预分频。t_{PSC}也用于 SCL 高电平和低电平计数器。 $t_{PSC}=(PSC+1)*t_{I2CCLK}$。
27:24	保留	必须保持复位值。
23:20	SCLDELY[3:0]	数据建立时间 这些位用于在 SDA 边沿和 SCL 上升沿之间生成延时$t_{SCLDELY}$。在主机模式下和在从机模式下 SS=0 时，在$t_{SCLDELY}$期间 SCL 线被拉低。 $t_{SCLDELY}=(SCLDELY+1)*t_{PSC}$。
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。

26.4.6. 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。



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。

26.4.7. 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。

26.4.8. 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			

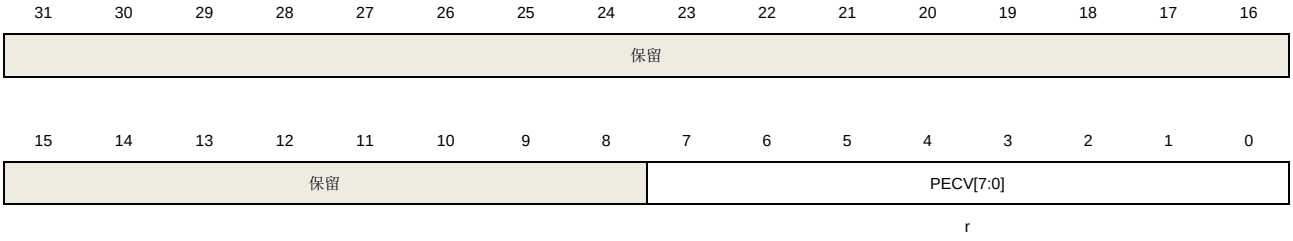
位/位域	名称	描述
31:14	保留	必须保持复位值。
13	SMBALTC	SMBus 报警标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 SMBALT 位。
12	TIMEOUTC	TIMEOUT 标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 TIMEOUT 位。
11	PECERRC	PEC 错误标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 PECERR 位。
10	OUERRC	上溢 / 下溢标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 OUERR 位。
9	LOSTARBC	仲裁丢失标志清零 软件对该位写 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。

26.4.9. 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。

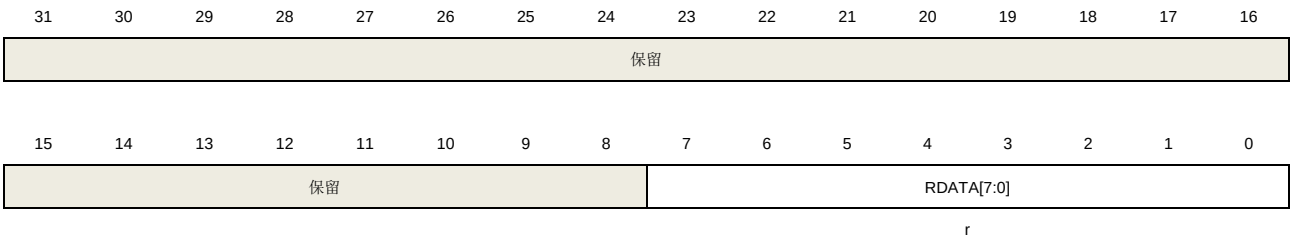


位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	PECV[7:0]	在 PEC 使能时，由硬件计算出来的 PEC 值。 当 I2CEN = 0 时，PECV 由硬件清零。

26.4.10. 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

26.4.11. 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。

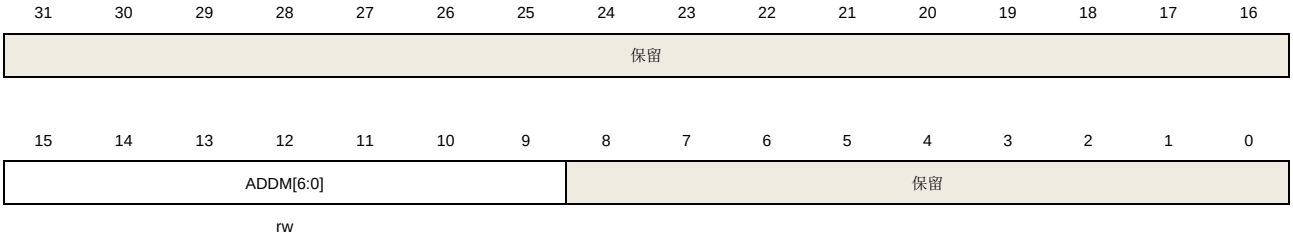


位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	TDATA[7:0]	发送的数据

26.4.12. 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。



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。

27. 串行外设接口/片上音频接口（SPI/I2S）

27.1. 简介

SPI/I2S模块可以通过SPI协议或I2S音频协议与外部设备进行通信。

串行外设接口（Serial Peripheral Interface，缩写为SPI）提供了基于SPI协议的数据发送和接收功能，可以工作于主机或从机模式。SPI接口支持具有硬件CRC计算和校验的全双工、半双工和单工模式。SPI3 / 4还支持SPI四线主机模式。

片上音频接口（Inter-IC Sound，缩写为I2S）支持四种音频标准，分别是I2S飞利浦标准，MSB对齐标准，LSB对齐标准和PCM标准。它可以在四种模式下运行，包括主机发送模式，主机接收模式，从机发送模式和从机接收模式。

27.2. 主要特性

27.2.1. SPI 主要特性

- 具有全双工、半双工和单工模式的主从操作；
- 32位宽度，独立的发送和接收FIFO；
- 4位到32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，MOSI与MISO引脚复用功能的交换；
- 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；
- 多主机多从机功能；
- 配置和设置保护；
- 可调的数据帧之间的最小延时和NSS与数据流之间的最小延时；
- 主机模式错误可触发中断，上溢、下溢和CRC错误检测；
- 可调的主设备接收器采样时间；
- 可配置的FIFO阈值（数据打包）；
- 在从机模式，下溢条件可配置；
- 支持SPI四线功能的主机模式（只有SPI3 / 4）。

27.2.2. I2S 主要特性

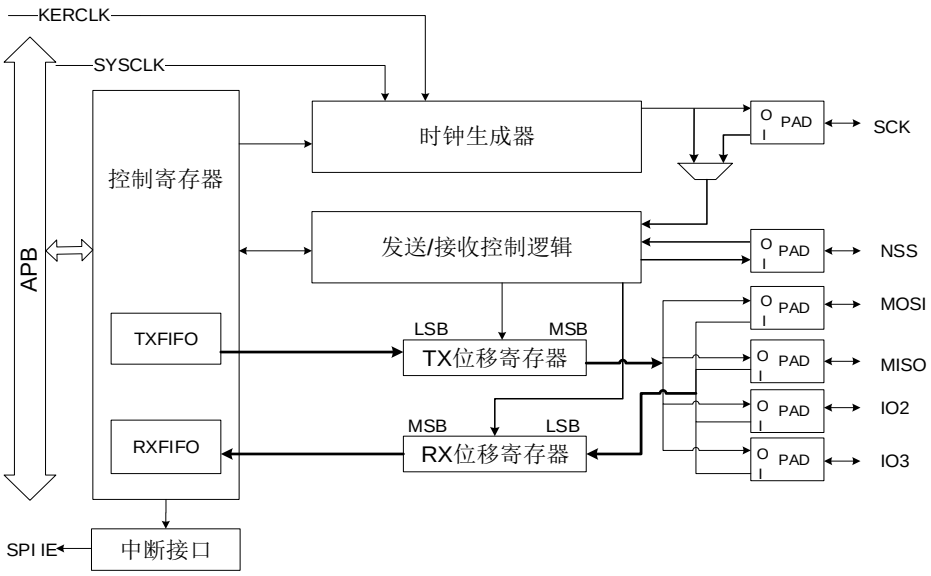
- 具有发送和接收功能的主从操作；
- 支持四种I2S音频标准：飞利浦标准，MSB对齐标准，LSB对齐标准和PCM标准；
- 数据长度可以为16位，24位和32位；
- 通道长度为16位或32位；
- 提高可靠性的错误信号：下溢、上溢和帧格式错误；
- 32位宽的发送和接收缓冲区；

- 通过I2S时钟分频器，可以得到8 kHz到192 kHz的音频采样频率；
- 可编程空闲状态时钟极性；
- 可以输出主时钟（MCK）；
- 发送和接收支持DMA功能；
- 32位宽度，独立的发送和接收FIFO。

27.3. SPI 功能说明

27.3.1. SPI 结构框图

图 27-1. SPI 结构框图



- SYSCLK：系统时钟，由 APB 总线提供。需要访问 SPI 寄存器时，该时钟必须有效；
- KERCLK：内核时钟，由 RCU 提供，和系统时钟是异步的关系；
- 时钟信号的频率没有特定限制，但需与用户使用条件及数据传输速度匹配，防止数据丢失；（**注**：建议 SYSCLK 大于等于 KERCLK 的频率）
- SPI 从机的 SCK 信号由 SPI 主机提供。

27.3.2. SPI 信号线描述

常规配置（非 SPI 四线模式）

表 27-1. 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默认配置为单线模式，当SPI_QCTL中的QMOD位置1时，配置为SPI四线模式（只适用于SPI3 / 4）。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 27-2. 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

串口数据线交换配置

SPI可以通过设置SPI_CFG1寄存器的SWPMIO位去交换MOSI与MISO的功能。

表 27-3. MISO / MISO 信号交换描述

MODE	SWPMIO	MOSI	MISO
主发送	0	发送	-
	1	-	发送
从发送	0	-	发送
	1	发送	-
主接收	0	-	接收
	1	接收	-
从接收	0	接收	-
	1	-	接收
主全双工	0	发送	接收
	1	接收	发送
从全双工	0	接收	发送
	1	发送	接收

27.3.3. SPI 时序和数据帧格式

SPI_CFG1寄存器中的CKPL位和CKPH位决定了SPI时钟和数据信号的时序。CKPL位决定了空闲状态时SCK的电平，CKPH位决定了第一个或第二个时钟跳变沿为有效采样边沿。在TI模式下，这两位没有意义。

图 27-2. SPI 常规模式下的时序图

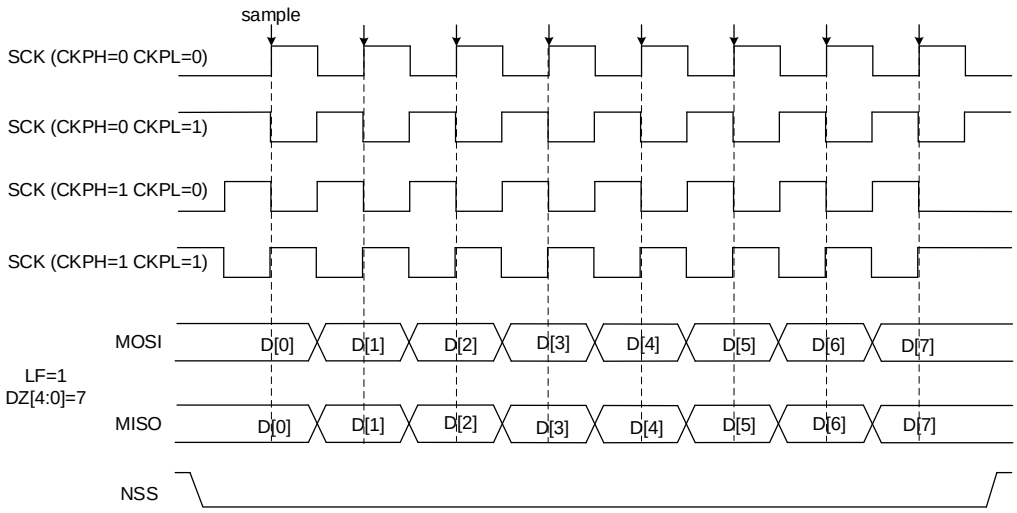
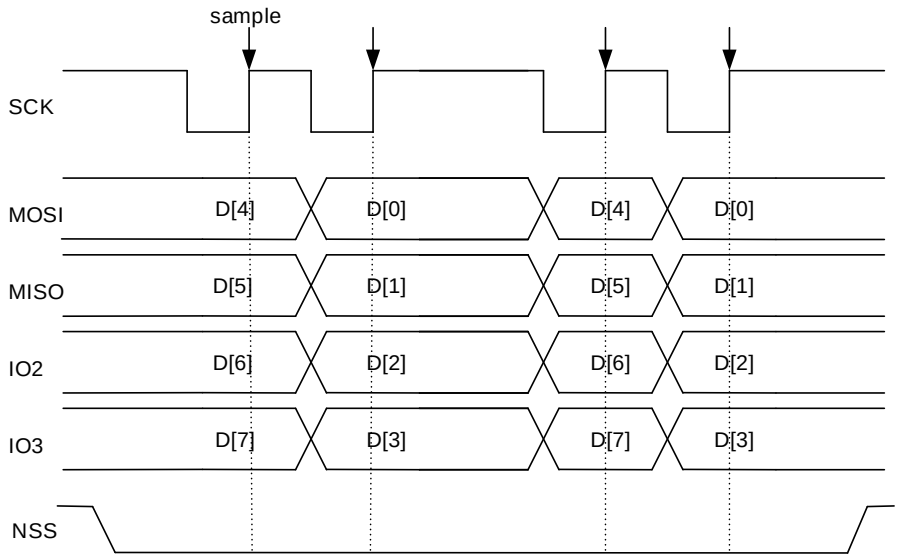


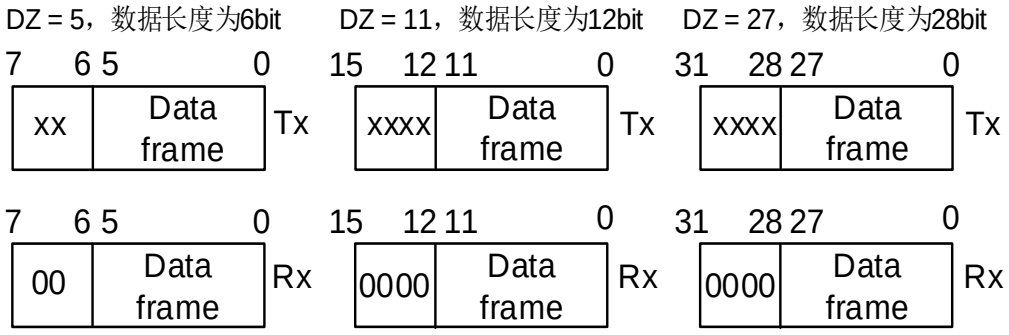
图 27-3. SPI3 / 4 四线模式下的 SPI 时序图（CKPL = 1, CKPH = 1, LF = 0）



在常规模式中，通过SPI_CFG0中的DZ[4:0]位域配置数据长度，可以设置为4位至32位。该设置不仅适用于数据的发送也适用于数据的接收。通过设置SPI_CFG1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。在SPI四线模式下，数据长度固定为8位。

当访问SPI_TDATA / SPI_RDATA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）或一个半字或一个字。通讯时，只有数据长度内的位会随时钟输出。

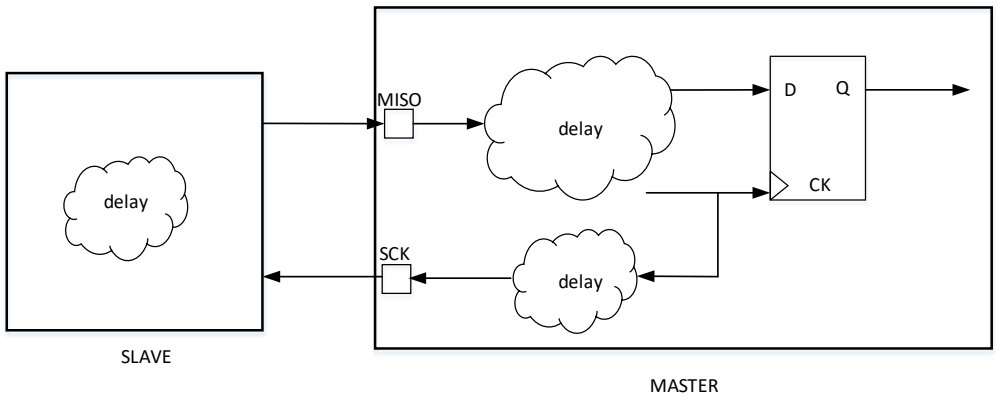
图 27-4. SPI 数据帧右对齐示意图



27.3.4. SPI 时钟延迟模式

SPI可以被配置为主机或者从机。当SPI被配置为主机模式，时钟SCK从SPI主机发出，经过延迟到达从机，从机驱动MISO数据发送，MISO数据从从机再次经过延迟到达主机采样端，这一系列延迟会导致SPI主机接收的数据和时钟有相位差，从而导致数据采样错误，这一点在较高速度的比特率下会更为明显。

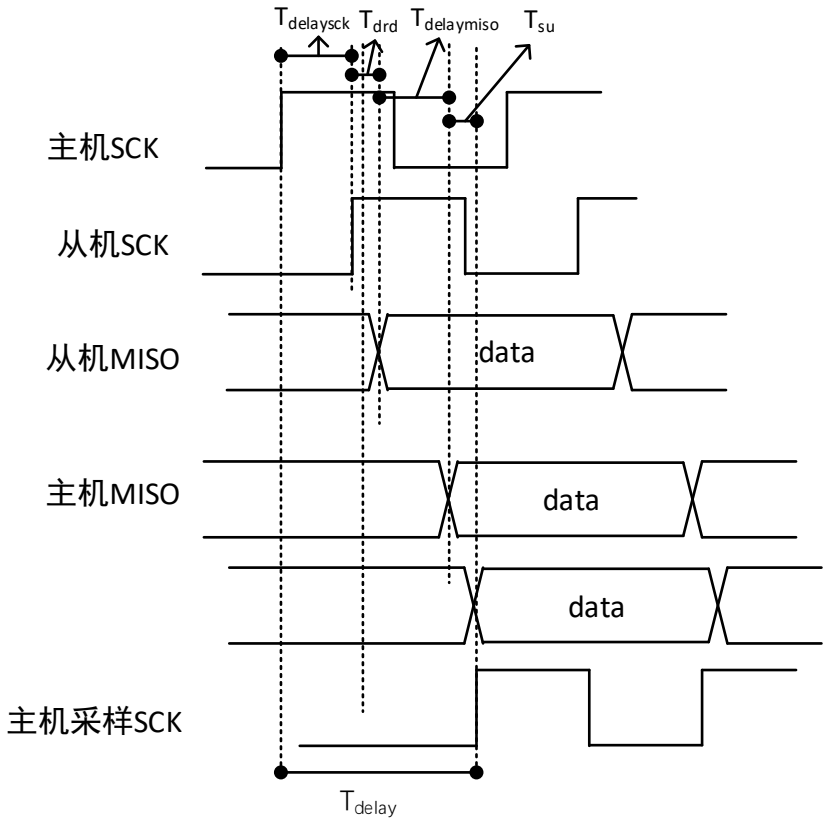
图 27-5. SPI 数据时钟传输路径示意图



为了解决这一问题，可以通过配置SPI_RXDLYCK调节SPI内部主机的接收时钟相位使得满足正确的采样时序（此配置是非必须的，需要结合实际的场景使用）。

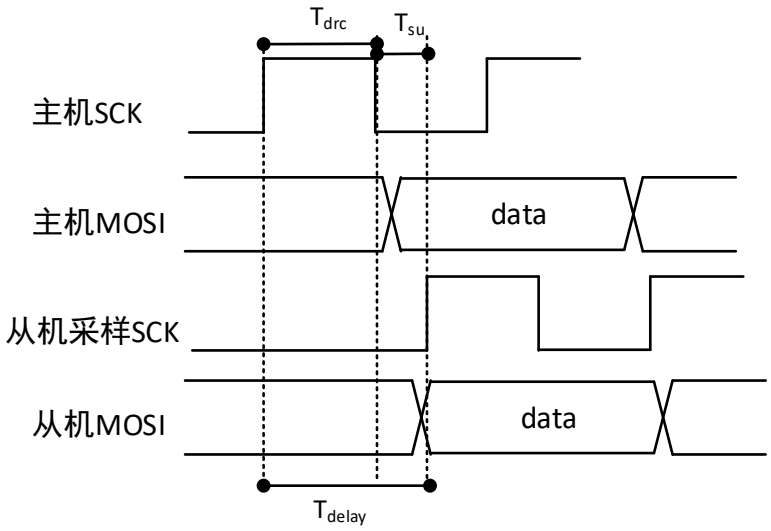
SPI_RXDLYCK的MRXDEN为1’b0开启延迟功能，配置1’b1关闭延迟功能。MRXD配置5’b00000~5’b11111可将延迟长度配置为1~32个延迟单元（常温下一个延迟单元的延迟是0.5ns）。用户需要根据自身场景配置延迟T_{delay}。（$T_{delay} > T_{delaysck} + T_{drd} + T_{delaymiso} + T_{su}$）

图 27-6. SPI 主机接收延迟配置时序图



当SPI被配置为从机模式同理。从机时钟延迟的原因有两个，一方面是MOSI和SCK可能会有相位差T_{drc}，从而优化从机采样的误差，另一方面可以通过调节延迟而调整MISO的相位，从而给主机的采样提供时序上的优化。SPI_RXDLYCK的SRXDEN为1’b0开启延迟功能，配置1’b1关闭延迟功能。SRXD配置5’b00000~5’b11111可将延迟长度配置为1~32个延迟单元（常温下一个延迟单元的延迟是0.5ns）。用户需要根据自身场景配置延迟T_{delay}。（$T_{delay} > T_{drc} + T_{su}$）

图 27-7. SPI 从机接收延迟配置时序图



27.3.5. RxFIFO 和 TxFIFO

RxFIFO和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，并且可以防止当数据帧长度较短或中断 / DMA延迟太长时发生的上溢。

对SPI_TDATA寄存器的写访问会将写入的新数据存储在TxFIFO的末尾，而对SPI_RDATA的读访问则返回RxFIFO中最早的数据。FIFO处理取决于数据交换模式（双工和单工）、数据帧格式（DZ值）、访问FIFO寄存器的大小（8、16或32位）以及数据包中数据的组织方式。TxFIFO / RxFIFO的范围为16x32位，最大访问数据帧长度为32位，*表27-4. SPIX FIFO最大存储数据帧数量*描述了在不同帧尺寸时，FIFO中可存放的最大帧数量。（N = FIFO范围 / 32= 16 x 32 / 32 = 16）

表 27-4. SPIX FIFO 最大存储数据帧数量

数据帧尺寸（DZ）	DZ <= 8 位	8 位 < DZ <= 16 位	16 位 < DZ <= 24 位	DZ > 24 位
FIFO 存储帧数（BYTEN = 1，WORDEN = 0）	N	-	-	-
FIFO 存储帧数（BYTEN = 0，WORDEN = 0）	2N	N	-	-
FIFO 存储帧数（WORDEN = 1）	4N	2N	N	N

注意：当SPI设备被禁止时（SPIEN = 0），RxFIFO和TxFIFO中的数据将被清空。

RxFIFO 接收

对SPI_RDATA的读访问由RP事件管理。当RxFIFO非空（至少一个完整的数据包在RxFIFO中）时，该事件被触发。当RP被清除时，RxFIFO被认为是空的（或者RxFIFO中的数据包是不完整的）。RP在RPIE位置1时触发中断，或者在DMAREN位置1触发DMA请求。

TxFIFO 发送

对SPI_TDATA的写访问由TP事件管理。当TxFIFO有足够的可用空间接收数据包时触发此事件。如果TxFIFO由软件或DMA填充，TP标志被清除。当TXF设置为1或SPI禁用时，如果没有足够的空间存储至少一个数据包，那么对TxFIFO的写入将被忽略。TP在TPIE位置1时触发中断，或在DMATEN位置1时触发DMA请求。当TXF标志设置为1时，TPIE屏蔽被硬件清除。

双工数据包处理

在全双工模式下，DP位可以监控TP和RP事件。将DP标志设置为1时，应用程序将适当数量的数据写入SPI_TDATA寄存器以上传一个完整的包，然后从SPI_RDATA寄存器读取等量的数据以下载一个完整的包。在一个包被上传和下载后，应用程序检查DP值，看看它是否可以推送和弹出其他包，如果可能，一个包一个包地上传 / 下载它们，直到DP读取0。DP在DPIE位置1触发中断，或当DMATEN和DMAREN置1时触发DMA请求。当TXTF标志设置为1时，DPIE屏蔽被硬件清除。

如果在RxFIFO满时接收下一个数据，则会发生接收上溢事件。上溢事件可以由中断或轮询处理。这种情况可能发生在从模式或主模式（全双工或只读模式，MASP = 0）。主设备处于只读模式，当MASP = 1时，如果RxFIFO已满，生成的时钟将自动停止，以防止上溢事件。

数据打包

当数据帧尺寸（DZ）<= 8位时，在对SPI_RDATA或SPI_TDATA中进行16位或32位的读写访问时（BYTEN = 0或WORDEN = 1），将自动开启数据打包模式。在这种情况下，多个数据帧并行处理。在发送端，如果FIFOLVL = 1（数据包中有2个数据帧）或FIFOLVL = 3（数据包中有4个数据帧），则在单个16位或32位访问发送端SPI_TDATA寄存器后发送2或4个数据帧。在接收端，如果FIFOLVL = 1（数据包中有2个数据帧）或FIFOLVL = 3（数据包中有4个数据帧），则在单个16位或32位访问接收端SPI_RDATA寄存器时，同时接收2个或4个数据帧，在接收端只能产生1个RP事件。然后接收端必须从SPI_RDATA中以16位或32位读取所有数据帧。如果FIFOLVL = 0（数据包中有1个数据帧），接收端从SPI_RDATA读取16位或32位的数据帧时会产生2个或4个RP事件。

如果9位 <= DZ <= 16位，在对SPI_RDATA或SPI_TDATA中进行32位的读写访问时（WORDEN = 1），将自动开启数据打包模式。将使用最低有效半字节去存储有效数据。在发送端，如果FIFOLVL = 1（数据包中有2个数据帧），则在单个32位访问发送端SPI_TDATA寄存器后发送2个数据帧。在接收端，如果FIFOLVL = 1（数据包中有2个数据帧），则在单个32位访问接收端SPI_RDATA寄存器时，同时接收2个数据帧，在接收端只能产生1个RP事件。然后接收端必须从SPI_RDATA中以32位读取所有数据帧。如果FIFOLVL = 0（数据包中有1个数据帧），接收端从SPI_RDATA读取32位的数据帧时会产生2个RP事件。

当短数据帧（< 8或< 16位）与大数据访问模式（16或32位）配对时，FIFOLVL值必须配置为数据帧数量的倍数，如果32位访问用于8位以下的帧，则为4的倍数。如果16位访问用于8位以下的帧，用2的倍数。如果32位访问用于16位的帧，用2的倍数。

FIFOLVL设置必须始终高于后续的读访问大小，否则将读取额外的伪数据。不允许小于配置数据大小的FIFO数据访问（数据帧大小由DZ设置，FIFO数据访问由BYTEN / WORDEN设置）。始终确保至少有一个完整的数据帧被访问。

顺序传输处理

用户可以根据TXSIZE和TXSER值处理消息中的多个数据。当通过设置MSTART位启用SPI时，消息的传输事务开始，在所需的数据数量已被传输时结束。如果当MSTART设置为1时TXSIZE保持为零，则无限传输事务开启。通过设置MSPDR位（清除MSTART位），事务可以在任何时候挂起。

在主模式下，TXSIZE中的数据量传输完毕后，如果TXSER的值不为零，则将TXSER的值复制到TXSIZE中，并自动清除TXSER的值。然后，传输将增加与TXSIZE中新加载值对应的数据数量。在重新加载操作之后，如果TXSERFIE被设置为1，则TXSERF标志被设置为1，并将触发中断。用户可以在下次重新加载之前将下一个非零值写入TXSER，这样它就可以处理多个数据。在这种情况下，ET事件不会发生，因为传播仍在继续。

如果TXSIZE或TXSER定义的数据量（数据帧数）不能与FIFOLVL中定义数据包长度对齐，那么在发送结束前的最后一个不完整的数据包需做打包处理。*数据打包*详细描述了打包原理。

注意：为防止传输下溢，可将从机SPI_URDATA寄存器中写入特定值。在从机TxFIFO变为空时，该值将作为下一个数据自动送出，并且在主机接收后通过软件进行解析，以便通过软件挂起主机接收器。

传输延时处理

如果从机的接收速度小于主机的传输速度，主机必须降低传输速度，通过降低时钟频率或增加数据帧之间的时延。主控模式下，SPI_CFG1寄存器中的MFDF[3:0]位用于增加数据帧之间的延迟，主控模式下，MSSD[3:0]用于增加NSS有效沿与开始传输或接收数据之间的延迟。详细描述可参见*NSS信号时序*。

27.3.6. NSS 功能

从机模式

当配置为从机模式（MSTMOD = 0）时，在硬件NSS模式（NSSIM = 0）下，SPI从NSS引脚获取NSS电平，在软件NSS（NSSIM = 1）下，SPI根据NSSI位得到NSS电平。只有当NSS为有效电平时，发送或接收数据。在软件NSS模式下，不使用NSS引脚。用户可以设置NSSIOPL位来决定输入/输出外部信号的有效电平（在NSS引脚上）。

表 27-5. 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 NSSIM = 0	SPI 从机 NSS 电平从 NSS 引脚获取。
从机软件 NSS 模式	MSTMOD = 0 NSSIM = 1	SPI 从机 NSS 电平由 NSSI 位决定。 NSSI = 0：NSS 电平为低 NSSI = 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（NSSIM = 0，NSSDRV = 0）或者软件模式（NSSIM = 1）。一旦NSS引脚（在硬件NSS模式下）或NSSIM位（在软件NSS模式下）变无效，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（NSSIM = 0，NSSDRV = 1）。使能SPI并置位MSTART之后，NSS保持有效电平，直至传输了当前要传输的数据量后（ET标志位置位），NSS变为无效电平。

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。

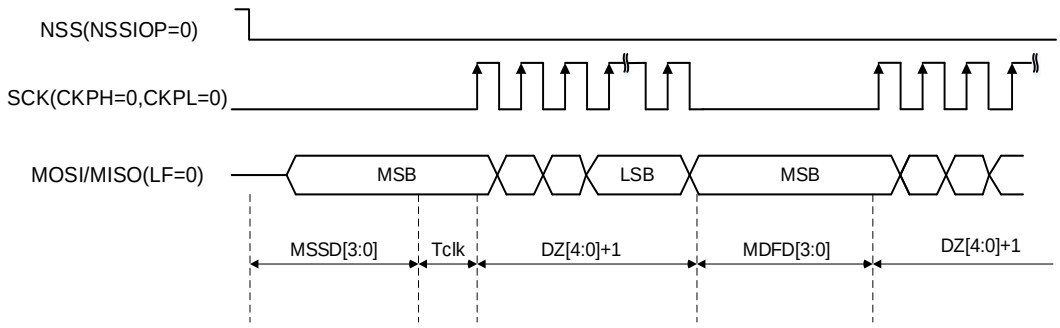
表 27-6. 主机模式 NSS 功能

模式	寄存器配置	描述
主机硬件 NSS 输出模式	MSTMOD = 1 NSSIM = 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 并置位 MSTART 后 NSS 为有效电平。
主机硬件 NSS 输入模式	MSTMOD = 1 NSSIM = 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 引脚被拉无效，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 NSSIM = 1 NSSI = 0 NSSDRV：不要求	适用于多主机模式，一旦 NSS 无效，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 NSSIM = 1 NSSI = 1 NSSDRV：不要求	从机可以使用硬件或软件 NSS 模式

NSS 信号时序

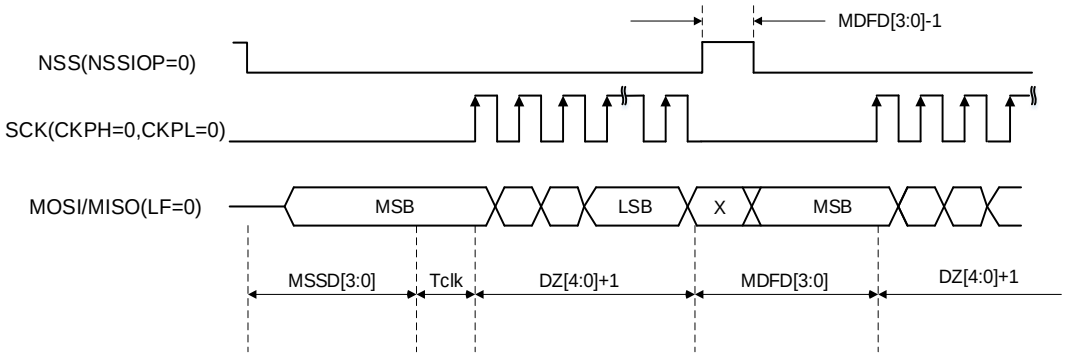
当应用硬件输出 NSS 控制（NSSIM = 0, NSSDRV = 1）时，用户可以配置 MDFD[3:0]和MSSD[3:0]位域来控制数据帧之间的 NSS 信号时序，并在每次事务开始时插入额外的延迟（以分离 NSS 和时钟启动）。*图 27-8. NSS 信号延时时序图（MSSD[3:0] = 0011（3 x T_{clk}）, MDFD = 0011（3 x T_{clk}））*描述了 MSSD[3:0] = 3 设置下，数据采集相对于 NSS 信号有效的时延和 MDFD[3:0] = 3 设置下，数据帧之间采集的时延。

图 27-8. NSS 信号延时时序图（MSSD[3:0] = 0011（3 x T_{clk}）, MDFD = 0011（3 x T_{clk}））



当NSSCTL = 1且MDFD[3:0] > 1时，SPI数据帧之间可插入交错脉冲。*图27-9. NSS交错脉冲时序图（MSSD[3:0] = 0011（3 x T_{clk}）, MDFD = 0011（3 x T_{clk}））*描述了MDFD[3:0] > 1时，NSS信号的脉冲状态。

图 27-9. NSS 交错脉冲时序图（MSSD[3:0] = 0011（3 x T_{clk}）, MDFD = 0011（3 x T_{clk}））



27.3.7. SPI 运行模式

表 27-7. SPI 运行模式

模式	描述	寄存器配置	数据引脚用法
MFD	全双工主机模式	MSTMOD = 1 RO = 0 BDEN = 0 BDOEN：不要求	MOSI：发送 MISO：接收
MTU	单向线连接主机发送模式	MSTMOD = 1 RO = 0 BDEN = 0 BDOEN：不要求	MOSI：发送 MISO：不使用
MRU	单向线连接主机接收模式	MSTMOD = 1 RO = 1 BDEN = 0 BDOEN：不要求	MOSI：不使用 MISO：接收
MTB	双向线连接主机发送模式	MSTMOD = 1 RO = 0 BDEN = 1 BDOEN = 1	MOSI：发送 MISO：不使用
MRB	双向线连接主机接收模式	MSTMOD = 1 RO = 0 BDEN = 1 BDOEN = 0	MOSI：接收 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	MOSI：不使用 MISO：发送

模式	描述	寄存器配置	数据引脚用法
		BDOEN：不要求	
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 27-10. 典型的全双工模式连接

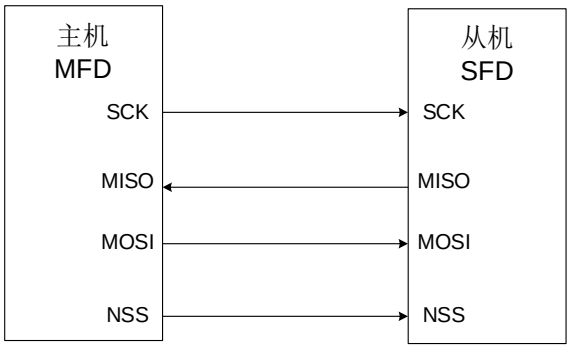


图 27-11. 典型的单工模式连接（主机：接收，从机：发送）

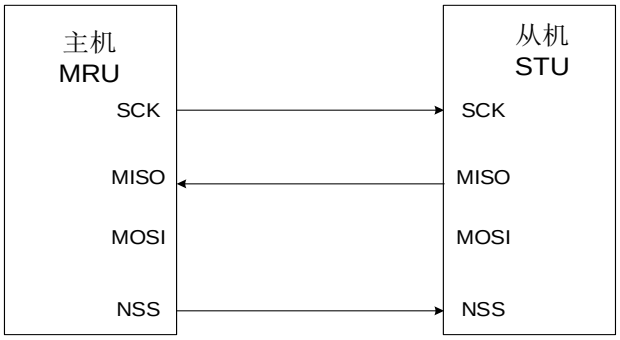


图 27-12. 典型的单工模式连接（主机：只发送，从机：接收）

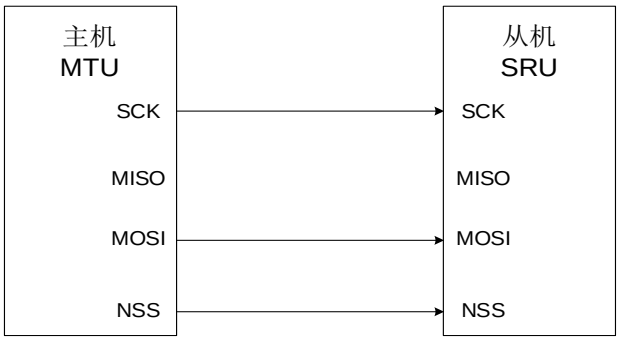
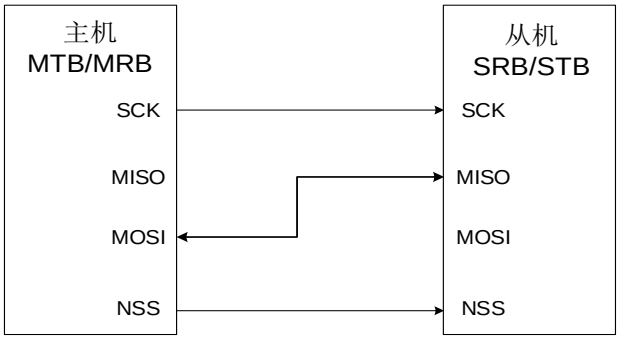


图 27-13. 典型的双向线连接



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_CFG0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_CFG1中的CKPL位和CKPH位）。
3. 配置帧格式（SPI_CFG1中的LF位）。
4. 配置数据格式（SPI_CFG0中的DZ[4:0]位域）。
5. 配置FIFO等级（SPI_CFG0中的FIFOLVL[3:0]），以及访问FIFO方式（WORDEN和BYTEN）。
6. 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_CFG1中的NSSIM / NSSDRV / NSSIOPL / NSSCTL / MDFD[3:0] / MSSD[3:0]位和SPI_CTL0中的NSSI位）。
7. 如果是从机模式，配置SPI_CFG0的TXURDT[1:0]与TXUROP[1:0]位域。
8. 如果工作在TI模式，需要将SPI_CFG1中的TMOD位置1，否则，忽略此步骤。
9. 根据***表27-7. SPI运行模式***，配置MSTMOD位、RO位、BDEN位和BDOEN位。
10. 配置SPI_CTL1寄存器以选择传输的长度，如果该值是未知的，则必须将TXSIZE设为零。
11. 配置SPI_CRCPOLY寄存器，并根据CRC多项式和CRC计算所需配置CRCSZ[4:0]位域和CRCFS位，相关描述在***CRC功能***章节。
12. 据***DMA功能***，当使用DMA时，需要初始化DMATEN / DMAREN位。
13. 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。（只有SPI3 / 4）
14. 如果需要配置保护，配置SPI_CTL0寄存器的IOAFEN位。
15. 使能SPI（将SPIEN位置1）。

16. 如果是主机模式（MSTMOD = 1），当SPIEN = 1，配置SPI_CTL0中MSTART位去传输数据。如果无需传输数据，忽略此步骤。

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TxFIFO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引脚电平有效，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TxFIFO中。

当SPI开始发送一个数据帧时，首先将这个数据帧从TxFIFO加载到移位寄存器中，然后开始发送加载的数据。相关操作可参考***RxFIFO和TxFIFO***描述。

对SPI_TDATA的写访问由TP事件管理。当TP标志设置为1时，应用程序对SPI数据寄存器写入适当数量的数据，以传输数据包的内容。在上传新的完整包后，应用程序检查TP值，检查TxFIFO是否可以接收额外的数据包，如果TP = 1，则逐包上传，直到TP读取0。如果传输大小和数据包大小没有对齐，则最后要传输的数据包数无法达到配置的大小（由FIFOLVL设置）。应用程序仍然可以将标准数量的先前完整数据包写入TxFIFO，而不会产生不良影响：只有一致的数据（完整的数据帧）将传输到TxFIFO，而冗余的写入时间（或任何不完整的数据）将被忽略。

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_TDATA寄存器中。只要TxFIFO中存在数据，数据发送便一直继续，直至TxFIFO变为空。

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RxFIFO，且RP（RxFIFO非空）位置1。软件通过读SPI_RDATA寄存器获得接收的数据，此操作会自动清除RP标志位（当RxFIFO数据量少于FIFOLVL标准）。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TxFIFO非空时，硬件才接收下一个数据帧。相关操作可参考***RxFIFO和TxFIFO***描述。

对SPI_RDATA的读访问由RP事件管理。当RP标志设置为1时，应用程序读取SPI数据寄存器相当数量的数据，以下载单个数据包内容。下载完整数据包后，应用程序会检查RP值，查看RxFIFO中是否有其他数据包，如果有，则逐包下载，直到RP读到0。在接收结束时，可能会出现RxFIFO中仍然有一些数据可用，但没有达到FIFOLVL级别，因此RP不会被设置为1。在这种情况下，RxFIFO中剩余的RX数据帧的数量将由SPI_STAT寄存器中的RWNE和RPLVL表示。如果传输大小和数据包大小没有对齐，当最后接收的数据包数量不能达到配置的大小(由FIFOLVL设置)时，就会出现上述情况。然而，应用程序仍然可以从RxFIFO读取标准数量的以前完整的数据包，而不会产生不良影响：只有一致的数据（完整的数据帧）将从RxFIFO读取，而冗余的读取（或任何不完整的数据）将读取0。

接收数据时，主机提供时钟信号，当主机停止或挂起SPI时才会停止接收流程。主机通过将MSTART位置1来启动流程，可通过向SPI_CTL0寄存器的MSPDR为写1来请求挂起，或者向MASP位写1来设置上溢挂起。当完成TXSIZE和TXSER中的数据帧传输后，接收流程也会结束。

SPI 不同模式下的操作流程（非 SPI 四线模式，TI 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RP标志位和TP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RP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TP标志位，并且在RP位置1后，读出RxFIFO内的数据，否则，将会产生接收过载错误。

除了忽略TP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， CKPL、CKPH、LF、NSSIM、NSSIOPL、NSSDRV位是没有意义的，SCK信号的采样边沿为下降沿。

图 27-14. 主机 TI 模式在不连续发送时的时序图

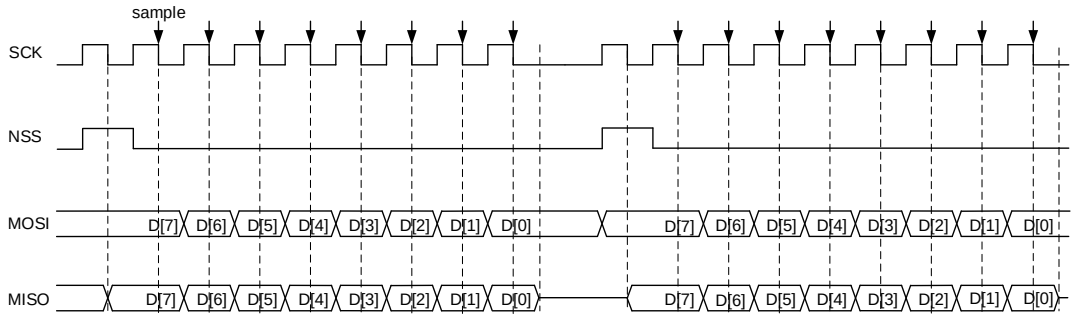
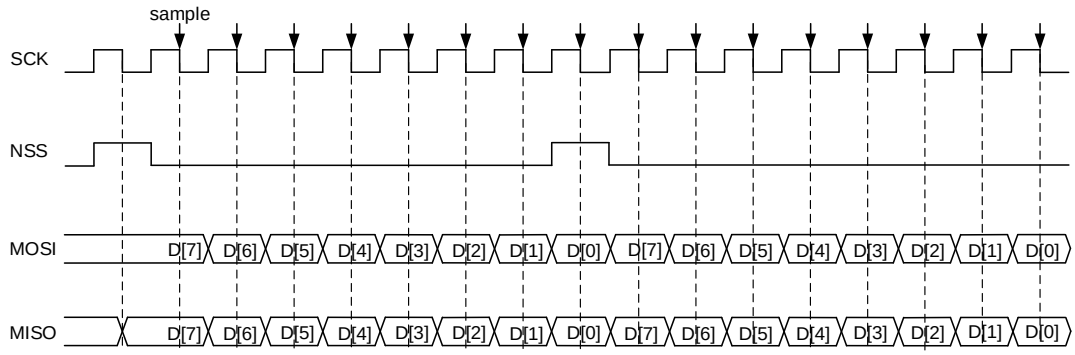
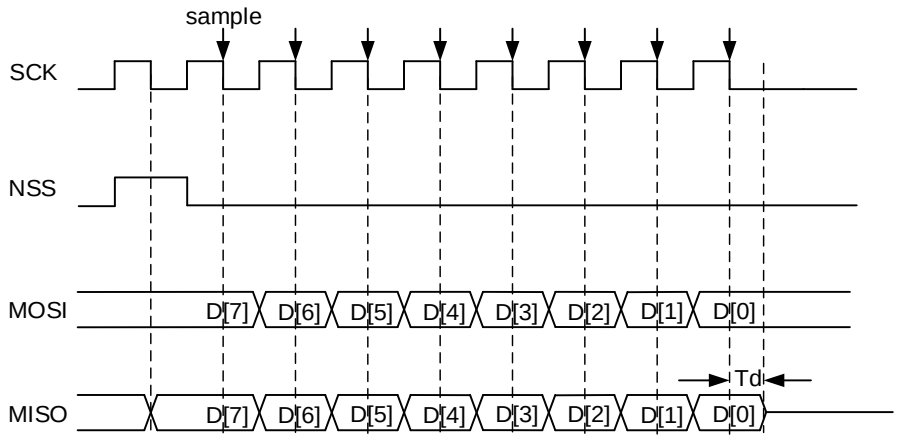


图 27-15. 主机 TI 模式在连续发送时的时序图



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_TDATA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 27-16. 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MISO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_CFG0寄存器中的PSC[2:0]位来设置。

$$\frac{T_{bit}}{2} + 2 * T_{kerclk} \leq T_d \leq \frac{T_{bit}}{2} + 4 * T_{kerclk} \quad (27\text{-}1)$$

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。

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TP位与TC位置1，然后将SPI_QCTL寄存器中的QMOD位置1。在SPI四线模式，BDEN位、BDOEN位、CRCEN位、CRCSZ位、RO位和LF位保持清零，WORDEN配置为1，DZ[4:0]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、MSTART位、TXSIZE、TXSER位、PSC位、CKPL位和CKPH位根据需要进行配置。

注意：四线模式不支持CRC功能。PSC不能配置为两分频和四分频。

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。

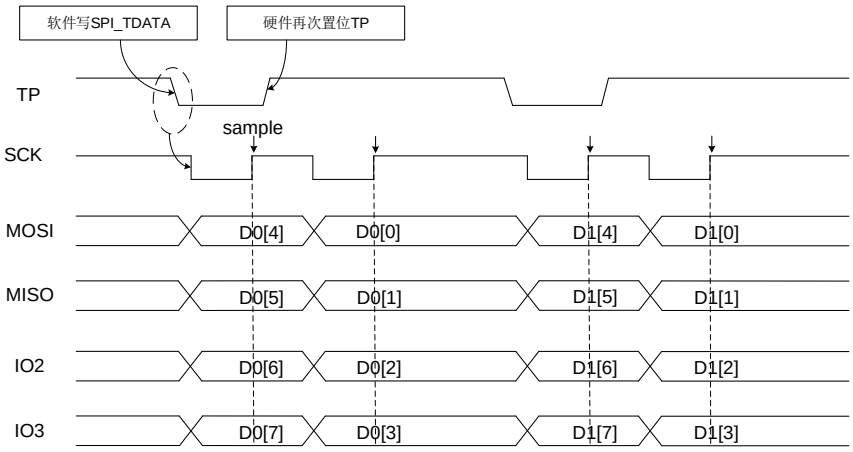
四线写模式

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_TDATA寄存器（TP位清零）且SPIEN和MSTART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TP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、SPI_CTL1、SPI_CFG0、SPI_CFG1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_TDATA寄存器中写入一个字节的数据，TP标志位将会清零。
4. 等待硬件将TP位重新置位，然后写入下一个字节数据。

图 27-17. SPI 四线模式四线写操作时序图



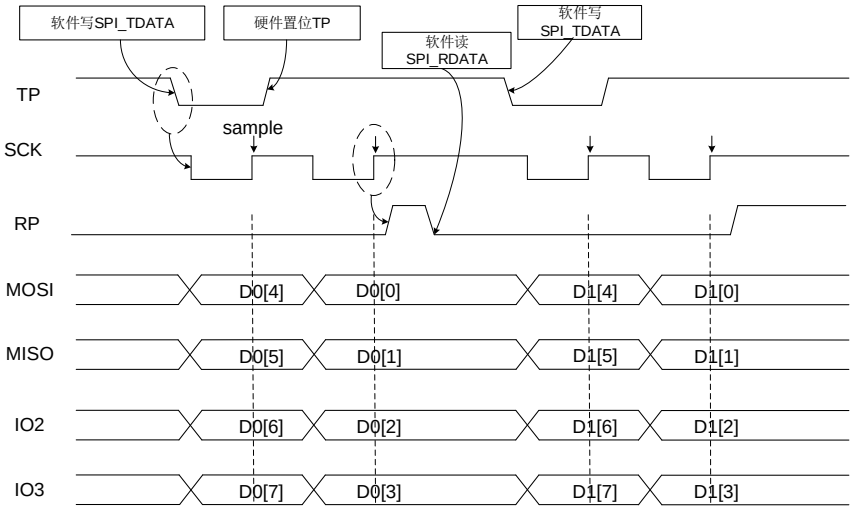
四线读模式

当SPI_QCTL寄存器中的QMOD位和QRD位都置1时，SPI工作在四线读模式。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_TDATA寄存器（TP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_TDATA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_TDATA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、SPI_CTL1、SPI_CFG0、SPI_CFG1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_TDATA寄存器。
4. 等待RP位置1，然后读SPI_RDATA寄存器来获取接收的数据。
5. 写任意数据（例如0xFF）到SPI_TDATA寄存器，以接收下一个字节数据。

图 27-18. SPI 四线模式四线读操作时序图



SPI 停止流程

不同运行模式下采用不同的流程来停止SPI功能。

MFD SFD MTU MTB STU STB

当设备处于全双工或只发送模式，主器件停止提供要发送的数据时，任何传输事务都可以被终止。在这种情况下，时钟在最后一个数据传输完成后停止。TC标志可以被轮询（或者通过ESTCIE = 1使能中断）来等待最后一个数据帧被发送。等待TC = 1或ET = 1（不再发送数据，发送最后一帧数据）。如果使能CRC功能，则在最后一次数据处理后自动发送CRC。在这种情况下，TC / ET将在CRC帧完成后被设置为1。当发送被挂起时，软件必须等待MSTART位被清除。然后通过清除SPIEN位禁用SPI。

MRU MRB

要停止外围设备，必须首先挂起SPI通信。当主设备处于仅接收模式时，将MSPDR设置为1，或者通过ET等待数据传输结束。如果接收流被暂停，请等待SPD = 1。当SPI挂起时，接收但未读取的数据总是存储在RxFIFO中（禁止SPIEN时，RxFIFO会被清空）。读取所有RxFIFO数据（直到RWNE = 0和RPLVL = 0），然后通过清除SPIEN位禁用SPI。

SRU SRB

当应用程序不想接收数据时，可以禁用SPI，任何正在进行的数据都将丢失。

TI 模式

TI模式的停止流程与上面描述过程相同。

SPI 四线模式

应用程序可以作为MFD模式运行，然后清除SPI_QCTL寄存器中的QMOD位和SPI_CTL0寄存器中的SPIEN位。

27.3.8. 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_CFG0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。

在初始化完成后，如果设置了DMATEN，当TP = 1时，SPI每次都会生成一个DMA请求，DMA将确认此请求并自动将数据写入SPI_TDATA寄存器。如果发送数据未准备好，则TP和TXURERR将置1。在这种情况下，将根据TXUROP位选择去发送数据。如果设置了DMAREN，那么当RP = 1时，SPI每次都会生成一个DMA请求，然后DMA将确认这个请求并自动从SPI_RDATA寄存器读取数据。如果在事务结束时ET被设置为1，并且最后一个包不完整，DMA请求将根据RWNE和RPLVL[1:0]设置（在SPI_STAT寄存器中）自动激活以读取剩余的数据。

DMA 数据打包传输

如果传输由DMA（DMATEN = 1或DMAREN = 1）管理，当DZ[4:0] <= 8位，并且SPI_TDATA寄存器以16位或32位访问，或当8位 < DZ[4:0] <= 16位，并且SPI_TDATA寄存器以32位访问，DMA数据打包模式被启用，DMA应该自动管理对SPI_TDATA寄存器的写操作。

不管使用的数据打包模式，也不管要传输的数据数量是DMA数据大小（16位或32位）的倍数。当帧大小很小时，DMA会根据TXSIZE字段设置自动完成传输。在配置DMA时，禁止访问小于配置数据大小的DMA数据。总是确保数据访问至少一个完整的数据帧。

27.3.9. 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。SPI_CRCPOLY寄存器中的值的最高有效位定义多项式长度。如果DZ <= 32位，可提供5 - 33位CRC多项式长度。如果DZ <= 16位，可提供5 - 17位的CRC多项式长度。多项式长度必须大于DZ字段中定义的数据帧长度的值。如果DZ = 32位，必须将SPI_CTL0寄存器中CRCFS位置1，以确保CRC多项式处于全尺寸模式。SPI_CFG0中的CRCSZ位域定义CRC计算单元中被处理并与CRC帧进行比较的最高有效位数。

通过配置SPI_CFG0中的CRCEN位使能CRC功能，必须在SPI使能之前配置。CRCPOLY值只能在没有数据传输的时候修改，建议在CRCEN=0的时候配置。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。CRC的发送与接收都以数据帧的形式实现，数据帧的长度等于SPI_CFG0寄存器中的CRCSZ的值。

在发送阶段，应用程序在最后一个数据写入TxFIFO之后，硬件自动发送存储在SPI_TCRC寄存器中计算完成的CRC值。在接收阶段，最后一个数据从RxFIFO读出后，SPI_RCRC寄存器被存入CRC值，CRC计算单元对所接收到的数据进行CRC计算，并将计算的值与SPI_RCRC中保存的值进行效验，如果校验失败时，SPI_STAT寄存器中CRCERR错误标志位将会置1。可通过软件向SPI_STATC寄存器的CRCERRC位写1来清除CRCERR位。

注意：当SPI被禁止时，或数据传输完成后的新数据采样初期，SPI_TCRC与SPI_RCRC寄存器会被初始化，初始化的值可通过SPI_CTL0寄存器的TXCRCI与RXCRCI设置。

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）。

27.3.10. SPI 中断

状态标志位

- **发送包空间可用标志（TP）**

当TxFIFO有足够的可用位置来容纳一个数据包时设置此位，软件可以通过写入SPI_TDATA寄存器将下一个数据包写入TxFIFO。当TxFIFO没有足够的空间放置下一个数据包时，该位被清除，软件不能通过写入SPI_TDATA寄存器将下一个数据包写入TxFIFO。

- **接收包空间可用标志（RP）**

当RxFIFO非空时设置该位，这意味着至少有一个数据包被接收并存储在接收缓冲区中，并且软件可以通过读取SPI_RDATA寄存器来读取数据包。当RxFIFO为空或RxFIFO中存储的数据不能到达FIFOLVL时，该位被清除。因此，当RxFOFO为空时，软件无法通过读取SPI_RDATA寄存器来读取数据包。或在这种情况下, RxFIFO剩余的数据帧的数量将由SPI_STAT寄存器的RWNE和RPLVL表示,应用程序仍然可以从RxFIFO读取标准数量完整数据包不产生不利影响。

- **传输/接收结束标志（ET）**

ET是一个状态标志，表示传输/接收是正在进行还是结束。在完成传输后，即基于SPI发送或接收TXSIZE数据量时，该标志由硬件设置，并可通过SPI_STATC寄存器中的软件设置ETC位清除。设置为1时，ET标志触发ESTCIE中断。

- **双工数据包标志（DP）**

如果TP和RP标志设置为1，则DP标志设置为1，这意味着TxFIFO有空间进行写操作，而RxFIFO至少包含一个包进行读操作。DP适用于全双工通信，优化数据上传/下载性能，从而最大限度地减少对CPU带宽和系统功率的需求，特别是当SPI在停止模式下运行时。

- **TXFIFO已被填充标志（TXF）**

当应用程序或DMA发送一次传输的所有数据包时，这意味着TXSIZE数据值已被推入TxFIFO, TXF标志将被硬件设置为1。该位可以通过软件将SPI2S_TCRC寄存器的TXFC位写1来清除。TXF标志在TXFIE被设置为1时触发中断。

■ **额外的数据量已被重载标志（TXSERF）**

处理完TXSIZE中的数据数量后，如果TXSER的值不为零，则将TXSER的内容复制到TXSIZE中，并自动清除TXSER的值。然后，传输将增加与TXSIZE中新加载值相对应的数据量。当数据量被发送到TxFIFO后，TXSERF标志被设置为1，并在TXSERFIE上触发一个中断。

■ **挂起标志（SPD）**

在主模式下，当前帧完成或RxFIFO满时(SPI2S_CTL0寄存器中的MASP设置为1)，设备自动挂起接收模式，执行MASPR后，硬件将SDP设置为1。当ESTCIE设置为1，SPD标志设置为1时触发中断。SPD标志可以通过SPI_STATC寄存器SPDC位写入1来清除。

■ **传输完成标志（TC）**

此标志由硬件更改。如果TXSIZE = 0，或TxFIFO为空，TC被设置为1，代表总线上没有活动。如果TXSIZE > 0, TC将在传输结束时设置为1，无论TxFIFO使用情况如何。TC设置为1时，表示传输结束。启用CRC校验模式，发送CRC校验码后TC设置为1。当ESTCIE设置为1，TC标志置1后触发中断被。

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NSSI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。可以通过将SPI_STATC寄存器的CONFERRC位写1来清除CONFERR。当CONFEIE设置为1，CONFERR标志置1触发中断。

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）**

如果RxFIFO没有足够的空间存储接收到的数据，则RXORERR位置1。RxFIFO内容不会被新传入的数据覆盖，因此新传入的数据将丢失。当RXOREIE被设置为1时，RXORERR标志置1触发中断。可以通过在SPI_STATC寄存器的RXORERRC位写入1来清除RXORERR。

■ **帧格式错误（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。当FEIE被设置为1时，FERR标志置1触发中断。通过向SPI_STATC寄存器的FERRC位写入1可以清除FERR。

■ **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。CRCERR标志在CRCERIE设置为1时触发中断。通过向SPI_STATC寄存器的CRCERRC位写入1可以清除CRCERR。

■ **传输下溢错误（TXURERR）**

在从传输模式中TxFIFO是空的，但需要将新数据传入移位寄存器时发送下溢错误标志TXURERR置1。在捕获下溢错误后，提供用于发送的下一个数据取决于TXUROP位，WORDEN位，BYTEN位。TXURERR标志在TXUREIE设置为1时触发中断。可以通过将SPI_STATC寄存器的TXURERRC位写1来清除TXURERR。

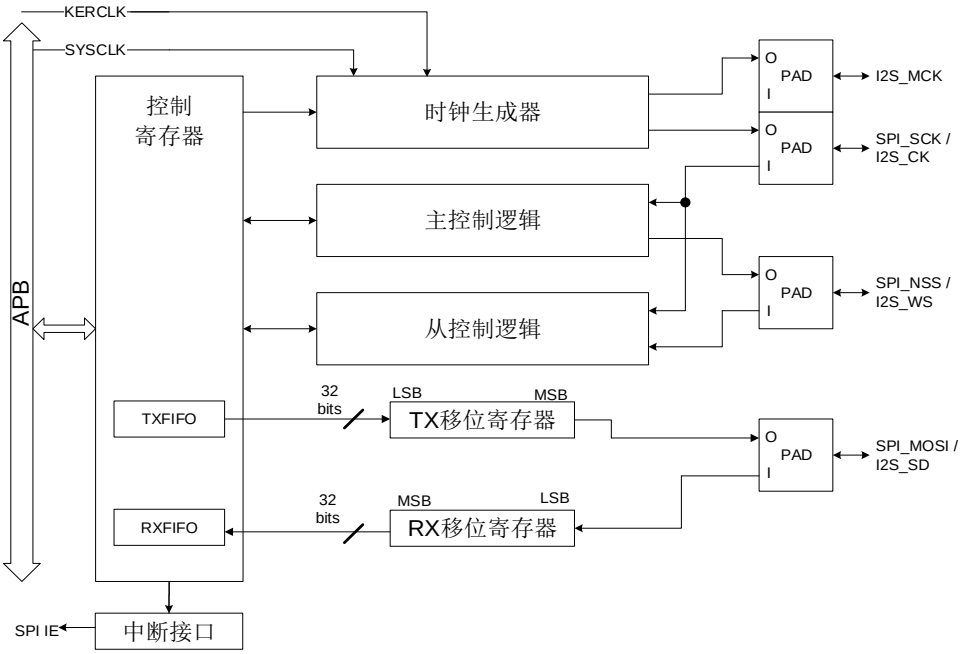
表 27-8. SPI 中断请求

中断标志	描述	清除方式	中断使能位
TP	发送包空间可用标志	当TxFIFO空间少于FIFOLVL，TP被硬件清除	TPIE
RP	接收包空间可用标志	当RxFIFO数据量少于FIFOLVL，RP被硬件清除	RPIE
ET	传输/接收完成标志	ETC 置 1	ESTCIE
DP	双工数据包标志	当 TP 与 RP 清 0，DP 被硬件清除	DPIE
TXF	TxFIFO已被重载标志	TXFC 置 1	TXFIE
TXSERF	额外的数据已被重载标志	TXSERFC 置 1	TXSERFIE
SPD	挂起标志	SPDC 置 1	ESTCIE
TC	传输结束标志	当传输开始时，TC 被硬件清除	ESTCIE
CONFERR	配置错误	CONFERRC 置 1	CONFEIE
RXORERR	接收上溢错误	RXORERRC 置 1	RXOREIE
FERR	帧格式错误	FERRC 置 1	FEIE
CRCERR	CRC错误	CRCERRC 置 1	CRCERIE
TXURERR	传输下溢错误	TXURERRC 置 1	TXUREIE

27.4. I2S 功能说明

27.4.1. I2S 结构框图

图 27-19. I2S 结构框图



- SYSCLK：系统时钟，由APB总线提供。需要访问I2S寄存器时，该时钟必须有效；
- KERCLK：内核时钟，由RCU提供，和系统时钟是异步的关系；
- 时钟信号的频率没有特定限制，但需与用户使用条件及数据传输速度匹配，防止数据丢失；（**注**：建议SYSCLK大于等于KERCLK的频率）
- I2S从机的SCK信号由I2S主机提供。

I2S功能有5个子模块，分别是控制寄存器、时钟生成器、主机控制逻辑、从机控制逻辑和移位寄存器。所有的用户可配置寄存器都在控制寄存器模块实现，其中包括TxFIFO和RxFIFO。时钟生成器用来在主机模式下生成I2S通信时钟。此时钟生成器也是MCK的源。主机控制逻辑用来在主机模式下生成I2S_WS信号并控制通信。从机控制逻辑根据接收到的I2S_CK和I2S_WS信号来控制从机模式的通信。移位寄存器控制I2S_SD上的串行数据发送和接收。

27.4.2. I2S 信号线描述

I2S接口有4个引脚，分别是I2S_CK、I2S_WS、I2S_SD和I2S_MCK。I2S_CK是串行时钟信号，与SPI_SCK共享引脚。I2S_WS是数据帧控制信号，与SPI_NSS共享引脚。I2S_SD是串行数据信号，与SPI_MOSI共享引脚。I2S_MCK是主时钟信号，它最大可提供一个256倍于Fs的时钟频率，其中Fs是音频采样率。

27.4.3. I2S 音频标准

I2S音频标准是通过设置SPI_I2SCTL寄存器中的I2SSTD位来选择的，可以选择四种音频标准：

I2S飞利浦标准，MSB对齐标准，LSB对齐标准和PCM标准。除PCM之外的所有标准都是两个通道（左通道和右通道）的音频数据分时复用I2S接口的，并通过I2S_WS信号来区分当前数据属于哪个通道。对于PCM标准，I2S_WS信号表示帧同步信息。

数据长度和通道长度可以通过SPI_I2SCTL寄存器中的DTLEN位和CHLEN位来设置。由于通道长度必须大于或等于数据长度，所以有四种数据包类型可供选择。它们分别是：16位数据打包成16位数据帧格式，16位数据打包成32位数据帧格式，24位数据打包成32位数据帧格式，32位数据打包成32位数据帧格式。

对于所有标准和数据包类型来说，数据的最高有效位总是最先被发送的。对于所有基于两通道分时复用的标准来说，总是先发送左通道，然后是右通道。

I2S 飞利浦标准

对于I2S飞利浦标准，I2S_WS和I2S_SD在I2S_CK的下降沿变化，I2S_WS在数据的前一个时钟开始有效。各种配置情况的时序图如下所示。

图 27-20. I2S 飞利浦标准时序图（DTLEN = 00，CHLEN = 0，CKPL = 0）

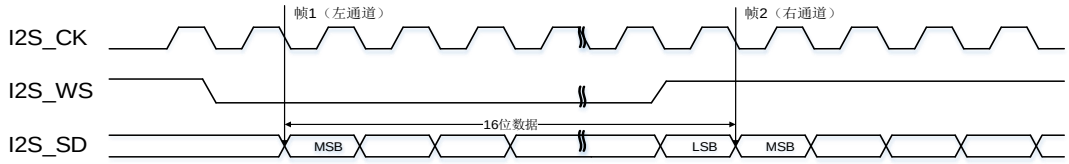


图 27-21. I2S 飞利浦标准时序图（DTLEN = 00，CHLEN = 0，CKPL = 1）

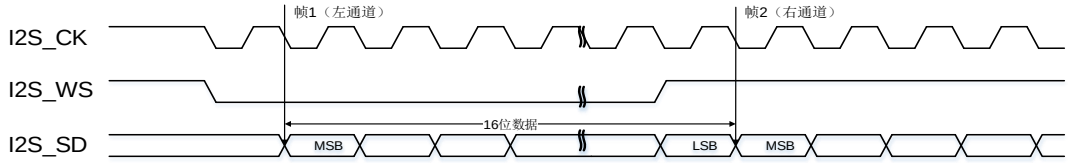


图 27-22. I2S 飞利浦标准时序图（DTLEN = 10，CHLEN = 1，CKPL = 0）

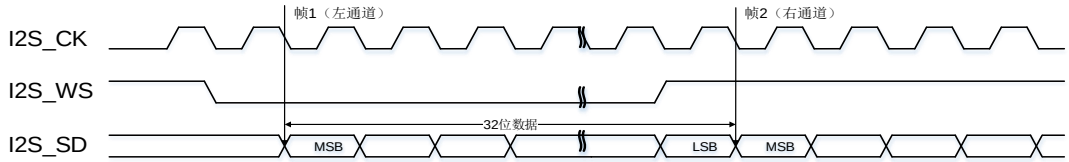


图 27-23. I2S 飞利浦标准时序图（DTLEN = 10，CHLEN = 1，CKPL = 1）

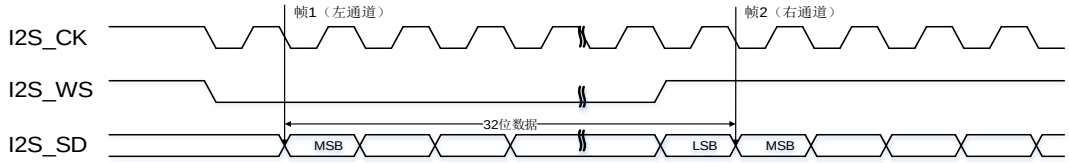


图 27-24. I2S 飞利浦标准时序图（DTLEN = 01，CHLEN = 1，CKPL = 0）

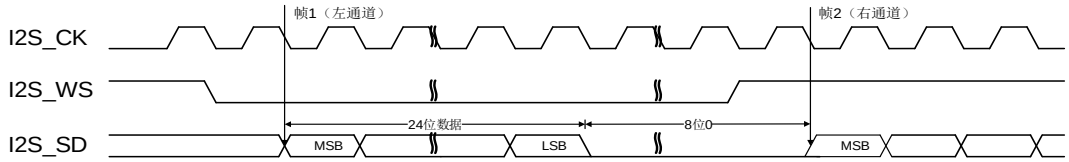


图 27-25. I2S 飞利浦标准时序图（DTLEN = 01，CHLEN = 1，CKPL = 1）

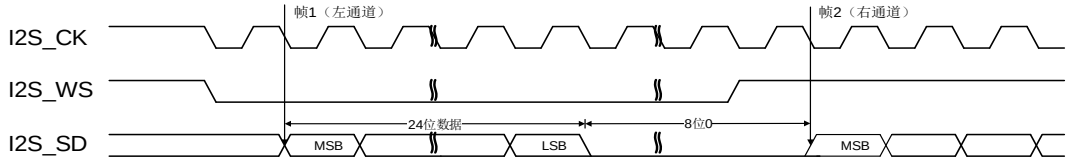


图 27-26. I2S 飞利浦标准时序图（DTLEN = 00，CHLEN = 1，CKPL = 0）

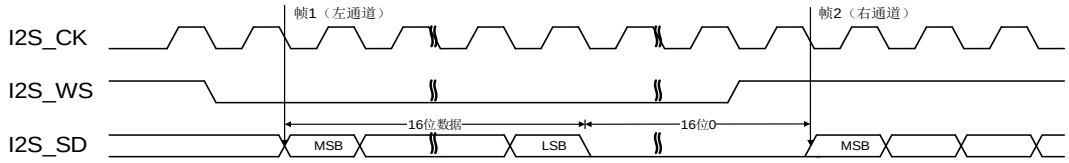
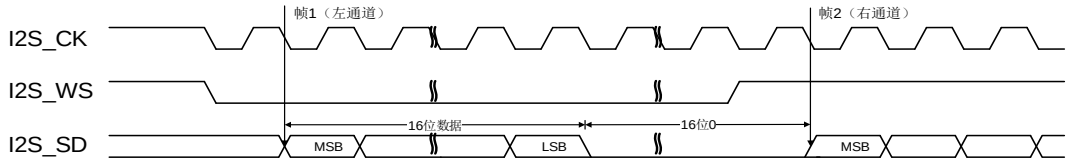


图 27-27. I2S 飞利浦标准时序图（DTLEN = 00，CHLEN = 1，CKPL = 1）



MSB 对齐标准

对于MSB对齐标准，I2S_WS和I2S_SD在I2S_CK的下降沿变化。各个配置情况的时序图如下所示。

图 27-28. MSB 对齐标准时序图（DTLEN = 00，CHLEN = 0，CKPL = 0）

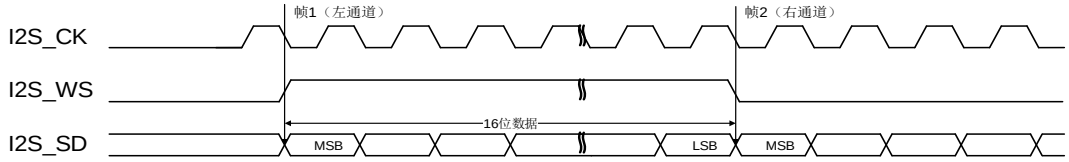


图 27-29. MSB 对齐标准时序图（DTLEN = 00，CHLEN = 0，CKPL = 1）

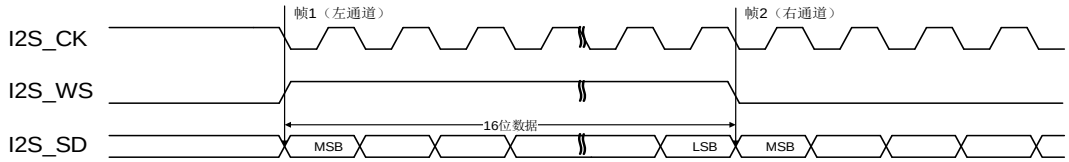


图 27-30. MSB 对齐标准时序图（DTLEN = 10，CHLEN = 1，CKPL = 0）

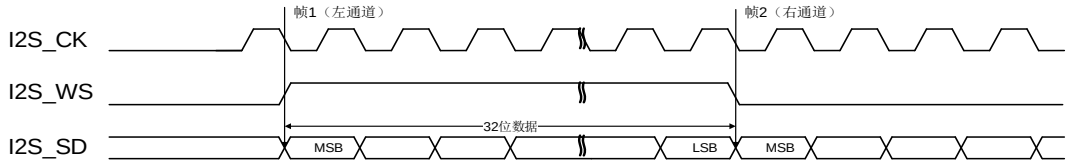


图 27-31. MSB 对齐标准时序图（DTLEN = 10，CHLEN = 1，CKPL = 1）

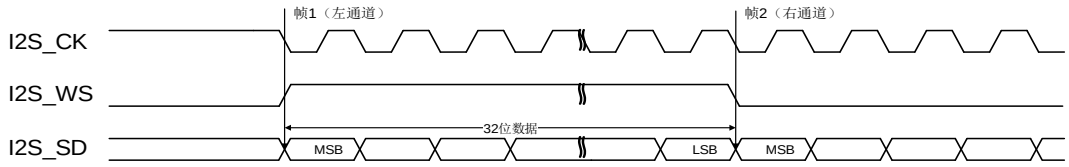


图 27-32. MSB 对齐标准时序图（DTLEN = 01，CHLEN = 1，CKPL = 0）

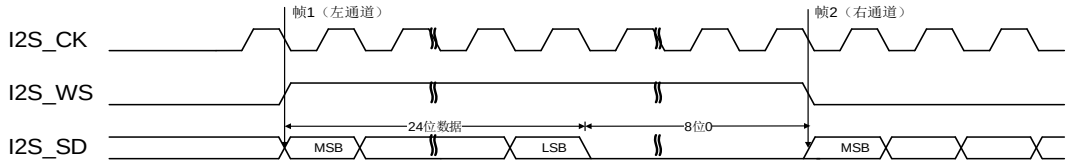


图 27-33. MSB 对齐标准时序图（DTLEN = 01，CHLEN = 1，CKPL = 1）

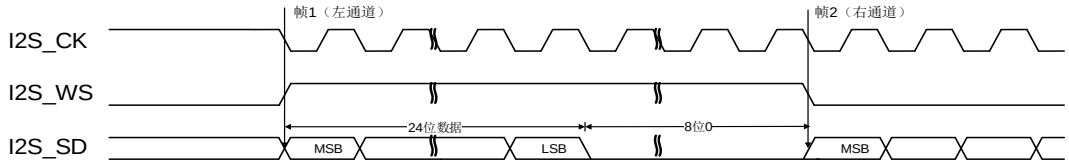


图 27-34. MSB 对齐标准时序图（DTLEN = 00，CHLEN = 1，CKPL = 0）

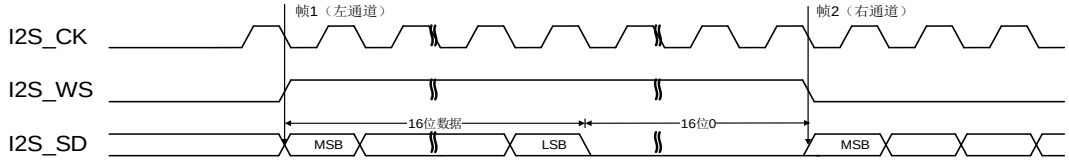
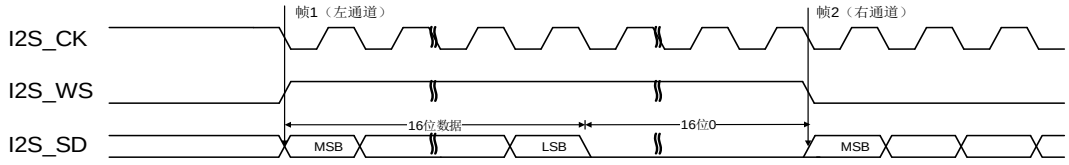


图 27-35. MSB 对齐标准时序图（DTLEN = 00，CHLEN = 1，CKPL = 1）



LSB 对齐标准

对于LSB对齐标准，I2S_WS和I2S_SD在I2S_CK的下降沿变化。在通道长度与数据长度相同的情况下，LSB对齐标准和MSB对齐标准是完全相同的。对于通道长度大于数据长度的情况，LSB对齐标准的有效数据与最低位对齐，而MSB对齐标准的有效数据与最高位对齐。通道长度

大于数据长度的各种配置情况时序图如下所示。

图 27-36. LSB 对齐标准时序图（DTLEN = 01，CHLEN = 1，CKPL = 0）

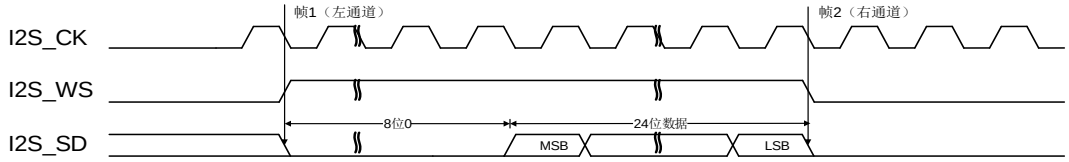


图 27-37. LSB 对齐标准时序图（DTLEN = 01，CHLEN = 1，CKPL = 1）

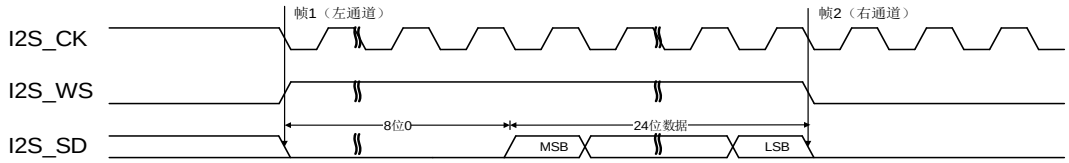


图 27-38. LSB 对齐标准时序图（DTLEN = 00，CHLEN = 1，CKPL = 0）

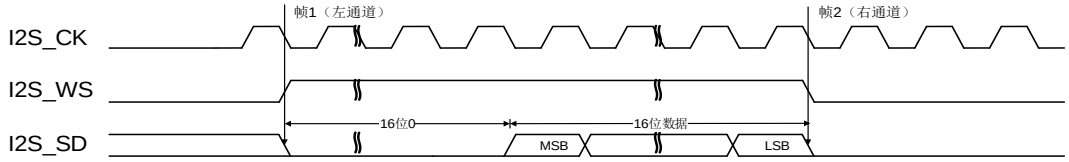
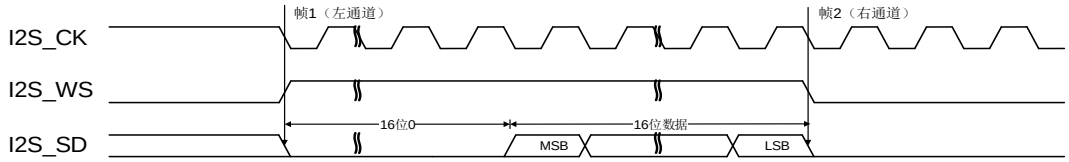


图 27-39. LSB 对齐标准时序图（DTLEN = 00，CHLEN = 1，CKPL = 1）



PCM 标准

对于PCM标准，I2S_WS和I2S_SD在I2S_CK的上升沿变化，I2S_WS信号表示帧同步信息。可以通过SPI_I2SCTL寄存器的PCMSMOD位来选择短帧同步模式和长帧同步模式。SPI_TDATA / SPI_RDATA寄存器的处理方式与I2S飞利浦标准完全相同。短帧同步模式的各种配置情况时序图如下所示。

图 27-40. PCM 标准短帧同步模式时序图（DTLEN = 00，CHLEN = 0，CKPL = 0）

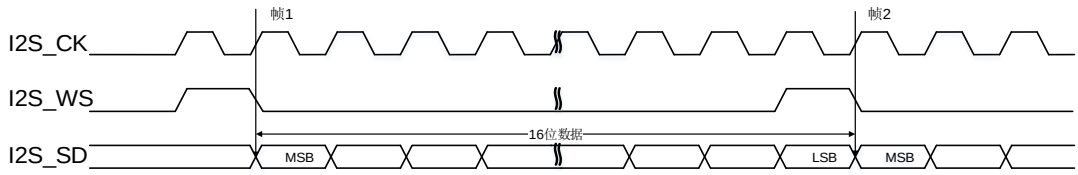


图 27-41. PCM 标准短帧同步模式时序图（DTLEN = 00，CHLEN = 0，CKPL = 1）

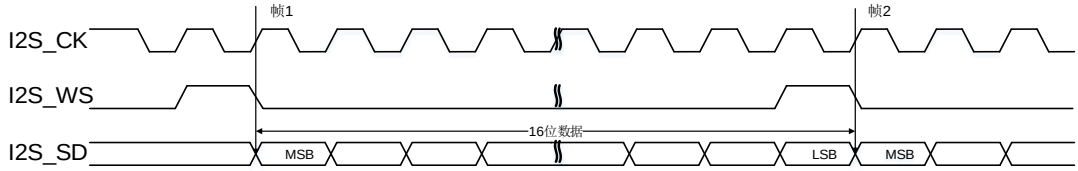


图 27-42. PCM 标准短帧同步模式时序图（DTLEN = 10，CHLEN = 1，CKPL = 0）

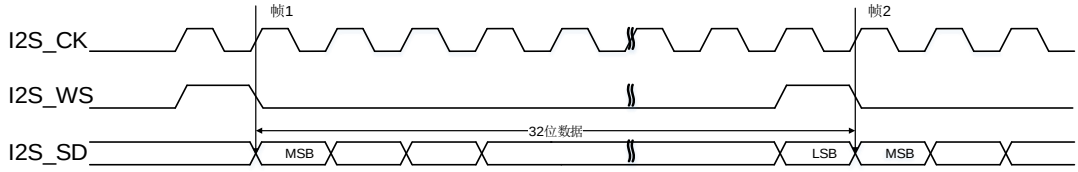


图 27-43. PCM 标准短帧同步模式时序图（DTLEN = 10，CHLEN = 1，CKPL = 1）

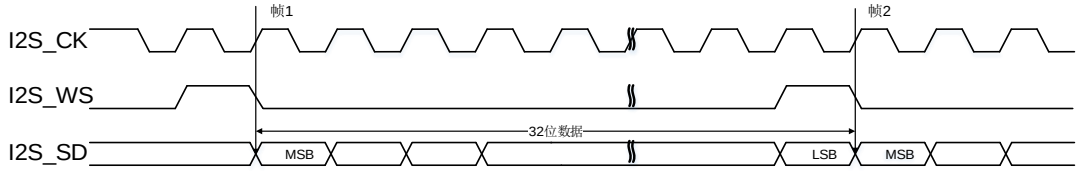


图 27-44. PCM 标准短帧同步模式时序图（DTLEN = 01，CHLEN = 1，CKPL = 0）

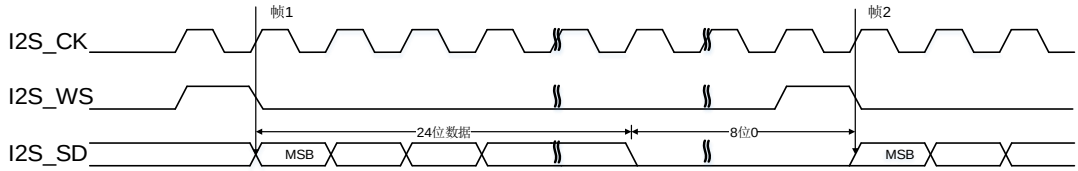


图 27-45. PCM 标准短帧同步模式时序图（DTLEN = 01，CHLEN = 1，CKPL = 1）

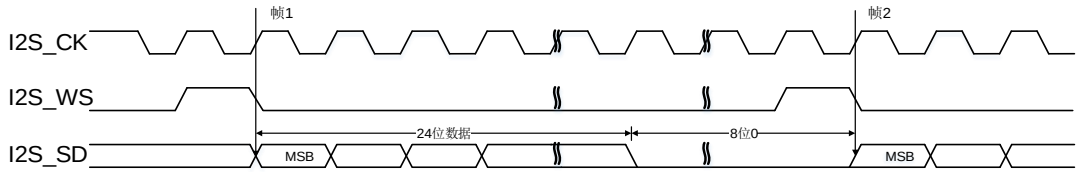


图 27-46. PCM 标准短帧同步模式时序图（DTLEN = 00，CHLEN = 1，CKPL = 0）

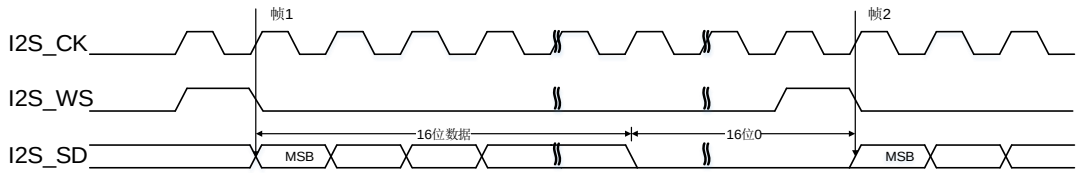
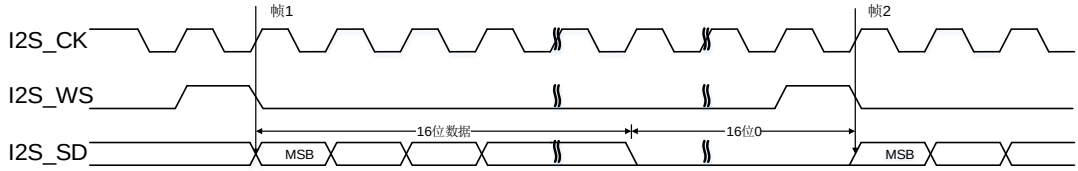


图 27-47. PCM 标准短帧同步模式时序图（DTLEN = 00，CHLEN = 1，CKPL = 1）



长帧同步模式的各种配置情况时序图如下所示。

图 27-48. PCM 标准长帧同步模式时序图（DTLEN = 00，CHLEN = 0，CKPL = 0）

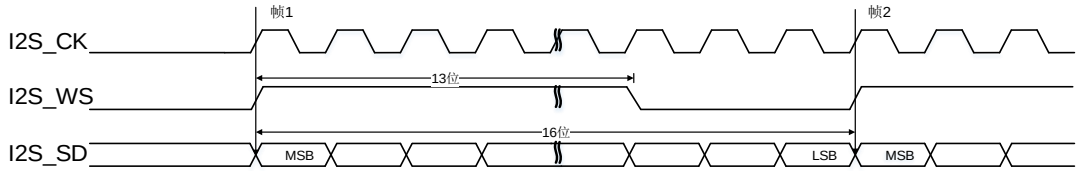


图 27-49. PCM 标准长帧同步模式时序图（DTLEN = 00，CHLEN = 0，CKPL = 1）

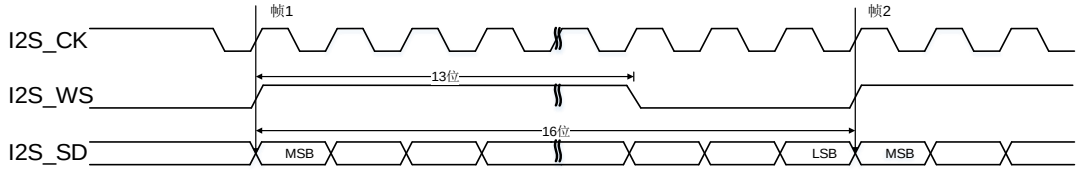


图 27-50. PCM 标准长帧同步模式时序图（DTLEN = 10，CHLEN = 1，CKPL = 0）

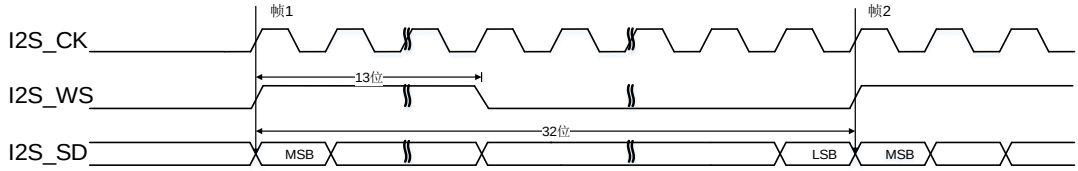


图 27-51. PCM 标准长帧同步模式时序图（DTLEN = 10，CHLEN = 1，CKPL = 1）

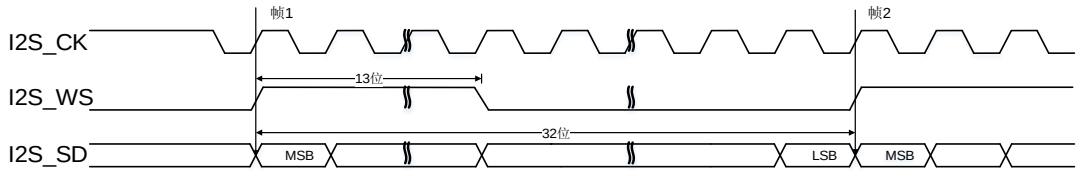


图 27-52. PCM 标准长帧同步模式时序图（DTLEN = 01，CHLEN = 1，CKPL = 0）

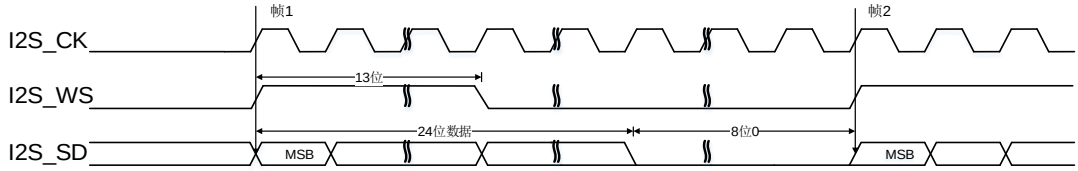


图 27-53. PCM 标准长帧同步模式时序图（DTLEN = 01，CHLEN = 1，CKPL = 1）

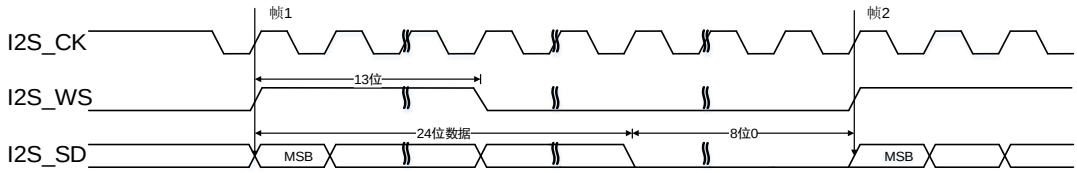


图 27-54. PCM 标准长帧同步模式时序图（DTLEN = 00，CHLEN = 1，CKPL = 0）

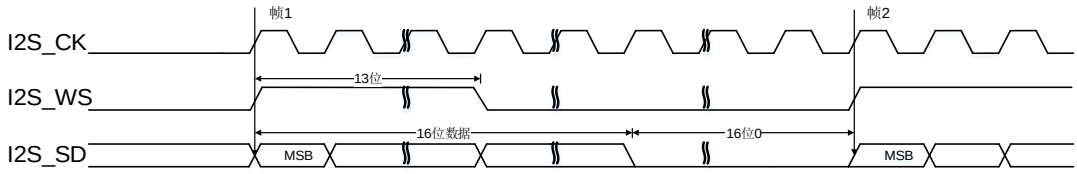
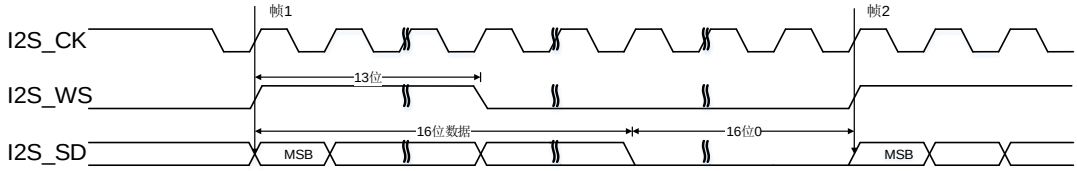
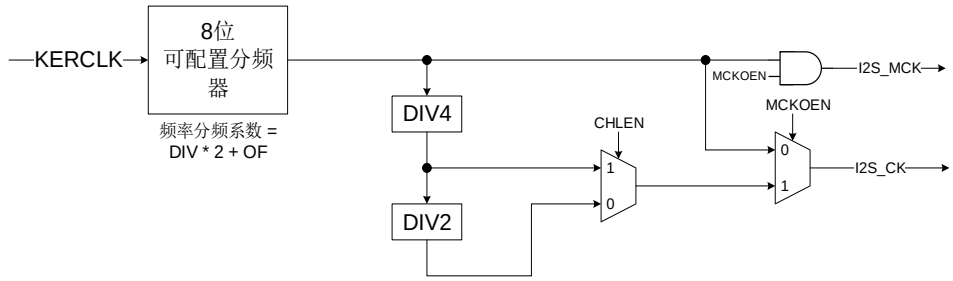


图 27-55. PCM 标准长帧同步模式时序图（DTLEN = 00，CHLEN = 1，CKPL = 1）



27.4.4. I2S 时钟

图 27-56. I2S 时钟生成结构框图



I2S时钟生成器框图如*图27-56. I2S时钟生成结构框图*所示。I2S接口时钟是通过SPI_I2SCTL寄存器的DIV位，OF位和MCKOEN位以及SPI_I2SCTL寄存器的CHLEN位来配置的。时钟源是内核时钟（KERCLK）。I2S比特率可以通过*表27-9. I2S比特率计算公式*所示的公式计算。

表 27-9. I2S 比特率计算公式

MCKOEN	CHLEN	公式
0	0	KERCLK /（DIV * 2 + OF）
0	1	KERCLK /（DIV * 2 + OF）
1	0	KERCLK /（8 *（DIV * 2 + OF））
1	1	KERCLK /（4 *（DIV * 2 + OF））

音频采样率（Fs）和I2S比特率的关系由如下公式定义：

Fs = I2S比特率 /（通道长度 * 通道数）

所以，为了得到期望的音频采样率，时钟生成器需要按*表27-10. 音频采样频率计算公式*所列的公式进行配置。

注意：I2S串行时钟的配置值建议设置为低于I2S挂载APB对应的PCLK时钟的1/6倍（不包含1/6）。

表 27-10. 音频采样频率计算公式

MCKOEN	CHLEN	公式
0	0	KERCLK /（32 *（DIV * 2 + OF））
0	1	KERCLK /（64 *（DIV * 2 + OF））
1	0	KERCLK /（256 *（DIV * 2 + OF））
1	1	KERCLK / （256 * （DIV * 2 + OF））

27.4.5. RxFIFO 和 TxFIFO

RxFIFO 和 TxFIFO 用于 I2S 数据传输的不同方向，它们可以使 I2S 工作在一个连续的流中，可以防止短数据帧或中断/DMA 延迟太长发生溢出错误。

对 SPI_TDATA 寄存器的写访问将写入的数据存储在 TxFIFO 的末尾，而对 SPI_RDATA 的读访问将返回 RxFIFO 中尚未被读取的最早的值。在 I2S 模式下，左音频采样和右音频采样在 FIFO 中交错进行。这意味着对于发送操作，用户必须先用左通道数据填充 TxFIFO，然后是右通道，以此类推。对于接收模式，从 RxFIFO 读取的第一个数据是左通道，下一个数据是右通道，以此类推。

FIFO 处理取决于数据长度（DTLEN 值）、访问 FIFO 寄存器的大小（8、16 或 32 位）。TxFIFO / RxFIFO 的范围为 16x32 位，最大访问数据长度为 32 位，*表 27-11. I2SX FIFO 最大存储数据帧数量*描述了在不同数据长度时，FIFO 中可存放的最大帧数量。（N = FIFO 范围 / 32= 16 x 32 / 32 = 16）

表 27-11. I2SX FIFO 最大存储数据帧数量

数据长度（DTLEN）	DTLEN = 16 位	DTLEN = 24 位	DTLEN = 32 位
FIFO 存储帧数（WORDEN = 0）	N	-	-
FIFO 存储帧数（WORDEN = 1）	2N	N	N

可根据可编程的 FIFO 阈值生成中断或者 DMA 请求。FIFOLVL 的影响和 SPI 的一致。

注意：SPI_TDATA和SPI_RDATA内数据是默认右对齐的。当I2S设备被禁止时（I2SEN = 0），RxFIFO和TxFIFO中的数据将被清空。

27.4.6. 运行

运行模式

运行模式是通过 SPI_I2SCTL 寄存器的 I2SOPMOD 位来选择的。共有四种运行模式可供选择：主机发送模式，主机接收模式，从机发送模式和从机接收模式。各种运行模式下 I2S 接口信号

的方向如*表 27-12. 各种运行模式下 I2S 接口信号的方向*所示。

表 27-12. 各种运行模式下 I2S 接口信号的方向

运行模式	I2S_MCK	I2S_CK	I2S_WS	I2S_SD
主机发送	输出或 NU（1）	输出	输出	输出
主机接收	输出或 NU（1）	输出	输出	输入
从机发送	输入或 NU（1）	输入	输入	输出
从机接收	输入或 NU（1）	输入	输入	输入

1. NU表示该引脚没有被I2S使用，可以用于其他功能。

I2S 初始化流程

I2S初始化过程包括以下五个步骤。如果要初始化I2S工作在主机模式，五个步骤都要执行，如果要初始化I2S工作在从机模式，只需要执行步骤2、3、4、5、6、7。

- 步骤1：配置SPI_I2SCTL寄存器的DIV[7:0]位，OF位和MCKOEN位，定义I2S的比特率和选择是否需要提供I2S_MCK信号；
- 步骤2：配置SPI_I2SCTL寄存器的CKPL位，定义空闲状态的时钟极性；
- 步骤3：配置FIFO等级（在SPI_CFG0寄存器中的FIFOLVL[3:0]位）；
- 步骤4：配置SPI_I2SCTL寄存器的I2SSEL位，I2SSTD[1:0]位，PCMSMOD位，I2SOPMOD[1:0]位，DTLEN[1:0]位和CHLEN位，定义I2S的特性；
- 步骤5：配置TPIE位，RPIE位，TXUREIE位，RXOREIE位，FEIE位，DMATEN位和DMAREN位，选择中断源和DMA功能。此步骤可选；
- 步骤6：将SPI_I2SCTL寄存器的I2SEN位置1，来启动I2S；
- 步骤7：将SPI_CTL0寄存器中的MSTART置1，来激活串行接口。

I2S 基本发送和接收流程

发送流程

在完成初始化过程之后，I2S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TxFIFO时，发送过程开始。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TxFIFO中。

当I2S开始发送一个数据帧时，首先将这个数据帧从TxFIFO加载到移位寄存器中，然后开始发送加载的数据。相关操作可参考描述。

对SPI_TDATA的写访问由TP事件管理。当TP标志设置为1时，应用程序对I2S数据寄存器写入适当数量的数据，以传输数据包的内容。在上传新的完整包后，应用程序检查TP值，检查TxFIFO是否可以接收额外的数据包，如果TP = 1，则逐包上传，直到TP读取0。如果传输大小和数据包大小没有对齐，则最后要传输的数据包数无法达到配置的大小（由FIFOLVL设置）。应用程序仍然可以将标准数量的先前完整数据包写入TxFIFO，而不会产生不良影响：只有一致的数据（完整的数据帧）将传输到TxFIFO，而冗余的写入时间（或任何不完整的数据）将被忽略。

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_TDATA寄存器中。只要TxFIFO中存在数据，数据发送便一直继续，直至TxFIFO

变为空。

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RxFIFO，且RP（RxFIFO非空）位置1。软件通过读SPI_RDATA寄存器获得接收的数据，此操作会自动清除RP标志位（当RxFIFO数据量少于FIFOLVL标准）。

对SPI_RDATA的读访问由RP事件管理。当RP标志设置为1时，应用程序读取I2S数据寄存器相当数量的数据，以下载单个数据包内容。下载完整数据包后，应用程序会检查RP值，查看RxFIFO中是否有其他数据包，如果有，则逐包下载，直到RP读到0。在接收结束时，可能会出现RxFIFO中仍然有一些数据可用，但没有达到FIFOLVL级别，因此RP不会被设置为1。在这种情况下，RxFIFO中剩余的RX数据帧的数量将由SPI_STAT寄存器中的RWNE和RPLVL表示。如果传输大小和数据包大小没有对齐，当最后接收的数据包数量不能达到配置的大小（由FIFOLVL设置）时，就会出现上述情况。然而，应用程序仍然可以从RxFIFO读取标准数量的以前完整的数据包，而不会产生不良影响：只有一致的数据（完整的数据帧）将从RxFIFO读取，而冗余的读取（或任何不完整的数据）将读取0。

I2S 停止流程

I2S主机停止流程：

- 步骤1：停止总线时钟和DMA功能；
- 步骤2：将I2SEN置0，禁止I2S模块。

I2S从机停止流程：

- 步骤1：将I2SEN置0，禁止I2S模块；
- 步骤2：停止总线时钟和DMA功能。

27.4.7. DMA 功能

DMA功能与SPI模式完全一样，唯一不同的地方就是I2S模式不支持CRC功能。

27.4.8. I2S 中断

状态标志位

SPI_STAT寄存器中有两个可用的标志位（TP、RP），SPI_I2SCTL寄存器中有一个可用的标志位（I2SCH），用户通过这些标志位可以全面监视I2S总线的状态。

- **发送包空间可用标志（TP）**

当TxFIFO有足够的可用位置来容纳一个数据包时设置此位，软件可以通过写入SPI_TDATA寄存器将下一个数据包写入TxFIFO。当TxFIFO没有足够的空间放置下一个数据包时，该位被清除，软件不能通过写入SPI_TDATA寄存器将下一个数据包写入TxFIFO。

- **接收包空间可用标志（RP）**

当RxFIFO非空时设置该位，这意味着至少有一个数据包被接收并存储在接收缓冲区中，并且软件可以通过读取SPI_RDATA寄存器来读取数据包。当RxFIFO为空或RxFIFO中存储的数据不能到达FIFOLVL时，该位被清除。因此，当RxFOFO为空时，软件无法通过读取SPI_RDATA寄存器来读取数据包。或在这种情况下, RxFIFO剩余的数据帧的数量将由SPI_STAT寄存器的RWNE和RPLVL表示,应用程序仍然可以从RxFIFO读取标准数量完整数据包不产生不利影响。

- **I2S通道标志（I2SCH）：**

I2SCH用来表明当前传输数据的通道信息，对PCM音频标准来说没有意义。在发送模式下，I2SCH标志在每次发送通道切换时更新，在接收模式下，I2SCH标志在每次接收通道切换时更新。该标志位不会产生任何中断。

注：因为FIFO的存在，该位的变化，与TP / RP不再有相关性。变化会发生在一个channel传输结束时（channel的传输结束，并不代表数据传输结束，例如channel 32位，data 16位时，channel传输结束代表32位完成的时候）。

错误标志

有三个错误标志：

- **接收上溢错误标志（RXORERR）：**

当接收缓冲区已满且又接收到一个新的数据时，接收过载错误标志RXORERR置位。当接收过载发生时，接收缓冲区中的数据没有更新，新接收的数据丢失。当RXOREIE被设置为1时，RXORERR标志置1触发中断。可以通过在SPI_STATC寄存器的RXORERRC位写入1来清除RXORERR。

注：I2S模式存在一种硬件机制，可防止因上溢导致的左右通道数据互换的错误。比如，数据接收顺序为L0 -> R0 -> L1 -> R1 -> L2 -> R2 -> L3 -> R3…LN -> RN（L代表左通道数据，R代表右通道数据）。当上溢发生在R1接收后，L2数据丢失，当RxFIFO恢复后（可接收数据），硬件会自动丢弃R2数据，并接收L3数据到左通道，接收R3到右通道。当上溢发生在L2接收后，R2数据丢失，当RxFIFO恢复后（可接收数据），硬件会自动丢弃L3数据，并接收R3数据到右通道，接收L4数据到左通道。

- **帧格式错误（FERR）：**

在从I2S模式下，I2S模块监视I2S_WS信号，如果I2S_WS信号在一个错误的位置发生翻转，将会置位FERR帧错误标志位。当FEIE被设置为1时，FERR标志置1触发中断。通过向SPI_STATC寄存器的FERRC位写入1可以清除FERR。

- **发送下溢错误标志（TXURERR）：**

在从传输模式中TxFIFO是空的，但需要将新数据传入移位寄存器时发送下溢错误标志TXURERR置1，这种情况发生时，至少会丢失一个数据。TXURERR标志在TXUREIE设置为1时触发中断。可以通过将SPI_STATC寄存器的TXURERRC位写1来清除TXURERR。

注：I2S模式存在一种硬件机制，可防止因下溢导致的左右通道数据互换的错误。比如，数据发送顺序为L0 -> R0 -> L1 -> R1 -> L2 -> R2 -> L3 -> R3…LN- > RN（L代表左通道数据，R代表右通道数据）。当下溢发生在R1发送后，L2数据未及时传入TxFIFO导致TxFIFO为空，硬件会自动将R1数据传入左通道，再传入右通道，当L2数据传入TxFIFO后，再将L2数据传入左通

道，R2数据传入右通道。当上溢发生在L2发送后，R2数据未及时传入TxFIFO导致TxFIFO为空，硬件会自动将L2数据传入右通道，再传入左通道，当R2数据传入TxFIFO后，再将R2数据传入右通道，L3数据传入左通道。

*表27-13. I2S中断*总结了I2S中断事件和相应的使能位。

表 27-13. I2S 中断

中断标志	描述	清除方式	中断使能位
TP	发送包空间可用标志	当 TxFIFO 空间少于 FIFOLVL，TP 被硬件清除	TPIE
RP	接收包空间可用标志	当 RxFIFO 数据量少于 FIFOLVL，RP 被硬件清除	RPIE
TXURERR	发送下溢错误	TXURERRC 置 1	TXUREIE
RXORERR	接收上溢错误	RXORERRC 置 1	RXOREIE
FERR	帧格式错误	FERRC 置 1	FEIE

27.5. SPI / I2S 寄存器

SPI0 / I2S0基地址：0x4001 3000

SPI1 / I2S1基地址：0x4000 3800

SPI2 / I2S2基地址：0x4000 3C00

SPI3基地址：0x4001 3400

SPI4基地址：0x4001 5000

SPI5 / I2S5基地址：0x4001 3800

27.5.1. 控制寄存器 0（SPI_CTL0）

地址偏移：0x00
复位值：0x0000 0000

该寄存器可以按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXCRCI	RXCRCI	CRCFS	NSSI	保留	MSPDR	MSTART	MASP	保留							SPIEN
rw	rw	rw	rw		w	rw	rw								rw

位/位域	名称	描述
31:17	保留	必须保持复位值。
16	IOAFEN	相关IO的AF配置功能使能 0：相关IO的AF配置功能使能

		1：相关IO的AF配置功能禁止 该位可被软件设置，并且可被硬件清0，不论SPIEN位是否由1变0，或者当SPIEN位和CONFERR位为0时可由软件清0。当CONFERR位置1时，该位被清0，并且不能置1。当SPIEN位使能后，该位被写保护。当该位被置1后，SPI_CFG1寄存器不能改变。
15	TXCRCI	发送器CRC初始化配置 0：全0模式使用 1：全1模式使用
14	RXCRCI	接收器CRC初始化配置 0：全0模式使用 1：全1模式使用
13	CRCFS	全尺寸CRC多项式配置 0：不使用全尺寸CRC多项式 1：使用全尺寸CRC多项式
12	NSSI	内部NSS信号输入电平 0：NSS引脚被拉低 1：NSS引脚被拉高 只有当NSSIM位置1，该位有效。该位的值作用到外设NSS引脚的输入状态，并且NSS引脚的IO值无效。
11	保留	必须保持复位值。
10	MSPDR	SPI主机模式挂起请求 0：无挂起请求 1：有挂起请求 该位读取为0。在SPI主机模式中，如果该位被软件置1，当MSTART在当前帧传输结束后被复位，SPI交互将被挂起。用户需要通过检查SPI_STAT寄存器中的SPD标志去判断传输是否结束。在SPI禁止前，主机通信必须被挂起（可通过该位或清空SPI_TDATA寄存器实现）。
9	MSTART	主机启动传输 0：主机处于空闲状态 1：主机开始传输，或者被自动挂起功能临时挂起 该位可被软件置1，当SPI_STAT寄存器中的ET = 1或当收到挂起请求时被硬件清零。 在SPI模式中，只有当SPIEN = 1和SPI_CFG1寄存器中的MSTMOD = 1时，该位可被设置。 在I2S/PCM模式中，只有当I2SEN = 1，该位可被设置。
8	MASP	主机在接收模式时被自动挂起 0：不论上溢是否发生，SPI的数据流和时钟都持续 1：当上溢出现之前，当RxFIFO已满时，SPI数据流被挂起。SPI_STAT寄存器中的SPD标志将置1 当SPI通信被暂停以防止上溢时，下一帧的几个位可能由于内部同步延迟而被同步

		出去。读取RxFIFO后，通信恢复，后续的位传输继续不受任何限制。 出于同样的原因，当数据大小小于8位时，自动挂起不是很可靠。在这种情况下，通过设置MDFD参数值，应用的数据帧之间的插入延迟来实现安全挂起；数据大小和交错SPI周期的总和，应该始终产生至少8个SPI时钟周期的间隔。 **注意：**MASP只能在接收模式下开启，否则可能会引发RXORERR（接收上溢）错误。
7:1	保留	必须保持复位值。
0	SPIEN	SPI使能 0：SPI设备禁止 1：SPI设备使能 该位可被软件置1或清零，并且当SPI_STAT寄存器的CONFERR位置1时，该位不能置1。

27.5.2. 控制寄存器 1（SPI_CTL1）

地址偏移：0x04

复位值：0x0000 0000

该寄存器可以按字（32位）访问。

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

位/位域	名称	描述
31:16	TXSER[15:0]	当先前保存在TXSIZE中的数据量被传输完后，将重新加载存储在TXSER中的扩展数据量到TXSIZE中。 这些位只能在其值为0时被软件设置。TXSIZE重新加载后，它会被硬件清除。如果是最后一次重加载必须在CTXSIZE（在SPI_STAT寄存器中）计数器到达1之前编写最后一次TXSER，除此之外的重加载必须在CTXSIZE（在SPI_STAT寄存器中）计数器达到1（如果配置CRCEN则要在2）之前，并且CTXSIZE计数器在小于上次配置TXSER减1时预先编写新的TXSER值，否则将不考虑重载，通信将正常终止。 **注意：**TXSER设置需要大于1。
15:0	TXSIZE[15:0]	当前要传输的数据量 这些位可以通过软件修改，当MSTART位设置为1时不能修改。当TXSIZE为0，MSTART设置为1时，将开始无限传输。当CRC使能时，TXSIZE不能设置为0xFFFF / 0x0001。

27.5.3. 配置寄存器 0（SPI_CFG0）

地址偏移：0x08

复位值：0x0007 0007

该寄存器可以按字（32位）访问。当SPI使能后，除了DMATEN和DMAREN位，该寄存器被写保护。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	PSC[2:0]			保留			WORDEN	BYTEN	CRCEN	保留	CRCSZ[4:0]				
	rw						rw	rw	rw				rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMATEN	DMAREN	保留	TXURDT[1:0]		TXUROP[1:0]		FIFOLVL[3:0]				DZ[4:0]				
rw	rw		rw		rw		rw				rw				

位/位域	名称	描述
31	保留	必须保持复位值。
30:28	PSC[2:0]	主时钟预分频选择 000：KERCLK / 2 001：KERCLK / 4 010：KERCLK / 8 011：KERCLK / 16 100：KERCLK / 32 101：KERCLK / 64 110：KERCLK / 128 111：KERCLK / 256 **注意：**四线模式下，000 / 001配置不可用。 TI模式下，当DZ为4时，000配置不可用。
27:25	保留	必须保持复位值。
24	WORDEN	字访问使能 该位用于指示对FIFO的访问宽度，并设置产生RWNE的RXFIFO的阈值。 0：按照BYTEN访问 1：字访问 在I2S模式下，为了保证声道数据稳定，WORDEN与DTLEN有关。 当DTLEN = 0时，WORDEN必须为0。 当DTLEN > 0时，WORDEN必须为1。
23	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RWNE的RXFIFO的阈值。 0：半字访问 1：字节访问 在I2S模式下，为了保证声道数据稳定，该位必须始终为0。
22	CRCEN	CRC计算使能 0：CRC计算禁止

		1：CRC计算使能
21	保留	必须保持复位值。
20:16	CRCSZ[4:0]	CRC长度 该位域必须等于DZ值或DZ值的倍数。 00000：未使用 00001：未使用 00010：未使用 00011：4位 00100：5位 00101：6位 ….. 11101：30位 11110：31位 11111：32位
15	DMATEN	发送缓冲区DMA使能 0：发送缓冲区DMA禁止 1：发送缓冲区DMA使能
14	DMAREN	接收缓冲区DMA使能 0：接收缓冲区DMA禁止 1：接收缓冲区DMA使能
13	保留	必须保持复位值。
12:11	TXURDT[1:0]	从机发送时检测下溢 00：在数据帧开始时检测到下溢（无第一位保护） 01：在最后一个数据帧结束时检测到下溢 10：在NSS信号开始时检测到下溢 11：保留
10:9	TXUROP[1:0]	从机发送时检测到下溢后的处理 00：从机发送定义在SPI_URDATA寄存器中的常数 01：从机发送从主机获取的最后一帧数据 10：从机发送最后一次发送的数据帧（该数据帧存储在TxFIFO中） 11：保留
8:5	FIFOLVL	FIFO阈值 定义单个数据包中包含的数据帧数。数据包的大小不应超过FIFO空间的一半。 0000：1个数据帧 0001：2个数据帧 0010：3个数据帧 0011：4个数据帧 …. 1101：14个数据帧 1110：15个数据帧

1111：16个数据帧

如果配置的数据包大小与数据寄存器访问对齐，SPI接口将更有效。如果SPI数据寄存器作为16位访问，并且DZ <= 8位，最好选择FIFOLVL = 2、4、6等。如果SPI数据寄存器作为32位访问，并且DZ > 8位，最好选择FIFOLVL = 2、4、6等。而如果DZ <= 8位，则最好选择FIFOLVL = 4、8、12等。

4:0	DZ[4:0]	数据位宽 这些位配置一帧数据的位数： 00000：保留 00001：保留 00010：保留 00011：4位，（当数据宽度是4 bit时，必须使用字 / 半字访问FIFO，否则会有数据错乱的风险） 00100：5位 00101：6位 00110：7位 …… 11101：30位 11110：31位 11111：32位

27.5.4. 配置寄存器 1（SPI_CFG1）

地址偏移：0x0C

复位值：0x0000 0000

该寄存器可以按字（32位）访问。当SPI使能或IOAFEN位置1，该寄存器被写保护。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFCTL	NSSCTL	NSSDRV	NSSIOPL	保留	NSSIM	CKPL	CKPH	LF	MSTMOD	TMOD	保留		BDEN	BDOEN	RO
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
SWPMIO	保留							MDFD[3:0]					MSSD[3:0]		
rw								rw					rw		

位/位域	名称	描述
31	AFCTL	AF GPIOs控制 当SPI禁止时，该位可被设置或清零。 0：外设在禁止时不能控制GPIOs 1：外设总是控制相关的GPIOs 当由于特定的配置原因（如CRC重置，或CKPH更改）必须临时禁用SPI主服务器时，将此位设置为1将强制为备用功能模式配置的相关输出处于与当前SPI配置对应的状态，以防止出现毛刺。在从模式下，不应该使用这个位，因为一旦SPI被禁用，任何从发送器都不能强制其MISO输出。 **注意：**在MFD、MRU和MRB模式下，禁止使用AFCTL。

30	NSSCTL	主机模式时NSS引脚输出控制 0：NSS保持有效电平直到数据传输完成，之后通过ET标志变为无效电平 1：当MDFD[3:0] > 1时，SPI数据帧之间插入交错脉冲
29	NSSDRV	主机模式NSS输出使能 0：输出禁止 1：输出使能
28	NSSIOPL	NSS输入 / 输出极性选择 0：低电平有效 1：高电平有效
27	保留	必须保持复位值。
26	NSSIM	NSS输入信号管理模式 0：NSS输入值由NSS PAD决定 1：NSS输入值由SPI_CTL0寄存器的NSSI位决定
25	CKPL	时钟信号极性选择 0：SPI空闲时，CLK引脚拉低 1：SPI空闲时，CLK引脚拉高
24	CKPH	时钟信号相位选择 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
23	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。
22	MSTMOD	主机模式使能 0：从机模式 1：主机模式
21	TMOD	SPI TI模式使能 0：SPI TI模式禁止 1：SPI TI模式使能
20:19	保留	必须保持复位值。
18	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
17	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

16	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
15	SWPMIO	MOSI与MISO引脚交换 0：不交换 1：交换 该位置1，MISO与MISO引脚复用功能交换。
14:8	保留	必须保持复位值。
7:4	MDFD[3:0]	SPI主机模式时，数据帧之间延时 0000：无延时 0001：1 clock延时 …. 1111：15 clock延时 该位在SPI TI模式下没有意义。
3:0	MSSD[3:0]	SPI主机模式时，NSS有效沿与数据开始传输或接收之间的延时 0000：无延时 0001：1 clock延时 …. 1111：15 clock延时 该位在SPI TI模式下没有意义。

27.5.5. 中断寄存器（SPI_INT）

地址偏移：0x10
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					TXSERFIE	CONFEIE	FEIE	CRCERIE	RXOREIE	TXUREIE	TXFIE	ESTCIE	DPIE	TPIE	RPIE
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:11	保留	必须保持复位值。
10	TXSERFIE	TXSER重载中断使能 0：TXSER中断禁止 1：TXSER中断使能
9	CONFEIE	SPI配置错误中断使能

0：SPI配置错误中断禁止
1：SPI配置错误中断使能

8	FEIE	TI帧错误中断使能 0：TI帧错误中断禁止 1：TI帧错误中断使能
7	CRCERIE	CRC错误中断使能 0：CRC错误中断禁止 1：CRC错误中断使能
6	RXOREIE	上溢错误中断使能 0：上溢中断禁止 1：上溢中断使能
5	TXUREIE	下溢错误中断使能 0：下溢中断禁止 1：下溢中断使能
4	TXFIE	传输已填充中断使能 0：TXF中断禁止 1：TXF中断使能
3	ESTCIE	传输结束、挂起、TxFIFO清空中断使能 0：ESTC中断禁止 1：ESTC中断使能
2	DPIE	DP中断使能 0：DP中断禁止 1：DP中断使能 该位由软件置1，当TXF位置1时清除。
1	TPIE	TP中断使能 0：TP中断禁止 1：TP中断使能 该位由软件置1，当TXF位置1时清除。
0	RPIE	RP中断使能 0：RP中断禁止 1：RP中断使能

27.5.6. 状态寄存器（SPI_STAT）

地址偏移：0x14
复位值：0x0000 1002

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
CTXSIZE[15:0]															

r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RWNE	RPLVL[1:0]		TC	SPD	TXSERF	CONFERR	FERR	CRCERR	RXORERR	TXURERR	TXF	ET	DP	TP	RP
r	r		r	r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:16	CTXSIZE[15:0]	TXSIZE（在SPI_CTL1寄存器中）区域中剩余的数据帧数。 当总线上有数据传输时，这个值不是很可靠。
15	RWNE	RxFIFO中的数据字长非空 0：RxFIFO中包含的数据量少于一个字长 1：RxFIFO中包含的数据量至少达到一个字长
14:13	RPLVL[1:0]	RxFIFO数据包级别 这些位定义了RxFIFO最后32位的字区中存储的数据帧的数量。 如果数据帧大小<=8位（DZ[4:0] <= 7）： 00：有0个（RWNE = 0）或4的倍数（RWNE = 1）个数据帧存储在RXFIFO中 01：有1个数据帧存储在RxFIFO中（RWNE = 0） 10：有2个数据帧存储在RxFIFO中（RWNE = 0） 11：有3个数据帧存储在RxFIFO中（RWNE = 0） 如果数据帧大小>8位并且<=16位（7< DZ[4:0] <= 15）： 00：有0个（RWNE = 0）或2的倍数（RWNE = 1）个数据帧存储在RXFIFO中 01：有1个数据帧存储在RxFIFO中（RWNE = 0） 其他：不用。 如果数据帧大小>16位（DZ[4:0] > 15）： 00：只读 其他：不用。
12	TC	TxFIFO传输完成标志 0：有数据保存在TxFIFO中，或者TxFIFO正进行最后一帧数据的传输（包含CRC） 1：最后一个数据帧或CRC帧已发送结束
11	SPD	挂起标志 0：SPI未挂起 1：SPI主模式被挂起
10	TXSERF	额外的SPI数据已被重载 0：未接收数据 1：已接收额外的数据量，传输继续进行
9	CONFERR	SPI配置错误 0：无配置错误 1：配置错误发生
8	FERR	SPI TI格式错误 0：无TI格式错误

1：TI格式错误发生

7	CRCERR	SPI CRC错误 0：无CRC错误 1：CRC错误发生
6	RXORERR	接收上溢错误 0：无接收上溢错误 1：接收上溢错误发生
5	TXURERR	传输下溢错误 0：无传输下溢错误 1：传输下溢错误发生
4	TXF	TxFIFO传输已被填充 0：TxFIFO数据上传正在进行中或未启动 1：TxFIFO数据上传已完成
3	ET	传输 / 接收结束标志 0：传输 / 接收正在进行中或未启动 1：传输 / 接收完成
2	DP	双工数据包 0：TxFIFO已满和 / 或RxFIFO已空 1：TxFIFO有空间可用于写一个完整的数据包（TP = 1），并且RxFIFO有至少一个数据包可读（RP = 1）
1	TP	TxFIFO数据包空间有效标志 0：TxFIFO没有足够的空间去接收下个数据包 1：TxFIFO有足够的空间去接收下个数据包
0	RP	RxFIFO数据包空间有效标志 0：RxFIFO已空或接收的数据包不完整（达不到FIFOLVL） 1：RxFIFO至少包含一个完整的数据包

27.5.7. 中断/状态标志清除寄存器（SPI_STATC）

地址偏移：0x18
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				SPDC	TXSERFC	CONFERRC	FERRC	CRCERRC	RXORERRC	TXURERRC	TXFC	ETC	保留		
				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	SPDC	清除挂起标志 对该位写1可以清除SPI_STAT寄存器的SPD位。
10	TXSERFC	清除TXSERF标志 对该位写1可以清除SPI_STAT寄存器的TXSERF位。
9	CONFERRC	清除配置错误标志 对该位写1可以清除SPI_STAT寄存器的CONFERR位。
8	FERRC	清除SPI TI格式错误标志 对该位写1可以清除SPI_STAT寄存器的FERR位。
7	CRCERRC	清除CRC错误标志 对该位写1可以清除SPI_STAT寄存器的CRCERR位。
6	RXORERRC	清除接收上溢错误标志 对该位写1可以清除SPI_STAT寄存器的RXORERR位。
5	TXURERRC	清除传输下溢错误标志 对该位写1可以清除SPI_STAT寄存器的TXURERR位。
4	TXFC	清除TxFIFO传输填充标志 对该位写1可以清除SPI_STAT寄存器的TXF位。
3	ETC	清除传输/接收结束标志 对该位写1可以清除SPI_STAT寄存器的ET位。
2:0	保留	必须保持复位值。

27.5.8. 数据发送寄存器（SPI_TDATA）

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

位/位域	名称	描述
31:0	TDATA[31:0]	数据发送寄存器 硬件有两个FIFO，包括TxFIFO和RxFIFO。TDATA寄存器充当TxFIFO的接口。当数

据写入TDATA后会将数据保存到TxFIFO。数据始终右对齐，根据WORDEN，BYTEN，DZ放置数据。例如：如果WORDEN置1，DZ为8位时，TDATA[7:0]为data 0，TDATA[15:8]为data 1，TDATA[23:16]为data 2，TDATA[31:24]为data 3。如果WORDEN置0，BYTEN置0，DZ为8位时，TDATA[7:0]为data 0，TDATA[15:8]为data 1，TDATA[31:16]数据无效。如果WORDEN置0，BYTEN置1，DZ为8位时，TDATA[7:0]为data 0，TDATA[31:8]数据无效。如果DZ大于8位，只能按字或半字访问。

27.5.9. 数据接收寄存器（SPI_RDATA）

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

位/位域	名称	描述
31:0	RDATA[31:0]	数据接收寄存器 硬件有两个FIFO，包括TxFIFO和RxFIFO。SPI_RDATA寄存器充当RxFIFO的接口。读取SPI_RDATA的值将从RxFIFO中获取数据。数据始终右对齐，根据WORDEN，BYTEN，DZ放置数据。例如：如果WORDEN置1，DZ为8位时，RDATA[7:0]为data 3，RDATA[15:8]为data 2，RDATA[23:16]为data 1，RDATA[31:24]为data 0。如果WORDEN置0，BYTEN置0，DZ为8位时，RDATA[15:0]数据无效，RDATA[23:16]为data 1，RDATA[31:24]为data 0。如果WORDEN置0，BYTEN置1，DZ为8位时，RDATA[23:0]数据无效，RDATA[31:24]为data 0。如果DZ大于8位，只能按字或半字访问。

27.5.10. CRC 多项式寄存器（SPI_CRCPOLY）

地址偏移：0x40

复位值：0x0000 0107

该寄存器可以按字（32位）访问。当SPI使能后，该寄存器被写保护。

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

位/位域	名称	描述
31:0	CRCPOLY[31:0]	CRC多项式寄存器 该寄存器包含CRC多项式，用于CRC计算。 默认值0x107对应DZ为8位设置。它与多项式字符串长度固定的某些其他GD产品使用的默认值0x07兼容。 多项式的长度由存储在该寄存器中的值的最高有效位决定。必须将其设置为大于DZ的值。此外，如果DZ = 32位，还必须将CRCFS位置1，以使多项式长度大于数据大小。 如果DZ = 16位，SPI_CRCPOLY寄存器的位16 - 31保留。对该寄存器进行32位访问时，位16 - 31始终读为零，写入无效。

27.5.11. 发送 CRC 寄存器（SPI_TCRC）

地址偏移：0x44
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
TCRC[31:16]															
r															

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
31:0	TCRC[31:0]	发送CRC寄存器。 当设置SPI_CFG0的CRCEN位时，硬件计算传输字节的CRC值，并将它们保存在SPI_TCRC寄存器中。 这些位在I2S模式下无意义。

27.5.12. 接收 CRC 寄存器（SPI_RCRC）

地址偏移：0x48
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
RCRC[31:16]															
r															

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
31:0	RCRC[31:0]	接收CRC寄存器。

当设置SPI_CFG0的CRCEN位时，硬件计算接收字节的CRC值并将其保存在SPI_RCRC寄存器中。
这些位在I2S模式下无意义。

27.5.13. 下溢数据寄存器（SPI_URDATA）

地址偏移：0x4C
复位值：0x0000 0000

该寄存器可以按字（32位）访问。当SPI使能后，该寄存器被写保护。

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

位/位域	名称	描述
31:0	URDATA[31:0]	从机模式传输下溢数据。 该寄存器仅在从机模式和下溢条件下被考虑。所考虑的位数取决于SPI_CFG0寄存器DZ位设置。下溢状态的处理取决于SPI_CFG0寄存器的TXURDT和TXUROP位。

27.5.14. I2S 控制寄存器（SPI_I2SCTL）

地址偏移：0x50
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
I2SCH	保留					MCKOEN	OF	DIV[7:0]							
r						rw	rw	rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				I2SSEL	I2SEN	I2SOPMOD[1:0]		PCMSMOD	保留	I2SSTD[1:0]		CKPL	DTLEN[1:0]		CHLEN
				rw	rw	rw		rw		rw		rw	rw		rw

位/位域	名称	描述
31	I2SCH	I2S通道标志 0：下一个将要发送或当前接收的数据属于左通道 1：下一个将要发送或当前接收的数据属于右通道 该位由硬件置位和清除。 SPI模式下该位无用，I2S PCM模式下该位无意义。 对于TX来说，只有FIFOLVL = 15，即TxFIFO只用于1个data的收发时，该bit有意义。 对于RX来说，只有FIFOLVL = 0，即RxFIFO只用于1个data的收发时，该bit有意

		义。 其他配置时，该bit无意义。
30:26	保留	必须保持复位值。
25	MCKOEN	I2S_MCK输出使能 0：I2S_MCK输出禁止 1：I2S_MCK输出使能 当I2S关闭时配置该位。SPI模式不使用该位。
24	OF	预分频器的奇系数 0：实际分频系数为DIV * 2 1：实际分频系数为DIV * 2 + 1 当I2S关闭时配置该位。SPI模式下不使用该位。
23:16	DIV[7:0]	预分频器的分频系数 实际分频系数是DIV * 2 + OF。 DIV不能为0。 当I2S关闭时配置该位。SPI模式下不使用该位。
15:12	保留	必须保持复位值。
11	I2SSEL	I2S模式选择 0：SPI模式 1：I2S模式 当SPI或I2S关闭时配置该位。
10	I2SEN	I2S使能 0：I2S禁止 1：I2S使能 SPI模式不使用该位。
9:8	I2SOPMOD[1:0]	I2S运行模式 00：从机发送模式 01：从机接收模式 10：主机发送模式 11：主机接收模式 当I2S关闭时配置该位。SPI模式不使用该位。
7	PCMSMOD	PCM帧同步模式 0：短帧同步 1：长帧同步 只有在PCM标准下，该位才有意义。 当I2S关闭时配置该位。SPI模式不使用该位。
6	保留	必须保持复位值。
5:4	I2SSTD[1:0]	I2S标准选择 00：I2S飞利浦标准 01：MSB对齐标准

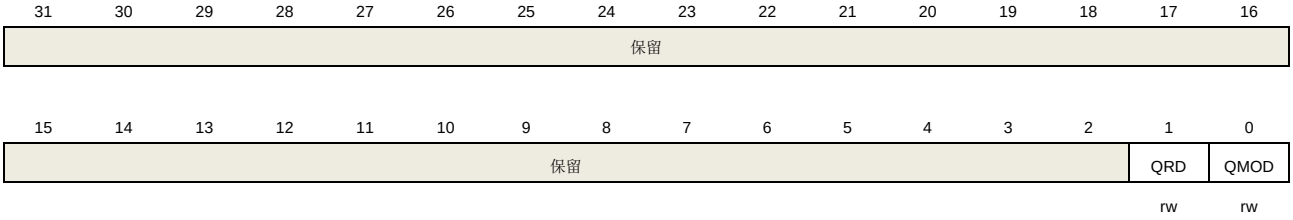
		10：LSB对齐标准 11：PCM标准 当I2S关闭时配置该位。SPI模式不使用该位。
3	CKPL	空闲状态时钟极性 0：I2S_CK空闲状态为低电平 1：I2S_CK空闲状态为高电平 当I2S关闭时配置该位。SPI模式不使用该位。
2:1	DTLEN[1:0]	数据长度 00：16位 01：24位 10：32位 11：保留 当I2S关闭时配置该位。SPI模式不使用该位。
0	CHLEN	通道长度 0：16位 1：32位 通道长度必须大于或等于数据长度。 当I2S关闭时配置该位。SPI模式不使用该位。

27.5.15. 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（32位）访问。



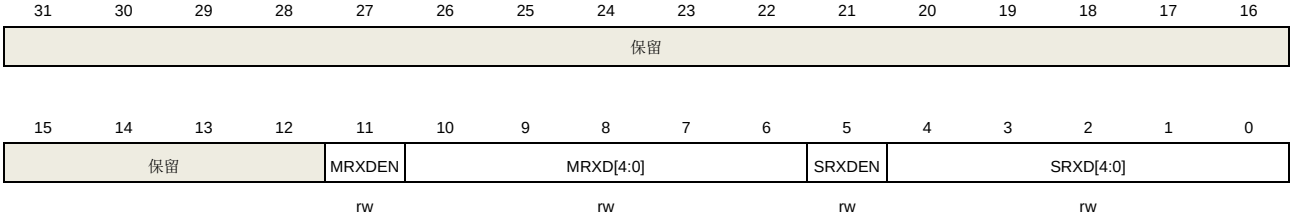
位/位域	名称	描述
31:2	保留	必须保持复位值。
1	QRD	四线SPI模式读选择 0：SPI四线模式写操作 1：SPI四线模式读操作 该位仅能在SPI未通信时配置。 该位仅适用于SPI3 / 4。
0	QMOD	四线SPI模式使能 0：SPI工作在单线模式 1：SPI工作在四线模式

该位仅能在SPI未通信时配置
该位仅适用于SPI3 / 4。

27.5.16. 接收时钟延迟寄存器（SPI_RXDLYCK）

地址偏移：0xFC
复位值：0x0000 0000

该寄存器可以按字（32位）访问。



位/位域	名称	描述
31:12	保留	必须保持复位值。
11	MRXDEN	当主机接收时，采样时钟延迟使能 0：采样时钟延迟使能 1：采样时钟延迟禁止
10:6	MRXD[4:0]	当主机接收时，采样时钟延迟时钟单元 00000：延迟1个时钟单元 00001：延迟2个时钟单元 ….. 11111：延迟32个时钟单元
5	SRXDEN	当从机接收时，采样时钟延迟使能 0：采样时钟延迟使能 1：采样时钟延迟禁止
4:0	SRXD[4:0]	当从机接收时，采样时钟延迟时钟单元 00000：延迟1个时钟单元 00001：延迟2个时钟单元 ….. 11111：延迟32个时钟单元

28. OSPI I/O 管理器（OSPIM）

28.1. 简介

OSPIM 支持对 OSPI 全 IO 矩阵引脚进行分配。

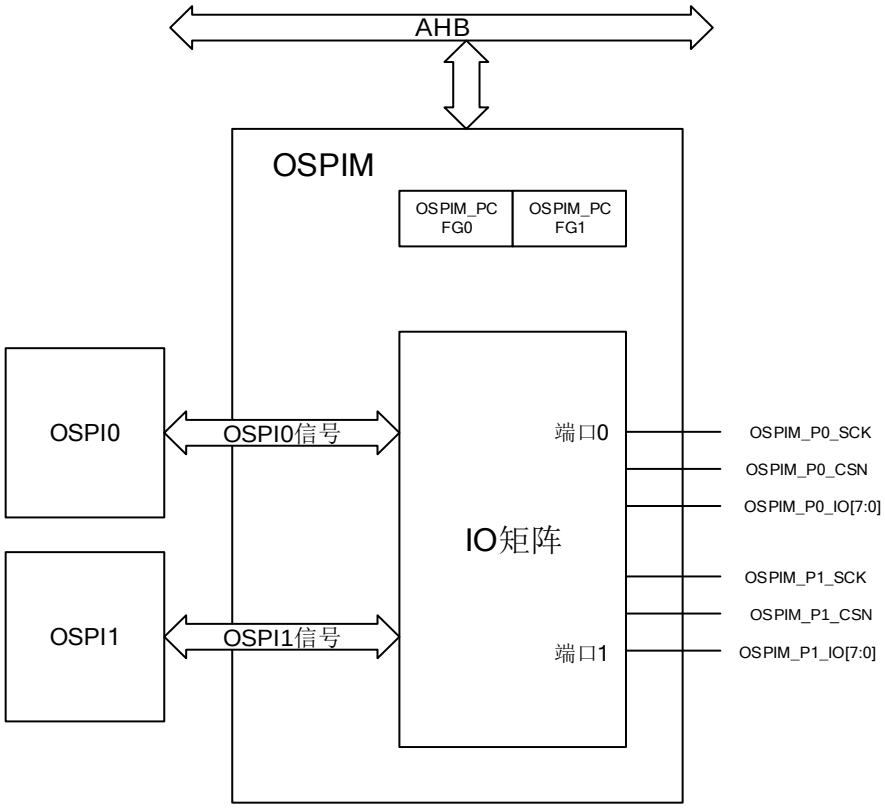
28.2. 主要特征

- 支持两个 SPI（单线，双线，四线或八线）接口。
- 支持两个端口进行引脚分配。
- 完全可编程 IO 矩阵，可按功能对引脚进行分配。

注意：对于 GD32H75E，OSPI1 和 OSPIM 端口 1 仅支持芯片内部 ESC 使用，不可用于外部器件。

28.3. 功能说明

28.3.1. OSPIM 结构框图



28.3.2. OSPIM 矩阵

OSPIM 矩阵完全可编程，可按功能对引脚进行预映射，如*表 28-1. OSPIM 矩阵映射*所示：

表 28-1. OSPIM 矩阵映射

引脚	映射
OSPIM_P0_SCK, OSPIM_P1_SCK	可独立映射到 OSPI0_SCK 或 OSPI1_SCK
OSPIM_P0_CSN, OSPIM_P1_CSN	可独立映射到 OSPI0_CSN 或 OSPI1_CSN
OSPIM_P0_IO[3:0], OSPIM_P0_IO[7:4], OSPIM_P1_IO[3:0], OSPIM_P1_IO[7:4]	可独立映射到 OSPIM0_IO[3:0], OSPIM0_IO[7:4], OSPIM1_IO[3:0]或 OSPIM1_IO[7:4]

默认情况下，OSPI0 和 OSPI1 的信号分别映射到端口 0 和端口 1。OSPIM 的端口 0 和端口 1 可分别通过 OSPIM_PCFGx 寄存器进行独立配置。若 OSPI 被禁用，OSPIM 矩阵必须被配置，防止总线中出现意外事务。

28.4. OSPIM 寄存器

OSPIM基地址：0x5200 B400

28.4.1. 端口配置寄存器（OSPIM_PCFGx）（x = 0, 1）

地址偏移：0x04*（x+1）
复位值：0x0301 0111（x = 0），0x0705 0333（x = 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
保留					SRCPHIO[1:0]		POHEN	保留					SRCPLIO[1:0]		POLEN
					rw		rw						rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						SRCPCS	NCSEN	保留						SRCPCK	SCKEN
						rw	rw							rw	rw

位/位域	名称	描述
31:27	保留	必须保持复位值。
26:25	SRCPHIO[1:0]	端口x的IO[7:4]源选择 00：选择OSPI0_IO[3:0]。 01：选择OSPI0_IO[7:4]。 10：选择OSPI1_IO[3:0]。 11：选择OSPI1_IO[7:4]。
24	POHEN	使能端口x的IO[7:4]

		0：禁用端口x的IO[7:4]。 1：使能端口x的IO[7:4]。
23:19	保留	必须保持复位值。
18:17	SRCPLIO[1:0]	端口x的IO[3:0]源选择 00：选择OSPI0_IO[3:0]。 01：选择OSPI0_IO[7:4]。 10：选择OSPI1_IO[3:0]。 11：选择OSPI1_IO[7:4]。
16	POLEN	使能端口x的IO[3:0] 0：禁用端口x的IO[3:0]。 1：使能端口x的IO[3:0]。
15:10	保留	必须保持复位值。
9	SRCPCS	端口 x 的 CSN 源选择 0：CSN 源为 OSPI0_CSN。 1：CSN 源为 OSPI1_CSN。
8	NCSEN	端口x的CSN使能 0：禁用端口x的CSN。 1：使能端口x的CSN。
7:2	保留	必须保持复位值。
1	SRCPCK	端口 x 的 SCK 源选择 0：SCK 源为 OSPI0_SCK。 1：SCK源为OSPI1_SCK。
0	SCKEN	端口x的SCK使能 0：禁用端口x的SCK。 1：使能端口x的SCK。

29. 八线 SPI 接口（OSPI）

29.1. 简介

OSPI是一种专用于和外部存储器通信的接口，支持单线，双线，四线和八线SPI模式。

29.2. 主要特征

- 三种模式：
  - 间接模式：使用OSPI的寄存器执行所有操作。
  - 状态轮询模式：周期性读取并检测外部存储器的状态寄存器值。
  - 内存映射模式：外部存储器映射到MCU地址空间（OSPI0：0x9000 0000 - 0x9FFF FFFF，OSPI1：0x7000 0000- 0x 7FFF FFFF），和访问内部存储空间一样访问存储器。
- 支持内存映射模式下读；
- 支持单线、双线、四线和八线通信；
- 可用于间接模式和内存映射模式的完全可编程命令格式；
- 支持SDR（单倍数据速率）和DTR（双倍传输速率，仅支持读GD25LX512ME）；
- 接收和发送的FIFO功能；
- 允许8位、16位或32位数据访问；
- 间接模式支持DMA操作；
- 中断：FIFO达到阈值，状态匹配、传输完成、访问错误中断。

注意：对于GD32H75E，OSPI1仅支持芯片内部ESC使用，不可用于外部器件。

29.3. OSPI 功能描述

29.3.1. OSPI 结构框图

OSPI 引脚在下表中描述。

表 29-1. OSPI 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。 八线模式：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。

引脚	方向	描述
		八线模式：数据输入或输出。
IO2	I/O	单线模式：输出模式并强制置“0”，连接外部存储器的 WP 引脚，控制“写保护”功能。 双线模式：输出模式并强制置“0”，连接外部存储器的 WP 引脚，控制“写保护”功能。 四线模式：数据输入或输出。 八线模式：数据输入或输出。
IO3	I/O	单线模式：输出模式并强制置“1”，连接外部存储器的 HOLD 引脚，控制“保持”功能。 双线模式：输出模式并强制置“1”，连接外部存储器的 HOLD 引脚，控制“保持”功能。 四线模式：数据输入或输出。 八线模式：数据输入或输出。
IO4	I/O	单线模式：输出模式并强制置“0”。 双线模式：输出模式并强制置“0”。 四线模式：输出模式并强制置“0”。 八线模式：数据输入或输出。
IO5	I/O	单线模式：输出模式并强制置“0”。 双线模式：输出模式并强制置“0”。 四线模式：输出模式并强制置“0”。 八线模式：数据输入或输出。
IO6	I/O	单线模式：输出模式并强制置“0”。 双线模式：输出模式并强制置“0”。 四线模式：输出模式并强制置“0”。 八线模式：数据输入或输出。
IO7	I/O	单线模式：输出模式并强制置“0”。 双线模式：输出模式并强制置“0”。 四线模式：输出模式并强制置“0”。 八线模式：数据输入或输出。

图 29-1. OSPI 八线通信模式结构框图

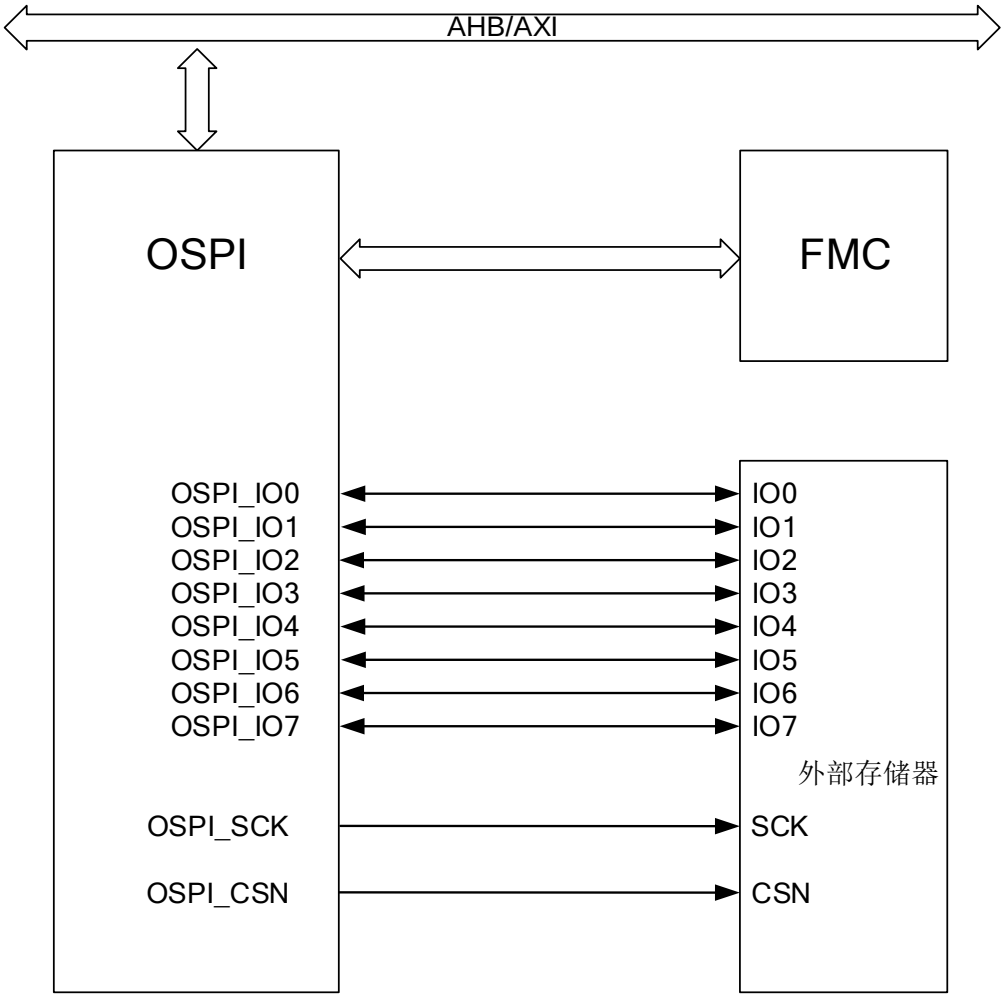
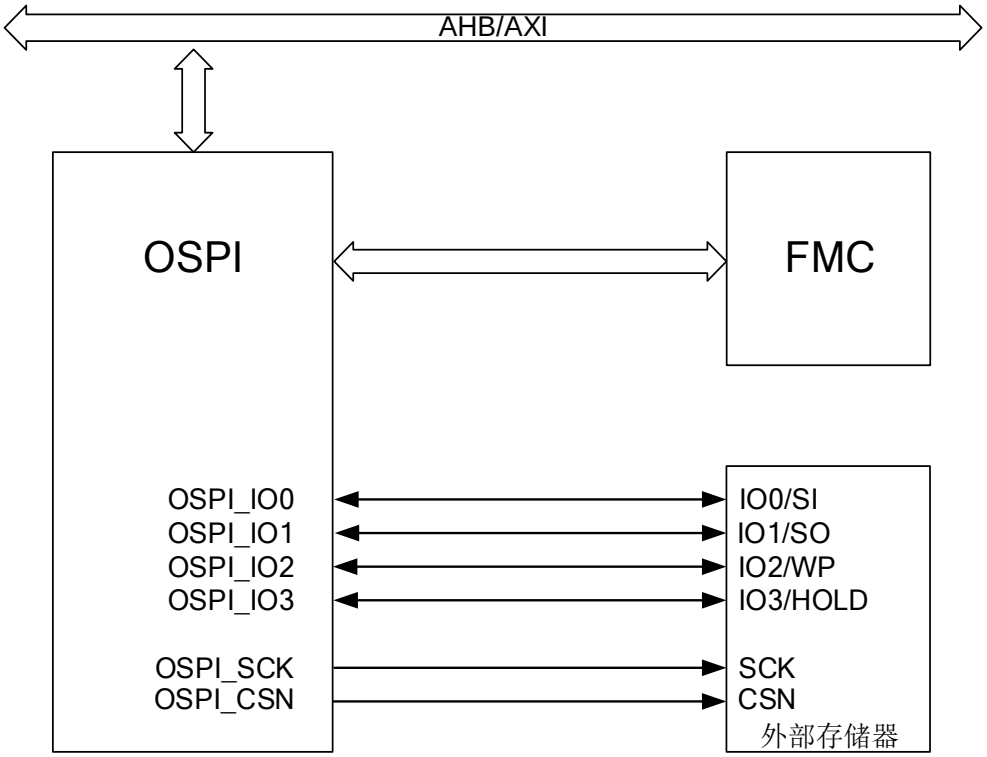


图 29-2. OSPI 四线通信模式结构框图



29.3.2. OSPI 常规命令模式

常规命令模式下，每个命令最多有五个阶段：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 29-3. 八线模式 OSPI 命令格式

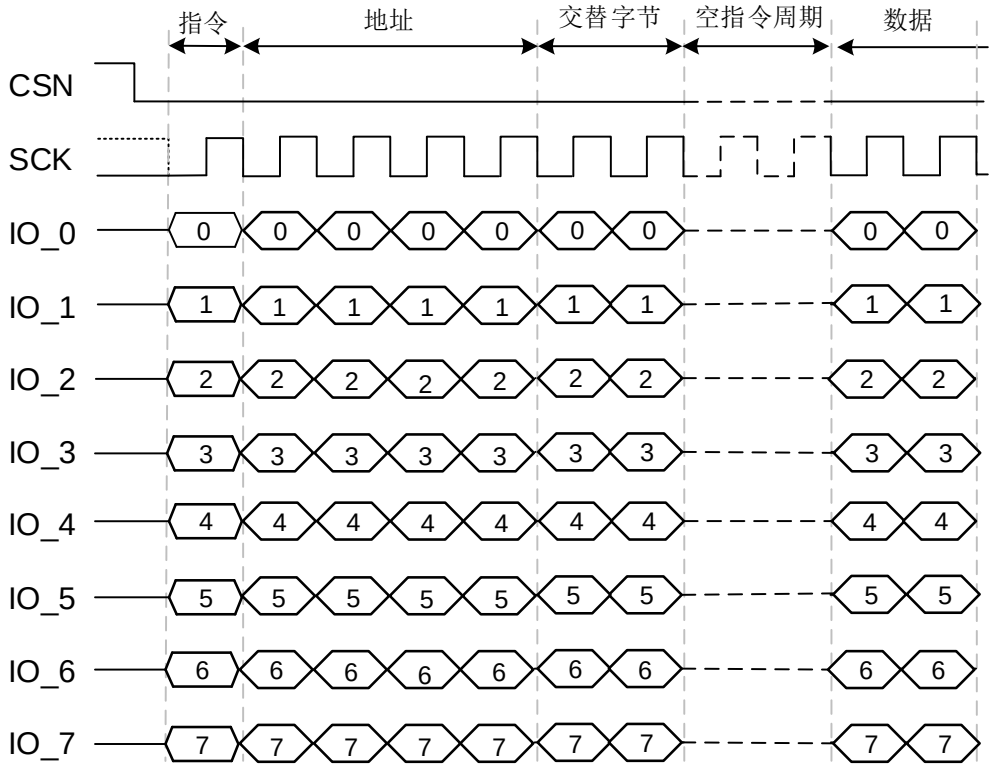
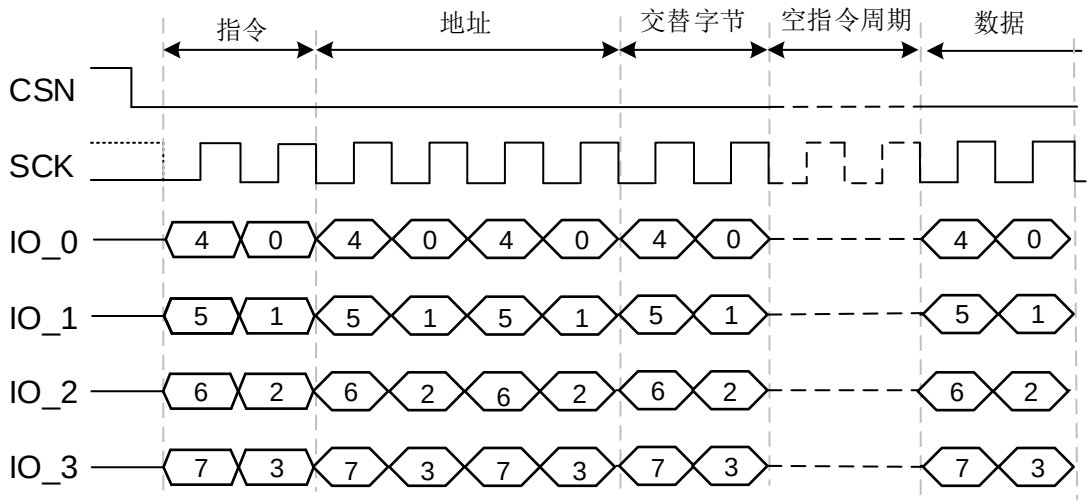


图 29-4. 四线模式 OSPI 命令格式



指令阶段

指令阶段，在OSPI_TCFG寄存器的INSSZ[1:0]位域配置需要发送的指令大小（8位，16位，24位或32位），在OSPI_INS寄存器中配置待发送给外部存储器的指令。OSPI_TCFG寄存器的IMOD[2:0]位域定义了指令阶段的模式（无指令，1线，2线，4线或者8线）。

当OSPI_CTL寄存器的FMOD[1:0]位域为0b11，则OSPI工作在内存映射模式下。在内存映射模式下，在OSPI_WINS寄存器中定义写入操作的指令，在OSPI_WTCFG寄存器配置指令的格式（大小，模式）。在OSPI_INS寄存器和OSPI_TCFG寄存器中配置读取操作指令以及指令的格

式。

地址阶段

地址阶段，在OSPI_TCFG寄存器的ADDRSZ[1:0]位域中定义待发送的地址大小（8位，16位，24位或32位），在OSPI_TCFG寄存器的ADDRMOD[2:0]位域中定义地址阶段的模式（无地址，1线，2线，4线或者8线）。将OSPI_TCFG寄存器的ADDRDTR位置1可使能DTR模式，则地址在时钟的每个边沿被发送。在间接模式和自动轮询模式下，在OSPI_ADDR寄存器中定义待发送的地址信息。

在内存映射模式下，待发送的地址由AXI（内核或者DMA）直接给出。在OSPI_WTCFG寄存器配置写入操作的地址格式（大小，模式以及是否开启DTR）。在OSPI_TCFG寄存器中配置读取操作的地址格式。

交替字节阶段

交替字节阶段，在OSPI_TCFG寄存器的ALTESZ[1:0]位域配置需要发送的交替字节大小（8位，16位，24位或32位），在OSPI_ALTE寄存器中配置待发送给外部存储器的交替字节。OSPI_TCFG寄存器的ALTEMOD[2:0]位域定义了指令阶段的模式（无交替字节，1线，2线，4线或者8线）。将OSPI_TCFG寄存器的ALTEDTR位置1可使能DTR模式，则交替字节在时钟的每个边沿被发送。

在内存映射模式下，在OSPI_WALTE寄存器中定义写入操作的交替字节，在OSPI_WTCFG寄存器配置交替字节的格式（大小，模式以及是否开启DTR）。在OSPI_ALTE寄存器和OSPI_TCFG寄存器中配置读取操作指令以及指令的格式。

空指令阶段

空指令阶段，在OSPI_TIMCFG寄存器的DUMYC[4:0]位域指定空指令的周期数（0到31个），这期间OSPI与外部存储器没有数据交互，为了等待外部存储器准备数据。

在内存映射模式下，在OSPI_WTIMCFG寄存器的DUMYC[4:0]位域指定写入操作的的空指令周期数，而读取操作的空指令周期数在OSPI_TIMCFG寄存器的DUMYC[4:0]位域指定。

注意：

1. 当OSPI工作在2线、4线或者8线模式下从外部存储器中接收数据时，至少设置一个空指令周期。

数据阶段

数据阶段，任意数量的字节可以在外部存储器和OSPI之间传输。OSPI_TCFG寄存器的DATAMOD[2:0]位域定义了数据阶段的模式（无数据，1线，2线，4线或者8线，其中无数据仅可用于间接写入模式）。将OSPI_TCFG寄存器的DADTR位置1可使能DTR模式，则数据在时钟的每个边沿被发送或接收。在间接模式下，OSPI_DTLEN寄存器定义了待接收或者待发送数据的字节数。在写操作时，需要发送的数据写入OSPI_DATA寄存器，在读操作时，接收数据从OSPI_DATA寄存器中获取。在自动轮询模式下，OSPI_DTLEN寄存器定义了待接收的字节数，接收数据从OSPI_DATA寄存器中获取。

在内存映射模式下，数据的传输通过AXI直接发送或接收自内核或DMA。传输的字节个数确定

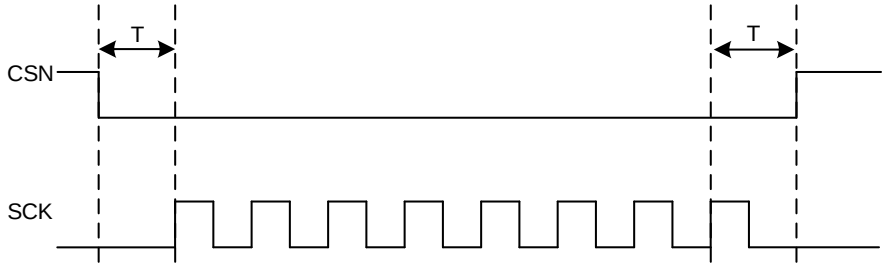
AXI总线的访问操作，可以8位，16位或者32位读写访问，相应传输1个，2个，4个字节。在OSPI_WTCFG寄存器配置发送数据的格式（模式以及是否开启DTR）。在OSPI_TCFG寄存器中配置读取数据的格式。

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信号输出的一个闸信号，内部SCK信号是一直存在的。

CSN在第一个SCK有效上升沿的之前一个SCK时钟周期拉低，在最后一个SCK有效上升沿之后一个SCK时钟周期拉高。如*图29-5. CSN和SCK的行为*所示：

图 29-5. CSN 和 SCK 的行为



29.4. 操作模式

29.4.1. 间接模式

在间接模式写操作时，要发送的数据写入到OSPI_DATA寄存器，读操作时，接收的数据从OSPI_DATA寄存器读取。

OSPI_DTLEN寄存器定义了需要传输的字节数。如果DTLEN = 0XFFFF_FFFF，数据的字节数被认为没有定义，传输会一直持续到MESZ定义的存储器大小的边界。如果DTLEN = 0XFFFF_FFFF并且MESZ = 0x1F，传输会一直持续到OSPI关闭。

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。在没有定义传输长度情况下，在接受或者发送的字节数达到存储器大小时，TC会被置1。

如果TCIE和TC都被置1，则会产生中断，可以通过写TCC位为1清除。

常规模式下触发命令序列

命令序列在根据通信需求配置好最后信息之后立即开始。命令开始后，BUSY位置1。

当没有地址并且没有数据时，在访问OSPI_INS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OSPI_ADDR寄存器之后立即开始命令序列。

当在间接模式写操作时需要地址并且有数据，在访问OSPI_DATA寄存器之后立即开始命令序列。

FIFO 和标志控制

32字节的FIFO用于传输数据。在间接模式写操作时，32位访问写入4字节，16位访问写入2字节，8位访问写入1个字节。

FIFO阈值由FTL定义，在间接模式读操作时，FIFO中的字节数等于或者超过定义的阈值时，FIFO阈值标志FT会被置1。在数据阶段完成后如果FIFO不为空，FT也会被置1。在间接模式写操作时，FIFO空的字节数超过阈值，FT会被置1。

如果FTIE和FT都被置1，会产生中断。如果OSPI DMA使能，DMA请求由FT产生，直到标志清除。

在间接模式读操作时，当FIFO变为满时，OSPI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个字节为空。

29.4.2. 状态轮询模式

在状态轮询模式时，OSPI周期性的开始读命令，每帧最多读取4字节的数据，如果OSPI_DTLEN寄存器中定义的数据长度大于4，则忽略多余的长度仅读取4字节。接收的数据会按位屏蔽，并且与定义的数据内容比较，如果一个匹配发生，当SMIE置位时，会产生一个中断。

状态轮询访问和间接模式读序列一样，在周期性间隔时BUSY位保持高电平。

轮询匹配模式位SPMOD控制比较匹配的模式，如果SPMOD=0，与模式被选择。该模式下，只有在所有非屏蔽位都匹配时，状态匹配标志SM被置位。如果SPMOD=1，或模式被选择，该模式下，任何非屏蔽位只要有一位匹配，状态匹配标志SM被置位。

如果状态轮询停止位SPS被置位，当一个匹配被检测到，状态轮询停止，在数据阶段结束时BUSY位会被清除。

在状态轮询模式下，FIFO是禁用的，读状态的字节都存储在DATA寄存器中，存储的状态字节不会被MASK控制域影响。DATA寄存器的内容在数据阶段开始时会被更新。

如果FT位在数据阶段结束时被置位，这时候表示外部寄存器的状态字节被读取，当DATA寄存器被读取时，该位被清除。

在状态轮询模式下，必须将外部存储器配置为固定延迟模式。

29.4.3. 内存映射模式

在内存映射模式下，外部存储被当做内部存储来访问，最大访问地址为256MB，即使外部存储器大小大于256MB。

内存映射模式不允许地址超过MESZ定义的范围，即使MESZ范围在256MB范围内。如果该情况发生，AXI会产生错误。错误的影响取决于AXI主机。如果主机是CPU，会产生硬件错误中断，如果是DMA，传输错误中断产生，并且相应DMA通道会关闭。

在该模式下，字节，半字，字或者突发访问都可以支持。

本地执行（XIP）也可以支持，在完成最近一次访问后会继续将字节加载到地址。如果随后连续访问后续的字节，由于结果已经预取，这一系列访问操作将很快完成。否则，读序列会重新

开始，并且在开始之前CSN保持低电平。

FIFO为空后，OSPI进入保持状态，没有时钟输出，在此期间CSN保持低电平。在开始传输时，在CSN下降之前BUSY位会变为高电平。

29.5. OSPI 配置

29.5.1. OSPI 系统配置

OSPI系统配置的具体如下：

1. 通过FMOD[1:0]位域配置OSPI的工作模式。
2. 若OSPI工作在状态轮询模式，则需要配置SPMOD和SPS位选择轮询匹配模式和自动轮询模式的停止方式。
3. 通过配置FTL[4:0]位域，设置FIFO的阈值。
4. 若需要使用DMA，设置DMAEN位为1。在OSPI配置期间不得使能DMA通道，否则可能产生意外请求。
5. 如需使用中断，则可设置相关中断的使能位。

29.5.2. OSPI 器件配置

通过OSPI_DCFG0和OSPI_DCFG1寄存器配置OSPI与外部器件的相关参数，具体如下：

1. 通过设置DTYSEL[2:0]位域的值，配置外部存储器类型。
2. 通过设置MESZ[4:0]位域值，配置外部存储器的大小。
3. 通过设置CSHC[5:0]位域值，配置片选信号最短高电平时间。
4. 通过设置PSC[7:0]位域值，配置预分频系数。

29.5.3. OSPI 常规命令模式配置

间接模式

当OSPI工作在间接模式时，具体配置如下：

1. 通过设置OSPI_DTLEN寄存器，配置数据长度。
2. 通过设置OSPI_TIMCFG寄存器，配置帧时序。
3. 通过设置OSPI_TCFG寄存器，配置帧格式。
4. 通过设置OSPI_INS寄存器，指定要发送到外部存储器的指令。
5. 通过设置OSPI_ALTE寄存器，指定在发送地址后立即发送到外部存储器的可选交替字节。
6. 通过设置OSPI_ADDR寄存器，指定要发送到外部存储器的地址。
7. 通过OSPI_DATA寄存器读取/写入数据。

状态标志轮询模式

当 OCTALSLPI 工作在状态标识轮询模式时，具体配置如下：

1. 通过设置OSPI_STATMK寄存器，指定对接收的状态字节进行屏蔽。

2. 通过设置OSPI_STATMATCH寄存器，指定与OSPI_STATMK寄存器比较的值。
3. 通过设置OSPI_INTERVAL寄存器，指定读取操作之间的时钟周期数。
4. 通过设置OSPI_DTLEN寄存器，配置数据长度。
5. 通过设置OSPI_TIMCFG寄存器，配置帧时序。
6. 通过设置OSPI_TCFG寄存器，配置帧格式。
7. 通过设置OSPI_INS寄存器，指定要发送到外部存储器的指令。
8. 通过设置OSPI_ALTE寄存器，指定在发送地址后立即发送到外部存储器的可选交替字节。
9. 通过设置OSPI_ADDR寄存器，指定要发送到外部存储器的地址。

内存映射模式

在内存映射模式下，外部存储被当做内部存储来访问。在内存映射模式下，对 OSPI 的配置需在首次访问存储器区域之前完成。具体配置如下：

1. 通过设置OSPI_TIMCFG寄存器，配置读取操作帧时序。
2. 通过设置OSPI_TCFG寄存器，配置读取操作帧格式。
3. 通过设置OSPI_INS寄存器，指定要发送到外部存储器的指令。
4. 通过设置OSPI_ALTE寄存器，指定在发送地址后立即发送到外部存储器的可选交替字节。
5. 通过设置OSPI_WTIMCFG寄存器，配置写操作帧时序。
6. 通过设置OSPI_WTCFG寄存器，配置写操作帧格式。
7. 通过设置OSPI_WINS寄存器，指定要发送到外部存储器的指令。
8. 通过设置OSPI_WALTE寄存器，指定在发送地址后立即发送到外部存储器的可选交替字节。

29.6. OSPI 延迟数据采样

复位后，OSPI 在外部存储器驱动数据半个时钟周期之后对数据进行采样。配置 OSPI_TIMCFG 寄存器的 SSAMPLE 位可以将数据采样移位半个时钟周期。当 DADTR 置 1 时，软件必须清除 SSAMPLE 位。

29.7. 繁忙状态

OSPI开始操作外部存储器时，BUSY位置1。

在间接模式下，如果命令阶段结束且FIFO为空时，BUSY位清0。在状态标志轮询模式下，只有在发生匹配，BUSY位才会请零。

29.8. 错误管理

下列情况会产生错误：

在间接模式下，根据MESZ设置的外部存储器地址，向ADDR寄存器写入错误地址会产生TERR错误。

在间接模式下，如果ADDR寄存器的值加上DTLEN寄存器的值大于外部存储器的大小，一旦

OSPI被触发，TERR错误会产生。

在内存映射模式下，AXI超出范围的访问会返回AXI总线错误。

29.9. OSPI 中断

表 29-2. SPI 中断请求

标志	描述	清除	中断使能位
FT	FIFO阈值	硬件清除	FTIE
TC	传输完成	设置OSPI_STATC寄存器TCC位为1	TCIE
TERR	传输错误	设置 OSPI_STATC 寄存器 TERRC 位为 1	TERRIE
SM	状态匹配	设置 OSPI_STATC 寄存器 SMC 位为 1	SMIE

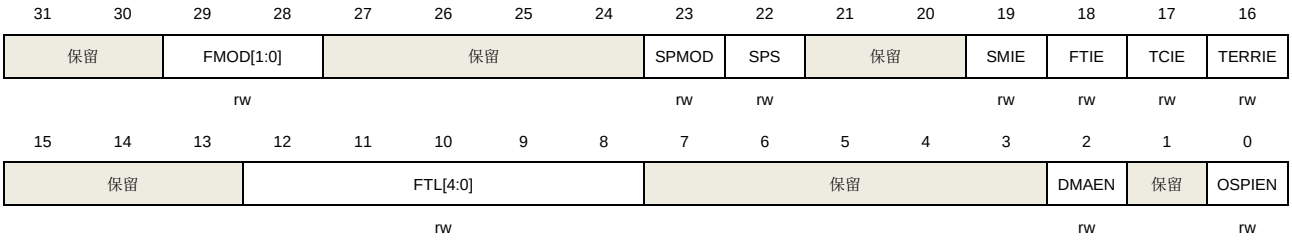
29.10. OSPI 寄存器

OSPI0基地址：0x5200 5000
OSPI1基地址：0x5200 A000

29.10.1. 控制寄存器（OSPI_CTL）

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字（32位）访问。
当BUSY位为1，该寄存器不可修改。



位/位域	名称	描述
31:30	保留	必须保持复位值。
29:28	FMOD[1:0]	功能模式 00：间接写入模式。 01：间接读取模式。 10：状态轮询模式。 11：内存映射模式。 当DMAEN位为1，在更改该位域之前必须禁止相应通道的DMA控制器。
27:24	保留	必须保持复位值。
23	SPMOD	状态轮询匹配模式 0：与模式，如果存储器返回的字节所有非屏蔽位都和匹配寄存器相应位匹配，状态匹配标志SM被置位。 1：或模式，如果存储器返回的字节任何一个非屏蔽位都和匹配寄存器相应位匹配，状态匹配标志SM被置位。
22	SPS	状态轮询模式停止 该位表明在产生匹配后停止状态轮询模式。 0：保留 1：在产生匹配后自动轮询停止。
21:20	保留	必须保持复位值。
19	SMIE	状态匹配中断使能 0：禁用状态匹配中断。 1：使能状态匹配中断。

18	FTIE	FIFO阈值中断使能 0：禁用FIFO阈值中断。 1：使能FIFO阈值中断。
17	TCIE	传输完成中断使能 0：禁用传输完成中断。 1：使能传输完成中断。
16	TERRIE	传输错误中断使能。 0：禁用传输错误中断使能。 1：使能传输错误中断使能。
15:13	保留	必须保持复位值。
12:8	FTL[4:0]	FIFO阈值等级 该位在间接模式下使用，FIFO中的字节数会触发FIFO阈值标志置位。 间接模式写操作时（FMOD = 00）： 0：如果有1个或者更多字节可以有效写入FIFO，FT置位。 1：如果有2个或者更多字节可以有效写入FIFO，FT置位。 ... 31：如果有32个字节可以有效写入FIFO，FT置位。 间接模式读操作时（FMOD = 01）： 0：如果有1个或者更多有效数据能从FIFO中读取，FT置位。 1：如果有2个或者更多有效数据能从FIFO中读取，FT置位。 ... 31：如果有32个有效数据能从FIFO中读取，FT置位。 如果DMAEN为1，在改变FTL之前，DMA控制器的相应通道必须是禁用的。
7:3	保留	必须保持复位值。
2	DMAEN	DMA使能 间接模式下，可以使用DMA通过OSPI_DATA寄存器传输数据。当FT位置1时，DMA传输开始。 0：DMA禁用。 1：DMA使能。
1	保留	必须保持复位值。
0	OSPIEN	使能OSPI 0：禁用OSPI。 1：使能OSPI。

29.10.2. 设备配置寄存器 0（OSPI_DCFG0）

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字（32位）访问。
当BUSY位为1，该寄存器不可修改。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留					DTYSEL[2:0]			保留			MESZ[4:0]				
					rw						rw				

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

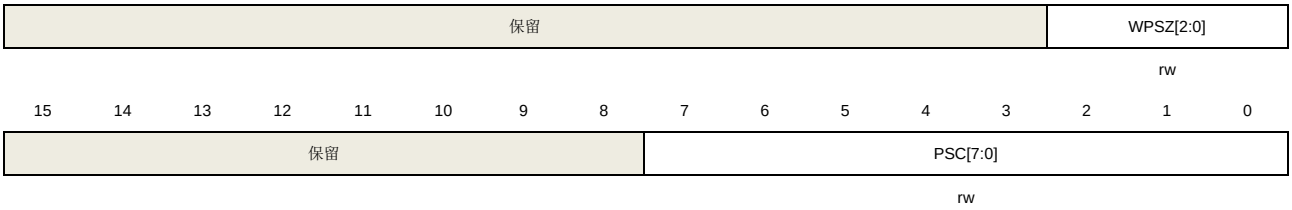
位/位域	名称	描述
31:27	保留	必须保持复位值。
26:24	DTYSEL[2:0]	选择设备类型 000：Micron模式，DTR 8数据位模式下按D0/D1排序。八线/四线/双线/单线模式下的常规SPI协议。 001：Macronix模式，DTR 8数据位模式下按D1/D0排序。八线/四线/双线/单线模式下的常规SPI协议。 010：标准模式。 011：Macronix RAM模式，DTR 8位数据模式下D1/D0排序。具有专用地址映射的八线/四线/双线/单线模式下的常规SPI协议。 其它：保留。
23:21	保留	必须保持复位值。
20:16	MESZ[4:0]	存储器大小 该位定义外部存储器大小，使用下列公式： 存储器字节数 = $2^{[MESZ+1]}$。 MESZ+1是存储器地址位数。间接模式下，存储器容量最大到4GB。在内存映射模式下，最大256MB。
15:14	保留	必须保持复位值。
13:8	CSHC[5:0]	片选高电平周期数 CSHC+1定义了在两个命令序列之间CSN保持高电平最少的SCK周期数。 0：CSN保持高电平至少1个SCK周期。 1：CSN保持高电平至少2个SCK周期。 ... 63：CSN保持高电平至少64个SCK周期。
7:0	保留	必须保持复位值。

29.10.3. 设备配置寄存器 1（OSPI_DCFG1）

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。
当BUSY位为1，该寄存器不可修改。

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



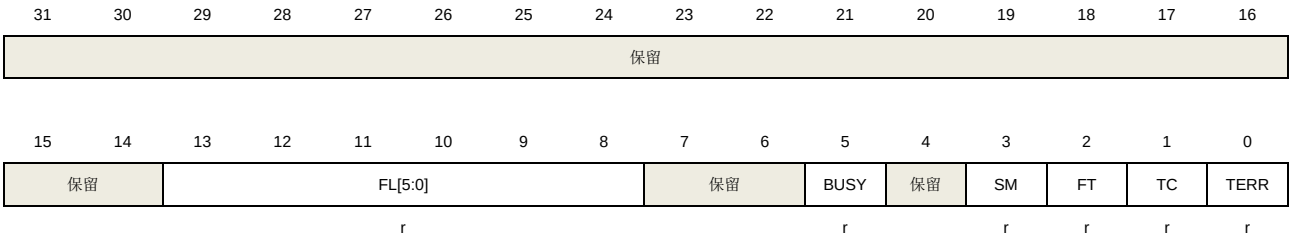
位/位域	名称	描述
31:19	保留	必须保持复位值。
18:16	WPSZ[2:0]	回卷大小 000：外部存储器设备不支持回卷读取。 001：保留。 010：外部存储器设备支持16字节回卷大小。 011：外部存储器设备支持32字节回卷大小。 100：外部存储器设备支持64字节回卷大小。 101：外部存储器设备支持128字节回卷大小。 110：保留。 111：保留。
15:8	保留	必须保持复位值。
7:0	PSC[7:0]	该位域定义了从内核时钟分频产生OSPI时钟的分频因子（位域值+1）。 0：$F_{CLK} = F_{KERNEL}$。 1：$F_{CLK} = F_{KERNEL} / 2$。 2：$F_{CLK} = F_{KERNEL} / 3$。 … 255：$F_{CLK} = F_{KERNEL} / 256$。 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。

29.10.4. 状态寄存器（OSPI_STAT）

地址偏移：0x20

复位值：0x0000 0004

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	FL[5:0]	FIFO等级 该位域给出FIFO有效字节数。在状态轮询模式下，FL为0。
7:6	保留	必须保持复位值。
5	BUSY	忙状态 该位在命令传输时置1，在对存储器一次操作完成后并且FIFO为空时清0。
4	保留	必须保持复位值。
3	SM	状态匹配标志 在状态轮询模式下，当接收到数据匹配期望值时置1，向SMC位写1清0。
2	FT	FIFO阈值标志 在间接模式下，当FIFO阈值到达或者最后读操作时FIFO非空，该位置1。 在状态轮询模式下，每次从外部存储器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 在间接模式下，当传输数据达到设置长度时，该位置1。通过对TCC位置1来清除。
0	TERR	传输错误标志 在间接模式下，当无效地址被访问时该位置1，通过对TERRC位置1来清除。

29.10.5. 状态清除寄存器（OSPI_STATC）

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												SMC	保留	TCC	TERRC
												w		w	w

位/位域	名称	描述
31:4	保留	必须保持复位值。
3	SMC	清除状态匹配标志 写1清除状态寄存器的SM标志。
2	保留	必须保持复位值。
1	TCC	清除传输完成标志 写1清除状态寄存器的TC标志。
0	TERRC	清除传输错误标志 写1清除状态寄存器的TERR标志。

29.10.6. 数据长度寄存器（OSPI_DTLEN）

地址偏移：0x40
复位值：0x0000 0000

该寄存器只能按字（32位）访问。
当BUSY位为1，该寄存器不可修改。

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

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

位/位域	名称	描述
31:0	DTLEN[31:0]	数据长度 在间接模式和状态轮询模式下数据长度为DTLEN+1，对于状态轮询模式，DTLEN的值不大于3。在间接模式下全为1表明未定义长度，OSPI会持续通信直到MESZ设定的存储器容量大小。 0x0000 0000：1个字节将要被传输。 0x0000 0001：2个字节将要被传输。 0x0000 0002：3个字节将要被传输。 0x0000 0003：4个字节将要被传输。 ... 0xFFFF FFFD：4,294,967,294 (4G-2)个字节将要被传输。 0xFFFF FFFE：4,294,967,295 (4G-1)个字节将要被传输。 0xFFFF FFFF：未定义长度 – 所有字节都会被传输直到存储器最后（由MESZ定义），如果MESZ为0x1F，无限读数据。 内存映射模式下，该位无影响。

29.10.7. 地址寄存器（OSPI_ADDR）

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
ADDR[31:16]															
rw															

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

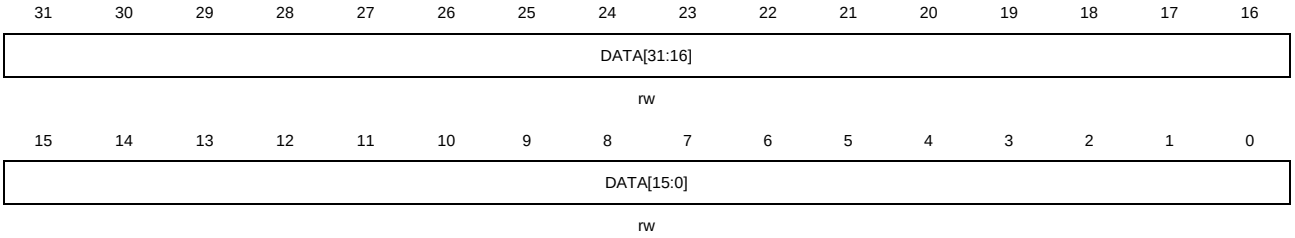
位/位域	名称	描述

31:0	ADDR[31:0]	地址 发送到存储器的访问地址。 该位域当BUSY位为0时才能写入并且内存映射模式不被配置。

29.10.8. 数据寄存器（OSPI_DATA）

地址偏移：0x50
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

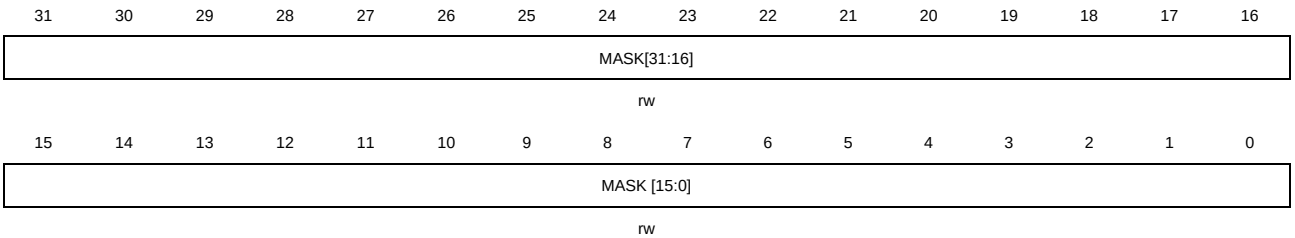


位/位域	名称	描述
31:0	DATA[31:0]	将要与存储器交互的数据。 在间接模式下写操作时，在发送到存储器之前，写入到该寄存器数据会被存储到FIFO中。如果FIFO为满，写操作会停止直到FIFO有足够空间。 在间接模式下读操作时，读该寄存器获取从存储器接收的数据。如果FIFO没有足够的字节数来满足读命令请求，并且BUSY位为1，那么读操作会被停止直到FIFO中有足够的数据或者传输已经完成。 在状态轮询模式下，该寄存器包含从读取的最后数据。

29.10.9. 状态屏蔽寄存器（OSPI_STATMK）

地址偏移：0x80
复位值：0x0000 0000

该寄存器只能按字（32位）访问。
当BUSY位为1，该寄存器不可修改。



位/位域	名称	描述
31:0	MASK[31:0]	状态屏蔽 用来屏蔽接收的状态字节。 对于第n位：

0：接收数据的第n位屏蔽，该位不参与匹配逻辑。
1：接收数据的第n位没有屏蔽，该位参与匹配逻辑。

29.10.10. 状态匹配寄存器（OSPI_STATMATCH）

地址偏移：0x88
复位值：0x0000 0000

该寄存器只能按字（32位）访问。
当BUSY位为1，该寄存器不可修改。

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]	状态匹配 与屏蔽状态寄存器比较进行匹配的值。

29.10.11. 间隔寄存器（OSPI_INTERVAL）

地址偏移：0x90
复位值：0x0000 0000

该寄存器只能按字（32位）访问。
当BUSY位为1，该寄存器不可修改。

31	30	29	28	27	26	25	24	23	22	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 状态轮询模式下两次读命令之间的SCK周期数。

29.10.12. 传输配置寄存器（OSPI_TCFG）

地址偏移：0x100
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

当BUSY位为1，该寄存器不可修改。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				DADTR	DATAMOD[2:0]			保留		ALTESZ[1:0]		ABDTR	ALTEMOD[2:0]		
				rw	rw					rw		rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		ADDRSZ[1:0]		ADDRDTR	ADDRMOD[2:0]			保留		INSSZ[1:0]		保留	IMOD[2:0]		
		rw		rw	rw					rw			rw		

位/位域	名称	描述
31:28	保留	必须保持复位值。
27	DADTR	数据双倍传输速率 0：数据阶段禁用DTR。 1：数据阶段使能DTR。 **注：**仅支持GD25LX512ME。
26:24	DATAMOD[2:0]	数据模式 该位定义数据阶段的操作模式。 000：无数据。 001：单线传输数据。 010：双线传输数据。 011：四线传输数据。 100：八线传输数据。 101：保留。 110：保留。 111：保留。
23:22	保留	必须保持复位值。
21:20	ALTESZ[1:0]	交替字节大小 该位域定义交替字节大小。 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。
19	ABDTR	交替字节双倍传输速率 0：交替字节阶段禁用DTR。 1：交替字节阶段使能DTR。 **注：**仅支持GD25LX512ME。
18:16	ALTEMOD[2:0]	交替字节模式 该位定义交替字节阶段的操作模式： 000：无交替字节。 001：单线传输交替字节。 010：双线传输交替字节。

011：四线传输交替字节。
100：八线传输交替字节。
101：保留。
110：保留。
111：保留。

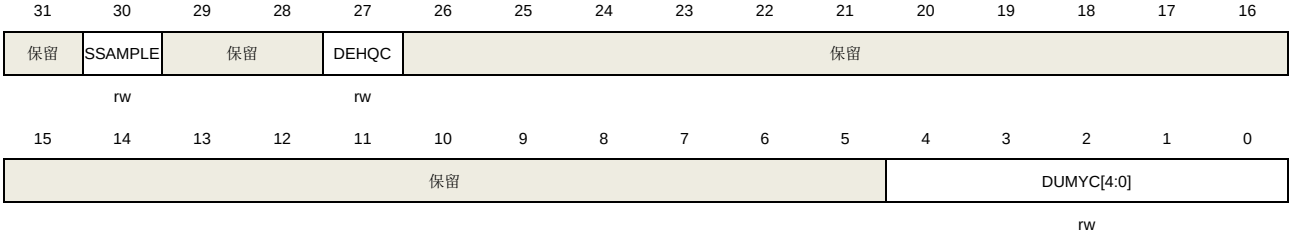
15:14	保留	必须保持复位值。
13:12	ADDRSZ[1:0]	地址大小 该位域定义地址大小。 00：8位地址。 01：16位地址。 10：24位地址。 11：32位地址。
11	ADDRDTR	地址双倍传输速率 0：地址阶段禁用DTR模式。 1：地址阶段使能DTR模式。 **注：**仅支持GD25LX512ME。
10:8	ADDRMOD[2:0]	地址模式 该位定义地址阶段的操作模式： 000：无地址。 001：单线传输地址。 010：双线传输地址。 011：四线传输地址。 100：八线传输地址。 101：保留。 110：保留。 111：保留。
7:6	保留	必须保持复位值。
5:4	INSSZ[1:0]	指令大小 该位域定义指令大小。 00：8位指令。 01：16位指令。 10：24位指令。 11：32位指令。
3	保留	必须保持复位值。
2:0	IMOD[2:0]	命令模式 该位定义指令阶段的操作模式： 000：无指令。 001：单线传输指令。 010：双线传输指令。 011：四线传输指令。

100：八线传输指令。
101：保留。
110：保留。
111：保留。

29.10.13. 时序配置寄存器（OSPI_TIMCFG）

地址偏移：0x108
复位值：0x0000 0000

该寄存器只能按字（32位）访问。
当BUSY位为1，该寄存器不可修改。



位/位域	名称	描述
31	保留	必须保持复位值。
30	SSAMPLE	采样移位 默认情况下，OSPI在外部存储器驱动数据后二分之一个SCK时钟周期采样。该位允许外部信号延迟的原因采样延迟。 0：不移位。 1：移位二分之一个周期。 **注：**当通信速率大于40M，SSAMPLE必须设置为1。
29:28	保留	必须保持复位值。
27	DEHQC	延迟保持1/4周期 0：不延迟保持。 1：延迟保持1/4周期。
26:5	保留	必须保持复位值。
4:0	DUMYC[4:0]	空指令周期数 该位域定义空指令阶段持续时间。

29.10.14. 指令寄存器（OSPI_INS）

地址偏移：0x110
复位值：0x0000 0000

该寄存器只能按字（32位）访问。
当BUSY位为1，该寄存器不可修改。



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

位/位域	名称	描述
31:0	INSTRUCTION[31:0]	指令 发送到外部存储器的命令信息。

29.10.15. 交替字节寄存器（OSPI_ALTE）

地址偏移：0x120
复位值：0x0000 0000

该寄存器只能按字（32位）访问。
当BUSY位为1，该寄存器不可修改。

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 发送给外部存储器的选项字节。

29.10.16. 回卷传输配置寄存器（OSPI_WPTCFG）

地址偏移：0x140
复位值：0x0000 0000

该寄存器只能按字（32位）访问。
当BUSY位为1，该寄存器不可修改。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				DADTR	DATAMOD[2:0]			保留		ALTESZ[1:0]		ABDTR	ALTEMOD[1:0]		
				rw	rw					rw		rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		ADDRSZ[1:0]		ADDRDTR	ADDRMOD[2:0]			保留		INSSZ[1:0]		保留	IMOD[2:0]		
		rw		rw	rw					rw			rw		

位/位域	名称	描述
31:28	保留	必须保持复位值。

27	DADTR	数据双倍传输速率 0：数据阶段禁用DTR。 1：数据阶段使能DTR。 **注：**仅支持GD25LX512ME。
26:24	DATAMOD[2:0]	数据模式 该位定义数据阶段的操作模式。 000：无数据。 001：单线传输数据。 010：双线传输数据。 011：四线传输数据。 100：八线传输数据。 101：保留。 110：保留。 111：保留。
23:22	保留	必须保持复位值。
21:20	ALTESZ[1:0]	交替字节大小 该位域定义交替字节大小。 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。
19	ABDTR	交替字节双倍传输速率 0：交替字节阶段禁用DTR。 1：交替字节阶段使能DTR。 **注：**仅支持GD25LX512ME。
18:16	ALTEMOD[2:0]	交替字节模式 该位定义交替字节阶段的操作模式： 000：无交替字节。 001：单线传输交替字节。 010：双线传输交替字节。 011：四线传输交替字节。 100：八线传输交替字节。 101：保留。 110：保留。 111：保留。
15:14	保留	必须保持复位值。
13:12	ADDRSZ[1:0]	地址大小 该位域定义地址大小。 00：8位地址。 01：16位地址。 10：24位地址。

		11：32位地址。
11	ADDRDTR	地址双倍传输速率 0：地址阶段禁用DTR模式。 1：地址阶段使能DTR模式。 **注：**仅支持GD25LX512ME。
10:8	ADDRMOD[2:0]	地址模式 该位定义地址阶段的操作模式： 000：无地址。 001：单线传输地址。 010：双线传输地址。 011：四线传输地址。 100：八线传输地址。 101：保留。 110：保留。 111：保留。
7:6	保留	必须保持复位值。
5:4	INSSZ[1:0]	指令大小 该位域定义指令大小。 00：8位指令。 01：16位指令。 10：24位指令。 11：32位指令。
3	保留	必须保持复位值。
2:0	IMOD[1:0]	命令模式 该位定义指令阶段的操作模式： 000：无指令。 001：单线传输指令。 010：双线传输指令。 011：四线传输指令。 100：八线传输指令。 101：保留。 110：保留。 111：保留。

29.10.17. 回卷时序配置寄存器（OSPI_WPTIMCFG）

地址偏移：0x148
复位值：0x0000 0000

该寄存器只能按字（32位）访问。
当BUSY位为1，该寄存器不可修改。

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

保留	SSAMPLE	保留	DEHQC	保留
	rw		rw	

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

位/位域	名称	描述
31	保留	必须保持复位值。
30	SSAMPLE	采样移位 默认情况下，OSPI在外部存储器驱动数据后二分之一个SCK时钟周期采样。该位允许外部信号延迟的原因采样推迟。 0：不移位。 1：移位二分之一个周期。
29	保留	必须保持复位值。
28	DEHQC	延迟保持1/4周期 0：不延迟保持。 1：延迟保持1/4周期。
27:5	保留	必须保持复位值。
4:0	DUMYC[4:0]	空指令周期数 该位域定义空指令阶段持续时间。

29.10.18. 回卷指令寄存器（OSPI_WPINS）

地址偏移：0x150
复位值：0x0000 0000

该寄存器只能按字（32位）访问。
当BUSY位为1，该寄存器不可修改。

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

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

位/位域	名称	描述
31:0	INSTRUCTION[31:0]	指令 发送到外部存储器的命令信息。

29.10.19. 回卷交替字节寄存器（OSPI_WPALTE）

地址偏移：0x160

复位值：0x0000 0000

该寄存器只能按字（32位）访问。
当BUSY位为1，该寄存器不可修改。

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 发送给外部存储器的可选数据。

29.10.20. 写入传输配置寄存器（OSPI_WTCFG）

地址偏移：0x180
复位值：0x0000 0000

该寄存器只能按字（32位）访问。
当BUSY位为1，该寄存器不可修改。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				DADTR	DATAMOD[2:0]			保留		ALTESZ[1:0]		ABDTR	ALTEMOD[1:0]		
				rw	rw					rw		rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		ADDRSZ[1:0]		ADDRDTR	ADDRMOD[2:0]			保留		INSSZ[1:0]		保留	IMOD[2:0]		
		rw		rw	rw					rw			rw		

位/位域	名称	描述
31:28	保留	必须保持复位值。
27	DADTR	数据双倍传输速率 0：数据阶段禁用DTR。 1：数据阶段使能DTR。 **注：**仅支持GD25LX512ME。
26:24	DATAMOD[2:0]	数据模式 该位定义数据阶段的操作模式。 000：无数据。 001：单线传输数据。 010：双线传输数据。 011：四线传输数据。 100：八线传输数据。 101：保留。 110：保留。

		111：保留。
23:22	保留	必须保持复位值。
21:20	ALTESZ[1:0]	交替字节大小 该位域定义交替字节大小。 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。
19	ABDTR	交替字节双倍传输速率 0：交替字节阶段禁用DTR。 1：交替字节阶段使能DTR。 **注：**仅支持GD25LX512ME。
18:16	ALTEMOD[2:0]	交替字节模式 该位定义交替字节阶段的操作模式： 000：无交替字节。 001：单线传输交替字节。 010：双线传输交替字节。 011：四线传输交替字节。 100：八线传输交替字节。 101：保留。 110：保留。 111：保留。
15:14	保留	必须保持复位值。
13:12	ADDRSZ[1:0]	地址大小 该位域定义地址大小。 00：8位地址。 01：16位地址。 10：24位地址。 11：32位地址。
11	ADDRDTR	地址双倍传输速率 0：地址阶段禁用DTR模式。 1：地址阶段使能DTR模式。 **注：**仅支持GD25LX512ME。
10:8	ADDRMOD[2:0]	地址模式 该位定义地址阶段的操作模式： 000：无地址。 001：单线传输地址。 010：双线传输地址。 011：四线传输地址。 100：八线传输地址。

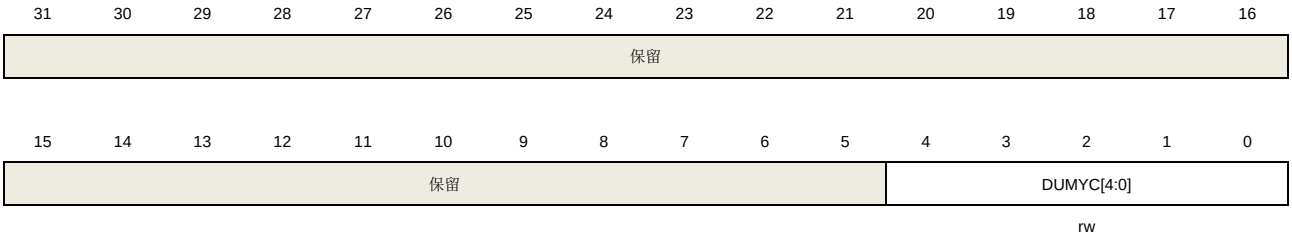
101：保留。
110：保留。
111：保留。

7:6	保留	必须保持复位值。
5:4	INSSZ[1:0]	指令大小 该位域定义指令大小。 00：8位指令。 01：16位指令。 10：24位指令。 11：32位指令。
3	保留	必须保持复位值。
2:0	IMOD[1:0]	命令模式 该位定义指令阶段的操作模式： 000：无指令。 001：单线传输指令。 010：双线传输指令。 011：四线传输指令。 100：八线传输指令。 101：保留。 110：保留。 111：保留。

29.10.21. 写入时序配置寄存器（OSPI_WTIMCFG）

地址偏移：0x188
复位值：0x0000 0000

该寄存器只能按字（32位）访问。
当BUSY位为1，该寄存器不可修改。



位/位域	名称	描述
31:5	保留	必须保持复位值。
4:0	DUMYC[4:0]	空指令周期数 该位域定义空指令阶段持续时间。

29.10.22. 写入指令寄存器（OSPI_WINS）

地址偏移：0x190
复位值：0x0000 0000

该寄存器只能按字（32位）访问。
当BUSY位为1，该寄存器不可修改。

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

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

位/位域	名称	描述
31:0	INSTRUCTION[31:0]	指令 发送到外部存储器的命令信息。

29.10.23. 写入交替字节寄存器（OSPI_WALTE）

地址偏移：0x1A0
复位值：0x0000 0000

该寄存器只能按字（32位）访问。
当BUSY位为1，该寄存器不可修改。

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 发送给外部存储器的可选数据。

30. 时钟相位延迟模块（CPDM）（适用于 GD32H737_757_759）

30.1. 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。

30.2. 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。

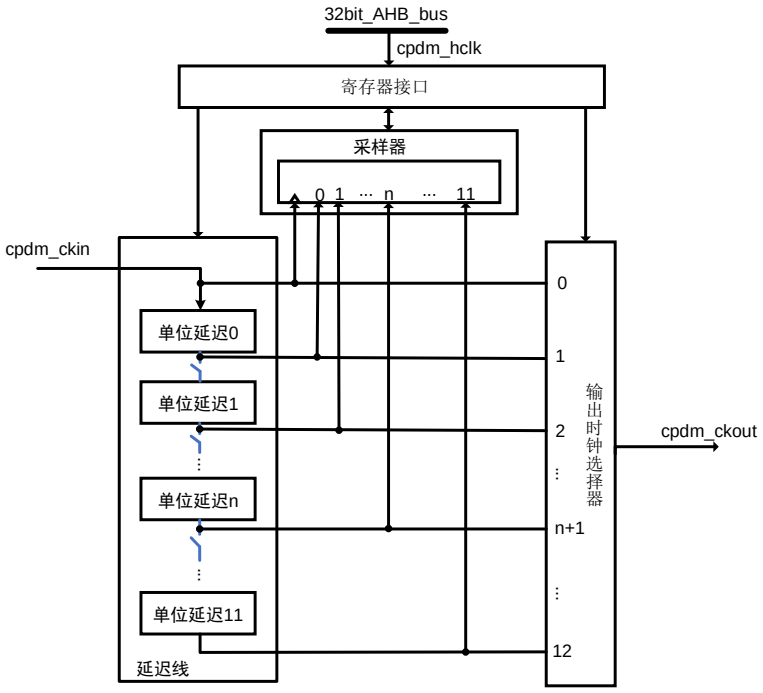
30.3. 功能说明

时钟相位延迟模块（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 SDIO 中使用，请参考*图 36-6. SDIO 框图*。

图 30-1. 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。

30.3.1. 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[6:0]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。

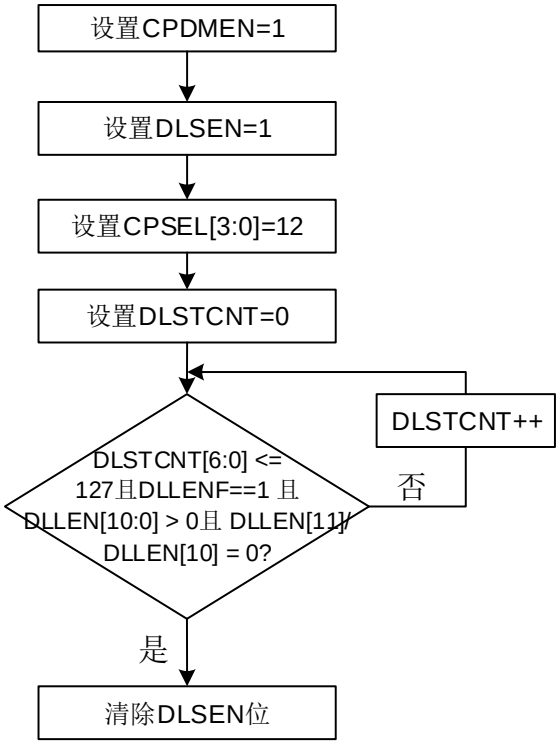
注意：只有当DLSEN为1时才能更改单位延迟及输出时钟相位。

30.3.2. 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30-2. CPDM延迟线长度配置流程图*。

图 30-2. CPDM 延迟线长度配置流程图



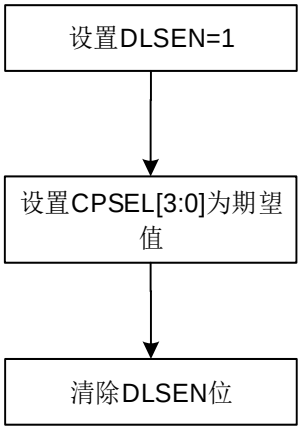
注意：

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 30-3. CPDM 输出时钟相位配置流程图*。

图 30-3. CPDM 输出时钟相位配置流程图



30.4. CPDM 寄存器

CPDM(SDIO0)基地址：0x5200 8000

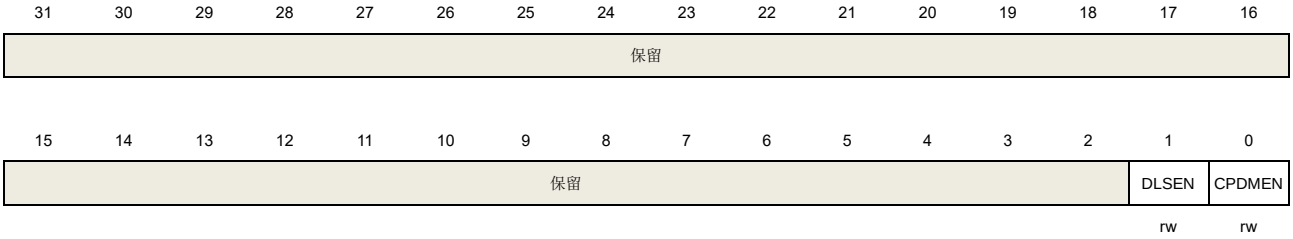
CPDM(SDIO1)基地址：0x4802 2800

30.4.1. 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。



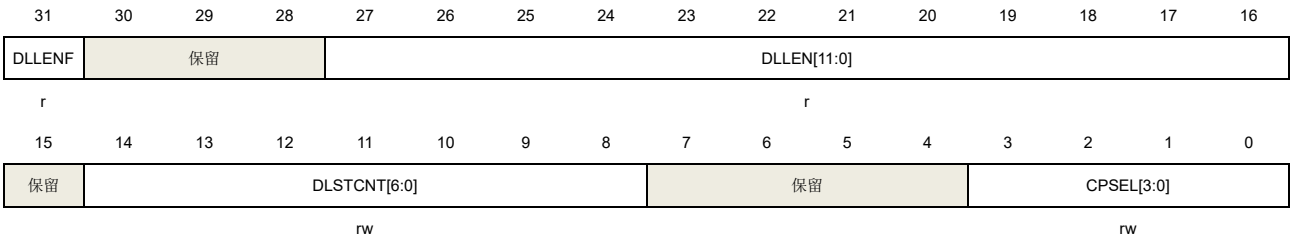
位/位域	名称	描述
31:2	保留	必须保持复位值。
1	DLSEN	延迟线采样模块使能位 0：禁能 CPDM 延迟线采样模块 1：使能 CPDM 延迟线采样模块
0	CPDMEN	CPDM 使能位 0：禁能 CPDM 1：使能 CPDM

30.4.2. 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

31. 数字摄像头接口（DCI）（适用于 GD32H737_757_759）

31.1. 简介

数字摄像头接口是一个同步并行接口，可以从数字摄像头捕获视频和图像信息。它支持不同的颜色空间图像，例如YUV/RGB，以及压缩格式如JPEG。支持CCIR656视频解码器格式并执行额外的图像处理。

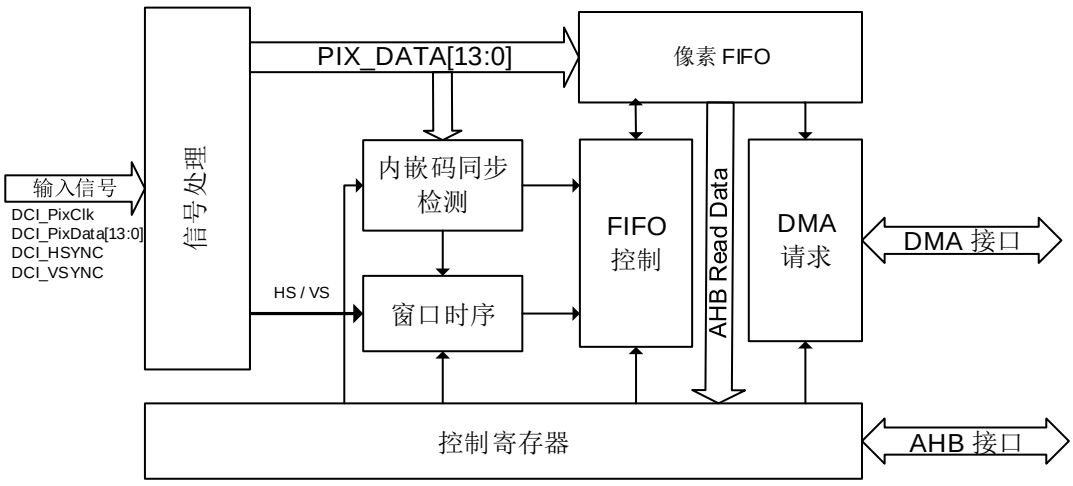
31.2. 主要特性

- 数字视频和图像的捕获；
- 支持8位、10位、12位或14位并行接口；
- DMA高效传输；
- 支持视频和图像裁剪；
- 支持不同的像素数字编码格式，如YCbCr422/RGB565/YUV420/Bayer；
- 支持JPEG压缩格式；
- 支持内嵌码同步和硬件同步；
- 支持CCIR656视频接口和传统传感器接口。

31.3. 结构框图

数字摄像头接口包含以下模块：信号处理单元、像素FIFO、FIFO控制器、窗口时序发生器、内嵌码同步检测器、DMA接口和控制寄存器。

图 31-1. DCI 模块示意图



信号处理单元根据外部输入信号，产生有用的信号信息，为其他的内部模块所用。为确保信号处理单元工作正常，HCLK的频率要高于像素时钟频率的2.5倍。

内嵌码同步检测用于内嵌码同步模式。DCI使用内嵌码同步模式时，视频同步信息内嵌于像素

数据，并无硬件水平或垂直同步信号（DCI_HSYNC或DCI_VSYNC）。DCI通过内嵌码同步检测器从像素数据提取同步信息，然后根据这些信息重新恢复水平和垂直同步信号。

窗口时序模块具有图片剪裁功能。该模块通过来自DCI接口或内嵌码同步检测器的同步信号计算像素点的位置，然后根据寄存器DCI_CWSPOS和DCI_CWSZ的配置决定是否接收该像素点数据。

DCI用一个4字（32位）FIFO缓存接收到的数据。如果DMA模式使能，当收到一个完整的32位数据的时候，DMA接口置位一个DMA请求。控制寄存器提供DCI和软件之间的接口。

31.4. 信号描述

表 31-1. DCI 引脚

方向	引脚名称	信号	位宽	描述
I	DCI_PIXCLK	DCI_PixClk	1	DCI 像素时钟
I	DCI_Dx	DCI_PixData	14	DCI 像素数据
I	DCI_HSYNC	DCI_HSYNC	1	DCI 水平同步
I	DCI_VSYNC	DCI_VSYNC	1	DCI 垂直同步

31.5. 功能描述

31.5.1. DCI 硬件同步模式

在DCI硬件同步模式（DCI_CTL寄存器的ESM为0），DCI_HSYNC和DCI_VSYNC分别用来表示一行的开始和一帧的开始。DCI在DCI_PixClk的上升沿或下降沿（时钟的极性通过DCI_CTL寄存器的CKS位配置），从DCI_PixData[13:0]，捕获像素数据。

图 31-2. 硬件同步模式

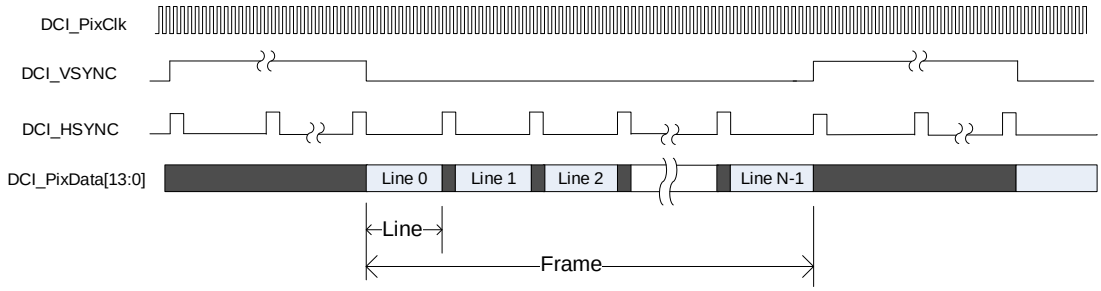
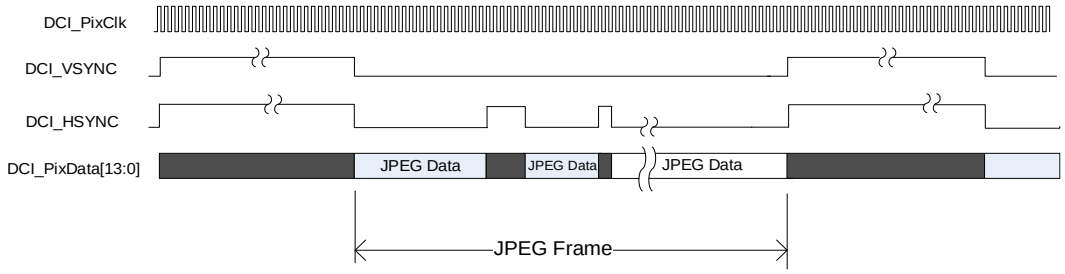


*图31-2. 硬件同步模式*假设DCI_HSYNC和DCI_VSYNC消隐期间的极性为高电平，所以DCI_PixData线仅在DCI_HSYNC和DCI_VSYNC都为低电平期间是有效的。

JPEG 模式

DCI在硬件同步模式时，支持JPEG视频/图像压缩格式。在JPEG模式（DCI_CTL寄存器JM置1），DCI_VSYNC表示一帧的开始，DCI_HSYNC用作数据流有效信号。

图 31-3. 硬件同步模式之 JPEG 格式



31.5.2. 内嵌码同步模式

DCI支持内嵌码同步模式。这一模式仅用到DCI接口的DCI_PixData和DCI_PixClk信号，同步信息内嵌在像素数据中。通过置位DCI_CTL寄存器的ESM位，并且清除JM位，使能内嵌码同步模式。

在内嵌码同步模式，行和帧同步信息被编码为同步码并嵌入像素数据中。有4种同步码：行开始（LS），行结束（LE），帧开始（FS）和帧结束（FE）。该模式数据宽度强制为8，并且每个同步码由4字节序列组成：FF-00-00-XY，MN在DCI_SC寄存器定义。在内嵌码同步模式，0xFF和0x00不应出现在像素数据中以避免误解。

使能内嵌码同步模式之后，DCI开始检测同步码，并恢复行/帧同步信息。例如，如果DCI检测到一个帧结束码以及一个帧开始码，它开始捕获新的帧。

当检测到一个同步码，通过配置DCI_SCUMSK，可能仅需要比较FF_00_00_XY序列XY字节的几位。DCI仅比较DCI_SCUMSK寄存器的非屏蔽位。例如：DCI_SC寄存器的LS位为A5，DCI_SCUMSK的LSM位是F0，DCI将仅比较LS同步码的高4位，因此FF-00-00-A6也将被检测为LS码。

31.5.3. CCIR656 模式

隔行扫描模式

CCIR656标准使用嵌入式时序编解码来代替VSYNC和HSYNC信号。在CCIR656隔行模式下，仅使用PIXCLK和DCI_PixData[7:0]信号。SAV表示每个有效行的开头，EAV是每个有效行的结尾。在EAV和SAV码之间插入数字消隐。DCI解码嵌入式时序，根据场的信息，在软件中重新排列原始的YCbCr图像。

表 31-2. 典型的单行数据组成

EAV码				消隐数据				SAV码				有效数据			
FF	0	0	XY	Cb	Y	Cr	V	FF	0	0	XY	Cb	Y	Cr	Y
4 字节				280（268）字节				4 字节				1440 字节			

EAV和SAV的4字节格式指定如下（以下以十六进制表示）：FF 00 00 XY

前三个字节是固定的，必须是FF 00 00，而第四个字节（XY）由字段和隐藏信息决定，其八位定义如下：1 F V H P3 P2 P1 P0。

F：标记场信息，发送偶数场时为0，发送奇数场时为1。

V：标记消隐信息，传输消隐数据时为1，传输有效视频数据时为0。

H：标记EAV或SAV，其中SAV为0，EAV为1

P0~P3为保护位，其值取决于F、H、V，起到校准的作用。计算方法如下：

表 31-3. SAV 和 EAV 编码

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	F	V	H	P3	P2	P1	P0
1	0	0	0	0	0	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	0	1	1
1	0	1	1	0	1	1	0
1	1	0	0	0	1	1	1
1	1	0	1	1	0	1	0
1	1	1	0	1	1	0	0
1	1	1	1	0	0	0	1

DCI解码并过滤掉数据流中的时序编码，恢复VSYNC和HSYNC信号以供内部使用，例如统计块控制。数据按顺序转发到数据接收端，无需重新排序，场0后跟场1。必须在软件中重新排序场以取回原始图像。

当场发生交替时，场变化中断标志（COFIF）将被置位。CCIR656标准图像一般为625/50 PAL或525/60 NTSC格式。图像由奇偶场、水平和垂直消隐和有效数据组成。一个字段由三个部分组成，顶部垂直消隐、有效数据和底部垂直消隐。

下表显示了PAL/NTSC系统中的帧信息。

表 31-4. 图像格式定义

行数		场/VBlk		线路说明
PAL	NTSC	F	V	
22	19	0	1	场 0 顶部垂直消隐
288	240	0	0	场 0 有效数据
2	3	0	1	场 0 底部垂直消隐
23	20	1	1	场 1 顶部垂直消隐
288	240	1	0	场 1 有效数据
2	3	1	1	场 1 底部垂直消隐
625	525			

逐行扫描模式

图像以逐行方式扫描和排列。此有效场为场1，DCI解码忽略SAV/EAV代码中的F位。

表 31-5. 逐行模式的一般情况

EAV	消隐	SAV	消隐	场 1
⋮				（F = 1）

EAV	消隐	SAV	消隐	
EAV	消隐	SAV	有效数据	
⋮				
EAV	消隐	SAV	有效数据	

在逐行扫描模式下，COFIF将被忽略，但VSIF可以产生中断。当VSYNC信号由摄像头传感器提供时，称为外部VSYNC模式，当从嵌入代码中解码VSIF标志位时，称为内部VSYNC模式。DCI可以执行内部和外部VSYNC模式。

CCIR656 编码的纠错

根据CCIR编码算法，SAV和EAV中的保护位的编码方式允许1比特错误被纠正，或者2比特错误被解码器检测到。DCI中的隔行模式CCIR解码器支持此功能。

对于1比特错误情况，用户可以选择自动更正错误，或者简单地显示为状态标志。对于2比特错误情况，由于解码器无法进行更正，错误将仅显示为状态标志。

当CCIR错误中断被使能时（CCEIE置位），检测到错误时会产生中断。如果启用自动纠错（DCI_CTL寄存器中的AECEN置位），1比特错误将被自动纠正。如果启用2比特错误，则CCEIF错误标志位将置位。

31.5.4. 用快照或连续捕获模式捕获数据

DCI支持两种捕获模式：快照和连续捕获。捕获模式通过DCI_CTL寄存器的SNAP位配置。

正确配置之后，使能DCI并置位DCI_CTL寄存器的CAP位，DCI开始检测帧开始信号。一旦检测到帧开始信号，DCI开始捕获数据。在快照模式（SNAP=1），当一帧被捕获之后，DCI自动停止捕获并清除CAP位，而若在连续模 式，DCI将准备捕获下一帧。在连续模式，DCI捕获频率在FR[1:0]位域定义。如果FR[1:0]=00，DCI捕获每一帧，如果FR[1:0]=01，DCI将每隔一帧捕获一次。

在连续模式，当DCI正在捕获数据的时候，软件可以在任意时间清除CAP位，但DCI并不立即停止捕获。它总是在捕获当前帧之后停止。软件应读回CAP位，以确认是否DCI停止生效。

31.5.5. 窗口功能

DCI支持窗口功能，该功能能够从捕获到的帧剪裁图像的一部分。该功能通过设置DCI_CTL寄存器的WDEN位，在JPEG子模式使能该功能是禁止的。

在捕获期间，DCI不断的计数和计算像素的水平和垂直位置，并且将该位置与剪裁窗口寄存器（DCI_CWSPOS和DCI_CWSZ）的值进行比较，然后丢弃剪裁窗口外的像素数据，仅将位于窗口内的数据压入数据FIFO。

如果一帧已经结束，但DCI_CWSZ定义的垂直行数还没有到达，这种情况下也将触发帧结束标志并且DCI停止捕获。

31.5.6. 像素格式，数据填充和 DMA 接口

DCI支持包含YCbCr422/RGB565等多种像素编码格式，但是DCI只接收这些像素数据，将像素数据补充成全字，并将其压入像素FIFO。DCI不执行任何像素格式转换或数据处理，不关心像素格式细节。

DCI使用32位宽的数据缓冲器在DCI接口和像素FIFO之间传递数据。在这一模块有两种填充方法：字节填充和半字填充，具体使用哪一种取决于DCI接口的数据宽度。数据宽度由DCI_CTL寄存器的DCIF[1:0]配置，在JPEG子模式和内嵌码同步模式，数据宽度固定为8。

当收到一个完整的32位数据的时候，DMA接口发送DMA请求。

字节填充模式

如果DCI接口是8位，使用字节填充模式。在字节填充模式下，四个字节被填充到32位数据缓冲区，在Non-JPEG模式，如果数据缓冲区满或者到达行尾，DCI将压32位数据缓冲区的数据进入像素FIFO。在JPEG子模式，如果数据缓冲区满或者到达帧结束，DCI接口将压32位数据缓冲区的数据进入像素FIFO。

表 31-6. 字节填充模式下的存储视图

D3[7:0]	D2[7:0]	D1[7:0]	D0[7:0]
D7[7:0]	D6[7:0]	D5[7:0]	D4[7:0]

半字填充模式

如果DCI接口配置为10/12/14位，使用半字填充模式。在该模式下，通过高位填0，每像素数据扩展为16位。所以32位宽的数据缓冲区可以包含两个像素数据。当缓冲区满或行结束的时候，DCI将压数据进入像素FIFO。

表 31-7. 半字填充模式下的存储视图

2’b00	D1[13:0]	2’b00	D0[13:0]
2’b00	D3[13:0]	2’b00	D2[13:0]
2’b00	D5[13:0]	2’b00	D4[13:0]
2’b00	D7[13:0]	2’b00	D6[13:0]

31.6. 状态、错误和中断

DCI有几个状态和错误标志位，中断可以根据这些标志判断。如果使能DCI_INTEN的相应使能位，这些状态和错误标志触发DCI全局中断。这些标志可以通过写1到DCI_INTC寄存器清除。

表 31-8. 状态/错误标志

状态标志名	解释
CCEIF	CCIR 错误中断标志
COFIF	CCIR 场转换中断标志
F1IF	CCIR 场 1 中断标志
F0IF	CCIR 场 0 中断标志

ELF	行结束标志
EFF	帧结束标志
OVRF	FIFO 溢出标志
VSF	帧垂直同步消隐标志
ESEF	内嵌同步错误标志

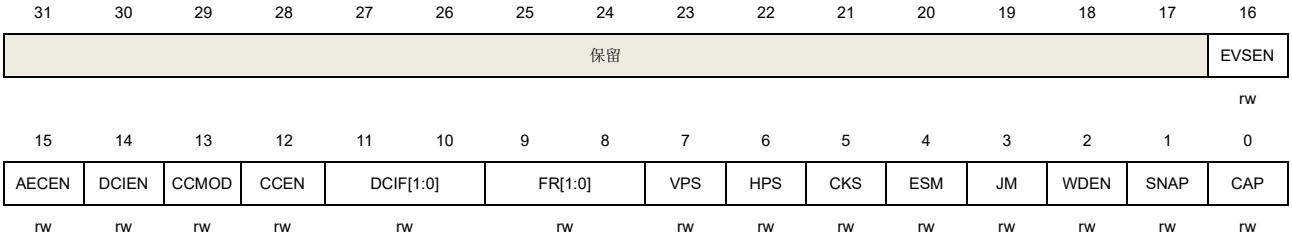
31.7. DCI 寄存器

DCI基地址：0x4802 0000

31.7.1. 控制寄存器（DCI_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
31:17	保留	必须保持复位值。
16	EVSEN	外部 Vsync 使能 0：禁用外部 Vsync 模式 1：使能外部 Vsync 模式 仅适用于 CCIR 逐行扫描模式
15	AECEN	自动纠错使能，1bit 纠错 0：禁能自动纠错 1：使能自动纠错 仅适用于 CCIR 隔行扫描模式
14	DCIEN	DCI 使能 0：DCI 禁止 1：DCI 使能
13	CCMOD	CCIR 模式选择 0：CCIR 逐行模式 1：CCIR 隔行模式
12	CCEN	CCIR 使能 0：禁能 CCIR 1：使能 CCIR
11:10	DCIF[1:0]	DCI 数据格式 00：每个像素时钟捕获 8 位数据 01：每个像素时钟捕获 10 位数据 10：每个像素时钟捕获 12 位数据

		11：每个像素时钟捕获 14 位数据
9:8	FR[1:0]	帧频率 在连续捕获模式，FR 定义帧捕获频率 00：捕获所有帧 01：每隔一帧捕获一次 10：每隔三帧捕获一次 11：保留
7	VPS	垂直同步极性选择 0：消隐期间低电平 1：消隐期间高电平
6	HPS	水平同步极性选择 0：消隐期间低电平 1：消隐期间高电平
5	CKS	时钟极性选择 0：下降沿捕获 1：上升沿捕获
4	ESM	内嵌码同步模式 0：禁止内嵌码同步模式 1：使能内嵌码同步模式
3	JM	JPEG 子模式 0：禁止 JPEG 子模式 1：使能 JPEG 子模式
2	WDEN	窗口使能 0：禁止窗口功能 1：使能窗口功能
1	SNAP	快照模式 0：连续捕获模式 1：快照模式
0	CAP	使能捕获 0：禁止帧捕获 1：使能帧捕获

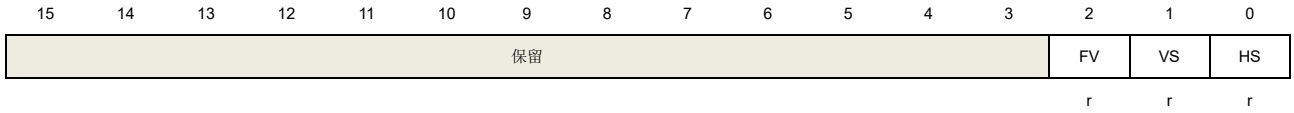
31.7.2. 状态寄存器 0（DCI_STAT0）

地址偏移：0x04
复位值：0x0000 0003

该寄存器只能按字（32位）访问。

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



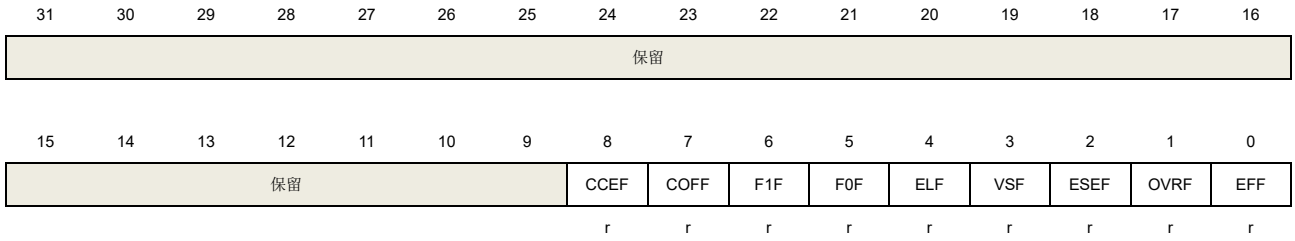


位/位域	名称	描述
31:3	保留	必须保持复位值。
2	FV	FIFO 有效 0：FIFO 没有有效像素数据 1：FIFO 中像素数据有效
1	VS	VS 引脚状态 0：不在垂直消隐期间 1：处于垂直消隐期间
0	HS	HS 引脚状态 0：不在水平消隐期间 1：处于水平消隐期间

31.7.3. 状态寄存器 1（DCI_STAT1）

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
31:9	保留	必须保持复位值。
8	CCEF	CCIR 错误标志 0：没有错误出现 1：在 SVA 或 EVA 码上检测到错误 仅适用于 CCIR 隔行扫描模式
7	COFF	CCIR 场改变标志 0：没有场的改变事件出现 1：场发生了改变

仅适用于 CCIR 隔行扫描模式

6	F1F	CCIR 场 1 0：当前不是场 1 1：当前是场 1 仅适用于 CCIR 隔行扫描模式（在当前场不匹配时自动清除）
5	F0F	CCIR 场 0 0：当前不是场 0 1：当前是场 0 仅适用于 CCIR 隔行扫描模式（在当前场不匹配时自动清除）
4	ELF	行结束标志 0：没有行结束标志 1：DCI 捕获到一行
3	VSF	垂直同步标志 0：没有垂直同步标志 1：检测到垂直同步消隐
2	ESEF	内嵌码同步错误标志 0：没有内嵌码同步错误标志 1：检测到内嵌码同步错误
1	OVRF	FIFO 溢出标志 0：没有 FIFO 溢出 1：发生 FIFO 溢出
0	EFF	帧结束标志 0：没有帧结束标志 1：帧被 DCI 捕获

31.7.4. 中断使能寄存器（DCI_INTEN）

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							CCEIE	COFIE	F1IE	F0IE	ELIE	VSIE	ESEIE	OVRIE	EFIE
							rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:9	保留	必须保持复位值。

8	CCEIE	CCIR 错误中断使能 0：CCIR 错误标志不会产生中断 1：CCIR 错误标志会产生中断
7	COFIE	场中断使能的 CCIR 变化 0：CCIR 场标志变化不会产生中断 1：CCIR 场标志变化将产生中断
6	F1IE	CCIR 场 1 中断使能 0：CCIR 场 1 标志不会产生中断 1：CCIR 场 1 标志将产生中断
5	F0IE	CCIR 场 0 中断使能 0：CCIR 场 0 不会产生中断 1：CCIR 场 0 标志将产生中断
4	ELIE	行结束中断使能 0：行结束标志不产生中断 1：行结束标志产生中断
3	VSIE	垂直同步中断使能 0：垂直同步标志不产生中断 1：垂直同步标志产生中断
2	ESEIE	内嵌码同步错误中断使能 0：内嵌码同步错误标志不产生中断 1：内嵌码同步错误标志产生中断
1	OVRIE	FIFO 溢出中断使能 0：FIFO 溢出不产生中断 1：FIFO 溢出产生中断
0	EFIE	帧结束中断使能 0：帧结束标志不产生中断 1：帧结束标志产生中断

31.7.5. 中断标志寄存器（DCI_INTF）

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							CCEIF	COFIF	F1IF	F0IF	ELIF	VSIF	ESEIF	OVRIF	EFIF
							r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:9	保留	必须保持复位值。
8	CCEIF	CCIR 错误中断标志 仅适用于 CCIR 隔行扫描模式，在 SVA 或 EVA 代码上发现错误。
7	COFIF	场中断标志的 CCIR 变化 仅适用于 CCIR 隔行扫描模式
6	F1IF	CCIR 场 1 中断标志 仅适用于 CCIR 隔行扫描模式（当前场不匹配时自动清零）
5	F0IF	CCIR 场 0 中断标志 仅适用于 CCIR 隔行扫描模式（当前场不匹配时自动清零）
4	ELIF	行结束中断标志
3	VSMF	垂直同步中断标志
2	ESEMF	内嵌码同步错误中断标志
1	OVRMF	FIFO 溢出中断标志
0	EFMF	帧结束中断标志

31.7.6. 中断标志清除寄存器（DCI_INTC）

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							CCEFC	COFFC	F1IF	F0IF	ELFC	VSFC	ESEFC	OVRFC	EFFC
							w	w	w	w	w	w	w	w	w

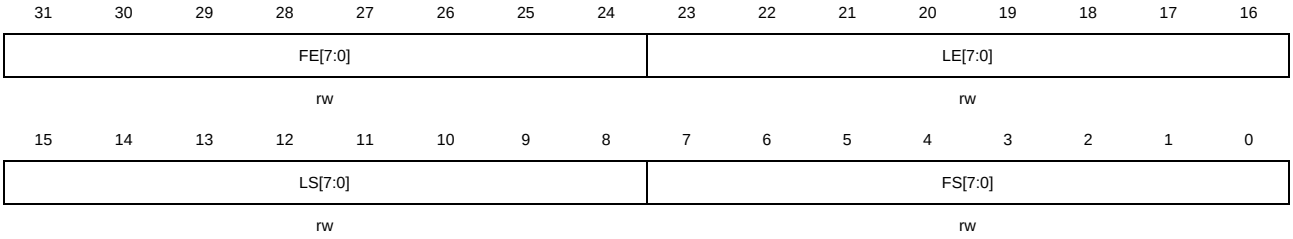
位/位域	名称	描述
31:9	保留	必须保持复位值。
8	CCEFC	CCIR 错误标志清除 写 1 清除 CCIR 错误标志
7	COFFC	CCIR 场转换标志清除 写 1 清除 CCIR 场转换标志
6	F1FC	CCIR 场 1 中断标志清除

		写 1 清除 CCIR 场 1 中断标志
5	F0FC	CCIR 场 0 中断标志 写 1 清除 CCIR 场 0 中断标志
4	ELFC	行结束中断标志清除 写 1 清除行结束中断标志
3	VSFC	垂直同步标志清除 写 1 清除垂直同步标志
2	ESEFC	内嵌码同步错误标志清除 写 1 清除内嵌码同步错误标志
1	OVRFC	FIFO 溢出标志清除 写 1 清除 FIFO 溢出标志
0	EFFC	帧结束中断标志清除 写 1 清除帧结束中断标志

31.7.7. 同步码寄存器（DCI_SC）

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

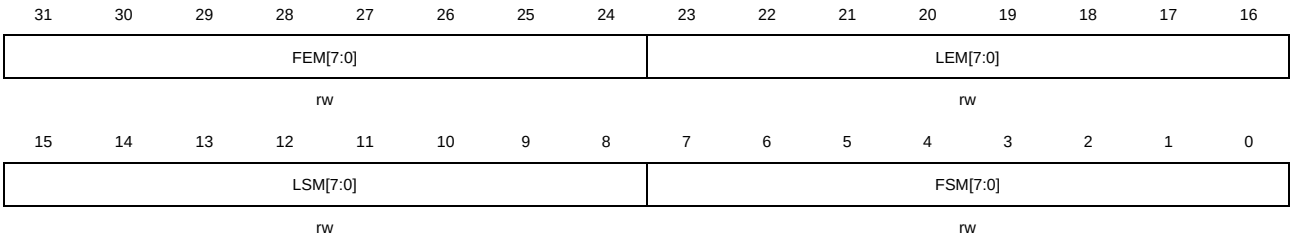


位/位域	名称	描述
31:24	FE[7:0]	内嵌同步模式的帧结束码
23:16	LE[7:0]	内嵌同步模式的行结束码
15:8	LS[7:0]	内嵌同步模式的行开始码
7:0	FS[7:0]	内嵌同步模式的帧开始码

31.7.8. 同步码屏蔽寄存器（DCI_SCUMSK）

地址偏移：0x1C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

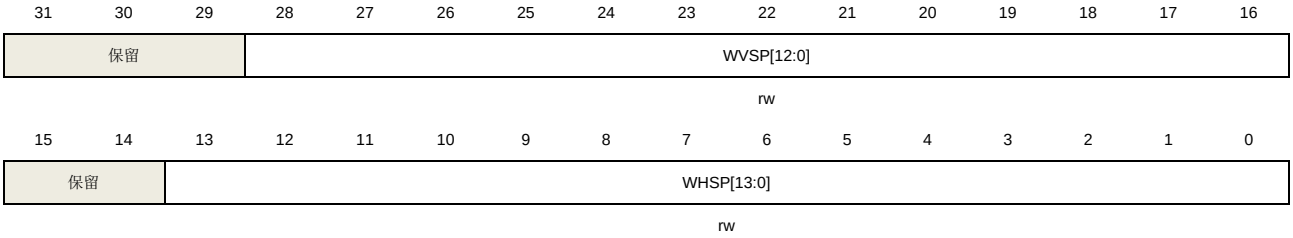


位/位域	名称	描述
31:24	FEM[7:0]	内嵌码同步模式下非屏蔽帧结束码
23:16	LEM[7:0]	内嵌码同步模式下非屏蔽行结束码
15:8	LSM[7:0]	内嵌码同步模式下非屏蔽行开始码
7:0	FSM[7:0]	内嵌码同步模式下非屏蔽帧开始码

31.7.9. 剪裁窗口开始位置寄存器（DCI_CWSPOS）

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
31:29	保留	必须保持复位值。
28:16	WVSP[12:0]	窗口垂直开始位置 值为 0 表示着第一行，以此类推
15:14	保留	必须保持复位值。
13:0	WHSP[13:0]	窗口水平开始位置 值为 0 表示着第一个像素时钟，以此类推

31.7.10. 剪裁窗口大小寄存器（DCI_CWSZ）

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		WVSZ[13:0]													
		rw													

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

位/位域	名称	描述
31:30	保留	必须保持复位值
29:16	WVSZ[13:0]	窗口垂直大小 WVSZ=X 表示 X+1 行
15:14	保留	必须保持复位值
13:0	WHSZ[13:0]	窗口水平大小 WHSZ=X 表示某一行有 X+1 个像素时钟

31.7.11. 数据寄存器（DCI_DATA）

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
DT3[[7:0]								DT2[7:0]							
r								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DT1[7:0]								DT0[7:0]							
r								r							

位/位域	名称	描述
31:24	DT3[7:0]	像素字节 3
23:16	DT2[7:0]	像素字节 2
15:8	DT1[7:0]	像素字节 1
7:0	DT0[7:0]	像素字节 0

32. TFT-LCD 接口（TLI）（适用于 GD32H737_757_759）

32.1. 简介

TLI(TFT-LCD接口)连接同步的LCD接口，并且为无源LCD显示屏提供像素数据，时钟以及时序信号。它支持不同的完全可编程的时序参数显示。一个内嵌的DMA不断的从系统存储器搬移数据到TLI然后输出到外部的LCD显示。TLI支持两个独立的显示层，并支持层窗口和层混叠功能。

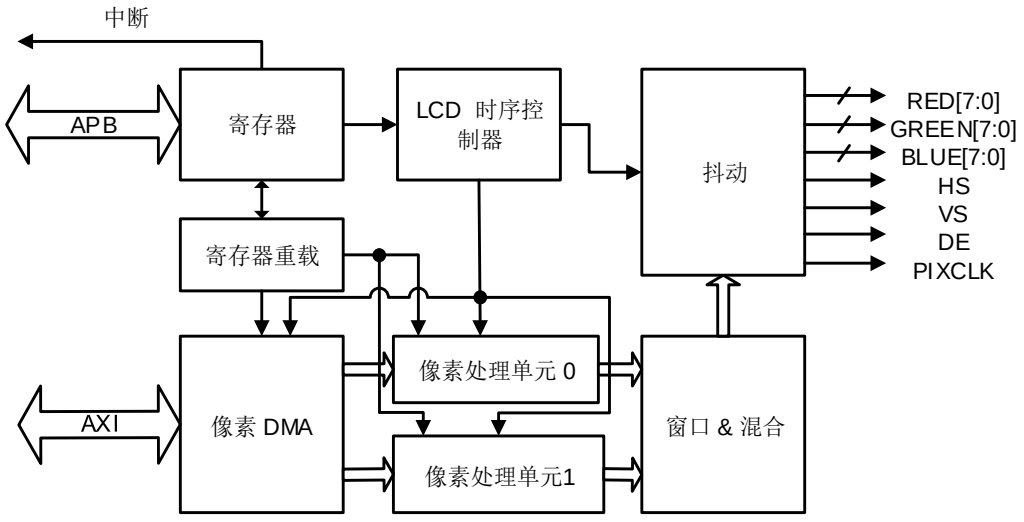
32.2. 主要特点

- 每像素最多24位并行数据输出；
- 支持高达2048*2048的分辨率；
- 完全可编程的时序参数；
- 内嵌DMA处理像素数据拷贝；
- 带有窗口和混合功能的两个独立的层；
- 支持多种像素格式：ARGB8888，RGB888，RGB565等；
- 支持CLUT(颜色查找表)和色键格式；
- 像素低位的抖动操作。

32.3. 结构框图

*图 32-1. TLI 模块框图*展示了 TLI 模块的结构框图。在 TLI 模块有 3 个时钟域。寄存器工作在 APB 时钟域，通过 APB 总线访问。像素 DMA 模块工作在 AXI 时钟域，从系统存储器获取像素数据需要使用 AXI 总线。剩下的模块工作在 TLI 时钟域。TLI 时钟由 PLL2 分频而得到。PLL2 参数和分频因子在 RCU 模块配置。

图 32-1. TLI 模块框图



32.4. 信号线描述

TLI 提供一个 24 位 RGB 并行显示接口，如*表 32-1. TLI 提供的显示接口的引脚*所示。

表 32-1. TLI 提供的显示接口的引脚

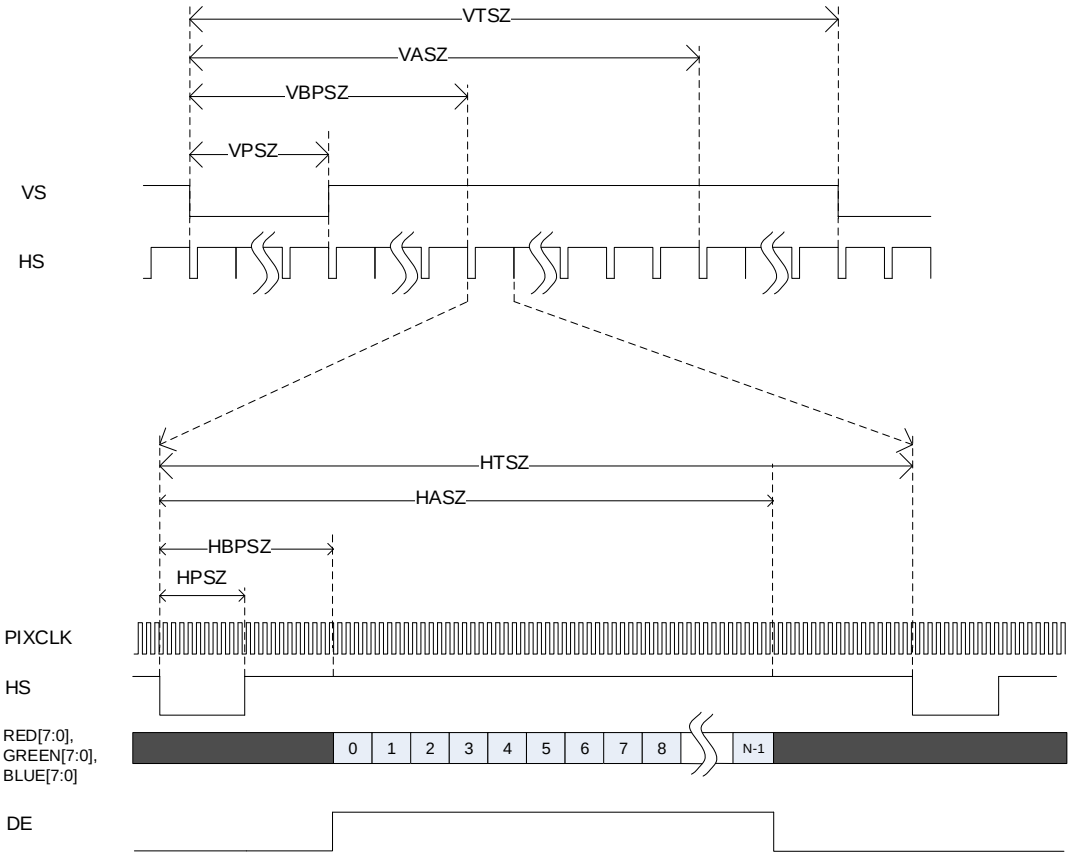
方向	名称	位宽	描述
输出	HS	1	水平同步
输出	VS	1	垂直同步
输出	DE	1	数据使能
输出	PIXCLK	1	像素时钟
输出	RED[7:0]	8	红色像素数据
输出	GREEN[7:0]	8	绿色像素数据
输出	BLUE[7:0]	8	蓝色像素数据

32.5. 功能描述

32.5.1. LCD 显示时序

LCD 接口是一个同步数据接口，包括像素时钟，像素数据以及水平和垂直同步信号。*图 32-2. 显示时序图*展示了一帧的 HS 和 VS 信号时序。时序参数在 TLI_SPSZ，TLI_BPSZ，TLI_ASZ 和 TLI_TSZ 寄存器中配置。这些时序值寄存器假设第一个点的位置是(0,0)。

图 32-2. 显示时序图



32.5.2. 像素 DMA 功能

根据寄存器模块的配置，像素DMA不断的从存储器读像素数据到内部PPU(像素处理单元)像素缓冲区。

使能之后，像素DMA开始从系统取像素数据，只要像素缓冲区未满，就将这些数据压进像素缓冲区。TLI总是用AXI BURST16方式取像素数据。

TLI支持两个独立的帧层，在系统中每层有不同的帧缓冲区地址。像素DMA仅有一个AXI访问端口，所以如果都使能的话，在取像素的时候，它将交替读取两层的数据。

TLI_LxFBADDR寄存器的FBADD位定义了每层的帧缓冲区地址。

TLI_LxFLLEN寄存器的FLL定义了一行的长度，以字节为单位。如果一行的长度是N，FLL=N+3。

在系统存储器里，两行之间可能有一些间隔存储空间，这一间隔空间信息在TLI_LxFLLEN寄存器的STDOFF位域定义。如果某行第一个像素的地址是M，那么下一行第一个像素的地址将是M+STDOFF。如果两行之间无间隔存储空间，则STDOFF=FLL-7。

TLI_LxFTLN寄存器的FTLN位域定义了一帧的行数。

32.5.3. 像素格式

像素 DMA 以双字为单位，将像素数据压入 PPU，然后由 PPU 负责将各种像素格式转换

成内部 ARGB8888 格式。如*表 32-2. 八种像素格式*所示，TLI 支持多达 8 种像素格式。TLI_LxPPF 寄存器的 PPF[2:0]位域定义了像素格式。

ARGB8888格式要求每通道(Alpha, Red, Green和Blue)有8位数据。但是ARGB1555和ARGB4444格式的某些通道是少于8位的。 PPU通过拷贝高位填充到低位的方式，将其转换成ARGB8888。当处理RGB888和RGB565格式时，PPU假设Alpha=255，并且如果通道的位数少于8，也将拷贝高位填充到低位。

AL88, AL44和L8格式是LUT(颜色查找表)格式。在这些通道里，L是颜色查找表的地址。TLI有两个内部颜色查找表：每层各一个。内部颜色查找表的大小是256x24bits (256个节点，每节点存储24位RGB值)。当处理LUT格式像素时，PPU从颜色查找表读出一个节点，并用这个节点值作为RGB值。由于颜色查找表的地址是8位的，如果L通道的位数少于8位，PPU也将拷贝高位填充到低位。颜色查找表的节点在复位后是不会被初始化的，因此在显示一个颜色查找表格式层之前，应用程序应该用TLI_LxLUT寄存器，写入适当的值初始化颜色查找表。

每层都支持色键模式。TLI_LxCKEY寄存器定义了一个RGB值。当某层的色键模式使能，PPU将会把该层每一个像素的RGB值与TLI_LxCKEY中的值相比较，如果值匹配，将会置该像素的ARGB值为0。

表 32-2. 八种像素格式

PPF[2:0]	像素格式
000	ARGB8888
001	RGB888
010	RGB565
011	ARGB1555
100	ARGB4444
111	AL88
101	L8
110	AL44

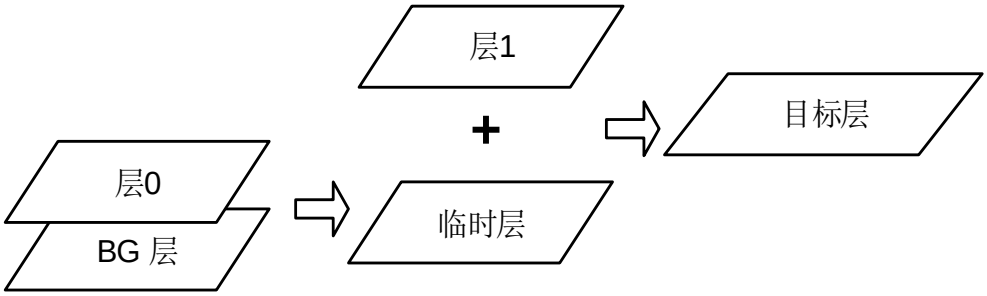
32.5.4. 层窗口和混合功能

TLI每层都支持窗口功能以及两层的混合功能。TLI首先执行每层的窗口操作，然后将两层混合成一帧。

窗口功能定义了一个显示窗口，每层有独立的窗口参数定义寄存器LTI_LxHPOS和LTI_LxVPOS。窗口参数定义了一层内的显示窗。窗口内的像素将保持它的原值，但是窗口外的像素值将被TLI_LxDC 寄存器定义的像素值替代。

混合单元首先混合层0和BG层得到一个临时层，然后混合层1和临时层，得到目的层。BG层的ARGB值在TLI_BGC寄存器定义。如果某层被禁止，混合功能使用该层的默认颜色。

图 32-3. 混合过程框图



混合公式

通用混合公式:

$$BC=BF_1*C+BF_2*C_S \quad (32\text{-}1)$$

- BC 混合的颜色
- BF1 混合因子 1
- C 当前层颜色
- BF_2 混合因子 2
- C_S 下一级层混合的颜色

当前像素的混合因子有两种取值，由寄存器配置。一种是归一化的像素Alpha乘以归一化的恒定Alpha，另一种是归一化的恒定Alpha。

32.5.5. Layer 配置重载

如上面所描述的，每层有自己的帧缓冲区，像素格式，窗口，默认颜色配置寄存器并且每个寄存器都有影子寄存器。影子寄存器与真正的寄存器共享相同的地址。每次当应用程序对层相关的寄存器地址执行写操作，相应的影子寄存器立即更新，但是直到一个重载操作前真正寄存器的值是不会改变的，而只有真正寄存器的值才是影响TLI功能的。

应用程序有两种方法触发一个重载操作：请求重载和帧消隐重载。对于请求重载模式，在应用程序设置TLI_RL寄存器的RQR位之后，TLI立即加载影子寄存器的值到真正寄存器。对于帧消隐重载模式，设置TLI_RL寄存器的FBR位之后，TLI等待帧垂直消隐，然后加载影子寄存器的值到真正寄存器。在两种模式下，重载成功完成之后，硬件自动清除RQR或FBR位。

32.5.6. 抖动

抖动模块为每一个像素通道加一个2位的伪随机值当18位接口用来显示24位数据的时候，该功能能够使图像更平滑。应用程序可以用TLI_CTL寄存器的DFEN位开启这一功能。

32.6. 中断

TLI中有以下错误和状态标志位。中断可以从这些状态判定，状态标志可以触发一个全局中断，错误标志将触发错误中断。

表 32-3. 状态标志

状态标志位	描述
LMF	行标记标志
LCRF	层配置重载标志

表 32-4. 错误标志

错误标志位	描述
TEF	传输错误标志
FEF	FIFO 错误标志

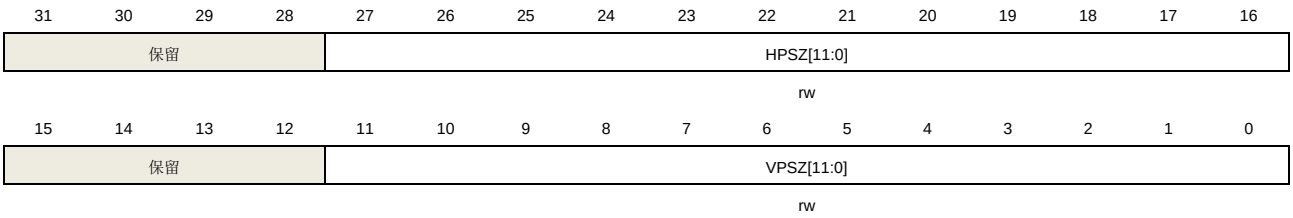
32.7. TLI 寄存器

TLI基地址：0x5000 1000

32.7.1. 同步脉冲宽度寄存器（TLI_SPSZ）

偏移地址：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
31:28	保留	必须保持复位值。
27:16	HPSZ[11:0]	水平同步脉冲宽度 HPSZ 值应该配置成水平同步脉冲像素的个数减 1。
15:12	保留	必须保持复位值。
11:0	VPSZ[11:0]	垂直同步脉冲宽度 VPSZ 值应该配置成垂直同步脉冲像素的个数减 1。

32.7.2. 后沿宽度寄存器（TLI_BPSZ）

偏移地址：0x0C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



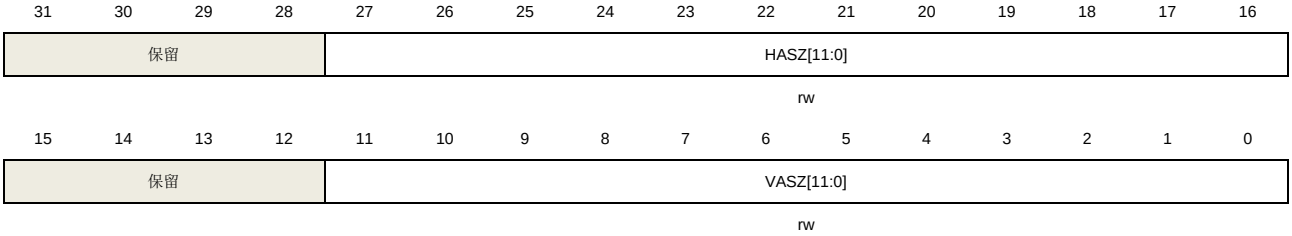
位/位域	名称	描述
31:28	保留	必须保持复位值。
27:16	HBPSZ[11:0]	水平后沿加同步脉冲的宽度 HBPSZ 值应该配置成水平后沿像素个数加同步脉冲像素个数减 1。

15:12	保留	必须保持复位值。
11:0	VBPSZ[11:0]	垂直后沿加同步脉冲的宽度 VBPSZ 值应该配置成垂直后沿像素个数加同步脉冲像素个数减 1。

32.7.3. 有效宽度寄存器（TLI_ASZ）

偏移地址：0x10
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

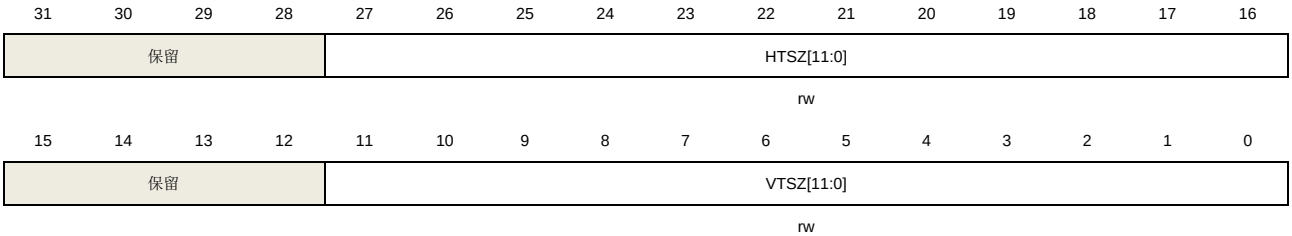


位/位域	名称	描述
31:28	保留	必须保持复位值。
27:16	HASZ[11:0]	水平有效宽度加后沿像素和水平同步像素宽度 HASZ 值应该配置成水平有效宽度加后沿像素和水平同步像素个数减 1。
15:12	保留	必须保持复位值。
11:0	VASZ[11:0]	垂直有效宽度加后沿像素和垂直同步像素宽度 VASZ 值应该配置成垂直有效宽度加后沿像素和垂直同步像素个数减 1。

32.7.4. 总宽度寄存器（TLI_TSZ）

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
31:28	保留	必须保持复位值。
27:16	HTSZ[11:0]	显示器的水平总宽度，包括有效宽度，后沿，同步脉冲和前沿 HTSZ 值应该配置成水平有效宽度像素的个数加后沿像素，前沿像素和同步脉冲像素

		减 1。
15:12	保留	必须保持复位值。
11:0	VTSZ[11:0]	显示器的垂直总宽度，包括有效宽度，后沿,同步脉冲和前沿 VTSZ 值应该配置成垂直有效宽度像素的个数加后沿像素，前沿像素和同步脉冲像素减 1。

32.7.5. 控制寄存器（TLI_CTL）

偏移地址：0x18
复位值：0x0000 2220

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HPPS	VPPS	DEPS	CLKPS	保留											DFEN
rw	rw	rw	rw												rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	RDB[2:0]			保留	GDB[2:0]			保留	BDB[2:0]			保留			TLIEN
	r				r				r						rw

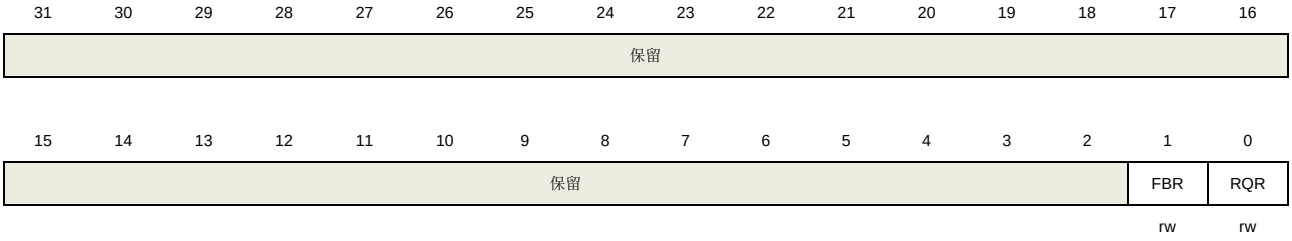
位/位域	名称	描述
31	HPPS	水平脉冲极性选择 0：水平同步脉冲低电平有效 1：水平同步脉冲高电平有效
30	VPPS	垂直脉冲极性选择 0：垂直同步脉冲低电平有效 1：垂直同步脉冲高电平有效
29	DEPS	非数据使能极性选择 0：非数据使能低电平有效 1：非数据使能高电平有效
28	CLKPS	像素时钟极性选择 0：像素时钟是 TLI 时钟 1：像素时钟是 TLI 时钟翻转
27:17	保留	必须保持复位值。
16	DFEN	抖动功能使能 0：禁止抖动功能 1：使能抖动功能
15	保留	必须保持复位值。
14:12	RDB[2:0]	红色通道抖动位数 固定为 2，只读

11	保留	必须保持复位值。
10:8	GDB[2:0]	绿色通道抖动位数 固定为 2，只读
7	保留	必须保持复位值。
6:4	BDB[2:0]	蓝色通道抖动位数 固定为 2，只读
3:1	保留	必须保持复位值。
0	TLIEN	TLI 使能位 0：禁止 TLI 1：使能 TLI

32.7.6. 重载层配置寄存器（TLI_RL）

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
31:2	保留	必须保持复位值。
1	FBR	帧消隐重载请求 此位通过软件置位，在重载之后由硬件清除。 0：禁止重载 1：层配置将在帧消隐时被重载进入真正寄存器。
0	RQR	立即重载请求 此位通过软件置位，在重载之后由硬件清除。 0：禁止重载 1：层配置将在该位置位之后被重载进入真正寄存器。

32.7.7. 背景色配置寄存器（TLI_BGC）

偏移地址：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								BVR[7:0]							
								rw							

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

位/位域	名称	描述
31:24	保留	必须保持复位值。
23:16	BVR[7:0]	背景红色值
15:8	BVG[7:0]	背景绿色值
7:0	BVB[7:0]	背景蓝色值

32.7.8. 中断使能寄存器（TLI_INTEN）

偏移地址：0x34
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												LCRIE	TEIE	FEIE	LMIE
												rw	rw	rw	rw

位/位域	名称	描述
31:4	保留	必须保持复位值。
3	LCRIE	层配置重载中断使能 0：层配置重载标志将不产生中断 1：层配置重载标志将产生中断
2	TEIE	传输错误中断使能 0：传输错误标志将不产生中断 1：传输错误标志将产生中断
1	FEIE	FIFO 错误中断使能 0：FIFO 错误标志将不产生中断 1：FIFO 错误标志将产生中断
0	LMIE	行标记中断使能 0：行标记标志将不产生中断 1：行标记标志将产生中断

32.7.9. 中断标志寄存器（TLI_INTF）

地址偏移：0x38

复位值：0x0000 0000

该寄存器只能按字(32位)访问。



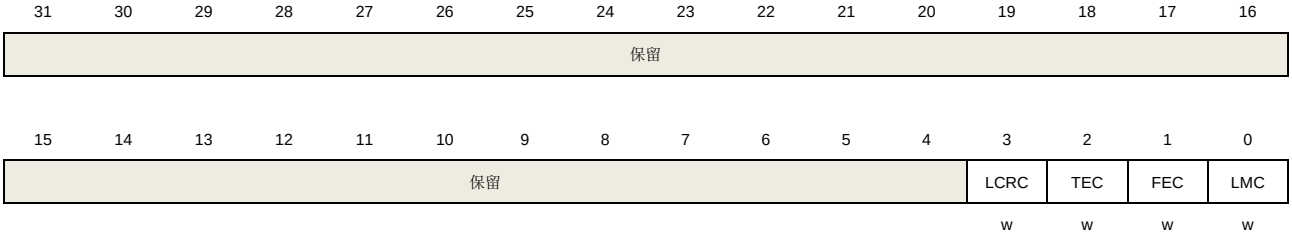
位/位域	名称	描述
31:4	保留	必须保持复位值。
3	LCRF	层配置重载标志 0：无层配置重载标志出现 1：由 TLI_RL 寄存器的 FBR 位置位触发了层配置重载
2	TEF	传输错误标志 0：无传输错误 1：一个传输错误在 AXI 总线上出现
1	FEF	FIFO 错误标志 0：无 FIFO 错误标志 1：出现 FIFO 下溢错误
0	LMF	行标记标志 0：没有行标记标志 1：行数达到 TLI_LM 寄存器中设置的特定值

32.7.10. 中断标志清除寄存器（TLI_INTC）

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

31:4	保留	必须保持复位值。
3	LCRC	层配置重载标志清除 写 1 清除层配置重载标志
2	TEC	传输错误标志清除 写 1 清除传输错误标志
1	FEC	FIFO 错误标志清除 写 1 清除 FIFO 错误标志
0	LMC	行标记标志清除 写 1 清除行标记标志

32.7.11. 行标记寄存器（TLI_LM）

地址偏移: 0x40
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

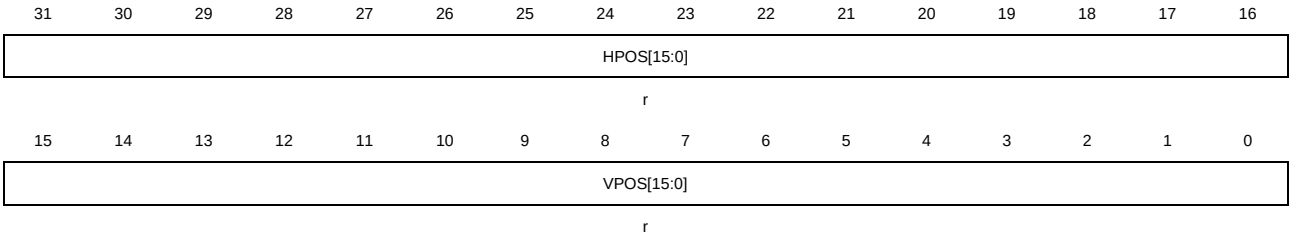


位/位域	名称	描述
31:11	保留	必须保持复位值。
10:0	LM[10:0]	行标记值 当行数到达该值，TLI_INTF 寄存器的 LMF 位将置位。

32.7.12. 当前像素位置寄存器（TLI_CPPOS）

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

31:16	HPOS[15:0]	水平位置 当前显示的像素的水平位置
15:0	VPOS[15:0]	垂直位置 当前显示的像素的垂直位置

32.7.13. 状态寄存器（TLI_STAT）

地址偏移：0x48
复位值：0x0000 000F

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	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												HS	VS	HDE	VDE
												r	r	r	r

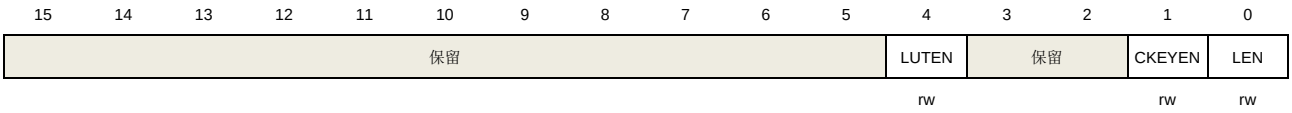
位/位域	名称	描述
31:4	保留	必须保持复位值。
3	HS	TLI 当前的 HS 状态
2	VS	TLI 当前的 VS 状态
1	HDE	当前的 HDE 状态 0：TLI_CPPOS 寄存器 HPOS 并未位于 TLI_BPSZ 寄存器 HBPSZ 与 TLI_ASZ 寄存器 HASZ 之间 1：TLI_CPPOS 寄存器 HPOS 位于 TLI_BPSZ 寄存器 HBPSZ 与 TLI_ASZ 寄存器 HASZ 之间
0	VDE	当前的 VDE 状态 0：TLI_CPPOS 寄存器 VPOS 并未位于 TLI_BPSZ 寄存器 VBPSZ 与 TLI_ASZ 寄存器 HASZ 之间 1：TLI_CPPOS 寄存器 VPOS 位于 TLI_BPSZ 寄存器 VBPSZ 与 TLI_ASZ 寄存器 VASZ 之间

32.7.14. 第 x 层控制寄存器（TLI_LxCTL）（x = 0, 1）

地址偏移：0x84 + 0x80 * x
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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



位/位域	名称	描述
31:5	保留	必须保持复位值。
4	LUTEN	LUT 使能 0：禁止 LUT 1：使能 LUT
3:2	保留	必须保持复位值。
1	CKEYEN	色键使能 0：禁止色键功能 1：使能色键功能
0	LEN	层使能 0：禁止层 1：使能层

32.7.15. 第 x 层水平位置参数寄存器（TLI_LxHPOS）（x = 0, 1）

偏移地址：0x88 + 0x80 * 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
31:28	保留	必须保持复位值。
27:16	WRP[11:0]	窗口右侧位置
15:12	保留	必须保持复位值。
11:0	WLP[11:0]	窗口左侧位置

32.7.16. 第 x 层垂直位置参数寄存器（TLI_LxVPOS）（x = 0, 1）

地址偏移：0x8C + 0x80 * x
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

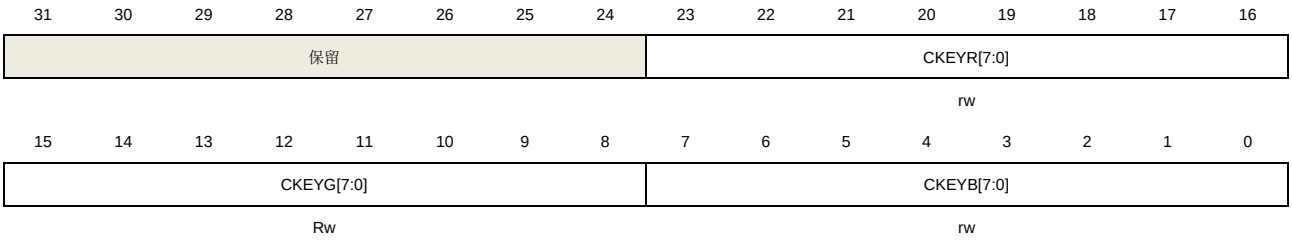


位/位域	名称	描述
31:28	保留	必须保持复位值。
27:16	WBP[11:0]	窗口底部位置
15:12	保留	必须保持复位值。
11:0	WTP[11:0]	窗口顶部位置

32.7.17. 第 x 层色键值寄存器（TLI_LxCKEY）（x = 0, 1）

地址偏移：0x90 + 0x80 * x
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



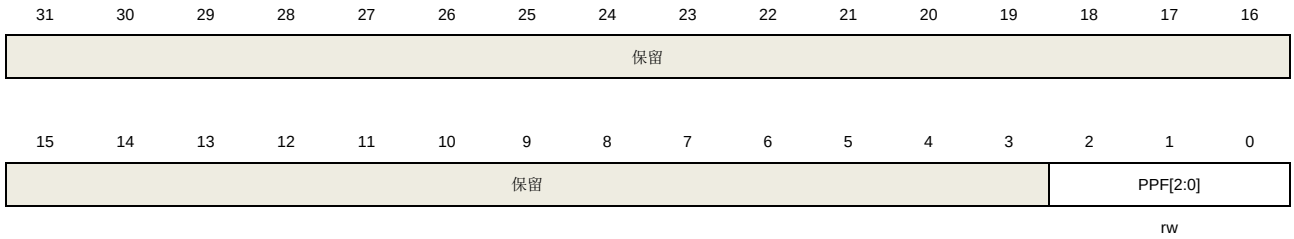
位/位域	名称	描述
31:24	保留	必须保持复位值。
23:16	CKEYR [7:0]	色键红色值
15:8	CKEYG [7:0]	色键绿色值
7:0	CKEYB [7:0]	色键蓝色值

注意：如果某层的像素RGB等于TLI_LxCKEY寄存器定义的值，该像素RGB值复位为0。这意味着这些像素对其它层来说是透明的。

32.7.18. 第 x 层像素格式寄存器（TLI_LxPPF）（x = 0, 1）

地址偏移：0x94 + 0x80 * x
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

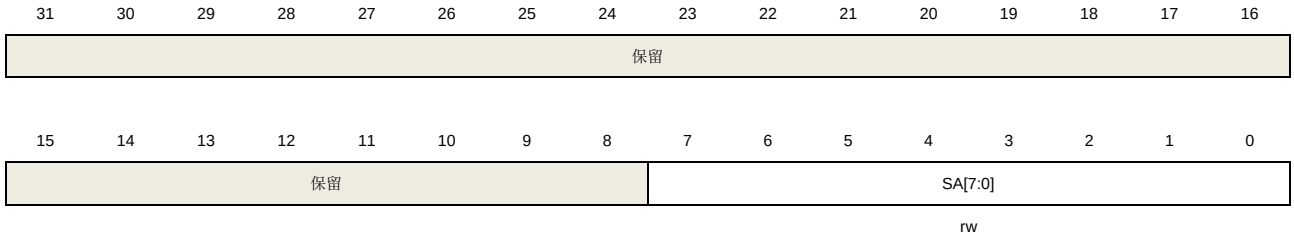


位/位域	名称	描述
31:3	保留	必须保持复位值。
2:0	PPF[2:0]	像素格式 这些位配置像素格式 000：ARGB8888 001：RGB888 010：RGB565 011：ARGB1555 100：ARGB4444 101：L8 110：AL44 111：AL88

32.7.19. 第 x 层恒定 Alpha 寄存器（TLI_LxSA）（x = 0, 1）

地址偏移：0x98 + 0x80 * x
复位值：0x0000 00FF

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	SA[7:0]	恒定 Alpha 可用于计算混合因子。

32.7.20. 第 x 层默认颜色寄存器（TLI_LxDC）（x = 0, 1）

地址偏移：0x9C + 0x80 * x
复位：0x0000 0000

该寄存器只能按字(32位)访问。

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

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

位/位域	名称	描述
31:24	DCA[7:0]	默认颜色 ALPHA
23:16	DCR[7:0]	默认颜色红色
15:8	DCG[7:0]	默认颜色绿色
7:0	DCB[7:0]	默认颜色蓝色

注意：当该层被禁止或TLI_LxHPOS和TLI_LxVPOS定义的窗口之外，默认颜色值生效。

32.7.21. 第 x 层混合寄存器（TLI_LxBLEND）（x = 0, 1）

地址偏移：0xA0 + 0x80 * x

复位值：0x0000 0607

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	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					ACF1[2:0]			保留					ACF2[2:0]		
					rw								rw		

位/位域	名称	描述
31:11	保留	必须保持复位值。
10:8	ACF1[2:0]	Alpha 混合因子 1 计算方法 000：保留 001：保留 010：保留 011：保留 100：归一化的恒定 Alpha 101：保留 110：归一化的像素 Alpha 乘以归一化的恒定 Alpha 111：保留
7:3	保留	必须保持复位值。
2:0	ACF2[2:0]	Alpha 混合因子 2 计算方法 000：保留

001：保留
010：保留
011：保留
100：保留
101：1-归一化的恒定 Alpha
110：保留
111：1-归一化的像素 Alpha 乘以归一化的恒定 Alpha

32.7.22. 第 x 层帧基地址寄存器（TLI_LxFBADDR）（x = 0, 1）

地址偏移：0xAC + 0x80 * 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
FBADD[31:16]															
rw															

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

位/位域	名称	描述
31:0	FBADD[31:0]	帧缓冲区基地址 帧缓冲区基地址

32.7.23. 第 x 层行长度寄存器（TLI_LxFLLEN）（x = 0, 1）

地址偏移：0xB0 + 0x80 * x
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		STDOFF[13:0]													
		rw													

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

位/位域	名称	描述
31:30	保留	必须保持复位值。
29:16	STDOFF[13:0]	步幅偏移 这个值定义了从某行起始处到下一行起始处之间的字节数
15:14	保留	必须保持复位值。

13:0	FLL[13:0]	行长度 这个值为一行的字节数+7

32.7.24. 第 x 层总行数寄存器（TLI_LxFTLN）（x = 0, 1）

地址偏移：0xB4 + 0x80 * x
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

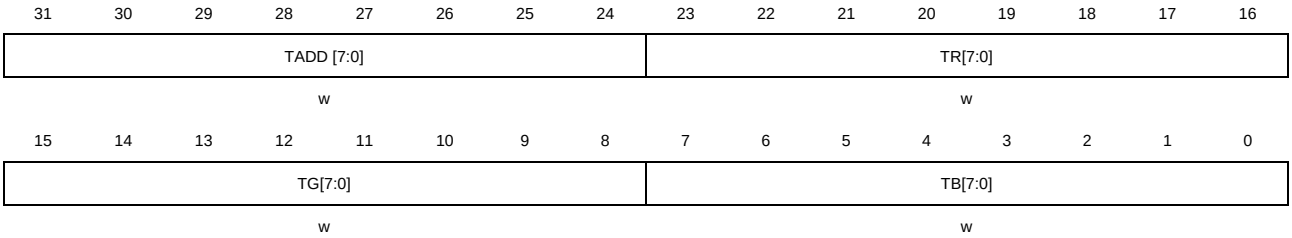


位/位域	名称	描述
31:11	保留	必须保持复位值。
10:0	FTLN[10:0]	总行数 这个值定义了一帧行数

32.7.25. 第 x 层颜色查找表寄存器（TLI_LxLUT）（x = 0, 1）

地址偏移：0xC4 + 0x80 * 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	TADD[7:0]	颜色查找表写地址 颜色查找表位于该地址的节点的值，将由写入的 TR，TG 和 TB 值更新。
23:16	TR[7:0]	LUT 节点的红色值
15:8	TG[7:0]	LUT 节点的绿色值
7:0	TB[7:0]	LUT 节点的蓝色值

33. S/P 数字音频接口接收器（RSPDIF）（适用于GD32H737_757_759）

33.1. 简介

S/P数字音频接口接收器（RSPDIF）模块提供了接收及解码SPDIF音频数据流的功能。

33.2. 主要特征

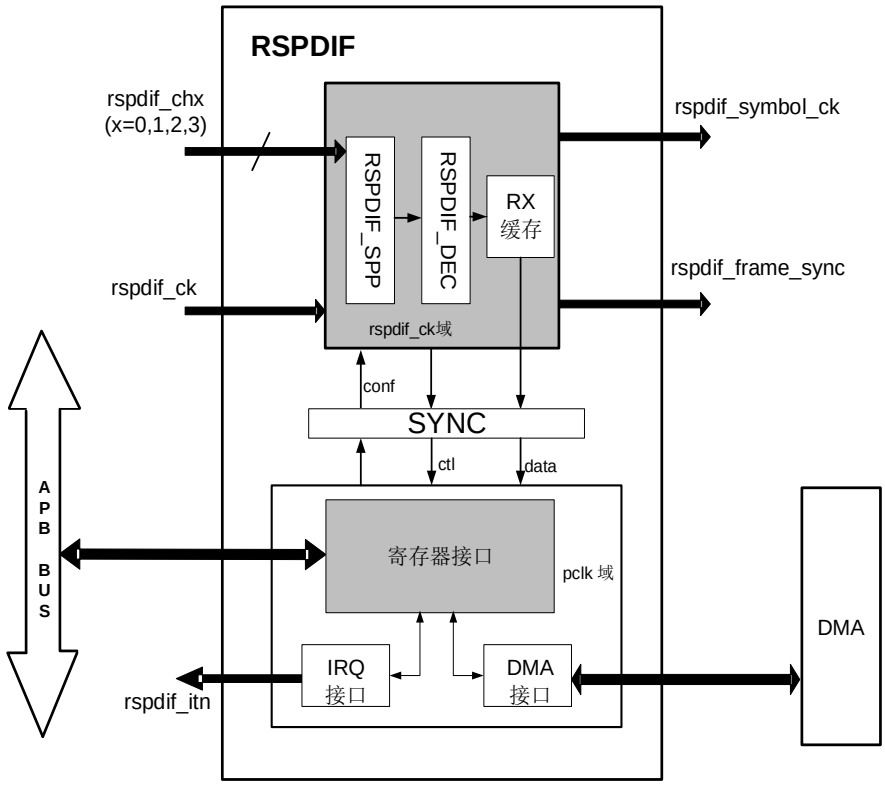
- 支持IEC-60958和IEC-61937音频协议。
- 支持高达4路输入。
- 支持接收双缓冲功能。
- 双时钟域：用于寄存器接口的PCLK和用于其他部分的RSPDIF_CK。
- 支持最大符号率：12.288MHz。
- 支持8KHz到192KHz立体声。
- 支持自动符号率检测。
- 生成符号时钟。
- 检测接收的数据的奇偶校验位。
- 支持多种数据处理的方式，可以分别处理音频数据和用户通道信息或者一起处理。
- 支持分别使用DMA通信接收音频数据和用户通道信息。
- 支持中断功能。

33.3. 功能说明

33.3.1. RSPDIF 结构框图

S/P 数字音频接口接收器（RSPDIF）模块提供了接收及解码 SPDIF 音频数据流的功能。音频数据、信道状态（CS）和用户数据（U）均可通过 DMA 接口接收。RSPDIF 收发器允许处理 S/PDIF 通道状态（CS）和用户数据（U），并支持输入采样频率的精确测量。

图 33-1. RSPDIF 模块框图



说明：

rspdif_frame_sync：RSPDIF 帧同步信号

rspdif_symbol_ck：RSPDIF 符号时钟

RSPDIF 模块负责解码从 RSPDIF_CH[3:0]接收到的 S/PDIF 流，用于接收符合 IEC-60958 和 IEC-61937 标准的 S/P 音频数据。这些标准支持高采样率的简单立体声流，以及压缩的多声道环绕声。

RSPDIF 重新采样传入信号，解码曼彻斯特流，识别子帧，帧和块元素，并将解码后的数据和相关的状态标志传递给 RSPDIF 寄存器。RSPDIF 部分可以通过 APB 总线完全控制，并为用户信息和通道信息提供了专用路径，可以处理音频样本和通道、用户信息两个 DMA 通道。中断服务也可作为 DMA 的替代功能，用于发送错误信号或关键状态。

RSPDIF 解码传入的音频数据流时，还提供了两个信号：

- rspdif_frame_sync：该信号频率等于帧速率，在RSPDIF每次检测到子帧报头时进行翻转，占空比为50%。
- rspdif_symbol_ck：该信号频率等于符号速率。该信号可为系统内部组件提供驱动，如SAI端口，和外部组件，如A/Ds或D/ As，可通过相关寄存器提供时钟控制。具体生成条件请参考***RSPDIF时钟管理***。

33.3.2. S/PDIF 协议

S/PDIF（Sony/Philips Digital Interface）是一种数字音频互连，用于消费类音频设备在一定的

短距离内输出音频。信号通过带有 RCA 连接器的同轴电缆或带有 TOSLINK 连接器的光纤电缆传输。S/PDIF 可用于连接家庭影院的组件和其他数字高保真系统。

S/PDIF 协议是一种数据链路层协议，也是一套物理层规范，用于通过光缆或电缆在设备和组件之间传输数字音频信号。该名称代表索尼/飞利浦数字互连格式，但也被称为索尼/飞利浦数字接口。索尼和飞利浦是 S/PDIF 的主要设计者。S/PDIF 符合 IEC 60958 标准。

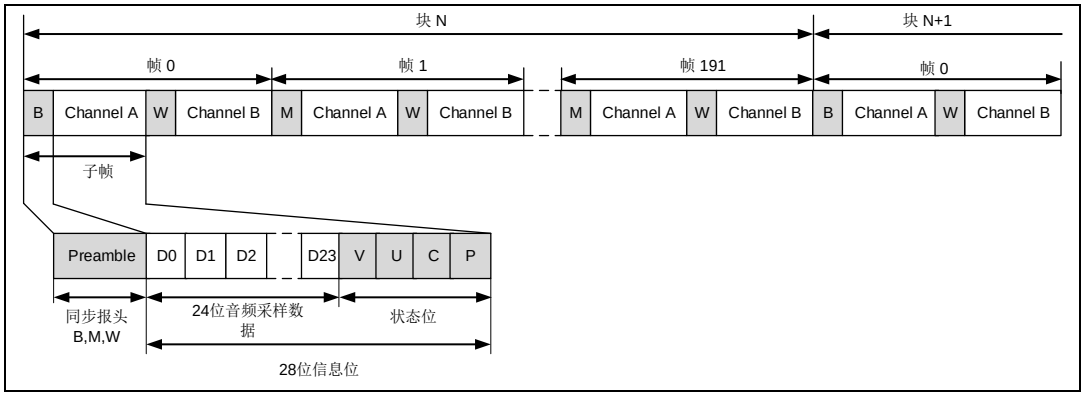
S/PDIF 块

一个S/PDIF块包含192帧，每帧包含2个子帧（左通道和右通道），即一个S/PDIF块包含384个子帧。每个子帧包含32位。块和子帧的数据格式如*表33-1. 子帧格式*及*图33-2. S/PDIF块及子帧格式*所示。

表 33-1. 子帧格式

比特位	描述
0~3	同步报头，有 3 种类型：B，M，W。 M 代表此时传送的是 A 通道（左声道），W 代表此时传输的是 B 通道（右声道），而 B 比较特殊，代表此时传送的是 A 通道（左声道），并且是一个块的起始子帧。
4~27	以线性 2 的补码表示的音频样本数据。第 27 位总是最高位。当使用 20 位编码范围时，第 8~27 位为音频样本数据，第 8 位为最低位。
28	有效位“V”，表示数据是否有效
29	用户数据位“U”，用户自定义
30	通道状态位“C”
31	奇偶位“P”，携带一个奇偶校验位，使 4 到 31 位包含偶数个 1 和偶数个 0。

图 33-2. S/PDIF 块及子帧格式



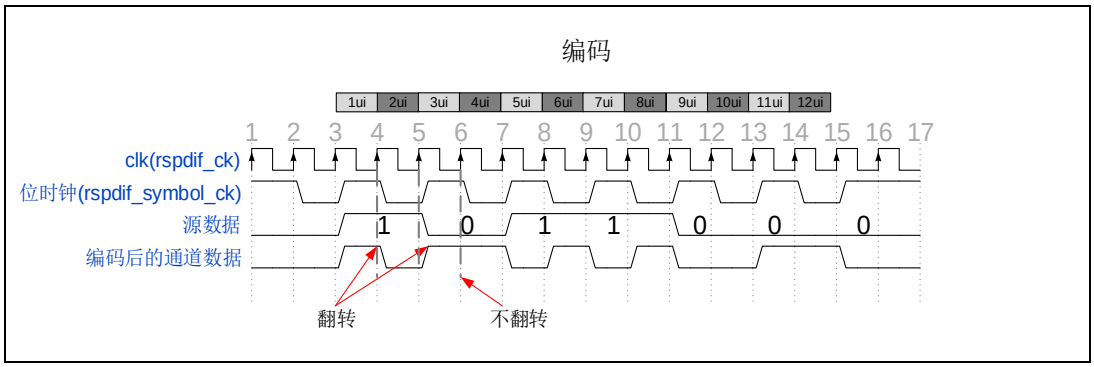
信息位编码

IEC60958 在传输数据时，位 4 到 31 使用双相符号编码（曼彻斯特协议）。其原理是使用一个两倍于传输位率的时钟频率做为基准，把原来一位数据拆成两份，每一个要传输的位用一个包含两个连续二进制状态的符号表示，当数据为 1 的时候，在其时钟周期内转变一次电位（0->1 或 1->0）让数据变成两个不同电位，变成 10 或 01，而当数据为 0 则不转变电位，变成 11 或 00。这些二进制状态在 IEC-60958 规范中称为“UI”（单位间隔）。

注意：信息位编码方式如下。符号表示由两个单位间隔组成的数据位。符号率表示*图33-3. 信息位编码示例*中位时钟的频率。RSPDIF模块使用的时钟频率是位时钟的两倍。24个数据位首

先传输LSB。

图 33-3. 信息位编码示例

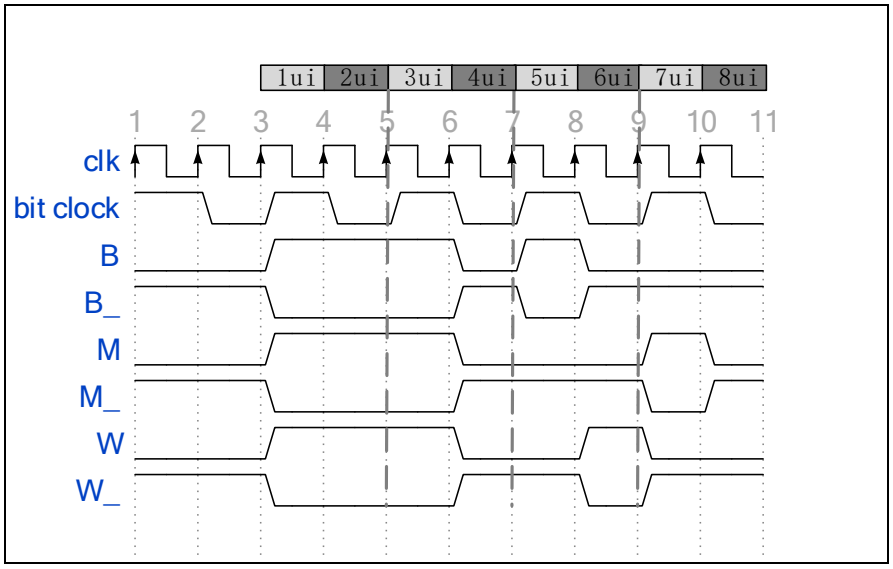


报头同步

根据前一个半比特值决定是否反转报头模式。这个前半比特值是在使能第一帧的第一个“B”报头启用传输之前的线路电平。对于其他报头，这个前半比特值是前一个子帧的奇偶校验位的第二个半比特。

报头模式B、M、W如*图33-4. S/PDIF报头*及*表33-2. 报头模式*所示。

图 33-4. S/PDIF 报头



注意：编码数据时，转换应该发生在第二个UI的末尾。然而，转换可能在第二个UI结束时缺少报头。

表 33-2. 报头模式

预先状态（前一个半比特值）	0	1	描述
报头	编码		
B	11101000	00010111	通道 A，且为一个块的起始子帧
W	11100100	00011011	通道 B
M	11100010	00011101	通道 A

33.3.3. RSPDIF

RSPDIF主要包括RSPDIF_SPP信号预处理和RSPDIF_DEC信号解码两部分，基于测量两个连续边缘之间的时间间隔来解码S/PDIF流。在S/PDIF流中可以找到三种时间间隔，如*表33-3. RSPDIF时间间隔*。

表 33-3. RSPDIF 时间间隔

时间间隔	描述
TL	长时间间隔，持续时间为 3 x UI，仅出现在报头中。
TM	中时间间隔，持续时间为 2 x UI，出现在一些报头或信息字段中。
TS	短时间间隔，持续时间为 1 x UI，出现在一些报头或信息字段中。

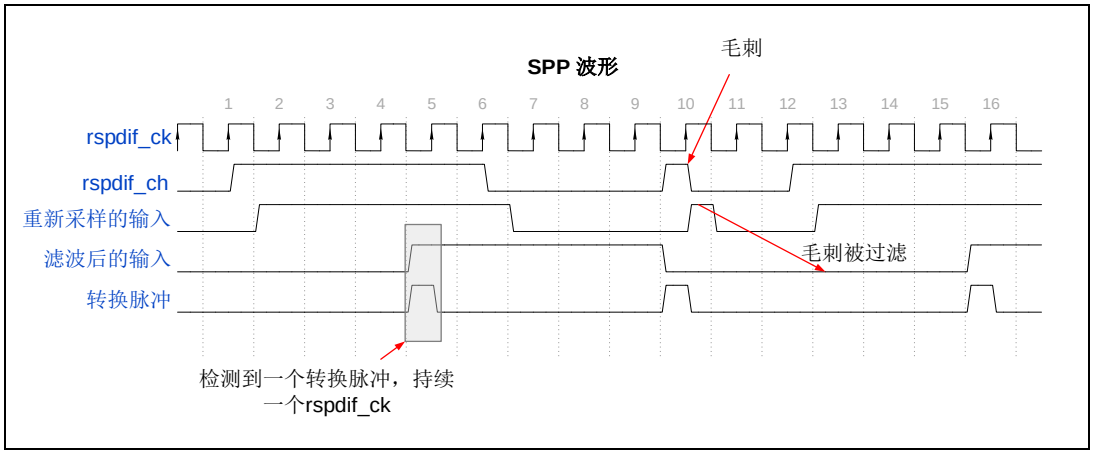
RSPDIF_SPP

RSPDIF_SPP 信号预处理阶段主要完成噪声的滤波和上升/下降边缘的检测。

RSPDIF 共支持四个输入信号，配置 RSPDIF_CTL 寄存器中 RXCHSEL[2:0]选择需要的输入。在选定的 RSPDIF_CH 上接收到 S/PDIF 信号使用 rspdif_ck 时钟重新采样。为了消除毛刺，RSPDIF 采用了一个简单的滤波。

这是通过检测边缘转换的阶段来实现的。边缘转换检测时，当采样到序列 0 后面跟着两个 1 时，即检测到上升边沿。当采样到序列 1 后面跟着两个 0 时，即检测到下降边沿。在上升边沿之后，预计将出现下降边沿序列。在下降边沿之后，预计将出现上升边沿序列。

图 33-5. 噪声滤波及上升\下降边沿检测



注意：转换脉冲是RSPDIF_DEC模块判断转换类型和正确解码输入位流的一个重要指标。

RSPDIF_DEC

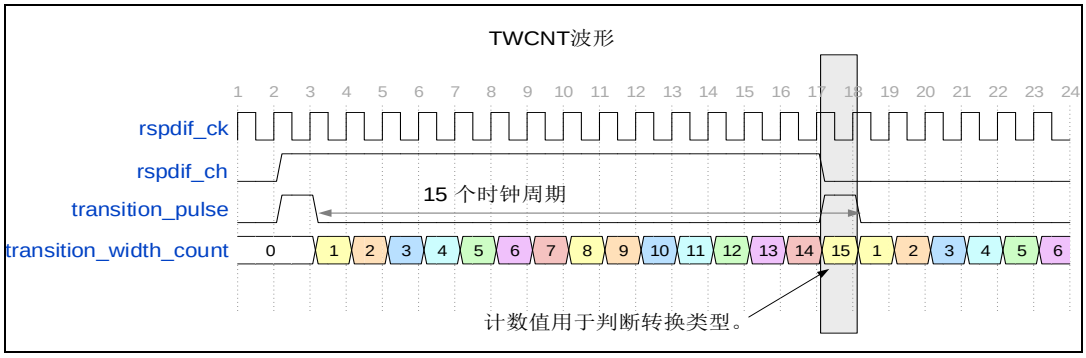
RSPDIF_DEC 模块检测报头类型并解码信息位，打包这些数据并写入 RX 缓存或 RSPDIF_CHSTAT 寄存器。RSPDIF_DEC 模块可完成时间间隔估计，符号速率和同步的估计，块和子帧报头的检测，解码数据及连续跟踪符号速率等功能。

TWCNT

RSPDIF_DEC 模块提供了一个 TWCNT 计数器，用于测量时间间隔持续时间。它由 rspdif_ck

信号计时。在每一个转换脉冲上，计数器值被存储并且计数器被重置以重新开始计数。若两个转换之间的时间间隔过长，TWCNT 溢出时，RSPDIF 将停止工作，此时 RSPDIF_STAT 寄存器的 TMOUTERR 标志被置位。转换定时器就像看门狗定时器，在传入信号经过 70 个转换后产生一个触发，计算 70 个转换确保了一个比一个子帧更长一点的延迟。

图 33-6. TWCNT 波形



注意：存储的计数器值等于两个转换脉冲之间的实际时钟周期。例如，如*图33-6. TWCNT波形*所示，两个转换脉冲之间的时间间隔是15个rspdif_ck时钟周期，所以存储的数字是15（不是14!）这样就方便了CKCNT5、WIDTH24和WIDTH40的计算。在TWCNT设计中，计数器值被重置为1（不是0!）。

SYNC

RSPDIF_DEC 模块提供了一个同步模块，具体请参考 *RSPDIF 同步过程*。同步阶段计算 WIDTH24 和 WIDTH40，它们是计算用于判断转换类型的阈值的参考。WIDTH24 和 WIDTH40 分别表示连续 24 个符号和 40 个符号的时间间隔时间。如上文所述，很明显：

$$WIDTH24 = 48\ UI \rightarrow TL_{LO} = 1.5\ UI = \frac{WIDTH24}{32} \quad (33\text{-}1)$$

$$WIDTH40 = 80\ UI \rightarrow TH_{HI} = 2.5\ UI = \frac{WIDTH40}{32} \quad (33\text{-}2)$$

TH_{HI} 和 TH_{LO} 是用于判断转换类型的阈值，这部分将在 DECODER 模块和 RSPDIF 同步过程的描述中详细介绍。WIDTH24 和 WIDTH40 也可以用来生成符号时钟，这将在 *RSPDIF 时钟管理*介绍。

此外，RSPDIF_STAT寄存器中CKCNT5用于估计S/PDIF符号率，也在SYNC阶段中计算。

DECODER

RSPDIF_DEC 模块提供了一个解码器模块，具有转换编码器和报头检测器的功能。解码输出报头类型（PREF）、28 位信息位和状态位（C、U、V、P）。在 RSPDIF_CH 位流的解码器过程中，可生成一个恢复的符号时钟。解码器块从 TWCNT 计数器接收当前转换宽度，该块通过比较当前转换宽度与两个不同的阈值（TH_{HI} 和 TH_{LO}）来编码当前转换宽度，具体如*表 33-4. 转换编码器编码规则*。

表 33-4. 转换编码器编码规则

转换宽度 TH	编码值	描述
$TH < (TH_{LO} - 1)$	TS	接收的数据是数据位“1”的一半

转换宽度 TH	编码值	描述
(TH_{LO} – 1)< TH <TH_{HI}	TM	接收的数据为数据位“0”
TH > TH_{HI}	TL	接收的数据为报头的长脉冲
其他	-	FRERR 置位

TH_{HI}和TH_{LO}的计算方法见*表33-5. 临界值计算*。

表 33-5. 临界值计算

临界值	同步阶段
TH_{LO}	WIDTH24 / 32
TH_{HI}	WIDTH40 / 32

RSPDIF同步的详细信息请参见*图33-7. 同步流程图*。

TH_{LO}理想情况下等于1.5个UI，TH_{HI}等于2.5个UI。

报头检测器

解码器提供报头检测的功能，检查特定序列的四个连续转换，以确定它们是否构成报头部分。假设TRANS0、TRANS1、TRANS2和TRANS3代表上述编码的四个连续转换。这四个转换的值如*表33-6. 报头的转换序列*所示。缺少这种模式表明这些转换不构成报头，而是构成子帧数据的一部分并且可用双相解码器解码。

表 33-6. 报头的转换序列

报头格式	双相编码数据	TRANS3	TRANS2	TRANS1	TRANS0
B	11101000	TL	TS	TS	TL
M	11100010	TL	TL	TS	TS
W	11100100	TL	TM	TS	TM

双相解码器

双相解码器使用由转换编码器及报头检测器提供的转换信息来解码输入的双相符号数据流。当报头已正确检测到后，双相解码器将解码接下来的数据信息，见*表 33-7. 双相解码器解码规则*。

表 33-7. 双相解码器解码规则

输入信息	解码值
TM	0
两个连续的 TS	1
其他	FRERR 置位

33.3.4. RSPDIF 同步过程

当将RSPDIF_CTL寄存器中RXCFG[1:0]设置为2'b01或2'b11时，同步阶段开始。如果设置RSPDIF_CTL寄存器的位WFRXA为1，则在同步之前，RSPDIF将首先检测选定的RSPDIF_CH线的活动。仅当选定的RSPDIF_CH线上检测到四个转换时，才切换至同步阶段。该功能能有效避免同步错误。同步流程如*图33-7. 同步流程图*所示。该功能在RSPDIF_DEC模块中实现。

同步阶段在 RSPDIF 正确解码后，计算得到精确的同步阈值。当 RSPDIF 能够正确测量 24 和

40 个连续符号的持续时间时，同步阶段完成，阈值就会更新，标志 SYNDO 被设置为 1。

RSPDIF_CH 线上可能存在干扰，可能会发生同步过程没有正确执行的情况。RSPDIF 提供了在同步之前可设置同步重试次数（MAXRT）的功能。直到到达设置的 MAXRT 次数，同步还未正确执行，SYNERR 错误标志置位。如果 RSPDIF_CH 线上没有有效的 SPDIF 数据流，TWCNT 溢出，TMOUTERR 错误标志置位。

同步完成后，当检测到下一个报头“B”，RSPDIF 开始接收通道状态（C）和用户数据（U）（见*图 33-11. 通道/用户数据格式*）。用户通过 RSPDIF_CHSTAT 寄存器读取，并根据读取到的 C 和 U，配置 RXDF[1:0]和 RXSTEOMEN 位。

注意：当 RXCFG[1:0] = 2'b11 时，对 RXDF[1:0]和 RXSTEOMEN 位的修改无效。可参考*表 33-9. 不同 RSPDIF 状态下 RSPDIF_CTL 比特位访问特性*。

图 33-7. 同步流程图

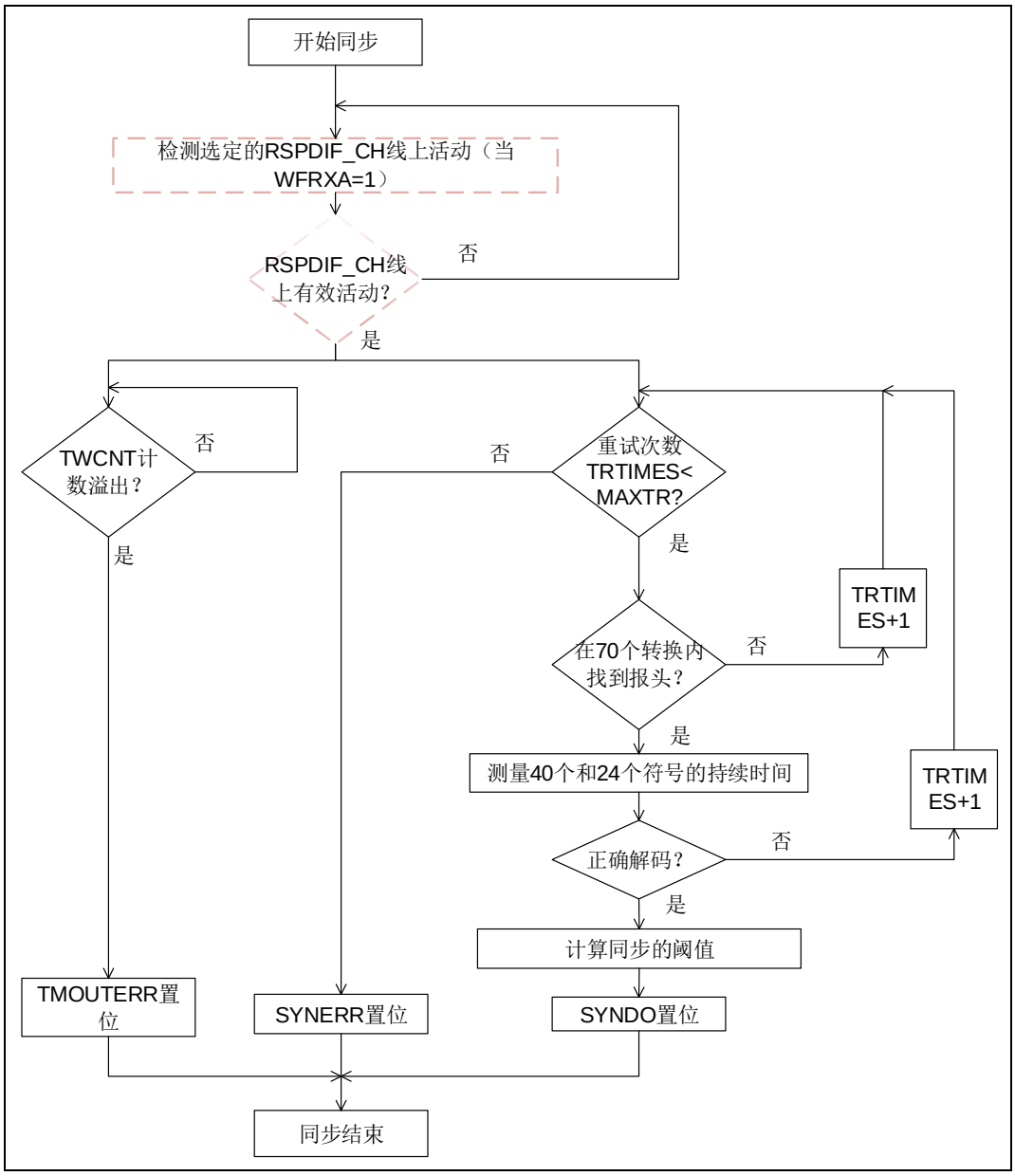
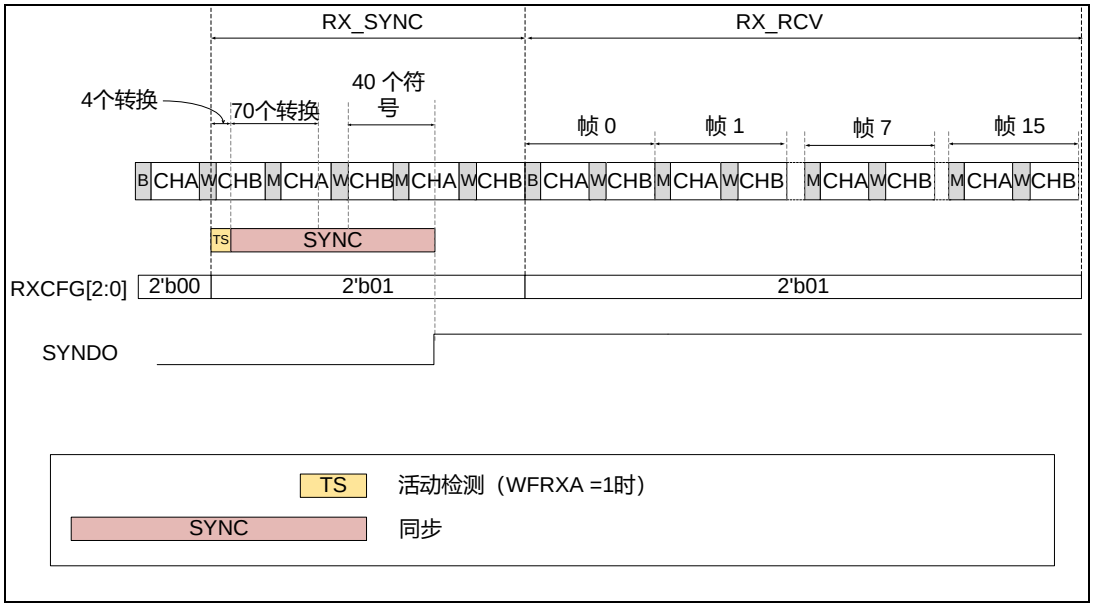


图 33-8. 同步过程时序



33.3.5. RSPDIF 状态机

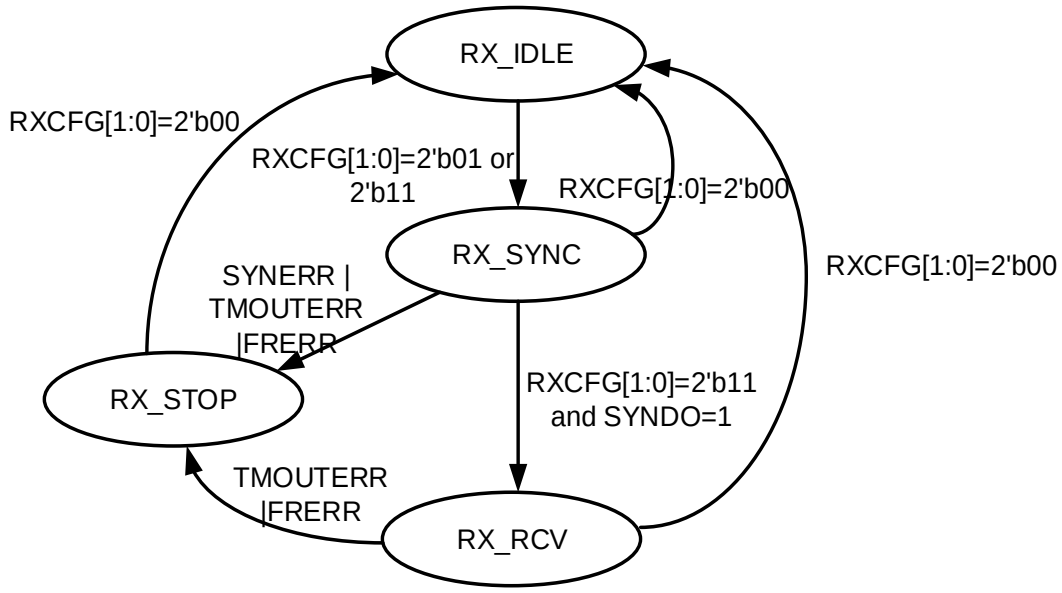
RSPDIF状态机包括以下状态，如*表33-8. RSPDIF状态*。

表 33-8. RSPDIF 状态

RSPDIF 状态	描述
RX_IDLE	空闲状态，此时 RSPDIF 禁能，rspdif_ck 域复位，rspdif_pclk 域功能正常。
RX_SYNC	同步状态，RSPDIF 与数据流同步，阈值定期更新，可以通过中断或 DMA 获取用户数据和通道状态。
RX_RCV	接收状态：RSPDIF 与数据流同步，阈值定期更新，可以通过中断或 DMA 通道获取用户，通道状态和音频数据。当 RXCFG[1:0]转为 2’b11 时，RSPDIF 在开始保存音频数据之前等待“B”报头。
RX_STOP	停止状态：RSPDIF 不再同步，用户和通道状态和音频数据接收停止。

*图33-9. RSPDIF状态机*显示了RSPDIF状态之间如何转换。

图 33-9. RSPDIF 状态机



RX_IDLE

状态机将从以下条件进入RX_IDLE状态：

- 设置 RXCFG[1:0]为 2’b00。
- RX_SYNC 时，设置 RXCFG[1:0]为 2’b00。
- RX_RCV 时，设置 RXCFG[1:0]为 2’b00。
- RX_STOP 时，设置 RXCFG[1:0]为 2’b00。

RX_IDLE 的下一个状态是：

- RX_SYNC：设置 RXCFG[1:0]为 2’b01 或 2’b11。

注意：软件可以在任何时候将 RXCFG[1:0]设置为 0，RSPDIF 立即返回 RX_IDLE 状态。如果 DMA 传输正在进行中，它将等待 DMA 传输完成。

RX_SYNC

状态机将从以下条件进入RX_SYNC状态：

- RX_IDLE 时，设置 RXCFG[1:0]为 2’b01 或 2’b11。

RX_SYNC 的下一个状态是：

- RX_IDLE：设置 RXCFG[1:0]为 2’b00。
- RX_RCV：设置 RXCFG[1:0]为 2’b11 且同步成功完成 SYNDO = 1。
- RX_STOP：同步失败（若设置了同步重试次数，也已经达到设置的最大值）或者接收到的数据没有正确被解码（FRERR 或 SYNERR 或 TMOUTERR = 1）。

注意：当同步阶段完成时，如果 RXCFG[1:0] = 2'b01，外设仍然处于这个状态。

RX_RCV

状态机将从以下条件进入RX_RCV状态：

- RX_SYNC 时，设置 RXCFG[1:0]为 2'b11 且同步成功完成 SYNDO = 1。

RX_RCV 的下一个状态是：

- RX_IDLE：设置 RXCFG[1:0]为 2'b00。
- RX_STOP：接收到的数据没有正确被解码（FRERR 或 TMOUTERR = 1）。

RX_STOP

状态机将从以下条件进入RX_STOP状态：

- RX_SYNC 时，同步失败（若设置了同步重试次数，也已经达到设置的最大值）或者接收到的数据没有正确被解码（FRERR 或 SYNERR 或 TMOUTERR = 1）。
- RX_RCV 时，接收到的数据没有正确被解码（FRERR 或 TMOUTERR = 1）。

RX_STOP 的下一个状态是：

- RX_IDLE：设置 RXCFG[1:0]为 2'b00。

注意：当 RXCFG[1:0]设置为 0 时，硬件 IP 关闭，即复位所有状态机，并刷新 RX 缓存。标志 FRERR, SYNERR 和 TMOUTERR 被重置。

不同状态下 RSPDIF_CTL 寄存器访问权限

在RSPDIF状态机的不同的状态阶段，对RSPDIF_CTL寄存器的不同位域的访问权限不同，具体请参考*表33-9. 不同RSPDIF状态下RSPDIF_CTL比特位访问特性*。硬件这样处理，可以避免对RSPDIF_CTL寄存器的错误配置。请注意，即使在RSPDIF任何状态下，都可以修改PTNCPEN\CUNCPEN\VNCPEN\PNCPEN位，但这些修改不会影响已经保存到DATA寄存器中的值了。

表 33-9. 不同 RSPDIF 状态下 RSPDIF_CTL 比特位访问特性

RSPDIF 状态	RXCHSEL[2:0]	WFRXA	MAXRT[1:0]	CFCHSEL	DMACBEN	PTNCPEN	CUNCPEN	VNCPEN	PNCPEN	RXDF[1:0]	RXSTEOMEN	DMAREN
RX_IDLE (RXCFG[1:0] = 2'b00)	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RX_SYNC (RXCFG[1:0] = 2'b01)	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw
RX_RCV (RXCFG[1:0] =	r	r	r	r	rw	rw	rw	rw	rw	r	r	rw

RSPDIF 状态	RXCHSEL[2:0]	WFRXA	MAXRT[1:0]	CFCHSEL	DMACBEN	PTNCPEN	CUNCPEN	VNCPEN	PNCPEN	RXDF[1:0]	RXSTEOMEN	DMAREN
2'b11)												

33.3.6. RSPDIF 数据接收管理

RSPDIF 数据双缓冲区

RSPDIF实现了数据接收时的双缓冲功能。数据接收双缓冲区由一个32位的RX缓存和RSPDIF_DATA寄存器构成。当RSPDIF_DATA寄存器为空且已检测到奇偶校验位（P）和下一个报头之间的转换时，RX缓存中的数据将立即转移到RSPDIF_DATA寄存器中。

数据的双缓冲区机制提高了对延迟的容忍度，现允许的最大延迟是T_{SAMPLE} - $2T_{pclk}$ - $2T_{rspdif_ck}$（T_{SAMPLE}为所接收的立体声音频样本的音频采样率，T_{PCLK}为rspdif_pclk时钟的周期，T_{rspdif_ck}为rspdif_ck时钟的周期。）

软件可以通过读RSPDIF_DATA寄存器或者DMA方式获取接收到的数据。推荐DMA操作，详细请参阅*DMA功能*。

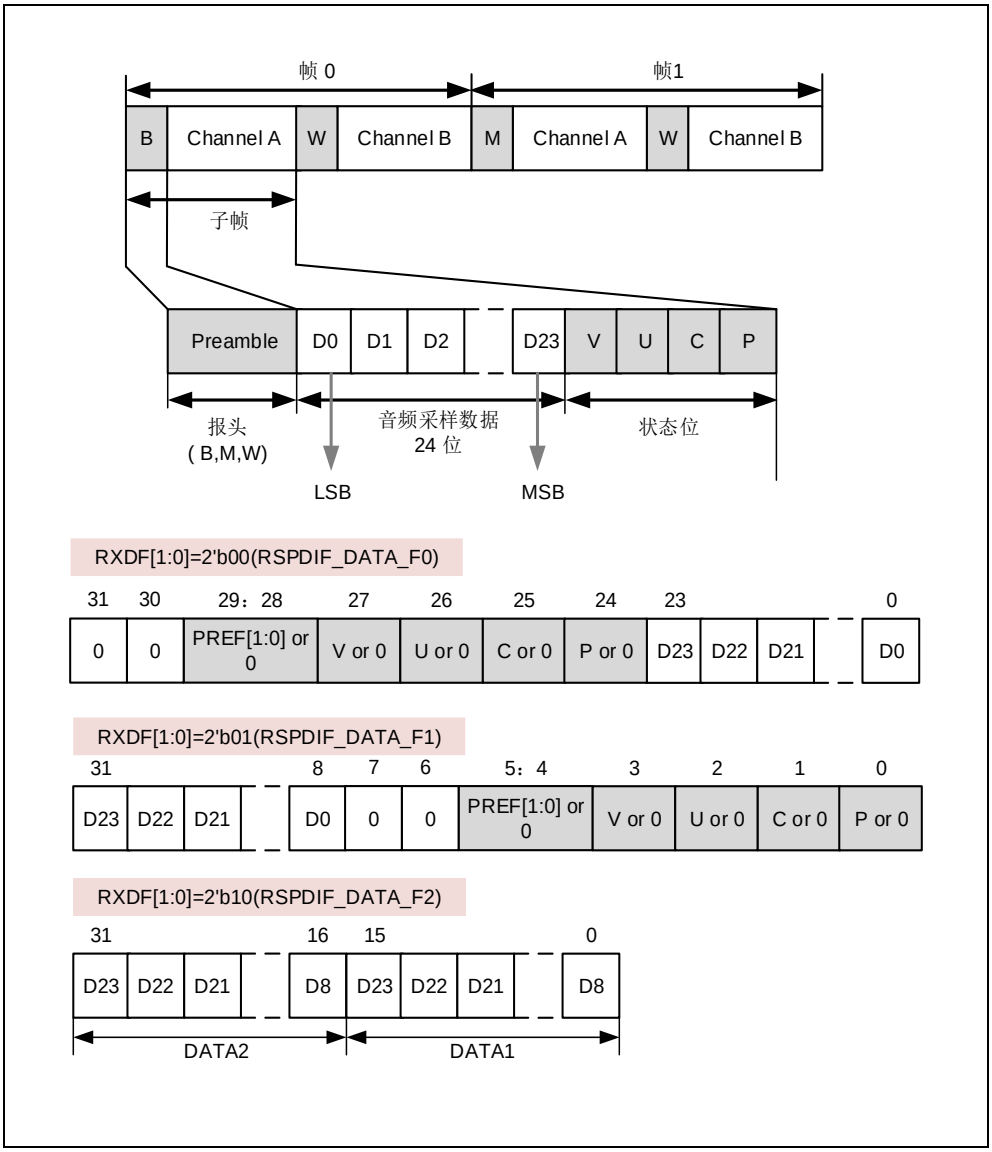
RSPDIF_DATA 寄存器

除V，U，C，P状态位，和报头，每个子帧最多包括24位的数据。RSPDIF_DATA寄存器根据RSPDIF_CTL寄存器中RXDF[1:0]的值不同，有三种格式，可对接收到的音频数据流有三种处理方式：

- 当 RXDF[1:0] = 2'b00 时，数据寄存器的格式为 RSPDIF_DATA_F0 所描述的格式，强制数据右对齐，PREF\C\U\V\P 位可根据需要启动或者强制为 0。
- 当 RXDF[1:0] = 2'b01 时，数据寄存器的格式为 RSPDIF_DATA_F1 所描述的格式，强制数据左对齐，PREF\C\U\V\P 位可根据需要启动或者强制为 0。
- 当RXDF[1:0] = 2'b11时，数据寄存器的格式为RSPDIF_DATA_F2所描述的格式。该格式在非线性模式下，每个子帧仅使用16位。两个连续子帧的数据将存放在一个RSPDIF_DATA寄存器中。该格式PREF\C\U\V\P位和数据位不能混合，但软件仍可以通过读取RSPDIF_CHSTAT寄存器或者专用的DMA通道获取。当RXSTEOMEN = 1时，没有不对齐的风险（即来自通道A的数据总是存储在RSPDIF_DATA[31:16]中）。如果RXSTEOMEN = 0，则存在超量情况下的错位风险。在这种情况下，RSPDIF_DATA[31:16]总是包含最老的值，而RSPDIF_DATA[15:0]总是包含最近的值（如*图33-12. RSPDIF上溢错误（RXSTEOMEN = 0且RXDF[1:0] = 2'b0x）*和*图33-13. RSPDIF上溢错误（RXSTEOMEN = 0且RXDF[1:0] = 2'b10）*所示）。

注意：本文档描述了3个数据寄存器：RSPDIF_DATA_Fx（x = 0,1,2），但实际上只有一个物理数据寄存器。

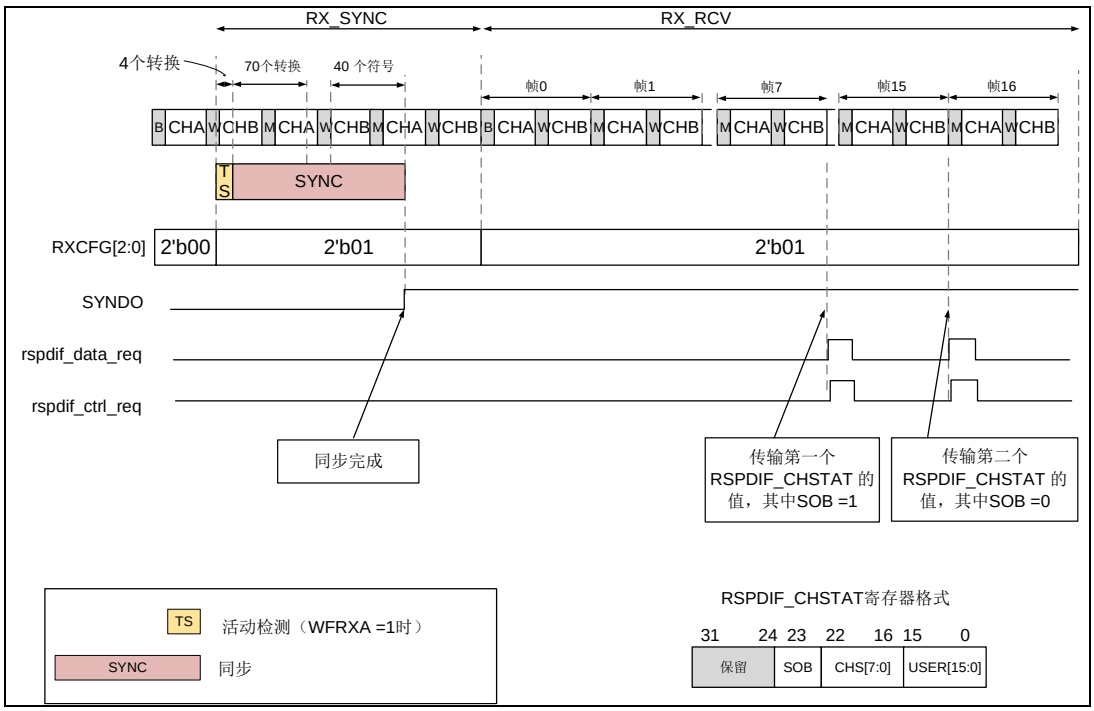
图 33-10. RSPDIF_DATA 寄存器格式



RSPDIF 特定控制流

RSPDIF提供了接收用户数据U和信道状态C的专用通道。通过配置RSPDIF_CTL寄存器中CFCHSEL位，可需选择信道状态位C时从通道A还是通道B中获取。当RXCFG[1:0]设置为2'b01或2'b11时，同步阶段结束后开始采集。当接收到8个信道状态位和16个用户数据位时，将它们打包存储到RSPDIF_CHSTAT寄存器中。若配置RSPDIF_CTL寄存器DMACBNE位为1，使能了控制流的DMA功能，则触发一个DMA请求。如果CHS[0]是新块的第一个状态位，则SOB位设为1。请参考*接收通道状态寄存器（RSPDIF_CHSTAT）*。

图 33-11. 通道/用户数据格式



混合数据和控制流

当RXDF[1:0] = 2'b00或2'b01时，RSPDIF_DATA寄存器格式为RSPDIF_DATA_F0或RSPDIF_DATA_F1，该格式允许对V，U，C，P状态位，报头和数据混合接收管理。通过配置RSPDIF_CTL寄存器，用户可以灵活选择将V，U，C，P状态位，报头的哪些内容保存到RSPDIF_DATA寄存器中，如*表33-10. 混合数据和控制流*。

表 33-10. 混合数据和控制流

位域	值	描述
PNCPEN	1	奇偶校验错误信息被屏蔽（设置为 0）
	0	奇偶校验错误信息被复制到 RSPDIF_DATA 中
VNCPEN	1	有效性信息被屏蔽
	0	有效性信息被复制到 RSPDIF_DATA 中
CUNCPEN	1	通道和用户信息被屏蔽
	0	通道和用户信息被复制到 RSPDIF_DATA 中
PTNCPEN	1	报头类型被屏蔽
	0	报头类型被复制到 RSPDIF_DATA 中

33.3.7. RSPDIF 时钟管理

由RSPDIF框图*图33-1. RSPDIF模块框图*可知，RSPDIF块需要用于寄存器接口rspdif_pclk时钟和用于RSPDIF DEC模块的rspdif_ck时钟。rspdif_ck不应该是锁相的，在SYNC块中，所有穿过这些时钟域的信号都被重新同步。

为了正确解码传入的S/PDIF流，解码rspdif_ck时钟应至少高于最大符号率11倍或高于音频采样率704倍的时钟重新采样所接收的数据。例如，如果音频速率为192KHz，则符号率为12.288MHz（音频速率*通道数*子帧位数），则至少需要135.2MHz的解码时钟（符号率*11）。

对于rspdif_pclk，rspdif_pclk不能小于符号速率。

表 33-11. 最小 rspdif_ck 频率与音频采样率

符号率	最小 rspdif_ck 频率	音频
3.072 MHz	33.8 MHz	48 kHz 数据流
6.144 MHz	67.6 MHz	96 kHz 数据流
12.288 MHz	135.2 MHz	192 kHz 数据流

在RSPDIF模块中，

RSPDIF_DEC：解码器被设计解码输入RSPDIF输入位流。当RSPDIF未使能时，解码器模块的时钟被门控。该模块的时钟源是rspdif_ck。

RSPDIF 符号时钟

RSPDIF在解码时，使用WIDTH24、WIDTH40和符号边界的值构建了一个符号时钟rspdif_symbol_ck。它可以用作其他音频设备（如SAI或I2S）的参考内核时钟，可用于RSPDIF到I2S桥接功能。

在接收子帧同步报头时，WIDTH24和WIDTH40值构建了符号时钟的下降沿和上升沿。当RSPDIF为RX_STOP或RX_IDLE时，WIDTH24和WIDTH40也被用于生成符号时钟。在接收子帧时，RSPDIF使用符号边界生成上升边沿，WIDTH24和WIDTH40值生成下降边沿，符号时钟的占空比接近50%。但是，当RSPDIF从用WIDTH24和WIDTH40生成符号时钟转换到由符号边界生成符号时钟时，符号时钟占空比可以改变，反之亦然。符号时钟生成模式发生转换时或使用rspdif_ck时钟对S/PDIF信号的重新采样时，符号时钟可能会产生大幅抖动。

配置RSPDIF_CTL寄存器中SCKEN位可使能或禁能符号时钟的生成。配置RSPDIF_CTL寄存器中BKSCKEN位可使能或禁能备份符号时钟的生成。但是，当标志SYNERR置位时，符号时钟和备份时钟都不能生成，因为同步出错。当SCKEN和BKSCKEN都设置为1时，当RSPDIF从RX_SYNC或RX_RCV切换到RX_STOP或RX_IDLE时，符号时钟将失去一些转换。符号时钟的具体生成条件请参考*表33-12. 符号时钟spdif_symbol_ck生成条件*。

表 33-12. 符号时钟 spdif_symbol_ck 生成条件

RSPDIF 状态	SCKEN	BKSCKEN	RSPDIF CH 接收到有效数据	存在 WIDTH40 和 WIDTH24 的有效值	WIDTH40 和 WIDTH24 包含来自上一次同步的有效值	RSPDIF CH 未检测到转换	完成同步（SYNDO = 1）	符号时钟状态
所有状态	0	x	x	x	x	x	x	输出关闭
RX_IDLE	1	0	x	x	x	x	1	无输出
		1	x	0	x	x	x	
	1	1	x	1	x	x	x	有输出
RX_SYNC	1	0	x	x	x	x	0	无输出
		1	x	0	x	x	0	
	1	0	x	x	x	x	1	有输出
		1	x	x	x	x	1	
			x	x	1	x	0	

RSPDIF 状态	SCKEN	BKSCKEN	RSPDIF CH 接收到有效数据	存在 WIDTH40 和 WIDTH24 的有效值	WIDTH40 和 WIDTH24 包含来自上一次同步的有效值	RSPDIF CH 未检测到转换	完成同步（SYNDO = 1）	符号时钟状态
RX_REV	1	0	x	x	x	1	x	无输出
		1	x	x	x	1	x	
	1	0	1	x	x	x	x	有输出
		1	1	x	x	x	x	
RX_STOP	1	0	x	x	x	x	x	无输出
		1	x	0	x	x	x	
	1	1	x	1	x	x	x	有输出

注意：“0”表示该条件必须不满足；“1”表示该条件必须满足；“x”表示该条件对符号时钟的生成与否无影响。

33.3.8. DMA 功能

RSPDIF可以通过对应的专用的DMA通道获取数据和控制信息。具体使用通道请参考*直接存储器访问控制器（DMA）*。设置RSPDIF_CTL寄存器中的DMAREN位，配置RSPDIF块进行数据的DMA传输。当RSPDIF_DATA不空时，就会向DMA发送一个数据传输请求，DMA将直接读取数据。设置RSPDIF_CTL寄存器中的DMACBNE位，配置RSPDIF块进行通道及用户信息的DMA传输。关于控制数据的DMA使用，请参阅***RSPDIF数据接收管理***。不推荐在已经开始接收数据后，再配置DMAREN或DMACBNE位来启用DMA传输。

33.3.9. 状态、错误和中断

状态

接收缓冲区非空（RBNE）

当RX缓存非空时，RBNE位置位。表示此时接收到一个数据，当RSPDIF_DATA寄存器为空且已检测到奇偶校验位（P）和下一个报头之间的转换时，存入RSPDIF_DATA，软件可通过读取RSPDIF_DATA获取此数据。

控制流接收缓冲区非空（CBNE）

当RSPDIF_CHSTAT非空时，CBNE位置位。表示此时接收到控制流数据，并存入RSPDIF_CHSTAT，软件可通过读取RSPDIF_CHSTAT获取此数据。

同步完成（SYNDO）

当RSPDIF能够正确测量24和40个连续符号的持续时间时，SYNC完成时，该位置位。表示同步完成。

新块开始（SYNDB）

当RSPDIF已检测到报头B时，该位置位。表示为一个block的起始。

注意：SYNDB 事件只能在 RSPDIF 与输入流同步时（SYNDO）才能发生。

错误

帧错误

当28个信息位中出现转换序列不正确时或报头出现在意外的位置或预期的报头没有收到时，RSPDIF帧错误发生，FRERR位置位。如果RSPDIF_INTEN寄存器的RXDCERRIE位置位，则会产生一个相应的中断。设置RXCFG[1:0]为2'b00，可清除该错误标志。

同步错误

当同步失败时（若设置了最大重试次数MAXTR，也已经超过MAXTR），RSPDIF同步错误发生，SYNERR位置位。如果RSPDIF_INTEN寄存器的RXDCERRIE位置位，则会产生一个相应的中断。设置RXCFG[1:0]为2'b00，可清除该错误标志。

超时错误

当在rspdif_ck时钟的8192周期期间没有检测到转换，TWCNT发生溢出时，RSPDIF超时错误发生，TMOUTERR位置位。如果RSPDIF_INTEN寄存器的RXDCERRIE位置位，则会产生一个相应的中断。设置RXCFG[1:0]为2'b00，可清除该错误标志。

奇偶校验错误

当在一个子帧的28个信息位中，0和1的个数不是偶数个时，RSPDIF奇偶校验错误发生，PERR位置位。如果RSPDIF_INTEN寄存器的PERRIE位置位，则会产生一个相应的中断。设置PERRC位为1，可清除PERR标志。

注意：即使中断挂起，接收传入的数据也不会被暂停，RSPDIF也会继续向RSPDIF_DATA发送数据。如果软件想要保证在RSPDIF_DATA寄存器中读取的数据和PERR位的值之间的一致性，PNCPEN位必须设置为0。

上溢错误

当RSPDIF_DATA和RX缓存都已满，同时RSPDIF_DEC仍向RX缓存中写入一个新的数据时，RSPDIF上溢错误发生，RXORERR位置位，且这个新数据也并不会写入缓存。如果RSPDIF_INTEN寄存器的RXORERRIE位置位，则会产生一个相应的中断。设置RXORERRC位为1，可清除RXORERR标志。

注意：即使RXORERR标志挂起，当RXSTEOMEN = 0，RX缓存为空时，下一个传入的数据仍会被存储。参见*图33-12. RSPDIF上溢错误（RXSTEOMEN = 0且RXDF[1:0] = 2'b0x）*和*图33-13. RSPDIF上溢错误（RXSTEOMEN = 0且RXDF[1:0] = 2'b10）*。

图 33-12. RSPDIF 上溢错误（RXSTEOMEN = 0 且 RXDF[1:0] = 2’b0x）

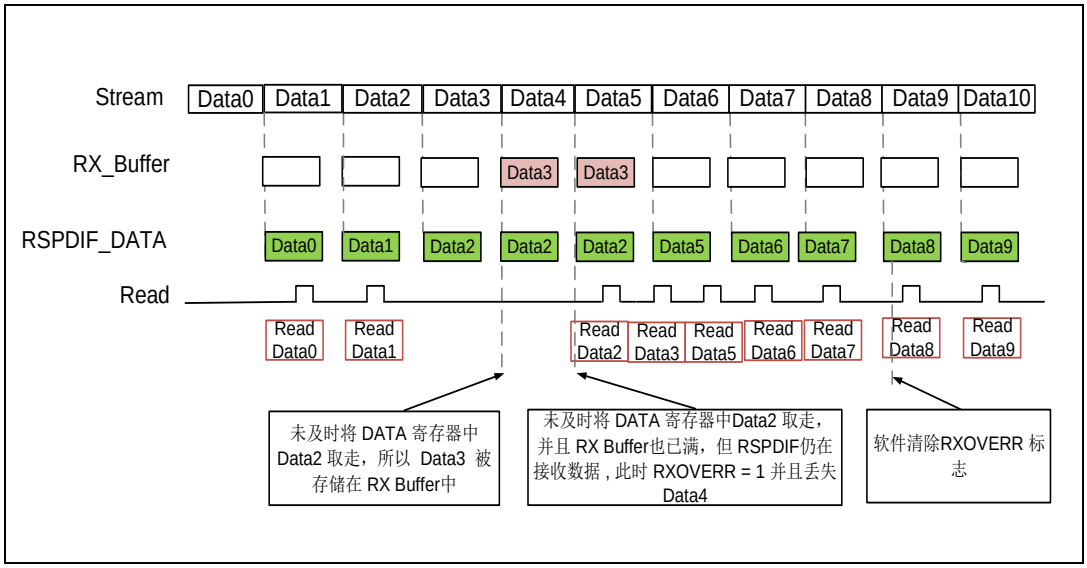
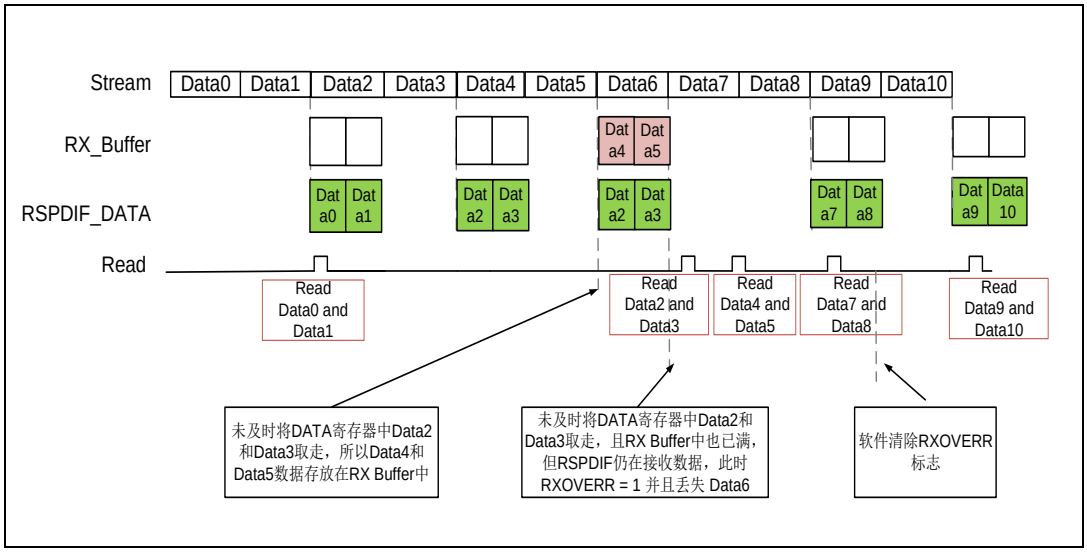


图 33-13. RSPDIF 上溢错误（RXSTEOMEN = 0 且 RXDF[1:0] = 2’b10）



如果RXSTEOMEN = 1，RSPDIF传输的是立体声数据，那么为了避免左右通道不对准的问题，RSPDIF必须丢弃第二个数据，即使RX缓存内部有空间。之后再传入的数据可以正常写入RX缓存，即使RXORERR标志仍然挂起。参见 *图33-14. RSPDIF上溢错误（RXSTEOMEN = 1且RXDF[1:0] = 2’b0x）*和*图33-15. RSPDIF上溢错误（RXSTEOMEN = 1且RXDF[1:0] = 2’b10）*所示。

图 33-14. RSPDIF 上溢错误（RXSTEOMEN = 1 且 RXDF[1:0] = 2’b0x）

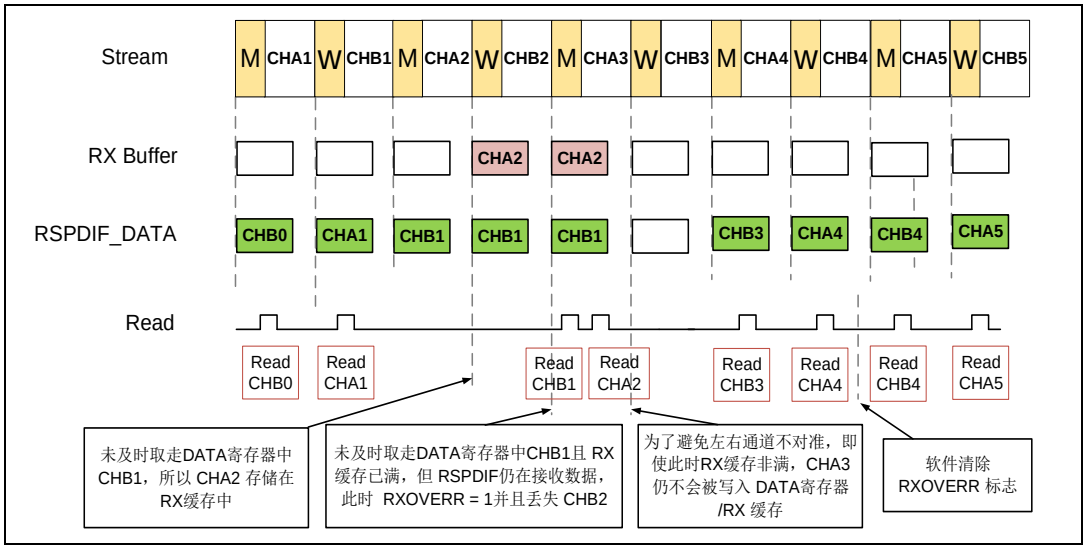
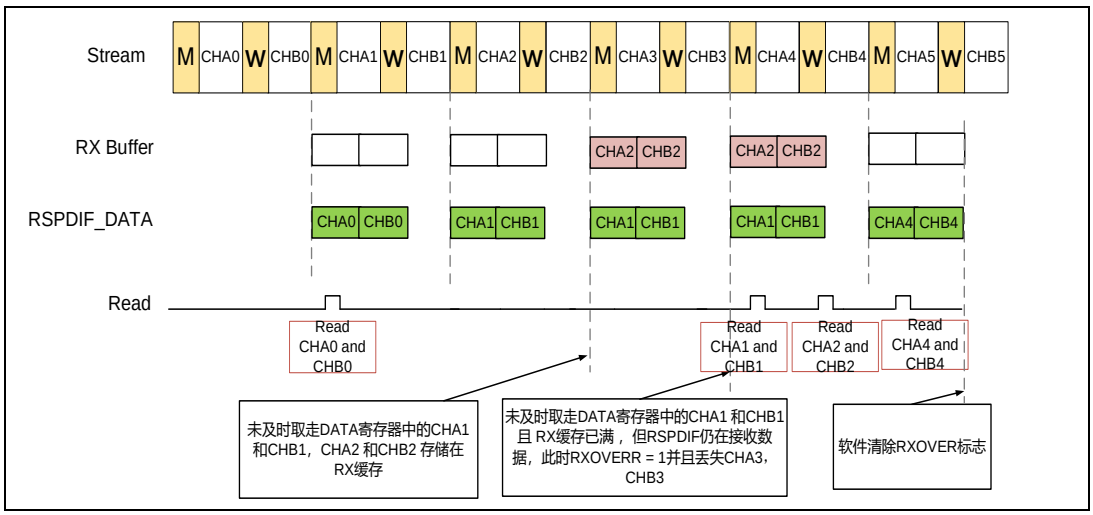


图 33-15. RSPDIF 上溢错误（RXSTEOMEN = 1 且 RXDF[1:0] = 2’b10）



中断

RSPDIF 模块的中断及事件标注如*表 33-13. RSPDIF 中断事件*。

表 33-13. RSPDIF 中断事件

事件标志	中断事件	清除方式	中断使能位
SYNDO	同步完成	设置 RSPDIF_STATC 寄存器 SYNDBC 位为 1	SYNDOIE
RBNE	RSPDIF 接收数据寄存器非空	读 RSPDIF_DATA 寄存器	RBNEIE
PERR	子帧中 28 个信息位内 0 和 1 不是偶数个	设置 RSPDIF_STATC 寄存器 PERRC 位为 1	PERRIE
RXORERR	RSPDIF_DATA 和 RX 缓存都已满，且继续向 RX 缓存中写入一个新数据	设置 RSPDIF_STATC 寄存器 RXORERRC 位为 1	RXORERRIE
CBNE	RSPDIF 控制流接收寄存器非空	读 RSPDIF_CHSTAT 寄存器	CBNEIE

事件标志	中断事件	清除方式	中断使能位
SYNDB	当已检测到报头 B	设置 RSPDIF_STATC 寄存器 SYDBC 位为 1	SYNDBIE
SYNERR	同步失败	设置 RSPDIF_CTL 寄存器 RXCFG[1:0]位为 2'b00	RXDCERRIE
FRERR	当 28 个信息位中出现转换序列不正确时或报头出现在意外的位置或预期的报头没有收到时		
TMOUTERR	TWCNT 发生溢出		

33.4. RSPDIF 寄存器

RSPDIF基地址：0x4000 4000

33.4.1. 控制寄存器（RSPDIF_CTL）

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留										BKSCKEN	SCKEN	保留	RXCHSEL[2:0]		
										rw	rw			rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	WFRXA	MAXRT[1:0]		CFCHSEL	DMACBNE	PTNCPEN	CUNCPEN	VNCPEN	PNCPEN	RXDF[1:0]		RXSTEOMEN	DMAREN	RXCFG[1:0]	
	rw	rw		rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	

位/位域	名称	描述
31:22	保留	必须保持复位值。
21	BKSCKEN	备份符号时钟使能位 软件置 1 和清 0。 0：RSPDIF 不生成备用符号时钟。 1：当 SCKEN = 1，RSPDIF 生成一个备用符号时钟。 请参考 ***RSPDIF 时钟管理***如何编程该区域。
20	SCKEN	符号时钟使能位 软件置 1 和清 0。 0：RSPDIF 不生成符号时钟。 1：RSPDIF 生成一个符号时钟。 请参考 ***RSPDIF 时钟管理***如何编程该区域。
19	保留	必须保持复位值。
18:16	RXCHSEL[2:0]	RSPDIF 输入通道选择 000：选择 RSPDIF_CH0 001：选择 RSPDIF_CH1 010：选择 RSPDIF_CH2 011：选择 RSPDIF_CH3 100 ~ 111：保留
15	保留	必须保持复位值。
14	WFRXA	等待选定的 RSPDIF 通道的 4 个有效转换信号 软件置 1 和清 0。 0：在启动同步进程之前，RSPDIF 不等待选定的 RSPDIF 通道的有效转换信号。

		1：在启动同步进程之前，RSPDIF 等待选定的 RSPDIF 通道的有效转换信号。
13:12	MAXRT[1:0]	RSPDIF 同步阶段允许的最大重试次数 00：不允许重试（仅一次机会） 01：允许最多 3 次重试 10：允许最多 15 次重试 11：允许最多 63 次重试
11	CFCHSEL	控制流获取通道状态源选择 软件置 1 和清 0。 0：从通道 A 获取通道状态 1：从通道 B 获取通道状态
10	DMACBNE	控制流的控制缓冲区 DMA 使能位 软件置 1 和清 0。 0：禁能控制流 DMA 模式。 1：使能控制流 DMA 模式。 当该位置位时，每当 CBNE 标志置位就会生成 DMA 请求。
9	PTNCPEN	报头类型位域不复制使能位 软件置 1 和清 0。 0：将报头类型位域复制到 RSPDIF_DATA 中。 1：不将报头类型位域复制到 RSPDIF_DATA 中，而是写 0 来替代。
8	CUNCPEN	通道信息位和用户信息位不复制使能位 软件置 1 和清 0。 0：将通道信息位和用户信息位复制到 RSPDIF_DATA 中。 1：不将通道信息位和用户信息位复制到 RSPDIF_DATA 中，而是写 0 来替代。
7	VNCPEN	有效位不复制使能位 软件置 1 和清 0。 0：将有效位复制到 RSPDIF_DATA 中。 1：不将有效位复制到 RSPDIF_DATA 中，而是写 0 来替代。
6	PNCPEN	奇偶校验位不复制使能位 软件置 1 和清 0。 0：将奇偶校验位复制到 RSPDIF_DATA 中。 1：不将奇偶校验位复制到 RSPDIF_DATA 中，而是写 0 来替代。
5:4	RXDF[1:0]	接收数据格式选择 软件置 1 和清 0。 00：数据格式为 RSPDIF_DATA_F0 寄存器所描述的格式，音频数据右对齐（LSB） 01：数据格式为 RSPDIF_DATA_F1 寄存器所描述的格式，音频数据左对齐（MSB） 10：数据格式为 RSPDIF_DATA_F2 寄存器所描述的格式，将两个 16 位的音频数据打包为一个 32 位的数据 11：保留
3	RXSTEOMEN	接收立体声模式使能位 软件置 1 和清 0。

0：单声道模式
1：立体声模式
这一比特位用于在溢出情况下，以处理不对齐的情况。

2	DMAREN	数据流接收 DMA 使能位 软件置 1 和清 0。 0：禁能接收 DMA 1：使能接收 DMA 当设置此位时，每当 RBNE 置位时，就生成接收 DMA 请求。
1:0	RXCFG[1:0]	RSPDIF 配置 00：禁用 RSPDIF 01：只使能 RSPDIF 同步 10：保留 11：使能 RSPDIF

33.4.2. 中断使能寄存器（RSPDIF_INTEN）

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									RXDCERRIE	SYNDOIE	SYNDBIE	RXORERRIE	PERRIE	CBNEIE	RBNEIE
									rw	rw	rw	rw	rw	rw	rw

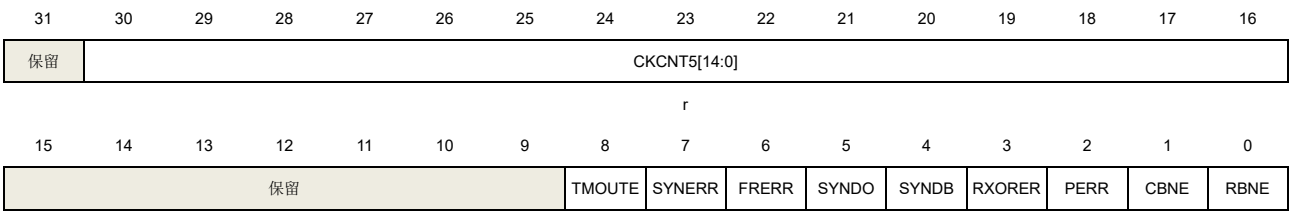
位/位域	名称	描述
31:7	保留	必须保持复位值。
6	RXDCERRIE	RSPDIF 数据解码错误中断使能 软件置 1 和清 0。 0：禁能中断 1：使能中断 当该位置位时，如果 RSPDIF_STAT 寄存器中的 SYNERR 或 TMOUTERR 或 FRERR 置位，就会产生 RSPDIF 数据解码错误中断。
5	SYNDOIE	同步完成中断使能 软件置 1 和清 0。 0：禁能中断 1：使能中断 当该位置位时，如果 RSPDIF_STAT 寄存器中的 SYNDO 位置位，就会产生同步完成中断。

4	SYNDBIE	同步块检测中断使能 软件置 1 和清 0。 0：禁能中断 1：使能中断 当该位置位时，如果 RSPDIF_STAT 寄存器中的 SYNDB 位置位，就会产生同步块检测中断。
3	RXORERRIE	接收上溢错误中断使能 软件置 1 和清 0。 0：禁能中断 1：使能中断 当该位置位时，如果 RSPDIF_STAT 寄存器中的 RXORERR 位置位，就会产生接收上溢错误中断。
2	PERRIE	校验错误中断使能 软件置 1 和清 0。 0：禁能中断 1：使能中断 当该位置位时，如果 RSPDIF_STAT 寄存器中的 PERR 位置位，就会产生校验错误中断。
1	CBNEIE	RSPDIF_CHSTAT 寄存器非空中断使能 软件置 1 和清 0。 0：禁能中断 1：使能中断 当该位置位时，如果 RSPDIF_STAT 寄存器中的 CBNE 位置位，就会产生控制流接收寄存器非空中断。
0	RBNEIE	RSPDIF_DATA 寄存器非空中断使能 软件置 1 和清 0。 0：禁能中断 1：使能中断 当该位置位时，如果 RSPDIF_STAT 寄存器中的 RBNE 位置位，就会产生接收数据寄存器非空中断。

33.4.3. 状态寄存器（RSPDIF_STAT）

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



	RR					R			
	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31	保留	必须保持复位值。
30:16	CKCNT5[14:0]	使用 rspdif_ck 计数的 5 个符号的连续时间的时钟周期数 该值可用于估算 S/PDIF 符号率。其精度受 rspdif_ck 的频率限制。例如，如果 rspdif_ck 固定为 84 MHz，则 CKCNT5= 147d。S/PDIF 数据流的采样率估算值为：Fs = 5 x rspdif_ck / （CKCNT5 x 64） ~ 44.6 kHz，因此最接近的标准采样率为 44.1 kHz。 请注意，当 SYNDO 变为高电平时，硬件更新 CKCNT5，并且每帧数据更新时 CKCNT5 都更新。
15:9	保留	必须保持复位值。
8	TMOUTERR	超时错误 0：未检测到超时错误 1：检测到超时错误 当计数器 TWCNT 值达到最大值时，该位由硬件置 1。它表示两次转换之间的时间间隔过长。这通常意味着 RSPDIF_CH 上没有有效信号。 如果 RSPDIF_INTEN 寄存器中 RXDCERRIE=1，则会生成中断。 该标志通过将 RXCFG[1:0]写为 0 来清零。
7	SYNERR	同步错误 0：未检测到同步错误 1：检测到同步错误 当由于 MAXRT 重试次数方面的原因同步失败时，该位由硬件置 1。 如果 RSPDIF_INTEN 寄存器中 RXDCERRIE=1，则会生成中断。 该标志通过将 RXCFG[1:0]写 0 来清零。
6	FRERR	帧错误 0：未检测到曼彻斯特编码错误 1：检测到曼彻斯特编码错误 当在接收数据期间发生错误时，该位由硬件置 1。报头未出现在预期位置，短转换未按对分组。仅当同步完成时（SYNDO = 1），该位才能由硬件置 1。 如果 RSPDIF_INTEN 寄存器中 RXDCERRIE=1，则会生成中断。 该标志通过将 RXCFG[1:0]写为 0 来清零。
5	SYNDO	同步完成 0：同步挂起 1：同步完成 当同步阶段完成时，该位由硬件置 1。 如果 RSPDIF_INTEN 寄存器中 SYNDOIE = 1，则会生成中断。 置位 RSPDIF_STATC 寄存器中 SYNDOC 位，即可将该标志清零。
4	SYNDB	已检测到同步块 0：未检测到“B”报头 1：已检测到“B”报头

当检测到“B”报头时，该位由硬件置 1。
如果 RSPDIF_INTEN 寄存器中 SYNDBIE = 1，则会生成中断。
置位 RSPDIF_STATC 寄存器中 SYNDBC 位，即可将该标志清零。

3	RXORERR	接收上溢错误 0：无上溢错误 1：检测到上溢错误 在 RBNE = 1 且 RSPDIF_DATA 和 RX 缓存均已满的情况下，RSPDIF 仍在接收数据时，该位由硬件置 1。 如果 RSPDIF_INTEN 寄存器中 RXORERRIE = 1，则会生成中断。 置位 RSPDIF_STATC 寄存器中 RXORERRC 位，即可将该标志清零。 **注意：**当该位置 1 时，RSPDIF_DATA 寄存器的内容不会丢失，但最后接收的数据会丢失。
2	PERR	校验错误 0：无奇偶校验错误 1：奇偶校验错误 当所接收子帧的数据和状态位中包含奇数个 0 和 1 时，该位由硬件置 1。 如果 RSPDIF_INTEN 寄存器中 PERRIE = 1，则会生成中断。 置位 RSPDIF_STATC 寄存器中 PERRC 位，即可将该标志清零。
1	CBNE	RSPDIF_CHSTAT 寄存器非空 0：RSPDIF_CHSTAT 寄存器中无控制流数据 1：RSPDIF_CHSTAT 寄存器中有控制流数据 当 RSPDIF_CHSTAT 寄存器中有控制流数据时，该位由硬件置 1。 如果 RSPDIF_INTEN 寄存器中 CBNEIE = 1，则会生成中断。 当读取 RSPDIF_CHSTAT 寄存器时，该标志清零。
0	RBNE	RX 缓存非空 0：未接收到数据 1：已接收到数据 当有效数据进入 RX 缓存时，该位由硬件置 1。 如果 RSPDIF_INTEN 寄存器中 RBNEIE = 1，则会生成中断。 可通过读取 RSPDIF_DATA 寄存器清零该标志。

33.4.4. 状态清除寄存器（RSPDIF_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										SYNDO	SYNDBC	RXORER	PERRC	保留	

	C		RC		
	w	w	w	w	

位/位域	名称	描述
31:6	保留	必须保持复位值。
5	SYNDOC	清除同步完成标志 将该位置 1，清除 RSPDIF_STAT 寄存器中 SYNDO 标志。
4	SYNDBC	清除已检测到同步块标志 将该位置 1，清除 RSPDIF_STAT 寄存器中 SYNDB 标志。
3	RXORERRC	清除接收上溢错误标志 将该位置 1，清除 RSPDIF_STAT 寄存器中 RXORERR 标志。
2	PERRC	清除校验错误标志 将该位置 1，清除 RSPDIF_STAT 寄存器中 PERR 标志。
1:0	保留	必须保持复位值。

33.4.5. 接收数据寄存器（RSPDIF_DATA）

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

根据 RXDF[1:0]值，接收数据寄存器有三种不同格式。

当 RXDF[1:0]= 2’b00，格式如下：

接收数据寄存器格式 0（RSPDIF_DATA_F0）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		PREF[1:0]		C	U	V	P	DATA[23:16]							
		r		r	r	r	r	r							

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

位/位域	名称	描述
31:30	保留	必须保持复位值。
29:28	PREF[1:0]	报头类型位 这些位表示接收到的报头类型。 00：未使用 01：接收到报头 B 10：接收到报头 M 11：接收到报头 W

请注意，如果 PTNCPEN = 1，该位域将强制被设置为零。

27 C 通道状态位
请注意，如果 CUNCPEN = 1，该位域将强制被设置为零。

26 U 用户位
请注意，如果 CUNCPEN = 1，该位域将强制被设置为零。

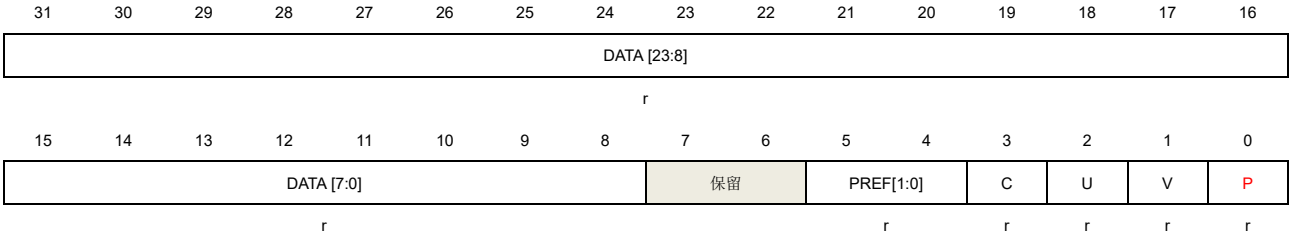
25 V 有效位
请注意，如果 VNCPEN = 1，该位域将强制被设置为零。

24 P 校验位
请注意，如果 PNCPEN = 1，该位域将强制被设置为零。

23:0 DATA[23:0] 数据位
包含 24 个接收到的数据位，数据右对齐。

当 RXDF[1:0]= 2’b01，格式如下：

接收数据寄存器格式 1（RSPDIF_DATA_F1）



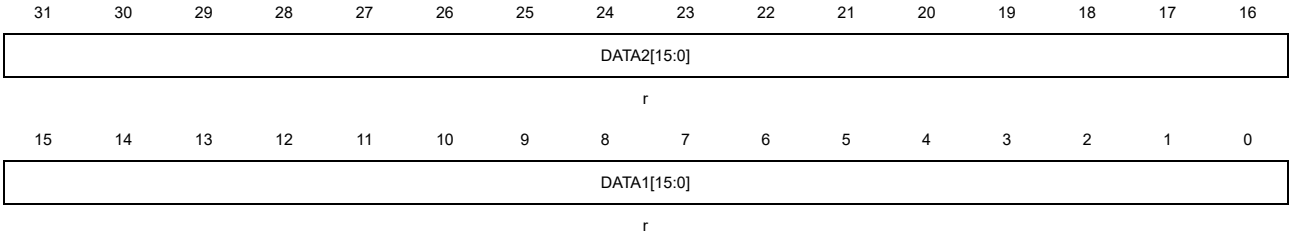
位/位域	名称	描述
31:8	DATA[23:0]	数据位 包含 24 个接收到的数据位，数据左对齐。
7:6	保留	必须保持复位值。
5:4	PREF[1:0]	报头类型位域 这些位表示接收到的报头类型。 00：未使用 01：接收到报头 B 10：接收到报头 M 11：接收到报头 W 请注意，如果 PTNCPEN = 1，该位域将强制被设置为零。
3	C	通道状态位 请注意，如果 CUNCPEN = 1，该位域将强制被设置为零。
2	U	用户位 请注意，如果 CUNCPEN = 1，该位域将强制被设置为零。
1	V	有效位

请注意，如果 VNCPEN = 1，该位域将强制被设置为零。

0 P 校验位
请注意，如果 PNCPEN = 1，该位域将强制被设置为零。

当 RXDF[1:0]=2'b10，格式如下：

接收数据寄存器格式 2（RSPDIF_DATA_F2）

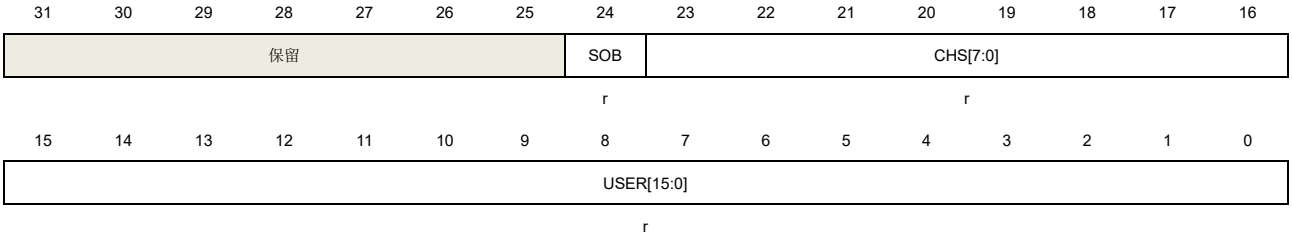


位/位域	名称	描述
31:16	DATA2[15:0]	当为立体声模式时，包含通道 A 数据。 当为单声道模式时，包含的为最老的数据。
15:0	DATA1[15:0]	当为立体声模式时，包含通道 B 数据。 当为单声道模式时，包含的为最新的数据。

33.4.6. 接收通道状态寄存器（RSPDIF_CHSTAT）

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
31:25	保留	必须保持复位值。
24	SOB	块起始 该位表示位 CHS[0]是否对应于新块的第一位 0：CHS[0]不是新块的第一位 1：CHS[0]是新块的第一位
23:16	CHS[7:0]	通道状态信息 CHS[0]是最早的值。

15:0	USER[15:0]	用户数据信息位 USER[0]是最早的值，来自通道 A，USER[1]来自通道 B。因此，n 为偶数时，USER[n]位来自通道 A，否则来自通道 B。

33.4.7. 接收阈值寄存器（RSPDIF_DTH）

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
31:29	保留	必须保持复位值。
28:16	THLO[12:0]	阈值下限 THLO = 1.5 x UI / T_{rspdif_ck} 该位域包含当前阈值下限的估算值。该值可用于估算接收数据流的采样率。THLO 的精度受限于 rspdif_ck 的周期。采样率按如下公式估算：采样率 = [2 x THLO x T_{rspdif_ck} +/- T_{rspdif_ck}] x 2/3 请注意，当 SYNDO 变为高电平时，THLO 将由硬件更新，随后每帧都更新。
15:13	保留	必须保持复位值。
12:0	THHI[12:0]	阈值上限 THHI = 2.5 x UI / T_{rspdif_ck} 该位域包含当前阈值上限的估算值。该值可用于估算接收数据流的采样率。THHI 的精度受限于 rspdif_ck 的周期。采样率按如下公式估算：采样率= [2 x THHI x T_{rspdif_ck} +/- T_{rspdif_ck}] x 2/5 请注意，当 SYNDO 变为高电平时，THHI 将由硬件更新，随后每帧都更新。

34. 串行音频接口（SAI）（适用于 GD32H737_757_759）

34.1. 简介

串行音频接口（SAI）用于支持各种通用的音频协议，如I2S、PCM/DSP、AC’97、LSB或MSB对齐和TDM，它适用于单声道和立体声。

为了初始化这些配置，SAI用了两个完全独立的音频子模块。每个音频子模块包含多达4个IO引脚（SD，SCK，FS和MCLK）。当两个音频子模块配置成相互同步时，部分IO引脚可以共用。

SAI可以配置成主/从、发送/接收的任何组合模式，根据音频子模块同步/异步配置，可以设置其操作模式为全双工/单工。

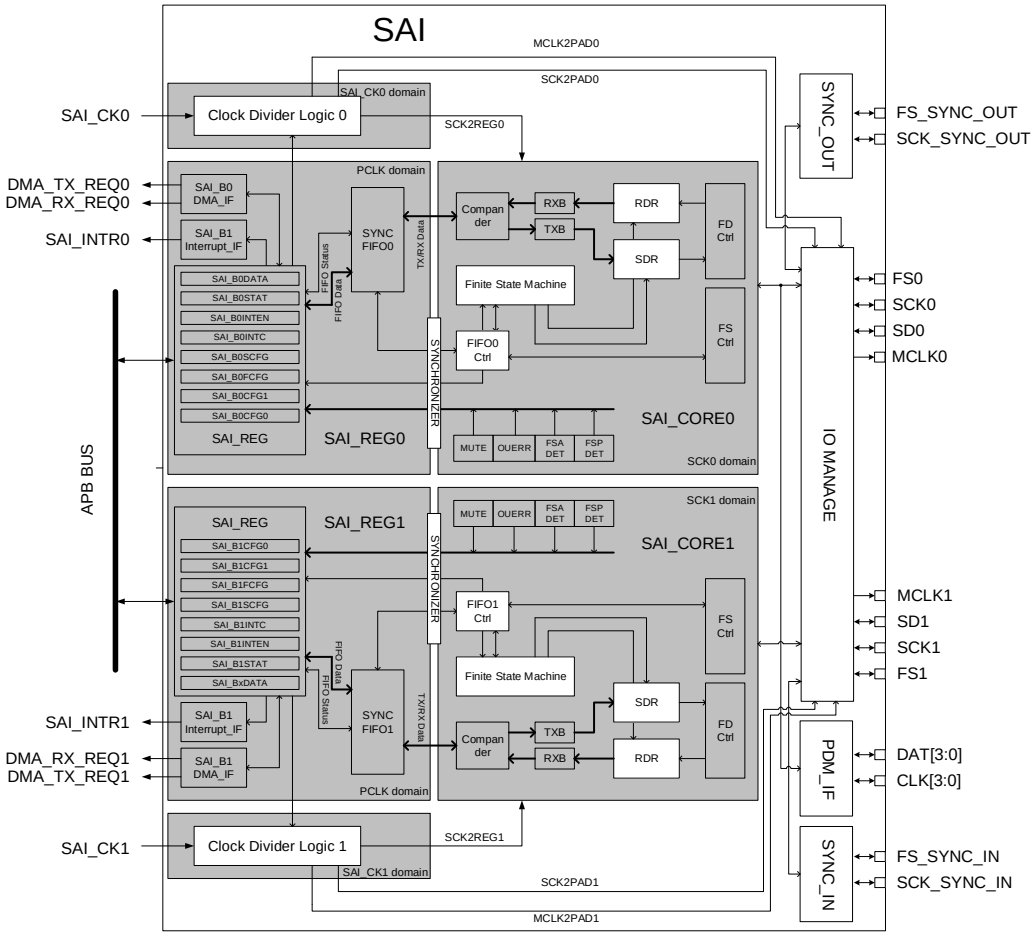
34.2. 主要特征

- 两个独立的音频子模块；
- 每个音频子模块可以配置成主/从、发送/接收的任何组合，并都具有一个8字的FIFO；
- 本地时钟分频逻辑用于满足各种音频采用率；
- 可灵活配置的音频协议，如I2S，PCM/DSP，AC’97，LSB或MSB对齐和TDM；
- PDM接口，最多支持4对麦克风（GD32H73x_75x支持3对麦克风）；
- 具有单声道/立体声音频能力，支持静音设置；
- 帧同步配置（有效电平、有效长度和偏移）；
- 每个音频帧包含多达16个可配置的slot；
- 灵活的配置slot长度，并且可以配置slot为有效或无效；
- 每个slot能够支持一个大小为8位、10位、16位、20位、24位或32位的数据，并且可以配置这些数据的第一位偏移、LSB或MSB传输；
- 串行时钟选通边沿选择（SCK）；
- 错误标志位和中断源：
  - FIFO上溢和下溢；
  - 从模式时，帧同步提前检测；
  - 从模式时，帧同步滞后检测；
  - AC’97编解码器未就绪；
  - 时钟配置错误；
- 每个音频子模块都有2个独立的DMA接口，支持频率高达4MHz的从机模式。

34.3. 功能描述

34.3.1. 模块框图

图 34-1. 模块框图



注意：GD32H73x_75x PDM接口仅支持DAT[2:0]和CLK[1:0]。

灵活的音频收发器整合了两个相同的独立子模块，并具有一个连接到输出的IO管理模块。每个音频子模块由三个独立的时域组成，分别为SAI_CK、SCK和PCLK域。定义音频采样率时钟分频逻辑设计在SAI_CK域，它的时钟输出到SCK域。SCK域包含SAI主要功能状态机、压缩/解压、发送/接收逻辑和中断产生逻辑。主要的控制寄存器和同步FIFO位于PCLK域。同步FIFO可以被ARM CPU的APB总线或DMA控制器访问。

每个音频子模块可以配置成主/从、发送/接收两者的任意组合。帧同步（FS）和串行时钟（SCK）在主模式下产生，在从模式下，音频子模块从外部主机或同步模式下另一个音频子模块接收这两个信号。主时钟（MCLK）只有在主模式时才产生，用来供外部DAC/ADC操作。有一个例外，当SAI配成支持AC’97协议时，FS强制变成输出信号，这与主/从配置无关。串行数据（SD）IO引脚在发送时配成输出，接收时配成输入。

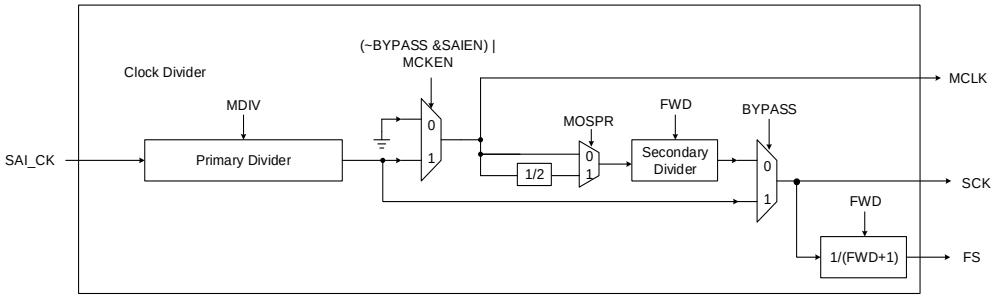
IO管理模块控制每个音频子模块的IO引脚，当两个音频子模块被声明为互相同步时，FS、SCK

和MCLK可以共用，同步子模块的这些引脚被释放，并可用作通用IO。

34.3.2. 时钟分频器

SAI的两个音频子模块的时钟分频逻辑只有在当它们配置成主设备时才打开，否则是关闭的，并且MCLK和SCK的输出都保持低电平。分频器的时钟源SAI_CK，推荐采用45.1584MHz和49.152MHz这两个特定值来产生标准的音频采样率。时钟分频逻辑由主时钟分频器和子时钟分频器组成，其中主时钟分频器用于产生所需的主时钟（MCLK），子时钟分频器用于产生位时钟（SCK）。时钟分频逻辑如*图34-2. 时钟分频逻辑*所示。

图 34-2. 时钟分频逻辑



主时钟分频比 MDIV 直接链接到 SAI 控制寄存器内的主时钟分频比控制字段，其输出频率可通过以下公式获得。

$$f_{MCLK} = \begin{cases} \frac{f_{SAI_CK}}{MDIV}, MDIV \neq 0 \\ f_{SAI_CK}, MDIV = 0 \end{cases} \quad (34\text{-}1)$$

注意：以上公式仅在 BYPASS 无效、SAI 开启且 MCKEN 开启时成立，否则 MCLK 保持低电平。

辅助时钟分频器的比率 MDIV 连接到 SAI 控制寄存器内的帧长度（FWD）和 MCLK 过采样率（MOSPR）控制字段。以下公式决定了 SAI_CK 与位时钟（SCK）采样率之间的关系。

$$f_{SCK} = \frac{f_{SAI_CK} \times (FWD+1)}{MDIV \times (MOSPR+1) \times 256} \quad (34\text{-}2)$$

帧同步频率 f_{FS}:

$$f_{FS} = \frac{f_{SAI_CK}}{MDIV \times (MOSPR+1) \times 256} \quad (34\text{-}3)$$

当 BYPASS 置位时，主时钟（MCLK）以固定输出值 0 关闭，而位时钟（SCK）取决于 MDIV。另外，帧长值没有限制，只要帧长大于等于 8 即可。

当 BYPASS 清零时，需要设置（FWD+1）等于 master 模式下基数为 2 的指数函数的结果，以保证 SCK 可被 MCLK 整除。

*表 34-1. 常用的音频采用率*列出了帧长度为 256 位时一些常用的音频采样率配置。

表34-1. 常用的音频采用率

SAI_CK时钟频率	标准音频采样率	主时钟分频率
192kHz x 256	192 kHz	MDIV = 1
	96 kHz	MDIV = 2
	48 kHz	MDIV = 4
	16 kHz	MDIV = 12
	8 kHz	MDIV = 24
44.1kHz x 256	44.1 kHz	MDIV = 1
	22.05 kHz	MDIV = 2
	11.025 kHz	MDIV = 4

34.3.3. 操作模式

SAI音频子模块可以独立的配置成主/从、发送/接收任何组合的操作模式。

主设备

帧同步（FS）是由主设备在FIFO不为空且帧开始时生成，它用来协调帧开始或通道识别。串行时钟（SCK）和主时钟（MCLK）都是由主设备生成的信号，SCK信号专门被从设备用来作为位时钟。和FS不同，SCK和MCLK的产生不受FIFO是否为空的制约，只要音频子模块被使能，他们就会生成。

从设备

从设备接收来自主设备的FS和SCK信号，这些信号的来源取决于音频子模块是声明为同步还是异步。当选择异步模式时，FS和SCK信号源被直接关联到芯片级IO端口。当选择同步模式时，FS和SCK信号源被连接到另一个音频子模块的FS和SCK信号端。用户必须在使能主设备前先使能从设备，否则从设备将不能完整地接收主设备的数据。

发送器

当音频子模块被配置成发送器时，串行数据（SD）为输出。如果在音频子模块使能之后FIFO还是为空，则会发送数值0，并产生下溢标志（OUERR）。

接收器

当音频子模块被配置成接收器时，串行数据（SD）为输入。从接收器总会监测FS信号，当检测到第一个有效边沿时，音频子模块存储接收到的数据，然后由内部有限态机器处理后续数据的接收。当SAI失能时，接收器会在帧结束时才停止接收。

34.3.4. 同步模式

SAI的同步模式可以分为内部同步模式和外部同步模式。

内部同步

内部同步模式具有减少通信时占用外部引脚数量的优点，即SAI子模块SAI_B0和SAI_B1可以

同步运行，二者将共用SAI_FS和SAI_SCK信号，从而释放SCKx、FSx和MCLKx的GPIO引脚。

内部同步模式下的SAI子模块在全双工通信中可以配置为如下几种模式：

1. 子模块0（或者1）配置为主模块，子模块1（或者0）配置为从模块；
2. 子模块0和子模块1都配置为从模块；
3. 子模块0（或者1）配置为异步模块，子模块1（或者0）配置为同步模块。

注意：由于存在内部重新同步阶段，因此PCLK APB频率必须大于比特率时钟频率的二倍。

外部同步

外部同步即SAI音频子模块与其他SAI进行同步。通过配置SAI_SYNCFG寄存器中的SYNO[1:0]确定为其他SAI提供FS和SCK信号的同步源，即SAI子模块0或者SAI子模块1。通过配置SAI_SYNCFG 寄存器中的SYNI[1:0]确定接收同步信号的SAI选择哪个SAI进行同步。通过配置SAI_BxCFG0寄存器中的SYNCMOD[1:0]指定SAI音频子模块是否与其它SAI同步。

如果SAI中的两个音频子模块都需要与另一个SAI同步，则可以通过配置SYNCMOD[1:0]位来配置每个音频子模块与另一个SAI模块同步。或者通过配置SYNCMOD[1:0]位来配置一个音频子模块与另一个SAI模块同步。然后通过配置 SYNCMOD[1:0]位配置其他音频子模块与第二个SAI音频子模块同步。

参考*表34-2. 外部同步配置*进行外部同步配置。

表34-2. 外部同步配置

SAI 模块	SYNI =2	SYNI =1	SYNI =0
SAI0	SAI2 同步	SAI1 同步	保留
SAI1	SAI2 同步	保留	SAI0 同步
SAI2	保留	SAI1 同步	SAI0 保留

注意：当 SAI 子模块配置为从机模式时，存在同步信号方向限制：该从机子模块只能接收同步信号，无法提供帧同步(FS)信号给其他 SAI 子模块。

在以下场景中需要特别注意：

1. 内部同步模式：从机子模块不能作为同步信号源，只能接收来自主机模块的 FS 和 SCK 信号。
2. 外部同步模式：若从机模块需要与外部 SAI 同步，必须通过 GPIO 引脚接收外部 FS 和 SCK 信号，无法通过内部路径向外部 SAI 提供同步信号。

在系统设计阶段应充分考虑此限制，合理规划 SAI 模块的主从关系和引脚分配

34.3.5. 帧配置

帧同步

帧同步信号是主设备和从设备之间初始化一个传输的协调信号。许多参数用于控制它的波形。

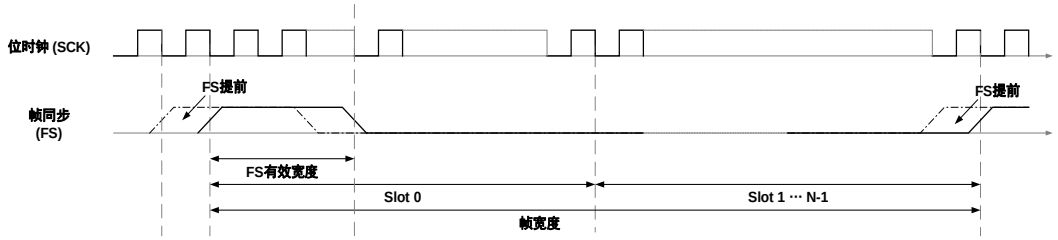
帧同步提前

帧同步有效边沿可以和第一个slot中第一个比特位的开始或者其前一个SCK位时钟对齐，这取决于SAI_SxFCFG寄存器中FSOST控制字段。*图34-3. FS有效宽度*展示了FS波形是如何改变的。

帧同步有效宽度

*图34-3. FS有效宽度*中帧同步信号的有效宽度取决于SAI_BxFCFG寄存器的FSAWD控制字段的配置，它的实际宽度等于（FSAWD+1）个SCK时钟周期，且其最小值为1个SCK时钟周期，最大值为128个SCK时钟周期，即为最大帧宽的一半。当FSFUNC置1时，FS信号不仅表示帧开始，还能表示通道识别，这种情况下，（FSAWD+1）必须等于帧宽的一半，否则音频子模块的功能将不能得到保证。

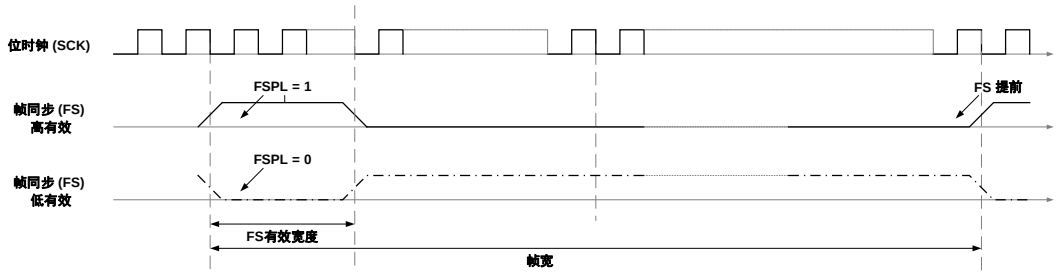
图 34-3. FS 有效宽度



帧同步极性

帧同步有效电平可以通过SAI_BxFCFG寄存器的FSPL控制字段配置，如*图34-4. FS极性*所示。

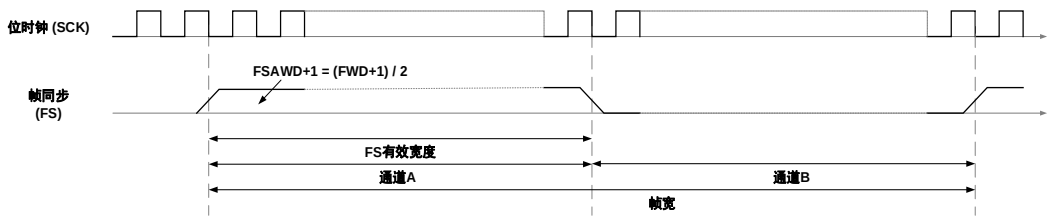
图 34-4. FS 极性



帧同步功能

帧同步功能的定义通过SAI_BxFCFG寄存器的FSFUNC控制字段进行配置。有两个指定的功能可被选择，当FSFUNC置1时，FS不仅表示帧开始还表示通道编号的识别，在这种情况下，帧同步有效宽度（FSAWD+1）应该配置成帧宽的一半，如*图34-5. FS功能*所示，否则音频子模块的行为将不能得到保证。当FSFUNC为0时，FS只表示帧开始。

图 34-5. FS 功能



帧宽

帧宽不能小于8位（相当于一个字节的数据），也不能大于256位。

在主模式中，如果BYPASS清0，帧宽（FWD+1）的值应该设置为8到256之间且等于2的几次幂的值，以保证每个SCK时钟周期包含整数个MCLK时钟周期，这是外部DAC/ADC能正确操作的必须要求，否则SAI_BxINTEN寄存器中的错误时钟标志位（ERRCK）会置位，若还使能了SAI_BxINTEN寄存器的错误时钟中断位（ERRCKIE），则产生一个中断。在主模式中，如果BYPASS置1，这将对帧宽的配置没有约束，主时钟自动关闭。

在从模式中，帧宽配置用于配合内部有限状态机来获取有效帧的开始和结束。它还有另一个用途，就是用于检测帧同步信号的提前或滞后，如果出现帧同步提前或滞后现象，则一个错误标志位会被置位，如果使能了相应的中断，则产生一个中断，具体可以参考*错误标志位*和*中断*章节。

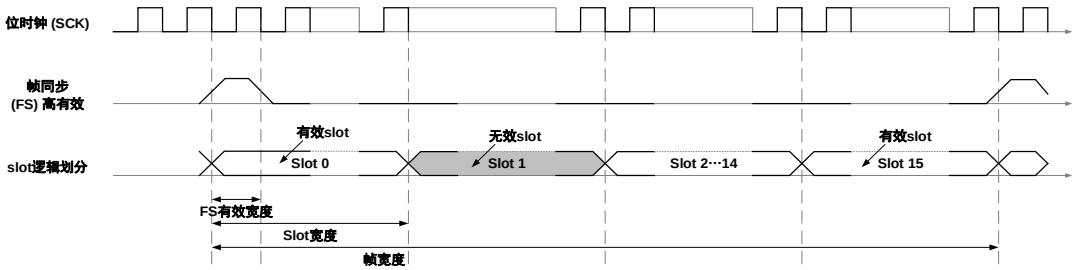
34.3.6. Slot 配置

每个SAI帧逻辑上最多分为16个slots，每个slot的有效状态和它们的分布通过slot配置寄存器进一步控制。Slot宽度可以通过SAI_BxSCFG寄存器的SLOTWD控制字段配成16位、32位或是和数据宽度一致。

Slot 激活

每个slot的激活状态可以通过SAI_BxSCFG寄存器的slot激活向量（SLOTAV）独立配置。SLOTAV是一个16位宽的控制字段，每个比特位控制相应的一个slot的激活状态。Slot的逻辑划分如*图34-6. Slot激活*所示。

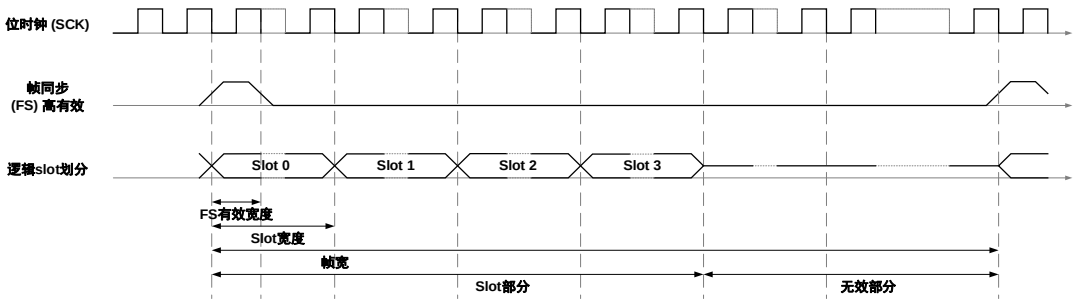
图 34-6. Slot 激活



Slot 分布

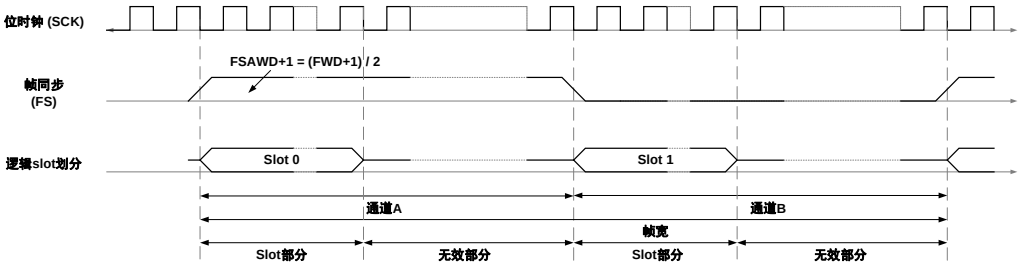
在slot个数和slot宽度的乘积小于帧宽的特殊情况下，存在非slot的分布。Slot部分即为有slot分布的部分，其他的为无效部分。当FSFUNC为0时，FS仅表示信号帧的开始，从最后一个slot结束到下一个帧的开始之间为slot的无效部分，如*图34-7. 当FSFUNC=0时，slot分布*所示。

图 34-7. 当 FSFUNC=0 时，slot 分布



当FSFUNC=1，FS不仅表示帧开始，还表示通道识别，slot部分和无效部分均匀分布在两个通道上。无效部分为从当前通道的最后一个slot到下一个通道的slot开始之间的部分。

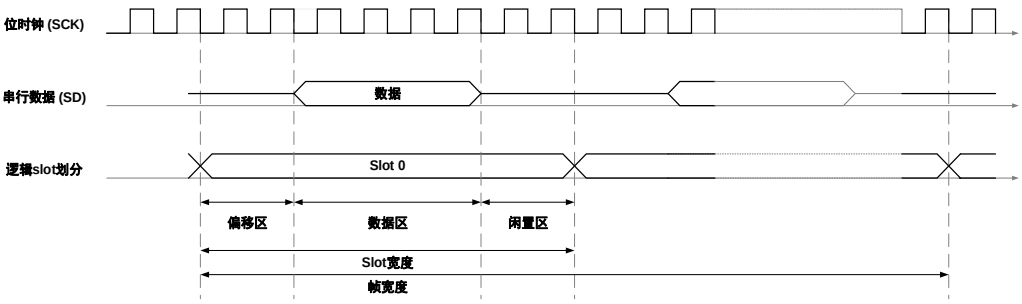
图 34-8. 当 FSFUNC=1 时，slot 分布



在无效 slot 上的串行数据输出管理

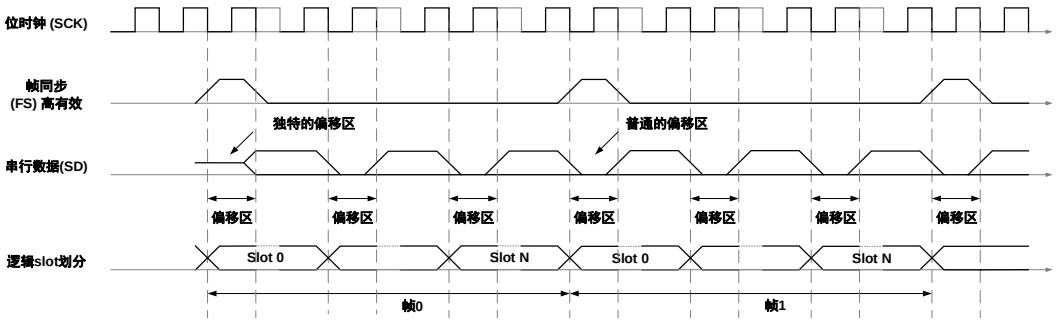
在无效slot附近的串行数据（SD）输出行为可以根据SAI_BxCFG1寄存器中的串行数据输出模式（SDOM）定义的管策略来决定是SAI释放还是驱动输出0。SD输出行为，偏移和空闲区域这三项需要特别注意。在该用户手册中，将slot部分规定为偏移区、数据区和闲置区，具体描述如*图34-9. Slot部分的规定*所示。

图 34-9. Slot 部分的规定



首先，偏移区域的SD输出由SDOM决定,SDOM为1，那么SAI将会释放SD的输出，否则SD输出0，其区别如*图34-10. 偏移区的处理*所示。

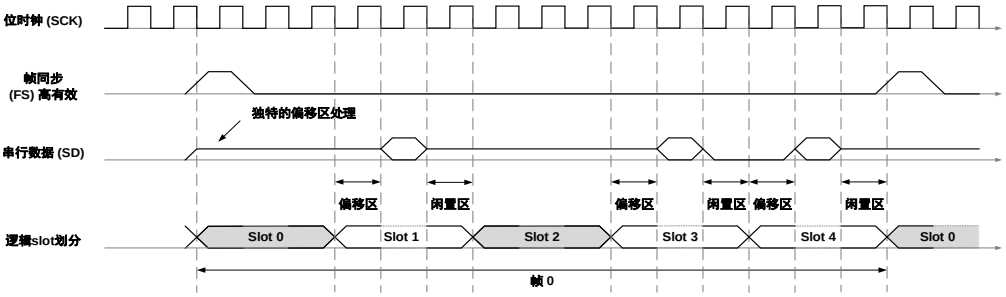
图 34-10. 偏移区的处理



其次，当SDOM为1时，一个帧的最后一个slot的闲置区期间SD输出行为将参考第一个slot的有效状态。如果slot0是无效的时候，SD输出为释放状态，否则如果slot0是有效的，则SD输出0。当SDOM为0时，则SD的输出0，这和其他slot的有效状态无关。

最后，位于帧中间的slot的偏移区和闲置区的SD输出参考它们上一个slot和下一个slot的有效状态。如果上一个slot是无效的，并且存在偏移区，那么当SDOM=1时，SD输出线释放，当SDOM=0时，SD输出0。同样，如果下一个slot是无效的，且存在闲置区，那么当SDOM=1时，SD线释放，当SDOM=0时，SD输出0。在有效slot和无效slot附近的偏移区和闲置区的SD输出行为如*图34-11. SD输出管理*所示。

图 34-11. SD 输出管理



34.3.7. 数据配置

数据宽度也是灵活的，它可以通过SAI_BxCFG0寄存器的DATAWD位将其配置成8位、10位、16位、20位、24位和32位宽。通过设置SAI_BxSCFG寄存器的数据偏移（DATAOST）位，可以将有效slot中的数据向前移或是向后移。如串行数据输出管理部分(*注意：当SAI子模块配置为从机模式时，存在同步信号方向限制：该从机子模块只能接收同步信号，无法提供帧同步*(FS)信号给其他SAI子模块。

在以下场景中需要特别注意：

3. 内部同步模式：从机子模块不能作为同步信号源，只能接收来自主机模块的 FS 和 SCK 信号。

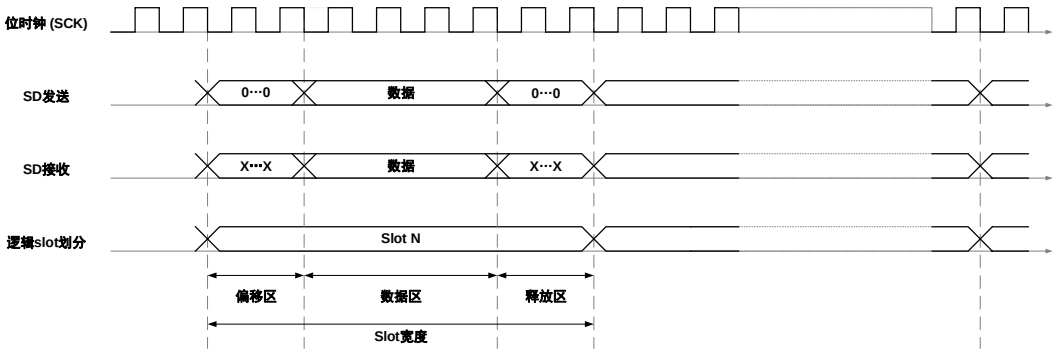
4. 外部同步模式：若从机模块需要与外部 SAI 同步，必须通过 GPIO 引脚接收外部 FS 和 SCK 信号，无法通过内部路径向外部 SAI 提供同步信号。

在系统设计阶段应充分考虑此限制，合理规划 SAI 模块的主从关系和引脚分配

帧配置节)所描述的那样，一个slot开始处与里面数据的第一个比特位之间的空间称为偏移区，

数据的最后一个比特位和slot结束处之间的空间称为闲置区。当音频子模块配置成发送器，且存在偏移区或闲置区，那么在这些区期间，SD输出0。SD线的实际行为不仅取决于输出值，还取决于线管理条件和附近slot的有效状态。当音频子模块配置成接收器，且存在偏移区或闲置区，那么在这些区期间的数据接收将会被忽略。数据发送和接收如*图34-12. 数据配置*所示。

图 34-12. 数据配置



34.3.8. 同步 FIFO

在每个SAI音频子模块内部独立应用一个8字深的同步FIFO以提高传输效率。这些FIFO可以被CPU或是DMA访问，FIFO请求中断机制用于请求CPU和DMA访问。FIFO请求的产生取决于操作模式、FIFO阈值、FIFO状态和DMA突发传输大小。FIFO请求中断的产生概括在*表34-3. FIFO请求的产生条件*中，如果根本条件不满足，则中断请求就会被清除。

表 34-3. FIFO 请求的产生条件

发送: OPTMOD[0] = 0				接收: OPTMOD[0] = 1			
FIFO 阈值	FFTH	FIFO状态	FFSTAT	FIFO阈值	FFTH	FIFO状态	FFSTAT
空	= 000	空	= 000	空	= 000	不空	≥ 001
1/4 满	= 001	<1/4满	<010	1/4满	= 001	≥ 1/4满	≥ 010
1/2满	= 010	<1/2满	<011	1/2满	= 010	≥ 1/2满	≥ 011
3/4满	= 011	<3/4满	<100	3/4满	= 011	≥ 3/4满	≥ 100
全满	= 100	不满	<101	全满	= 100	全满	= 101

通过设置SAI_BxCFG1寄存器的FLUSH控制字段可以实现FIFO刷新，当FLUSH置1时，FIFO中的所有数据内容将被清除，读写指针复位到0。

注意：DMA请求的产生取决于FIFO请求，DMA接口章节会给出详细信息。

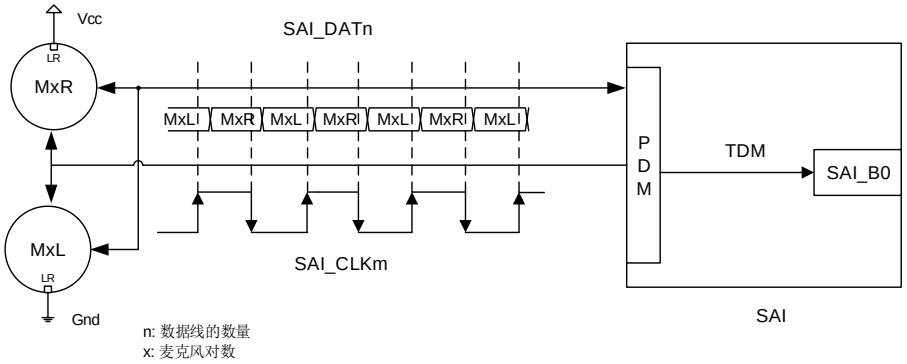
34.3.9. PDM 接口

数字麦克风可以通过PDM接口实现数据传输，PDM接口最多支持4对数字麦克风并联使用。

时序与连接

采用PDM接口连接的两个麦克风示意图如*图34-13. PDM典型连接和时序图*所示。

图 34-13. PDM 典型连接和时序图



注意：GD32H73x_75x PDM接口仅支持DAT[2:0]和CLK[1:0]。

麦克风LR引脚接入VCC的为左麦克风通道，接入GND的为右麦克风通道，左通道与右通道麦克风共用一路时钟线SAI_CLK，时序信号由SAI_B0的TDM接口进行调整后产生，通过在上升沿采样左声道数据，在下降沿采样右声道数据，完成双声道数据的采集。

注意：PDM接口只支持与配置为TDM主模式的SAI_B0子模块配合使用。

PDM 接口使能

PDM 接口启用流程如下：

1.配置 TDM 为主模式，SAI_B0CFG0 寄存器的 OPTMOD[0:1]为主机接收模式，PROT[1:0]位为自由协议；

2.配置 PDM 接口：通过配置 SAI_PDMCTL 寄存器的 NBMIC[1:0]位选择麦克风的数量，配置 SAI_PDM 寄存器中的 CLKLx 位使能时钟；

3.配置 SAI_PDMCTL 寄存器中的 PDMEN 位使能 PDM 接口；

4.配置 SAI_B0CFG0 寄存器中的 SAIEN 位并使能 SAI_B0。

数据处理

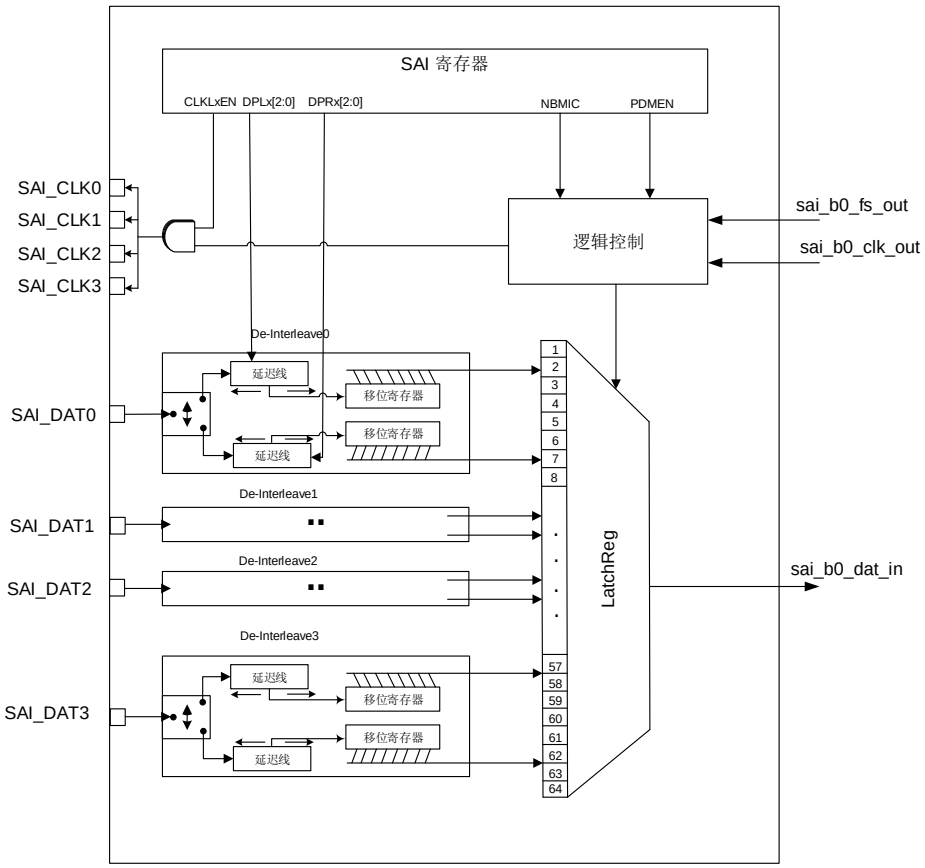
PDM中的数据处理序列如下：

1.SAI_B0 产生时钟经 TDM 链路到 PDM 接口产生比特流时钟

2.通过配置 SAI_PDMCFG 寄存器的 DPLx[2:0]和 DPRx[2:0]位，实现 PDM 接口对来自麦克风产生的比特流数据 SAI_DATx 进行交错和延迟处理，从而对麦克风所产生的延迟进行调整。

3.移位寄存器将数据流转换为字节，通过 TDM 链路将数据传输到 SAI_B0。

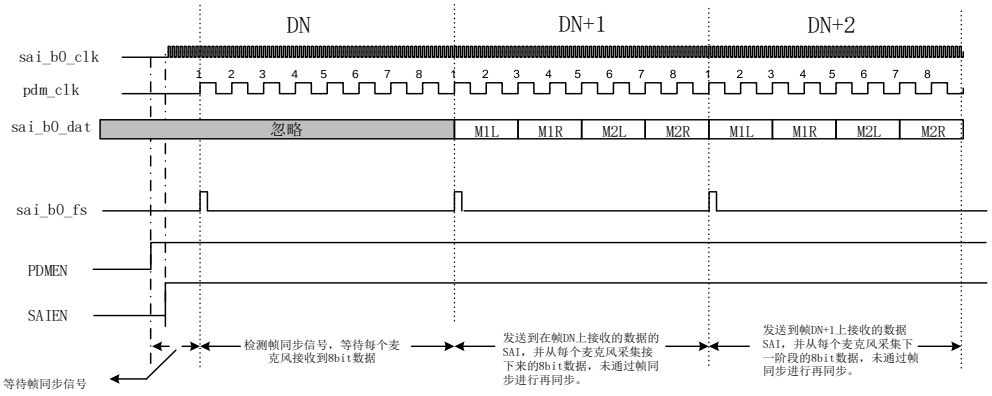
图 34-14. PDM 数据处理示意图



数据传输启动过程

在使能PDM接口后，麦克风数据采样的开始是在帧同步事件发生后进行，在8个SAI_CK时钟之后，麦克风的数据将会通过TDM接口传输到SAI。*图34-15. PDM数据传输的启动过程*显示了PDM数据传输的启动过程。

图 34-15. PDM 数据传输的启动过程



数据格式

获取到的麦克风数据的数据格式主要与以下配置有关，

1.在PDMCTL寄存器中配置NBMIC[1:0]设置麦克风的数量；

2.在BxSCFG寄存器中配置SLOTWD[1:0]设置槽宽；

3.配置SHIFTDIR位设置传输数据的MSB/LSB。

表 34-4. 不同配置下，获取麦克风数据需要读取寄存器次数

麦克风数量	Slot 宽度	获取麦克风数据需要读取 BxDATA 寄存器次数
8	32	2
	16	4
	8	8
4	32	1
	16	2
	8	4
2	16	1
	8	2

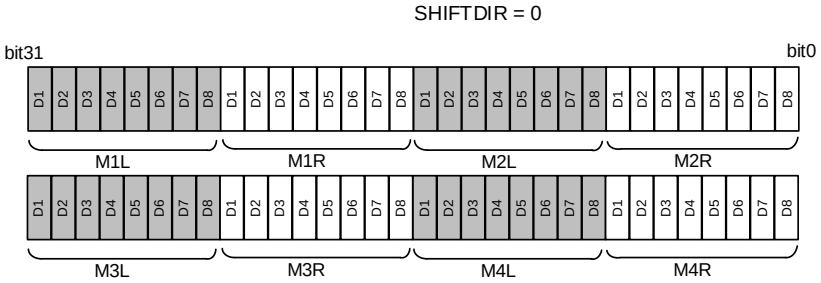
注意：GD32H73x_75x 支持3对麦克风。

下图给出了不同麦克风数量和Slot宽度下，寄存器数据排列示意图。

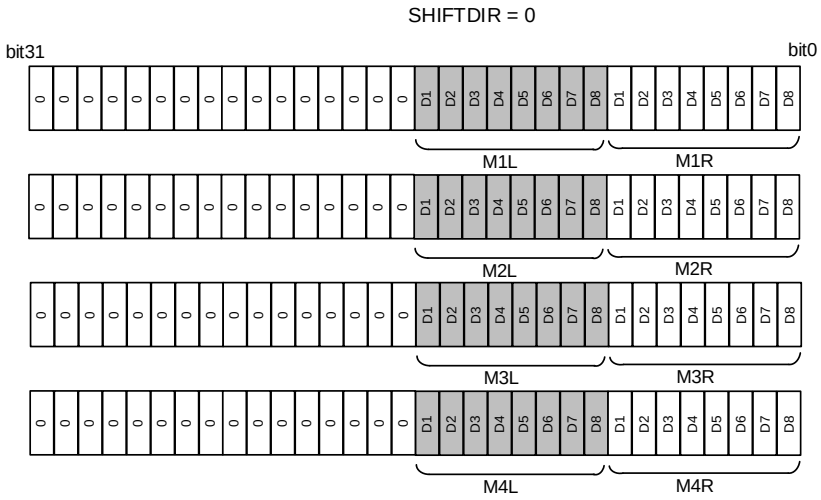
图 34-16. 八麦克风，不同 slot 宽度下，BxDATA 寄存器数据格式

八麦克风配置

32位slot宽度



16位slot宽度



8位slot宽度

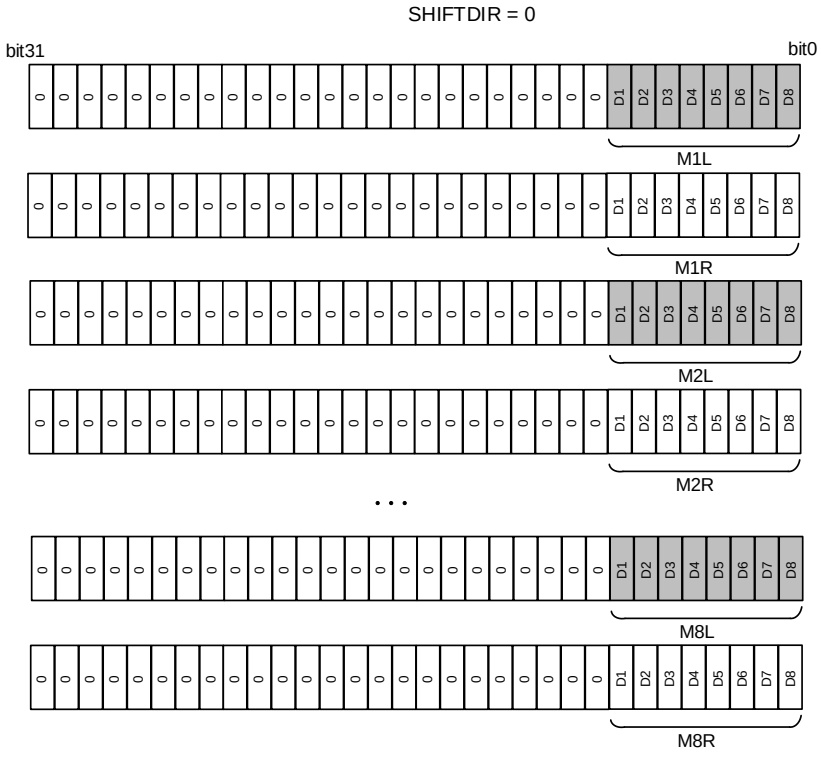


图 34-17. 四麦克风，不同 slot 宽度下，BxDATA 寄存器数据格式

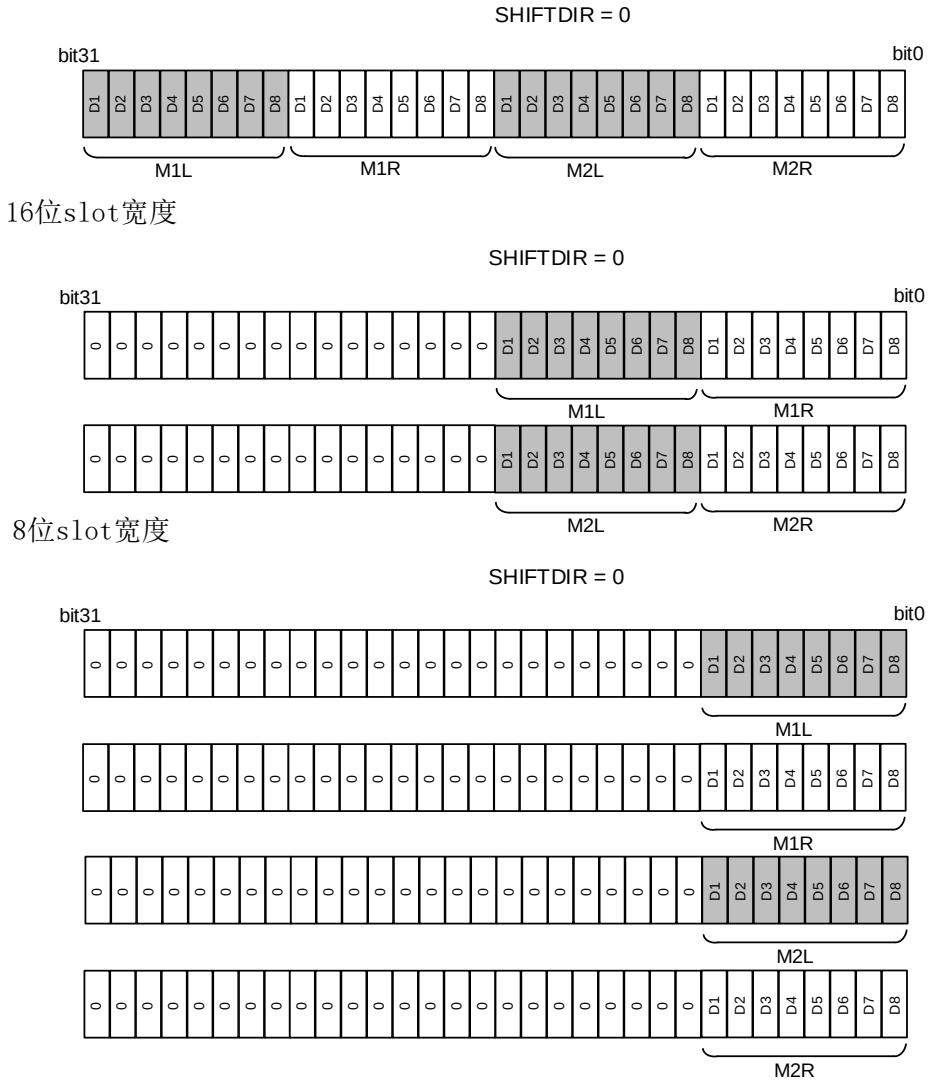
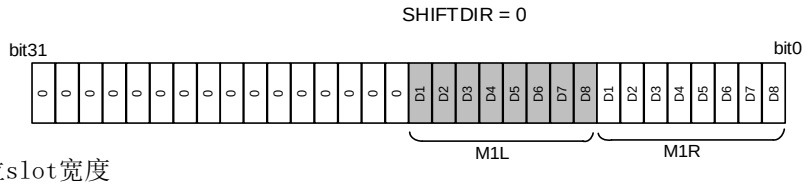


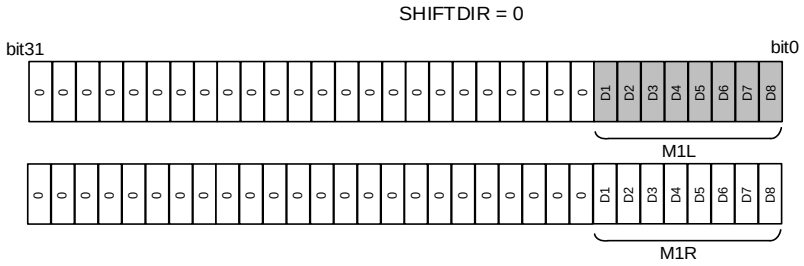
图 34-18. 双麦克风，不同 slot 宽度下，BxDATA 寄存器数据格式

双麦克风配置

16位slot宽度



8位slot宽度



TDM 配置

针对PDM接口的TDM配置参考*表34-5. TDM配置表*。

表 34-5. TDM 配置表

寄存器	位域	值	描述
SAI_B0CFG0	OPTMOD[1:0]	0b01	工作模式配置为主机接收接收模式
	PROT[1:0]	0b00	协议为自由协议
	DATAWD[2:0]	x	数据宽度
	SHIFTDIR	x	数据传输是 MSB 还是 LSB
	SAMPEDGE	0	数据采样时钟边沿下降沿采样
	MONO	0	立体声模式选择
	BYPASS	1	时钟分频器逻辑旁路
	MDIV[5:0]	x	主时钟分频器
SAI_B0FCFG	FWD[7:0]	x	帧宽度
	FSAWD[6:0]	0	FS 有效宽度为 1 个 SCK 时钟周期
	FSFUNC	0	FS 只定义帧开始
	FSPL	1	FS 有效极性为高
	FSOST	0	FS 有效边沿声明为第一个 slot 的第一个位开始处
SAI_B0SCFG	DATAOST[4:0]	0	数据无偏移
	SLOTWD[1:0]	0	Slot 宽等于数据位宽
	SLOTNUM[3:0]	x	一个帧中的 slot 个数
	SLOTAV[15:0]	x	Slot 激活向量

注意：在配置PDM时，时钟频率、帧长度和Slot大小需要遵循以下三条要求：

1.时钟频率配置遵循以下公式：

$$f_{SCK_B0} = f_{PDM_CLK} * (NBMIC+1) * 2 \quad (30\text{-}4)$$

2.帧长度需要满足以下公式：

$$FWD=(16*(NBMIC+1))-1 \quad (30\text{-}5)$$

3.Slot大小需要配置为（FWD+1）的整数倍。

34.3.10. AC’97 链路控制器

AC’97链路控制器模式是通过SAI_BxCFG0寄存器的PROT位配置的。当选择了这个协议，有许多配置字段会被忽略，包括数据移位方向、数据宽度、帧和slot的大部分配置以及部分中断控制字段，具体可以参考寄存器定义部分的描述。

AC’97协议的帧宽固定为256位，每个帧被分成13个slot，第一个slot固定为16位宽，其他的12个slot的宽度固定为20位。用户必须设置SAI_BxCFG0寄存器的数据宽度（DATAWD）控制字段为16位或20位，否则将不能保证音频子模块的行为。

TAG（即Slot0）中的位2为保留位，无论写什么值到TAG中，位2均会被写0。

TAG（slot 0）中的位3到位14为自由协议的slot激活向量（SLOTAV），其中TAG slot（即slot 0）总为有效，位3对应slot12，位14对应slot1.

TAG（slot 0）的位15是编解码就绪状态指示位，当音频子模块配置为接收时，接收到的TAG（slot0）的位15为0，则表明音频编解码器没有就绪，相应的ACNRDY标志位置1。如果ACNRDY标志位和音频编解码器未就绪中断使能位（ACNRDYIE）都置1，则产生一个中断。

帧同步有效边沿被声明为数据的第一个比特位的前一个时钟周期，如*图34-19. AC’97的slot划分*所示。

图 34-19. AC’97 的 slot 划分

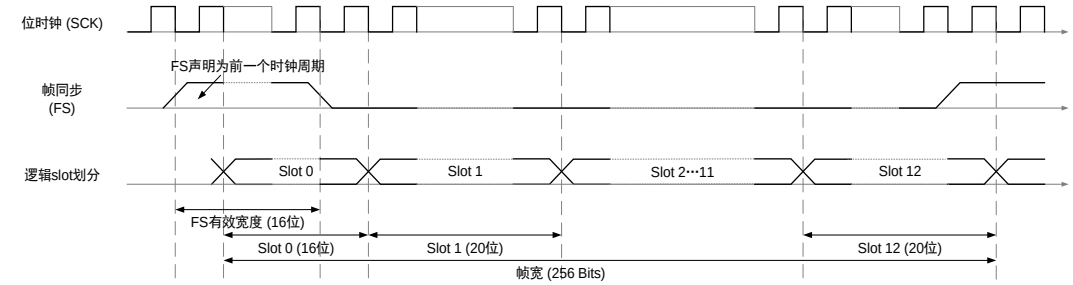


*图34-20. AC’97 TAG定义*给出了AC’97slot划分的综述。

图 34-20. AC’97 TAG 定义

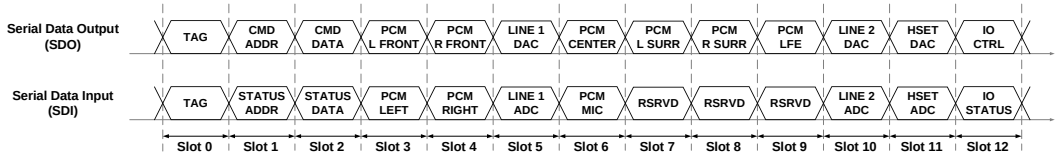


*表34-6. AC’97发送slot定义*和*表34-7. AC’97接收slot定义*概括了每个slot的定义和意义。

当AC’97链路控制器作为发送器时。

表 34-6. AC'97 发送 slot 定义

Slot	名称	描述
0	输出目标	高位指示哪个slot包含有效数据，低位指示传达编解码器ID
1	命令地址端口	读/写命令和7位的编解码器寄存器地址
2	命令数据端口	16位命令寄存器写数据
3,4	PCM回放	左右声道输入的16、18、20位PCM数据
5	Modem Line1 DAC	Modem line1输出的16位Modem数据
6,7,8,9	中置，左右环绕，LEF数据	中置，左右环绕与LEF通道的16、18、20位PCM数据
10	Modem Line2 DAC	Modem line2输出的16位Modem数据
11	Modem听筒	听筒的16位Modem数据
12	Modem IO控制	用于Modem控制的GPIO写端口I
10-11	SPDIF输出	AC-link可选SPDIF输出带宽
6-12	双倍音频数据	88.2或者96kHz的AC-link可选左，中，右声道带宽。实际使用时间片由DRSS位控制

当AC'97链路控制器作为接收器时。

表 34-7. AC'97 接收 slot 定义

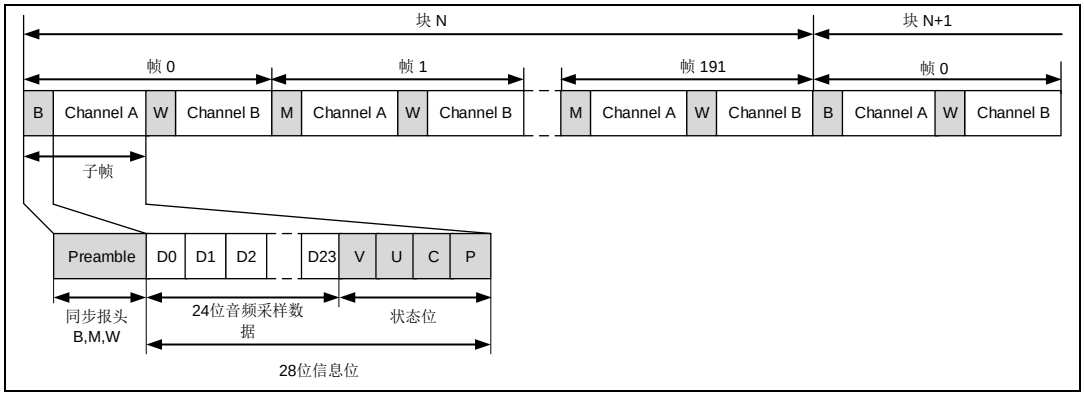
Slot	Name	Description
0	输入目标	高位指示哪个slot包含有效数据;
1	状态地址端口	高位指示寄存器地址，低位指示请求数据的时间片
2	状态数据端口	读取到的16位寄存器数据
3,4	PCM录音	左右声道输出的16、18、20位PCM数据
5	Modem Line 1 ADC	Modem line1输入的16位Modem数据
6	话筒专用ADC	用于第三个可选ADC的16、18、20位PCM数据
7,8,9	供应商保留	供应商特定（增强的输入扩充口，或者麦克风阵列等）
10	Modem Line 2 ADC	Modem line2输入的16位Modem数据
11	Modem话筒ADC	话筒的16位Modem数据
12	Modem IO状态	Modem状态读取GPIO端口

34.3.11. SPDIF 输出

SPDIF（索尼/飞利浦数字接口）是一种用于消费音频设备的数字音频互连，用于在合理的短距离内输出音频。 SPDIF 支持 IEC 60958 标准。

*图 34-21. SPDIF 数据格式*显示了 SPDIF 块格式和子帧格式。

图 34-21. SPDIF 数据格式



每个 SPDIF 块包含 192 帧数据，每个帧由左通道子帧（32 位）和右通道子帧（32 位）组成，每个子帧由 4bit 的 SOPD 模式、24bit 的数据信息和 4bit 的状态信息组成。

SOPD 模式编码参考*表 34-8. SOPD 模式*。

表 34-8. SOPD 模式

预先状态（前一个半比特值）	0	1	描述
报头	编码		
B	11101000	00010111	通道 A，且为一个块的起始子帧
W	11100100	00011011	通道 B
M	11100010	00011101	通道 A

SPDIF 的数据传输在 SAI_BxDATA 寄存器的数据填充应遵循：SAI_BxDATA[26:24]包含通道状态位、用户位和有效性位，SAI_BxDATA[23:0]包含所考虑通道的 24 位数据。

注意：如果数据大小为 20/16 位，应将数据映射到 SAI_BxDATA[23:4] /SAI_BxDATA[23:8]上。

通过配置 SAI_BxCFG0 寄存器中 OPTMOD[1]位为 0，强制选择为主模式，同时将忽略 SAI_BxCFG0 寄存器中 DATAWD[2:0]数据位宽设置，强制设置为 24 位，通过时钟发生器配置符号率，并通过曼彻斯特协议进行编码。

SAI 首先在块中发送每个子帧的适当报头。随后在 SD 线上发送 SAI_BxDATA（以曼彻斯特协议进行编码）。SAI 通过传输按*表 34-9. 校验位奇数*奇偶校验位来结束子帧。

表 34-9. 校验位奇数

SAI_BxDATA [26:0]	传输校验位 P 的值
奇数个 0	0
奇数个 1	1

对于 SPDIF 发生器，SAI 应提供一个符号率两倍的位时钟。通常情况下，音频采样率（F_S）和比特时钟率（F_{SCK_x}）之间的关系由以下公式给出：

$$F_s = \frac{F_{SCK_x}}{128} \tag{34-6}$$

比特时钟率由以下公式给出:

$$F_{SCK_x}=\frac{F_{SAI_CK_x}}{MDIV} \quad (34\text{-}7)$$

注意：仅当 SAI_BxCFG0 寄存器中 BYPASS 设置为 1 时，上述公式才有效。

34.3.12. 立体声/单声道

SAI音频子模块通过设置SAI_BxCFG0寄存器的MONO位进行立体声和单声道模式的转换。注意，如果选择单声道，则slot的个数必须配置为2，否则音频子模块的行为将不能保证。

当音频子模块配置为发送器时，在第一个slot（slot0）期间发送的数据将会复制到第二个slot（slot1），在这种情况下，FIFO的访问次数是立体声模式的一半。

当音频子模块配置成接收器时，在第一个slot期间接收到的数据被放入FIFO，第二个slot期间接收的数据将会被丢弃。

34.3.13. 静音

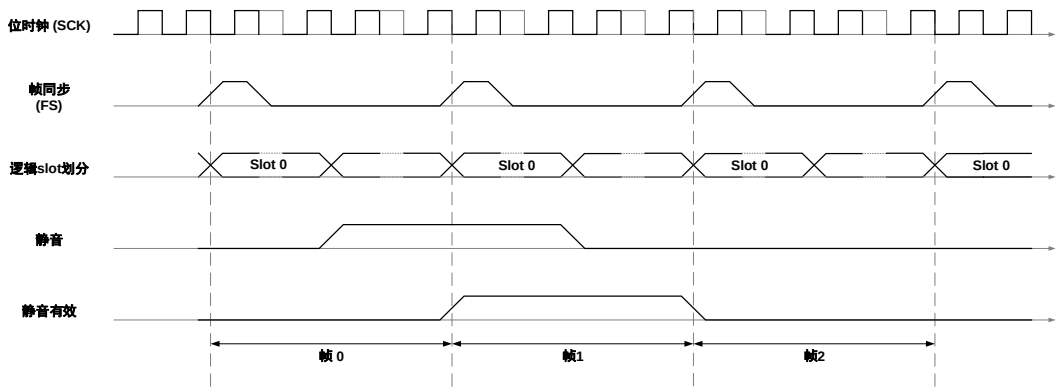
用户可以在一个帧传输期间的任何时候设置静音属性，这通过SAI_BxCFG1寄存器的MT位来配置，但是静音只会到下一个帧才生效。

如果SAI音频子模块作为发送器且已配置静音，当静音在下一个帧生效时，数据照常会从FIFO中取出，然后送入移位寄存器。唯一不同的是，SD输出是否强制为一个特定值，这个值由SAI_BxCFG1寄存器的MTVAL位决定。当MTVAL位为0时，在静音帧期间SD强制输出0，相反，当MTVAL置1时，SD输出行为得进一步根据slot总个数的配置。当slot总数小于或等于2时，静音有效的前一个帧内容会被赋值到当前静音帧。当slot总数大于2时，SD强制输出0。

配置成接收器的SAI音频子模块能够检测静音帧和产生相应的中断。一个静音帧计数器被应用到每个音频子模块上，如果接收到每个有效slot都为0的帧，那么这个帧就会被视为一个静音帧，内部的静音帧计数器增1。当SAI音频子模块使能或接收到一个非静音帧时，这个静音计数器就会复位。如果连续接收到的静音帧的个数达到SAI_BxCFG1寄存器MTFCNT位定义的值，则SAI_BxSTAT寄存器中的MTDET静音检测标志位就会置1，同时，如果使能了SAI_BxINTEN寄存器的MTDETIE位，则产生一个中断。

静音帧有效如*图34-22. 静音帧有效*所示。

图 34-22. 静音帧有效



不同配置下SD输出行为概括在*表34-10. 静音帧输出值*中。

表 34-10. 静音帧输出值

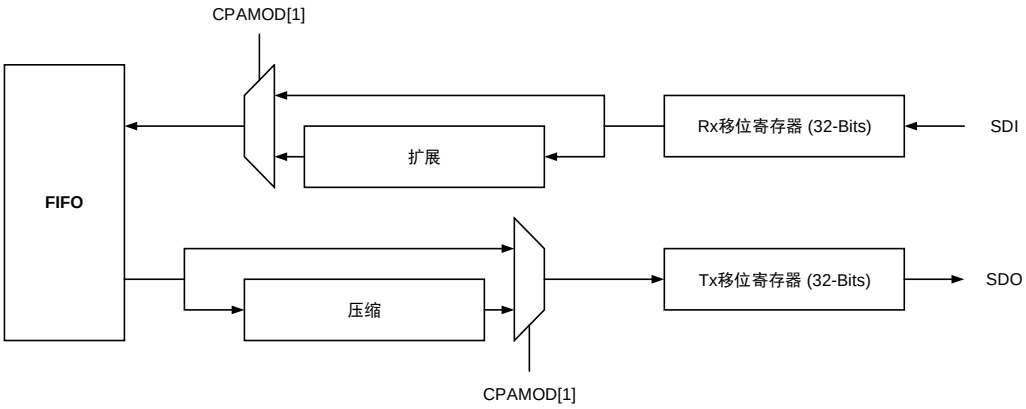
Slot个数	MTVAL= 1	MTVAL = 0
≤ 2	静音有效前的一个帧内容被赋值到SD线上输出	强制为0
>2	强制为0	强制为0

34.3.14. 压缩扩展器

压缩扩展器仅仅是一个系统，里面的信息首先经过压缩，然后在一个有限带宽的通道上传输，最后在接收端进行扩展。它常被用于减小传输电话优质语音所需的带宽，它能将13位数据压缩成8位密语，该密语由1位符号位，3位量化级以及4位分量组成。有两个支持将信号数据编码成8位编码的国际标准：A-law和Mu-law。A-law是欧洲所公认的标准，Mu-law是美国和日本所公认的标准。

A-law和Mu-law都可以应用在SAI上，这需要通过对SAI_BxCFG1寄存器进行配置来选择。音频子模块根据操作模式（OPTMOD）来选择压缩还是扩展。当音频子模块配置为发送器时，即选择压缩，相反，如果配成接收器，则选择扩展。用户可以通过设置SAI_BxCFG1寄存器的补码模式（CPLMOD）来选择1或者2的补码作为默认的数据表示。在发送模式时，无论选择哪种压缩模式，硬件首先将补码表示转换成符号量值表示，然后再送入压缩扩展器。在接收模式时，线性输出的数据从符号量值表示转换成补码表示，然后存储到FIFO中。

图 34-23. 压缩扩展数据通路



A-law 压缩扩展

A-law是CCITT推荐的压缩扩展标准，在欧洲被广泛地使用，它将线性样本值限制在12位量级。*表34-11. A-law编码*阐述了A-law编码算法，线性输入数据用符号量表示，用S代指这个符号，之后的12位表示量级。编码后输出8位宽，且按MSB表示这个符号，下表中两端的符号位S不是同一个值。A、B、C、D取0或1，x代表不关心。

表 34-11. A-law 编码

线性输入数据													A-law编码输出							
S	0	0	0	0	0	0	0	A	B	C	D	X	S	0	0	0	A	B	C	D
S	0	0	0	0	0	0	1	A	B	C	D	X	S	0	0	1	A	B	C	D
S	0	0	0	0	0	1	A	B	C	D	X	X	S	0	1	0	A	B	C	D
S	0	0	0	0	1	A	B	C	D	X	X	X	S	0	1	1	A	B	C	D
S	0	0	0	1	A	B	C	D	X	X	X	X	S	1	0	0	A	B	C	D
S	0	0	1	A	B	C	D	X	X	X	X	X	S	1	0	1	A	B	C	D
S	0	1	A	B	C	D	X	X	X	X	X	X	S	1	1	0	A	B	C	D
S	1	A	B	C	D	X	X	X	X	X	X	X	S	1	1	1	A	B	C	D

输入的数据在经过表中定义的逻辑编码后，一个反向模式应用到这个8位编码上来增加传输线上的转变密度，这对硬件性能有益。8位编码与0x55异或后再应用这个反向模式。

对A-law编码的数据进行解码从本质上来说是编码步骤的颠倒问题。*表34-12. A-law解码*说明了A-law解码算法，它在反向模式颠倒之后应用。在编码过程中丢弃的最低有效位近似的取间隔的中间值。这在线性输出数据中体现为D后紧接着的1…0。

表 34-12. A-law 解码

A-law编码输入								线性输出数据												
S	0	0	0	A	B	C	D	S	0	0	0	0	0	0	0	A	B	C	D	1
S	0	0	1	A	B	C	D	S	0	0	0	0	0	0	1	A	B	C	D	1
S	0	1	0	A	B	C	D	S	0	0	0	0	0	1	A	B	C	D	1	0
S	0	1	1	A	B	C	D	S	0	0	0	0	1	A	B	C	D	1	0	0
S	1	0	0	A	B	C	D	S	0	0	0	1	A	B	C	D	1	0	0	0
S	1	0	1	A	B	C	D	S	0	0	1	A	B	C	D	1	0	0	0	0
S	1	1	0	A	B	C	D	S	0	1	A	B	C	D	1	0	0	0	0	0
S	1	1	1	A	B	C	D	S	1	A	B	C	D	1	0	0	0	0	0	0

Mu-Law 压缩扩压

美国和日本使用Mu-law压缩扩压标准，将线性样本值限制在13位量级。Mu-law的编码和解码过程和A-law类似，不过还是有一些值得注意的差异：

1. Mu-law编码器一般操作在13位量级数据，而A-law为12位量级数据；
2. 在量化级计算之前，一个值为33的偏差被加到线性输入数据的绝对值上，用来简化量化值和分量的计算；
3. 符号位的定义是相反的，也就说，输入符号位和输出符号位相反；
4. 反向模式应用在8位编码的所有比特位上。

*表34-13. Mu-law编码*阐述了Mu-law编码算法，线性输入数据的符号位S取编码数据符号位的相反值。

表 34-13. Mu-law 编码

线性输入数据														Mu-law编码输出							
S	0	0	0	0	0	0	0	1	A	B	C	D	X	~S	0	0	0	A	B	C	D
S	0	0	0	0	0	0	1	A	B	C	D	X	X	~S	0	0	1	A	B	C	D

线性输入数据														Mu-law编码输出							
S	0	0	0	0	0	1	A	B	C	D	X	X	X	~S	0	1	0	A	B	C	D
S	0	0	0	0	1	A	B	C	D	X	X	X	X	~S	0	1	1	A	B	C	D
S	0	0	0	1	A	B	C	D	X	X	X	X	X	~S	1	0	0	A	B	C	D
S	0	0	1	A	B	C	D	X	X	X	X	X	X	~S	1	0	1	A	B	C	D
S	0	1	A	B	C	D	X	X	X	X	X	X	X	~S	1	1	0	A	B	C	D
S	1	A	B	C	D	X	X	X	X	X	X	X	X	~S	1	1	1	A	B	C	D

输入数据通过上表定义的算法编码之后，一个反向模式应用到这个8位编码上来增加传输线上的密度，这对硬件性能有益。8位编码与0xFF异或后再应用这个反向模式。

Mu-law的解码本质上是编码步骤的颠倒问题。*表34-14. Mu-law解码*说明了Mu-law解码过程，它应用在反向模式颠倒之后。在编码处理中丢弃的最低有效位近似等于这个间隔的中间值。这在线性输出数据中体现为D后紧接着的1…0。

表 34-14. Mu-law 解码

Mu-law 编码输入								线性输出数据												
S	0	0	0	A	B	C	D	~S	0	0	0	0	0	0	0	1	A	B	C	D	1
S	0	0	1	A	B	C	D	~S	0	0	0	0	0	0	1	A	B	C	D	1	0
S	0	1	0	A	B	C	D	~S	0	0	0	0	0	1	A	B	C	D	1	0	0
S	0	1	1	A	B	C	D	~S	0	0	0	0	1	A	B	C	D	1	0	0	0
S	1	0	0	A	B	C	D	~S	0	0	0	1	A	B	C	D	1	0	0	0	0
S	1	0	1	A	B	C	D	~S	0	0	1	A	B	C	D	1	0	0	0	0	0
S	1	1	0	A	B	C	D	~S	0	1	A	B	C	D	1	0	0	0	0	0	0
S	1	1	1	A	B	C	D	~S	1	A	B	C	D	1	0	0	0	0	0	0	0

34.3.15. 输出驱动

SAI可以根据SAI使能状态独立驱动每个音频子模块的帧同步（FS）、串行时钟（SCK）和串行数据（SD），这通过配置SAI_BxCFG0寄存器的输出驱动（ODRIV）来实现。

输出驱动的设定必须在SAI寄存器配置之后、SAI使能之前进行配置。

34.3.16. IO 管理

IO管理模块连接SAI的两个音频子模块，它也是两者进行连接的唯一中介。当通过设置SAI_BxCFG0寄存器的同步模式位（SYNCMOD）将音频子模块配置成与另一个子模块同步时，它们的FS、SCK和MCLK引脚会共用，同步子模块的这些引脚会释放，并可用作通用IO。当一个音频模块配置为与另一个音频模块同步时，那么它必须配成从设备。

当一个音频子模块作为发送器，且与另一个作为接收器的音频子模块同步的时候，如果它被配置为主设备，那么同步子模块会通过IO管理模块接收来自异步模块的FS和SCK信号，如果它被配置为从设备，那么会接收来自外部IO的FS和SCK信号。这个功能在双工模式中是非常有用的。

34.3.17. DMA 接口

每一个音频子模块都拥有自己的DMA接口。DMA访问的使能通过SAI_BxCFG0寄存器的DMA使能位（DMAEN）进行配置。DMA请求和FIFO请求（FFREQ）一起产生，而FIFO请求产生状态取决于FIFO阈值（FFTH）和FIFO状态（FFSTAT），这在使用DMA突发传输时是非常重要的。当音频子模块配成发送模式时，FIFO阈值必须设成一个特定的值，以保证在最坏的情况下也有足够的剩余空间来实现一个完整的DMA突发写操作，否则有可能出现FIFO上溢错误。当音频子模块配成接收模式时，FIFO阈值必须设成一个特定的值，以保证FIFO中有足够的数据来实现一个完整的DMA突发读操作，从而避免出现FIFO下溢错误。

DMA的方向和音频子模块的操作配置相关。当配置为发送器时，DMA请求将数据从数据寄存器SAI_BxDATA中加载到内部FIFO中。当配置为接收器时，DMA请求将数据从内部FIFO读到数据寄存器SAI_BxDATA中。

注意：DMA SAI通道必须在SAI寄存器配置之后使能。

34.3.18. 使能/失能

SAI音频子模块通过设置SI_BxCFG0寄存器的SAIEN位来使能，用户必须确保这个操作在音频子模块配置之后进行，SAI不支持在已经使能后再进行配置，否则将不能保证硬件行为的正确。从音频子模块必须在主音频模块使能前使能。

用户可以在有效帧传输期间的任何时候失能音频子模块，只是必须等到当前帧结束后才完全失能。

34.3.19. 错误标志位

时钟错误配置检查

时钟错误配置检测机制只有在音频子模块配置为主设备，并且时钟分频旁路（BYPASS）为0时才会使能。在这个操作模式下，用户必须保证帧宽（FWD+1）等于8到256之间且等于2的几次幂的一个值，否则状态寄存器SAI_BxSTAT中的时钟错误标志位（ERRCK）将会被置位。帧宽必须设置为2的几次幂，这是为确保在每个位时钟周期（SCK）中包含整数个主时钟（MCLK），以使得声音质量更好。

如果将中断使能寄存器SAI_BxINTEN中的时钟错误配置检测中断使能位（ERRCKIE）置1，则在出现时钟错误配置时会产生中断。

当检测到时钟错误时，SAI音频子模块将自动失能，即SAI_BxCFG0寄存器的SAIEN位被硬件清零。

音频编解码器未就绪检测

音频子模块只有在使用AC’97协议，并选择为接收器时才会检测音频编解码器未就绪状态。音频子模块从TAG（slot0）中读取音频编解码器就绪状态标志，当接收到的TAG的位15为0时，状态寄存器SAI_BxSTAT寄存器的ACNRDY会被置1，如果设置了SAI_BxINTEN中断使能寄存器中的ACNRDYIE音频编解码未就绪中断位，则产生一个中断。当检测到音频编解码器未就绪

时，当前帧的后续slot的内容将不会被送入FIFO中。

音频编解码器未就绪检测标志位通过设置SAI_BxINTC寄存器的ACNRDYC位来清除。

帧同步提前检测

音频子模块只有在配置为从设备时，才会使能帧同步提前检测机制，由于从设备才接收FS信号，FS信号到达时间对当前数据的解析至关重要。帧同步提前检测是可能的，因为帧宽、帧有效极性和帧偏移在音频子模块使能前已经确定。

帧同步提前对当前帧是没有影响的，因为FS有效边沿只有在帧结束时才能预料到。

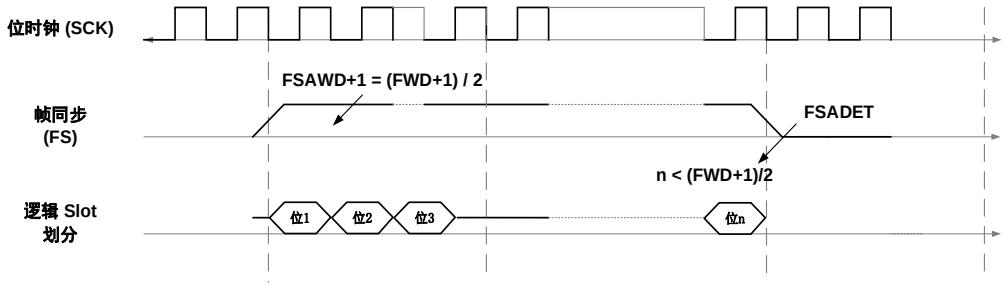
当状态寄存器SAI_BxSTAT中的帧同步提前检测标志位（FSDET）和中断使能寄存器SAI_bxINTEN中的帧同步提前检测中断使能位都置1时，产生中断。

在出现帧同步提前后，需要按照下面的步骤来进行重新同步：

1. 失能音频子模块，用户必须等到相应的音频子模块的SAIEN控制字段完全失能；
2. 设置FLUSH控制字段刷新内部FIFO；
3. 设置SAIEN再一次使能音频子模块；
4. 等待FS重新同步。

注意： 在AC’97配置模式中，这个标志位不会产生，因为SAI仅作为一个链路控制器，即使音频子模块配置为从设备，也会生成FS信号。

图34-24. 帧同步提前检测示意图



帧同步滞后检测

音频子模块只有在配置为从设备时，才会使能帧同步滞后检测机制，由于从设备才接收FS信号，FS信号到达时间对当前数据的解析至关重要。帧同步滞后检测是可能的，因为帧宽、帧有效极性和帧偏移在音频子模块使能前已经确定。

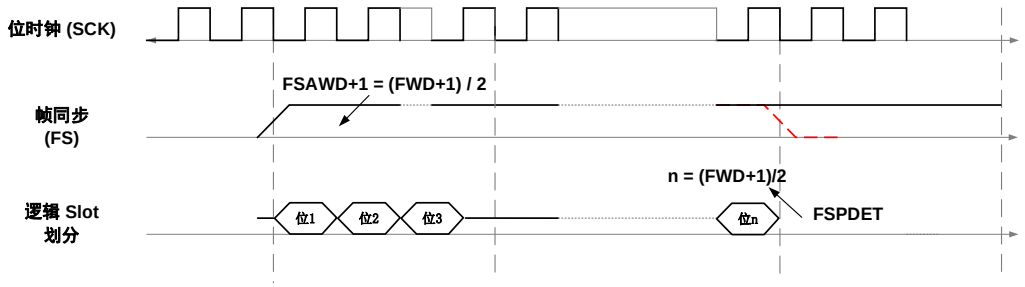
帧同步滞后可能的原因有主设备的延迟产生、外因延迟、噪音感应故障。错误的FS时序将会破坏音频子模块内部有限状态机，从而影响数据的正确传输。

当状态寄存器SAI_BxSTAT中的帧滞后提前检测标志位（FSPDET）和中断使能寄存器SAI_bxINTEN中的帧同步滞后检测中断使能位都置1时，产生中断。

为了和主设备重新同步，需要应用重新同步的步骤。

注意： 在AC’97配置模式中，这个标志位不会产生，因为SAI仅作为一个链路控制器，即使音频子模块配置为从设备，也会生成FS信号。

图34-25. 帧同步滞后检测示意图



FIFO 上溢或下溢检测

FIFO上溢和下溢标志位（OUERR）在SAI_BxSTAT状态寄存器中占同一个位，因为每个音频子模块只能配置成发送或接收。

当音频子模块配置成发送器时，在有效帧传输过程中如果FIFO为空，并且发送一个空的数据的slot，则产生下溢。如果中断使能寄存器SAI_BxINTEN的上溢或下溢中断使能位（OUERRIE）置位，则产生中断。如果发生下溢，一个重新同步过程需要按如下所示步骤进行：

1. 失能音频子模块，用户必须等到相应的音频子模块的SAIEN控制字段完全失能；
2. 设置FLUSH控制字段刷新内部FIFO；
3. 将要发送的数据填充到FIFO中；
4. 设置SAIEN再一次使能音频子模块。

通过设置SAI_BxINTC寄存器的OUERRC位来清除下溢标志位。

当音频子模块配置为接收器时，在帧传输过程中如果FIFO已满，并有一个新的slot数据接收时，发生上溢。当上溢发生时，最新接收的数据将被丢弃，也不会写值到FIFO。如果中断使能寄存器SAI_BxINTEN的上溢或下溢中断使能位（OUERRIE）置位，则产生中断。

通过设置SAI_BxINTC寄存器的OUERRC位来清除上溢标志位

注意：当DMA使能时，用户必须保证正确的DMA配置，尤其是使用DMA突发操作的时候，否则上溢和下溢都可能发生在发送或接收操作模式中。

34.3.20. 中断

*表34-15. 中断控制*概括了每个音频子模块出现的所有中断源

表 34-15. 中断控制

中断源	中断划分	中断出现条件	中断使能控制	中断清除控制
FFREQ	请求	OPTMOD为任意值	FFREQIE	读或写SAI_BxDATA
MTDET	静音	OPTMOD 为接收方	MTDETIE	MTDETC
ERRCK	错误	OPTMOD为主模式；BYPASS = 0	ERRCKIE	ERRCKC
ACNRDY	错误	OPTMOD为从模式；PROT = AC’97	ACNRDYIE	ACNRDYC
FSADET	错误	OPTMOD为从模式；PROT ≠AC’97	FSADETIE	FSADETC
FSPDET	错误	OPTMOD为从模式；PROT ≠AC’97	FSPDETIE	FSPDETC
OUERR	错误	OPTMOD为任意值	OUERRIE	OUERRC

使用下面所列的过程可以使得音频子模块从错误中断中恢复：

1. 使能相应的中断；
2. 配置SAI功能寄存器；
3. 使能中断；
4. 使能SAI音频子模块。

34.4. SAI 寄存器

SAI0 基地址：0x4001 5800

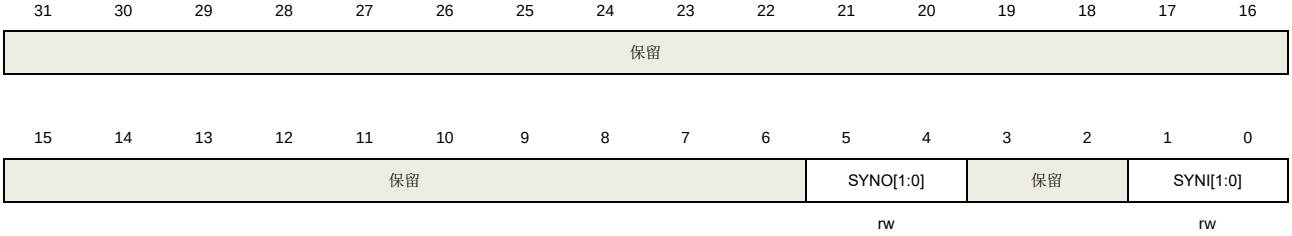
SAI1 基地址：0x4001 5C00

SAI2 基地址：0x4001 6000

34.4.1. 同步配置寄存器（SAI_SYNCFG）

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
31:6	保留	必须保持复位值。
5:4	SYNO[1:0]	同步输出 该位由软件清零或置位。 00：无同步输出信号 01：音频模块 0 与其他 SAI 进行同步 10：音频模块 1 与其他 SAI 进行同步 11：保留。必须在音频模块 1 和音频模块 2 失能时，设置这些位。 注意：当音频模块配置成 SPDIF 模式时，选择无同步输出信号。
3:2	保留	必须保持复位值。
1:0	SYNI[1:0]	同步输入 参考*表 34-2. 外部同步配置*。 必须在音频模块 0 和音频模块 1 失能时，设置这些位。 如果将两个音频模块之一定义为与外部 SAI 在同步模式下工作（SAI_BxCFG0 寄存器中的 SYNCMOD[1:0] = 10），这些位起作用。

34.4.2. 子模块 x 配置寄存器 0（SAI_BxCFG0）（x = 0, 1）

地址偏移：0x04 + 0x20 * x
复位值：0x0000 0040

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				MCLKEN	MOSPR	MDIV[5:0]						BYPASS	保留	DMAEN	SAIEN
				rw	rw	rw						rw		rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		ODRIV	MONO	SYNCMOD[1:0]		SAMPEDGE	SHIFTDIR	DATAWD[2:0]			保留	PROT[1:0]		OPTMOD[1:0]	
		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	MCLKEN	主时钟使能 0：主时钟使能 1：主时钟独立于 SAIEN 位使能
26	MOSPR	主时钟过采样率 0: MCLK = 256 * F_{fs} 1: MCLK = 512 *F_{fs}
25:20	MDIV[5:0]	主时钟分频器 0000：主分频器逻辑旁路 否则，其输出频率请参考章节*时钟分频器*公式（30-1）计算。 **注意：**当 SAI 配置为从机模式时，该控制字段无效。 **注意：**必须在使能 SAI 之前设置此控制字段。
19	BYPASS	时钟分频器逻辑旁路 0：时钟分频器应用于初级和次级分频器逻辑 1：时钟分频器逻辑被旁路
18	保留	必须保持复位值。
17	DMAEN	DMA 使能 0：DMA 失能 1：DMA 使能 **注意：**如果 SAI 被配置为接收器，则必须在 OPTMOD 控制字段之后设置 DMAEN，以避免不必要的 DMA 请求，因为 SAI 在复位后是发送器。
16	SAIEN	SAI 子模块使能 0：SAI 子模块失能 1：SAI 子模块使能。当 SPI_STAT 中的 TBE 置位时，将会在相应的 DMA 通道上产生一个 DMA 请求。
15:14	保留	必须保持复位值。
13	ODRIV	输出驱动 0：当 SAIEN 置 1 时，驱动 SAI 音频子模块输出 1：当 ODRIV 位置 1 时，立即驱动 SAI 音频子模块输出

		注意：该控制位必须在 SAI 配置后且使能前置 1。
12	MONO	立体声和单声道模式选择 0：立体声模式 1：单声道模式 单声道模式要求 slot 数等于 2，在发送器模式下，第一个 slot 的数据被复制到第二个 slot，而在接收器模式下，第二个 slot 的数据被忽略。
11:10	SYNCMOD[1:0]	同步模式 00：与其他子块异步 01：与其他子块同步，选择该模式时，用户必须配置工作模式为从机 10：音频子块与外部 SAI 嵌入式外设同步。 在这种情况下，音频子块应配置为从模式 11：保留 **注意：**在音频模块失能的情况下配置该位。 **注意：**如果协议选择为 SPDIF，则模式应配置为异步。
9	SAMPEDGE	采样时钟边沿 0：在 SCK 下降沿采样数据 1：在 SCK 上升沿采样的数据 **注意：**此控制字段在 SPDIF 模式下被忽略。 **注意：**在音频模块失能的情况下配置该位。
8	SHIFTDIR	数据传输移动方向 0：数据传输采用高位在前 1：数数据传输采用低位在前 **注意：**此控制字段在 AC’97 模式下被忽略，因为数据传输被强制为 MSB；此控制字段在 SPDIF 模式下被忽略，因为数据传输被强制为 LSB。
7:5	DATAWD[2:0]	数据宽度 000：保留 001：保留 010：8 位宽 011：10 位宽 100：16 位宽 101：20 位宽 110：24 位宽 111：32 位宽 在压扩模式下，数据宽度由算法本身固定为 8 位宽度。 **注意：**在音频模块失能的情况下配置该位。 **注意：**此控制字段在 SPDIF 模式下被忽略。 **注意：**如果选择 AC’97 协议，则只有 16 位或 20 位是可行的，否则无法保证音频子块的行为。
4	保留	必须保持复位值。
3:2	PROT[1:0]	协议选择 00：自由协议

01：SPDIF 协议
10：AC'97 洗衣
10：保留
自由协议配置允许用户调整所有帧和帧配置选项，以形成他选择的协议，如 I2S、LSB/MSB 对齐、TDM、PCM/DSP 等。
注意：在音频模块失能的情况下配置该位。

1:0	OPTMOD[1:0]	工作模式选择 00：主机发送 01：主机接收 10：从机发送 11：从机接收 如果协议选择为 SPDIF，工作模式将被强制配置为主机发送。 **注意：**在音频模块失能的情况下配置该位。

34.4.3. 子模块 x 配置寄存器 1（SAI_BxCFG1）（x = 0, 1）

地址偏移：0x08 + 0x20 * x
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPAMOD[1:0]		CPLMOD	MTFCNT[5:0]						MTVAL	MT	SDOM	FLUSH	FFTH[2:0]		
rw		rw	rw						rw	rw	rw	rw	rw		

位/位域	名称	说明
31:16	保留	必须保持复位值
15:14	CPAMOD[1:0]	压缩扩展模式 00：不使用压缩扩展模式 01：保留. 10：Mu-law 算法 11：A-law 算法 ITU-T G.711 定义了 Mu-law 和 A-law 两个主要的压缩扩展算法，它们分别将 13 位和 12 位有符号的线性 PCM 信号编码成 8 位样本。 前者为较高范围信号提供更多分辨率，后者为低信号电平提供更多量化等级。 压缩或扩展模式通过 OPTMOD[0]控制位进行选择，当音频子模块配置为发送器时，自动应用压缩模式，当音频子模块配置为接收器时，将自动应用扩展模式。补码模式通过 CPLMOD 控制位进行选择。 **注意：**只有配置 TDM 协议，才会使用压缩扩展模式
13	CPLMOD	补码模式 该控制位用于在压缩扩展模式中选择补码选项

0：数据以 1 的补码表示
1：数据以 2 的补码表示

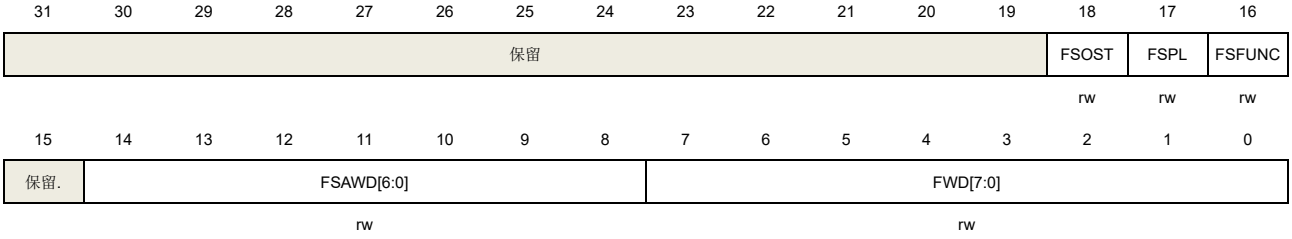
12:7	MTFCNT[5:0]	静音帧计数器 该控制位只有在接收模式中才有用。当接收到连续的静音帧数等于 MTFCNT 时，MTFDET 标志位置位，如果 MTFDETIE 置'1'，则产生中断
6	MTVAL	静音值 0：当静音打开时，串行数据线上发送 0 1：当 SLOTNUM 小于或等于 2 时，如果静音打开，则在重新数据线上重发上一个帧，否则，在静音帧期间串行数据线上发送 0 **注意：**只有当音频子模块配置为发送器时，该控制位才有意义。 **注意：**接收器只能检测 0 值静音帧，当 MTVAL 置 1 并且在静音期间重复发送上一帧时，接收器实际接收了这个值，但静音帧计数器不会计数，同样 MTDET 标志位也不会置位。
5	MT	静音 0：静音模式关闭 1：静音模式打开 **注意：**只有当音频子模块配置为发送器时，该控制位才有意义，当静音模式打开时，SD 的输出取决于 MTVAL 的配置 **注意：**如果在帧传输期间设置静音模式，静音将会在下个帧生效。
4	SDOM	串行数据输出管理 0：在音频帧期间，完全由 SAI 驱动 SD 输出 1：SD 输出在无效 slot 附近释放 **注意：**如果第一个帧的第一个 slot 的数据偏移不为 0，SD 保持释放状态直到第一个有效数据位到达。如果当前帧不是连续传输中的第一个帧，第一个 slot 的偏移区 SD 输出是否释放，这取决于上一个帧的最后一个 slot 的状态。如果上一个 slot 有效，则驱动偏移区，否则将释放。 **注意：**如果数据偏移区加上数据位宽小于 slot 宽时，我们将数据最后一位到 slot 结束之间的区域成为空白区（闲置区），在空白区 SD 输出是否释放取决于下一个 slot 是否有效，如果当前 slot 为最后一个 slot，那么它的空白区 SD 输出行为将取决于当前帧的第一个 slot，这与当前帧是否为最后一个帧无关。 **注意：**Slot 前的空白区和 slot 后的偏移区期间 SD 输出驱动与否，只取决于这个 slot。如果这个 slot 有效，则 SD 线将被驱动，否则 SD 线将被释放。
3	FLUSH	FIFO 刷新 0：无 FIFO 刷新 1：执行 FIFO 刷新 **注意：**FIFO 刷新清除 FIFO 中的所有数据，并且复位读写指针。当 SAI 失能时配置 FIFO 刷新。
2:0	FFTH[2:0]	FIFO 阈值 000：FIFO 为空 001：FIFO 1/4 满 010：FIFO 半满

011：FIFO 3/4 满
100：FIFO 全满
101：保留
110：保留
111：保留
注意：FIFO 阈值和 FIFO 状态（FFSTAT）控制位配合使用来产生 CPU 和 DMA 的 FIFO 请求（FFREQ）。

34.4.4. 子模块 x 帧配置寄存器（SAI_BxFCFG）（x = 0, 1）

地址偏移：0x0C + 0x20 * x
复位值：0x0000 0007

该寄存器只能按字（32位）访问。



位/位域	名称	说明
31:19	保留	必须保持复位值。
18	FSOST	帧同步偏移 0：FS 有效边沿声明为第一个 slot 的第一个位开始处 1：FS 有效边沿声明为当 FSOST 为 0 时的 FS 的前一个位时钟周期。 **注意：**该控制域必须在音频子模块使能之前配置，并且它在 AC’97 或 SPDIF 中是没意义的。
17	FSPL	帧同步有效极性 0：FS 有效极性为低 1：FS 有效极性为高 **注意：**该控制位必须在音频子模块使能前配置，并且它在 AC’97 或 SPDIF 中是没意义的。
16	FSFUNC	帧同步功能 0：FS 只定义帧开始 1：FS 定义帧开始和通道号 **注意：**该控制位必须在音频子模块使能前配置。 **注意：**当 FSFUNC 置 1 时，一个帧中的 slot 数（SLOTNUM+1）必须为偶数，在这种情况下，一半的 slot 将会被分派给通道 A，另一半的 slot 被分派给通道 B。如果分派到一个通道的所有 slot 数小于帧宽的一半，则在 slot 未定义时 SD 输出线释放，这和 SDOM 无关。 **注意：**当 FSFUNC 置 1 时，FS 有效宽度（FSAWD+1）必须配置为帧长的一半。

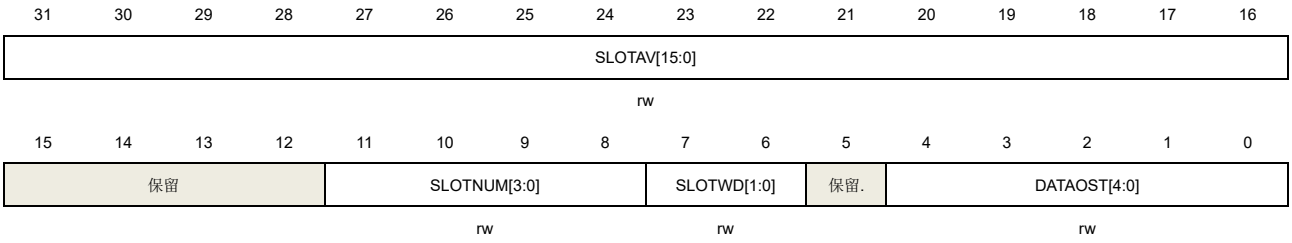
15	保留	必须保持复位值。
14:8	FSAWD[6:0]	帧同步有效宽度 **注意：**该控制位必须在音频子模块使能前配置，并且它在 AC'97 模式中是没有意义的。 **注意：**该控制位指定 FS 有效宽度为（FSAWD+1）个 SCK 时钟周期。
7:0	FWD[7:0]	帧宽度 **注意：**该控制位必须在音频子模块使能前配置，并且它在 AC'97 模式中是没有意义的。 **注意：**该控制位指定帧宽为（FWD +1）个 SCK 时钟周期，当音频子模块配置为主模式，且 BYPASS=0 时，FWD+1 的值必须等于 8 到 256 之间且为 2 的几次幂的一个值。

34.4.5. 子模块 x slot 配置寄存器（SAI_BxSCFG）（x = 0, 1）

地址偏移：0x10 + 0x20 * x

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



位/位域	名称	说明
31:16	SLOTAV[15:0]	Slot 激活向量 0：Slot 无效 1：Slot 有效 SLOTAV 向量中的每一个比特位对应到 slot0~15，如果 SLOTNUM 小于 15，则不对应的比特位被忽略。 **注意：**该控制位必须在音频子模块使能前配置。 **注意：**该控制位在 AC'97 或 SPDIF 模式中没有意义。
15:12	保留	必须保持复位值。
11:8	SLOTNUM[3:0]	一个帧中的 slot 个数 一个帧中实际的帧数为（SLOTNUM+1），并且不能超过 16。当 FSFUNC 置 1 时，slot 数必须为偶数，并且均分到两个通道。 **注意：**该控制位必须在音频子模块使能前配置。 **注意：**该控制位在 AC'97 模式中没有意义。
7:6	SLOTWD[1:0]	slot 宽度 00： Slot 宽等于数据位宽 01： Slot 为 16 位宽

10： Slot 为 32 位宽

11： 保留.

Slot 的位宽必须大于或等于数据位宽才能包含一个数据，否则 SAI 的行为将不能保证正确。

注意：该控制位必须在音频子模块使能前配置。

注意：该控制位在 AC’97 或 SPDIF 模式中没有意义。

5	保留	必须保持复位值。
4:0	DATAOST[4:0]	数据偏移 定义了在一个有效 slot 中第一个数据位的出现位置，在发送模式时，偏移区和空白区的 SD 输出 0 或 Hi-Z，这取决于 SDOM 和附近 slot 的有效状态。在接收模式时，偏移区和空白区的数据内容将会忽略。 **注意：**该控制位必须在音频子模块使能前配置。 **注意：**该控制位在 AC’97 模式中没有意义。

34.4.6. 子模块 x 中断使能寄存器（SAI_BxINTEN）（x = 0, 1）

地址偏移：0x14 + 0x20 * x

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
									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									FSPDETIE	FSADETIE	ACNRDYIE	FFREQIE	ERRCKIE	MTDETIE	OUERRIE
									rw	rw	rw	rw	rw	rw	rw

位/位域	名称	说明
31:7	保留	必须保持复位值。
6	FSPDETIE	帧同步滞后检测中断使能 0：中断失能 1：中断使能 如果 FSPDET 和 FSPDETIE 都置 1，则产生中断。 **注意：**当音频子模块配置为主模式时，该控制位无意义。 **注意：**该控制位在 AC’97 模式中没有意义。
5	FSADETIE	帧同步提前检测中断使能 0：中断失能 1：中断使能 如果 FSADET 和 FSADETIE 都置 1，则产生中断。 **注意：**当音频子模块配置为主模式时，该控制位无意义。 **注意：**该控制位在 AC’97 模式中没有意义。

4	ACNRDYIE	音频编解码器未就绪中断使能 0：中断失能 1：中断使能 如果 ACNRDY 和 ACNRDYIE 都置 1，则产生中断。 **注意：**当音频子模块配置为接收器时，该控制位才有意义。 **注意：**该控制位只有在选择 AC'97 模式时才有意义。
3	FFREQIE	FIFO 请求中断使能 0：中断失能 1：中断使能 如果 FFREQ 和 FFREQIE 都置 1，则产生中断。 **注意：**当音频子模块配置为接收器时，OPTMOD 必须在 FFREQIE 使能之前设置，以保证不会产生错误的 FIFO 请求，因为音频子模块在复位之后默认处于发送模式。
2	ERRCKIE	错误时钟中断使能，该位通过软件置 1 和清 0 0：中断失能 1：中断使能 如果 ERRCK 和 ERRCK 都置 1，则产生中断。 **注意：**该控制位只有当子模块配置为发送器，并且 BYPASS 置 0 时才可时钟分频逻辑相关。 **注意：**该控制位只用于 TDM 模式，在其他模式中是没有意义的。
1	MTDETIE	静音检测中断使能 0：中断失能 1：中断使能 如果 MTDET 和 MTDETIE 都置 1，则产生中断。 **注意：**该控制位只有在音频子模块配置为接收器时才有意义。
0	OUERRIE	FIFO 上溢或下溢中断使能 0：中断失能 1：中断使能 如果 OUERR 和 OUERRIE 都置 1，则产生中断。

34.4.7. 子模块 x 状态寄存器（SAI_BxSTAT）（x = 0, 1）

地址偏移：0x18 + 0x20 * x
复位值：0x0000 0008

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留													FFSTAT[2:0]		
														r	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									FSPDET	FSADET	ACNRDY	FFREQ	ERRCK	MTDET	OUERR
									r	r	r	r	r	r	r

位/位域	名称	说明
31:19	保留	必须保持复位值。
18:16	FFSTAT[2:0]	FIFO 状态 指示 FIFO 的满/空状态，它由硬件单独控制，根据音频子模块的操作模式有着不同的评估标准。 在 OPTMOD 配置为接收器的情况下： 000：空 001：空 <FIFO 级别<= 1/4 满 010：1/4 满 <FIFO 级别<= 1/2 满 011：1/2 满 <FIFO 级别<= 3/4 满 100：3/4 满 <FIFO 级别< 全满 101：全满 在 OPTMOD 配置为发送器的情况下： 000：空 001：空 <FIFO 级别< 1/4 满. 010：1/4 满 <= FIFO 级别< 1/2 满 011：1/2 满 <= FIFO 级别< 3/4 满 100：3/4 满 <= FIFO 级别< 全满 101：全满
15:7	保留	必须保持复位值。
6	FSPDET	帧同步滞后检测 0：收到正确的 FS 边沿 1：FS 边沿滞后接收 如果 FSPDETIE 置 1，FS 边沿接收滞后将产生中断。该标志位由 FSPDETC 控制位进行清 0。 **注意：**当音频子模块配置为接收器时，该控制位才有意义
5	FSADET	帧同步提前检测 0：收到正确的 FS 边沿 1：FS 边沿提前接收 如果 FSADETIE 置 1，FS 边沿接收提前将产生中断。该标志位由 FSADETC 控制位进行清 0。 **注意：**当音频子模块配置为接收器时，该控制位才有意义。
4	ACNRDY	音频编解码器未就绪 0：AC’97 音频编解码器就绪 1：AC’97 音频编解码器未就绪 每个帧的 TAG slot 的位 15 是 AC’97 音频编解码器就绪指示位，0 表示音频编解码器未就绪，反之，1 表示就绪。 如果 ACNRDYIE 置 1，AC’97 音频编解码器未就绪将产生中断。该标志位由 ACNRDYC 控制位进行清 0。 **注意：**该控制位只有在 AC’97 模式中才有用。
3	FFREQ	FIFO 请求

0：没有 FIFO 请求
1：FIFO 写或读请求
如果 FFREQIE 置 1，FIFO 请求将产生中断。
FIFO 的请求类型取决于音频子模块的配置，当 OPTMOD 配置为发送器，并且所有的条件满足，则产生写请求，如果配置为接收器时，则产生读请求。

2	ERRCK	时钟错误 0：正确的时钟配置 1：错误的时钟配置 如果 ERRCKIE 置 1，时钟配置错误将产生中断。该标志位由 ERRCKC 控制位进行清 0。 该控制位只有当音频子模块配置为主模式且 BYPASS 置 0 时才有意义。
1	MTDET	静音检测 0：没检测到静音 1：检测到静音 如果 MTDETIE 置 1，检测到静音将产生中断。该标志位由 MTDETC 控制位进行清 0。 当接收到 slot 全为 0 的帧的个数达到 MTFCNT 中定义的帧数时，静音检测标志位置 1.当 slot 数小于 2，且 MTVAL 置 1 时，将不能检测到静音，在发送器中，在静音之前的帧将被重复传输。
0	OUERR	上溢或下溢 0：未检测到 FIFO 上溢或下溢 1：检测到 FIFO 上溢或下溢 如果 OUERRIE 置 1，FIFO 上溢或下溢将产生中断。该标志位由 OUERRC 控制位进行清 0。 当音频子模块配置为接收器时，如果将接收到的数据存入已满 FIFO，则产生 FIFO 上溢。 当音频子模块配置为发送时，如果在 FIFO 为空出现传输请求，则产生 FIFO 下溢。

34.4.8. 子模块 x 中断标志清除寄存器（SAI_BxINTC）（x = 0, 1）

地址偏移：0x1C + 0x20 * x
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									FSPDETC	FSADETC	ACNRDYC	保留.	ERRCKC	MTDETC	OUERRC
									rw	rw	rw		rw	rw	rw

位/位域	名称	说明

31:7	保留	必须保持复位值。
6	FSPDETC	帧同步滞后检测中断清除 写 1 清除 FSPDET 标志位。 **注意：**该控制位在 AC'97 模式中没用。 **注意：**读该位将始终返回 0。
5	FSADETC	帧同步提前检测中断清除 写 1 清除 FSADET 标志位。 **注意：**该控制位在 AC'97 模式中没用。 **注意：**读该位将始终返回 0。
4	ACNRDYC	音频编解码器未就绪中断清除 写 1 清除 ACNRDY 标志位。 **注意：**该控制位只用在 AC'97 模式中。 **注意：**读该位将始终返回 0。
3	保留	必须保持复位值。
2	ERRCKC	时钟错误中断清除 写 1 清除 ERRCK 标志位。 **注意：**该控制位只有在音频模块配置为主模式，并且 BYPASS 置 0 时才有用。 **注意：**读该位将始终返回 0。
1	MTDETC	静音检测中断清除 写 1 清除 MTDET 标志位。 **注意：**读该位将始终返回 0。
0	OUERRC	上溢或下溢中断清除 写 1 清除 OUERR 标志位。 **注意：**读该位将始终返回 0。

34.4.9. 子模块 x 数据寄存器（SAI_BxDATA）（x = 0, 1）

地址偏移：0x20 + 0x20 * 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
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	说明
31:0	DATA[31:0]	数据

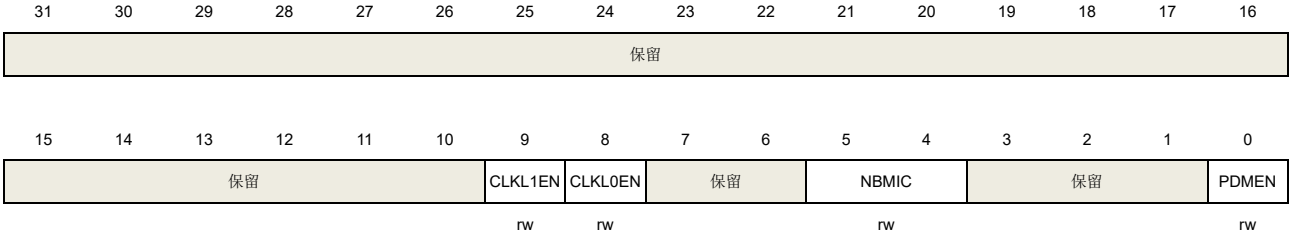
写和读操作直接体现在 FIFO 中。

34.4.10. PDM 控制寄存器（SAI_PDM）

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	说明
31:10	保留	必须保持复位值。
9	CLKL1EN	PDM 时钟线 1 使能 0：PDM 时钟线 1 失能 1：PDM 时钟线 1 使能
8	CLKL0EN	PDM 时钟线 0 使能 0：PDM 时钟线 0 失能 1：PDM 时钟线 0 使能
7:6	保留	必须保持复位值。
5:4	NBMIC	选择麦克风数量 00：2 个麦克风 01：4 个麦克风 10：6 个麦克风 11：8 个麦克风
3:1	保留	必须保持复位值。
0	PDMEN	PDM 使能 0：PDM 失能 1：PDM 使能

34.4.11. PDM 配置寄存器（SAI_PDMCFG）

地址偏移：0x48

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



		rw				rw				rw				rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	DPR1[2:0]			保留	DPL1[2:0]			保留	DPR0[2:0]			保留	DPL0[2:0]		
		rw				rw				rw				rw	

位/位域	名称	说明
31	保留	必须保持复位值。
30:28	DPR3[2:0]	第三组麦克风右通道数据流延迟周期 000：无延迟 010：延迟 1 个 T_{SAI_CK} 周期 … 111：延迟 7 个 T_{SAI_CK} 周期
27	保留	必须保持复位值。
26:24	DPL3[2:0]	第三组麦克风左通道数据流延迟周期 000：无延迟 010：延迟 1 个 T_{SAI_CK} 周期 … 111：延迟 7 个 T_{SAI_CK} 周期
23	保留	必须保持复位值。
22:20	DPR2[2:0]	第二组麦克风右通道数据流延迟周期 000：无延迟 010：延迟 1 个 T_{SAI_CK} 周期 … 111：延迟 7 个 T_{SAI_CK} 周期
19	保留	必须保持复位值。
18:16	DPL2[2:0]	第二组麦克风左通道数据流延迟周期 000：无延迟 010：延迟 1 个 T_{SAI_CK} 周期 … 111：延迟 7 个 T_{SAI_CK} 周期
15	保留	必须保持复位值。
14:12	DPR1[2:0]	第一组麦克风右通道数据流延迟周期 000：无延迟 010：延迟 1 个 T_{SAI_CK} 周期 … 111：延迟 7 个 T_{SAI_CK} 周期
11	保留	必须保持复位值。
10:8	DPL1[2:0]	第一组麦克风左通道数据流延迟周期 000：无延迟

		010：延迟 1 个 T_{SAI_CK} 周期 … 111：延迟 7 个 T_{SAI_CK} 周期
7	保留	必须保持复位值。
6:4	DPR0[2:0]	第零组麦克风右通道数据流延迟周期 000：无延迟 010：延迟 1 个 T_{SAI_CK} 周期 … 111：延迟 7 个 T_{SAI_CK} 周期
3	保留	必须保持复位值。
2:0	DPL0[2:0]	第零组麦克风左通道数据流延迟周期 000：无延迟 010：延迟 1 个 T_{SAI_CK} 周期 … 111：延迟 7 个 T_{SAI_CK} 周期

35. 图像处理加速器（IPA）（适用于 GD32H737_757_759）

35.1. 简介

IPA 提供从某一个或两个源图像到目标图像的可配置的，灵活的图像处理功能。它支持以下四种转换模式：

- 复制某一源图像到目标图像中；
- 复制某一源图像到目标图像中并同时进行特定的格式转换；
- 将两个不同的源图像进行混合，并将得到的结果进行特定的颜色格式转换；
- 用特定的颜色填充目标图像区域。

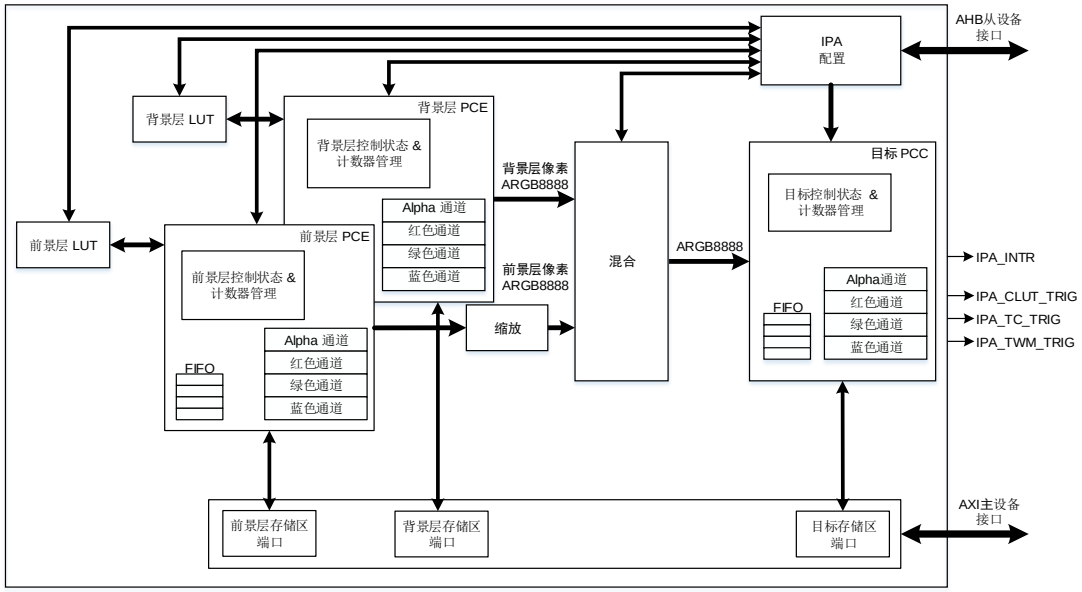
前景层图像支持 16 种像素格式，背景层图像支持 11 种像素格式，每像素从 4 位到最高 32 位，对于目标图像支持 5 种像素格式，每像素从 16 位到最高 32 位。采用间接像素模式时，IPA 为两个源图像分别提供了 256*32 的颜色查找表。

35.2. 主要特性

- 一个访问存储器的 AXI 主设备接口；
- 一个支持 8 位，16 位，32 位的 IPA 配置的 AHB 从设备接口；
- 3 个 4 双字深度的 64 位 FIFO 独立用于源图像和目标图像；
- 支持四种像素格式转换模式：
  - 复制某一源图像到目标图像中；
  - 复制某一源图像到目标图像中并同时进行特定的颜色格式转换；
  - 将两个不同的源图像进行混合，并将得到的结果进行特定的颜色格式转换；
  - 用特定的颜色填充目标图像区域。
- 支持两源图像独立配置 LUT 大小；
- 支持两种 LUT 像素格式；
- 支持 LUT 自动加载；
- 支持传输挂起或停止；
- 对于源图像和目标图像，支持独立配置行偏移量；
- 支持源图像和目标图像独立预定义的像素通道值；
- 分别支持两源图像 3 种 alpha 通道值计算算法；
- 对于前景层图像，支持 16 种像素格式；
- 对于背景层图像，支持 11 种像素格式；
- 对于目标图像，支持 5 种像素格式；
- 支持配置图像大小；
- 支持 AXI 总线带宽自动调节；
- 支持一个带有六种事件标志位的中断；
- 支持中断使能和清除；
- 支持十进制缩放和双线性缩放；
- 支持图像旋转（0、90、180、270 度）；
- 前景层图像支持隔行输入。

35.3. 结构框图

图 35-1. IPA 模块框图



如*图 35-1. IPA 模块框图*所示，IPA 包含 7 个主要部分：

- 通过 AHB 从设备接口配置 IPA；
- 通过 AXI 主设备接口访问图像数据；
- 前景层和背景层 LUT；
- 前景层和背景层像素通道扩展（PCE）；
- 前景层缩放；
- 前景层和背景层像素混合；
- 目标像素通道压缩（PCC）。

35.4. 信号描述

IPA 信号的描述如*表 35-1. IPA 信号描述*所示。

表 35-1. IPA 信号描述

I/O 口	类型	描述
IPA_INTR	O	IPA 全局中断信号
IPA_CLUT_TRIG	O	IPA CLUT 传输完成触发信号
IPA_TC_TRIG	O	IPA 传输完成突发信号
IPA_TWM_TRIG	O	IPA 传输水印触发信号

35.5. 功能概述

IPA是一个像素格式转换器，它支持多种转换模式，允许用户通过配置IPA对应寄存器的相应位

灵活的配置转换模式，前景层，背景层，目标像素格式及其行偏移。除了LUT仅支持32位访问外，其他所有IPA寄存器都可以通过AHB从设备接口进行8位，16位或32位访问。

IPA支持4种转换模式，它可以通过IPA_CTL寄存器的PFCM位配置，如*表35-2. IPA转换模式*所示：

- 复制前景层图像到目标图像中
  在这种模式中，前景层存储区的像素数据复制到目标存储区而不进行像素转换，所以前景层和目标图像的像素格式没有意义。前景层的像素格式仅定义了每像素的位数。
- 转换前景层图像到目标图像
  在这种模式中，前景层的像素数据从前景层的像素格式转换成目标像素格式，然后写入目标存储区中。如果前景层的像素格式是非直接的（L8, AL44, AL88, L4），读取前景层存储区域中的数据作为索引从前景层 LUT 获取像素数据。
- 转换和混合前景层和背景层图像到目标图像
  在这种模式中，前景层和背景层的像素数据首先由其原来的格式转换为‘ARGB8888’，然后前景层和背景层像素数据成对的混合并从‘ARGB8888’转换为目标像素格式，写入目标存储区中。
  如果前景层的像素格式是非直接的，读取前景层存储区域中的数据作为索引从前景层 LUT 获取像素数据。
  如果背景层的像素格式是非直接的，读取背景层存储区域中的数据作为索引从背景层 LUT 获取像素数据。
- 用特定颜色填充到目标图像中
  在这种模式中，目标图像被特定的像素填充，该像素的值被预先定义在对应的目标寄存器中。

表 35-2. IPA 转换模式

PFCM[1:0]	转换模式		像素转换	混合
	源	目的		
00	前景层图像	目标图像	否	否
01	前景层图像	目标图像	是	否
10	前景层图像和背景层图像	目标图像	是	是
11	在寄存器中预定义的像素值	目标图像	否	否

35.5.1. 转换操作

一次 IPA 操作包含以下 7 个步骤：

1) 从前景层存储区（基地址配置在 IPA_FMADDR 寄存器中）读取数据，如果前景层像素格式是非直接的，则从前景层 LUT 获取像素数据。
2) 扩展前景层像素数据到一个 32 位的值，并根据 IPA_FPCTL 寄存器的 FAVCA 位计算 alpha 通道的值。
3) 从背景层存储区（基地址配置在 IPA_BMADDR 寄存器中）读取数据，如果背景层像素格式是非直接的，则从背景层 LUT 获取像素数据。
4) 扩展背景层像素数据到一个 32 位的值，并根据 IPA_BPCTL 寄存器的 BAVCA 位计算 alpha 通道的值。
5) 混合扩展后的前景层和背景层像素数据。

6) 压缩像素数据为 IPA_DPCTL 寄存器 DPF 位指定的目标区像素格式。
7) 将转换后的像素数据写到目标存储区（基地址配置在 IPA_DMADDR 寄存器中）。

对于前景层和背景层的像素数据处理，IPA 提供了 3 个 8 双字深度的 64 位 FIFO。对于目标层像素数据处理，IPA 提供了 1 个 4 双字深度的 64 位 FIFO。前景层和背景层 FIFO 存储从对应的源存储区读得的数据，目标 FIFO 存储处理过的像素数据，当 AXI 总线空闲的时候，这些数据将会被写入目标存储器。

如果 IPA_CTL 寄存器的 PFCM 位域被配置成 '00' 或 '01'，只有前景层 FIFO 和目标层 FIFO 被激活。如果 IPA 的操作为用特定的颜色填充目标图像，则不需要任何一个 FIFO。

35.5.2. 前景层和背景层 LUT

IPA 提供了两个 LUT 来存储像素值，以便非直接像素格式使用。如果像素格式是非直接的，使能 IPA 传输之前，像素数据必须已经被写入 LUT 中。LUT 中的像素数据可以通过以下两种方法更新：

- 自动加载：
  使能 IPA_FPCTL/IPA_BPCTL 寄存器的 FLLEN/BLLEN 位。IPA_FPCTL 或 IPA_BPCTL 寄存器的 FCNP 或 BCNP 位定义了要自动加载的像素的数目，它等于 FCNP+1 或 BCNP+1。
- 软件编程：
  像素数据可直接通过 IPA 从设备接口写入相应的 LUT 存储器地址。前景层 LUT 的基地址偏移是 0x0400，背景层 LUT 的基地址偏移是 0x0800。

LUT支持两种像素格式，分别为'ARGB8888'和'RGB888'，由IPA_FPCTL或IPA_BPCTL寄存器的FLPF或BLPF位决定，如*表35-3. 前景层和背景层CLUT像素格式*所示。

表 35-3. 前景层和背景层 CLUT 像素格式

BLPF/FLPF	LUT 像素格式	存储地址			
		基地址+0x3	基地址+0x2	基地址+0x1	基地址+0x0
0	ARGB8888	A_0[7:0]	R_0[7:0]	G_0[7:0]	B_0[7:0]
1	RGB888	R_3[7:0]	G_3[7:0]	B_3[7:0]	R_2[7:0]
		G_2[7:0]	B_2[7:0]	R_1[7:0]	G_1[7:0]
		B_1[7:0]	R_0[7:0]	G_0[7:0]	B_0[7:0]

注意：如果 LUT 的像素格式是'RGB888'，当自动加载 LUT 的像素数据时，alpha 值为固定的 0xFF。

35.5.3. 前景层和背景层像素通道扩展（PCE）

若 IPA 传输模式需要进行像素格式转换，前景层或背景层像素数据需要由原来的格式扩展为'ARGB8888'格式。

IPA_FPCTL和IPA_BPCTL寄存器的FPF和BPF位定义了前景层和背景层的像素格式。如*表35-4. 前景层和背景层像素格式*所示。

一个像素包含以下 5 个通道：

- Alpha 通道 A：透明度，0x00：透明的；0xFF：不透明的。
- 红色通道 R：红色值，0x00：没有红色；0xFF：全红色。
- 绿色通道 G：绿色值，0x00：无绿色；0xFF：全绿色。
- 蓝色通道 B：蓝色值，0x00：无蓝色；0xFF：全蓝色。
- 亮度通道：在 IPA 中，亮度通道的值是索引值，从背景层或前景层的 LUT 中获得像素数据。

表 35-4. 前景层和背景层像素格式

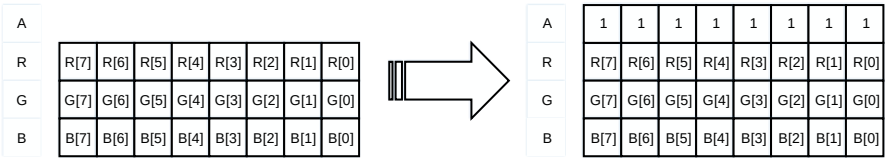
BPF[3:0]/FPF[3:0]	像素格式	存储地址			
		基地址 + 0x3	基地址 + 0x2	基地址 + 0x1	基地址 + 0x0
0000	ARGB8888	A_0[7:0]	R_0[7:0]	G_0[7:0]	B_0[7:0]
0001	RGB888	R_3[7:0]	G_3[7:0]	B_3[7:0]	R_2[7:0]
		G_2[7:0]	B_2[7:0]	R_1[7:0]	G_1[7:0]
		B_1[7:0]	R_0[7:0]	G_0[7:0]	B_0[7:0]
0010	RGB565	R_1[4:0]G_1[5:3]	G_1[2:0]B_1[4:0]	R_0[4:0]G_0[5:3]	G_0[2:0]B_0[4:0]
0011	ARGB1555	A_1[0]R_1[4:0]G_1[4:3]	G_1[2:0]B_1[4:0]	A_0[0]R_0[4:0]G_0[4:3]	G_0[2:0]B_0[4:0]
0100	ARGB4444	A_1[3:0]R_1[3:0]	G_1[3:0]B_1[3:0]	A_0[3:0]R_0[3:0]	G_0[3:0]B_0[3:0]
0101	L8	L_3[7:0]	L_2[7:0]	L_1[7:0]	L_0[7:0]
0110	AL44	A_3[3:0]L_3[3:0]	A_2[3:0]L_2[3:0]	A_1[3:0]L_1[3:0]	A_0[3:0]L_0[3:0]
0111	AL88	A_1[7:0]	L_1[7:0]	A_0[7:0]	L_0[7:0]
1000	L4	L_7[3:0]L_6[3:0]	L_5[3:0]L_4[3:0]	L_3[3:0]L_2[3:0]	L_1[3:0]L_0[3:0]
1001	A8	A_3[7:0]	A_2[7:0]	A_1[7:0]	A_0[7:0]
1010	A4	A_7[3:0]A_6[3:0]	A_5[3:0]A_4[3:0]	A_3[3:0]A_2[3:0]	A_1[3:0]A_0[3:0]
1011	YUV444*	Y_3[7:0]	U_3[7:0]	V_3[7:0]	Y_2[7:0]
		U_2[7:0]	V_2[7:0]	Y_1[7:0]	U_1[7:0]
		V_1[7:0]	Y_0[7:0]	U_0[7:0]	V_0[7:0]
1100	UYVY422*	Y_1[7:0]	V_{01}[7:0]	Y_0[7:0]	U_{01}[7:0]
1101	VYUY422*	Y_1[7:0]	U_{01}[7:0]	Y_0[7:0]	V_{01}[7:0]
1110	YUV420*	…	…	…	…
		V_{2367}[7:0]	U_{2367}[7:0]	V_{0145}[7:0]	U_{0145}[7:0]
		…	…	…	…
		Y_7[7:0]	Y_6[7:0]	Y_5[7:0]	Y_4[7:0]@line2
		Y_3[7:0]	Y_2[7:0]	Y_1[7:0]	Y_0[7:0]@line1
1111	YVU420*	…	…	…	…
		U_{2367}[7:0]	V_{2367}[7:0]	U_{0145}[7:0]	V_{0145}[7:0]
		…	…	…	…
		Y_7[7:0]	Y_6[7:0]	Y_5[7:0]	Y_4[7:0]@line2
		Y_3[7:0]	Y_2[7:0]	Y_1[7:0]	Y_0[7:0]@line1

注意*： YUV 格式仅前景层图像支持。

如果像素格式是‘RGB888’，当扩展像素数据时，alpha通道值被设置为0xFF，如*图35-2. 从‘RGB888’到‘ARGB8888’像素格式扩展*所示。

图 35-2. 从‘RGB888’到‘ARGB8888’像素格式扩展

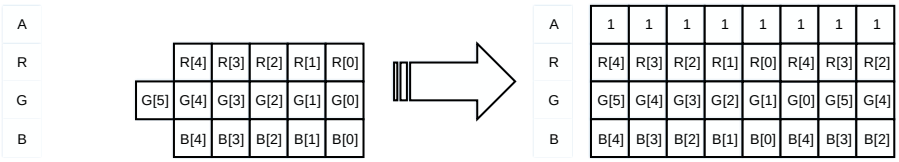
- RGB888 → ARGB8888



如果像素格式是‘RGB565’，当扩展像素数据时，alpha通道值等于0xFF。红绿蓝通道值扩展到8位，扩展后高位为通道值，通道值的高位值填充到低位。如*图35-3. 从‘RGB565’到‘ARGB8888’像素格式扩展*所示。

图 35-3. 从‘RGB565’到‘ARGB8888’像素格式扩展

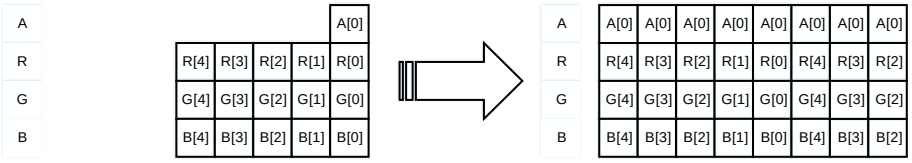
- RGB565 → ARGB8888



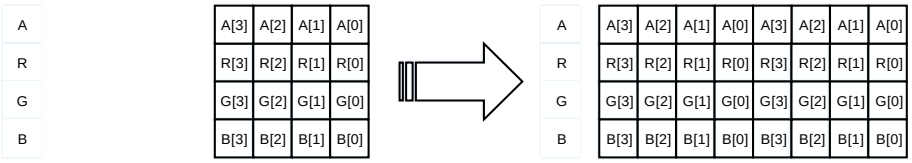
如果像素格式是‘ARGB1555’或‘ARGB4444’，每通道值将扩展到8位，扩展后高位为通道值，通道值的高位值填充到低位。如*图35-4. 从‘ARGB1555’或‘ARGB4444’到‘ARGB8888’像素格式扩展*所示。

图 35-4. 从‘ARGB1555’或‘ARGB4444’到‘ARGB8888’像素格式扩展

- ARGB1555 → ARGB8888



- ARGB4444 → ARGB8888



如果像素格式是‘L8’或‘L4’，8 位亮度通道值(当像素格式为‘L4’，高位补 0)作为索引值从 LUT 获得像素数据。

如果像素格式是‘AL44’，8 位亮度通道值(高位补 0)作为索引值从 LUT 获得红、绿、蓝通道值。Alpha 通道值将扩展到 8 位，扩展后高位为通道值，填充通道值的高位值到低位。

如果像素格式是‘AL88’，只有红，绿，蓝通道值通过 8 位亮度通道从 LUT 获得。

如果像素格式是‘A8’，红，绿，蓝通道值分别等于 IPA_FPV 寄存器的 FPDRV，FPDGV 位以及 FPDBV 位（或 IPA_BPV 寄存器的 BPDRV 位，BPDGV 位以及 BPDBV 位）。

如果像素格式是‘A4’，Alpha 通道值将扩展到 8 位，扩展后高位为通道值，填充通道的高位值

到低位。红，绿，蓝通道值分别等于 IPA_FPV 寄存器的 FPDRV，FPDGV 位以及 FPDBV 位（或 IPA_BPV 寄存器的 BPDRV 位，BPDGV 位以及 BPDBV 位）。

PCE 还支持从 YUV / YCbCr 到 ARGB8888 格式的颜色转换。以下等式用于执行此过程。常量将作为二进制补码值存储在 IPA_CSCC_CFGx（x = 0…2）控制寄存器中，这种方式带来了实现的灵活性以及视频编码和解码操作的差异性。此外，它还提供了一种软件机制来操纵亮度或对比度。

$$R = C0(Y+Yoffset) + C1(V+UVoffset) \quad (35\text{-}1)$$

$$G = C0(Y+Yoffset) + C3(U+UVoffset) + C2(V+UVoffset) \quad (35\text{-}2)$$

$$B = C0(Y+Yoffset) + C4(U+UVoffset) \quad (35\text{-}3)$$

*表 35-5. YUV 和 YCbCr 模式的期望系数*给出了 YUV 和 YCbCr 模式下的期望系数。

表 35-5. YUV 和 YCbCr 模式的期望系数

Coff	YUV	YCbCr
YOFF	0x000	0x1F0 (-16)
UVOFF	0x000	0x180 (-128)
C0	0x100 (1.00)	0x129 (1.164)
C1	0x123 (1.140)	0x198 (1.596)
C2	0x76B (-0.581)	0x72F (-0.813)
C3	0x79B (-0.394)	0x79B (-0.392)
C4	0x208 (2.032)	0x204 (2.017)

IPA 支持通过 3 种算法调制 alpha 通道值，由 IPA_FPCTL 或 IPA_BPCTL 寄存器的 FAVCA 或 BAVCA 位决定。如*表 35-6. Alpha 通道值调制*所述。

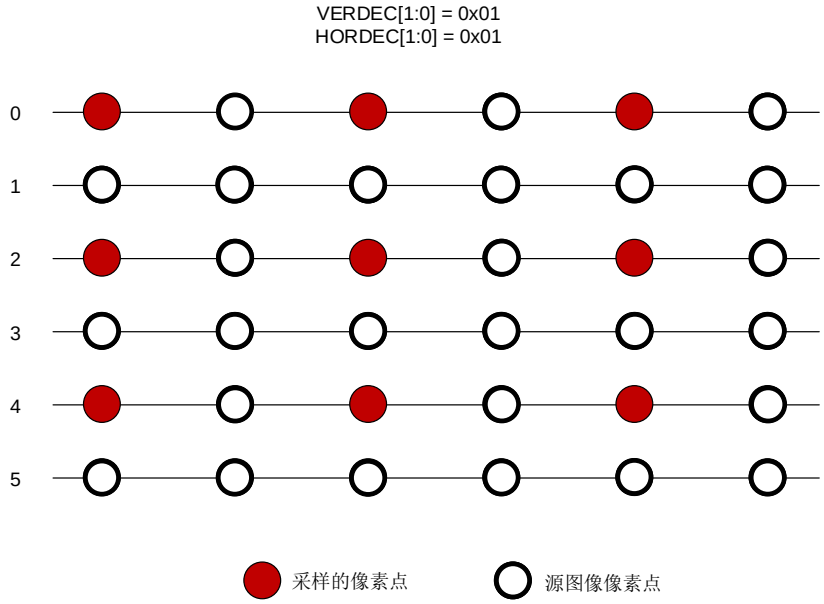
表 35-6. Alpha 通道值调制

FAVCA/BAVCA[1:0]	alpha 计算算法
00/11	无影响，等于原来的值
01	等于 IPA_FPCTL 或 IPA_BPCTL 寄存器 FPDAV 或 BPDAV 位的值
10	等于 FPDAV 或 BPDAV 位的值乘以原来 alpha 的值再除以 255

35.5.4. 前景通道缩放

IPA 实现了一个双线性缩放过滤器，在前景通道中将输入图像调整为不同的分辨率。IPA_DPCTL 寄存器中的 HORDEC 和 VERDEC 定义了水平和垂直预抽取滤波器控制。IPA_BSCTL 寄存器中的 XSCALE 和 YSCALE 为前景提供 X 和 Y 缩放因子。抽取滤波器和双线性滤波器的最大缩减因子分别为 8 和 2，因此最大缩放因子可以达到 16。抽取滤波器的详细采样方法如*图 35-5. 抽取滤波器的采样方法*所示。

图 35-5. 抽取滤波器的采样方法

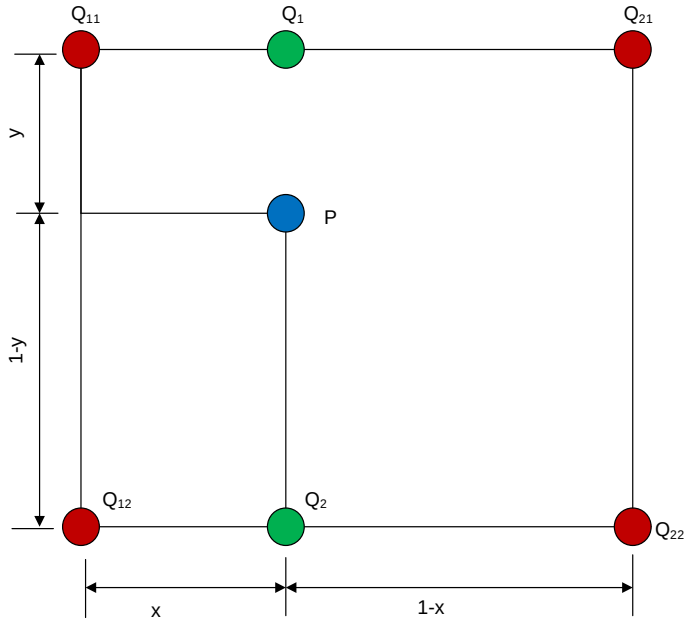


每个轴的最大缩小系数为 1/2，最大放大系数为 2^12。为了实现缩放功能，需要将缩放因子的倒数写入 IPA_BSCTL 寄存器。例如：

```
REG32(IPA_BSCTL) = 0x20002000;      //1/2 倍缩放(0x2.0)
REG32(IPA_BSCTL) = 0x18001800;      //2/3 倍缩放(0x1.8)
REG32(IPA_BSCTL) = 0x08000800;      //2 倍缩放(0x0.8)
REG32(IPA_BSCTL) = 0x04000400;      //4 倍缩放(0x0.4)
```

双线性滤波器是四个最近像素的加权平均值。假设 P 是输出像素，Q_{11}、Q_{12}、Q_{21}、Q_{22}是周围的四个源图像的像素点。如*图 35-6. 双线性缩放图*所示。

图 35-6. 双线性缩放图



输出像素 P 的值由以下等式计算。

$$P = Q_1(1-y) + Q_2*y \tag{35-4}$$

其中Q_1和Q_2的等式如下。

$$Q_1 = Q_{11}*(1-x) + Q_{21}*x \tag{35-5}$$

$$Q_2 = Q_{12}*(1-x) + Q_{22}*x \tag{35-6}$$

仅当 PFCM 等于 2'b01 或 2'b10 时，缩放才有效。

35.5.5. 混合

若 IPA 的传输模式需要进行像素混合时，扩展之后的前景层和背景层像素数据需要成对的混合并获得一个 32 位像素值。

Alpha 通道值的混合基于下面的公式(A_F 是前景层 alpha 值, A_B 是背景层 alpha 值)：

$$A_{mix} = \frac{A_F \times A_B}{255} \tag{35-7}$$

$$A_{blend} = A_F + A_B - A_{mix} \tag{35-8}$$

红，绿，蓝通道值的混合基于下面的公式(R_F, G_F, B_F 是前景层的红，绿，蓝值; R_B, G_B, B_B 是背景层的红，绿，蓝值)：

$$R_{blend} = \frac{R_F \times A_F + R_B \times A_B - R_B \times A_{mix}}{A_{blend}} \tag{35-9}$$

$$G_{blend} = \frac{G_F \times A_F + G_B \times A_B - G_B \times A_{mix}}{A_{blend}} \tag{35-10}$$

$$B_{blend} = \frac{B_F \times A_F + B_B \times A_B - B_B \times A_{mix}}{A_{blend}} \tag{35-11}$$

注意：1)上述公式中的除法结果是向下取整。

2)如果 A_{blend} 等于 0, R_{blend}, G_{blend} 和 B_{blend} 等于‘0xFF’。

3)背景和目标层的宽度和高度要一致。

35.5.6. 目标像素通道压缩（PCC）

如果在 IPA 传输模式需要进行像素转换，在像素数据写入目标存储区之前，需要由‘ARGB8888’压缩为目标像素格式。

IPA_DPCTL 寄存器的 DPF 位定义了目标图像的像素格式。如*表 35-7. 目标像素格式*所示。

表 35-7. 目标像素格式

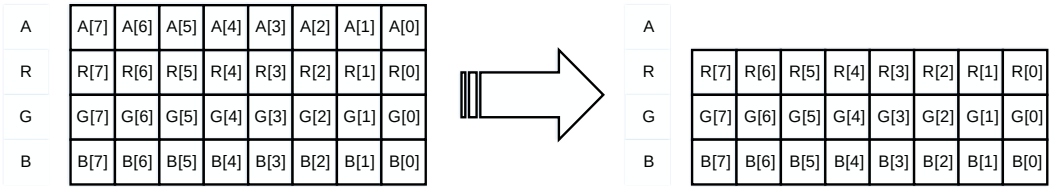
DPF[2:0]	像素格式	存储地址			
		基地址+0x3	基地址+0x2	基地址+0x1	基地址+0x0
000	ARGB8888	A_0[7:0]	R_0[7:0]	G_0[7:0]	B_0[7:0]
001	RGB888	R_3[7:0]	G_3[7:0]	B_3[7:0]	R_2[7:0]
		G_2[7:0]	B_2[7:0]	R_1[7:0]	G_1[7:0]
		B_1[7:0]	R_0[7:0]	G_0[7:0]	B_0[7:0]
010	RGB565	R_1[4:0]G_1[5:3]	G_1[2:0]B_1[4:0]	R_0[4:0]G_0[5:3]	G_0[2:0]B_0[4:0]
011	ARGB1555	A_1[0]R_1[4:0]G_1[4:3]	G_1[2:0]B_1[4:0]	A_0[0]R_0[4:0]G_0[4:3]	G_0[2:0]B_0[4:0]
100	ARGB4444	A_1[3:0]R_1[3:0]	G_1[3:0]B_1[3:0]	A_0[3:0]R_0[3:0]	G_0[3:0]B_0[3:0]

注意：如果 IPA_CTL 寄存器的 PFCM 位等于‘00’（拷贝前景层图像到目标图像），DPF 位无意义，IPA_FPCTL 寄存器的 FPF 位决定了源图像和目标图像每像素的位数。

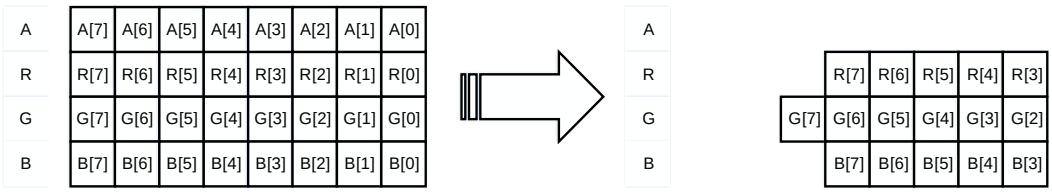
如*图35-7. 像素压缩*所示，目标像素通道压缩通过丢弃低位实现。

图 35-7. 像素压缩

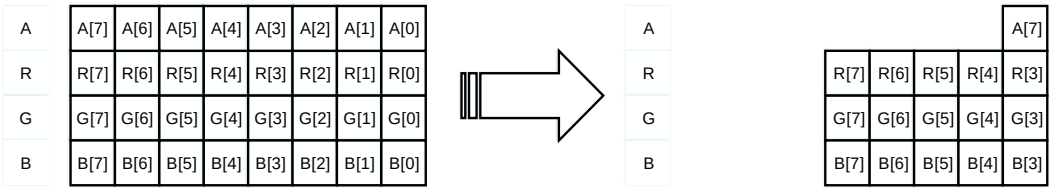
- ARGB8888 → RGB888



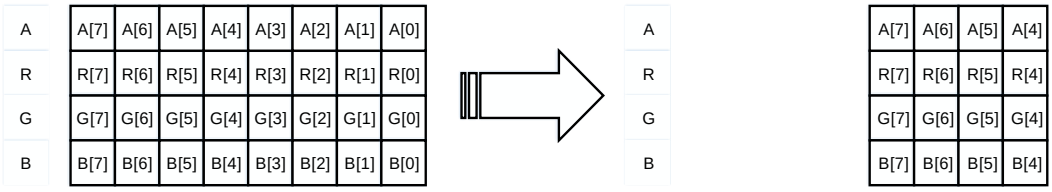
- ARGB8888 → RGB565



- ARGB8888 → ARGB1555



- ARGB8888 → ARGB4444



35.5.7. 旋转

IPA 支持 0、90、180 和 270 度的旋转。当 IPA_DPCTL 中的 ROT 位域被配置时，IPA 逐行读取输入图像并将每个转换后的输出像素放入旋转后生成的输出像素地址。

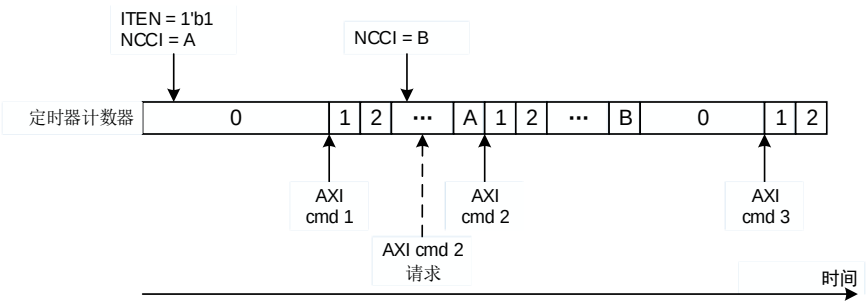
35.5.8. 内部定时器

为了减少 IPA 使用的 AXI 总线的带宽，在 IPA 传输与 LUT 自动加载时，IPA 会自动在两个连续的 AXI 请求之间插入若干时钟周期，这个功能通过一个内部定时器实现。

置位 IPA_ITCTL 寄存器的 ITEN 位，内部定时器使能。IPA_ITCTL 寄存器的 NCCI 位定义了两个连续的 AXI 写或读地址通道命令之间插入的时钟周期数的最小值。若内部定时器没有使能，NCCI 没有意义。

当IPA传输或LUT自动加载正在进行时，若改变NCCI的值，对内部计数器的当前计数没有影响，从下次计数有作用；如*图35-8. 内部定时器操作*所示。

图 35-8. 内部定时器操作



35.5.9. 行标记

软件可通过标记行号来了解当前 IPA 传输的进度，被标记的行号可以通过 IPA_LM 寄存器的 LM 位配置。当且仅当标记行的最后一个像素数据被写入目标存储区时，IPA_INTF 寄存器中的 TLMIF 位会被置起。

注意：如果 LM 位等于 0，无标志位置位。

35.5.10. 传输流

软件置位 IPA_FPCTL/IPA_BPCTL 寄存器的 FLLEN/BLLEN 位，前景层或背景层 LUT 开始自动加载。一旦 LUT 自动加载开始，FLLEN/BLLEN 位变为传输标志位，用于指示 LUT 自动加载是否完成，且软件向其写 0 没有意义；当加载完成时，FLLEN/BLLEN 位会被自动清 0。

软件置位 IPA_CTL 寄存器的 TEN 位，IPA 开始传输。一旦传输开始，TEN 位变为传输标志位，用于指示 IPA 传输是否完成，且软件向其写 0 没有意义；当传输完成时，TEN 位会被自动清 0。

在 IPA 传输或 LUT 自动加载正在工作时，软件可通过置位 IPA_CTL 寄存器的 THU 位挂起当前传输；当软件清除 THU 位后，传输继续。若无前景层/背景层 LUT 自动加载和 IPA 传输使能时，设置 THU 位无影响，读该位的值为 0。

通过置位 IPA_CTL 寄存器的 TST 位可以停止前景层/背景层 LUT 自动加载和 IPA 传输。复位 IPA_FPCTL/IPA_BPCTL 寄存器的 FLLEN/BLLEN 位或 IPA_CTL 寄存器的 TEN 位可以让 LUT 加载或 IPA 传输立即停止，即使 IPA 传输或 LUT 自动加载正在工作或挂起。当当前的传输停止的后，TST 位自动复位。若无前景层/背景层 LUT 自动加载和 IPA 传输使能时，设置 TST 位无影响，读该位的值为 0。

前景层 LUT 自动加载，背景层 LUT 自动加载和 IPA 传输同一时间只能有一个在工作。例如，当 IPA 传输正在进行的时候，若软件置位 FLLEN 或 BLLEN 位，前景层或背景层 LUT 自动加载不会启动，且 FLLEN 或 BLLEN 位自动复位。

35.5.11. 配置

开始任何传输之前，软件需要读 TEN，FLLEN 和 BLLEN 位检查是否有 IPA 传输或 LUT 加载正在工作。如果有任何一个正在进行，可以设置 TST 位使其停止或等待其完成。当读取 TEN，

FLLEN 和 BLLEN 的值都为 0 时，可以开始一个新的传输。

前景层 LUT 加载

当开始一个新的前景层 LUT 加载的时候，建议按如下步骤进行：

1. 配置 IPA_FLMADDR 寄存器设置前景层 LUT 存储区基地址；
2. 配置 IPA_FPCTL 寄存器的 FLPF 位设置前景层 LUT 像素格式；
3. 配置 IPA_FPCTL 寄存器的 FCNP 位设置前景层 LUT 要加载的像素数目；
4. 配置 IPA_CTL 寄存器的错误配置中断，LUT 加载完成中断，LUT 访问冲突中断和传输访问错误中断使能位；
5. 配置 IPA_FPCTL 寄存器的 FLLEN 位为‘1’以使能前景层 LUT 的自动加载。

背景层 LUT 加载

当开始一个新的背景层 LUT 加载的时候，建议按如下步骤进行：

1. 配置 IPA_BLMADDR 寄存器设置背景层 LUT 存储区基地址；
2. 配置 IPA_BPCTL 寄存器的 BLPF 位设置背景层 LUT 像素格式；
3. 配置 IPA_BPCTL 寄存器的 BCNP 位设置背景层 LUT 要加载的像素数目；
4. 配置 IPA_CTL 寄存器的错误配置中断，LUT 加载完成中断，LUT 访问冲突中断和传输访问错误中断使能位；

配置 IPA_FPCTL 寄存器的 FLLEN 位为‘1’以使能背景层 LUT 的自动加载。

IPA 传输

当开始一个新的 IPA 传输的时候，对应不同像素转换模式的配置步骤如下所示：

复制前景层图像到目标图像

1. 配置 IPA_FMADDR 和 IPA_DMADDR 寄存器设置前景层和目标层存储区基地址；
2. 配置 IPA_FPCTL 寄存器的 FPF 位域设置前景层像素格式；
3. 配置 IPA_FLOFF 和 IPA_DLOFF 寄存器的 FLOFF 和 DLOFF 位域设置前景层和目标层的行偏移；
4. 若需要，配置 IPA_LM 寄存器的 LM 位域设置行标记；
5. 配置 IPA_IMS 寄存器的 WIDTH 和 HEIGHT 位域设置图像大小；
6. 配置 IPA_CTL 寄存器的错误配置中断，LUT 访问冲突中断、传输行标记中断、完全传输完成中断和传输访问错误中断所需的使能位；
7. 配置 IPA_CTL 寄存器的 TEN 位为‘1’以使能 IPA 传输。

转换前景层图像到目标图像

如果前景层像素格式是非直接的，像素数据应在开始 IPA 传输之前加载到前景层 LUT。LUT 的自动加载过程可参考*前景层 LUT 加载*。

1. 配置 IPA_FMADDR 和 IPA_DMADDR 寄存器设置前景层和目标存储区基地址；
2. 配置 IPA_FPCTL 寄存器的 FAVCA 和 FPF 位域设置前景层 alpha 值的计算方法和前景层像素格式。当前景层格式是 YUV/YCbCr 时需要配置 IPA_CSCC_CFGx（x = 0…2）寄存器；

3. 如果前景层格式不是 ARGBxxxx 类型，配置 IPA_FPCTL 和 IPA_FPV 寄存器设置预定义像素值，包括 alpha，红，绿，蓝颜色值；
4. 配置 IPA_DPCTL 寄存器的 DPF 位域设置目标像素格式；
5. 配置 IPA_FLOFF 和 IPA_DLOFF 寄存器的 FLOFF 和 DLOFF 位设置前景层和目标层行偏移；
6. 若需要使用缩放功能，配置 IPA_DPCTL 寄存器的 VERDEC/HORDEC 位域，IPA_BSCTL 寄存器的 XSCAL/YSCAL 位域，IPA_DIMS 寄存器的 DWIDTH/DHEIGHT 位域；
7. 若需要旋转功能，配置 IPA_DPCTL 寄存器的 ROT 位域；
8. 若需要，配置 IPA_LM 寄存器的 LM 位设置行标记；
9. 配置 IPA_IMS 寄存器的 WIDTH 和 HEIGHT 位域设置图像大小；
10. 配置 IPA_CTL 寄存器的错误配置中断，LUT 访问冲突中断、传输行标记中断、完全传输完成中断和传输访问错误中断所需的使能位；
11. 配置 IPA_CTL 寄存器的 TEN 位为‘1’以使能 IPA 传输。

转换并混合前景层和背景层图像到目标图像

如果前景层和背景层像素格式是非直接的，开始 IPA 传输之前，像素数据必须被加载到对应的 LUT。前景层和背景层 LUT 的自动加载过程可参考*前景层 LUT 加载*和*背景层 LUT 加载*。

1. 配置 IPA_FMADDR, IPA_BMADDR 和 IPA_DMADDR 寄存器设置前景层，背景层和目标存储区基地址；
2. 配置 IPA_FPCTL 寄存器的 FAVCA 和 FPF 位域设置前景层 alpha 值的计算方法和前景层像素格式。当前景层格式是 YUV/YCbCr 时需要配置 IPA_CSCC_CFGx（x = 0…2）寄存器；
3. 如果前景层格式不是 ARGBxxxx 类型，配置 IPA_FPCTL 和 IPA_FPV 寄存器设置预定义像素值，包括 alpha，红，绿，蓝颜色值；
4. 配置 IPA_BPCTL 寄存器的 BAVCA 和 BPF 位域设置背景层 alpha 值的计算方法和背景层像素格式。当前景层格式是 YUV/YCbCr 时需要配置 IPA_CSCC_CFGx（x = 0…2）寄存器；
5. 如果背景层格式不是 ARGBxxxx 类型，配置 IPA_BPCTL 和 IPA_BPV 寄存器设置预定义像素值，包括 alpha，红，绿，蓝颜色值；
6. 配置 IPA_DPCTL 寄存器的 DPF 位域设置目标像素格式；
7. 配置 IPA_FLOFF, IPA_BLOFF 和 IPA_DLOFF 寄存器的 FLOFF、BLOFF 和 DLOFF，设置背景层，前景层和目标的行偏移；
8. 若需要，配置 IPA_LM 寄存器的 LM 位设置行标记；
9. 配置 IPA_IMS 寄存器的 WIDTH 和 HEIGHT 位域设置图像大小；
10. 若需要使用缩放功能，配置 IPA_DPCTL 寄存器的 VERDEC / HORDEC 位域，IPA_BSCTL 寄存器的 XSCAL / YSCAL 位域，IPA_DIMS 寄存器的 DWIDTH / DHEIGHT 位域；
11. 若需要旋转功能，配置 IPA_DPCTL 寄存器的 ROT 位域；
12. 配置 IPA_CTL 寄存器的错误配置中断，LUT 访问冲突中断、传输行标记中断、完全传输完成中断和传输访问错误中断所需的使能位；
13. 配置 IPA_CTL 寄存器的 TEN 位为‘1’以使能 IPA 传输。

用特定的颜色填充目标图像

1. 配置 IPA_DMADDR 寄存器设置目标存储区的基地址；
2. 配置 IPA_DPCTL 寄存器的 DPF 位域设置目标像素格式；

3. 配置 IPA_DPV 寄存器设置目标区预定义像素值，包括 alpha，红，绿，蓝颜色值；
4. 配置 IPA_DLOFF 寄存器的 DLOFF 位域设置目标的行偏移；
5. 若需要，配置 IPA_LM 寄存器的 LM 位域设置行标记；
6. 配置 IPA_IMS 寄存器的 WIDTH 和 HEIGHT 位域设置图像大小；
7. 配置 IPA_CTL 寄存器的错误配置中断，LUT 访问冲突中断、传输行标记中断、完全传输完成中断和传输访问错误中断所需的使能位；
8. 配置 IPA_CTL 寄存器的 TEN 位为‘1’以使能 IPA 传输。

配置规则

IPA 配置必须遵守一些规则，否则在它使能之后，传输或加载将会自动复位，IPA_INTF 寄存器的 WCFIF 位将会立即置位。规则描述如下：

当前景层 LUT 自动加载使能：

- 当 IPA_FPCTL 寄存器的 FLPF 位等于 0 时，寄存器 IPA_FLMADDR 的 FLMADDR 位域必须是 32 位对齐的。

当背景层 LUT 自动加载使能：

- 当 IPA_BPCTL 寄存器的 BLPF 位等于 0 时，寄存器 IPA_BLMADDR 的 BLMADDR 位必须是 32 位对齐的。

当 IPA 传输使能：

1) 当 IPA_FPCTL 寄存器的 FPF 位是‘ARGB8888’，’UYVY422’，‘VYUY422’，’YUV420’或‘YVU420’时，IPA_FMADDR 寄存器的 FMADDR 位必须是 32 位对齐。当 FPF 位是‘RGB565’, ‘ARGB1555’, ‘ARGB4444’ 或 ‘AL88’，必须是 16 位对齐。
2) 当 IPA_FPCTL 寄存器的 FPF 位是’A4’，’L4’，‘UYVY422’ 或‘VYUY422’时，IPA_FLOFF 寄存器的 FLOFF 位域必须是偶数。当 IPA_FPCTL 寄存器的 FPF 位为‘YUV420’或‘YVU420’时，必须是 4 位对齐。
3) 当 IPA_BPCTL 寄存器的 BPF 位是‘ARGB8888’时，IPA_BMADDR 寄存器的 BMADDR 位必须是 32 位对齐。当 BPF 位是‘RGB565’, ‘ARGB1555’, ‘ARGB4444’ 或 ‘AL88’，必须是 16 位对齐。
4) 当 IPA_BPCTL 寄存器的 BPF 位是 A4 或 L4 时，IPA_BLOFF 寄存器的 BLOFF 位域必须是偶数。
5) 当 PFCM 位不为‘0b00’或‘0b11’时，IPA_FPCTL 寄存器的 FPF 位必须小于或等于‘0b1111’。当 PFCM 位为‘0b00’时，FPF 位必须小于或等于‘0b1101’
6) IPA_BPCTL 寄存器的 BPF 位必须小于或等于‘0b1010’。
7) IPA_DPCTL 寄存器的 DPF 位必须小于或等于‘0b100’。
8) 当 IPA_DPCTL 寄存器的 DPF 位是‘ARGB8888’时，IPA_DMADDR 寄存器的 DMADDR 位必须是 32 位对齐。当 DPF 位是‘RGB565’, ‘ARGB1555’, ‘ARGB4444’ 或 ‘AL88’，必须是 16 位对齐。
9) 当 IPA_FPCTL 寄存器的 FPF 位是’A4’或’L4’时，IPA_DLOFF 寄存器的 DLOFF 位必须是偶数。
10) 当 IPA_FPCTL 寄存器的 FPF 位是’A4’，’L4’，‘UYVY422’或‘VYUY422’时，IPA_IMS 寄存器的 WIDTH 位必须是偶数。当 IPA_FPCTL 寄存器的 FPF 位是’YUV420’或‘YVU420’，必须是 4 位对齐。
11) 当 IPA_BPCTL 寄存器的 BPF 位是’A4’或’L4’时，IPA_IMS 寄存器的 WIDTH 位必须是偶

数。

12) IPA_IMS 寄存器的 WIDTH 位必须大于 0。
13) IPA_IMS 寄存器的 HEIGHT 位必须大于 0。
14) IPA_BSCTL 寄存器中的 XSCALE 和 YSCALE 位必须大于零且小于 0x2001。
15) 当 FIIMEN 位置位时，IPA_FPCTL 寄存器中的 FPF 位不能为'YUV420'和'YVU420'。当 IPA_FPCTL 寄存器中的 FPF 位为'ARGB8888'、'UYVY422'、'VYUY422'时，IPA_EF_UV_MADDR 寄存器的 EFUVMADDR 位的值必须为 32 位对齐。当 FPF 位为'RGB565'、'ARGB1555'、'ARGB4444'或'AL88'时 EFUVMADDR 位的值必须为 16 位对齐。
16) 当 IPA_FPCTL 寄存器中的 FPF 位为'YUV420'和'YVU420'时，IPA_EF_UV_MADDR 寄存器的 EFUVMADDR 位的值必须为 16 位对齐。
17) 启用前景层通道缩放功能时，HORDEC、VERDEC、XSCALE、YSCALE 不为默认值。当 IPA_BPCTL 寄存器中的 BPF 位为'A4'、'L4'时，IPA_DIMS 寄存器中的 DWIDTH 位必须是偶数。
18) 启用前景层通道缩放和旋转功能时，HORDEC、VERDEC、XSCALE、YSCALE 和 ROT 不为默认值。当 IPA_BPCTL 寄存器中的 BPF 位为'A4'、'L4'时，IPA_DIMS 寄存器中的 DHEIGHT 位必须是偶数。

当 PFCM 位等于‘00’，仅考虑 1)、2)、5)、9)、10)、12)、13)。
当 PFCM 位等于‘01’，仅考虑 1)、2)、5)、7)、8)、10)、12)、13)、14)、15)、16)。
当 PFCM 位等于‘10’，除了 9)所有的规则都考虑。
当 PFCM 位等于‘11’，仅考虑 7)、8)、12)、13)。

35.6. 中断

有 6 个中断事件连接到 IPA 中断，包括错误配置中断，LUT 加载完成中断，LUT 访问冲突中断，传输行标记中断，传输完成中断和传输访问错误中断。任何一个中断事件发生都将产生一个 IPA 中断。

每一个中断事件在 IPA_INTF 寄存器有一个专用的状态位，在 IPA_INTC 寄存器有一个专门的清除位，在 IPA_CTL 寄存器有一个专用的使能位。相应位之间的关系如*表 35-8. IPA 中断事件*所述。

表 35-8. IPA 中断事件

中断事件	状态位	使能位	清除位
	IPA_INTF	IPA_CTL	IPA_INTC
配置错误中断	WCFIF	WCFIE	WCFIFC
LUT加载完成中断	LLFIF	LLFIE	LLFIFC
LUT访问冲突中断	LACIF	LACIE	LACIFC
传输行标记中断	TLMIF	TLMIE	TLMIFC
全传输完成中断	FTFIF	FTFIE	FTFIFC
传输访问错误中断	TAEIF	TAEIE	TAEIFC

配置错误中断

当 LUT 加载或 IPA 传输被使能之后，若*配置规则*小节列出的任何一个配置规则被破坏，配置错误中断状态位将立即置位。LUT 加载或 IPA 传输将自动停止且不产生任何访问操作。

当配置错误中断状态位被置位并且相应使能位使能时，将产生一个 IPA 中断。

LUT 加载完成中断

当最后一个像素数据被写入到前景层或背景层 LUT 后，LUT 加载完成中断状态状态位将立即置位，在加载期间的停止操作不能置位 LUT 加载完成中断状态位。

当 LUT 加载完成中断状态位被置位并且相应使能位使能，将产生一个 IPA 中断。

LUT 访问冲突中断

当通过软件访问前景层和背景层 LUT 时，必须遵守以下规则：

- 当前景层 LUT 自动加载时，禁止软件访问前景层 LUT；
- 当背景层 LUT 自动加载时，禁止软件访问背景层 LUT；
- 当配置 IPA 传输模式 PFCM 位等于‘01’或‘10’时，如果前景层像素格式是非直接的，在 IPA 传输正在进行时，禁止软件访问前景层 LUT；
- 当配置 IPA 传输模式 PFCM 位等于‘10’时，如果背景层像素格式是非直接的，在 IPA 传输正在进行时，禁止软件访问背景层 LUT。

当违背上述规则之一时，LUT 访问冲突中断状态位置位，且软件访问无作用（写访问不被执行，读访问返回一个无效值）。

当 LUT 访问冲突中断状态位寄存器被置位并且相应使能位使能时，将产生一个 IPA 中断。

传输行标记中断

当标记行的最后一个像素数据被写入目标存储区后，传输行标记中断状态位立即置位。如果 IPA_LM 的 LM 位等于 0，在 IPA 传输期间，传输行标记中断状态位不置位。

当传输行标记中断状态位被置位并且相应使能位使能时，将产生一个 IPA 中断。

传输完成中断

当最后一个像素数据被写入目标存储区后，传输完成中断状态位立即置位。在 IPA 传输期间的停止操作不置位传输完成中断状态位。

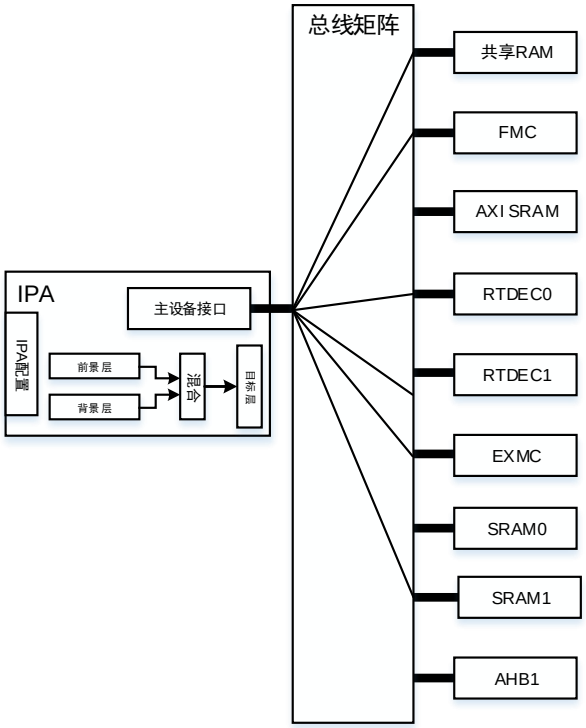
当传输完成中断状态位被置位并且相应使能位使能时，将产生一个 IPA 中断。

传输访问错误中断

当 IPA 访问的地址超出允许的地址，IPA 将收到一个错误反馈；传输（LUT 加载或 IPA 传输）将被立即禁止并且没有 LUT 加载完成中断状态位或传输完成中断状态位置位。IPA 允许和禁止的访问区域在*图 35-9. IPA 的系统连接*中列出。

当传输访问错误中断状态位被置位并且相应使能位使能时，将产生一个 IPA 中断。

图 35-9. IPA 的系统连接



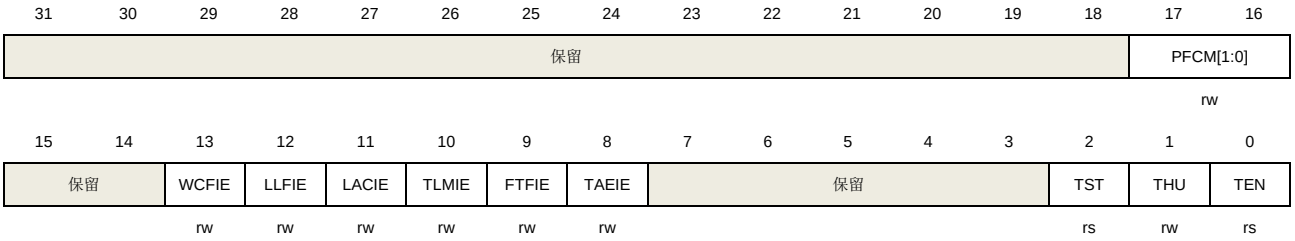
35.7. IPA 寄存器

IPA基地址：0x5200 1000

35.7.1. 控制寄存器（IPA_CTL）

地址偏移：0x00
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域	名称	描述
31:18	保留	必须保持复位值。
17:16	PFCM[1:0]	像素格式转换模式 软件置位和清除. 00：前景层存储区到目标存储区无像素格式转换 01：前景层存储区到目标存储区有像素格式转换 10：混合前景层和背景层存储区到目标存储区 11：用特定的颜色填充目标存储区 当TEN为‘1’时，该位不可写。
15:14	保留	必须保持复位值。
13	WCFIE	配置错误中断使能位 软件置位和清除 0：配置错误中断禁止 1：配置错误中断使能
12	LLFIE	LUT加载完成中断使能位 软件置位和清除 0：LUT加载完成中断禁止 1：LUT加载完成中断使能
11	LACIE	LUT访问冲突中断使能位 软件置位和清除 0：LUT访问冲突中断禁止 1：LUT访问冲突中断使能
10	TLMIE	传输行标记中断使能位 软件置位和清除

0：传输行标记中断禁止
1：传输行标记中断使能

9	FTFIE	传输完成中断使能位 软件置位和清除 0：传输完成中断禁止 1：传输完成中断使能
8	TAEIE	传输访问错误中断使能位 软件置位和清除 0：传输访问错误中断禁止 1：传输访问错误中断使能
7:3	保留	必须保持复位值。
2	TST	传输停止 软件置位，软件和硬件清除 0：无影响 1：停止当前的传输 当该位使能后，当前传输停止。当当前传输停止后，该位立即被硬件自动清0。
1	THU	传输挂起 软件置位，软件和硬件清除. 0：无影响 1：挂起当前传输 当该位使能后，当前传输暂停。当该位清0后，当前传输继续。当当前传输被停止时，该位立即被硬件自动清0。
0	TEN	传输使能 软件置位，硬件清除 0：传输禁止 1：传输使能 当该位使能后，IPA传输开始。当下述情况之一发生时，该位自动清0。 - 使能TST位停止当前传输 - 传输完全完成 - 检测到配置错误或传输访问错误 - 前景层LUT或背景层LUT正在自动加载（IPA_FPCTL寄存器的FLLEN或IPA_BPCTL寄存器的BLLEN位为‘1’）

35.7.2. 中断状态寄存器（IPA_INTF）

地址偏移：0x04
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留										WCFIF	LLFIF	LACIF	TLMIF	FTFIF	TAEIF
										r	r	r	r	r	r

位/位域	名称	描述
31:6	保留	必须保持复位值。
5	WCFIF	配置错误中断标志 硬件置位，软件置位IPA_INTC 寄存器的‘WCFIFC’位清除该位。 0：当IPA传输完成或LUT加载使能时，没检测到配置错误 1：当IPA传输完成或LUT加载使能时，检测到配置错误
4	LLFIF	LUT加载完成中断标志LUT 硬件置位，软件置位IPA_INTC 寄存器的‘LLFIFC’位清除该位。 0：没检测到LUT加载完成 1：检测到一个LUT加载完成
3	LACIF	LUT访问冲突中断标志位 硬件置位，软件置位IPA_INTC 寄存器的‘LACIFC’位清除该位。 0：没检测到LUT访问冲突 1：检测到一个LUT访问冲突
2	TLMIF	传输行标记中断标志 硬件置位，软件置位IPA_INTC 寄存器的‘CTCLIF’位清除该位。 0：传输的像素数目，没有准确的达到标记行 1：传输的像素数目，准确的达到标记行
1	FTFIF	传输完成中断标志 硬件置位，软件置位IPA_INTC 寄存器的‘CTFIF’位清除该位。 0：没检测到传输完成 1：检测到传输完成
0	TAEIF	传输访问错误中断标志 硬件置位，软件置位IPA_INTC 寄存器的‘CTEIF’位清除该位。 0：没检测到传输访问错误 1：检测到传输访问错误

35.7.3. 中断标志清除寄存器（IPA_INTC）

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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留										WCFIFC	LLFIFC	LACIFC	TLMIFC	FTFIFC	TAEIFC
										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	WCFIFC	配置错误中断标志清除位 软件置位，硬件清除 0：无影响 1：清除配置错误中断标志
4	LLFIFC	LUT加载完成中断标志清除位 软件置位，硬件清除 0：无影响 1：清除LUT加载完成中断标志
3	LACIFC	LUT访问冲突中断标志清除位 软件置位，硬件清除 0：无影响 1：清除LUT访问冲突中断标志
2	TLMIFC	传输行标记中断标志清除位 软件置位，硬件清除 0：无影响 1：清除传输行标记中断标志
1	FTFIFC	传输完成中断标志清除位 软件置位，硬件清除 0：无影响 1：清除传输完成中断标志
0	TAEIFC	传输访问错误中断标志清除位 软件置位，硬件清除 0：无影响 1：清除传输访问错误中断标志

35.7.4. 前景层存储区基地址寄存器（IPA_FMADDR）

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
FMADDR[31:16]															
rw															

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

FMADDR[15:0]

rw

位/位域	名称	描述
31:0	FMADDR[31:0]	前景层存储区基地址 这些位必须是8位，16位，32位对齐，具体对齐方式与前景层像素格式相对应。如果前景层像素格式是ARGB8888，UYVY422，VYUY422，YUV420或YVU420，这些位必须是32位对齐，如果前景层像素格式是RGB565，ARGB1555，ARGB4444或AL88，这些位必须是16位对齐，如果违背以上对齐规则，当传输使能时，将产生配置错误。 当IPA_CTL寄存器的TEN位为‘1’时，该位不可写。

35.7.5. 前景层行偏移寄存器（IPA_FLOFF）

地址偏移：0x10

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

rw

位/位域	名称	描述
31:14	保留	必须保持复位值。。
13:0	FLOFF[13:0]	前景层行偏移 该位表明当前行最后一个像素和下一行第一个像素之间的像素数目。如果前景层像素格式是A4或L4，FLOFF必须被配置成一个偶数，如果前景层像素格式为YUV420或YVU420，则FLOFF必须配置为4位对齐，否则当传输使能的时候将检测到一个配置错误。 当IPA_CTL寄存器的TEN位为‘1’时，该位不可写。

35.7.6. 背景层存储区基地址寄存器（IPA_BMADDR）

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
BMADDR[31:16]															

rw

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

rw

位/位域	名称	描述
31:0	BMADDR[31:0]	背景层存储区基地址 这些位必须是8位，16位，32位对齐，具体对齐方式与背景层像素格式相对应。如果背景层像素格式是ARGB8888，UYVY422和VYUY422，这些位必须是32位对齐，如果背景层像素格式是RGB565, ARGB1555, ARGB4444或AL88,这些位必须是16位对齐，如果违背以上对齐规则，当传输使能时，将产生配置错误。 当IPA_CTL寄存器的TEN位为‘1’时，该位不可写。

35.7.7. 背景层行偏移寄存器（IPA_BLOFF）

地址偏移：0x18
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:14	保留	必须保持复位值。
13:0	BLOFF[13:0]	背景层行偏移 该位表明当前行最后一个像素和下一行第一个像素之间的像素数目。如果背景层像素格式是A4或L4，BLOFF必须被配置成一个偶数，否则当传输使能的时候将检测到一个配置错误。 当IPA_CTL寄存器的TEN位为‘1’时，该位不可写。

35.7.8. 前景层像素控制寄存器（IPA_FPCTL）

地址偏移：0x1C
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FPDAV[7:0]								FIIMEN	保留					FAVCA[1:0]	
rw								rw						rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCNP[7:0]								保留		FLLEN	FLPF	FPF[3:0]			
rw										rc_w1	rw	rw			

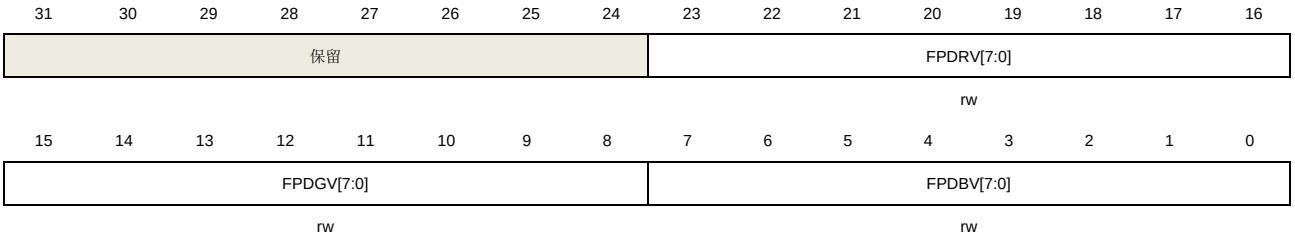
位/位域	名称	描述
31:24	FPDAV[7:0]	前景层预定义alpha通道值 软件置位和清除 该位域预定义前景层的alpha通道值。该位域结合从前景层存储区或前景层LUT读取的alpha数据根据前景层alpha计算算法计算前景层的alpha通道值。 当IPA_CTL寄存器的TEN位为‘1’时，该位不可写。
23	FIIMEN	前景层隔行输入模式 0：禁能前景层隔行输入模式 1：使能前景层隔行输入模式
22:18	保留	必须保持复位值。
17:16	FAVCA[1:0]	前景层alpha值计算算法 软件置位和清除 00：无影响 01：FPDAV[7:0]被选作前景层alpha通道值 10：FPDAV[7:0]乘以从前景层存储区或前景层LUT读取的alpha数据除以255作为前景层alpha通道值。 11：保留 当IPA_CTL寄存器的TEN位为‘1’时，该位不可写。
15:8	FCNP[7:0]	前景层LUT像素数目 软件置位和清除 前景层LUT的像素数目等于FCNP + 1. 当FLLEN为‘1’时，该位不可写。
7:6	保留	必须保持复位值。
5	FLLEN	前景层LUT加载使能 软件置位，硬件清除 0：禁止前景层LUT加载 1：使能前景层LUT加载 当该位使能，前景层LUT自动加载开始，当下述情况之一发生时，该位自动清0 - 使能TST位停止当前传输 - 前景层LUT自动加载完成 - 检测到配置错误或传输错误 - IPA传输或背景层LUT自动加载正在进行
4	FLPF	前景层LUT像素格式 软件置位和清除 0：ARGB8888 1：RGB888 当FLLEN为‘1’时，该位不可写。
3:0	FPF[3:0]	前景层像素格式 软件置位和清除 0000：ARGB8888

0001：RGB888
0010：RGB565
0011：ARGB1555
0100：ARGB4444
0101：L8
0110：AL44
0111：AL88
1000：L4
1001：A8
1010：A4
1011：YUV444 (1 Plane)
1100：UYVY422 (1 Plane)
1101：VYUY422 (1 Plane)
1110：YUV420 (2 Plane)
1111：YVU420 (2 Plane)
当IPA_CTL寄存器的TEN位为‘1’时，该位不可写。

35.7.9. 前景层像素值寄存器（IPA_FPV）

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。



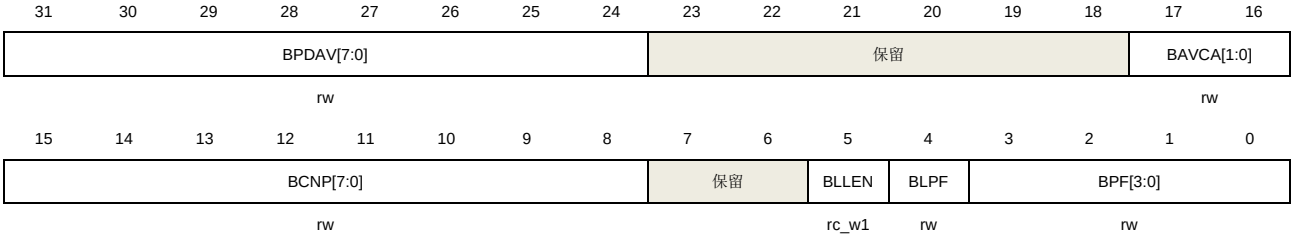
位/位域	名称	描述
31:24	保留	必须保持复位值。
23:16	FPDRV[7:0]	前景层预定义红色值 当前景层像素格式是A4或A8时，该位域被用作前景层红色值。 当IPA_CTL寄存器的TEN位为’1’时，该位不可写。
15:8	FPDGV[7:0]	前景层预定义绿色值 当前景层像素格式是A4或A8时，该位域被用作前景层绿色值。 当IPA_CTL寄存器的TEN位为’1’时，该位不可写。
7:0	FPDBV[7:0]	前景层预定义蓝色值 当前景层像素格式是A4或A8时，该位域被用作前景层蓝色值。 当IPA_CTL寄存器的TEN位为’1’时，该位不可写。

35.7.10. 背景层像素控制寄存器（IPA_BPCTL）

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。



位/位域	名称	描述
31:24	BPDAV[7:0]	背景层预定义alpha通道值 软件置位和清除 该位域预定义背景层的alpha通道值。该位域结合从背景层存储区和背景层LUT读取的alpha数据根据背景层alpha计算算法计算背景层alpha通道值。 当IPA_CTL寄存器的TEN位为‘1’时，该位不可写。
23:18	保留	必须保持复位值。
17:16	BAVCA[1:0]	背景层alpha值计算算法 软件置位和清除 00：无影响 01：BPDAV [7:0] 被选作背景层alpha值 10：BPDAV [7:0] 乘以从背景层存储区或背景层LUT读取的alpha数据除以255作为背景层alpha值。 11：保留 当IPA_CTL寄存器的TEN位为‘1’时，该位不可写。
15:8	BCNP[7:0]	背景层LUT像素数目 软件置位和清除 背景层LUT的像素数目等于BCNP + 1. 当BLLEN为‘1’时，该位不可写。
7:6	保留	必须保持复位值。
5	BLLEN	背景层LUT加载使能 软件置位，硬件清除. 0：禁止背景层LUT加载 1：使能背景层LUT加载 当该位使能，背景层LUT加载开始，当下述情况之一发生时，该位自动清0： - 使能TST停止当前传输 - 背景层LUT加载完成 - 检测到配置错误或传输错误 - IPA传输或背景层LUT自动加载正在进行（IPA_CTL寄存器TEN位或IPA_FPCTL

寄存器FLLEN位为1）

4	BLPF	背景层LUT像素格式 软件置位和清除 0：ARGB8888 1：RGB888 当BLLEN为‘1’时，该位不可写。
3:0	BPF[3:0]	背景层像素格式 软件置位和清除 0000：ARGB8888 0001：RGB888 0010：RGB565 0011：ARGB1555 0100：ARGB4444 0101：L8 0110：AL44 0111：AL88 1000：L4 1001：A8 1010：A4 1011 ~ 1111：保留 当IPA_CTL寄存器的TEN位为‘1’时，该位不可写。

35.7.11. 背景层像素值寄存器（IPA_BPV）

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								BPDRV[7:0]							
								rw							

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

位/位域	名称	描述
31:24	保留	必须保持复位值。
23:16	BPDRV[7:0]	背景层预定义红色值 当背景层像素格式是A4或A8时，该位域被用作背景层红色值。 当IPA_CTL寄存器的TEN为‘1’时，该位不可写。
15:8	BPDGV[7:0]	背景层预定义绿色值 当背景层像素格式是A4或A8时，该位域被用作背景层绿色值。

当IPA_CTL寄存器的TEN位为’1’时，该位不可写。

7:0	BPDBV[7:0]	背景层预定义蓝色值 当背景层像素格式是A4或A8时，该位域被用作背景层蓝色值。 当IPA_CTL寄存器的TEN位为’1’时，该位不可写。

35.7.12. 前景层 LUT 存储区基地址寄存器（IPA_FLMADDR）

地址偏移：0x2C

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:0	FLMADDR[31:0]	前景层LUT存储区基地址 软件置位和清除 这些位必须是8位，16位，32位对齐，具体对齐方式与前景层LUT像素格式相对应。 如果前景层LUT像素格式是ARGB8888，这些位必须是32位对齐。如果违背以上对齐规则，当前景层LUT加载使能时，将产生配置错误。 当IPA_FPCTL寄存器的FLLEN 位为‘1’的时候，该位域不可写。

35.7.13. 背景层 LUT 存储区基地址寄存器（IPA_BLMADDR）

地址偏移：0x30

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:0	BLMADDR[31:0]	背景层LUT存储区基地址 软件置位和清除 这些位必须是8位，16位，32位对齐，具体对齐方式与背景层LUT像素格式相对应。 如果背景层LUT像素格式是ARGB8888，这些位必须是32位对齐。如果违背以上对齐

规则，当背景层LUT加载使能时，将产生配置错误。

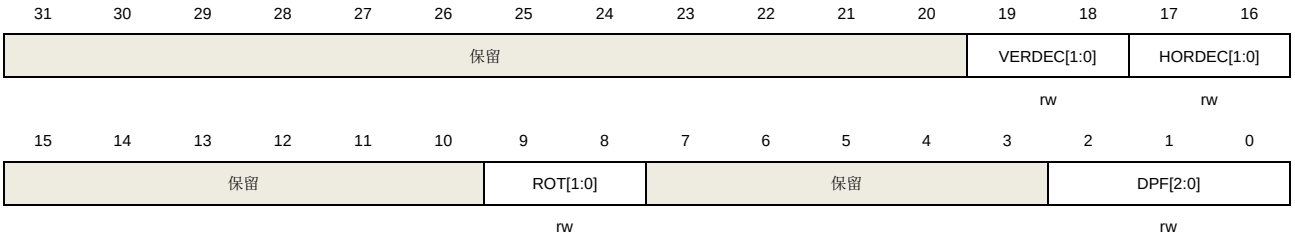
当IPA_BPCTL寄存器的BLLEN位为‘1’的时候，该位域不可写。

35.7.14. 目标像素控制寄存器（IPA_DPCTL）

地址偏移：0x34

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域	名称	描述
31:20	保留	必须保持复位值。
19:18	VERDEC[1:0]	垂直预抽取滤波器控制 00：禁用预抽取滤波器 01：抽取滤波器的缩减因子为2 10：抽取滤波器的缩减因子为4 11：抽取滤波器的缩减因子为8
17:16	HORDEC[1:0]	水平预抽取滤波器控制 00：禁用预抽取滤波器 01：抽取滤波器的缩减因子为2 10：抽取滤波器的缩减因子为4 11：抽取滤波器的缩减因子为8
15:10	保留	必须保持复位值。
9:8	ROT[1:0]	指示要应用于输出的顺时针旋转 00：不旋转 01：旋转90度 10：旋转180度 11：旋转270度
7:3	保留	必须保持复位值。
2:0	DPF[2:0]	目标像素格式 软件置位和清除 000：ARGB8888 001：RGB888 010：RGB565 011：ARGB1555

100：ARGB4444

101~111：保留

当IPA_CTL寄存器的TEN位为’1’时，该位不可写。

35.7.15. 目标像素值寄存器（IPA_DPV）

地址偏移：0x38

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
DPDAV[7:0]								DPDRV[7:0]							
MEANINGLESS								DPDRV[7:0]							
MEANINGLESS															
MEANINGLESS															
MEANINGLESS															

rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPDGV[7:0]								DPDBV[7:0]							
DPDGV[7:0]								DPDBV[7:0]							
DPDRV[4:0]					DPDGV[5:0]						DPDBV[4:0]				
DPDAV	DPDRV[4:0]					DPDGV[4:0]					DPDBV[4:0]				
DPDAV[3:0]				DPDRV[3:0]				DPDGV[3:0]				DPDBV[3:0]			

rw

当目标像素格式是 ARGB8888 时，第 1 行有效。

位/位域	名称	描述
31:24	DPDAV[7:0]	目标层预定义alpha值 软件置位和清除 当IPA配置为用特定的颜色填充目标存储区的时候，这些位用作目标层alpha值。 当IPA_CTL寄存器的TEN位的值为’1’时，该位不可写。
23:16	DPDRV[7:0]	目标层预定义红色值 软件置位和清除 当IPA配置为用特定的颜色填充目标存储区的时候，这些位用作目标层红色值。 当IPA_CTL寄存器的TEN位的值为’1’时，该位不可写。
15:8	DPDGV[7:0]	目标层预定义绿色值 软件置位和清除 当IPA配置为用特定的颜色填充目标存储区的时候，这些位用作目标层绿色值。 当IPA_CTL寄存器的TEN位的值为’1’时，该位不可写。
7:0	DPDBV[7:0]	目标层预定义蓝色值 软件置位和清除 当IPA配置为用特定的颜色填充目标存储区的时候，这些位用作目标层蓝色值。

当IPA_CTL寄存器的TEN 的值为‘1’时，该位不可写。

当目标像素格式是 RGB888 时，第 2 行有效。

位/位域	名称	描述
31:24	Meaningless	该位域可以软件置位和清除，但当目标像素格式是RGB888时，这些位没有意义。
23:16	DPDRV[7:0]	目标层预定义红色值 软件置位和清除 当IPA配置为用特定的颜色填充目标存储区的时候，这些位用作目标层红色值。 当IPA_CTL寄存器的TEN位的值为’1’时，该位不可写。
15:8	DPDGV[7:0]	目标层预定义绿色值 软件置位和清除 当IPA配置为用特定的颜色填充目标存储区的时候，这些位用作目标层绿色值。 当IPA_CTL寄存器的TEN位的值为’1’时，该位不可写。
7:0	DPDBV[7:0]	目标层预定义蓝色值 软件置位和清除 当IPA配置为用特定的颜色填充目标存储区的时候，这些位用作目标层蓝色值。 当IPA_CTL寄存器的TEN位的值为’1’时，该位不可写。

当目标像素格式是 RGB565 时，第 3 行有效。

位/位域	名称	描述
31:16	Meaningless	该位域可以软件置位和清除，但当目标像素格式是RGB565时，这些位没有意义。
15:11	DPDRV[4:0]	目标层预定义红色值 软件置位和清除 当IPA配置为用特定的颜色填充目标存储区的时候，这些位用作目标层红色值。 当IPA_CTL寄存器的TEN位的值为’1’时，该位不可写。
10:5	DPDGV[5:0]	目标层预定义绿色值 软件置位和清除 当IPA配置为用特定的颜色填充目标存储区的时候，这些位用作目标层绿色值。 当IPA_CTL寄存器的TEN位的值为’1’时，该位不可写。
4:0	DPDBV[4:0]	目标层预定义蓝色值 软件置位和清除 当IPA配置为用特定的颜色填充目标存储区的时候，这些位用作目标层蓝色值。 当IPA_CTL寄存器的TEN位的值为’1’时，该位不可写。

当目标像素格式是 ARGB1555 时，第 4 行有效。

位/位域	名称	描述
31:16	Meaningless	该位域可以软件置位和清除，但当目标像素格式是ARGB1555时，这些位没有意义。
15	DPDAV	目标层预定义alpha值 软件置位和清除

		当IPA配置为用特定的颜色填充目标存储区的时候，这些位用作目标层alpha值。 当IPA_CTL寄存器的TEN位的值为'1'时，该位不可写。
14:10	DPDRV[4:0]	目标层预定义红色值 软件置位和清除 当IPA配置为用特定的颜色填充目标存储区的时候，这些位用作目标层红色值。 当IPA_CTL寄存器的TEN位的值为'1'时，该位不可写。
9:5	DPDGV[4:0]	目标层预定义绿色值 软件置位和清除 当IPA配置为用特定的颜色填充目标存储区的时候，这些位用作目标层绿色值。 当IPA_CTL寄存器的TEN位的值为'1'时，该位不可写。
4:0	DPDBV[4:0]	目标层预定义蓝色值 软件置位和清除 当IPA配置为用特定的颜色填充目标存储区的时候，这些位用作目标层蓝色值。 当IPA_CTL寄存器的TEN位的值为'1'时，该位不可写。

当目标像素格式是 ARGB4444 时，第 5 行有效。

位/位域	名称	描述
31:16	Meaningless	该位域可以软件置位和清除，但当目标像素格式是ARGB4444时，这些位没有意义。
15:12	DPDAV[3:0]	目标层预定义alpha值 软件置位和清除 当IPA配置为用特定的颜色填充目标存储区的时候，这些位用作目标层alpha值。 当IPA_CTL寄存器的TEN位的值为'1'时，该位不可写。
11:8	DPDRV[3:0]	目标层预定义红色值 软件置位和清除 当IPA配置为用特定的颜色填充目标存储区的时候，这些位用作目标层红色值。 当IPA_CTL寄存器的TEN位的值为'1'时，该位不可写。
7:4	DPDGV[3:0]	目标层预定义绿色值 软件置位和清除 当IPA配置为用特定的颜色填充目标存储区的时候，这些位用作目标层绿色值。 当IPA_CTL寄存器的TEN位的值为'1'时，该位不可写。
3:0	DPDBV[3:0]	目标层预定义蓝色值 软件置位和清除 当IPA配置为用特定的颜色填充目标存储区的时候，这些位用作目标层蓝色值。 当IPA_CTL寄存器的TEN位的值为'1'时，该位不可写。

35.7.16. 目标存储区基地址寄存器（IPA_DMADDR）

地址偏移：0x3C
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
DMADDR[31:16]															
rw															

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

位/位域	名称	描述
31:0	DMADDR[31:0]	目标存储区基地址 软件置位和清除 如果目标层像素格式是ARGB8888，这些位必须是32位对齐，如果目标层像素格式是RGB565，ARGB1555或ARGB4444,这些位必须是16位对齐，如果违背以上对齐规则，当传输使能的时候，将检测到一个配置错误。 当IPA_CTL寄存器的TEN位为‘1’时，该位不可写。

35.7.17. 目标行偏移寄存器（IPA_DLOFF）

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															

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

位/位域	名称	描述
31:14	保留	必须保持复位值。
13:0	DLOFF[13:0]	目标行偏移 该位表明当前行最后一个像素和下一行第一个像素之间的像素数目。当IPA_CTL寄存器的PFCM配置为“00”时，如果前景层像素格式是A4或L4，DLOFF必须被配置成一个偶数，否则当传输使能的时候将检测到一个配置错误。 当IPA_CTL寄存器的TEN位为’1’时，该位不可写。

35.7.18. 图像大小寄存器（IPA_IMS）

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
保留		WIDTH[13:0]													

rw

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

rw

位/位域	名称	描述
31:30	保留	必须保持复位值。
29:16	WIDTH[13:0]	待处理的图像的宽度 软件置位和清除 该位域表示待处理的图像每行像素的数目。如果背景层或前景层像素格式是A4或L4，这些位必须配置成偶数，否则当传输使能的时候将检测到一个配置错误。 当IPA_CTL寄存器的TEN位为'1'时，该位不可写。
15:0	HEIGHT[15:0]	待处理图像的高度 软件置位和清除 该位域表明待处理图像的行数。 当IPA_CTL寄存器的TEN为'1'时，该位不可写。

35.7.19. 行标记寄存器（IPA_LM）

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。

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

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

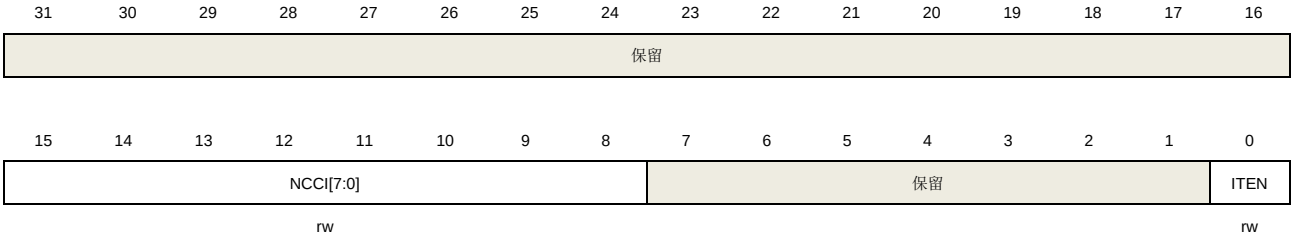
rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	LM[15:0]	行标记 软件置位和清除 该位域定义了一个行号以表明传输的进度，当且仅当标记行的最后一个像素已经写入了目标存储区，传输行标记中断标志位将置位。 当IPA_CTL寄存器的TEN位为'1'时，该位不可写。

35.7.20. 内部定时器控制寄存器（IPA_ITCTL）

地址偏移：0x4C
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



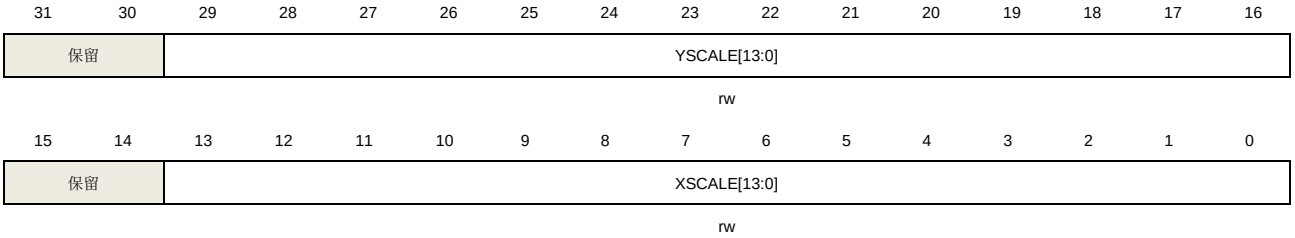
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	NCCI[7:0]	间隔时钟周期数 软件置位和清除 如果ITEN等于‘0’，该位域没有意义。如果ITEN等于‘1’，该位域表示两个连续的AHB请求之间插入的时钟周期数的最小值。
7:1	保留	必须保持复位值。
0	ITEN	内部定时器使能 IPA使用一个内部定时器用来减少AHB总线使用带宽。 0：禁止内部定时器 1：使能内部定时器

35.7.21. 双线性缩放控制寄存器（IPA_BSCTL）

地址偏移：0x50

复位值：0x1000 1000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



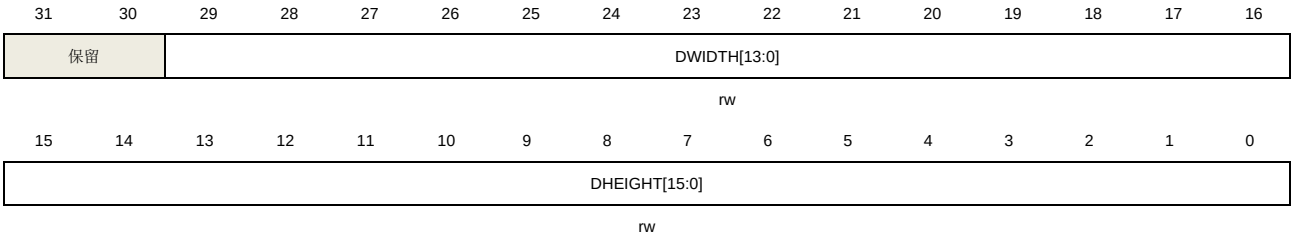
位/位域	名称	描述
31:30	保留	必须保持复位值。
29:16	YSCALE[13:0]	它由2位整数和12位小数组成，表示前景层的Y缩放因子。 **注意：**最大缩小因子为1/2，最大放大因子为2^12。
15:14	保留	必须保持复位值。
13:0	XSCALE[13:0]	它由2位整数和12位小数组成，表示前景层的X缩放因子。 **注意：**最大缩小因子为1/2，最大放大因子为2^12。

35.7.22. 缩放目标图像大小寄存器（IPA_DIMS）

地址偏移：0x54

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。



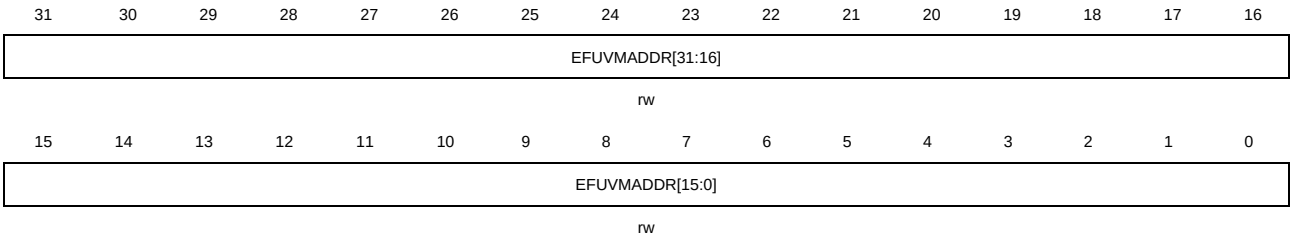
位/位域	名称	描述
31:30	保留	必须保持复位值。
29:16	DWIDTH[13:0]	待处理的图像的宽度 软件置位和清除 该位域表示待处理的图像每行像素的数目。如果背景层或前景层像素格式是A4或L4，这些位必须配置成偶数，否则当传输使能的时候将检测到一个配置错误。 当IPA_CTL寄存器的TEN位为'1'时，该位不可写。 使用缩放功能的时候配置这些位域。
15:0	DHEIGHT[15:0]	待处理图像的高度 软件置位和清除 该位域表明待处理图像的行数。 当IPA_CTL寄存器的TEN为'1'时，该位不可写。 使用缩放功能的时候配置这些位域。

35.7.23. 前景层偶数帧/UV 存储区基地址寄存器（IPA_EF_UV_MADDR）

地址偏移：0x5C

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域	名称	描述
31:0	EFUVMADDR[31:0]	前景层偶数帧/UV存储区基地址 地址必须与目标层像素格式对应的8位、16位或32位对齐。如果前景层像素格式为

ARGB8888、UYVY422或VYUY422，则这些位必须是32位对齐；如果前景层像素格式为 YUV420、YVU420、RGB565、ARGB1555、ARGB4444或AL88，则这些位必须是16位对齐。如果违背以上对齐规则，当传输使能的时候，将检测到一个配置错误。当IPA_CTL寄存器的TEN位为‘1’时，该位不可写。

35.7.24. 色彩空间转换系数配置寄存器 0（IPA_CSCC_CFG0）

地址偏移：0x60

复位值：0x0400 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
CONVMOD	保留							C0[10:0]						UVOFF[8:7]	
rw								rw						rw	

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

位/位域	名称	描述
31	CONVMOD	色彩空间转换模式 0：YUV转RGB，UV值为8位二进制补码 1：将YCbCr转换为RGB，CbCr值为8位无符号数
30:29	保留	必须保持复位值。
28:18	C0[10:0]	Y乘数系数。对于YUV，它通常是0x100(1.000)。对于YCbCr，通常为0x12A(1.164)。
17:9	UVOFF[8:0]	UV/CbCr数据的隐式相位偏移，用于YUV/YCbCr到RGB的转换。对于YUV，它通常是0x000。对于YCbCr，通常为0x180(-128)。
8:0	YOFF[8:0]	Y数据的隐式幅度偏移。对于YUV，通常为0。对于YCbCr，通常为0x1F0(-16)。

35.7.25. 色彩空间转换系数配置寄存器 1（IPA_CSCC_CFG1）

地址偏移：0x64

复位值：0x0123 0208

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					C1[10:0]										
					rw										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					C4[10:0]										
					rw										

位/位域	名称	描述
31:27	保留	必须保持复位值。
26:16	C1[10:0]	V/Cr红色乘数系数。对于YUV，通常为0x123 (1.140)。对于YCbCr，它通常为0x198 (1.596)。
15:11	保留	必须保持复位值。
10:0	C4[10:0]	U/Cb蓝色乘数系数。对于YUV，它通常是0x208(2.032)。对于YCbCr，它通常是0x204(2.017)。

35.7.26. 色彩空间转换系数配置寄存器 2（IPA_CSCC_CFG2）

地址偏移：0x68

复位值：0x 076B 079C

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					C2[10:0]										
					rw										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					C3[10:0]										
					rw										

位/位域	名称	描述
31:27	保留	必须保持复位值。
26:16	C2[10:0]	V/Cr绿色乘数系数。对于YUV，通常为0x76B(-0.581)。对于YCbCr，通常为0x730(-0.813)。
15:11	保留	必须保持复位值。
10:0	C3[10:0]	U/Cb绿色乘数系数。对于YUV，它通常是0x79C(-0.394)。对于YCbCr，通常为0x79C(-0.392)。

36. SDIO 接口（SDIO）（适用于 GD32H737_757_759）

36.1. 简介

安全的数字输入/输出接口（SDIO）定义了 SD 卡、SD I/O 卡和嵌入式多媒体卡（e•MMC）主机接口，提供 AHB 系统总线与 SD 存储卡、SD I/O 卡以及 e•MMC 设备之间的命令和数据传输。

所支持的 SD 存储卡和 SD I/O 卡系统规格书可以通过 SD 卡协会网站（*www.sdcard.org*）获取。

所支持的嵌入式多媒体卡（e•MMC）系统规格书可以通过多媒体卡协会网站（*www.jedec.org*）获取，由 JEDEC 固态技术协会出版。

36.2. 主要特征

SDIO 的主要特征如下：

- **e•MMC**：兼容嵌入式多媒体卡系统规格书 V4.51 及之前的版本。有五种不同的数据总线模式：1 位（默认）、4 位（包括 SDR 和 DDR）和 8 位（包括 SDR 和 DDR）；
- **SD 卡**：与 SD 存储卡规格版本 3.0 全兼容；
- **SD I/O**：与 SD I/O 卡规格版本 3.0 全兼容，有三种不同的数据总线模式：1 位（默认）和 4 位（包括 SDR 和 DDR）；
- 104MHz 数据传输频率和 8 位数据传输模式；
- 中断和 DMA 请求；
- 数据传输支持 DDR 模式。

注意：SDIO 在同一时间仅支持一个 SD、SD I/O 卡或 e•MMC，但可支持多个 e•MMC。

两个 SDIO 接口的实现特点如*表 36-1. 两个 SDIO* 所示。

表 36-1. 两个 SDIO

SDIO features	SDIO0	SDIO1
延迟模块（SDR104，HS200）	+	+
SDIO_CLKIN	+	-
SDIO_CMDDIR，SDIO_DAT0DIR，SDIO_DAT123DIR	+	-
MDMA 命令结束、MDMA 数据结束和 MDMA 缓存结束的触发信号	+	-

注意：“-”表示不支持，“+”表示支持。

36.3. SDIO 功能说明

36.3.1. SDIO 总线拓扑

上电复位之后，主机必须通过特殊的基于消息的总线协议来初始化卡。

每个消息是由以下部分中的一个来表示：

命令：命令是启动一个操作的令牌，从主机发送到卡。命令串行传输在 CMD 线上。

响应：响应是从卡发送到主机，作为先前接收到的命令的回应。响应串行传输在 CMD 线上。

数据：数据可以从卡传输到主机或者从主机传输到卡。数据通过数据线传送。用于数据传输的数据线的数目可以是 1（DAT0）、4（DAT0-DAT3）或 8（DAT0-DAT7）。

命令，响应和数据块的结构在*卡功能描述*章节中介绍。一次数据传输就是一个总线操作。

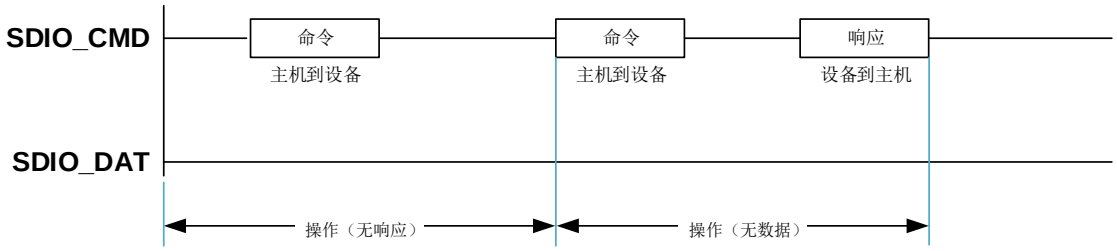
有几种不同类型的操作。一般操作总是包含一个命令和响应。此外，一些操作还有一个数据令牌。还有一些其他操作直接将他们的信息包含在命令或响应结构中。在这种情况下，操作没有数据令牌。在 DAT0-DAT7 和 CMD 信号线上的比特位根据主机时钟同步传输。

三种类型的数据传输命令定义如下：

- 流命令：这些命令发起连续的数据流，只有当 CMD 信号线上出现停止命令时，数据传输终止。该模式将命令的开销减少到最低（仅支持 e•MMC）。
- 面向块的命令：这些命令成功发送一个数据块后紧跟一个 CRC 校验。读和写操作允许单个或多个块传输。与流命令相同，当 CMD 信号线上出现停止命令时，多块传输终止。
- 多字节模式：支持大小为 1-512 字节范围的数据块。（仅 SD/SD I/O 支持）

总线上的基本操作是命令/响应操作（参考*图 36-1. SDIO “无响应”和“无数据”操作*。这种类型的总线事务直接在命令或响应结构中传递它们的信息。此外，有些操作还有数据令牌。卡与设备之间的数据传输通过块完成。

图 36-1. SDIO “无响应”和“无数据”操作



多块操作模式比单块操作速度更快。当 CMD 信号线上出现停止命令时，多块传输终止。主机数据传输可以使用单个或多个数据线。多个块的读操作如*图 36-2. SDIO 多块读操作*所示，多个块的写操作如*图 36-3. SDIO 多块写操作*所示。块的写操作在数据（DAT0）信号线上使用忙

信号。

图 36-2. SDIO 多块读操作

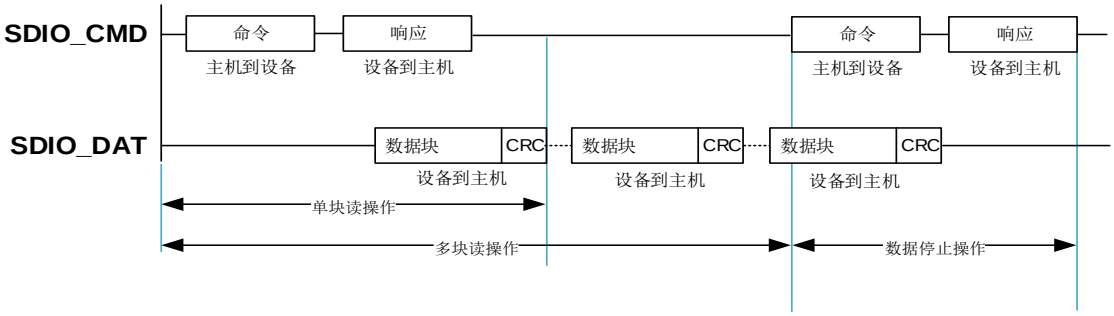
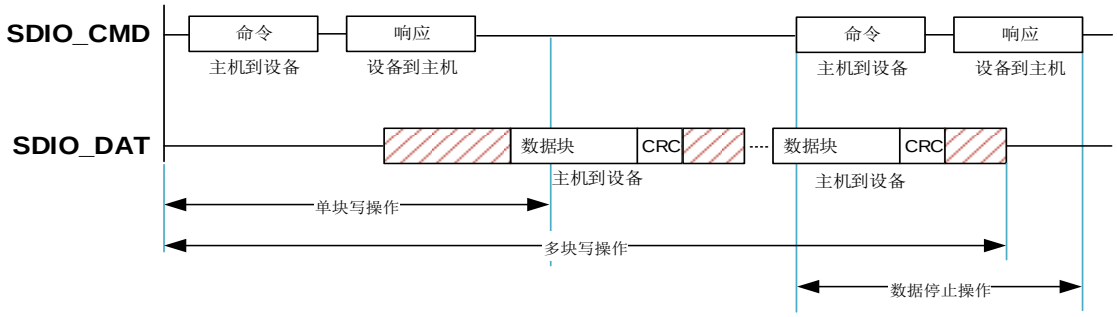


图 36-3. SDIO 多块写操作



SD 存储卡和 SD I/O 卡（包括仅 IO 卡和组合卡）直接的数据传输是以数据块的方式完成的。e•MMC 卡以数据块或数据流方式进行数据传输。*图 36-4. SDIO 数据流读操作*和*图 36-5. SDIO 数据流写操作*分别是数据流的读和写操作。

图 36-4. SDIO 数据流读操作

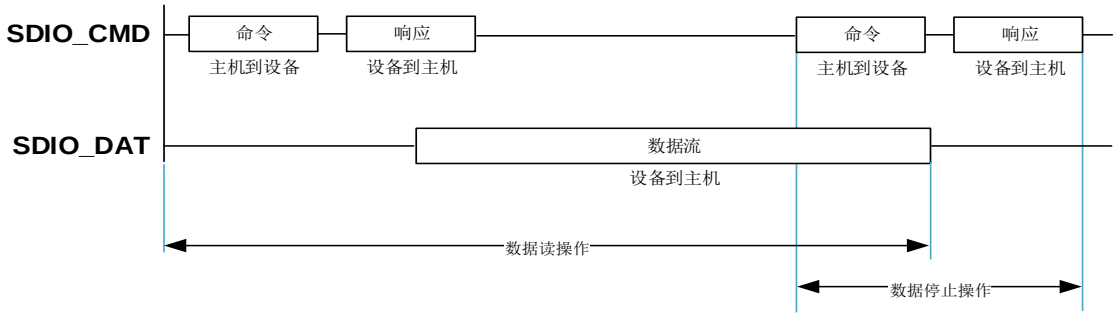
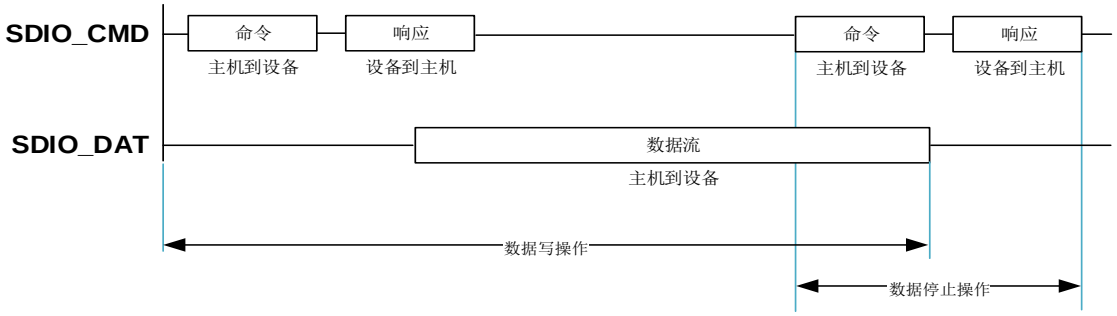


图 36-5. SDIO 数据流写操作



36.3.2. SDIO 操作模式

表 36-2. SDIO 适用 SD & SD/IO 卡的操作模式

总线速度模式	最大总线速度[Mbyte/s]	最大时钟频率[MHz]	电压[V]
DS	12.5	25	3.3
HS	25	50	3.3
SDR12	12.5	25	1.8
SDR25	25	50	1.8
DDR50	50	50	1.8
SDR50	50	100	1.8
SDR104	104	208	1.8

表 36-3. SDIO 适用 e•MMC 卡的操作模式

总线速度模式	最大总线速度[Mbyte/s]	最大时钟频率[MHz]	电压[V]
向后兼容 MMC 卡	26	26	3/1.8
HS SDR	52	52	3/1.8
HS DDR	104	52	3/1.8
HS200	200	200	1.8

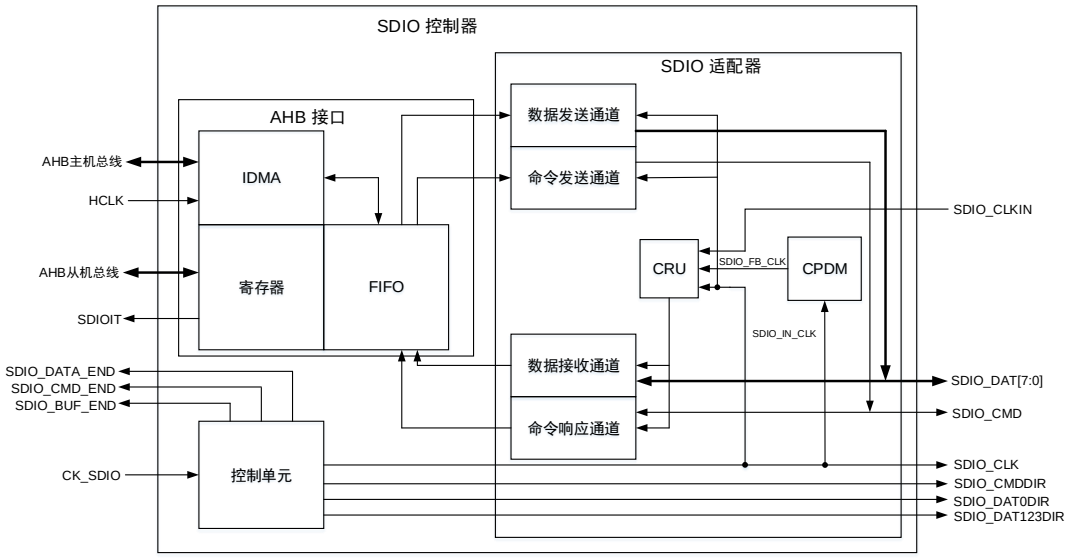
注意：1. DS 表示默认速度，HS 表示高速模式；2. SDR 表示单倍数据速率信号传输，DDR 表示双倍数据速率信号传输；3. SD/SD I/O 卡最大的总线速度时，总线宽度为 4 位；e•MMC 最大的总线速度时，总线宽度为 8 位；4. 最大的频率取决于 I/O 速度；5. SDR104 模式和 HS200 模式需要使用延迟模块来支持采样调谐，但 SDR50 模式时延迟模块是可选择的。

36.3.3. SDIO 框图

*图 36-6. SDIO 框图*显示了 SDIO 的结构框图，主要有三大部分：

- SDIO 适配器：由控制单元、命令单元和数据单元组成。控制单元管理时钟信号，命令单元管理命令的传输，数据单元管理数据的传输。
- AHB 接口：包括通过 AHB 总线访问的寄存器、用于数据传输的 FIFO 单元以及产生中断和 DMA 请求信号。
- 内部 DMA（IDMA）以及 AHB 主机接口

图 36-6. SDIO 框图



36.3.4. SDIO 引脚和内部信号

表 36-4. SDIO 内部输入/输出信号

信号	信号类型	描述
CK_SDIO	输入	SDIO 内核时钟
HCLK	输入	AHB 时钟
SDIOIT	输出	SDIO 全局中断
SDIO_CMD_END	输出	MDMA 的 SDIO 命令结束触发信号
SDIO_DATA_END	输出	MDMA 的 SDIO 数据结束触发信号
SDIO_BUF_END	输出	MDMA 的 SDIO IDMA 缓存结束触发
SDIO_IN_CLK	输入	卡的反馈时钟，信号连接到 SDIO_CLK 脚，用于 DS/HS 模式
SDIO_FB_CLK	输入	CPDM 延迟模块后的 SD/SD I/O/e•MMC 卡调节反馈时钟（用于 SDR50、DDR50、SDR104 以及 HS200）

表 36-5. SDIO 引脚介绍

引脚名称	信号	信号类型	描述
SDIO_CK	SDIO_CLK	输出	SD/SD I/O/e•MMC 卡的时钟
SDIO_CKIN	SDIO_CLKIN	输入	来自卡的外部驱动的时钟反馈（用于 SDR12，SDR25，SDR50，DDR50）
SDIO_CMD	SDIO_CMD	输入/输出	双向命令/响应信号
SDIO_CMDDIR	SDIO_CMDDIR	输出	SDIO_CMD 的方向指示信号
SDIO_D[7:0]	SDIO_DAT[7:0]	输入/输出	用于数据输入/输出的数据线
SDIO_DAT0DIR	SDIO_DAT0DIR	输出	SDIO_DAT0 数据线的方向指示
SDIO_DAT123DIR	SDIO_DAT123DIR	输出	SDIO_DAT[3:1]数据线的方向指示

36.3.5. SDIO 介绍

SDIO_DAT[7:0]线具有不同的工作模式：

- 上电后默认情况下，使用 1 位数据总线（SDIO_DAT0），初始化后，主机可以通过修改寄存器来更改数据总线宽度。
- 对于 SD 或 SDIO 卡，可以使用 1 位（SDIO_DAT0）或 4 位（SDIO_DAT[3:0]）的数据总线宽度。所有数据线均以推挽模式运行。
- 对于 e•MMC，可以使用 1 位（SDIO_DAT0）、4 位（SDIO_DAT[3:0]）或 8 位（SDIO_DAT[7:0]）的数据总线宽度。

为了连接电压切换收发器，使用 I/O 方向信号指示数据线上的数据流方向。SDIO_DAT0DIR 信号指示 SDIO_DAT0 数据线的 I/O 方向，SDIO_DAT123DIR 则指示 SDIO_DAT[3:1]数据线的方向。SDIO_CMD 线上的数据流方向用 I/O 方向信号 SDIO_CMDDIR 表示。SDIO_CMD 只能在推挽模式下运行。

卡的时钟 SDIO_CLK 源自 CK_SDIO 时钟：

- 当 CK_SDIO 的占空比为 50%时，即使分频因子 DIV = 0，也可以使用。
- 当 CK_SDIO 的占空比不是 50%时，必须满足分频因子 DIV > 0，才可以使用。
- SDIO_CMD 和 SDIO_DAT[7:0]输出与 SDIO_CLK 之间的相位关系可以通过 CLKEDGE 位选择，相位关系取决于 DIV、CLKEDGE 以及 DRSEL 设置，详细如*表 36-6. SDIO 命令和数据输出的相位选择*。

表 36-6. SDIO 命令和数据输出的相位选择

DIV	DRSEL	CLKEDGE	SDIO_CLK	命令输出	数据输出
= 0	-	-	= CK_SDIO	在 CK_SDIO 的下降沿生成	
> 0	0	0	产生于 CK_SDIO 的上升沿	在 SDIO_CLK 上升沿后的 CK_SDIO 下降沿生成	
> 0	0	1		在 SDIO_CLK 下降沿后的 CK_SDIO 上升沿生成	
> 0	1	0	产生于 CK_SDIO 的上升沿	在 SDIO_CLK 上升沿后的 CK_SDIO 下降沿生成	在 SDIO_CLK 边沿后的 CK_SDIO 下降沿生成
> 0	1	1		在 SDIO_CLK 下降沿后的 CK_SDIO 上升沿生成	

默认情况下，选择源自 SDIO_CLK 引脚的 SDIO_IN_CLK 反馈时钟输入来采样 SDIO 接收通道中的传入的数据。为了调整采样时钟的相位以适应接收数据的时序，可以将设备上的 CPDM 延迟块连接到 SDIO_FB_CLK 时钟信号和 SDIO_IN_CLK 信号之间。然后选择 SDIO_FB_CLK 作为数据接收通道的时钟，对接收的数据使用相位调整的采样时钟。该功能是 SDR104、HS200 工作模式下必须满足的以及 SDR50、DDR50 工作模式下可选的。

当使用外部驱动器（电压开关收发器）时，可以选择 SDIO_CLKIN 反馈时钟输入信号来采样接收的数据。

SD/SD I/O/e•MMC 卡的时钟频率范围为 0 到 208MHz（最高的时钟频率受限于 I/O 速度）。

根据所选的总线模式（SDR 或 DDR），每个时钟周期在 SDIO_DAT[7:0]线上传输一位或两位。SDIO_CMD 线每个时钟周期仅传输一位。

36.3.6. SDIO 适配器

SDIO 适配器包括控制单元、命令单元和数据单元，并且可以向卡生成信号。这些信号的具体描述如下：

SDIO_CLK：SDIO 控制器提供给卡的时钟。每个时钟周期在命令线（SDIO_CMD）上只发送一位命令或数据。对于 MMC 卡 V4.2 版本可以在 0 MHz 到 48MHz 之间，对于 e•MMC V4.51 版本可以在 0 MHz 到 200MHz 之间，对于 SD 或 SD I/O 卡可以在 0 MHz 到 208MHz。

SDIO 使用两个时钟信号：SDIO 适配器时钟和 AHB 总线时钟（HCLK）。

SDIO_CMD：该信号是双向命令通道，用于卡的初始化和命令的传输。命令从 SDIO 控制器发送到卡，响应从卡发送到主机。CMD 信号有两种操作模式：用于初始化的开漏模式（仅用于 e•MMC 卡的初始化）和用于命令传送的推挽模式（SD 卡/SD I/O 卡的初始化时以及 e•MMC 卡的数据传输时使用）。

SDIO_DAT[7:0]：这些信号线都是双向数据通道。数据信号线操作在推挽模式。每次只有卡或者主机会驱动这些信号。默认情况下，上电或者复位后仅 DAT0 用于数据传输。SDIO 适配器可以配置更宽的数据总线用于数据传输，使用 DAT0-DAT3 或者 DAT0-DAT7（仅适用于 e•MMC）。SDIO 对数据信号线 DAT1-DAT7 有内部上拉。在进入 4 位模式后，卡断开 DAT1-DAT3 的内部上拉。相应地，在进入 8 位模式后，断开 DAT1-DAT7 的内部上拉。

SDIO_CLKIN：该信号线是数字输入信号线，用于 SD/SD I/O/e•MMC 卡的外部驱动的时钟反馈。（用于 SDR12，SDR25，SDR50，DDR50）

SDIO_CMDDIR：该信号是数字输出信号，作为 SDIO_CMD 信号线上数据流的输入/输出方向指示信号。

SDIO_DAT0DIR：该信号是数字输出信号，作为 SDIO_DAT0 数据线上数据流的输入/输出方向指示信号。

SDIO_DAT123DIR：该信号是数字输出信号，作为 SDIO_DAT[3:1]数据线上数据流的输入/输出方向指示信号。

SDIO 适配器是总线设备，提供与 SD/SD I/O /e•MMC 卡的接口，它由几个子单元组成：

控制单元

控制单元包含电源管理功能、用于存储卡时钟的时钟管理功能以及 I/O 方向管理功能。

电源管理是由 SDIO_PWRCTL 寄存器控制的，实现电源的掉电、上电。电源管理子单元会在复位后、断电阶段和上电阶段中禁止卡总线输出信号。通过设置 SDIO_CLKCTL 的 CLKPWRSAV 位来配置省电模式，实现当总线空闲时，关闭 SDIO_CLK。

时钟管理使用 CK_SDIO 向卡生成 SDIO_CLK 时钟信号，并提供分频控制。适配器寄存器和 FIFO 使用 AHB 时钟域（HCLK）。控制单元、命令发送通道和数据发送通道使用 SDIO 适配器时钟域（CK_SDIO）。命令响应通道和数据接收通道使用来自 SDIO_IN_CLK、SDIO_CLKIN 或 SDIO_FB_CLK（由 CPDM 生成）的 SDIO 配器反馈时钟域。

I/O 方向管理控制外部电压收发器以及控制 SDIO_CMDDIR、SDIO_D0DIR 和 SDIO_D123DIR 的信号。

命令单元

命令单元包含命令发送通道和响应接收通道，作用是在 SDIO_CMD 线上实现向卡发送命令和接收响应。命令发送通道由 SDIO_CLK 提供时钟，响应接收通道有专用的 SDIO 内部接收时钟，数据传输流由命令状态机（CSM）控制。在对 SDIO_CMDCTL 寄存器进行一次写操作并设置该寄存器的 CSMEN 位为 1 后，命令传输开始。首先向卡发送一个命令，这个命令包含 48 位，通过 SDIO_CMD 线发出，每个 SDIO_CLK 发送一个比特数据。这 48 位命令包含 1 位起始位、1 位传输位、6 位命令索引（由 SDIO_CMDCTL 寄存器的 CMDIDX 位定义）、32 位参数（由 SDIO_CMDAGMT 定义）、7 位 CRC 和 1 位停止位。然后接收来自卡的响应（在 SDIO_CMDCTL 寄存器的 CMDRESP 位不为 0b00 的情况下），响应分为 48 位的短响应和 136 位的长响应，响应都存在 SDIO_RESP0 - SDIO_RESP3 寄存器中。命令单元同样可以产生命令状态标志（在 SDIO_STAT 寄存器中定义）。

命令状态机

CS_Idle	复位后准备发送命令	
1.CSM 被使能并且 WAITDEND 使能	→	CS_Pend
2.CSM 被使能并且 WAITDEND 失能并且 BOOT 未使能	→	CS_Send
3.CSM 被关闭	→	CS_Idle
4.BOOTMODEN 置位	→	CS_Boot
注意：命令状态机在空闲状态至少保持 8 个 SDIO_CLK 周期，以满足 N_{CC} 和 N_{RC} 时序限制。N_{CC} 是两个主机命令之间的最小时间间隔，N_{RC} 是主机命令与卡响应之间的最小时间间隔。		

CS_Pend	等待数据传输结束	
1.数据传送完成	→	CS_Send
2.CSM 被关闭	→	CS_Idle
注意：DATALEN =< 5 时，CSM 直接变为 CS_Send 状态；DATALEN > 5 时，CSM 会等待 DSM 的信号后再变为 CS_Send 状态。		

CS_Send	发送命令	
1.命令发送后有响应	→	CS_Wait
2.命令发送后无响应	→	CS_Idle
3.CSM 被关闭	→	CS_Idle

CS_Wait	等待响应起始位	
1.接收到响应（检测到起始位）	→	CS_Receive
2.接收响应超时	→	CS_Idle
3.CSM 被关闭	→	CS_Idle
注意：命令超时时间固定为 64 个 SDIO_CLK 时钟周期。		

CS_Receive	接收响应并检测 CRC

1.CSM 被关闭	→	CS_Idle
2.收到响应	→	CS_Idle
3.命令 CRC 检测失败	→	CS_Idle

CS_Boot	从卡读取引导数据	
1.选择正常引导模式且使能引导模式	→	CS_Boot
2.选择正常引导模式且禁止引导模式	→	CS_Idle
3.选择备用引导模式	→	CS_Send

数据单元

数据单元实现主机与卡之间的数据传输。当数据宽度为 8 位（SDIO_CLKCTL 寄存器的 BUSMODE 位为 0b10）时，数据传输使用 SDIO_DAT[7:0]信号线；当数据宽度为 4 位（SDIO_CLKCTL 寄存器的 BUSMODE 位为 0b01）时，数据传输使用 SDIO_DAT[3:0]信号线；当数据宽度为 1 位（SDIO_CLKCTL 寄存器的 BUSMODE 位为 0b00）时，数据传输使用 SDIO_DAT[0]信号线。数据传输流由数据状态机（DSM）控制。在对 SDIO_DATACTL 寄存器进行一次写操作并将 SDIO_DATACTL 寄存器的 DATAEN 位为 1，数据传输开始。当 SDIO_DATACTL 寄存器的 DATADIR 位为 0 时，数据是从控制器到卡；当 DATADIR 位为 1 时，数据是从卡到控制器。数据单元同样可以产生数据状态标志（在 SDIO_STAT 寄存器中定义）。数据接收时，对于包含引导确认操作时，DSM 变为 DS_WaitACK 状态并等待引导确认，然后再变为 DS_WaitR 状态。

数据状态机

DS_Idle	数据单元不工作，等待发送和接收数据	
1.（DSM 使能或者 DATAEN 位使能）并且（DSM 不忙且数据传输方向为主机到卡）	→	DS_WaitS
2.（DSM 使能或者 DATAEN 位使能）且禁用引导确认且数据传输方向为卡到主机	→	DS_WaitR
3.DSM 使能、使引导确认且数据传输方向为卡到主机	→	DS_WaitACK

DS_WaitS	等待数据 FIFO 为空标志无效或者数据传输结束	
1.数据传输结束	→	DS_Idle
2.DSM 被关闭且数据 FIFO 为空	→	DS_Idle
3.数据传送保持标志置位	→	DS_Idle
4.数据 FIFO 为空标志无效且数据保持标志清零	→	DS_Send

DS_Send	发送数据到卡	
1.数据块已发送	→	DS_Busy
2.DSM 被关闭	→	DS_Busy
3.内部 CRC 错误	→	DS_Busy

DS_Busy	等待 CRC 状态标志	
1.接收到正确 CRC 状态并且卡不繁忙	→	DS_WaitS
2.没有接收到正确 CRC 状态且卡不繁忙	→	DS_WaitS
3.接收到错误的 CRC 状态且卡不繁忙	→	DS_Idle
4.DSM 被关闭且卡不繁忙	→	DS_Idle
注意：命令超时时间设置在数据超时寄存器（SDIO_DATATO）中。		

DS_WaitR	等待接收数据的起始位	
1.数据接收结束	→	DS_Idle
2.DSM 被关闭	→	DS_Idle
3.数据保持且 FIFO 为空	→	DS_Idle
4.在超时前收到起始位	→	DS_Receive
注意：命令超时时间设置在数据超时寄存器（SDIO_DATATO）中。		

DS_Receive	接收卡的数据并将其写入数据 FIFO	
1.数据块已接收且读等待模式禁用	→	DS_WaitR
2.数据传输结束且读等待模式禁用	→	DS_WaitR
3.数据 FIFO 上溢错误发生	→	DS_Idle
4.DSM 被关闭且 FIFO 为空	→	DS_Idle
5.CRC 错误且数据接收完成	→	DS_Idle

DS_WaitACK	等待引导确认令牌	
1.及时收到引导确认，且检验 OK	→	DS_WaitR
2.确认超时或收到错误的确认	→	DS_WaitACK
3.DSM 被关闭且 FIFO 为空	→	DS_Idle

CRU 单元

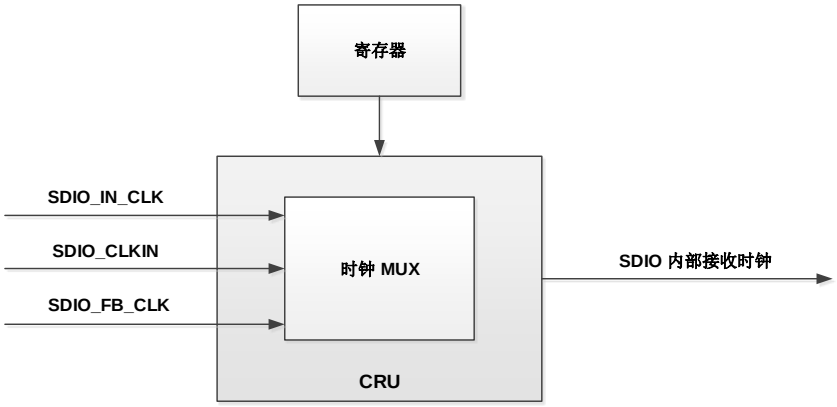
CRU 单元为 SDIO 内部接收时钟选择时钟源，SDIO 内部接收时钟用于接收数据和命令响应。

通过配置 RCLK[1:0]寄存器来选择 SDIO 内部接收时钟的时钟源，有三种时钟可供选择：

- SDIO_IN_CLK 总线主反馈时钟
- SDIO_CLKIN 外部总线反馈时钟
- SDIO_FB_CLK 总线调谐反馈时钟

注意：1. 当没有外部驱动且使用 DS/HS 总线模式时，选择 SDIO_IN_CLK。2. 当有外部总线驱动且在 SDR12、SDR25、SDR50、DDR50 总线模式时，选择 SDIO_CLKIN 时钟。3. 使用 CPDM 延迟模块时，如果是 SDR104、HS200 模式，则必须选择 SDIO_FB_CLK 时钟输入；如果是 SDR50、DDR50 则可选择 SDIO_FB_CLK 时钟输入。4. 如果 CPM 和 DSM 都处于空闲状态时，SDIO 内部接收时钟的时钟源必须被改变。

图 36-7. CRU 单元



36.3.7. AHB 从接口

AHB 从接口用于访问 SDIO 寄存器、数据 FIFO 以及生成中断。包括数据 FIFO 单元、寄存器单元和中断。

AHB 从机接口包含以下子单元：

中断

当选择的状态标志中至少有一位为高时，中断逻辑产生中断。单元提供了中断使能寄存器逻辑来控制相应中断产生的开关。如果设置了相应的中断使能标志，则状态标志寄存器中产生一个中断请求。一些状态标志需要在中断标志清除寄存器中隐式清除。

寄存器单元

寄存器单元包含所有系统寄存器，目的是管理控制器与卡之间通信的信号。

数据 FIFO

数据 FIFO 单元有一个数据缓冲区，用于发送和接收 FIFO。FIFO 包含一个每个字的宽度为 32 位，深度为 32 字的数据缓冲区。当使用半字或字节访问 FIFO 时，会产生 AHB 总线故障。

发送 FIFO 用于写数据到卡的操作。将要发送的数据通过 AHB 总线写入发送 FIFO，SDIO 适配器中的数据单元从发送 FIFO 读取数据，然后将数据发送到卡上。当 DATALEN 不是 4 的整数倍时，最后剩余的数据（1/2/3 字节）将以字传输的方式写入。

接收 FIFO 用于从卡读取数据的操作。将要传输的数据从卡读取，然后写入接收 FIFO。接收 FIFO 中的数据在需要时读取到 AHB 总线。当 DATALEN 不是 4 的整数倍时，读取最后剩余的数据（1/2/3 字节）时使用填充 0 值字节的字传输。

36.3.8. AHB 主接口

AHB 主接口使用 SDIO 内部 DMA（IDMA）在存储和 FIFO 之间进行数据传输。

IDMA

IDMA 在存储和 FIFO 之间提供了一个双向高速数据传输通道。

通过 IDMAEN 位来使能 IDMA，IDMA 支持 8 节拍的突发模式传输。

突发模式具体分为突发模式发送和突发模式接收：

- 突发模式发送
  - 只要 FIFO 对突发传输次数为空，数据就会以突发的形式从内存中获得，直到 DATALEN 所指示的所有数据都传输完毕。如果 DATALEN 不是突发大小的倍数，则小于突发大小的剩余数据将以单次传输方式传输。当 DATALEN 不是 4 的倍数时，最后剩余的数据（1、2 或 3 字节）通过字传输获得。
- 突发模式接收
  - 只要 FIFO 对突发传输次数为空，数据就会以突发的形式存储在内存中，直到 DATALEN 所指示的所有数据都传输完毕。如果 DATALEN 不是突发大小的倍数，则小于突发大小的剩余数据将以单次传输方式传输。当 DATALEN 不是 4 的倍数时，最后剩余的数据（1、2 或 3 字节）以半字或字节传输的方式存储。

此外，IDMA 还提供了两种通道配置（由 BUFMOD 位选择）。

- 单缓冲通道
  - 在单个缓冲区配置中，访问内存端的数据是从基址 IDMAADDR0 线性访问的。当 IDMA 完成所有数据的传输，DSM 也完成传输时，DTEND 标志设为 1。
- 双缓冲通道
  - 在双缓冲区配置中，内存端的数据从两个缓冲区访问，一个来自 IDMAADDR0，另一个来自 IDMAADDR1。这样，当 IDMA 访问其中一个内存缓冲区时，固件可以处理另一个。内存缓冲区的大小由 IDMASIZE 定义。缓冲区大小应该是突发大小的倍数。当通道被启用时，缓冲区的基址可以立即被更新。

在双缓冲通道模式下，通过 BUFSEL 寄存器位配置访问内存的地址。

- 当 BUFSEL 位为“0”时，IDMA 硬件使用 IDMAADDR0 访问内存。当尝试通过固件写入该位寄存器时，写入操作将被忽略，IDMAADDR0 数据不会改变。但允许固件写入 IDMAADDR1。

- 当 BUFSEL 位为“1”时，IDMA 硬件使用 IDMAADDR1 访问内存。当尝试通过固件写入该寄存器时，写入操作将被忽略，IDMAADDR1 数据将不会更改。但允许固件写入 IDMAADDR0。

当 IDMA 在其中一个缓冲区完成数据传输时，将缓冲区传输完成标志（IDMAEND）设置为 1，并翻转 BUFSEL 位，然后 IDMA 将继续从另一个缓冲区传输数据。当 IDMA 完成所有数据的传输，DSM 也完成传输时，DTEND 标志设为 1。

IDMAADDR0 和 IDMAADDR1 地址应该是字对齐的。

36.3.9. MDMA 请求

SDIO 的内部触发线（SDIO_DATA_END、SDIO_CMD_END 和 SDIO_BUF_END 线）可以直接向 MDMA 控制器发送请求，实现从/到不同内部 RAM 地址的连续传输，而不需要 CPU 参

与。

MDMA 的请求输入信号通过 SDIO_DATA_END 引脚传输。同时，输入信号会触发清除 DTEND 和 CMDRECV 两个标志，通过 MDMA 直接访问 SDIO 控制和配置寄存器，开始新的传输，而无需 CPU 干预。

当成功接收到对命令的响应时，将置位 CMDRECV 标志。R1b 响应忙状态结束后，清除状态寄存器 DAT0BSY 的标志位并置位 DAT0BSYEND 标志位。当与最终繁忙信号相关联的序列命令响应结束时，设置连接到 MDMA 的 SDIO_CMD_END 输出。因此，MDMA 可以通过清除 CMDRECV 和 DAT0BSYEND 状态标志来管理 CMD12（STOP_TRANSMISSION）命令（需要支持开放模式传输）。

在使用 Linux 操作系统时，要通过 SDIO 总线传输的数据包含在设备内部内存中的不连续地址的 1-4Kbyte 大小的数据块中。双缓冲模式允许改变 IDMA 在内部内存中的目标地址。每次缓冲区传输完成时，会置位状态寄存器的 IDMAEND 标志。通过连接到 MDMA 请求输入的 SDIO_BUF_END 输出将此事件发送给 MDMA，新的缓冲区基地址可以在不需要 CPU 干预的情况下自动填充 IDMAADDR0 / IDMAADDR1 字段。

*表 36-7. SDIO 与 MDMA 的连接*显示了根据 SDIO 的请求在 MDMA 中的编程动作：

表 36-7. SDIO 与 MDMA 的连接

触发信号	事件	条件	MDMA 传输配置	MDMA 动作
SDIO_DATA_END	数据成功传输结束	DTEND = 1	单独	置位 DTENDC
SDIO_CMD_END	命令序列结束	CMDSEND = 1，或（CMDRECV = 1 且 DAT0BSY = 0）	单独	置位 CMDSENDC 置位 CMDRECVC 置位 DAT0BSYENDC
SDIO_BUF_END	到达缓冲区尾	IDMAENDC = 1	联动	置位 IDMAENDC 更新 IDMAADDR0/1

36.3.10. AHB 总线与 SDIO_CLK 时钟的关系

AHB 总线传输速度带宽应当是 SDIO 总线带宽的 3 倍以上。

表 36-8. AHB 和 e•MMC 时钟频率关系

SDIO 总线模式	总线宽度	最大 SDIO_CLK 时钟频率 [MHz]	最小 AHB 时钟频率 [MHz]
DS	8	26	19.5
HS	8	52	39
HS DDR	8	52	78
HS200	8	200	150

表 36-9. AHB 和 SD/SD I/O 卡时钟频率关系

SDIO 总线模式	总线宽度	最大 SDIO_CLK 时钟频率 [MHz]	最小 AHB 时钟频率 [MHz]
SDR12（DS）	4	25	9.4
SDR25（HS）	4	50	18.8
SDR50	4	100	37.5

DDR50	4	50	37.5
SDR104	4	208	78

36.3.11. 硬件流控制

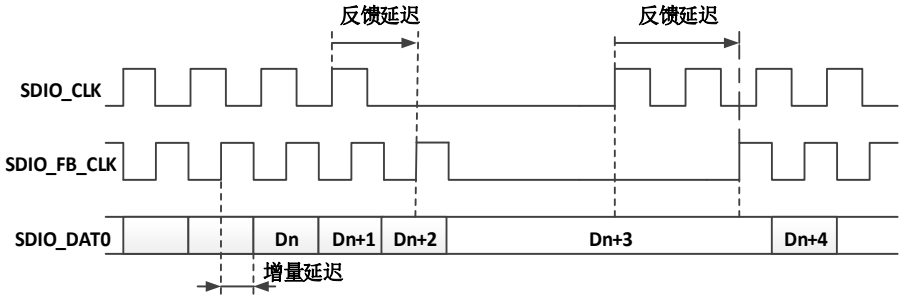
硬件流控制功能在数据传输期间通过停止 SDIO_CLK 来冻结状态机，从而阻止 FIFO 下溢（发送数据）和上溢（接收数据）的错误发生。

当 FIFO 不能够发送或接收数据时，数据传输会暂停，保持暂停状态直到发送 FIFO 数据发送了一半或 DATALEN 长度的数据被存储，或直到接收 FIFO 数据已经接收了一半。当状态机冻结时，AHB 接口会保持活跃状态，因此即使执行硬件流控制时 FIFO 依旧可以操作。

硬件流控制通过置位 HWEN 寄存器位使能。

只有当 SDIO_DAT 数据与 SDIO_CLK 周期对齐时，才使用硬件流控制。SDR104 模式使用 CPDM 延迟块不能使用硬件流控制。

图 36-8 硬件流控制时序



36.4. 卡功能描述

36.4.1. 卡寄存器

卡内部定义了接口寄存器：OCR，CID，CSD，EXT_CSD，RCA，DSR 和 SCR。这些寄存器只能通过相应的命令来访问。OCR，CID，CSD 和 SCR 寄存器包含卡的一些特定信息，而 RCA 和 DSR 寄存器是配置寄存器，存储实际的配置参数。EXT_CSD 寄存器同时包含卡的特定信息和实际的结构参数。有关具体信息，请参考相关的规范。

OCR 寄存器：32 位操作条件寄存器（OCR）储存卡的 V_{DD} 电压描述和存取模式指示（e•MMC）。另外，该寄存器包括一个状态信息位。如果卡上电过程已经完成该状态位被置位。该寄存器在 e•MMC 和 SD 卡之间有一点不同。主机可以使用 CMD1（e•MMC），ACMD41（SD 存储卡），CMD5（SD I/O）来获取该寄存器的内容。

CID 寄存器：卡识别寄存器（CID）是 128 位宽。它包含在卡识别阶段使用的卡识别信息。每个读/写（RW）卡应具有唯一的标识号。主机可以使用 CMD2 和 CMD10 得到这个寄存器的内容。

CSD 寄存器：卡特定数据寄存器提供访问卡中的内容信息。CSD 定义了数据格式、错误校正类型、最大数据访问时间、数据传输速度、DSR 寄存器是否可以使用等。寄存器的可编程部分

可通过 CMD27 来修改。主机可以使用 CMD9 得到这个寄存器的内容。

扩展 CSD 寄存器：只有 *e*•MMC4.51 有该寄存器。扩展 CSD 寄存器定义卡属性和选择模式。它的长度为 512 字节。最高 320 字节为属性段，定义了卡的功能，并且不能由主机修改。最低 192 字节是模式段，定义了卡工作在哪种配置下。这些模式可以由主机通过 SWITCH 命令来修改。主机可以使用 CMD8（仅 *e*•MMC 支持这个命令），以获取该寄存器的内容。

RCA 寄存器：可写的 16 位相对卡地址寄存器存放卡地址，该地址在卡的初始化期间由卡向外发布。这个地址用于卡识别过程之后，所寻址的主机和卡通信。主机可以使用 CMD3 要求卡发布一个新的相对地址（RCA）。

注意：RCA 的寄存器的缺省值是 0x0001（*e*•MMC）或 0x0000（SD/SD I/O）。这个数值是保留值，用于通过 CMD7 设置所有卡到待机（Stand-by）状态。

DSR 寄存器（可选）：16 位驱动阶段寄存器是可选的，可用于在扩展操作条件中提高总线性能（取决于类似于总线长度，传输速率和卡数目这些参数）。CSD 寄存器中有 DSR 寄存器使用情况的信息。DSR 寄存器的默认值是 0x404。主机可以使用 CMD4 得到这个寄存器的内容。

SCR 寄存器：仅 SD/ SD I/O（如果有存储模块）有这个寄存器。除了 CSD 寄存器，还有另一种配置寄存器名为 SD 卡配置寄存器（SCR），它仅用于 SD 卡。SCR 提供了被配置到特定 SD 存储卡的特殊功能的信息。SCR 寄存器的大小是 64 位。该寄存器应在出厂前通过 SD 存储卡制造商进行设置。主机可以使用 ACMD51 得到这个寄存器的内容。

36.4.2. 命令

命令类型

有四种控制卡的命令：

- 广播命令（bc），发送到所有卡，没有响应；
- 带响应的广播命令（bcr），发送到所有卡，同时从所有卡收到响应；
- 寻址（点对点）命令（ac），发送到寻址的卡上，DAT 信号线没有数据传输；
- 寻址（点对点）的数据传输的命令（adtc），发送到寻址的卡上，DAT 信号线进行数据传输。

命令格式

所有命令都是 48 位的固定码长，如*图 36-9. 命令标记格式*所示，需要 1.92 us（25 MHz）0.96 us（50 MHz）和 0.92us（52 MHz）的发送时间。

图 36-9. 命令标记格式

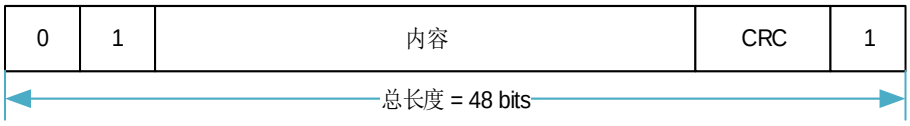


表 36-10. 命令格式

位	47	46	[45:40]	[39:8]	[7:1]	0
宽度	1	1	6	32	7	1
数值	'0'	'1'	x	x	x	'1'
描述	起始位	传输位	命令索引	参数	CRC7	结束位

一个命令总是从一个起始位（始终为 0）开始，随后的位表示传输的方向（主机=1）。接下来的 6 位表示命令的索引，该值被解释为一个二进制编码的数字（0 到 63 之间）。一些命令需要一个参数（例如，一个地址），由 32 位编码。上面表中的表示为“x”的值表示这个变量依赖于该命令。所有的命令有一个 CRC 7 位校验，由结束位（总是 1）终止。

命令分类

卡的命令集分为几类（见*表 36-11. 卡命令类 （CCCs）*）。每类支持一组卡的功能。*表 36-11. 卡命令类 （CCCs）*根据卡支持的命令来决定 CCC 的设置。

对于 SD 卡，类别为 0，2，4，5 和 8 的命令是强制的，应被 SD 卡支持。类别 7 中除了 CMD40 以外都是强制性用于 SDHC。其他类是可选的。所支持的卡命令类（CCC）被编码为参数，设置在每个卡的卡特定数据（CSD）寄存器，提供给主机如何访问该卡信息。

对于 *e*•MMC 卡，类别为 0 的命令是强制性的，应被 *e*•MMC 卡支持。其他类只对特定类型的卡是强制或是可选的。通过使用不同的类，可以选择几种配置（例如，一个块可写的卡或流可读的卡）。所支持的卡命令类（CCC）被编码为参数，设置在每个卡的卡的特定数据（CSD）寄存器，提供给主机如何访问该卡信息。

表 36-11. 卡命令类 （CCCs）

	卡命令类（CCC）	0	1	2	3	4	5	6	7	8	9	10	11
支持的命令	类描述	basic	Stream read	Block read	Stream write	Block write	erase	write protection	Lock card	application specific	I/O mode	switch	reserved
CMD0	M	+											
CMD1	M	+											
CMD2	M	+											
CMD3	M	+											
CMD4	M	+											
CMD5	M	+									+		
CMD6	M	+											
CMD7	M	+											
CMD8	M	+											
CMD9	M	+											
CMD10	M	+											

CMD11	M	+	+										
CMD12	M	+											
CMD13	M	+											
CMD14	M	+											
CMD15	M	+											
CMD16	M			+		+			+				
CMD17	M			+									
CMD18	M			+									
CMD19	M	+											
CMD20	M				+								
CMD21	M			+									
CMD23	M			+		+							
CMD24	M					+							
CMD25	M					+							
CMD26	M					+							
CMD27	M					+							
CMD28	M							+					
CMD29	M							+					
CMD30	M							+					
CMD32	M						+						
CMD33	M						+						
CMD34	M												+
CMD35	O						+						
CMD36	O						+						
CMD37	O												+
CMD38	M						+						
CMD39											+		
CMD40											+		
CMD42									+				
CMD49						+							
CMD52	O										+		
CMD53	O										+	+	
CMD54	O											+	
CMD55	M									+			
CMD56	M									+			
CMD57	O												+
ACMD6	M									+			
ACMD13	M									+			
ACMD22	M									+			
ACMD23	M									+			
ACMD41	M									+			
ACMD42	M									+			
ACMD51	M									+			

注意： 1. M：强制，O：可选。
2. SD 卡中 CMD5 是 0 类命令，在 e•MMC 中，CMD5 是 9 类命令。
3. 支持 UHS-I 的 SD 卡中 CMD11 是 0 类命令且强制的，其他卡中是可选的；在 e•MMC 中，CMD11 是 1 类命令。
4. CMD14，CMD20，CMD21，CMD26，CMD39，CMD40，CMD49 和 CMD54 仅用于 e•MMC；CMD32，CMD33，CMD52，CMD57 和 ACMDx 仅用于 SD 存储卡。
5. 在使用 ACMDx 命令之前发送 APP_CMD 命令（CMD55）。
6. CMD5，CMD8，CMD11 和 CMD53 对于 e•MMC 卡和 SD 卡有不同的含义。
7. 命令类 1 和 3 是被废弃的。

详细的命令描述

下列表详细描述了所有的总线命令。响应 R1-R7 将在*响应*章节说明。寄存器 CID，CSD 和 DSR 在*卡功能描述*介绍。卡应忽略参数中填充位和保留位。

表 36-12. 基本命令（class 0）

命令索引	类型	参数	响应格式	简称	描述
CMD0	bc	[31:0] 00000000	-	GO_IDLE_STATE	复位所有的卡到空闲状态。
	bc	[31:0] F0F0F0F0	-	GO_PRE_IDLE_STATE	复位所有的卡到 pre-idle 状态
	-	[31:0] FFFFFFFA	-	BOOT_INITIATION	开始备用引导操作
CMD1	bc	[31:0] OCR	R3	SEND_OP_COND	在空闲状态，请求卡通过 CMD 线发送响应（包含操作条件寄存器的内容）。
CMD2	bcr	[31:0] 填充位	R2	ALL_SEND_CID	请求任何卡通过 CMD 线发送发送 CID 数据（任何连接到主机的卡都会响应）。
CMD3	bcr	[31:0] 填充位	R6	SEND_RELATIVE_ADDR	请求卡发布新的相对卡地址（RCA）。
CMD4	bc	[31:16] DSR [15:0] 填充位	-	SET_DSR	设置所有卡的 DSR 寄存器。
CMD5	bcr	[31:25]保留位 [24]S18R [23:0] I/O OCR	R4	IO_SEND_OP_COND	仅适用于 I/O 卡。它类似于用于 SD 存储卡的 ACMD41 命令，用于查询所需要的 I/O 卡的电压范围。
CMD5	ac	[31:16]RCA [15]睡眠/唤醒 [14:0] 填充位	R1b	SLEEP_AWAKE	仅适用于 e•MMC，在睡眠和待机（stand-by）状态之间切换

命令索引	类型	参数	响应格式	简称	描述
CMD6	adtc	[31] 模式 0：检查功能 1：切换功能 [30:24] 保留（全‘0’） [23:20]为功能组 6 保留（“0”或“F”） [19:16] 为功能组5 保留（“0”或“F”） [15:12] 为功能组4 保留（“0”或“F”） [11:8] 为功能组 3 保留（“0”或“F”） [7:4] 功能组 2 命令系统 [3:0] 功能组 1 访问模式	R1	SWITCH_FUNC	仅适用于 SD 卡。检查可以切换的功能（模式 0），卡切换功能（模式 1）。
CMD6	ac	[31:26] 设为 0 [25:24] 访问 [23:16] 索引 [15:8] 值 [7:3] 设为 0 [2:0] 命令集	R1b	SWITCH	仅适用于 e•MMC 卡。切换所选卡的操作模式，或修改 EXT_CSD 寄存器。
CMD7	ac	[31:16] RCA [15:0] 填充位	R1b	SELECT/DESELECT_CARD	这个命令用于卡在待机（stand-by）状态和发送（transfer）状态之间切换，或编程（programming）状态和断开（disconnects）状态之间切换。在两种情况下，要选中该卡用它自己的相对地址，若不选中该卡用任何其他地址。地址 0 用于取消选择该卡。
CMD8	bcr	[31:12]保留位 [11:8] 工作电压（VHS） [7:0]检查模式	R7	SEND_IF_COND	向 SD 存储卡发送接口条件，包括主机供电电压信息和询问卡是否支持电压。保留位应设为 0。
CMD8	adtc	[31:0] 填充位	R1	SEND_EXT_CSD	仅用于 e•MMC 卡。卡发送自己的 EXT_CSD 寄存器作为数据块。
CMD9	ac	[31:16] RCA [15:0] 填充位	R2	SEND_CSD	被选定的卡通过CMD线发送它的卡特定数据（CSD）。

命令索引	类型	参数	响应格式	简称	描述
CMD10	ac	[31:16] RCA [15:0] 填充位	R2	SEND_CID	被选定的卡通过CMD线发送它的卡标识（CID）。
CMD11	ac	[31:0] 00000000	R1	VOLTAGE_SWITCH	切换到 1.8V 总线信号水平。
CMD12	ac	[31:0] 填充位	R1b	STOP TRANSMISSION	强制卡停止传输。
CMD13	ac	[31:16] RCA [15:0] 填充位	R1	SEND_STATUS	被选定的卡发送它的状态寄存器。
CMD14	adtc	[31:0] 填充位	R1	BUSTEST_R	主机从卡中读取反向的总线测试数据模式。
CMD15	ac	[31:16] RCA [15:0] 保留位	-	GO_INACTIVE_STATE	将被选定的卡转换到非激活（Inactive）状态。这个命令被用于当主机明确地想停用一张卡的时候。
CMD19	adtc	[31:0] 填充位	R1	BUSTEST_W	主机向卡发送总线测试模式。

表 36-13. 面向块的读命令（class 2）

命令索引	类型	参数	响应格式	简称	描述
CMD16	ac	[31:0]块长度	R1	SET_BLOCKLEN	在标准容量 SD 卡和 e•MMC 卡的情况下，该命令为所有后续块命令（读，写，锁）设置块长度（以字节为单位）。默认值是 512 字节。只有在 CSD 中局部块读操作被允许时，设置长度对于存储器访问命令有效。 在高容量 SD 存储卡的情况下，块长度是由 CMD16 命令设置，不会影响内存读和写命令。总是使用 512 字节的固定块长度。在这两种情况下，如果块长度设置大于 512 字节，BLOCK_LEN_ERROR 位会被卡置位。
CMD17	adtc	[31:0]数据地址	R1	READ_SINGLE_BLOCK	在标准容量 SD 卡和 e•MMC 卡的情况下，通过 SET_BLOCKLEN 命令读取所选择大小的块。 在高容量存储卡的情况下，块长度是固定的 512 字节，忽略 SET_BLOCKLEN 命令。
CMD18	adtc	[31:0]数据地址	R1	READ_MULTIPLE_BLOCK	不断从卡传输数据块到主机，直到收到 STOP_TRANSMISSION 命令才中断。块长度规定和

命令索引	类型	参数	响应格式	简称	描述
					READ_SINGLE_BLOCK 命令是一样的。
CMD21	adtc	[31:0] 填充位	R1	SEND_TUNNING_BLOCK	e•MMC 卡，为优化 HS200 采样点发送 128 时钟的调谐模板（4 bit 总线宽度下 64 字节，8 bit 下 128 字节）
注意：传输的数据不能跨越物理块边界，除非 READ_BLK_MISALIGN 在 CSD 寄存器中被设置。					

表 36-14. 流读取命令（class 1）和流写入命令（class 3）

命令索引	类型	参数	响应格式	简称	描述
CMD11	adtc	[31:0]数据地址	R1	READ_DAT_UNTIL_STOP	从卡中读取数据流，起始于给定的地址，直至收到 STOP_TRANSMISSION 命令。
CMD20	adtc	[31:0]数据地址	R1	WRITE_DAT_UNTIL_STOP	从主机写数据流，起始于给定的地址，直至收到 STOP_TRANSMISSION 命令。
注意：传输的数据不能跨越物理块边界，除非 READ_BLK_MISALIGN 在 CSD 寄存器中被设置。					

表 36-15. 面向块的写命令（class 4）

命令索引	类型	参数	响应格式	简称	描述
CMD16	ac	[31:0] 块长度	R1	SET_BLOCKLEN	见 *表 36-13. 面向块的读命令（class 2）* 描述。
CMD23	ac	[31:16] 设为 0 [15:0] 块数目	R1	SET_BLOCK_COUNT	定义了将要在后续多个块的读或写命令被传输块的数目。如果参数为全 0，随后的读/写操作将被被认为无终止的。
CMD24	adtc	[31:0] 数据地址	R1	WRITE_BLOCK	在标准容量 SD 卡的情况下，该命令写入由 SET_BLOCKLEN 命令所选择的块长度。在高容量 SD 卡的情况下，块长度是固定的 512 字节忽略 SET_BLOCKLEN 命令。
CMD25	adtc	[31:0] 数据地址	R1	WRITE_MULTIPLE_BLOCK	连续写入数据块，直至收到 STOP_TRANSMISSION 命令。块长度是和 WRITE_BLOCK 命令规定一样的。
CMD26	adtc	[31:0] 填充位	R1	PROGRAM_CID	对卡识别寄存器进行编程。此命令必须一次发出。该编程涉及硬件，以防止首次编程以后的操作。通常情况下这个命令是针对厂家保留。
CMD27	adtc	[31:0] 填充位	R1	PROGRAM_CSD	对 CSD 的可编程位编程。

命令索引	类型	参数	响应格式	简称	描述
CMD49	adtc	[31:0] 填充位	R1	SET_TIME	根据 512 字节数据块中的 RTC 信息设置实时时钟
注意： 1. 传输的数据不得跨越物理块边界。除非是在 CSD 设置 WRITE_BLK_MISALIGN。在写入部分块不支持的情况下，块长度=默认块长度（CSD 中给出）。 2. 标准容量 SD 存储卡数据地址以字节为单位，高容量 SD 存储卡数据地址以块（512 字节）为单位。					

表 36-16. 擦除命令（class 5）

命令索引	类型	参数	响应格式	简称	描述
CMD32	ac	[31:0]数据地址	R1	ERASE_WR_BLK_START	设置要被擦除数据的第一个块的地址。（SD）
CMD33	ac	[31:0]数据地址	R1	ERASE_WR_BLK_END	设置要被擦除数据的最后一个块地址。（SD）
CMD35	ac	[31:0]数据地址	R1	ERASE_GROUP_START	在选择的擦除范围内，设置第一个擦除组的地址。（e•MMC）
CMD36	ac	[31:0]数据地址	R1	ERASE_GROUP_END	在选择的连续擦除范围内，设置最后一个擦除组的地址。（e•MMC）
CMD38	ac	[31:0]填充位	R1b	ERASE	擦除所有之前选择的数据块.
注意： 1. CMD34 和 CMD37 被保留，以便保持与旧版本 e•MMC 的兼容性 2. 标准容量 SD 存储卡数据地址以字节为单位，高容量 SD 存储卡数据地址以块（512 字节）为单位。					

表 36-17. 面向块的写保护命令（class 6）

命令索引	类型	参数	响应格式	简称	描述
CMD28	ac	[31:0] 数据地址	R1b	SET_WRITE_PROT	如果卡有写保护功能，该命令将设置地址组的写保护位。写保护的特性被编码在卡的特定数据（WP_GRP_SIZE）中。高容量 SD 存储卡不支持此命令。
CMD29	ac	[31:0] 数据地址	R1b	CLR_WRITE_PROT	如果卡有写保护功能，该命令将清除寻址组的写保护位。
CMD30	adtc	[31:0] 写保护数据地址	R1	SEND_WRITE_PROT	如果卡有写保护功能，该命令请求卡发送写保护位状态。
注意： 1. 高容量 SD 存储卡不支持这三个命令。					

表 36-18. 锁卡命令（class 7）

命令索引	类型	参数	响应格式	简称	描述
CMD16	ac	[31:0] 块长度	R1	SET_BLOCK_LEN	见 *表 36-13. 面向块的读命令（class 2）* 描述。
CMD42	adtc	[31:0] 保留位（所有位设为 0）	R1	LOCK_UNLOCK	用于设置/重置密码或者对卡上锁/解锁。数据块长度由命令 SET_BLOCK_LEN 设置。

					参数及锁卡数据结构里的保留位应设为 0。

表 36-19. 特定应用命令（class 8）

命令索引	类型	参数	响应格式	简称	描述
ACMD41	bcr	[31]保留位 [30]HCS [29:24]保留位 [23:0]V_{DD} 电压窗口（OCR[23:0]）	R3	SD_SEND_OP_COND	发送主机容量支持信息（HCS），并请求访问的卡在响应中发送操作条件寄存器（OCR）的内容。当卡接收到 SEND_IF_COND 命令，HCS 是有效的。CCS 位被分配到 OCR[30]。
ACMD42	ac	[31:1] 填充位 [0] set_cd	R1	SET_CLR_CARD_DETECT	在卡的 CD/DAT3 （引脚 1）上连接[1]/断开[0] 50K 上拉电阻。
ACMD51	adtc	[31:0] 填充位	R1	SEND_SCR	读 SD 卡配置寄存器（SCR）。
CMD55	ac	[31:16] RCA [15:0] 填充位	R1	APP_CMD	表明卡的下一个命令是特定应用命令而不是标准命令。
CMD56	adtc	[31:1] 填充位 [0] RD/WR	R1	GEN_CMD	对于通用/特定应用命令，该命令用于向卡传输一个数据块，或从卡读取一个数据块。主机设 RD/WR=1 时是从卡中读数据，RD/WR=0 时啊写数据到卡中。
注意： 1. ACMDx 是针对 SD 存储卡的特定应用命令。					

表 36-20. I/O 模式命令（class 9）

命令索引	类型	参数	响应格式	简称	描述
CMD39	ac	[31:16] RCA [15] 寄存器写标志 [14:8] 寄存器地址 [7:0] 寄存器数据	R4	FAST_IO	用于写入和读取 8 位（寄存器）的数据字段。如果写标志被设置，该命令寻址寄存器，并提供数据写入。如果写标志被清为 0，R4 的响应中包含从寻址寄存器中读取的数据。该命令用于访问未在 MMC 标准定义的应用程序相关的寄存器。
CMD40	bcr	[31:0] 填充位	R5	GO_IRQ_STATE	设置系统进入中断模式。
CMD52	adtc	[31] R/W 标志 [30:28] 功能数目 [27] RAW 标志 [26] 填充位 [25:9] 寄存器地址 [8] 填充位	R5	IO_RW_DIRECT	IO_RW_DIRECT 命令提供简单的方式访问任意 I/O 功能的 128K 存储空间的寄存器。此命令可以实现使用单个命令对寄存器的读写。一个常见的用途是初始化寄存器或查询 I/O 功能状

命令索引	类型	参数	响应格式	简称	描述
		[7:0] 写数据/填充位			态。这个命令是读或写单 I/O 寄存器最快的方法，因为它仅需要一对单一的命令/响应。
CMD53	adtc	[31] R/W 标志 [30:28] 功能数目 [27] 块模式 [26] OP 码 [25:9] 寄存器地址 [8:0] 字节/块数		IO_RW_EXTENDED	该命令允许用一个简单命令读取或写入大量的 I/ O 寄存器。
注意： 1.CMD39, CMD40 仅用于 e•MMC 卡 2. CMD52, CMD53 仅用于 SD I/O 卡					

表 36-21. 切换功能命令（class 10）

命令索引	类型	参数	响应格式	简称	描述
CMD53	adtc	[31:16] 安全协议详细 [15:8] 安全协议 [7:0] 保留	R1	PROTOCOL_RD	仅用于 SD 存储卡和 SD I/O 卡。从卡到主机连续不断的传输数据块，数据块的个数由 CMD23 提供，数据传输可以被 STOP_TRANSMISSION 中断，该命令不支持包命令，数据块的大小固定为 512 字节。
CMD54	adtc	[31:16] 安全协议详细 [15:8] 安全协议 [7:0] 保留	R1	PROTOCOL_WR	仅用于 SD 存储卡和 SD I/O 卡。从主机到卡连续不断的传输数据块，数据块的个数由 CMD23 提供，数据传输可以被 STOP_TRANSMISSION 中断，该命令不支持打包命令，数据块的大小固定为 512 字节。

36.4.3. 响应

所有的响应都是通过 CMD 信号线发送，方向是由卡到主机。响应传输总是从对应响应字串的最左位开始。响应字串的长度依赖于响应类型。

响应类型

响应的类型有七种，分别如下：

- **R1 / R1b：**普通命令响应
- **R2：**CID，CSD 寄存器

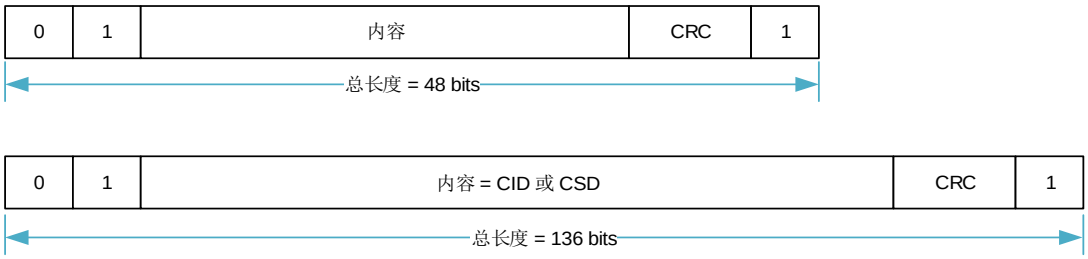
- **R3**：OCR 寄存器
- **R4**：Fast I/O
- **R5**：中断请求
- **R6**：发布的 RCA 响应
- **R7**：卡接口条件

SD 存储卡支持其中的五种响应，R1 / R1b, R2, R3, R6, R7。SD I/O 卡和 *e*•MMC 卡支持支持额外的响应类型，名为 R4 和 R5，但对于 SD I/O 卡和 *e*•MMC 卡，这两种响应并不完全相同。

响应格式

响应有两种格式，如*图 36-10. 响应令牌格式*所示，所有响应经由 CMD 线发出。代码的长度取决于响应类型。除了 R2 的长度是 136 位，其他的长度均为 48 位。

图 36-10. 响应令牌格式



响应总是从一个起始位（始终为 0）开始，随后第二位表示传输的方向（卡= 0）。下面表中的“x”的值表示为可变的部分。除了 R3 类型的所有响应由 CRC 校验。每个响应字段由结束位（总是 1）终止。

R1（普通命令响应）

代码长度为 48 位。位 45:40 指示要响应的命令索引，该值被解释为一个二进制编码的数字（0 到 63 之间）。卡的状态被 32 位编码。注意，如果写数据到卡上，在每个数据块传输之后会出现 BUSY 信号，在每个数据块传输完成后主机需要检查 BUSY 信号。卡状态在章节*卡的两种状态*中描述。

表 36-22. R1 响应

位	47	46	[45:40]	[39:8]	[7:1]	0
位宽	1	1	6	32	7	1
数值	‘0’	‘0’	x	x	x	‘1’
描述	起始位	传输位	命令索引	卡状态	CRC7	结束位

R1b

R1b 格式与 R1 相同，但可以在数据线 DAT0 上发送忙信号。收到命令后，依据收到命令之前的状态，卡可能变为忙状态。主机应在响应中检查忙状态。

R2（CID, CSD 寄存器）

代码长度为 136 位。CID 寄存器的内容作为对命令 CMD2 和 CMD10 的响应被发送。CSD 寄存器的内容将作为以 CMD9 响应被发送。卡只响应发送 CID 和 CSD 的位[127.. 1]，这两个寄存器保留位[0]被替换为响应的结束位。

表 36-23. R2 响应

位	135	134	[133:128]	[127:1]	0
位宽	1	1	6	127	1
数值	'0'	'0'	'111111'	x	'1'
描述	起始位	传输位	保留	CID 或 CSD 寄存器，内部 CRC7	结束位

R3（OCR 寄存器）

代码长度为 48 位。该 OCR 寄存器的内容作为 ACMD41（SD 存储卡），CMD1（*e*•MMC）的响应被发送。不同卡的响应可能有一点不同。

表 36-24. R3 响应

位	47	46	[45:40]	[39:8]	[7:1]	0
位宽	1	1	6	32	7	1
数值	'0'	'0'	'111111'	x	'1111111'	'1'
描述	起始位	传输位	保留	OCR 寄存器	保留	结束位

R4（Fast I/O）

仅适用于 *e*•MMC 卡。代码长度为 48 位。参数域包括选定卡的 RCA，被读取或写入寄存器的地址，和它的内容。如果操作成功，参数域状态位置位。

表 36-25. R4 响应（e•MMC）

位	47	46	[45:40]	[39:8] 参数域				[7:1]	0
位宽	1	1	6	16	1	7	8	7	1
数值	'0'	'0'	'100111'	x	x	x	x	x	'1'
描述	起始位	传输位	CMD39	RCA [31:16]	状态 [15]	寄存器地址 [14:8]	读寄存器的内容 [7:0]	CRC7	结束位

R4b

仅适用于 SD I/O 卡。代码长度为 48 位。SD I/O 卡接收到 CMD5 命令后会返回一个唯一的 SD I/O 卡响应 R4。

表 36-26. R4 响应（SD I/O）

位	47	46	[45:40]	39	[38:36]	35	[34:32]	31	[30:8]	[7:1]	0
位宽	1	1	6	1	3	1	3	1	23	7	1
数值	'0'	'0'	'111111'	x	x	x	'000'	x	x	'1111111'	1
描述	起始位	传输位	保留	C	I/O 功能数目	存储功能存在	填充位	S18A	I/O OCR	保留	结束位

R5（中断请求）

仅适用于 *e*•MMC 卡。代码长度为 48 位。若这个响应由主机产生，参数中 RCA 域为 0x0。

表 36-27. R5 响应（MMC）

位	47	46	[45:40]	[39:8] 参数域		[7:1]	0
位宽	1	1	6	16	16	7	1
数值	‘0’	‘0’	‘101000’	x	x	x	‘1’
描述	起始位	传输位	CMD40	成功的卡或主机的 RCA [31:16]	[15:0]未定义，可能作为中断数据	CRC7	结束位

R5b

仅适用于 SD I/O 卡。SD I/O 卡对于 CMD52 和 CMD53 命令的响应是 R5。如果卡和主机之间的通信是在 1 位或 4 位 SD 模式下，响应应是 48 位响应（R5）。

表 36-28. R5 响应（SD I/O）

位	47	46	[45:40]	[39:24]	[23:16]	[15:8]	[7:1]	0
位宽	1	1	6	16	8	8	7	1
数值	‘0’	‘0’	‘11010x’	‘0’	x	x	x	‘1’
描述	起始位	传输位	CMD52/53	填充位	响应标志	读或写的数据	CRC7	结束位

R6（发布的 RCA 响应）

代码长度为 48 位。位[45:40]表示对 CMD3 响应的命令索引。参数字段的 16 个最高位比特用于已发布的 RCA 号。

表 36-29. R6 响应

位	47	46	[45:40]	[39:8] 参数域		[7:1]	0
位宽	1	1	6	16	16	7	1
数值	‘0’	‘0’	‘000011’	x	x	x	‘1’
描述	起始位	传输位	CMD3	新发布卡的 RCA	卡的状态位：23,22,19,12:0	CRC7	结束位

R7（卡接口条件）

仅适用于 SD 存储卡。代码长度为 48 位。卡支持电压信息由 CMD8 的响应发送。位[19:16]表明该卡支持的电压范围。接受了供电电压的卡返回 R7 响应。在响应中，卡回送的参数设置电压范围和检查模式。

表 36-30. R7 响应

位	47	46	[45:40]	[39:20]	[19:16]	[15:8]	[7:1]	0
位宽	1	1	6	20	4	8	7	1
数值	‘0’	‘0’	‘001000’	‘00000h’	x	x	x	‘1’
描述	起始位	传输位	CMD8	保留位	可接受电压	检查模式字节	CRC7	结束位

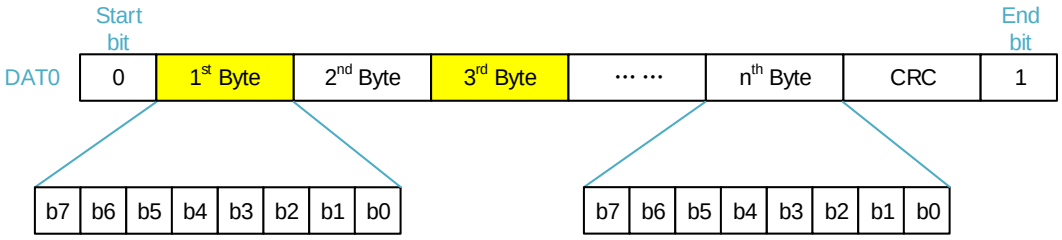
36.4.4. 数据包格式

数据总线模式有 5 种，1 位、4 位 SDR/DDR 和 8 位 SDR/DDR 总线模式。1 位模式是强制的，4 位和 8 位总线模式是可选的。

1 位数据包格式

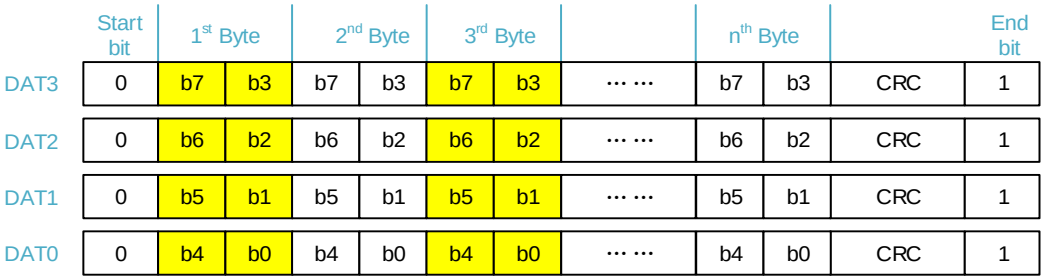
卡复位和初始化之后，只有 DAT0 被用于传输数据。其他引脚可以用于其他用处。*图 36-11. 1 位数据总线宽度*，*图 36-12. 4 位数据总线宽度*和*图 36-13. 8 位数据总线宽度*显示了数据宽度是 1 位，4 位和 8 位时的数据包格式。

图 36-11. 1 位数据总线宽度



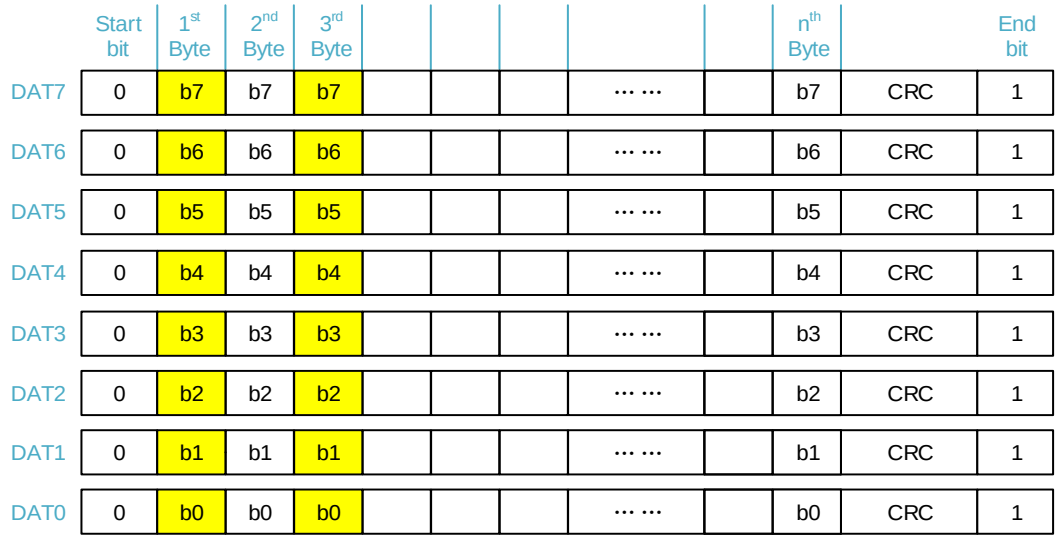
4 位数据包格式

图 36-12. 4 位数据总线宽度



8 位数据包格式

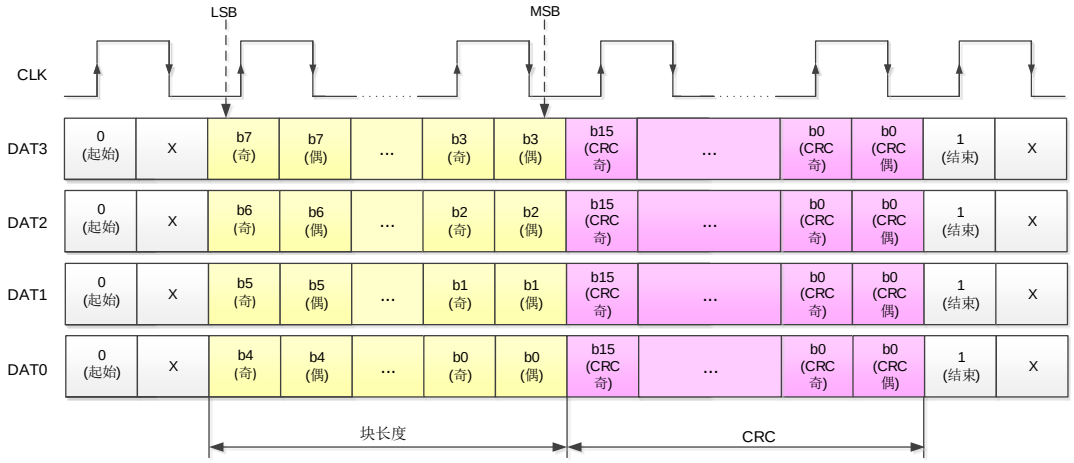
图 36-13. 8 位数据总线宽度



4 位 DDR 数据包格式（DAT3-DAT0）

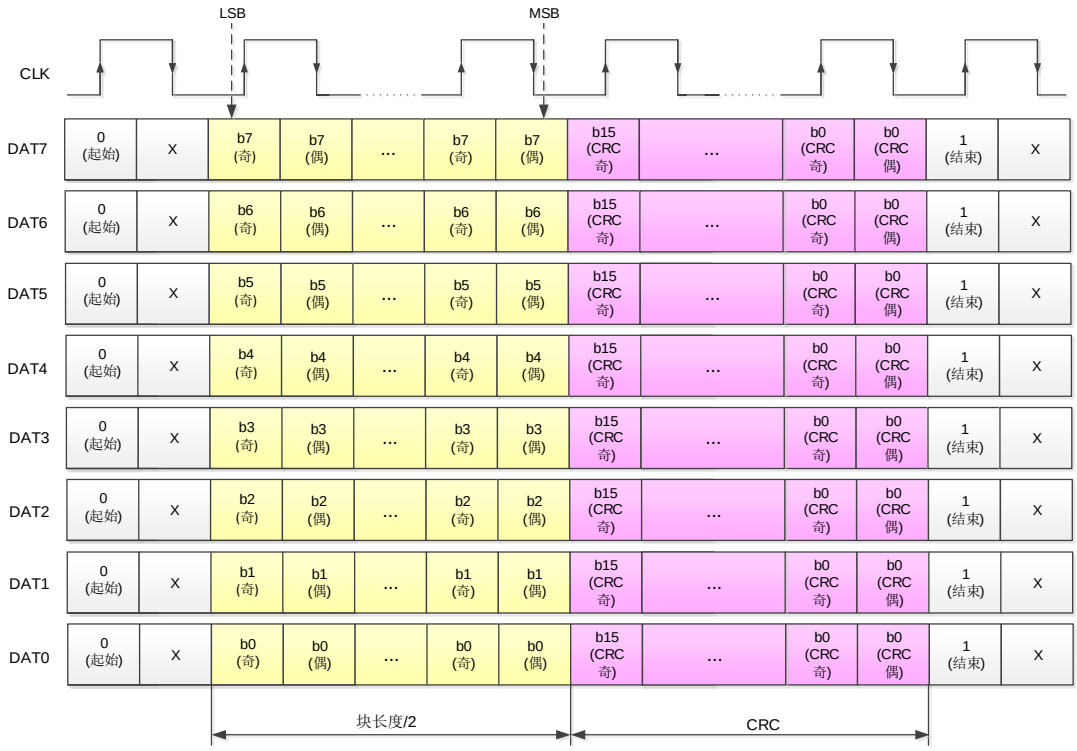
对于每条数据线，数据可以以每个时钟周期一位（单数据速率）或两位（双数据速率）的速率传输。DDR（双倍数据速率）信号，数据在两个 SDIO_CLK 时钟沿采样。*图 36-14. 4 位 DDR 数据包格式*和*图 36-15. 8 位 DDR 数据包格式*显示了数据总线宽度为 4 位 DDR 和 8 位 DDR 时的数据包格式。

图 36-14. 4 位 DDR 数据包格式



8 位 DDR 数据包格式（DAT7-DAT0）

图 36-15. 8 位 DDR 数据包格式



注意：对于 DDR 数据总线：1.字节数据不交错，但 CRC 是交错的。2.起始位在上升沿和下

降沿均有效。3.结束位仅在上升沿有效（“x”未定义）。

36.4.5. 卡的两种状态

SD 存储卡支持两种状态字段，而其他的卡只支持第一种：

卡状态：执行命令的错误和状态信息，在响应中指示。
SD 状态：512 位的扩展状态信息，支持特定功能的 SD 存储卡和未来应用特定功能。

卡状态

响应格式 R1 包含一个名为卡状态的 32 位字段。该字段用来传送该卡的状态的信息（可以存储在本地状态寄存器）到主机。除非特别说明，卡的状态信息总是与之前发出的命令相关。

表中的类型和清除条件的缩写如下：

类型

•E：错误位。向主机发送错误条件。这些位一旦响应（报告错误）被发出去就会清除。
•S：状态位。这些位仅作为信息字段，并不因为对命令的响应而改变。这些位是持久性的，它们根据卡状态被设置或被清除。
•R：卡在命令解释和验证阶段（响应模式）检测到异常。
•X：卡在命令执行阶段（执行模式）检测到异常。

清除条件

•A：根据卡当前状态。
•B：始终与之前命令相关。接收到有效命令可清除该状态（有命令延迟）。
•C：读可清除。

表 36-31. 卡状态

位	标识符	类型	数值	说明	清除条件
31	OUT_OF_RANGE	ERX	’0’= 无错误 ’1’= 错误	命令的参数超出卡的允许范围。	C
30	ADDRESS_ERROR	ERX	’0’= 无错误 ’1’= 错误	在命令中使用与块长度不匹配的未对齐地址。	C
29	BLOCK_LEN_ERROR	ERX	’0’= 无错误 ’1’= 错误	所传输的块长度是卡不允许的，或者传输的字节数不匹配块的长度。	C
28	ERASE_SEQ_ERROR	ER	’0’= 无错误 ’1’= 错误	擦除命令顺序发生错误。	C
27	ERASE_PARAM	ERX	’0’= 无错误 ’1’= 错误	擦除时选择了无效的擦除块。	C
26	WP_VIOLATION	ERX	’0’= 未保护 ’1’= 已保护	当主机试图写一个受保护的块或暂时或永久写保护卡时置位。	C
25	CARD_IS_LOCKED	SX	‘0’ = 卡未锁 ‘1’ = 卡已锁	当设置该位，表示卡已经被主机锁住。	A
24	LOCK_UNLOCK_FAIL	ERX	’0’= 无错误	在上锁/解锁中有命令的顺序错	C

位	标识符	类型	数值	说明	清除条件
	ED		’1’= 错误	误或检测到密码错误时置位。	
23	COM_CRC_ERROR	ER	’0’= 无错误 ’1’= 错误	之前命令的 CRC 校验错误。	B
22	ILLEGAL_COMMAND	ER	’0’= 无错误 ’1’= 错误	对于当前状态，命令非法。	B
21	CARD_ECC_FAILED	ERX	’0’= 成功 ’1’= 失败	卡的内部实施了 ECC 校验，但在更正数据时失败。	C
20	CC_ERROR	ERX	’0’= 无错误 ’1’= 错误	卡内部控制器错误。	C
19	ERROR	ERX	’0’= 无错误 ’1’= 错误	在操作过程中发生一般的或者未知的错误。	C
18	UNDERRUN	ERX	’0’= 无错误 ’1’= 错误	仅针对 e•MMC。该卡不支持在流读取模式下的数据传输。	C
17	OVERRUN	ERX	’0’= 无错误 ’1’= 错误	仅针对 e•MMC. 该卡不支持在流写入模式下的数据编程。	C
16	CID/ CSD_OVERWRITE	ERX	’0’= 无错误 ’1’= 错误	可能是下面两种错误之一： - CSD 的只读部分与卡内容不匹配 - 试图进行拷贝或永久写保护的反向操作，即恢复原状或解除写保护	C
15	WP_ERASE_SKIP	ERX	’0’= 未保护 ’1’= 已保护	若置位，因为存在写保护数据块仅有部分地址空间被擦除；被暂时或者永久写保护的卡被擦除。	C
14	CARD_ECC_DISABLED	SX	’0’= 使能 ’1’= 失能	执行命令时未使用内部 ECC。	A
13	ERASE_RESET	SR	’0’= 清除 ’1’= 设置	因为收到一个擦除顺序之外的命令，擦除序列在执行前被清除。	C
[12:9]	CURRENT_STATE	SX	0 = 空闲 1 = 就绪 2 = 识别 3 = 待机 4 = 传输 5 = 发送数据 6 = 接收数据 7 = 编程 8 = 断开 9 = 总线测试 10 = 睡眠 11-14 = 保留 15 = 保留（I/O 模式）	当收到命令时卡的状态。如果命令的执行导致状态的变化，这个变化将会在下个命令的响应中反映出来。这四个位按 0 到 15 的二进制编码数解释。 睡眠状态只在 e•MMC 卡中。	B

位	标识符	类型	数值	说明	清除条件
8	READY_FOR_DATA	SX	'0'= 未就绪 '1'= 就绪	与总线上的缓冲器空的信号一致。	A
7	SWITCH_ERROR	EX	'0'= 无错误 '1'= 切换错误	如果置位，卡没有通过 SWITCH 命令切换到期望的模式。	B
6	保留				
5	APP_CMD	SR	'0'= 使能 '1'= 失能	卡期望 ACMD，或指示命令已经被解释为 ACMD 命令。	C
4	保留				
3	AKE_SEQ_ERROR	ER	'0'= 无错误 '1'= 错误	仅针对 SD 存储卡。验证过程的顺序有错误。	C
2	保留给与应用特定命令。				
[1:0]	保留给厂商测试模式。				

注意： 18, 17, 7 位仅适用于 e•MMC。14, 3 位仅适用于 SD 存储卡。

SD 状态寄存器

在 SD 状态寄存器中含有与 SD 存储卡的专有特征相关的状态位，并且可以被用于未来的特定应用使用。SD 状态寄存器是一个 512 比特大小的数据块。该寄存器的内容连同一个 16 位 CRC 通过 DAT 总线被发送到主机上。SD 状态通过 DAT 总线被发送到主机上，作为 ACMD13 的响应（CMD55 接着用 CMD13）。ACMD13 只能在“传送状态”被发送到存储卡（卡被选中）。SD 状态结构将在下面描述。

“类型”和“清除条件”的缩写与上述卡状态描述相同。

表 36-32. SD 状态

位	标识符	类型	数值	描述	清除条件
[511:510]	DAT_BUS_WIDTH	SR	'00'= 1 （默认） '01'= 保留 '10'= 4 位宽 '11'= 保留	由 SET_BUS_WIDTH 命令显示当前定义的数据总线宽度	A
509	SECURED_MODE	SR	'0'= 未处于安全模式 '1'= 处于安全模式	卡处于操作的安全模式（参考“SD 安全规范”）。	A
[508:496]	保留				
[495:480]	SD_CARD_TYPE	SR	下列卡目前被定义为： '0000'= 通用 SD 读/写卡 '0001'= SD ROM 卡 '0002'= OTP	低 8 位在未来被用来定义 SD 存储卡的不同变种（每个位将定义不同的 SD 卡类型）。高 8 位将被用来定义不符合当前 SD 物理层规范的 SD 卡。	A
[479:	SIZE_OF_PROTECT	SR	受保护区域的大	（见下面描述）	A

位	标识符	类型	数值	描述	清除条件
448]	ED_AREA		小。		
[447:440]	SPEED_CLASS	SR	卡的速度类型。	（见下面描述）	A
[439:432]	PERFORMANCE_MOVE	SR	以 1MB/s 为单位的传输性能。	（见下面描述）	A
[431:428]	AU_SIZE	SR	AU 大小	（见下面描述）	A
[427:424]	保留				
[423:408]	ERASE_SIZE	SR	一次要被擦除的 AU 数目。	（见下面描述）	A
[407:402]	ERASE_TIMEOUT	SR	UNIT_OF_ERASE_AU 指定的擦除区域的超时时间。	（见下面描述）	A
[401:400]	ERASE_OFFSET	SR	擦除时间增加固定偏移值。	（见下面描述）	A
[399:312]	保留				
[311:0]	保留给生产厂商				

SIZE_OF_PROTECTED_AREA

对于标准容量卡（SDSC）和高容量卡（SDHC/SDXC）设置该位域不同。

对于标准容量卡（SDSC），受保护区域容量计算方式如下：
受保护区域 = SIZE_OF_PROTECTED_AREA* MULT * BLOCK_LEN。
SIZE_OF_PROTECTED_AREA 以 MULT*BLOCK_LEN 为单位。

对于高容量卡（SDHC/SDXC），受保护区域容量计算方式如下：
受保护区域 = SIZE_OF_PROTECTED_AREA 。
SIZE_OF_PROTECTED_AREA 以字节为单位。

SPEED_CLASS

这 8 位字段表示速度等级。

00h: Class 0
01h: Class 2
02h: Class 4
03h: Class 6
04h: Class 10
05h–FFh: 保留

PERFORMANCE_MOVE

这 8 位域指示 Pm，该值可被设为以 1MB/秒为单位。如果卡不用 RU 移动数据，应该认为 Pm

是无穷大。设置这个域为 FFh 表示无穷大。Pm 的最小值由*表 36-33. 移动性能字段*中定义。

表 36-33. 移动性能字段

PERFORMANCE_MOVE	数值定义
00h	顺序写入
01h	1 [MB/sec]
02h	2 [MB/sec]
.....	.....
FEh	254 [MB/sec]
FFh	无穷大

AU_SIZE

这 4 位字段指示 AU 大小，数值是 16K 字节为单位 2 的幂次的倍数。

表 36-34. AU_SIZE 字段

AU_SIZE	数值定义
0h	未定义
1h	16 KB
2h	32 KB
3h	64 KB
4h	128 KB
5h	256 KB
6h	512 KB
7h	1 MB
8h	2 MB
9h	4 MB
Ah	8 MB
Bh	12 MB
Ch	16 MB
Dh	24 MB
Eh	32 MB
Fh	64 MB

最大 AU 大小，取决于卡的容量，由*表 36-34. AU_SIZE 字段*中定义。卡可以任意的设置 AU 大小（由*表 36-35. 最大 AU 大小*定义），只要小于或等于该卡容量所允许的最大 AU 大小。卡应该尽可能小地设置 AU 尺寸。

表 36-35. 最大 AU 大小

卡容量	最大 64MB	最大 256MB	最大 512MB	最大 32GB	最大 2TB
最大 AU 大小	512 KB	1 MB	2 MB	4 MB	64MB

ERASE_SIZE

这 16 位字段表示 N_{ERASE}。当 N_{ERASE} 个数的 AU 被擦除，超时时间由 ERASE_TIMEOUT 规定（参考 ERASE_TIMEOUT）。主机应确定在一次操作中要被擦除的 AU 的适当数目，以便主机可以预示擦除操作的进度。如果该字段设置为 0，则不支持擦除的超时计算。

表 36-36. 擦除大小字段

ERASE_SIZE	数值定义
0000h	不支持擦除的超时计算。
0001h	1 AU
0002h	2 AU
0003h	3 AU
.....	.....
FFFFh	65535 AU

ERASE_TIMEOUT

这 6 位字段表示 T_{ERASE}，当 ERASE_SIZE 指示的多个 AU 被擦除时，这个数值给出了从偏移量算起的擦除超时时间。ERASE_TIMEOUT 的范围可以被定义为最多 63 秒，卡的制造商可以根据具体实现选择 ERASE_SIZE 和 ERASE_TIMEOUT 的任意组合。一旦 ERASE_TIMEOUT 被确定下来，那么 ERASE_SIZE 也确定了。主机可以通过以下公式计算任意数目的 AU 的擦除超时时间：

$$\text{Erase timeout of X AU} = \frac{T_{ERASE}}{N_{ERASE}} * X + T_{OFFSET} \quad \text{（式 36-1）}$$

表 36-37. 擦除超时字段

ERASE_TIMEOUT	数值定义
00	不支持擦除的超时计算
01	1 秒
02	2 秒
03	3 秒
.....	.....
63	63 秒

如果 ERASE_SIZE 字段被设置为 0，则该字段应该设置为 0。

ERASE_OFFSET

这 2 位字段表示 T_{OFFSET}，可以选择如*表 36-38. 擦除偏移字段*所示的四个数值之一。若 ERASE_SIZE 和 ERASE_TIMEOUT 字段都设为 0，该字段无意义。

表 36-38. 擦除偏移字段

ERASE_OFFSET	数值定义
0h	0 秒
1h	1 秒
2h	2 秒
3h	3 秒

36.5. 编程序列

36.5.1. 卡识别

主机复位后进入卡识别模式，寻找总线上的新卡。在卡识别模式下，主机复位所有的卡，验证工作电压范围，识别卡并询问每个卡的相对卡地址（RCA）。这个操作是在每个卡自己的命令信号线 CMD 上分别完成的。在卡识别模式中的所有数据通信只使用命令信号线（CMD）。

在卡识别过程中，卡应该工作在时钟频率为 F_{OD}（400 kHz）的情况下。

卡复位

命令 GO_IDLE_STATE（CMD0）是软件复位命令，并设置 e•MMC 和 SD 存储卡进入空闲状态（Idle State），不管当前卡的状态是什么。复位命令（CMD0）仅用于存储器或组合卡的存储器部分。为了重置只有 I/O 卡或组合卡的 I/O 部分，使用 CMD52 写 1 到 CCCR 的 RES 位。在非激活状态（Inactive State）的卡不受此命令的影响。

主机上电后，所有的卡都处于空闲状态（Idle State），包括之前已在非激活状态（Inactive State）的卡。上电或 CMD0 后，所有卡的 CMD 线处于输入模式，等待下一个命令的起始位。这些卡都是用缺省的相对卡地址（RCA）初始化，并用默认 400 kHz 的时钟频率驱动器。

工作电压范围验证

在主机和卡之间开始通信时，主机可能不知道卡支持的电压，并且卡可能不知道主机能否提供其支持的电压。为了验证电压，下面的命令都在相关规范中定义。

在协议规范中定义的命令包括：SEND_OP_COND（CMD1 用于 e•MMC），SD_SEND_OP_COND（ACMD41 用于 SD 存储卡），IO_SEND_OP_COND（CMD5 用于 SD I/O 卡），这些命令提供给主机一种机制去识别和拒绝那些不匹配主机所需的 V_{DD} 范围的卡。这是由主机发送所需的 V_{DD} 电压窗口作为此命令的操作数来实现的。如果卡不能在指定的范围内进行数据传输，必须从总线断开并进入非激活状态（Inactive State）。否则，该卡将响应返回它的 V_{DD} 范围。

如果该卡可以工作在所提供的电压下，响应将返回供电电压和在命令参数中设置的检查模式。

如果该卡不能在提供的电压下工作，它不返回响应，并保持在空闲状态。初始化 SDHC 卡时强制性的在 ACMD41 命令之前发送 CMD8。收到 CMD8 是让该卡知道主机支持物理层 2.00 协议及卡支持高版本的功能。

卡识别过程

对于不同的卡，卡的识别过程不同。这些卡包括 e•MMC、SD，或 SD I/O 卡。支持所有类型的 SD I/O 卡，即 SDIO_IO_ONLY 卡、SDIO_MEM_ONLY 卡和 SDIO COMBO 卡。卡识别过程步骤如下：

1. 检测卡是否连接。
2. 识别卡的类型：SD 卡、e•MMC 或 SD I/O 卡。

- 发送 CMD5 命令。如果主机接收到响应，则是 SD I/O 卡；
- 如果没有响应，发送 ACMD41。如果主机接收到响应，则是 SD 卡；
- 否则，是 e•MMC 卡。

3. 根据卡的类型初始化卡。

使用 F_{OD} （400 KHz）为时钟源，并按照下列命令顺序发送命令：

- SD 卡 - 发送 CMD0，ACMD41，CMD2，CMD3；
- SDHC 卡 - 发送 CMD0，CMD8，ACMD41，CMD2，CMD3；
- SD I/O 卡 - 如果卡没有存储器端口，发送 CMD52，CMD0，CMD5，CMD3；否则，发送 CMD52，CMD0，CMD5，ACMD41，CMD11 （可选），CMD2，CMD3；
- e•MMC - 发送 CMD0，CMD1，CMD2，CMD3。

36.5.2. 引导操作

如果在上电或复位后（硬件复位或通过参数 0xF0F0F0F0 的 CMD0）发送第一个命令之前将 CMD 线拉低 74 时钟周期以上（正常引导模式）或发送了参数为 0xFFFFFFFA 的 CMD0（备用引导模式），卡则识别出启动了引导模式，并开始内部准备引导数据。

主机通过 EXT_CSD 寄存器[179]字节的位[5:3]来选择从哪个分区读取引导数据，主机在引导操作期间可读取的数据长度可按 128KB x BOOT_SIZE_MULT（EXT_CSD 寄存器的[226]字节）计算。

主机通过 EXT_CSD 寄存器[177]字节的位[4:3]设置合适的值，来选择向后兼容接口时序的单倍数据率模式或选择高速接口时序的 SDR 或 DDR（如果支持）。EXT_CSD 寄存器[228]字节的位[2]告诉主机在引导期间设备是否支持高速时序。EXT_CSD 寄存器[228]字节的位[1]告诉主机在引导期间设备是否支持双倍数据率模式。

引导操作期间不支持 HS200 模式。

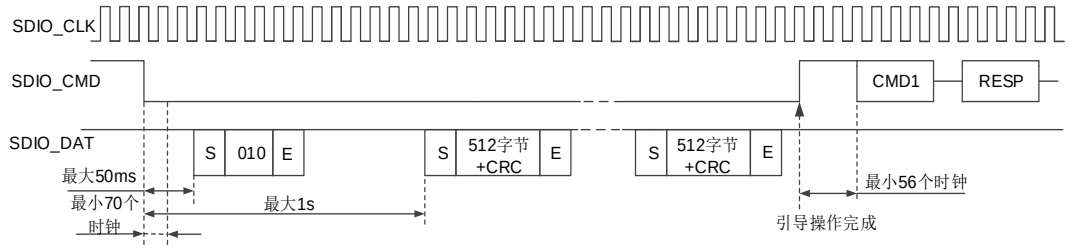
主机可以通过在 EXT_CSD 寄存器[179]字节的位[6]设置为“1”来选择从卡接收引导确认，这样主机就可以识别卡工作在引导模式下。

如果引导确认被使能，卡必须在 CMD 线被拉低后 50ms 之内向主机发送确认模式“010”。如果引导确认被禁用，卡不发送确认模式“010”。

正常引导操作

如果在上电后发送第一个命令之前保持 CMD 线低电平至少 74 个时钟周期，卡将识别引导模式正在启动。SDIO_CMD 线拉低 1 秒之内，卡开始在 SDIO_DAT 线上向主机发送第一个引导数据。主机必须保持 CMD 线为低电平以读取所有引导数据，主机必须采用推挽模式直至引导操作结束。期间主机可以通过拉高 SDIO_CMD 线以终止引导模式。

图 36-16. 引导操作时序



执行正常的引导过程，要按照以下步骤操作：

1. 复位卡或给卡上电。

2. 如果使能引导确认（ACK），需要使能 ACKEN，设置 ACKTIME，并使能 ACKFAIL 和 ACKTO 标志。

3. 通过置位 DATADIR 来使能接收数据，并设置接收的引导数据的字节数 DATALEN。使能 DTTMOUT、DTEND 和 CMDSEND 中断，用于完成引导命令确认。

4. 通过设置 BOOTMOD = 0 选择正常引导操作模式，并通过 BOOTMODEN 使能启动。ACK 引导确认超时被启用，且 CMD 线保持低电平。

5. ACKTO 或 ACKFAIL 标志可用于检测是否收到引导确认。
- 如果没有及时收到引导确认时，将出现 ACKTO 标志。
- 如果没有收到正确的引导确认时，将出现 ACKFAIL 标志。

6. 当接收到所有引导数据时，将出现 DTEND 标志。
- 当数据 CRC 失败时，会生成 DTCRCERR 标志。
- 当发生接收数据超时时，会生成 DTTMOUT 标志。

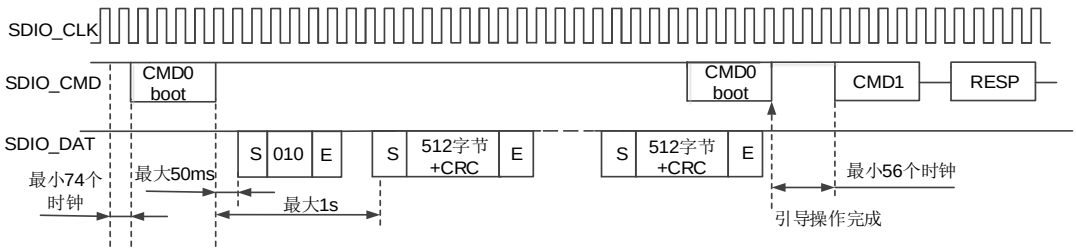
7. 当接收到最后一个数据时，从 FIFO 读取数据直到 FIFO 为空（RFE = 1），之后将生成数据结束标志（DTEND）。

8. 通过清零 BOOTMODEN 位来终止引导过程，这会导致 SDIO_CMD 线变为高电平。在 56 个时钟之后将生成 CMDSEND 标志，用于指示引导过程结束，卡已准备好接收新命令。

备用引导操作

卡在上电或复位后，如果主机在 74 时钟周期之后，发送 CMD1 之前，发送了参数为 0xFFFFFFFA 的 CMD0，则卡识别出备用的引导模式被启动，并内部开始准备引导数据。在主机发送带参数 0xFFFFFFFA 的 CMD0 后的 1 秒内，卡开始在 SDIO_DAT 线上发送第一个引导数据。主机可以通过发送 CMD0（复位）来终止引导操作。

图 36-17. 备用的引导操作时序



执行备用的引导操作，要按照以下步骤操作：

1. 复位卡或给卡上电。

2. 如果要使能引导 ACK，需要通过使能 ACKEN，设置 ACKTIME，并使能 ACKFAIL 和 ACKTO 标志。

3. 通过置位 DATADIR 来使能接收数据，并设置接收的引导数据的字节数 DATALEN。使能 DTTMOUT 和 DTEND 标志。

4. 通过设置 BOOTMOD = 1 来选择备用的引导操作模式，设置命令寄存器装载参数为 0xFFFFFFFA 的 CMD0。使能 CMDSEND 标志以指示完成引导命令确认，通过 BOOTMODEN 使能引导，并且设置 CSMEN 为 1。ACK 引导确认超时被启用，且 CMD 线保持低电平。当命令被发送时，将生成 CMDSEND 标志位，此时 BOOTMODEN 位应清零。

5. ACKTO 或 ACKFAIL 标志可用于检测是否收到引导确认。
- 如果没有及时收到引导确认时，将出现 ACKTO 标志。
- 如果没有收到正确的引导确认时，将出现 ACKFAIL 标志。

6. 当接收到所有引导数据时，将出现 DTEND 标志。
- 当数据 CRC 失败时，会生成 DTCRCERR 标志。
- 当接收数据发生超时时，会生成 DTTMOUT 标志。

7. 当接收到最后一个数据时，从 FIFO 读取数据直到 FIFO 为空（RFE=1），之后将生成数据结束 DTEND 标志。

8. 通过发送 CMD0 来终止备用的引导操作之前需要清零 BOOTMODEN 位，这会导致 56 个时钟之后生成 CMDSEND 标志。该标志用于指示引导过程结束，卡已准备好接收新命令。当 CMD0（复位）被成功发送，BOOTMOD 位必须被清除才能终止备用的引导操作。

36.5.3. 无数据命令

发送任何无数据命令时，软件需要用适当的参数设置 SDIO_CMDCTL 寄存器和 SDIO_CMDAGMT 寄存器。通过这两个寄存器，主机形成命令，并将其发送到命令总线上。主机通过 SDIO_STAT 寄存器的错误标志来反映命令响应的错误。

当接收到响应时，主机设置 SDIO_STAT 寄存器 CMDRECV（CRC 校验通过）位或 CCRCERR（CRC 校验失败）位为 1。短响应被复制到 SDIO_RESP0，而长响应被复制到所有四个响应寄存器。SDIO_RESP3 寄存器的第 31 位代表的长响应的最高位，而 SDIO_RESP0 寄存器的第 0 位表示长响应最低位。

36.5.4. 单个数据块或多个数据块写

在发送块写入命令（CMD24 - CMD27）时，一个或多个数据块从主机传到卡。数据块由起始位（1 位或 4 位低电平），数据块，CRC 和结束位（1 位或 4 位高电平）组成。如果 CRC 失败，则卡通过 SDIO_DAT 线指示传输失败，传送数据被丢弃而不写入，并且后续发送的数据块将被忽略。

如果主机传输的部分数据累积长度不是数据块对齐，并且块错位是不允许的（未设置 CSD 参

数 WRITE_BLK_MISALIGN），卡将在第一个未对齐块的开始之前检测块错位错误（设置状态寄存器的 ADDRESS_ERROR 错误位），并同时忽略后续的数据传输。如果主机试图写一个写保护区的数据，写操作也将被终止。在这种情况下，卡将设置状态寄存器中 WP_VIOLATION 位。

设置 CID 和 CSD 寄存器不需要先设置块长度，传送的数据也通过 CRC 保护。如果 CSD 或 CID 寄存器的一部分被存储在 ROM 中，那么不可改变部分必须与接收缓冲区的对应部分相匹配。如果匹配失败，卡将报告一个错误同时不改变任何寄存器的内容。

一些卡可能需要很长的或者不可预测的时间写入一个数据块。接收一个数据块并完成 CRC 校验后，卡将开始写操作，如果写缓冲区已满则保持 DAT0 线拉低，并且无法通过新的命令 WRITE_BLOCK 接收新的数据。主机可以在任何时间用 SEND_STATUS 命令（CMD13）查询卡的状态，并且卡将返回当前状态。状态位 READY_FOR_DATA 表示卡是否可以接受新的数据或写入操作是否仍在进行中。主机可以通过发出 CMD7 命令不选中该卡（选择另外的卡），将该卡置于断开状态（Disconnect State），并释放 DAT 信号线而不中断写操作。当重新选择卡，如果写操作仍在进行中并且写缓冲区不可用，它会拉低 DAT 信号线重新激活忙指示。

对于 SD 卡。设置一些块被预擦除（ACMD23）操作将使多块写操作比没有 ACMD23 操作更快。主机将使用此命令来定义下一次操作将会有多少个数据块被发送。

单块或多块写操作步骤为：

1. 在 SDIO_DATALEN 寄存器中设置数据大小（以字节为单位）。
2. 在 SDIO_DATACTL 寄存器中设置数据块大小（BLKSZ，以字节为单位）；主机每次发送 BLKSZ 大小的数据块。
3. 在 SDIO_CMDAGMT 寄存器中设置数据应该被写入的地址。
4. 设置 SDIO_CMDCTL 寄存器。对于 SD 存储卡和 e•MMC 卡，使用 CMD24 命令为单块写和 CMD25 命令为多块写。对于 SD I/O 卡，使用 CMD53 命令来进行单块和多块传输。
5. 将数据写入 SDIO_FIFO。
6. 软件应查询数据错误中断。如果需要，软件可以通过发送停止命令（CMD12）终止数据传输。
7. 当收到 DTEND 中断时，数据传送结束。对于开放式的块传输，如果字节计数为 0，则软件必须发送 STOP 命令。如果字节计数不为 0，则在给定的字节数传送结束时，主机应该发送停止命令。

36.5.5. 单个数据块或多个数据块读

读数据块是基于块的数据传输。数据传输的基本单位是块，最大块大小在 CSD（READ_BL_LEN）中被定义，块的大小始终是 512 字节。如果 READ_BL_PARTIAL（在 CSD 中）被设置时，更小的块也可以被传输，其开始和结束地址被完全包含在 512 个字节的边界中。

CMD17（READ_SINGLE_BLOCK）表示开始读一个数据块，完成传输后卡返回发送状态。CMD18（READ_MULTIPLE_BLOCK）开始读连续的数据块。为了确保数据传输的完整性，每个数据块后都有一个 CRC 校验。

块长度由 CMD16 设置，可以设置为 512 字节而忽略 READ_BL_LEN 的设置。

数据块将不断传输，直到主机发出 STOP_TRANSMISSION 命令（CMD12）。由于串行命令传

输原因，停止命令有一个执行的延迟。在停止命令的结束位之后停止数据传输。

当使用 CMD18 读到用户区的最后一个块时，主机应该忽略可能会出现的 OUT_OF_RANGE 错误，即使序列是正确的。

如果主机传输的部分块的累积长度不是块对齐并且不允许块错位，卡将在第一个未对齐块的开始检测出块错位，并设置状态寄存器的 ADDRESS_ERROR 错误位，中断传输和等待在数据状态的停止命令。

单块或多块读操作步骤为：

1. 在 SDIO_DATALEN 寄存器中设置数据大小的字节数。
2. 在 SDIO_DATACTL 寄存器中设置块大小（BLKSZ）。主机每次从卡中读取 BLKSZ 大小的数据。
3. 在 SDIO_CMDAGMT 寄存器中设置需要读取数据的开始地址。
4. 设置 SDIO_CMDCTL 寄存器。对于 SD 和 e•MMC 卡，使用 CMD17 用于单块读取和 CMD18 为多块读取。对于 SD I/O 卡，使用 CMD53 用于单块和多块传输。
5. 软件应查询数据错误中断。如果需要，软件可以通过发送停止命令（CMD12）终止数据传输。
6. 软件应从 FIFO 中读数据，并腾出 FIFO 的空间用于接收更多的数据。
7. 当收到 DTEND 中断时，软件应读出 FIFO 中剩余的数据。

36.5.6. 数据流写和数据流读（仅适用于 e•MMC）

数据流写

数据流写（CMD20）开始从主机将数据传送到卡，从起始地址开始，直到主机发出停止命令。如果允许部分块传输（如果 CSD 参数 WRITE_BL_PARTIAL 被设置），数据流可以在卡地址空间内的任何地址启动和停止，否则应仅在块边界启动和停止。由于不预先确定要传输的数据量，CRC 不能使用。

如果主机提供了一个超出范围的地址作为参数传递给 CMD20，卡将拒绝该命令，留在传输状态，并将 ADDRESS_OUT_OF_RANGE 置位。

需要注意的是数据流写命令只适用于 1 位总线配置（DAT0 信号线上）。如果 CMD20 在其它总线配置中发出的，它被认为是非法的命令。

为了使卡保持在流模式的数据传输，接收数据所花费的时间（由总线时钟速率定义）必须比它需要写入到主存储器字段（由卡定义在 CSD 寄存器）的时间少。因此，流写入操作最大的时钟频率由下面给出的公式计算：

$$\text{max write frequence} = \min\left(\text{TRAN_SPEED}, \frac{8*2^{\text{WRITE_BL_LEN}}-100*\text{NSAC}}{\text{TAAC}*\text{R2W_FACTOR}}\right) \quad \text{（式 36-2）}$$

其中，TRAN_SPEED：最大的总线时钟频
WRITE_BL_LEN：最大写数据块长度
NSAC：以 CLK 周期计算的数据读访问时间 2
TAAC：数据读访问时间 1
R2W_FACTOR：写速度因子

所有的参数在 CSD 寄存器中定义。如果主机试图使用更高频率，卡可能不能够对数据进行处理，并将停止编程，同时忽略所有后续的数据传输并等待（在接收数据状态）一个停止指令。由于主机发送 CMD12，该卡将 TXURE 位置位并返回传输状态。

数据流读

由 READ_DAT_UNTIL_STOP（CMD11）控制数据流的数据传输。此命令指示卡从指定地址发送数据，直到主机发送一个 STOP_TRANSMISSION（CMD12）命令。由于串行命令传输停止的原因，命令有一个执行的延迟。停止命令的结束位之后数据传输停止。

如果主机提供了一个超出范围的地址作为参数传递给 CMD11，该卡将拒绝该命令，留在传输状态，并将 ADDRESS_OUT_OF_RANGE 位置位。

需要注意的是数据流读取命令只工作在 1 位总线配置（DAT0 信号线）。如果 CMD11 在其它总线配置中发出的，它被认为是非法的命令。

如果数据传输的地址到达存储范围的结束处时，主机还没有发送停止命令，则后续传输的有效载荷的内容是不确定的。由于主机发送 CMD12 命令，卡将 ADDRESS_OUT_OF_RANGE 位置位并返回传输状态。

为了使卡保持在流模式的数据传输，传输数据所花费的时间（由总线时钟速率定义）必须比它需要从主存储器字段（在 CSD 寄存器中由卡定义）读出的时间少。因此，流读取操作最大的时钟频率由下面给出的公式计算：

$$\text{max read frequence} = \min\left(\text{TRAN_SPEED}, \frac{8*2^{\text{READ_BL_LEN}}-100*\text{NSAC}}{\text{TAAC}*\text{R2W_FACTOR}}\right) \quad \text{（式 36-3）}$$

其中，TRAN_SPEED: 最大总线时钟频率
READ_BL_LEN: 最大读数据块长度
NSAC: 以 CLK 周期计算的数据读访问时间 2
TAAC: 数据读访问时间 1
R2W_FACTOR: 写速度因子

所有的参数在 CSD 寄存器中定义。如果主机试图使用更高频率，卡可能不能够对数据进行处理，并将停止编程，同时忽略所有后续的数据传输并等待（在接收数据状态）一个停止指令。由于主机发送 CMD12，该卡将 RXORE 位置位并返回传输状态。

36.5.7. 擦除

SD/e•MMC 存储卡的可擦除单位是“擦除组”，擦除组是以写数据块计算的，写数据块是卡的基本写入单元。擦除组的大小是一个卡特定的参数，在 CSD 中定义。

主机可以擦除连续范围的擦除组。开始擦除操作有三个步骤。首先，主机使用 ERASE_GROUP_START（CMD35）/ ERASE_WR_BLK_START（CMD32）命令定义了连续范围内的开始地址，然后使用 ERASE_GROUP_END（CMD36）/ ERASE_WR_BLK_END（CMD33）命令定义了连续范围内的结束地址，最后发送 ERASE（CMD38）命令启动擦除操作。在擦除命令中的地址字段是以字节为单位的擦除组地址。卡会舍弃未与擦除组大小对齐的部分，把地址边界对齐到擦除组的边界。

如果未按照定义的步骤接收到擦除命令（CMD35，CMD36 和 CMD38），卡应设置状态寄存器

的 ERASE_SEQ_ERROR 位，并重置整个序列。

如果主机提供了一个超出范围的地址作为参数传递给 CMD35 或 CMD36，卡将拒绝该命令，同时设置 ADDRESS_OUT_OF_RANGE 位，并重置整个擦除序列。

如果收到“非擦除”命令（既不是 CMD35，CMD36，CMD38 也不是 CMD13），卡应该设置 ERASE_RESET 位，重置擦除序列并执行最后一个命令。

如果擦除范围包括写保护块，它们应不被擦除，只有非保护块被擦除。应设置状态寄存器的 WP_ERASE_SKIP 状态位。

如上所述，对于块写入，卡将通过保持 DAT0 为低来指示擦除过程正在进行。实际擦除时间可能很长，主机可以发送 CMD7 命令以取消选择该卡。

36.5.8. 总线宽度选择

在主机已经验证了总线上的功能引脚后，卡初始化后可以改变总线宽度的配置。

对于 e•MMC 卡，使用 SWITCH 命令（CMD6）。总线宽度的配置是通过在 EXT_CSD 寄存器模式字段的 BUS_WIDTH 字节设置而改变的。上电或软件复位后，BUS_WIDTH 字节的内容为 0x00。如果主机试图写一个无效的值时，BUS_WIDTH 字节不会改变，同时设置 SWITCH_ERROR 位，另外该寄存器是只写的。

对于 SD 存储卡，使用 SET_BUS_WIDTH 命令（ACMD6）改变总线宽度。上电或 GO_IDLE_STATE 命令（CMD0）后默认总线宽度为 1 位。 SET_BUS_WIDTH（ACMD6）仅在传送状态有效，这表明仅在由 SELECT/DESELECT_CARD （CMD7）命令选择卡之后总线宽度才可以改变。

36.5.9. 保护管理

为了允许主机保护数据，使得其不被擦除或改写，有三种卡保护方式：

CSD 寄存器用于卡保护 （可选的）

通过在 CSD 寄存器中设置永久或临时的写保护位，整个卡可以被写保护。一些卡通过设置 CSD 的 WP_GRP_ENABLE 位支持一组扇区的写保护。它的大小在 CSD 寄存器中的 WP_GRP_SIZE 单元定义。SET_WRITE_PROT 命令设置指定写保护组的写保护，CLR_WRITE_PROT 命令清除指定写保护组的写保护。

高容量 SD 存储卡不支持写保护，不响应写保护命令（CMD28，CMD29 和 CMD30）。

写保护开关 （SD 存储卡和 SD I/O 卡）

在卡的侧面有一个机械的滑动开关，提供给用户设置是否对卡进行写保护。如果滑动片处在窗口打开的位置表明该卡被写保护。如果在窗口关闭的位置则卡没有写保护。

卡密码上锁/解锁

卡密码上锁/解锁的保护方式在章节 *卡上锁/解锁操作* 中描述。

36.5.10. 卡上锁/解锁操作

密码保护的功能允许主机使用密码锁住卡，当解锁卡的时候也使用该密码。其中密码存储在 128 位的 PWD 寄存器当中，密码的长度存储在 PWD_LEN 的 8 位寄存器中。这些寄存器是非易失性的，以至于电源开关不会清除他们。

已经上锁的卡支持所有的基本命令（class 0），ACMD41，CMD16 和锁卡命令（class 7）。因此主机可以对卡进行复位，初始化，选择，状态查询，但是无法获取卡上的数据。如果卡之前被设置过密码（PWD_LEN 的值不为 0），卡在每次上电后会自动上锁。

与存在的 CSD 寄存器写命令相同，上锁/解锁命令也只在卡的传输态有效。这意味着，上锁/解锁命令不包含地址参数，且必须在使用该命令前卡必须被选中。

卡上锁/解锁命令与卡单块写命令有着相同的结构和总线事务类型。传输的数据块包含命令所有需要的信息（密码设置模式，密码本身，卡上锁/解锁等）。*表 36-39. 上锁/解锁数据结构*为上锁/解锁命令的结构。

表 36-39. 上锁/解锁数据结构

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	保留（全设置为 0）				ERASE	LOCK_UNLOCK	CLR_PWD	SET_PWD
1	PWDS_LEN							
2	密码数据（PWD）							
……								
PWDS_LEN+1								

ERASE：该位为 1 时定义了强制擦除操作。字节 0 的位 3 将被设为 1（其他位应为 0）。所有该命令的其他字节将被卡忽略。

LOCK/UNLOCK：1 = 上锁，0 = 解锁。注意，此位可以和 SET_PWD 一起设置，不可以和 CLR_PWD 一起设置。

CLR_PWD：1 = 清除 PWD.

SET_PWD：1 = 设置新的密码到 PWD

PWDS_LEN：定义密码长度（字节）。在改变密码的情况下，这个长度应该是新旧密码长度之和。密码长度可达 16 个字节。在密码替换的情况下，新旧密码长度总和可达 32 个字节。

密码数据（PWD）：在设置一个新密码的情况下，它包含这个新的密码。如果修改密码，它包含旧的密码，后面是设置的新密码。

设置密码

- 如果卡之前未被选中，使用 CMD7 选中卡。
- 使用 CMD16 定义数据块长度，8 位卡上锁/解锁模式，8 位密码长度（字节为单位），新密码的字节数。在密码替换完成的情况下，块的大小应考虑新旧密码都会与命令一起被发送出去。
- 在数据线上，以合适的数据块大小发送卡上锁/解锁命令，包含 16 位的 CRC。数据块应指示模式（SET_PWD），密码长度（PWDS_LEN）和密码本身。在密码替换完成的情况下，密码长度值（PWDS_LEN）应为新旧密码长度之和，密码数据字段应包括旧的密码

（当前使用），后面是新的密码。需要注意的是卡需要内部处理新密码长度的计算，通过从 PWDS_LEN 字段减去旧密码长度。

- 当发送的旧密码不正确（大小和内容不相同），状态寄存器中的 LOCK_UNLOCK_FAILED 会被置位，并且旧的密码不会改变。如果发送的旧密码正确（大小和内容相同），新的密码数据及其长度会分别保存在 PWD 和 PWD_LEN 中。

复位密码

- 如果卡之前未被选中，使用 CMD7 选中卡。
- 使用 CMD16 定义数据块长度，8 位卡上锁/解锁模式，8 位密码长度（字节为单位），当前使用的密码的字节数。
- 在数据线上，以合适的数据块大小发送卡上锁/解锁命令，包含 16 位的 CRC。数据块指示模式（CLR_PWD），密码长度（PWDS_LEN）和密码本身。如果 PWD 和 PWD_LEN 的内容与发送的密码和其大小匹配，PWD 寄存器的内容会被清除，同时 PWD_LEN 被设为 0。如果密码不正确，状态寄存器中的 LOCK_UNLOCK_FAILED 会被置位。

卡上锁

- 如果卡之前未被选中，使用 CMD7 选中卡。
- 使用 CMD16 定义数据块长度，8 位卡上锁/解锁模式，8 位密码长度（字节为单位），当前使用的密码的字节数。
- 在数据线上，以合适的数据块大小发送卡上锁/解锁命令，包含 16 位的 CRC。数据块指示 LOCK 模式，密码长度（PWDS_LEN）和密码本身。

如果 PWD 内容等于发送的密码，卡将会被上锁，并且状态寄存器中卡上锁状态位（CARD_IS_LOCKED）会被置位。如果密码不正确，状态寄存器中 LOCK_UNLOCK_FAILED 会被置位。

卡解锁

- 如果卡之前未被选中，使用 CMD7 选中卡。
- 使用 CMD16 定义数据块长度，8 位卡上锁/解锁模式，8 位密码长度（字节为单位），当前使用的密码的字节数。
- 在数据线上，以合适的数据块大小发送卡上锁/解锁命令，包含 16 位的 CRC。数据块指示 UNLOCK 模式，密码长度（PWDS_LEN）和密码本身。

如果 PWD 内容等于发送的密码，卡将会被解锁，并且状态寄存器中卡上锁状态位（CARD_IS_LOCKED）会被清除。如果密码不正确，状态寄存器中 LOCK_UNLOCK_FAILED 会被置位。

36.5.11. 睡眠

e•MMC 卡可以通过 CMD5 在睡眠模式和待机模式之间切换。在睡眠状态下，卡的功耗最小化，此时可以关闭 V_{CC} 电源。

– 睡眠命令：CMD5 的参数的第 15 位为 1.
– 唤醒命令：CMD5 的参数的第 15 位为 0.

睡眠命令用于启动从待机状态到睡眠状态的状态转换。卡通过下拉 SDIO_DAT0 线来指示过渡

阶段繁忙。在繁忙期间不应该发送其他命令。当卡停止下拉 SDIO_DAT0 线时，则达到睡眠状态，完成了状态的转换。

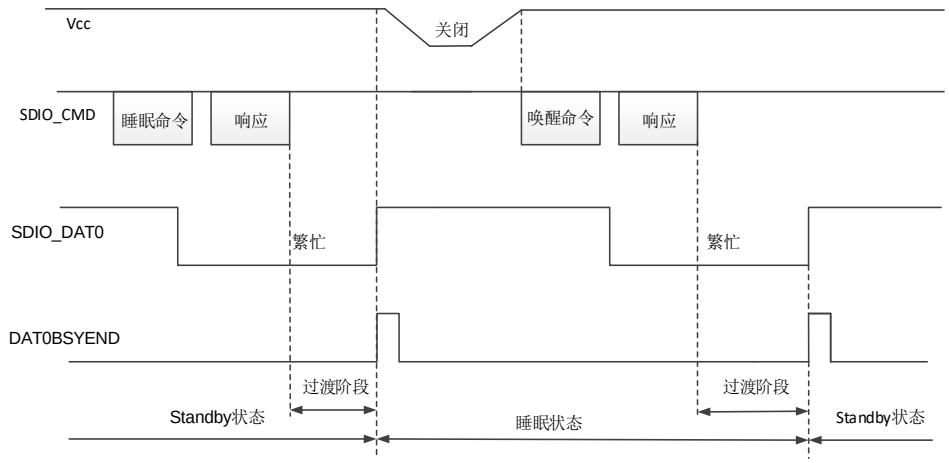
唤醒命令用于启动从睡眠状态到待机状态的状态转换。卡通过下拉 SDIO_DAT0 线来指示过渡阶段繁忙。在繁忙期间不应该发送其他命令。当卡停止下拉 SDIO_DAT0 线时，则达到了待机状态，完成了状态的转换。

设置卡睡眠，需要遵从以下步骤：
1. 使能 DAT0BSYEND 中断。
2. 发送 CMD5（睡眠命令）。
3. 当 DAT0BSYEND 中断出现，则卡处于睡眠状态。
4. 允许关闭 Vcc 电源。

设置卡进入待机状态，需要遵从以下步骤：
1. 打开 Vcc 电源且等待卡进入最小工作电压。
2. 使能 DAT0BSYEND 中断。
3. 发送 CMD5（唤醒命令）
4. 当 DAT0BSYEND 中断产生时，则卡已经处于待机状态。

在睡眠状态期间，Vcc 电源可能会关闭。这是为了进一步节省系统功耗。Vcc 电源只允许在达到到睡眠状态（卡已停止下拉 SDIO_DAT0 线）后，才允许关闭。在允许启动从睡眠状态到待机状态的状态过渡（唤醒命令）之前，Vcc 电源必须至少提高到最小工作电压水平。

图 36-18. CMD5 时序



36.5.12. CMD12 发送时序

CMD12 被用于停止或中止数据传输。卡的数据传输在停止传输命令的结束位后 2 个时钟周期后终止。

所有的读写命令都可以在任何时间被停止传输命令 CMD12 中止。如果数据传输正在进行，CMD12 的发送时序需要遵从以下流程：

1. 在寄存器中配置 CMD12 命令并且置位 TRSTOP 寄存器。当 DSM 收到命令，CSM 将产生中止信号。

2. 清零 WAITDEND 寄存器位。

3. 当 IDMAEN = 0 时，则置位 FIFOREST 位。
- 主机发送数据，当 CMDRECV 标志出现时，固件将停止传输数据到 FIFO。随后置位 FIFOREST 并且刷新 FIFO。
- 主机接收数据，当 CMDRECV 标志出现时，固件将从 FIFO 读取剩余的数据。随后置位 FIFOREST。

4. 当 IDMAEN = 1 时，硬件将操作 FIFO。
- 主机发送数据，当中止信号出现时，硬件将停止 IDMA。随后刷新 FIFO。
- 主机接收数据，当中止信号出现时，硬件将通过 IDMA 将 FIFO 中剩余的数据传输到 RMA。

5. 当 FIFO 是空或者复位状态，将生成 DATABOR 标志。

表 36-40. CMD12 的用法

数据操作类型	CMD12 的作用
预定块数的多块写	卡接收完需求块数的数据块，然后停止传输并返回卡的传输状态。在多块写末尾的 CMD12 停止命令不是必要的，除非发生错误。
预定块数的多块读	卡传送完需求块数的数据块，然后停止传输并返回卡的传输状态。在多块读末尾的 CMD12 停止命令不是必要的，除非发生错误。
开放终点的多块写	多块写的块数没有定义，卡将一直接收并写数据直到收到 CMD12 的停止命令。
开放终点的多块读	多块读的块数没有定义，卡将一直接收并读数据直到收到 CMD12 的停止命令。
流写入	通过发送 CMD12 停止命令来停止或中止数据传输。
流读取	通过发送 CMD12 停止命令来停止或中止数据传输。

块操作中 CMD12 的使用

要在数据结束时停止块传输，需要在最后一个数据块结束位之后发送 CMD12 结束位.

当写数据到卡时，需要在写数据的 CRC 令牌结束位之后发送 CMD12 结束位。CMD12 传送过程应遵循块传输时序。

停止开放终点的多块写操作遵循以下步骤：

1. 开始数据传输前设置 TRANSMOD[1:0]为“11”。

2. 等待 DTEND 标志置位，DSM 发送的数据不会超过 SDIO_DATALEN。

3. CSM 发送 CMD12，卡被设置为空闲状态。

当从卡读取数据时，CMD12 的结束位应该尽早发送，即在卡读取数据块的最后一位时发送。

停止开放终点的多块读操作遵循如下步骤：

1. 在开始数据传输前设置 TRANSMOD[1:0]为“11”。

2. 等待 DTEND 标志置位，即使卡发送更多的数据，DSM 接收的数据不会超过 SDIO_DATALEN。

3. CSM 发送 CMD12。卡发送数据被中止且被设置在空闲状态。

流操作中 CMD12 的使用

要在待传输的最后一个字节之后停止流传输，应在数据流的最后一个字节结束时发送 CMD12 结束位时序。请按照以下的流写入步骤操作：

1. 初始化 DSM，设置 TRANSMOD[1:0]为“10”。

2. 置位 TREN，发送 WRITE_DATA_STREAM 命令。

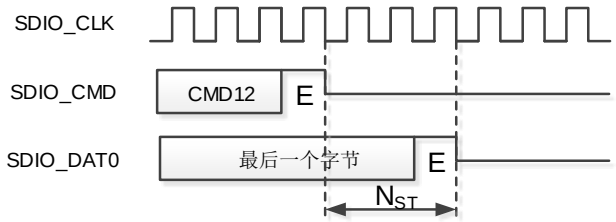
3. 在命令寄存器中预加载 CMD12 命令，置位 TRSTOP 位。

4. 配置 CSM 为仅在 DATALEN 数据长度的最后一笔数据的等待挂起（WAITDEND = 1）结束后发送命令。

5. CSM 发送 CMD12，流数据结束位和命令结束位应对齐。
- 如果 DATALEN > 5，CMD12 在 CSM 中等待与数据传输结束位对齐。
- 如果 DATALEN < 5，CMD12 将提前启动，DSM 将保持 WaitS 状态，使数据传输结束位与 CMD12 结束位对齐。

6. 通过清零 WAITDEND 位，流数据写入过程可以在任何时候停止。这将导致预加载的 CMD12 立即发送并停止流写入过程。

图 36-19. CMD12 影响流操作的时序



要在流读取的最后一个字节之后停止流传输过程，CMD12 结束位时序应该在数据流的最后一个字节之后发送。

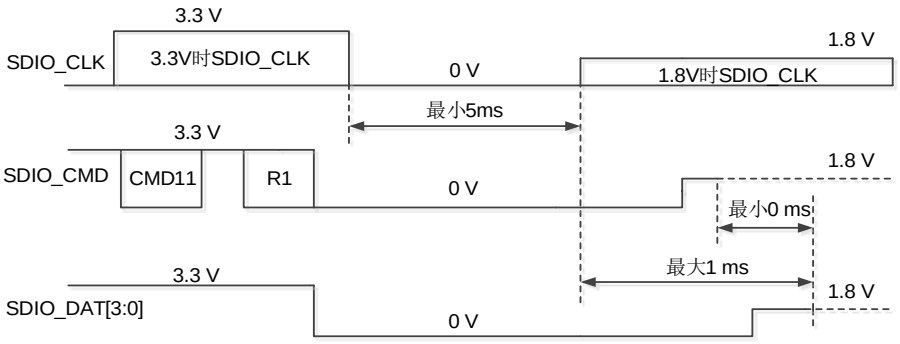
注意：1. CMD12 发送前等待 DTEND=1（数据接收完成），即使卡发送更多的数据，但是 DSM 不会读取超过 SDIO_DATALEN 长度的数据。2. 一旦 DATACNT=0，即使卡继续发送数据，SDIO 也不会再接收。

36.6. 特定操作

36.6.1. UHS-I 电压切换

UHS-I（即超高速总线速度 I 相，包括：SDR12、SDR25、SDR50、 SDR104 和 DDR50）工作在 1.8V 电压下，而卡上电时是 3.3V 的电压启动，所以 UHS-I 模式需要支持 3.3V 到 1.8V 的电压切换。当电压切换序列成功完成时，卡将以默认 SDR12 进入 UHS-I 模式，卡输入和输出时序将发生变化。

图 36-20. CMD11 电压切换时序



电压切换时序需要遵循以下流程进行：

1. 在开始电压切换前，SDIO_CLK 时钟频率必须配置在 100kHz-400kHz 范围内。
2. 主机置位电压切换序列使能位（VSEN = 1）开始电压切换，并发送 CMD11。
3. 主机收到卡 R1 类型的命令响应。
   - 如果响应的 CRC 检查通过，电压切换序列将继续，直到电压切换序列完成前主机将不再驱动 CMD 和 SDIO_DAT[3：0]信号线。响应后的若干时钟周期，SDIO_CLK 时钟停止并产生 CLKSTOP 标志。
   - 如果响应的 CRC 检查错误（CCRCERR = 1）或者响应超时（CMDTMOUT = 1），电压切换序列被停止。
4. 在 R1 响应的下一个时钟周期，卡拉低 CMD 线和 SDIO_DAT[3：0]线。
5. 收到 R1 响应后，主机可以使用 DAT0BSY 寄存器位监视 SDIO_DAT0 线。在两个 SDIO_CLK 时钟周期后采样 SDIO_D0 线。主机可以读取 DAT0BSY 标志：
   - 当检测到 DAT0BSY 低电平时，主机将电压调节器切换到 1.8V，然后置位寄存器位 VSSTART，以指示 SDIO 启动电压切换序列的时序关键部分。硬件通过保持 SDIO_CLK 低至少 5ms 来继续停止时钟。
   - 当检测到 DAT0BSY 为高电平时，主机将中止电压切换序列并对卡重新上电。
6. 如果 SDIO_CLK 信号线是低电平时，卡开始将电压切换到 1.8V。
7. SDIO 的硬件将在至少 5ms 后重新启动 SDIO_CLK 时钟。
8. 检测到 SDIO_CLK 切换后的 1ms 内，卡将拉高 CMD 线和 DAT[3：0]线至少 1 个时钟周期后，卡停止驱动 CMD 线和 DAT[3：0]。
9. SDIO 硬件在 SDIO_CLK 重启的 1ms 后，主机采样 SDIO_DAT0 通过 DAT0BSY 位，且生成电压切换关键时序完成（VSEND）标志。
10. 如果生成 VSEND 标志，主机需要读取 DAT0BSY 寄存器位来判断 SDIO_DAT0 线，来确定电压切换序列是否完成：
    - 如果 DAT0BSY 为高，电压切换序列成功完成。
    - 如果 DAT0BSY 为低，电压切换序列失败，主机对卡进行重新上电

36.7. SDIO 寄存器

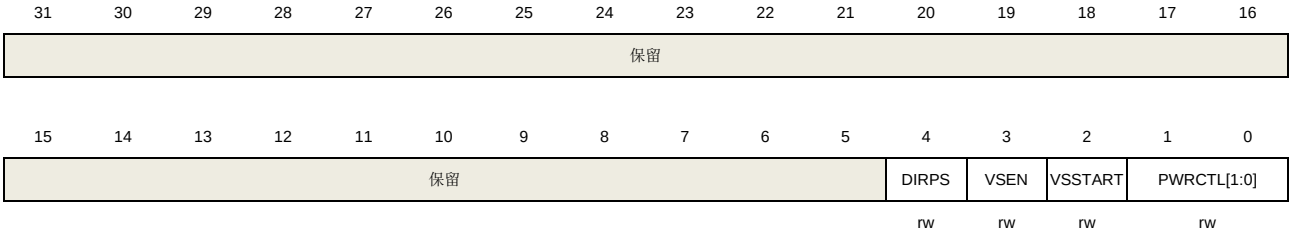
SDIO0 基地址：0x5200 7000

SDIO1 基地址：0x4802 2400

36.7.1. 电源控制寄存器（SDIO_PWRCTL）

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
31:5	保留	必须保持复位值。
4	DIRPS	命令和数据方向极性选择位 只有在 PWRCTL[1:0]清零状态时，才能写入该位。 0：方向信号为低电平时，电压收发器驱动 IOs 作为输出 1：方向信号为高电平时，电压收发器驱动 IOs 作为输出
3	VSEN	电压切换使能位 该位用于在电压切换命令响应后停止 SDIO_CLK。 只有在 CSMEN 位清零的条件下，固件才能修改该位。 0：在成功接收到命令响应后，SDIO_CLK 时钟保持不变 1：在成功接收到命令响应后，SDIO_CLK 时钟停止
2	VSSTART	电压切换启动位 该位用于启动电压切换的时序关键部分。 0：电压切换没有激活也没有启动 1：电压切换激活或启动
1:0	PWRCTL[1:0]	SDIO 电源控制位 该位域控制 SDIO 状态，卡输入或输出。 该位域只能在 SDIO 断电状态下写入。 00：复位之后（复位：SDIO 被禁用，时钟停止，SDIO_CMD 和 SDIO_DAT 处于高阻状态，且 SDIO_CLK 是低电平），如果写 00，SDIO 掉电（掉电：SDIO 被禁用，时钟停止，SDIO_DAT、SDIO_CMD 和 SDIO_CLK 都为高电平） 01：保留（写 01，寄存器值不变） 10：掉电再上电（SDIO 被禁用且卡的时钟停止，SDIO_DAT、SDIO_CMD 和 SDIO_CLK 都为低电平）

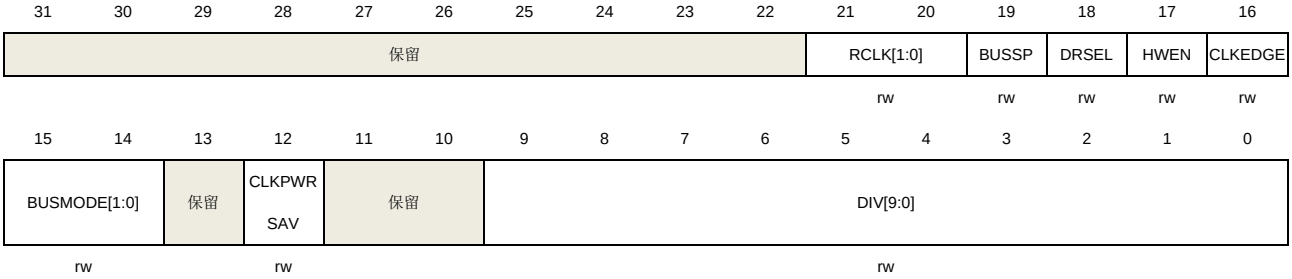
11：SDIO 上电（上电后，使能 SDIO 有 74 个时钟周期的延迟，延迟期间任何写操作都被忽略，PWRCTL 寄存器保持 11）

36.7.2. 时钟控制寄存器（SDIO_CLKCTL）

地址偏移：0x04
复位值：0x0000 0000

该寄存器控制输出时钟 SDIO_CLK。

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。
21:20	RCLK[1:0]	接收时钟选择位 该位域只能在 CMDSTA = 0 且 DATSTA = 0 时写入。 00：选择 SDIO_IN_CLK 时钟 01：选择 SDIO_CLKIN 时钟 10：选择 SDIO_FB_CLK 时钟 11：保留（默认选择 SDIO_IN_CLK 时钟）
19	BUSSP	总线速度模式选择位 该位只能在 CMDSTA = 0 且 DATSTA = 0 时写入。 0：DS，HS，SDR12，SDR25 总线模式 1：SDR50，DDR50，SDR104 总线模式
18	DRSEL	数据速率选择位 该位只能在 CMDSTA = 0 且 DATSTA = 0 时写入。 0：SDR 模式被选择 1：DDR 模式被选择
17	HWEN	硬件流控制使能位 如果该位置位，TFF 和 RFF 寄存器的意义被改变。 该位只能在 CMDSTA = 0 且 DATSTA = 0 时写入。 0：硬件流控制被禁用 1：硬件流控制被使能
16	CLKEDGE	命令和数据 SDIO_CLK 移相选择位 该位只能在 CMDSTA = 0 且 DATSTA = 0 时写入。

		当 DIV = 0 时，该位不起作用。数据和命令在 SDIO_CLK 的下降沿变化。 0：如果 DIV > 0 且在 SDR 模式，命令和数据在 SDIO_CLK 上升沿后的 CK_SDIO 下降沿改变。SDIO_CLK 在 CK_SDIO 的上升沿产生。 如果 DIV > 0 且在 DDR 模式，命令在 SDIO_CLK 上升沿后的 CK_SDIO 下降沿改变。数据在 SDIO_CLK 上升沿后的 CK_SDIO 下降沿改变。SDIO_CLK 在 CK_SDIO 的上升沿产生。 1：如果 DIV > 0 且在 SDR 模式，命令和数据在 SDIO_CLK 下降沿后的 CK_SDIO 下降沿改变。SDIO_CLK 在 CK_SDIO 的上升沿产生。 如果 DIV > 0 且在 DDR 模式，命令在 SDIO_CLK 下降沿后的 CK_SDIO 上升沿改变。数据在 SDIO_CLK 边沿后的 CK_SDIO 下降沿改变。SDIO_CLK 在 CK_SDIO 的上升沿产生。
15:14	BUSMODE[1:0]	SDIO 卡总线模式控制位 该位域只能在 CMDSTA = 0 且 DATSTA = 0 时写入。 00：选择 1 位总线模式（默认选择且只能 SDR 模式），SDIO_DAT0 01：选择 4 位总线模式，SDIO_DAT[3：0] 10：选择 8 位总线模式，SDIO_DAT[7：0] 11：保留
13	保留	必须保持复位值。
12	CLKPWRSAV	SDIO_CLK 时钟动态开启/关闭以节省功耗 该位在总线空闲的时候，控制 SDIO_CLK 时钟动态开启/关闭以节省功耗。 该位只能在 CMDSTA = 0 且 DATSTA = 0 时写入。 0：SDIO_CLK 时钟总是开启 1：SDIO_CLK 时钟在总线空闲时关闭
11:10	保留	必须保持复位值。
9:0	DIV[9:0]	时钟分频 该位域只能在 CMDSTA = 0 且 DATSTA = 0 时写入。 该位域定义了分频因子来向卡产生 SDIO_CLK 时钟。SDIO_CLK 是由 CK_SDIO 分频得到，并且 SDIO_CLK 频率= CK_SDIO / （DIV[9:0] * 2） 0x000：SDIO_CLK = CK_SDIO / 1（不支持 DDR 模式） 0x001：SDIO_CLK = CK_SDIO / 2 0x002：SDIO_CLK = CK_SDIO / 4 …… 0x3FF：SDIO_CLK = CK_SDIO / 2046

注意：SD 卡、SD I/O 卡或 e•MMC 卡在卡识别模式时，SDIO_CLK 频率必须不大于 400kHz。

如果 RCA 已经被分配给所有的卡时，可以改变时钟频率到最大的卡总线频率。

在两次对该寄存器的写访问之间，至少需要 7 个 HCLK 周期，用于同步寄存器到 SDIO_CLK 时钟域。

36.7.3. 命令参数寄存器（SDIO_CMDAGMT）

地址偏移：0x08
复位值：0x0000 0000

该寄存器定义了 32 位命令参数，这些参数将被用作为命令的一部分（位 39 到位 8）。

该寄存器只能按字（32 位）访问。

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

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

位/位域	名称	描述
31:0	CMDAGMT[31:0]	SDIO 卡命令参数 该位域定义了将被发送到卡的 SDIO 卡命令参数。这个域是命令消息的位[39:8]。如果命令消息包含一个参数，在发送命令时，这个域应该在写 SDIO_CMDCTL 寄存器前更新。 只有在 CSMEN 位清零时，该位才能被固件写入。

36.7.4. 命令控制寄存器（SDIO_CMDCTL）

地址偏移：0x0C
复位值：0x0000 0000

SDIO_CMDCTL 寄存器包含命令索引和其他命令控制位来控制命令状态机（CSM）。

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOOTMODEN	BOOTMOD	HOLD	CSMEN	WAITDEND	INTWAIT	CMDRESP[1:0]		TRSTOP	TREN	CMDIDX[5:0]					
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	CMDSR	将命令视为挂起或恢复命令并发出中断周期开始/结束位。 只有在 CSMEN 位清零时，该位才能被固件写入。 0：无作用 1：如果 TREN = 0，CSM 将命令视为挂起命令； 如果 TREN = 1，CSM 将命令视为恢复命令
15	BOOTMODEN	引导模式使能位 0：禁用引导模式 1：使能引导模式
14	BOOTMOD	引导模式选择位 只有在 CSMEN 位清零时，该位才能被固件写入。

		0：正常引导模式 1：备用引导模式
13	HOLD	保持 DSM 发送和接收新的数据块 该位置位时，DSM 不会从 DS_WaitS 状态切换到 DS_Send 状态，或者从 DS_WaitR 状态切换到 DS_Receive 状态。 只有在 CSMEN 位清零时，该位才能被固件写入。 0：无影响 1：保持在数据传输或接收的状态
12	CSMEN	命令状态机（CSM）使能位 该位固件写入，硬件清零。 0：命令状态机失能（停留在 CS_Idle 状态） 1：命令状态机使能
11	WAITDEND	等待数据传输结束 如果该位置位，命令状态机开始发送命令前需要等待数据传输结束。 0：无影响 1：等待数据传输结束
10	INTWAIT	等待中断请求 如果在 Wait 状态下清零该位，将导致中断模式中止。 0：无影响 1：在命令响应时，禁止命令超时，并等待卡中断请求。
9:8	CMDRESP[1:0]	命令响应类型位 这些位定义了发送一个命令消息后的响应类型。 只有在 CSMEN 位清零时，该位域才能被固件写入。 00：无响应 01：短响应 10：短响应（无 CRC） 11：长响应
7	TRSTOP	数据传输停止命令模式使能位 传输停止命令模式（CSM 将命令当做数据停止传输的命令） 只有在 CSMEN 位清零时，该位才能被固件写入。 0：无影响 1：当发送命令时，使能传输停止命令模式且 CSM 将向 DSM 发中止信号
6	TREN	数据传输命令模式使能位 传输命令模式（CSM 将命令当做数据传输的命令） 只有在 CSMEN 位清零时，该位才能被固件写入。 0：无影响 1：如果命令被发送，使能传输命令模式且 DSM 数据传输并且中断周期结束
5:0	CMDIDX[5:0]	命令索引 只有在 CSMEN 位清零时，该位才能被固件写入。 该位域定义了将被发送到 SDIO 卡的命令索引。

注意：两次对该寄存器写访问之间，至少需要 7 个 HCLK 时钟周期，用于将寄存器同步到 SDIO_CLK 时钟域。

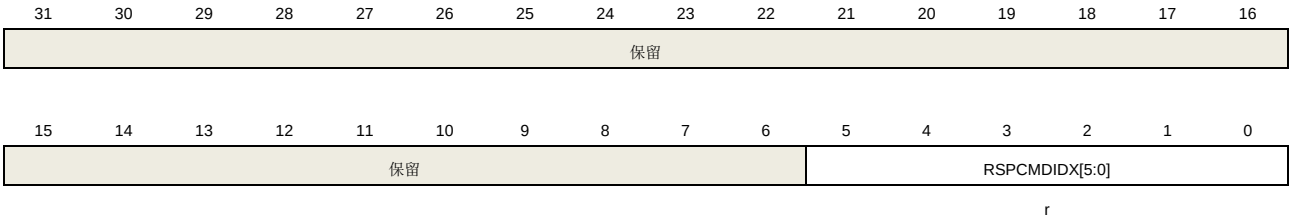
e•MMC 可以发送两种响应:短响应，48 位，或长响应，136 位。

SD 卡和 SD I/O 卡只能发送短响应，参数可以根据响应的类型而变化：软件根据发送命令来区分响应的类型。

36.7.5. 命令索引响应寄存器（SDIO_RSPCMDIDX）

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字（32位）访问。



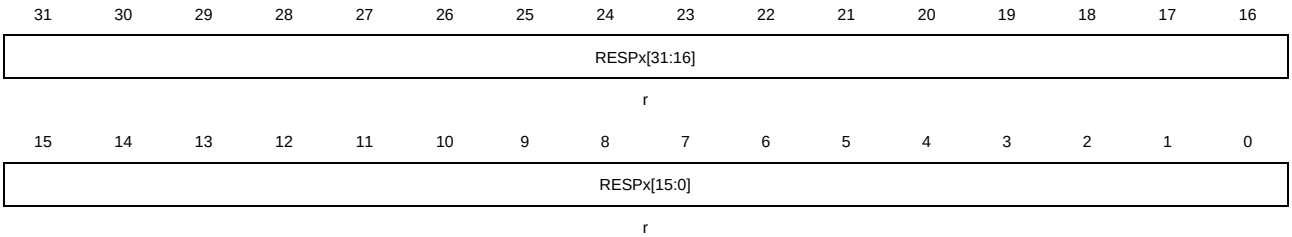
位/位域	名称	描述
31:6	保留	必须保持复位值。
5:0	RSPCMDIDX[5:0]	最后响应的命令索引 只读位域。这个域包含收到的最后命令响应的命令索引。如果响应没有命令索引（R3的长响应和短响应），这个寄存器的内容是未定义的。

36.7.6. 响应寄存器（SDIO_RESPx x = 0..3）

地址偏移：0x14+（4*x）, x = 0..3
复位值：0x0000 0000

这些寄存器包含最后收到的卡响应的内容。

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:0	RESPx[31:0]	卡状态。响应内容由*表 36-41. 不同响应类型对应的 SDIO_RESPx 寄存器*所示。

短响应为 32 位，长响应为 127 位（位 128 是结束位 0）。

表 36-41. 不同响应类型对应的 SDIO_RESPx 寄存器

寄存器	短响应	长响应
SDIO_RESP0	卡响应 [31:0]	卡响应 [127:96]
SDIO_RESP1	保留	卡响应 [95:64]
SDIO_RESP2	保留	卡响应 [63:32]
SDIO_RESP3	保留	卡响应 [31:1]，加上位 0

36.7.7. 数据超时寄存器（SDIO_DATATO）

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
DATATO[31:16]															
rw															

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

位/位域	名称	描述
31:0	DATATO[31:0]	数据超时时间 这些位定义了数据超时时间，由 SDIO_CLK 计数。当 DSM 进入 WaitR 或 BUSY 状态，该寄存器的值加载到内部计数器开始递减。DSM 超时并进入空闲状态，当计数器的值减至 0 时设置 DTTMOUT 标志。 该域只能在 CMDSTA = 0 且 DATSTA = 0 时写入。

注意：当需要数据传输时，数据超时器寄存器和数据长度寄存器应在写数据控制寄存器前更新。

36.7.8. 数据长度寄存器（SDIO_DATALEN）

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							DATALEN[24:16]								
							rw								

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

位/位域	名称	描述
31:25	保留	必须保持复位值。
24:0	DATALEN[24:0]	数据传输长度

该寄存器定义了需要传输的字节数。当数据传输开始时，这个寄存器的值将加载到数据计数器并开始递减。
只有在 DATSTA=0 时，该位才能写入。
当 DRSEL=1 时，DATALEN 会被截断为 2 的倍数，最后一个奇数字节不传输。
当 DATALEN 为 0 时，无数据传输。当有 CSMEN 和 TREN=1 的请求时，也没有命令传输。DATAEN 和 CSMEN 被清零。

注意：如果选择了数据块传输，该寄存器的内容应该为块大小的倍数（参考 SDIO_DATACTL 寄存器）。当需要数据传输时，数据计时器寄存器和数据长度寄存器应在写数据控制寄存器前更新。

对于多字节传输，数据长度寄存器中的取值必须在 1 到 512 之间。

36.7.9. 数据控制寄存器（SDIO_DATACTL）

地址偏移：0x2C
复位值：0x0000 0000

该寄存器控制 DSM。该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	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		FIFORES T	ACKEN	保留				BLKSZ[3:0]				TRANSMOD[1:0]		DATADIR	DATAEN
		rw	rw							rw		rw		rw	rw

位/位域	名称	描述
31:14	保留	必须保持复位值。
13	FIFOREST	FIFO 复位，刷新所有的数据 该位只有在 IDMAEN = 0 且 DATSTA = 1 时，才能被固件写入。当 DATSTA = 0 时，该位被硬件自动清零。 如果有传输错误或传输保持时，该位才有效。 0：无影响 1：刷新所有剩余的数据并复位 FIFO 指针
12	ACKEN	引导确认使能位 该位只有在 DATSTA 位清零时，才能被固件写入。 0：引导确认禁用 1：引导确认使能
11:8	保留	必须保持复位值。
7:4	BLKSZ[3:0]	数据块大小 这些位定义了当数据传输是块传输时数据块的大小。 该位只有在 DATSTA 位清零时，才能被固件写入。 0000：块大小 = 2^0 = 1 字节 0001：块大小 = 2^1 = 2 字节

0010：块大小 = 2^2 = 4 字节
0011：块大小 = 2^3 = 8 字节
0100：块大小 = 2^4 = 16 字节
0101：块大小 = 2^5 = 32 字节
0110：块大小 = 2^6 = 64 字节
0111：块大小 = 2^7 = 128 字节
1000：块大小 = 2^8 = 256 字节
1001：块大小 = 2^9 = 512 字节
1010：块大小 = 2^{10} = 1024 字节
1011：块大小 = 2^{11} = 2048 字节
1100：块大小 = 2^{12} = 4096 字节
1101：块大小 = 2^{13} = 8192 字节
1110：块大小 = 2^{14} = 16384 字节
1111：保留
注意：当 DATALEN 不是 BLKSZ 的倍数时，传输的数据将在 BLKSZ 的倍数处截断。

DDR 模式，如果块大小设置为 1 字节，则不会发生数据传输。

3:2	TRANSMOD[1:0]	数据传输模式 该位域只有在 DATSTA 位清零时，才能被固件写入。 00：块传输模式 01：多字节传输模式（只适用于 SD/SD I/O 卡） 10：流传输（只适用于 e•MMC 卡） 11：需要 CMD12 终止传输（开放终点）
1	DATADIR	数据传输方向 该位只有在 DATSTA 位清零时，才能被固件写入。 0：写数据到卡上 1：从卡中读取数据
0	DATAEN	数据传输使能位 数据传输命令未使用时，该位才被用于数据传输。 该位只有在 DATSTA 位清零时，才能被固件写入。该位在数据传输完成时，硬件自动清零。 0：无影响 1：启动数据传输，无 CSM 参与。

36.7.10. 数据计数寄存器（SDIO_DATACNT）

地址偏移：0x30
复位值：0x0000 0000

该寄存器为只读类型。当 DSM 从空闲状态进入 DS_WaitR 或者 DS_WaitS 时，该寄存器从数据长度寄存器（SDIO_DATALEN）加载数值。随着数据传输，数值不断递减直至为 0，随后 DSM 进入空闲状态并设置数据结束标志 DTEND。

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:25	保留	必须保持复位值。
24:0	DATACNT[24:0]	数据计数值 只读位域。当读取这些位时，返回待传输剩余数据的字节数。

注意：该寄存器只在数据传输完成或保持时读取。在错误事件后读取该值时，读取的数据计数值可能与实际传输的数据字节数不相等。

36.7.11. 状态寄存器（SDIO_STAT）

地址偏移：0x34
复位值：0x0000 0000

该寄存器为只读类型。下面描述标志的类型：

位[28:21, 11:0]的标志只能通过向中断清除寄存器（SDIO_INTC）中相应的位写'1'清除。

位[20:12]的标志是根据硬件逻辑而发送变化的。

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			IDMAEND	IDMAERR	CLKSTOP	VSEND	ACKTO	ACKFAIL	SDIOINT	DAT0BSYEND	DAT0BSY	RFE	TFE	RFF	TFF
			r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RFH	TFH	DATSTA	CMDSTA	DATABOR	DTBLKEND	DATHOLD	DTEND	CMDSEND	CMDRECV	RXORE	TXURE	DTTMOUT	CMDTMOUT	DTCRCERR	CCRCERR
r	r	r	r	r	r	r	r	r	r	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	IDMAEND	IDMA 传输结束
27	IDMAERR	IDMA 传输错误
26	CLKSTOP	电压切换期间 SDIO_CLK 停止
25	VSEND	电压切换时关键时序完成

24	ACKTO	引导确认超时
23	ACKFAIL	引导确认接收且检查错误
22	SDIOINT	SD I/O 中断已接收
21	DAT0BSYEND	DAT0 线繁忙到准备好
20	DAT0BSY	DAT0 线繁忙 硬件状态标志，不会产生中断。
19	RFE	接收 FIFO 为空 硬件状态标志，不会产生中断。
18	TFE	发送 FIFO 为空 当 FIFO 中有一个变满时，该位清零。
17	RFF	接收 FIFO 为满 当 FIFO 中有一个变空时，该位清零。
16	TFF	发送 FIFO 为满 硬件状态标志，不会产生中断。 该位在 FIFO 变空时清零。
15	RFH	接收 FIFO 半满：FIFO 中至少还有一半数目的字可以被读取 当 FIFO 中有（half+1）个空字时，该位被清零。
14	TFH	发送 FIFO 半空：至少还有一半数目的字可以被写到 FIFO 中 当 FIFO 中的数据变为（half+1）个时，该位被清零。
13	DATSTA	数据通道激活状态 该位只是硬件状态标志，不产生中断。
12	CMDSTA	命令通道激活状态 该位只是硬件状态标志位，不产生中断。
11	DATABOR	数据传输被 CMD12 中止
10	DTBLKEND	数据块已发送/已接收（CRC 检测通过）
9	DATHOLD	数据传输保持
8	DTEND	数据结束（数据计数器，SDIO_DATACNT 为零）
7	CMDSEND	命令已发送（不需响应）
6	CMDRECV	命令响应已接收（CRC 检测通过）
5	RXORE	接收 FIFO 上溢错误发生
4	TXURE	发送 FIFO 下溢错误发生
3	DTTMOUT	数据超时，数据超时时间取决于 SDIO_DATATO 寄存器。
2	CMDTMOUT	命令响应超时，命令超时时间为 64 个 SDIO_CLK 时钟周期的固定值。

1	DTCRCERR	数据块已发送/已接收（CRC 检测失败）
0	CCRCERR	命令响应已接收（CRC 检测失败）

注意：如果使用 IDMA 模式，FIFO 中断不能够被使能。

36.7.12. 中断清除寄存器（SDIO_INTC）

地址偏移：0x38
复位值：0x0000 0000

该寄存器为只写。对该寄存器的位写 1 可以清除 SDIO_STAT 寄存器中相应的状态位。

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			IDMAENDC	IDMAERRC	CLKSTOPC	VSENDC	ACKTOC	ACKFAILC	SDIOINTC	DAT0BSYENDC	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				DATABORC	DTBLKENDC	DATHOLDC	DTENDC	CMDSENDC	CMDRECVC	RXOREC	TXUREC	DTTMOUTC	CMDTMOUTC	DTCRCERRC	CCRCERRC
				w	w	w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:29	保留	必须保持复位值。
28	IDMAENDC	IDMAEND 标志清楚位 写 1 清除标志。
27	IDMAERRC	IDMAERR 标志清楚位 写 1 清除标志。
26	CLKSTOPC	CLKSTOP 标志清楚位 写 1 清除标志。
25	VSENDC	VSEND 标志清楚位 写 1 清除标志。
24	ACKTOC	ACKTO 标志清楚位 写 1 清除标志。
23	ACKFAILC	ACKFAIL 标志清除位 写 1 清除标志。
22	SDIOINTC	SDIOINT 标志清除位 写 1 清除标志。
21	DAT0BSYENDC	DAT0BSYEND 标志清除位 写 1 清除标志。

20:12	保留	必须保持复位值。
11	DATABORC	DATABOR 标志清除位 写 1 清除标志。
10	DTBLKENDC	DTBLKEND 标志清除位 写 1 清除标志。
9	DATHOLDC	DATHOLD 标志清除位 写 1 清除标志。
8	DTENDC	DTEND 标志清除位 写 1 清除标志。
7	CMDSENDC	CMDSEND 标志清除位 写 1 清除标志。
6	CMDRECVC	CMDRECV 标志清除位 写 1 清除标志。
5	RXOREC	RXORE 标志清除位 写 1 清除标志。
4	TXUREC	TXURE 标志清除位 写 1 清除标志。
3	DTTMOUTC	DTTMOUT 标志清除位 写 1 清除标志。
2	CMDTMOUTC	CMDTMOUT 标志清除位 写 1 清除标志。
1	DTCRCERRC	DTCRCERR 标志清除位 写 1 清除标志。
0	CCRCERRC	CCRCERR 标志清除位 写 1 清除标志。

36.7.13. 中断使能寄存器（SDIO_INTEN）

地址偏移：0x3C
复位值：0x0000 0000

该寄存器使能 SDIO_STAT 寄存器中相应状态位的中断。

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			IDMAENDIE	IDMAERRIE	CLKSTOPIE	VSENDIE	ACKTOIE	ACKFAILIE	SDIOINTIE	DAT0BSYENDIE	保留		TFEIE	RFFIE	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

RFHIE	TFHIE	保留	DATABORIE	DTBLKENDIE	DATHOLDIE	DTENDIE	CMDSENDIE	CMDRECVIE	RXOREIE	TXUREIE	DTTMOUTIE	CMDTMOUTIE	DTCRCERRIE	CCRCERRIE
rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:29	保留	必须保持复位值
28	IDMAENDIE	IDMA 传输结束中断使能 写 1 使能中断。
27	IDMAERRIE	IDMA 传输错误中断使能 写 1 使能中断。
26	CLKSTOPIE	电压切换时钟停止中断使能 写 1 使能中断。
25	VSENDIE	电压切换关键时序结束中断使能 写 1 使能中断。
24	ACKTOIE	引导确认超时使能 写 1 使能中断。
23	ACKFAILIE	引导确认接收和检验失败中断使能 写 1 使能中断。
22	SDIOINTIE	SD I/O 中断已接收中断使能 写 1 使能中断。
21	DAT0BSYENDIE	DAT0 线繁忙结束中断使能 写 1 使能中断。
20:19	保留	必须保持复位值
18	TFEIE	发送 FIFO 空中断使能 写 1 使能中断。
17	RFFIE	接收 FIFO 满中断使能 写 1 使能中断。
16	保留	必须保持复位值
15	RFHIE	接收 FIFO 半满中断使能 写 1 使能中断。
14	TFHIE	发送 FIFO 半空中断使能 写 1 使能中断。
13:12	保留	必须保持复位值
11	DATABORIE	数据传输中止中断使能 写 1 使能中断。
10	DTBLKENDIE	数据块已发送/已接收中断使能

		写 1 使能中断。
9	DATHOLDIE	数据保持中断使能 写 1 使能中断。
8	DTENDIE	数据结束中断使能 写 1 使能中断。
7	CMDSENDIE	命令已发送中断使能 写 1 使能中断。
6	CMDRECVIE	命令响应已接收中断使能 写 1 使能中断。
5	RXOREIE	接收 FIFO 上溢错误中断使能 写 1 使能中断。
4	TXUREIE	发送 FIFO 下溢错误中断使能 写 1 使能中断。
3	DTTMOUTIE	数据超时中断使能 写 1 使能中断。
2	CMDTMOUTIE	命令响应超时中断使能 写 1 使能中断。
1	DTCRCERRIE	数据 CRC 错误中断使能 写 1 使能中断。
0	CCRCERRIE	命令响应 CRC 错误中断使能 写 1 使能中断。

36.7.14. ACK 超时寄存器（SDIO_ACKTO）

地址偏移：0x40
复位值：0x0000 0000

当DSM进入WaitAck状态时，该寄存器的值将装载到计数器中，用于计数引导确认是否超时。如果计数器计数到0时，ACK超时状态标志会置位。

数据传输时，必须先设置SDIO_ACKTO寄存器，然后再设置SDIO_DATACTL寄存器。

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:25	保留	必须保持复位值。
24:0	ACKTO[24:0]	引导确认超时时间 只有在 CSMEN 位清零时，该位才能被固件写入。

36.7.15. FIFO 数据寄存器（SDIO_FIFO）

地址偏移：0x80
复位值：0x0000 0000

该寄存器占用了 16 个 32 位的字，地址偏移从 0x80 到 0xBC。

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:0	FIFODT[31:0]	接收 FIFO 数据或发送 FIFO 数据 这些位为接收 FIFO 或发送 FIFO 的数据。读或写该寄存器相当于对 FIFO 读或写数据。 只有在 DATSTA 位置位时，该位才能被固件写入。

36.7.16. 内部 DMA（IDMA）控制寄存器（SDIO_IDMACTL）

地址偏移：0x50
复位值：0x0000 0000

该寄存器作为发送或接收 FIFO 只能按字（32 位）进行访问。

FIFO 包含 32 个连续地址作为 32 个入口。允许 CPU 使用该寄存器存储/装载多操作命令去写/读 FIFO。

31	30	29	28	27	26	25	24	23	22	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													BUFSEL	BUFMOD	IDMAEN
													rw	rw	rw

位/位域	名称	描述
31:3	保留	必须保持复位值。

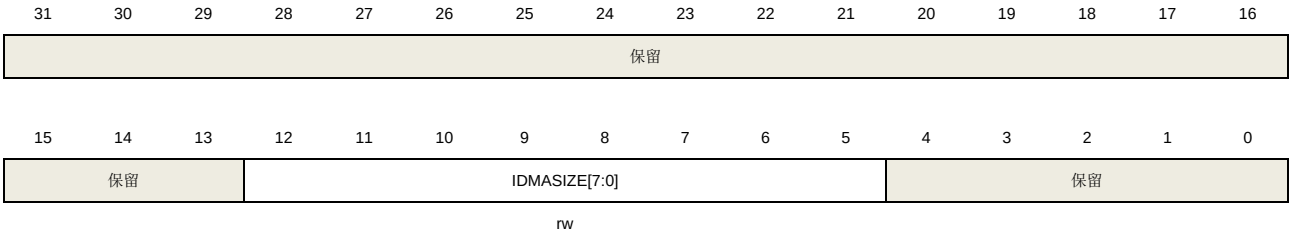
2	BUFSEL	IDMA 双缓冲区选择位 只有在 DATSTA 位清零时，该位才能被固件写入。 当 IDMA 使能时，该位由硬件翻转。 0：IDMA 使能时，使用缓冲区 0，禁止固件对 IDMAADDR0 进行写访问 1：IDMA 使能时，使用缓冲区 1，禁止固件对 IDMAADDR1 进行写访问
1	BUFMOD	双缓冲区模式使能位 只有在 DATSTA 位清零时，该位才能被固件写入。 0：单缓冲区模式 1：双缓冲区模式
0	IDMAEN	FIFO 内部 DMA 使能位 只有在 DATSTA 位清零时，该位才能被固件写入。 0：IDMA 禁用 1：IDMA 使能

36.7.17. 内部 DMA（IDMA）缓冲大小寄存器（SDIO_IDMASIZE）

地址偏移：0x54
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

该寄存器包含双缓冲区模式下的缓冲区大小。



位/位域	名称	描述
31:13	保留	必须保持复位值。
12:5	IDMASIZE[7:0]	每个缓冲区的字节数 IDMA 区大小 = IDMASIZE[7:0]*8 字 0x00：缓冲区大小 = 0 字 0x01：缓冲区大小 = 8 字 0x02：缓冲区大小 = 16 字 …… 0xFF：缓冲区大小 = 2040 字
4:0	保留	必须保持复位值。

36.7.18. IDMA 缓冲区 0 基地址寄存器（SDIO_IDMAADDR0）

地址偏移：0x58

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

该寄存器包含单缓冲区配置下缓冲区的基地址，和双缓冲区配置的缓冲区 0 的基地址。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDMAADDR0[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDMAADDR0[15:0]															
rw															

位/位域	名称	描述
31:0	IDMAADDR0[31:0]	地址是 4 的倍数。IDMAADDR0[1:0]总是 0 且只能被读。 只有在 DATSTA 位清零时，该位域才能被固件写入。当 DATSTA 和 BUFSEL 都为 1 时，可以通过固件动态写入。

36.7.19. IDMA 缓冲区 1 基地址寄存器（SDIO_IDMAADDR1）

地址偏移：0x5C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

该寄存器包含双缓冲区配置的缓冲区 1 的基地址。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDMAADDR1[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDMAADDR1[15:0]															
rw															

位/位域	名称	描述
31:0	IDMAADDR1[31:0]	地址是 4 的倍数。IDMAADDR1[1:0]总是 0 且只能被读。 只有在 DATSTA 位清零时，该位域才能被固件写入。当 DATSTA 为 1 且 BUFSEL 都为 0 时，可以通过固件动态写入。

37. 管理数据输入/输出接口（MDIO）（适用于GD32H737_757_759）

37.1. 简介

MDIO 接口可以接收完整的 MDIO 帧。只要在接收到读数据帧或读后增量地址帧的转换位（TA）之前将数据写入寄存器，MDIO 接口就可以传输完整的 MDIO 帧。中断在每个完整帧的末尾生成，可以在正确的时间使用或提供中断。中断也可以在每个有效的 PHYADR 和 DEVADD 之后生成，这支持了在帧内进行更复杂的控制。

37.2. 主要特性

- 支持从机模式下最高频率4MHz；
- 支持CFP / CFP2 MSA管理接口规范；
- 支持多种中断；
- 物理地址可配置：
  - 软件配置；
  - 硬件引脚配置。

37.3. 引脚和内部信号

*图37-1. CFP管理接口结构*提供了MDIO结构框图。*表37-1. MDIO引脚定义*提供了MDIO内部信号和引脚介绍。

表37-1. MDIO引脚定义

名称	信号类型
MDIO_MDIO	输入输出，数字信号
MDIO_MDC	输入，数字时钟信号
MDIO_Ax(x=0…4)	输入，数字地址信号

37.4. 功能描述

图37-1. CFP管理接口结构

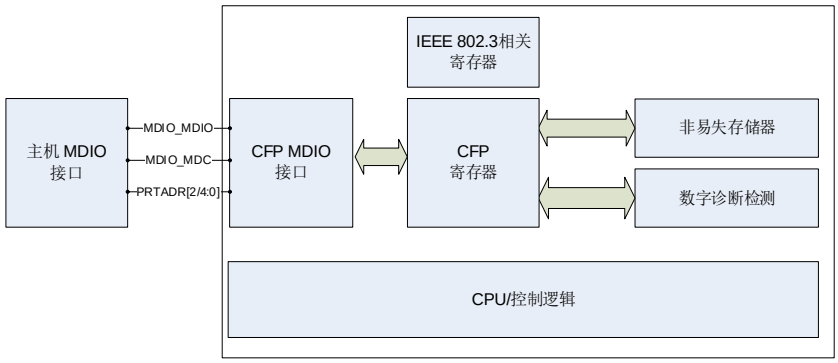
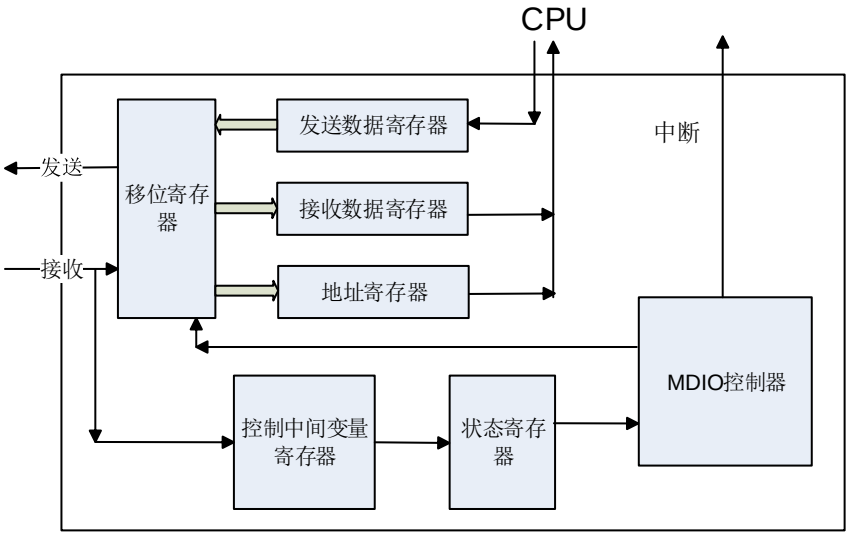


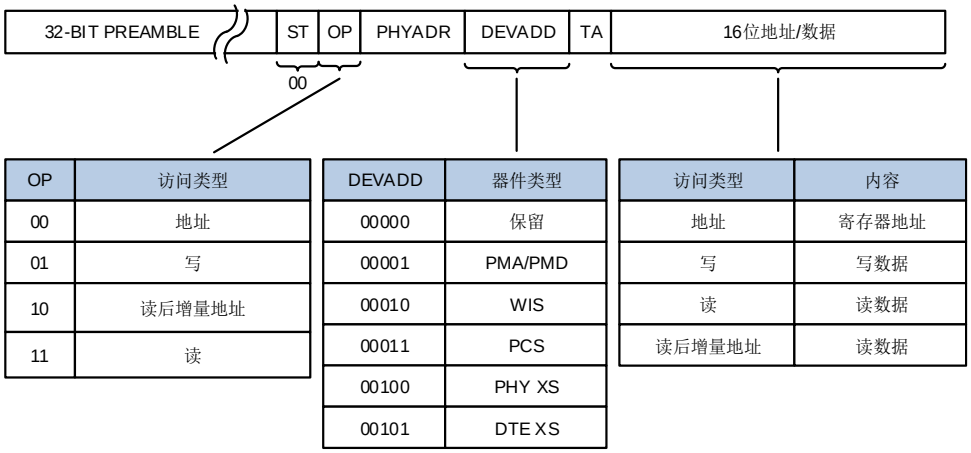
图37-2. MDIO框图



37.4.1. 帧结构

CFP MDIO 接口的通讯数据帧格式符合 IEEE 802.3 标准的第 45 条。帧可以是地址帧或者数据帧。帧总长度为 64 位，包括 32 位的前导码，以及帧命令主体。帧命令主体包括 6 个部分，详见*图 37-3. MDIO 帧格式*。

图37-3. MDIO帧格式



OP	访问类型
00	地址
01	写
10	读后增量地址
11	读

DEVADD	器件类型
00000	保留
00001	PMA/PMD
00010	WIS
00011	PCS
00100	PHY XS
00101	DTE XS

访问类型	内容
地址	寄存器地址
写	写数据
读	读数据
读后增量地址	读数据

注意：

1. ST = 开始位（2 位）
2. OP = 操作码（2 位）
3. PHYADR = 物理端口地址（5 位）
4. DEVADD = MDIO 器件地址（或称为器件类型，5 位）
5. TA = 转换位（2 位）
6. 16 位地址/数据为帧数据

表37-2. 不同帧类型具体描述(1)

帧	Idle	管理帧域							Idle
		PRE	ST	OP	PHYADR	DEVADD	TA	Address/Data	
写地址	Z	1…1	00	00	aaaaa	aaaaa	10	aaaaaaaaaaaaaaaa	Z
写数据	Z	1…1	00	01	aaaaa	aaaaa	10	dddddddddddddddd	Z
读数据	Z	1…1	00	11	aaaaa	aaaaa	z0	dddddddddddddddd	Z
读后增量地址帧	Z	1…1	00	10	aaaaa	aaaaa	z0	dddddddddddddddd	Z

(1).在空闲状态期间，MDIO 接口时钟（MDC 信号）和 MDIO 信号无源驱动。读数据帧或者读后增量地址帧的 TA 域为 1.5 位，TA 域的前 0.5 位时钟周期，MDIO 信号由主机来驱动，TA 域的后 1 位时钟周期以及在 16 位读数据期间，MDIO 信号由 MDIO 管理的器件（即从机）来驱动，随后还会有 0.5 位周期的 TA 转换位，MDIO 信号由主机来驱动。在其他时候，MDIO 信号由 STA 即主机来驱动。

空闲状态（Idle）

MDIO 信号的空闲状态为高阻态。

前导码（PRE）

在每次传输的开始，站管理者（主机）一次发送一串不少于 32 位（对应 MDC 脚上 32 个 MDIO 时钟）连续的’1’到 MDIO 脚，来建立一个帧的开始。

帧的开始（ST）

在前导码后，帧开始位（包含 2 位的’0’）指示了帧的起始。

操作码（OP）

操作码描述了要求的动作，详见*表 37-3. 操作码*。

表37-3. 操作码

OP	描述
00	为后续的写数据或读数据帧设置地址
01	写数据到之前设置的地址
11	从之前设置的地址读取数据
10	从之前设置的地址读取数据，然后增加地址。用户代码必须增加地址。

物理地址（PHYADR）

5 位物理地址，允许最多 32 个不同的地址。PHYADR 可由 5 个硬件引脚或者软件来设置。

器件地址（DEVADD）

5 位器件地址用于器件类型选择。在 CFP 规范中，仅支持 MDIO 器件地址为 1。

转换位（TA）

在 TA 转换位期间，MDIO 信号由主机驱动转换为从机驱动。

地址 / 数据

地址 / 数据域为 16 位。

37.4.2. 典型使用流程

MDIO 接口大部分在硬件上实现，需要正确的软件操作顺序。使用流程如下所示：

1. 复位 MDIO 模块，配置 GPIO 模块，将相应的功能脚映射到复用功能上。
2. 通过写 MDIO_CTL 和 MDIO_CFG 寄存器来配置帧参数。并根据需求来设置 MDIO_TO 寄存器。
3. 通过写 MDIO_INTEN 寄存器和要求的系统中断设置来设置中断。
4. 对于一个写操作帧，用户可以在接收完整个帧后从 MDIO_RADDR 和 MDIO_RDATA 寄存器分别取得地址和写的数据。对于一个读数据帧或者读后增量地址帧，用户必须在该帧取数之前将数据放到 MDIO_TDATA 以便使该数据能够自动插入到帧里。

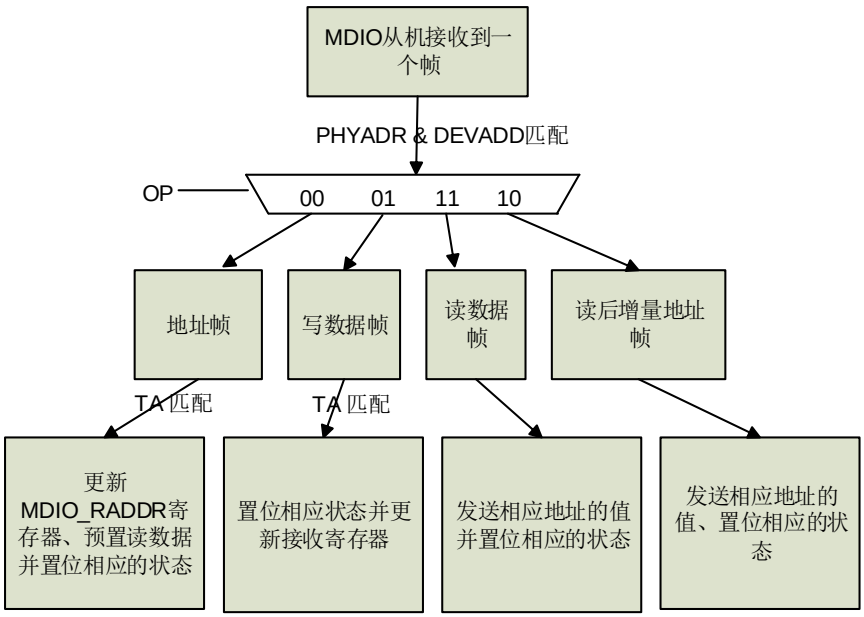
在这个过程中，不需要软件来干预。当发生帧的域匹配或不匹配事件时，可以在帧期间或结束的时候通过 MDIO_RFRM 寄存器来监测帧进程。由于 MDIO_STAT 的位[9：0]是自动清零的，所以不要使用 MDIO_STAT 来监测帧进程，在不恰当的时间读取 MDIO_STAT 寄存器，寄存器数据可能会丢失。每帧只读 MDIO_STAT 寄存器一次。为了监测帧进程，可以根据中断或轮询 NVIC 中 SETPEND1 寄存器的位 8（中断号 40）来读取 MDIO_STAT 中状态。MDIO 中断优先级必须高于所有其他外设，以避免数据丢失。

注意：

1. 系统时钟频率是 MDC 时钟的三倍以上。
2. 请务必确保发送数据在 TA 之前给到 MDIO_TDATA 寄存器。

3. 如果出现超时等错误情况请配置软复位后再使用。

图37-4. MDIO从机通讯流程



在接收到主机发送的写地址帧以后，无论下一帧是写还是读，都可以提前把数据预置到发送数据寄存器里，如果下一帧是读操作的话 MDIO 会将预置的数据发出去。如果给了写地址帧，但没有进行读操作，可在接受完下一次的地址帧之后将新的数据覆盖原来的发送寄存器。

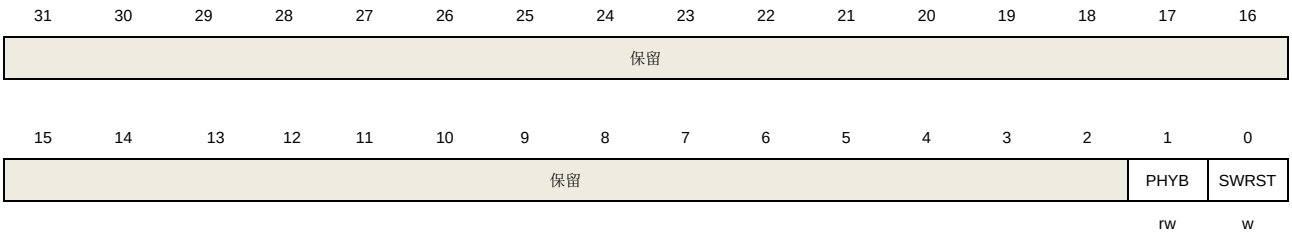
37.5. MDIO 寄存器

MDIO 基地址：0x4000 D800

37.5.1. 控制寄存器（MDIO_CTL）

地址偏移：0x00
复位值：0x0000 XXXX

该寄存器可以按字（32位）访问。

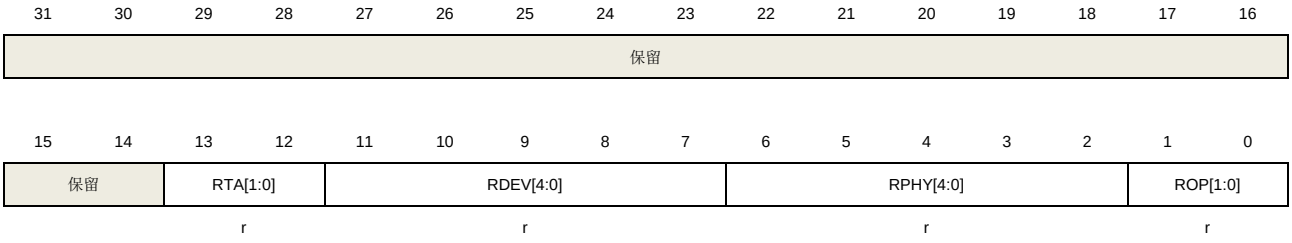


位/位域	名称	描述
31:2	保留	必须保持复位值。
1	PHYB	MDIO PHY 位长度 0: MDIO PHY 为 5 位 1: MDIO PHY 为 3 位，未使用的 PHY 位被忽略。
0	SWRST	写 1 以复位 MDIO 模块。寄存器不会被复位。硬件自动清零该位。

37.5.2. 接收帧信息寄存器（MDIO_RFRM）

地址偏移：0x04
复位值：0x0000 0000

该寄存器可以按字（32位）访问。



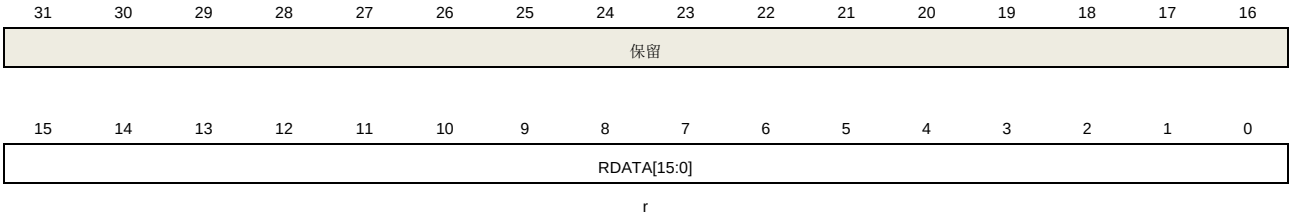
位/位域	名称	描述
31:14	保留	必须保持复位值。
13:12	RTA[1:0]	接收到的帧 TA 位信息（只支持写数据帧或者写地址帧的 TA 位）
11:7	RDEV[4:0]	接收到的帧 DEVADD 位信息

6:2	RPHY[4:0]	接收到的帧 PHYADR 位信息
1:0	ROP[1:0]	接收到的帧 OP 位信息 00：写地址帧 01：写数据帧 10：读后增量地址帧 11：读数据帧

37.5.3. 数据接收寄存器（MDIO_RDATA）

地址偏移：0x08
复位值：0x0000 0000

该寄存器可以按字（32位）访问。

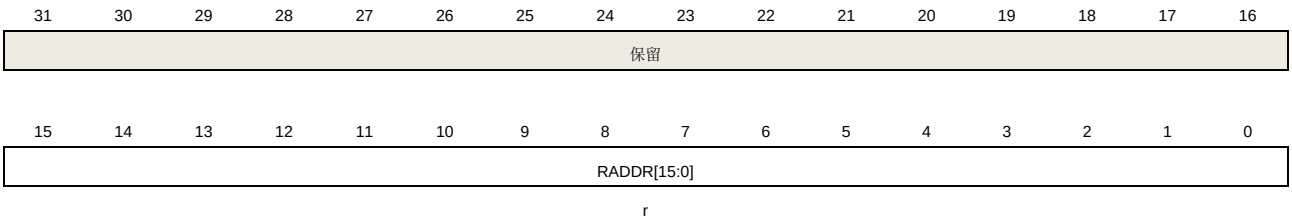


位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	RDATA[15:0]	接收到的帧 DATA 位数据

37.5.4. 地址接收寄存器（MDIO_RADDR）

地址偏移：0x0C
复位值：0x0000 0000

该寄存器可以按字（32位）访问。



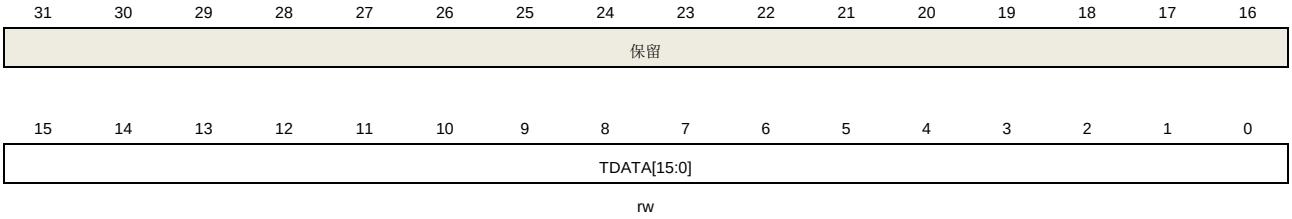
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	RADDR[15:0]	接收到的帧 ADDRESS 位数据

37.5.5. 数据发送寄存器（MDIO_TDATA）

地址偏移：0x10

复位值：0x0000 0000

该寄存器可以按字（32位）访问。



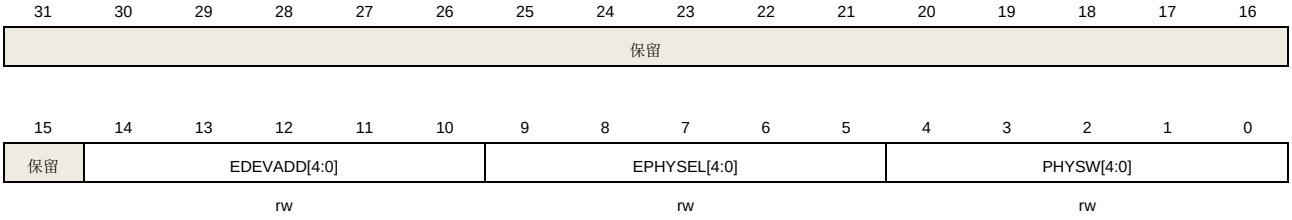
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	TDATA[15:0]	数据由后一个读数据帧或者读后增量地址帧发送。在一个读数据帧或者读后增量地址帧之前，主机发送一个写地址帧来指定要读取的数据。在这个地址帧之后，用户软件必须在读数据帧或者读后增量地址帧到来之前将要求的数据放置到 TDATA[15:0]位。最迟在读操作帧的 TA 位的前 3 个 MDIO 时钟周期之前，需完成这个动作。

37.5.6. 配置寄存器（MDIO_CFG）

地址偏移：0x14

复位值：0x0000 0000

该寄存器可以按字（32位）访问。



位/位域	名称	描述
31:15	保留	必须保持复位值。
14:10	EDEVADD[4:0]	期望的 DEVADD 值。通常为 01。
9:5	EPHYSEL[4:0]	选择期望的 PHYADR，对于 5 位 EPHYSEL 中的每一位 x： 0：设置期望的 PHYADR.x =PHYPIN[4:0].x 1：设置期望的 PHYADR.x = PHYSW[4:0].x
4:0	PHYSW[4:0]	软件配置的 PHYADR。根据 EPHYSEL[4:0]位域值选择 PHYADR。

37.5.7. 状态寄存器（MDIO_STAT）

地址偏移：0x18

复位值：0x0000 1000

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		RBNE	保留	OVR	UDR	TO	TANM	PHYNM	PHYM	DEVNM	DEVM	RDFRM	RDINCFRM	ADDRFRM	WRFRM
		r		r	r	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc

位/位域	名称	描述
31:14	保留	必须保持复位值。
13	RBNE	数据接收缓冲区非空标志。已接收到数据并可以读取。读 MDIO_RDATA 寄存器清零该位。
12	保留	必须保持复位值。
11	OVR	数据接收上溢标志。在 RBNE 位置位的情况下，如果接收移位寄存器的数据传递给 MDIO_RDATA 寄存器，将会由硬件置位。读 MDIO_RDATA 寄存器清零该位。
10	UDR	数据发送下溢标志。在数据发送缓冲区为空的情况下，如果在读数据帧的 TA 位的前 3 个 MDIO 时钟周期之前没有将数据写到 MDIO_TDATA 寄存器，将会由硬件置位。写 MDIO_TDATA 寄存器清零该位。
9	TO	超时标志。在一个完整的帧的两个位之间的时间，包括前导码到数据部分，如果已达到设置的超时时间，仍没有收到新的位或者准备的数据位未发送出去，将会由硬件置位。读 MDIO_STAT 寄存器清零该位。
8	TANM	在写地址/写数据帧的 TA 最后一位接收完时，如果接收到的 TA，与期望的‘10’不匹配，则置位该位。读 MDIO_STAT 寄存器清零该位。
7	PHYNM	在帧的 PHYADR 最后一位接收完时，如果接收到的 PHYADR 与配置的期望 PHYADR 不匹配，则置位该位。读 MDIO_STAT 寄存器清零该位。
6	PHYM	在帧的 PHYADR 最后一位接收完时，如果接收到的 PHYADR 与配置的期望 PHYADR 相匹配，则置位该位。读 MDIO_STAT 寄存器清零该位。
5	DEVNM	在帧的 DEVADD 最后一位接收完时，如果接收到的 DEVADD 与配置的期望 DEVADD 不匹配，则置位该位。读 MDIO_STAT 寄存器清零该位。
4	DEVM	在帧的 DEVADD 最后一位接收完时，如果接收到的 DEVADD 与配置的期望 DEVADD 相匹配，则置位该位。读 MDIO_STAT 寄存器清零该位。
3	RDFRM	在读数据帧的最后一位（帧 DATA 位发送完成之后）接收完时，如果接收到的 DEVADD 和 PHYADR，与配置的期望 DEVADD 和期望 PHYADR 相匹配，则置位该位。读 MDIO_STAT 寄存器清零该位。
2	RDINCFRM	在读后增量地址帧的最后一位（帧 DATA 位发送完成之后）接收完时，如果接收到的 DEVADD 和 PHYADR，与配置的期望 DEVADD 和期望 PHYADR 相匹配，则置位该位。读 MDIO_STAT 寄存器清零该位。
1	ADDRFRM	在地址帧的最后一位接收完时，如果接收到的 DEVADD 和 PHYADR，与配置的期

		望 DEVADD 和期望 PHYADR 相匹配，则置位该位。读 MDIO_STAT 寄存器清零该位。
0	WRFRM	在写数据帧的最后一位接收完时，如果接收到的 DEVADD 和 PHYADR，与配置的期望 DEVADD 和期望 PHYADR 相匹配，则置位该位。读 MDIO_STAT 寄存器清零该位。

37.5.8. 中断使能寄存器 (MDIO_INTEN)

地址偏移：0x1C
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	120	19	18	17	16
保留															

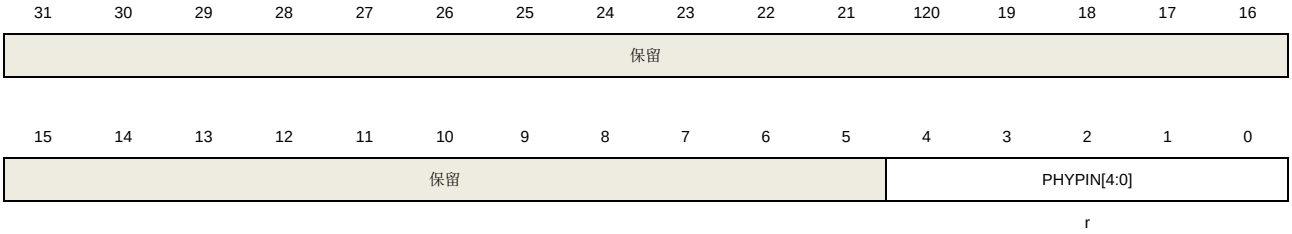
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		RBNEIE	保留	OVRIE	UDRIE	TOIE	TANMIE	PHYNMIE	PHYMIE	DEVNMIE	DEVMIE	RDFRMIE	RDINCFRMIE	ADDRFRMIE	WRFRMIE
		rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:14	保留	必须保持复位值。
13	RBNEIE	若置位该位，则当 MDIO_STAT 中的 RBNE 位变为有效时产生中断请求。
12	保留	必须保持复位值。
11	OVRIE	若置位该位，则当 MDIO_STAT 中的 OVR 位变为有效时产生中断请求。
10	UDRIE	若置位该位，则当 MDIO_STAT 中的 UDR 位变为有效时产生中断请求。
9	TOIE	若置位该位，则当 MDIO_STAT 中的 TO 位变为有效时产生中断请求。
8	TANMIE	若置位该位，则当 MDIO_STAT 中的 TANM 位变为有效时产生中断请求。
7	PHYNMIE	若置位该位，则当 MDIO_STAT 中的 PHYNM 位变为有效时产生中断请求。
6	PHYMIE	若置位该位，则当 MDIO_STAT 中的 PHYM 位变为有效时产生中断请求。
5	DEVNMIE	若置位该位，则当 MDIO_STAT 中的 DEVNM 位变为有效时产生中断请求。
4	DEVMIE	若置位该位，则当 MDIO_STAT 中的 DEVM 位变为有效时产生中断请求。
3	RDFRMIE	若置位该位，则当 MDIO_STAT 中的 RDFRM 位变为有效时产生中断请求。
2	RDINCFRMIE	若置位该位，则当 MDIO_STAT 中的 RDINCFRM 位变为有效时产生中断请求。
1	ADDRFRMIE	若置位该位，则当 MDIO_STAT 中的 ADDRFRM 位变为有效时产生中断请求。
0	WRFRMIE	若置位该位，则当 MDIO_STAT 中的 WRFRM 位变为有效时产生中断请求。

37.5.9. 引脚数值寄存器（MDIO_PIN）

地址偏移：0x20
复位值：0x0000 0000

该寄存器可以按字（32位）访问。



位/位域	名称	描述
31:5	保留	必须保持复位值。
4:0	PHYPIN[4:0]	读取的硬件引脚 MDIO_Ax(x=0…4)的数值

37.5.10. 超时寄存器（MDIO_TO）

地址偏移：0x24
复位值：0x0000 0000

该寄存器可以按字（32位）访问。



位/位域	名称	描述
31:17	保留	必须保持复位值。
16:1	TOCNT[15:0]	MDIO 超时=TOCNT[15:0]*PCLK1
0	TOEN	使能超时 0：超时失能 1：超时使能

38. 外部存储器控制器（EXMC）

38.1. 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AXI协议转换为专用的片外存储器通信协议，包括SRAM，ROM，NOR Flash，NAND Flash和SDRAM。用户还可以调整相关的时间参数来提高通信效率。EXMC模块划分为许多个子Bank，每个Bank支持特定的存储器类型，用户可以通过对Bank的寄存器配置来控制外部存储器。

38.2. 主要特性

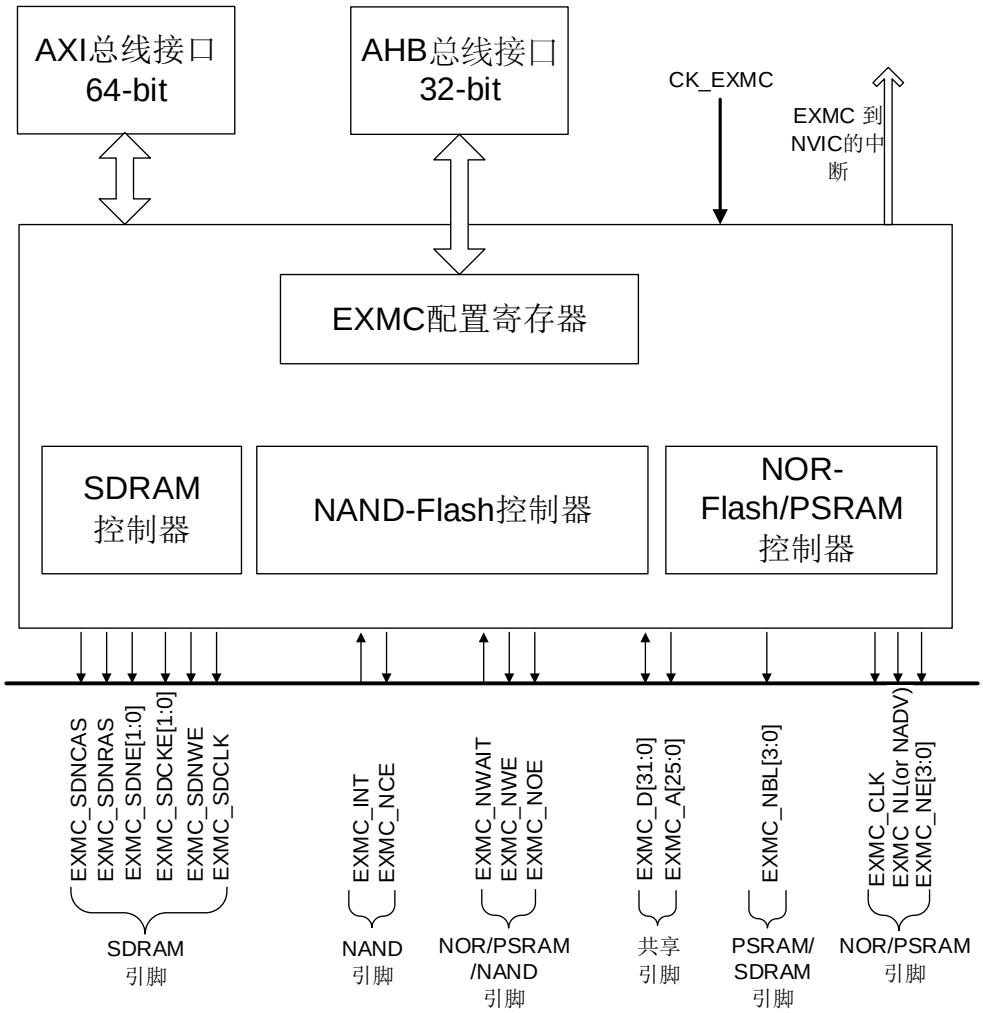
- 支持的片外存储器类型：
  - SRAM
  - PSRAM
  - ROM
  - NOR Flash
  - 8位或16位NAND Flash
  - SDRAM
- AXI协议与各种片外存储器协议转换；
- 时序参数可编程可以满足用户特定需求；
- 每个Bank有独立的片选信号；
- 对于部分存储器类型支持独立的读写时序；
- 对于NAND Flash内置硬件ECC；
- 支持8位，16位，32位总线带宽；
- NOR Flash和PSRAM支持地址总线和数据总线的复用；
- 提供写使能和字节选择信号；
- 当AXI总线宽度与外部存储器数据宽度不同时，会自动分割操作。

38.3. 功能描述

38.3.1. 结构框图

EXMC由7个模块组成：AHB总线接口，AXI总线接口，EXMC配置寄存器，NOR/PSRAM控制器，NAND控制器，SDRAM控制器和外部设备接口。AHB时钟（HCLK）是参考时钟，用于配置EXMC寄存器。

图 38-1. 系统架构



注意：GD32H75E 系列并非所有设备都支持 EXMC_NE3、EXMC_A0-A4 和 EXMC_A10-A15。详情请参考数据手册。

38.3.2. 总线接口

AHB 总线接口：CPU 通过 AHB 从接口配置 EXMC 寄存器。

AXI 总线接口：CPU 和 AXI 总线主设备通过 AXI 总线从接口访问外部存储器。

NOR、NAND、SDRAM 控制器的时钟是异步 CK_EXMC 始时钟（具体参考***时钟配置寄存器 4（RCU_CFG4）***）。

38.3.3. AXI 错误

访问未使能的 EXMC BANK region x (x = 0,.., 3)将产生 AXI 从错误。

如果 EXMC_SNCTLx (x = 0,.., 3) 寄存器中的 NREN 位被置为 0，访问 EXMC NOR Flash 区将产生 AXI 从错误。

对已被写保护的 SDRAM 设备（WPEN 设置为 1）进行写操作，会产生 AXI 从机错误。

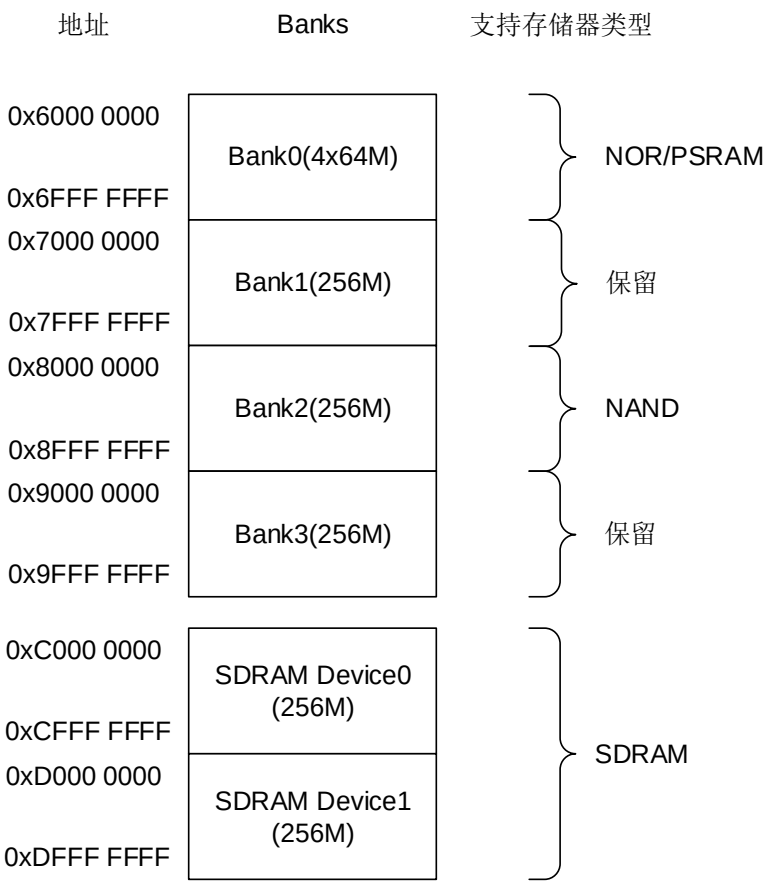
38.3.4. EXMC 访问基本规范

EXMC是AXI总线至外部设备协议的转换接口。由于AXI数据总线的位宽为64位，因此AXI事务会根据数据大小将一次访问拆分为多个连续的8位、16位或32位访问。在数据传输的过程中，AXI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：

- AXI事务数据宽度等于存储器宽度，则没有问题；
- AXI事务数据宽度大于存储器宽度，则自动将AXI访问分割成几个连续的存储器数据宽度的传输；
- AXI事务数据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、SDRAM，则可通过它的字节通道EXMC_NBL[3:0]来访问对应的字节。否则禁止写操作，只允许读操作。

38.3.5. 外部设备地址映射

图 38-2. EXMC Bank 划分



EXMC将外部存储器分成多个Bank，每个Bank占256M字节，其中Bank0又分为4个Region，每个Region占64M字节。Bank2又都被分成2个空间，分别是属性存储空间和通用存储空间。

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。

Bank2用于连接NAND Flash。

SDRAM Device0和SDRAM Device1用于连接SDRAM。

EXMC bank映射可以通过EXMC_SNCTL寄存器中的BKREMAP[1:0]位域进行修改。EXMC bank映射如*表38-1. EXMC bank映射*所示。

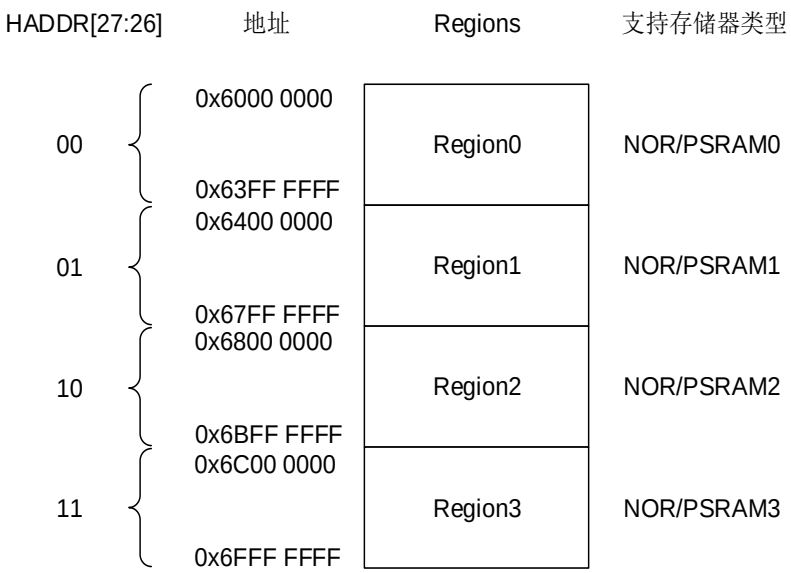
表 38-1. EXMC bank 映射

地址	BKREMAP[1:0]=00	BKREMAP[1:0]=01
0x6000 0000 – 0x6FFF FFFF	NOR/PSRAM bank	SDRAM Device 0
0x7000 0000 – 0x7FFF FFFF	保留	
0x8000 0000 – 0x8FFF FFFF	NAND bank	
0x9000 0000 – 0x9FFF FFFF	保留	
0xC000 0000 – 0xCFFF FFFF	SDRAM Device 0	NOR/PSRAM bank
0xD000 0000 – 0xDFFF FFFF	SDRAM Device 1	

NOR/PSRAM 的地址映射

*图 38-3. Bank0 地址映射*是 Bank0 四个 Region 的地址映射。AXI 地址线 HADDR[27:26]作为四个 Region 的片选信号。

图 38-3. 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：

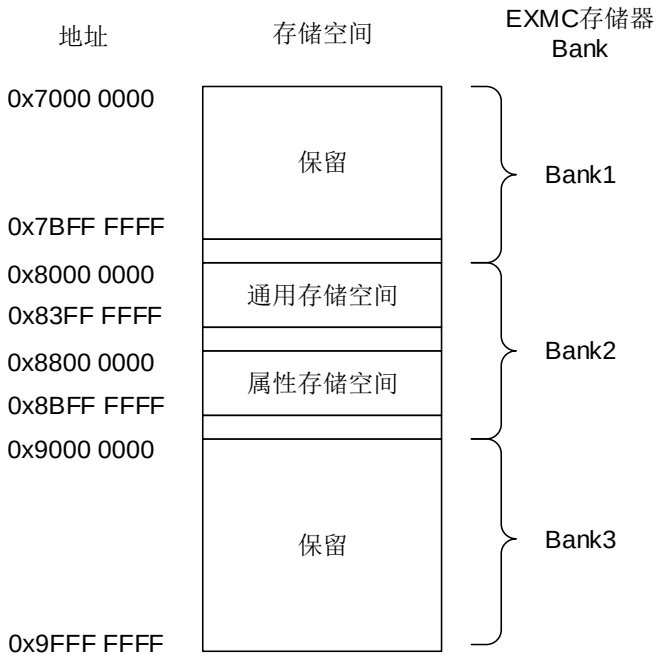
- 如果外部存储器的数据宽度是 8 位按字节对齐，EXMC 内部将 HADDR[25:0]与 EXMC_A[25:0]相连，然后 EXMC_A[25:0]与外部存储器的地址线相连；

- 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；
- 如果外部存储器的数据宽度是 32 位按字对齐，就需要将 HADDR 的字节地址转化为字地址之后再连接外存储器。EXMC 内部将 HADDR[25:2]与 EXMC_A[23:0]相连，然后 EXMC_A[23:0]与外部存储器的地址线相连。

NAND 地址映射

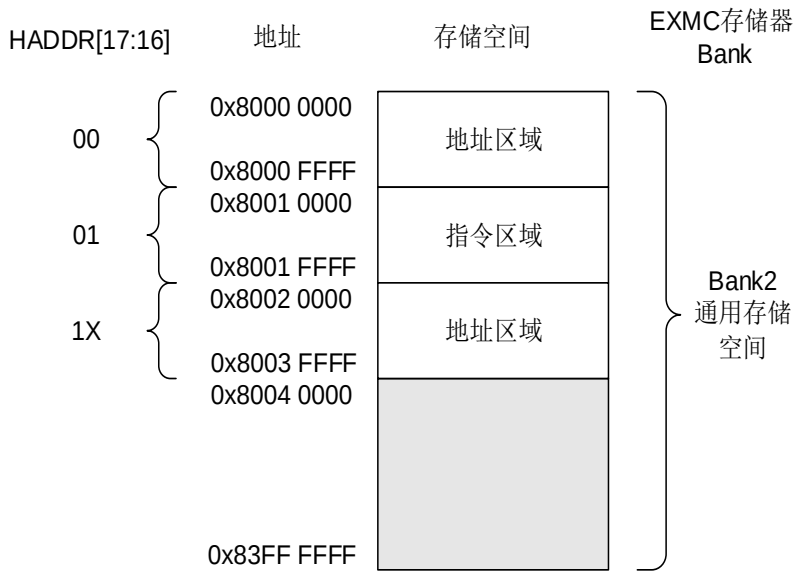
Bank2用来访问NAND Flash，Bank1和Bank3保留。Bank如*图38-4. NAND地址映射*被分为多个存储空间。

图 38-4. NAND 地址映射



对于NAND FLASH，通用和属性空间又可以细划分为3个区域。*图38-5. Bank2通用空间*为Bank2通用存储空间的数据区域，指令区域和地址区域的划分。

图 38-5. Bank2 通用空间



利用HADDR[17:16]来实现对以上三个区的选择：

- HADDR[17:16]=00，即选择数据区；
- HADDR[17:16]=01，即选择命令区；
- HADDR[17:16]=1X，即选择地址区。

应用软件使用这3个区访问NAND FLASH。操作规则如下：

指令区：指定NAND FLASH将要执行的指令，软件在命令区写入指令。在指令传输过程中，EXMC会使能命令锁存信号（CLE），CLE映射到EXMC_A[16]。

地址区：指定操作NAND FLASH的地址，软件在地址区写入地址。在地址传输过程中，EXMC会使能地址锁存信号（ALE），ALE映射到EXMC_A[17]。

数据区：NAND FLASH读写数据，软件在数据区读出或写入数据。当EXMC在数据发送模式，软件需要在数据区写入数据，当EXMC在数据接收模式，软件需要在数据区读取数据。由于NAND FLASH会自动累加其内部操作地址，故在读写时不需要软件修改操作地址。

SDRAM 地址映射

HADDR[28]位用来选两个SDRAM Device，如*图38-6. SDRAM地址映射*所示。

图 38-6. SDRAM 地址映射

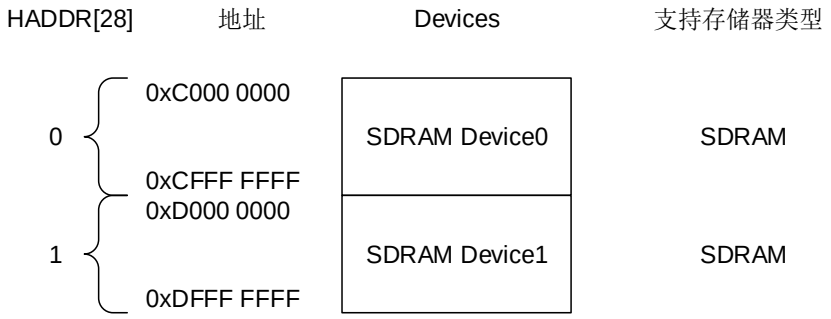


*表38-2. SDRAM地址映射*展示了SDRAM的13位行地址和11位列地址的配置映射：

表 38-2. SDRAM 地址映射

存储器数据宽度	内部 bank	行地址	列地址	最大存储容量
8-bit	HADDR[25:24]	HADDR[23:11]	HADDR[10:0]	64 Mbytes: 4 x 8K x 2K
16-bit	HADDR[26:25]	HADDR[24:12]	HADDR[11:1]	128 Mbytes: 4 x 8K x 2K x 2
32-bit	HADDR[27:26]	HADDR[25:13]	HADDR[12:2]	256 Mbytes: 4 x 8K x 2K x 4

38.3.6. 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AXI总线时钟（CK_EXMC）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 存储器接口描述

表 38-3. 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 引脚	传输方向	模式	功能描述
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 38-4. 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
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38-5. EXMC Bank0 支持的访问模式*列出了存储器数据总线为16位时EXMC对NOR，PSRAM和SRAM支持的访问模式。

表 38-5. EXMC Bank0 支持的访问模式

存储器类型	访问模式	读/写	AXI 事务宽度	存储器传输宽度	注释
NOR Flash	异步	R	8	16	
	异步	W	8	16	不允许
	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
	异步	R	64	16	分成 4 次 EXMC 访问
	异步	W	64	16	分成 4 次 EXMC 访问
	同步	R	8	16	不允许
	同步	R	16	16	
	同步	R	32	16	
	同步	R	64	16	
PSRAM	异步	R	8	16	
	异步	W	8	16	使用字节信号 EXMC_NBL[1:0]
	异步	R	16	16	
	异步	W	16	16	
	异步	R	32	16	分成 2 次 EXMC 访问

存储器类型	访问模式	读/写	AXI 事务宽度	存储器传输宽度	注释
	异步	W	32	16	分成 2 次 EXMC 访问
	异步	R	64	16	分成 4 次 EXMC 访问
	异步	W	64	16	分成 4 次 EXMC 访问
	同步	R	8	16	不允许
	同步	R	16	16	
	同步	R	32	16	
	同步	R	64	16	
	同步	W	8	16	使用字节信号 EXMC_NBL[1:0]
	同步	W	16	16	
	同步	W	32	16	
	同步	W	64	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	R	64	16	分成 4 次 EXMC 访问
	异步	W	8	16	使用字节信号 EXMC_NBL[1:0]
	异步	W	16	16	
	异步	W	32	16	使用字节信号 EXMC_NBL[1:0]
	异步	W	64	16	使用字节信号 EXMC_NBL[1:0]

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 38-6. 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	CK_EXMC	2	16
DLAT	数据延迟	同步	EXMC_CLK	2	17
BUSLAT	总线延迟	异步/同步读	CK_EXMC	0	15
DSET	数据建立时间	异步	CK_EXMC	1	255
AHLD	地址保持时间	异步（复用）	CK_EXMC	1	15
ASET	地址建立时间	异步	CK_EXMC	0	15

表 38-7. EXMC 时序模型

时序模型		扩展模式	模式描述	写时序参数	读时序参数
异步	模式 1	0	SRAM/PSRAM/CRAM	DSET ASET	DSET ASET
	模式 2	0	NOR Flash	DSET ASET	DSET ASET

时序模型		扩展模式	模式描述	写时序参数	读时序参数
	模式 A	1	SRAM/PSRAM/CRAM 在数据阶段 EXMC_OE 翻转	WDSET WASET	DSET ASET
	模式 B	1	NOR Flash	WDSET WASET	DSET ASET
	模式 C	1	NOR Flash 在数据阶段 EXMC_OE 翻转	WDSET WASET	DSET ASET
	模式 D	1	有地址保持功能	WDSET WAHLD WASET	DSET AHLD ASET
	模式 AM	0	NOR Flash 数据/地址复用	DSET AHLD ASET BUSLAT	DSET AHLD ASET BUSLAT
同步	模式 E	0	NOR/PSRAM/CRAM 同步读 PSRAM/CRAM 同步写	DLAT CKDIV	DLAT CKDIV
	模式 SM	0	NOR Flash 数据/地址复用	DLAT CKDIV	DLAT CKDIV

如*表38-7. 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38-6. 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 38-7. 模式 1 读时序

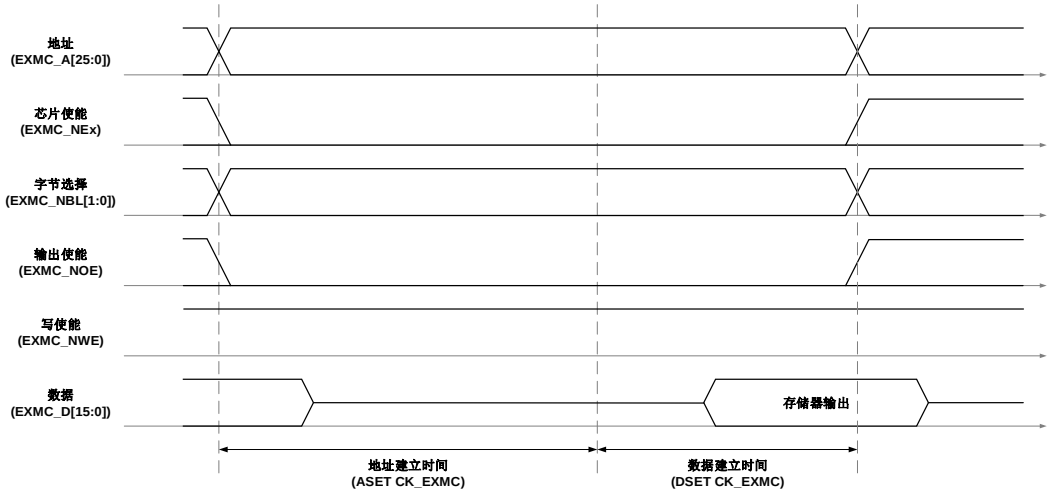


图 38-8. 模式 1 写时序

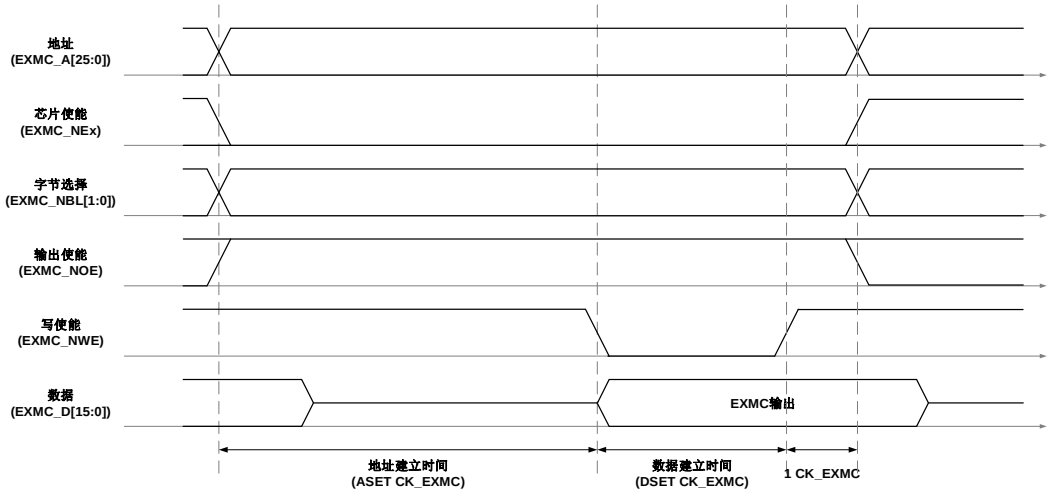


表 38-8. 模式 1 相关寄存器配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
31-21	保留	0x000
20	CCK	取决于存储器与用户
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWTEN	取决于存储器
14	EXMODEN	0x0
13	NRWTEN	0x0
12	WEN	取决于用户
11	NRWTCFG	无影响
10	保留	0x0
9	NRWTPOL	仅当位 15 为 1 时有效
8	SBRSTEN	0x0
7	保留	0x1
6	NREN	无影响
5-4	NRW	取决于存储器
3-2	NRTP	取决于存储器，除了 2（Nor Flash）
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFGx		
31-30	保留	0x0000
29-28	ASYNCMOD	无影响
27-24	DLAT	无影响
23-20	CKDIV	无影响
19-16	BUSLAT	EXMC_NE[x]上升沿到下降沿的时间
15-8	DSET	取决于存储器与用户（写操作为 DSET+1 CK_EXMC 时钟周期，读操作为 DSET CK_EXMC 时钟周期）

位域/位	名称	参考设定值
EXMC_SNCTLx		
7-4	AHLD	无影响
3-0	ASET	取决于存储器与用户

模式A - SRAM/PSRAM(CRAM) OE翻转

图 38-9. 模式 A 读时序

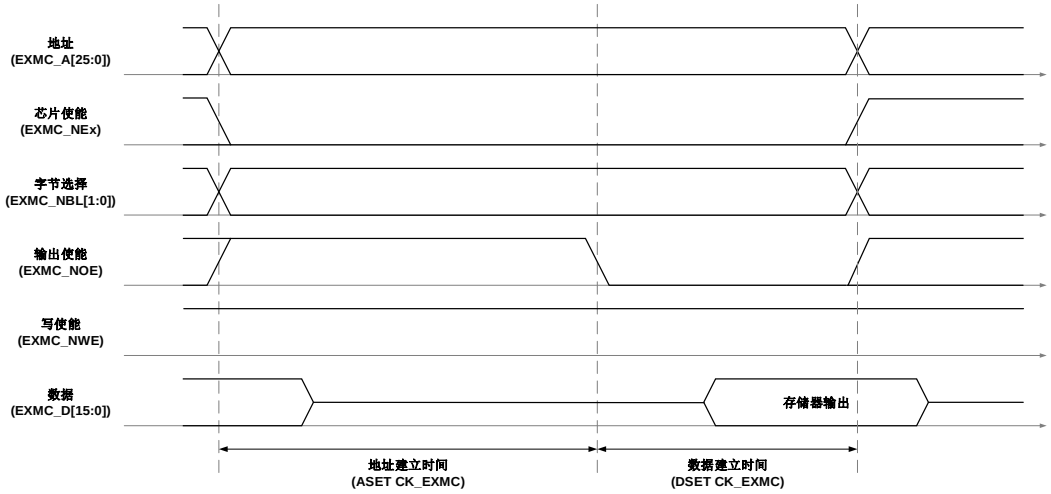
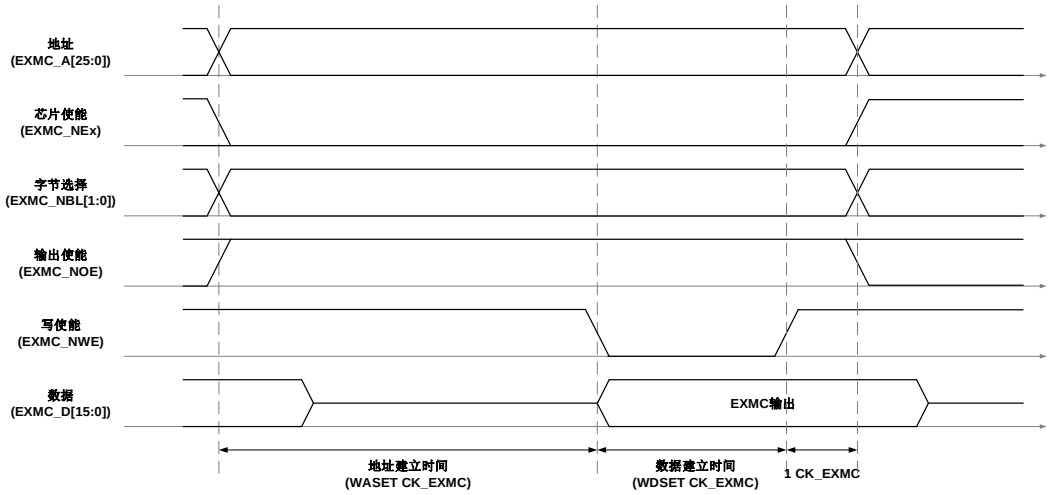


图 38-10. 模式 A 写时序



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 38-9. 模式 A 相关寄存器配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
31-21	保留	0x000
20	CCK	取决于存储器与用户
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWTEN	取决于存储器

位域/位	名称	参考设定值
EXMC_SNCTLx		
14	EXMODEN	0x1
13	NRWTEN	0x0
12	WEN	取决于用户
11	NRWTCFG	无影响
10	保留	0x0
9	NRWTPOL	仅当位 15 为 1 时有效
8	SBRSTEN	0x0
7	保留	0x1
6	NREN	无影响
5-4	NRW	取决于存储器
3-2	NRTP	取决于存储器，除了 2（Nor Flash）
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（写操作为 DSET+1 CK_EXMC 时钟周期，读操作为 DSET CK_EXMC 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	0x00
19-16	WBUSLAT	EXMC_NE[x]上升沿到下降沿的时间
15-8	WDSET	取决于存储器与用户
7-4	WAHLD	0x0
3-0	WASET	取决于存储器与用户

模式2/B - NOR Flash

图 38-11. 模式 2/B 读时序

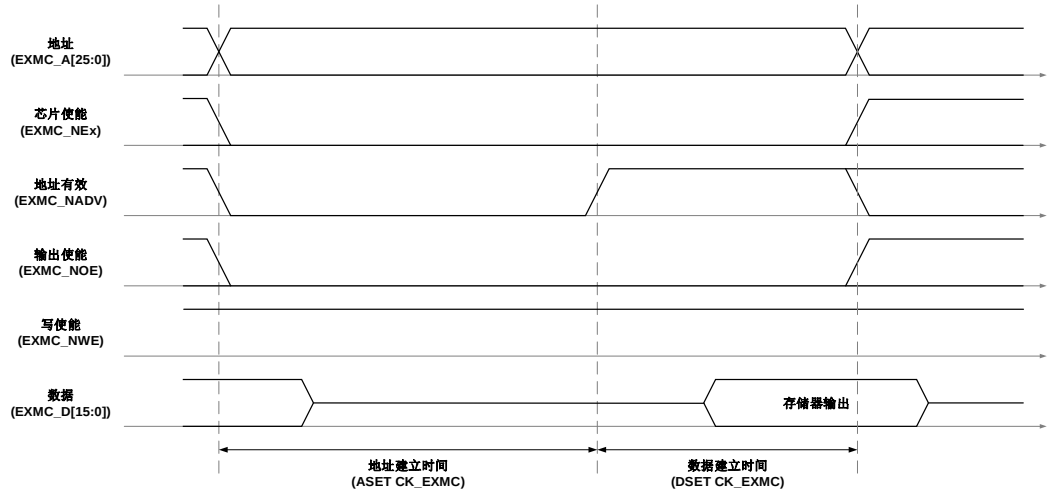


图 38-12. 模式 2 写时序

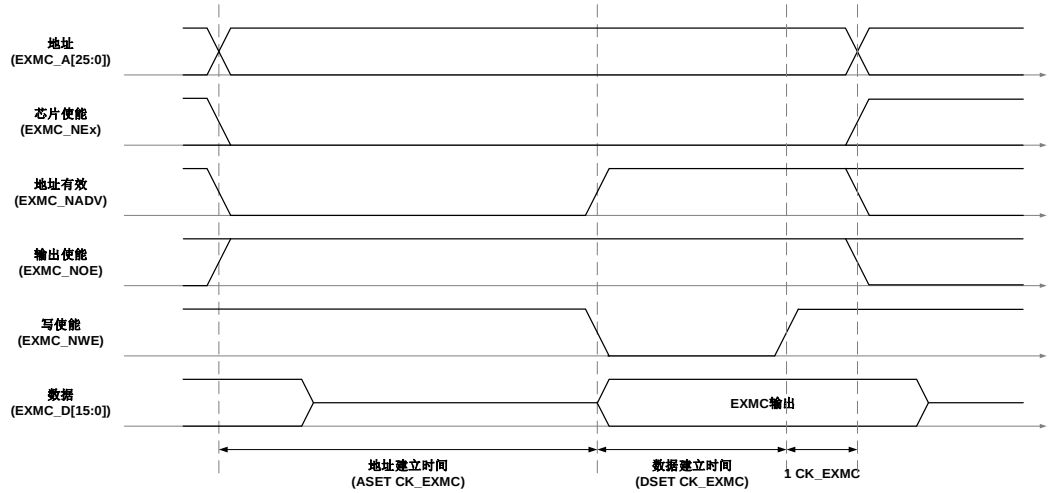


图 38-13. 模式 B 写时序

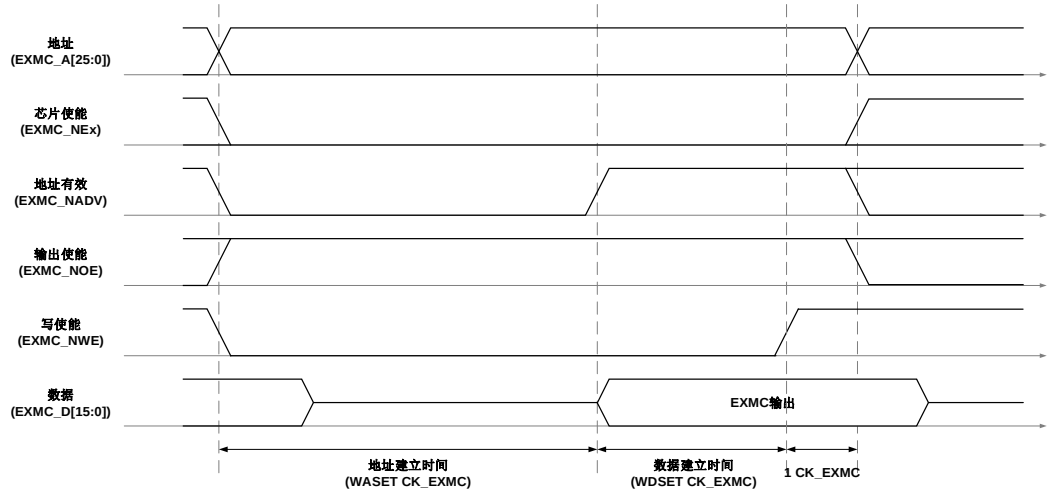


表 38-10. 模式 2/B 相关寄存器配置

位域/位	名称	参考设定值
EXMC_SNCTLx（模式 2，模式 B）		
31-21	保留	0x000

位域/位	名称	参考设定值
EXMC_SNCTLx（模式 2，模式 B）		
20	CCK	取决于存储器与用户
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWTEN	取决于存储器
14	EXMODEN	模式 2：0x0，模式 B：0x1
13	NRWTEN	0x0
12	WEN	取决于用户
11	NRWTCFG	无影响
10	保留	0x0
9	NRWTPOL	仅当位 15 为 1 时有效
8	SBRSTEN	0x0
7	保留	0x1
6	NREN	0x1
5-4	NRW	取决于存储器
3-2	NRTP	0x2，NOR Flash
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFGx（模式 2 读/写操作，模式 B 读操作）		
31-30	保留	0x0000
29-28	ASYNCMOD	模式 B：0x1
27-24	DLAT	无影响
23-20	CKDIV	无影响
19-16	BUSLAT	EXMC_NE[x]上升沿到下降沿的时间
15-8	DSET	取决于存储器与用户
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	0x0000
29-28	WASYNCMOD	模式 B：0x1
27-20	保留	0x000
19-16	WBUSLAT	EXMC_NE[x]上升沿到下降沿的时间
15-8	WDSET	取决于存储器与用户
7-4	WAHLD	0x0
3-0	WASET	取决于存储器与用户

模式C - NOR Flash OE翻转

图 38-14. 模式 C 读时序

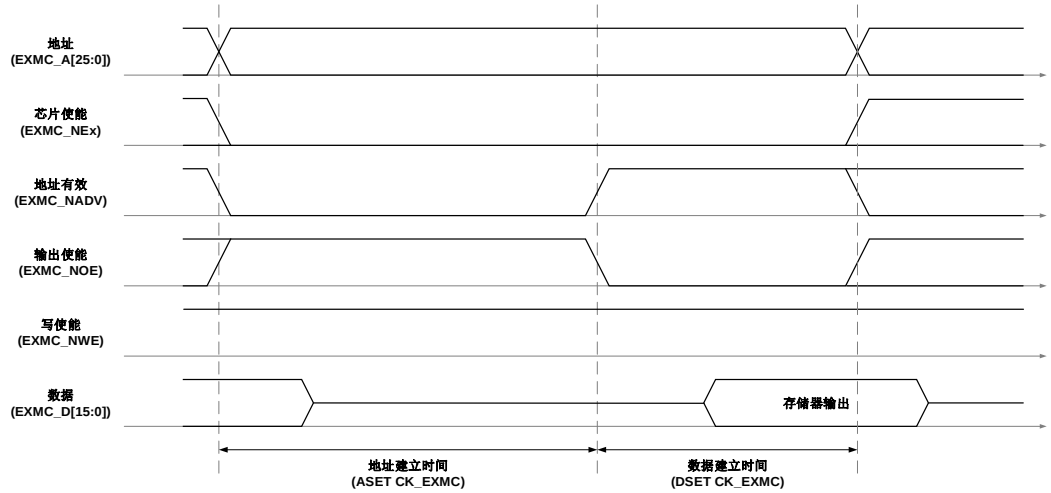
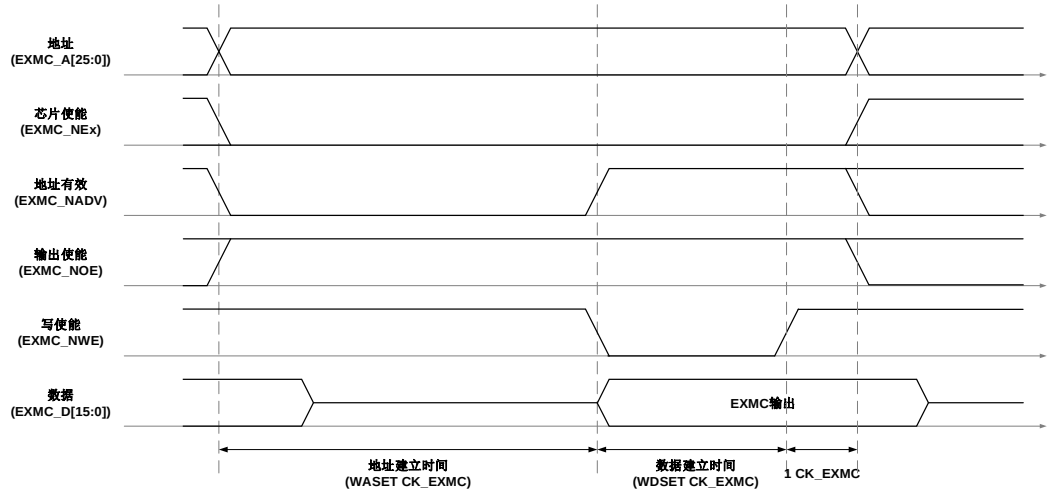


图 38-15. 模式 C 写时序



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。

表 38-11. 模式 C 相关寄存器配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
31-21	保留	0x000
20	CCK	取决于存储器与用户
19	SYNCWR	0x0
18-16	CPS	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	保留	0x0
9	NRWTPOL	仅当位 15 为 1 时有效

位域/位	名称	参考设定值
EXMC_SNCTLx		
8	SBRSTEN	0x0
7	保留	0x1
6	NREN	0x1
5-4	NRW	取决于存储器
3-2	NRTP	0x2，NOR Flash
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFGx		
31-30	保留	0x0000
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
7-4	AHLD	0x0
3-0	ASET	取决于存储器与用户
EXMC_SNWTCFGx		
31-30	保留	0x0
29-28	WASYNCMOD	模式 C：0x2
27-20	保留	0x00
19-16	WBUSLAT	EXMC_NE[x]上升沿到下降沿的时间
15-8	WDSET	取决于存储器与用户
7-4	WAHLD	0x0
3-0	WASET	取决于存储器与用户

模式D - 带地址扩展的异步操作

图 38-16. 模式 D 读时序

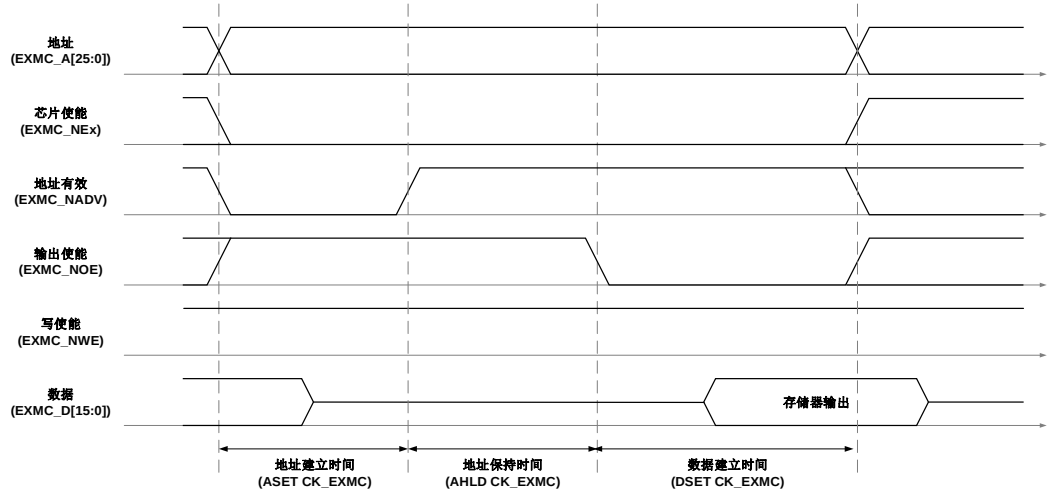


图 38-17. 模式 D 写时序

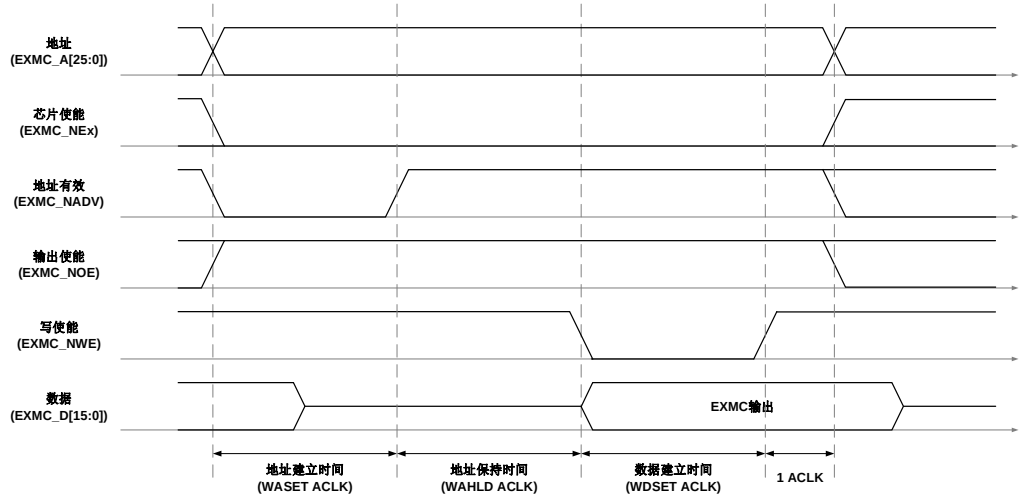


表 38-12. 模式 D 相关寄存器配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
31-21	保留	0x000
20	CCK	取决于存储器与用户
19	SYNCWR	0x0
18-16	CPS	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	保留	0x0
9	NRWTPOL	仅当位 15 为 1 时有效
8	SBRSTEN	0x0
7	保留	0x1
6	NREN	取决于存储器
5-4	NRW	取决于存储器
3-2	NRTP	取决于存储器
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFGx		
31-30	保留	0x0
29-28	ASYNCMOD	模式 D：0x3
27-24	DLAT	无关
23-20	CKDIV	无影响
19-16	BUSLAT	EXMC_NE[x]上升沿到下降沿的时间
15-8	DSET	取决于存储器与用户
7-4	AHLD	取决于存储器与用户
3-0	ASET	取决于存储器与用户

位域/位	名称	参考设定值
EXMC_SNCTLx		
EXMC_SNWTCFGx		
31-30	保留	0x0
29-28	WASYNCMOD	模式 D：0x3
27-20	保留	0x00
19-16	WBUSLAT	EXMC_NE[x]上升沿到下降沿的时间
15-8	WDSET	取决于存储器与用户
7-4	WAHLD	取决于存储器与用户
3-0	WASET	取决于存储器与用户

模式M - NOR Flash地址/数据总线复用

图 38-18. 复用模式读时序

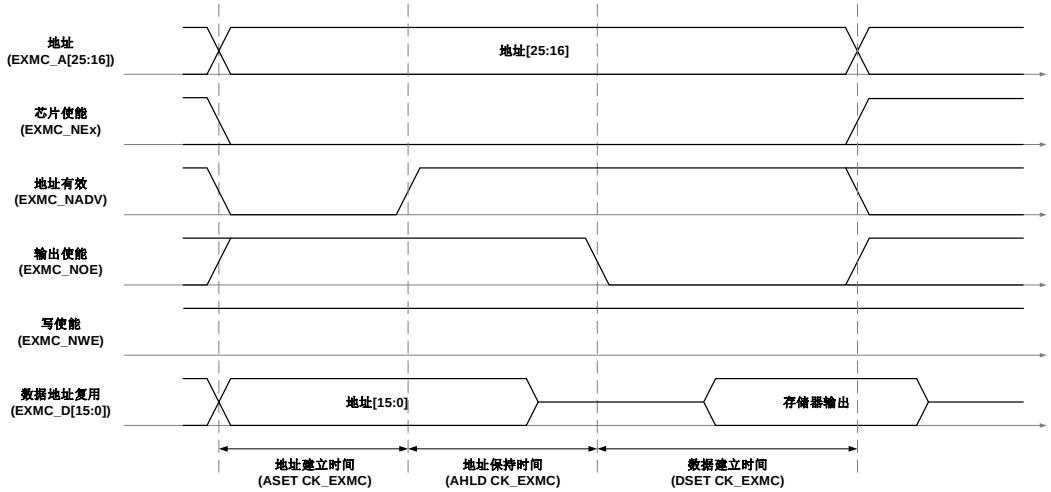


图 38-19. 复用模式写时序

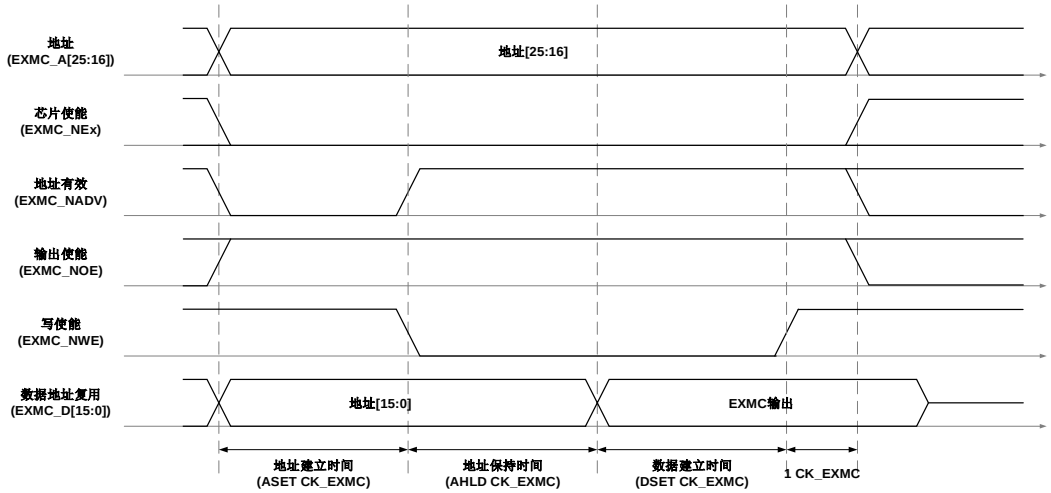


表 38-13. 复用模式相关寄存器配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
31-21	保留	0x000
20	CCK	取决于存储器

位域/位	名称	参考设定值
EXMC_SNCTLx		
19	SYNCWR	0x0
18-16	CPS	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	保留	0x0
9	NRWTPOL	仅当位 15 为 1 时有效
8	SBRSTEN	0x0
7	保留	0x1
6	NREN	0x1
5-4	NRW	取决于存储器
3-2	NRTP	0x2：NOR Flash
1	NRMUX	0x1
0	NRBKEN	0x1
EXMC_SNTCFGx		
31-30	保留	0x0
29-28	ASYNCMOD	0x0
27-24	DLAT	无影响
23-20	CKDIV	无影响
19-16	BUSLAT	EXMC_NE[x]上升沿到下降沿的时间
15-8	DSET	取决于存储器与用户
7-4	AHLD	取决于存储器与用户
3-0	ASET	取决于存储器与用户

异步通信的等待时间

等待功能由寄存器EXMC_SNCTLx位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：

$$T_{DATA_SETUP} \geq \max T_{WAIT_ASSERTION} + 4CK_EXMC \quad (38\text{-}1)$$

若存储器等待信号与EXMC_NE信号对齐：

如果

$$\max T_{WAIT_ASSERTION} \geq T_{ADDRES_PHASE} + T_{HOLD_PHASE} \quad (38\text{-}2)$$

则

$$T_{DATA_SETUP} \geq (\max T_{WAIT_ASSERTION} - T_{ADDRES_PHASE} - T_{HOLD_PHASE}) + 4CK_EXMC \quad (38\text{-}3)$$

否则

$$T_{DATA_SETUP} \geq 4CK_EXMC \quad (38\text{-}4)$$

图 38-20. 异步等待有效时的读时序

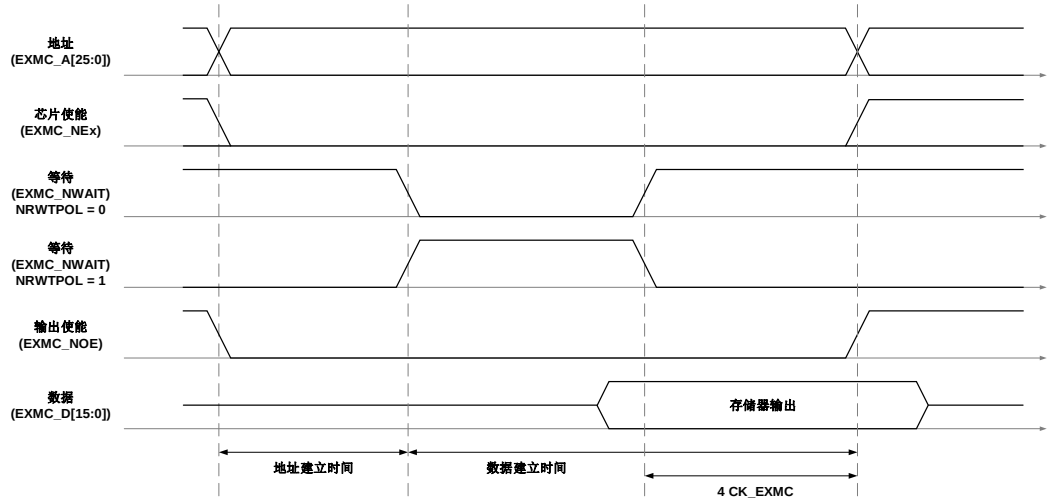
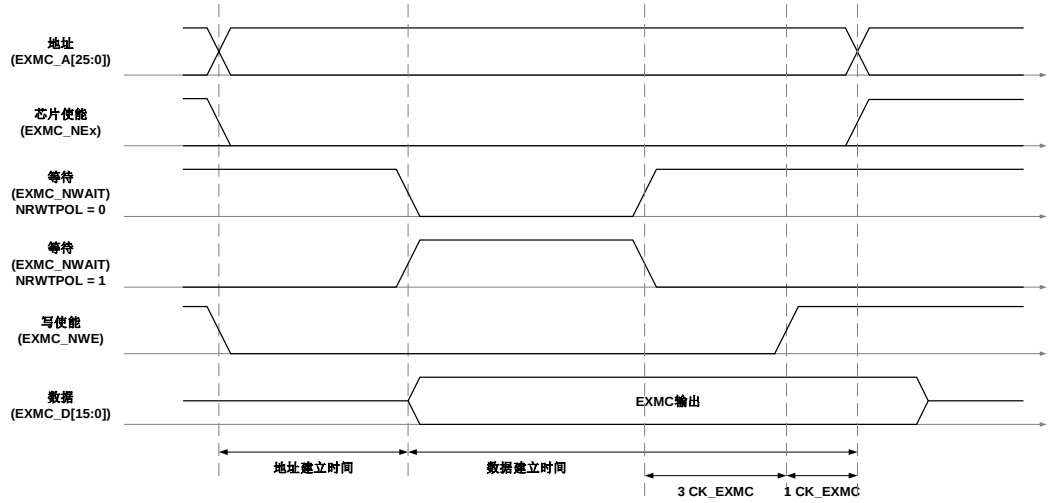


图 38-21. 异步等待有效时的写时序



同步访问时序

存储器时钟（EXMC_CLK）与系统时钟（CK_EXMC）关系如下：

$$EXMC_CLK=\frac{CK_EXMC}{CKDIV+1} \quad (38\text{-}5)$$

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：

$$NOR\ 闪存延迟=DLAT+2 \quad (38\text{-}6)$$

NOR闪存延迟包含EXMC_NADV，二者之间的关系为：

NOR 闪存延迟=DLAT+3 (38-7)

1. 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。

2. 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。

3. 模式 SM – 单次突发传输

对于同步突发传输，如果AXI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AXI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*表38-5. EXMC Bank0支持的访问模式*。

同步复用突发读时序 - NOR, PSRAM (CRAM)

图 38-22. 同步复用突发传输读时序

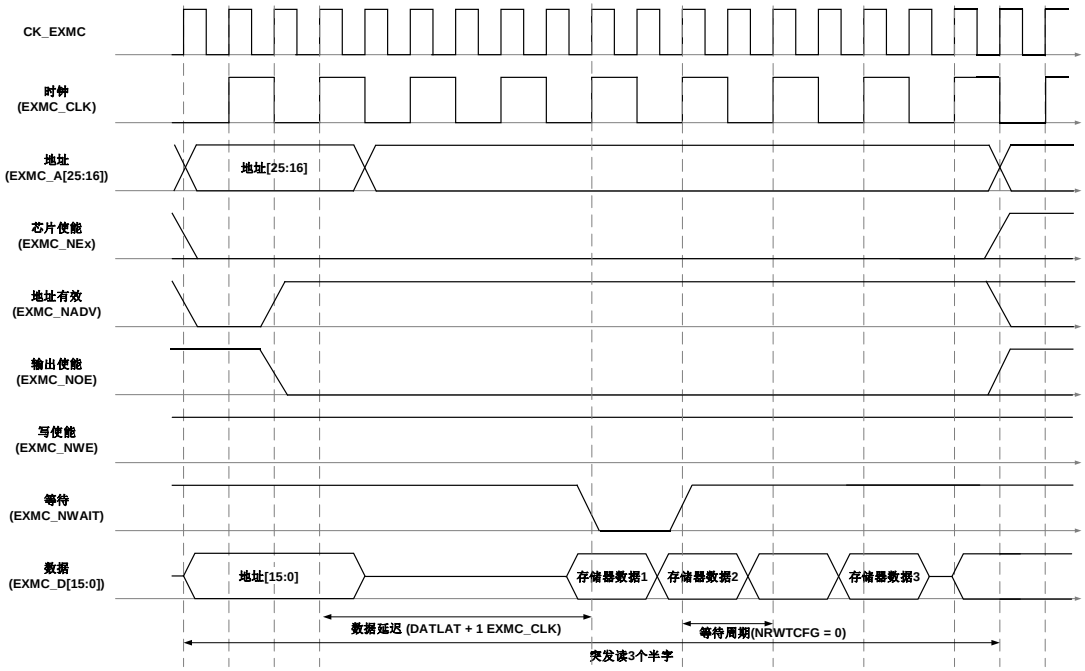


表 38-14. 同步复用模式读时序配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
31-21	保留	0x000
20	CCK	取决于存储器
19	SYNCWR	无影响
18-16	CPS	取决于存储器
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	保留	0x0
9	NRWTPOL	取决于存储器
8	SBRSTEN	0x1，突发读使能
7	保留	0x1
6	NREN	取决于存储器
5-4	NRW	0x1
3-2	NRTP	取决于存储器，0x1/0x2
1	NRMUX	0x1，取决于存储器与用户
0	NRBKEN	0x1
EXMC_SNTCFGx（读）		
31-30	保留	0x0
29-28	ASYNCMOD	0x0
27-24	DLAT	数据延迟

23-20	CKDIV	上图设置 0x1，EXMC_CLK=2 CK_EXMC
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

模式SM – 同步复用突发写时序 – NOR, PSRAM (CRAM)

图 38-23. 同步复用突发传输写时序

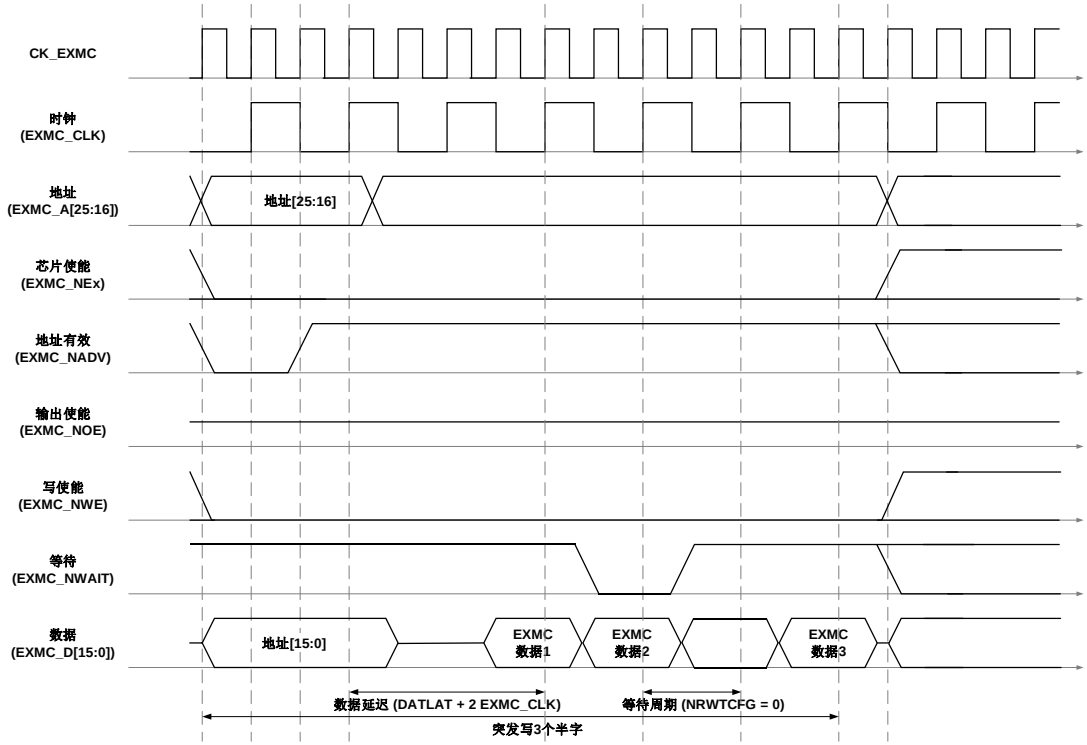


表 38-15. 同步复用模式写时序配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
31-21	保留	0x000
20	CCK	取决于存储器
19	SYNCWR	0x1，同步写使能
18-16	CPS	取决于存储器
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	保留	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

位域/位	名称	参考设定值
EXMC_SNCTLx		
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	上图设置：0x1，EXMC_CLK=2 CK_EXMC
19-16	BUSLAT	无影响
15-8	DSET	无影响
7-4	AHLD	无影响
3-0	ASET	无影响

38.3.7. NAND flash 控制器

EXMC模块Bank2支持NAND FLASH，Bank1和Bank3保留。对于每个Bank，EXMC提供独立的寄存器来配置访问时序，支持8位、16位的NAND FLASH。对于NAND FLASH，EXMC还提供ECC计算模块，保证数据传输和保存的鲁棒性。

NAND flash 接口功能

表 38-16. 8 位/16 位 NAND 接口信号描述

EXMC 引脚	传输方向	功能描述
EXMC_A[17]	输出	NAND Flash 地址锁存（ALE）
EXMC_A[16]	输出	NAND Flash 命令锁存（CLE）
EXMC_D[7:0]/ EXMC_D[15:0]	输入/输出	8 位复用，双向地址/数据总线
		16 位复用，双向地址/数据总线
EXMC_NCE	输出	片选
EXMC_NOE(NRE)	输出	输出使能
EXMC_NWE	输出	写使能
EXMC_NWAIT/ EXMC_INT	输入	NAND Flash 就绪/忙输入信号到 EXMC

支持的存储器访问模式

表 38-17. EXMC Bank2 支持的访问模式

存储器	模式	读/写	AXI 传输宽度	注释
8-bit NAND	异步	R	8	
	异步	W	8	
	异步	R	16	分成 2 次 EXMC 访问
	异步	W	16	

存储器	模式	读/写	AXI 传输宽度	注释
	异步	R	32	分成 4 次 EXMC 访问
	异步	W	32	
	异步	R	64	分成 8 次 EXMC 访问
	异步	W	64	
16-bit NAND	异步	R	8	
	异步	W	8	不支持此操作
	异步	R	16	
	异步	W	16	
	异步	R	32	分成 2 次 EXMC 访问
	异步	W	32	
	异步	R	64	分成 4 次 EXMC 访问
	异步	W	64	

NAND flash 的控制时序

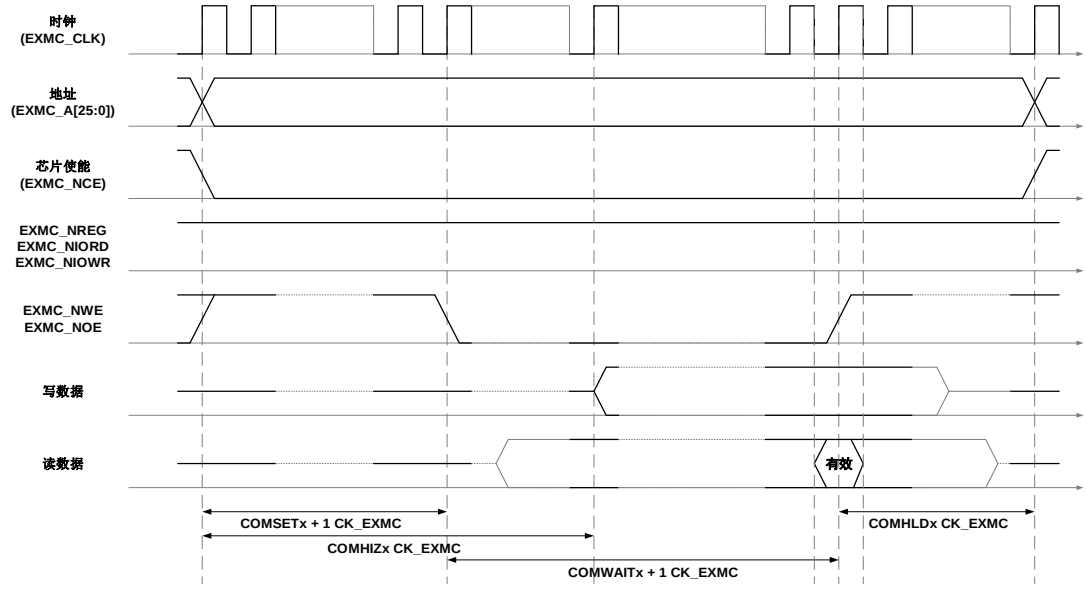
EXMC能够为NAND Flash等设备产生合适的时序信号。每个Bank都有相应的寄存器来对外部存储器进行管理和控制，EXMC_NCTL、EXMC_NINTEN、EXMC_NCTCFG、EXMC_NATCFG、EXMC_NECC，其中寄存器EXMC_NCTCFG、EXMC_NATCFG都可以配置4个时序参数，可以根据用户需求和外部存储器的特性来进行相应的配置。

表 38-18. NAND flash 可编程参数

参数	读/写	单位	功能描述	NAND Flash	
				最小值	最大值
存储器数据总线高阻时间（HIZ）	W/R	CK_EXMC	启动写操作之后保持数据总线为高阻态的时间	1	255
存储器保持时间（HLD）	W/R	CK_EXMC	在发送命令结束后保持地址的(CK_EXMC)时钟周期数目，写操作时也是数据的保持时间	1	254
存储器等待时间（WAIT）	W/R	CK_EXMC	发出命令的最短持续时间(CK_EXMC)时钟周期数目	2	255
存储器建立时间（SET）	W/R	CK_EXMC	发出命令之前建立地址的(CK_EXMC)时钟周期数目	1	256

*图38-24. NAND flash通用存储空间操作时序*给出了在通用存储空间中操作的可编程参数定义，属性存储空间中操作与此相似。

图 38-24. NAND flash 通用存储空间操作时序



NAND flash 操作

EXMC在对NAND Flash发送命令或地址时，需要利用其命令锁存信号（A[16]）或地址锁存信号（A[17]）这两条地址线，即MCU需要在特定的地址进行写操作。

示例：NAND Flash读操作步骤：

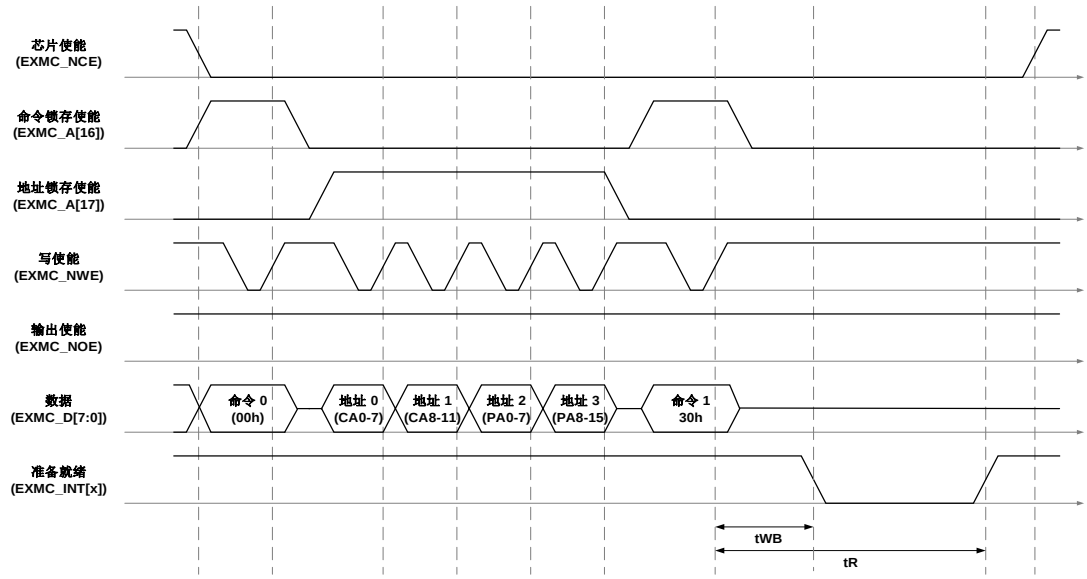
1. 配置EXMC_NCTL、EXMC_NCTCFG，若需要预等待功能，还需配置EXMC_NATCFG；
2. 往通用空间写入NAND Flash读数据命令，即在EXMC_NCE和EXMC_NWE有效期间，EXMC_CLE（A[16]）变为有效电平（高），则被NAND认为写入命令；
3. 往通用空间写入读操作的起始地址，即在EXMC_NCE和EXMC_NWE有效期间，EXMC_ALE（A[17]）变为有效电平（高），则被NAND认为写入地址；
4. 等待NAND就绪信号，NAND控制器会在这期间将和EXMC_NCE一直保持有效；
5. 从通用空间的数据区逐字节的读出数据；
6. 在不写入新的命令和地址，可以自动读出NAND下一页数据；或转到3）写入新的地址进行下一页的读取；或转到2）写入新的命令和地址。

NAND flash 预等待功能

某些NAND Flash要求在输入最后一个地址字节后，控制器等待NAND Flash就绪，并且还有一些对EXMC_NCE敏感型的NAND Flash还要求在其就绪前NCE必须保持有效。

下面以TOSHIBA128M*8bit NAND Flash为例：

图 38-25. NCE 敏感 NAND Flash 访问时序



1. 往NAND的通用空间命令区写入命令CMD0
2. 往 NAND 的通用空间地址区写入操作地址 ADD0
3. 往 NAND 的通用空间地址区写入操作地址 ADD1
4. 往 NAND 的通用空间地址区写入操作地址 ADD2
5. 往 NAND 的通用空间地址区写入操作地址 ADD3
6. 往 NAND 的属性空间命令区写入命令 CMD1

在步骤 6 中写命令操作，EXMC 使用的是寄存器 EXMC_NATCFG 定义的时序。经过 ATTHLD 时间后，NAND Flash 等待 EXMC_INT 信号，ATTHLD 要大于 t_{WB}（EXMC_NWE 高到 EXMC_INT 低的时间）。对于那些对片选信号敏感的 NAND Flash，在输入最后一个地址字节后的第一个命令字节之后，一直到 B/NB 就绪状态（EXMC_INT 从低电平变为高电平）到来的这段时间中，要求片选信号 EXMC_NCE 一直保持低电平。这里可以通过在 EXMC_NATCFG 寄存器配置属性空间的 ATTHLD 值来满足 t_{WB} 的时序要求。MCU 只有在最后一个地址字节之后写入第一个命令字节时才使用属性空间的时序，而在其他时候，都使用通用空间的时序。

NAND Flash 的 ECC 计数模块

EXMC模块中的Bank2有一个ECC计算的硬件模块，用户可以根据EXMC_NCTL中的ECCSZ来选择ECC计算的页面大小，通过ECC计算可以纠正1个bit的错误并且能检测2个bit的错误。

当NAND存储器块使能，ECC模块就会检测EXMC_D[15:0]以及EXMC_NCE、EXMC_NEW信号。当已经完成ECCSZ大小字节的读写操作时，软件必须读出EXMC_NECC中的结果值。如果需要再次开始ECC计算，软件需要先将EXMC_NECC中ECCEN清0来清除EXMC_NCTL中的值，再将ECCEN置1来重新启动ECC计算。

38.3.8. SDRAM 控制器

主要特性

- 两个可独立配置的SDRAM devices；

- 8位，16位，32位数据带宽；
- 多达 13 位行地址、11 位列地址、2 位内部 bank 地址；
- 支持存储器大小：4x16Mx32bit(256 MB), 4x16Mx16bit (128 MB), 4x16Mx8bit (64 MB)；
- AXI 字、半字、字节访问；
- 为每个存储器 bank 提供独立的片选控制；
- 每个存储器 bank 可独立配置；
- 写使能和字节选择输出；
- 自动进行行和 bank 边界管理；
- 多个 bank 的乒乓访问；
- SDRAM 时钟可以为 f_{CK_EXMC}/(2、3、4 或 5)；
- 可编程的时序参数；
- 可编程的刷新速率的自动刷新操作；
- 通过软件进行上电初始化；
- CAS 延迟可设置为 1、2、3 个时钟周期；
- 具有 16x35 位深度的写数据 FIFO；
- 具有 16x31 位深度的写地址 FIFO；
- 6x32 位深度的可缓存的读数据 FIFO；
- 6x14 位深度的可缓存读地址 FIFO；
- 可调整的读数据采样时钟；
- 自刷新模式；
- 掉电模式。

SDRAM 简介

同步动态随机存储器（SDRAM）是通过外部同步时钟刷新的动态随机存储器（DRAM），它的同步时钟由EXMC的EXMC_SDCLK引脚提供，通过配置寄存器EXMC_SDCTLx位SDCLK时钟频率可设置为f_{CK_EXMC}/(2、3、4或5)。指令和数据在时钟的上升沿锁存，在下降沿改变。

SDRAM内部分为多个叫做Bank的区域，允许设备以交错的方式进行访问，以获取更大的并发性和数据传输量。每个Bank可以认为是一个矩阵，其中每个地址对应存储器存储宽度的空间，矩阵由行和列构成，因此存储器的Bank大小可以认为是存储器数据宽度*行数*列数。用户可以通过设置寄存器EXMC_SDCTLx位NBK，SDW，RAW，CAW使EXMC可以与不同的SDRAM进行通信。

由于易失的本征特性，SDRAM需要周期性的刷新。EXMC支持两种刷新模式，自刷新和自动刷新。自刷新是在EXMC挂起的低功耗模式中使用，由SDRAM内部计数提供时钟，内部进行刷新。自动刷新是由EXMC周期性的提供刷新命令，因为此时SDRAM需要进行数据传输，刷新间隔由寄存器EXMC_SDARI位ARINTV决定，连续刷新次数由寄存器EXMC_SDCMD位NARF决定。刷新命令优先级高于其他的包括读写命令，来保证数据的正常存储，当SDRAM同时收到刷新命令与读写命令时，读写命令需要等待刷新命令完成才能进行。如果在前一个刷新命令未完成时，再次接收到刷新命令，寄存器EXMC_SDSTAT刷新错误标志位（REIF）会被置位，同时如果刷新错误中断使能（REIE），将会发生刷新错误中断。

CAS延迟是读命令和数据线出现第一个可读数据之间的延迟，可以通过寄存器EXMC_SDCTLx位CL设置。

对不同的SDRAM需要参考其手册，使用模式寄存器进行设置，包含突发长度，突发类型，CAS

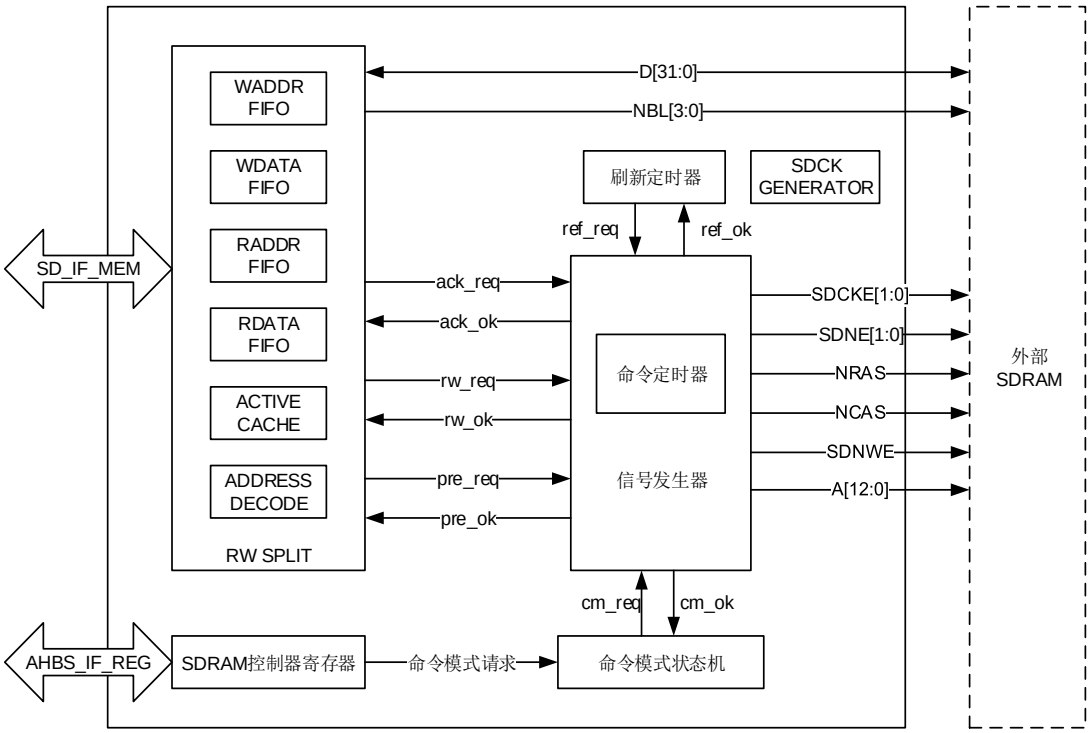
延迟，写模式。在寄存器EXMC_SDCMD位MRC中设置，会通过CMD命令发送给SDRAM。在读写操作之前，需要发送读取模式寄存器命令，否则SDRAM无法工作。

SDRAM 控制器简介

同步动态随机存储器控制器（SDRAMC）是MCU和SDRAM的接口。它把AXI的操作根据SDRAM协议转换为对SDRAM的操作，同时配置寄存器EXMC_SDTCFG满足时序要求。

SDRAMC包含4个模块，读写预处理模块，控制寄存器，有限状态机和信号发生器。使用两组FIFO来提高存储器访问效率，一组用来写地址和数据，另外一组用来读地址数据。SDRAMC模块由*图38-26. SDRAM系统架构*所示。

图 38-26. SDRAM 系统架构



信号发生器处理状态机，刷新定时器，读写模块产生的请求。

命令定时器由遵守SDRAM时序协议的计数器组成。

SDRAM命令由SDRAM控制器接口发出，可见*表38-19. SDRAM命令真值表*。

表 38-19. SDRAM 命令真值表

SDNE	NRAS	NCAS	SDNWE	A[n]	A[10]	A[m]	命令
H	X	X	X	X	X	X	命令禁止（无操作）
L	H	H	H	X	X	X	无操作
L	H	H	L	X	X	X	中止突发传输
L	H	L	H	Bank	L	Col	突发读选择行
L	H	L	H	Bank	H	Col	预充电完成后，突发读选择行

SDNE	NRAS	NCAS	SDNWE	A[n]	A[10]	A[m]	命令
L	H	L	L	Bank	L	Col	突发写选择行
L	H	L	L	Bank	H	Col	预充电完成后，突发写选择行
L	L	H	H	Bank	Row	Row	行使能命令，之后可进行读写
L	L	H	L	Bank	L	X	预充电命令，关闭当前 Bank 的选择行
L	L	H	L	X	H	X	全局预充电命令，关闭所有 Bank 的选择行
L	L	L	H	X	X	X	SDCKE = 1 时自动刷新模式 SDCKE = 0 时自刷新模式
L	L	L	L	L	Mode	Mode	加载模式寄存器

SDRAM 控制器操作序列

IO 配置

SDRAMC的IO口必须在与SDRAM通信之前配置，否则，它就被留作通用IO口，并且可以被其他模块使用。下表总结了与SDRAM操作相关的IO口。

表 38-20. SDRAM IO 口定义

信号	传输方向	描述
EXMC_SDCLK	O	SDRAM 存储器时钟
EXMC_SDCKE[0]	O	SDRAM device 0 的时钟使能信号
EXMC_SDCKE[1]	O	SDRAM device 1 的时钟使能信号
EXMC_SDNE[0]	O	SDRAM device 0 的片选信号，低电平有效
EXMC_SDNE[1]	O	SDRAM device 1 的片选信号，低电平有效
EXMC_NRAS	O	行地址选通，低电平有效
EXMC_NCAS	O	列地址选通，低电平有效
EXMC_SDNWE	O	写使能，低电平有效
EXMC_A[12:0]	O	地址
EXMC_A[15:14]	O	Bank 地址
EXMC_D[31:0]	I/O	读/写数据
EXMC_NBL[3:0]	O	写数据标记（掩码）

控制器初始化

用户需要按照以下步骤来初始化 SDRAM 控制器，初始化序列可以应用于单个 SDRAM，或同时初始化两个 SDRAM，由寄存器 EXMC_SDCMD 位 DS0 和 DS1 决定。为了保证读写的可靠性，必须先进行初始化，否则无法保证 EXMC 的行为。

1. 控制参数：控制配置寄存器EXMC_SDCTLx指定SDRAM的存储器行列数，时钟配置和读写方法。
2. 时序参数：时序配置寄存器EXMC_SDTCFGx需要根据SDRAM数据手册来配置，以与外部 SDRAM 的操作保持同步。RPD 和 ARFD 必须在 EXMC_SDTCFG0 来配置，EXMC_SDTCFG1中相应的位保留。
3. 使能SDCLK：通过将'0b001'写入EXMC_SDCMD寄存器中的CMD位域来完成SDCLK使

能命令应发送到相应的SDRAM设备，DS0和DS1决定选择哪个设备将接受命令并开始接收EXMC_SDCLK。

4. 上电延时：典型延时在100us左右。
5. 预充电：命令会对SDRAM的所有Bank进行复位，并使SDRAM回到空闲状态，等待后续操作。给寄存器EXMC_SDCMD位域CMD写'0b010'使能相应设备的SDCLK信号，DS0和DS1决定选择哪个设备将接受此命令。
6. 设置自刷新模式：给寄存器EXMC_SDCMD位域CMD写'0b011'发送自刷新命令。用户也可以通过设置位NARF来设置连续刷新次数，这个配置是SDRAM规范要求的，也是用户应该参考的地方，DS0和DS1决定选择哪个设备将接受此命令。
7. 模式寄存器配置：模式寄存器通过写寄存器EXMC_SDCMD位域MRC来设置，模式寄存器指定了SDRAM的工作模式，这些模式包括突发长度，突发类型，CAS延迟和读写模式，用户应参考SDRAM的规范进行正确配置。CAS延迟必须与寄存器EXMC_SDCTLx位域CL对应，突发长度设为1来保证数据正常传输。如果两个SDRAM的模式寄存器内容不同，需要通过DS0和DS1单独选择设备来配置。
8. 设置自刷新频率：自动刷新率对应刷新周期之间的时间间隔，用户必须确保刷新周期满足SDRAM的要求。

这里控制器已经完成初始化，可以与SDRAM通信。如果发生了复位，初始化需要按照上述步骤重复一遍。在读写操作之前，要保证控制器至少初始化一遍。

预充电

若SDRAM控制器在存取时需要进行行切换，那么首先需要将该行地址对应块的读写放大器去使能，使其进入空闲状态，为下一行的读写操作进行准备。这个过程叫做预充电，或者行去使能。预充电可以由控制器的全局预充电命令（Precharge-All）独立激发，或者是在读写完成后自动激发。行预充电延时（RPD）代表SDRAM行切换的最小时间，它是预充电完成到下一次行使能命令的最小时间间隔。

行使能

行使能命令将行地址所在的块使能，完整的行地址由2比特的块地址EXMC_A[15:14]和13比特的行地址EXMC_A[12:0]组成。行使能会将所选行的16384比特信息读入读写放大器，这个过程也叫做行开启，该命令的一个副作用就是对所选行的存储单元进行了刷新。

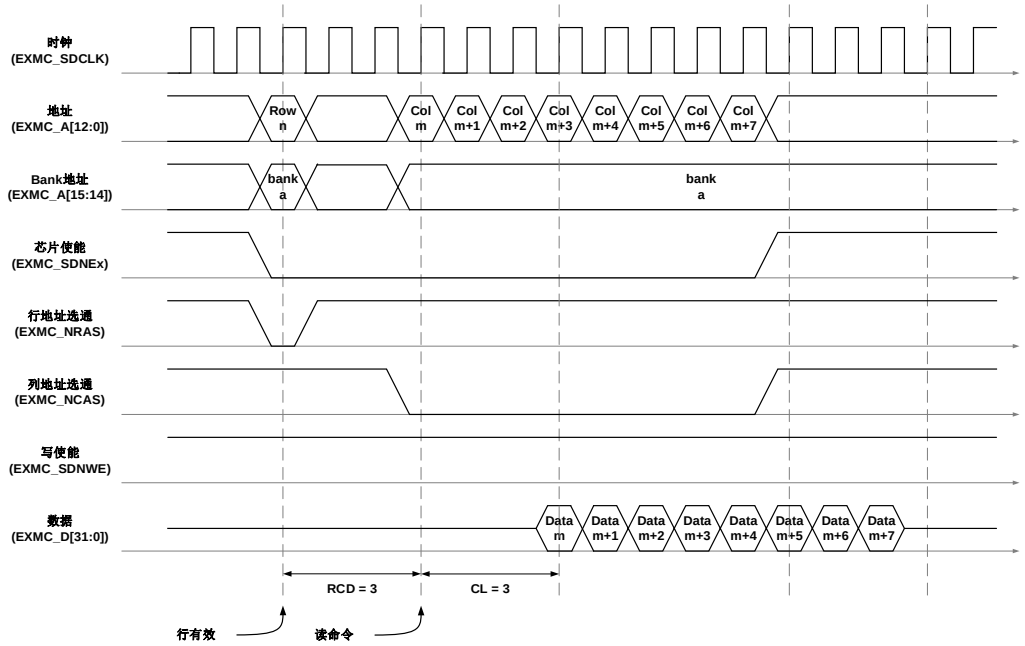
一旦行使能，读写操作就可以顺利的进行，但是行使能需要一定的时间，这个时间间隔叫做行列延时，它是行寻址到列寻址的最小时间间隔。对SDRAM控制器进行配置时的行列延时（RCD），是包含SDRAM行列延时的最小时钟周期数，它代表了行使能到SDRAM读写间的最小等待时间。在这段时间中，用户可向其它的块地址发出控制命令，因为SDRAM控制器对块的操作是独立进行的。

读写访问

控制器可以把AXI的单次或突发读操作转换成单次的存储器读操作。为了连续访问，控制器通常会保存之前操作的行号。若下一次的读取位置是在相同的行号或是已经使能的其他行号，那么读操作会未中断的执行，否则需要先执行取消使能当前行和使能需要操作的行，然后执行读取访问。读FIFO的设计用于在CAS延迟期间缓存读数据，必须设置管道延迟（PIPED）、突发读取（BRSTRD）以启用FIFO。

*图38-27. 突发读操作*是对一个未被使能的行突发读操作，在读之前发送了行使能指令。若对一个已经使能的行进行读操作，只需要发送列地址，行地址无需发送。

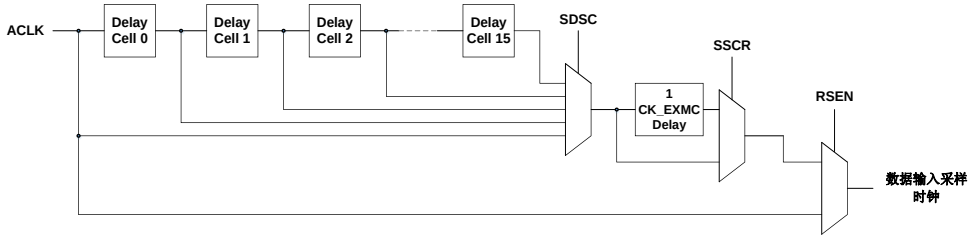
图 38-27. 突发读操作



内部生成的时钟（具有来自CK_EXMC的可调延迟）可用于从外部存储器采样读取数据。当CK_EXMC无法对读取的数据进行正确采样时，此时钟可能会有所帮助。当该时钟启用时，读取数据将首先存储在异步FIFO中，然后返回到AXI总线。读取命令过程中可能会带来大约2~3 CK_EXMC的额外延迟。

时钟延迟模块在CK_EXMC输入到信号发生器后添加，这个延迟的时钟作为输入数据的采样时钟。延迟模块可以通过寄存器EXMC_SDRSCTL来控制，其中RSEN位选择是否使用CK_EXMC延迟，SSCR位选择是否额外增加一个CK_EXMC延迟，SDSC选择增加多少个CK_EXMC延迟，可以添加的延迟单元数在0到15之间。*图38-28. 数据采样时钟延迟模块*显示了数据采样时钟延迟。

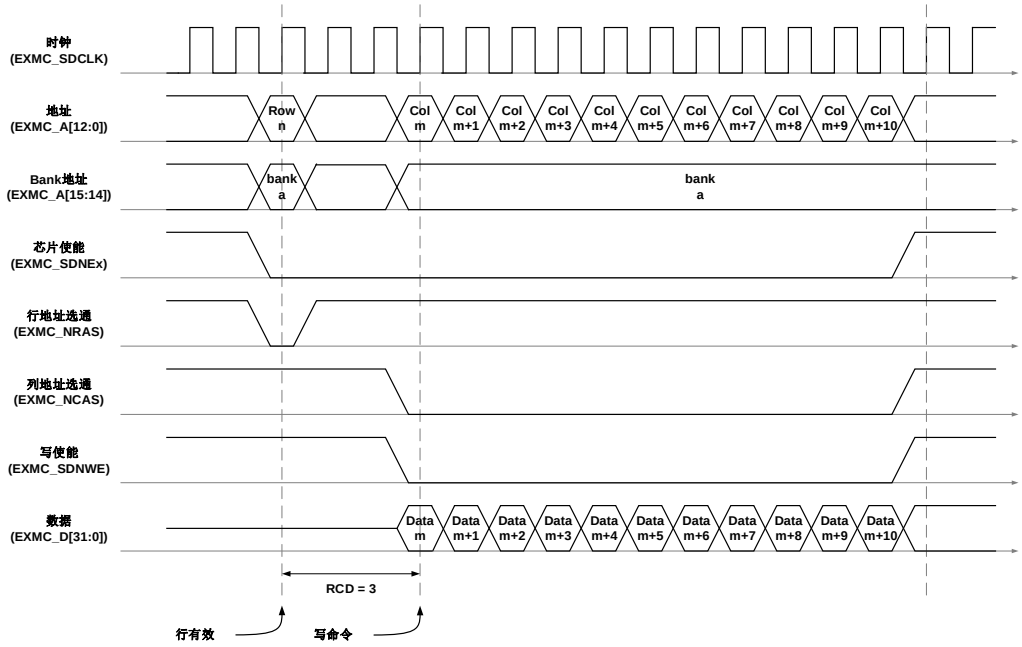
图 38-28. 数据采样时钟延迟模块



控制器可以把AXI的单次或突发写操作转换成单次的存储器写操作。写操作之前必须失能写保护位（寄存器EXMC_SDCTLx位WPEN）。为了连续访问，控制器通常会保存之前操作的行号，若下一次操作是在相同的行号或是已经使能的其他行号，操作会未中断的执行，否则需要先执行取消使能当前行和使能需要操作的行，最后才会执行写入访问。

*图38-29. 突发写操作*是对一个未被使能的行突发写操作，在写之前发送了行使能指令。若对一个已经使能的行写操作，则不需要行地址选通，只需要列地址选通。

图 38-29. 突发写操作



读写命令预处理模块接收AXI命令，然后根据AXI总线和SDRAM接口的数据总线宽度将AXI命令转换成单个的SDRAM读/写访问。

在读写命令预处理模块中，有两个写FIFO，用于缓冲AXI写命令的地址和数据。当两个写FIFO都不为空时，产生写访问。

当寄存器EXMC_SDCTLx位BRSTRD置1时，读写命令预处理模块能够预处理下一个读访问。读FIFO被用来存储在CAS延迟（由EXMC_SDCTLx中的CL位配置）和PIPED延迟（由EXMC_SDCTLx中的PIPED位配置）期间提前读出的数据。

读数据FIFO能够最多缓存6个32位的读数据字，同时地址FIFO携带6个14位的读地址标签，这些标签用来标识6个32位的读数据字中每一个。每个地址标签由11位列地址，2位Bank地址和1位SDRAM设备选择位。

当在AXI总线上出现一个读命令时，读写命令预处理模块将首先检查这个地址是否和某个地址标签匹配，如果匹配，则直接从FIFO中读取数据。否则，向存储器发一个新的读命令，FIFO会被新的数据更新。如果FIFO满了，旧的数据会被丢失。

读FIFO操作，如*图38-30. FIFO未命中时的读访问（BRSTRD=1，CL=2，SDCLK=2，PIPED=2）*和*图38-31. FIFO命中时的读访问（BRSTRD=1）*所示。

图 38-30. FIFO 未命中时的读访问（BRSTRD=1，CL=2，SDCLK=2，PIPED=2）

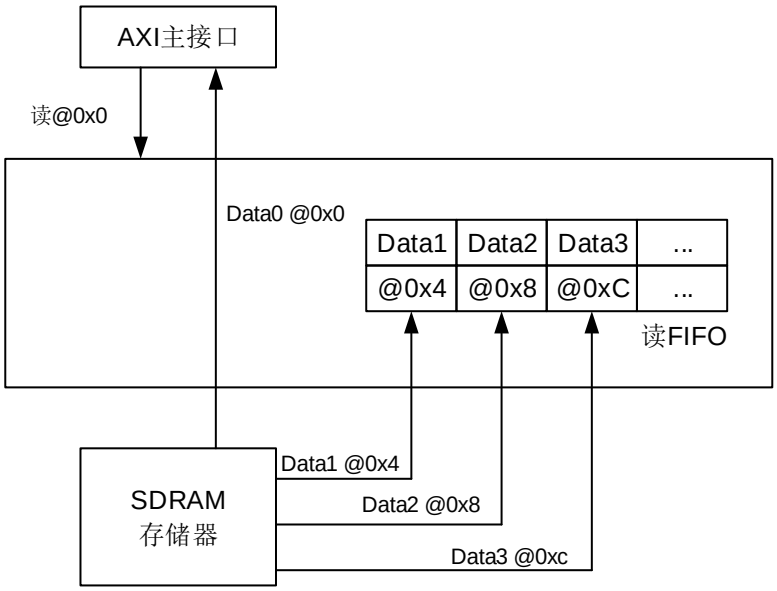
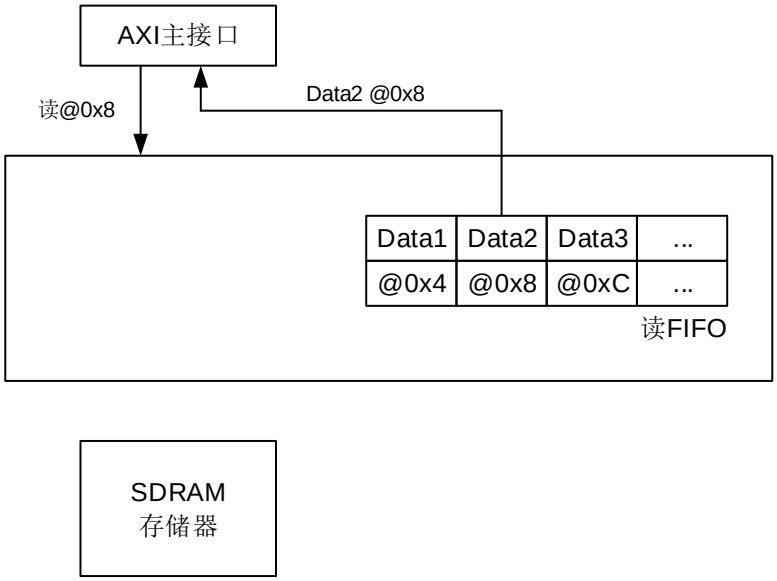


图 38-31. FIFO 命中时的读访问（BRSTRD=1）



当一个写访问或者预充电命令出现时，读FIFO缓冲区中的数据就会被清除掉，用以填充新的数据。

地址译码器子模块会根据外部存储器设备的配置将AXI总线地址转化成片选、内部bank地址、行地址和列地址。

使能缓存子模块记录着内部bank（最多8个）是否处于使能状态。当一个内部bank处于使能状态，则相应的行地址也会被记录。当AXI访问或者自动刷新命令出现时，读写命令预处理模块将会查询这个记录，并且决定是否生成使能或预充电命令。

读/写操作之前，目标行必须被使能，EXMC_A[15:14]选择Bank，EXMC_A[12:0]选择行。被选择的行在预充电命令出现前会一直有效。预充电命令用来取消选择特定Bank或者所有Bank使能的行。预充电命令必须在使能同一个Bank的不同行之前发出。使能和预充电由EXMC自动发出，它的正确性取决于之前描述的存储器的相关配置。有关自动行使能和预充电的读写时序如

*图38-32. 跨边界读操作*和*图38-33. 跨边界写操作*所示。

图 38-32. 跨边界读操作

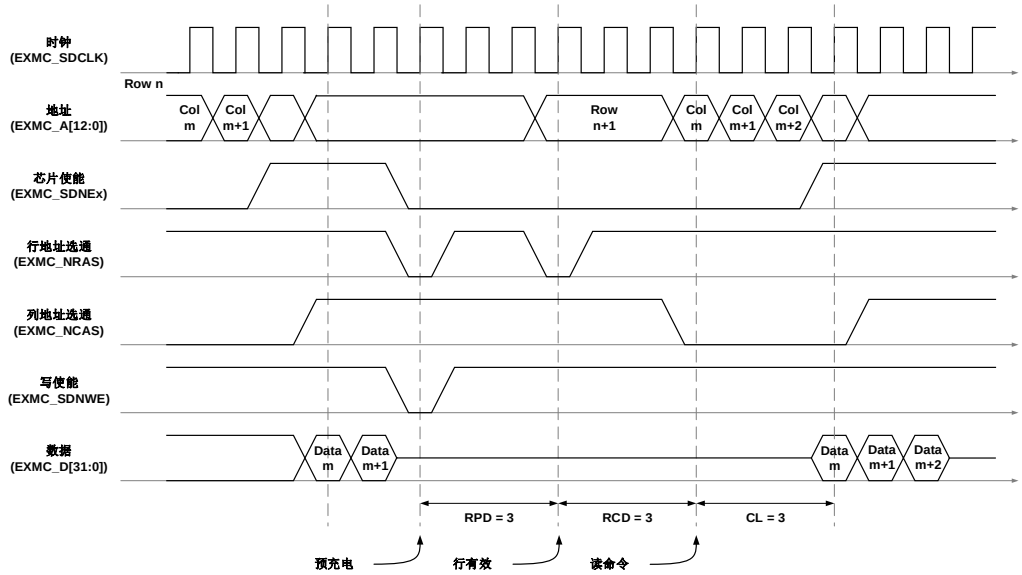
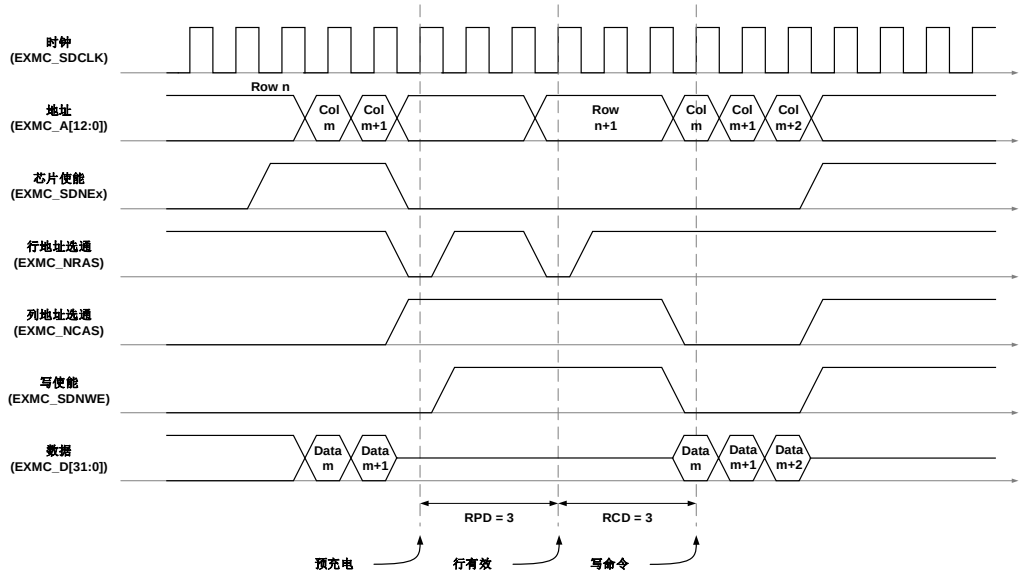


图 38-33. 跨边界写操作



上图描述了在跨行边界时的读写操作时序，会按照以下步骤自动执行：

1. 预充电当前行；
2. 使能下一行；
3. 读写操作。

预充电延迟（PRD）和行到列延迟（RCD）根据其在寄存器 EXMC_SDTCFGx 中的配置添加。其他时序参数必须参照 SDRAM 标准要求。

当读写操作发生在Bank边界时，会有以下两种情况：

1. 当前Bank不是最后一个Bank，使能下一个Bank的第一行，支持任意的行，列，总线宽度设置。
2. 当前Bank是最后一个Bank，行，列，总线宽度设置为13位，11位，32位。假设当前操作

的SDRAM位device0，控制器会在device1上继续操作。

低功耗模式

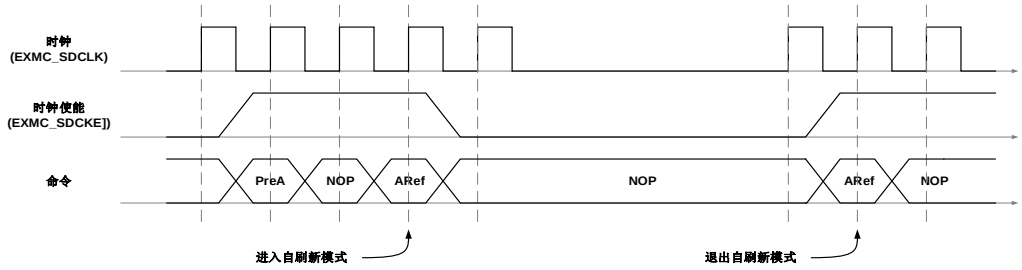
EXMC支持两种低功耗模式：

1. 自刷新模式：在自刷新模式中，在没有外部时钟（EXMC_CLK）的情况下，刷新由SDRAM本身提供，以此来保持数据的完整性。通过往寄存器EXMC_SDCMD位域CMD写入'0b101'进入自刷新模式，DS0和DS1决定哪个SDRAM设备接收到该命令。如果自刷新指令发送给两个SDRAM设备或一个未初始化的SDRAM设备，则在RASD延迟后EXMC_SDCLK停止运行。
2. 掉电模式：在掉电模式中，刷新由SDRAM控制器提供。通过往寄存器EXMC_SDCMD位域CMD写入'0b110'进入掉电模式，DS0和DS1决定哪个SDRAM设备接收到该命令。如果写数据FIFO非空，在掉电模式使能之前，所有数据都会发送给存储器。

命令模式状态机也控制正常模式和低功耗模式（自刷新/掉电）之间的转换过程。

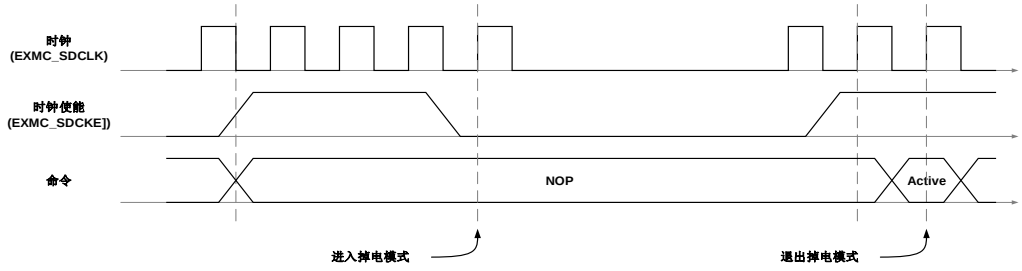
当读/写访问出现时，SDRAM控制器会从自刷新模式退出，返回到正常模式。如果在SDRAM控制器进入自刷新模式时出现读/写访问，则自刷新的进入过程会被中断，并且在读写访问完成后SDRAM控制器会停留在正常模式。

图 38-34. 自刷新模式进入和退出的处理



如果在SDRAM控制器处于掉电模式时出现自动刷新请求，那么SDRAM控制器会退出掉电模式并返回到正常模式，发"预充电所有存储区域"命令和"自动刷新"命令序列，然后再一次自动进入掉电模式。

图 38-35. 掉电模式进入和退出的处理



状态和中断

寄存器EXMC_SDSTAT的准备未完成状态位NRDY指示SDRAM是否准备完成接受新的命令。在控制器发送新的命令之后，该位会被清除。

寄存器EXMC_SDSTAT的STA0和STA1定义SDRAM的Device 0和Device 1的状态，0b00代表

普通模式，0b01表示相应的SDRAM Device处于自刷新模式，0b10代表掉电模式。

若前一个刷新指令未完成时，接收到了新的刷新指令，寄存器EXMC_SDSTAT刷新错误标志位（REIF）会被置位，该位通过寄存器EXMC_SDARI位REC置位来清除。

38.4. EXMC 寄存器

EXMC基地址：0x5200 4000

38.4.1. NOR/PSRAM 控制器寄存器

SRAM/NOR Flash 控制寄存器（EXMC_SNCTLx, x=0, 1, 2, 3）

地址偏移：0x00 + 8 * x, (x = 0, 1, 2, 3)
复位值：0x0000 30DA

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						BKREMAP[1:0]		保留			CCK	SYNCWR	CPS[2:0]		
						rw					rw	rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASYNCWTEN	EXMODEN	NRWTEN	WEN	NRWTCFG	保留	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

位/位域	名称	描述
31:26	保留	必须保持复位值。
25:24	BKREMAP[1:0]	Bank 重映射 00：默认映射 01：NOR/PSRAM bank 和 SDRAM device0 交换 10：保留 11：保留 **注意：**BKREMAP 位域只在 EXMC_SNCTL0 有效，EXMC_SNCTLx (x = 1, 2, 3)没有意义。
23:21	保留	必须保持复位值。
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, 2, 3)没有意义。当该位置为 1 时，只有 EXMC_SNTCFG0 寄存器的 CKDIV[3:0]可以影响 EXMC_CLK 的输出。
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 AXI 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	保留	必须保持复位值。
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 Flash 访问使能 0：禁止 NOR Flash 访问 1：允许 NOR Flash 访问
5:4	NRW[1:0]	NOR 存储器数据宽度 00：8 位 01：16 位(复位缺省值) 10：32 位 11：保留
3:2	NRTP[1:0]	NOR 存储器类型

00：SRAM、ROM
01：PSRAM（CRAM）
10：NOR Flash
11：保留

1	NRMUX	NOR 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	NOR 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 时序配置寄存器（EXMC_SNTCFGx, x=0, 1, 2, 3）

地址偏移：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（EXMC_SNCTLx 寄存器的 EXMODEN 位为 1）中使用。 00：模式 A 01：模式 B 10：模式 C 11：模式 D
27:24	DLAT[3:0]	NOR Flash 数据延时，仅在同步模式有效 0x0：第一数据的保持时间为 2 个 EXMC_CLK 时钟周期 0x1：第一数据的保持时间为 3 个 EXMC_CLK 时钟周期 …… 0xF：第一数据的保持时间为 17 个 EXMC_CLK 时钟周期
23:20	CKDIV[3:0]	同步模式时钟分频比，仅在同步模式有效 0x0：无 EXMC_CLK 时钟输出 0x1：EXMC_CLK 周期 = 2 * CK_EXMC 周期 ……

0xF：EXMC_CLK 周期 = 16 * CK_EXMC 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 * CK_EXMC 周期 0x1：总线延迟 = 1 * CK_EXMC 周期 …… 0xF：总线延迟 = 15 * CK_EXMC 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 * CK_EXMC 周期 …… 0xFF：数据建立时间 = 255 * CK_EXMC 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 * CK_EXMC …… 0xF：地址保持时间 = 15 * CK_EXMC
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 * CK_EXMC …… 0xF：地址建立时间 = 15 * CK_EXMC

SRAM/NOR Flash 写时序寄存器（(EXMC_SNWTCFGx, x=0, 1, 2, 3）

地址偏移：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_SNCTLx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		WASYNMOD[1:0]		保留								WBUSLAT[3:0]			
		rw										rw			

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

位/位域	名称	描述
31:30	保留	必须保持复位值。

29:28	WASYNCMOD[1:0]	异步访问模式 该位只有在扩展模式（EXMC_SNCTLx 寄存器的 EXMODEN 位为 1）中使用。 00：模式 A 01：模式 B 10：模式 C 11：模式 D
27:20	保留	必须保持复位值。
19:16	WBUSLAT[3:0]	总线延迟时间 在复用读模式中使用，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 * CK_EXMC 周期 0x1：总线延迟 = 1 * CK_EXMC 周期 …… 0xF：总线延迟 = 15 * CK_EXMC 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 * CK_EXMC 周期 …… 0xFF：数据建立时间 = 255 * CK_EXMC 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：地址保持时间 = 1 * CK_EXMC …… 0xF：地址保持时间 = 15 * CK_EXMC
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 * CK_EXMC 0x1：地址建立时间 = 1 * CK_EXMC …… 0xF：地址建立时间 = 15 * CK_EXMC

38.4.2. NAND Flash 控制器寄存器

NAND Flash 控制器寄存器（EXMC_NCTL）

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	ECCSZ[2:0]	ART[3]
	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATR[2:0]			CTR[3:0]				保留		ECCEN	NDW[1:0]		保留	NDBKEN	NDWTEN	保留
rw			rw						rw	rw			rw	rw	

位/位域	名称	描述
31:20	保留	必须保持复位值。
19:17	ECCSZ[2:0]	ECC 块大小 000：256 字节 001：512 字节 010：1024 字节 011：2048 字节 100：4096 字节 101：8192 字节
16:13	ATR[3:0]	ALE 至 RE 延迟 0x0：ALE 至 RE 延迟 = 1 * CK_EXMC …… 0xF：ALE 至 RE 延迟 = 16 * CK_EXMC
12:9	CTR[3:0]	CLE 至 RE 延迟 0x0：CLE 至 RE 延迟 = 1 * CK_EXMC 0x1：CLE 至 RE 延迟 = 2 * CK_EXMC …… 0xF：CLE 至 RE 延迟 = 16 * CK_EXMC
8:7	保留	必须保持复位值。
6	ECCEN	ECC 使能 0：关闭 ECC，并复位 EXMC_NECC 1：使能 ECC
5:4	NDW[1:0]	NAND 外部存储器宽度 00：8 位 01：16 位 其他：保留
3	保留	必须保持复位值。
2	NDBKEN	NAND 外部存储器使能 0：禁能对应的存储器块 1：使能对应的存储器块
1	NDWTEN	NWAIT 信号使能位 0：关闭等待功能 1：使能等待功能

0	保留	必须保持复位值。

NAND Flash 中断使能寄存器（EXMC_NINTEN）

地址偏移：0x84

复位值：0x0000 0042

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									FFEPT	INTFEN	INTHEN	INTREN	INTFS	INTHS	INTRS
									rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:7	保留	必须保持复位值。
6	FFEPT	FIFO 空标志位 0：FIFO 非空 1：FIFO 空
5	INTFEN	中断下降沿检测使能 0：禁用中断下降沿检测 1：使能中断下降沿检测
4	INTHEN	中断高电平检测使能 0：禁用中断高电平检测 1：使能中断高电平检测
3	INTREN	中断上升沿中断检测使能 0：禁用中断上升沿检测 1：使能中断上升沿检测
2	INTFS	中断下降沿状态 0：没有检测到中断下降沿 1：检测到中断下降沿
1	INTHS	中断高电平状态 0：没有检测到中断高电平 1：检测到中断高电平
0	INTRS	中断上升沿状态 0：没有检测到中断上升沿 1：检测到中断上升沿

NAND Flash 通用空间时序寄存器（EXMC_NCTCFG）

地址偏移：0x88

复位值：0xFFFF FFFF

这些操作适用于NAND Flash的外部存储器的通用存储空间。

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:24	COMHIZ[7:0]	通用空间数据总线的高阻时间 定义在通用空间进行写操作后数据总线保持高阻态时间 0x00：COMHIZ = 1 * CK_EXMC …… 0xFE：COMHIZ = 255 * CK_EXMC 0xFF：保留
23:16	COMHLD[7:0]	通用空间的保持时间 在发送地址后的地址保持时间，在写操作时，也作为数据信号保持的时间 0x00：保留 0x01：COMHLD = 1 * CK_EXMC …… 0xFE：COMHLD = 254 * CK_EXMC 0xFF：保留
15:8	COMWAIT[7:0]	通用空间的等待时间 定义了保持命令的最小时间 0x00：保留 0x01：COMWAIT = 2 * CK_EXMC（加上 NWAIT 时钟周期） …… 0xFE：COMWAIT = 255 * CK_EXMC（加上 NWAIT 时钟周期） 0xFF：保留
7:0	COMSET[7:0]	通用空间的建立时间 定义地址信号的建立时间 0x00：COMSET = 1 * CK_EXMC …… 0xFE：COMSET = 255 * CK_EXMC 0xFF：保留

NAND Flash 属性空间时序寄存器（EXMC_NATCFG）

地址偏移：0x8C

复位值：0xFFFF FFFF

如果必须应用其他时序，对于最后地址的写访问，它被用于NAND Flash的属性存储空间的8位访问。

该寄存器只能按字（32 位）访问。

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

位/位域	名称	描述
31:24	ATTHIZ[7:0]	属性空间数据总线的高阻时间 定义在属性空间进行写操作后数据总线保持高阻态时间 0x00：ATTHIZ = 0 * CK_EXMC …… 0xFE：ATTHIZ = 254 * CK_EXMC 0xFF：保留
23:16	ATTHLD[7:0]	属性空间的保持时间 在发送地址后的地址保持时间，在写操作时，也作为数据信号保持的时间 0x00：保留 0x01：ATTHLD = 1 * CK_EXMC …… 0xFE：ATTHLD = 254 * CK_EXMC 0xFF：保留
15:8	ATTWAIT[7:0]	属性空间的等待时间 定义了保持命令的最小时间 0x00：保留 0x01：ATTWAIT = 2 * CK_EXMC（加上 NWAIT 时钟周期） …… 0xFE：ATTWAIT = 255 * CK_EXMC（加上 NWAIT 时钟周期） 0xFF：ATTWAIT = 保留
7:0	ATTSET[7:0]	属性空间的建立时间 定义地址信号的建立时间 0x00：ATTSET = 1 * CK_EXMC …… 0xFE：ATTSET = 255 * CK_EXMC 0xFF：保留

NAND Flash ECC 寄存器（EXMC_NECC）

地址偏移：0x94

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:0	ECC[31:0]	ECC 计算结果

ECCSZ[2:0]	NAND Flash 页大小	ECC 位
0b000	256	ECC[21:0]
0b001	512	ECC[23:0]
0b010	1024	ECC[25:0]
0b011	2048	ECC[27:0]
0b100	4096	ECC[29:0]
0b101	8192	ECC[31:0]

38.4.3. SDRAM 控制器寄存器

SDRAM 控制寄存器（EXMC_SDCTLx, x=0, 1）

地址偏移：0x140+4*x, (x = 0, 1)
复位值：0x0000 02D0

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDCLK[2]	PIPED[1:0]		BRSTRD	SDCLK[1:0]		WPEN	CL[1:0]		NBK	SDW[1:0]		RAW[1:0]		CAW[1:0]	
rw	rw		rw	rw		rw	rw		rw	rw		rw		rw	

位/位域	名称	描述
31:16	保留	硬件强制清零。
15	SDCLK[2]	参考 SDCLK[1:0]的描述
14:13	PIPED[1:0]	流水线读数据延迟 这些位用于指定在 CAS 延迟之后再延迟多少个 CK_EXMC 时钟周期才去读数据 00：延迟 0 个 CK_EXMC 周期 01：延迟 1 个 CK_EXMC 周期 10：延迟 2 个 CK_EXMC 周期 11：保留

		注意：寄存器 EXMC_SDCTL1 相应位保留
12	BRSTRD	突发读开关 当该位被置位时，会在 CAS 延迟期间预期处理下一个读命令，并将数据存储到读 FIFO 中。 0：禁用突发读 1：使能突发读 注意：寄存器 EXMC_SDCTL1 相应位保留
11:10	SDCLK[1:0]	SDRAM 时钟配置 这些位指定了两个 SDRAM device 的时钟周期。如果需要修改存储器时钟配置，首先需要将存储器时钟禁用，并且在修改配置后将存储器重新初始化。 000：SDCLK 存储器时钟禁用 001：保留 010：SDCLK 存储器周期 = 2 * CK_EXMC 周期 011：SDCLK 存储器周期 = 3 * CK_EXMC 周期 110：SDCLK 存储器周期 = 4 * CK_EXMC 周期 111：SDCLK 存储器周期 = 5 * CK_EXMC 周期 其他：保留 注意：寄存器 EXMC_SDCTL1 相应位保留 SDCLK[2]位不连续，位于第 15 位。
9	WPEN	写保护 该位禁用写保护功能 0：禁用写保护，允许写访问 1：使能写保护，忽略写访问
8:7	CL[1:0]	CAS 延迟 这些位用于设定 SDRAM CAS 延迟多少个 SDRAM 存储器时钟周期单元 00：保留不使用 01：1 个周期 10：2 个周期 11：3 个周期
6	NBK	内部 Bank 的个数 该位指定内部 Bank 的个数 0：2 个内部 Bank 1：4 个内部 Bank
5:4	SDW[1:0]	SDRAM 数据总线宽度 该位指定 SDRAM 存储器数据总线宽度 00：8 位 01：16 位 10：32 位 11：保留
3:2	RAW[1:0]	行地址位宽 这些位用于指定行地址的比特宽度

00：11 位
01：12 位
10：13 位
11：保留

1:0	CAW[1:0]	列地址位宽 这些位用于指定列地址的比特宽度 00：8 位 01：9 位 10：10 位 11：11 位

SDRAM 时序配置寄存器（EXMC_SDTCFGx, x=0, 1）

地址偏移：0x148+4*x, (x = 0, 1)
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				RCD[3:0]				RPD[3:0]				WRD[3:0]			
				rw				rw				rw			

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

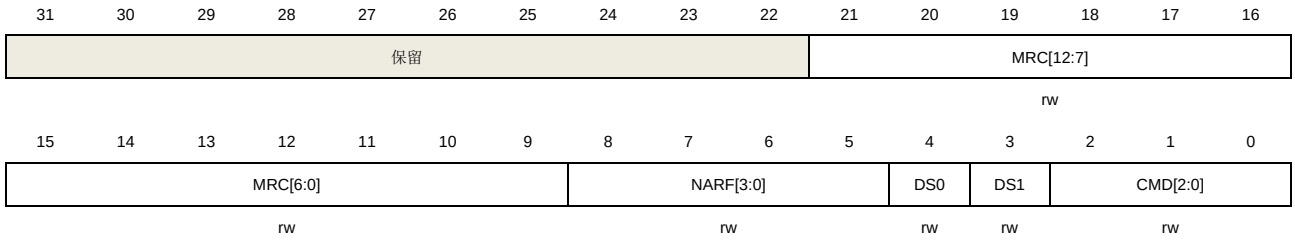
位/位域	名称	描述
31:28	保留	硬件强制清零。
27:24	RCD[3:0]	行到列的延迟 这些位指定了使能命令与读/写命令之间延迟多少 SDRAM 时钟周期单元。 0x0：1 个周期. 0x1：2 个周期 .... 0xF：16 个周期
23:20	RPD[3:0]	行预充电延迟 这些位指定了预充电命令与下一个命令之间延迟多少 SDRAM 存储器时钟周期单元。 0x0：1 个周期 0x1：2 个周期 .... 0xF：16 个周期 注意：寄存器 EXMC_SDTCFG1 相应位保留，如果两个 SDRAM 存储器都被使用，RPD 必须用较慢设备的时序来配置。
19:16	WRD[3:0]	写恢复延迟 这些位指定写命令和预充电命令之间延迟多少 SDRAM 存储器时钟周期单元。 0x0：1 个周期

		0x1：2 个周期 ..... 0xF：16 个周期 注意：寄存器 EXMC_SDTCFG1 相应位保留，如果两个 SDRAM 存储器都被使用，WRD 必须用较慢设备的时序来配置。
15:12	ARFD[3:0]	自动刷新延迟 这些位指定两个连续的刷新命令之间的延迟，在同一个内部 bank 上两个使能命令之间的延迟，以及刷新命令和使能命令之间的延迟，延迟时间以 SDRAM 存储器时钟周期为单位。 0x0：1 个周期 0x1：2 个周期 .... 0xF：16 个周期 注意：寄存器 EXMC_SDTCFG1 相应位保留，如果两个 SDRAM 存储器都被使用，ARFD 必须用较慢设备的时序来配置。
11:8	RASD[3:0]	行地址选择延迟 这些位指定了使能命令与预充电命令之间延迟多少 SDRAM 时钟周期单元，也指定了两个连续的自刷新命令之间的最小延迟。 0x0：1 个周期 0x1：2 个周期 ..... 0xF：16 个周期
7:4	XSRD[3:0]	退出自刷新延迟 这些位指定了从自刷新命令到使能命令之间延迟多少个 SDRAM 存储器时钟周期单元。 0x0：1 个周期 0x1：2 个周期 …… 0xF：16 个周期
3:0	LMRD[3:0]	加载模式寄存器延迟 这些位指定加载模式寄存器命令与刷新或使能命令之间延迟多少 SDRAM 存储器时钟周期单元。 0x0：1 个周期 0x1：2 个周期 ..... 0xF：16 个周期

SDRAM 命令寄存器（EXMC_SDCMD）

地址偏移：0x150
复位值：0x0000 0000

该寄存器只能按字（32位）访问。



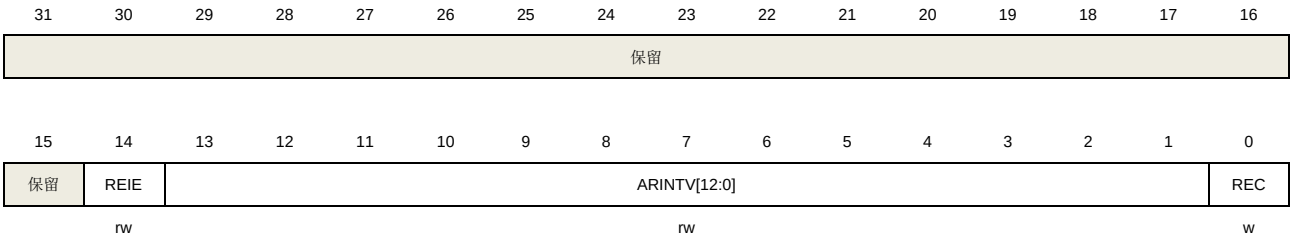
位/位域	名称	描述
31:22	保留	硬件强制清零。
21:9	MRC[12:0]	模式寄存器内容 这些位指定 SDRAM 模式寄存器的内容，这些内容在 CMD = ‘100’时进行编程
8:5	NARF[3:0]	连续的自动刷新个数 这些位指定在 CMD = ‘011’时，发出多少个连续自动刷新周期 0x0：1 个自动刷新周期 0x1：2 个自动刷新周期 .... 0xE：15 个自动刷新周期 0xF：保留
4	DS0	选择 SDRAM Device 0 该位指示 SDRAM Device 0 是否被选择 0：SDRAM Device 0 没有被选择 1：SDRAM Device 0 被选择
3	DS1	选择 SDRAM Device 1 该位指示 SDRAM Device 1 是否被选择 0：SDRAM Device 1 没有被选择 1：SDRAM Device 1 被选择
2:0	CMD[2:0]	命令 这些位指定发送到 SDRAM 设备上的命令 000：正常操作模式 001：时钟使能命令 010：所有存储区预充电命令 011：自动刷新命令 100：加载模式寄存器命令 101：自刷新命令 110：掉电模式进入命令 111：保留 注意：发送命令时，至少需要选择一个设备（DS1 或 DS0）。如果两个设备同时使用，必须同时选择两个设备发送命令。

SDRAM 自动刷新间隔寄存器（EXMC_SDARI）

地址偏移：0x154

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



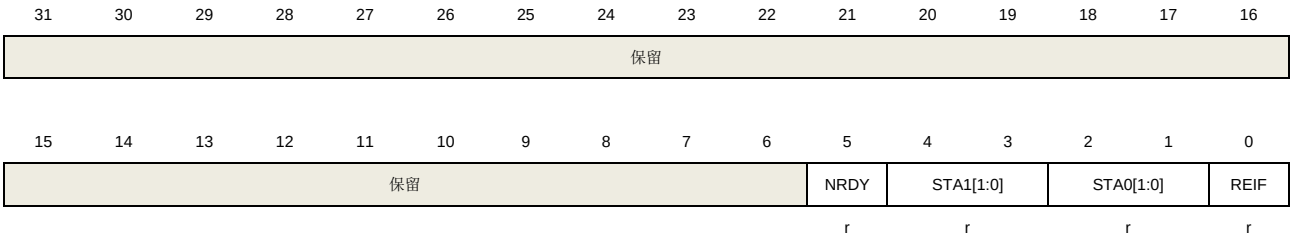
位/位域	名称	描述
31:15	保留	硬件强制清零。
14	REIE	刷新错误中断使能 0：中断禁止 1：状态寄存器 REIF 位置 1 发生中断
13:1	ARINTV[12:0]	自动刷新间隔 这些位指定两个连续的自动刷新命令之间间隔多少存储器时钟周期单元。 ARFITV = (SDRAM 刷新周期/行数) - 20
0	REC	清除刷新错误标志 该位置 1 会清除状态寄存器 REIF 位。 0：没有效果 1：清除刷新错误标志

SDRAM 状态寄存器（EXMC_SDSTAT）

地址偏移：0x158

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	硬件强制清零。
5	NRDY	非就绪状态 该位指定 SDRAM 控制器是否已经准备接收一个新的命令 0：SDRAM 控制器准备好接收新命令 1：SDRAM 控制器没有准备好接收新命令
4:3	STA1[1:0]	Device1 状态

		该位定义 SDRAM Device1 的状态 00：正常状态 01：自刷新状态 10：掉电状态
2:1	STA0[1:0]	Device 0 状态 该位定义 SDRAM Device0 的状态 00：正常状态 01：自刷新状态 10：掉电状态
0	REIF	刷新错误标志 0：无刷新错误 1：出现刷新错误。若中断使能位置 1（RFEIE），则产生中断 REIE

SDRAM 读采样控制寄存器（EXMC_SDRSCTL）

地址偏移：0x180
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	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								SDSC[3:0]				保留		SSCR	RSEN
								rw						rw	rw

位/位域	名称	描述
31:8	保留	硬件强制清零。
7:4	SDSC[3:0]	选择读数据的采样时钟的延迟单元 0x0：0 个延迟单元 0x1：1 个延迟单元 …… 0xF：15 个延迟单元
3:2	保留	硬件强制清零。
1	SSCR	选择读数据的采样周期 0：除延迟之外，为读数据采样时钟增加 0 个额外的 CK_EXMC 周期 1：除延迟之外，为读数据采样时钟增加 1 个额外的 CK_EXMC 周期
0	RSEN	读采样使能 0：禁止读采样 1：使能读采样

39. VREF

39.1. 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。

39.2. 主要特性

内部参考电压特性描述如下：

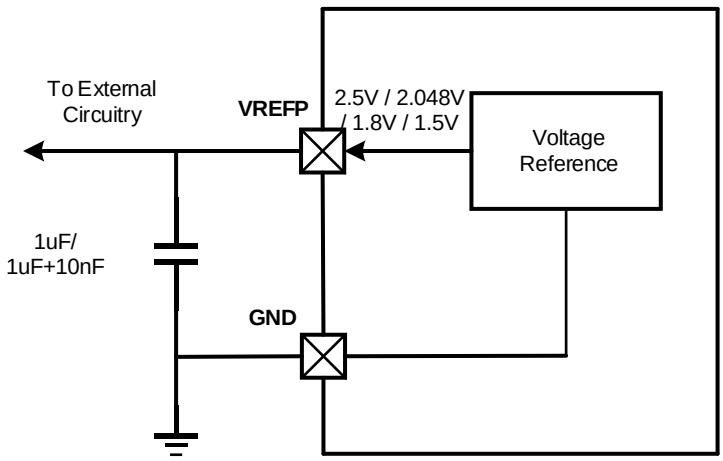
- 电压稳定，产品经过校准；
- 连接 VREFP 引脚可供片外电路使用；
- 1.5V、1.8V、2.048V 或者 2.5V 可配置的参考电压输出。

39.3. 功能描述

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 1.5V、1.8V、2.048V 或者 2.5V 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 1uF（或 1uF 和 10nF 并联）的旁路电容，并接地。

图 39-1. VREF 连接



如下*表 39-1. 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 39-1. 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。

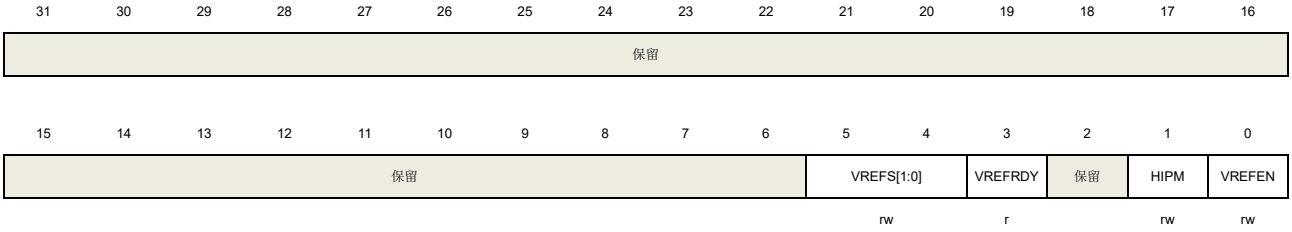
39.4. VREF 寄存器

VREF 基地址：0x5800 3C00

39.4.1. 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。

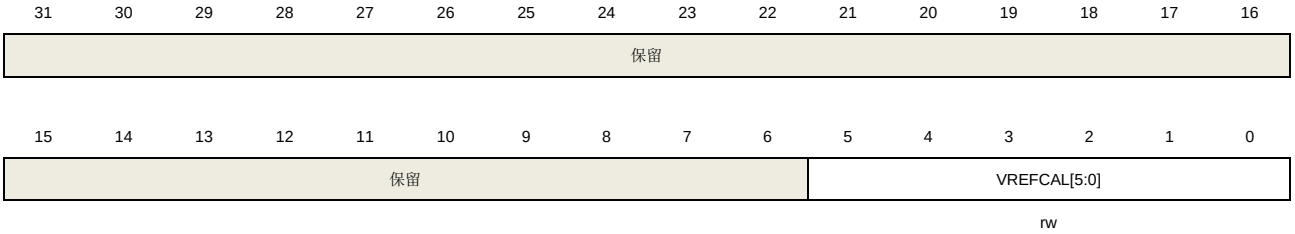


位/位域	名称	描述
31:6	保留	必须保持复位值。
5:4	VREFS[1:0]	参考电压选择 这些位定义了VREF产生的参考电压的数值。 00：参考电压在2.5V左右 01：参考电压在2.048V左右 10：参考电压在1.8V左右 11：参考电压在1.5V左右 此位只有在VREF失能（VREFEN位为0）的时候可以被更改。
3	VREFRDY	VREF就绪 0：VREF输出未就绪 1：VREF输出就绪
2	保留	必须保持复位值。
1	HIPM	高阻抗模式 0：VREFP 引脚内部连接到 VREF 输出 1：VREFP引脚为高阻抗模式
0	VREFEN	VREF使能 0：VREF失能 1：VREF使能

39.4.2. 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 复位后，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。写入这些位可调节内部VREF电压。 **注意：**如果用户执行校准程序，则校准值必须从0x00递增到0x3F。

40. 低功耗数字温度传感器（LPDTS）

40.1. 简介

低功耗数字温度传感器（LPDTS），提供了将温度转换为频率与绝对温度（CLK_PTAT）成正比的方波。其中，频率的测量是基于PCLK或LXTAL时钟。

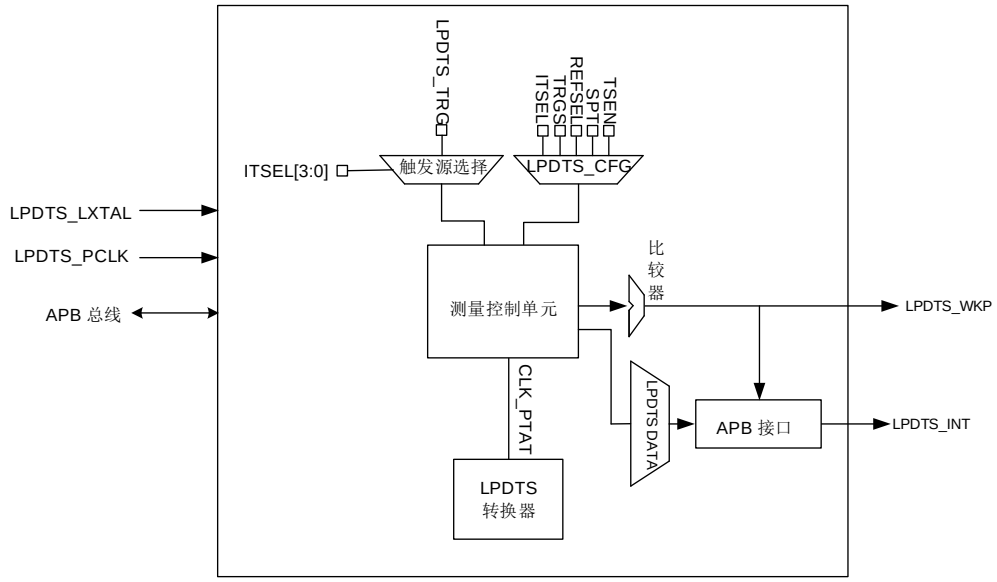
40.2. 主要特征

- 支持软件和硬件的触发源；
- 可编程的采样时间；
- 温度窗口看门狗；
- 温度低于/高于预设阈值以及在测量结束时产生中断；
- 温度高于/低于预定义阈值时产生的异步唤醒信号（仅LXTAL作为参考时钟）。

40.3. 模块框图

*图 40-1. LPDTS 模块框图*描述了 LPDTS 的模块框图。

图 40-1. LPDTS 模块框图



40.4. 功能说明

40.4.1. LPDTS 内部信号

表 40-1. LPDTS 信号

信号	类型	描述
LPDTS_LXTAL	输入	LXTAL时钟
LPDTS_PCLK	输入	APB时钟
LPDTS_INT	输出	模块中断
LPDTS_WKP	输出	模块唤醒

40.4.2. 操作模式

通过设置LPDTS_CFG中的REFSEL位，可以选择多种操作模式。

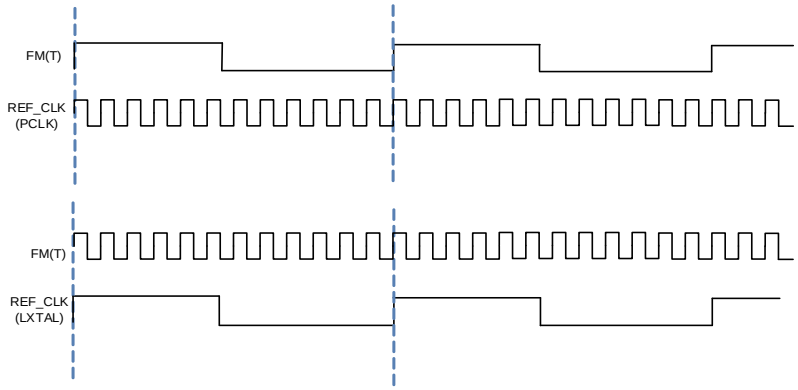
- PCLK模式（REFSEL = 0）
  寄存器能够通过软件进行读写。当REFSEL位设置为0，选择PCLK作为参考时钟。
- PCLK和LXTAL模式（REFSEL = 1）
  寄存器能够通过软件进行读写。当REFSEL位设置为1，选择LXTAL作为参考时钟。
- LXTAL模式（REFSEL = 1并且PCLK时钟关闭）
  此模式下温度传感器的寄存器无法访问，选择LXTAL作为参考时钟，从而使用硬件触发来退出深度睡眠模式。

40.4.3. 温度测量原理

温度传感器的模拟部分能够将温度转换为方波信号输出，其中，信号的FM(T)频率通常为641KHz。LPDTS模块内嵌了两个计数器，其计数方式和选择的参考时钟相关，计数结果存储在LPDTS_DATA寄存器中。

- 当参考时钟为 PCLK 时，测量方法为采样一个或多个 FM(T)周期，并在 PCLK 上升沿和下降沿计数。
- 当参考时钟为 LXTAL 时，测量方法为采样一个或多个 LXTAL 周期，并在 FM(T)上升沿和下降沿计数。

图 40-2. 测量方式



当参考时钟为PCLK时温度计算公式如下：

$$T=T0+((2\times F_{PCLK}/COVAL)\times SPT-100\times FREQ)/RF_CF \quad (40\text{-}1)$$

当参考时钟为LXTAL时温度计算公式如下：

$$T=T0+(((F_{LXTAL}\times COVAL)/(2\times SPT))-(100\times FREQ))/RF_CF \quad (40\text{-}2)$$

其中：

- T0 等于 25°C；
- COVAL 为温度传感器计数器输出值，其值存储在 LPDTS_DATA 寄存器中；
- SPT 为模块采样时间；
- FREQ 是温度传感器在温度为 T0 时测量并存储在 LPDTS_SDATA 寄存器中的频率值，其通常为几百 Hz；
- RF_CF 为温度传感器斜坡系数。

40.4.4. 采样时间

LPDTS 的测量精度可通过增加采样周期来提高，当参考频率设置在采样频率附近时效果最好。

采样时间的默认值应当设置为一个 REF_CLK 周期或一个 FM(T)周期，对应的模式有 LXTAL 模式和 PCLK 模式。

采样时间是通过 LPDTS_CFG 寄存器中的 SPT 位配置的。如*表 40-2. 采样时间设置*所示。

表 40-2. 采样时间设置

SPT[3:0]	LXTAL或FM(T)时钟周期（s）
0000	1
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6

SPT[3:0]	LXTAL或FM(T)时钟周期（s）
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

40.4.5. 触发源

温度测量可以由软件或外部事件触发。触发器源可以通过LPDTS_CFG中的ITSEL[3:0]位选择。

- 软件触发

  当在LPDTS_CFG中将ITSEL[3:0]设置为 '0000' 时，选择软件触发器。

  - 检查LPDTS_STAT中的TSRF是否设置为1。当TSRF位置1后，通过设置LPDTS_CFG寄存器中的TRGS位来开始温度测量。否则，忽略此步骤。
  - 测量完成后，如果TRGS位仍为1，当TSRF标志位变为1时，测量将重新开始。

- 硬件触发

  硬件仅能在TSRF设置为1时才能检测到触发源的上升沿信号，否则，该触发源信号将被忽略。

表 40-3. 触发源设置

编号	ITSEL[3:0]				描述
NA	0	0	0	0	无硬件触发源
0001	0	0	0	1	保留
0010	0	0	1	0	
0011	0	0	1	1	
0100	0	1	0	0	LPDTS_TRG
0101	0	1	0	1	保留
0110	0	1	1	0	
0111	0	1	1	1	
1000	1	0	0	0	
1001	1	0	0	1	
1010	1	0	1	0	
1011	1	0	1	1	
1100	1	1	0	0	
1101	1	1	0	1	
1110	1	1	1	0	
1111	1	1	1	1	

注意：LPDTS_TRG信号来源于TRIGSEL模块。TRIGSEL模块中TRIGSEL_LPDTS寄存器的INSELx[7:0]位用于选择LPDTS_TRG信号的触发输入源。

40.4.6. 开关控制

通过设置LPDTS_CFG寄存器中的TSEN位来启用LPDTS模块。温度传感器状态寄存器（LPDTS_STAT）中的TSRF标志置位表明LPDTS模块已准备好进行温度测量：当TSRF位设置为1时，LPDTS模块可以开始温度测量。一旦测量开始，TSRF位被重置，新的测量请求将被忽略。如果需要进行新的测量，则需要等待最后一次测量完成，并且再次设置TSRF位。

40.4.7. LPDTS 低功耗模式

表 40-4. 低功耗描述

模式	描述
睡眠模式	此模式下参考时钟为LXTAL或PCLK LPDTS中断会导致模块退出睡眠模式
深度睡眠模式	此模式下参考时钟为LXTAL LPDTS中断会导致模块退出深度睡眠模式

40.4.8. LPDTS 中断

LPDTS中断可以连接到CPU NVIC或EXTI控制器。

- LPDTS模块可以在以下两种情况下产生中断：
  - 测量结束时
  - 测量结果高于或低于预定义阈值
- LPDTS模块中有以下两种中断：
  - 同步中断：通过设置LPDTS_INTEN寄存器选择3个中断事件
  - 异步唤醒：通过设置LPDTS_INTEN寄存器中选择3个异步唤醒事件
- 允许所有中断的组合。

注意:只有选择LXTAL作为参考时钟时，才能使用异步唤醒。

*表40-5. 低功耗模式下温度传感器行为*显示了中断位及其描述。

表 40-5. 低功耗模式下温度传感器行为

中断事件	中断标志	中断使能位	中断清除位	退出睡眠模式	退出深度睡眠模式	同步/异步
测量结束	EMIF in LPDTS_STAT	EMIE in LPDTS_INTEN	EMIC in LPDTS_INTC	是	否	与PCLK同步
低于低阈值	LTIF in LPDTS_STAT	LTIE in LPDTS_INTEN	LTIC in LPDTS_INTC	是	否	
高于高阈值	HTIF in LPDTS_STAT	HTIE in LPDTS_INTEN	HTIC in LPDTS_INTC	是	否	
测量结束	EMAIF in LPDTS_STAT	EMAIE in LPDTS_INTEN	EMAIC in LPDTS_INTC	是	是	异步
低于低阈值	LTAIF in LPDTS_STAT	LTAIFE in LPDTS_INTEN	LTAIC in LPDTS_INTC	是	是	
高于高阈值	HTAIF in LPDTS_STAT	HTAIE in LPDTS_INTEN	HTAIC in LPDTS_INTC	是	是	

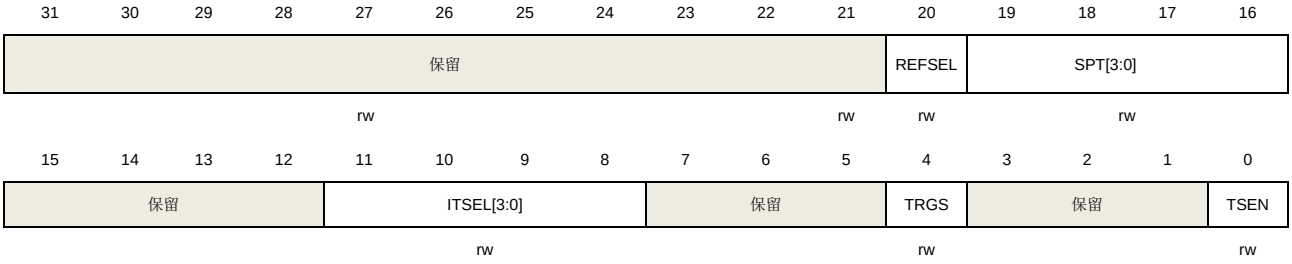
40.5. LPDTS 寄存器

LPDTS基地址：0x5800 6800

40.5.1. LPDTS 配置寄存器（LPDTS_CFG）

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字（32位）访问。



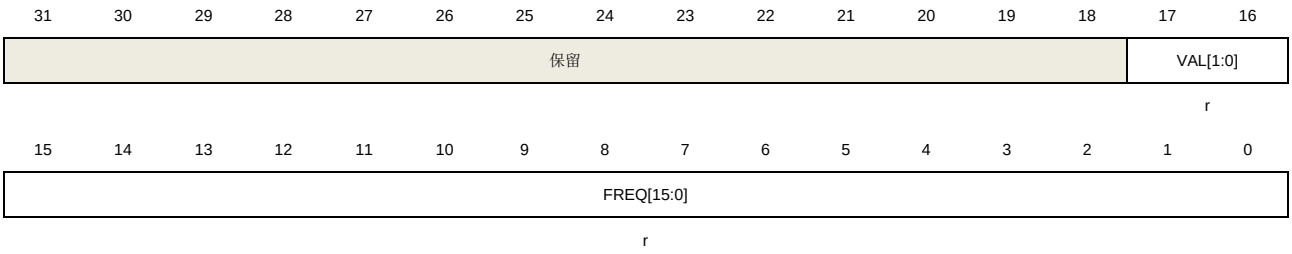
位/位域	名称	描述
31:21	保留	必须保持复位值。
20	REFSEL	参考时钟选择位 0：高速参考时钟（PCLK） 1：低速参考时钟（LXTAL）
19:16	SPT[3:0]	采样时间 采样时间的增加有利于提高采样精度
15:12	保留	必须保持复位值。
11:8	ITSEL[3:0]	触发输入选择位 设置此位能够选择温度测量的触发输入源
7:5	保留	必须保持复位值。
4	TRGS	频率测量触发选择位 0：无软件触发 1：当模块准备完成时软件触发频率测量
3:1	保留	必须保持复位值。
0	TSEN	使能LPDTS模块 0：禁止LPDTS模块 1：使能 LPDTS 模块

40.5.2. T0 传感器数据寄存器（LPDTS_SDATA）

地址偏移：0x08

复位值：0x000X XXXX

该寄存器只能按字（32位）访问。



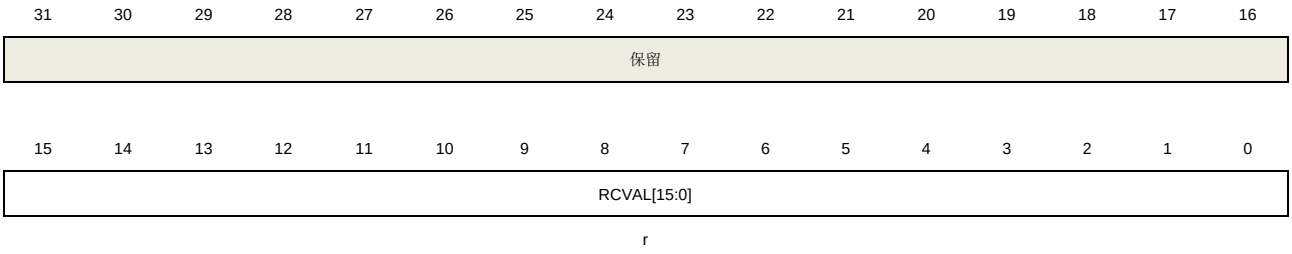
位/位域	名称	描述
31:18	保留	必须保持复位值。
17:16	VAL[1:0]	采集温度值 采集到的T0温度值 0x00：25 °C 0x01：-40 °C 其他：保留
15:0	FREQ[15:0]	频率值 当外界环境为T0时采集到的频率值 **注意**：采集单位设置为 0.1 kHz

40.5.3. 斜率数据寄存器（LPDTS_RDATA）

地址偏移：0x10

复位值：0xXXXX XXXX

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	RCVAL[15:0]	斜率值 这些位定义了LPDTS模块的温度测量斜率值 **注意**：采集单位设置为 1 Hz/°C

40.5.4. 中断阈值寄存器（LPDTS_IT）

地址偏移：0x014
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:16	INTHT[15:0]	中断高阈值 这些位定义了温度中断高阈值，当采集值高于此值时将产生中断。
15:0	INTLT[15:0]	中断低阈值 这些位定义了温度中断低阈值，当采集值低于此值时将产生中断。

40.5.5. 温度值寄存器（LPDTS_DATA）

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
COVAL[15:0]															
r															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	COVAL[15:0]	计数器输出值

40.5.6. 温度传感器状态寄存器（LPDTS_STAT）

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
TSRF	保留								HTAIF	LTAIF	EMAIF	保留	HTIF	LTIF	EMIF
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	TSRF	温度传感器准备标志 0：温度传感器准备未完成 1：温度传感器准备完成
14:7	保留	必须保持复位值。
6	HTAIF	高阈值异步中断标志位 当达到温度高阈值并且HTAIE位置位时，此位被硬件置位。 当LPDTS_INTC寄存器中的HTAIC位置位时，此位被软件复位。 0：未产生高阈值异步中断 1：产生高阈值异步中断
5	LTAIF	低阈值异步中断标志位 当达到温度低阈值并且LTAIE位置位时，此位被硬件置位。 当LPDTS_INTC寄存器中的LTAIC位置位时，此位被软件复位。 0：未产生低阈值异步中断 1：产生低阈值异步中断
4	EMAIF	测量完成异步中断标志位 当达到温度测量完成并且EMAIE位置位时，此位被硬件置位。 当LPDTS_INTC寄存器中的EMAIC位置位时，此位被软件复位。 0：未产生温度测量完成异步中断 1：产生温度测量完成异步中断
3	保留	必须保持复位值。
2	HTIF	高阈值中断标志位 当达到温度高阈值并且HTIE位置位时，此位被硬件置位（与PCLK同步）。 当LPDTS_INTC寄存器中的HTIC位置位时，此位被软件复位。 0：未产生高阈值中断 1：产生高阈值中断
1	LTIF	低阈值中断标志位 当达到温度低阈值并且LTIE位置位时，此位被硬件置位（与PCLK同步）。 当LPDTS_INTC寄存器中的LTIC位置位时，此位被软件复位。 0：未产生低阈值中断 1：产生低阈值中断
0	EMIF	测量完成中断标志位

当达到温度测量完成并且EMIE位置位时，此位被硬件置位（与PCLK同步）。
当LPDTS_INTC寄存器中的EMIC位置位时，此位被软件复位。
0：未产生温度测量完成中断
1：产生温度测量完成中断

40.5.7. 中断使能寄存器（LPDTS_INTEN）

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									HTAIE	LTAIE	EMAIE	保留	HTIE	LTIE	EMIE
									rw	rw	rw		rw	rw	rw

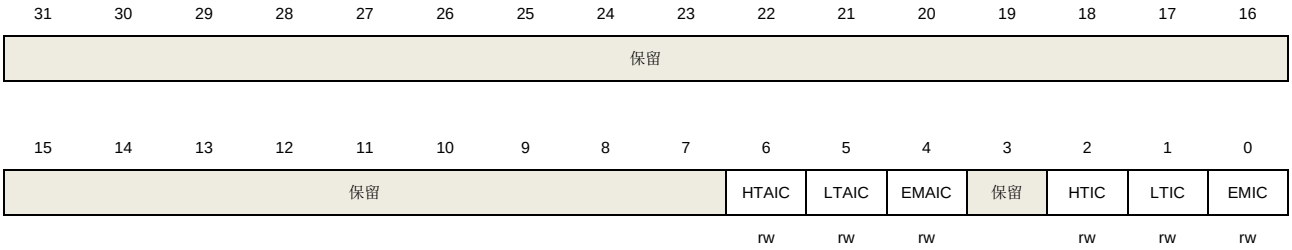
位/位域	名称	描述
31:7	保留	必须保持复位值。
6	HTAIE	高阈值异步中断使能位 通过软件复位/置位来失能/使能高阈值异步中断（仅当REFSEL = 1有效） 0：失能高阈值异步中断 1：使能高阈值异步中断
5	LTAIE	低阈值异步中断使能位 通过软件复位/置位来失能/使能低阈值异步中断（仅当REFSEL = 1有效） 0：失能低阈值异步中断 1：使能低阈值异步中断
4	EMAIE	测量完成异步中断使能位 通过软件复位/置位来失能/使能测量完成异步中断（仅当REFSEL = 1有效） 0：失能测量完成异步中断 1：使能测量完成异步中断
3	保留	必须保持复位值。
2	HTIE	高阈值中断使能位 通过软件复位/置位来失能/使能高阈值中断（与PCLK同步） 0：失能高阈值中断 1：使能高阈值中断
1	LTIE	低阈值中断使能位 通过软件复位/置位来失能/使能低阈值中断（与PCLK同步） 0：失能低阈值中断 1：使能低阈值中断

0	EMIE	测量完成中断使能位 通过软件复位/置位来失能/使能测量完成中断（与PCLK同步） 0：失能测量完成中断 1：使能测量完成中断

40.5.8. 中断标志清除寄存器（LPDTS_INTC）

地址偏移：0x28
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

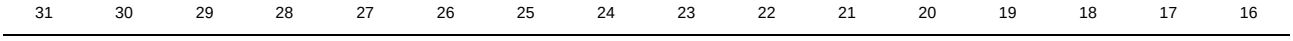


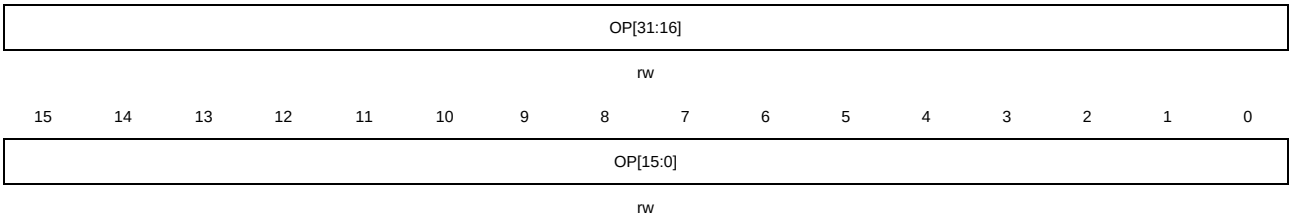
位/位域	名称	描述
31:7	保留	必须保持复位值。
6	HTAIC	高阈值异步中断清除位 通过软件置位来清除LPDTS_STAT寄存器中的HTAIF标志位
5	LTAIC	低阈值异步中断清除位 通过软件置位来清除LPDTS_STAT寄存器中的LTAIF标志位
4	EMAIC	测量完成异步中断清除位 通过软件置位来清除LPDTS_STAT寄存器中的EMAIF标志位
3	保留	必须保持复位值。
2	HTIC	高阈值中断清除位 通过软件置位来清除LPDTS_STAT寄存器中的HTIF标志位
1	LTIC	低阈值中断清除位 通过软件置位来清除LPDTS_STAT寄存器中的LTIF标志位
0	EMIC	测量完成中断清除位 通过软件置位来清除LPDTS_STAT寄存器中的EMIF标志位

40.5.9. 选择寄存器（LPDTS_OP）

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。





位/位域	名称	描述
31:0	OP [31:0]	通用选项位

41. 编码器分频输出控制器（EDOUT）

41.1. 简介

编码器分频输出控制器（EDOUT）用于把从编码器获取到的位置信息，以A相、B相和Z相脉冲的方式输出。

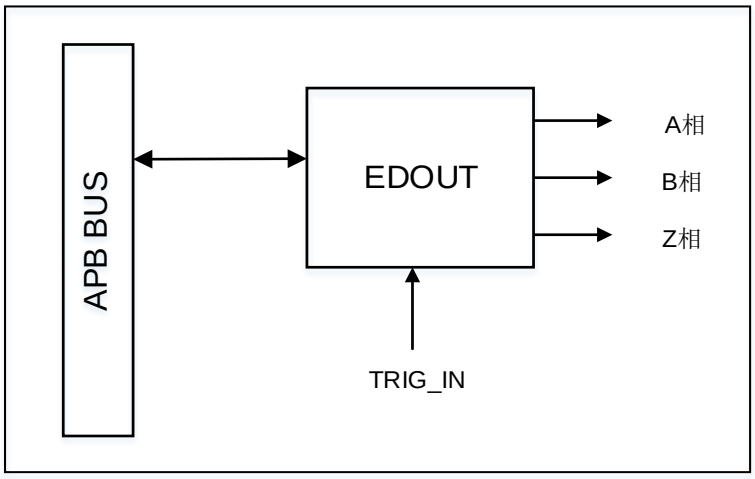
41.2. 主要特征

- 支持更改B相极性状态。
- 支持配置Z相输出位置和脉冲宽度。
- 每旋转一圈的边沿数量：16至65536（必须为4的倍数）。
- 支持从TRIGSEL获取更新周期事件输入信号。

41.3. 功能说明

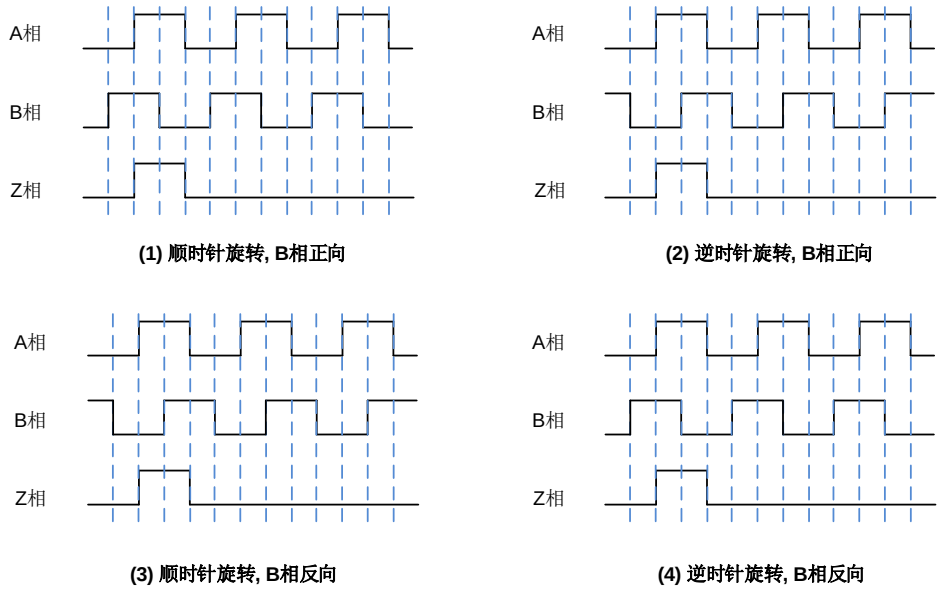
EDOUT模块结构框图如*图41-1. EDOUT模块结构框图*。A相、B相和Z相代表相关的脉冲信号输出引脚，TRIG_IN表示从TRIGSEL模块输入的信号。

图 41-1. EDOUT 模块结构框图



EDOUT输出波形与增量式编码器输出信号比较相似。通过对EDOUT的设置，可以反映出当前的位置信息，如*图41-2. ABZ相输出波形*所示。

图 41-2. ABZ 相输出波形



(1) 顺时针旋转, B相正向　(2) 逆时针旋转, B相正向

(3) 顺时针旋转, B相反向　(4) 逆时针旋转, B相反向

通过对寄存器的配置，EDOUT 可以将增量式编码器或者绝对式编码器的位置信息转换到 AB 相和 Z 相输出信号。

除了EDOUT初始设置外，CPU在每个更新周期内，从编码器获得位置变化量并更新EDOUT寄存器，得到AB相和Z相输出信号。

此时，来自定时器模块的事件信号必须通过TRIGSEL输入至EDOUT。

41.4. Z 相输出模式

Z相支持以下两种操作模式：

■ 操作模式0

Z相输出由当前位置决定。用户设置ZCR寄存器的ZOSP位域和ZOWH位域，在当前位置（EDOUT_LCNT寄存器的LOCCNT反映当前位置）与ZOSP相匹配时，Z相开始输出信号且脉冲宽度为ZOWH个边沿。

■ 操作模式1

Z相输出由边沿序号决定。用户设置ZCR寄存器的ZOSP位域和ZOWH位域，在一个更新周期内，当输出边沿序号与ZOSP匹配时，Z相开始输出信号且脉冲宽度为ZOWH个边沿。

在上面两种操作模式中，若将ZOWH位域设置为0，则表示Z相不输出。需要注意，必须在EDOUT运行时（即EDOUT_ENABLE寄存器中的EDOUTEN位为1），配置ZCR寄存器；否则，Z相不会产生有效输出。

41.5. 操作指导

EDOUT根据寄存器设置输出AB相和Z相信号。AB相和Z相输出的更新周期由TRIGSEL的输入决定。使用定时器等产生更新周期，从TRIGSEL输出信号至EDOUT。具体的设置步骤如下所

述。

41.5.1. EDOUT 初始化

EDOUT初始化处理步骤如下：

1. 配置EDOUT相关的输出引脚
2. EDOUT模块初始化配置

   对EDOUT_CTL寄存器和EDOUT_LOC寄存器进行初始化配置。例如，当B相输出为反向状态且每旋转一圈的边沿数量为88（4x22）。此时，需要设置EDOUT_CTL寄存器的POL位为1，EDOUT_LOC寄存器的LOCMAX位域为87（4x22-1）。

3. EDOUT_LCNT寄存器设置初始值

   初始值设置为绝对式编码器或增量式编码器的初始位置，用于产生AB相和Z相信号，由EDOUT_LCNT寄存器的LOCCNT位域表示，取值范围是0到LOCMAX。位置值的计算公式是[编码器位置值]×[每旋转一圈的边沿数量]/[编码器位置值的分辨率]（向下舍入）。

   例如，每旋转一圈的边沿数量为88，编码器位置值的分辨率为20位（1048576），初始值为931802。此时，需要将EDOUT_LCNT寄存器的LOCCNT位域设置为78（931802×88/1048576）。

4. 设置TRIGSEL

   例如，当选择定时器作为EDOUT输入的触发源，设置定时器生成更新周期。当然，EDOUT还支持其他触发源，但触发器输出的高电平宽度必须大于EDOUT的T_{PCLK}。

5. 使能AB相和Z相输出

   配置EDOUT_ENABLE寄存器的EDOUTEN位为1，使能EDOUT输出。

6. 启动定时器

   按照步骤4的TRIGSEL设置，定时器启动开始计数，生成更新周期信号。

41.5.2. EDOUT 更新处理

EDOUT更新处理步骤如下：

1. 获取位置信息

   从绝对式编码器或增量式编码器获取位置信息。

2. 设置EDOUT_OCNT寄存器值

   首先，计算写入EDOUT_OCNT寄存器PDC位域和EDGC位域的值。当前位置可以用0到LOCMAX表示，计算过程如下：

   （m）=[步骤1中获取的位置]×[每旋转一圈的边沿数量]/[编码器位置值的分辨率]（向下舍入）

此时，EDGC位域的值为：

（n）=（m）–[前一次计算的值(m)]

当（n）的绝对值大于旋转边沿数量的一半时，则：

A. 如果（n）为正数，则EDGC值为（n）－[每旋转一圈的边沿数量]。

B. 如果（n）为负数，则EDGC值为（n）+[每旋转一圈的边沿数量]。

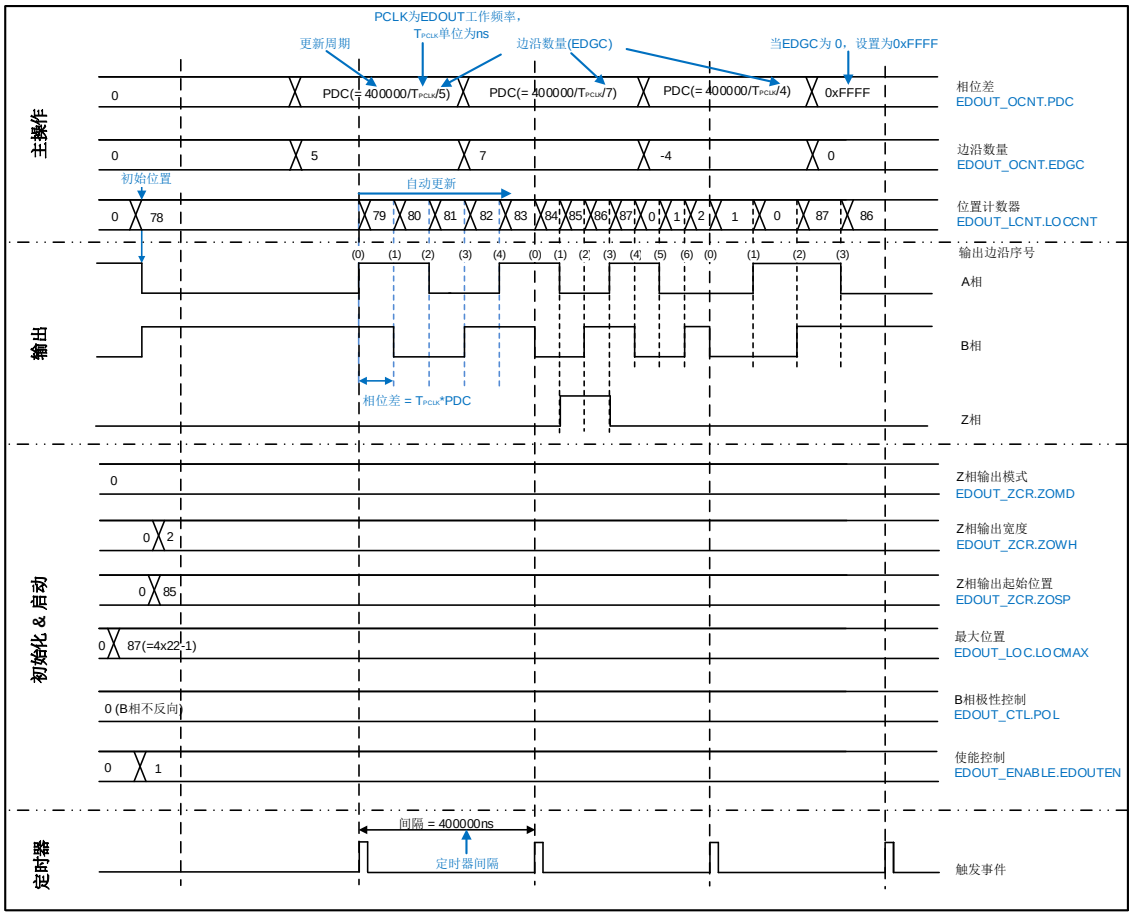
当EDGC位域不为0时，PDC位域为[更新周期]/[T_{PCLK}]/[EDGC位域的绝对值]（向下舍入）。当EDGC位域为0时，PDC位域配置为0xFFFF。

在EDOUT_OCNT寄存器中设置计算得到的PDC和EDGC值。

41.5.3. EDOUT 工作案例

如*图41-3. EDOUT 设置案例和AB 相及Z 相输出波形*所示。在该示例中，设置定时器来生成更新周期，并选择定时器触发输出作为EDOUT输入。更新周期为400000ns，每旋转一圈的边沿数量为88，B相极性为正，Z相输出由当前位置决定。同时，用0到LOCMAX表示的编码器位置值从初始值78变化到83、2和86。

图 41-3. EDOUT 设置案例和 AB 相及 Z 相输出波形



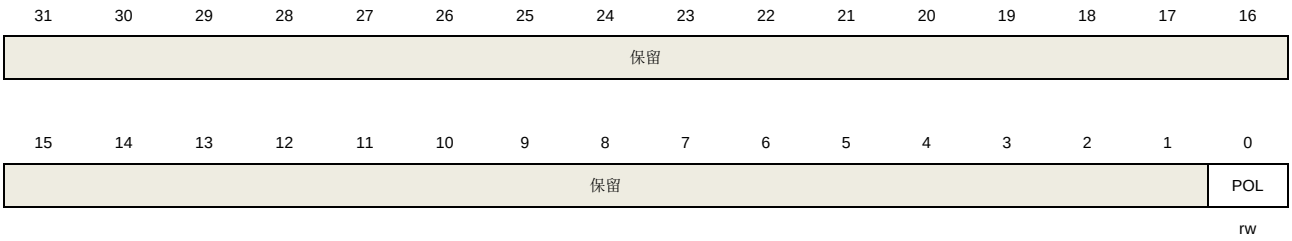
41.6. EDOUT 寄存器

EDOUT 基地址：0x4001 8800

41.6.1. 控制寄存器（EDOUT_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
31:1	保留	必须保持复位值。
0	POL	B 相工作极性 该位用于选择 B 相输出信号极性。如果 EDOUT_ENABLE 寄存器的 EDOUTEN 位为 0，则该位的设置会反映在 B 相输出。否则，此位设置无效。 0：工作极性为正向 1：工作极性为反向

41.6.2. 使能寄存器（EDOUT_ENABLE）

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



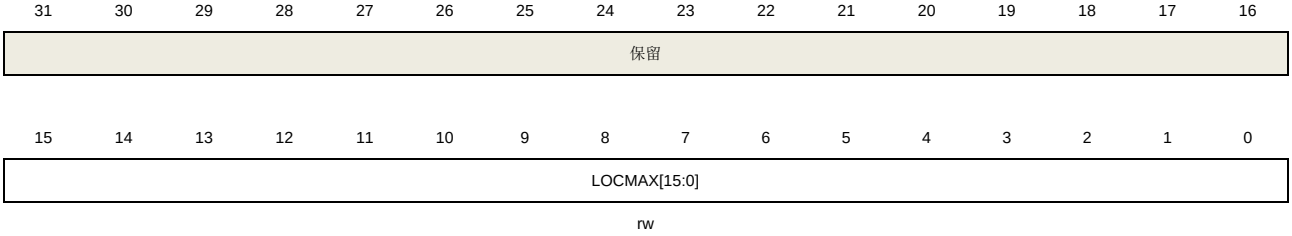
位/位域	名称	描述
31:1	保留	必须保持复位值。
0	EDOUTEN	EDOUT 使能位

当该位设置为 0 时，EDOUT_LCNT 寄存器配置完成后，AB 相会立即输出相对应的状态，Z 相输出值为 0。当该位设置为 1 时，EDOUT 启动并输出 AB 相和 Z 相信号。
0：禁止 EDOUT
1：使能 EDOUT

41.6.3. 位置寄存器（EDOUT_LOC）

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

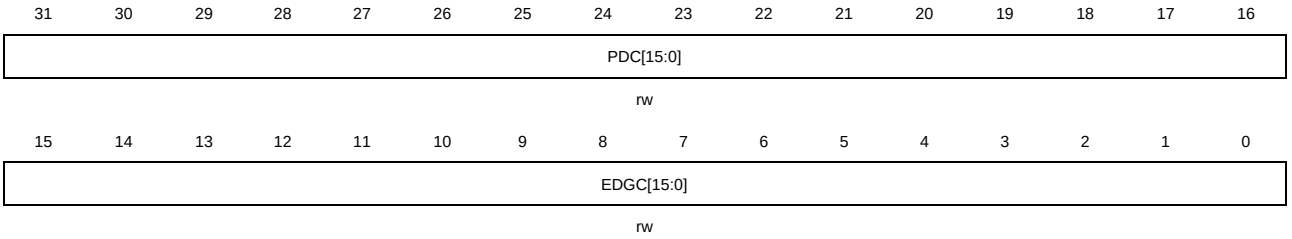


位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	LOCMAX[15:0]	最大位置 通过该位域设置一次旋转的最大位置。最大位置必须为 4 的倍数。如果最大位置为“4×M”，则在此寄存器中设置“4×M－1”。当 EDOUT_ENABLE 寄存器的 EDOUTEN 位从 0 更改为 1 时，该位域设置生效。 0x0000~0x000E：保留 0x000F：最大位置为 16 ... 0xyyyy：最大位置为 0xyyyy+1

41.6.4. 输出计数器寄存器（EDOUT_OCNT）

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	PDC[15:0]	相位差 该位域设置下一更新周期 A 相与 B 相信号之间的相位差。设定值的允许范围为 2 至 65535，单位为 T_{PCLK}。当 EDGC 位域设置为 0 时，将该位域设置为 65535；当不为 0 时，将其设置为“更新周期/T_{PCLK}/EDGC 的绝对值”（向下舍入）。当 EDOUT 运行时（即 EDOUT_ENABLE 寄存器的 EDOUTEN 位为 1），请确保在下一个更新周期事件之前设置该位域。
15:0	EDGC[15:0]	边沿数量 该位域设置下一更新周期的 A 相与 B 相信号的边沿数量。如果使用反方向旋转，需要设置为二进制补码表示的负值，值的允许范围为-32767（0x8001）到 32767（0x7FFF）。该位域的绝对值不得大于“更新周期/（2*T_{PCLK}）”。当 EDOUT 运行时（即 EDOUT_ENABLE 寄存器中的 EDOUTEN 位为 1），请确保在下一个更新周期事件之前设置该位域。

41.6.5. 位置计数寄存器（EDOUT_LCNT）

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
LOCCNT[15:0]															
rw															

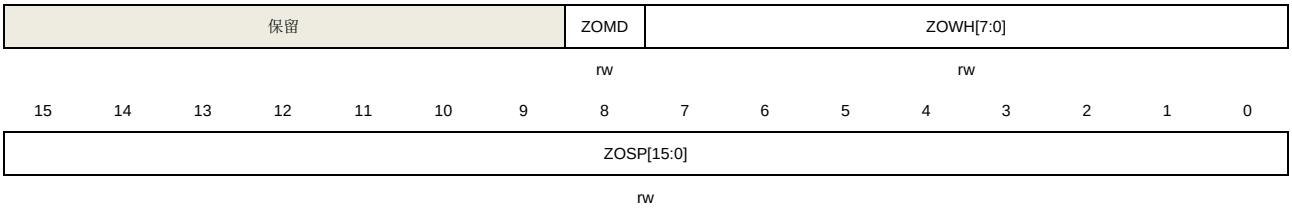
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	LOCCNT[15:0]	当前位置 该位域用在 EDOUT 停止时（即 EDOUT_ENABLE 寄存器的 EDOUTEN 位为 0），设置当前位置（设置范围为 0 到 LOCMAX）。当前位置设置完成后，A 相和 B 相会立即输出相应状态。当 EDOUT 运行时（即 EDOUT_ENABLE 寄存器的 EDOUTEN 位为 1），这些位反映出与 A 相和 B 相输出相关的位置变化。

41.6.6. Z 相配置寄存器（EDOUT_ZCR）

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



位/位域	名称	描述
31:25	保留	必须保持复位值。
24	ZOMD	Z 相输出模式 0：当前位置决定输出 1：边沿序号决定输出
23:16	ZOWH[7:0]	Z 相输出宽度
15:0	ZOSP[15:0]	Z 相输出起始位置

42. 控制器局域网络（CAN）

42.1. 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。

注：GD32H737xx 系列不支持 CAN FD 帧。

42.2. 主要特征

- 支持CAN总线协议2.0A/B；
- 遵循ISO 11898-1:2015规范；
- 支持CAN FD帧，最大64字节数据，通信波特率最大为8 Mbit/s；
- 支持CAN常规帧，最大8字节数据，通信波特率最大为1 Mbit/s；
- 支持发送和接收时间戳，基于16位内部计数器；
- 支持传输延迟补偿，用于CAN FD帧的高速率数据阶段；
- 中断可配置屏蔽；
- 支持4种通信模式：正常模式，暂停模式，回环静默模式，和监听模式；
- 支持2种省电模式：CAN_Disable模式，和虚拟联网模式；
- 支持2种从虚拟联网模式唤醒的方式：唤醒匹配事件，和唤醒超时事件；
- 最大32个邮箱，此时每个邮箱都配置为8字节数据长度，可灵活配置为发送或接收邮箱；
- 支持通过一个特殊帧同步全局网络时间。
- 协议控制器的时钟（CANCLK）源可配置(1)。

发送

- 支持发送中止；
- 发送邮箱状态可查看；
- 发送帧消息的CRC校验；
- 支持发送优先级：最小邮箱号优先，或最高优先级优先。

接收

- 接收私有过滤寄存器用于每个接收邮箱或者接收FIFO；
- 接收邮箱公有过滤寄存器用于接收邮箱，接收FIFO公有过滤寄存器用于接收FIFO；
- 支持接收优先级，可配置在匹配阶段的接收邮箱和接收FIFO的优先级；
- 接收FIFO的标识符过滤功能支持最大104个扩展标识符的过滤，或者208个标准标识符的过滤，或者416个对标识符部分8位的过滤；

- 深度为6帧的接收FIFO，支持DMA功能。

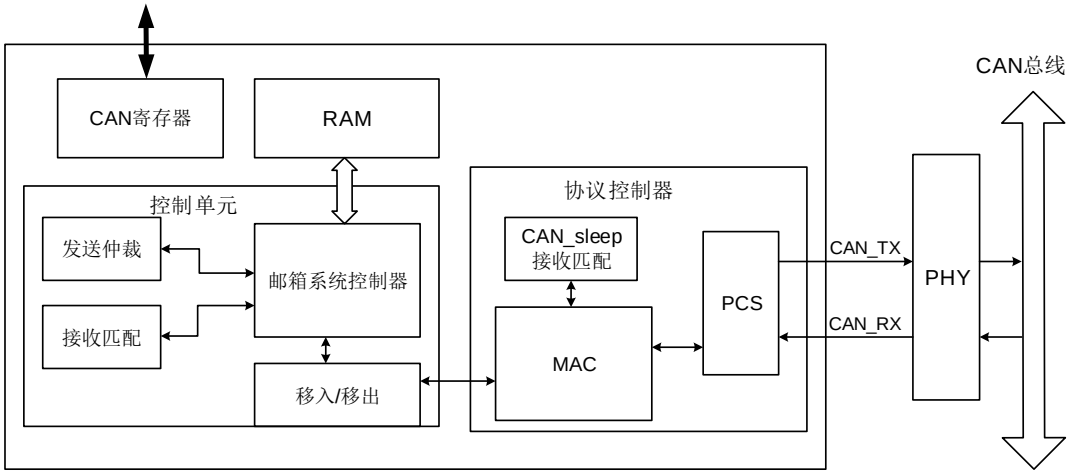
注意：

（1）CANCLK 时钟源的选择有使用限制，请查看 GD32H73x_75x 系列的勘误手册。

42.3. 功能说明

CAN 模块结构框图如*图 42-1. CAN 模块结构框图*所示。

图 42-1. 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APB总线的交互。

42.3.1. 邮箱描述符

邮箱描述符如*表42-1. 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 42-1. 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 42-3. 接收邮箱 CODE* 和*表 42-4. 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 42-2. 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 42-3. 接收邮箱 CODE

CODE	含义	接收后的 CODE	完成服务(1)	RRFRMS(2)	描述
0b0000	INACTIVE	-	-	-	邮箱不参与匹配过程。
0b0100	EMPTY	FULL	-	-	当成功接收了一个帧后（在移入过程之后），CODE域自动更新为FULL。
0b0010	FULL	FULL	是	-	保持为FULL。如果新的一帧在该邮箱完成服务之后移入该邮箱，则邮箱代码保持为FULL。
		OVERRUN	否		如果邮箱代码已经为FULL，而在该邮箱完成服务之前又有新的一帧移入该邮箱，则邮箱代码自动更新为OVERRUN。
0b0110	OVERRUN	FULL	是	-	如果邮箱代码为OVERRUN，在邮箱完成服务之后有新的一帧移入了该邮箱，则邮箱代码更新为FULL。
		OVERRUN	否		如果邮箱代码为OVERRUN，而有新的一帧必须移入，则该邮箱将再次被覆盖，邮箱代码保持为OVERRUN。
0b1010	RANSWER(3)	TANSWER(0x1110)	-	0	邮箱代码为RANSWER的邮箱用于远程请求帧接收的识别。在接收远程请求帧之后，如果CAN_CTL2寄存器的RRFRMS位为0，则该邮箱将自动设置发送一个响应帧，邮箱代码自动修改为TANSWER。
		-		1	邮箱在接收匹配和发送仲裁过程中被忽略。
CODE[0] = 1	BUSY(4)	FULL	-	-	表示邮箱正在更新。
		OVERRUN			

1. 完成服务：邮箱被CPU读取过，并且通过读取CAN_TIMER寄存器或者读取其他邮箱的方式解锁了该邮箱。
2. 远程请求帧存储位，参考***控制寄存器2（CAN_CTL2）***。
3. 邮箱代码为0b1010的不可被中止。
4. 对于接收邮箱，如果CODE[0]位置位，则对应的邮箱将不参与接收匹配过程。

表 42-4. 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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRIO[2:0]			ID_STD[10:0]											ID_EXD[17:16]	
rw			rw											rw	

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

位/位域	名称	描述
31:29	PRIO[2:0]	本地优先级 该位域只有当 CAN_CTL0 寄存器的 LAPRIOEN 位为 1 时才适用。 该位域仅用于发送邮箱，但在发送消息时不发送这些位，它们附加到标识符之前，共同用作发送优先级的判断。
28:18	ID_STD[10:0]	标准帧的标识符 对于标准帧，这 11 个有效意义位用作发送接收帧的标识符。邮箱描述符字 1 的低 18 位忽略不用。
17:0	ID_EXD[17:0]	扩展帧的标识符 对于扩展帧，ID_STD[10:0]和这 18 位共同用作发送接收帧的标识符。

MDESx：邮箱描述符字 x（x = 2..17）

地址偏移：0x80 + 0x04 * x（x = 2..17）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA_i[7:0]								DATA_i+1[7:0]							
rw								rw							

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

位/位域	名称	描述
31:24	DATA_i[7:0]	数据字节 i（i = 4*x – 8）

		参考 DATA_i+3[7:0]描述。
23:16	DATA_i+1[7:0]	数据字节 i+1（i = 4*x – 8） 参考 DATA_i+3[7:0]描述。
15:8	DATA_i+2[7:0]	数据字节 i+2（i = 4*x – 8） 参考 DATA_i+3[7:0]描述。
7:0	DATA_i+3[7:0]	数据字节 i+3（i = 4*x - 8） 一个数据帧最大包含 64 个字节数据，主要由邮箱的 DLC 值决定。 对于接收帧，该位域用于存储 CAN 总线上接收到的数据。

邮箱编号

当接收FIFO禁能时，专用的RAM空间只被邮箱占用，因此邮箱编号与邮箱描述符编号相同，邮箱描述符编号按整个邮箱描述符为单位递增，每个邮箱描述符的数据可以是8字节，16字节，32字节或者64字节。

当接收FIFO使能时（CAN FD模式处于禁能状态，因此数据是8字节长度），专用的RAM空间同时被邮箱和FIFO占用，因此统一以8字节数据的邮箱描述符为单位进行描述符计数编号，那么邮箱编号就是邮箱描述符所占用的计数编号。

CAN FD 模式下邮箱数目

当CAN FD模式使能时，由CAN_FDCTL寄存器的MDSZ[1:0]位域来配置邮箱的数目，决定512字节的RAM空间的分配。

表 42-5. 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

42.3.2. 接收 FIFO 描述符

接收FIFO描述符如*表42-6. 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 42-6. 接收 FIFO 描述符

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

FDES0	IDFMN[8:0]			SRR	IDE	RTR	DLC[3:0]	TIMESTAMP[15:0]
FDES1	保留	ID_STD[10:0]					ID_EXD[17:0]
FDES2	DATA_0[7:0]		DATA_1[7:0]		DATA_2[7:0]		DATA_3[7:0]
FDES3	DATA_4[7:0]		DATA_5[7:0]		DATA_6[7:0]		DATA_7[7:0]
0x90 - 0xDC	保留						
FDES4	标识符过滤元素 0						
…	…						
FDES107	标识符过滤元素 103						

FDES0：接收 FIFO 描述符字 0

地址偏移：0x80

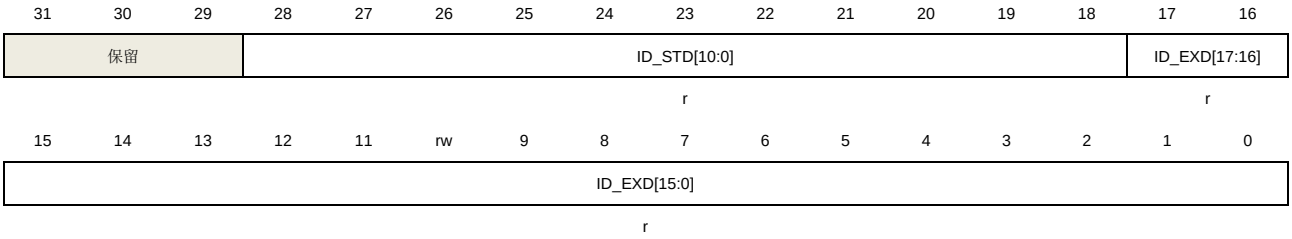
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDFMN[8:0]									SRR	IDE	RTR	DLC[3:0]			
r									r	r	r	r			

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

位/位域	名称	描述
31:23	IDFMN[8:0]	标识符过滤元素匹配序号 该位域表示在接收 FIFO 输出中的消息是与哪个标识符过滤元素相匹配。
22	SRR	替代远程请求 该位仅用于扩展帧格式。 对于接收，该位将存储 CAN 总线上接收到的该位值。
21	IDE	标识符扩展位 该位指示了该帧是标准帧还是扩展帧。 0：帧格式为标准帧 1：帧格式为扩展帧
20	RTR	远程传输请求 0：接收数据帧 1：接收远程帧
19:16	DLC[3:0]	数据字节长度代码 该位域表示接收帧的数据字节长度。 对于接收邮箱，不需要配置该位，该位域将被 CAN 总线上接收到的 DLC 域值改写。
15:0	TIMESTAMP[15:0]	时间戳 该位域是在接收帧的标识符域出现在 CAN 总线的时刻，抓取到的内部计数器的值。

FDES1：接收 FIFO 描述符字 1

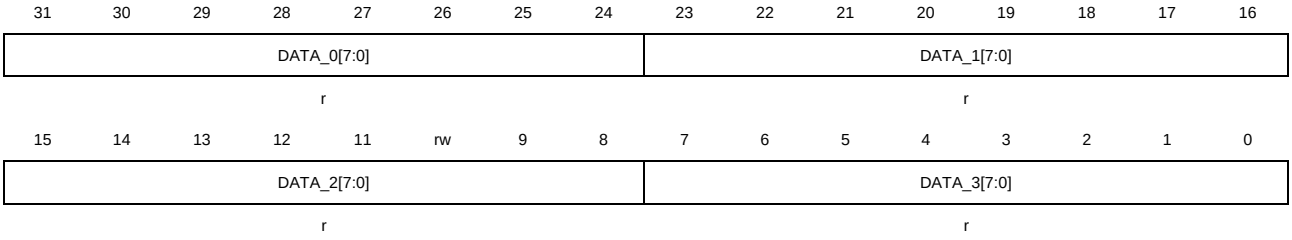
地址偏移：0x84



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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA_4[7:0]								DATA_5[7:0]							
r								r							
15	14	13	12	11	rw	9	8	7	6	5	4	3	2	1	0
DATA_6[7:0]								DATA_7[7:0]							
r								r							

位/位域	名称	描述
31:24	DATA_4[7:0]	数据字节 4 参考 DATA_7[7:0]描述。
23:16	DATA_5[7:0]	数据字节 5 参考 DATA_7[7:0]描述。
15:8	DATA_6[7:0]	数据字节 6 参考 DATA_7[7:0]描述。
7:0	DATA_7[7:0]	数据字节 7 一个数据帧最大包含 8 个字节数据，主要由邮箱的 DLC 值决定。 接收 FIFO 不支持接收 FD 帧。

FDESx：接收 FIFO 描述符字 x（x = 4..107）

地址偏移：0xE0 + 4 * (x - 4)

该描述符字给出了标识符过滤元素的3种不同格式，可以通过CAN_CTL0寄存器的FS[1:0]位域来配置。

注意：所有的标识符过滤元素只能同时使用同一种格式，不同的格式不能混合在一个标识符过滤器表中使用。

格式A模式：

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

位/位域	名称	描述
31	RTR_A	格式 A 远程帧 该位指示了是否接收匹配的远程帧到 FIFO。 0：远程帧都被拒绝，只存储数据帧 1：数据帧都被拒绝，只存储远程帧

30	IDE_A	格式 A 标识符扩展位 该位指示了是否接收匹配的扩展帧到 FIFO。 0：扩展帧都被拒绝，只存储标准帧 1：标准帧都被拒绝，只存储扩展帧
29	保留	必须保持复位值。
28:0	ID_STD_A[10:0]/ ID_EXD_A[28:0]	格式 A 标识符 该位域指示一个用于接收 FIFO 匹配过程的完整的标识符（标准格式或者扩展格式）。 如果 IDE_A 为 0，则 18 到 28 位用作标准格式标识符，其余位保留；否则，所有位用作扩展格式标识符。

格式B模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RTR_B0	IDE_B0	ID_STD_B_0[10:0]											保留		
		ID_EXD_B_0[13:0]													
rw	rw	rw													

15	14	13	12	11	rw	9	8	7	6	5	4	3	2	1	0
RTR_B1	IDE_B1	ID_STD_B_1[10:0]											保留		
		ID_EXD_B_1[13:0]													
rw	rw	rw													

位/位域	名称	描述
31	RTR_B0	格式 B 远程帧 0 该位指示了是否接收匹配的远程帧到 FIFO。 0：远程帧都被拒绝，只存储数据帧 1：数据帧都被拒绝，只存储远程帧
30	IDE_B0	格式 B 标识符扩展位 0 该位指示了是否接收匹配的扩展帧到 FIFO。 0：扩展帧都被拒绝，只存储标准帧 1：标准帧都被拒绝，只存储扩展帧
29:16	ID_STD_B_0[10:0]/ ID_EXD_B_0[13:0]	格式 B 标识符 0 该位域指示一个用于接收 FIFO 匹配过程的完整的标准格式标识符或者扩展格式标识符其中 14 位。 如果 IDE_B0 位为 0，则 19 到 29 位用作标准格式标识符，其余位保留；否则，这些位都用作扩展格式标识符其中 14 位，其与接收到的标识符的最高有效 14 位进行比较。
15	RTR_B1	格式 B 远程帧 1 参考 RTR_B0 描述。
14	IDE_B1	格式 B 标识符扩展位 1 参考 IDE_B0 描述。
13:0	ID_STD_B_1[10:0]/	格式 B 标识符 1

ID_EXD_B_1[13:0] 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。

42.3.3. 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。

42.3.4. 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IRC64MDIV，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。

42.3.5. 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42-5. 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42-7. 当本地优先级禁用时的邮箱仲裁值（32位）*和*表42-8. 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42-7. 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42-8. 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 42-7. 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 42-8. 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。

42.3.6. 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42-6. 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42-9. 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 42-9. 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42-10. 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 42-10. 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。

42.3.7. 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42-11. 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。

42.3.8. 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最长2位的显性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MDES0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42-2. 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：

$$t_{compensation} = t_{measure} + t_{offset} \quad (42\text{-}1)$$

其中：

$$t_{offset} = TDCO[4:0] \times t_{CANCLK} \quad (42\text{-}2)$$

$$t'_{offset} = t_{PBS1_FD} + t_{PTS_FD} + t_{SYNC_SEG} \quad (23\text{-}3)$$

$$t_{PBS1_FD} = (DPBS1[2:0] + 1) \times t_{q_FD} \quad (42\text{-}4)$$

$$t_{PTS_FD} = DPTS[4:0] \times t_{q_FD} \quad (42\text{-}5)$$

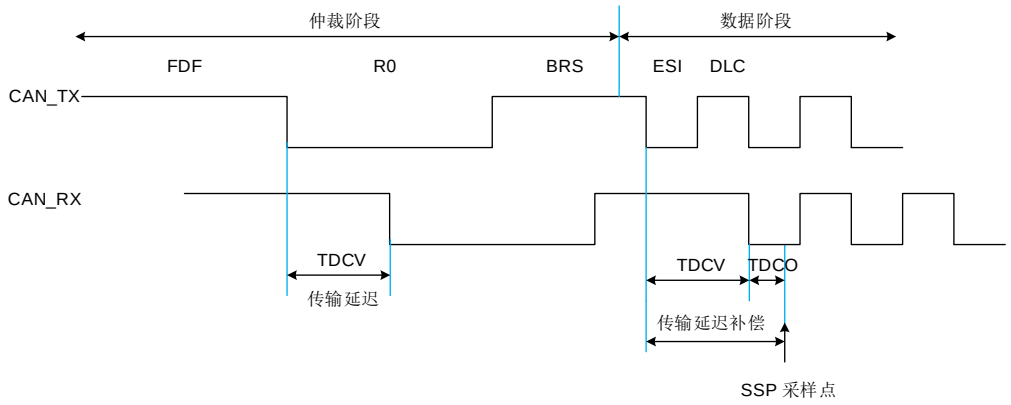
$$t_{q_FD} = (DBAUDPSC[9:0] + 1) \times t_{CANCLK} \quad (42\text{-}6)$$

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 42-2. 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。

42.3.9. 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。

42.3.10. 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（$1\times t_q$）。

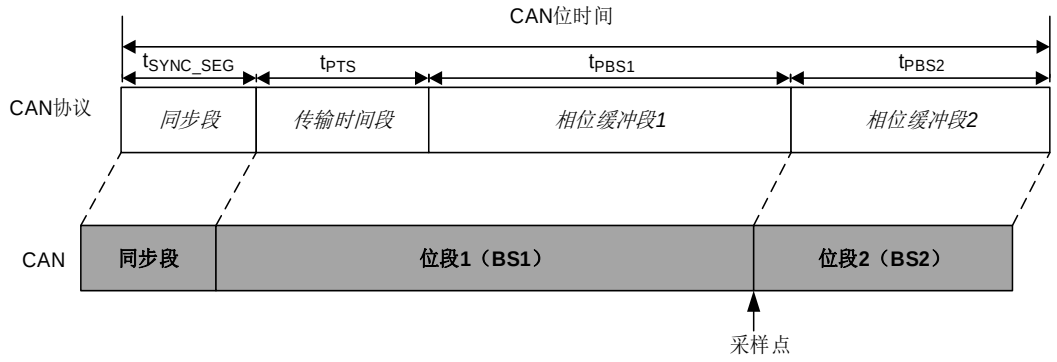
位段1（BS1）：该段包括CAN协议中的传播时间段和相位缓冲段1。该段可自动延长来补偿网络节点的频率不同引起的相位正漂移。

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42-3. CAN位时间*所示。

图 42-3. 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：

$$\text{BaudRate} = \frac{1}{\text{CAN Bit Time}} \quad (42\text{-}7)$$

$$CAN\ Bit\ Time = t_{SYNC_SEG} + t_{PTS} + t_{PBS1} + t_{PBS2} \quad (42\text{-}8)$$

其中：

$$t_{SYNC_SEG} = 1 \times t_q \quad (42\text{-}9)$$

$$t_{PTS} = (N_{PTS} + 1) \times t_q \ \text{ or } \ t_{PTS} = N_{DPTS} \times t_q \quad (42\text{-}10)$$

$$t_{PBS1} = (N_{PBS1} + 1) \times t_q \quad (42\text{-}11)$$

$$t_{PBS2} = (N_{PBS2} + 1) \times t_q \quad (42\text{-}12)$$

$$t_q = (N_{BAUDPSC} + 1) \times t_{CANCLK} \quad (42\text{-}13)$$

在公式中，对于正常波特率：

N_{PTS}，N_{PBS1}，N_{PBS2}和$N_{BAUDPSC}$分别由CAN_BT寄存器的PTS[5:0]，PBS1[4:0]，PBS2[4:0]和BAUDPSC[9:0]位域来配置。

对于数据波特率：

N_{DPTS}，N_{PBS1}，N_{PBS2}和$N_{BAUDPSC}$分别由CAN_FDBT寄存器的DPTS[4:0]，DPBS1[2:0]，DPBS2[2:0]和DBAUDPSC[9:0]位域来配置。

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。

42.3.11. 中断

CAN中断事件与标志如*表42-11. 中断事件*所示。

表 42-11. 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			

42.4. 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]位域来配置邮箱数据量大小。
  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PNMOD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。

42.5. CAN 寄存器

CAN0基地址：0x4001 A000

CAN1基地址：0x4001 B000

CAN2基地址：0x4001 C000

42.5.1. 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

42.5.2. 控制寄存器 1（CAN_CTL1）

地址偏移：0x04
复位值：0x0000 0000

该寄存器中位 12，7，5，4，3 只可在暂停模式下配置，它们在其他模式下被硬件锁定。

该寄存器中所有位都不会被 CAN_CTL0 寄存器中的软件复位 SWRST 位复位。

该寄存器只能按字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOIE	ERRSIE	保留	LSCMOD	TWERRIE	RWERRIE	保留		BSPMOD	ABORDIS	TSYNC	MTO	MMOD	保留		
rw	rw		rw	rw	rw			rw	rw	rw	rw	rw			

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	BOIE	离线中断使能 0：禁能离线中断 1：使能离线中断
14	ERRSIE	错误汇总中断使能 0：禁能错误汇总中断 1：使能错误汇总中断
13	保留	必须保持复位值。
12	LSCMOD	回环静默模式 0：禁能回环静默模式 1：使能回环静默模式 **注意：**在该模式下，不能置位CAN_CTL0寄存器的SRDIS位，和CAN_FDCTL寄存器的TDCEN位。
11	TWERRIE	发送错误警告中断使能 只有当CAN_CTL0寄存器的WERREN位置位时才可写该位。当CAN_CTL0寄存器的WERREN位为0时，该位读为0。 0：禁能发送错误警告中断 1：使能发送错误警告中断
10	RWERRIE	接收错误警告中断使能 只有当CAN_CTL0寄存器的WERREN位置位时才可写该位。当CAN_CTL0寄存器的WERREN位为0时，该位读为0。 0：禁能接收错误警告中断 1：使能接收错误警告中断
9:8	保留	必须保持复位值。
7	BSPMOD	位采样模式 0：接收位使用1个采样点 1：接收位使用3个采样点
6	ABORDIS	自动离线恢复不使能 0：使能自动离线恢复 1：不使能自动离线恢复
5	TSYNC	时间同步使能 0：禁能时间同步 1：使能时间同步

4	MTO	邮箱发送顺序 0：高优先级的邮箱先发送 1：低邮箱编号的邮箱先发送
3	MMOD	监听模式 0：禁能监听模式 1：使能监听模式
2:0	保留	必须保持复位值。

42.5.3. 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。

42.5.4. 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。

42.5.5. 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

42.5.6. 错误寄存器 1（CAN_ERR1）

地址偏移：0x20
复位值：0x000X 000X

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BRFERR	BDFERR	保留	CRCFERR	FMFERR	STFFERR	保留				ERROVR	ERRFSF	BORF	SYN	TWERRIF	RWERRIF
rc	rc		rc	rc	rc					rc_w1	rc_w1	rc_w1	r	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRERR	BDERR	ACKERR	CRCERR	FMERR	STFERR	TWERRF	RWERRF	IDLEF	TS	ERRSI[1:0]		RS	BOF	ERRSF	保留
rc	rc	rc	rc	rc	rc	r	r	r	r	r		r	rc_w1	rc_w1	

位/位域	名称	描述
31	BRFERR	FD帧BRS位为隐性位时数据阶段的位隐性错误 0：没有发生错误 1：至少有一个位发送为隐性位，接收为显性位
30	BDFERR	FD帧BRS位为隐性位时数据阶段的位显性错误 0：没有发生错误 1：至少有一个位发送为显性位，接收为隐性位
29	保留	必须保持复位值。
28	CRCFERR	FD帧BRS位为隐性位时数据阶段的CRC错误 0：没有发生错误 1：发生了一个 CRC 错误
27	FMFERR	FD帧BRS位为隐性位时数据阶段的格式错误 0：没有发生错误 1：发生了一个格式错误
26	STFFERR	FD帧BRS位为隐性位时数据阶段的填充错误 0：没有发生错误 1：发生了一个填充错误
25:22	保留	必须保持复位值。
21	ERROVR	错误溢出 该位表示在某一个错误标志位已经置位的情况下，又检测到了一个错误。 0：没有发生错误溢出 1：发生了错误溢出
20	ERRFSF	FD帧BRS位为隐性位时数据阶段的错误汇总标志 该位是下列位的逻辑或： CAN_ERR1[31]：位隐性错误 CAN_ERR1[30]：位显性错误 CAN_ERR1[28]：CRC错误 CAN_ERR1[27]：格式错误 CAN_ERR1[26]：填充错误
19	BORF	离线恢复标志 当检测到CAN总线上总线恢复序列时，该位置位，指示CAN节点可以从离线状态恢复。 0：没有事件发生 1：发生了离线恢复序列事件
18	SYN	同步标志 0：未与CAN总线同步 1：与 CAN 总线同步
17	TWERRIF	发送错误警告中断标志 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。

42.5.7. 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

42.5.8. 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*。

42.5.9. 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 42-12. 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

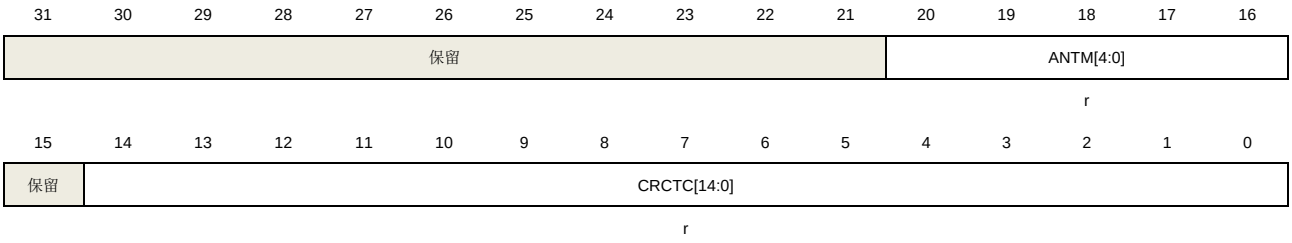
14 PREEN CAN规范中的协议异常检测使能

		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。

42.5.10. 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。

42.5.11. 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

42.5.12. 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。

42.5.13. 位时间寄存器（CAN_BT）

地址偏移：0x50
复位值：0x0100 0000

该寄存器中所有位都只可在暂停模式下配置，它们在其他模式下被硬件锁定。

该寄存器中所有位都不会被 CAN_CTL0 寄存器中的软件复位 SWRST 位复位。

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	BAUDPSC[9:0]										SJW[4:0]				

			rw								rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PTS[5:0]						PBS1[4:0]					PBS2[4:0]				
			rw					rw					rw		

位/位域	名称	描述
31	保留	必须保持复位值。
30:21	BAUDPSC[9:0]	波特率分频系数 CAN波特率分频系数= BAUDPSC[9:0] + 1。
20:16	SJW[4:0]	再同步补偿宽度 再同步补偿占用的时间单元数量 = SJW[4:0] + 1
15:10	PTS[5:0]	传播时间段 传播时间段占用的时间单元数量 = PTS[5:0] + 1
9:5	PBS1[4:0]	相位缓冲段1 相位缓冲段 1 占用的时间单元数量 = PBS1[4:0] + 1
4:0	PBS2[4:0]	相位缓冲段2 相位缓冲段 2 占用的时间单元数量 = PBS2[4:0] + 1

42.5.14. 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

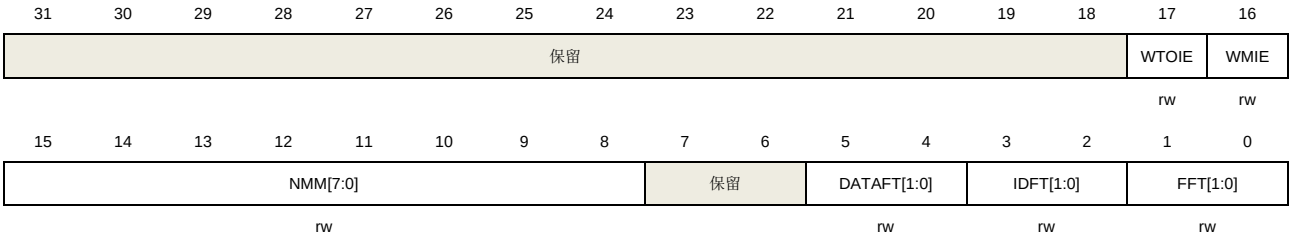
位/位域	名称	描述
31:0	FMFDx	FIFO / 邮箱过滤数据 当用作邮箱过滤时，参考CAN_RMPUBF寄存器的MFDx位。 当用作 Rx FIFO 过滤时，参考 CAN_RFIFOPUBF 寄存器的 FFDx 位。 0：不关心该位 1：参与比较

42.5.15. 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。



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

42.5.16. 虚拟联网模式超时寄存器（CAN_PN_TO）

地址偏移：0xB04

复位值：0x0000 0000

该寄存器中所有位都只可在暂停模式下配置，它们在其他模式下被硬件锁定。

该寄存器只能按字（32 位）访问。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	WTO[15:0]	超时唤醒 该超时值按照CAN位时间的64倍进行计数。默认关闭超时唤醒。

42.5.17. 虚拟联网模式状态寄存器（CAN_PN_STAT）

地址偏移：0xB08

复位值：0x0000 0080

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留														WTOS	WMS
														rc_w1	rc_w1

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

r r

位/位域	名称	描述
31:18	保留	必须保持复位值。
17	WTOS	超时唤醒标志状态 0：没有发生超时唤醒事件 1：发生了超时唤醒事件
16	WMS	匹配唤醒标志状态 0：没有发生匹配唤醒事件 1：发生了匹配唤醒事件
15:8	MMCNT[7:0]	在虚拟联网模式下的帧匹配计数 该位域指示了在虚拟联网模式下的匹配的帧的计数值。该位域在进入虚拟联网模式时由CAN模块复位，并且不受软件复位的影响。
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。

42.5.18. 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。

42.5.19. 虚拟联网模式期望 DLC 寄存器（CAN_PN_EDLC）

地址偏移：0xB10
复位值：0x0000 0008

该寄存器中所有位都只可在暂停模式下配置，它们在其他模式下被硬件锁定。

该寄存器只能按字（32 位）访问。

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

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

位/位域	名称	描述
31:20	保留	必须保持复位值。
19:16	DLCELT[3:0]	在虚拟联网模式下的期望DLC下限值
15:4	保留	必须保持复位值。
3:0	DLCEHT[3:0]	在虚拟联网模式下的期望DLC上限值

42.5.20. 虚拟联网模式期望数据低字 0 寄存器（CAN_PN_EDL0）

地址偏移：0xB14
复位值：0x0000 0000

该寄存器中所有位都只可在暂停模式下配置，它们在其他模式下被硬件锁定。

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DB0ELT[7:0]								DB1ELT[7:0]							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB2ELT[7:0]								DB3ELT[7:0]							
rw								rw							

位/位域	名称	描述
31:24	DB0ELT[7:0]	在虚拟联网模式下的期望数据字节 0 下限值 参考 DB3ELT[7:0]描述。
23:16	DB1ELT[7:0]	在虚拟联网模式下的期望数据字节 1 下限值 参考 DB3ELT[7:0]描述。
15:8	DB2ELT[7:0]	在虚拟联网模式下的期望数据字节 2 下限值 参考 DB3ELT[7:0]描述。
7:0	DB3ELT[7:0]	在虚拟联网模式下的期望数据字节 3 下限值 当 CAN_PN_CTL0 寄存器的 DATAFT[1:0]位域为 0 / 1 / 2 时，该位域用作期望的 DATA，当 DATAFT[1:0]位域为 3 时，该位域用作期望的 DATA 下限值。

42.5.21. 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。

42.5.22. 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。



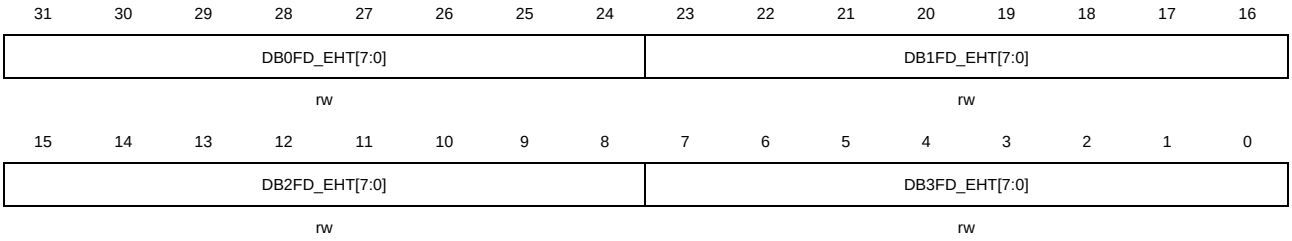
位/位域	名称	描述
31	保留	必须保持复位值。
30	IDEFD	在虚拟联网模式下的 IDE 过滤数据 0：不关心该位 1：参与比较
29	RTRFD	在虚拟联网模式下的 RTR 过滤数据 0：不关心该位 1：参与比较
28:0	IDFD_EHT[28:0]	在虚拟联网模式下的ID过滤数据 / 期望的ID上限值 **ID过滤数据**（当CAN_PN_CTL0寄存器的IDFT[1:0]位域为0时）： 0：不关心该位 1：参与比较 **ID期望上限值**（当IDFT[1:0]位域为3时）。 **保留**（当IDFT[1:0]位域为1或者2时）。 对于扩展格式帧，使用所有29位。 对于标准格式帧，使用位18到28。

42.5.23. 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。



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）。

42.5.24. 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							

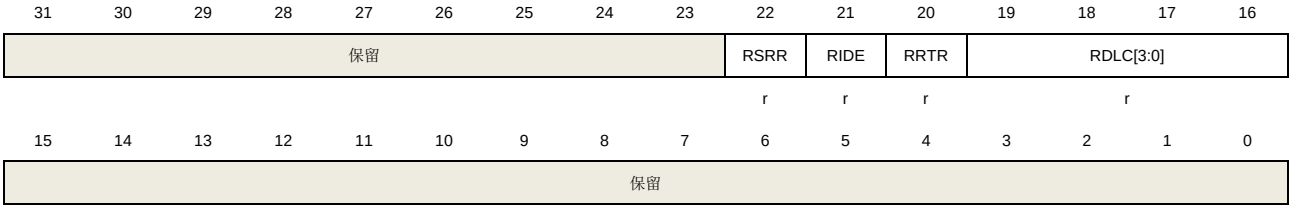
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB6FD_HTF[7:0]								DB7FD_HTF[7:0]							
rw								rw							

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。

42.5.25. 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。



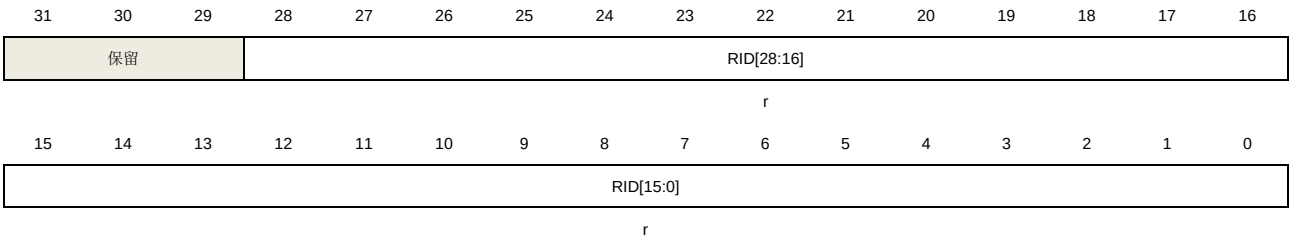
位/位域	名称	描述
31:23	保留	必须保持复位值。
22	RSRR	接收到的SRR位
21	RIDE	接收到的IDE位 0：帧为标准格式 1：帧为扩展格式
20	RRTR	接收到的RTR位 0：帧为数据帧 1：帧为远程帧
19:16	RDLC[3:0]	接收到的DLC域 该位域指示了有效的数据字节长度。
15:0	保留	必须保持复位值。

42.5.26. 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。

42.5.27. 虚拟联网模式接收唤醒邮箱 x 数据 0 寄存器（CAN_PN_RWMxD0）（x=0..3）

地址偏移：0xB48 + 16 * x

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDB0[7:0]								RDB1[7:0]							
r								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDB2[7:0]								RDB3[7:0]							
r								r							

位/位域	名称	描述
31:24	RDB0[7:0]	接收到的数据字节0
23:16	RDB1[7:0]	接收到的数据字节 1
15:8	RDB2[7:0]	接收到的数据字节 2
7:0	RDB3[7:0]	接收到的数据字节 3

42.5.28. 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

42.5.29. FD 控制寄存器（CAN_FDCTL）

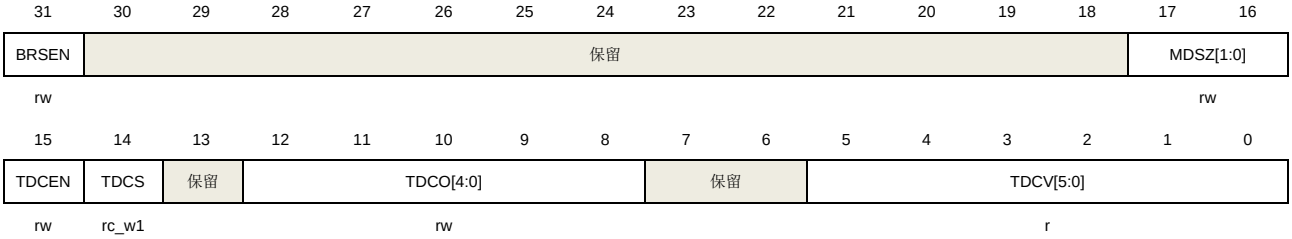
地址偏移：0xC00

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。



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。

42.5.30. 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频系数= DBAUDPSC[9:0] + 1。
19	保留	必须保持复位值。
18:16	DSJW[2:0]	数据位时间的再同步补偿宽度 再同步补偿占用的时间单元数量 = DSJW[2:0] + 1
15	保留	必须保持复位值。
14:10	DPTS[4:0]	数据位时间的传播时间段 传播时间段占用的时间单元数量 = DPTS[4:0]
9:8	保留	必须保持复位值。
7:5	DPBS1[2:0]	数据位时间的相位缓冲段1 相位缓冲段 1 占用的时间单元数量 = DPBS1[2:0] + 1
4:3	保留	必须保持复位值。
2:0	DPBS2[2:0]	数据位时间的相位缓冲段 2 相位缓冲段 2 占用的时间单元数量 = DPBS2[2:0] + 1

42.5.31. 常规帧和 FD 帧 CRC 寄存器（CAN_CRCCFD）

地址偏移：0xC08
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

保留	ANTM[4:0]	保留	CRCTCI[20:16]
	r		r

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

位/位域	名称	描述
31:29	保留	必须保持复位值。
28:24	ANTM[4:0]	发送 CRCTCI[20:0]值的相关联的邮箱的编号 该位域包含发送常规帧或者 FD 帧时，CRC 值为 CRCTCI[20:0]的邮箱的编号。
23:21	保留	必须保持复位值。
20:0	CRCTCI[20:0]	发送的常规帧 / FD帧的CRC计算值 对于CRC_15，使用位0到位14，其他位为0，并且该位域值与CAN_CRCC寄存器中的CRCTC[14:0]值相同。 对于CRC_17，使用位0到位16，其他位为0。 对于CRC_21，使用所有的21位。

43. 以太网（ENET）（适用于 GD32H737_757_759）

43.1. 简介

本章描述以太网外设模块。该以太网模块包含两个10/100Mbps以太网MAC（媒体访问控制器），采用DMA优化数据帧的发送与接收性能，支持MII（媒体独立接口）与RMII（简化的媒体独立接口）两种与物理层(PHY)通讯的标准接口，实现以太网数据帧的发送与接收。以太网模块遵守IEEE 802.3-2002标准和IEEE 1588-2008标准。

43.2. 主要特性

MAC特性

- 支持10Mbps或100Mbps数据传输速率；
- 支持MII和RMII接口；
- 支持调试用回环模式；
- 支持符合CSMA/CD协议的半双工背压通讯；
- 支持符合IEEE 802.3x的流控通讯。在当前帧发送完毕后，根据接收的暂停帧中暂停时间延迟发送。在全双工/半双工模式下，MAC根据RxFIFO的填充程度自动发送暂停帧/背压信号；
- 支持符合IEEE 802.3x的全双工流控通讯，当输入流控信号失效时，自动发送零时间片暂停帧。支持符合IEEE 802.3x的半双工流控通讯，支持根据RxFIFO的填充程度（直通模式）自动发送背压信号；
- 可选择在发送操作时自动生成校验/填充位；
- 可选择在接收操作时自动去除校验/填充位；
- 帧长度可配置；
- 帧间隙可配置；
- 支持多种模式的接收过滤；
- 支持检测接收帧的IEEE 802.1Q VLAN标签；
- 支持强制网络统计标准（RFC2819/RFC2665）；
- 支持两种唤醒帧检测：LAN远程唤醒帧和AMD的Magic Packet™帧；
- 支持校验和检查（IPv4报头，IPv4或IPv6数据格式封装的TCP、UDP或ICMP）；
- 支持IEEE 1588-2008标准定义的以太网帧时间戳，并将其按64位记录于帧状态中；
- 相互独立的两个FIFO分别用于发送与接收；
- 在延迟冲突、过度冲突、过度顺延和下溢情况下丢弃帧；
- 帧传输时，支持存储转发模式下的硬件校验和的计算和插入。

DMA特性

- 支持环结构或链结构两种形式的描述符列表；
- 每个描述符可以传输最高为8192字节的数据；
- 中断可配置，适用于多种工作状态；
- 支持轮询或固定优先级两种方式仲裁DMA发送和接收控制器的请求。

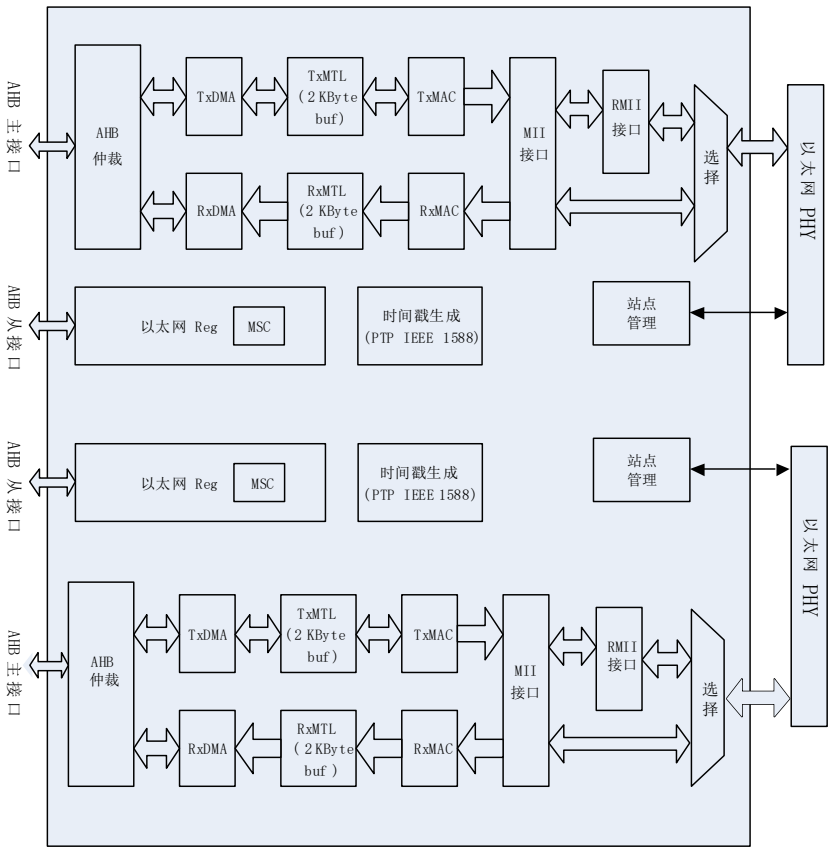
PTP特性

- 支持符合IEEE1588的时间同步功能；
- 支持粗/精调两种校正方法；
- 输出秒脉冲；
- 达到预设目标时间时触发中断。

43.2.1. 模块框图

以太网模块由两个MAC（介质访问控制器）模块、两个MII/RMII模块和两个以描述符形式控制的DMA模块组成。在使用以太网模块时，用户应确保配置的时钟频率不小于25MHz。

图 43-1. 以太网模块框图



MAC模块通过MII或RMII与片外PHY连接。通过对SYSCFG_PMCFG寄存器的相关位进行设置，可以选择使用哪种接口。SMI（站点管理接口）用于配置和管理外部PHY。

发送数据模块包括：

- TxDMA控制器，用于从存储器中读取描述符和数据，以及将状态写入存储器；
- TxMTL，用于对发送数据的控制，管理和存储。TxMTL内含TxFIFO，用于缓存待MAC发送的数据；
- MAC发送控制寄存器组，用于管理和控制数据帧的发送。

接收数据模块包括：

- RxDMA控制器，用于从存储器中读取描述符，以及将数据与状态写入存储器；

- RxMTL，用于对接收数据的控制，管理和存储。RxMTL实现了RxFIFO，用于存储待转发到系统存储的帧数据；
- MAC接收控制寄存器组，用于管理数据帧的接收和标示接收状态。MAC内含接收过滤器，采用多种过滤机制，滤除特定的以太网帧。

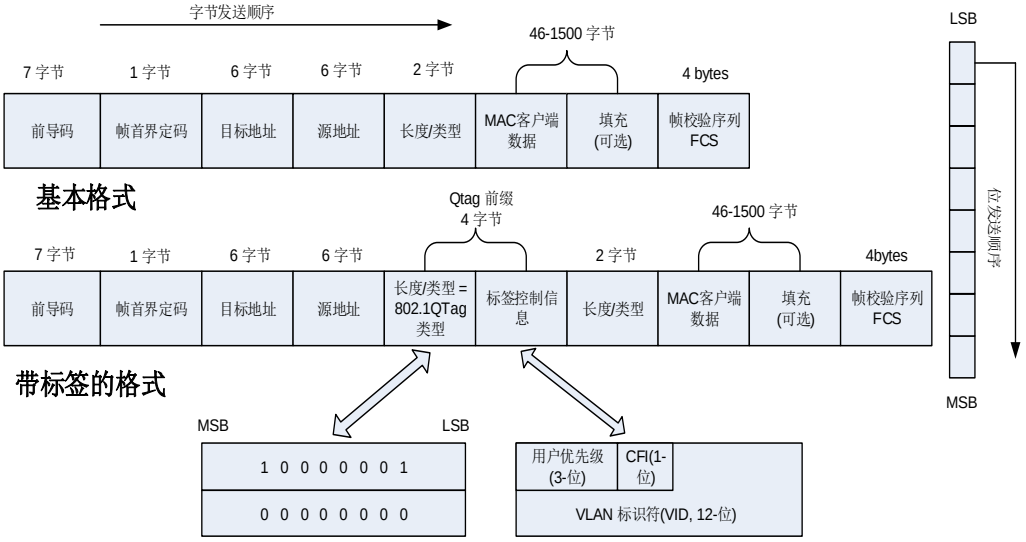
43.2.2. MAC 802.3 以太网数据包描述

MAC的数据通信可使用两种帧格式：

- 基本帧格式；
- 带标签的帧格式。

*图43-2. MAC/带标签的MAC帧格式*描述了帧结构（基本的和带标签的）：

图 43-2. MAC/带标签的 MAC 帧格式



注意：除了帧校验序列，以太网控制器发送每个字节时都按照低位先出的次序进行传输。

CRC计算包括帧数据的所有字节除去前导码和帧首界定码域。以太网帧的32位CRC生成多项式为0x04C11DB7，且此多项式用于以太网模块中所有的32位CRC计算，如下式所示：

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

43.2.3. 以太网信号描述

*表43-1. 以太网信号（MII）*和*表43-2. 以太网信号（RMII）*列出了MAC模块所用引脚在MII/RMII模式下默认及重映射的功能和具体配置。

表 43-1. 以太网信号（MII）

信号	引脚1	引脚2	引脚3	引脚模式
ETH0_MDC	PC1			推挽复用输出(AF11)
ETH0_MII_TXD2	PC2			推挽复用输出(AF11)
ETH0_MII_TX_CLK	PC3			推挽复用输出(AF11)
ETH0_MII_CRS	PA0	PH2		推挽复用输出(AF11)

信号	引脚1	引脚2	引脚3	引脚模式
ETH0_MII_RX_CLK	PA1			推挽复用输出(AF11)
ETH0_MDIO	PA2			推挽复用输出(AF11)
ETH0_MII_COL	PA3	PH3		推挽复用输出(AF11)
ETH0_MII_RX_DV	PA7			推挽复用输出(AF11)
ETH0_MII_RXD0	PC4			推挽复用输出(AF11)
ETH0_MII_RXD1	PC5			推挽复用输出(AF11)
ETH0_MII_RXD2	PB0	PH6		推挽复用输出(AF11)
ETH0_MII_RXD3	PB1	PH7		推挽复用输出(AF11)
ETH0_PPS_OUT	PB5	PG8		推挽复用输出(AF11)
ETH0_MII_TXD3	PB8	PE2		推挽复用输出(AF11)
ETH0_MII_RX_ER	PB10			推挽复用输出(AF11)
ETH0_MII_TX_EN	PB11	PG11		推挽复用输出(AF11)
ETH0_MII_TXD0	PB12	PG13		推挽复用输出(AF11)
ETH0_MII_TXD1	PB13	PG12	PG14	推挽复用输出(AF11)
ETH1_MDC	PG6			推挽复用输出(AF6)
ETH1_MII_TXD2	PG12			推挽复用输出(AF6)
ETH1_MII_TX_CLK	PG9			推挽复用输出(AF6)
ETH1_MII_CRS	PH15			推挽复用输出(AF6)
ETH1_MII_RX_CLK	PH12			推挽复用输出(AF6)
ETH1_MDIO	PH14			推挽复用输出(AF6)
ETH1_MII_COL	PH13			推挽复用输出(AF6)
ETH1_MII_RX_DV	PH11			推挽复用输出(AF6)
ETH1_MII_RXD0	PH8			推挽复用输出(AF6)
ETH1_MII_RXD1	PH9			推挽复用输出(AF6)
ETH1_MII_RXD2	PH6			推挽复用输出(AF6)
ETH1_MII_RXD3	PH7			推挽复用输出(AF6)
ETH1_PPS_OUT	PG8			推挽复用输出(AF6)
ETH1_MII_TXD3	PG15			推挽复用输出(AF6)
ETH1_MII_RX_ER	PH10			推挽复用输出(AF6)
ETH1_MII_TX_EN	PG11			推挽复用输出(AF6)
ETH1_MII_TXD0	PG13			推挽复用输出(AF6)
ETH1_MII_TXD1	PG14			推挽复用输出(AF6)

表 43-2. 以太网信号（RMII）

信号	引脚1	引脚2	引脚3	引脚模式
ETH0_MDC	PC1			推挽复用输出(AF11)
ETH0_RMII_REF_CLK	PA1			推挽复用输出(AF11)
ETH0_MDIO	PA2			推挽复用输出(AF11)
ETH0_RMII_CRS_DV	PA7			推挽复用输出(AF11)
ETH0_RMII_RXD0	PC4			推挽复用输出(AF11)
ETH0_RMII_RXD1	PC5			推挽复用输出(AF11)
ETH0_PPS_OUT	PB5			推挽复用输出(AF11)

信号	引脚1	引脚2	引脚3	引脚模式
ETH0_RMII_TX_EN	PB11	PG11		推挽复用输出(AF11)
ETH0_RMII_TXD0	PB12	PG13		推挽复用输出(AF11)
ETH0_RMII_TXD1	PB13	PG12	PG14	推挽复用输出(AF11)
ETH1_MDC	PG6			推挽复用输出(AF6)
ETH1_RMII_REF_CLK	PH12			推挽复用输出(AF6)
ETH1_MDIO	PH14			推挽复用输出(AF6)
ETH1_RMII_CRS_DV	PH11			推挽复用输出(AF6)
ETH1_RMII_RXD0	PH8			推挽复用输出(AF6)
ETH1_RMII_RXD1	PH9			推挽复用输出(AF6)
ETH1_PPS_OUT	PG8			推挽复用输出(AF6)
ETH1_RMII_TX_EN	PG11			推挽复用输出(AF6)
ETH1_RMII_TXD0	PG13			推挽复用输出(AF6)
ETH1_RMII_TXD1	PG14	PG15		推挽复用输出(AF6)

43.3. 功能描述

43.3.1. 接口配置

以太网模块通过MII/RMII接口与片外PHY连接，传送与接收以太网包。MII或RMII模式由软件选择并通过SMI接口对PHY进行管理。

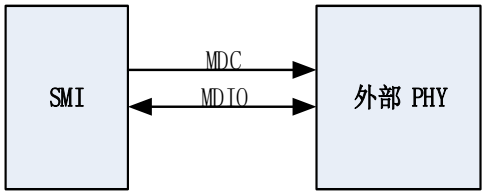
SMI：站点管理接口

SMI用于访问和设置PHY的配置。

站点管理接口（SMI）通过MDC时钟线与MDIO数据线与外部PHY通讯，可以通过其访问任意PHY的任意寄存器。SMI接口支持的最大PHY数量为32，但在同一时刻只能访问一个PHY的一个寄存器。MDC时钟线和MDIO数据线具体作用如下：

- MDC：最高频率为2.5MHz的时钟信号，在空闲状态下该引脚保持为低电平状态。在传输数据时该信号的高电平和低电平的最短保持时间为160ns，信号的最小周期为400ns；
- MDIO：用于与PHY之间的数据传输，与MDC时钟线配合，接收/发送数据。

图 43-3. 站点管理接口信号



写操作

应用程序需将要传输的数据写入ENET_MAC_PHY_DATA寄存器中，并对

ENET_MAC_PHY_CTL寄存器相关位进行操作：

1. 设置PHY设备地址和将要操作的PHY寄存器地址，并将PW位置为1，使能写模式；
2. 将PB位置1开始传输。在传输过程中PB位一直为高，直到传输完成，硬件将会自动清除PB位。

应用程序可以通过PB位判断传输是否完成。在PB位置1期间，由于操作正在运行，因此不能修改PHY控制寄存器和PHY数据寄存器的内容。在将PB位置位之前，应用程序必须确保该位读出为'0'。

读操作

应用程序对ENET_MAC_PHY_CTL寄存器相关位进行操作：

1. 设置PHY设备地址和将要操作的PHY寄存器地址，并将PW位置为0，使能读模式；
2. 将PB位置1开始数据接收。在接收过程中PB位一直为高，直到接收完成，硬件将会自动清除PB位。

应用程序可以通过PB位判断传输是否完成。在PB位置1期间，由于操作正在运行，因此不能修改PHY控制寄存器和PHY数据寄存器的内容。在将PB位置位之前，应用程序必须确保该位读出为'0'。

注意：由于PHY寄存器地址16-31的寄存器内容由各厂商自由定义，所以在访问不同PHY设备的这部分寄存器时，需要根据厂商手册对应用程序进行不同的设置。固件库当前支持的PHY设备详情请参考固件库相关手册说明。

时钟配置

SMI接口的时钟源由AHB时钟分频得到。为了保证MDC时钟频率不超过2.5MHz，需根据AHB时钟频率对PHY控制寄存器中相关位进行设置，选择合适的分频系数。*表43-3. 时钟范围*列出了对应AHB时钟范围的分频系数的选择。

表 43-3. 时钟范围

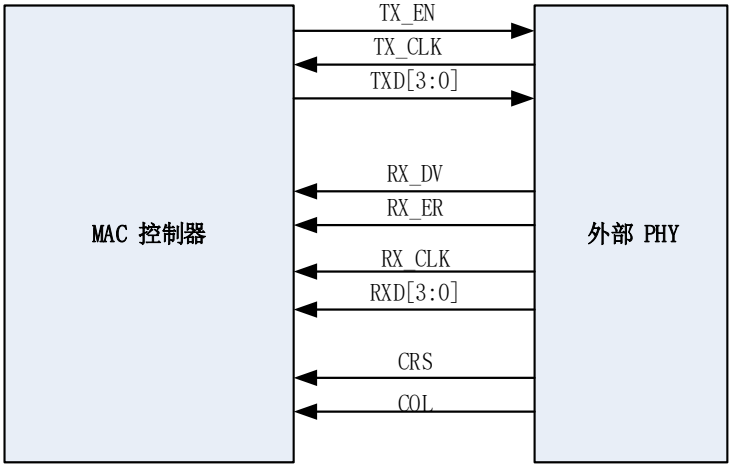
AHB时钟	MDC 时钟	ENET_MAC_PHY_CTL 位 CLR[2:0]
300~350MHz	AHB clock/142	0x6
250~300MHz	AHB clock/124	0x5
150~250MHz	AHB clock/102	0x4
35~60MHz	AHB clock/26	0x3
20~35MHz	AHB clock/16	0x2
100~150MHz	AHB clock/62	0x1
60~100MHz	AHB clock/42	0x0

MII/RMII 的选择

当使能以太网控制器时钟前或以太网控制器处于复位状态时，应用程序可通过配置SYSCFG_PMCFG寄存器的ENET0_PHY_SEL/ENET1_PHY_SEL位来选择使用MII或者RMII模式，默认为MII模式。

MII：媒体独立接口

图 43-4. 媒体独立接口（MII）信号线



- **MII_TX_CLK**：发送数据使用的时钟信号，对于10Mbit/s的数据传输，此时钟为2.5MHz，对于100M bit/s的数据传输，此时钟为25MHz。
- **MII_RX_CLK**：接收数据使用的时钟信号，对于10Mbit/s的数据传输，此时钟为2.5MHz，对于100M bit/s的数据传输，此时钟为25MHz。
- **MII_TX_EN**：发送使能信号，，当数据前导码的起始位出现时，该信号必须有效，并且需要在传输完毕前保持有效。
- **MII_TXD[3:0]**：发送数据线，每次传输4位数据，数据在MII_TX_EN信号有效时有效。当MII_TX_EN信号无效时，PHY将忽略传输的数据。
- **MII_CRS**：载波侦听信号，仅工作在半双工模式下，由PHY控制。该信号不需要与MII_TX_CLK和MII_RX_CLK保持同步。当它处于有效状态时，意味着发送或接收介质不处于空闲状态。MII_CRS信号一直保持有效，直到发送和接收介质都处于空闲状态。
- **MII_COL**：冲突检测信号，仅工作在半双工模式下，由PHY控制。该信号不需要与MII_TX_CLK和MII_RX_CLK保持同步。当检测到介质发生冲突时，此信号有效，并且在整个冲突的持续时间内，保持此信号有效。
- **MII_RXD[3:0]**：接收数据线，每次接收4位数据，数据在MII_RX_DV信号有效时有效。根据MII_RX_DV和MII_RX_ER信号的状态，MII_RXD[3:0]数据值可被用来传达一些特定信息（请参考*表43-4. 接收接口信号编码*）。
- **MII_RX_DV**：接收数据使能信号，由PHY控制，当PHY准备好数据供MAC接收时，该信号有效。当数据帧的第一个4位出现时，该信号必须有效，并且需要在传输完毕前保持有效。在传送最后4位数据后的第一个时钟之前，此信号必须变为无效状态。为确保正确地接收帧，MII_RX_DV信号应该在SFD字段出现之前有效。
- **MII_RX_ER**：接收出错信号，为了表明MAC在接收过程中检测到错误，MII_RX_ER信号必须在一个或多个时钟周期（MII_RX_CLK）内保持有效。具体错误原因需结合MII_RX_DV的状态及MII_RXD[3:0]的数据值，详见*表43-4. 接收接口信号编码*。

表 43-4. 接收接口信号编码

信号	正常的帧间隔		正常的数据接收	载波错误指示	数据接收出错
MII_RX_ER	0	1	0	1	1
MII_RX_DV	0	0	1	0	1

MII_RXD[3:0]	0000 to 1111	0000	0000 to 1111	1110	0000 to 1111

MII 时钟源

用户需要给外部PHY提供一个外部的25MHz时钟来产生TX_CLK和RX_CLK时钟信号。该时钟不需要与MAC时钟相同。可以使用外部的25MHz晶振或者微控制器的时钟输出引脚CK_OUT0提供这一时钟。当时钟来源为CK_OUT0引脚时需配置合适的PLL，保证CK_OUT0引脚输出的时钟为25MHz。

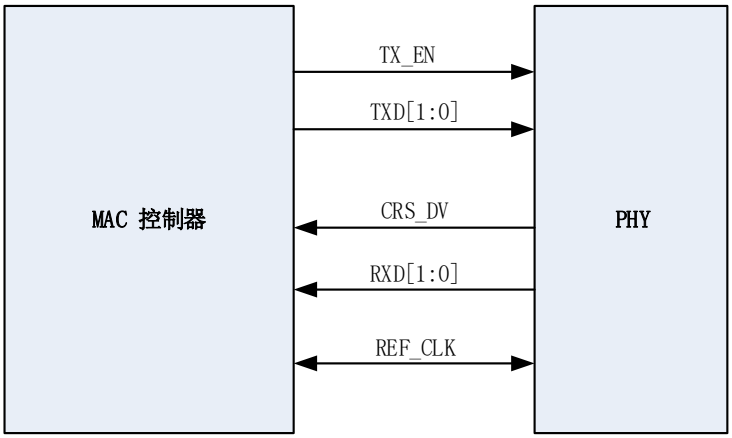
RMII：精简媒体独立接口

精简媒体独立接口（RMII）规范减少了以太网通信所需要的引脚数。根据IEEE 802.3标准，MII接口需要16个引脚用于数据和控制信号，而RMII标准则将引脚数减少到了7个。

RMII特性：

- 只有一个时钟信号，且该时钟信号需要提高到50MHz；
- MAC和外部的以太网PHY需要使用同样的时钟源；
- 使用2位宽度的数据收发。

图 43-5. 精简媒体独立接口（RMII）信号线



RMII 时钟源

通过将相同的时钟源接到MAC和以太网PHY的REF_CLK引脚保证两者时钟源的同步。可以通过外部的50MHz信号或者微控制器的CK_OUT0引脚提供这一时钟。当时钟来源CK_OUT0引脚时需配置合适的PLL，保证CK_OUT0引脚输出的时钟为50MHz。

MII/RMII 位传输顺序

不论选择的是MII还是RMII接口，发送接收的次序都是低位先出。

MII和RMII之间的区别主要是数据位数和发送次数的不同。MII上是先发送/接收低4位数据，再发送/接收高4位。在RMII上则是先发送/接收最低2位数据，再次低2位数据，次高2位数据，和最高2位数据。

例如：一个字节数据为10011101b（从左到右顺序：高位到低位）

使用MII发送需2个时钟周期：1101 -> 1001（从左到右顺序：高位到低位，其中1101对应

MII_T/RXD[3] - MII_T/RXD[0]）

使用RMII发送需4个时钟周期：01 -> 11 -> 01 -> 10（从左到右顺序：高位到低位，其中01对应RMII_T/RXD[1] - RMII_T/RXD[0]）

43.3.2. MAC 功能简介

MAC 模块可以在两种模式（半双工模式和全双工模式）下工作。半双工模式下，通过 CSMA/CD 算法来抢占对物理介质的访问，在同一时间只有一个传输方向的两个站点有效。全双工模式下，满足以下条件时，可同时进行收发而不发生冲突：1）物理介质支持同时进行收发操作。2）只有两个站点接入 LAN，且两个站点都配置为全双工模式。

MAC 模块能够实现以下功能：1）数据封装（发送和接收），包括检测/解码帧、帧边界定界、寻址（管理源地址和目的地址）、错误检测。2）半双工模式下的介质访问管理，包括介质分配和冲突解决。

MAC 的发送流程

所有的发送均由以太网模块中专用DMA控制器和MAC控制。在收到应用程序发送指令后，DMA将发送帧从系统存储区读出并存入深度为2K的TxFIFO中，之后根据选择的模式（直通或者存储转发模式，具体定义请查看下段）将数据取出到MAC控制器，通过MII/RMII接口发送到以太网PHY，并可以选择配置使MAC控制器自动将硬件计算的CRC值添加到数据帧的帧校验序列中。当MAC控制器收到来自TxFIFO的帧结束信号后，完成整个传输过程。传输完毕后，传输状态信息将会由MAC控制器生成并写回到DMA控制器中，应用程序可以通过DMA当前发送描述符查询发送状态。

TxFIFO取出数据到MAC控制器的操作有两种模式：

- 在直通模式下，一旦FIFO中的数据字节数等于或超过设置的阈值，或者描述符中的帧结束标志被写入，FIFO中的数据就会被送入到MAC控制器中。用户可通过ENET_DMA_CTL中的TTHC[2:0]配置阈值。
- 在存储转发模式下，只有当一个完整的帧写入FIFO之后，FIFO中的数据才会被送入到MAC控制器中。但还有一种情况，帧没有被完整写入FIFO，FIFO也会取出数据。这种情况为TxFIFO的大小小于要发送的以太网帧长度，那么在TxFIFO即将全满时，数据会被送入到MAC控制器。

特殊情况处理

在传输过程中，如果空闲的DMA发送描述符不足，或者误操作了ENET_DMA_CTL的FTF位清空了FIFO（此位置1时将清空TxFIFO中的数据并将FIFO的指针复位，清空操作完成后由硬件将此位清零），则将导致不能及时连续的发送数据，此时MAC控制器会标识数据下溢状态。对于只收到帧起始信号却没有收到帧结束信号的情况，MAC会忽略第二帧数据的帧起始，而将第二帧作为前一帧的延续。

若被发送的一帧占用两个DMA发送描述符，则第一个描述符的首段位（FSG）和末段位（LSG）应为10b，第二个描述符的应为01b。若第一个描述符与第二个描述符的FSG位都置位了，且第一个描述符的LSG位复位了，则将忽略第二个描述符的FSG位，并认为这两个描述符为只发送一个帧。

若发送MAC帧的数据域长度小于46或者带标签的MAC帧的数据域长度小于42，可以选择配置MAC控制器自动填充内容为0的数据，使帧数据域的长度符合IEEE 802.3规范的相关定义。若执行了自动填充0功能，则MAC将忽略DMA描述符DCRC位的配置，自动计算并添加CRC值到帧的帧校验序列中。

MAC 的发送管理

Jabber 定时器

为了防止出现一个站点长时间占用PHY的情况，以太网内置的Jabber定时器会在以太网帧发送超过2048字节后终止发送。默认情况下，Jabber定时器是使能的，因此当以太网帧发送超过2048字节，则MAC将只发送2048字节，并丢弃剩余的帧数据。

冲突处理机制：重发

在半双工模式下，MAC发送数据帧时可能会发生冲突。当发生冲突事件的时候如果FIFO中只有不超过96个字节的帧数据被取出到了MAC中，那么帧重发功能将被激活。重发功能激活后，MAC会中止当前的传输，然后重新从FIFO中读取数据并发送。当发生冲突事件的时候如果已有超过96个字节的数据从FIFO中取出到MAC中，那么MAC会中止当前的传输但不会激活重发功能，然后在描述符中置位LCO以通知应用程序。

清空 TxFIFO 操作

将ENET_DMA_CTL寄存器的FTF位置1将清空TxFIFO，并将FIFO数据指针复位。无论TxFIFO是否正在取出数据到MAC中，清空操作都会立刻执行。因此这也将导致MAC控制器产生数据下溢事件，并终止发送当前帧，同时返回该帧的状态信息和发送状态信息字到应用程序。并标记数据下溢位和清空位（发送描述符0的FRMF和UFE位）。在应用程序（DMA）接收到所有被清空帧的状态信息字以后，清空操作完成。清空操作完成后，ENET_DMA_CTL寄存器的FTF位将自动清'0'。当收到清空操作指令，所有从FIFO取出到MAC的数据都将被丢弃，直到收到FSG位为1的描述符。

帧间隔管理

MAC管理两个帧之间的时间间隔。两个帧之间的时间间隔称为帧间隙时间。在全双工模式下，在完成帧发送后，或者MAC进入空闲状态时，帧间隙计数器开始计数。如果在帧间隙时间未到达ENET_MAC_CFG寄存器中IGBS位所配置的值时，来了新的发送帧，则这个发送帧将被延迟发送直到达到帧间隙时间值。若这个新的发送帧在帧间隙时间之后到达，则会立即发送该帧。在半双工模式下，MAC遵循截断二进制指数退让算法，简要来说，就是在前一个发送帧发送完成之后，或者MAC进入空闲状态时，帧间隙计数器开始计数。在帧间隙时间内，可能会有3种情况会发生：

1. 如果在帧间隙时间的前2/3时间检测到载波信号，帧间隙计数器将复位并重新计数；
2. 如果在帧间隙时间的后1/3时间里检测到载波信号，帧间隙计数器不会复位，将继续计数，当帧间隙时间到达后，MAC发送新的帧；
3. 如果在整个帧间隙时间内都没有检测到载波信号，则在到达帧间隙时间后停止帧时隙计数器，并在之前有帧被延迟的情况下立即发送新的帧。

地址过滤模块

MAC过滤分为错误过滤（诸如过短帧、CRC错误以及坏帧的过滤）和地址过滤。此部分主要讨论地址过滤。

地址过滤利用静态物理地址（MAC地址）过滤和多播HASH列表过滤实现。若ENET_MAC_FRMF过滤器寄存器的FAR位为'0'（默认值），则只有通过地址过滤的帧才会被接收。该功能会根据应用程序设定的参数(帧过滤器寄存器)对单播帧或多播帧的目的与/或源地址进行过滤（通过目标地址的I/G位可判断是单播帧还是多播帧）并报告相应的地址过滤结果,所有不能通过过滤器的帧将被丢弃。

注意：若ENET_MAC_FRMF过滤器寄存器的FAR位为'1'，则所有帧都会被接收。在这种情况下，帧过滤结果仍会更新到接收描述符中，但帧过滤结果不会影响到帧是否会被过滤。

单播目标地址过滤器

通过对ENET_MAC_FRMF寄存器HUF位的设置，可以选择使用静态物理地址（HUF位为'0'）或者HASH列表（HUF位为'1'）的方式实现单播过滤。

- 静态物理地址过滤

MAC控制器支持多达4个MAC地址对单播地址进行完美过滤。在这种方式下，MAC会把接收到帧的6个字节单播地址与设好的MAC地址寄存器逐位比较，检查是否相符。对于MAC地址0寄存器始终使能，对于MAC地址1-MAC地址3寄存器分别有对应的使能位。MAC地址1-MAC地址3寄存器的每一个字节都可以通过相应MAC地址的高寄存器的屏蔽字节控制位（MB位）来设置是否与接收帧的目标地址相应字节比较。

- HASH列表过滤

这种过滤使用一种HASH机制。MAC利用64位的HASH列表对单播地址进行不完美过滤。这种过滤模式遵循以下两个过滤步骤：

1. MAC计算接收帧的目标地址的CRC值
2. 取CRC计算结果高6位作为索引检索HASH列表。如果CRC值对应的HASH列表上的相应位为'1'，则该帧能通过HASH过滤器，反之则该帧不能通过HASH过滤器。

这种类型过滤器的优点是可以仅用一个小表就覆盖任何可能的地址。缺点是过滤不完全，即有时应该丢弃的帧也会被接收。

单播源地址过滤器

使能MAC地址1-MAC地址3寄存器，并设置其对应MAC地址高寄存器的SAF位为'1'，MAC可以将MAC地址1-MAC地址3寄存器中设置的物理（MAC）地址与接收帧的源地址进行比较并过滤。MAC也支持对源地址的成组过滤。若设置帧过滤寄存器ENET_MAC_FRMF的SAFLT位为'1'，MAC会丢弃没能通过源地址过滤的帧，同时过滤结果会通过DMA接收描述符0的SAFF位反映出来。当SAFLT位为'1'的同时，目标地址过滤器也在工作，此时MAC控制器以两个滤波器结果的逻辑"与"形式判定帧是否通过。这意味着，只要帧没能通过其中一个过滤器，就会被丢弃。MAC只会把通过全部过滤器的帧转发给应用程序。

多播目标地址过滤器

将帧过滤寄存器ENET_MAC_FRMF的MFD位清零，以开启MAC多播地址过滤功能。此时根据帧过滤寄存器ENET_MAC_FRMF的HMF位的取值可以选择类似于单播目标地址过滤的两种方式进行地址过滤。

广播地址过滤器

默认情况下，MAC无条件的接收任何广播帧。但当设置帧过滤寄存器ENET_MAC_FRMF的BFRMD位为'1'时，MAC将丢弃接收到的所有广播帧。

HASH 或者完美地址过滤器

设置帧过滤器寄存器ENET_MAC_FRMF的HPFLT位为'1'，并设置相应的HUF位（对单播帧）或者HMF位（对多播帧）为'1'，则可以将过滤器配置成只要接收帧的目标地址匹配HASH过滤器或者物理地址过滤器之一，就令帧通过。

逆转过滤操作

无论是目标地址过滤还是源地址过滤，都能在过滤器输出端逆转过滤结果。即地址与过滤器匹配时，帧不通过；不匹配时帧通过。通过设置帧过滤寄存器ENET_MAC_FRMF的DAIFLT位和SAIFLT位为'1'可以实现这一功能。DAIFLT位作用于单播和多播帧的目标地址的过滤结果，SAIFLT位作用于单播和多播帧的源地址的过滤结果。

下面*表43-5. 目标地址过滤器结果列表*和*表43-6. 源地址过滤器结果列表*总结了目标地址和源地址过滤器在不同设置下的工作状态。

表 43-5. 目标地址过滤器结果列表

帧类型	PM	HPFLT	HUF	DAIFLT	HMF	MFD	BFRMD	目标地址过滤器操作
广播帧	1	-	-	-	-	-	-	通过
	0	-	-	-	-	-	0	通过
	0	-	-	-	-	-	1	不通过
单播帧	1	-	-	-	-	-	-	所有帧通过
	0	-	0	0	-	-	-	匹配完美/组过滤器时通过
	0	-	0	1	-	-	-	匹配完美/组过滤器时不通过
	0	0	1	0	-	-	-	匹配HASH过滤器时通过
	0	0	1	1	-	-	-	匹配HASH过滤器时不通过
	0	1	1	0	-	-	-	匹配HASH或者完美/组过滤器时通过
	0	1	1	1	-	-	-	匹配HASH或者完美/组过滤器时不通过
多播帧	1	-	-	-	-	-	-	所有帧通过
	-	-	-	-	-	1	-	所有帧通过
	0	-	-	0	0	0	-	匹配完美/组过滤器时通过，如果PCFRM = 0x，丢弃暂停控制帧
	0	0	-	0	1	0	-	匹配HASH过滤器时通过，如果PCFRM = 0x，丢弃暂停控制帧

帧类型	PM	HPFLT	HUF	DAIFLT	HMF	MFD	BFRMD	目标地址过滤器操作
	0	1	-	0	1	0	-	匹配HASH或者完美/组过滤器时通过，如果PCFRM = 0x，丢弃暂停控制帧
	0	-	-	1	0	0	-	匹配完美/组过滤器时不通过，如果PCFRM = 0x，丢弃暂停控制帧
	0	0	-	1	1	0	-	匹配HASH过滤器时不通过，如果PCFRM = 0x，丢弃暂停控制帧
	0	1	-	1	1	0	-	匹配HASH或者完美/组过滤器时不通过，如果PCFRM = 0x，丢弃暂停控制帧

表 43-6. 源地址过滤器结果列表

帧类型	PM	SAIFLT	SAFLT	源地址过滤器操作
单播帧	1	-	-	所有帧通过
	0	0	0	匹配完美/组过滤器时返回通过状态，不匹配时状态为不通过，但不丢弃不通过的帧
	0	1	0	匹配完美/组过滤器时返回不通过状态，但不丢弃帧
	0	0	1	匹配完美/组过滤器时通过，丢弃不通过的帧
	0	1	1	匹配完美/组过滤器时不通过，丢弃不通过的帧

混杂模式

若设置ENET_MAC_FRMF寄存器的PM位为'1'将使能混杂模式，此时地址过滤器无效，所有帧均可通过过滤器。同时接收状态信息的目标地址/源地址错误位总是为'0'。

暂停控制帧过滤

MAC会检测接收到的控制帧内的6字节目标地址域，若ENET_MAC_FCTL寄存器的UPFDT位设为0，则判断目标地址域的值是否符合IEEE 802.3规范控制帧的唯一值(0x0180 C200 0001)。若ENET_MAC_FCTL寄存器的UPFDT位设为1，则在与IEEE 802.3规范定义的唯一值比较外，同时与控制器所设置的MAC地址逐位比较。如果目标地址域比较通过且接收流控制被使能（ENET_MAC_FCTL的RFCEN位被置1），则相应暂停控制帧功能将被触发。这个通过过滤的暂停帧是否会被转发给应用取决于ENET_MAC_FRMF寄存器的PCFRM[1:0]位设置。

MAC 的接收流程

MAC接收到的帧都会被送入RxFIFO中。MAC接收到帧后会剥离其前导码和帧首界定码，并从帧首界定码后的第一个字节（目标地址）开始向FIFO发送帧数据。如果使能了IEEE 1588时间戳，MAC会在检测到帧的帧首界定码的时候记录下系统的当前时间。如果这个帧通过地址过滤器的检查，MAC会把这个时间戳通过接收描述符一并发给应用程序。

若ENET_MAC_CFG寄存器的APCD位置位，且接收到的帧长度/类型域的值小于0x600时，MAC将自动剥离填充域和帧校验序列。MAC会在向RxFIFO发送完帧长度/类型域规定字节数后，丢弃包括帧校验序列在内的余下字节。如果长度/类型域的值大于或等于0x600，则忽略APCD位，由TFCD位来确定是否自动剥离帧校验序列。

若看门狗定时器被使能（ENET_MAC_CFG寄存器中的WDD位被复位），当帧长度超过2048字

节时将被切断。即使看门狗定时器被禁能，MAC仍然会切断长度大于16384字节的帧。

当RxFIFO工作于直通模式时，如果FIFO中的数据量大于门限值（可通过ENET_DMA_CTL寄存器的RTHC位设置），就开始从FIFO中取出数据，并通知DMA接收。当FIFO完成取出整个帧后，MAC控制器将接收状态信息字发送给DMA控制器以回写到接收描述符中。在这种模式下，假如一个帧开始由FIFO取出由DMA发送到应用程序，则即使检测到错误，帧也会一直接收直到整个帧接收完毕。由于错误信息也要等到此时才会发送给DMA控制器，此时帧的前部分已经被DMA接收，所以在这种模式下将MAC设置成将所有错误帧丢弃将无效。

当RxFIFO工作于存储转发模式（通过ENET_DMA_CTL寄存器的RSFD位设置）时，DMA只在RxFIFO完整地收到一帧后，才将其读出。此模式下，如果MAC设置成将所有错误帧丢弃，那么DMA只会读出合法的帧，并转发给应用程序。一旦MAC在接口上检测到帧首界定码就会启动接收过程。MAC控制器在处理帧之前会剥离前导码和帧首界定码。会通过过滤器检查帧的报头，并用帧校验序列核对帧的CRC值。如果帧没能通过地址滤波器，MAC控制器就会丢弃该帧。

MAC 的接收管理

多个帧的接收处理

与TxFIFO不同，由于帧的状态信息紧随在帧数据之后，MAC可以判断接受帧的状态，因此第二个接收帧的传送是紧接着第一个接收帧的数据与状态信息的，只要RxFIFO未满，就可以存放任意数量的帧。

错误处理

- 在从MAC接收到EOF之前，RxFIFO已满。则MAC控制器会将整个帧丢弃并返回一个溢出状态。同时将溢出计数器加1；
- 若RxFIFO设置成存储转发模式，MAC可以过滤并丢弃所有的错误帧。但根据ENET_DMA_CTL寄存器的FERF和FUF位的设置，RxFIFO仍可以接收错误帧和长度低于最小帧长的帧；
- 若RxFIFO设置成直通模式，并不能将所有的错误帧都丢弃，仅当DMA从RxFIFO读出帧的SOF时，RxFIFO也已获得了该帧的错误状态时可以丢弃错误帧。

流控模块

MAC 控制器主要通过背压（半双工模式）和暂停控制帧（全双工模式）来管理帧的发送流控。

- 半双工模式流控：背压

当 MAC 采用半双工模式进行通讯时，如果设置了发送流控使能位(ENET_MAC_FCTL 寄存器的 TFCEN 位)，有两种情况可以触发背压流控。背压流控是通过发送一个 32 位的堵塞信号 0x5555 5555，通知所有其他站点发生了冲突。两种触发情况中，第一种是通过置位 ENET_MAC_FCTL 寄存器的 FLCB/BKPA 位来使能发送流控。第二种情况在接收帧时发生，MAC 在接收帧的过程中，RxFIFO 中字节数不断增大，当接收数目超过流控激活阈值（ENET_MAC_FCTH 寄存器中的 RFA 位），MAC 将置位背压挂起标志。若背压挂起标志置位了，且又有新的帧到来，MAC 将发送堵塞信号以延迟一段背压时间再接收帧。在背压时间结束后，PHY 会重新发送这个新的帧。若在背压期间，RxFIFO 中字节数大于等于流控失活阈值（ENET_MAC_FCTH 寄存器中的 RFD 位），则 MAC 会再次发送背压信号；反之，则 MAC

将复位背压挂起标志，并可以接收新的帧，不再发送堵塞信号。

■ 全双工模式流控：暂停帧

对于全双工模式，MAC 控制器使用“暂停帧”进行流控制。这种方式可以使接收端能够命令发送端暂停一段时间再发送，如当接收缓冲区快要溢出的情况。如果设置了发送流控使能位(ENET_MAC_FCTL 寄存器的 TFCEN 位)，在全双工模式下，MAC 会在以下两种情况下产生并发送暂停帧。两种情况分别为：

1) 应用程序把 ENET_MAC_FCTL 寄存器的 FLCB/BKPA 位置位，将立即发送一个暂停帧。这个暂停帧指定的暂停时间为ENET_MAC_FCTL寄存器中PTM位配置好的暂停时间值。如果应用程序前面要求了一段时间的暂停，但在这段时间内，应用程序准备好了，可以不需要剩余的暂停时间了，这时应用程序需要发一个零时间片暂停帧来通知发送方可以继续发送了。零时间片暂停帧是通过设置 ENET_MAC_FCTL 寄存器中的 PTM 位为 0，并将 FLCB/BKPA 位置位来发送的；
2) 在 RxFIFO 满足一定的条件下，MAC 会自动发送暂停帧。在接收过程中，RxFIFO 不停地有数据进来，同时 RxFIFO 也取出数据给 RxDMA，如果 RxFIFO 取出数据的频率小于其接收数据的频率，RxFIFO 中的数据就会越来越多。一旦 RxFIFO 中的数据量超过了流控的激活阈值（ENET_MAC_FCTH 寄存器中的 RFA 位），MAC 将发送一个暂停时间为 PTM 位定义的值的暂停帧。发送暂停帧之后，MAC 将启动一个计数器，计数器的时间由 ENET_MAC_FCTL 寄存器的 PLTS 位定义，当到了计数器规定的时间，MAC 将重新检查 RxFIFO。此时若 RxFIFO 中的数据量仍然大于流控激活阈值，MAC 将再次发送一个暂停帧。若 RxFIFO 中的数据量小于流控失活阈值，并且 ENET_MAC_FCTL 寄存器中的 DZQP 位被复位，则 MAC 将发送一个零时间片暂停帧。这个零时间片暂停帧用于指示远程站点结束暂停，本地缓存区已经准备好接收新的数据帧。

MAC 通过如下方式管理帧的接收流控：

在全双工模式下，MAC 能够检测暂停帧，并按照暂停帧中的暂停时间域参数，在暂停一定时间后再发送数据。可以通过设置 ENET_MAC_FCTL 寄存器的 RFCEN 位，使能或者取消暂停帧检测功能。如果没有使能该功能，则 MAC 会忽略接收到的暂停帧。若使能了该功能，MAC 将能够对接收到的暂停帧进行解码。类型域、操作数域和暂停时间域都将能够被 MAC 识别。在暂停期间，如果收到一个新的暂停帧，则新的暂停时间将立即被加载到暂停时间计数器中。如果接到收的暂停时间域值为 0，则 MAC 会停止暂停时间计数器，恢复数据的发送。通过配置 ENET_MAC_FRMF 寄存器的 PCFRM 位值，来处理这些接收到的控制帧。

校验和引擎

以太网控制器具有发送校验和的功能，支持计算校验和，并在发送时插入计算结果，以及在接收时侦测校验和错误。

如下描述了发送帧的校验和的操作功能。

注意：只有将 ENET_DMA_CTL 寄存器的 TSFD 位置为‘1’（TxFIFO 设置成存储转发模式），同时必须保证 TxFIFO 的深度足够容纳将要发送的完整帧时，才能使能此功能。若 FIFO 深度小于帧长度，则仅仅计算和插入 IPv4 报头的校验和域。

欲了解 IPv4、TCP、UDP、ICMP、IPv6 和 ICMPv6 报头的规范，请分别查阅 IETF 规范 RFC 791、RFC 793、RFC 768、RFC 792、RFC 2460 和 RFC 4443。

■ IP头校验和

若以太网帧的类型域值为 0x0800 同时 IP 数据包的版本域值为 0x4，则校验和模块标记其为 IPv4 数据包并会用计算结果取代帧的校验和域的内容。IPv6 的报头不包含校验和域，因此校验和模块不会改变 IPv6 报头的值。IP 头校验和计算完毕之后，其结果会写到发送描述符 0 的 IPHE 位。当发生下述情况时，IPHE 错误状态位会被硬件置'1'：

1) 对于IPv4数据帧：
   a) 接收到的以太网类型域值为0x0800，但IP报头版本域的值不等于0x4；
   b) IPv4报头长度域的值大于帧的总长度；
   c) IPv4报头长度域值小于IP报头总长0x5（20字节）。

2) 对于IPv6数据帧：
   a) 接收到的以太网类型域值为0x86dd，但IP报头版本域的值不等于0x6；
   b) 帧在完全接收IPv6报头或者扩展报头之前结束。IPv6标准报头长度为40字节，扩展报头包含相应的报头长度字段。

■ TCP/UDP/ICMP校验和

校验和模块通过分析IPv4或IPv6报头(包括扩展报头)来判断帧的类型（TCP、UDP或ICMP）。

当帧发生以下情况时，将绕过校验和功能，校验和模块不对这些帧进行处理：

1) 不完整的IPv4或IPv6帧；
2) 包含安全功能的IP帧(如验证报头或者封装有安全数据)；
3) 非TCP/UDP/ICMPv4/ICMPv6数据的IP帧；
4) 带路由报头的IPv6帧。

校验和模块会对TCP、UDP或者ICMP的数据进行计算，并插入报头的相应域。它有以下2种工作模式：

1) 校验和计算不包括TCP、UDP或者ICMPv6的伪首部。并假定输入帧的校验和字段已有值。校验和字段包含在校验和计算中，在计算完成后插入并替换原校验和域的值；
2) 校验和计算包括TCP、UDP或者ICMPv6的伪首部。将传输帧的校验和字段清零。进行校验和的计算，计算完成后插入传输帧的原校验和域。

校验和计算完毕之后，其结果会写到发送描述符0的IPPE位。当发生下述情况时，IPPE错误状态位会被硬件置'1'：

1) 在存储转发模式下，帧未被完整写入FIFO之前就被转发给MAC控制器；
2) 帧已发送完毕，但MAC从FIFO中取出的数据包字节数小于IP报头中数据长度域标明的字节数。

如果数据包长度大于标明的长度，不会报告错误，之后的数据会被当成填充字节而丢弃。如果检测到第一类错误情况，校验和的值不会插入TCP、UDP或者ICMP报头。如果检测到第二类错误情况，仍然会把校验和计算结果插入报头的相应域。

注意：无论采用哪种模式，对于 IPv4 上的 ICMP 数据包，由于这类数据包没有定义伪报头，为正确计算其校验和，校验和域内容必须为 0x0000。

接收帧校验和的操作功能描述如下所述。

置位 ENET_MAC_CFG 寄存器的 IPFCO 位，可以使能接收校验和模块。接收校验和模块可以计算 IPv4 报头的校验和，并检查它是否与 IPv4 报头的校验和域的内容相匹配以外。MAC 可根据检查接收到的以太网帧类型域是 0x0800 还是 0x86dd，来判别是 IPv4 帧还是 IPv6 帧，这个方法也用于带 VLAN 标签的帧识别。DMA 接收描述符的报头校验和错误位（接收描述符 0 中的 IPHERR 位）反映了对报头的校验和结果，该位在接收到的 IP 报头出现下述错误时被置 1：

- 计算的IPv4报头的校验和值与其校验和域的内容不匹配；
- 以太网类型域值指示的数据类型与IP报头版本域不匹配；
- 接收到的帧长少于IPv4报头长度域指示的长度，或者IPv4/IPv6报头少于20字节。

接收校验和模块还能识别IP数据包的数据类型是TCP、UDP还是ICMP，并按照TCP、UDP或ICMP的规范计算它们的校验和。计算过程包括TCP/UDP/ICMPv6伪报头的数据。DMA接收描述符（接收描述符0的PCERR位）的数据校验和错误位反映了对数据的校验和结果，该位在接收到的IP数据包数据出现下述错误时被置1：

- 计算的TCP、UDP或ICMP校验和与其帧的TCP、UDP或ICMP校验和域值不匹配；
- 收到的TCP、UDP或者ICMP数据长度与IP报头给出的长度不符。

接收校验和模块不计算下列情况：不完整的IP数据包、带安全功能的IP数据包、IPv6路由报头以及数据类型不是TCP、UDP或者ICMP的数据包。

MAC 回环模式

通常地，回环模式用于应用程序对系统硬件和软件的测试与调试。通过将ENET_MAC_CFG寄存器的LBM位置’1’，可以使能MAC回环模式。在该模式下，MAC发射端把帧发送到自身的接收端上。该模式默认为关闭。

43.3.3. DMA 控制器描述

为了减少CPU的干预，设计了以太网专用DMA控制器，用于实现FIFO和系统存储之间的帧数据传输。CPU和DMA之间的的通讯通过2种数据结构实现。分别是：1）描述符列表（链结构或环结构）和数据缓存；2）控制和状态寄存器。应用程序需要开辟存储描述符列表及数据缓存用到的物理内存。在存储器里，描述符是以指向缓存的指针的形式存放。有2个描述符队列，一个用作发送，另一个用作接收。两个队列的基地址分别存放在ENET_DMA_TDTADDR寄存器和ENET_DMA_RDTADDR寄存器中。当DFM位为0时，发送描述符由四个描述符字发送描述符0-4）组成，当DFM位为1时，发送描述符由八个描述符字（发送描述符0-8）组成。同样的，当DFM位为0时，接收描述符由四个描述符字（接收描述符0-4）组成，当DFM位为1时，接收描述符由八个描述符字（接收描述符0-8）组成。每个描述符可以指向最多2个缓存用来存储帧的数据。根据描述符列表类型是环结构还是链结构，来决定第二个缓存是被配置为第二个数据存储地址，还是下一个描述符地址。数据缓存存放在MCU的物理内存里，可以存放一个帧的全部或者部分，但是不允许存放不属于同一个帧的数据。描述符队列可以是显性（链结构）或者隐性（环结构）的方式前向连接的。通过设置接收描述符1的RCHM位和发送描述符的TCHM位为’1’，可以实现描述符的显性连接，此时接收描述符2及发送描述符2中将存放缓存地址，接收描述符3及发送描述符3中将存放下一个描述符的地址，这种链接的描述符也可以称为描述符的链结构。通过设置接收描述符1的RCHM位和发送描述符的TCHM位为’0’，可以实现描述符的隐性连接，此时接收描述符2/发送描述符2，接收描述符3/发送描述符3中都将存放缓

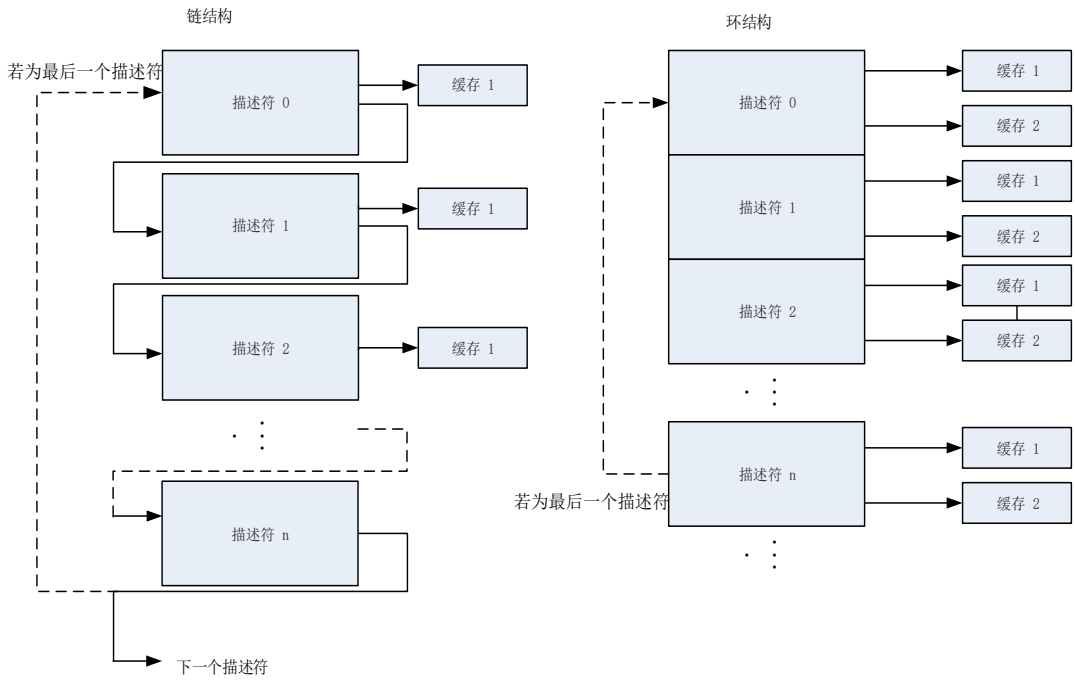
存地址，这种链接的描述符也可以称为描述符的环结构。在使用当前的描述符所指向的缓存地址时，描述符指针就指向下一个描述符。当使用链结构时，描述符指针指向的是第二个缓存。当使用环结构，根据下式计算描述符指针下一个所指向的地址：

DFM=0：下个描述符地址 = 当前描述符地址 + 16 + DPSL * 4

DFM=1：下个描述符地址 = 当前描述符地址 + 32 + DPSL * 4

若当前描述符是描述符列表的最后一个描述符，环结构下必须设置发送描述符0的位21或接收描述符1的位15以标识当前描述符为列表的最后一个。此时下一个描述符又指向描述符列表的第一个。链结构下还可以通过设置发送描述符3或接收描述符3的值指向描述符列表中第一个的地址。DMA一旦检测到帧结束就会跳到下一个帧的缓存。

图 43-6. 描述符的环结构和链结构



数据缓存地址对齐

以太网DMA控制器支持所有对齐类型：字节对齐，半字对齐，字对齐。这意味着应用程序可将发送和接收数据缓存地址配置到任意地址。但是，在DMA发起传输的时候，总是以字对齐的方式访问地址。对于读和写缓存的访问也不一样。示例如下：

- 读缓存示例：如果发送缓存的地址为0x2000 0AB2，并需要传输15字节。在开始读操作后，DMA实际会从地址0x2000 0AB0，0x2000 0AB4，0x2000 0AB8，0x2000 0ABC和0x2000 0AC0先读5个字，但是在往FIFO发送数据的时候，会丢弃头2个字节(0x2000 0AB0和0x2000 0AB1)和最后3个字节（0x2000 0AC1，0x2000 0AC2和0x2000 0AC3）。
- 写缓存示例：如果接收缓存的地址为0x2000 0CD2，并需要传输16字节。在开始写操作后，DMA实际会从地址0x2000 0CD0到0x2000 0CE0先写5个32位数据。但是头2个字节（0x2000 0CD0和0x2000 0CD1）和末尾的2个字节（0x2000 0CE2和0x2000 0CE3）会用虚拟字节替代。

注意：DMA控制器不会写任何数据到定义的缓存区之外的地址。

缓冲区有效长度

发送帧的过程中，TxDMA会传输与发送描述符1中标明的缓存有效长度的字节给MAC控制器。如前所述，一个发送帧可以用多个描述符来描述一个帧，即一个帧的数据可以处于多个不同的缓存中。如果DMA控制器读取的发送描述符0的FSG位为'1，那么DMA就明确了当前缓存存储的是一个新的帧，并标记发送的第一个字节是帧首。如果DMA控制器读取的发送描述符0的LSG位为'1'，则DMA就明确了当前缓存存储的是当前帧的最后一部分数据。通常来说一个帧只存在一个缓存里（因为缓存的大小对于一个正常的帧来说足够大了），因此FSG和LSG位会在一个相同的描述符中同时置位。

接收帧的过程中，接收帧的缓存长度域值必须是字对齐的。对于字对齐或非字对齐的缓存地址，接收操作与发送操作不大相同。如果接收缓存地址是字对齐的，则与发送流程是类似的，缓存的有效长度为由接收描述符1中配置的值。如果接收缓存地址是非字对齐的，则缓存的有效长度将小于接收描述符1中配置的值。缓存有效长度值应为接收描述符1中配置的值减去缓存地址的低2位值。例如，假设缓存的总大小为2048字节，缓存地址为0x2000 0001，地址的低2位值为0b01，那么缓存有效长度为2047个字节，范围从0x2000 0001（帧首）到0x2000 07FF。

当收到了一个帧起始SOF，则DMA控制器将FSG位置位，当收到一个帧结束EOF时，则LSG位被置位。如果接收缓存长度域值配置的足够大，能放下整个帧，则FSG和LSG位将在同个描述符中被置位。实际接收的帧长度可从接收描述符0的FRML位域获取。从而应用程序可计算未被使用的缓存空间。RxDMA总是用新的描述符来接收下一帧。

TxDMA 和 RxDMA 的仲裁器

DMA的仲裁器设计了两种仲裁方式用于提高DMA发送与接收控制器的效率：固定和轮询优先级。设置ENET_DMA_BCTL寄存器DAB位为'0'，选择轮询优先级，在TxDMA和RxDMA同时要求访问数据总线的时候，按照ENET_DMA_BCTL寄存器RTPR位设定的比例对其访问进行分配。设置DAB位为'1'选择固定优先级，此时RxDMA和TxDMA同时要求访问时，RxDMA总是对总线拥有更高的访问优先级。

DMA 错误状态

若DMA在传输过程中出现了错误的总线响应，那么DMA控制器认为发生了一个致命错误，会立刻停止所有操作，并更新状态寄存器ENET_DMA_STAT。在发生类似的致命错误（响应错误）之后，应用程序必须复位以太网外设并重新初始化DMA，DMA才能恢复操作。

TxDMA 与 RxDMA 控制器的初始化

在使用DMA控制器之前，必须按如下步骤对DMA进行初始化：

1. 对ENET_DMA_BCTL寄存器进行总线访问参数的相关设置；
2. 对ENET_DMA_INTEN寄存器进行设置，屏蔽不需要的中断源；
3. 将发送描述符列表和接收描述符列表的基地址分别写入ENET_DMA_TDTADDR寄存器与ENET_DMA_RDTADDR寄存器中；
4. 对相关的寄存器进行期望的过滤器配置；

5. 根据从PHY读出的自协商的结果，设置SPD位和DPM位的值，来选择通讯模式（半/全双工）及通讯速度（10Mbit/s或100Mbit/s）。将ENET_MAC_CFG寄存器的TEN和REN位置'1'，使能MAC的发送和接收操作；
6. 设置ENET_DMA_CTL寄存器的位STE和位SRE为'1'，使能DMA发送和接收器。

注意：如果HCLK频率过低，应用程序可以先使能DMA接收器，再将ENET_MAC_CFG寄存器的REN位置'1'，以避免RxFIFO在启动的时候溢出。

DMA 发送帧处理

如前所述，一个帧可以分散在不同缓存内，这意味着需要多个描述符。当 FSG 位置位，表示当前描述符指向的缓存为帧头，当 LSG 位置位，表示当前描述符指向的缓存为帧尾。对于当前帧其他描述符（LSG 位为'0'的描述符），TxDMA 控制器仅修改清零其 DAV 位。在这最后一个缓存的数据发送完毕以后，DMA 会将整个帧的发送状态信息，写入最后一个的发送描述符 0 并返回。将数据从系统存储传输到 FIFO，开始发送数据，但实际上真正的数据发送是由 TxDMA 模式决定的：直通模式和存储转发模式。直通模式在 FIFO 中的字节数大于所配置的阈值时，数据将取出到 MAC 发送。存储转发模式在整个帧数据都传入 FIFO 后或 FIFO 快要填满时再取出数据给 MAC 进行发送。

DMA 发送管理

发送缓存区中第二帧操作

如果ENET_DMA_CTL寄存器中OSF位为'0'，则发送顺序为：首先读取发送描述符，然后从系统存储读取数据写到FIFO，再将帧数据通过MAC放到接口上，最后等待数据发送完毕后将发送状态写回描述符。

上述是TxDMA的标准发送流程，但当HCLK远远大于TX_CLK时，在发送两个帧时发送效率将显著降低。

为避免上述提及的情况，应用程序可将OSF位置位。在此情况下，第二帧的数据可以不等待第一帧的描述符状态信息被写回，就先读取内存里的第二帧数据，并把它们送进FIFO。OSF功能仅在两相邻帧之间起作用。

TxDMA 操作模式（A）（默认）：非 OSF

在默认模式下，TxDMA控制器的工作流程如下：

1. 初始化帧数据到发送缓存，并对发送描述符0-3进行设置，置发送描述符0的DAV位为'1'；
2. 将ENET_DMA_CTL寄存器的STE位置为'1'，使能TxDMA控制器；
3. TxDMA控制器开始轮询发送描述符列表来获取待发送的帧。如果TxDMA检测到发送描述符0的位DAV为0，或者发生了错误，则控制器就会终止传输进入挂起状态，并设置ENET_DMA_STAT寄存器的发送缓存不可用位（位2）和正常中断汇总位（位16）为'1'。如果处于挂起状态，则发送控制器操作跳至步骤8；
4. 如果取到的描述符标志位显示该描述符由DMA占有（DAV位被置'1'），那么DMA从描述符中解析出所配置的发送帧以及发送数据缓存的地址；
5. DMA从内存中取出数据并将数据存入TxFIFO；
6. TxDMA控制器会一直轮询描述符列表直到帧结尾被传送出去（LSG位置位）。如果当前描

述符的LSG位为'0'，则在所有缓存数据送入TxFIFO之后，将DAV位清零以关闭这个描述符。然后TxDMA控制器等待写回描述符状态，以及IEEE 1588时间戳值（如果使能了时间戳功能）；

7. 在整个帧发送完成以后，仅当发送描述符0位INTC为'1'时，发送状态位（ENET_DMA_STAT寄存器中的TS位）会被置位。此时若使能了DMA中断，将进入相应中断。然后DMA控制器返回步骤3，继续处理下一帧；
8. 在挂起状态下，如果向发送查询使能寄存器ENET_DMA_TPEN写入任意值，并清除发送溢出标志位，TxDMA将重新回到运行状态，尝试重新获取描述符。发送控制器操作回到步骤3。

TxDMA 操作模式(B)：OSF

在操作第二帧(OSF)模式下，TxDMA可以不必等到前一帧的状态信息写回，就发送下一帧。如果系统时钟频率远远大于MAC频率（10Mbit/s或100Mbit/s），这种情况OSF模式可以提高发送效率。设置ENET_DMA_CTL寄存器的位OSF为'1'，进入此模式。DMA在发送完前一帧数据后，不必等到前一阵的状态写回，而是立即查询第二帧的发送描述符，如果第二帧发送描述符的DAV位与FSG位都置1，那么TxDMA立即读取第二帧的帧数据并将其存入MAC FIFO。

在OSF模式下，TxDMA的操作流程如下：

1. 按照TxDMA默认模式的步骤1-6操作；
2. DMA不等到关闭前一帧的最后一个描述符（LSG位为'1'），就直接取下一个描述符；
3. 如果取到的描述符标明被DMA占有（DAV位为'1'），那么就从解析的发送缓存地址中读取下一帧的数据。如果DAV位为'0'即DMA不占有这个描述符，则TxDMA进入挂起状态并跳到步骤7；
4. TxDMA控制器会一直轮询描述符列表直到帧结尾被传送出去。如果一个帧由多个描述符描述，则中间描述符会在获取之后就被关闭；
5. TxDMA等待前一帧的发送状态信息和时间戳（如果使能了时间戳功能），在接收到状态信息后，DMA会把DAV位为'0'的状态信息写入发送描述符0，将该描述符的占有权交还给CPU进行操作；
6. 在整个帧发送完成以后，仅当发送描述符0位INTC为'1'时，发送状态位（ENET_DMA_STAT寄存器中的TS位）会被置位。此时若使能了DMA中断，将进入相应中断。如果前一个帧返回的状态信息正常则跳到步骤3。若显示有数据下溢错误，TxDMA进入挂起状态，并跳到步骤7；
7. 在挂起状态下，如果TxDMA收到一个发送帧的待处理的状态信息和时间戳（若使能了时间戳），则TxDMA将这些信息写入发送描述符，并将相应描述符的的DAV位清零。随后设置相关的中断标志位并回到暂停状态；
8. 在挂起状态下，如果向发送查询使能寄存器ENET_DMA_TPEN写入任意值并将溢出中断标志位清零，TxDMA将回到运行状态，尝试重新获取描述符。发送控制器操作根据是否有待处理的状态信息跳到步骤1或者步骤2。

发送帧格式

根据前述的IEEE 802.3规范，一个正常的发送帧应该由以下及部分构成：前导码，帧首界定码SFD，目标地址DA，源地址SA，QTAG前缀（可选），长度/类型域LT，数据，PAD填充域（可选），和帧校验序列FCS。

前导码和帧首界定码都是由MAC自动生成的，因此应用程序只需要存储目标地址，源地址，QTAG（若需要），长度/类型，数据，填充域（若需要），帧校验序列（若需要）。如果帧需要填充位，即缓存中没有存储填充位和帧校验序列部分，则应用程序可配置自动生成帧校验序列和填充位功能。如果帧仅需帧校验序列，即缓存中没有存储帧校验序列部分，则应用程序可配置自动生成帧校验序列。DPAD位和DCRC位用于配置填充位和帧校验序列的自动生成。

发送查询挂起后的处理

当传输开始后DMA会不断对发送描述符进行查询，当发生如下情况时，会导致DMA进入挂起状态，并暂停发送。此时当前描述符固定为暂停前的最后一个描述符。

- DMA检测到发送描述符0的DAV位为0，此时CPU占有描述符，则会进入挂起状态，并暂停查询。同时设置ENET_DMA_STAT寄存器的正常中断总结位NI和发送缓存不可用位TBU为'1'；
- 当接口在发送帧的过程中MAC FIFO为空，意味着检测到了数据下溢错误。在此情况下，设置ENET_DMA_STAT寄存器的异常中断总结位AI和发送数据下溢位TU为'1'，同时把该信息写入发送描述符0。

带 IEEE 1588 时间戳的 TxDMA 描述符格式

如果设置TTSEN位为'1'，则使能了IEEE1588功能。TxDMA控制器会在帧发送完成后，将时间戳写入描述符同时设置TTMSS位为'1'。写时间戳值的地址由ENET_DMA_BCTL寄存器的DFM位决定。如果DFM位为'0'即描述符格式为常规模式，则时间戳将覆盖写到发送描述符2和发送描述符3上。如果DFM位为'1'即描述符格式为增强型模式，则时间戳将写到发送描述符6和发送描述符7上，发送描述符2和发送描述符3中的值保持不变。

常规 TxDMA 描述符

常规TxDMA描述符结构体包含4个32位字，发送描述符0~发送描述符3。发送描述符0~发送描述符3的的位定义如下：

注意：若一个帧由多个描述符表示，则对于描述符的控制位（除了INTC位）只有第一个描述符的才有效。状态信息和时间戳（若使能了时间戳功能）只写回到最后一个描述符。

图 43-7. 常规发送描述符



- 发送描述符0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DAV	INTC	LSG	FSG	DCRC	DPAD	TTSEN	保留	CM[1:0]		TERM	TCHM	保留		TTMSS	IPHE
rw	rw	rw	rw	rw	rw	rw		rw		rw	rw			rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ES	JT	FRMF	IPPE	LCA	NCA	LCO	ECO	VFRM	COCNT[3:0]				EXD	UFE	DB
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				rw	rw	rw

位/位域	名称	描述
31	DAV	DAV位 DMA会在将帧完整传输或者描述符指向的缓存里的数据全部被读出以后把该位清'0'。当一个帧位于多个缓存中时，第一个缓存描述符的DAV位，必须在后面缓存描述符的DAV位全部置'1'以后，才能置'1'。 0：表示CPU占有描述符 1：表示DMA占有描述符
30	INTC	完成时中断位 LSG位置位后，此位才有效。 0：帧发送完成时，ENET_DMA_STAT寄存器的TS位不被置位 1：帧发送完成时，ENET_DMA_STAT寄存器的TS位被置位
29	LSG	最后分块位 此位指示缓存是否包含帧的最后一个分块。 0：该描述符缓存中没有存放帧的最后一个分块 1：该描述符缓存中存放有帧的最后一个分块
28	FSG	第一分块位 此位指示缓存是否包含帧的第一个分块。 0：该描述符缓存中没有存放帧的第一个分块 1：该描述符缓存中存放有帧的第一个分块
27	DCRC	不计算CRC位 只有在FSG位置位时，此位才有效。 0：MAC 在传输帧末尾自动插入 CRC 域 1：MAC不在传输帧末尾自动插入CRC域
26	DPAD	不填充位 只有在 FSG 位置位时，此位才有效。 0：DMA 对传输帧自动添加填充字节，并且插入 CRC 数值。发生填充时，CRC 会被插入，忽略 DCRC 位的值。 1：MAC 不对传输帧自动填充字节 **注意**：此处的传输帧小于64字节。
25	TTSEN	使能发送时间戳位 只有在 FSG 位置位时，此位才有效。 0：发送时间戳功能失能 1：当ENET_PTP_TSCTL寄存器的TMSEN位为'1'时，传输帧的IEEE1588硬件时间戳功能使能。

24	保留	必须保持复位值。
23:22	CM[1:0]	校验和插入模式位 0x0：不插入校验和 0x1：只使能硬件IP报头的校验和计算和插入 0x2：使能硬件IP报头和数据域的校验和计算和插入，但是不计算伪报头的校验和 0x3：使能硬件IP报头和数据域的校验和计算和插入，也计算伪报头的校验和
21	TERM	环形发送结束模式位 该位仅在环模式下使用，且比TCHM位具有更高优先级。 0：当前描述符还不是描述符队列的最后一个 1：当前描述符到达描述符队列的最后一个，DMA返回列表的基地址
20	TCHM	第二地址链表模式位 该位在链模式下使用。该位为'1'时，忽略TB2S[12:0]的值。 0：描述符里的第二个地址是第二缓存的地址 1：描述符里的第二个地址是下一个描述符的地址，而不是第二个缓存的地址
19:18	保留	必须保持复位值。
17	TTMSS	发送时间戳状态位 只有在LSG位置位时，此位才有效。 0：还未记录帧的时间戳信息 1：记录下了描述符对应的帧时间戳，记录的时间戳放在发送描述符2（或发送描述符6，假如DFM=1）和发送描述符3（或发送描述符7，假如DFM=1）处。
16	IPHE	IP报头错误位 发生下列任意一种情况，则产生IP报头错误： IPv4帧： 1）报头长度域值小于0x5。 2）报头长度域值与报头的长度不符。 3）报头版本域值与帧长度/类型域值不匹配。 IPv6帧： 1）主报头长度不足40字节。 2）报头版本域值与帧长度/类型域值不匹配。 0：未发现IP数据包报头的错误 1：MAC发送端发现了IP数据包报头的错误
15	ES	错误汇总 该位为下列位的逻辑"或"： IPHE：IP 报头错误 JT：Jabber 超时 FRMF：帧清空 IPPE：IP 数据错误 LCA：载波丢失 NCA：无载波 LCO：延迟冲突 ECO：过度冲突

EXD：过度顺延

UFE：数据下溢错误

14	JT	Jabber超时位 该位仅当JBD位复位时才会被置'1'。 0：未发生Jabber超时 1：MAC发送端发生了Jabber超时
13	FRMF	帧清空位 置1时，清空TxFIFO中的数据。
12	IPPE	IP数据错误位 发送端会核对IPv4或者IPv6报头的数据长度域值与实际收到的TCP、UDP和ICMP数据数目，不符合就置'1'报错。 0：未发生IP数据错误 1：MAC发送端发现了IP数据包的TCP、UDP或者ICMP的IP数据错误。
11	LCA	载波丢失位 在发送时，如果CRS信号在一个或一个以上发送时钟周期中为无效状态，并且没有发生冲突，则载波丢失将概率性发生。 该位只有在半双工模式下有效。 0：未发生载波丢失 1：帧发送的时候发生了载波丢失
10	NCA	无载波位 0：PHY的载波侦听信号有效 1：帧发送的时候PHY的载波侦听信号无效
9	LCO	延迟冲突位 如果冲突在64字节（包括前导符）发送之后发生，则这种情况称作延迟冲突。 0：未发生延迟冲突 1：发生了延迟冲突 **注意：**如果溢出错误位UFE置'1'，该位无效。
8	ECO	过度冲突位 如果MAC设置寄存器的RTD（不进行重试）位为'1'，那么在发生一次冲突后，该位就置'1'。 如果MAC设置寄存器的RTD（不进行重试）位为'0'，那么在连续发生16次冲突后，该位置'1'。 若该位置位，则中止当前帧的发送。 0：未发生过度冲突 1：发生了过度冲突
7	VFRM	VLAN帧位 0：发送帧为普通帧 1：发送的帧是VLAN帧

6:3	COCNT[3:0]	冲突计数位 只有在 ECO 位为 0 时，此位才有效。 该4位计数值记录了帧发送出去前出现的冲突次数。
2	EXD	过度顺延位 当MAC设置寄存器的顺延位DFC为'1'时有效。 0：未发生过度顺延 1：由于顺延超过3036字节的时间而结束发送
1	UFE	数据下溢错误位 数据下溢错误表示由于从系统存储传输数据到FIFO的速度过慢，导致DMA在发送帧的时候遇到了空的缓存。发送过程进入挂起状态，并将ENET_DMA_STAT寄存器的发送数据下溢位TU（位5）和发送状态位TS（位0）都置'1'。 0：未发生数据下溢错误 1：发生了数据下溢错误，MAC中止帧的发送。
0	DB	顺延位 该位指示了是否由于载波侦听信号CRS在MAC发送帧之前被占用，而导致发生帧的顺延。 该位只在半双工模式下有效。 0：未发生发送顺延 1：MAC发生了顺延，推迟发送。

■ 发送描述符1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			TB2S[12:0]												
			rw												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			TB1S[12:0]												
			rw												

位/位域	名称	描述
31:29	保留	必须保持复位值。
28:16	TB2S[12:0]	发送缓存 2 大小 第二个数据缓存的大小（以字节记），如果发送描述符0的位TCHM为'1'时，这些位无效。
15:13	保留	必须保持复位值。
12:0	TB1S[12:0]	发送缓存 1 大小 第一个数据缓存的大小（以字节记），这些位的值为0时，则DMA忽略这个缓存，并根据发送描述符0的THCM位使用缓存2(TCHM=0)或者下一个描述符的缓存(TCHM=1)。

■ 发送描述符2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TB1AP/TTSL[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TB1AP/TTSL[15:0]															
rw															

位/位域	名称	描述
31:0	TB1AP/TTSL[31:0]	发送缓存1地址指针/发送帧时间戳低32位 在发送帧之前，应用程序必须对这些位进行配置发生缓存1地址（TB1AP），等到数据发送完后，DMA可以用它们存放帧的时间戳低32位（若DFM=0）。但若DFM=1，这些位将不会被修改，保持为缓存地址。 当这些位的值表示缓存1的物理地址（TTSL）时，对缓存的地址对齐不做限制。 当这些位的值表示时间戳低32位（TB1AP）时，当前描述符的TTSEN位和LSG位必须置位。

- 发送描述符3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TB2AP/TTSH[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TB2AP/TTSH[15:0]															
rw															

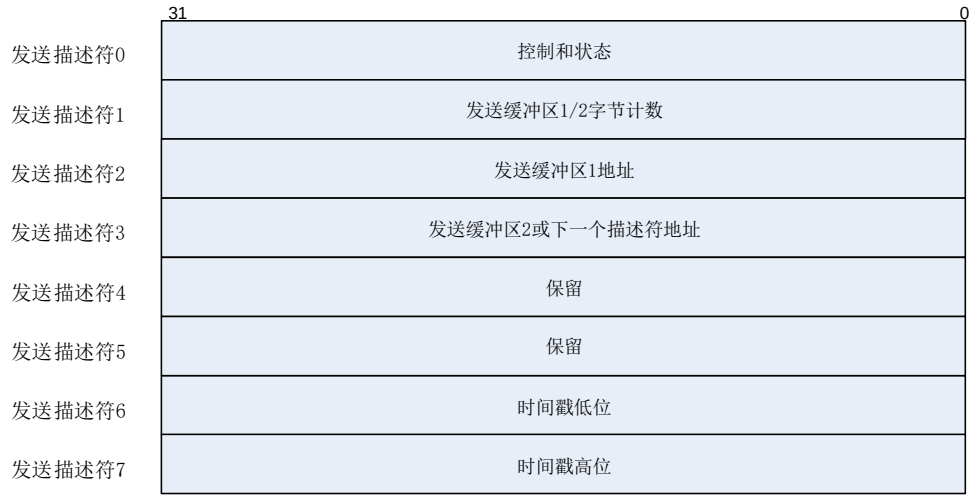
位/位域	名称	描述
31:0	TB2AP/TTSH[31:0]	发送缓存 2 地址指针（下个描述符地址）/发送帧时间戳高 32 位 在发送帧之前，应用程序必须对这些位进行配置发生缓存 2 地址（TB2AP），或者配置下一个描述符地址（由描述符类型是链型还是环型决定）。等到数据发送完后，DMA 可以用它们存放帧的时间戳高 32 位 TTSH（若 DFM=0，且 TTSEN=1）。但若 DFM=1 或 TTSEN=0，这些位将不会被修改。 当这些位的值表示缓存 2 的物理地址时（TCHM=0），对缓存的地址对齐不做限制。 当这些位的值表示下个描述符地址时（TCHM=1），这些位必须是字对齐的。 当这些位的值表示时间戳高32位时，当前描述符的TTSEN位和LSG位必须置位。

增强 TxDMA 描述符

增强TxDMA描述符结构体包含8个32位字：发送描述符0~7。发送描述符0~3的位定义与常规TxDMA描述符相同，发送描述符4~7的位定义如下：

注意：若一个帧由多个描述符表示，则对于描述符的控制位（除了INTC位）只有第一个描述符的才有效。状态信息和时间戳（若使能了时间戳功能）只写回到最后一个描述符。

图 43-8. 增强发送描述符



- 发送描述符4
  所有位保留。
- 发送描述符5
  所有位保留。
- 发送描述符6

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

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

位/位域	名称	描述
31:0	TTSL[31:0]	发送时间戳低 32 位 当 TTSEN 位和 LSG 位都为 1 时，这些位用于记录当前发送帧的时间戳低 32 位值。

- 发送描述符7

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

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

位/位域	名称	描述
31:0	TTSH[31:0]	发送时间戳高 32 位 当 TTSEN 位和 LSG 位都为 1 时，这些位用于记录当前发送帧的时间戳高 32 位值。

DMA 接收帧处理

当接口上出现一个帧的时候，MAC开始接收帧。同时，地址过滤模块开始工作，如果这个帧没有通过地址过滤，则MAC RxFIFO将忽略该帧，不会将其通过RxDMA转发给接收缓存。如果这个帧通过了地址过滤，则其在不同的转发条件满足时会被转发给接收缓存。在直通模式下，这个转发条件是指接收的帧长大于等于设好的接收阈值。在存储转发模式下，这个转发条件是指FIFO里存入了完整的帧时。在接收帧的过程中，当发生以下任意一种情况时，将丢弃RxFIFO中的数据，并且不转发数据：

- RxFIFO中数据少于64字节；
- 在接收过程中发生了冲突；
- 提前终止接收帧。

当满足转发条件时，RxDMA控制器开始将数据从RxFIFO中传输到接收缓存中。若当前缓存中包含了帧起始，则在RxDMA控制器写回帧接收状态的时候会将接收描述符0中的FDES位置位，以表明这个描述符中存储的是帧的第一部分。若当前缓存中包含了帧结尾，则在RxDMA控制器写回帧接收状态的时候会将接收描述符0中的LDES位置位，以表明这个描述符中存储的是帧的最后一部分。通常当接收缓存大小大于接收帧的长度时，FDES位和LDES位会在同一个描述符中置位。当缓存接收到了帧结尾，或者当前描述符的缓存不足以存储整个帧时，RxDMA将获取下一个接收描述符，并将上一个描述符的接收描述符0的DAV位清零以关闭上个描述符。当LDES位置位时，描述符其他状态也会更新，并且ENET_DMA_STAT寄存器中的RS位将置位（当DINTC=0时立刻置位，当DINTC=1时延迟置位）。当接收到一个新的帧时，如果描述符的DAV位为'1'，则重复上述的RxDMA控制器操作。如果描述符的DAV位为'0'，则DMA控制器进入挂起状态，并设置ENET_DMA_STAT寄存器的RBU位为'1'。记录描述符列表地址指针当前值，并在退出挂起状态后作为描述符开始的地址。

DMA 接收管理

RxDMA控制器的工作流程如下：

1. DMA接收描述符初始化，置接收描述符0的DAV位为'1'；
2. 将ENET_DMA_CTL寄存器的SRE位置为'1'，使能RxDMA控制器。DMA进入运行状态后，会从ENET_DMA_RDTADDR寄存器配置的描述符列表基地址获取接收描述符。如果获取的描述符DAV位为1，则当前描述符开始接收帧。但如果检测到取到的描述符正在被CPU操作而不可用（DAV=0），则DMA进入挂起状态，跳到步骤9；
3. 如果获取的描述符显示描述符由DMA占有（DAV=1），那么该描述符的控制位和缓存地址就会被DMA所记录；
4. 处理接收到的帧，并从RxFIFO将数据传输到接收缓存；
5. 如果缓存被填满或者帧传输结束，接收控制器会从描述符队列中获取下一个接收描述符；
6. 如果当前帧传输结束，DMA操作跳到步骤7。如果当前帧传输没有结束(未接收到帧尾EOF)，则可能发生两种情况：
   - 下一个描述符的DAV位为'0'。如果接收帧清空功能使能，则RxDMA控制器将接收描述符0的描述符错误位DERR位置位。然后RxDMA控制器将当前描述符的DAV位清零以关闭描述符，并根据帧清空功能是否使能来确定是否置位LSG位（若使能则置位LSG，反之则不置位LSG）。之后DMA操作跳到步骤8；
   - 下一个描述符的DAV位为'1'。那么RxDMA将DAV位清零以关闭当前描述符，之后操

作退回步骤4；

7. 如果使能了IEEE 1588时间戳功能，在接收帧完成后DMA控制器会把获取的时间戳的低位和高位（如果接收帧符合需要记录时间戳的帧的条件），分别写入当前描述符的接收描述符2和接收描述符3（当DFM=0时），或者写入当前描述符的接收描述符6和接收描述符7（当DFM=1时）。同时DMA把从MAC处返回的接收状态信息写入接收描述符0，并把DAV位清'0'，把LSG位置'1'；
8. 如果新获取的描述符DAV位为'1'，则RxDMA控制器操作跳动步骤4。如果DAV位为'0'，则RxDMA控制器进入挂起状态，并设ENET_DMA_STAT寄存器的RBU位为1。如果使能了接收帧清空功能，则在DMA进入挂起状态之前，控制器会清空接收帧；
9. 在挂起状态下，有两种方法退出该状态。一种方法是向接送查询使能寄存器ENET_DMA_RPEN中写入任意值。另一种方法是RxFIFO收到下一帧数据，这意味着在直通模式下时，帧数据字节数需要大于设置的阈值，或者在存储转发模式下，需要收到整个帧。当DMA退出暂停状态后，DMA会获取下一个描述符，并跳到步骤2。

获取接收描述符

只要满足下列条件任意一个或多个，DMA就会尝试获取接收描述符：

- 在寄存器ENET_DMA_CTL的接收开始/停止位SRE从'0'变为'1'，使DMA控制器进入运行状态的时候；
- 当前描述符的整个缓存大小（对于链结构为缓存1，对于环结构为缓存1和2）不足以接收整个帧，也就是说在接收到帧的结尾之前，当前描述符的缓存已满；
- 在一个完整的帧传送到接收缓存之后，并在当前描述符关闭之前；
- 在挂起状态时，MAC接收到新的帧；
- 向接送查询使能寄存器ENET_DMA_RPEN写入任意值。

挂起状态时接收到新的帧时的处理

在挂起状态时，当接收到一个新的帧，并且满足转发条件时（转发条件如上所述），RxDMA将获取帧的描述符。如果接收描述符0的DAV位为'1'，则RxDMA控制器退出挂起状态，返回运行状态开始接收帧。但当接收描述符0的DAV位为'0'，则应用程序可以通过配置ENET_DMA_CTL寄存器中DAFRF位来选择是否清空RxFIFO中的帧。如果DAFRF=0，则RxDMA控制器将丢弃FIFO所接收的帧数据，并将丢失帧计数器MSFC加1。若DAFRF=1，则可以阻止丢弃RxFIFO顶部的帧，除非RxFIFO满，丢失帧计数器MSFC的值不会增加。在DAV位为'0'时，ENET_DMA_STAT寄存器中的RBU位将被置位，RxDMA控制器仍处于挂起状态。

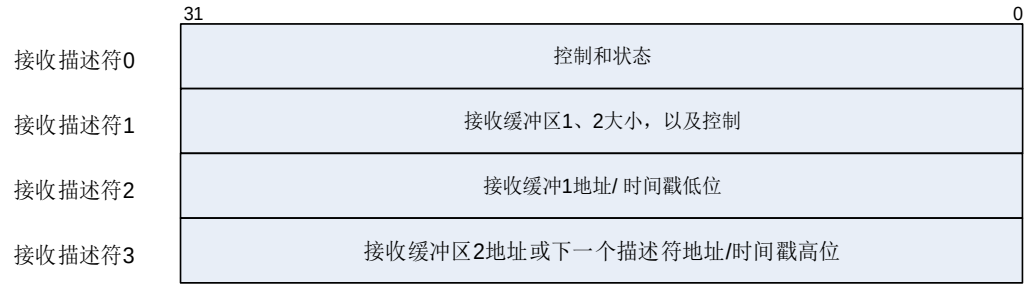
带 IEEE1588 时间戳的 RxDMA 描述符格式

如果使能了IEEE 1588功能，则MAC控制器会在带时间戳的帧接收完成之后，DMA关闭描述符之前（DAV位清'0'），将时间戳写入接收描述符2和接收描述符3(DFM=0)，或接收描述符6和接收描述符7(DFM=1)。

常规 RxDMA 描述符

常规RxDMA描述符结构体包含4个32位字，接收描述符0~接收描述符3。接收描述符0~接收描述符3的位定义如下：

图 43-9. 常规接收描述符



■ 接收描述符0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DAV	DAFF	FRML[13:0]													
rw	rw	rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRS	DERR	SAFF	LERR	OERR	VTAG	FDES	LDES	IPHERR/ TSV	LCO	FRMT	RWDT	RERR	DBERR	CERR	PCERR/ EXSV
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	DAV	描述符转发条件位 该位指示了DMA控制器是否可以占有该描述符。DMA会在帧完整接收或者描述符指向的缓存被填满以后把该位清’0’。 0：表示CPU占有描述符 1：表示DMA占有描述符
30	DAFF	未通过目标地址过滤器位 0：接收帧通过目标地址过滤器 1：接收帧没有通过目标地址过滤器
29:16	FRML	帧长位 这些位表示了传送到输入缓存的接受帧字节长度（如果输入帧不是类型帧则包括CRC部分，如果输入帧是类型帧，则是否包括CRC部分由ENET_MAC_CFG寄存器的TFCD位决定）。只有LDES位为1且DERR位为0时，此位才有效。若LDES位和错误汇总位ERRS位都为’0’，则这些位表示当前接收帧已经传送到内存里的累计字节数。 **注意：**帧长度值为0意味着由于某种原因（例如FIFO溢出或在接收过程中动态地修改了过滤器的值，导致未通过过滤器的情况等）造成写入FIFO的帧不完整。
15	ERRS	错误汇总位 该位只在LDES位为’1’时有效。 当DFM = 0时，该位为下列位的逻辑“或”： DERR：描述符错误 OERR：溢出错误

LCO：延迟冲突
RWDT：看门狗超时
RERR：接收错误
CERR：CRC错误
IPHERR = 0, FRMT = 1且PCERR = 1：数据校验和错误
IPHERR = 1, FRMT = 1且PCERR = 0：报头校验和错误
IPHERR = 1, FRMT = 1且PCERR = 1：报头及数据校验和错误
当DFM = 1时，该位为下列位的逻辑“或”：
IPPLDERR：IP帧数据错误
IPHERR：IP帧报头错误
DERR：描述符错误
OERR：溢出错误
LCO：延迟冲突
RWDT：看门狗超时
RERR：接收错误
CERR：CRC错误

14	DERR	描述符错误位 只有在LDES位置位时，此位才有效。 当当前描述符的缓存大小小于接收帧帧长同时DMA又无法占有下一个描述符，将发生描述符错误。 0：未发生描述符错误 1：发生了描述符错误
13	SAFF	未通过源地址过滤器位 0：未发生未通过源地址过滤器事件 1：帧没有通过源地址过滤器
12	LERR	长度错误位 只有在 FRMT 位为’0’时，此位才有效。 该位指示了接收到的以太网帧头长度/类型域的值是否与接收帧的实际长度匹配。 0：未发生长度错误 1：发生了长度错误
11	OERR	溢出错误位 当RxFIFO发生了溢出，而接收帧已有部分被传送到输入缓存时，溢出错误位置位。 0：未发生溢出错误 1：发生了RxFIFO溢出，帧数据无效
10	VTAG	VLAN标签位 0：接收帧非VLAN帧 1：当前的接收帧为VLAN帧
9	FDES	第一个描述符 该位表示当前描述符是否存放了接受帧的第一部分。 0：当前描述符未存放帧的第一部分 1：当前描述符存放了帧的第一部分

8	LDES	最后一个描述符 该位表示当前描述符是否存放了接受帧最后一部分。 0：当前描述符未存放帧的最后一部分 1：当前描述符存放了帧的最后一部分
7	IPHERR/TSV	IP帧报头错误/时间戳有效位 当DFM=0时，则将位7结合位5和位0，查询错误状态表，判断发生了什么错误。 当DFM=1时，则表示已获取时间戳并将时间戳值写到了接收描述符6和接收描述符中，该位仅在LDES位置位时有效。
6	LCO	延迟冲突位 延迟冲突表示在接收到64字节数据后是否发生了冲突。 该位仅在半双工模式下有效 。 0：未发生延迟冲突 1：接收帧的过程发生了延迟冲突
5	FRMT	帧类型 当DFM=0时，则将位7结合位5和位0，查询错误状态表，判断发生了什么错误。 当DFM=1时，则表示接受帧为以太网类型帧或标签帧。 当接收帧为过短帧时，该位无效。 0：接收到的帧是IEEE802.3帧 1：接收到的帧是以太网类型的帧（以太网帧头长度/类型域的值大于等于0x0600，或是一个标签帧）。
4	RWDT	接收看门狗超时 当WDD=0，该位表示已接收到超过2048字节的帧数据。 当WDD=1，该位表示已接收到超过16384字节的帧数据。 0：未发生接收看门狗超时 1：在接收帧的过程中发生了看门狗超时，当前接收帧将被截断。
3	RERR	接收错误 该位表示帧接收过程中，在RX_DV信号有效时是否收到有效接口信号RX_ER。 0：未发生接收错误 1：发生了接收错误
2	DBERR	Dribble位错误 该位表示接收到的数据帧中存在半字节，只有 MII 模式时该位有效。 0：未发生Dribble位错误 1：发生了Dribble位错误
1	CERR	CRC错误 该位表示接受帧的帧校验序列是否与硬件计算结果不匹配，只有在 LDES 位为’1’时该位有效。 0：未发生CRC错误 1：检测到接收帧发生了CRC错误
0	PCERR/EXSV	数据校验和错误/扩展状态有效位 当DFM=0时，则将位7结合位5和位0，查询错误状态表，判断发生了什么错误。

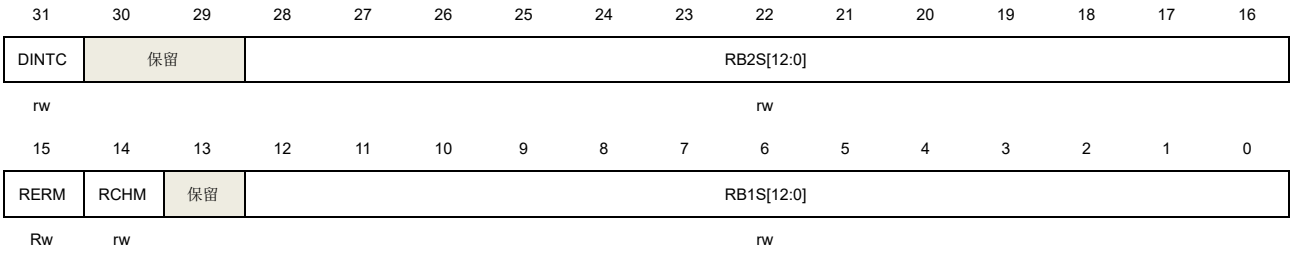
当DFM=1时，该位表示接收描述符4有效。
该位仅在LDES位置位时有效。
0：接收描述符4无效
1：接收描述符4有效

*表43-7. 接收描述符0错误状态描述，仅适用于常规描述符（DFM=0）*显示了接收描述符0中位IPHERR、FRMT、PCERR取值的含义：

表 43-7. 接收描述符 0 错误状态描述，仅适用于常规描述符（DFM=0）

位7: IPHERR	位5: FRMT	位0: PCERR	帧状态
0	0	0	IEEE802.3类型帧（长度域值小于0x0600或非标签帧）
0	0	1	IPv4/IPv6类型帧，未检测到IP报头校验和错误；由于不支持的数据格式，未执行数据校验和检测
0	1	0	IPv4/IPv6类型帧，未检测到校验和错误
0	1	1	IPv4/IPv6类型帧，检测到数据校验和错误。 该错误可能原因： 1）计算的校验和值与接收帧中的校验和域值不匹配 2）接收帧长度域值与实际接收帧字节数不符
1	0	0	保留
1	0	1	类型帧（长度/类型域值大于等于0x600）或标签帧，但非IPv4/IPv6帧，不检测数据校验和
1	1	0	IPv4/IPv6类型帧，检测到报头校验和错误。 该错误可能原因： 1) 类型域值与数据包中的版本值不一致 2) 计算的报头校验和值与报头校验和域值不匹配 3）IP报头字节数不足
1	1	1	IPv4/IPv6类型帧，检测到IP报头校验和错误和数据校验和错误

■ 接收描述符1



位/位域	名称	描述
31	DINTC	关闭接收完成中断 0：接收完成后ENET_DMA_STAT寄存器的RS位（位6）会立即置1，此时若使能了相应中断则会触发中断。 1：在接收帧成功接收完成后，ENET_DMA_STAT寄存器的RS位（位6）不会立即置1，但会在可配置的延时（由ENET_DMA_RSWDC寄存器配置）之后置位。
30:29	保留	必须保持复位值。

28:16	RB2S[12:0]	接收缓存2大小 表示接收缓存2的大小（以字节为单位）。缓存大小必须被设为4的倍数。这些位在RCHM位为'1'时被忽略。
15	RERM	接收描述符环形结构结尾 该位表示到达的是描述符列表中的最后一个描述符，下个描述符自动返回列表的基地址。 0：当前描述符不是最后一个描述符 1：到达描述符列表的最后一个描述符
14	RCHM	第二地址链表 0：描述符里的第二个地址指向第二缓存的地址 1：描述符里的第二个地址是下一个描述符的地址，该位仅在在链模式下使用。忽略RB2S [12:0]的值。 **注意：**当RERM=1，则即使该位置位，下个描述符也将返回列表基地址。
13	保留	必须保持复位值。
12:0	RB1S[12:0]	接收缓存1大小 表示接收缓存1的大小（以字节为单位）。缓存大小必须被设为4的倍数。如果这些位为0，DMA忽略该缓存，直接取接收缓存2中的数据（RCHM=0）或者下一个描述符（RCHM=1）。

■ 接收描述符2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RB1AP/RTSL[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RB1AP/RTSL[15:0]															
rw															

位/位域	名称	描述
31:0	RBAP1/RTSL[31:0]	接收缓存1地址指针/发送帧时间戳低32位 这些位有2个功能：缓冲区1的地址指针（RB1AP）和时间戳低32位（RTSL）。 RB1AP：在DMA控制器获取该描述符之前，配置该位为缓存区1的地址。如果位RB1S不为'0'，则用该位的地址来存储接收的数据帧。对缓存的地址对齐不做限制。 RTSL：当使能了时间戳功能同时LDES位为'1'时，如果接收帧通过了地址过滤，并且置位了对应的帧类型使能位，则DMA会将时间戳低32位写入这些位。如果接收帧没有置位对应的帧类型使能位，则这些位会保持RB1AP的值。

■ 接收描述符3

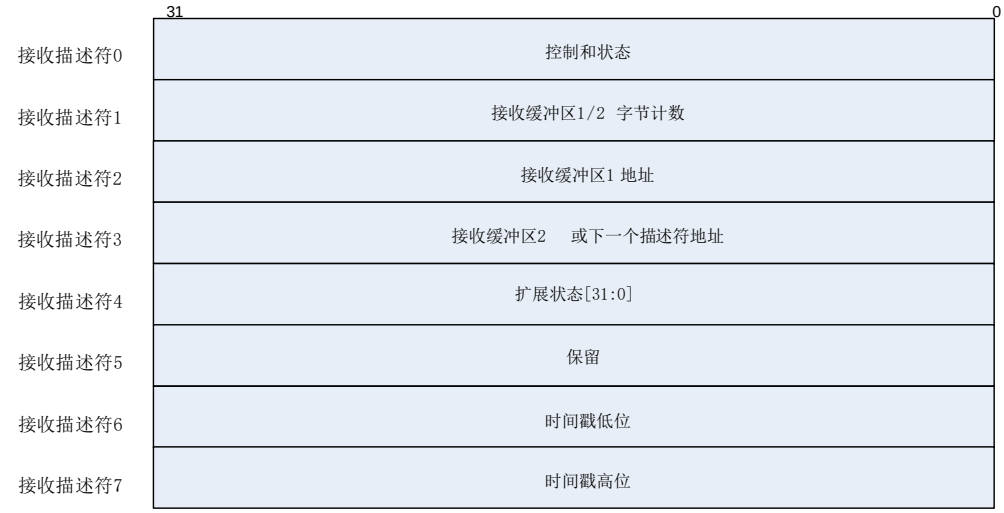
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RB2AP/RTSH[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RB2AP/RTSH[15:0]															
rw															

位/位域	名称	描述
31:0	RB2AP/RTSH[31:0]	接收缓存2地址指针（或者下一个描述符地址指针）/接收帧时间戳高32位 这些位有2个功能：数据接收时这些位存放缓冲区2的地址或下个描述符的地址（RB2AP），或帧时间戳高32位（RTSH）。 RB2AP：在DMA控制器获取该描述符之前，配置该位为缓存区2的地址（RCHM=0）或下个描述符地址（RCHM=1）。在RCHM=0时，如果位RB1S不为'0'，则用该位的地址来存储接收的数据帧。如果RCHM=1，且RERM=1，则忽略这些位。 如果该地址表示下个描述符地址，则地址需为字对齐的，没有其他的限制条件。 RTSH：当使能了时间戳功能同时帧的最后部分LDES位为'1'时，如果接收帧通过了地址过滤，并且置位了对应的帧类型使能位，则DMA会将时间戳高32位写入这些位。如果接收帧没有置位对应的帧类型使能位，则这些位会保持RB2AP的值。

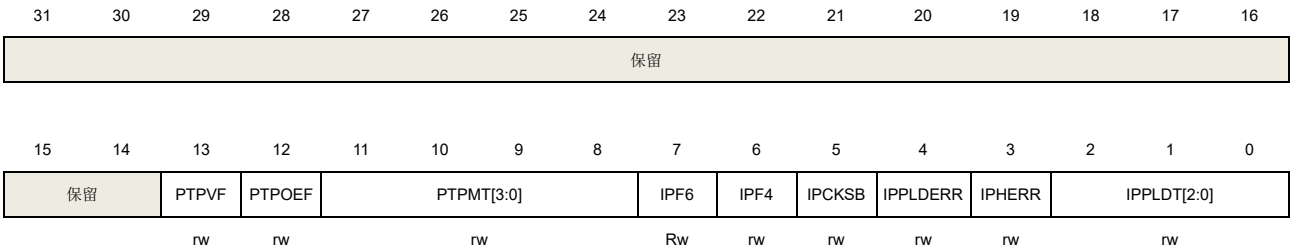
增强 RxDMA 描述符

增强RxDMA描述符结构体包含8个32位字：接收描述符0~接收描述符7。接收描述符0~接收描述符3的位定义与常规RxDMA描述符相同，接收描述符4~接收描述符7的位定义如下：

图 43-10. 增强接收描述符



■ 接收描述符4



位/位域	名称	描述
31:14	保留	必须保持复位值。
13	PTPVF	PTP 版本格式位 0：版本1格式

		1：版本2格式
12	PTPOEF	以太网PTP帧 0：当PTPMT不为0时，接收的PTP帧是IP-UDP帧。 1：接收的PTP帧是IEEE802.3以太网帧
11:8	PTPMT[3:0]	PTP消息类型 PTP消息类型解码为以下几种： 0x0：未收到PTP消息 0x1：SYNC 0x2：FOLLOW_UP 0x3：DELAY_REQ 0x4：DELAY_RESP 0x5：对于点对点透明时钟：PDELAY_REQ 对于普通时钟或边界时钟：ANNOUNCE 0x6：对于点对点透明时钟：PDELAY_RESP 对于普通时钟或边界时钟：MANAGEMENT 0x7：对于点对点透明时钟：PDELAY_RESP_FOLLOW_UP 对于普通时钟或边界时钟：SIGNALING
7	IPF6	IPv6帧位 0：接收帧不是IPv6帧 1：接收帧是IPv6帧
6	IPF4	IPv4帧位 0：接收帧不是IPv4帧 1：接收帧是IPv4帧
5	IPCKSB	绕过IP帧校验和 该位仅在接收帧为IPv6或IPv4帧时有效。 0：没有绕过接收帧校验和功能 1：绕过了接收帧校验和功能
4	IPPLDERR	IP帧数据错误位 在以下任意情形中，该位会被置位：1）硬件计算的校验和域TCP，UDP或ICMP帧校验和域值不匹配 2）IP报头数据长度域值与接收到的帧数据长度不符。 0：在接收帧中没有发生帧数据错误 1：在接收帧中发生了帧数据错误
3	IPHERR	IP报头错误 在以下任意情形中，该位会被置位：1）硬件计算的校验和值与IP报头校验和域值不匹配 2）以太网帧类型域值与IP数据包中版本域值不一致（例如类型域值为0x800，但版本域值不为0x4；类型域值为0x86dd，但版本域值不为0x6）。 0：没有发生IP报头错误 1：发生了IP报头错误
2:0	IPPLDT[2:0]	IP帧数据类型位 仅在IPFCO=1，IPHERR=0且LDES=1时，这些位有效。

0x0：不支持的数据类型或忽略的IP数据类型

0x1：数据类型为UDP

0x2：数据类型为TCP

0x3：数据类型为ICMP

0x4~0x7：保留

- 接收描述符5

  所有位保留。

- 接收描述符6

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

位/位域	名称	描述
31:0	RTSL[31:0]	接收时间戳低 32 位 当使能了时间戳功能同时帧的最后部分 LDES 位为'1'时，如果接收帧通过了地址过滤，并且置位了对应的帧类型使能位，则 DMA 会将时间戳低 32 位写入这些位。

- 接收描述符7

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

位/位域	名称	描述
31:0	RTSH[31:0]	接收时间戳高 32 位 当使能了时间戳功能同时帧的最后部分 LDES 位为'1'时，如果接收帧通过了地址过滤，并且置位了对应的帧类型使能位，则 DMA 会将时间戳高 32 位写入这些位。

43.3.4. MAC 统计计数器：MSC

为了了解发送和接收帧的统计情况，利用一组计数器来收集相关的统计数据。这些MAC计数器被称为MAC统计计数器（MSC）。在章节"*ENET寄存器*"中有这些寄存器的详细功能说明。

当发送帧没有出现帧下溢、没有载波、载波丢失、顺延(Deferral)过多、延迟冲突，过度冲突和jabber超时等情况时，可以称为“好帧”，MSC发送计数器会自动更新。

- 当接收帧没有出现对齐错误、CRC错误、过短帧、长度错误、超出范围和MII_RX_ER引脚上的错误信号有效等情况时，可以称为“好帧”，MSC接收计数器会自动更新。其中，CRC错误是指CRC计算结果与帧校验序列值不一致，过短帧表示帧长少于64字节，长度错误表示长度域值与实际接收到的字节数不符，超出范围表示长度域值超过IEEE 802.3所规定的最大值，即对于非标签帧最大值为1518字节，对于VLAN标签帧最大值为1522字节。

注意：当被丢弃的帧是长度小于6字节的过短帧（没有完整接收到目标地址）时，MSC接收计数器也会更新。

43.3.5. 唤醒管理：WUM

以太网模块支持两种将系统从深度睡眠模式唤醒的方法。分别为远程唤醒帧和Magic Packet唤醒帧。为了减小功耗，可以使主机系统和以太网模块进入低功耗状态，从而可以停止由HCLK驱动的电路以及发送时钟。但由接收时钟驱动的电路将继续工作，以监听唤醒帧。如果将ENET_MAC_WUM寄存器的PWD位置1，则以太网模块进入低功耗状态。在低功耗状态下，MAC会丢弃所有的帧，直到退出低功耗状态。此时可以采用上述的两种方式能够退出低功耗状态。将ENET_MAC_WUM寄存器的WFEN置1，以设置当收到远程唤醒帧时唤醒以太网模块，或将ENET_MAC_WUM寄存器的MPEN置1，以设置当收到Magic Packet唤醒帧时唤醒以太网模块。当任一个唤醒功能被使能，一旦MAC接收到相应的唤醒帧，以太网模块将产生一个唤醒中断，并退出低功耗状态。

远程唤醒帧检测

将ENET_MAC_WUM寄存器的WFEN置1可以使能远程唤醒检测。当MAC处于低功耗状态，且远程唤醒使能位为’1’时，MAC会进行唤醒帧过滤。如果输入帧通过了过滤器命令的地址过滤，而且过滤器CRC-16与被检查的输入帧匹配，则认为接收到唤醒帧，随后MAC即恢复正常工作。即便唤醒帧的长度超过了512字节，只要该帧有正确的CRC值，它仍然被认为是有效的。在接收到远程唤醒帧时还会将ENET_MAC_WUM寄存器的WUFR位置1。如果远程唤醒中断没有被屏蔽，那么此时还将产生一个WUM中断。

Magic Packet 检测

另一种唤醒方法是检测Magic Packet唤醒帧（见AMD公司的“Magic Packet技术”）。一个Magic Packet帧是一种特殊构成的数据包，专门用于唤醒。这种包可以被以太网模块接收、分析和识别，并用于触发一个唤醒事件。设置ENET_MAC_WUM寄存器的MPEN位为’1’可以使能此功能。这种类型的帧格式如下：目的和源地址域之后的任何位置连续6字节全1（0xFFFF FFFF FFFF），接着是在没有任何中断和暂停的情况下有16个重复的MAC地址；如果这16次重复间有任何的间断，则需要重新在输入帧里检测0xFFFF FFFF FFFF。WUM模块会持续监视每一个发向本节点的帧，那些通过地址过滤的Magic Packet帧，MAC会进一步检测其是否符合Magic Packet的格式，一旦通过检测将会使MAC从低功耗状态下唤醒。设备也接受多播帧作为Magic Packet帧。

下面是一个站地址为0xAABB CCDD EEFF的Magic Packet帧实例（MISC表示包内各种附加的数据字节）：

<DESTINATION><SOURCE><MISC>

……FF FF FF FF FF FF

AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF

AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF

AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF

AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF

<MISC><FCS>

一旦检测到Magic Packet帧，ENET_MAC_WUM寄存器的位MPKR会被置1。如果使能了Magic Packet中断，此时还将产生对应中断。

系统在低功耗期间注意事项

在MCU处于深度睡眠模式时，若使能外部中断线22（ENET1）/23（ENET0），则以太网的WUM模块仍能够检测帧。 由于MAC在低功耗状态也需要进行Magic Packet/远程唤醒帧检测，因此ENET_MAC_CFG寄存器的REN位必须保持为'1'。在低功耗状态时需要把ENET_MAC_CFG寄存器的TEN位清'0'来关闭发送功能。此外，由于不需要把Magic Packet/远程唤醒帧转发给应用，因此在低功耗状态时也要关闭以太网DMA模块，可以通过设置ENET_DMA_CTL寄存器的STE位和SRE位（分别对应TxDMA和RxDMA）为'0'来关闭以太网DMA。

推荐的进入低功耗状态和唤醒步骤如下：

1. 等待当前帧发送完毕，然后将ENET_DMA_CTL寄存器的STE位复位来关闭TxDMA；
2. 把ENET_MAC_CFG寄存器的TEN位和REN位清'0'，来关闭MAC发射器和MAC接收器；
3. 观察ENET_DMA_STAT寄存器位RS，等待RxDMA把RxFIFO里的所有帧读出，再关闭RxDMA；
4. 配置并使能外部中断线19，使其能产生事件或者中断。如果配置了外部中断线19产生中断，则还需要编写中断处理程序ENET_WKUP_IRQ，在其中清除外部中断线19的中断标志位；
5. 设置ENET_MAC_WUM寄存器的MPEN或WFEN位（或两位）为'1'，使能Magic Packet/远程唤醒帧检测（或两种功能）；
6. 设置ENET_MAC_WUM寄存器的PWD位为'1'，使能低功耗模式；
7. 设置ENET_MAC_CFG寄存器的REN位为'1'，打开MAC接收器；
8. 设置使MCU进入深度睡眠模式；
9. 在接收到有效的唤醒帧后，以太网模块退出低功耗状态；
10. 读取ENET_MAC_WUM寄存器来清除电源管理事件标志位，打开MAC发送器，以及TxDMA和RxDMA；
11. 设置系统时钟：使能HXTAL并配置RCU时钟参数。

远程唤醒帧过滤器寄存器

唤醒帧过滤器寄存器一共有8个，但这些寄存器共用一个相同的偏移地址。在完成对某个过滤器寄存器的读或写的时候，内部的指针会自动指到下一个过滤器寄存器。不论是读还是写操作，强烈建议连续8次的操作。也就是说，对其设置时需要将设置的值分为8次逐一写入唤醒帧过滤

器寄存器地址，读取的时候也是需要连续读8次唤醒帧过滤器寄存器，才能将所有值读出。

图 43-11. 唤醒帧过滤器寄存器

唤醒帧过滤器寄存器0	过滤器0字节屏蔽							
唤醒帧过滤器寄存器1	过滤器1字节屏蔽							
唤醒帧过滤器寄存器2	过滤器2字节屏蔽							
唤醒帧过滤器寄存器3	过滤器3字节屏蔽							
唤醒帧过滤器寄存器4	保留	过滤器3命令	保留	过滤器2命令	保留	过滤器1命令	保留	过滤器0命令
唤醒帧过滤器寄存器5	过滤器3偏移		过滤器2偏移		过滤器1偏移		过滤器0偏移	
唤醒帧过滤器寄存器6	过滤器1 CRC-16				过滤器0 CRC-16			
唤醒帧过滤器寄存器7	过滤器3 CRC-16				过滤器2 CRC-16			

- 过滤器n字节屏蔽

该寄存器定义了过滤器n（n=0，1，2，3）使用帧的哪些字节来检查判断是否为唤醒帧。其第31位必须为'0',位[30:0]是字节屏蔽位。如果过滤器n（n=0，1，2，3）的第m位（m=0~30）为'1'，则唤醒帧检测的CRC模块会处理输入帧的第[过滤器n偏移+m]字节，否则忽略之。

- 过滤器n命令

共4位控制过滤器n的工作模式。最高位位3为地址类型选择，如果该位为'1'，则只检测多播帧；如果该位为'0'，则只检测单播帧。位2和位1必须保持为0。位0是过滤器n的使能位，置位时使能过滤器n，反之禁能过滤器n。

- 过滤器n偏移

与过滤器n字节屏蔽配合使用。该寄存器定义了过滤器n要检查的首字节在帧内的偏移量。最小允许取值是12，代表了帧的第13个字节（偏移值为0表示帧的第1个字节）。

- 过滤器n CRC-16

该寄存器包含了预先写入的CRC-16码，用于与帧数据计算的CRC-16值进行比较。

43.3.6. 精确时间协议：PTP

协议的大部分是通过UDP层之上的应用程序软件实现的。MAC的PTP模块主要是支持记录PTP包从以太网端口发出和收到的准确时间，并将其返回给应用程序。

关于精确时间协议(PTP)的具体内容可参见IEEE 1588™相关文档。

基准时钟源

IEEE 1588协议规定，通过一个64位寄存器来获得系统基准时间，其中高32位提供秒级的时间信息，低32位提供亚秒级的时间信息。

PTP基准时钟输入用来生成系统基准时间（也称为系统时间），以及获取PTP帧的时间戳值。其

频率必须大于或等于时间戳计数器的分辨率。主节点和从节点之间的时间同步精度在0.1us左右。

同步精度

时间同步的精度取决于以下几个因素：

- PTP基准时钟输入的频率；
- 所用晶体振荡器的特性（频漂）；
- 同步流程的执行频度。

系统时间校准

64位PTP系统时间由PTP输入基准时钟来更新。这个PTP系统时间用来作为记录发送/接收时间戳的依据。该系统时间的初始化和校准支持两种模式：粗调和精调。校准的目的是纠正频率偏移。

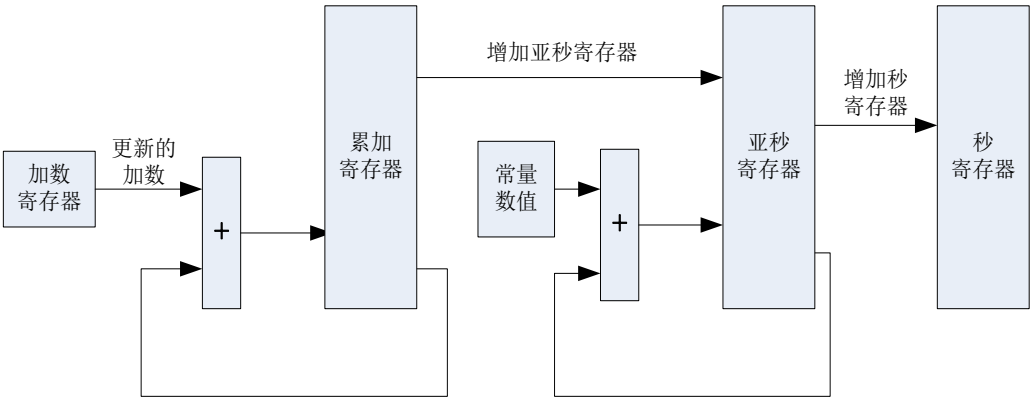
若选择了粗调的方式，则可通过配置PTP时间戳更新寄存器(ENET_PTP_TSUH和ENET_PTP_TSUL)，来进行系统时间初始化和校准。如果TMSSTI位被置位，则PTP时间戳更新寄存器被用于初始化。如果TMSSTU位被置位，则PTP时间戳更新寄存器被用于系统时间的调整，加上或者减去这个寄存器值进行校准。

若选择了精调的方式，则需要一段时间才能完成。由应用程序确定精调的频率，以确保从时钟能线性地同步于主时钟，避免不可预知的大的抖动。

这种方法是指，在每个HCLK周期把加数寄存器ENET_PTP_TSADDEND中的值加入累加器。当累加器溢出时会产生脉冲令时间戳低寄存器ENET_PTP_TSL的值增加。增加的值由亚秒递增寄存器ENET_PTP_SSINC中的值决定。

*图43-12. 系统时钟精细校准方法*演示了精调算法的流程：

图 43-12. 系统时钟精细校准方法



下面是一个具体例子用于说明精调方式如何更新系统时间：

假设系统时钟更新电路的精度需要达到25ns，即更新的频率为40MHz。假设基准时钟HCLK是72MHz，计算频率比得72 / 40 = 1.8。写入ENET_PTP_TSADDEND寄存器的值应当是2^{32} / 1.8，等于0x8E38 E38E。如果基准时钟频率漂低，假设降至68MHz，此时频率比变成68/40 = 1.7，

写入ENET_PTP_TSADDEND寄存器的值应当是2^{32} / 1.7 = 0x9696 9697。如果基准时钟漂高，假设升到76MHz，写入到ENET_PTP_TSADDEND寄存器的值应当是2^{32} / 1.9 = 0x86BC A1AF。。初始时，将加数寄存器设为从时钟频率Clock Addend Value(0)，该值按上述进行计算。除了配置加数计数器之外，还需对亚秒递增寄存器进行设置才能保证达到20ns的精度。每次累加寄存器溢出后，该寄存器的值会对时间戳低寄存器进行更新。因为ENET_PTP_TSL寄存器中的STMSS[30:0]位表示系统时间的亚秒值，其精度为10^9ns / 2^{31} = 0.46ns。所以为了使系统时间精度达到25ns，亚秒递增寄存器的值应该设为25 / 0.46 = 0d54。

注意：下文描述的算法是以主从设备之间传输的时延Master-to-Slave-Delay恒定为基础的，通过该算法在若干个Sync周期内确定同步频率比。

算法如下：

- 定义主设备发送一个SYNC消息到从设备时的时间：MSYNCT(n)
  定义从设备的本地时间：SLOCALT（n）
  定义主设备的本地时间：MLOCALT（n）
  计算：MLOCALT（n）= MSYNCT（n）+ Master-to-Slave-Delay（n）

- 定义发送两次SYNC消息之间的主设备时钟计数：MCLOCKC（n）
  计算：MCLOCKC（n）= MLOCALT（n）– MLOCALT（n-1）
  定义接收两次SYNC消息之间的从设备时钟计数：SCLOCKC（n）
  计算：SCLOCKC（n）= SLOCALT（n）- SLOCALT（n-1）

- 定义两个计算之间的差值：DIFFCC（n）
  计算：DIFFCC（n）= MCLOCKC（n）- SCLOCKC（n）

- 定义从时钟频率调整系数：SCFAF（n）
  计算：SCFAF（n）=（MCLOCKC（n）+ DIFFCC（n））/ SCLOCKC（n）

- 定义加数寄存器的时钟加数值：Clock Addend Value（n）
  Clock Addend Value（n）= SCFAF（n）* Clock Addend Value（n-1）

注意：实际操作中，可能需要多个SYNC消息来完成主从设备的同步。

系统时间初始化流程

设置ENET_PTP_TSCTL寄存器的位TMSEN为'1'，可以使能时间戳功能。不过在把该位置'1'以后，必须首先初始化时间戳计数器来开始时间戳操作。初始化步骤如下：

1. 置位ENET_MAC_INTMSK寄存器的TMSTIM位，以屏蔽时间戳触发中断；
2. 置位ENET_PTP_TSCTL寄存器TMSEN位，以使能时间戳；
3. 根据期望时钟精度配置亚秒递增寄存器；
4. 若希望采用精调校准方式，则配置时间戳加数寄存器，并置位ENET_PTP_TSCTL寄存器的TMSARU位。若希望采用粗调校准方式，则忽略第4-6步，直接跳至第7步；
5. 轮询ENET_PTP_TSCTL寄存器的位TMSARU，直到其变为'0'；
6. 将ENET_PTP_TSCTL寄存器的TMSFCU位置位，来选择使用精调校准方式；
7. 把希望设置的系统时间值写入时间戳更新高寄存器和时间戳更新低寄存器；
8. 置位ENET_PTP_TSCTL寄存器的TMSSTI位，以初始化时间戳；
9. 一旦初始化成功后，时间戳计数器就开始工作。

系统时间更新步骤

粗调方式

1. 在时间戳更新高寄存器和时间戳更新低寄存器中写入偏移值（可以是负值）；
2. 置位ENET_PTP_TSCTL寄存器的TMSSTU位，以更新时间戳寄存器；
3. 轮询TMSSTU位，直到其被清'0'后完成。

精调方式

1. 利用前述"*系统时间校准*"介绍的算法，计算出期望的系统时钟频率所对应的加数寄存器的值；
2. 将值写入加数计数器，并设置ENET_PTP_TSCTL寄存器的TMSARU位为'1'将该值更新到PTP模块；
3. 把要求的期望时间写入期望时间高和期望时间低寄存器，并设置ENET_MAC_INTMSK寄存器的TMSTIM位为'0'来允许时间戳中断；
4. 设置ENET_PTP_TSCTL寄存器的TMSITEN位为'1'使能时间戳中断；
5. 在这个事件产生中断时，读出ENET_MAC_INTF寄存器的值以清除相应的中断标志位；
6. 重新用旧值编写时间戳加数寄存器，并设置ENET_PTP_TSCTL寄存器的TMSARU位为'1'将值更新到PTP模块。

带 PTP 功能的帧的发送与接收

在使能了IEEE 1588（PTP）时间戳功能后，在发送帧的帧首界定码从MAC输出或者MAC接收到帧的帧首界定码的时候，时间戳值被记录。每一个等待发送的帧在DMA发送描述符中都有一个标志，指示是否需要记录这个帧的时间戳，这与发送的帧是否为PTP帧无关。若ENET_PTP_TSCTL寄存器的位ARFSEN为'1'，则所有接收到的帧的时间戳都将被记录。如果ARFSEN位为'0'，则通过地址过滤的接收帧需要与ENET_PTP_TSCTL寄存器的配置进行匹配。换句话说，只有与PTP配置相匹配的帧才标记为一个PTP帧，并且同时记录时间戳值到描述符中。鉴别接收帧是否为PTP帧，接收帧中的PTP版本需要与PFSV位相符，并且对应的帧类型使能位（寄存器ENET_PTP_TSCTL中的位13到位11）要置位。特殊地，对于非IP数据包的PTP帧（普通802.3以太网帧的PTP），以太网帧的目标地址域需要为特殊的MAC地址（比如对于PDELAY_REQ/PDELAY_RESP/PDELAY_RESP_FOLLOW_UP消息类型，目标地址应为0x0E00 00C2 8001，对于其他的消息类型，目标地址应为0x0000 0019 1B01，详细信息请参考IEEE1588-2008规范）。同时，为了增强灵活性，如果MAFEN位置位，除了上述两个特殊地址以外，MAC1-3寄存器中SAF位为复位状态的地址也可以认为是特殊地址。

记录下来的时间戳会和帧的发送/接收状态信息一起，存放在相应的发送/接收描述符里。64位的发送帧时间戳写入DMA发送描述符，64位的接受帧时间戳写入DMA接收描述符。具体描述见后续的"*带IEEE 1588时间戳的TxDMA描述符格式*"和"*带IEEE1588时间戳的RxDMA描述符格式*"。

内部连接触发

MAC可以在系统时间大于等于期望时间的时候提供触发中断。使用中断会引入一段已知的中断时延再加上不确定的命令执行时间。为了计算这部分已知的中断时延时间，在系统时间大于期

望值的时候，PTP会将一个输出信号置高。将SYSCFG_TIMERxCFG0（x = 1/2/30/31）寄存器中的TSCFG5[4:0]、TSCFG4[4:0]或TSCFG3[4:0]位域设置为5'b01001，可将ENET0的输出信号内部连接到TIMERx（x = 1/2/30/31）的事件模式配置、暂停模式配置或复位模式配置的ITI4输入上。同样的，将SYSCFG_TIMERxCFG1（x = 1/2/30/31）寄存器中的TSCFG7[4:0]或TSCFG6[4:0]位域设置为5'b01001，可将ENET1的输出信号内部连接到TIMERx（x = 1/2/30/31）的复位+事件模式配置或外部时钟模式0模式配置的ITI4输入上。利用这个信号，由于TIMER1/2/30/31的时钟与PTP基准时钟（HCLK）是同步的，因此不再有任何不确定的误差。

PPS 输出信号

将ETH0_PPS_OUT映射到AF11功能上或者ETH1_PPS_OUT映射到AF6功能上分别可以使能ENET0和ENET1的PPS输出功能。该功能可以输出脉冲宽度为默认125ms（其他宽度详见***PTP PPS控制寄存器（ENET_PTP_PPSCTL）***）的脉冲，用于检查网络全部节点之间的同步。为了测试本地从时钟和主时钟之间的差别，可以把主从设备的PPS（秒脉冲）输出都连接到示波器，以测量2个时钟之间的差别。

43.3.7. 典型的以太网配置流程示例

在上电复位或系统复位之后，应用程序可按以下的典型操作流程来配置并启动以太网模块：

- 使能以太网时钟：
  配置RCU模块来使能HCLK时钟和以太网发送/接收时钟。
- 配置通讯接口：
  配置SYSCFG_PMCFG，选择接口模式（MII或RMII）；
  配置GPIO模块，将相应的功能脚映射到复用功能上。
- 等待复位完成：
  轮询 ENET_DMA_BCTL 寄存器直到 SWR 位复位（SWR 位在上电复位后或系统复位后默认置位）。
- 获取并配置PHY寄存器参数：
  根据 HCLK 频率，配置 SMI 时钟频率，并访问 PHY 寄存器获取 PHY 的信息（例如是否支持半/全双工，是否支持 10M/100Mbit 速度等等）。根据外部 PHY 支持的模式，配置 ENET_MAC_CFG 寄存器使与 PHY 寄存器信息一致。
- 初始化以太网DMA模块用于数据传输：
  配置 ENET_DMA_BCTL，ENET_DMA_RDTADDR，ENET_DMA_TDTADDR 和 ENET_DMA_CTL 寄存器，完成 DMA 模块初始化（详细信息请参考 ***DMA 控制器描述***章节）。
- 初始化用于存放描述符列表以及数据缓存的物理内存空间：
  根据 ENET_DMA_RDTADDR 和 ENET_DMA_TDTADDR 寄存器中的地址，初始化发送和接收描述符(DAV=1)，以及数据缓存。
- 使能MAC和DMA模块，开始发送和接收：
  置位 ENET_MAC_CFG 寄存器中的 TEN 和 REN 位，开启 MAC 发送器和接收器。置位 ENET_DMA_CTL 寄存器中的 STE 位和 SRE 位，使能 DMA 的发送和接收。
- 如果有帧要发送：
  1. 选择一个或多个描述符发送描述符，将发送帧数据写到发送描述符中指定的缓存地址中；

2. 将这些发送帧描述符中的 DAV 位置位；
3. 写入任意值到 ENET_DMA_TPEN 寄存器中，使 TxDMA 退出挂起模式，开始发送数据；
4. 有两种方法来确定当前帧是否发送完毕。第一种方法为轮询当前描述符的 DAV 位直到其复位；第二种方法仅适用于当 INTC 位为 1 的情况，应用程序可以轮询 ENET_DMA_STAT 寄存器的 TS 位直到其置位。

- 如果有帧要接收：
  1. 查看描述符列表中的第一个接收描述符（其地址在 ENET_DMA_RDTADDR 寄存器中配置）；
  2. 如果接收描述符 0 的 DAV 位复位，则说明描述符已被使用过，且接收缓存空间已存储了接收帧；
  3. 处理接收帧数据；
  4. 置位当前描述符的 DAV 位，以复用当前描述符接收新的帧；
  5. 查看列表中的下一个描述符，跳到步骤 2。

43.3.8. 以太网中断

以太网部分一共有2个中断向量，一个用于以太网正常操作，另一个用于映射到EXTI线19的以太网唤醒事件（检测唤醒帧或者Magic Packet）。

第一个中断向量用于由MAC和DMA产生的中断。关于***MAC中断***和***DMA中断***的详细介绍请看下文。

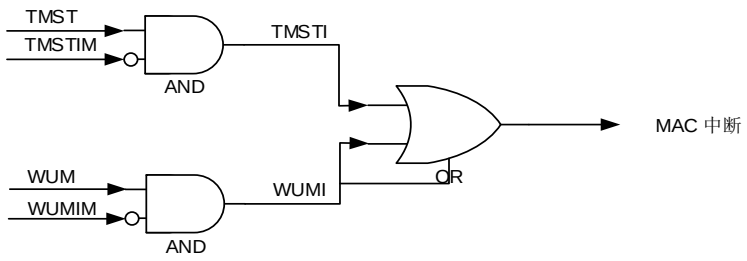
第二个中断向量用于WUM模块在唤醒事件时产生的中断，唤醒事件为远程唤醒帧接收事件或Magic Packet唤醒帧接收事件。唤醒事件映射到EXTI线19上，若使能了EXTI线19的上升沿中断，则唤醒事件可以使微控制器退出深度睡眠模式。此外，若使能了WUM中断，则EXTI线19中断和以太网中断都会被触发。

注意：由于WUM寄存器位于RX_CLK域，在应用程序读WUM寄存器后，到这些标志位被清除可能会有可观的延迟（延迟由HCLK和RX_CLK时钟频率之间差异决定）。为避免两次进入同一个中断，强烈建议应用程序在中断里等待唤醒帧接收标志位WUFR和Magic Packet接收标志位MPKR变为0后，再退出中断服务程序。

MAC 中断

MAC控制器有多个中断触发源。ENET_MAC_INTF寄存器描述了所有可产生的MAC中断类型，每个位都有对应的中断屏蔽位来防止某一事件引发中断。MAC中断信号为MAC所有中断的逻辑或。

图 43-13. MAC 控制器中断示意图



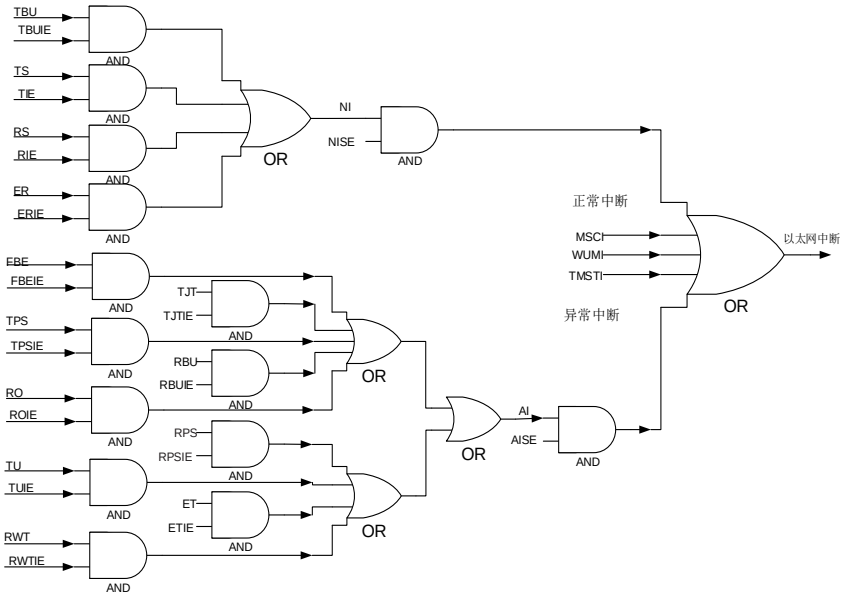
DMA 中断

DMA控制器有两种类型中断事件：正常类和异常类。

无论什么类型的中断事件，都具有相应的中断使能位（屏蔽位）来控制是否产生中断。当所有中断事件都被清除，或中断使能位被清除，则相应的中断汇总位也被清除。如果正常类和异常类中断都被清除，则DMA中断将被清除。

*图43-14. 以太网中断示意图*示意了以太网模块的中断连接：

图 43-14. 以太网中断示意图



43.4. ENET 寄存器

ENET0基地址：0x4002 8000

ENET1基地址：0x4002 A000

43.4.1. MAC 配置寄存器（ENET_MAC_CFG）

地址偏移：0x0000

复位值：0x0000 8000

MAC配置寄存器是MAC的工作模式寄存器。它定义了接收和发送的工作模式。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						TFCD	保留	WDD	JBD	保留		IGBS[2:0]			CSD
						rw		rw	rw				rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	SPD	ROD	LBM	DPM	IPFCO	RTD	保留	APCD	BOL[1:0]		DFC	TEN	REN	保留	
	rw	rw	rw	rw	rw	rw		rw	rw		rw	rw	rw		

位/位域	名称	描述
31:26	保留	必须保持复位值。
25	TFCD	类型帧CRC剥离位 0：帧的帧校验序列（最后4字节）在转发之前不会被剥离 1：帧的帧校验序列（最后4字节）在转发之前会被剥离 **注意：**该位仅在帧的LT域大于0x0600时有效。
24	保留	必须保持复位值。
23	WDD	关闭看门狗 该位表示已接收到了最大字节数的数据，超过的部分将被切断。 0：MAC允许接收小于或等于2048字节的帧 1：MAC关闭接收看门狗定时器，此时最多可接收16384字节的帧。
22	JBD	不检测Jabber 该位表示发送帧最大允许的发送字节数，超过的部分将被截断。 0：MAC允许的最大发送字节数为2048字节 1：MAC关闭发送Jabber定时器，此时最多可发送16384字节的帧。
21:20	保留	必须保持复位值。
19:17	IGBS[2:0]	帧间间隙选择位 这些位用于选择2个相邻发送帧之间的最短发送间隙。 0x0：96位时间 0x1：88位时间 0x2：80位时间 0x3：72位时间 0x4：64位时间 0x5：56位时间（半双工模式下不可用） 0x6：48位时间（半双工模式下不可用） 0x7：40位时间（半双工模式下不可用）

16	CSD	关闭载波侦听功能 0：MAC载波信号错误时会报错，并终止发送。 1：在半双工模式下，MAC在发送帧过程中忽略MII的CRS信号，发送过程中载波丢失或者没有载波都不会报错。
15	保留	必须保持复位值。
14	SPD	快速以太网速度 该位表示快速以太网模式下的速度： 0：10 Mbit/s 1：100 Mbit/s
13	ROD	关闭自接收功能 该位在全双工模式下可忽略 0：MAC在发送时接收所有来自PHY的数据包 1：MAC在半双工模式下不接受帧
12	LBM	回环模式 0：MAC在普通模式下工作 1：MAC在MII的回环模式下工作
11	DPM	双工模式 0：半双工模式使能 1：全双工模式使能
10	IPFCO	IP帧数据校验和 0：禁止接收端TCP/UDP/ICMP报头的校验和检验功能 1：使能接收端的帧数据校验和检测功能
9	RTD	不尝试重试 全双工模式下该位可被忽略 0：MAC会在发生冲突后按照BOL位的设定重发高达16次 1：帧仅发送一次
8	保留	必须保持复位值。
7	APCD	自动填充/CRC剥离 该位仅在非标签帧，且其长度域值小于等于1536时有效。 0：MAC会转发所有接收到的帧，而不改变帧的内容。 1：MAC会去除帧的填充字节和CRC域
6:5	BOL[1:0]	退后限制 在全双工模式下这些位可被忽略 在发生冲突后，MAC在重发当前帧之前需要延迟一段时间。这个延迟时间（dt）的时基单元称为时间间隙，一个时间间隙为512位时间。这个延迟时间（dt）是由下式计算得的随机整数值：$0 \le dt < 2^k$。 0x0：k = min (n, 10) 0x1：k = min (n, 8) 0x2：k = min (n, 4) 0x3：k = min (n, 1)

		其中n = 重发次数。
4	DFC	顺延检验 在全双工模式下这些位可被忽略 0：禁止MAC顺延检验功能。MAC会延迟发送直到CRS信号失效。 1：MAC顺延检验功能使能。如果延迟超过24288位时间，则会发生过度顺延错误，并且MAC将中止发送。但如果在顺延时间内检测到有效的CRS（载波侦听）信号，则会将顺延计数器重置为0，重新启动顺延计时。
3	TEN	使能发送器 0：MAC关闭发送状态机，若当前帧正在发送则在完成发送后关闭。 1：MAC使能发送状态机
2	REN	使能接收器 0：MAC关闭接收状态机，若当前帧正在接收则在接收完成后关闭。 1：MAC使能接收状态机
1:0	保留	必须保持复位值。

43.4.2. MAC 帧过滤器寄存器（ENET_MAC_FRMF）

地址偏移：0x0004
复位值：0x0000 0000

MAC帧过滤器寄存器包含了接收帧的过滤模式位。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					HPFLT	SAFLT	SAIFLT	PCFRM[1:0]		BFRMD	MFD	DAIFLT	HMF	HUF	PM
					rw	rw	rw	rw		rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	FAR	接收所有帧 该位控制帧过滤器功能。 0：只有通过了地址过滤器的接收帧才会被转发给应用程序 1：所有接收到的帧都会被转发给应用程序，但过滤的结果会反映在更新接收描述符状态信息的相应标志位。
30:11	保留	必须保持复位值。
10	HPFLT	HASH或者完美过滤 0：如果HMF位或者HUF位置'1'，符合HASH过滤器的帧才能通过接收地址过滤。 1：如果HMF位或者HUF位置'1'，接收帧通过HASH过滤器或者完美过滤器中任一种，就认为通过接收地址过滤。
9	SAFLT	源地址过滤器 除了目标地址过滤之外，使能源地址过滤器。

		过滤器将接收帧的源地址域值与使能的源地址寄存器中配置的值进行比较。如果源地址值相匹配，则接收描述符中的源地址匹配状态位将置位。 0：源地址过滤器关闭 1：源地址过滤器使能
8	SAIFLT	源地址过滤结果逆转 该位将源地址比较结果逆转。 0：仅在源地址过滤器结果逆转 1：使能源地址过滤器结果逆转，所有源地址符合源地址寄存器的帧会被标记为未通过源地址过滤。
7:6	PCFRM[1:0]	控制帧转发位 这些位用于设置所有控制帧的转发条件（包括单播和多播暂停帧）。 对于是否处理暂停控制帧，只取决于RFCEN位（ENET_MAC_FCTL[2]）的值。 0x0：MAC不转发任何控制帧给应用程序 0x1：MAC转发除了暂停帧以外的其他控制帧给应用程序 0x2：MAC转发所有的控制帧给应用程序，即使是没通过地址过滤器的控制帧。 0x3：MAC转发通过地址过滤器的控制帧给应用程序
5	BFRMD	不接收广播帧 0：过滤器接收所有广播帧 1：过滤器不接收所有广播帧
4	MFD	关闭多播过滤器 0：是否对多播帧进行过滤，取决于HMF位的取值。 1：所有的带多播目标地址的帧（帧的目标地址域中第一位为'1'，但不是所有位都为'1'）都能通过过滤器。
3	DAIFLT	目标地址过滤结果逆转 该位将目标地址过滤结果逆转。 0：禁用目标地址过滤结果逆转 1：使能目标地址过滤结果逆转
2	HMF	多播HASH过滤器 0：MAC会将接收到的多播帧的目标地址域和目标地址寄存器的设定值比较 1：MAC根据HASH列表对接收到的多播帧进行目标地址过滤
1	HUF	单播HASH过滤器 0：MAC会将接收到的单播帧目标地址域和目标地址寄存器的设定值比较 1：MAC根据HASH列表对接收到的单播帧进行目标地址过滤
0	PM	混杂模式 该位使地址过滤器无效，这意味着所有帧均可通过过滤器，同时接收描述符中状态信息的目标地址/源地址错误位总是为'0'。 0：禁用混杂模式 1：使能混杂模式

43.4.3. MAC hash 列表高寄存器（ENET_MAC_HLH）

地址偏移：0x0008

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:0	HLH[31:0]	HASH列表高位 这些位是HASH列表的高32位。

43.4.4. MAC hash 列表低寄存器（ENET_MAC_HLL）

地址偏移：0x000C

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:0	HLL[31:0]	HASH列表低位 这些位是HASH列表的低32位。

43.4.5. MAC PHY 控制寄存器（ENET_MAC_PHY_CTL）

地址偏移：0x0010

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PA[4:0]					PR[4:0]					保留	CLR[2:0]			PW	PB
rw					rw						rw			rw	rc_w1

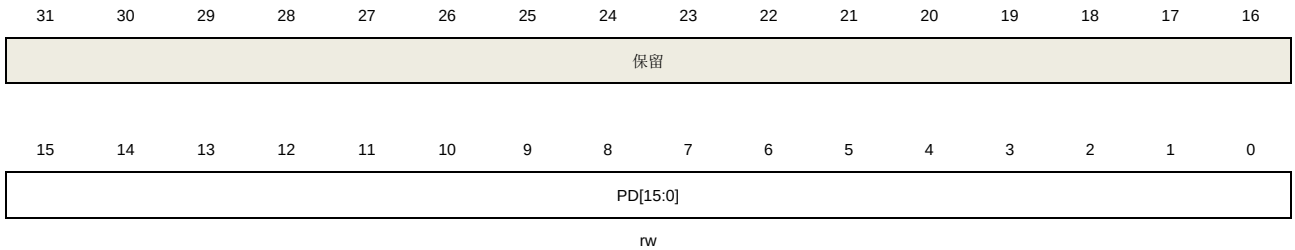
位/位域	名称	描述

31:16	保留	必须保持复位值。
15:11	PA[4:0]	PHY地址 这些位选择想要访问的PHY地址。
10:6	PR[4:0]	PHY寄存器 这些位选择想要访问的PHY寄存器。
5	保留	必须保持复位值。
4:2	CLR[2:0]	时钟范围 根据HCLK的频率来决定MDC的时钟分频系数。 0x0：HCLK/42（HCLK 范围：60-100 MHz） 0x1：HCLK/62（HCLK 范围：100-150 MHz） 0x2：HCLK/16（HCLK 范围：20-35 MHz） 0x3：HCLK/26（HCLK 范围：35-60 MHz） 0x4：HCLK/102（HCLK 范围：150-250 MHz） 0x5：HCLK/124（HCLK 范围：250-300 MHz） 0x6：HCLK/142（HCLK 范围：300-350 MHz） 其他：保留
1	PW	PHY写 该位指示了PHY的操作模式。 0：对PHY进行读操作 1：对PHY进行写操作
0	PB	PHY忙 该位指示了对PHY操作的状态。由应用程序置’1’后开始对PHY的进行读或者写操作，并需等到该位在操作完成后由硬件清’0’。在写ENET_MAC_PHY_CTL寄存器和读ENET_MAC_PHY_DATA寄存器之前，该位应当为’0’。

43.4.6. MAC PHY 数据寄存器（ENET_MAC_PHY_DATA）

地址偏移：0x0014
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	PD[15:0]	PHY数据位 对于读操作，这些位为从PHY中读取的数据。对于写操作，这些位为将要写到PHY中

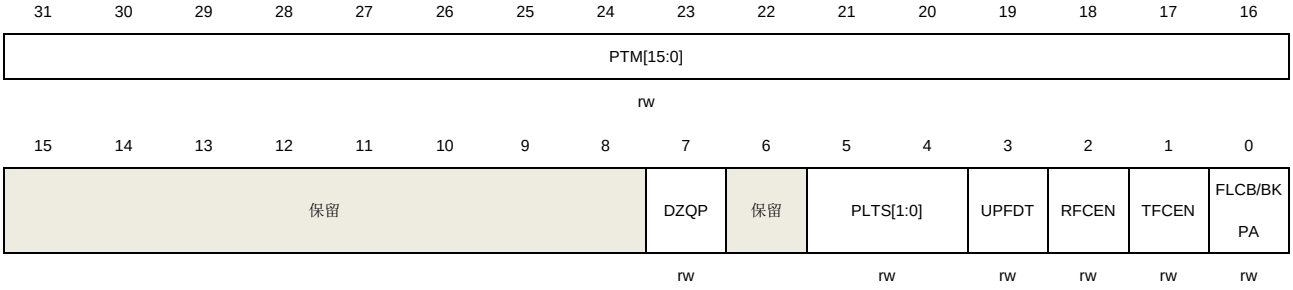
的数据。

43.4.7. MAC 流控寄存器（ENET_MAC_FCTL）

地址偏移：0x0018
复位值：0x0000 0000

该寄存器用于配置控制帧的生成和接收。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域	名称	描述
31:16	PTM[15:0]	暂停时间 这些位用来设置暂停控制帧时间域的值。
15:8	保留	必须保持复位值。
7	DZQP	关闭零时间片暂停功能 0：打开零时间片暂停控制帧自动生成功能 1：关闭零时间片暂停控制帧的自动生成
6	保留	必须保持复位值。
5:4	PLTS[1:0]	暂停低阈值 这些位设置了自动重发暂停帧的定时器阈值。这个阈值应当大于0，小于位[31:16]定义的暂停时间。低阈值的计算公式为PTM-PLTS。例如，PTM = 0x80（128个时间间隙），PLTS = 0x1（28个时间间隙），那么在第一个暂停帧发出100（128-28）个时间间隙后，将自动重发第二个暂停帧。 0x0：暂停时间 – 4个时间间隙 0x1：暂停时间 –28个时间间隙 0x2：暂停时间 – 144个时间间隙 0x3：暂停时间 – 256个时间间隙 **注意：**一个时间间隙是指MII接口发送512位（64字节）数据所需要的时间。
3	UPFDT	单播暂停帧检测 0：MAC只接收符合IEEE802.3规范定义的唯一多播地址的暂停帧 1：除了唯一多播地址的暂停帧，MAC同时还会使用MAC0地址（ENET_MAC_ADDR0H寄存器和ENET_MAC_ADDR0L寄存器）来检测暂停帧。
2	RFCEN	接收流控使能位 0：MAC不解析暂停帧 1：MAC解析并处理接收到的暂停帧。MAC关闭发送器一段指定的时间（接收帧中的

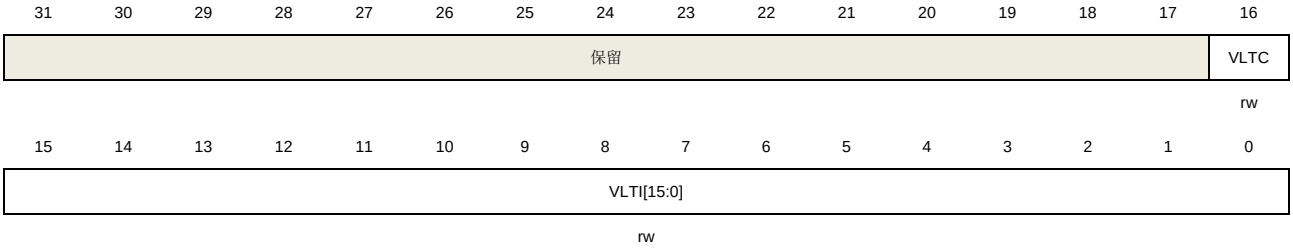
		暂停时间域值）。
1	TFCEN	发送流控使能位 0：MAC关闭发送流控功能。在全双工模式下，MAC不发送暂停帧；在半双工模式下，MAC关闭背压功能。 1：MAC开启发送流控功能。在全双工模式下，MAC使能暂停帧发送；在半双工模式下，MAC使能背压功能。
0	FLCB/BKPA	流控忙/背压激活 该位仅在TFCEN位置位时有效。 在全双工模式下，该位可发送暂停帧；在半双工模式下，该位可激活背压功能。 在全双工模式下，应用程序要确保在写ENET_MAC_FCTL寄存器之前该位为'0'。置位该位后，MAC将发送一个暂停帧到接口，在发送控制帧的过程中，该位始终为'1'，直到暂停控制帧发送完成以后，MAC将该位重置为'0'。 在半双工模式下，设置该位为'1'可以激活背压功能。在背压功能有效时，如果MAC接收到新的帧，就会在发送端发送阻塞信号，通知有冲突发生。

43.4.8. MAC VLAN 标签寄存器（ENET_MAC_VLT）

地址偏移：0x001C
复位值：0x0000 0000

该寄存器包含了用来识别VLAN帧的IEEE802.1Q VLAN标签。MAC把接收到帧的第13，14字节（长度/类型域）与0x8100比较，再把之后的2个字节（第15，16字节）和VLAN标签比较。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域	名称	描述
31:17	保留	必须保持复位值。
16	VLTC	12位VLAN标签比较位 该位选择用12位或16位VLAN标签来进行比较。 0：接收到的VLAN帧的全部16位数据（第15和16字节）都用来与VLTI位比对 1：仅用VLTI位[11:0]12位数据和接收到VLAN帧的相应域比对
15:0	VLTI[15:0]	VLAN标签标识符位 这些位用来识别VLAN帧的802.1Q VLAN标签格式。格式如下： VLTI[15:13]：UP（用户优先级） VLTI[12]：CFI（标准格式指示符） VLTI[11:0]：VID（VLAN标识符） 如果比较的位（当VLTC=1，则为VLTI[11:0]；当VLTC=0，则为VLTI[15:0]）值是全'0'，则MAC不再比对检验VLAN帧的第15、16字节，并将接收帧的类型域值是0x8100的帧都直接视为VLAN帧。

如果用于比较的位不是全为'0'，则使用VLTI[11:0]（VLTC=1）或VLTI[15:0]（VLTC=0）进行比较。

43.4.9. MAC 远程唤醒帧过滤器寄存器（ENET_MAC_RWFF）

地址偏移：0x0028
复位值：0x0000 0000

该寄存器实质上是指向8个不透明的唤醒帧过滤器寄存器的指针（使用同一个偏移地址）。对该寄存器地址(偏移为0x0028)的8次连续写操作，可以写入全部8个唤醒帧过滤器寄存器；对该寄存器地址（偏移为0x0028）的8次连续读操作，可以读出全部8个唤醒帧过滤器寄存器。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

图 43-15. 远程唤醒帧过滤器寄存器

	31 ... 0
Wakeup frame filter reg0	Byte Mask of Filter-0
Wakeup frame filter reg1	Byte Mask of Filter-1
Wakeup frame filter reg2	Byte Mask of Filter-2
Wakeup frame filter reg3	Byte Mask of Filter-3
Wakeup frame filter reg4	Reserved | Filter 3 Command | Reserved | Filter 2 Command | Reserved | Filter 1 Command | Reserved | Filter 0 Command
Wakeup frame filter reg5	Offset of Filter 3 | Offset of Filter 2 | Offset of Filter 1 | Offset Filter 0
Wakeup frame filter reg6	Filter 1 CRC - 16 | Filter 0 CRC - 16
Wakeup frame filter reg7	Filter 3 CRC - 16 | Filter 2 CRC - 16

43.4.10. MAC 唤醒管理寄存器（ENET_MAC_WUM）

地址偏移：0x002C
复位值：0x0000 0000

该寄存器设置并监控唤醒事件。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						GU	保留		WUFR	MPKR	保留		WFEN	MPEN	PWD
						rw			rc_r	rc_r			rw	rw	rs

位/位域	名称	描述
31	WUFFRPR	唤醒帧过滤器寄存器指针复位

		向该位写'1'，将会把远程唤醒帧过滤器寄存器指针ENET_MAC_RWFF复位，该位在指针复位完成后自动清'0'。 0：无作用 1：复位ENET_MAC_RWFF寄存器指针
30:10	保留	必须保持复位值。
9	GU	全局单播 向该位写1，所有能通过MAC地址过滤器的单播帧，都被认为是唤醒帧。 0：不是所有接收的单播帧都被认为是唤醒帧 1：所有能通过MAC地址过滤器的单播帧，都被认为是唤醒帧。
8:7	保留	必须保持复位值。
6	WUFR	接收到唤醒帧 读本寄存器可以清'0'该位。 0：没有接收到唤醒帧 1：接收到唤醒帧，并发生唤醒事件。
5	MPKR	接收到Magic Packet 读本寄存器可以清'0'该位。 0：没有接收到Magic Packet 1：接收到Magic Packet帧，并发生唤醒事件。
4:3	保留	必须保持复位值。
2	WFEN	唤醒帧使能位 0：禁能在接收到唤醒帧时产生唤醒事件 1：使能在接收到唤醒帧时产生唤醒事件
1	MPEN	Magic Packet使能位 0：禁能在接收到Magic Packet唤醒帧时产生唤醒事件 1：使能在接收到Magic Packet唤醒帧时产生唤醒事件
0	PWD	低功耗位 该位由软件置位，由硬件复位。当该位置位，MAC丢弃所有接收到的帧。当发生了唤醒事件，使得退出低功耗模式，硬件会自动将该位清'0'。

43.4.11. MAC 调试寄存器（ENET_MAC_DBG）

地址偏移：0x0034

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						TXFF	TXFNE	保留	TXFW	TXFRS[1:0]		PCS	SOMT[1:0]		MTNI
						ro	ro		ro	ro		ro	ro		ro

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						RXFS[1:0]		保留	RXFRS[1:0]		RXFW	保留	RXAFS[1:0]		MRNI
						ro			ro		ro		ro		ro

位/位域	名称	描述
31:26	保留	必须保持复位值。
25	TXFF	TxFIFO满标志位 0：TxFIFO未满 1：TxFIFO已满
24	TXFNE	TxFIFO非空标志位 0: TxFIFO空 1: TxFIFO不为空
23	保留	必须保持复位值。
22	TXFW	正在写TxFIFO 0: 没有向TxFIFO写帧数据 1: 正在向TxFIFO写帧数据
21:20	TXFRS[1:0]	TxFIFO读操作状态 0x0：空闲状态 0x1：读状态 0x2：等待MAC发送器返回Tx状态 0x3：写发送描述符状态，或清空TxFIFO。
19	PCS	暂停状态 0：MAC发送器不处于暂停状态 1：MAC发送器处于暂停状态，并暂停发送帧。
18:17	SOMT[1:0]	MAC发送器状态 0x0：空闲状态 0x1：等待前一帧的状态返回或IFG/BACKOFF周期结束 0x2：对于全双工模式，表示正在发送暂停控制帧。 0x3：从FIFO读取待发送的帧
16	MTNI	MAC发送器不空闲位 0：MAC发送器处于空闲状态 1：MAC发送器处于非空闲状态
15:10	保留	必须保持复位值。
9:8	RXFS	RxFIFO状态 0x0：RxFIFO空 0x1：RxFIFO中字节数低于流控下阈值 0x2：RxFIFO中字节数高于流控上阈值 0x3：RxFIFO满
7	保留	必须保持复位值。
6:5	RXFRS[1:0]	RxFIFO读操作状态 0x0：空闲状态 0x1：读数据帧状态

		0x2：读帧状态（包括时间戳） 0x3：清空帧
4	RXFW	正在写RxFIFO 0：没有向RxFIFO中写帧数据 1：正在向RxFIFO写帧数据
3	保留	必须保持复位值。
2:1	RXAFS[1:0]	Rx异步FIFO状态 RXAFS[1]：Rx异步FIFO在HCLK时钟域进行读操作（MAC取出数据） RXAFS[0]：Rx异步FIFO在MAC RX_CLK时钟域进行写操作（MAC存入数据）
0	MRNI	MAC接收器不空闲位 0：MAC接收器处于空闲状态 1：MAC接收器处于非空闲状态

43.4.12. MAC 中断状态寄存器（ENET_MAC_INTF）

地址偏移：0x0038
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	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						TMST	保留		MSCT	MSCR	MSC	WUM	保留		
						rc_r			r	r	r	r			

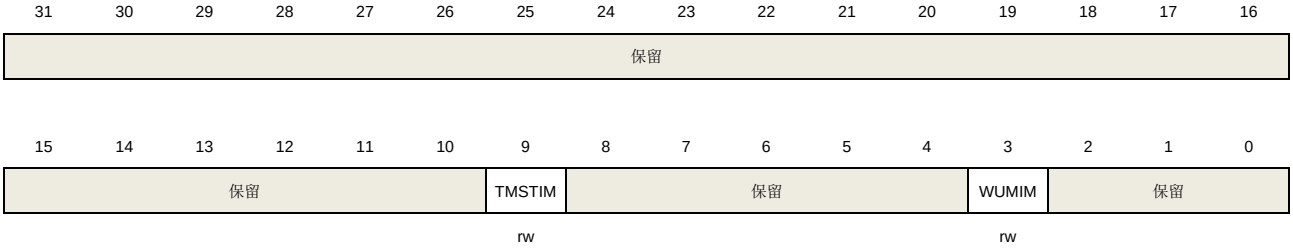
位/位域	名称	描述
31:10	保留	必须保持复位值。
9	TMST	时间戳触发状态 读 ENET_PTP_TSF 寄存器可以清'0'该位。 0：系统时间值小于期望时间值 1：系统时间值等于或者超过期望时间值
8:7	保留	必须保持复位值。
6	MSCT	MSC 发送状态 0：没有产生任一 ENET_MSC_TINTF 寄存器中的中断 1：产生任一 ENET_MSC_TINTF 寄存器中的中断
5	MSCR	MSC 接收状态 0：没有产生任一 ENET_MSC_RINTF 寄存器中的中断 1：产生任一 ENET_MSC_RINTF 寄存器中的中断

4	MSC	MSC 状态 该位为 MSCT 位与 MSCR 位的逻辑或。 0：MSCT 位和 MSCR 位均为'0' 1：MSCT 位和 MSCR 位中有位为'1'
3	WUM	WUM 状态 该位为 ENET_MAC_WUM 寄存器中的 WUFR 和 MPKR 位的逻辑或。 0：未接收到唤醒帧或者 Magic Packet 帧 1：在低功耗模式下，接收到唤醒帧或者 Magic Packet。
2:0	保留	必须保持复位值。

43.4.13. MAC 中断屏蔽寄存器（ENET_MAC_INTMSK）

地址偏移：0x003C
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

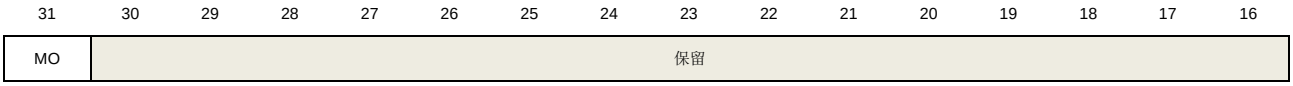


位/位域	名称	描述
31:10	保留	必须保持复位值。
9	TMSTIM	时间戳触发中断屏蔽位 0：允许产生时间戳中断 1：禁止产生时间戳中断
8:4	保留	必须保持复位值。
3	WUMIM	WUM中断屏蔽位 0：允许由于ENET_MAC_INTF寄存器的WUM状态位置位而引发的中断 1：禁止由于 ENET_MAC_INTF 寄存器的 WUM 状态位置 1 而引发的中断
2:0	保留	必须保持复位值。

43.4.14. MAC 地址 0 高寄存器（ENET_MAC_ADDR0H）

地址偏移：0x0040
复位值：0x8000 FFFF

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR0H[15:0]															

rw

位/位域	名称	描述
31	MO	该位总是为'1'。
30:16	保留	必须保持复位值。
15:0	ADDR0H[15:0]	MAC地址0高16位 这些位包含了 6 字节 MAC 地址 0 的高 16 位，这些位用于作为接收帧的地址过滤，还用于发送流控中发送暂停帧时插入作为帧的源地址。

43.4.15. MAC 地址 0 低寄存器（ENET_MAC_ADDR0L）

地址偏移：0x0044
复位值：0xFFFF FFFF

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR0L[31:16]															

rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR0L[15:0]															

rw

位/位域	名称	描述
31:0	ADDR0L[31:0]	MAC地址0低32位 这些位包含了 6 字节 MAC 地址 0 的低 32 位，这些位用于作为接收帧的地址过滤，还用于发送流控中发送暂停帧时插入作为帧的源地址。

43.4.16. MAC 地址 1 高寄存器（ENET_MAC_ADDR1H）

地址偏移：0x0048
复位值：0x0000 FFFF

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFE	SAF	MB[5:0]						保留							

rw rw rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR1H[15:0]															

rw

位/位域	名称	描述
31	AFE	地址过滤使能 0：不使用 MAC 地址 1 进行地址过滤 1：地址过滤器使用 MAC 地址 1 来进行完美过滤
30	SAF	源地址过滤器 0：MAC 地址 1[47:0]用来和接收帧的目标地址进行比对 1：MAC 地址 1[47:0]用来和接收帧的源地址进行比对
29:24	MB[5:0]	屏蔽字节位 当某个位置’1’时，MAC 不再把接收帧目标地址/源地址的对应字节与 MAC 地址 1 的相应字节进行比较。每个控制位对应的 MAC 地址字节如下： MB[5]：ENET_MAC_ADDR1H [15:8] MB[4]：ENET_MAC_ADDR1H [7:0] MB[3]：ENET_MAC_ADDR1L [31:24] MB[2]：ENET_MAC_ADDR1L[23:16] MB[1]：ENET_MAC_ADDR1L[15:8] MB[0]：ENET_MAC_ADDR1L [7:0]
23:16	保留	必须保持复位值。
15:0	ADDR1H[15:0]	MAC 地址 1 高[47:32]位 这些位包含了 6 字节的 MAC 地址 1 的高 16 位。

43.4.17. MAC 地址 1 低寄存器（ENET_MAC_ADDR1L）

地址偏移：0x004C
复位值：0xFFFF FFFF

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR1L[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR1L[15:0]															
rw															

位/位域	名称	描述
31:0	ADDR1L[31:0]	MAC 地址 1 低 32 位 这些位包含了 6 字节 MAC 地址 1 的低 32 位。

43.4.18. MAC 地址 2 高寄存器（ENET_ MAC_ADDR2H）

地址偏移：0x0050
复位值：0x0000 FFFF

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFE	SAF	MB[5:0]						保留							
rw	rw	rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR2H[15:0]															
rw															

位/位域	名称	描述
31	AFE	地址过滤使能 0：不使用 MAC 地址 2 进行地址过滤 1：地址过滤器使用 MAC 地址 2 来进行完美过滤
30	SAF	源地址过滤器 0：MAC 地址 2[47:0]用来和接收帧的目标地址进行比对 1：MAC 地址 2[47:0]用来和接收帧的源地址进行比对
29:24	MB[5:0]	屏蔽字节位 当某个位置'1'时，MAC 不再把接收帧目标地址/源地址的对应字节与 MAC 地址 2 的相应字节进行比较。每个控制位对应的 MAC 地址字节如下： MB[5]：ENET_MAC_ADDR2H [15:8] MB[4]：ENET_MAC_ADDR2H [7:0] MB[3]：ENET_MAC_ADDR2L [31:24] MB[2]：ENET_MAC_ADDR2L[23:16] MB[1]：ENET_MAC_ADDR2L[15:8] MB[0]：ENET_MAC_ADDR2L [7:0]
23:16	保留	必须保持复位值。
15:0	ADDR2H[15:0]	MAC 地址 2 高[47:32]位 这些位包含了 6 字节的 MAC 地址 2 的高 16 位。

43.4.19. MAC 地址 2 低寄存器（ENET_MAC_ADDR2L）

地址偏移：0x0054

复位值：0xFFFF FFFF

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR2L[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR2L[15:0]															
rw															

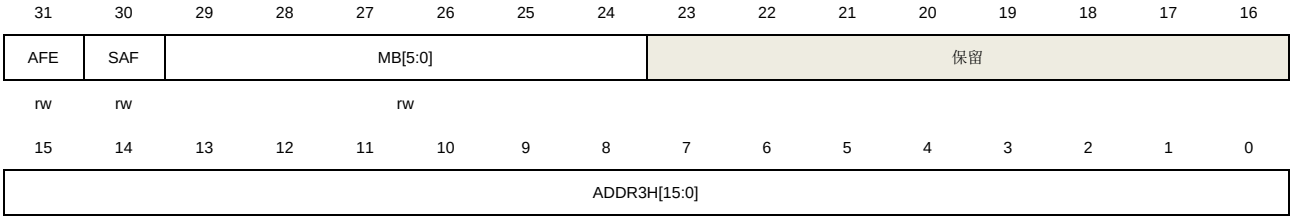
位/位域	名称	描述

31:0	ADDR2L[31:0]	MAC 地址 2 低 32 位 这些位包含了 6 字节 MAC 地址 2 的低 32 位。

43.4.20. MAC 地址 3 高寄存器（ENET_MAC_ADDR3H）

地址偏移：0x0058
复位值：0x0000 FFFF

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域	名称	描述
31	AFE	地址过滤使能 0：不使用 MAC 地址 3 进行地址过滤 1：地址过滤器使用 MAC 地址 3 来进行完美过滤
30	SAF	源地址过滤器 0：MAC 地址 3[47:0]用来和接收帧的目标地址进行比对 1：MAC 地址 3[47:0]用来和接收帧的源地址进行比对
29:24	MB[5:0]	屏蔽字节位 当某个位置'1'时，MAC 不再把接收帧目标地址/源地址的对应字节与 MAC 地址 3 的相应字节进行比较。每个控制位对应的 MAC 地址字节如下： MB[5]：ENET_MAC_ADDR3H [15:8] MB[4]：ENET_MAC_ADDR3H [7:0] MB[3]：ENET_MAC_ADDR3L [31:24] MB[2]：ENET_MAC_ADDR3L[23:16] MB[1]：ENET_MAC_ADDR3L[15:8] MB[0]：ENET_MAC_ADDR3L [7:0]
23:16	保留	必须保持复位值。
15:0	ADDR3H[15:0]	MAC 地址 3 高[47:32]位 这些位包含了 6 字节的 MAC 地址 3 的高 16 位。

43.4.21. MAC 地址 3 低寄存器（ENET_MAC_ADDR3L）

地址偏移：0x005C
复位值：0xFFFF FFFF

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR3L[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR3L[15:0]															
rw															

位/位域	名称	描述
31:0	ADDR3L[31:0]	MAC 地址 3 低 32 位 这些位包含了 6 字节 MAC 地址 3 的低 32 位。

43.4.22. MAC 流控阈值寄存器（ENET_MAC_FCTH）

地址偏移：0x1080

复位值：0x0000 0015

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	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									RFD[2:0]			保留	RFA[2:0]		
									rw				rw		

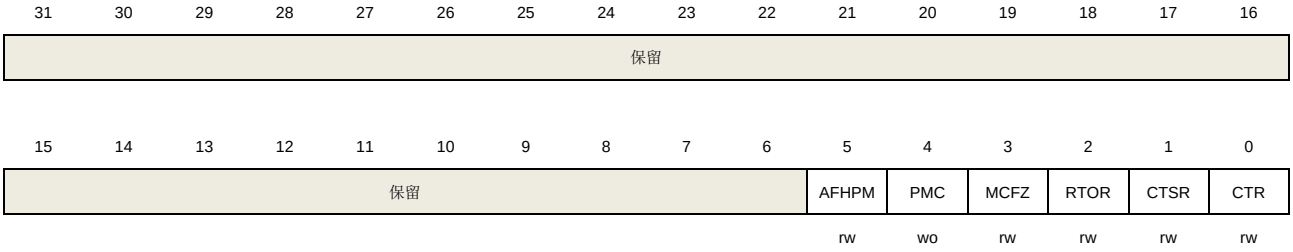
位/位域	名称	描述
31:7	保留	必须保持复位值。
6:4	RFD[2:0]	流控失效阈值 这些位设置了流控失效的阈值。这个值应当小于位[2:0]定义的流控激活阈值。当 RxFIFO 中未处理的数据低于这些位所设置的值，流控功能将自动失效。 0x0：256 字节 0x1：512 字节 0x2：768 字节 0x3：1024 字节 0x4：1280 字节 0x5：1536 字节 0x6，0x7：1792 字节
3	保留	必须保持复位值。
2:0	RFA[2:0]	流控激活阈值 这些位设置了流控激活的阈值。若使能了流控功能，当 RxFIFO 中未处理的数据超过了这些位所设置的值，流控功能将被激活。 0x0：256 字节 0x1：512 字节

0x2：768 字节
0x3：1024 字节
0x4：1280 字节
0x5：1536 字节
0x6，0x7：1792 字节

43.4.23. MSC 控制寄存（ENET_MSC_CTL）

地址偏移：0x0100
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

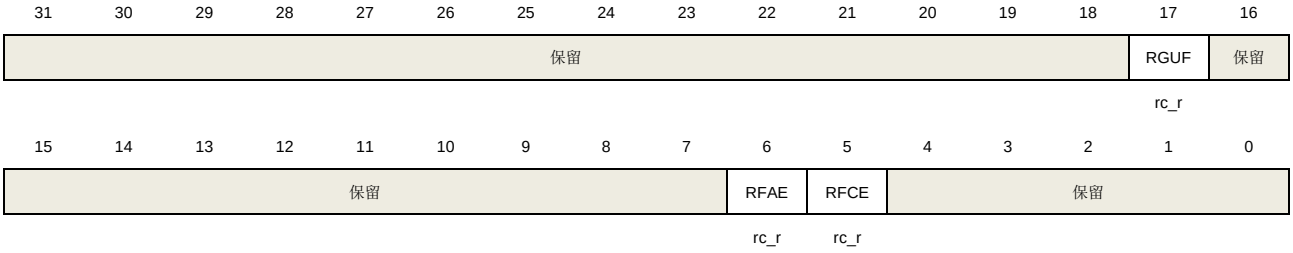


位/位域	名称	描述
31:6	保留	必须保持复位值。
5	AFHPM	近似全值或半值预设模式 0：预设 MSC 计数器的值为近似半值（0x7FFF FFF0） 1：预设 MSC 计数器的值为近似全值（0xFFFF FFF0） **注意：**该位仅在 PMC 位置位时有效。
4	PMC	MAC 计数器预设位 0：无作用 1：将 MSC 计数器预设为一个预设值。预设值取决于 AFHPM 位。
3	MCFZ	MSC 计数器冻结位 0：MSC 计数器正常工作 1：冻结 MSC 计数器，保持它们的当前值。RTOR 位可在计数器冻结状态时工作。
2	RTOR	读时复位 0：读 MSC 计数器后，计数器不复位。 1：读 MSC 计数器后，计数器复位。
1	CTSR	计数器停止回转 0：计数器在计数到最大值后，会重新从 0 开始计数。 1：计数器在计数到最大值后，不会重新从 0 开始计数。
0	CTR	计数器复位 该位置位后，会在 1 个时钟周期后由硬件自动清零。 0：无作用 1：复位所有计数器

43.4.24. MSC 接收中断状态寄存器（ENET_MSC_RINTF）

地址偏移：0x0104
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

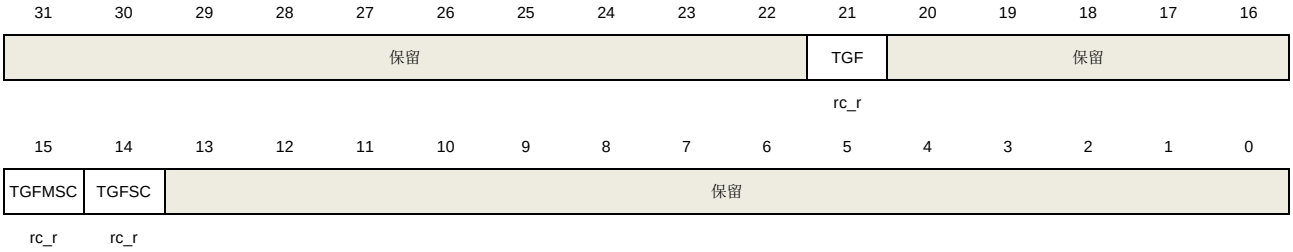


位/位域	名称	描述
31:18	保留	必须保持复位值。
17	RGUF	接收到”好”的单播帧 0：接收“好”单播帧计数器值小于最大值的一半 1：接收“好”单播帧计数器值达到最大值的一半
16:7	保留	必须保持复位值。
6	RFAE	接收到帧对齐错误 0：对齐错误接收帧计数器值小于最大值的一半 1：对齐错误接收帧计数器值达到最大值的一半
5	RFCE	接收到帧 CRC 错误 0：CRC 错误接收帧计数器值小于最大值的一半 1：CRC 错误接收帧计数器值达到最大值的一半
4:0	保留	必须保持复位值。

43.4.25. MSC 发送中断状态寄存器（ENET_MSC_TINTF）

地址偏移：0x0108
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域	名称	描述

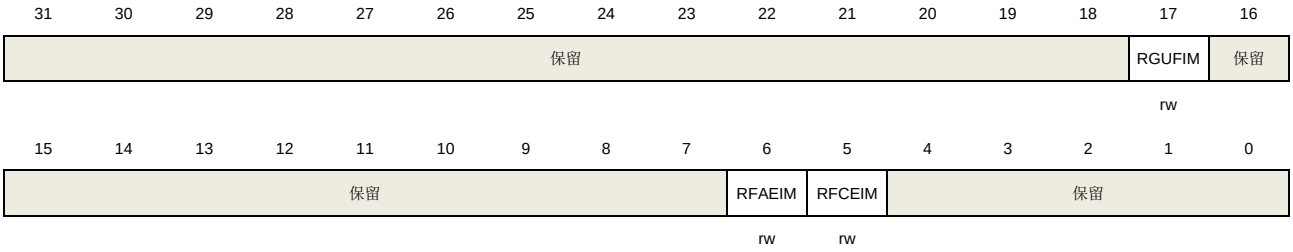
31:22	保留	必须保持复位值。
21	TGF	发送”好”的帧 0：发送“好”单播帧计数器值小于最大值的一半 1：发送“好”单播帧计数器值达到最大值的一半
20:16	保留	必须保持复位值。
15	TGFMSC	发送”好”的帧时遇到 1 个以上冲突 0：1 次以上冲突后发送”好”帧计数器值小于最大值的一半 1：1 次以上冲突后发送”好”帧计数器值达到最大值的一半
14	TGFSC	发送”好”的帧时仅遇到 1 个冲突 0：1 次冲突后发送”好“帧计数器值小于最大值的一半 1：1 次冲突后发送”好”帧计数器值达到最大值的一半
13:0	保留	必须保持复位值。

43.4.26. MSC 接收中断屏蔽寄存器（ENET_MSC_RINTMSK）

地址偏移：0x010C
复位值：0x0000 0000

该寄存器包含当接收统计计数器达到其最大值的一半时所产生的中断的屏蔽位。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域	名称	描述
31:18	保留	必须保持复位值。
17	RGUFIM	接收到”好”的单播帧的中断屏蔽位 0：不屏蔽当 RGUF 位为’1’时发生的中断 1：屏蔽当 RGUF 位为’1’时发生的中断
16:7	保留	必须保持复位值。
6	RFAEIM	接收帧对齐错误中断屏蔽位 0：不屏蔽当 RFAE 位为’1’时发生的中断 1：屏蔽当 RFAE 位为’1’时发生的中断
5	RFCEIM	接收帧 CRC 错误中断屏蔽位 0：不屏蔽当 RFCE 位为’1’时发生的中断 1：屏蔽当 RFCE 位为’1’时发生的中断

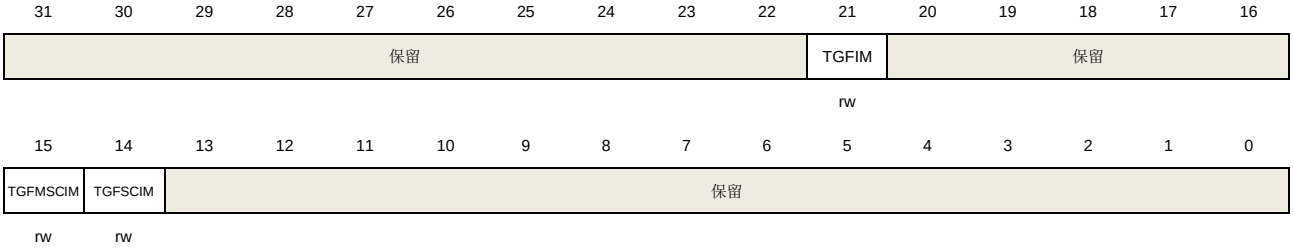
4:0	保留	必须保持复位值。

43.4.27. MSC 发送中断屏蔽寄存器（ENET_MSC_TINTMSK）

地址偏移：0x0110
复位值：0x0000 0000

该寄存器可以设置相应中断的屏蔽位。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



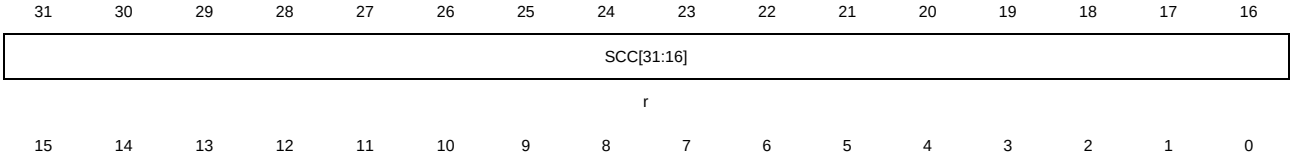
位/位域	名称	描述
31:22	保留	必须保持复位值。
21	TGFIM	发送”好”的帧的中断屏蔽位 0：不屏蔽当 TGF 位为’1’时发生的中断 1：屏蔽当 TGF 位为’1’时发生的中断
20:16	保留	必须保持复位值。
15	TGFMSCIM	遇到 1 个以上冲突后发送”好”帧中断屏蔽位 0：不屏蔽当 TGFMSC 位为’1’时发生的中断 1：屏蔽当 TGFMSC 位为’1’时发生的中断
14	TGFSCIM	仅遇到 1 个冲突后发送”好”帧中断屏蔽位 0：不屏蔽当 TFGSC 位为’1’时发生的中断 1：屏蔽当 TFGSC 位为’1’时发生的中断
13:0	保留	必须保持复位值。

43.4.28. MSC 1 次冲突后发送”好”帧的计数器寄存器（ENET_MSC_SCCNT）

地址偏移：0x014C
复位值：0x0000 0000

该寄存器统计在半双工模式下，在只遇到一次冲突后发送帧成功时的帧的数目。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



SCC[15:0]

r

位/位域	名称	描述
31:0	SCC[31:0]	1 次冲突后发送好帧计数器 这些位是 1 次冲突后发送的”好”帧的计数器。

43.4.29. MSC 1 次以上冲突后发送”好”帧的计数器寄存器（ENET_MSC_MSCCNT）

地址偏移：0x0150
复位值：0x0000 0000

该寄存器统计在半双工模式下，遇到一次以上冲突后发送帧成功时的帧的数目。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

位/位域	名称	描述
31:0	MSCC[31:0]	1 次以上冲突后发送”好”帧计数器 这些位是 1 次以上冲突后发送”好”帧的计数器。

43.4.30. MSC 发送”好”帧计数器寄存器（ENET_MSC_TGFCNT）

地址偏移：0x0168
复位值：0x0000 0000

该寄存器统计发送”好”帧的数目。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

位/位域	名称	描述
31:0	TGF[31:0]	发送”好”帧计数器 这些位是发送“好”帧的计数器。

43.4.31. MSC CRC 错误接收帧计数器寄存器（ENET_MSC_RFCECNT）

地址偏移：0x0194
复位值：0x0000 0000

该寄存器统计接收帧中有CRC错误的帧的数目。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:0	RFCER[31:0]	CRC 错误接收帧计数器 这些位是接收帧中有 CRC 错误的帧的计数器。

43.4.32. MSC 对齐错误接收帧计数器寄存器（ENET_MSC_RFAECNT）

地址偏移：0x0198
复位值：0x0000 0000

该寄存器统计接收帧中有对齐错误帧的数目。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:0	RFAER[31:0]	对齐错误接收帧计数器 这些位是接收帧中有对齐错误的帧的计数器。

43.4.33. MSC“好”单播帧接收帧计数器寄存器（ENET_MSC_RGUFCNT）

地址偏移：0x01C4
复位值：0x0000 0000

该寄存器统计接收到”好”单播帧的数目。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:0	RGUF[31:0]	“好”单播帧接收帧计数器 这些位是接收到”好“的单播帧的计数器。

43.4.34. PTP 时间戳控制寄存器（ENET_PTP_TSCTL）

地址偏移：0x0700

复位值：0x0000 2000

该寄存器用于配置时间戳的产生和更新。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留													MAFEN	CKNT[1:0]	
													rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MNMSEN	ETMSEN	IP4SEN	IP6SEN	ESEN	PFSV	SCROM	ARFSEN	保留		TMSARU	TMSITEN	TMSSTU	TMSSTI	TMSFCU	TMSEN
rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:19	保留	必须保持复位值。
18	MAFEN	PTP 帧 MAC 地址过滤使能 0：无作用 1：当接收帧的类型域值为 0x88f7，则使能 MAC 地址 1-3 用于 PTP 帧过滤。
17:16	CKNT[1:0]	时间戳时钟节点类型 0x0：普通类型时钟 0x1：边界类型时钟 0x2：端对端透明类型时钟 0x3：点对点透明类型时钟
15	MNMSEN	接收主节点消息时时间戳快照使能 该位仅在 CKNT=0x0 或 0x1 时有效。 0：从节点消息时间戳快照使能 1：主节点消息时间戳快照使能
14	ETMSEN	接收事件类型的消息时时间戳快照使能 0：接收到除了 Announce，Management 和 Signaling 以外的所有其他类型的消息

		时，时间戳快照使能。 1：只有接收到事件类型的消息（SYNC，DELAY_REQ，PDELAY_REQ 和 PDELAY_RESP）时，时间戳快照使能。
13	IP4SEN	接收 IPv4 帧时时间戳使能 0：接收到 IPv4 帧时，时间戳失能。 1：接收到 IPv4 帧时，时间戳使能。
12	IP6SEN	接收 IPv6 帧时时间戳使能 0：接收到 IPv6 帧时，时间戳失能。 1：接收到 IPv6 帧时，时间戳使能。
11	ESEN	接收以太网帧时时间戳使能 0：接受到非类型帧时，时间戳失能。 1：接受到非类型帧时，时间戳使能。
10	PFSV	监听 PTP 帧版本 0：版本 1（版本为 IEEE STD. 1588-2002/1588-2008） 1：版本 2（版本为 IEEE STD. 1588-2008）
9	SCROM	亚秒计数器回转模式 0：二进制回转模式，亚秒计数器在达到 0x7FFF FFFF 以后重新从 0 计数。 1：十进制回转模式，亚秒计数器在达到 0x3B9A C9FF（0d999 999 999）以后重新从 0 计数。
8	ARFSEN	所有接收帧时间戳快照使能 0：不对所有接收帧使能时间戳功能 1：对所有接收帧使能时间戳功能
7:6	保留	必须保持复位值。
5	TMSARU	时间戳加数寄存器更新位 该位在更新完成后清'0'。该位在置位前必须确保读出为'0'。 0：不将时间戳加数寄存器的值更新到 PTP 模块进行精调 1：将时间戳加数寄存器的值更新到 PTP 模块进行精调
4	TMSITEN	时间戳中断触发使能 0：禁止时间戳中断 1：使能时间戳中断，当系统时间超过期望时间寄存器的值时将会产生中断。 **注意：**产生时间戳中断后，该位将会清'0'。
3	TMSSTU	时间戳系统时间更新位 置位该位之前，必须确保 TMSSTU 位和 TMSSTI 位读出为'0'。 0：系统时间保持不变 1：更新系统时间，在原有系统时间上加上或者减去时间戳高和低更新寄存器的值。 完成更新后，硬件将会清除该位。
2	TMSSTI	时间戳系统时间初始化位 置位该位之前，必须确保该位读出为'0'。 0：系统时间保持不变

1：初始化系统时间，将原有系统时间替换为时间戳高和低更新寄存器的值。在初始化完成后，硬件将会清除该位。

1 TMSFCU 时间戳粗调或者精调更新位
0：用粗调的方式更新系统时间戳
1：用精调的方式更新系统时间戳

0 TMSEN 时间戳使能位
0：禁止时间戳功能
1：使能接收和发送帧的时间戳功能
注意：每次设置该位为'1'后，都需要重新初始化系统时间。

表 43-8. 支持的 PTP 时间戳及其寄存器配置

CKNT (位 17:16)	**0X**			**10**		**11**	
MNMSEN (位 15)	X(*)	1	0	X			
ETMSEN (位 14)	0	1	1	0	1	0	1
支持的时间戳消息类型	SYNC, FOLLOW_UP, DELAY_REQ, DELAY_RESP	DELAY_REQ	SYNC	SYNC, FOLLOW_UP, DELAY_REQ, DELAY_RESP	SYNC, FOLLOW_UP	SYNC, FOLLOW_UP, DELAY_REQ, DELAY_RESP, PDELAY_REQ, PDELAY_RESP	SYNC, PDELAY_REQ, PDELAY_RESP

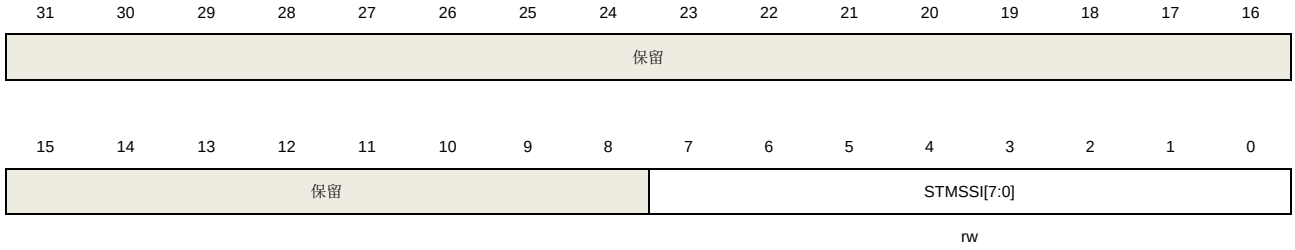
*：指无关值。

43.4.35. PTP 亚秒递增寄存器（ENET_PTP_SSINC）

地址偏移：0x0704
复位值：0x0000 0000

该寄存器用于配置亚秒递增寄存器的8位递增值。在粗调模式下，每个HCLK时钟周期，系统时间就加一次该寄存器的值。在精调模式下，在累加器溢出时，系统时间才加一次该寄存器的值。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域 名称 描述

31:8	保留	必须保持复位值。
7:0	STMSSI[7:0]	系统时间亚秒递增 在每次系统时间递增时，把这些位的值加到系统时间的亚秒值上。

43.4.36. PTP 时间戳高寄存器（ENET_PTP_TSH）

地址偏移：0x0708
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:0	STMS[31:0]	系统时间秒位 这些位表示了当前系统时间的秒值。

43.4.37. PTP 时间戳低寄存器（ENET_PTP_TSL）

地址偏移：0x070C
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31	STS	系统时间符号位 0：时间值是正的 1：时间值是负的
30:0	STMSS[30:0]	系统时间亚秒位 这些位表示了当前系统时间的亚秒值，亚秒精度为 0.46ns。

43.4.38. PTP 时间戳高更新寄存器（ENET_PTP_TSUH）

地址偏移：0x0710
复位值：0x0000 0000

使用该寄存器的值对当前系统时间替换、加或减。时间戳高和低更新寄存器可以用来初始化或更新MAC的当前系统时间。应当先写这2个寄存器，再置位时间戳控制寄存器的TMSSTI位或TMSSTU位。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:0	TMSUS[31:0]	时间戳秒更新位 这些位表示的值在初始化时用于替换系统时间，在更新时表示在系统时间上加上或减去的秒值。

43.4.39. PTP 时间戳低更新寄存器（ENET_PTP_TSUL）

地址偏移：0x0714
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31	TMSUPNS	时间戳更新正或者负符号位 TMSSTI 位置’1’时，该位应当为’0’。 0：在系统时间上加上时间戳更新值 1：从系统时间中减去时间戳更新值
30:0	TMSUSS[30:0]	时间戳更新亚秒位 这些位表示的值在初始化时用于替换系统时间，在更新时表示在系统时间上加上或减去的亚秒值。

43.4.40. PTP 时间戳加数寄存器（ENET_PTP_TSADDEND）

地址偏移：0x0718
复位值：0x0000 0000

该寄存器只用于系统时间更新方式为精调模式。该寄存器的值在每个时钟周期都会累加到32位累加器上，一旦该累加器溢出就更新系统时间。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:0	TMSA[31:0]	时间戳加数 这些位用于时钟同步时加到累加器上的值，以实现时间同步。

43.4.41. PTP 期望时间高寄存器（ENET_PTP_ETH）

地址偏移：0x071C
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

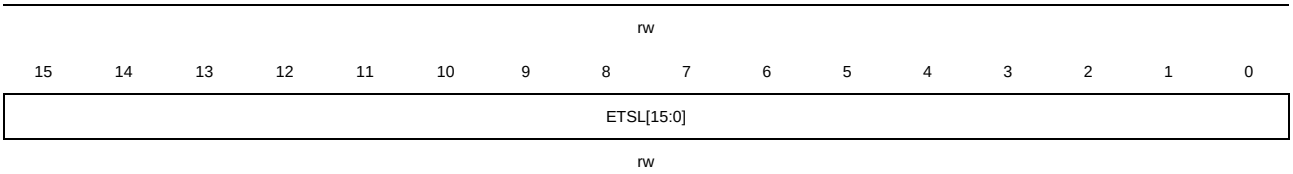
位/位域	名称	描述
31:0	ETSH[31:0]	期望时间戳高位 这些位表示了期望时间的秒值。

43.4.42. PTP 期望时间低寄存器（ENET_PTP_ETL）

地址偏移：0x0720
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

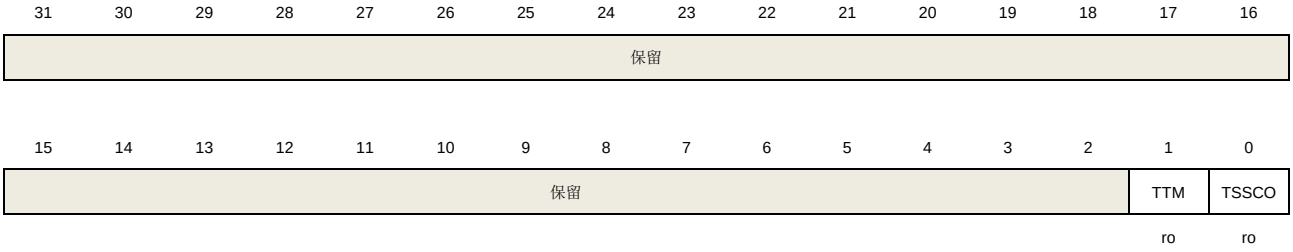


位/位域	名称	描述
31:0	ETSL[31:0]	期望时间戳低位 这些位表示了期望时间的纳秒值。

43.4.43. PTP 时间戳标志寄存器（ENET_PTP_TSF）

地址偏移：0x0728
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

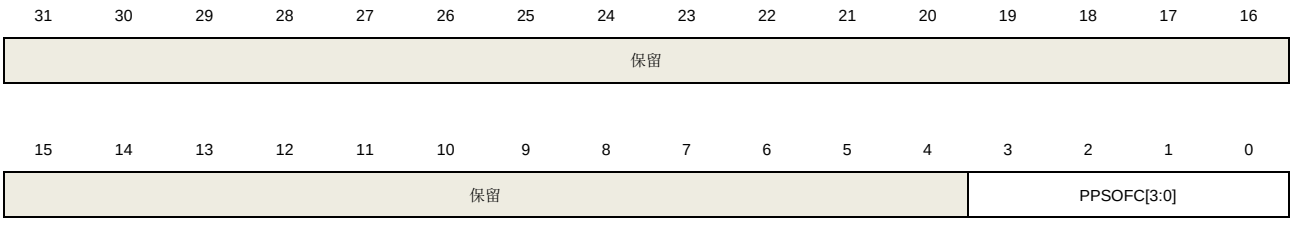


位/位域	名称	描述
31:2	保留	必须保持复位值。
1	TTM	期望时间比较位 0：系统时间小于期望时间 1：系统时间大于或等于期望时间 **注意：**读 ENET_PTP_TSF 寄存器将清除该位。
0	TSSCO	时间戳秒计数器上溢位 0：时间戳秒计数器没有发生上溢 1：时间戳秒计数器值大于 0xFFFF FFFF

43.4.44. PTP PPS 控制寄存器（ENET_PTP_PPSCTL）

地址偏移：0x072C
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



rw

位/位域	名称	描述
31:4	保留	必须保持复位值。
3:0	PPSOFC[3:0]	PPS 输出频率配置位 0x0：1Hz（脉冲宽度：二进制回转模式下为 125ms，十进制回转模式下为 100ms） 0x1：2Hz（脉冲宽度：二进制回转模式下 50%占空比） 0x2：4Hz（脉冲宽度：二进制回转模式下 50%占空比） …. 0xF：32768（2^{15}）Hz（脉冲宽度：二进制回转模式下 50%占空比） **注意：** 如果选择的是十进制回转模式，则建议仅使用 PPSOFC=0。

43.4.45. DMA 总线控制寄存器（ENET_DMA_BCTL）

地址偏移：0x1000
复位值：0x0002 0101

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留					MB	AA	FPBL	UIP	RXDP[5:0]						FB
					rw	rw	rw	rw	rw						rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTPR[1:0]		PGBL[5:0]						DFM	DPSL[4:0]					DAB	SWR
rw		rw						rw	rw					rw	rs

位/位域	名称	描述
31:27	保留	必须保持复位值。
26	MB	混合传输位 0：AHB 主接口仅传输小于或等于 16 固定长度的传输 1：AHB 主接口将以 INCR 传输大于 16 长度的传输 **注意：**MB 和 FB 位应当且必须只有其中一位为‘1’。
25	AA	地址对齐 0：关闭传输地址对齐功能 1：使能传输地址对齐，如果 FB 位为’1’，AHB 接口对齐所有连续传输至起始地址的 LS 位（位 1 到位 0）。如果 FB 位为’0’，除第一次 AHB 访问的地址（访问数据缓存的起始地址）不对齐，后续的传输与地址均对齐。
24	FPBL	4×PGBL 模式 0：PGBL 值（位[22:17]和位[13:8]）作为 DMA 传输长度值 1：PGBL 值（位[22:17]和位[13:8]）乘以 4 作为 DMA 传输长度值
23	UIP	使用分散 PGBL 0：PGBL 值（位[13:8]）对 DMA 接收和发送控制器都有效

		1：RXDP[5:0]位用于 RxDMA 的传输长度值，PGBL[5:0]位用于 TxDMA 的传输长度值。
22:17	RXDP[5:0]	RxDMA PGBL 位 如果 UIP=0，则这些位无效。仅当 UIP=1 时，这些位定义了一次 DMA 转发的最大数据传输次数。 0x01：最大数据传输次数为 1 0x02：最大数据传输次数为 2 0x04：最大数据传输次数为 4 0x08：最大数据传输次数为 8 0x10：最大数据传输次数为 16 0x20：最大数据传输次数为 32 其他：保留。
16	FB	固定传输位 0：AHB 在连续传输时，只用 SINGLE 和 INCR 数据传输操作。 1：AHB 在连续传输时，用 SINGLE，INCR4，INCR8 和 INCR16 数据传输操作。 **注意**：MB 和 FB 位应当且必须只有其中一位为'1'。
15:14	RTPR[1:0]	接收发送优先级比率 这些位表示 RxDMA 和 TxDMA 之间的访问优先级比率。 0x0：RxDMA：TxDMA = 1：1 0x1：RxDMA：TxDMA = 2：1 0x2：RxDMA：TxDMA = 3：1 0x3：RxDMA：TxDMA = 4：1 **注意**：该位只在 DMA 仲裁模式为循环模式（DAB=0）时有效。
13:8	PGBL[5:0]	可编程的数据传输长度位 这些位定义了一次 DMA 转发的最大数据传输次数。如果 UIP=1，则这些位仅用于 TxDMA 传输。如果 UIP=0 时，则这些位同时用于 TxDMA 和 RxDMA 传输。 0x01：最大数据传输次数为 1 0x02：最大数据传输次数为 2 0x04：最大数据传输次数为 4 0x08：最大数据传输次数为 8 0x10：最大数据传输次数为 16 0x20：最大数据传输次数为 32 其他：保留。
7	DFM	描述符模式 0：常规描述符模式 1：增强描述符模式
6:2	DPSL[4:0]	描述符跳跃长度 这些位仅对于环模式的两个描述符有效，定义了两个无链接的描述符之间从当前描述符的结尾到下一个描述符开头的地址差值，单位为字（32 位）。若 DPSL 域为 0 则 DMA 认为描述符是相邻地连续排列的。
1	DAB	DMA 仲裁位

		该位指示了 TxDMA 和 RxDMA 之间的仲裁模式。 0：根据 RTPR 位的值以循环方式仲裁 1：固定模式，接收的优先级高于发送
0	SWR	软件复位 在所有时钟域的复位操作完成之后，该位将由硬件清零。 **注意：**在写任何 MAC 的寄存器前，应当确保该位为'0'。 0：MAC 内部寄存器正常工作 1：复位 MAC 所有内核寄存器

43.4.46. DMA 发送查询使能寄存器（ENET_DMA_TPEN）

地址偏移：0x1004
复位值：0x0000 0000

该寄存器用于TxDMA对发送描述符列表的查询。TxDMA通常因为发送帧的数据下溢错误或者描述符被CPU占有(DAV=0)而进入暂停状态。可以对该寄存器写任意值使能发送查询。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

位/位域	名称	描述
31:0	TPE[31:0]	发送查询使能位 对这些位写任意值，DMA 使能发送查询，将查询当前描述符（描述符地址在 ENET_DMA_CTDADDR 寄存器中）是否被 CPU 占有。如果不是（DAV=1），则描述符可用，DMA 退出暂停状态并恢复工作。如果是（DAV=0），则 TxDMA 回到暂停状态，并把 ENET_DMA_STAT 的位 TBU 置'1'。

43.4.47. DMA 接收查询使能寄存器（ENET_DMA_RPEN）

地址偏移：0x1008
复位值：0x0000 0000

该寄存器用于RxDMA对接收描述符列表的查询。对该寄存器写任意值可以使能接收查询。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

RPE[15:0]

rw_wt

位/位域	名称	描述
31:0	RPE[31:0]	接收查询使能位 对这些位写任意值，DMA 使能接收查询，将查询当前描述符（描述符地址在 ENET_DMA_CRDADDR 寄存器中）是否被 CPU 占有。如果不是（DAV=1），则描述符可用，DMA 退出暂停状态并恢复工作。如果是（DAV=0），则 TxDMA 回到暂停状态，并把 ENET_DMA_STAT 的位 RBU 置'1'。

43.4.48. DMA 接收描述符列表地址寄存器（ENET_DMA_RDTADDR）

地址偏移：0x100C
复位值：0x0000 0000

接收描述符列表寄存器指向接收描述符队列的开始。描述符队列位于物理内存，并且其地址必须字对齐。只有在接收停止的时候，才允许写该寄存器。在开启RxDMA接收流程之前，必须正确配置该寄存器。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

位/位域	名称	描述
31:0	SRT[31:0]	接收队列基址 这些位包含了接收描述符队列第一个描述符的地址。SRT[1:0]的取值默认为'0'， 因此 SRT[1:0]这两个最低位是只读的。

43.4.49. DMA 发送描述符列表地址寄存器（ENET_DMA_TDTADDR）

地址偏移：0x1010
复位值：0x0000 0000

该寄存器指向发送描述符队列的起始。描述符队列位于物理内存，并且其地址必须字对齐。只有在发送停止的时候，才允许写该寄存器。在开启TxDMA发送流程之前，必须正确配置该寄存器。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:0	STT[31:0]	发送队列基址 这些位包含了发送描述符列表第一个描述符的地址。STT[1:0]的取值默认为'0'，因此STT[1:0]这两个最低位是只读的。

43.4.50. DMA 状态寄存器（ENET_DMA_STAT）

地址偏移：0x1014
复位值：0x0000 0000

该寄存器表示DMA的状态位。读ENET_DMA_STAT寄存器并不能清除其中的标志位。对寄存器位[16:0]（除保留位）需要写'1'才能清除，而写'0'是无效的。通过设置ENET_DMA_INTEN寄存器里的相应位，可以屏蔽位[16:0]触发的中断。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		TST	WUM	MSC	保留	EB[2:0]			TP[2:0]			RP[2:0]			NI
		r	r	r		r			r			r			rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AI	ER	FBE	保留		ET	RWT	RPS	RBU	RS	TU	RO	TJT	TBU	TPS	TS
rc_w1	rc_w1	rc_w1			rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位/位域	名称	描述
31:30	保留	必须保持复位值。
29	TST	时间戳触发状态 该位指示发生了一个时间戳中断事件。通过清除 TMST 标志，可以清零该位。当该位置'1'时，如果相应中断屏蔽位复位，则产生中断。 0：未发生时间戳中断事件 1：发生了时间戳中断事件
28	WUM	WUM 状态 该位指示发生了一个 WUM 事件。当两个事件触发源状态都被清除时，可以清零该位。当该位置'1'时，如果相应中断屏蔽位复位，则产生中断。 0：WUM 模块未发生中断事件 1：WUM 模块发生了中断事件
27	MSC	MSC 状态 该位指示发生了一个 MSC 事件。当所有事件触发源状态都被清除时，可以清零该位。当该位置'1'时，如果相应中断屏蔽位复位，则产生中断。 0：MSC 模块未发生中断事件

		1：MSC 模块发生了中断事件
26	保留	必须保持复位值。
25:23	EB[2:0]	错误位状态 当 FBE=1 时，这些位将对 AHB 总线上的总线响应错误进行错误类型解析。 EB[0] 1：TxDMA 传输数据时出错 0：RxDMA 传输数据时出错 EB[1] 1：读数据转发时出错 0：写数据转发时出错 EB[2] 1：访问描述符时出错 0：访问数据缓存时出错
22:20	TP[2:0]	发送流程状态 这些位表示 TxDMA 的状态。 0x0：停止，接到复位或者停止发送命令。 0x1：运行，正在取发送描述符。 0x2：运行，正在等待状态信息。 0x3：运行，正在读取内存中数据并存入 TxFIFO 中。 0x4，0x5：保留。 0x6：暂停，发送描述符不可用或发送缓存数据下溢。 0x7：运行，正在关闭发送描述符。
19:17	RP[2:0]	接收流程状态 这些位表示 RxDMA 的状态。 0x0：停止，接到复位或者停止接收命令。 0x1：运行，正在取接收描述符。 0x2：保留。 0x3：运行，正在等待接收数据包。 0x4：暂停，接收描述符不可用。 0x5：运行，正在关闭接收描述符。 0x6：保留。 0x7：运行，正在把接收到数据包从 RxFIFO 转发到内存中。
16	NI	正常中断汇总 该位是下列位在相应中断使能位（ENET_DMA_INTEN 寄存器）使能了的情况下，其各取值的逻辑或： TS：发送中断 TBU：发送缓存不可用 RS：接收中断 ER：提前接收中断 **注意：**该位置’1’后，只有把造成该位置’1’的位清’0’（写’1’），才能把该位清’0’。
15	AI	异常中断汇总

		该位是下列位在相应中断使能位（ENET_DMA_INTEN 寄存器）使能了的情况下，其各取值的逻辑或： TPS：发送流程停止 TJT：发送 Jabber 超时 RO：RxFIFO 上溢 TU：发送数据下溢 RBU：接收缓存不可用 RPS：接收流程停止 RWT：接收看门狗超时 ET：提前发送中断 FBE：总线致命错误 **注意：**该位置’1’后，只有把造成该位置’1’的位清’0’（写’1’），才能把该位清’0’。
14	ER	提前接收状态 在接收中断位 RS 置’1’时，该位自动清’0’。 0：未接收到帧数据 1：接收到的数据帧已由 DMA 填满了第一个缓存
13	FBE	总线致命错误状态 该位指示发生了一个 AHB 接口响应错误，其错误类型可以由 EB[2:0]位进行解释。 0：未发生总线错误 1：发生了总线错误，相应的 DMA 控制器停止所有操作。
12:11	保留	必须保持复位值。
10	ET	提前发送状态 0：发送的帧还未完全传输到 TxFIFO 中 1：发送的帧已经完全传输到 TxFIFO 中
9	RWT	接收看门狗超时状态 0：接收到的帧长度小于 2048 字节 1：接收到的帧长度超过 2048 字节
8	RPS	接收流程停止状态 0：接收流程未停止 1：接收流程进入停止状态
7	RBU	接收缓存不可用状态 0：下一个接收描述符的 DAV 位为’1’ 1：下一个接收描述符的 DAV 位为’0’，RxDMA 进入暂停状态。
6	RS	接收状态 0：帧接收未完成 1：帧接收完成
5	TU	发送数据下溢状态 0：未发生发送数据下溢错误 1：发送帧的过程中发生数据下溢，同时发送进入暂停状态。

4	RO	接收上溢状态 0：未发生接收数据上溢错误 1：接收帧的过程中发生上溢错误。如果已有一部分帧数据转发到内存，则设置接收描述符 0 的上溢错误位 OERR 为'1'。
3	TJT	发送 Jabber 超时状态 0：未发生发送 Jabber 定时器超时事件 1：发送 Jabber 定时器超时。此时中止发送进程并进入停止状态，同时设置发送描述符 0 的 Jabber 超时位 JT 为'1'。
2	TBU	发送缓存不可用状态 0：下一个发送描述符的 DAV 位为'1' 1：下一个发送描述符的 DAV 位为'0'，TxDMA 进入暂停状态。
1	TPS	发送流程停止状态 0：发送未停止 1：发送停止
0	TS	发送状态 该位仅在发送描述符 0 中 LSG 和 INTC 位都置位时，才可被置位。 0：当前帧发送未完成 1：当前帧发送完成

43.4.51. DMA 控制寄存器（ENET_DMA_CTL）

地址偏移：0x1018
复位值：0x0000 0000

该寄存器设定了接收和发送的工作模式和命令。在整个DMA的初始化流程中，应当最后写该寄存器。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留					DTCERFD	RSFD	DAFRF	保留		TSFD	FTF	保留			TTHC[2]
					rw	rw	rw			rw	rs				rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTHC[1:0]		STE	保留					FERF	FUF	保留	RTHC[1:0]		OSF	SRE	保留
rw		rw						rw	rw		rw		rw	rw	

位/位域	名称	描述
31:27	保留	必须保持复位值。
26	DTCERFD	不丢弃 TCP/IP 校验和错误帧 0：当 FERF 位为'0'，MAC 丢弃所有有错的帧。 1：当 FERF 位为'1'，接收到帧仅有校验和错误时，MAC 不会丢弃该帧。
25	RSFD	接收存储转发 0：RxFIFO 工作在直通模式，转发阈值由 RTHC 位决定。

		1：RxFIFO 工作在存储转发模式，只有在帧完整写入 RxFIFO 后，RxDMA 才会把它转发给应用程序，此时 RTHC 位的取值会被忽略。
24	DAFRF	不清空接收帧 0：当接收描述符不可用时，RxDMA 就清空 RxFIFO 里的接收帧。 1：RxDMA 不清空接收帧，即使接收描述符不可用。
23:22	保留	必须保持复位值。
21	TSFD	发送存储转发 0：TxFIFO 工作在直通模式，发送阈值由 ENET_DMA_CTL 寄存器中的 TTHC 位决定。 1：TxFIFO 工作在存储转发模式，只有在帧完整写入 TxFIFO 后，MAC 才会把其发送出去，TTHC 位的取值会被忽略。 **注意：**在发送处于停止状态时，可以修改该位。
20	FTF	清空 TxFIFO 当此位为 1 时，TxFIFO 控制逻辑电路被复位到初始状态，TxFIFO 里所有的数据被清空/丢失。在清空操作完成后该位被自动清'0'。 **注意：**在该位为'0'之前，不允许写 ENET_DMA_CTL 寄存器。
19:17	保留	必须保持复位值。
16:14	TTHC[2:0]	发送阈值控制 这三位控制直通模式下 TxFIFO 的阈值。 当 TSFD=1 时，忽略这些位。 0x0：64 0x1：128 0x2：192 0x3：256 0x4：40 0x5：32 0x6：24 0x7：16
13	STE	开始/停止发送 0：在发送完当前帧或 TxDMA 进入暂停状态后，发送进程进入停止模式。保存发送描述符队列里下一发送描述符的位置，在传输重新开始时，这个描述符就变成当前描述符。 1：TxDMA 进入运行状态。DMA 获取当前发送描述符，发送帧描述符可从 ENET_DMA_TDTADDR 基址获取，若上一次发送为停止状态，则也可从发送描述符队列的指针位置获取。如果当前描述符的 DAV 位为'0'，则 TxDMA 进入暂停状态，并设置 TBU 位为'1'。如果在未设置完其他 DMA 寄存器的情况下就置位该位，则会引起不可预料的后果。
12:8	保留	必须保持复位值。
7	FERF	转发错误帧 0：当 RxFIFO 工作于直通模式（RSFD=0）时，如果在将 RxFIFO 数据转发到内存

		之前检测到了帧错误（CRC 错误、冲突错误、校验和错误、看门狗超时、溢出），则 RxFIFO 会丢弃这个错误的帧。但如果在将 RxFIFO 数据转发到内存之后才检测到了帧错误，则就不会丢弃该帧。当 RxFIFO 工作于存储转发模式时，在接收过程中一旦检测到帧错误，就会丢弃该帧。 1：除了过短帧外的所有帧都会转发给 DMA
6	FUF	转发长度不够的”好”帧 0：RxFIFO 丢弃所有长度小于 64 字节的帧，但如果在检测到过短帧之前，帧已开始转发该帧给应用程序（例如在直通模式下，帧长小于接收阈值），则将转发整个帧。 1：RxFIFO 把长度不够的”好”帧（帧长小于 64 字节但没有错误）转发给应用程序
5	保留	必须保持复位值。
4:3	RTHC[1:0]	接收阈值控制 这两位设置了在直通模式下 RxFIFO 的阈值。 **注意：**只有在 RSFD 位（位 21）为’0’时，这些位才有效。在 RSFD 位为’1’时忽略这些位。 0x0：64 0x1：32 0x2：96 0x3：128
2	OSF	操作第二帧 0：TxDMA 仅在接收到前一个帧的发送状态信息后，才开始发送下一个帧的数据。 1：TxDMA 在前一帧数据全部存入到 TxFIFO 之后，在接收到前一个帧的发送状态信息前，就开始发送下一个帧的数据。
1	SRE	开始/停止接收 0：在转发完当前接收帧后，RxDMA 进入停止模式。保存接收描述符队列里下一接收描述符的位置，在传输重新开始时，这个描述符就变成当前描述符。只有在接收运行时或接收暂停时，可“停止接收”。 1：把接收进程置为运行状态，DMA 检查接收描述符队列的当前位置，用来处理下一个收到的帧。接收帧描述符可从 ENET_DMA_RDTADDR 基址获取，若上一次接收为停止状态，则也可从接收描述符队列的指针位置获取。如果获取的描述符 DAV=0，那么接收进程进入暂停状态，并设置 RBU 位为’1’。只有在接收停止时或接收暂停时，”开始接收”命令才有效。在未设置完所有其他 DMA 寄存器之前发出”开始接收”命令，会引起不可预料的后果。
0	保留	必须保持复位值。

43.4.52. DMA 中断使能寄存器（ENET_DMA_INTEN）

地址偏移：0x101C
复位值：0x0000 0000

该寄存器可以使能ENET_DMA_STAT寄存器反映的中断。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AIE	ERIE	FBEIE	保留		ETIE	RWTIE	RPSIE	RBUIE	RIEN	TUIE	ROIE	TJTIE	TBUIE	TPSIE	TIE
rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:17	保留	必须保持复位值。
16	NIE	正常中断汇总使能 0：屏蔽正常中断 1：使能正常中断 该位使能下列位： TS：发送中断 TBU：发送缓存不可用 RS：接收中断 ER：提前接收中断
15	AIE	异常中断汇总使能 0：屏蔽异常中断 1：使能异常中断 该位使能下列位： TPS：发送流程停止 TJT：发送 Jabber 超时 RO：RxFIFO 上溢 TU：发送下溢 RBU：接收缓存不可用 RPS：接收流程停止 RWT：接收看门狗超时 ET：提前发送中断 FBE：总线致命错误
14	ERIE	提前接收中断使能 0：屏蔽提前接收中断 1：使能早接收中断
13	FBEIE	总线致命错误中断使能 0：屏蔽总线致命错误中断 1：使能总线致命错误中断
12:11	保留	必须保持复位值。
10	ETIE	提前发送中断使能 0：屏蔽提前发送中断 1：使能提前发送中断

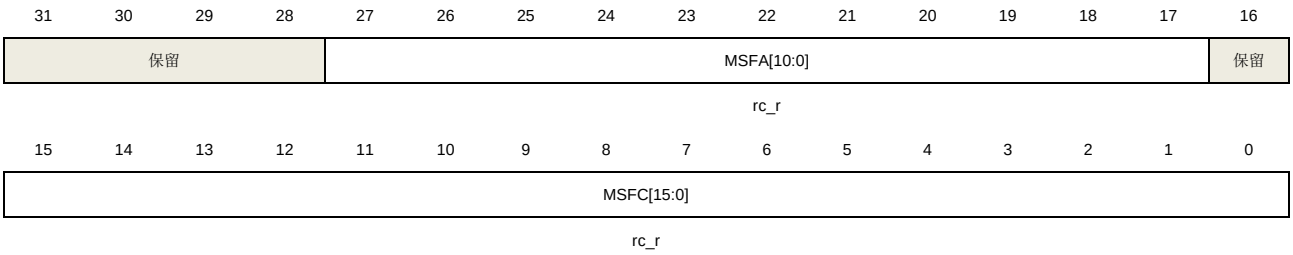
9	RWTIE	接收看门狗超时中断使能 0：屏蔽接收看门狗超时中断 1：使能接收看门狗超时中断
8	RPSIE	接收流程停止中断使能 0：屏蔽接收流程停止中断 1：使能接收流程停止中断
7	RBUIE	接收缓存不可用中断使能 0：屏蔽接收缓存不可用中断 1：使能接收缓存不可用中断
6	RIE	接收中断使能 0：屏蔽接收中断 1：使能接收中断
5	TUIE	发送下溢中断使能 0：屏蔽发送数据下溢中断 1：使能发送下溢中断
4	ROIE	接收上溢中断使能 0：屏蔽接收上溢中断 1：使能接收上溢中断
3	TJTIE	发送 Jabber 超时中断使能 0：屏蔽发送 Jabber 超时中断 1：使能发送 Jabber 超时中断
2	TBUIE	发送缓存不可用中断使能 0：屏蔽发送缓存不可用中断 1：使能发送缓存不可用中断
1	TPSIE	发送流程停止中断使能 0：屏蔽发送流程停止中断 1：使能发送流程停止中断
0	TIE	发送中断使能 0：屏蔽发送中断 1：使能发送中断

43.4.53. DMA 丢失帧和缓存溢出计数器寄存器（ENET_DMA_MFBOCNT）

地址偏移：0x1020
复位值：0x0000 0000

DMA有2个计数器，用来统计接收过程中丢失帧的数目。可通过读本寄存器来获取计数器的当前值。这个计数器通常用作故障诊断。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



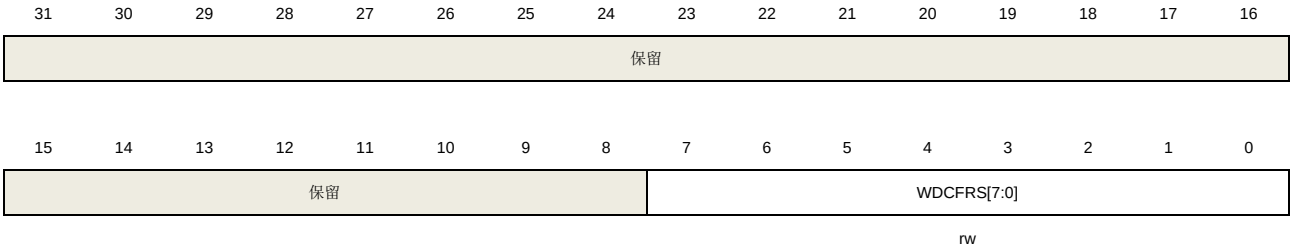
位/位域	名称	描述
31:28	保留	必须保持复位值。
27:17	MSFA[10:0]	应用程序丢失的帧 这些位指示了 RxFIFO 丢失的帧数目。
16	保留	必须保持复位值。
15:0	MSFC[15:0]	控制器丢失的帧 这些位表示了由于 MCU 的接收缓存不可用而导致 RxDMA 丢失的帧的数目。每当 DMA 清空一个输入帧时，这个计数器加 1。

43.4.54. DMA 接收状态看门狗计数器寄存器（ENET_DMA_RSWDC）

地址偏移：0x1024
复位值：0x0000 0000

向该寄存器写入一个值，可在延时一段可配置的时间之后，使能针对RS位（ENET_DMA_STAT寄存器）的看门狗定时器。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	WDCFRS[7:0]	接收状态看门狗计数器 这些位仅在接收描述符 1 的 DINTC 位置位时有效。当 DINTC=1 时，并接收到一个帧，则 RS 位会在接收完毕后延时 WDCFRS*256 个 HCLK 时钟周期再置位。

43.4.55. DMA 当前发送描述符地址寄存器（ENET_DMA_CTDADDR）

地址偏移：0x1048
复位值：0x0000 0000

当前发送描述符寄存器指向TxDMA正在读取的发送描述符起始地址。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:0	TDAP[31:0]	发送描述符地址指针 这些位在复位时清'0'，由 TxDMA 在操作过程中自动更新。

43.4.56. DMA 当前接收描述符地址寄存器（ENET_DMA_CRDADDR）

地址偏移：0x104C
复位值：0x0000 0000

当前接收描述符寄存器指向RxDMA正在读取的接收描述符起始地址。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:0	RDAP[31:0]	接收描述符地址指针 这些位在复位时清'0'，由 RxDMA 在操作过程中自动更新。

43.4.57. DMA 当前发送缓存地址寄存器（ENET_DMA_CTBADDR）

地址偏移：0x1050
复位值：0x0000 0000

该寄存器指向TxDMA正在读取的发送缓存的地址。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

TBAP[15:0]

r

位/位域	名称	描述
31:0	TBAP[31:0]	发送缓存地址指针 这些位在复位时清'0'，由 TxDMA 在工作过程中更新。

43.4.58. DMA 当前接收缓存地址寄存器（ENET_DMA_CRBADDR）

地址偏移：0x1054
复位值：0x0000 0000

该寄存器地址指向RxDMA正在读取的接收缓存地址。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

位/位域	名称	描述
31:0	RBAP[31:0]	接收缓存地址指针 这些位在复位时清'0'，由 RxDMA 在工作过程中更新。

44. 比较器（CMP）

44.1. 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。

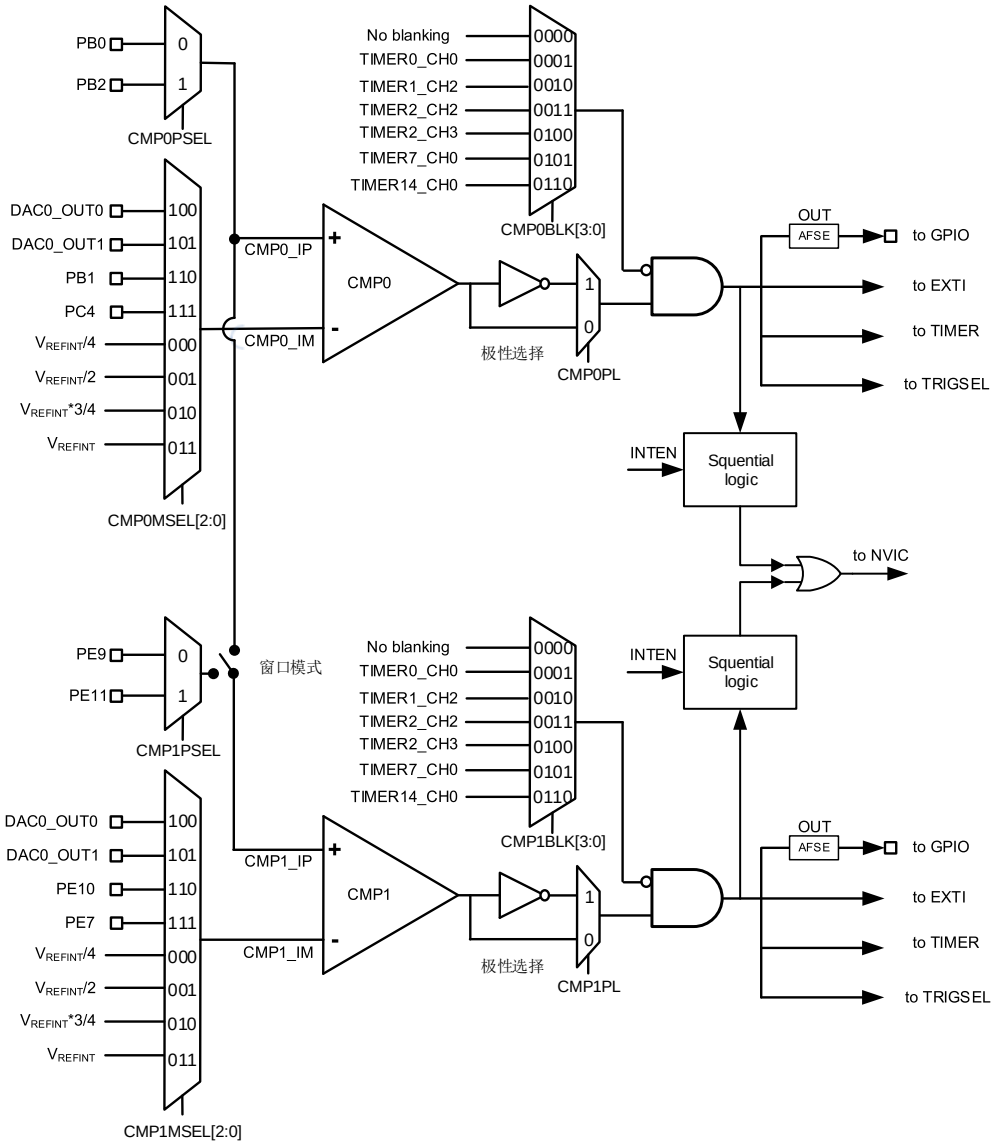
44.2. 主要特征

- 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；
- 窗口比较器；
- 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。

44.3. 功能描述

比较器的框图展示如下：

图 44-1. 比较器框图



注意：V_{REFINT} 是 1.2V。

44.3.1. 比较器时钟

比较器与 APB 总线连接，时钟与 PCLK 同步。

44.3.2. 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 输出连接到定时器输入通道；

- CMP 输出连接到定时器中止功能。

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 44-1. CMP 的输入和输出总结*详细描述了 CMP 的输入和输出。

表 44-1. CMP 的输入和输出总结

	CMP0	CMP1
CMP 同相输入连接到 I/O	PB0 PB2	PE9 PE11
CMP 反相输入连接到 I/O	PB1 PC4	PE10 PE7
CMP 反相输入连接到内部信号	V_{REFINT}/4, V_{REFINT}/2, V_{REFINT}*3/4, V_{REFINT}, DAC0_OUT0, DAC0_OUT1	V_{REFINT}/4, V_{REFINT}/2, V_{REFINT}*3/4, V_{REFINT}, DAC0_OUT0, DAC0_OUT1
CMP 输出连接到 I/O	PC5(AF13) PE12(AF13)	PE8(AF13) PE13(AF13)
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	•	
CMP 输出连接到 TIMER break	•	
CMP_MUX_OUT（由 AFSE[x]控制）	PA6(AF10) PA8(AF12) PB12(AF13) PE6(AF11) PE15(AF13) PG2(AF11) PG3(AF11) PG4(AF11) PK0(AF11) PK1(AF11) PK2(AF10)	

注意：

1. 有关CMP输出连接到内部信号的详细信息，请参阅系统配置寄存器；
2. 有关CMP输出连接到TIMER break的详细信息，请参阅TIMER模块；

44.3.3. 比较器供电模式

对于给定的程序，在比较器功耗和传输迟滞之间存在着权衡，可通过寄存器 CMPx_CS 的位 CMPxM [1:0]的配置进行调整。当 CMPxM [1:0]为 2'b 00 时，比较器以运行速度最快和功耗最大模式工作，但当 CMPxM[1:0]位 2'b 11 时，比较器以运行速度最慢和功耗最小的模式工作。

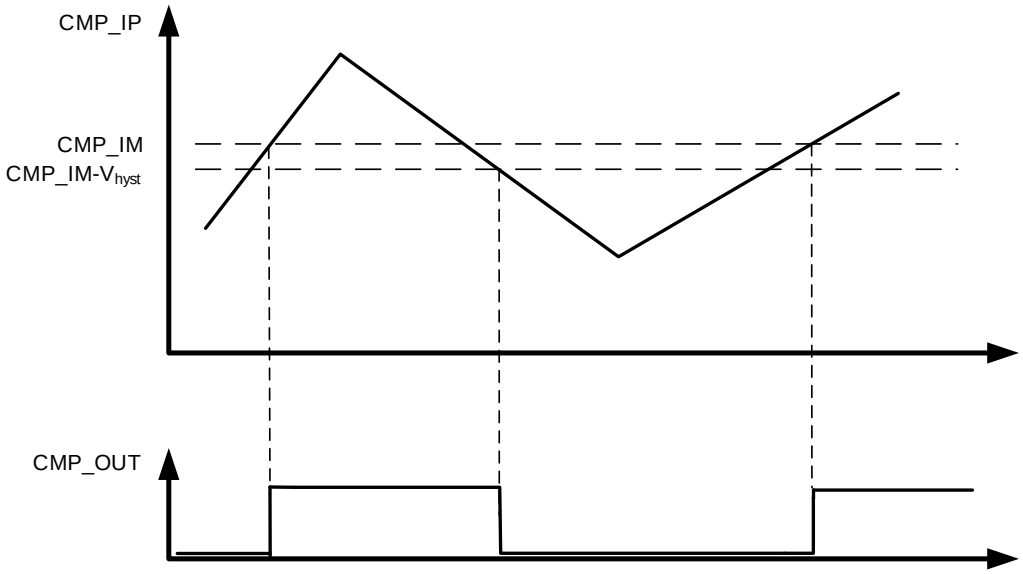
44.3.4. 比较器窗口模式

如果寄存器 CMP1_CS 的 WNDEN 位被置位，比较器的窗口模式被使能，比较器 1 的同相输入端即与比较器 0 的同相输入端相连。如果 CMP0 和 CMP1 的反相输入端连接不同的内部电压，可以通过分析 CMP0 和 CMP1 的输出结果监测输入电压的范围，该范围的上下限由反相输入端所连接的内部电压值决定。

44.3.5. 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 44-2. 比较器迟滞



44.3.6. 比较器寄存器写保护

比较器的控制状态寄存器（CMPx_CS）和外部选择寄存器（CMP_SR）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。

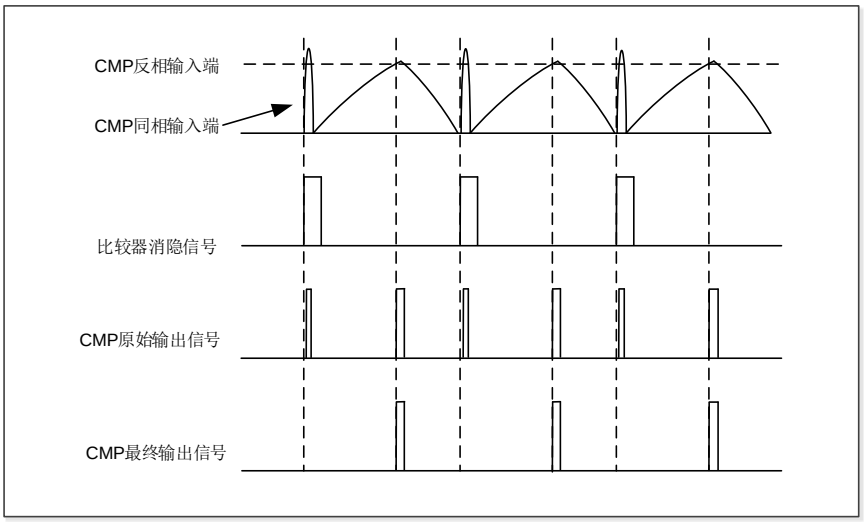
44.3.7. 比较器输出消隐

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 44-3. 比较器的输出消隐*显示了比较器的输出消隐功能。

图 44-3. 比较器的输出消隐



44.3.8. 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。

44.3.9. 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。

44.4. 比较器寄存器

CMP 基地址：0x5800 3800

44.4.1. 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														CMP1IF	CMP0IF
														r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														CMP1O	CMP0O
														r	r

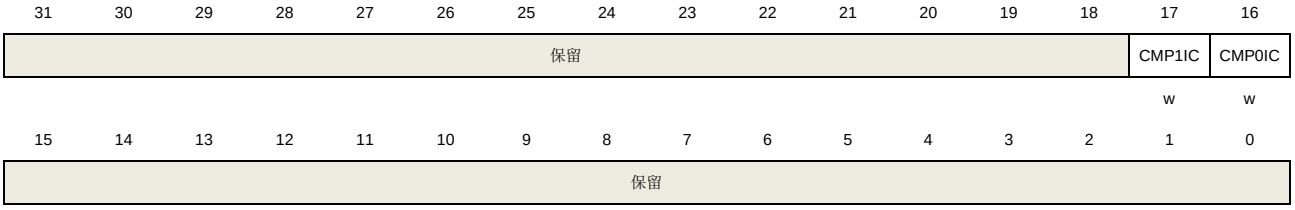
位/位域	名称	描述
31:18	保留	必须保持复位值。
17	CMP1IF	CMP1中断标志位 0：无CMP1中断发生 1：CMP1中断发生 当CMP1输出置位时，该位由硬件置1。 软件写1至CMP_IFC寄存器CMP1IC位清0。
16	CMP0IF	CMP0中断标志位 0：无CMP0中断发生 1：CMP0中断发生 当CMP0输出置位时，该位由硬件置1。 软件写1至CMP_IFC寄存器CMP0IC位清0。
15:2	保留	必须保持复位值。
1	CMP1O	CMP1 输出 该位反映 CMP1 输出状态，是只读位 0：同相输入端低于反相输入端，输出为低电平 1：同相输入端高于反相输入端，输出为高电平
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

44.4.2. 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。



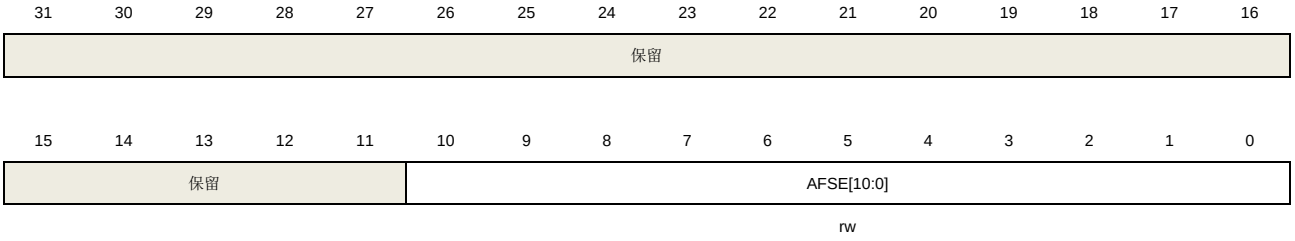
位/位域	名称	描述
31:18	保留	必须保持复位值。
17	CMP1IC	CMP1中断标志位清除 0：无清除中断标志位发生 1：清除中断标志位
16	CMP0IC	CMP0中断标志位清除 0：无清除中断标志位发生 1：清除中断标志位
15:0	保留	必须保持复位值。

44.4.3. 备用选择寄存器（CMP_SR）

地址偏移： 0x08

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



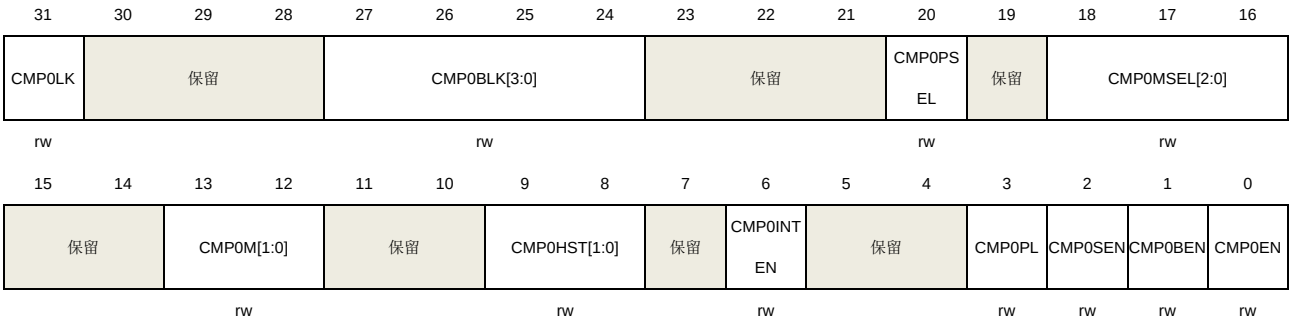
位/位域	名称	描述
31:11	保留	必须保持复位值。
10:0	AFSE[10:0]	CMP 备用输出端口选择，它们连接在 GPIO。 对于每一位，选择 0 为 CMP0_OUT 与相应的备用功能，选择 1 为 CMP1_OUT 与相应的备用功能。 位 0: PA6 位 1: PA8 位 2: PB12

位 3: PE6
位 4: PE15
位 5: PG2
位 6: PG3
位 7: PG4
位 8: PK0
位 9: PK1
位 10: PK2

44.4.4. CMP0 控制状态寄存器（CMP0_CS）

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	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 和 CMP_SR 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_CH0 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 TIMER2_CH3 输出比较信号为消隐源 0101：选择 TIMER7_CH0 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 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：PB0 1：PB2
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：DAC0_OUT0 101：DAC0_OUT1 110：PB1 111：PC4
15:14	保留	必须保持复位值。
13:12	CMP0M[1:0]	CMP0 模式 该位域用于控制 CMP0 的运行模式以调整速度和功耗。 00：高速 / 全功耗 01 / 10：中速 / 中功耗 11：超低速 / 超低功耗
11:10	保留	必须保持复位值。
9:8	CMP0HST[1:0]	CMP0 迟滞 该位域用于控制迟滞水平 00：无迟滞 01：低迟滞 10：中迟滞 11：高迟滞
7	保留	必须保持复位值。
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

44.4.5. CMP1 控制状态寄存器（CMP1_CS）

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
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		CMP1M[1:0]		保留		CMP1HST[1:0]		保留	CMP1INTEN	保留	WNDEN	CMP1PL	CMP1SEN	CMP1BEN	CMP1EN
		rw				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 和 CMP_SR 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_CH0 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 TIMER2_CH3 输出比较信号为消隐源 0101：选择 TIMER7_CH0 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 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：PE9 1：PE11
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：DAC0_OUT0 101：DAC0_OUT1 110：PE10 111：PE7
15:14	保留	必须保持复位值。
13:12	CMP1M[1:0]	CMP1 模式 该位域用于控制 CMP1 的运行模式以调整速度和功耗。 00：高速 / 全功耗 01 / 10：中速 / 中功耗 11：超低速 / 超低功耗
11:10	保留	必须保持复位值。
9:8	CMP1HST[1:0]	CMP1 迟滞 该位域用于控制迟滞水平 00：无迟滞 01：低迟滞 10：中迟滞 11：高迟滞
7	保留	必须保持复位值。
6	CMP1INTEN	CMP1中断使能 0：禁能 1：使能
5	保留	必须保持复位值。
4	WNDEN	窗口模式使能 该位用来选择 CMP1_IP 输入源。 0：CMP1_IP 连接到 CMP1 的同相输入端 1：CMP1_IP连接到CMP0_IP

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

45. 高性能数字滤波器（HPDF）

45.1. 简介

GD32H73x_75x 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。

45.2. 主要特性

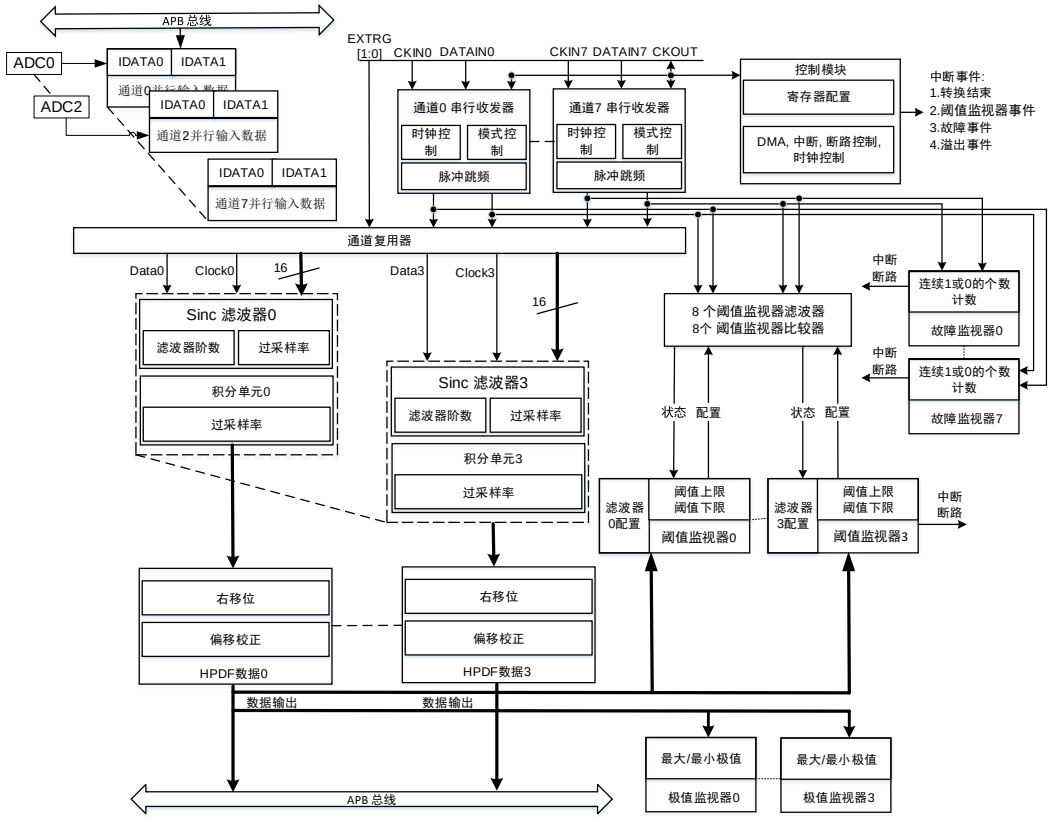
- 8 个复用数字串行输入通道
  - 可配置的SPI和曼切斯特接口
- 8 个内部数字并行输入通道
  - 高达 16 位分辨率的输入
  - 内部源：ADC 数据或内存（CPU/DMA 写）数据流
- 可配置的 Sinc 滤波器和积分器
  - 可配置 Sinc 滤波器的阶数、过采样率（抽取率）
  - 可配置积分器的采样率
- 阈值监视功能
  - 独立的 Sinc 滤波器，可配置阶数和过采样率（抽取率）
  - 可配置的数据输入源：串行通道输入数据或 HPDF 输出数据
- 故障监视功能
  - 拥有 8 位的计数器，用于监视串行通道输入数据流中连续的 0 或 1
- 极值监视器功能
  - 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

45.3. 功能描述

45.3.1. HPDF 结构框图

HPDF 的结构框图如*图 45-1. HPDF 结构框图*所示。

图 45-1. HPDF 结构框图



HPDF 接口通过*表 45-1. 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 45-1. 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 1bit 位的数据流。

表 45-2. HPDF 断路连接

断路名称	断路目标
HPDF_BREAK[0]	TIMER0 break0 / TIMER14 break0 / TIMER41 break0
HPDF_BREAK[1]	TIMER0 break1 / TIMER15 break0 / TIMER42 break0
HPDF_BREAK[2]	TIMER7 break0 / TIMER16 break0 / TIMER43 break0
HPDF_BREAK[3]	TIMER7 break1 / TIMER40 break0 / TIMER44 break0

45.3.2. 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。

45.3.3. 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_CFG2 寄存器中的 SAI0SEL[2:0]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。

45.3.4. 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。

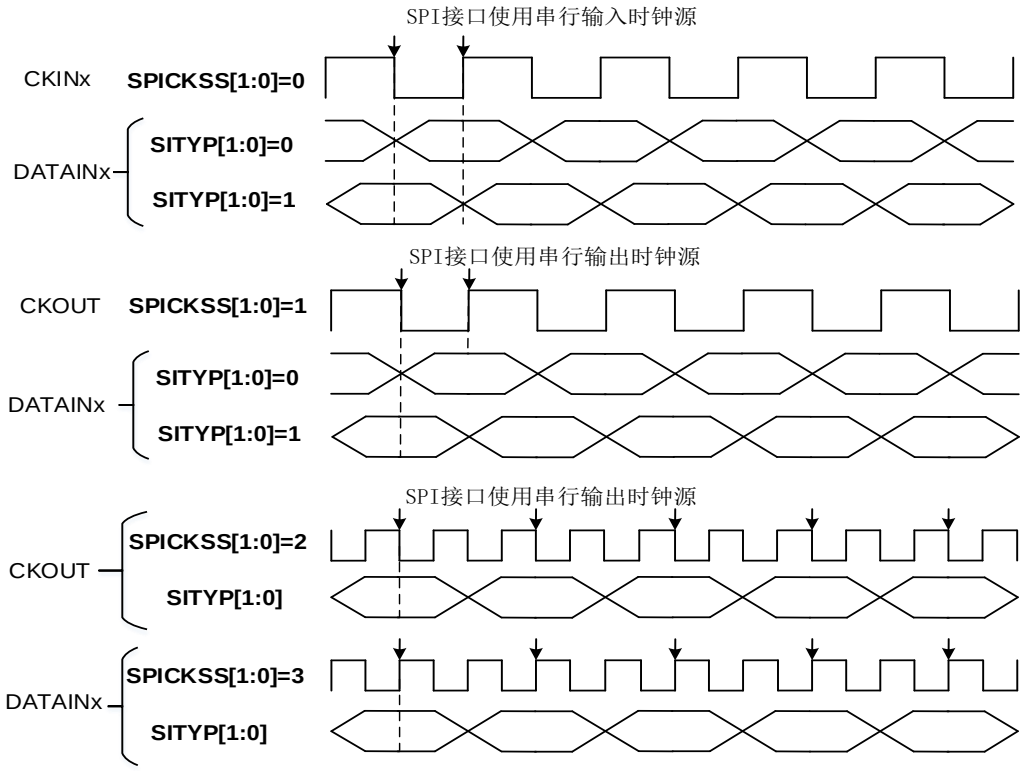
SPI 通信时的数据采样点由 HPDF_CHxCTL 寄存器中的 SITYP[1:0]位域和 SPICKSS[1:0]位域共同决定。SPI 通信时的数据采样点如表。

表 45-3. 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 45-3. SPI 接口时钟配置*，SPI 数据传输的时序图如下图所示。

图 45-2. 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 $f_{HPDFCLK}/4$。

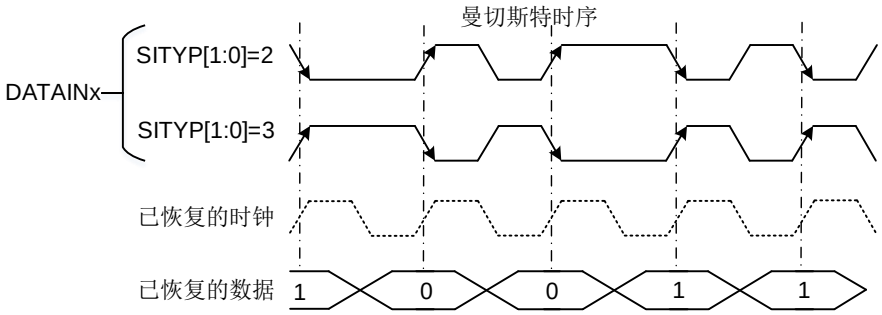
曼切斯特数据接口

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 45-3. 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：

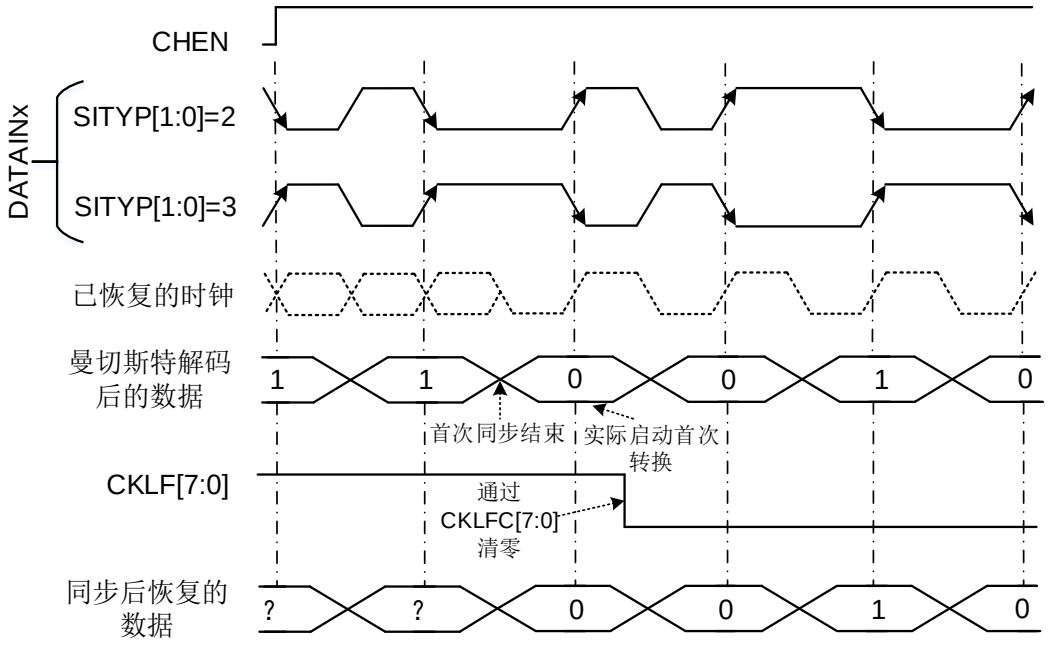
$$((CKOUTDIV+1)xT_{SYSCLK})<T_{Manchester_clock}<(2xCKOUTDIV\times T_{SYSCLK}) \quad (45\text{-}1)$$

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 45-4. 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：

$$T = CNVCNT / f_{HPDFCLK} = [SFOR * IOR] / f_{CKIN})$$

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。

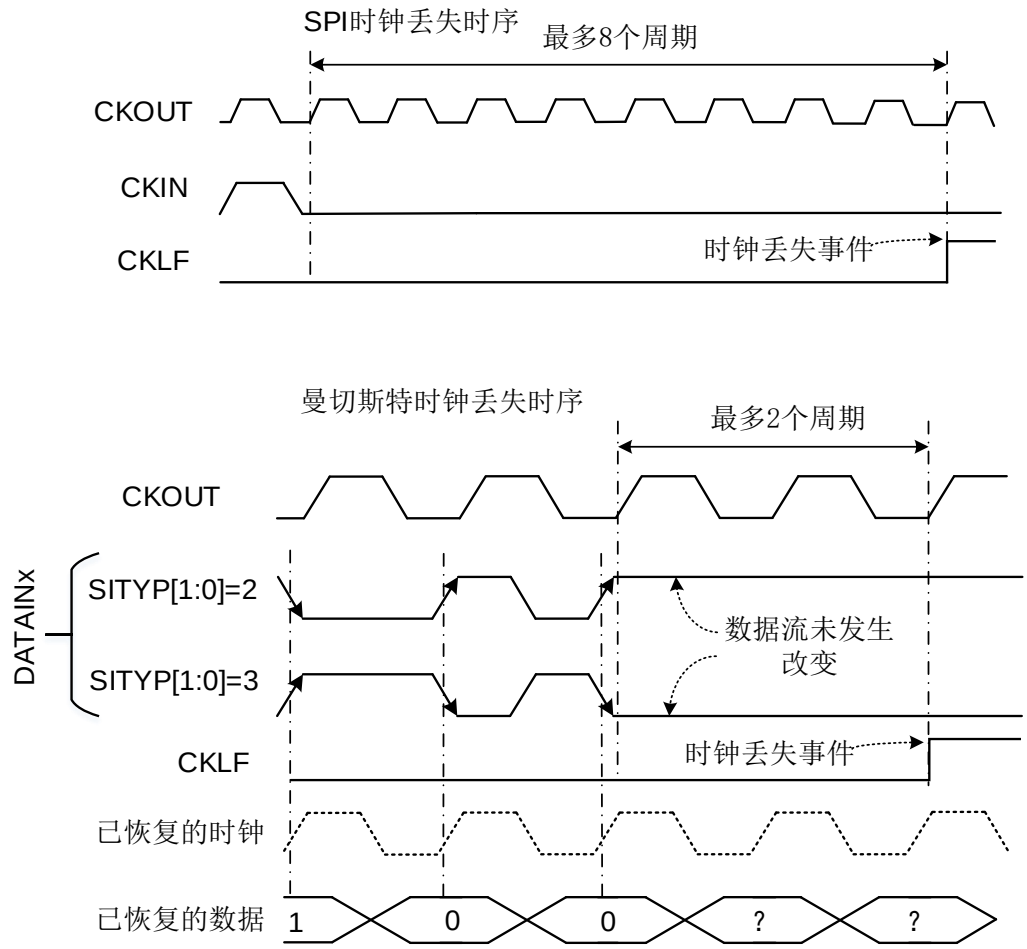
当通道的串行接口的收发器尚未被同步时，通道的时钟丢失标志位被置 1，且无法将相应的时钟丢失标志位清零。所以正确的使用时钟丢失功能的步骤如下：

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 45-5. 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：

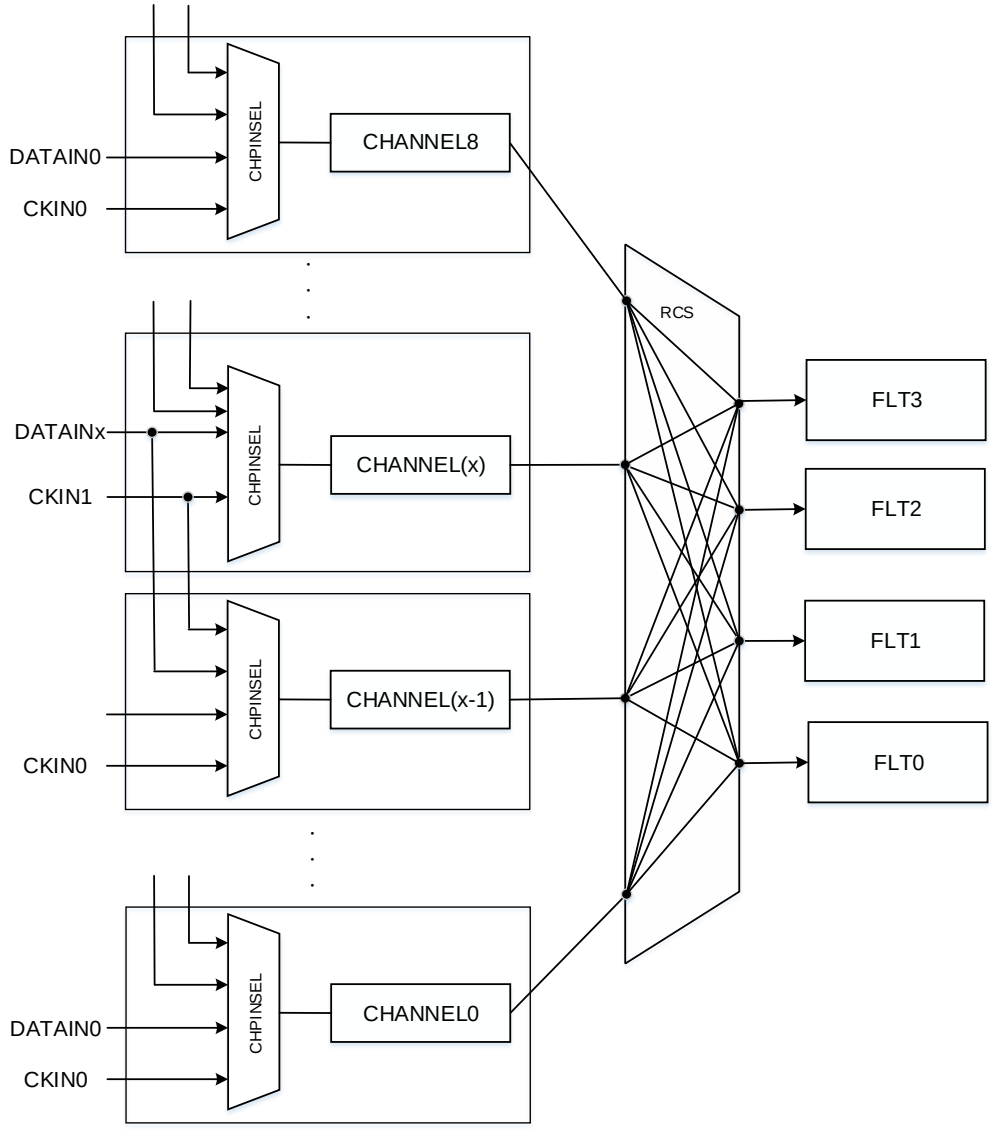
1. 选择 PDM 麦克风数据流输入的 HPDF 串行通道 1。
2. 将通道 1 的 HPDF_CHxCTL 寄存器中的 CHPINSEL 位写 0，通道 1 的输入引脚为自身引脚 DATAIN1 和 CKIN1。将 SITYP[1:0]=0，串行数据流在时钟信号的上升沿被采样，即通道 1 输入的为左通道数据。
3. 将通道 0 的 CHPINSEL 位置 1，通道 0 的输入引脚为引脚 DATAIN1 和 CKIN1。将

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 45-6. 通道引脚重定向*所示。

图 45-6. 通道引脚重定向



脉冲跳频

当 PCLK2 作为 HPDF 的系统时钟源时，可以使用脉冲跳频功能。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。

45.3.5. 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 45-4. 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 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	CH7 第一次	写保护	CH7 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
			采样	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。

45.3.6. 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。

45.3.7. 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 45-5. 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
HPDF_ITRG5	TIMER3_TRGO0
HPDF_ITRG6	TIMER15_CH1
HPDF_ITRG7	TIMER5_TRGO0
HPDF_ITRG8	TIMER6_TRGO0
HPDF_ITRG[9~10]	保留
HPDF_ITRG11	TIMER22_TRGO0
HPDF_ITRG12	TIMER23_TRGO0
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。

45.3.8. 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 45-6. $Sinc^X$ 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	$\pm x^2$	$\pm 2x^2$	$\pm x^3$	$\pm x^4$	$\pm x^5$
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。

45.3.9. 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 45-7. 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	±(2$SFOR^2$×IOR)
$Sinc^3$	±($SFOR^3$ ×IOR)
$Sinc^4$	±($SFOR^4$×IOR)
$Sinc^5$	±($SFOR^5$×IOR)

45.3.10. 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 45-8. 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 45-2. 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。

45.3.11. 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。

45.3.12. 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[1:0]位域，可使一个极值监视器采集多个通道的极值。

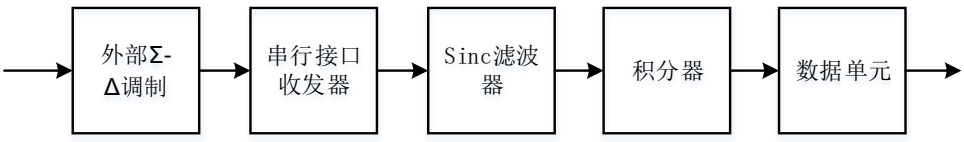
如果采集的最终输出数据字高于在极值监视器最大值寄存器中的值（HPDF_FLTyEMMAX 寄存器中的 MAXVAL[23:0]位），则用该寄存器的值被更新为当前的最终输出数据。如果采集的最终输出数据字小于在极值监视器最小值寄存器中的值（HPDF_FLTyEMMIN 寄存器中的 MINVAL[23:0]位），则该寄存器的值被更新为当前的最终输出数据。MAXDC 位和 MINDC 位的值分别指明了最大值/最小值来自哪个通道。

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。

45.3.13. 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 45-7. 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 45-9. 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	$\frac{f_{DATA}}{SFOR\times(IOR-1+4)+(2+1)}$
	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。

45.3.14. 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

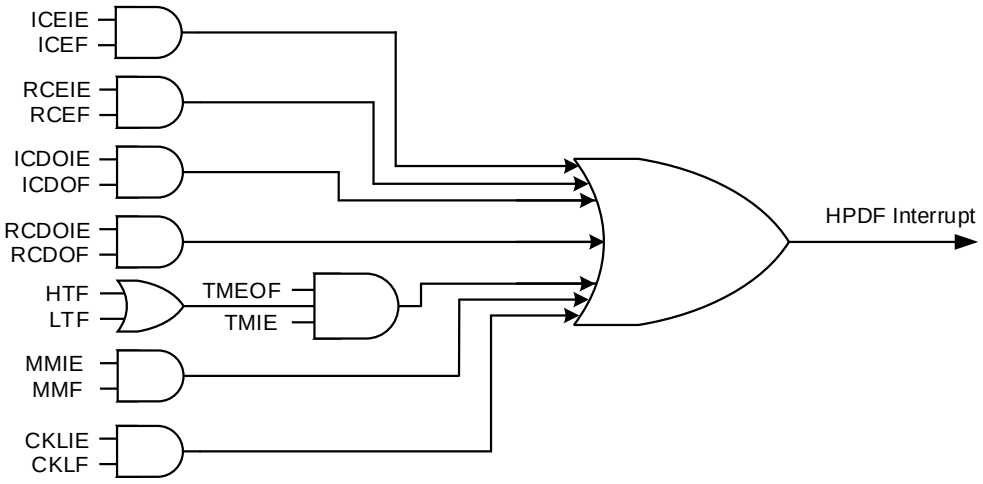
道时钟丢失中断事件。具体的中断事件描述如*表 45-10. HPDF 中断事件*所示。

表 45-10. 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 HTF[7:0] LTF[7:0]	阈值监视器事件	写 1 到 HTFC[7:0]位域 写 1 到 LTFC[7:0]位域	TMIE
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 45-8. HPDF 中断逻辑图*所示。

图 45-8. HPDF 中断逻辑图



45.4. HPDF 寄存器

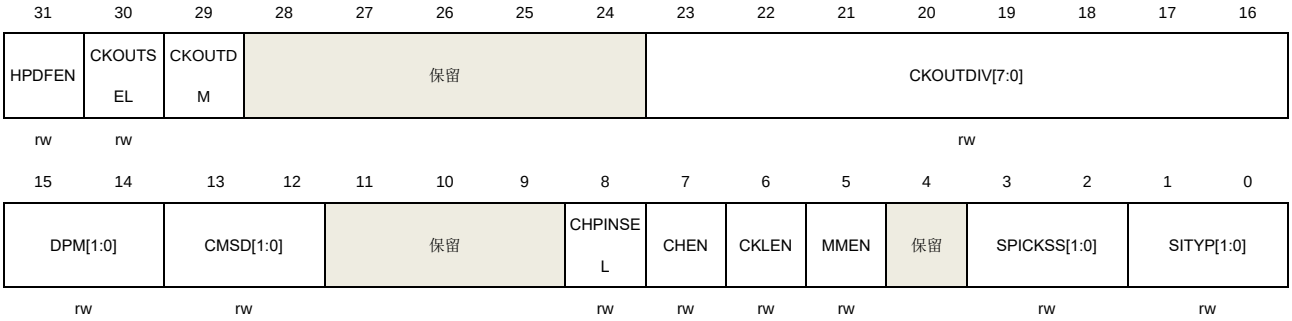
HPDF访问基地址：0x4001 7000

45.4.1. 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 ADCx 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

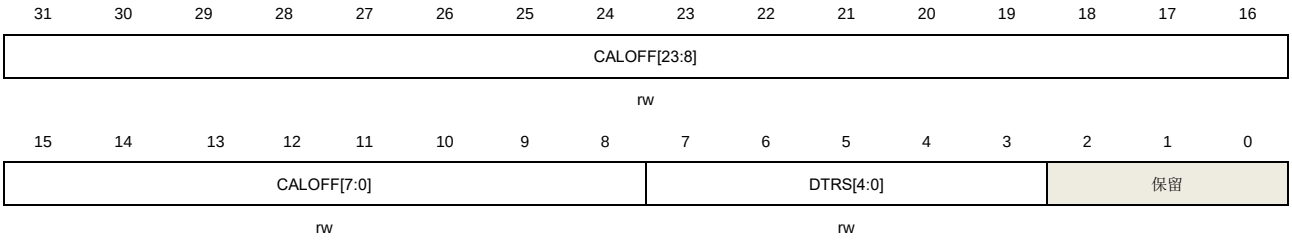
上升降沿 SPI 接口时钟源的详细介绍请参考*表 45-3. SPI 接口时钟配置*。
只有当 CHEN=0 时，此位域才能被修改。

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。

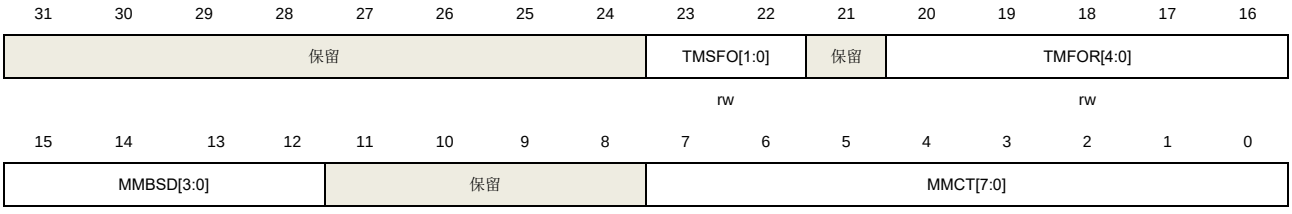


位/位域	名称	描述
31:8	CALOFF[23:0]	24 位偏移校正 通道的每一次转换数据后必须执行偏移校正。 此位域值由软件写入
7:3	DTRS[4:0]	数据右移位数 0~31：该值表示数据执行右移的位数 数据移位在偏移校正之间执行，数据移位将结果四舍五入到最接近的整数值，并保留符号位。 只有当 HPDF_CHxCTL 寄存器中的 CHEN=0 时，此位域才能被修改。
2:0	保留	必须保持复位值。

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。



rw rw

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。

31	30	29	28	27	26	25	24	23	22	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										PLSK[5:0]					
										rw					

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。

45.4.2. 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。

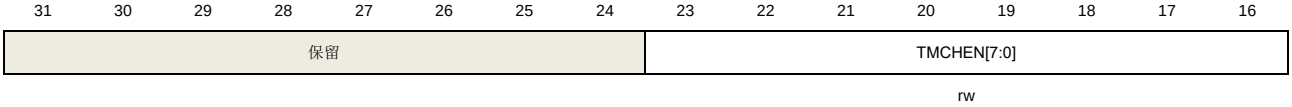
		只有当 FLTEN=0 时，此位才能被修改。
28:27	保留	必须保持复位值。
26:24	RCS[2:0]	选择规则转换通道 0：通道 0 作为规则转换通道 1：通道 1 作为规则转换通道 … 7：通道 7 作为规则转换通道 当 RCPF=1 时，写此位，被选中的通道在下一个规则转换开始转换。
23:22	保留	必须保持复位值。
21	RCDMAEN	使能读取规则转换数据的 DMA 0：禁止 DMA 读取规则转换数据 1：使能 DMA 读取规则转换数据 只有当 FLTEN=0 时，此位才能被修改。
20	保留	必须保持复位值。
19	RCSYN	同步启动规则转换 0：禁止同步启动规则转换 1：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：触发信号的没个上升沿产生启动注入转换请求 10：触发信号的没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。



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。

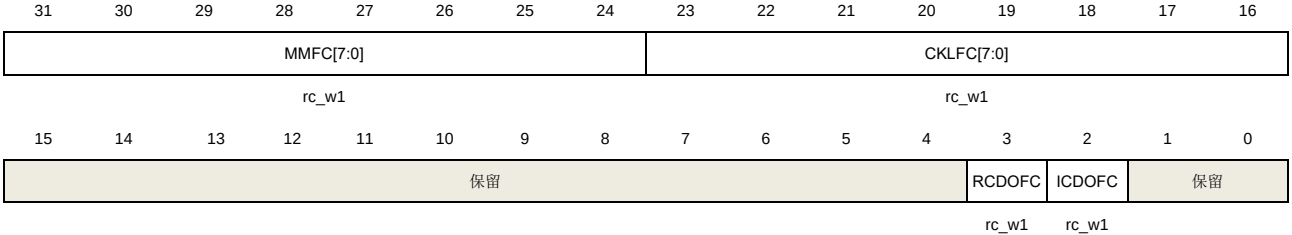
3	RCDOF	规则转换数据溢出标志 0：没有规则转换数据溢出产生 1：产生规则转换数据溢出 如果此位置位，表示规则转换已经完成，RCEF 也已经置位，FLTyRDATA 不受溢出影响。 此位由硬件置位，通过置位 HPDF_FLTyINTC 中的 RCDOFC 位，可清除此位。
2	ICDOF	注入转换溢出标志 0：没有注入转换数据溢出产生 1：产生注入转换数据溢出 如果此位置位，表示规则转换已经完成，ICEF 也已经置位，FLTyIDATA 不受溢出影响。 此位由硬件置位，通过置位 HPDF_FLTyINTC 中的 ICDOFC 位，可清除此位。
1	RCEF	规则转结束标志 0：未完成规则转换 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

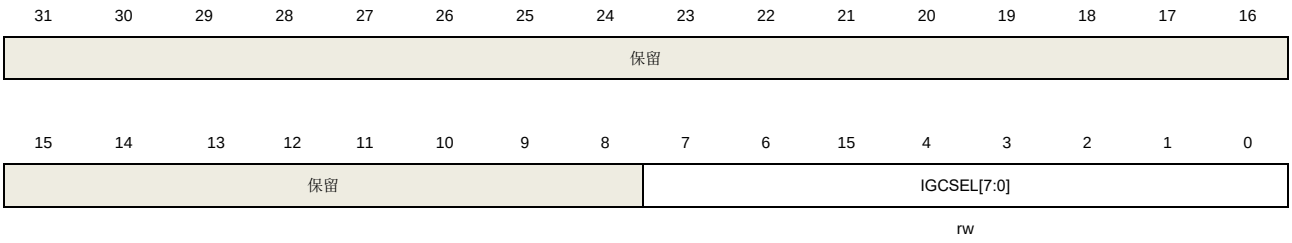
31:24	MMFC[7:0]	清除故障监视器标志 MMFC[x]=0：没有影响 MMFC[x]=1：清除通道 x 的故障监视器标志 此位仅在 HPDF_FLT0INTC 寄存器中有效（滤波器 y=0）
23:16	CKLFC[7:0]	清除时钟丢失标志 CKLFC[x]=0：没有影响 CKLFC[x]=1：清除通道 x 的时钟丢失标志 当串行接口尚未完成时钟同步，时钟丢失标志被置位，此时不能通过 CKLFC[7:0]清除时钟丢失标志。 此位仅在 HPDF_FLT0INTC 寄存器中有效（滤波器 y=0）
15:4	保留	必须保持复位值。
3	RCDOFC	清除规则转换数据溢出标志 0：没有影响 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。

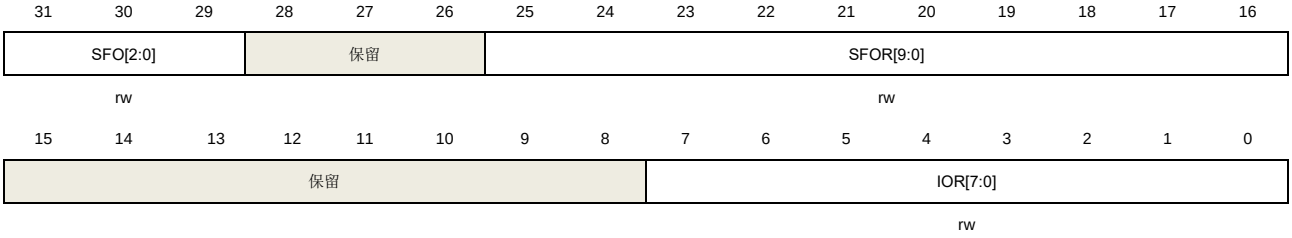


位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	IGCSEL[7:0]	注入组通道选择 ICGSEL[x]=0：通道 x 不属于注入组 ICGSEL[x]=1：通道 x 属于注入组 如果 SCMOD=1，由最小编号的通道开始，依次转换每一个所选通道。 如果 SCMOD=0，只转换其中一个所选通道，然后选择下一个通道。 当 SCMOD=0 时，对 IGCSEL[7:0]写操作将通道选择复位为最小编号通道。 注入组中必须至少有 1 个通道，所有将 IGCSEL[7:0]变为 0 的写操作都被忽略。

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。



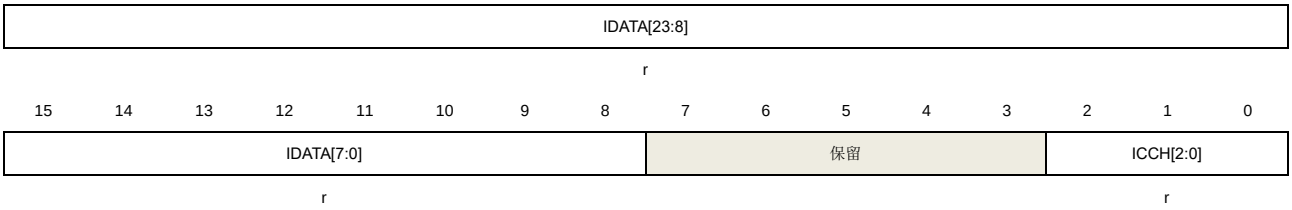
位/位域	名称	描述
31:29	SFO[2:0]	滤波器阶数 000：FastSinc 滤波器 001：Sinc^1 滤波器 010：Sinc^2 滤波器 011：Sinc^3 滤波器 100：Sinc^4 滤波器 101：Sinc^5 滤波器 110~111：保留 只有当 HPDF_FLTyCTL0 寄存器中的 FLTEN=0 时，此位域才能被修改。
28:26	保留	必须保持复位值。
25:16	SFOR[9:0]	Sinc 滤波器过采样率（抽取率） 0~1023：Sinc 滤波器过采样率 SFOR= SFOR[9:0] +1 如果 SFOR[9:0]=0，即过采样率为 SFOR=1，表示滤波器被旁路 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。



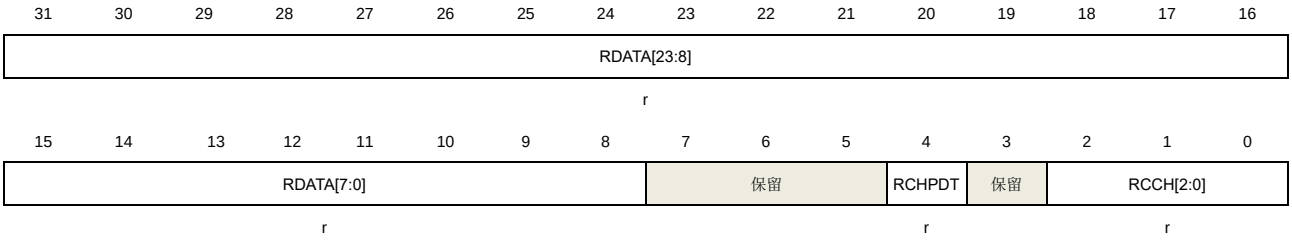
位/位域	名称	描述
31:8	IDATA[23:0]	注入通道转换数据 当每个注入中的一个通道转换完成，数据被保存在此位域。 当 ICEF=1 时，转换数据为有效的。读此寄存器清除 RCEF 位。
7:3	保留	必须保持复位值。
2:0	ICCH[2:0]	最近转换的注入通道 每个注入组通道转换完成时，ICCH[2:0]被更新，指示哪个通道完成了规则转换。故 IDATA[23:0]中的数据对应为该通道的值。

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。

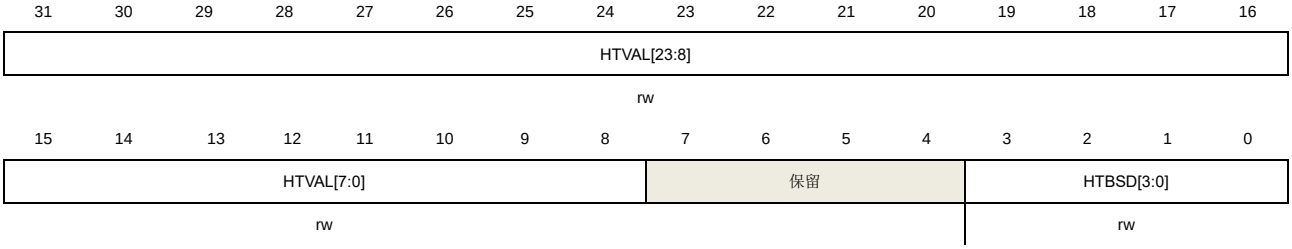


位/位域	名称	描述
31:8	RDATA[23:0]	规则通道转换数据 当每个规则转换完成，数据被保存在此位域。当 RCEF=1 时，转换数据为有效的。读此寄存器清除 RCEF 位。
7:5	保留	必须保持复位值。
4	RCHPDT	规则通道等待处理数据 在规则转换期间，被注入转换请求中断，导致 RDATA[23:0]中的规则转换数据被延迟处理。
3	保留	必须保持复位值。
2:0	RCCH[2:0]	最近转换的规则通道 每个规则转换完成时，RCCH[2:0]被更新，指示哪个通道完成了规则转换。故 RDATA[23:0]中的数据对应为该通道的值。

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。

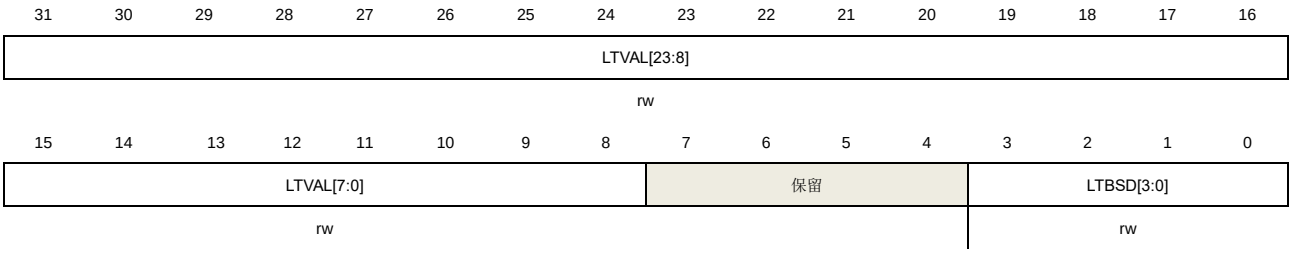


位/位域	名称	描述
31:8	HTVAL[23:0]	阈值监视器上限阈值 此位域通过软写入阈值监视器的上限阈值 在阈值监视器快速模式下（TMFM=1），此位域的高 16 位定义上限阈值，并与阈值监视器数据寄存器中的 TMDATA[15:0]值比较。此时 HTVAL[7:0]被忽略。
7:4	保留	必须保持复位值。
3:0	HTBSD[3:0]	上限阈值事件断路信号分配 HTBSD[i] = 0：断路信号 i 未分配到阈值监视器上限阈值事件。 HTBSD[i] = 1：断路信号 i 分配到阈值监视器上限阈值事件。

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

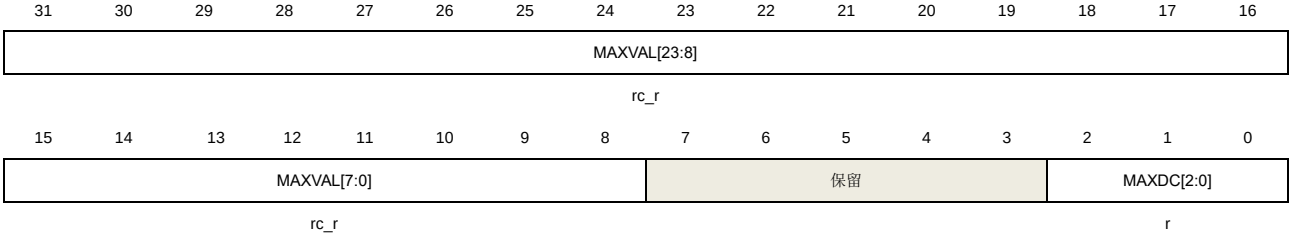
HTFC[x]=0：没有影响
HTFC[x]=1：清除通道 x 的阈值监视器上限阈值标志

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。

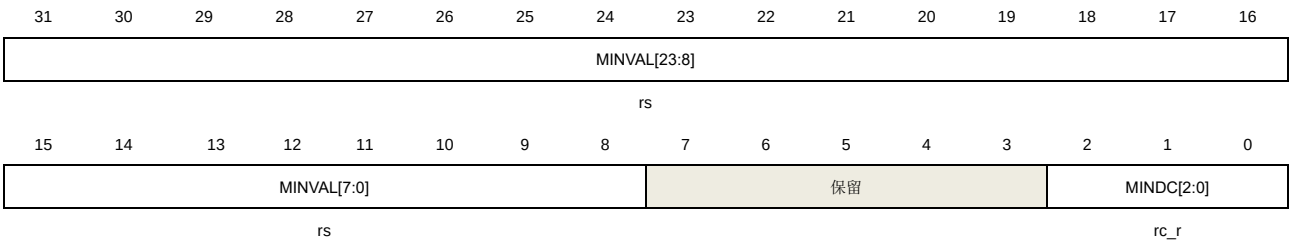


位/位域	名称	描述
31:8	MAXVAL[23:0]	极值监视器最大值 此位域通过硬件置位，表示 HPDF_FLTy 所转换的最大值。 读取该寄存器。此位域被复位。
7:3	保留	必须保持复位值。
2:0	MAXDC[2:0]	极值监视器最大值数据通道 该位域表示哪个通道的值被保存在 MAXVAL[23:0]位域中。 读取该寄存器此位被清 0

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。



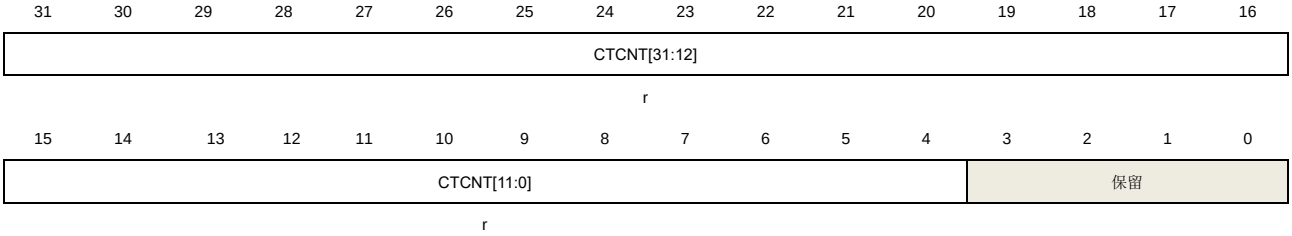
位/位域	名称	描述
31:8	MINVAL[23:0]	极值监视器最小值 此位域通过硬件置位，表示 HPDF_FLTy 所转换的最小值。

		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	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] / 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。

46. 实时解密（RTDEC）（适用于 GD32H737_757_759）

46.1. 简介

根据读请求地址的信息，实时解密（RTDEC）模块可以进行实时解密。RTDEC可以配置四个独立、不同的加密区域。每个区域都可被选择配置为只执行或从不执行。

为了实现良好的实时性能，RTDEC使用AES-128加密算法的计数器（CTR）模式。由于RTDEC在CTR模式下使用AES-128，当一个加密区域的数据或代码发生更改时，这个区域必须使用一个新的加密上下文（密钥或初始化向量）重新加密。RTDEC的这个特性决定了RTDEC仅适用于解密只读内容，如存储在外部闪存中的内容。

注意：RTDEC与Octal-SPI（OSPI）一起使用时，需要使用存储器映射模式访问外部只读存储器。需要特别注意的是，MCU内存和RTDEC模块均遵循小端表示法，而AES硬件加速器则遵循大端表示法。

46.2. 主要特性

- 软件可配置的加密区域多达4个；
- RTDEC区域中的粒度为4096字节；
- 区域配置可上锁；
- 可选的解密模式：只执行或从不执行；
- 每个区域都可配置独立的128位密钥、16位区域固件版本和64位应用程序定义的随机数。用户每次执行加密时，必须更改其中的一项或多项；
- 加密密钥的机密性和完整性保护；
  - 128 位只写密钥寄存器，且具有软件锁定机制
  - 硬件自动计算的 8 位 CRC 作为公钥信息
- OSPI内存读操作时的实时解密；
  - CTR 模式下使用 AES-128
  - 支持深度为 4 的密钥流 FIFO
  - 支持各种读取大小
  - 读取的物理地址用于加密/解密
- 支持GD32 OSPI预取机制。

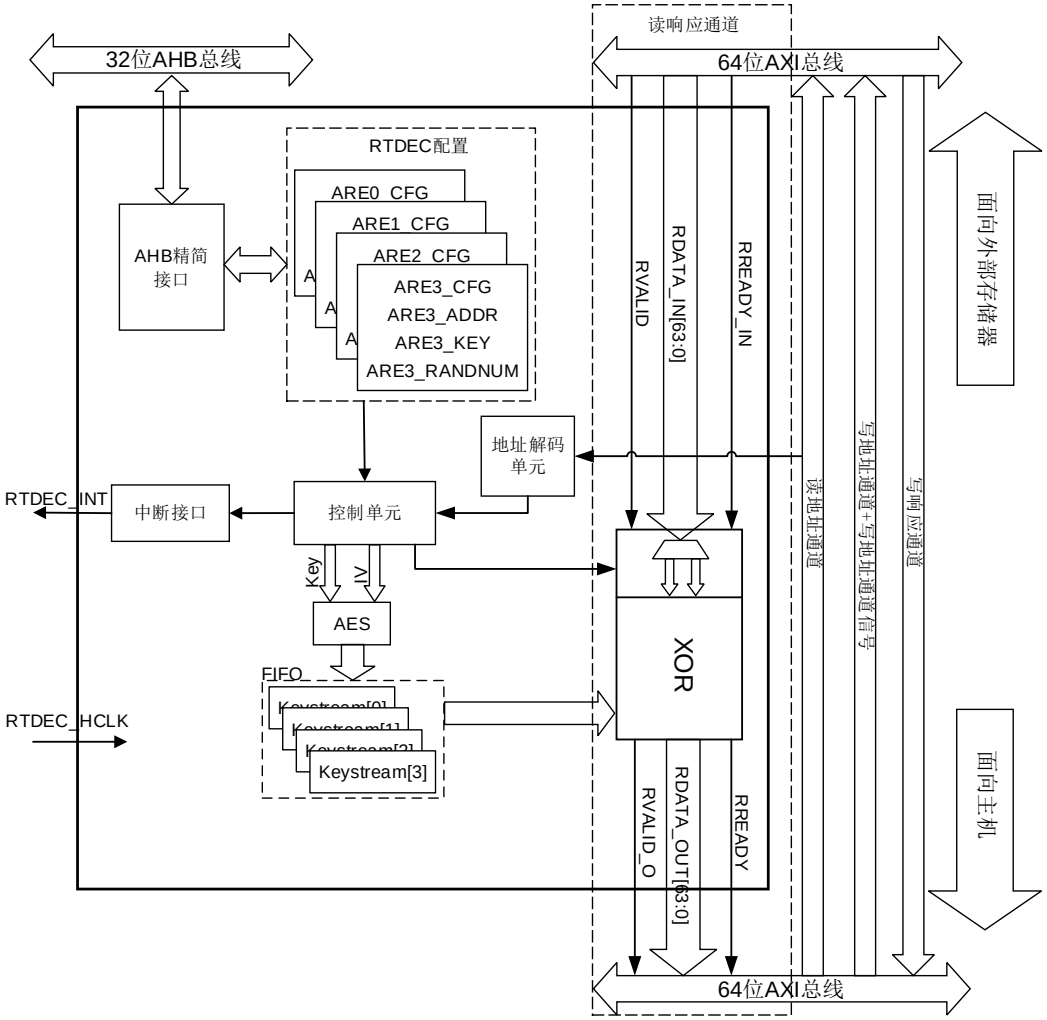
46.3. 功能说明

根据读取请求地址信息，RTDEC（实时解密）模块可以对Arm® AXI或AHB总线数据进行实时解密。RTDEC的最初目的是保护存储在外部SPI NOR Flash设备中的只读固件的机密性。RTDEC在OSPI存储器读操作期间进行实时解密。

46.3.1. 结构框图

*图 46-1. RTDEC 结构框图*给出了 RTDEC 的主要组成结构。

图 46-1. RTDEC 结构框图



注意：RTDEC_HCLK 是从 AHB 时钟输入的 RTDEC 时钟，RTDEC_INT 是 RTDEC 全局中断。

46.3.2. RTDEC 实时解密介绍

RTDEC 典型用例

RTDEC 的根本目的是保护只执行代码的机密性，只执行代码可以是从外部存储中执行的只执行固件。RTDEC 还可以保护只读的“代码+数据库”，以及在外部存储器中的只读和从不执行数据。

为了保护解密密钥的完整性，以及保护 RTDEC 的其他配置不受到软件拒绝服务攻击，RTDEC 提供了区域配置锁定机制。

RTDEC 与 OSPI 一起使用时，需要使用存储器映射模式访问外部只读存储器。

RTDEC 原理

RTDEC 原理是分析 MCU 和外设的所有读地址通道事务，这些读地址通道事务发生在 AXI 总线上，如中的 OSPI 控制器。

如果读取请求在 RTDEC 中编程的四个区域之一内，控制逻辑将触发密钥流计算。密钥流由 RTDEC 在计数器模式下使用 AES-128 计算。

RTDEC 编程区域之外的任何访问都属于非加密区域。RTDEC 区域定义的粒度为 4K 字节，也就是说，RTDEC 编程区域的大小是 4K 字节的整数倍。

每个 RTDEC 区域可以通过寄存器 AREx_CFG、AREx_SADDR、AREx_EADDR、AREx_NONCE 和 AREx_KEY 进行编程，其中 x=0~3。在寄存器 AREx_CFG 中，通过 MODE[1:0]位可定义区域是代码（只执行）、数据（从不执行），还是两个都是。

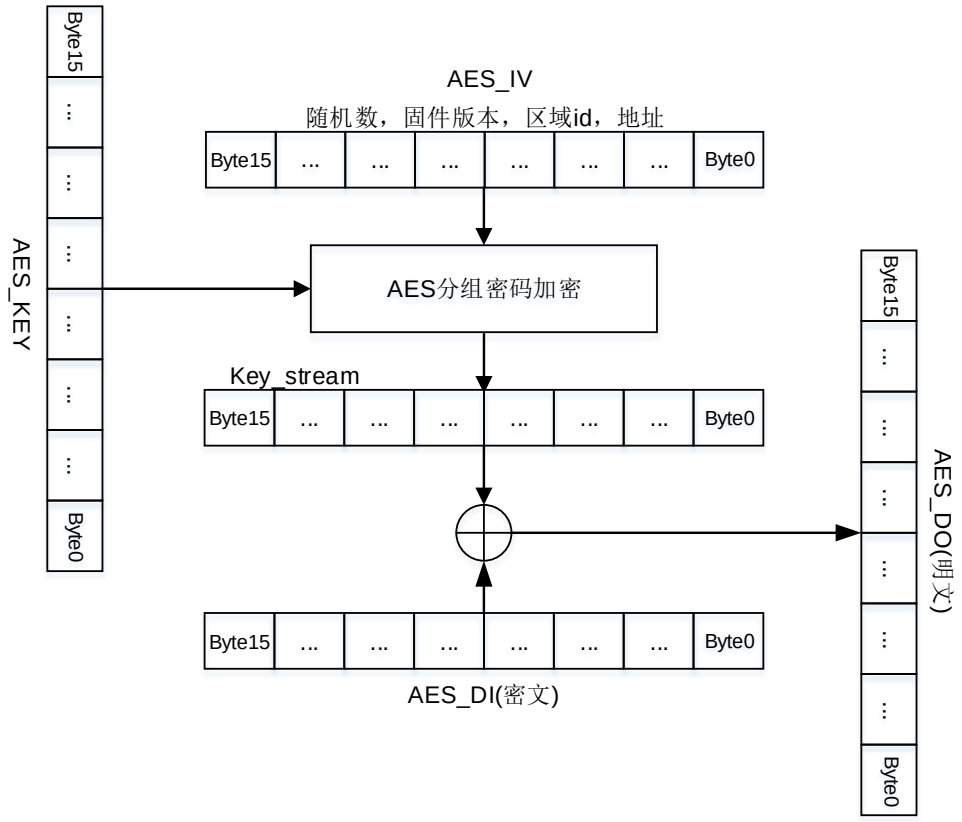
注意：虽然 RTDEC 不禁止区域重叠，但它是一种无效的编程，用户应该在应用软件中尽量避免。

当 RTDEC 解密增量或回卷突发时，它们不应跨越 4096 字节对齐的地址边界。RTDEC 无法解密 AXI 总线传输的部分数据（不完整数据）。

46.3.3. 计数器模式下使用 AES 解密

*图 46-2. CTR 下的 AES-128 解密流*表明了 RTDEC 如何在计数器模式下使用高级加密标准（AES）算法。此模式由 NIST 在 ***Special Publication 800-38A, Recommendation for Block Cipher Modes of Operation*** 中指定。

图 46-2. CTR 下的 AES-128 解密流



在解密过程中，使用 AES 块密码为每个 128 位数据块计算一个特定的密钥流信息。AES_IV 和 AES_KEY 定义如下：

1. 初始化向量 AES_IV[127:0] = AExNONCE[63:0] || 16'b0000 0000 0000 0000 || AExCFG[31:16] || 2'b00 || x || ReadAddress[31:4]。
2. 密钥 AES_KEY[127:0] = AExKEY3[31:0] || AExKEY2[31:0] || AExKEY1[31:0] || AExKEY0[31:0]。

注意：上述 x 为所选加密区域的区域 ID（x= 2'b 00 到 2'b11）

计算出 128 位密钥流后，将 128 位密文数据与对应的 128 位密钥流进行异或可得到 128 位明文信息。

128 位 AES_DIN 和 AES_DOUT 数据块的构造遵循以下规则：AES_Dx[127:0] = AXI_word(@+0x8)[63:0] || AXI_word(@)[63:0]，其中@是用于计算密钥流的十六进制地址。

当读取请求在非加密区域内，或在加密区域但未启用解密时，128 位 AXI 数据不会被更改。

46.3.4. RTDEC 配置

RTDEC 初始化过程

RTDEC 寄存器的可信初始化是使用 RTDEC 时的一个重要方面。因为它涉及密钥初始化和 MODE[1:0]位关键选项的初始化。

文中介绍两种 RTDEC 可信初始化方案。

注意：建议这些配置序列用于生产代码，因为在代码开发过程中，并不建议设置 ARE_K_LK 或 ARE_CFG_LK 位来锁定密钥或者去配置。

当 ARE_CFG_LK 位为 0 时，即使 ARE_EN 为已经置为 1，写入区域配置寄存器仍有效。

所有区域一个密钥解决方案

该解决方案中，用于解密 RTDEC 配置的 4 个保护区域的密钥属于同一个实体。

推荐的 RTDEC 配置序列如下：

- 将正确的MODE[1:0]值写入RTDEC_AREx_CFG寄存器（其中x=0~3）。
- 使用ARE_K_CRC中描述的序列对RTDEC_AREx_KEY0~3（其中x=0~3）寄存器进行编程，以确保生成有效的CRC。注意，密钥寄存器是只写寄存器，读该寄存器为非法操作。
- 检查密钥CRC。如果正常，将RTDEC_AREx_CFG寄存器中的ARE_K_LK位置位。区域x中的ARE_K_LK置位后不能被软件清除，即此区域中的密钥寄存器不能再被更改。其中x=0~3。
- 当你有一个需要解密的区域x时做以下操作。当这项工作进行时，不一定必须由实体（拥有解密密钥）执行。
  - 检查密钥 CRC 是否与存储在该区域中的加密二进制文件匹配。
  - 配置与加密二进制文件对应的详细信息，这些信息是随机数、固件版本、开始和结束地址。
  - 设置 RTDEC_AREx_CFG（其中 x=0~3）寄存器中的 ARE_EN 位，以使能该区域的解密。
  - 设置 RTDEC_AREx_CFG（其中 x=0~3）寄存器中的 ARE_CFG_LK 位，该位不能被软件清除，即区域配置不会再被更改。

注意：对于给定的区域，当 MODE[1:0]位改变时，密钥寄存器和相关的 CRC 会自动由硬件清零。因此，必须先写入 MODE[1:0] 位，然后再进行其他操作，并且在编程 RTDEC_AREx_KEY0~3（其中 x=0~3）寄存器后不得修改 MODE[1:0]位。

一区域一密钥解决方案

在该解决方案中，用于解密一个（或多个）保护区域的密钥可属于一个实体。建议遵循以下序列配置 RTDEC：

当你有一个需要解密的区域 x 时做以下操作，例如区域 0。当这项工作进行时，必须由拥有相应密钥的实体来执行。

- 将正确的MODE[1:0]值写入RTDEC_ARE0_CFG。
- 使用ARE_K_CRC中描述的序列对RTDEC_ARE0_KEY0~3寄存器进行编程，以确保生成有效的CRC。注意，密钥寄存器是只写寄存器，读该寄存器为非法操作。
- 检查密钥CRC。如果正常，将RTDEC_ARE0_CFG寄存器中的ARE_K_LK位置位。区域x中的ARE_K_LK置位后不能被软件清除，即此区域中的密钥寄存器不能再被更改。
- 配置与受保护固件对应的详细信息，这些信息是随机数、固件版本、开始和结束地址。

- 设置RTDEC_ARE0_CFG寄存器中的ARE_EN位，以使能该区域的解密。
- 设置RTDEC_ARE0_CFG寄存器中的ARE_CFG_LK位，该位不能被软件清除，即区域配置不会再被更改

注意：对于给定的区域，当 MODE[1:0]位改变时，密钥寄存器和相关的 CRC 会自动由硬件清零。因此，必须先配置 MODE[1:0]，再进行其他操作，且在编程 RTDEC_ARE0_KEY0~3 寄存器后，不得修改 MODE[1:0]。

RTDEC 复位

RTDEC 每次复位时，必须按照章节 ***RTDEC 初始化过程***描述的正确的密钥加载序列执行初始化。在这种情况下，RTDEC_AREx_CFG 寄存器位 ARE_K_CRC 位为零。当 RTDEC 复位时，我们建议用户的应用软件来保证正确的密钥加载顺序。

46.3.5. RTDEC 错误管理

RTDEC 可以自动管理如下的错误：

- 非法读取RTDEC_AREx_KEY0~3寄存器；
- 当RTDEC_AREx_CFG寄存器中的ARE_CFG_LK或ARE_K_LK位置位时，非法写入RTDEC_AREx_KEY0~3寄存器。其中x=0~3；
- 当RTDEC_AREx_CFG寄存器中的ARE_CFG_LK位置位时，非法写入RTDEC_AREx_CFG，RTDEC_AREx_SADDR，RTDEC_AREx_EADDR或RTDEC_AREx_NONCE0~1寄存器，其中x=0~3；
- 当MODE[1:0]=2'b00时，非法读取只执行区域。当MODE[1:0]=2'b01时，非法执行请求到从不执行区域。这些非法请求返回0，无总线错误；
- 密钥错误：当中止事件发生，密钥寄存器将被清除。此时如果有一些读请求，这些请求将返回0，不会出现总线错误。中止事件可能篡改检测，未授权调试连接，不受信任的引导或SPC级别低至无保护降级。

如果 SECIE、ECONEIE 或 KEIE 位置位，则发生上述一个或多个错误将产生中断。关于中断信息，可参考 ***RTDEC 中断***。

注意：发生密钥错误后，必须重新初始化 RTDEC 密钥。如果寄存器被锁定，可能还需要复位 RTDEC。

46.4. RTDEC 中断

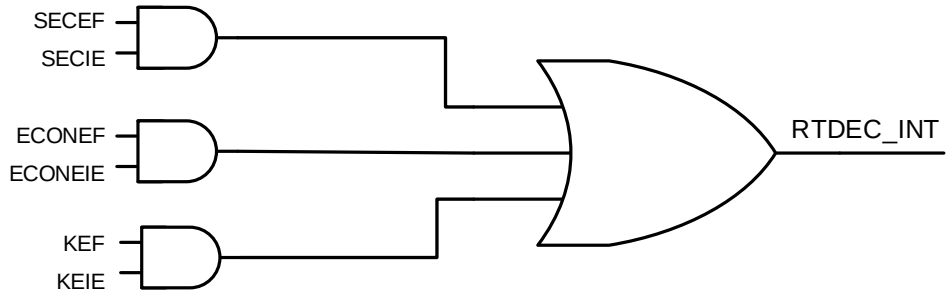
RTDEC 可产生三个独立的可屏蔽中断源，以下安全事件发生后中断标志将置位：

- 非法读取或写入密钥。当ARE_CFG_LK=1时，非法写入区域的配置寄存器。两者都将触发SECEF标志置位，参考***中断标志寄存器（RTDEC_INTF）***；
- 当MODE[1:0]为2'b00时，对只执行区域进行读访问。当MODE[1:0]为2’b01时，执行访问到从不执行区域（数据区）。两者都将触发ECONEF标志置位，参考***中断标志寄存器（RTDEC_INTF）***；
- 密钥错误（加密区域读为零），触发KEF标志置位。参考***中断标志寄存器（RTDEC_INTF）***

在被发送到中断控制寄存器之前，所有的中断事件都进行了 OR 操作。因此 RTDEC 在任何给定时间只能向控制器发送单个中断请求。

*图 46-3. RTDEC 中断框图*表明了 RTDEC 模块中断连接。

图 46-3. RTDEC 中断框图



通过设置 RTDEC_INTEN 寄存器中相应的 SECIE、ECONEIE 或 KEIE 位，可以使能或禁能 RTDEC 中断源，如*表 46-1. RTDEC 中断请求*所示。中断事件的状态可在 RTDEC_INTF 寄存器中查询，可以使用 RTDEC_INTC 寄存器清除中断标志。

表 46-1. RTDEC 中断请求

中断事件	事件标志	控制使能位
安全错误	SECEF	SECIE
仅执行，从不执行错误	ECONEF	ECONEIE
密钥错误	KEF	KEIE

46.5. RTDEC 寄存器

RTDEC0基地址: 0x5200 B800

RTDEC1基地址: 0x5200 BC00

46.5.1. 区域 x 配置寄存器（RTDEC_AREx_CFG）

地址偏移：0x20 + 0x30 * x（x = 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	ARE_FMVER[15:0]	区域固件版本 在RTDEC_AREx_CFG寄存器中的ARE_EN位使能之前，需要先配置好区域固件版本。
15:8	ARE_K_CRC[7:0]	区域密钥的 8 位 CRC 当ARE_K_LK = 0时，如果用户按照KEY0 -> KEY1 -> KEY2 -> KEY3的顺序加载该区域的密钥，则 ARE_K_CRC[7:0]将由硬件自动计算。当启动新的有效序列时，立即开始新的计算。直到有效序列完成前，取ARE_K_CRC[7:0]值都将为0。 当ARE_K_LK = 1时，直到下一次复位前，ARE_K_CRC值保持不变。 CRC 计算采用标准 CRC-8-CCITT 算法 X8 + X2 + X + 1，计算结果为 8 位校验和。 这些位是只读位。 **注意**：当密钥的最后一位写入后，才会更新 CRC 信息。
7:6	保留	必须保持复位值
5:4	MODE[1:0]	RTDEC 模式位 这些位配置此区域的 RTDEC 操作模式： 00：仅对代码访问进行解密。 01：仅解密数据访问。 10：所有读访问都被解密（代码或数据）。 11：保留。 当 MODE[1:0]位改变时，区域的密钥和相关的 CRC 被硬件自动清除。
3	保留	必须保持复位值
2	ARE_K_LK	区域密钥锁定位

		0：允许对 RTDEC_AREx_KEY 寄存器写入。 1：禁止对 RTDEC_AREx_KEY 寄存器写入，ARE_K_CRC[7:0]也被锁定。 该位只能由软件置位一次，且 RTDEC 复位之前无法清除。当该位被设置时，它禁止对该区域密钥寄存器的写访问。
1	ARE_CFG_LK	区域配置锁定位 0：允许对 RTDEC_AREx_CFG、RTDEC_AREx_SADDR、RTDEC_AREx_EADDR、RTDEC_AREx_NONCE0~1、RTDEC_AREx_KEY0~3 寄存器的写入。 1：禁止对 RTDEC_AREx_CFG、RTDEC_AREx_SADDR、RTDEC_AREx_EADDR、RTDEC_AREx_NONCE0~1 、 RTDEC_AREx_KEY0~3 寄存器的写入。ARE_K_CRC[7:0]也被锁定 该位只能由软件置位一次，且 RTDEC 复位之前无法清除。当该位置 1 时，ARE_K_LK 位被强制为 1 时，它禁止对该区域配置、起始地址、结束地址和随机数寄存器的写访问。
0	ARE_EN	0：禁能区域实时解密。 1：使能区域实时解密。 在该位置位之前，区域配置信息必须有效。

46.5.2. 区域 x 起始地址寄存器（RTDEC_AREx_SADDR）

地址偏移：0x24 + 0x30 * x（x = 0~3）
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:0	ARE_SADDR[31:0]	区域 x 起始地址位 必须在RTDEC_AREx_CFG寄存器中的ARE_EN位置1前写入这些位。如果RTDEC_AREx_CFG寄存器中的ARE_CFG_LK位置1，则写入这些位无效。 **注意**：确定区域时，低12位（LSB）和高4位（MSB）将被忽略。读取该寄存器时，4个MSB位和12个LSB位返回0。

46.5.3. 区域 x 结束地址寄存器（RTDEC_AREx_EADDR）

地址偏移：0x28 + 0x30 * x（x = 0~3）
复位值：0x0000 0FFF

该寄存器只能按字（32位）访问。

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

ARE_EADDR[31:16]
rw

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

位/位域	名称	描述
31:0	ARE_EADDR[31:0]	区域 x 结束地址位 必须在RTDEC_AREx_CFG寄存器中的ARE_EN位置1前写入这些位，且ARE_EADDR不能小于ARE_SADDR。 如果RTDEC_AREx_CFG寄存器中的ARE_CFG_LK位置1，则写入这些位无效。 **注意**：确定区域时，低12位（LSB）和高4位（MSB）将被忽略。读取该寄存器时，4个MSB位和12个LSB位返回1。

46.5.4. 区域 x 随机数寄存器 0（RTDEC_AREx_NONCE0）

地址偏移：0x2C + 0x30 * x（x = 0~3）
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARE_NONCE0[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARE_NONCE0[15:0]															
rw															

位/位域	名称	描述
31:0	ARE_NONCE0[31:0]	区域 x 随机数寄存器位，对应 ARE_NONCE[31:0]。 必须在RTDEC_AREx_CFG寄存器中的ARE_EN位置1前写入这些位。 如果RTDEC_AREx_CFG寄存器中的ARE_CFG_LK位置1，则写入这些位无效。

46.5.5. 区域 x 随机数寄存器 1（RTDEC_AREx_NONCE1）

地址偏移：0x30 + 0x30 * x（x = 0~3）
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARE_NONCE1[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARE_NONCE1[15:0]															
rw															

位/位域	名称	描述
31:0	ARE_NONCE1[31:0]	区域x随机数寄存器位。对应ARE_NONCE[63:32]。 参考ARE_NONCE0[31:0]描述

46.5.6. 区域 x 秘钥寄存器 0（RTDEC_AREx_KEY0）

地址偏移：0x34 + 0x30 * x（x = 0~3）
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARE_KEY0[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARE_KEY0[15:0]															
w															

位/位域	名称	描述
31:0	ARE_KEY0[31:0]	区域 x 密钥位，ARE_KEY[31:0]. 必须在RTDEC_AREx_CFG寄存器中的ARE_EN位置1前写入这些位。 如果RTDEC_AREx_CFG寄存器中的ARE_CFG_LK位或ARE_K_LK位置1，则写入这些位无效。当读取这些位时，返回0。 **注意**：当应用程序成功更改RTDEC_AREx_CFG寄存器中的MODE[1:0]位时，RTDEC_AREx_KEY0寄存器和相关的ARE_K_CRC将被擦除。

46.5.7. 区域 x 秘钥寄存器 1（RTDEC_AREx_KEY1）

地址偏移：0x38 + 0x30 * x（x = 0~3）
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARE_KEY1[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARE_KEY1[15:0]															
w															

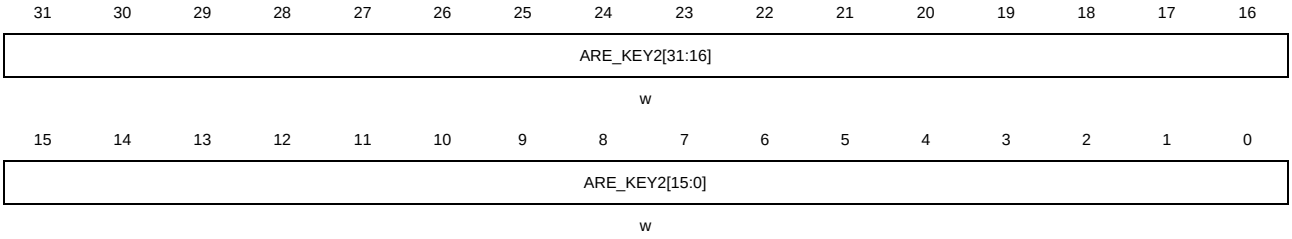
位/位域	名称	描述
31:0	ARE_KEY1[31:0]	区域 x 密钥位，ARE_KEY[63:32]。 参考ARE_KEY0[31:0]描述

46.5.8. 区域 x 秘钥寄存器 2（RTDEC_AREx_KEY2）

地址偏移：0x3C + 0x30 * x（x = 0~3）

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



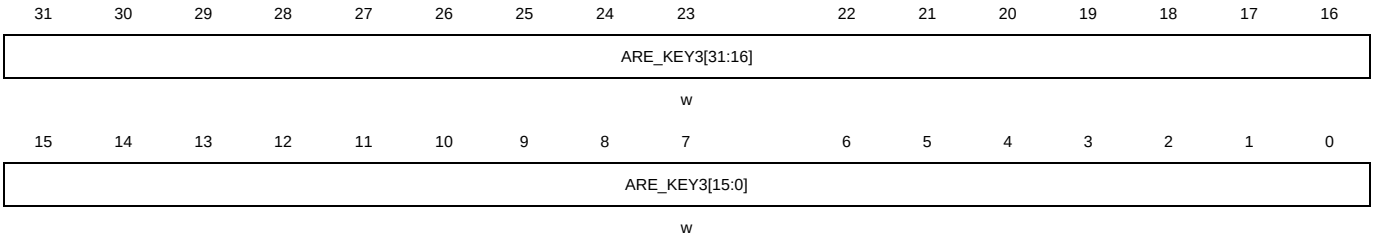
位/位域	名称	描述
31:0	ARE_KEY2[31:0]	区域 x 密钥位，对应 ARE_KEY[95:64]。 参考ARE_KEY0[31:0]描述

46.5.9. 区域 x 秘钥寄存器 3（RTDEC_AREx_KEY3）

地址偏移：0x40 + 0x30 * x（x = 0~3）

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



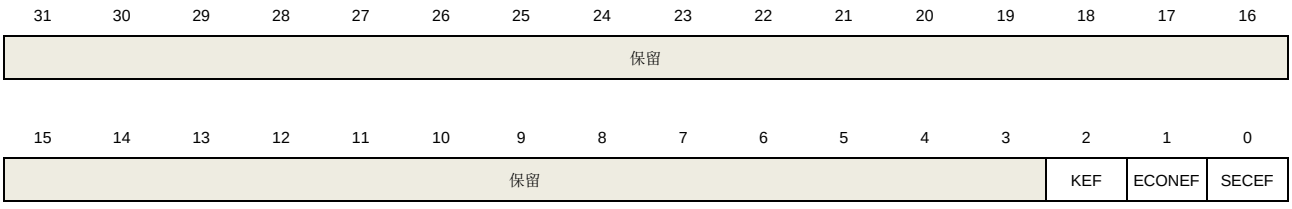
位/位域	名称	描述
31:0	ARE_KEY3[31:0]	区域 x 密钥位，对应 ARE_KEY[127:96]。 参考ARE_KEY0[31:0]描述

46.5.10. 中断标志寄存器（RTDEC_INTF）

地址偏移：0x300

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



r r r

位/位域	名称	描述
31:3	保留	必须保持复位值
2	KEF	密钥错误中断标志位 0：未检测到密钥错误 1：检测到密钥错误。中止事件后在使能的加密区域上检测到读取访问。如果RTDEC_INTEN寄存器中的KEIE位置1，则产生中断。 由于发生中止事件（篡改检测、未授权的调试器连接、不受信的引导、SPC级别低至无保护降级）重置密钥寄存器后，在加密区和非加密区上发生读访问时，该位由硬件置位。 该位通过向RTDEC_INTC寄存器中的KEC位写1清零。 KEF位置1后，后续对加密区域和非加密区域的读取都将返回0。直到再次初始化RTDEC密钥前，该位将一直保持置位。
1	ECONEF	只执行或从不执行错误中断标志 0：未检测到只执行错误或从不执行错误。 1：检测到只执行错误或从不执行错误。在MODE[1:0]设置为00的区域上检测到读取访问，或在MODE[1:0]设置为01的区域上检测到执行访问。如果RTDEC_INTEN寄存器中的ECONEIE位置1，则产生中断。 当在MODE[1:0]设置为00的任何加密区域上检测到读数据而不是取指令访问时，或当在MODE[1:0]设置为00的任何加密区域上检测到取指令而不是读数据访问时，该位都将由硬件置1。 通过向RTDEC_INTC寄存器中的ECONEC位写1清零。 **注意**：对于非法访问，RTDEC返回0。
0	SECEF	安全错误中断标志 0：未检测到安全错误 1：检测到安全错误。如果RTDEC_INTEN寄存器中的SECEIE位置1，则产生中断。 当检测到至少一个安全错误（非法访问密钥、配置锁定后的非法写入）时，由硬件置1。 通过向RTDEC_INTC寄存器中的SECEC位写1清零。

46.5.11. 中断标志清除寄存器（RTDEC_INTC）

地址偏移：0x304
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	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													KEC	ECONEC	SECEC

w w w

位/位域	名称	描述
31:3	保留	必须保持复位值
2	KEC	密钥错误标志位清除位 0：无影响 1：清除密钥错误标志 **注意**：RTDEC密钥寄存器应正确重新初始化，而不仅仅是清除KEF，以便能再次读取或执行任何加密区域。
1	ECONEC	只执行或从不执行错误标志清除位 0：无影响 1：清除只执行或从不执行错误标志
0	SECEC	安全错误标志清除位 0：无影响 1：清除安全错误标志

46.5.12. 中断使能寄存器（RTDEC_INTEN）

地址偏移：0x308
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	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													KEIE	ECONEIE	SECEIE
													rw	rw	rw

位/位域	名称	描述
31:3	保留	必须保持复位值
2	KEIE	密钥错误中断使能位 软件置位和清除 0：禁能密钥错误中断 1：使能密钥错误中断
1	ECONEIE	仅执行或从不执行错误中断使能位 软件置位和清除 0：禁能仅执行或从不执行错误中断 1：使能仅执行或从不执行错误中断
0	SECEIE	安全错误中断使能位 软件置位和清除

0：禁能安全错误中断

1：使能安全错误中断

47. 滤波算法加速器（FAC）

47.1. 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。

47.2. 主要特征

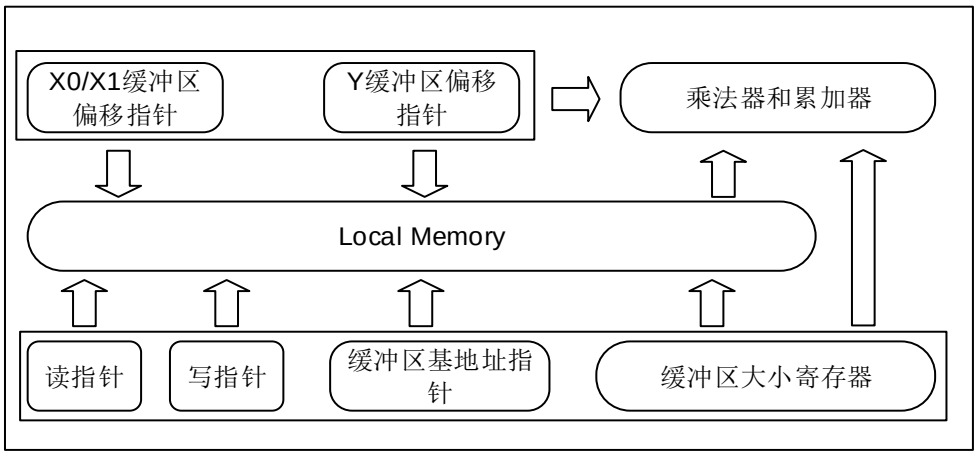
- 定点或浮点乘法器和累加器；
- 256 x 32位本地存储器；
- 16位定点和32位浮点输入端和输出端；
- 3个数据缓冲区，两个输入数据缓冲区一个输出数据缓冲区；
- 数据缓冲区可循环使用；
- 可实现FIR滤波和IIR滤波；
- 矢量函数支持卷积、点乘、相关函数；
- DMA读写数据通道。

47.3. 功能描述

47.3.1. 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 47-1. 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。

47.3.2. 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。

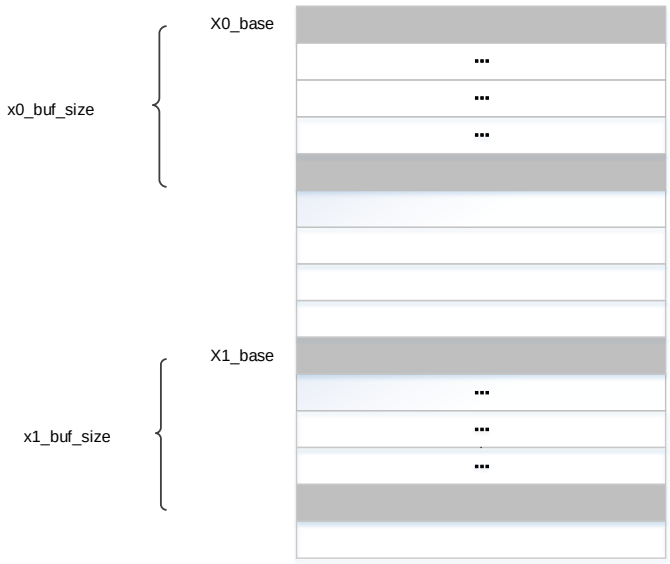
如果需要缓冲区循环使用，则可在缓冲区添加一个可选的净空区。此外，为了调节CPU或DMA的运行，有必要设置水印区级别。为了满足应用性能需求，净空区和水印区的数值需要配置。

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。

47.3.3. 输入缓冲区

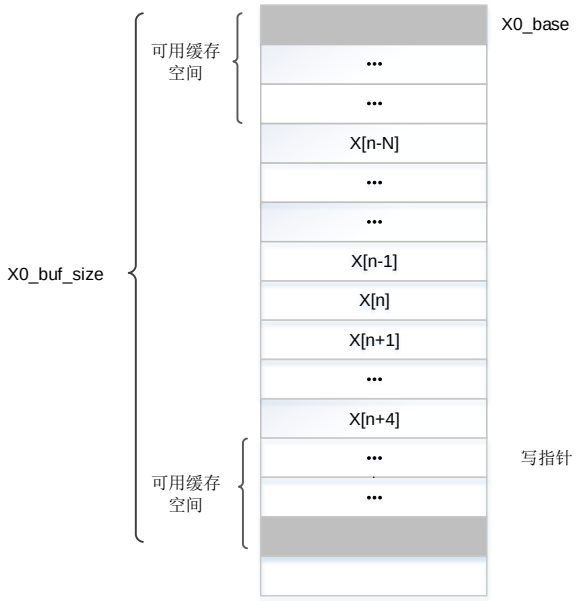
X0缓冲区和X1缓冲区用来保存输入到MAC的数据。每次乘法操作从X0缓冲区提取数值，从X1缓冲区提取数值，并将二者相乘。根据当前函数，控制单元的指针产生读地址偏移，该地址偏移是相对于缓冲区基地址。

图 47-2. 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 47-3. 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47-3. 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

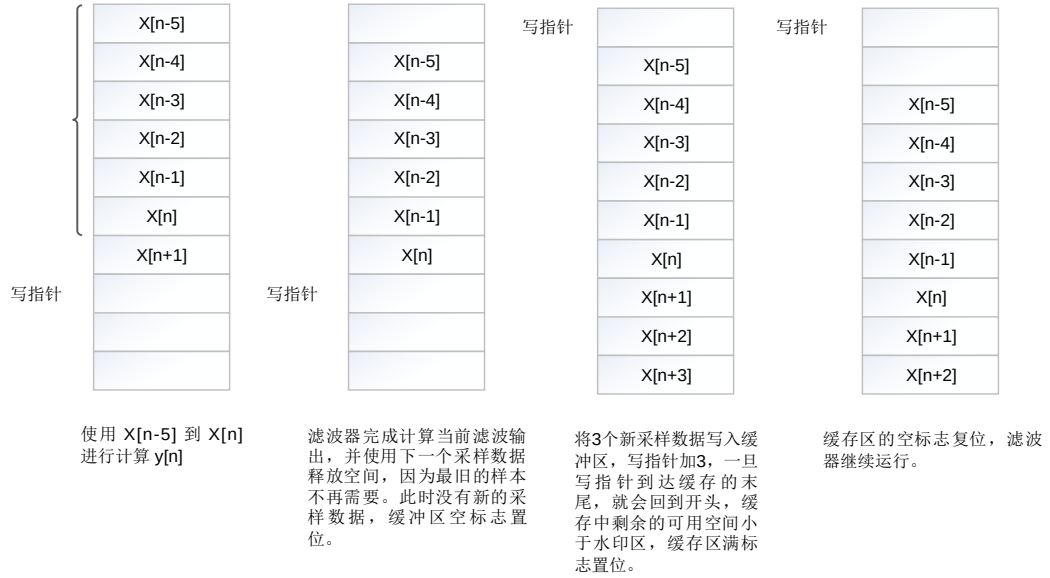
采样序列x[n-N+1]到x[n+1]，然后开始计算y[n+1]。输入序列丢弃最早的输入采样(x[n-N])，增加新的采样值(x[n+1])。

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47-4. 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 47-4. 循环输入缓冲区区域

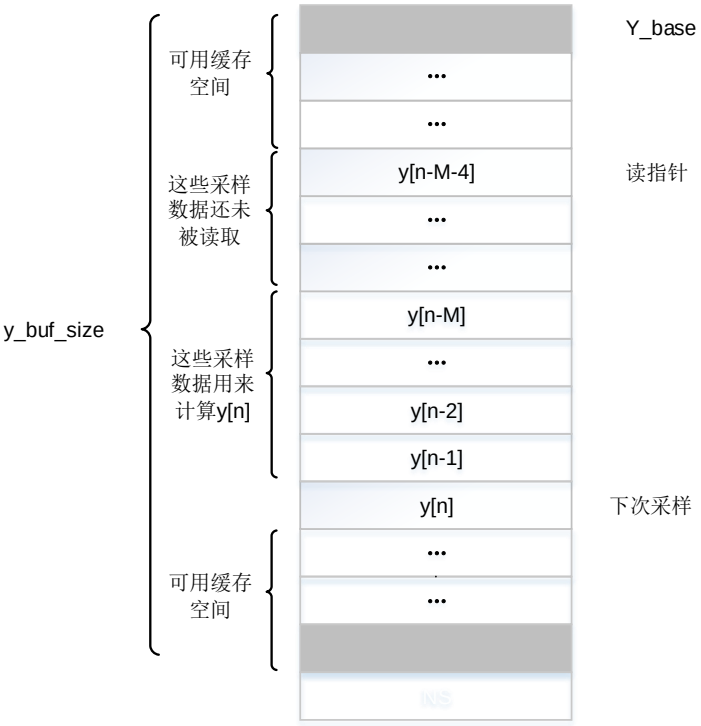


47.3.4. 输出缓冲区

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 47-5. 循环输出缓冲区



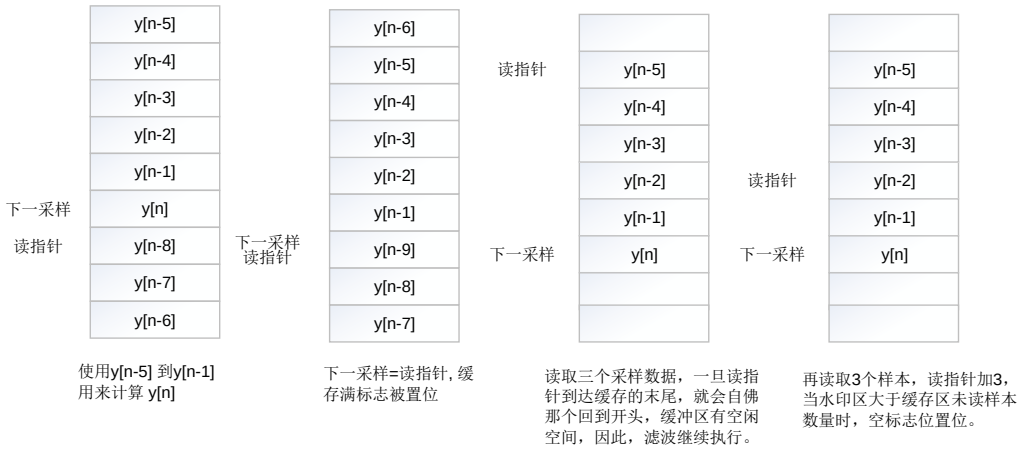
Y缓冲区也会工作在循环缓冲区模式。如果下一个输出数据的地址与读指针所指示的地址相等，缓冲区被标识为‘满’，然后滤波被中止，直至采样值被读取。

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47-6. 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 47-6. 循环输出缓冲区区域



47.3.5. 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。

47.3.6. 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。

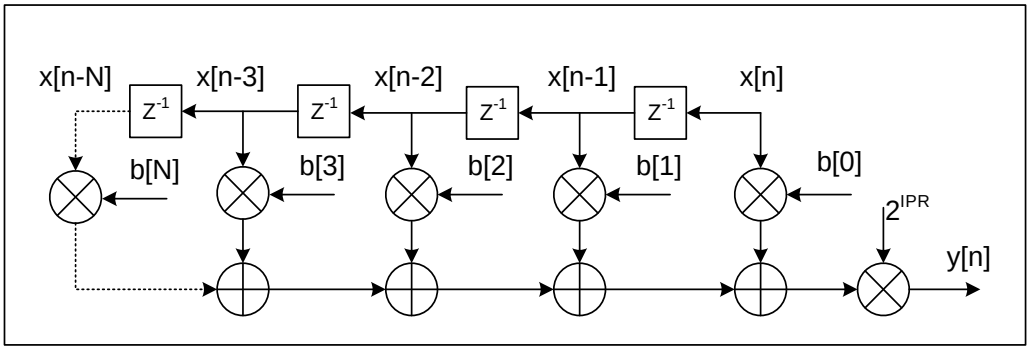
FIR filter: $\underline{Y} = \underline{B} * \underline{X}$

$$y_n = 2^{IPR} \sum_{k=0}^{N} (x_{n-k} b_k) \quad (47\text{-}1)$$

向量$\underline{B}$包含N+1个滤波系数，$\underline{X}$包含无限长的输入采样。$\underline{Y}$的元素按照点乘$y_n = \underline{B}.\underline{X}_n$计算得到，$\underline{X}_n = [x_{n-N},...,x_n]$，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47-7. FIR滤波器结构图*所示。

图 47-7. FIR 滤波器结构图



X0缓冲区是一个循环缓冲区

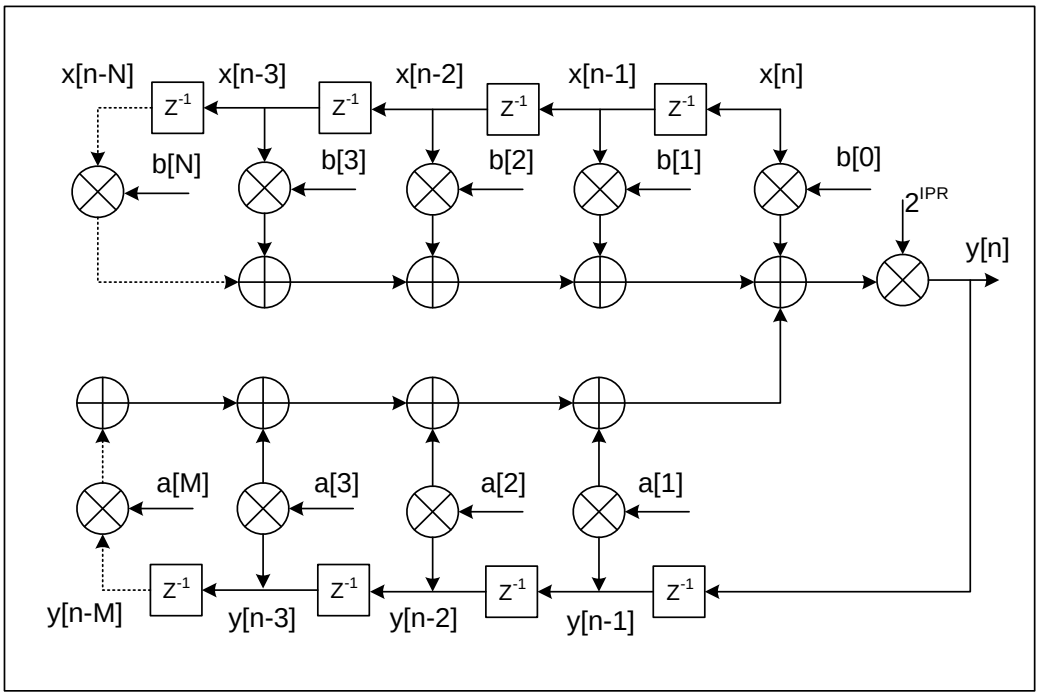
，由向量X的元素组成，缓冲区的长度是N+1+d，d是净空区长度，X1缓冲区由向量B的元素组成，缓冲区的长度是N+1，Y缓冲区是一个循环缓冲区，由输出值（y_n）组成，缓冲区的长度是d，IPP的长度是N+1，向量B在[2:127]之间，参数IPR是增益，乘以2^{IPR}得到最终的滤波输出结果，IPR的范围是[0:7]，IPQ未使用。

IIR filter: Y = B*X+A*Y

$$y_n=2^{IPR}\left(\sum_{k=0}^{N}(x_{n-k}\times b_k)+\sum_{k=1}^{M}\left(y_{n-k}\times a_k\right)\right) \quad (47\text{-}2)$$

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47-8. IIR滤波器结构图*所示。

图 47-8. 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]。

47.3.7. 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。

47.3.8. 浮点数据格式

如*表 47-1. 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 47-1. 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。

47.3.9. 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。

47.3.10. 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。

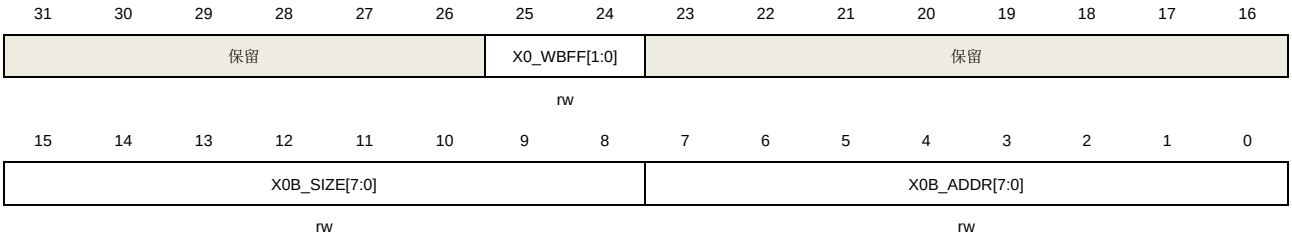
47.4. FAC 寄存器

FAC 基地址：0x4802 4800

47.4.1. 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。

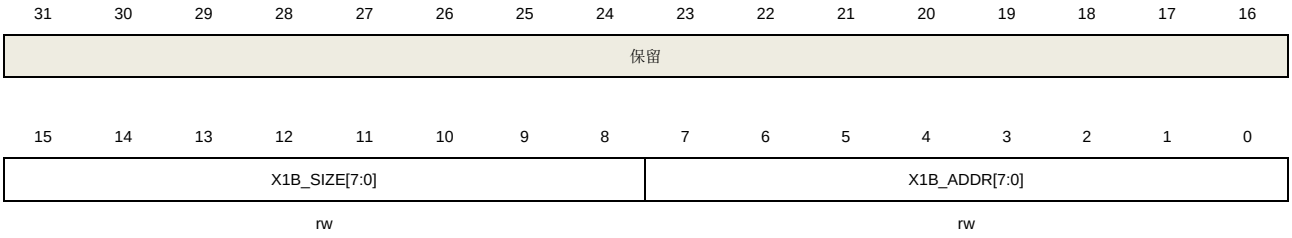


位/位域	名称	描述
31:26	保留	必须保持复位值。
25:24	X0_WBFF[1:0]	缓冲区水印区满标志位 如果缓冲区可用空间的数目少于 2^{X0_WBFF}，标志位置位 00：如果DMA写请求使能，阈值为1 01：阈值为2 10：阈值为4 11：阈值为8 如果若干数据在一次中断中传输到缓冲区，设置阈值大于1。
23:16	保留	必须保持复位值。
15:8	X0B_SIZE[7:0]	X0 缓冲区大小，滤波器前馈滤波系数数目。
7:0	X0B_ADDR[7:0]	X0 缓冲区基地址

47.4.2. 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。



位/位域	名称	描述

31:16	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。

47.4.3. 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。

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

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

47.4.4. 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。

47.4.5. 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。

47.4.6. 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															

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
					r	r	r							r	r

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。

47.4.7. 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	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	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。

47.4.8. 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	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	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。

48. 硬件信号量（HWSEM）（适用于 GD32H737_757_759）

48.1. 简介

硬件信号量（HWSEM）模块提供了一种非阻塞机制来保证不同进程之间的同步。HWSEM以原子方式实现了32个信号量，支持信号量的写锁定与读锁定，支持只有当总线主控ID和进程ID都与信号量锁信息相匹配时才解锁信号量。

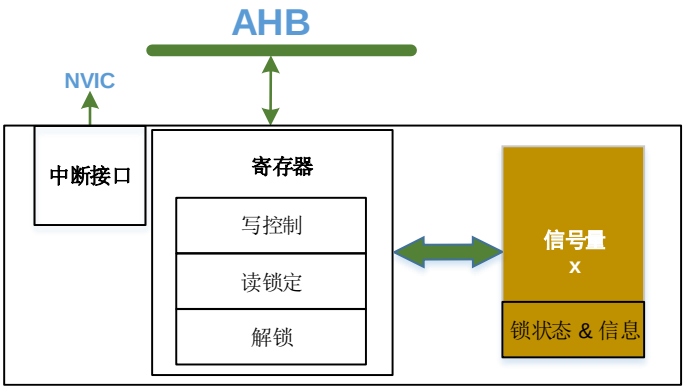
48.2. 主要特征

- 32个信号量；
- 当一个信号量解锁时会产生一个中断；
- 只有当MID[3:0]和PID[7:0]都与信号量锁信息相匹配时才解锁信号量。

48.3. 功能说明

48.3.1. 模块框图

图 48-1. HWSEM 框图



如*图48-1. HWSEM框图*所示，HWSEM模块包含三个子模块：

- 信号量 x 子模块，包含信号量锁状态和锁信息。
- 寄存器，用于访问信号量。
- 中断接口，用于中断控制和状态获取。

48.3.2. 信号量 x

锁状态

HWSEM_CTLx（x=0…31）寄存器中的LK位指示了信号量x的锁状态。

当LK位为0时，信号量未锁定。

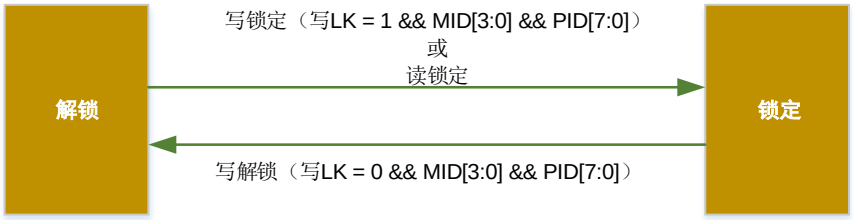
当LK位为1时，信号量已锁定。

锁信息

当信号量未锁定时，HWSEM_CTLx（x=0…31）寄存器中的MID[3:0]和PID[7:0]位值都为0。

当信号量已锁定时，HWSEM_CTLx（x=0…31）/ HWSEM_RLKx（x=0…31）寄存器中的MID[3:0]位值指示了锁定信号量的AHB总线主控，HWSEM_CTLx（x=0…31）/ HWSEM_RLKx（x=0…31）寄存器中的PID[7:0]位值指示了锁定信号量的进程。

图 48-2. HWSEM 状态机



48.3.3. 锁定信号量

HWSEM支持两种信号量锁定方式：

- 写锁定；
- 读锁定。

注意：仅支持当 MID[3:0]为 0b1011 时，锁定与解锁信号量。

写锁定

当由写锁定操作锁定一个信号量时，HWSEM_CTLx（x=0…31）/ HWSEM_RLKx（x=0…31）寄存器中的MID[3:0]位为锁定信号量的AHB总线主控，HWSEM_CTLx（x=0…31）/ HWSEM_RLKx（x=0…31）寄存器中的PID[7:0]位是配置的锁定信号量的进程ID。

每个AHB总线进程都必须有唯一的PID[7:0]值。

写锁定操作按照以下步骤进行：

1. 设置LK位为1，配置HWSEM_CTLx（x=0…31）寄存器的MID[3:0]和PID[7:0]位以锁定信号量。
2. 回读HWSEM_CTLx（x=0…31）寄存器，确认MID[3:0]和PID[7:0]位值是否是配置的值。
3. 如果回读的HWSEM_CTLx（x=0…31）寄存器的MID[3:0]和PID[7:0]位值与配置的值相匹配，则信号量被该进程锁定。否则，信号量已被其他AHB总线主控或进程锁定，返回步骤1重试。

读锁定

当由读锁定操作锁定一个信号量时，HWSEM_CTLx（x=0…31）/ HWSEM_RLKx（x=0…31）寄存器中的MID[3:0]位为锁定信号量的AHB总线主控，而HWSEM_CTLx（x=0…31）/ HWSEM_RLKx（x=0…31）寄存器中的PID[7:0]位始终为0。

如果超过一个进程使用读锁定操作锁定同一个信号量，当信号量被其中一个进程读锁定了，其他进程也会读到信号量被锁定了，但无法区分是哪个进程锁定的。

读锁定操作按照以下步骤进行：

1. 读HWSEM_RLKx（x=0…31）寄存器以锁定信号量。
2. 如果读到的MID[3:0]值与读锁定操作的AHB总线主控ID相匹配，并且PID[7:0]值读为0，则信号被读锁定。如果信号量被其他总线主控锁定时（读到的MID[3:0]值与读锁定操作的AHB总线主控ID不匹配），或者被其他进程写锁定时（读取的PID[7:0]值不为0），返回步骤1重试。

48.3.4. 解锁信号量

当信号量被锁定时，可使用写解锁操作来解锁信号量。写解锁操作可以避免信号量被其他不具有解锁权限的进程解锁。

只有与信号量锁信息相匹配的MID[3:0]和PID[7:0]值才可解锁信号量。在有效的写解锁操作后，HWSEM_CTLx（x=0…31）/ HWSEM_RLKx（x=0…31）寄存器的LK位，MID[3:0]和PID[7:0]位将都清零，此时如果HWSEM_INTEN寄存器的SIEx位为1，将产生一个中断。

写解锁操作按照以下步骤进行：

1. 设置LK位为0，配置HWSEM_CTLx（x=0…31）寄存器的MID[3:0]和PID[7:0]位以解锁信号量。
2. 回读HWSEM_CTLx（x=0…31）寄存器，确认LK位为0。
3. 如果配置的HWSEM_CTLx（x=0…31）寄存器的MID[3:0]和PID[7:0]位值与信号量的锁信息相匹配，且写入的LK位为0，则信号量将被解锁。否则，将忽略写解锁操作，LK位将始终保持为1。

48.3.5. 解锁所有信号量

当AHB总线主控工作不正常时，可使用全解锁操作来清除所有信号量的锁状态和锁信息。

全解锁操作按照以下步骤进行：

1. 配置HWSEM_UNLK寄存器的KEY[15:0]和MID[3:0]位域以解锁所有信号量。
2. 读取所有HWSEM_CTLx（x=0…31）寄存器，确认所有的LK位均为0。
3. 如果配置的HWSEM_CTLx（x=0…31）寄存器的MID[3:0]位值与信号量的锁信息相匹配，且写入的LK位为0，则信号量将被解锁，当HWSEM_INTEN寄存器的SIEx位为1时，将产生一个中断。否则，将忽略全解锁操作，LK位将始终保持为1。

48.3.6. 中断

信号量的锁定与解锁操作应在用户进程中进行，而不能在中断服务程序中进行。推荐的锁定操作为尝试两次锁定信号量动作，只有当第二次锁定失效时，再去使能信号量中断并等待中断。

*图48-3. 第一次尝试锁定操作时成功*中表示当目标进程第一次尝试锁定信号量时，锁定成功，接着应当执行清除标志操作，此时无需使能信号量中断。

*图48-4. 第二次尝试锁定操作时成功*中表示当目标进程第一次尝试锁定信号量失败，而在第二次锁定操作之前，应当执行清除标志操作，接着第二次尝试锁定信号量时，锁定成功，在这种情况下，也无需使能信号量中断。

*图48-5. 第二次尝试锁定操作时失败*中表示当目标进程第一次尝试锁定信号量失败，在第二次锁定操作之前，应当执行清除标志操作，接着第二次尝试锁定信号量再次失败，在这种情况下，再使能信号量中断并等待中断，在中断到来后进行后续的尝试锁定操作。

图 48-3. 第一次尝试锁定操作时成功

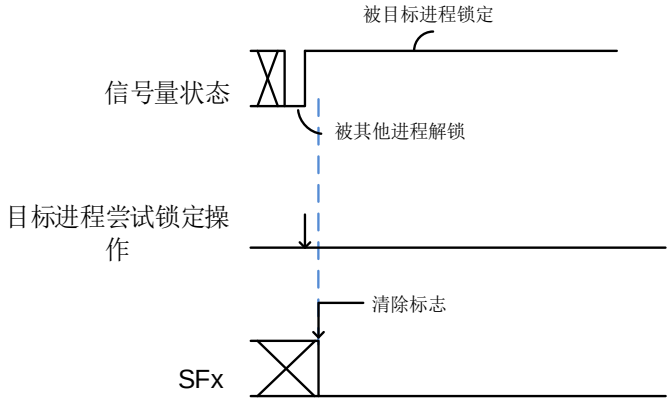


图 48-4. 第二次尝试锁定操作时成功

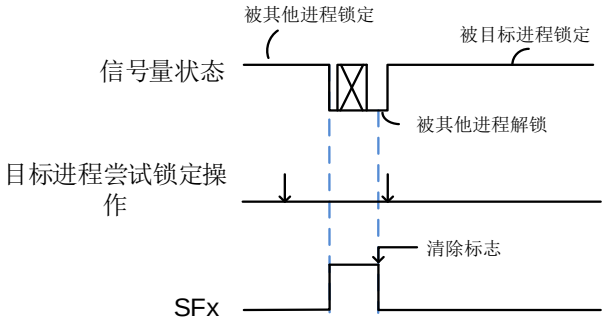
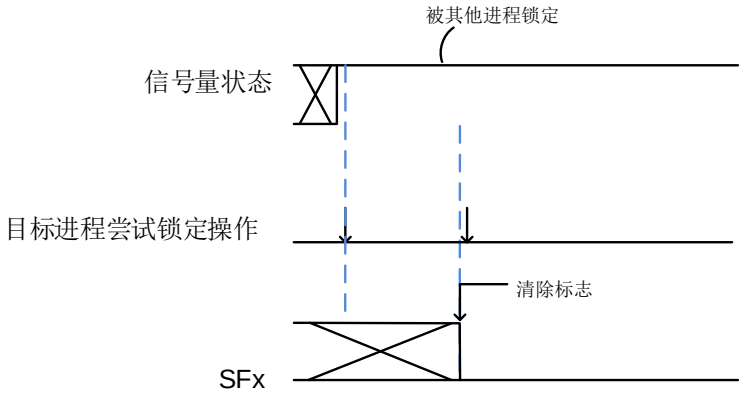


图 48-5. 第二次尝试锁定操作时失败



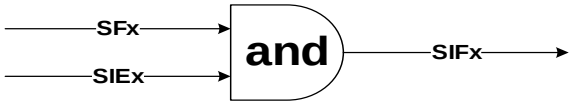
每个信号量均可产生一个信号量解锁中断。每个中断事件在HWSEM_INTF寄存器中专用的中断标志位，在HWSEM_INTC寄存器中有专用的清除标志位，在HWSEM_INTEN寄存器中有专用的中断使能位。*表48-1. 中断事件*描述了其对应关系。

表 48-1. 中断事件

中断事件	标志位	清除位	使能位
	HWSEM_INTF	HWSEM _INTC	HWSEM _INTEN
信号量 x 解锁	SIFx	SIFCx	SIEx

HWSEM 中断逻辑如*图 48-6. HWSEM 中断逻辑图*所示，中断使能时，相应中断事件会产生中断。

图 48-6. HWSEM 中断逻辑图



注意：“x”表示信号量序号（x=0...31）。

48.4. HWSEM 寄存器

HWSEM基地址：0x5802 6400

48.4.1. 控制寄存器（HWSEM_CTLx）（x=0…31）

地址偏移：0x00 + 0x4 * 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	LK	锁定信号 0：当MID[3:0]和PID[7:0]与信号量信息匹配时，解锁信号量 1：尝试锁定信号量
30:12	保留	必须保持复位值。
11:8	MID[3:0]	AHB总线主控ID 软件写入，只有当信号量未锁定，并且要写入的MID[3:0]值与写锁定操作的AHB总线主控相匹配时，要写入的MID[3:0]值才会写入该位域。 在写解锁操作中，当要写入的MID[3:0]值与写解锁操作的AHB总线主控相匹配时，该位域将清零。
7:0	PID[7:0]	进程ID 软件写入，只有当信号量未锁定时，才可通过写锁定操作将PID[7:0]值写入该位域。 在写解锁操作中，该位域将被清零。

48.4.2. 读锁定寄存器（HWSEM_RLKx）（x=0...31）

地址偏移：0x80 + 0x4 * x
复位值：0x0000 0000

HWSEM_RLKx（x=0...31）访问的物理地址与HWSEM_LKx（x=0...31）寄存器相同。

该寄存器只能按字（32位）访问。



保留	MID[3:0]	PID[7:0]
	r	r

位/位域	名称	描述
31	LK	读操作锁定信号 当通过匹配的AHB总线主控读该位时，将始终返回1，如果信号量在读操作之前是解锁状态，则此时信号量会被硬件读锁定，如果信号量在读操作之前是锁定状态，则此时该位不变。 0：信号量未锁定 1：信号量已锁定
30:12	保留	必须保持复位值。
11:8	MID[3:0]	总线主控ID 当通过匹配的AHB总线主控读该位域时，如果信号量在读操作之前是解锁状态，则此时AHB总线主控ID将被写入该位域，如果信号量在读操作之前是锁定状态，则此时将返回锁定该信号量的MID[3:0]。
7:0	PID[7:0]	进程ID 当通过匹配的AHB总线主控读该位域时，如果信号量在读操作之前是解锁状态，则此时该位域将被写为0且读为0，如果信号量在读操作之前是锁定状态，则此时将返回锁定该信号量的PID[7:0]。

48.4.3. 中断使能寄存器（HWSEM_INTEN）

地址偏移：0x100
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIE31	SIE30	SIE29	SIE28	SIE27	SIE26	SIE25	SIE24	SIE23	SIE22	SIE21	SIE20	SIE19	SIE18	SIE17	SIE16
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
SIE15	SIE14	SIE13	SIE12	SIE11	SIE10	SIE9	SIE8	SIE7	SIE6	SIE5	SIE4	SIE3	SIE2	SIE1	SIE0
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	SIEx	信号量中断使能位 0：禁能信号量中断 1：使能信号量中断

48.4.4. 中断状态清除寄存器（HWSEM_INTC）

地址偏移：0x104
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIFC31	SIFC30	SIFC29	SIFC28	SIFC27	SIFC26	SIFC25	SIFC24	SIFC23	SIFC22	SIFC21	SIFC20	SIFC19	SIFC18	SIFC17	SIFC16
w1	w1	w1	w1	w1	w1	w1	w1	w1	w1	w1	w1	w1	w1	w1	w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIFC15	SIFC14	SIFC13	SIFC12	SIFC11	SIFC10	SIFC9	SIFC8	SIFC7	SIFC6	SIFC5	SIFC4	SIFC3	SIFC2	SIFC1	SIFC0
w1	w1	w1	w1	w1	w1	w1	w1	w1	w1	w1	w1	w1	w1	w1	w1

位/位域	名称	描述
31:0	SIFCx	信号量中断标志清除位 软件写入，始终读为0。 0：无影响 1：清除信号量标志和中断标志

48.4.5. 状态寄存器（HWSEM_STAT）

地址偏移：0x108
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SF31	SF30	SF29	SF28	SF27	SF26	SF25	SF24	SF23	SF22	SF21	SF20	SF19	SF18	SF17	SF16
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SF15	SF14	SF13	SF12	SF11	SF10	SF9	SF8	SF7	SF6	SF5	SF4	SF3	SF2	SF1	SF0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:0	SFx	信号量标志 硬件置位，软件置位HWSEM_INTC寄存器的SIFCx位来清零该位。 0：无信号量解锁事件发生 1：发生了一个信号量解锁事件

48.4.6. 中断状态寄存器（HWSEM_INTF）

地址偏移：0x10C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIF31	SIF30	SIF29	SIF28	SIF27	SIF26	SIF25	SIF24	SIF23	SIF22	SIF21	SIF20	SIF19	SIF18	SIF17	SIF16
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIF15	SIF14	SIF13	SIF12	SIF11	SIF10	SIF9	SIF8	SIF7	SIF6	SIF5	SIF4	SIF3	SIF2	SIF1	SIF0

r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:0	SIFx	信号量中断标志位 硬件置位，软件置位HWSEM_INTC寄存器的SIFCx位来清零该位。 0：没有中断挂起 1：有一个中断挂起

48.4.7. 解锁寄存器（HWSEM_UNLK）

地址偏移：0x140
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:16	KEY[15:0]	清零密钥 只写，读为0。 只有当该位域值与HWSEM_KEY寄存器的KEY[15:0]值相匹配时，才可解锁所有与该寄存器的MID[3:0]相匹配的信号量。
15:12	保留	必须保持复位值。
11:8	MID[3:0]	要清零的信号量总线主控ID 只写，读为0。 指示将要解锁被该位域值的总线主控ID占有的所有信号量。
7:0	保留	必须保持复位值。

48.4.8. 键值寄存器（HWSEM_KEY）

地址偏移：0x144
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[15:0]															
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
31:16	KEY[15:0]	解锁键值，用于解锁某个总线主控ID占有的所有信号量。
15:0	保留	必须保持复位值。

49. 通用串行总线高速接口（USBHS）

49.1. 概述

USB高速（USBHS）控制器为便携式设备提供了一套USB互联解决方案。USBHS不仅支持主机模式和设备模式，也支持遵循HNP（主机协商协议）和SRP（会话请求协议）的OTG模式。USBHS包含了一个内部的USB PHY，可以配置成全速或高速，并且不再需要外部PHY芯片。对于GD32H75E，USBHS还为外部USB物理层（PHY）提供了一个ULPI接口。USBHS可以支持USB 2.0协议所定义的所有四种传输方式（控制传输、批量传输、中断传输和同步传输）。另外，在USBHS内部还有一个DMA引擎操作，可作为AHB总线主机在USBHS和系统之间加速数据传输。对于全速设备的操作，还支持电池充电检测（BCD）、附加检测协议（ADP）和链路层电源管理（LPM）。

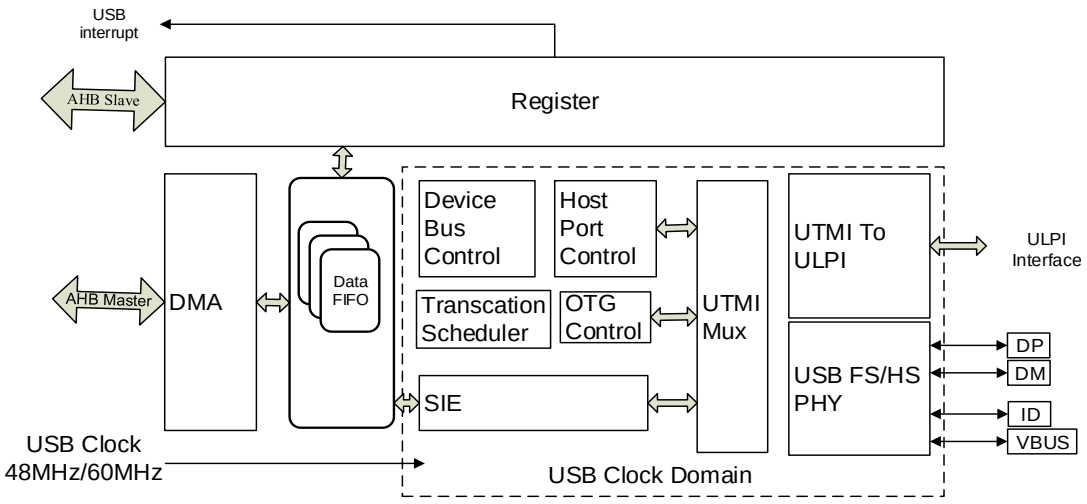
49.2. 主要特性

- 支持USB 2.0高速（480Mb/s）/全速（12Mb/s）/低速（1.5Mb/s）主机模式；
- 支持USB 2.0高速（480Mb/s）/全速（12Mb/s）设备模式；
- 支持遵循HNP（主机协商协议）和SRP（会话请求协议）的OTG协议；
- 支持所有的4种传输方式：控制传输、批量传输、中断传输和同步传输；
- 支持高带宽中断和同步传输；
- 在主机模式下，包含USB事务调度器，用于有效地处理USB事务请求；
- 包含一个4KB的FIFO RAM；
- 在主机模式下，支持16个通道；
- 在主机模式下，包含2个发送FIFO（周期性发送FIFO和非周期性发送FIFO）和1个接收FIFO（由所有的通道共享）；
- 在设备模式下，包含8个发送FIFO（每个IN端点一个发送FIFO）和1个接收FIFO（由所有的OUT端点共享）；
- 在主机模式下，若在高速模式下操作，支持PING协议；
- 在设备模式下，支持8个OUT端点和8个IN端点；
- 在设备模式下，支持远程唤醒功能；
- 包含一个支持USB OTG协议的USB PHY；
- 包含一个内部DMA调度器和引擎，每个应用请求都可在USBHS和系统之间执行数据拷贝；
- 在主机模式下，SOF的时间间隔可动态调节；
- 可将SOF脉冲输出到PAD；
- 可检测ID引脚电平和VBUS电压；
- 在主机模式或者OTG A设备模式下，需要外部部件为连接的USB设备提供电源；
- 支持1.2版电池充电规范中描述的电池充电检测（BCD）；
- 支持2.0版USB OTG补充协议中描述的附加检测协议（ADP）；
- 支持USB 2.0链路层电源管理附录和USB2.0工程变更通知单勘误表中描述的链路电源管理（LPM）。

注意：对于GD32H75E，同时支持片上PHY与连接外部高速PHY的ULPI接口。

49.3. 结构框图

图 49-1. USBHS 结构框图



该系列有两个USB HS模块（USB_HS0和USB_HS1），均支持ULPI接口，允许外部HS收发器高速传输USB的数据。对于GD32H75E， 仅USBHS0支持ULPI接口。

49.4. 信号线描述

表 49-1. USBHS 信号线描述

I/0 端口	类型	描述	注意
VBUS	输入	总线电源端口	仅内部 PHY 使用
DM	输入/输出	差分信号线-端口	仅内部 PHY 使用
DP	输入/输出	差分信号线+端口	仅内部 PHY 使用
ID	输入	USB 识别：微连接器识别接口	仅内部 PHY 使用
ULPI_D[7:0]	输入/输出	ULPI 数据线	外部 ULPI PHY 使用
ULPI_NXT	输入	ULPI 下个信号线	外部 ULPI PHY 使用
ULPI_DIR	输入	ULPI 方向	外部 ULPI PHY 使用
ULPI_STP	输出	ULPI 停止	外部 ULPI PHY 使用
ULPI_CLK	输入	ULPI 时钟	外部 ULPI PHY 使用

49.5. 功能描述

49.5.1. USBHS PHY 选择、时钟及工作模式

USBHS可以作为一个主机、一个设备或者一个DRD（双角色设备），并且支持两种连接类型：

内部嵌入式PHY和外部ULPI PHY。根据用户需求，应用程序可以选择两种连接类型的任何一种。

应用程序可以在主机模式下使用USBHS_HCTL寄存器内的SPDFSLS控制位和在设备模式下使用USBHS_DCFG寄存器内的DS[1:0]控制位将内部PHY和外部ULPI PHY的最大速度限制至全速。

表 49-2. USBHS 支持速度列表

寄存器配置		主机支持速度	设备支持速度
EMBPHY_FS=1 EMBPHY_HS=0 （内部FS PHY）		全速 低速	全速
EMBPHY_FS=0 EMBPHY_HS=1 （内部 HS PHY）	DS =01（设备模式） SPDFSLS=1（主机模式）	全速 低速	全速
	DS =00（设备模式） SPDFSLS=0（主机模式）	高速 全速 低速	高速 全速
EMBPHY_FS=0 EMBPHY_HS=0 （外部 ULPI PHY）	DS = 01（设备模式） SPDFSLS = 1（主机模式）	全速 低速	全速
	DS = 00（设备模式） SPDFSLS = 0（主机模式）	高速 全速 低速	高速 全速

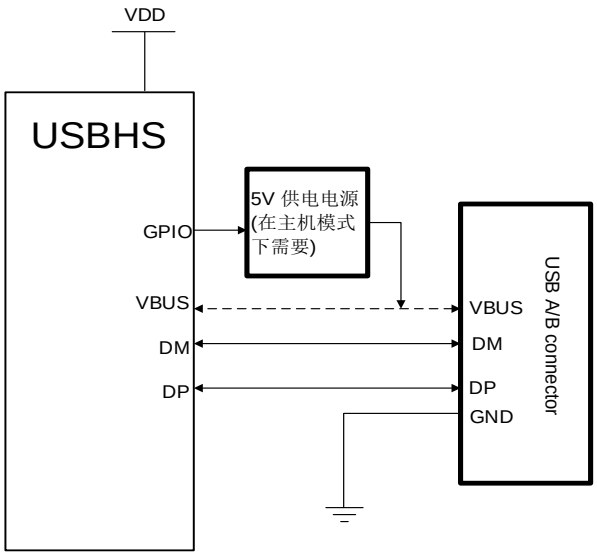
应用可以使用USBHS_GUSBCS寄存器中的FHM和FDM控制位选择USBHS的工作模式：主机模式(FHM=1)或设备模式(FDM=1)。当这两个控制位被清除时，USBHS工作在OTG模式，即系统复位后的默认模式。

内部嵌入式PHY

USBHS包含一个内部嵌入式PHY，该内部嵌入式PHY支持主机模式下的高速、全速和低速、设备模式下高速和全速，以及具备HNP和SRP的OTG协议。软件需要置位USBHS_GUSBCS寄存器中的EMBPHY_FS控制位并清除USBHS_GUSBCS寄存器中的EMBPHY_HS控制位来使用该内部嵌入式PHY的全速模式；或者清除EMBPHY_FS控制位并置位EMBPHY_HS控制位来使用该内部嵌入式PHY的高速模式。如果内部全速PHY被选择，USBHS在全速模式下所使用的USB时钟需要配置为48MHz，在高速模式需要配置为60MHz。该48MHz USB时钟从系统内部时钟产生，并且其时钟源和分频器需要在RCU模块中配置，而60MHz USB时钟由480MHz PLLUSB产生。

上拉或下拉电阻已经集成在内部全速PHY的内部，并且USBHS可根据当前模式（主机、设备或OTG模式）和连接状态进行自动控制。一个利用内部PHY的典型连接示意图如*图49-2. 在主机或设备模式下连接示意图*所示。

图 49-2. 在主机或设备模式下连接示意图

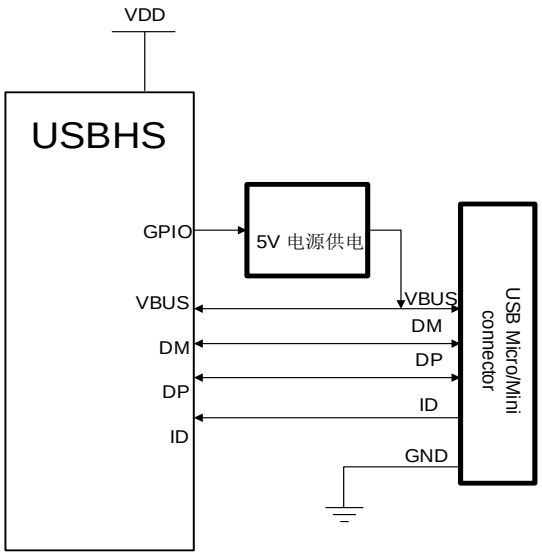


当USBHS工作在主机模式下时（FHM控制位置位、FDM控制位清除），VBUS为USB协议所定义的5V电源检测引脚。内部PHY不能提供5V VBUS电源，仅在VBUS信号线上具有电压比较器和充电、放电电路。所以，如果应用程序需要提供VBUS电源，那么则需要一个外部的供电电源IC。在主机模式下，USBHS和USB连接头之间的VUBS连接可以被忽略，这是由于USBHS并不检测VBUS引脚的电平状态，并假定5V供电电源一直存在。

当USBHS工作在设备模式下时（FHM控制位清除、FDM控制位置位），VBUS检测电路由USBHS_GCCFG寄存器中的VDEN控制位所配置。因此，如果设备不需要检测VBUS引脚电压，可以配置VDEN控制位，并可释放VBUS引脚作为其他用途。否则，VBUS引脚的连接不能够被忽略，并且USBHS需要不断的检测VBUS电平状态，一旦VBUS电压降至所需有效值以下，需要立即关闭DP信号线上的上拉电阻，从而产生一个断开状态。

OTG 模式连接示意图如*图49-3. OTG 模式下使用内部嵌入式PHY 连接示意图*所示。当USBHS工作在 OTG 模式下时，USBHS_GUSBCS 寄存器内的 FHM、FDM 控制位和 USBHS_GCCFG 寄存器的 VDEN 位都应该被清除。在这种模式下，USBHS 需要以下四个引脚：DM、DP、VBUS 和 ID，并且需要使用若干个电压比较器检测这些引脚的电压。USBHS 也包含 VBUS 充电和放电电路，用以完成 OTG 协议中所描述的 SRP 请求。OTG A 设备或 B 设备由 ID 引脚的电平状态所决定。在实现 HNP 协议的过程中，USBHS 控制上拉和下拉电阻。

图 49-3. OTG 模式下使用内部嵌入式 PHY 连接示意图

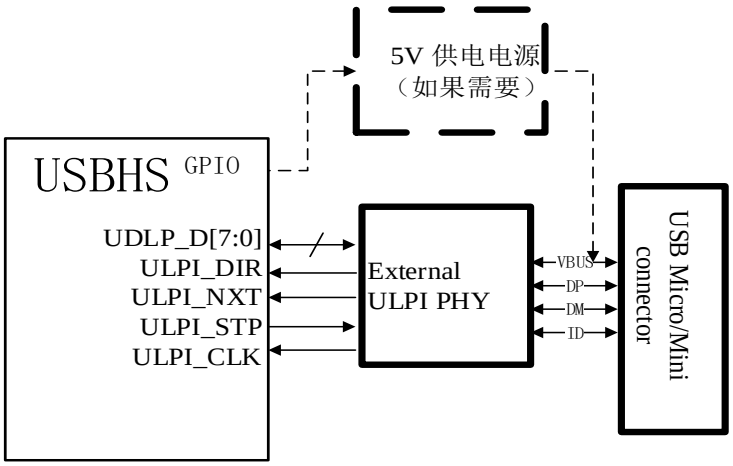


外部ULPI PHY

USBHS为外部PHY提供了一个ULPI接口。如果需要使用USBHS模块完成高速USB应用，那么则需要一个外部高速ULPI PHY。结合外部ULPI PHY，USBHS支持高速主机和设备，也支持前文中内部嵌入式全速PHY所描述的所有模式。

软件需要清除USBHS_GUSBCS寄存器中的EMBPHY_FS和EMBPHY_HS控制位以使能ULPI接口。当ULPI模式能使，USB时钟需要配置到60MHz，并且需要从ULPI_CLK引脚引入。软件可以在RCU模块中打开或关闭该60MHz ULPI时钟。

图 49-4. 使用外部 ULPI PHY 的连接示意图

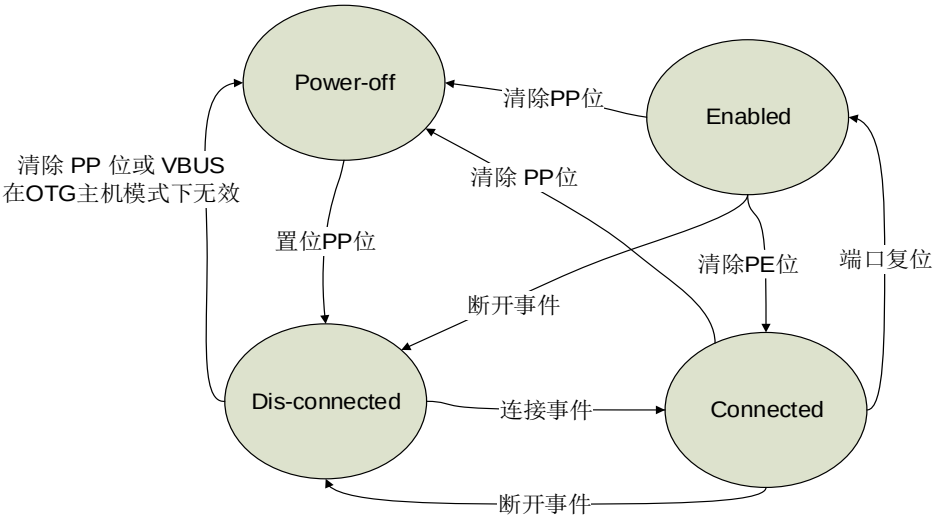


49.5.2. USB 主机功能

USB主机端口状态

主机应用可以通过USBHS_HPCS寄存器控制USB端口状态。系统初始化之后，USB端口保持掉电状态。通过软件置位PP控制位后，USB PHY（内部或外部）将被上电，并且USB端口变为断开状态。检测到连接后，USB端口变为连接状态。在USB总线上产生一个复位后，USB端口将变为使能状态。

图 49-5. 主机端口状态转移图



连接、复位和速度识别

作为USB主机，在检测到一个连接事件后，USBHS会为应用程序触发一个连接标志；同样，若检测到一个断开事件后，将会触发一个断开标志。

PRST控制位用于实现USB复位序列。应用程序可以置位该控制位以启动一个USB复位序列，或者清除该控制位以结束USB复位序列。仅当端口在连接或使能状态时，该控制位有效。

USBHS在对设备连接和复位时执行速度检测，并且速度检测的结果会反馈在USBHS_HPCS寄存器的PS位域中。

如果最大支持速度被配置为全速（SPDFSLS=1），USBHS仅仅在设备连接的过程中执行速度识别，并且从DM或DP的电平状态决定设备速度。就像USB协议中所描述的那样，全速设备上拉DP信号线，而低速设备上拉DM信号线。

如果最大支持速度被配置为高速（SPDFSLS=0），USBHS首先在连接的过程中执行速度检测，如果检测到全速设备连接，USBHS会在连接事件后的每个USB复位序列中，尝试执行高速检测（USB2.0协议中所描述的CHIRP序列）。所以，在主机上的应用应该在一个连接事件后提供一个USB复位，并且再次检查PS[1:0]标志位，以确定其连接的是否为高速设备。

挂起和复位

USBHS支持挂起和复位状态，当USBHS端口在使能状态时，向USBHS_HPCS寄存器的PSP控制位写1，USBHS会进入到挂起状态。在挂起状态下，USBHS停止在USB总线上发送SOF，并且这样会让所连接的USB设备在3ms后进入挂起状态。应用程序能够置位USBHS_HPCS寄

存器中的PREM控制位以启动一个恢复序列，从而唤醒挂起的设备，当清除该控制位时，则可以停止恢复序列。如果主机在挂起状态下检测到一个远程唤醒信号，将会置位USBHS_GINTF寄存器的WKUPIF标志位，并且触发USBHS唤醒中断。

SOF产生器

在主机模式下，USBHS向USB总线发送SOF令牌包。如USB2.0协议所描述，全速连接下，SOF令牌包每1ms产生一次(由主机控制器或者HUB事务转换器产生)；高速连接下，SOF令牌包将在接下来的7个125 μs周期后产生。

每当USBHS进入到使能状态后，它将会按照USB2.0所定义的周期发送SOF令牌包。然而，应用程序可以通过写USBHS_HFT寄存器中的FRI[15:0]位来调整一帧的间隔。FRI位定义了在一帧中的USB时钟周期个数，并且应用程序应该基于USBHS所使用的USB时钟频率计算该值。FRT[14:0]位显示当前帧剩余的时钟周期个数，并且在挂起状态时，该值将停止改变。

USBHS能够在每个SOF令牌包中产生一个脉冲信号，并且将其输出至一个引脚。该脉冲信号长度为12个HCLK周期。如果应用程序希望使用该功能，需要置位USBHS_GCCFG寄存器的SOFOEN控制位，并且配置相应的引脚寄存器为GPIO功能。

USB通道和事务

USBHS在主机模式下包含16个独立的通道。每个通道能够与一个USB设备端点通信。通道的传输类型、方向、数据包长和其他信息都在通道相应的寄存器中配置，例如USBHS_HCHxCTL和USBHS_HCHxLEN寄存器。

USBHS支持所有的四种传输类型：控制、批量、中断和同步。USB 2.0协议将这些传输类型划分为两类：非周期性传输（控制和批量）和周期性传输（中断和同步）。基于此，为了有效地进行事务调度，USBHS包含两种请求队列：周期性请求队列和非周期性请求队列。在上述请求队列中的请求条目可能代表一个USB事务请求或者一个通道操作请求。

在无DMA模式下，如果应用程序想要在USB总线上启动一个OUT事务，应用程序需要通过AHB寄存器接口向数据FIFO中写入数据包。USBHS硬件会在应用程序写完整包数据后，自动产生一个事务请求并进入请求队列。在DMA模式下，应用程序仅需要配置通道属性和通道数据缓冲区地址，USBHS内部的DMA引擎会执行数据包拷贝和请求条目的产生工作。当应用程序使能IN通道时，USBHS自动产生IN请求条目。

请求队列中的请求条目通过事务控制模块按顺序处理。USBHS通常首先尝试处理周期性请求队列，然后处理非周期性请求队列。

帧起始后，USBHS首先开始处理周期性队列，直到队列为空抑或当前周期性请求队列所需时间不够，然后处理非周期性队列。这种做法保证了一帧中周期性传输的带宽。每次USBHS从请求队列中读取并取出一个请求条目。如果取出的是通道禁用请求，这将直接禁用通道并准备处理下个条目。

如果当前请求是一个事务请求并且USB总线时间能够处理这个请求，USBHS会使用SIE在USB总线上产生该事务。

在当前帧内，当前请求所需的总线时间不足时，如果当前请求为周期性请求，USBHS停止处理该周期性请求队列，并启动处理非周期性请求。如果当前请求为非周期性请求，USBHS会停止处理任何队列，并等待直到当前帧结束。

LPM

USBHS模块添加了电源管理状态（LPM状态）和机制，这种机制影响主机和集线器用于有效管理总线和系统电源的状态更改。LPM只是添加了一个新特性和总线休眠状态（L1），它与USB2.0定义的suspend（L2）/resume共存。

L1类似于L2，但是使用起来比L2更加细致。进入到转换成L1是通过对集线器或主机端口的请求启动的。LPM事务被发送到下游设备，该事务请求的转换只能在设备响应了ACK握手时才发生。通过远程唤醒、恢复信令、重置信令或断开连接从L1退出。主机或设备可以在L1中启动恢复信令。尽管resume的信号等级与L2相同，但与L1、L0（活动状态）转换相关的信号和过渡延迟的持续时间要短得多。

49.5.3. USB 设备功能

USB设备连接

在设备模式下，USBHS在初始化后保持掉电状态。利用VBUS引脚上的5V电源连接USB主机后或者置位USBHS_GCCFG寄存器中VDEN控制位，USBHS将进入供电状态。USBHS首先打开DP信号线上的上拉电阻，之后主机将会检测到一个连接事件。

复位和速度识别

USB主机在检测到设备连接之后，总是会启动一个USB复位序列，并且在设备模式下，检测到USB总线复位事件后，USBHS会为软件触发一个复位中断。

如果最大支持速度被配置为全速（USBHS_DCFG寄存器内DS[1:0] = 01），USBHS会以全速设备操作，然而如果最大支持速度被配置为高速（USBHS_DCFG寄存器内DS[1:0] = 00），在复位序列中，USBHS设备会尝试和主机启动一个速度识别（USB2.0协议中描述的一个CHIRP序列）。如果和主机的CHIRP序列握手成功，设备将会进入高速模式，否则，仍然停留在全速模式。

在复位序列和速度识别过程完成后，USBHS将会触发USBHS_GINTF寄存器中的ENUMF标志/中断，并且利用USBHS_DSTAT寄存器内的ES标志位反映当前枚举设备速度。所以，如果软件想要实现一个高速设备，必须等待ENUMF中断，然后读取ES[1:0]控制位以获得速度识别结果。

如USB2.0协议所需要，USBHS在外设模式下不支持低速。

挂起和唤醒

USB总线保持IDLE状态并且数据线3ms无变化，USB设备将会进入挂起状态。当USB设备在挂起状态时，软件能够关闭大部分的时钟以节省电能。USB主机可以通过在USB总线上产生恢复信号，来唤醒挂起的设备。USBHS检测到恢复信号后，将置位USBHS_GINTF寄存器的WKUPIF标志位并且触发USBHS唤醒中断。

在挂起设备模式，USBHS也能够远程唤醒USB总线。软件可以通过置位USBHS_DCTL寄存器的RWKUP控制位来发送一个远程唤醒信号，并且如果USB主机支持远程唤醒，主机会在USB总线上启动发送一个恢复信号。

软件断开

USBHS支持软件断开。设备进入到供电状态后，USBHS会打开DP信号线的上拉电阻，并且这样主机会检测到设备连接。然后，软件可以通过置位USBHS_DCTL寄存器中SD控制位进行强制断开。SD控制位置位后，如果当前设备速度为高速，USBHS会首先返回到全速设备，然后关闭DP信号线上的上拉电阻；如果当前设备速度为全速，USBHS将会直接关闭上拉电阻。这样，USB主机将会在USB总线上检测到设备断开。

SOF跟踪

当USBHS在USB总线上接收到一个SOF令牌包时，将触发一个SOF中断，并且开始利用本地USB时钟计算总线时间。当前帧的帧号将会反应在USBHS_DSTAT寄存器的FNRSOF[13:0]位域中。当USB总线时间达到EOF1或EOF2点（帧结束，在USB 2.0协议中描述），USBHS会触发USBHS_GINTF寄存器中的EOPFIF中断。软件能够使用这些标志位和寄存器以获得当前总线时间和位置信息。

BCD

支持第1.2版电池充电规范中描述的充电端口检测（BCD）。为了使PD（便携式设备）确定允许从上游USB端口吸取多少电流，需要PD有区分标准下游端口和充电端口的机制。

在BCD机制中，包括USB VBUS检测（VD）、数据接触检测（DCD）、主检测（PD）和次检测（SD）。关于BCD的控制和配置位在USBHS_GCCFG寄存器中描述。

49.5.4. OTG 功能概述

USBHS支持OTG协议1.3/2.0中所描述的OTG功能，OTG功能包括SRP和HNP。

A设备和B设备

当标准A或微型A插头插入相应的插座时，具有OTG能力的USB设备为A设备。A设备向VBUS供电，并且在会话开始时默认为主机。当标准B、微型B、迷你B插头插入相应的插座或采用一端为标准A插头的不可分离电缆时，具有OTG能力的USB设备为B设备。B设备在会话开始时默认为外设。USBHS使用ID引脚电平状态决定A设备或B设备。ID引脚状态反馈在USBHS_GOTGCS寄存器的IDPS状态位。为了了解A设备和B设备之间传输的详细状态，请参考OTG1.3/2.0协议。

HNP

主机协商协议（HNP）允许主机功能在两个直接连接的OTG设备之间转换，并且用户不需要为了设备之间通信控制的改变而切换电缆线的连接。典型地，HNP协议是由B设备上的用户或应用启动，HNP只能通过设备上的微型AB插座执行。

一旦OTG设备具有一个微型AB插座，该OTG设备可通过插入的插头类型决定默认为主机或设备（微型A插头插入为主机，微型B插头插入为设备）。通过使用主机协商协议（HNP），一个默认为外设的OTG设备可以请求成为主机。主机角色切换的过程在下段中描述。此协议使用户不需要为了更改连接设备的角色而切换电缆线的连接。

当USBHS工作在OTG A主机模式时，并且其想放弃主机角色，可以首先置位USBHS_HPCS寄存器的PSP控制位来使USB总线进入挂起状态，然后B设备在3ms后进入挂起状态。如果B设备想要变为主机，软件需要置位USBHS_GOTGCS寄存器的HNPREQ控制位，然后USBHS会开始在总线上执行HNP协议，最后，HNP的结果会反馈在USBHS_GOTGCS寄存器的HNPS状态

位。另外，软件总能从USBHS_GINTF寄存器的COPM状态位获取当前设备角色（主机或外设）。

SRP

会话请求协议（SRP）允许B设备请求A设备打开VBUS并启动一个会话。该协议允许A设备（或许是电池供电）当总线无活动时通过关闭VBUS以节省电能，并为B设备启动总线活动提供了一种方法。如OTG协议中所描述，OTG设备必须和几个阈值比较VBUS电压，并且将比较结果反馈在USBHS_GOTGCS寄存器的ASV和BSV状态位中。

当USBHS工作在B设备OTG模式时，软件可以通过置位USBHS_GOTGCS寄存器的SRPREQ控制位来启动一个SRP请求，并且如果SRP请求成功，USBHS会在USBHS_GOTGCS寄存器中产生一个成功标志位SRPS。

当USBHS工作在OTG A设备模式且从B设备检测到一个SRP请求时，USBHS将会置位USBHS_GINTF寄存器中的SESIF标志位。软件获取该标志位后，需要准备为VBUS引脚打开5V供电电源。

ADP

附加检测协议（ADP）是一种允许本地设备检测远程设备何时被连接或分离的协议。远程设备可以是任何USB设备。ADP通过检测两个设备连接或分离时VBUS电容的变化来工作。电容的检测方法是先对VBUS线放电，然后用已知的电流源测量VBUS充电到已知电压所需的时间。通过寻找充电时间的变化来检测电容的变化。

软件可以设置ADPMEN、ADPEN和ENAPRB位来执行ADP探测，并且应至少执行一个ADP探测周期，以便在首次启动具有ADP功能的A设备或B设备时获得TADP_RISE的初始值。对于B设备，可以通过设置位ENASNS来执行ADP sense。如果USBHS_ADPCTL寄存器中的RITM发生变化，则表明远程设备已连接或分离。

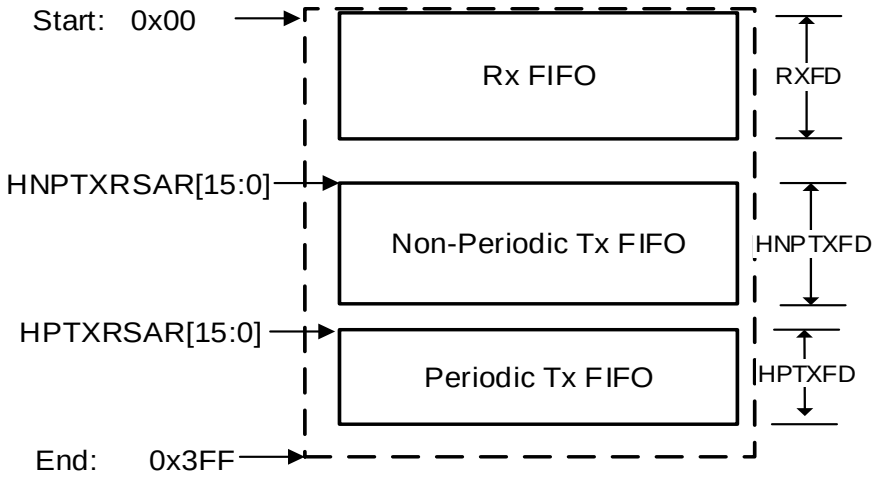
49.5.5. 数据 FIFO

USBHS中采用4K字节数据FIFO存储包数据，数据FIFO是通过USBHS的内部SRAM实现的。

主机模式

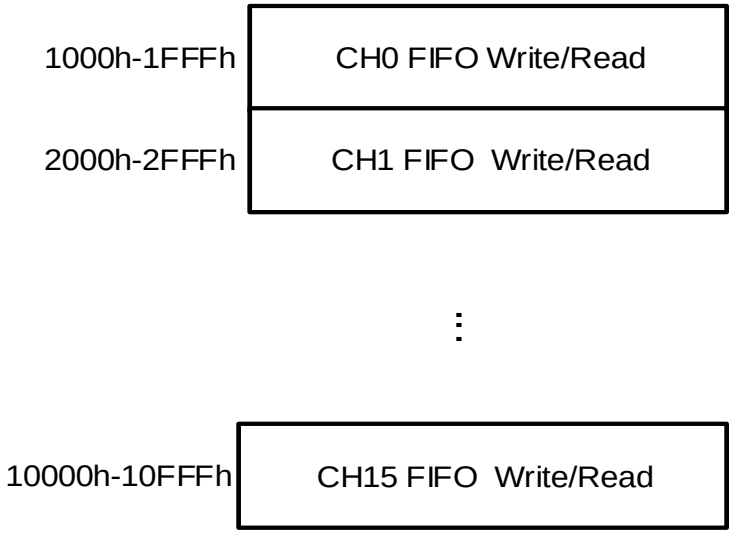
主机模式下，数据 FIFO 空间分为三个部分，分别是：用于接收数据包的 Rx FIFO、用于非周期性发送数据包的非周期性 Tx FIFO 和用于周期性发送数据包的周期性 Tx FIFO。所有的 IN 通道通过共享 Rx FIFO 接收数据。所有的周期性 OUT 通道通过共享周期性 Tx FIFO 来发送数据，所有的非周期性 OUT 通道通过共享非周期性 Tx FIFO 来发送数据。通过寄存器 USBHS_GRFLEN、USBHS_HNPTFLEN 和 USBHS_HPTFLEN，软件可以配置以上数据 FIFO 的大小和起始偏移地址。*图 49-6. 主机模式 FIFO 空间*所描述的是 SRAM 中各 FIFO 的结构，图中的数值是按照 32 位为单位写的。

图 49-6. 主机模式 FIFO 空间



在 DMA 模式下，DMA 负责系统存储区和数据 FIFO 之间的数据包传输。在非 DMA 模式下，程序将包数据写入数据 FIFO 或从数据 FIFO 读取包数据。USBHS 为程序提供了专有寄存器空间来读写数据 FIFO。*图 49-7. 主机模式 FIFO 访问寄存器映射表*所描述的是数据 FIFO 所访问的寄存器存储空间，图中的数值是以字节为单位寻址。尽管所有的非周期通道共享相同的 FIFO 以及所有的周期通道共享相同的 FIFO，每个通道都拥有它们的 FIFO 访问寄存器空间。对 USBHS 而言，获知当前压入数据包的通道号是非常重要的，通过寄存器 USBHS_GRXTATR/USBHS_GRSTATP 来访问数据包所从属的 Rx FIFO。

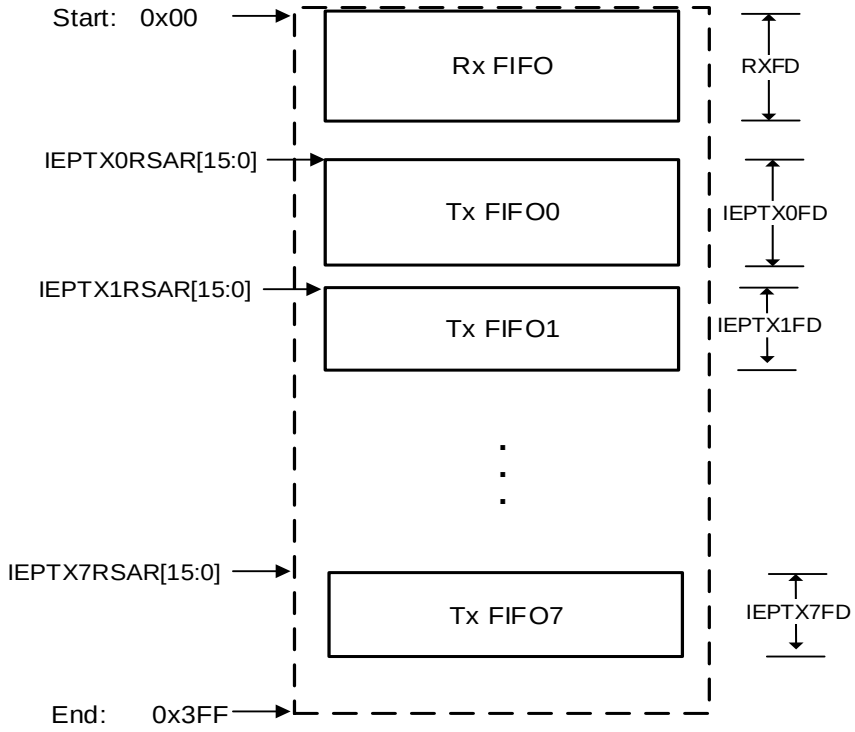
图 49-7. 主机模式 FIFO 访问寄存器映射表



设备模式

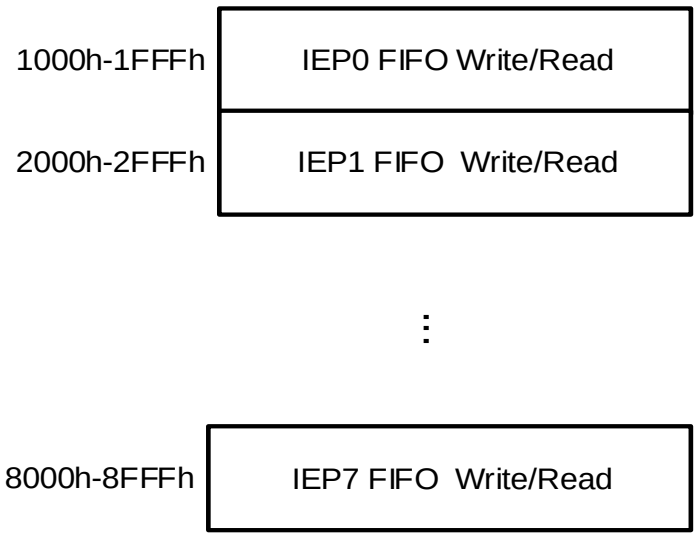
在设备模式下，数据 FIFO 分为多个部分，其中包含 1 个 Rx FIFO 和 8 个 Tx FIFO，每个 Tx FIFO 对应着一个 IN 端点，所有的 OUT 端点通过共享 Rx FIFO 接收数据包。通过寄存器 USBHS_GRFLEN 和 USBHS_DIEPxTFLEN （x=0…7），程序可配置数据 FIFO 的大小和起始偏移地址。*图 49-8. 设备模式 FIFO 空间*所描述的是 SRAM 中各 FIFO 的结构，图中的数值是以按照 32 位写的。

图 49-8. 设备模式 FIFO 空间



在 DMA 模式下，DMA 负责系统存储区和数据 FIFO 之间的数据包传输。在非 DMA 模式下，程序将包数据写入数据 FIFO 或从数据 FIFO 读取包数据。USBHS 为程序提供了专有寄存器空间来读写数据 FIFO。*图 49-9. 设备模式 FIFO 访问寄存器映射表*所描述的是数据 FIFO 所访问的寄存器存储空间，图中的数值是以字节为单位寻址。每个端点都拥有它们的 FIFO 访问寄存器空间。通过寄存器 USBHS_GRXTATR/USBHS_GRSTATP 来访问 Rx FIFO。

图 49-9. 设备模式 FIFO 访问寄存器映射表



49.5.6. DMA 功能

该部分描述USBHS的DMA调度器和引擎。

DMA请求和调度器

DMA功能通过置位寄存器USBHS_GAHBCS的位DMAEN获得使能。当一个IN/OUT通道或IN端点被适当地配置和使能，或Rx FIFO非空，USBHS将生成DMA请求。USBHS的DMA调度器负责应答这些DMA请求。

当同时存在多个请求时，DMA调度器负责仲裁这些请求。这些请求分为三类：Rx FIFO DMA请求、周期性传输DMA请求和非周期性传输DMA请求。在仲裁中，Rx FIFO DMA请求是最高优先级，周期性传输DMA请求是中级优先级，非周期性传输DMA请求是最低优先级。在处理周期性和非周期性传输DMA请求中，DMA调度器实行循环仲裁方法。

综上所述，DMA将自动处理Rx FIFO非空事件，所以，在DMA模式下，程序中可以忽略寄存器USBHS_GINTF的RXFNEIF标志位。

DMA引擎

接收：

在主机或设备模式下，当Rx FIFO DMA请求获得仲裁后，DMA驱动器开始从Rx FIFO读取包数据或状态条目。对于包数据而言，DMA将数据写到特定的系统地址，该地址配置在寄存器USBHS_HCHxDMAADDR或USBHS_DIEPxDMAADDR / USBHS_DIEPxDMAADDR。对于表目状态而言，在相关的通道或端点，DMA将生成特定的标志位或中断。

主机传输：

当一个IN周期性或非周期性通道DMA请求获得仲裁后，DMA将IN请求条目写入周期性或非周期性请求队列。当一个预期的IN传输完成，或一个AHB/USB总线错误发生后，DMA停止特定的通道，生成寄存器USBHS_HCHxINTF的TF和CH标志位。如上文所述，在Rx FIFO DMA请求生成后，在IN传输的过程中所接受的包数据被复制到系统存储区。

当一个OUT周期性或非周期性通道DMA请求获得仲裁后，DMA从系统存储区读取包数据，或将包数据写到内部的Tx FIFO。当每次完成包数据复制后，DMA总是将OUT请求条目写入请求队列。当一个预期的OUT传输完成，或一个AHB/USB总线错误发生后，DMA停止特定的通道，生成寄存器USBHS_HCHxINTF的TF和CH标志位。

设备传输：

在设备模式下，当一个IN端点DMA请求获得仲裁后，DMA从系统存储区读取包数据，或将包数据写到端点的Tx FIFO。当USBHS获取IN端点的IN令牌后，将发送DMA引擎所复制的包数据。

49.5.7. 操作手册

该部分描述的是USBHS的操作手册。

主机模式

全局寄存器初始化顺序：

1、 根据应用的需求，如是否使能DMA、DMA的传输类型、Tx FIFO的空阈值等，设置寄存器USBHS_GAHBCS，此时，GINTEN位需要保持清零状态；

2、 根据应用的需求，如操作模式（主机、设备或OTG）、某些OTG参数、ULPI和USB协议，设置寄存器USBHS_GUSBCS；

3、 根据应用的需求，设置寄存器USBHS_GCCFG；

4、 根据应用的需求，设置寄存器USBHS_GRFLEN、USBHS_HNPTFLEN_DIEP0TFLEN、USBHS_HPTFLEN，配置数据FIFO；

5、 通过设置寄存器USBHS_GINTEN使能模式错误和主机端口中断，置位USBHS_GAHBCS寄存器的GINTEN位使能全局中断；

6、 通过设置寄存器USBHS_HCTL的SPDFSLS位，判断是否将设备速度限制为全速。

7、 设置寄存器USBHS_HPCS，置位PP位；

8、 等待设备连接，当设备连接后，触发寄存器USBHS_HPCS的PCD位，然后置位PRST位，执行一次端口复位，等待至少10毫秒后，清除PRST位；

9、 等待USBHS_HPCS寄存器的PEDC中断，然后读取PE位以确认端口被成功地使能，读取PS位以获取连接的设备速度，之后，如果软件需要改变SOF间隔，设置USBHS_HFT寄存器。

通道初始化和使能顺序：

1、 根据期望的传输类型、方向、包大小等信息，设置寄存器USBHS_HCHxCTL，在设置期间，要保证位CEN和CDIS保持清除；

2、 设置寄存器USBHS_HCHxINTEN，设置期望的中断使能位；

3、 在DMA使能的前提下，设置寄存器USBHS_HCHxDMAADDR；

4、 设置寄存器USBHS_HCHxLEN，PCNT表示一次传输中的包数，TLEN表示一次传输中发送或接收的包数据的总字节数；

5、 对于OUT通道，如果PCNT为1，单包的大小等于TLEN。如果PCNT大于1，前PCNT-1个包被认定为最大包长度的包，其大小是由寄存器USBHS_HCHxCTL的位MPL所定义。最后一包的大小可通过PCNT、TLEN和MPL计算得到。如果程序想要发出一个零长度的包，应该设定TLEN为0，PCNT位1；

6、 对于IN通道，因为在IN事务结束之前，程序不知道实际接收的数据大小，程序可将TLEN设定为Rx FIFO所支持的最大值；

7、 置位寄存器USBHS_HCHxCTL中的CEN位以使能通道。

通道除能顺序：

程序可以通过同时置位CEN和CDIS除能通道。在寄存器操作后，USBHS将在请求队列中产生一个通道除能请求条目。当这个请求条目到达请求队列的顶部时，USBHS立即进行处理。

对于OUT通道而言，特定的通道将被立即除能。然后，会产生CH标志，USBHS将清除CEN和CDIS位。

对于IN通道而言，USBHS将通道除能状态条目压入Rx FIFO，然后，程序应该处理Rx FIFO非空事件：读和取出该状态条目，然后会产生CH标志，USBHS将清除CEN和CDIS位。

IN传输操作顺序（DMA除能）：

1、 初始化USBHS全局寄存器；

2、 初始化相应的通道；

3、 使能相应的通道；

4、 通过软件使能IN通道后，USBHS在相应请求队列中生成一个Rx请求条目；

5、 当Rx请求条目到达请求队列的顶部时，USBHS开始执行该请求条目。对于由请求条目所指示的事务而言，如果总线时间足够，USBHS在USB总线上开始IN事务；

6、 当IN事务结束时（收到ACK握手包），USBHS将接收到的数据包压入Rx FIFO，ACK标志位被触发，否则，状态标志（NAK）会指示事务结果；

7、 如果步骤5所描述的IN事务完成后，步骤2的PCNT的数值比1大，程序将会返回步骤3，继续接收剩下的数据包。如果步骤5中描述的IN事务没有成功完成，程序将会返回步骤3来再次发送该数据包；

8、 在所有的传输中的所有事务都被成功接收后，USBHS将TF状态条目压入Rx FIFO的最后的数据包的顶部，这样，软件在读取所有接收的数据包后，再读取TF状态条目。USBHS生成TF标志来指示传输成功结束；

9、 除能通道，当通道处于空闲状态，即可为其他传输做准备。

IN传输操作顺序（DMA使能）：

1、 初始化USBHS全局寄存器；

2、 初始化并使能相应通道；

3、 在通过软件使能IN通道后，USBHS在相应请求队列中生成一个Rx请求条目；

4、 USBHS逐一处理IN请求队列中的请求条目，并将它们所指示的IN事务发到USB总线上；

5、 当一个IN事务获得NAK握手包时，DMA可以自动地再发IN令牌直至USBHS获得预期的数据包的数目；

6、 在USBHS获取寄存器USBHS_HCHxTLEN的位PCNT中期望数据包数目后，USBHS生成TF和CH标志来表示传输成功完成，相应通道除能。如果在这些事务期间发生USB总线错误或DMA取值错误，DMA将触发相关的错误标志，停止该通道的操作，最后除能该通道，触发CH标志。

注意：在DMA模式下，因为DMA将自动处理Rx FIFO，程序不再使能或处理RXFNEIF中断。

OUT传输操作顺序（DMA除能）：

1、 初始化USBHS全局寄存器；

2、 初始化及使能相应通道；

3、 将数据包写入通道的Tx FIFO（周期性Tx FIFO或非周期性Tx FIFO）。在所有的数据包都被写入FIFO后，USBHS在相应的请求队列中产生一个Tx请求条目，并且将USBHS_HCHxTLEN中的TLEN值减少，减少的数值等于已写的包大小；

4、 当请求条目到达请求队列的顶部时，USBHS开始执行该请求条目。如果请求条目对应的事务的总线时间足够，USBHS在USB总线上开展OUT事务；

5、 当由请求条目所指示的OUT事务结束时，寄存器USBHS_HCHnTLEN的位PCNT减1。如果该事务完成（收到ACK握手包），ACK标志位被触发，否则，状态标志（NAK）会指示事务结果；

6、 如果步骤5所描述的OUT事务完成后且步骤2的PCNT的数值比1大，程序将会返回步骤3，继续发送剩下的数据包。如果步骤5中描述的OUT事务没有成功完成，程序将会返回步骤3来再次发送该包；

7、 在所有的传输中的所有事务都被成功送达后，USBHS生成TF标志来指示传输成功结束；

8、 除能通道，当通道处于空闲状态，即可为其他传输做准备。

OUT传输操作顺序（DMA使能）：

1、 初始化USBHS全局寄存器；

2、 初始化并使能相应通道；

3、 USBHS的DMA开始从寄存器USBHS_HCHxDMAADDR的位DMAADDR中所指示的地址取包数据，并且将数据写入相应通道的Tx FIFO（周期性Tx FIFO或非周期性Tx FIFO）。每当一个完整的包数据被写入FIFO中，USBHS在相应的请求队列中生成一个Tx请求条目，并减少寄存器USBHS_HCHxTLEN的位TLEN的数值，所减少的数值与所完成写操作的包大小相同；

4、 USBHS逐一处理请求队列中的请求条目，并将它们所指示的事务发到USB总线上；

5、 当一个事务获得NAK或PING握手包时，DMA可以再取或是再发数据包，在执行PING协议时也会自动像这样执行；

6、 如果所有的事务都被成功发送到USB总线上，USBHS生成TF和CH标志来表示传输成功完成，相应通道除能。如果在这些事务期间发生USB总线错误或DMA取值错误，DMA将触发相关的错误标志，停止该通道的操作，最后除能该通道，触发CH标志。

注意：在DMA模式下，因为DMA将自动处理Rx FIFO，程序不再使能或处理RXFNEIF中断。

设备模式

全局寄存器初始化顺序：

1、 根据应用的需求，如是否使能DMA、DMA的传输类型、Tx FIFO的空阈值等，设置寄存器USBHS_GAHBCS，此时，GINTEN位需要保持清零状态；

2、 根据应用的需求，如操作模式（主机、设备或OTG）、某些OTG参数、ULPI和USB协议，设置寄存器USBHS_GUSBCS；

3、 根据应用的需求，设置寄存器USBHS_GCCFG；

4、 根据应用的需求，设置寄存器USBHS_GRFLEN、USBHS_HNPTFLEN_DIEP0TFLEN、USBHS_HPTFLEN，配置数据FIFO；

5、 通过设置寄存器USBHS_GINTEN使能模式错误、挂起、SOF、枚举完成和USB复位中断，

置位USBHS_GAHBCS寄存器的GINTEN位使能全局中断；

6、 根据应用的需求，如设备的地址和设备的速度等，设置寄存器USBHS_DCFG；

7、 在设备连接上主机上后，主机在USB总线上执行端口复位，触发寄存器USBHS_GINTF的RST中断；

8、 等待寄存器USBHS_GINTF的ENUMF中断。

端点初始化和使能顺序：

1、 根据预期的传输类型、包大小等信息，设置寄存器 USBHS_DIEPnCTL 或 USBHS_DOEPxCTL；

2、 设定寄存器 USBHS_DIEPINTEN 或 USBHS_DOEPINTEN，置位相应中断使能位；

3、 如果 DMA 使能，设定寄存器 USBHS_DIEPxDMAADDR 或 USBHS_DOEPxDMAADDR；

4、 设定寄存器 USBHS_DIEPxLEN 或 USBHS_DOEPxLEN，PCNT 表示一次传输中的包数，TLEN 表示一次传输中发送或接收的数据包的总字节数；

5、 对于 IN 端点，如果 PCNT 等于 1，单数据包的大小等于 TLEN。如果 PCNT 大于 1，前 PCNT-1 个包被认定为最大包长度的包，其大小是由寄存器 USBHS_DIEPxCTL 的位 MPL 所定义。最后一包的大小可通过 PCNT、TLEN 和 MPL 计算得到。如果程序想要发出一个零长度的包，应该设定 TLEN 为 0，PCNT 位 1；

6、 对于 OUT 端点，因为在 IN 事务结束之前，程序不知道实际接收的数据大小，程序可将 TLEN 设定为 Rx FIFO 所支持的最大值；

7、 置位 USBHS_DIEPxCTL 或 USBHS_DOEPxCTL 寄存器 EPEN 位使能端点。

端点除能顺序

当USBHS_DIEPnCTL或USBHS_DOEPnCTL寄存器的EPEN位被清除时，程序可以在任何时候除能端点

IN传输操作顺序（DMA除能）：

1、 初始化USBHS全局寄存器；

2、 初始化和使能IN端点；

3、 将数据包写入端点的Tx FIFO，每当数据包写入FIFO，USBHS减少USBHS_DIEPxLEN寄存器的TLEN域的数值，其减少的数值等于已写的数据包大小；

4、 当IN令牌接收后，USBHS发送数据包，在USB总线上的事务完成后，USBHS_DIEPxLEN寄存器的PCNT值减1。如果事务成功完成（接收到ACK握手包），ACK标志被触发，或者，其他状态标志表示事务的结果；

5、 在一次传输的所有数据包都被成功发送，USBHS生成一个TF标志位以表明传输成功结束，除能相应IN端点。

IN传输操作顺序（DMA使能）：

1、 初始化USBHS全局寄存器；

2、 初始化并使能相应端点；

3、 将数据包写入端点的Tx FIFO，每当包数据写入FIFO，USBHS减少USBHS_DIEPxLEN寄存器的TLEN域的数值，其减少的数值等于已写的包数据大小；

4、 当IN令牌接收后，USBHS发送数据包，在USB总线上的事务完成后，USBHS_DIEPxLEN寄存器的PCNT值减1。如果事务成功完成（接收到ACK握手包），ACK标志被触发，或者，其他状态标志表示事务的结果；

5、 在一次传输的所有数据包都被成功发送，USBHS生成一个TF和EPDIS标志位表明传输成功结束，除能相应IN端点。如果在事务期间出现USB总线错误或DMA取值错误，DMA将触发相关错误标志。

注意：在DMA模式下，因为DMA将自动处理Rx FIFO，程序不再使能或处理RXFNEIF中断。

OUT传输操作顺序（DMA除能）：

1、 初始化USBHS全局寄存器；

2、 初始化和使能端点；

3、 当OUT令牌接收后，USBHS接收数据包或基于Rx FIFO状态和寄存器配置回复NAK握手包。如果事务成功完成（USBHS接收并保存数据到Rx FIFO，发送ACK握手包），USBHS_DOEPxLEN寄存器的PCNT值减1。如果事务成功完成（接收到ACK握手包），ACK标志被触发，或者，其他状态标志表示事务的结果；

4、 在一次传输的所有数据包都被成功接收，USBHS将TF状态条目压入Rx FIFO的最后的数据包的顶部，这样，软件在读取所有接收的数据包后，再读取TF状态条目。USBHS生成TF标志来指示传输成功结束。USBHS生成一个TF标志位以表明传输成功结束，除能相应OUT端点。

OUT传输操作顺序（DMA使能）：

1、 初始化USBHS全局寄存器；

2、 初始化并使能相应OUT端点；

3、 当OUT令牌接收后，USBHS接收包数据或基于Rx FIFO状态和寄存器配置回复NAK握手包。如果事务成功完成（USBHS接收并保存数据到Rx FIFO，发送ACK握手包），USBHS_DOEPxLEN寄存器的PCNT值减1。如果事务成功完成（接收到ACK握手包），ACK标志被触发，或者，其他状态标志表示事务的结果；

4、 在一次传输的所有数据包都被成功发送，USBHS生成一个TF和EPDIS标志位表明传输成功结束，除能相应端点。如果在事务期间出现USB总线错误或DMA取值错误，DMA将触发相关错误标志。

注意：在DMA模式下，因为DMA将自动处理Rx FIFO，程序不再使能或处理RXFNEIF中断。

49.6. 中断

USBHS 有四种中断：全局中断、唤醒中断、端点 1 IN 中断和端点 1 OUT 中断。

全局中断是软件需要处理的主要中断，全局中断的标志位可在 USBHS_GINTF 寄存器读取，列举在*表 49-3. USBHS 全局中断*中。

表 49-3. USBHS 全局中断

中断标志	描述	运行模式
SESIF	会话中断	主机或设备模式
DISCIF	断开连接中断标志	主机模式
IDPSC	ID 引脚状态变化	主机或设备模式
LPMIF	LPM 中断标志	主机或设备模式
PTXFEIF	周期性 Tx FIFO 空中断标志	主机模式
HCIF	主机通道中断标志	主机模式
HPIF	主机端口中断	主机模式
ISOONCIF/PXNCIF	周期性传输未完成中断标志 / 同步OUT传输未完成中断标志	主机或设备模式
ISOINCIF	同步 IN 传输未完成中断标志	设备模式
OEPIF	OUT 端点中断标志	设备模式
IEPIF	IN 端点中断标志	设备模式
EOPFIF	周期性帧尾中断标志	设备模式
ISOOPDIF	同步 OUT 丢包中断标志	设备模式
ENUMF	枚举完成	设备模式
RST	USB 复位	设备模式
SP	USB挂起	设备模式
ESP	早挂起	设备模式
GONAK	全局 OUT NAK 有效	设备模式
GNPINAK	全局非周期 IN NAK 有效	设备模式
NPTXFEIF	非周期 Tx FIFO 空中断标志	主机模式
RXFNEIF	Rx FIFO 非空中断标志	主机或设备模式
SOF	帧首	主机或设备模式
OTGIF	OTG 中断标志	主机或设备模式
MFIF	模式错误中断标志	主机或设备模式

唤醒中断可以在 USBHS 处于挂起状态时触发，即使 USBHS 的时钟停止。寄存器 USBHS_GINTF 的位 WKUPIF 是唤醒源。

端点 1 IN/OUT 中断是适用于端点 1 的两个特殊中断，程序可通过这两个中断快速回应端点 1 的事件。这两个是中断通过寄存器 USBHS_DEP1INT 各自使能，这两个中断源来自于寄存器 USBHS_DIEP1INTF 和 USBHS_DOEP1INTF，其中断使能位定义在寄存器 USBHS_DIEP1INTEN 和 USBHS_DOEP1INTEN。

49.7. USBHS 寄存器

USBHS0 基地址：0x4004 0000

USBHS1 基地址：0x4008 0000

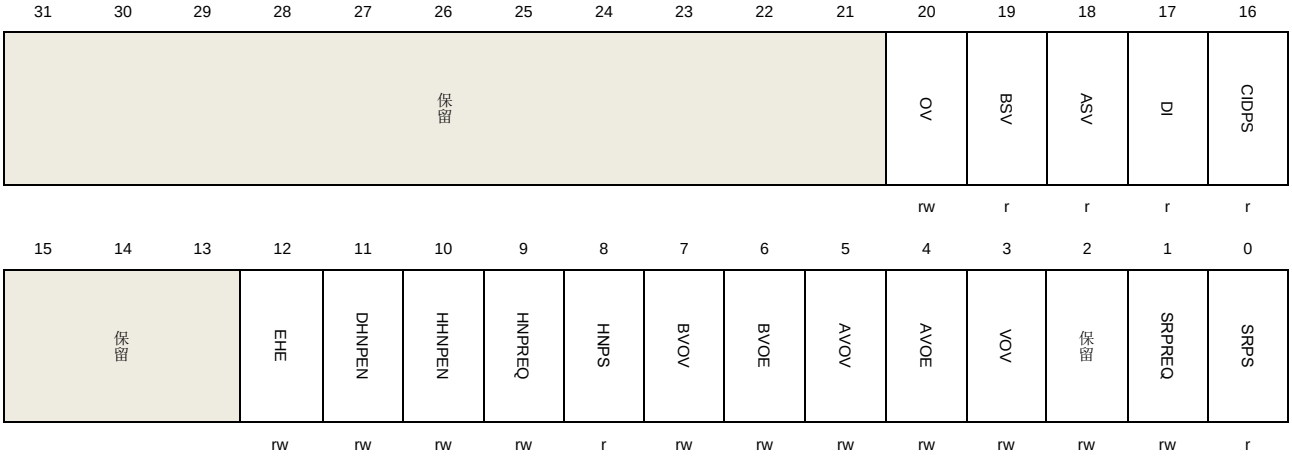
49.7.1. 全局控制与状态寄存器组

全局 OTG 控制和状态寄存器（USBHS_GOTGCS）

地址偏移：0x0000

复位值：0x0000 0800

该寄存器只能按字（32位）访问



位/位域	名称	描述
31:21	保留	必须保持复位值。
20	OV	OTG版本选择 0：1.3版本，SRP支持数据线脉冲和VBUS脉冲 1：2.0版本，SRP仅支持数据线脉冲
19	BSV	B会话有效（在OTG协议中描述） 0：OTG B设备VBUS电压水平低于VBSESSVLD 1：OTG B设备VBUS电压水平不低于VBSESSVLD 注意：仅在OTB B设备模式下可访问
18	ASV	A会话有效 A主机模式收发器状态 0：OTG A设备VBUS电压水平低于VASESSVLD 1：OTG A设备VBUS电压水平不低于VASESSVLD 在会话的开始，A设备默认是主机。 注意：仅在OTG A设备模式下可访问
17	DI	去抖动间隔

		检测到连接的去抖动间隔。 0：当USB总线上发生插入和连接时，表示长去抖动间隔 1：当HNP协议中使用一个软连接时，指示短去抖动间隔 注意：仅在主机模式下可访问
16	CIDPS	ID引脚状态 连接器ID引脚的电压水平 0：USBHS工作在A设备模式 1：USBHS工作在B设备模式 注意：在设备和主机模式下均可访问
15:13	保留	必须保持复位值。
12	EHE	嵌入式主机使能 0：选择OTG A设备状态 1：选择嵌入式主机状态
11	DHNPEN	设备HNP使能 使能B设备HNP功能。如果该控制位清除，当应用置位USBHS_GOTGCS寄存器中的HNPREQ控制位c时，USBHS并不启动HNP协议。 0：HNP功能不使能 1：HNP功能使能 注意：仅在设备模式下访问
10	HHNPEN	主机HNP使能 使能A设备HNP功能。如果该控制位清除，USBHS不能够响应B设备的HNP请求。 0：HNP功能不使能 1：HNP功能使能 注意：仅在主机模式下访问
9	HNPREQ	HNP请求 软件通过置位该控制位在USB总线上启动一个HNP。当USBHS_GOTGINTF寄存器中HNPEND控制位置位时，软件可以通过向该控制位写0或者清除USBHS_GOTGINTF寄存器中的HNPEND控制位来清除该控制位。 0：不发送HNP请求 1：发送HNP请求 注意：仅在设备模式下访问
8	HNPS	HNP成功标志位 当HNP成功时，该标志位由内核置位。当HNPREQ置位时，该控制位被清除。 0：HNP失败 1：HNP成功 注意：仅在设备模式下访问
7	BVOV	B外设会话有效覆盖值 0：BVOE=1时，B外设会话有效值为0 1：BVOE=1时，B外设会话有效值为1 注意：仅在设备模式下可访问

6	BVOE	B外设会话有效覆盖使能 0：覆盖禁用，从PHY接收内部B外设会话有效值 1：覆盖使能，从PHY接收的内部B外设会话有效值被BVOV值覆盖 注意：仅在设备模式下可访问
5	AVOV	A外设会话有效覆盖值 0：AVOE=1时，A外设会话有效值为0 1：AVOE=1时，A外设会话有效值为1 注意：仅在设备模式下可访问
4	AVOE	A外设会话有效覆盖使能 0：覆盖禁用，从PHY接收内部A外设会话有效值 1：覆盖使能，从PHY接收的内部A外设会话有效值被AVOV值覆盖 注意：仅在设备模式下可访问
3	VOV	VBUS有效覆盖值 0：VOE=1时，VBUS有效值为0 1：VOE=1时，VBUS有效值为1 注意：仅在设备模式下可访问
2	VOE	VBUS有效覆盖使能 0：覆盖禁用，从PHY接收VBUS有效值 1：覆盖使能，从PHY接收的VBUS有效值被VOV值覆盖 注意：仅在设备模式下可访问
1	SRPREQ	SRP请求 软件通过置位该控制位在USB总线上启动一个SRP会话请求。当USBHS_GOTGINTF寄存器中的SRPEND控制位置位时，软件可以通过向该控制位写0或者清除USBHS_GOTGINTF寄存器中的SRPEND控制位来清除该控制位。 0：没有会话请求 1：会话请求 注意：仅在设备模式下访问
0	SRPS	SRP会话请求成功 当SRP会话请求成功时，该标志位由内核置位。当SRPREQ控制位被置位时，该标志位被清除。 0：SRP会话请求失败 1：SRP会话请求成功 注意：仅在设备模式下访问

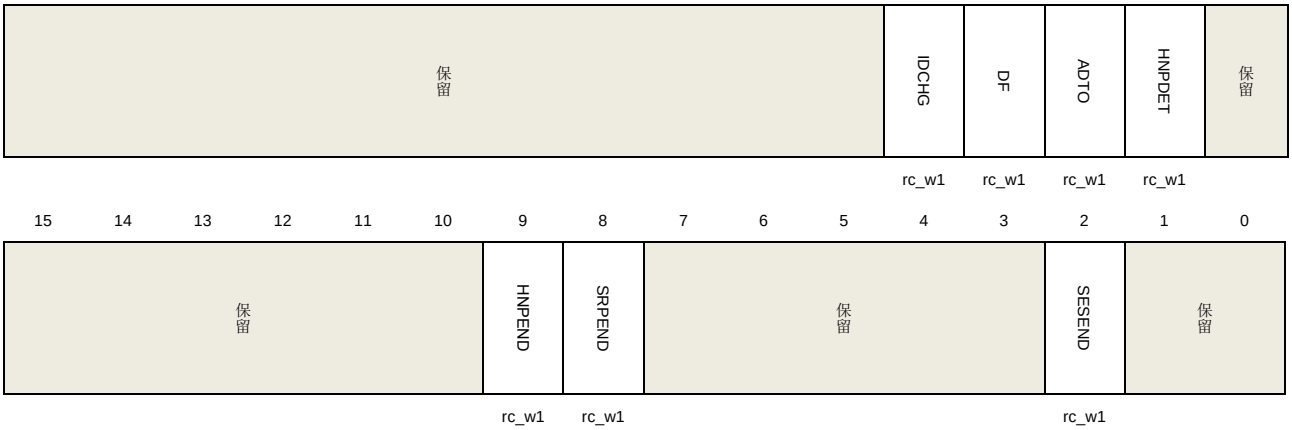
全局 OTG 中断状态寄存器（USBHS_GOTGINTF）

地址偏移：0x0004

复位值：0x0000 0000

该寄存器只能按字（32位）访问

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



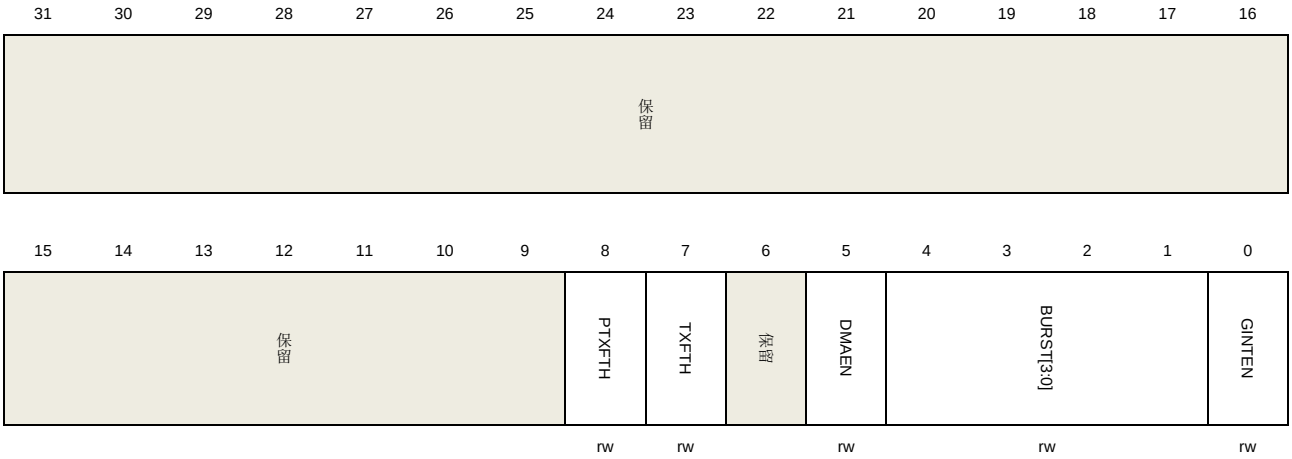
位/位域	名称	描述
31:21	保留	必须保持复位值。
20	IDCHG	ID输入的值有变化
19	DF	去抖动完成 当设备连接去抖动完成时，USBHS置位该控制位 注意：仅在主机模式下可访问
18	ADTO	A设备超时 当A设备等待B设备连接发生超时，USBHS置位该控制位 注意：在设备和主机模式下，均可访问
17	HNPDET	检测到主机协商请求 当A设备检测到一个HNP请求时，USBHS置位该标志位 注意：在设备和主机模式下，均可访问
16:10	保留	必须保持复位值。
9	HNPEND	HNP结束 当一个HNP结束时，内核置位该标志位。软件应该读取USBHS_GOTGCS寄存器中HNPS标志位，以获取HNP结果。 注意：在设备和主机模式下，均可访问。
8	SRPEND	SRPEND 当一个SRP结束时，内核置位该标志位。软件应该读取USBHS_GOTGCS寄存器中SRPS标志位，以获取SRP结果。 注意：在设备和主机模式下，均可访问。
7:3	保留	必须保持复位值。
2	SESEND	会话结束 当VBUS电压低于Vb_ses_vld时，内核置位该标志位。
1:0	保留	必须保留复位值。

全局 AHB 控制和状态寄存器（USBHS_GAHBCS）

地址偏移：0x0008

复位值：0x0000 0000

该寄存器只能按字（32位）访问



位/位域	名称	描述
31:9	保留	必须保持复位值。
8	PTXFTH	周期性Tx FIFO阈值 0：当周期性发送FIFO半空时，将触发PTXFEIF标志位 1：当周期性发送FIFO全空时，将触发PTXFEIF标志位 注意：只在主机模式下访问
7	TXFTH	Tx FIFO阈值 设备模式： 0：当IN端点发送FIFO半空时，将触发TXFEIF标志位 1：当IN端点发送FIFO全空时，将触发TXFEIF标志位 主机模式： 0：当非周期性发送FIFO半空时，将触发NPTXFEIF标志位 1：当非周期性发送FIFO全空时，将触发NPTXFEIF标志位
6	保留	必须保持复位值。
5	DMAEN	DMA功能使能 0：DMA功能禁用 1：DMA功能使能
4:1	BURST[3:0]	DMA使用的AHB突发类型 0000：单次 0001：INCR 0011：INCR4 0101：INCR8 0111：INCR16
0	GINTEN	全局中断使能

0：全局中断不使能

1：全局中断使能

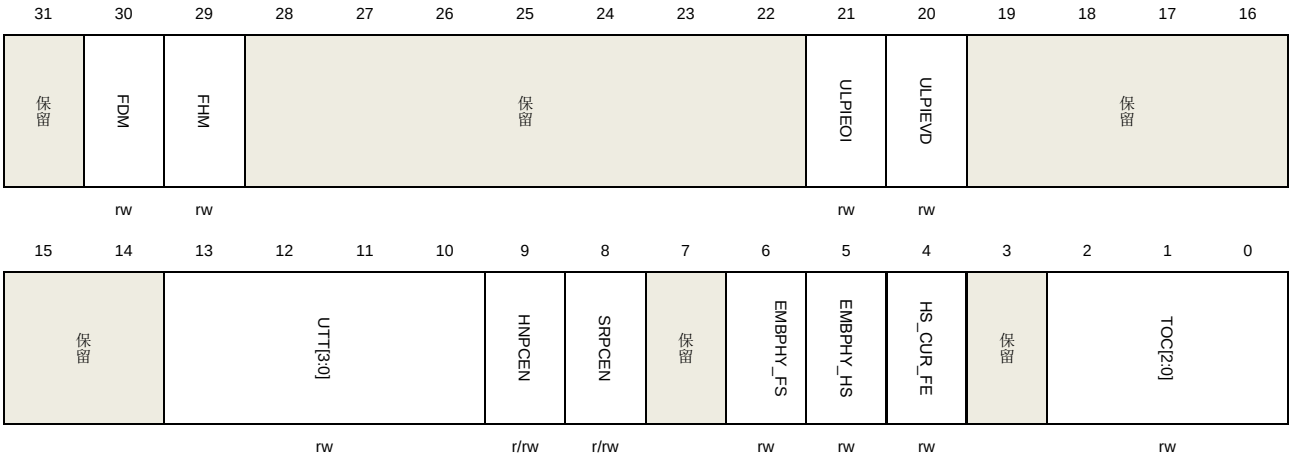
注意：在主机和设备模式下，均可访问

全局 USB 控制和状态寄存器（USBHS_GUSBCS）

地址偏移：0x000C

复位值：0x0000 1400

该寄存器只能按字（32位）访问



位/位域	名称	描述
31	保留	必须保持复位值。
30	FDM	强制设备模式 通过置位该控制位，可强制USB内核为设备模式，并且忽略USBHS ID引脚的输入状态 0：正常模式 1：设备模式 设置该控制位后，应用必须等待至少25ms，让变化产生作用。 注意：在设备和主机模式下，均可访问。
29	FHM	强制主机模式 通过置位该控制位，可强制USB内核为主机模式，并且忽略USBHS ID引脚的输入状态 0：正常模式 1：主机模式 设置该控制位后，应用必须等待至少25ms，让变化产生作用。 注意：在设备和主机模式下，均可访问
28:22	保留	必须保持复位值。
21	ULPIEOI	ULPI外部过流指示器 ULPI PHY使用该控制位决定使用内部或者外部过流指示器。该控制位只在外部ULPI PHY被使用时（本寄存器中EMBPHY_HS和EMBPHY_FS控制位为0），才有效。

		0：ULPI PHY使用内部过流指示器 1：ULPI PHY使用外部过流指示器
20	ULPIEVD	ULPI外部VBUS驱动器 ULPI PHY使用该控制位决定是由ULPI PHY还是外部电源驱动。该控制位仅在外部ULPI PHY被使用时（本寄存器中EMBPHY_HS和EMBPHY_FS控制位为0），才有效。 0：VBUS由ULPI PHY驱动 1：VBUS由外部电源驱动
19:14	保留	必须保持复位值。
13:10	UTT[3:0]	USB运转时间 以物理时钟数来设定运转时间 注意：仅在设备模式下访问
9	HNPCEN	HNP能力使能 控制HNP能力是否使能 0：HNP能力禁用 1：HNP能力使能 注意：在设备和主机模式下，均可访问
8	SRPCEN	SRP能力使能 控制SRP能力是否使能 0：SRP能力禁用 1：SRP能力使能 注意：在设备和主机模式下，均可访问
7	保留	必须保持复位值。
6	EMBPHY_FS	嵌入式全速PHY选择 0：嵌入式全速PHY禁用 1：嵌入式全速PHY使能 注意：该位仅在EMBPHY_HS为0时才能置1，在设备和主机模式下，均可访问
5	EMBPHY_HS	嵌入式高速PHY选择 0：嵌入式高速PHY禁用 1：嵌入式高速PHY使能 注意：该位仅在EMBPHY_FS为0时才能置1，在设备和主机模式下，均可访问
4	HS_CUR_FE	HS当前软件使能 0：释放HS模式，TX当前使能 1：强制HS模式，TX当前使能
3	保留	必须保持复位值。
2:0	TOC[2:0]	超时校准 当等待一个包时，USBHS需要使用USB2.0协议中需要的超时数值。应用可以使用TOC[2:0]增加该数值（以PHY时钟为单位）。PHY时钟频率由使用的PHY所决定：

48MHz（内部嵌入式PHY）和60MHz（外部ULPI PHY）。

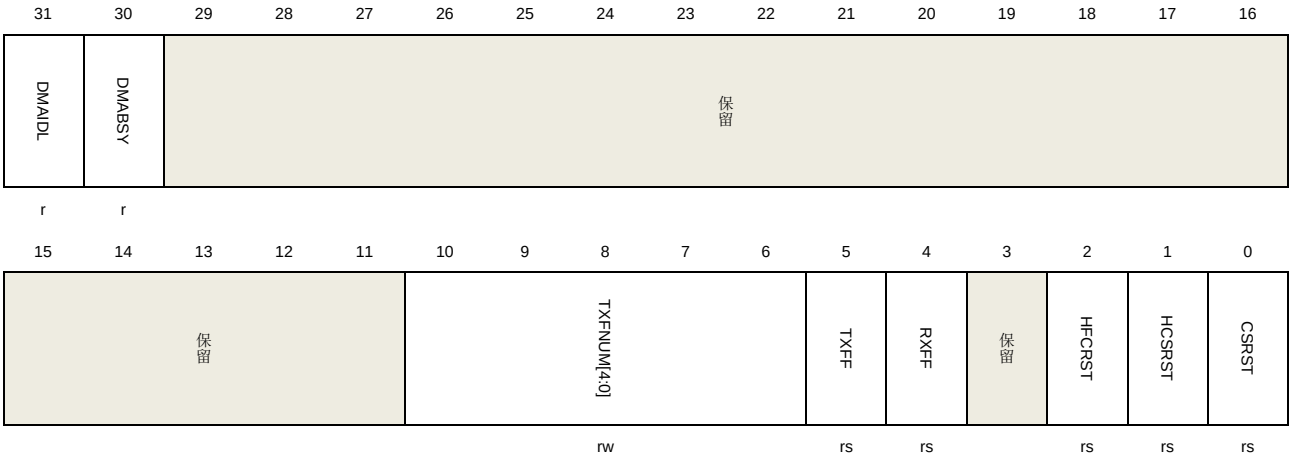
全局复位控制寄存器（USBHS_GRSTCTL）

地址偏移：0x0010

复位值：0x8000 0000

应用通过该寄存器来复位内核的不同硬件特性。

该寄存器只能按字（32位）访问



位/位域	名称	描述
31	DMAIDL	DMA空闲状态 该标志位体现了DMA是否在空闲状态 0：DMA在空闲状态 1：DMA不在空闲状态 注意：在设备和主机模式下，均可访问
30	DMABSY	DMA忙标志位 该标志位体现了DMA是否忙 0：DMA不忙 1：DMA忙 注意：在设备和主机模式下，均可访问
29:11	保留	必须保持复位值。
10:6	TXFNUM[4:0]	Tx FIFO数目 当本寄存器中TXFF控制位置位时，该标志位决定那个Tx FIFO会被冲刷 主机模式： 00000：仅非周期性Tx FIFO被冲刷 00001：仅周期性Tx FIFO被冲刷 1xxxx：周期性和非周期性Tx FIFO均被冲刷 其他：没有数据被冲刷 设备模式： 00000：仅Tx FIFO0被冲刷

00001：仅Tx FIFO1被冲刷
…
00111：仅Tx FIFO7被冲刷
1XXXX：所有的Tx FIFO均被冲刷
其他：没有数据被冲刷

5	TXFF	Tx FIFO冲刷控制位 应用通过置位该控制位来冲刷Tx FIFO数据，并且TXFNUM[4:0]决定冲刷的FIFO数目。当冲刷完成后，硬件自动清除该控制位。置位该控制位后，应用应该等待该控制位清除，并且，在此之前USBHS不应有其他操作。 注意：在设备和主机模式下，均可访问
4	RXFF	Rx FIFO冲刷控制位 应用通过置位该控制位来冲刷Rx FIFO数据。当冲刷完成后，硬件自动清除该控制位。置位该控制位后，应用应该等待该控制位清除，并且，在此之前USBHS不应有其他操作。 注意：在设备和主机模式下，均可访问
3	保留	必须保持复位值。
2	HFCRST	主机帧计数器复位 应用通过置位该控制位来复位USBHS内的帧计数器。该控制位置位后，接下来SOF的帧计数器将变为0。当复位操作完成后，硬件自动清除该控制位。置位该控制位后，应用应该等待该控制位清除，并且，在此之前USBHS不应有其他操作。 注意：仅在主机模式下访问
1	HCSRST	HCLK软件复位 应用通过置位该控制位来复位ABH时钟域电路 在复位操作完成后，硬件自动清除该控制位。置位该控制位后，应用应该等待该控制位清除，并且，在此之前USBHS不应有其他操作。 注意：在设备和主机模式下，均可访问
0	CSRST	USB内核软件复位 复位AHB和USB时钟域电路，以及大多数的寄存器。

全局中断标志寄存器（USBHS_GINTF）

地址偏移：0x0014
复位值：0x0400 0021

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WKUPIF	SESIF	DISCIF	IDPSC	LPMIF	PTXFEIF	HCIF	HPIF	保留		PXNCIF/ISOONCIF	ISOINCIF	OEPIF	IEPIF	保留	
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	r	r	r			rc_w1	rc_w1	r	r		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

EOPFIF	ISOOPDIF	ENUMF	RST	SP	ESP	保留	GONAK	GNPINAK	NPTXFEIF	RXFNEIF	SOF	OTGIF	MFIF	COPM
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1		r	r	r	r	rc_w1	r	rc_w1	r

位/位域	名称	描述
31	WKUPIF	唤醒中断标志位 当在USB总线上检测到一个恢复信号（在设备模式下）或者一个远程唤醒信号（在主机模式下），硬件将置位该中断标志位。 注意：在设备和主机模式下，均可访问
30	SESIF	会话中断标志位 当在A设备模式下检测到一个SRP会话请求或在B设备模式下B设备的Vbus变为可用时，硬件将置位该中断标志位 注意：在设备和主机模式下，均可访问
29	DISCIF	断开中断标志位 当设备断开后，将触发该标志位。 注意：仅在主机模式下访问
28	IDPSC	ID引脚状态改变中断标志位 当ID引脚状态改变时，内核将置位该标志位 注意：在设备和主机模式下，均可访问
27	LPMIF	LPM中断标志位 在主机模式下，当设备以ACK、NYET或STALL响应LPM事务，或者主机已经发送了RECNT（USBHS_LPMCFG寄存器）次LPM事务，中断会被触发 在从机模式下，当设备已经接收到LPM事务并且以ACK、NYET或STALL进行了响应，中断会被触发
26	PTXFEIF	周期性Tx FIFO空中断标志位 当周期性发送FIFO半空或全空时，将触发该标志位。空阈值由USBHS_GAHBCS寄存器中周期性Tx FIFO空等级控制位（PTXFTH）决定。 注意：仅在主机模式下访问
25	HCIF	主机通道中断标志位 当在主机模式下其中一个通道挂起一个中断时，USBHS将置位该标志位。软件应该首先读取USBHS_HACHINT寄存器以获取通道号，然后读取相应的USBHS_HCHxINTF寄存器以获取产生中断的通道标志位。当产生通道中断的独立通道标志位被清除后，该中断标志位将自动清除。 注意：仅在主机模式下访问
24	HPIF	主机端口中断标志位 当USBHS在主机模式下检测到端口状态改变时，USB内核将置位该标志位。软件应该读取USBHS_HPCSR寄存器以获取该中断源。当产生端口中断的标志被清除后，该中断标志位将自动清除。

		注意：仅在主机模式下访问
23:22	保留	必须保持复位值。
21	PXNCIF	周期性传输未完成中断标志位 在当前帧内，当帧结束时，周期性传输未完成，USBHS将置位该标志位（主机模式）。
	ISOONCIF	同步OUT传输未完成中断标志位 在周期性帧结束时（由USBHS_DCFG寄存器的EOPFT控制位定义），如果仍有同步OUT端点未完成传输，USBHS将置位该标志位（设备模式）。
20	ISOINCIF	同步IN传输未完成中断标志位 在周期性帧结束时（由USBHS_DCFG寄存器的EOPFT控制位定义），如果仍有同步IN端点未完成传输，USBHS将置位该标志位（设备模式）。 注意：仅在设备模式下访问
19	OEPIF	OUT端点中断标志位 当在设备模式下，其中一个OUT端点挂起一个中断时，USBHS将置位该中断标志位。软件应该首先读取USBHS_DAEPINT寄存器以获取设备号，然后读取相应的USBHS_ DOEPxINTF寄存器以获取产生中断的端点标志位。当产生中断的相应端点标志位被清除后，该中断标志位被自动清除。 注意：仅在设备模式下访问
18	IEPIF	IN端点中断标志位 当在设备模式下，其中一个IN端点挂起一个中断时，USBHS将置位该标志位。软件应该首先读取USBHS_DAEPINT寄存器以获取设备号，然后读取相应的USBHS_DIEPxINTF寄存器以获取产生中断的端点标志位。当相应产生中断的端点标志位被清除后，该中断标志位被自动清除。
17:16	保留	必须保持复位值。
15	EOPFIF	周期性帧结束中断标志位 当一帧内USB总线时间已经达到USBHS_DCFG寄存器中EOPFT控制位所定义的数值时，USBHS将置位该中断标志位。 注意：仅在设备模式下访问
14	ISOOPDIF	同步OUT包丢失中断标志位 如果USBHS接收到一个同步OUT包，但是Rx FIFO没有足够的空间来接收该OUT包，USBHS将置位该标志位。 注意：仅在设备模式下访问
13	ENUMF	枚举完成中断标志位 在速度枚举完成后，USBHS将置位该中断标志位。软件能够读取USBHS_DSTAT寄存器以获取当前设备速度。 注意：仅在设备模式下访问
12	RST	USB复位中断标志位 当USBHS在USB总线上检测到一个USB复位信号后，USBHS将置位该中断标志位。

		注意：仅在设备模式下访问
11	SP	USB挂起中断标志位 当USBHS检测到USB总线空闲3ms并且进入挂起状态，USBHS将置位该中断标志位。 注意：仅在设备模式下访问
10	ESP	早期挂起中断标志位 当USBHS检测到USB总线空闲3ms时，USBHS将置位该中断标志位。
9:8	保留	必须保持复位值。
7	GONAK	全局OUT NAK有效标志位 软件能够向USBHS_DCTL寄存器的SGONAK控制位写1，并且USBHS将会在SGONAK写入有效后，置位GONAK标志位。软件可通过向USBHS_DCTL寄存器的CGONAK控制位写1，清除该标志位 注意：仅在设备模式下可访问
6	GNPINAK	全局非周期性IN NAK有效标志位 软件能够向USBHS_DCTL寄存器中的SGINAK控制位写1，并且USBHS将会在SGINAK写入有效后，置位GNPINAK标志位。软件可通过向USBHS_DCTL寄存器的CGINAK控制位写1，清除该标志位 注意：仅在设备模式下可访问
5	NPTXFEIF	非周期性Tx FIFO空中断标志位 当非周期性Tx FIFO为半空或全空时，将置位该中断标志位。该阈值由USBHS_GAHBCS寄存器中的非周期Tx FIFO空等级控制位（TXFTH）决定。 注意：仅在主机模式下访问
4	RXFNEIF	Rx FIFO非空中断标志位 当至少有一个包或状态条目在Rx FIFO中时，USBHS将置位该标志位。 注意：在主机和设备模式下，均可访问
3	SOF	帧起始中断标志位 主机模式： 当准备在USB总线上发送一个SOF或保持有效信号，USBHS将置位该中断标志位。软件可以通过写1清除该中断标志位。 设备模式： 当USBHS接收到一个SOF令牌包后，USBHS置位该标志位。应用可以读取设备状态寄存器以获取当前帧号。软件可以通过写1清除该中断标志位。 注意：在设备和主机模式下，均可访问
2	OTGIF	OTG中断标志位 当USBHS_GOTGINTF寄存器中标志位产生一个中断时，USBHS置位该中断标志位。软件应该读取USBHS_GOTGINTF寄存器以获取产生该中断的信号源，当USBHS_GOTGINTF寄存器中产生该中断的标志位被清除后，该中断标志位也被自动清除。 注意：在设备和主机模式下，均可访问

1	MFIF	模式错误中断标志位 如果软件在设备模式下操作仅主机可访问的寄存器或者在主机模式下操作仅设备可访问的寄存器，USBHS将置位该中断标志位。这些错误操作不会产生作用。 注意：在主机和设备模式下，均可访问
0	COPM	当前操作模式 0：设备模式 1：主机模式 注意：在主机和设备模式下，均可访问

全局中断使能寄存器（USBHS_GINTEN）

地址偏移：0x0018

复位值：0x0000 0000

这个寄存器同全局中断标志寄存器（USBHS_GINTF）一起工作来中断应用程序。当中断使能位被禁止后，相应的中断就不会产生。然而，相应的全局中断标志位依然会被置位。

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WKUPIE	SESIE	DISCIE	IDPSCIE	LPMIE	PTXFEIE	HCIE	HPIE	保留		PXNCIE/ISOONCIE	ISOINCIE	OEPIE	IEPIE	保留	
rw	rw	rw	rw	rw	rw	rw	r			rw	rw	rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOPFIE	ISOOPDIE	ENUMFIE	RSTIE	SPIE	ESPIE	保留		GONAKIE	GNPINAKIE	NPTXFEIE	RXFNEIE	SOFIE	OTGIE	MFIE	保留
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	

位/位域	名称	描述
31	WKUPIE	唤醒中断使能 0：禁用唤醒中断 1：使能唤醒中断 注意：在主机和设备模式下，均可访问
30	SESIE	会话中断使能 0：禁用会话中断 1：使能会话中断 注意：在主机和设备模式下，均可访问
29	DISCIE	断开中断使能 0：禁用断开中断 1：使能断开中断

		注意：仅在设备模式下使用
28	IDPSCIE	ID引脚状态改变中断使能 0：禁用连接器ID引脚状态中断 1：使能连接器ID引脚状态中断 注意：在主机和设备模式下，均可访问
27	LPMIE	LPM中断使能 0：禁用LPM中断 1：使能LPM中断 注意：在主机和设备模式下，均可访问
26	PTXFEIE	周期性Tx FIFO空中断使能 0：禁用周期性Tx FIFO空中断 1：使能周期性Tx FIFO空中断 注意：仅在主机模式下访问
25	HCIE	主机通道中断使能 0：禁用主机通道中断 1：使能主机通道中断 注意：仅在主机模式下访问
24	HPIE	主机端口中断使能 0：禁止主机端口中断 1：使能主机端口中断 注意：仅在主机模式下访问
23:22	保留	必须保持复位值。
21	PXNCIE	周期性传输未完成中断使能 0：禁止周期性未完成传输中断 1：使能周期性未完成传输中断 注意：仅在主机模式下访问
	ISOONCIE	同步OUT传输未完成中断使能 0：禁止同步OUT传输未完成中断 1：使能同步OUT传输未完成中断 注意：仅在设备模式下访问
20	ISOINCIE	同步IN传输未完成中断使能 0：禁止同步IN传输未完成中断 1：使能同步IN传输未完成中断 注意：仅在设备模式下访问
19	OEPIE	OUT端点中断使能 0：禁止OUT端点中断 1：使能OUT端点中断 注意：仅在设备模式下访问
18	IEPIE	IN端点中断使能

		0：禁止IN端点中断 1：使能IN端点中断 注意：仅在设备模式下访问
17:16	保留	必须保持复位值。
15	EOPFIE	周期性帧结束中断使能 0：禁止周期性帧结束中断 1：使能周期性帧结束中断 注意：仅在设备模式下访问
14	ISOOPDIE	同步OUT包丢失中断使能 0：禁止同步OUT包丢失中断 1：使能同步OUT包丢失中断 注意：仅在设备模式下访问
13	ENUMFIE	枚举完成中断使能 0：禁止枚举完成中断 1：使能枚举完成中断 注意：仅在设备模式下访问
12	RSTIE	USB复位中断使能 0：禁止USB复位中断 1：使能USB复位中断 注意：仅在设备模式下访问
11	SPIE	USB挂起中断使能 0：禁止USB挂起中断 1：使能USB挂起中断 注意：仅在设备模式下访问
10	ESPIE	早期挂起中断使能 0：禁止早期挂起中断 1：使能早期挂起中断 注意：仅在设备模式下访问
9:8	保留	必须保持复位值。
7	GONAKIE	全局OUT NAK有效中断使能 0：禁止全局OUT NAK有效中断 1：使能全局OUT NAK有效中断 注意：仅在设备模式下访问
6	GNPINAKIE	全局非周期性IN NAK有效中断使能 0：禁止全局非周期性IN NAK有效中断 1：使能全局非周期性IN NAK有效中断 注意：仅在设备模式下访问
5	NPTXFEIE	非周期性发送FIFO空中断使能 0：禁止非周期性发送FIFO空中断

		1：使能非周期性发送FIFO空中断 注意：仅在主机模式下访问
4	RXFNEIE	接收FIFO非空中断使能 0：禁止接收FIFO非空中断 1：使能接收FIFO非空中断 注意：在设备模式与主机模式下，均可访问
3	SOFIE	帧首中断使能 0：禁止帧首中断 1：使能帧首中断 注意：在设备模式下与主机模式下，均可访问
2	OTGIE	OTG中断使能 0：禁止OTG中断 1：使能OTG中断 注意：在设备模式下与主机模式下，均可访问
1	MFIE	模式错误中断使能 0：禁止模式错误中断 1：使能模式错误中断 注意：在设备模式下与主机模式下，均可访问
0	保留	必须保持复位值。

全局接收状态读取/接收状态读取和弹出寄存器（USBHS_GRSTATR/USBHS_GRSTATP）

读地址偏移：0x001C
弹出地址偏移：0x0020
复位值：0x0000 0000

对接收状态读寄存器的读操作，将返回接收FIFO中顶部的条目。对接收状态读取和弹出寄存器的读操作，将额外的弹出Rx FIFO的顶部条目。

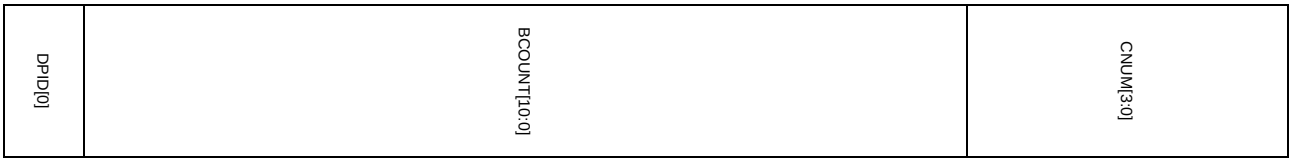
在主机模式和设备模式下，Rx FIFO中的条目具有不同的含义。当全局中断标志寄存器（USBHS_GINTF）中的接收FIFO非空中断标志位（RXFNEIF）置位后，软件应该读取该寄存器。

该寄存器只能按字(32位)访问

主机模式：

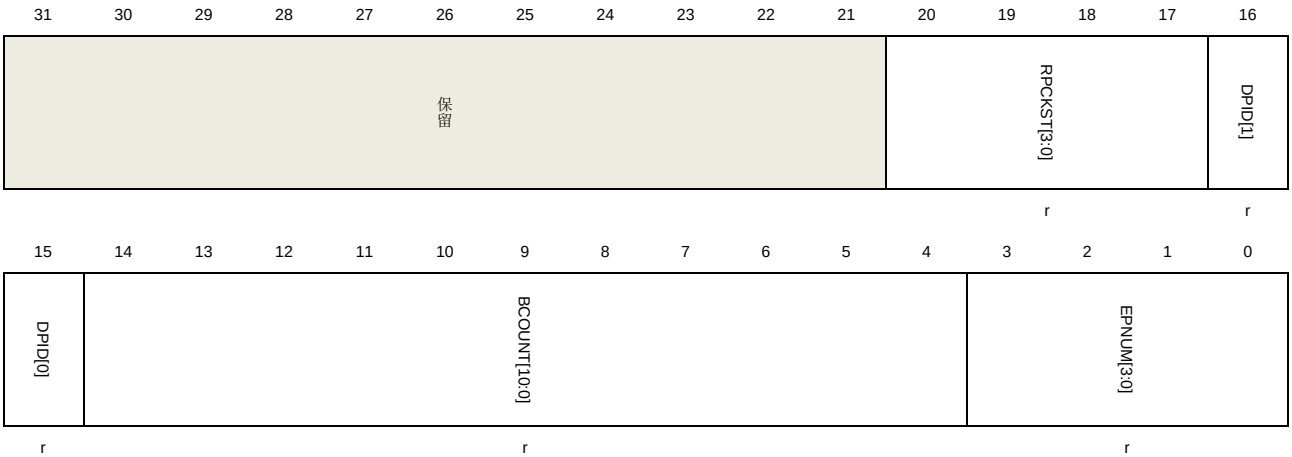
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留											RPCKST[3:0]				DPID[1]
											r				r

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



位/位域	名称	描述
31:21	保留	必须保持复位值。
20:17	RPCKST[3:0]	接收包状态 0010：接收到IN数据包 0011：IN传输完成（如果取出，触发一个中断） 0101：数据翻转错误（如果取出，触发一个中断） 0111：通道中止（如果取出，触发一个中断） 其他：保留
16:15	DPID[1:0]	数据PID 接收包的数据PID 00：DATA0 10：DATA1 01：DATA2 11：MDATA
14:4	BCOUNT[10:0]	字节数 接收IN数据包字节数。
3:0	CNUM[3:0]	通道数 当前接收包所属通道编号。

设备模式：



位/位域	名称	描述
31:21	保留	必须保持复位值。
20:17	RPCKST[3:0]	接收包状态 0001：全局OUT NAK（产生一个中断）

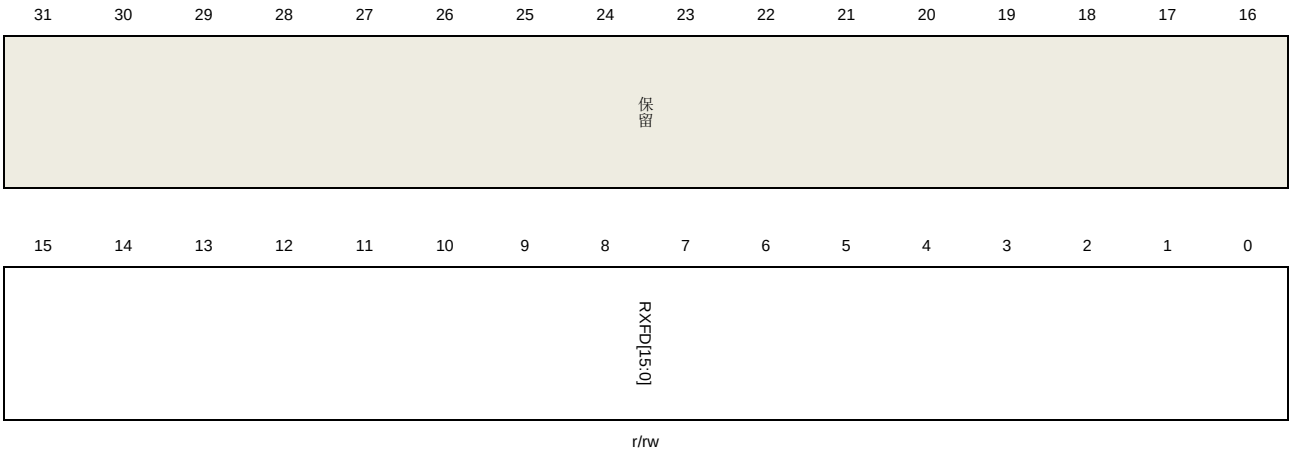
0010：接收到OUT数据包
0011：OUT传输完成（产生一个中断）
0100：SETUP传输完成（产生一个中断）
0110：接收到SETUP数据包
其他：保留

16:15	DPID[1:0]	数据PID 接收到OUT数据包的数据PID 00：DATA0 10：DATA1 01：DATA2 11：MDATA
14:4	BCOUNT[10:0]	字节数 接收数据包的字节数
3:0	EPNUM[3:0]	端点号 当前接收包所属端点编号

全局接收 FIFO 长度寄存器（USBHS_GRFLEN）

地址偏移：0x0024
复位值：0x0000 0200

该寄存器只能按字（32位）访问



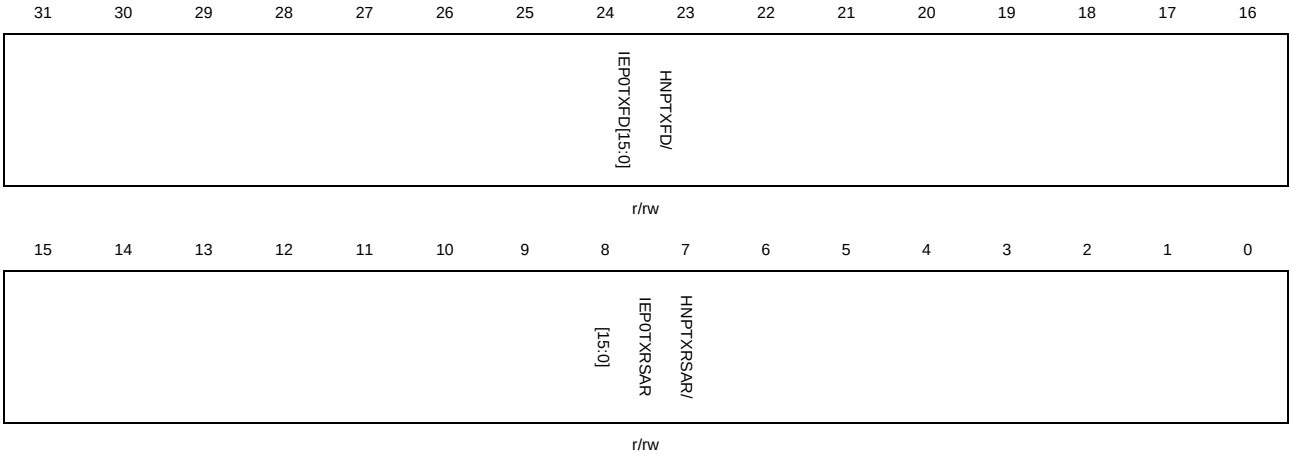
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	RXFD[15:0]	Rx FIFO 深度 以32位字计数 1≤RXFD≤1024

主机非周期性发送 FIFO 长度寄存器/设备 IN 端点 0 发送 FIFO 长度寄存器（USBHS_HNPTFLEN/USBHS_DIEP0TFLEN）

地址偏移：0x0028

复位值：0x0200 0200

该寄存器只能按字（32位）访问



主机模式下：

位/位域	名称	描述
31:16	HNPTXFD[15:0]	主机非周期性Tx FIFO深度 以32位字计数 1≤HNPTXFD≤1024
15:0	HNPTXRSAR[15:0]	主机非周期性Tx RAM起始地址 非周期性发送FIFO RAM的起始地址

设备模式下：

位/位域	名称	描述
31:16	IEP0TXFD[15:0]	输入端点0 Tx FIFO深度 以32位字计数 16≤IEP0TXFD≤140
15:0	IEP0TXRSAR[15:0]	输入端点0 TX RAM起始地址 端点0发送FIFO RAM的起始地址

主机非周期性发送 FIFO/队列状态寄存器（USBHS_HNPTFQSTAT）

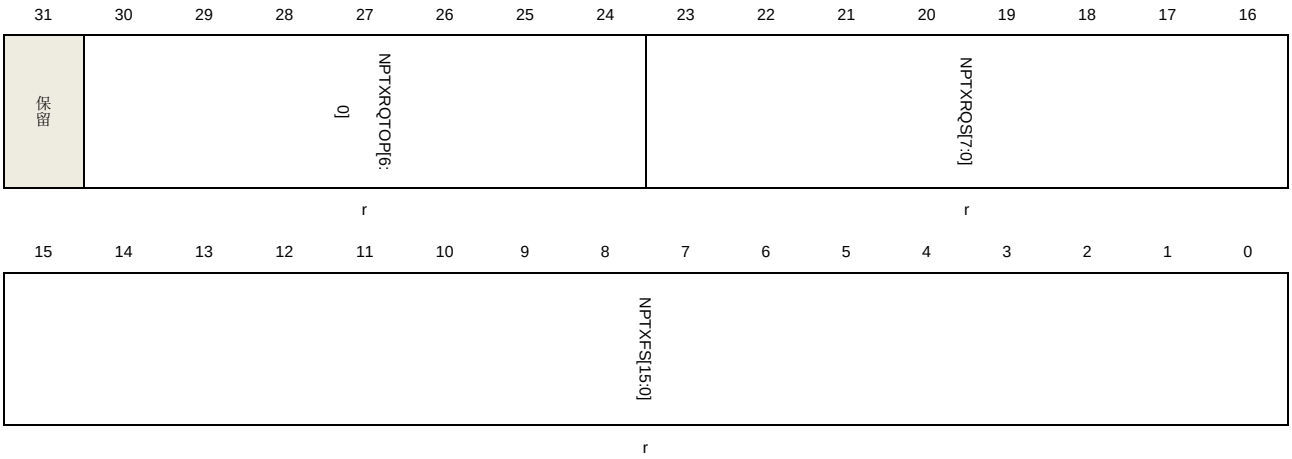
地址偏移：0x002C

复位值：0x0008 0200

该寄存器只能按字（32位）访问

该寄存器反映了非周期性Tx FIFO和请求队列的当前状态。
请求队列包括在主机模式下的IN、OUT或其他请求条目。

注意：在设备模式下，该寄存器不可用。



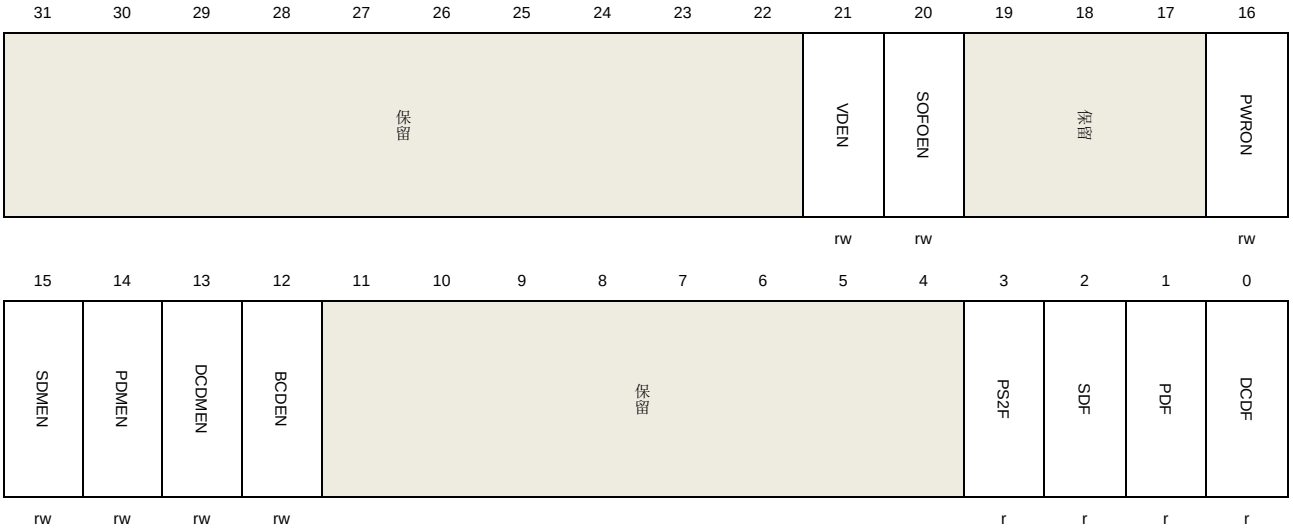
位/位域	名称	描述
31	保留	必须保持复位值。
30:24	NPTXRQTOP[6:0]	非周期性发送请求队列的顶部条目 在非周期性传输请求队列中的条目。 位30:27：通道号 位26:25： – 00：IN/OUT令牌 – 01：0长度OUT包 – 11：通道中止请求 位24：结束标志位，表明所选通道的最后一个条目
23:16	NPTXRQS[7:0]	非周期性发送请求队列空间 非周期性请求队列的剩余空间 0：请求队列空 1：1个条目 2：2个条目 … n：n个条目 (0≤n≤8) 其他：保留
15:0	NPTXFS[15:0]	非周期性Tx FIFO空间 非周期性发送FIFO剩余空间 以32位字计数 0：非周期性Tx FIFO为空 1：1个字 2：2个字 … n：n个字(0≤n≤NPTXFD) 其他：保留

全局内核配置寄存器（USBHS_GCCFG）

地址偏移：0x0038

复位值：0x0000 0000

该寄存器只能按字（32位）访问



位/位域	名称	描述
31:22	保留	必须保持复位值。
21	VDEN	启用VBUS感应比较器检测VBUS有效。如果支持HNP或SRP，则自动启用VBUS比较器。 0：禁用V_{BUS}检测 1：使能V_{BUS}检测
20	SOFOEN	SOF输出使能 0：SOF脉冲输出禁止 1：SOF脉冲输出使能
19:17	保留	必须保持复位值。
16	PWRON	上电 该控制位为内部嵌入式全速PHY的电源开关 0：嵌入式全速PHY掉电 1：嵌入式全速PHY上电
15	SDMEN	二次检测模式使能 0：二次检测模式禁止 1：二次检测模式使能
14	PDMEN	主检测模式使能 0：主检测模式禁止 1：主检测模式使能
13	DCDMEN	数据连接检测模式使能 0：数据连接检测模式禁止 1：数据连接检测模式使能
12	BCDEN	电池充电检测使能

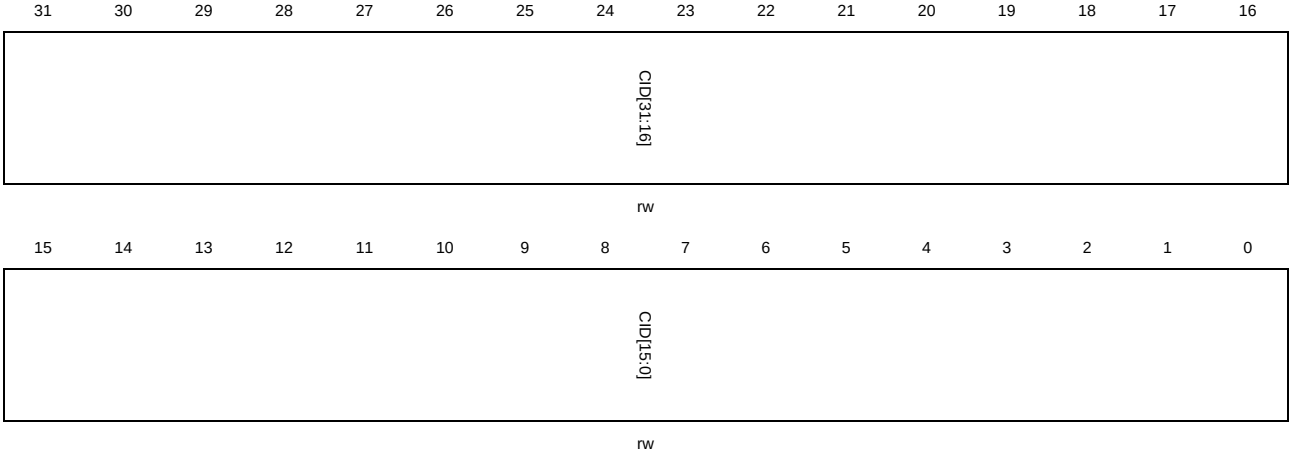
		0：电池充电检测禁止 1：电池充电检测使能
11:4	保留	必须保持复位值。
3	PS2F	PS2检测状态，仅在主检测模式下激活 0：检测到普通端口 1：检测到PS2端口
2	SDF	二次检测状态 0：检测到CDP 1：检测到DCP
1	PDF	主检测状态 0：检测到BCD支持 1：检测到BCD支持
0	DCDF	数据连接检测状态 0：未检测到数据线连接 1：检测到数据线连接

内核 ID 寄存器（USBHS_CID）

地址偏移：0x003C
复位值：0x0000 1000

该寄存器包含产品ID

该寄存器只能按字（32位）访问



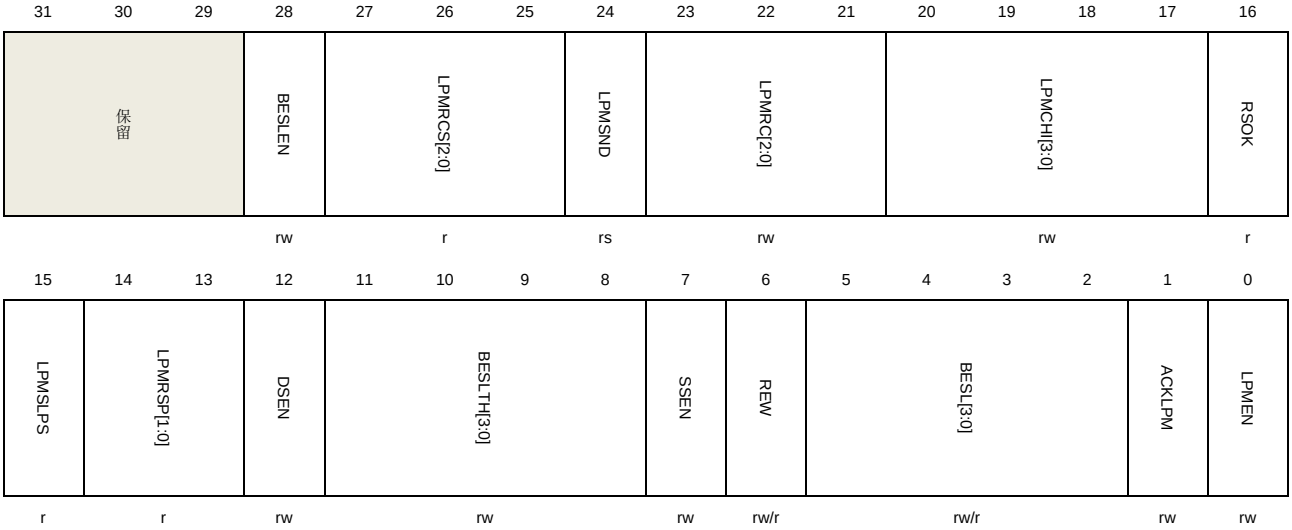
位/位域	名称	描述
31:0	CID[31:0]	内核ID 软件能够写入或读取该域值，并利用该域值为应用产生一个唯一ID。

全局内核 LPM 配置寄存器（USBHS_GLPMCFG）

地址偏移：0x0054

复位值：0x0000 0000

该寄存器只能按字（32位）访问



位/位域	名称	描述
31:29	保留	必须保持复位值。
28	BESLEN	使能 LPM 勘误表选项 0：选择 USB2.0 规范的链路电源管理附录工程变更通知 1：选择 USB2.0 规范的链路电源管理工程变更通知的勘误表
27:25	LPMRCS[2:0]	LPM 重试计数状态 注意：仅在主机模式下访问
24	LPMSND	发送 LPM 事务 当收到 ACK、STALL 或 NYET 响应，或者 LPM 发送次数达到重试计数，由硬件清除该位 注意：仅在主机模式下访问
23:21	LPMRC[2:0]	LPM 重试计数 接收到错误响应时的重试计数，直到接收到 ACK、STALL 或 NYET 响应 注意：仅在主机模式下访问
20:17	LPMCHI[3:0]	发送 LPM 事务时的通道号索引 注意：仅在主机模式下访问
16	RSOK	睡眠状态可以发送唤醒信号 主机或设备可以在进入睡眠状态 50us 后发送唤醒（TI1Residency） 当 LPMSLPS 为 0 时该位为 0 1：睡眠状态可以启动唤醒 0：睡眠状态不能启动唤醒

15	LPMSLPS	睡眠状态 **主机模式：** 主机在接收到 ACK 响应后转换到睡眠状态。 **从机模式：** 发送 ACK 响应后，设备进入睡眠状态，TL1 令牌重发计时器已过期。 1：内核在睡眠状态 0：内核不在睡眠状态
14:13	LPMRSP[1:0]	LPM 响应 11：ACK 10：NYET 01：STALL 00：ERROR（没有响应）
12	DSEN	深度睡眠使能 使能在深度睡眠模式下挂起 PHY
11:8	BESLTH[3:0]	BESL 阈值 设备模式： 当 BESL 大于或等于 BESLTH 值时，设备进入深低功耗模式。 主机模式： BESLTH 表示当检测到设备初始化的恢复时，恢复信号（TL1HubDrvResume2）的持续时间。 0000：75us 0001：100us 0010：150us 0011：250us 0101：450us 0110：950us
7	SSEN	浅睡眠下使能 在浅睡眠模式下使能挂起 PHY
6	REW	b-远程唤醒值 主机模式： LPM 事务中发送的远程唤醒值 设备模式（只读）： 当 ACK、STALL 或 NYET 已经发送，用接收到的 LPM 事务中的 b-远程唤醒值更新
5:2	BESL[3:0]	最大限度的服务延迟 主机模式： 要在 LPM 事务中发送的 BESL 值。这也是主机初始化 resume 时 resume（TL1HubDrvResume1）的持续时间。 设备模式： 当 ACK、STALL 或 NYET 已经发送，用接收到的 LPM 事务中的 BESL 值更新 0000：125us 0001：150us

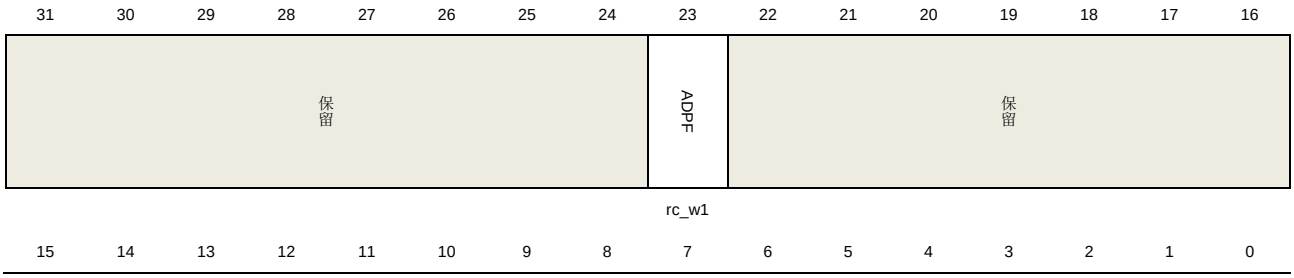
0010：200us
0011：300us
0100：400us
0101：500us
0110：1000us
0111：2000us
1000：3000us
1001：4000us
1010：5000us
1011：6000us
1100：7000us
1101：8000us
1110：9000us
1111：10000us

1	ACKLPM	在 LPM 事务中使能 ACK 1：ACK 设备仅在成功的 LPM 事务上使用 ACK 进行响应 - LPM 事务中无错误 - 无数据挂起错误 - bLinkState = 0001 在接收的 LPM 事务中 0：NYET 设备使用 NYET 进行响应 - 接收的 bLinkState 值不是 0001 - 接收 LPM 事务时产生了一个错误 注意：仅在设备模式下访问
0	LPMEN	LPM 使能 1：使能 LPM 0：禁用 LPM

断电寄存器（USBHS_PWRD）

地址偏移：0x0058
复位值：0x0000 0000

该寄存器只能按字（32位）访问





位/位域	名称	描述
31:24	保留	必须保持复位值。
23	ADPF	ADP 事件中断标志位
22:1	保留	必须保持复位值。
0	ADPMEN	ADP 模块使能 1：ADP 模块使能 0：ADP 模块禁用

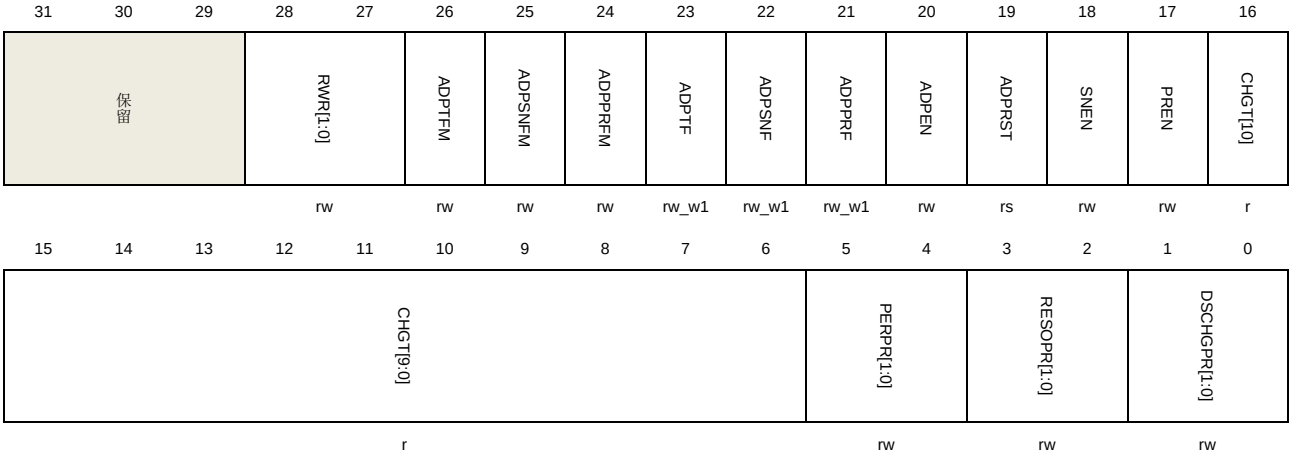
ADP 控制和状态寄存器（USBHS_ADPCTL）

地址偏移：0x0060
复位值：0x0000 0000

为了写入该寄存器，需要写入RWR为10，并保持轮询直到RWR=00。

为了读取该寄存器，需要等待任何ADP标志被置位或用01写入RWR并保持轮询直到RWR=00。

该寄存器只能按字（32位）访问



位/位域	名称	描述
31:29	保留	必须保持复位值。
28:27	RWR[1:0]	读和写请求 00：读和写有效（通过内核更新） 01：读请求 10：写请求

26	ADPTFM	ADP 超时中断标志的掩码
25	ADPSNFM	ADP 检测中断标志的掩码
24	ADPPRFM	ADP 探测中断标志的掩码
23	ADPTF	ADP 超时中断标志
22	ADPSNF	ADP 检测中断标志
21	ADPPRF	ADP 探测中断标志
20	ADPEN	ADP 使能 1：ADP 使能 0：ADP 禁用
19	ADPRST	ADP 复位 复位完成后自动清除
18	SNEN	ADP 检测使能 1：检测使能 0：检测禁止
17	PREN	ADP 探测使能 1：探测使能 0：探测禁止
16:6	CHGT[10:0]	VBUS 从 VADPSINK 渐变到 VADPPRB 的最新时间。这些位以 32 kHz 时钟周期为单位定义。 000：1 个周期 001：2 个周期 002：3 个周期 003：4 个周期 … 7ff：2048 个周期
5:4	PERPR[1:0]	探测周期 00：0.625s 到 0.925s 01：1.25s 到 1.85s 10：1.9s 到 2.6s
3:2	RESOPR[1:0]	CHGT 值的分辨率。这些位以 32 kHz 时钟周期为单位定义。如果选择 10，则 CHGT 每 3 个 32 kHz 时钟周期递增一次。 00：1 个周期 01：2 个周期 10：3 个周期 11：4 个周期
1:0	DSCHGPR[1:0]	探测放电时间 00：4 ms 01：8 ms

10：16 ms

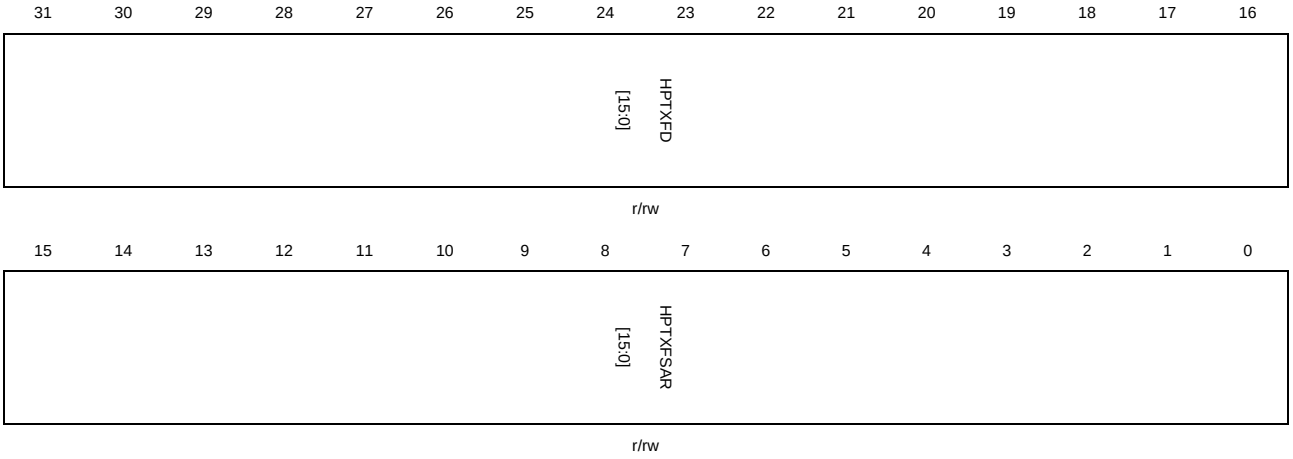
11：32 ms

主机周期性发送 FIFO 长度寄存器（USBHS_HPTFLEN）

地址偏移：0x0100

复位值：0x0200 0600

该寄存器只能按字（32位）访问



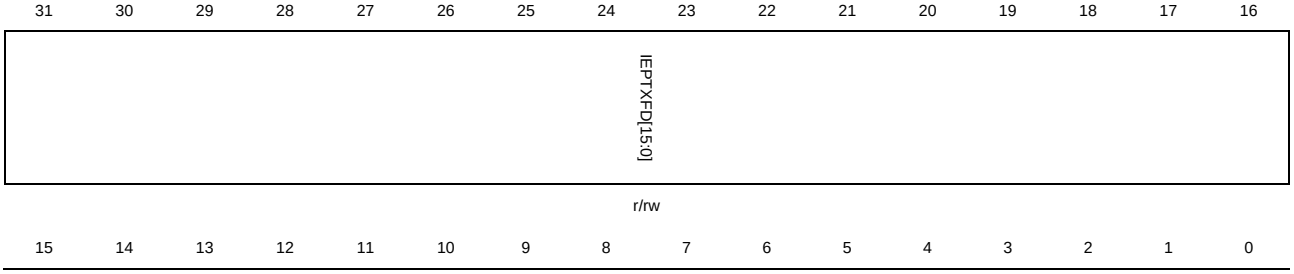
位/位域	名称	描述
31:16	HPTXFD[15:0]	主机周期性Tx FIFO深度 以32位字计数 1≤HPTXFD≤1024
15:0	HPTXFSAR[15:0]	主机周期性Tx RAM起始地址 主机周期性发送FIFO RAM起始地址

设备 IN 端点发送 FIFO 长度寄存器 （USBHS_DIEPxTFLEN）（x = 1..7，其中 x 为 FIFO 编号）

地址偏移：0x0104 + (FIFO_number – 1) × 0x04

复位值：0x0200 0400

该寄存器只能按字（32位）访问





位/位域	名称	描述
31:16	IEPTXFD[15:0]	IN端点Tx FIFO深度 以32位字计数 1≤HPTXFD≤1024
15:0	IEPTXRSAR[15:0]	IN端点FIFOx Tx RAM起始地址 以32位字为单位的IN端点发送FIFOx起始地址

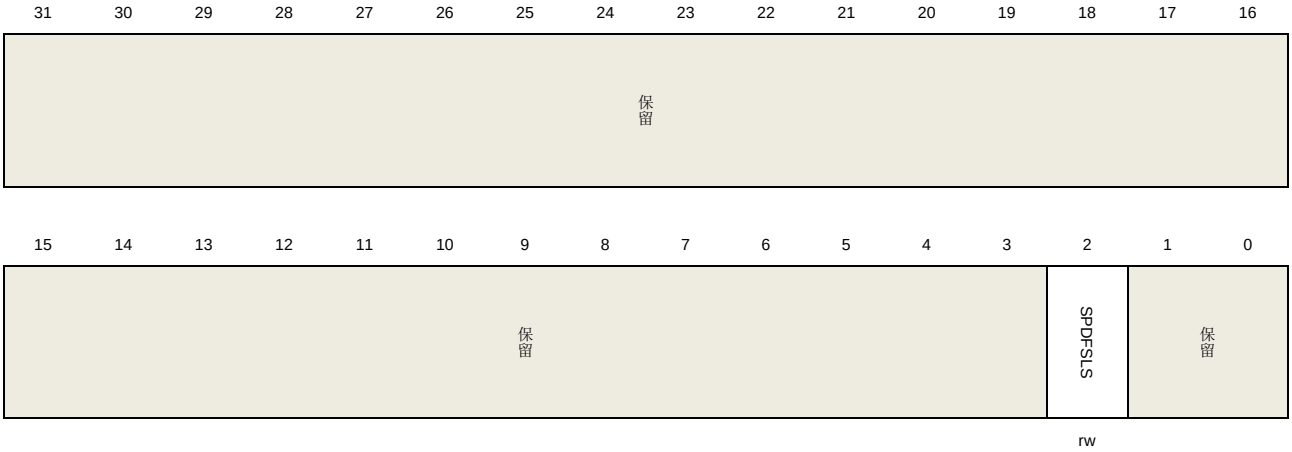
49.7.2. 主机控制和状态寄存器

主机控制寄存器（USBHS_HCTL）

地址偏移：0x0400
复位值：0x0000 0000

在主机模式下，上电后，该寄存器有USB内核配置。主机初始化后，无需修改。

该寄存器只能按字（32位）访问



位/位域	名称	描述
31:3	保留	必须保持复位值。
2	SPDFSLS	限制速度为FS和LS 软件可以利用该控制位限制USBHS的枚举速度为FS/LS，并且使USBHS在复位的过程中不执行高速枚举。 0：不限制速度 1：限制速度仅为FS/LS
1:0	保留	必须保持复位值。

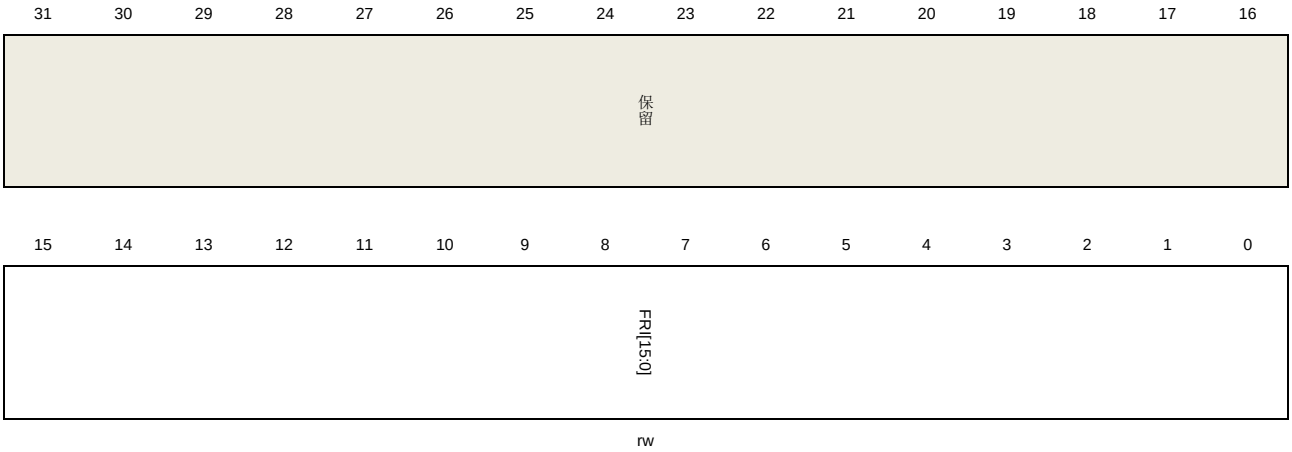
主机帧间隔寄存器（USBHS_HFT）

地址偏移：0x0404

复位值：0x0000 EA60

当USBHS控制器正在枚举中时，该寄存器为当前枚举速度设置帧间隔。

该寄存器只能按字（32位）访问



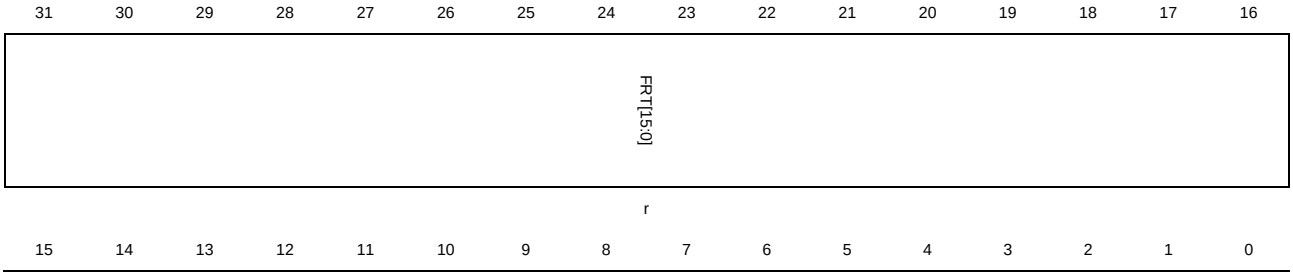
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	FRI[15:0]	帧间隔 该值描述了以PHY时钟为单位的帧周期。每次端口复位操作后，端口被使能，USBHS根据当前速度，采用一个固有值，并且软件可以向该位域写值以改变该固有值。该值需要采用以下描述的频率来进行计算： 内部嵌入式PHY 高速：60MHz 全速 / 低速：48MHz 外部ULPI PHY：60MHz 对于GD32H75E，低速：6MHz

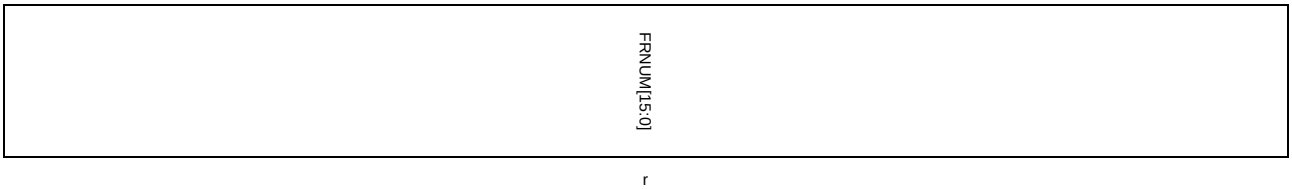
主机帧信息保持寄存器（USBHS_HFINFR）

地址偏移：0x0408

复位值：0xEA60 0000

该寄存器只能按字（32位）访问





位/位域	名称	描述
31:16	FRT[15:0]	帧剩余时间 该位域以PHY时钟为单位反映了当前帧剩余时间。
15:0	FRNUM[15:0]	帧号 该位域反映了当前帧的帧号，当其增加到0x3FFF后，其值变为0。

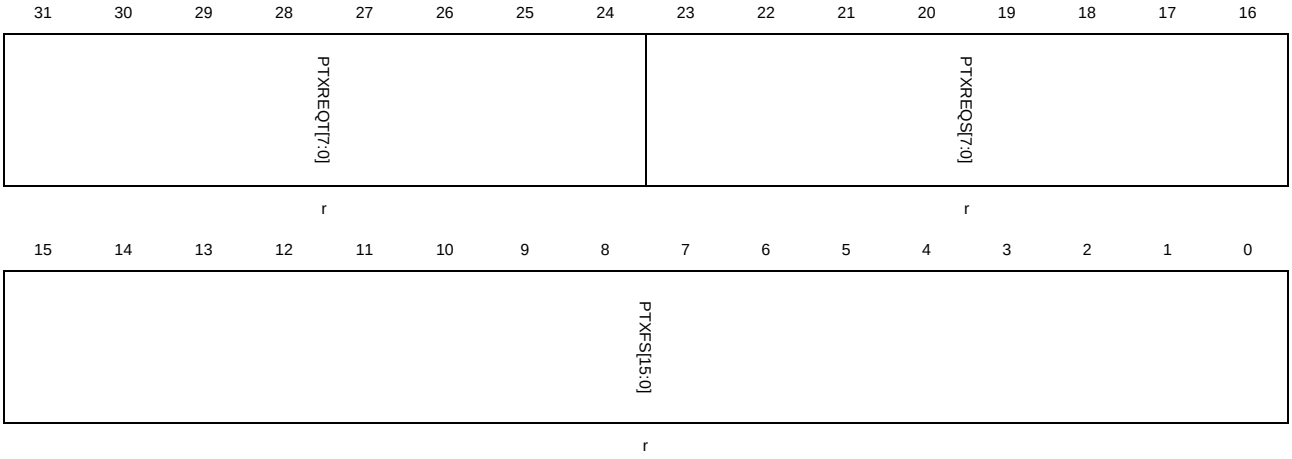
主机周期性发送 FIFO/队列状态寄存器（USBHS_HPTFQSTAT）

地址偏移：0x0410

复位值：0x0008 0200

该寄存器反映了主机周期性Tx FIFO和请求队列的当前状态。请求队列包括在主机模式下的IN、OUT或其他请求条目。

该寄存器只能按字（32位）访问



位/位域	名称	描述
31:24	PTXREQT[7:0]	周期性Tx 请求队列的顶部条目 在周期性发送请求队列中的条目 位31：奇偶帧 – 0：奇数帧 – 1：偶数帧 位30:27：通道号 位26:25： – 00：IN/OUT 令牌 – 01：0长度OUT包 – 11：通道中止请求

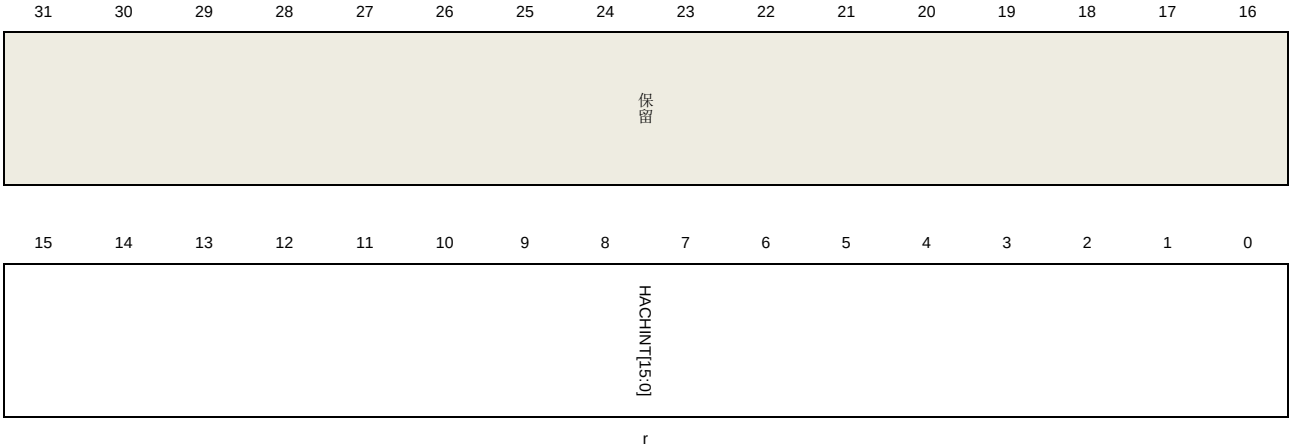
		位24：中止标志，指示所选通道的最后一个条目
23:16	PTXREQS[7:0]	周期性发送请求队列空间 周期性发送请求队列剩余空间 0：请求队列为空 1：1个条目 2：2个条目 … n：n个条目 (0≤n≤8) 其他：保留
15:0	PTXFS[15:0]	周期性发送FIFO空间 周期性发送FIFO剩余空间 以32位字计数 0：周期性发送FIFO为空 1：1个字 2：2个字 … n：n个字（0≤n≤PTXFD)） 其他：保留

主机所有通道中断寄存器（USBHS_HACHINT）

地址偏移：0x0414
复位值：0x0000 0000

当触发一个通道中断时，USBHS在该寄存器中置位相应的位，并且软件可以读取该寄存器以获取产生中断的通道。

该寄存器只能按字（32位）访问



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	HACHINT[15:0]	主机所有通道中断

每一位表示一个通道：位0代表通道0，位15表示通道15

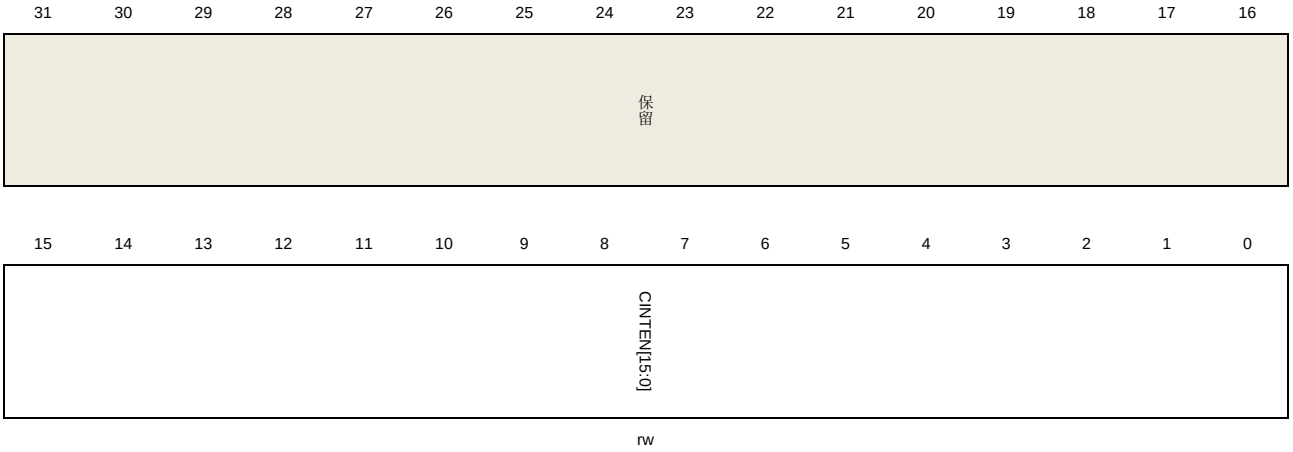
主机所有通道中断使能寄存器（USBHS_HACHINTEN）

地址偏移：0x0418

复位值：0x0000 0000

软件可以使用该寄存器使能或禁用一个通道的中断。只有该寄存器中相应通道的中断使能控制位被置位，USBHS_GINTF寄存器中的通道中断标志位HCIF标志位才可产生。

该寄存器只能按字（32位）访问



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CINTEN[15:0]	通道中断使能 0：禁用通道n中断 1：使能通道n中断 每一位表示一个通道：位0代表通道0，位15代表通道15

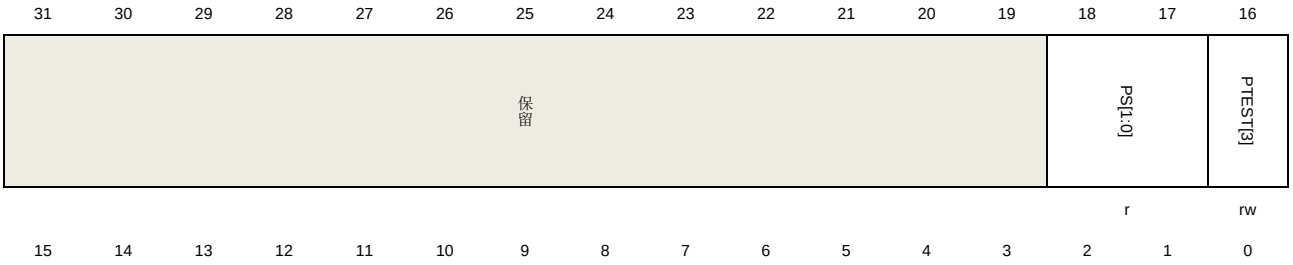
主机端口控制和状态寄存器（USBHS_HPCS）

地址偏移：0x0440

复位值：0x0002 0000

该寄存器控制端口行为，并且也包含一些反映端口状态的标志位。如果本寄存器中的PRST、PEDC和PCD标志位被USBHS置位的话，USBHS_GINTF寄存器中的HPIF标志位会被置位。

该寄存器只能按字（32位）访问



PTEST[2:0]	PP	PLST[1:0]	保留	PRST	PSP	PREM	保留	PEDC	PE	PCD	PCST
rw	rw	r		rw	rs	rw		rc_w1	rc_w1	rc_w1	r

位/位域	名称	描述
31:19	保留	必须保持复位值。
18:17	PS[1:0]	端口速度 反映连接到该端口的设备的枚举速度。 00：高速 01：全速 10：低速 其他：保留
16:13	PTEST[3:0]	端口测试控制 软件向该字段写入一个非零值以使端口进入测试模式，相应的模式发送到端口上。使用测试模式时，还应设置USBHS_GUSBCS寄存器中的HS_CUR_FE位。 0000：测试模式禁止 0001：Test_J模式 0010：Test_K模式 0011：Test_SE0_NAK模式 0100：Test_Packet模式 0101：强制测试使能 其他：保留
12	PP	端口供电 在端口被使用后，该控制位应该被置位。由于USBHS不具有电源供应能力，它只能使用该控制位以获取该端口是否在供电状态。软件应该在设置该控制位之前，保证在V_{BUS}引脚上具有电源供应。 0：端口掉电 1：端口供电
11:10	PLST[1:0]	端口线状态 反映USB数据线当前状态 位10：DP线状态 位11：DM线状态
9	保留	必须保持复位值。
8	PRST	端口复位 应用通过设置该控制位以在USB端口上启动一个复位信号。当应用希望停止复位信号时，应用应该清除该控制位。 0：端口不在复位状态 1：端口处于复位状态
7	PSP	端口挂起

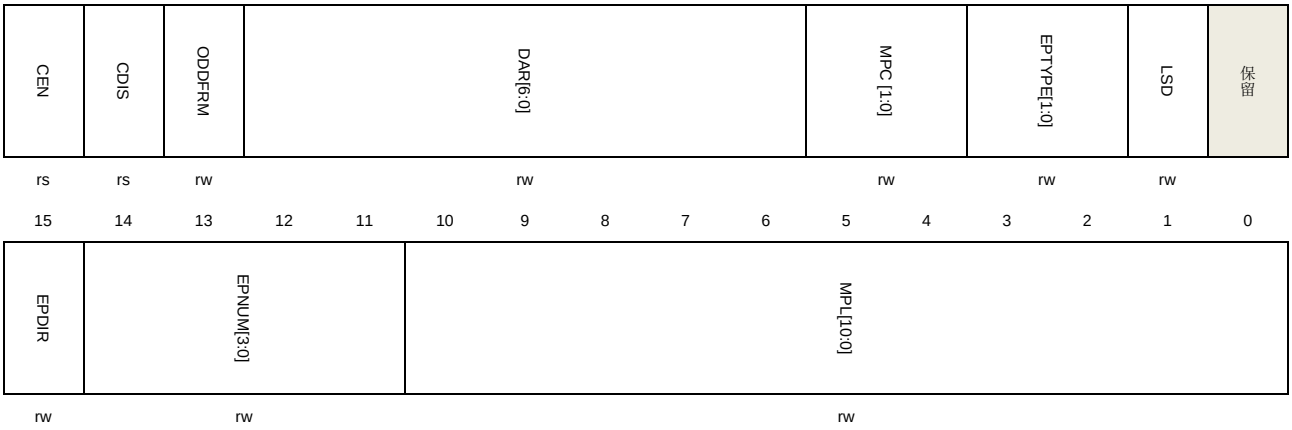
		应用设置该控制位来将端口进入挂起状态。当该控制位被置位后，端口停止发送SOF令牌包。该控制位只能够通过以下操作清除。 – 应用置位该寄存器中的PRST控制位 – 置位该寄存器中的PREM控制位 – 检测到一个远程唤醒信号 – 检测到一个设备断开 0：端口不在挂起状态 1：端口处于挂起状态
6	PREM	端口恢复 应用通过置位该控制位以在USB端口上启动一个恢复信号。当应用希望停止恢复信号时，应用可以清除该控制位。 0：无恢复驱动 1：恢复驱动 当应用程序在睡眠状态将PREM置1时，内核继续驱动恢复信号，直到计时器计数到BESLTH指定的时间。当内核检测到USB远程唤醒时，开始驱动恢复信号，并在恢复结束时自动将其清除。
5:4	保留	必须保持复位值。
3	PEDC	端口使能/禁止更改 当该寄存器中的位2端口使能控制位更改时，USB内核置位该标志位。
2	PE	端口使能 当USB复位信号完成后，USBHS自动置位该位，并且该位不可由软件置位。 该位可通过以下事件清除： – 一个断开状态 – 软件清除该位 0：端口禁止 1：端口使能
1	PCD	端口连接检测 当检测到设备连接时，USBHS置位该标志位。可通过向该位写1清除该标志位。
0	PCST	端口连接状态 0：设备没有连接到该端口 1：设备连接到该端口

主机通道 x 控制寄存器（USBHS_HCHxCTL）（x = 0..15，其中 x 为通道号）

地址偏移：0x0500 + （通道号 × 0x20）
复位值：0x0000 0000

该寄存器只能按字（32位）访问

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



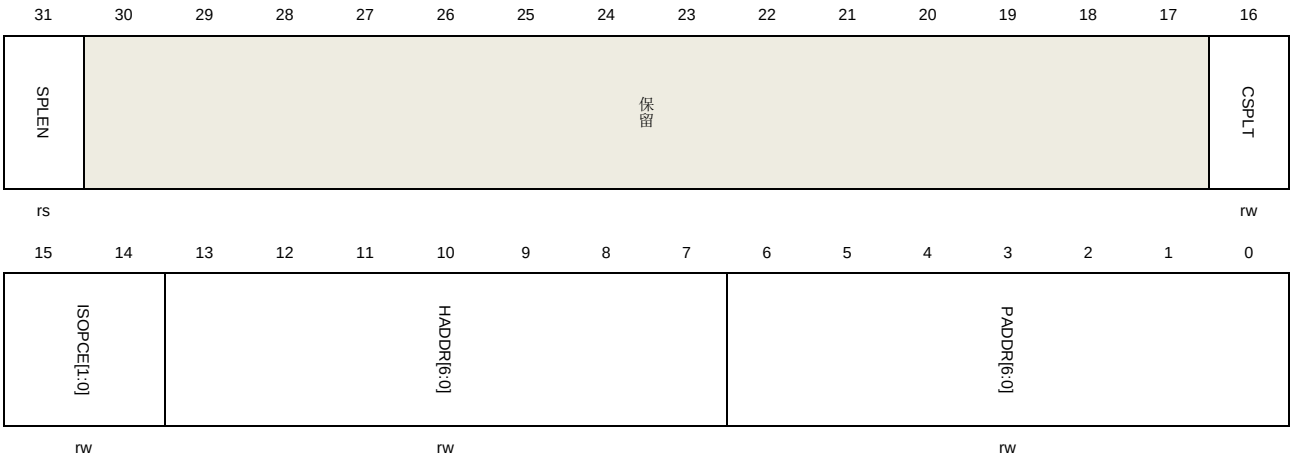
位/位域	名称	描述
31	CEN	通道使能 由应用设置，并且由USBHS清除 0：通道禁止 1：通道使能 软件应该遵循操作指南来禁用或者使能一个通道
30	CDIS	通道禁止 软件可以置位该控制位，来从处理事务中禁用该通道。软件应该遵循操作指南来禁用或者使能一个通道。
29	ODDFRM	奇偶帧控制 对于周期性传输（中断或同步传输），该位控制将要处理的通道事务为奇数帧还是偶数帧。
28:22	DAR[6:0]	设备地址 与该通道通信的USB设备地址。
21:20	MPC[1:0]	多包计数 对于周期性传输，该位域指定主机每个微帧必须执行的事务数量。对于非周期性传输，该位域指定在内部DMA引擎更改仲裁之前，DMA为此通道获取或写入的包数量。 00：保留 01：每微帧发出1个事务 10：每微帧发出2个事务 11：每微帧发出3个事务
19:18	EPTYPE[1:0]	端点类型 与该通道通信的端点的传输类型 00：控制 01：同步 10：批量 11：中断
17	LSD	低速设备 与该通道通信的设备是一个低速设备。

16	保留	必须保持复位值。
15	EPDIR	端点方向 与该通道通信的端点的传输方向 0：OUT 1：IN
14:11	EPNUM[3:0]	端点号 与该通道通信的端点号
10:0	MPL[10:0]	最大包长 目标端点的最大包长

主机通道 x 分裂事务控制寄存器（USBHS_HCHxSTCTL）（x = 0..15，其中 x = 通道号）

地址偏移：0x0504 + （通道号 × 0x20）
复位值：0x0000 0000

该寄存器只能按字（32位）访问



位/位域	名称	描述
31	SPLEN	使能高速分裂事务 软件可以置位该控制位以使能在该通道上的高速分裂事务。分裂事务用于通过HUB和一些全速和低速设备端点初始化全速/低速事务。
30:17	保留	必须保持复位值。
16	CSPLT	完全分裂使能 软件能够置位该控制位来使USBHS执行完全分裂事务，另外，USBHS执行起始分裂事务。
15:14	ISOPCE[1:0]	同步OUT负载延续编码 对于全速同步OUT起始分裂，该位域指明高速数据负载如何对应全速数据包。 00：高速数据在全速数据负载的中间 01：高速数据在全速数据负载的尾端

		10：高速数据在全速数据负载的起始 11：高速数据为全部的全速数据包
13:7	HADDR[6:0]	HUB地址 该位域为处理全速和低速事务并支持全速和低速设备的HUB地址
6:0	PADDR[6:0]	端口地址 该位域包含当前全速或低速事务传输中目标HUB的端口号

主机通道 x 中断标志寄存器（USBHS_HCHxINTF）（x = 0..15，其中 x = 通道号）

地址偏移：0x0508 + （通道号 × 0x20）
复位值：0x0000 0000

该寄存器包含一个通道的状态和事件，当软件获取一个通道中断时，软件需要为相应通道读取该寄存器以获取产生中断的中断源。该寄存器中的标志位均由硬件置位，并且写1清除。

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	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					DTER	REQOVR	BBER	USBER	NYET	ACK	NAK	STALL	DMAER	CH	TF
					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
31:11	保留	必须保持复位值。
10	DTER	数据切换错误 IN事务获取一个数据包，但是该包的PID和USBHS_HCHxLEN寄存器中的DPID[1:0]控制位不匹配。
9	REQOVR	请求队列上溢 当软件启动新的传输时，请求队列上溢。
8	BBER	串扰错误 USB总线上发生一个串扰事件。产生串扰事件的典型原因是端点发送了一个数据包，但是数据包长度超过了端点的最大包长。
7	USBER	USB总线错误 当在接收一个数据包的过程中，发生以下事件时，将置位USB总线错误标志位： – 接收包有一个错误的CRC域 – 在USB总线上检测到填充错误

– 当等待一个响应包时，超时

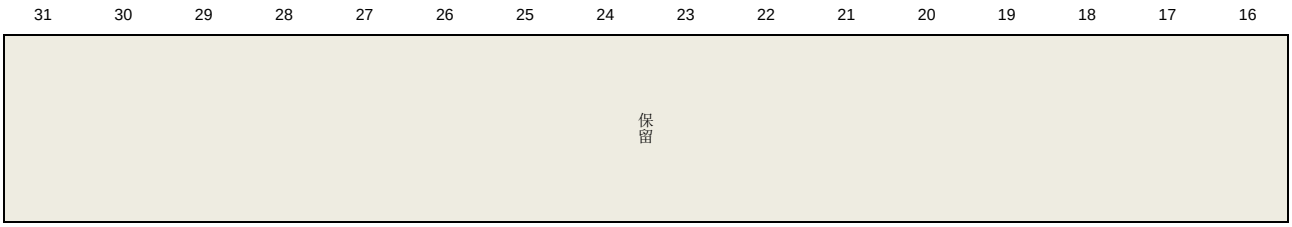
6	NYET	NYET 接收到一个NYET响应包（在高速模式下）
5	ACK	ACK 接收或者发送一个ACK响应包
4	NAK	NAK 接收到一个NAK响应包
3	STALL	STALL 接收到一个STALL响应包
2	DMAER	DMA 错误 当DMA尝试为当前通道获取或写如包数据时，产生一个错误
1	CH	通道中止 当DMA未被使能时： 通道被当前请求禁用。 当DMA使能时： 通道被DMA禁用，可能由于该通道的所有事务成功完成或者发生一个USB错误。
0	TF	发送完成 该通道所有的事务成功完成并且无错误发生。 对于IN通道，在USBHS_HCHxLEN寄存器的PCNT位减到0后，该标志位被置位。 对于OUT通道，当软件从RxFIFO中读取和取出一个TF状态条目时，该标志位被置位。

主机通道 x 中断使能寄存器 （USBHS_HCHxINTEN）（x = 0..15，其中 x = 通道号）

地址偏移：0x050C + （通道号 × 0x20）
复位值：0x0000 0000

该寄存器包含USBHS_HCHxINTF寄存器内中断标志位的中断使能位。如果该寄存器的某位被软件置位，USBHS_HCHxINTF寄存器内的相应位能够触发一个通道中断。该寄存器内的位可由软件置位和清除。

该寄存器只能按字（32位）访问



保留	DTERIE	REQOVRIE	BBERIE	USBERIE	NYETIE	ACKIE	NAKIE	STALLIE	DMAERIE	CHIE	TFIE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

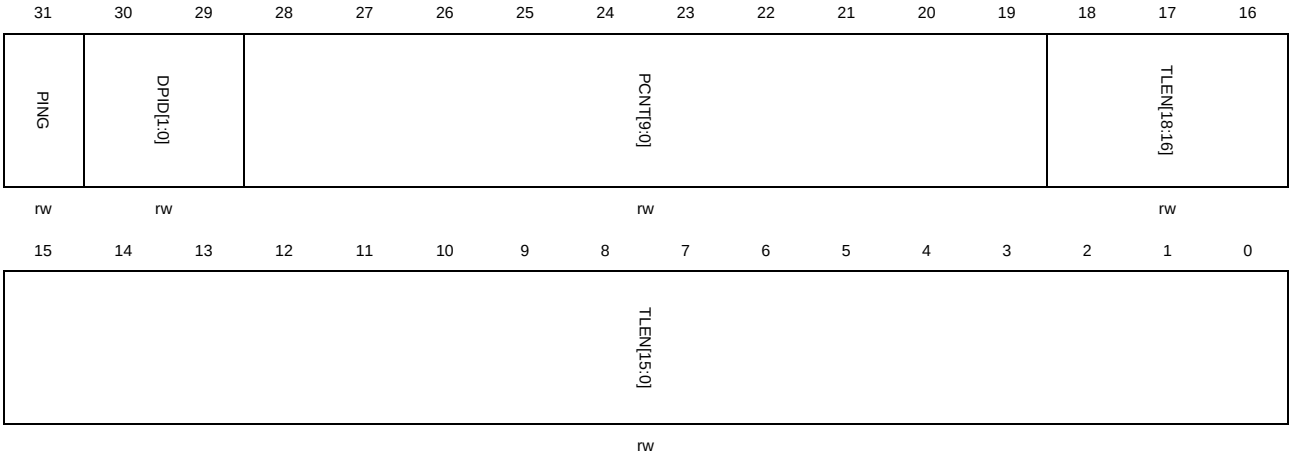
位/位域	名称	描述
31:11	保留	必须保持复位值。
10	DTERIE	数据切换错误中断使能 0：禁用数据切换错误中断 1：使能数据切换错误中断
9	REQOVRIE	请求队列上溢中断使能 0：禁用请求队列上溢中断 1：使能请求队列上溢中断
8	BBERIE	串扰错误中断使能 0：禁用串扰错误中断 1：使能串扰错误中断
7	USBERIE	USB总线错误中断使能 0：禁用USB总线错误中断 1：使能USB总线错误中断
6	NYETIE	NYET中断使能 0：禁用NYET中断 1：使能NYET中断
5	ACKIE	ACK中断使能 0：禁用ACK中断 1：使能ACK中断
4	NAKIE	NAK中断使能 0：禁用NAK中断 1：使能NAK中断
3	STALLIE	STALL中断使能 0：禁用STALL中断 1：使能STALL中断
2	DMAERIE	DMA 错误中断使能 0：禁用DMA错误中断 1：使能DMA错误中断
1	CHIE	通道中止中断使能 0：禁用通道中止中断 1：使能通道中止中断
0	TFIE	传输完成中断使能

0：禁用传输完成中断
1：使能传输完成中断

主机通道 x 长度寄存器（USBHS_HCHxLEN）（x = 0..15，其中 x = 通道号）

地址偏移：0x0510 + （通道号 × 0x20）
复位值：0x0000 0000

该寄存器只能按字（32位）访问



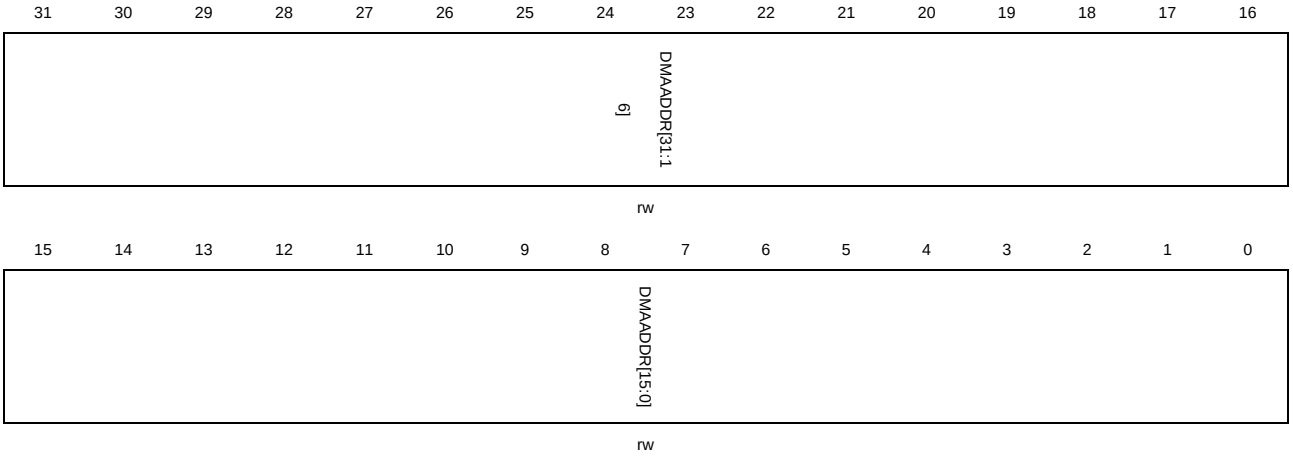
位/位域	名称	描述
31	PING	PING 令牌请求 对于OUT传输，如果软件置位该控制位，USBHS会执行PING协议。当OUT事务接收到一个NAK或NYET握手包时，USBHS会自动置位该控制位。不要为IN传输置位该控制位。
30:29	DPID[1:0]	数据PID 软件应该在传输起始之前写该段位域。对于OUT传输，该位域包含第一个传输包的数据PID。对于IN传输，该位域包含第一个接收包的数据PID，并且如果数据PID不匹配的话，将会触发DTER标志位。在传输开始之后，USBHS遵循USB协议自动改变和切换该位域。 00：DATA0 01：DATA2 10：DATA1 11：MDATA（非控制）/SETUP（控制）
28:19	PCNT[9:0]	包计数 在一个传输中希望发送（OUT）或接收（IN）的数据包个数。 软件应该在通道使能之前写该位域。在传输启动之后，该位域在USBHS正确传输每个数据包后，自动减少。
18:0	TLEN[18:0]	传输长度 一次传输的总数据字节数。 对于OUT传输，该位域为OUT传输中期望发送的所有数据包总数据字节数。软件应该在通道使能之前写该位域。当软件或DMA正确向通道的数据FIFO中写入一个包时，

该位域以包中字节大小进行减少。
对于IN传输，每次软件或DMA从RxFIFO中读取一个包后，该位域也以包中字节大小进行减少。

主机通道 x DMA 地址寄存器（USBHS_HCHxDMAADDR）（x = 0..15，其中 x = 通道号）

地址偏移：0x0514 + （通道号 × 0x20）
复位值：0x0000 0000

该寄存器只能按字（32位）访问



位/位域	名称	描述
31:0	DMAADDR[31:0]	DMA地址 该位域定义了端点DMA地址。DMA使用该地址来为该通道取出或写入包数据。

49.7.3. 设备控制和状态寄存器

设备配置寄存器（USBHS_DCFG）

地址偏移：0x0800
复位值：0x0000 0000

在上电、枚举或执行某些控制命令后，该寄存器配置内核为设备模式。在设备初始化后，不可以改变该寄存器值。

该寄存器采用字（32位）访问





位/位域	名称	描述
31:13	保留	必须保留为复位值。
12:11	EOPFT[1:0]	周期性帧尾时间 该域定义周期性帧时间的帧尾标志触发的时间点 00：80%的帧时间 01：85%的帧时间 10：90%的帧时间 11：95%的帧时间
10:4	DAR[6:0]	设备地址 该位定义USB设备地址，USBHS采用该位匹配接收的设备令牌地址域，在接收到来自主机的设置地址的命令后，软件设置该域
3	保留	必须保留为复位值。
2	NZLSOH	非零长度OUT状态阶段握手 在控制传输的OUT状态阶段，当USB设备接收到一个非零长度数据包时，该域控制控制USBHS是接收该包，还是用STALL握手信号拒绝该包。 0：把该包视为正常包，根据设备OUT端点控制寄存器的NAKS和STALL位，回复握手相应握手包 1：发送STALL握手，不保存接收到的OUT数据包
1:0	DS[1:0]	设备速度 该域控制设备连入主机后的设备速度 00：高速 01：全速 其他：保留

设备控制寄存器（USBHS_DCTL）

地址偏移：0x0804

复位值：0x0000 0000

该寄存器采用字（32位）访问



保留	POIF	CGONAK	SGONAK	CGINAK	SGINAK	DTEST[2:0]	GONS	GINS	SD	RWKUP
	rw	w	w	w	w	rw	r	r	rw	rw

位/位域	名称	描述
31:19	保留	必须保留为复位值。
18	L1RJCT	禁止深度睡眠 当这一位被设置，内核以NYET响应LPM事务，同时BESL大于BSELTH
17:12	保留	必须保留为复位值。
11	POIF	上电初始化完成 软件通过设置该位，通知USBHS寄存器在从掉电模式下唤醒，然后完成初始化。
10	CGONAK	清零全局OUT NAK 软件设置该位从而清零该寄存器的GONS位
9	SGONAK	设置全局OUT NAK 软件设置该位从而实现该寄存器的位GONS置位。 当GONS位为零，设置该位会引起USBHS_GINTF寄存器的GONAK标志触发，软件应该在再写该位前清除GONAK标志。
8	CGINAK	清零全局IN NAK 软件设置该位从而清零该寄存器的GINS位
7	SGINAK	设置全局IN NAK 软件设置该位从而实现该寄存器的位GINS置位 当GINS位为零，设置该位会引起USBHS_GINTF寄存器的GINAK标志触发，软件应该在再写该位前清除GINAK标志。
6:4	DTEST[2:0]	设备测试控制 软件向该字段写入一个非零值以使端口进入测试模式，相应的模式发送到端口上。使用测试模式时，还应设置USBHS_GUSBCS寄存器中的HS_CUR_FE位。 0000：测试模式禁止 0001：Test_J模式 0010：Test_K模式 0011：Test_SE0_NAK模式 0100：Test_Packet模式 0101：强制测试使能 其他：保留
3	GONS	全局OUT NAK状态 0：USBHS回复OUT事务的握手信号以及是否保存OUT数据包由Rx FIFO状态、端点的NAKS、STALL位确定。 1：USBHS回复OUT事务NAK握手信号，不保存接收的OUT数据包。

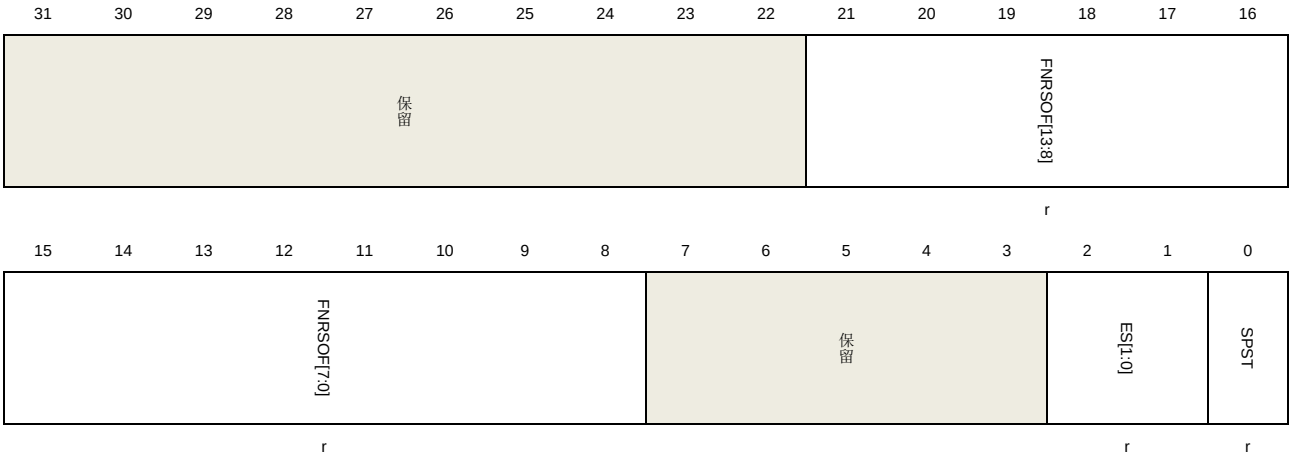
2	GINS	全局IN NAK状态 0：USBHS回复IN事务的握手信号由Tx FIFO状态、端点的NAKS、STALL位确定。 1：USBHS通常回复IN事务NAK握手信号
1	SD	软断开 软件可实现USB总线上的软断开，在置1该位后，如果当前是高速模式，USBHS先退回到全速模式，然后在关掉DP线上的上拉电阻，从而引起主机检测设备的断开。 0：没有软断开生成 1：生成软断开
0	RWKUP	远程唤醒 在挂起状态，软件可通过该位来生成一个远程唤醒信号来通知主机恢复USB总线 0：没有远程唤醒信号生成 1：生成远程唤醒信号 当核心已启用LPM且处于睡眠状态时，如果设置了此位，则核心将继续驱动它，并在50us后自动清除它（TL1DevDrvResume）。当从LPM事务接收的bRemoteWake值为零时，应用程序无法设置此位。

设备状态寄存器（USBHS_DSTAT）

地址偏移：0x0808
复位值：0x0000 0000

该寄存器包含设备模式下的 USBHS 的状态和信息。

该寄存器采用字（32位）访问



位/位域	名称	描述
31:22	保留	必须保留为复位值。
21:8	FNRSOF[13:0]	所接收的SOF帧编号 USBHS会在接收到一个SOF令牌后更新该域。
7:3	保留	必须保留为复位值。
2:1	ES[1:0]	枚举速度

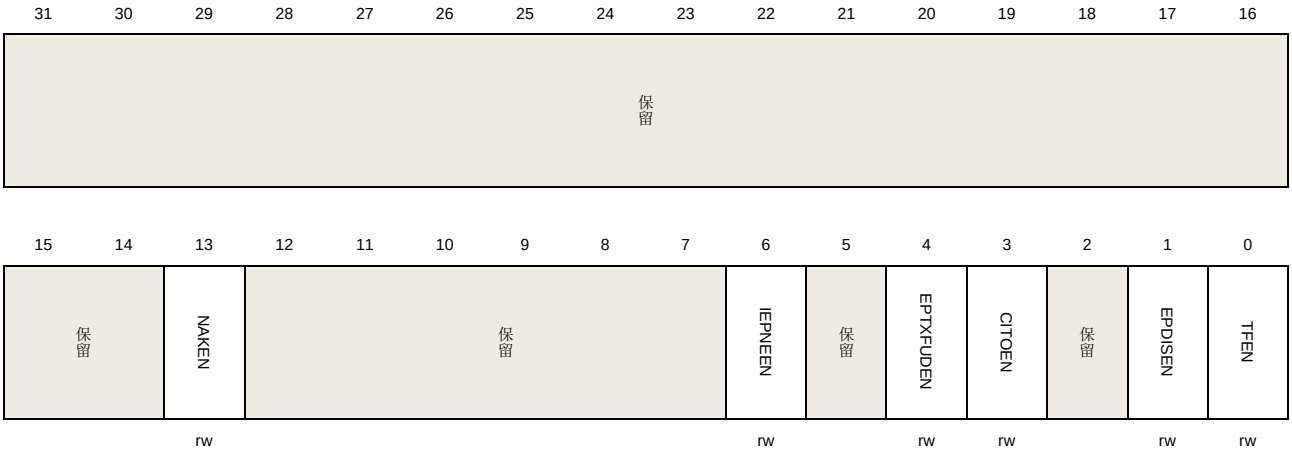
		该域指示所枚举的设备速度，在寄存器USBHS_GINTF的ENUMF标志触发后，软件可以读取该域。 00：高速 01：全速 其他：保留
0	SPST	挂起状态 该位指示设备是否处于挂起状态。 0：设备不在挂起状态 1：设备在挂起状态

设备 IN 端点通用中断使能寄存器（USBHS_DIEPINTEN）

地址偏移：0x810
复位值：0x0000 0000

该寄存器包含寄存器 USBHS_DIEPxINTF 中的标志的中断使能位，如果软件置 1 某位，其在寄存器 USBHS_DIEPxINTF 中对应的位可以触发一个寄存器 USBHS_DAEPINT 端点中断。该位可以通过软件置位和清零。

该寄存器采用字（32位）访问



位/位域	名称	描述
31:14	保留	必须保留为复位值。
13	NAKEN	发送NAK握手中断使能位 0：除能中断 1：使能中断
12:7	保留	必须保留为复位值。
6	IEPNEEN	IN端点NAK有效中断使能位 0：除能中断 1：使能中断
5	保留	必须保留为复位值。

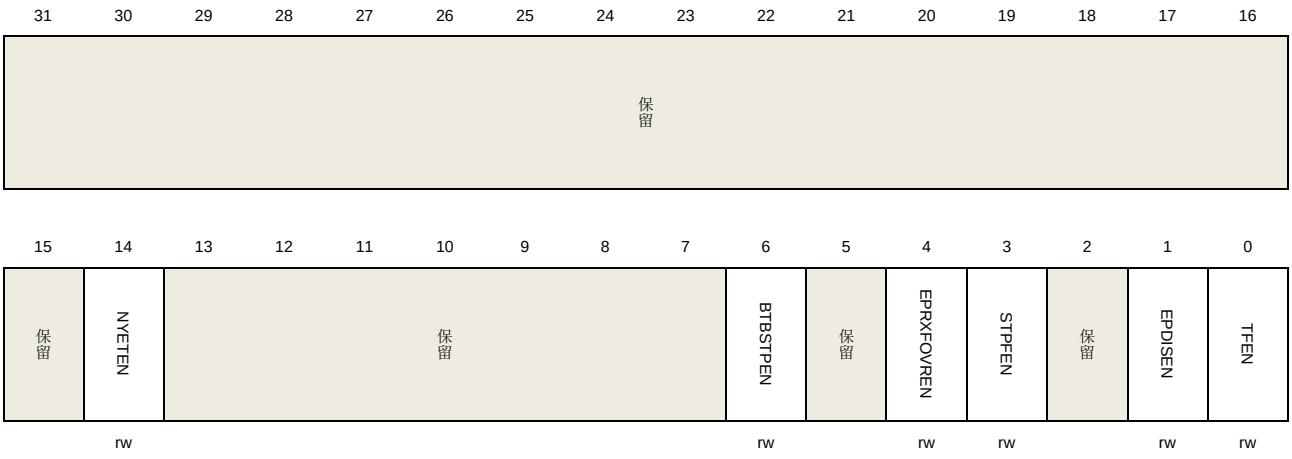
4	EPTXFUDEN	端点Tx FIFO下溢中断使能位 0：除能中断 1：使能中断
3	CITOEN	控制IN事务超时中断使能位 0：除能中断 1：使能中断
2	保留	必须保留为复位值。
1	EPDISEN	端点除能中断使能位 0：除能中断 1：使能中断
0	TFEN	传输完成中断使能位 0：除能中断 1：使能中断

设备 OUT 端点通用中断使能寄存器（USBHS_DOEPINTEN）

地址偏移：0x0814
复位值：0x0000 0000

该寄存器包含寄存器 USBHS_DOEPxINTF 中的标志的中断使能位，如果软件置 1 某位，其在寄存器 USBHS_DOEPxINTF 中对应的位可以触发一个寄存器 USBHS_DAEPINT 端点中断。该位可以通过软件置位和清零。

该寄存器采用字（32位）访问



位/位域	名称	描述
31:15	保留	必须保留为复位值。
14	NYETEN	发送NYET握手中断使能位 0：除能中断 1：使能中断
13:7	保留	必须保留为复位值。

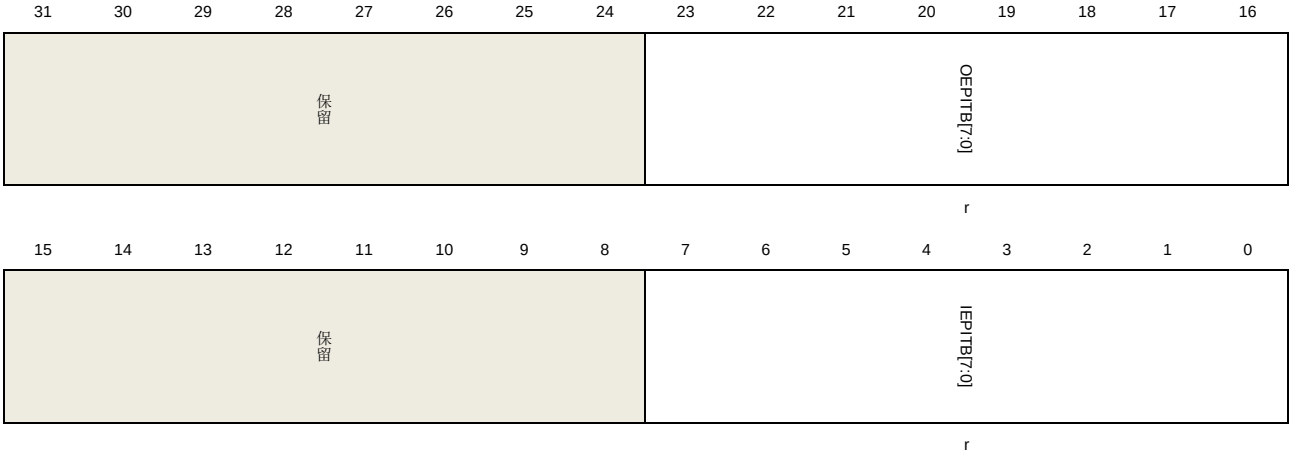
6	BTBSTPEN	连续SETUP包中断使能位（仅适用于控制OUT端点） 0：除能中断 1：使能中断
5	保留	必须保留为复位值。
4	EPRXFOVREN	端点Rx FIFO上溢中断使能位 0：除能中断 1：使能中断
3	STPFEN	SETUP阶段完成中断使能位（仅适用于控制OUT端点） 0：除能中断 1：使能中断
2	保留	必须保留为复位值。
1	EPDISEN	端点除能中断使能位 0：除能中断 1：使能中断
0	TFEN	传输完成中断使能位 0：除能中断 1：使能中断

设备端点中断寄存器（USBHS_DAEPINT）

地址偏移：0x0818
复位值：0x0000 0000

当一个端点的中断被触发，USBHS 置 1 该寄存器的相应位，软件可通过该寄存器知道在本次中断中的端点号。

该寄存器采用字（32位）访问



位/位域	名称	描述
31:24	保留	必须保留为复位值。
23:16	OEPITB[7:0]	设备OUT端点中断位

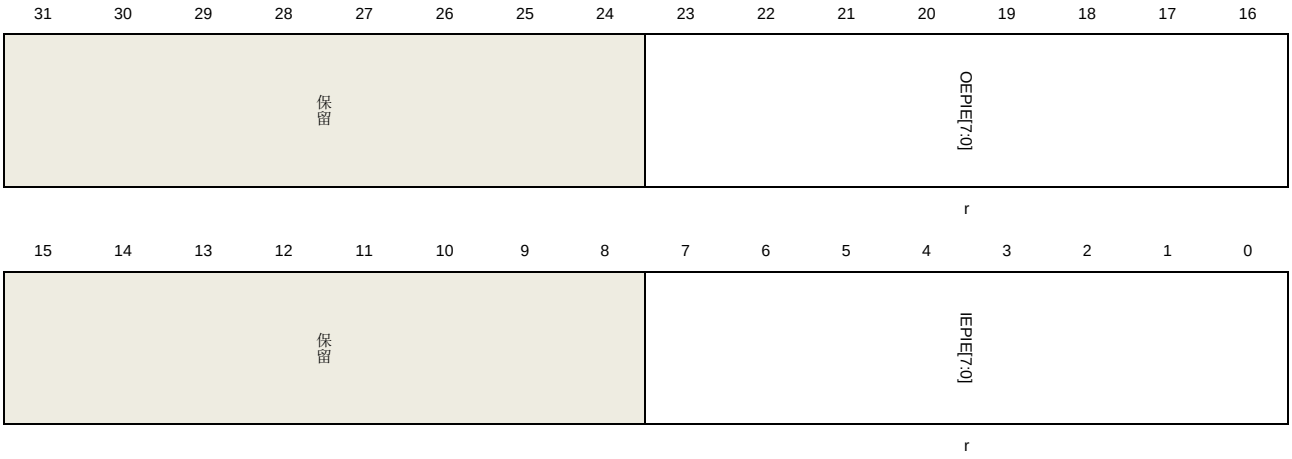
		每个位代表一个OUT端点：Bit16代表OUT端点0，Bit23代表OUT端点7
15:8	保留	必须保留为复位值。
7:0	IEPITB[7:0]	设备IN端点中断位 每个位代表一个IN端点：Bit0代表IN端点0，Bit7代表IN端点7

设备端点中断使能寄存器（USBHS_DAEPINTEN）

地址偏移：0x081C
复位值：0x0000 0000

该寄存器可通过软件使能或除能端点的中断，只有当端点在该寄存器中相应位被置 1 才能触发寄存器 USBHS_GINTF 的端点中断标志 OEPIF 或 IEPIF。

该寄存器采用字（32位）访问

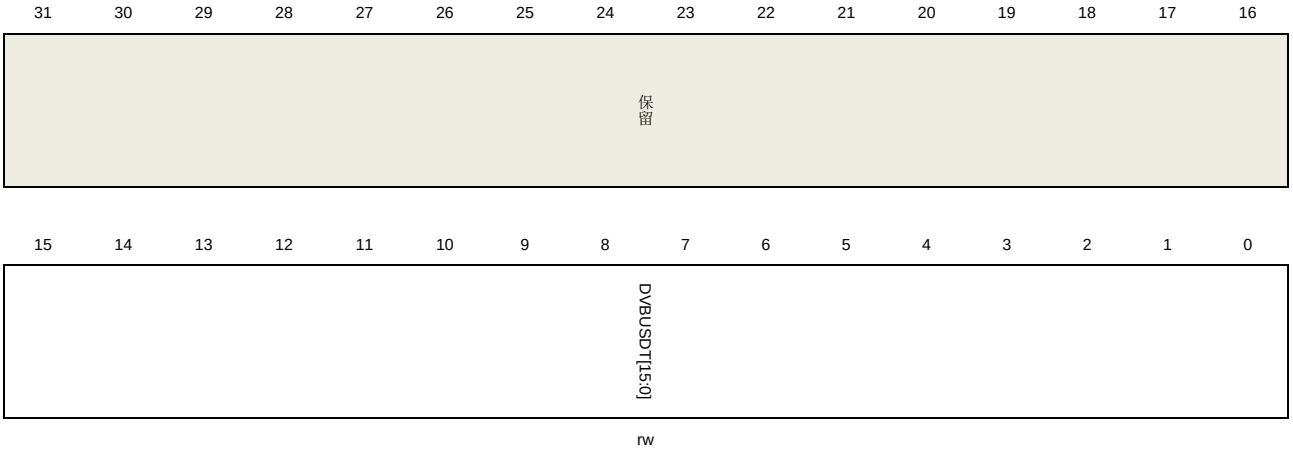


位/位域	名称	描述
31:24	保留	必须保留为复位值。
23:16	OEPIE[7:0]	OUT端点中断使能位 0：除能OUT端点n中断 1：使能OUT端点n中断 每个位代表一个OUT端点：Bit16对应OUT端点0，Bit23对应OUT端点7
15:8	保留	必须保留为复位值。
7:0	IEPIE[7:0]	IN端点中断使能位 0：除能IN端点n中断 1：使能IN端点n中断 每个位代表一个IN端点：Bit0对应IN端点0，Bit7对应IN端点7

设备 VBUS 放电时间寄存器（USBHS_DVBUSDT）

地址偏移：0x0828
复位值：0x0000 17D7

该寄存器采用字（32位）访问



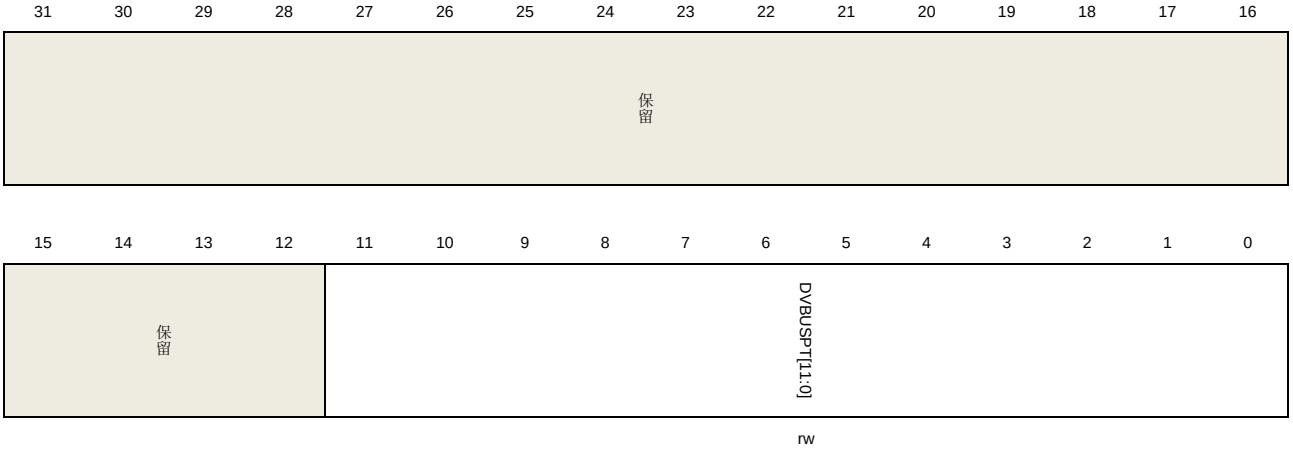
位/位域	名称	描述
31:16	保留	必须保留为复位值。
15:0	DVBUSDT[15:0]	设备V_{BUS}放电时间 在SRP协议中，在V_{BUS}脉冲产生后，有一个放电过程，该域定义了V_{BUS}的放电时间，真正的放电时间是1024*DVBUSDT[15:0] *$T_{USBCLOCK}$，$T_{USBCLOCK}$是USB时钟周期时间。

设备 VBUS 脉冲时间寄存器（USBHS_DVBUSPT）

地址偏移：0x082C

复位值：0x0000 05B8

该寄存器采用字（32位）访问



位/位域	名称	描述
31:12	保留	必须保留为复位值。
11:0	DVBUSPT[11:0]	设备V_{BUS}脉冲时间 该域定义V_{BUS}的脉冲时间，真正的充电时间是1024*DVBUSPT[1:0] *$T_{USBCLOCK}$，$T_{USBCLOCK}$是USB时钟周期时间

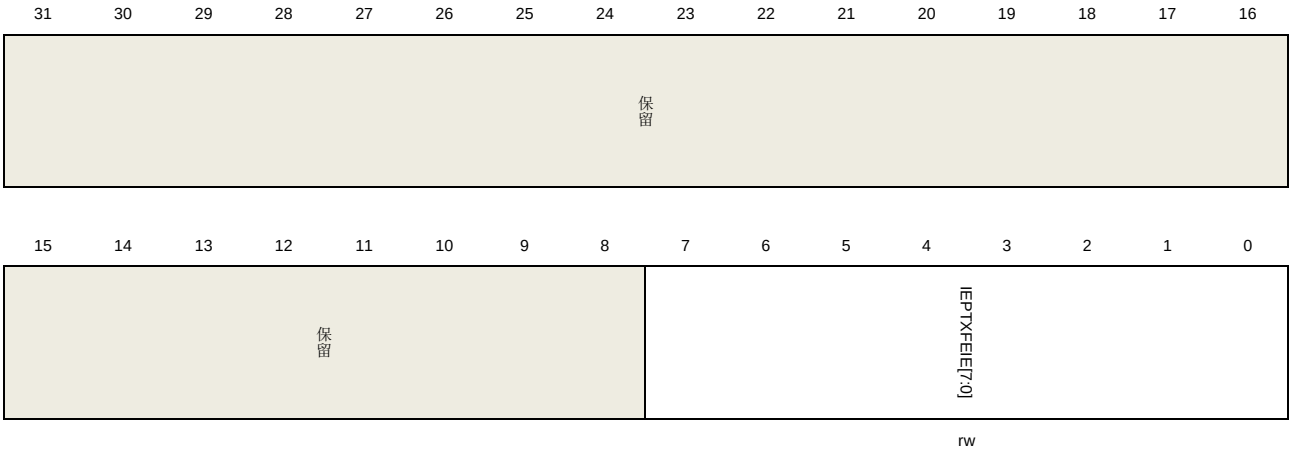
设备 IN 端点 FIFO 空中断使能寄存器（USBHS_DIEPFEINTEN）

地址偏移：0x0834

复位值：0x0000 0000

该寄存器包含 IN 端点 Tx FIFO 空中断的使能位

寄存器采用字（32位）访问



位/位域	名称	描述
31:8	保留	必须保留为复位值。
7:0	IEPTXFEIE[7:0]	IN端点Tx FIFO空中断的使能位 该域控制着USBHS_DIEPxINTF寄存器的TXFE位能否生成一个寄存器USBHS_DAEPINT的端点中断位 Bit0对应IN端点0，Bit7对应IN端点7 0：除能FIFO空中断 1：使能FIFO空中断

设备端点 1 中断寄存器（USBHS_DEP1INT）

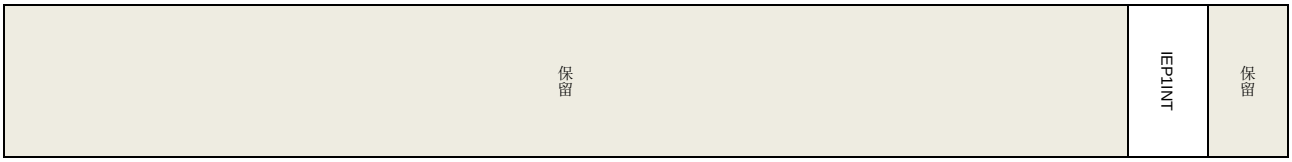
地址偏移：0x0838

复位值：0x0000 0000

当端点 1 OUT 或 IN 被触发，USBHS 置位该寄存器的相应位，软件可通过读取该位知道端点 1 的 IN 或 OUT 被触发。

该寄存器采用字（32位）访问





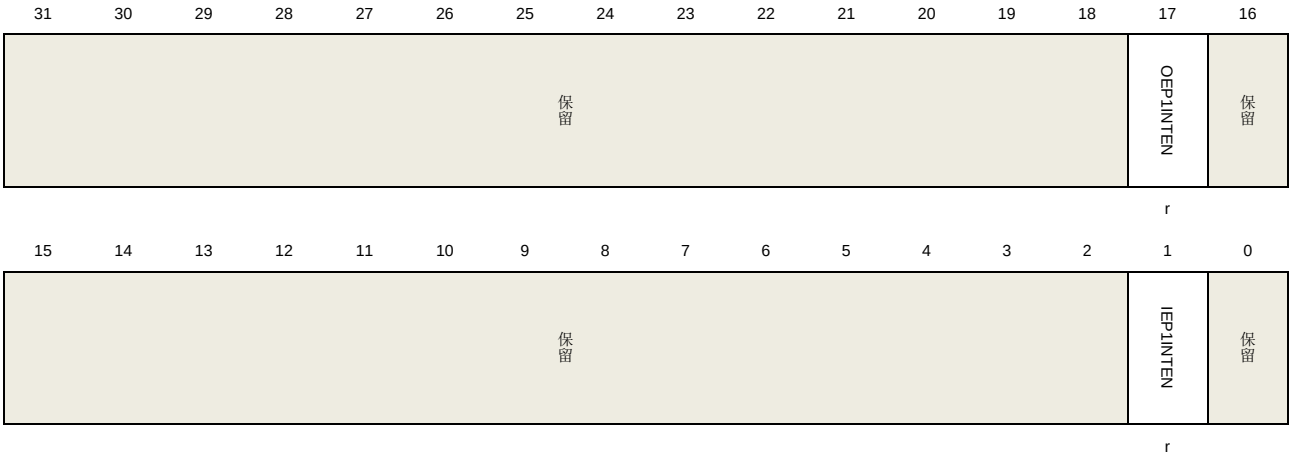
位/位域	名称	描述
31:18	保留	必须保留为复位值。
17	OEP1INT	OUT端点1中断
16:2	保留	必须保留为复位值。
1	IEP1INT	IN端点1中断
0	保留	必须保留为复位值。

设备端点 1 中断使能寄存器（USBHS_DEP1INTEN）

地址偏移：0x083C
复位值：0x0000 0000

该寄存器可以用软件使能或除能端点1的中断，只有该寄存器中的相应位被置位才可以引起端点1 IN或者OUT的中断。

该寄存器采用字（32位）访问



位/位域	名称	描述
31:18	保留	必须保留为复位值。
17	OEP1INTEN	OUT端点1中断使能
16:2	保留	必须保留为复位值。
1	IEP1INTEN	IN端点1中断使能
0	保留	必须保留为复位值。

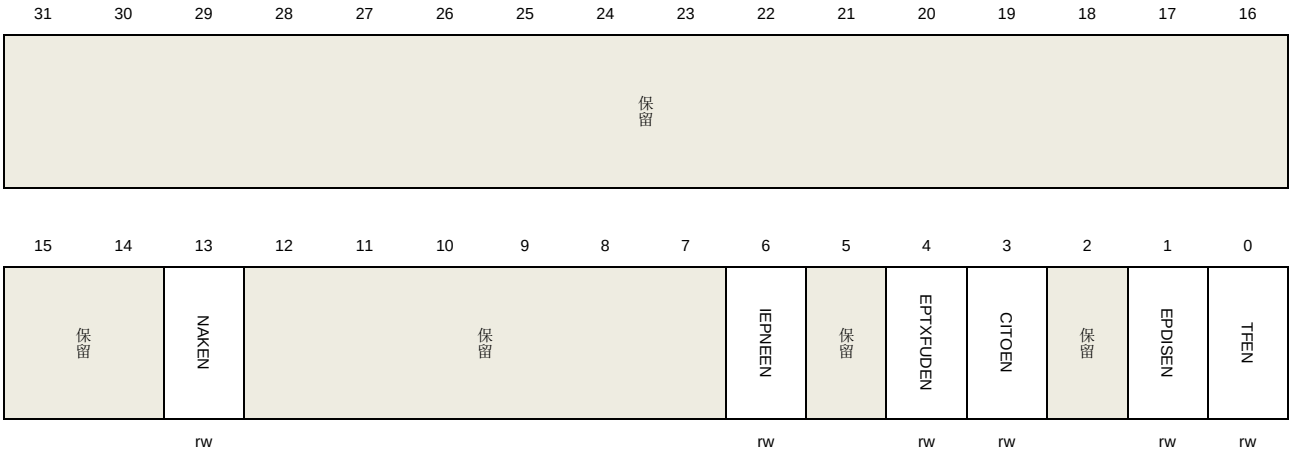
设备 IN 端点 1 中断使能寄存器（USBHS_DIEP1INTEN）

地址偏移：0x844

复位值：0x0000 0000

该寄存器包含对应寄存器 USBHS_DIEP1INTF 标志的中断使能位，如果软件置位该寄存器的某一位，其在寄存器 USBHS_DIEP1INTF 中对应的位会触发一个寄存器 USBHS_DEP1INT 中的端点中断，该寄存器的位可以通过软件置位或清零。

该寄存器采用字（32位）访问



位/位域	名称	描述
31:14	保留	必须保留为复位值。
13	NAKEN	发送NAK握手中断使能位 0：除能中断 1：使能中断
12:7	保留	必须保留为复位值。
6	IEPNEEN	IN端点NAK有效中断使能位 0：除能中断 1：使能中断
5	保留	必须保留为复位值。
4	EPTXFUD	端点Tx FIFO下溢中断使能位 0：除能中断 1：使能中断
3	CITOEN	控制传输IN超时中断使能位 0：除能中断 1：使能中断
2	保留	必须保留为复位值。
1	EPDISEN	端点除能中断使能位 0：除能中断

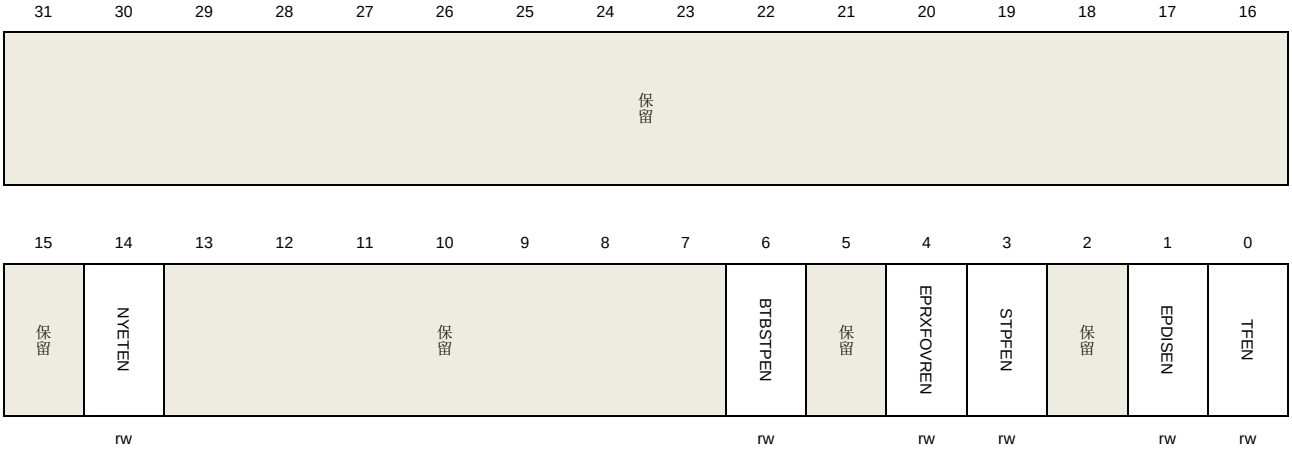
		1：使能中断
0	TFEN	传输完成中断使能位 0：除能中断 1：使能中断

设备 OUT 端点 1 中断使能寄存器（USBHS_DOEP1INTEN）

地址偏移：0x0884
复位值：0x0000 0000

该寄存器包含对应寄存器 USBHS_DOEP1INTF 标志的中断使能位，如果软件置位该寄存器的某一位，其在寄存器 USBHS_DOEP1INTF 中对应的位会触发一个寄存器 USBHS_DEP1INT 中的端点中断，该寄存器的位可以通过软件置位或清零。

该寄存器采用字（32位）访问



位/位域	名称	描述
31:15	保留	必须保留为复位值。
14	NYETEN	发送NYET握手中断使能位 0：除能中断 1：使能中断
13:7	保留	必须保留为复位值。
6	BTBSTPEN	连续SETUP包中断使能位（仅适用于控制OUT端点） 0：除能中断 1：使能中断
5	保留	必须保留为复位值。
4	EPRXFOVREN	端点Rx FIFO上溢中断使能位 0：除能中断 1：使能中断
3	STPFEN	SETUP阶段完成中断使能位（仅适用于控制OUT端点） 0：除能中断

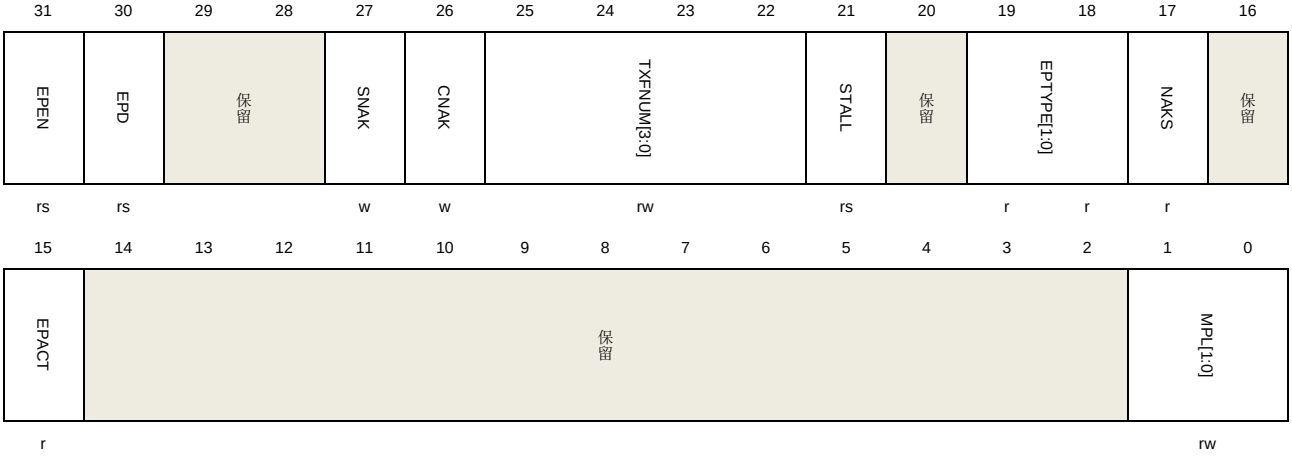
		1：使能中断
2	保留	必须保留为复位值
1	EPDISEN	端点除能中断使能位 0：除能中断 1：使能中断
0	TFEN	传输完成中断使能位 0：除能中断 1：使能中断

设备 IN 端点 0 控制寄存器（USBHS_DIEP0CTL）

地址偏移：0x0900

复位值：0x0000 8000

该寄存器采用字（32位）访问



位/位域	名称	描述
31	EPEN	端点使能 软件置位、USBHS清零 0：端点除能 1：端点使能 软件应该按照操作指南使能或除能端点
30	EPD	端点除能 软件可通过置位该位从而除能端点，软件应该按照操作指南使能或除能端点。
29:28	保留	必须保留为复位值
27	SNAK	置位NAK 软件置位该位来设置该寄存器的NAKS位
26	CNAK	清零NAK 软件置位该位来清零该寄存器的NAKS位

25:22	TXFNUM[3:0]	Tx FIFO编号 定义IN端点0的Tx FIFO编号
21	STALL	STALL握手 当接收IN令牌时，软件可以通过置1该位发送STALL握手包，对于相应的OUT端点0，在接收SETUP令牌后，USBHS清除此位。该位比该寄存器的NAKS位和寄存器USBHS_DCTL的GINS位优先级要高，如果STALL和NAKS位都被置位，STALL位生效。
20	保留	必须保留为复位值。
19:18	EPTYPE[1:0]	端点类型 该域固定为'00',控制端点。
17	NAKS	NAK状态 当该寄存器的STALL位和寄存器USBHS_DCTL的位GINS被清零，该位控制USBHS的NAK状态。 0：根据端点Tx FIFO的状态，USBHS发送数据或握手包 1：USBHS总为IN令牌发送NAK握手包 该位是只读位，可以通过该寄存器的位CNAK和位SNAK控制该位
16	保留	必须保留为复位值。
15	EPACT	端点激活 对于端点0来说，该域固定为'1'
14:2	保留	必须保留为复位值。
1:0	MPL[1:0]	最大包长 域定义了控制数据包的最大包长，如USB 2.0协议所描述，对控制传输而言，有四种包长度： 00：64字节 01：32字节 10：16字节 11：8字节

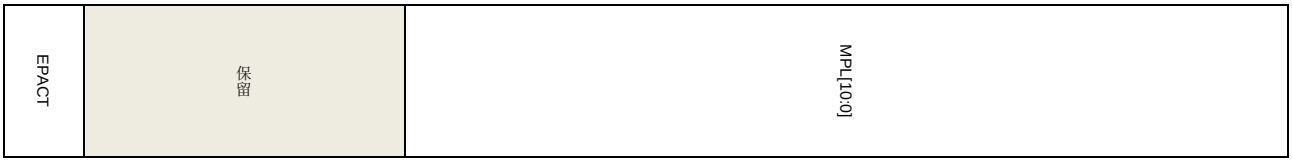
设备 IN 端点 x 控制寄存器（USBHS_DIEPxCTL）（x = 1..7， x 是端点编号）

地址偏移：0x0900 + (x * 0x20)

复位值：0x0000 0000

该寄存器采用字（32 位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPEN	EPD	SODDFRM/SD1PID	SD0PID/SEVENFRM	SNAK	CNAK	TXFNUM[3:0]				STALL	保留	EPTYPE[1:0]		NAKS	EOFRM/DPID
rs	rs	w	w	w	w	rw				rw/rs		rw		r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0



位/位域	名称	描述
31	EPEN	端点使能 软件置位，USBHS清零 0：端点除能 1：端点使能 软件应该按照操作指南使能或除能端点
30	EPD	端点除能 软件可通过置位该位从而除能端点，软件应该按照操作指南使能或除能端点。
29	SODDFRM	设置奇数帧（适用于同步IN端点） 软件通过置1该位置1该寄存器的EOFRM位
	SD1PID	设置DATA1 PID(适用于中断和大容量IN端点) 软件可通过置1该位置1该寄存器的DPID位
28	SEVENFRM	设置偶数帧(适用于同步IN端点) 软件通过置1该位清零该寄存器的EOFRM位
	SD0PID	设置DATA0 PID(适用于中断和大容量IN端点) 软件可通过置1该位清零该寄存器的DPID位
27	SNAK	设置NAK 软件置1该位置1该寄存器的NAKS位
26	CNAK	清零NAK 软件置1该位清零该寄存器的NAKS位
25:22	TXFNUM[3:0]	Tx FIFO编号 该位定义了IN端点的Tx FIFO编号
21	STALL	STALL握手 当接收IN令牌时，软件可以通过置1该位发送STALL握手包。该位比该寄存器的NAKS位和寄存器USBHS_DCTL的GINS位优先级要高，如果STALL和NAKS位都被置位，STALL位生效。 对于控制IN端点： 当对应的OUT端点接收到SETUP令牌时，只有USBHS可以清零此位，软件不可清除此位。 对于中断或大容量IN端点： 只有软件可以清零此位。
20	保留	必须保留为复位值。
19:18	EPTYPE[1:0]	端点类型

		该域定义端点的传输类型： 00：控制 01：同步 10：大容量 11：中断
17	NAKS	NAK状态 当该寄存器的STALL位和寄存器USBHS_DCTL的位GINS被清零，该位控制USBHS的NAK状态： 0：根据端点Tx FIFO的状态，USBHS发送数据或握手包 1：USBHS总为IN令牌发送NAK握手包 该位是只读位，可以通过该寄存器的位CNAK和位SNAK控制该位
16	EOFRM	奇偶帧（适用于同步IN端点） 对于同步传输，软件通过使用该位控制USBHS只在奇数帧或偶数帧为IN事务发送数据包，如果当前帧号的奇偶性不匹配该位，USBHS回复一个零长度的包： 0：只在偶数帧发送数据 1：只在奇数帧发送数据
	DPID	端点数据PID（适用于中断或大容量IN端点） 在端点或大容量传输中，有数据PID翻转机制，在传输开始之前，软件通过设定SD0PID来设置此位，按照USB协议中描述的数据PID翻转机制，USBHS在传输过程中保持该位。 0：数据包的PID是DATA0 1：数据包的PID是DATA1
15	EPACT	端点激活 该位控制端点是否激活，当端点没有激活，忽略任何令牌，不做任何回复。
14:11	保留	必须保留为复位值。
10:0	MPL[10:0]	该域定义最大包长

设备 OUT 端点 0 控制寄存器（USBHS_DOEP0CTL）

地址偏移：0x0B00
复位值：0x0000 8000

该寄存器采用字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPEN	EPD	保留		SNAK	CNAK	保留				STALL	SNOOP	EPTYPE[1:0]		NAKS	保留
rs	r			w	w					rs	rw	r		r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0



位/位域	名称	描述
31	EPEN	端点使能 软件置位，USBHS清零 0：端点除能 1：端点使能 软件应该按照操作指南使能或除能端点。
30	EPD	端点除能 对于OUT端点0，该位固定为0
29:28	保留	必须保留为复位值。
27	SNAK	设置NAK 软件置1该位置1该寄存器的NAKS位
26	CNAK	清零NAK 软件置1该位清零该寄存器的NAKS位
25:22	保留	必须保留为复位值。
21	STALL	STALL握手 在OUT事务中，软件可以通过置1该位发送STALL握手包，对于OUT端点0，在接收SETUP令牌后，USBHS清除此位。该位比该寄存器的NAKS位和寄存器USBHS_DCTL的GINS位优先级要高，即如果STALL和NAKS位都被置位，STALL位生效。
20	SNOOP	调查模式 该位控制OUT端点的调查模式，在调查模式中，USBHS不再检查接收数据包的CRC值 0：调查模式除能 1：调查模式使能
19:18	EPTYPE[1:0]	端点类型 对于控制端点，该位固定为”00“
17	NAKS	NAK状态 当该寄存器的STALL位和寄存器USBHS_DCTL的位GINS被清零，该位控制USBHS的NAK状态： 0：根据端点Rx FIFO的状态，USBHS发送数据或握手包 1：USBHS为OUT事务发NAK握手包 该位是只读位，通过该寄存器的CNAK和SNAK位控制该位
16	保留	必须保留为复位值。

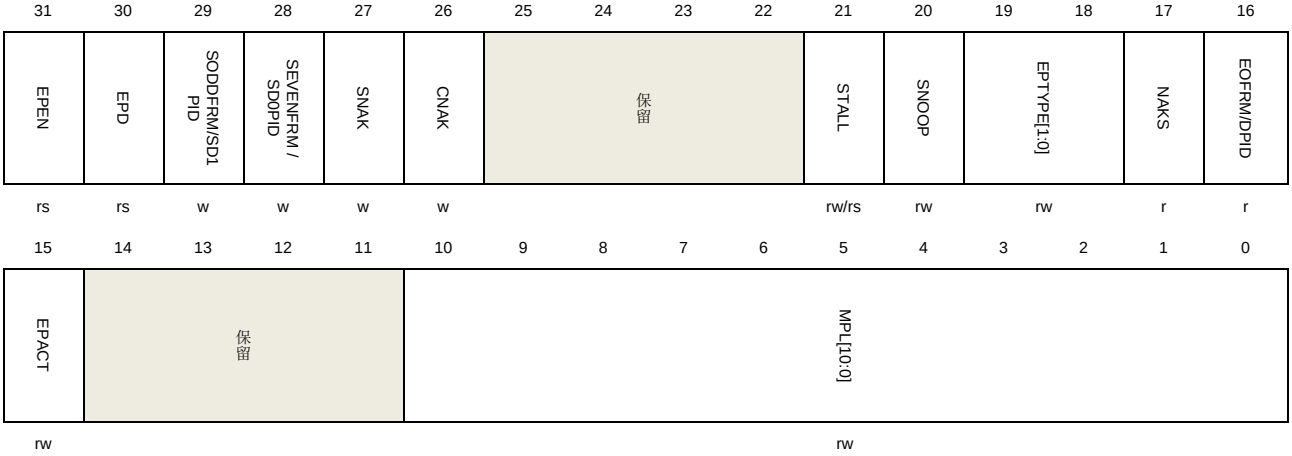
15	EPACT	端点激活 对于端点0，该域固定为1
14:2	保留	必须保留为复位值。
1:0	MPL[1:0]	最大包长 该位是只读位，其数值来自于寄存器USBHS_DIEP0CTL的位MPL： 00：64字节 01：32字节 10：16字节 11：8字节

设备 OUT 端点 x 控制寄存器（USBHS_DOEPxCTL）（x= 1..7, x 是端点编号）

地址偏移：0x0B00 + (x * 0x20)
复位值：0x0000 0000

软件用该寄存器控制 OUT 端点 0 以外的每个逻辑 OUT 端点

该寄存器采用字（32位）访问



位/位域	名称	描述
31	EPEN	端点使能 软件置位，USBHS清零 0：端点除能 1：端点使能 软件应该按照操作指南使能或除能端点。
30	EPD	端点除能 软件通过置1该位除能端点，软件应该按照操作指南使能或除能端点。
29	SODDFRM	设置奇数帧（适用于同步OUT端点） 该位只针对同步OUT端点有效 软件置1该位来置位该寄存器的EOFRM位
	SD1PID	设置DATA1 PID(适用于中断和大容量OUT端点)

		软件置1该位来置位该寄存器的DPID位
28	SEVENFRM	设置偶数帧（适用于同步OUT端点） 软件置1该位来清零该寄存器的EOFRM位
	SD0PID	设置DATA0 PID(适用于中断和大容量OUT端点) 软件置1该位来清零该寄存器的DPID位
27	SNAK	设置NAK 软件置1该位从而置1该寄存器的NAKS位
26	CNAK	清零NAK 软件置1该位从而清零该寄存器的NAKS位
25:22	保留	必须保留为复位值。
21	STALL	STALL握手 在OUT事务中，软件可以通过置1该位发送STALL握手包。该位比该寄存器的NAKS位和寄存器USBHS_DCTL的GINS位优先级要高，如果STALL和NAKS位都被置位，STALL位生效。 对于控制OUT端点： 当OUT端点接收SETUP令牌时，只有USBHS可以清零该位，软件不可清零此位。 对于中断或大容量OUT端点 只有软件可以清零该位
20	SNOOP	调查模式 该位控制OUT端点的调查模式，在调查模式中，USBHS不再检查接收数据包的CRC值 0：调查模式除能 1：调查模式使能
19:18	EPTYPE[1:0]	端点类型 该域定义端点的传输类型 00：控制 01：同步 10：大容量 11：中断
17	NAKS	NAK状态 当该寄存器的STALL位和寄存器USBHS_DCTL的位GONS被清零，该位控制USBHS的NAK状态： 0：根据端点的Rx FIFO的状态，发送握手包 1：USBHS为OUT事务发送NAK握手 该位是只读位，通过该寄存器的CNAK和SNAK位控制该位
16	EOFRM	奇偶帧（适用于同步OUT端点） 对于同步传输，软件通过使用该位控制USBHS只在奇数帧或偶数帧发送数据包给OUT事务，如果当前帧号的奇偶性不匹配该位，USBHS不保存数据包 0：只在偶数帧发送数据

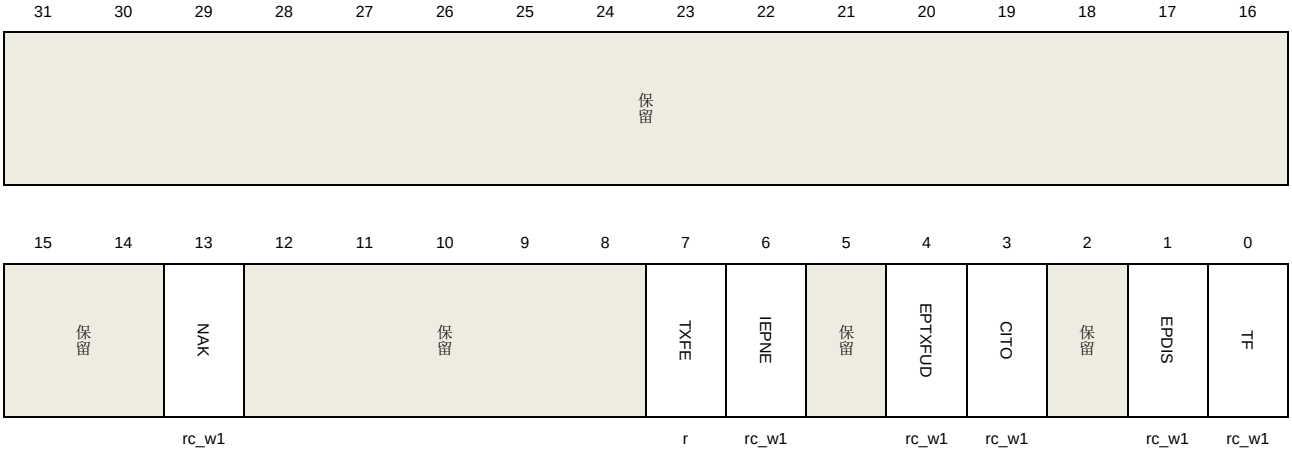
		1：只在奇数帧发送数据
	DPID	端点数据PID（适用于中断或大容量端点） 在端点或大容量传输中，有数据PID翻转机制，在传输开始之前，软件通过设定SD0PID来设置此位，按照USB协议中描述的数据PID翻转机制，USBHS在传输过程中保持该位。 0：数据包PID是DATA0 1：数据包PID是DATA1
15	EPACT	端点激活 位控制端点是否激活，当端点没有激活，忽略任何令牌，不做任何回复
14:11	保留	必须保留为复位值。
10:0	MPL[10:0]	该位定义最大包长

设备 IN 端点 x 中断标志寄存器（USBHS_DIEPxINTF）（x = 0..7, x 是端点编号）

地址偏移：0x0908 + (x * 0x20)
复位值：0x0000 0080

该寄存器包含 IN 端点的状态和事件，当获得一个 IN 端点的中断时，应该读取该端点的中断标志寄存器，从而获知中断源。该寄存器的标志位通常硬件置位，除了 TXFE 位，各位写 1 清零。

该寄存器采用字（32位）访问



位/位域	名称	描述
31:14	保留	必须保留为复位值。
13	NAK	发送NAK握手 USBHS在发出一个NAK握手包后置该位，因为寄存器USBHS_DIEPxCTL的NAKS位被置位，或端点的Tx FIFO里没有包数据。
12:8	保留	必须保留为复位值。
7	TXFE	发送FIFO空 端点的Tx FIFO达到寄存器USBHS_GAHBCS的位TXFTH定义的空阈值。

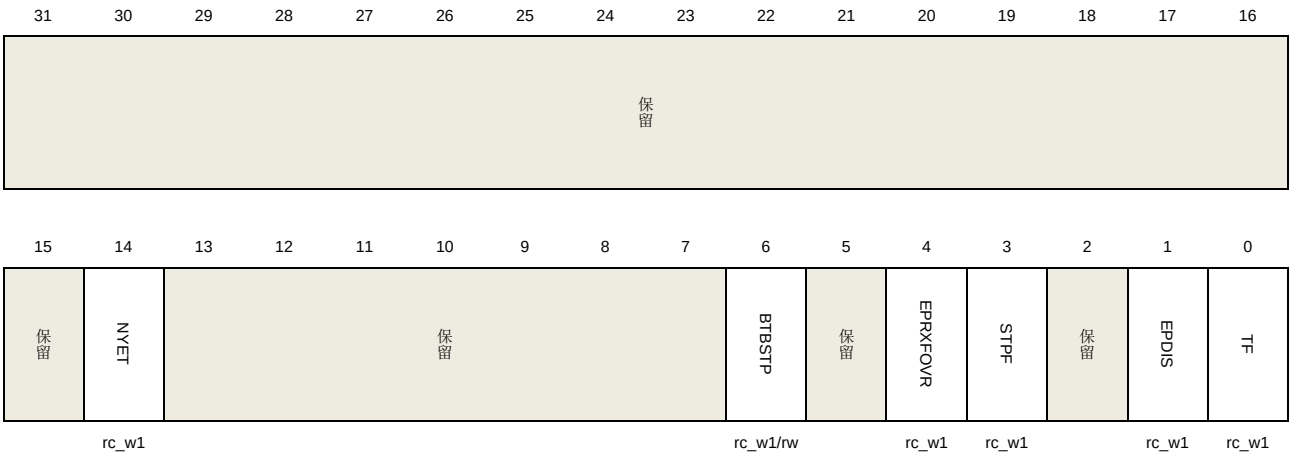
6	IEPNE	IN端点NAK有效 寄存器USBHS_DIEPxCTL的位SNAK的设置生效，该位可以通过写1清零或设置CNAK位
5	保留	必须保留为复位值。
4	EPTXFUD	端点Tx FIFO下溢 如果当IN令牌被接收后，Tx FIFO没有包数据，该标志被触发。
3	CITO	控制IN事务超时中断 在控制IN事务中，如果设备等待的握手包超时，该标志位被触发
2	保留	必须保留为复位值。
1	EPDIS	端点除能 端点除能时，该标志位被触发
0	TF	传输完成 当该端点的所有IN事务完成，该标志位被触发。

设备 OUT 端点 x 中断标志寄存器（USBHS_DOEPxINTF）（x = 0..7, x 是端点编号）

地址偏移：0x0B08 + (x * 0x20)
复位值：0x0000 0000

该寄存器包含 OUT 端点的状态和事件，当获得一个 OUT 端点的中断时，应该读取该端点的中断标志寄存器，从而获知中断源。该寄存器的标志位通常硬件置位，各位写 1 清零。

该寄存器采用字（32位）访问



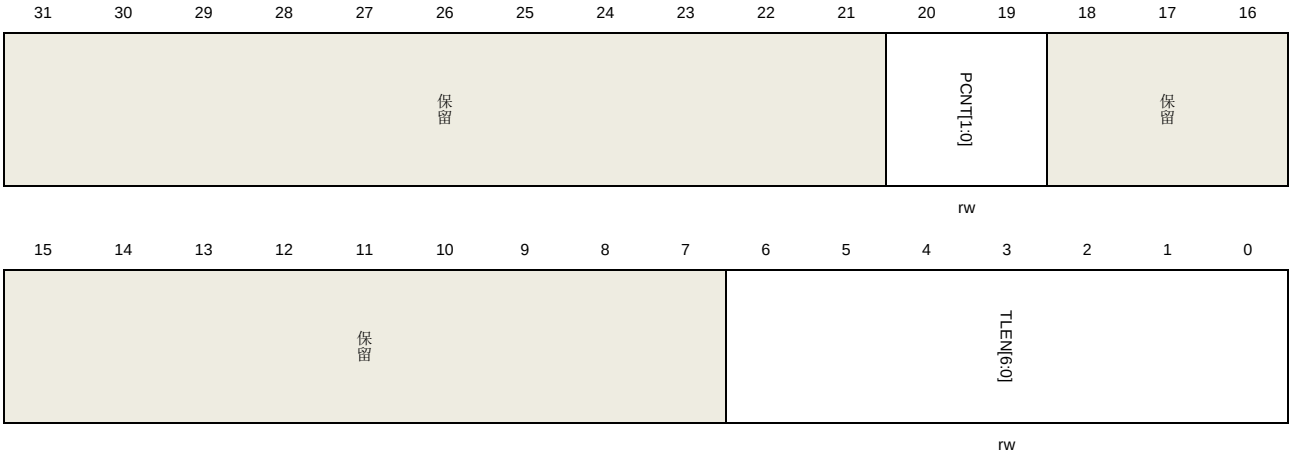
位/位域	名称	描述
31:15	保留	必须保留为复位值。
14	NYET	发送NYET握手 发送NYET握手包时，该标志被触发
13:7	保留	必须保留为复位值。
6	BTBSTP	连续SETUP包（适用于控制OUT端点）

		当一个控制OUT端点接收超过连续3个SETUP包时，该标志被触发。
5	保留	必须保留为复位值。
4	EPRXFOVR	端点Rx FIFO上溢 当OUT令牌被接收时，如果OUT端点的Rx FIFO没有足够的空间存放数据包，该位被触发。在这种情况下，USBHS不能接收OUT数据包，发送一个NAK握手包。
3	STPF	SETUP阶段完成（适用于控制OUT端点） 当一个SETUP阶段完成，也就是USBHS在一个setup令牌后接收了一个IN或OUT令牌，该位被置位。
2	保留	必须保留为复位值。
1	EPDIS	端点除能 端点除能时，该标志位被触发
0	TF	传输完成 当该端点的所有OUT事务完成，该标志位被触发

设备 IN 端点 0 传输长度寄存器（USBHS_DIEP0LEN）

地址偏移：0x0910
复位值：0x0000 0000

该寄存器采用字（32位）访问



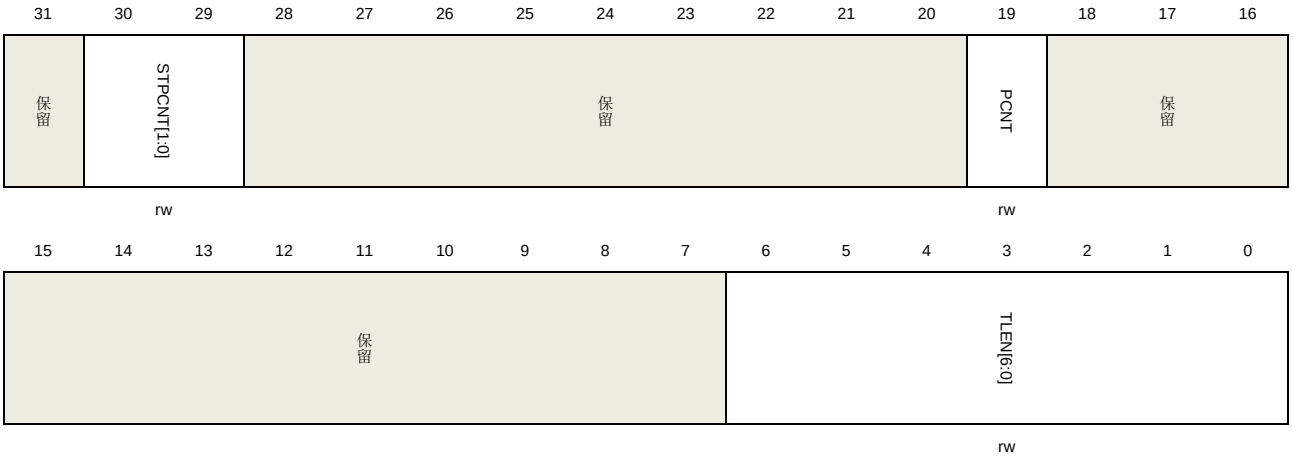
位/位域	名称	描述
31:21	保留	必须保留为复位值。
20:19	PCNT[1:0]	包数 传输中被发送的数据包数量 在端点使能之前，软件设置该位，在传输开始后，该域在每次数据包成功发送后自动减少。
18:7	保留	必须保留为复位值。
6:0	TLEN[6:0]	传输长度

一次传输的数据总字节数
该域是IN传输中需要发送的包数据的总字节数，在端点使能之前，软件设置该位，在软件或DMA成功地将包数据写入端点的Tx FIFO中，该域减少与包数据大小相同的数值。

设备 OUT 端点 0 传输长度寄存器（USBHS_DOEP0LEN）

地址偏移：0x0B10
复位值：0x0000 0000

该寄存器采用字（32位）访问



位/位域	名称	描述
31	保留	必须保留为复位值。
30:29	STPCNT[1:0]	SETUP包计数 该域定义端点可以接受的最大连续SETUP包数量 在SETUP传输之前，设置该域，每当连续SETUP包接收到时，该域值减1，当该域达到0时，寄存器USBHS_DOEP0INTF的BTBSTP标志被触发。 00：0个包 01：1个包 10：2个包 11：3个包
28:20	保留	必须保留为复位值。
19	PCNT	包计数 一次传输中应该接收到包数量。 在端点使能前，软件设置该位，在传输开始后，每当数据包接收到后，该域数值自动减少。
18:7	保留	必须保留为复位值。
6:0	TLEN[6:0]	传输长度 传输中数据总字数。 该域是OUT传输中需要接收的包数据的总字节数，在端点使能之前，软件设置该

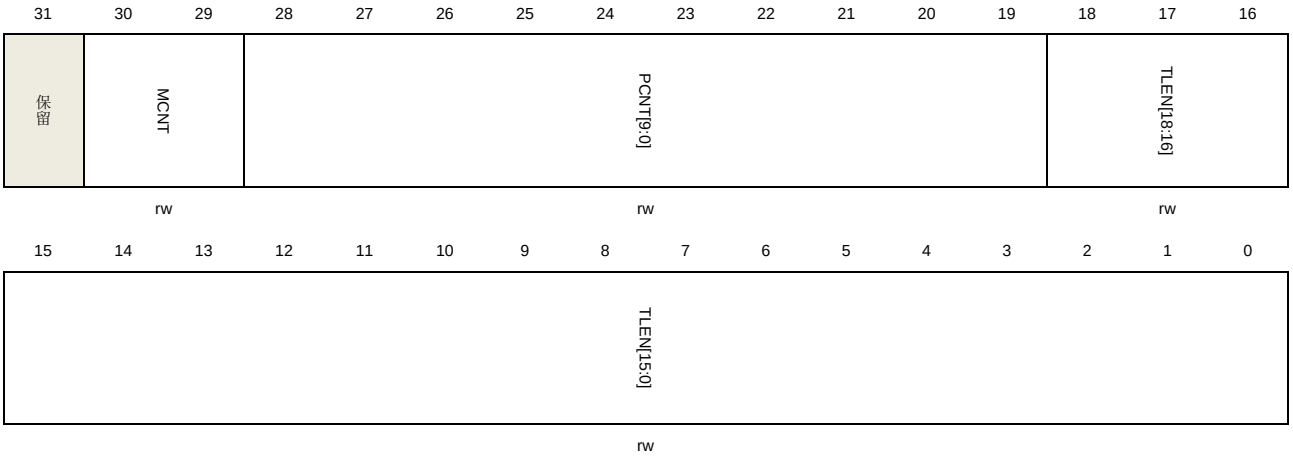
位，在软件或DMA成功地将包数据读取端点的Rx FIFO中，该域减少与包数据大小相同的数值。

设备 IN 端点 x 传输长度寄存器（USBHS_DIEPxLEN）（x = 1..7， x 是端点编号）

地址偏移：0x910 + (x * 0x20)

复位值：0x0000 0000

该寄存器采用字（32位）访问



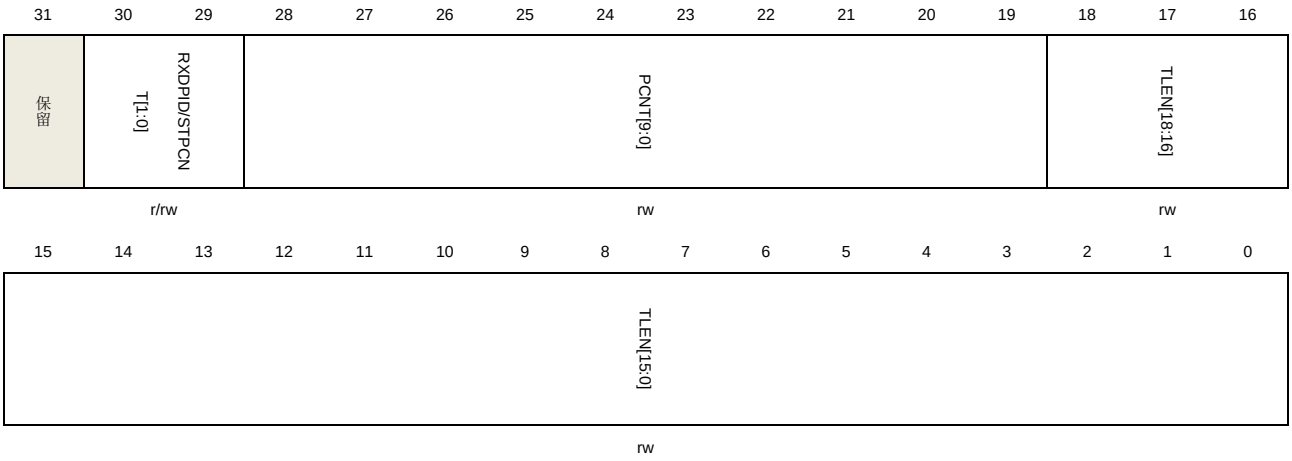
位/位域	名称	描述
31	保留	必须保留为复位值。
30:29	MCNT[1:0]	多包个数 该域描述在一帧内需要传输的包的个数 01:1个包 10:2个包 11:3个包
28:19	PCNT[9:0]	包数量 传输中被发送的数据包数量 在端点使能之前，软件设置该位，在传输开始后，该域在每次数据包成功发送后自动减少。
18:0	TLEN[18:0]	传输长度 传输的数据总字节数 该域是IN传输中需要发送的包数据的总字节数，在端点使能之前，软件设置该位，在软件或DMA成功地将包数据写入端点的Tx FIFO中，该域减少与包数据大小相同的数值。

设备 OUT 端点 x 传输长度寄存器（USBHS_DOEPxLEN）（x = 1..7， x 是端点编号）

地址偏移：0x0B10 + (x * 0x20)

复位值：0x0000 0000

该寄存器采用字（32位）访问

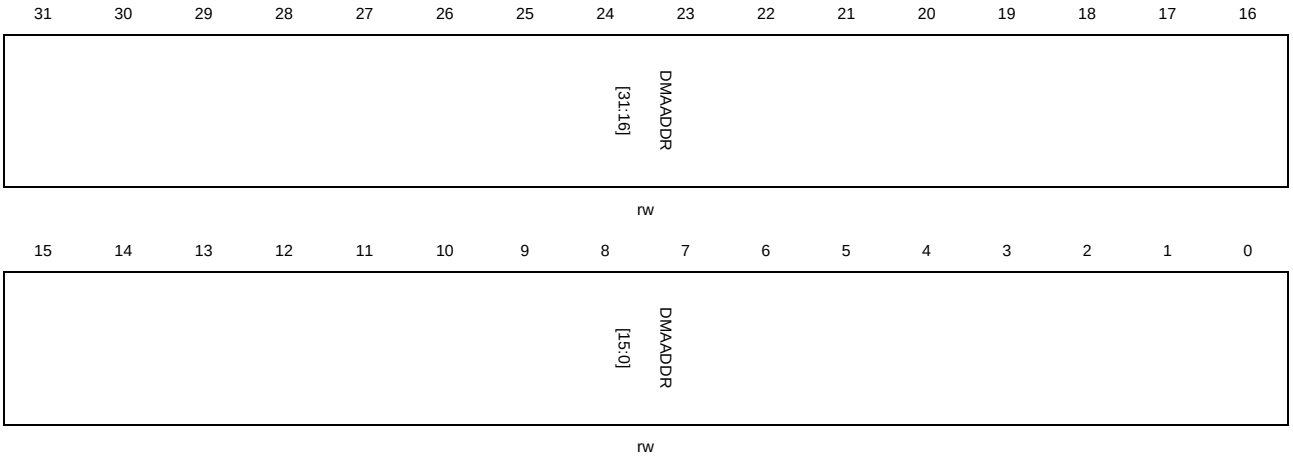


位/位域	名称	描述
31	保留	必须保留为复位值。
30:29	RXDPID[1:0]	接收数据PID（适用于同步OUT端点) 该域保存该端点该数据包所接受的最后一个数据包的PID 00：DATA0 01：DATA2 10：DATA1 11：MDATA
	STPCNT[1:0]	SETUP包数（适用于控制OUT端点) 该位定义该端点可以接受连续SETUP最大包数 在SETUP传输之前，设置该域，每当连续SETUP包接收到时，该域值减1，当该域达到0时，寄存器USBHS_DOEP0INTF的BTBSTP标志被触发。 00：0个包 01：1个包 10：2个包 11：3个包
28:19	PCNT[9:0]	包数 传输中应该接收到包数量 在端点使能前，软件设置该位，在传输开始后，每当数据包接收到后，该域数值自动减少。
18:0	TLEN[18:0]	传输长度 传输中数据总字数 该域是IN传输中需要接收的包数据的总字节数，在端点使能之前，软件设置该位，在软件或DMA成功地将包数据读取端点的Rx FIFO中，该域减少与包数据大小相同的数值。

设备 IN 端点 x DMA 地址寄存器（USBHS_DIEPxDMAADDR）/设备 OUT 端点 x DMA 地址寄存器（USBHS_DOEPxDMAADDR）（x = 0..7， x 是端点编号）

地址偏移：
IN endpoint：0x0914 + (x * 0x20)
OUT endpoint：0x0B14 + (x * 0x20)
复位值：0x0000 0000

该寄存器采用字（32位）访问



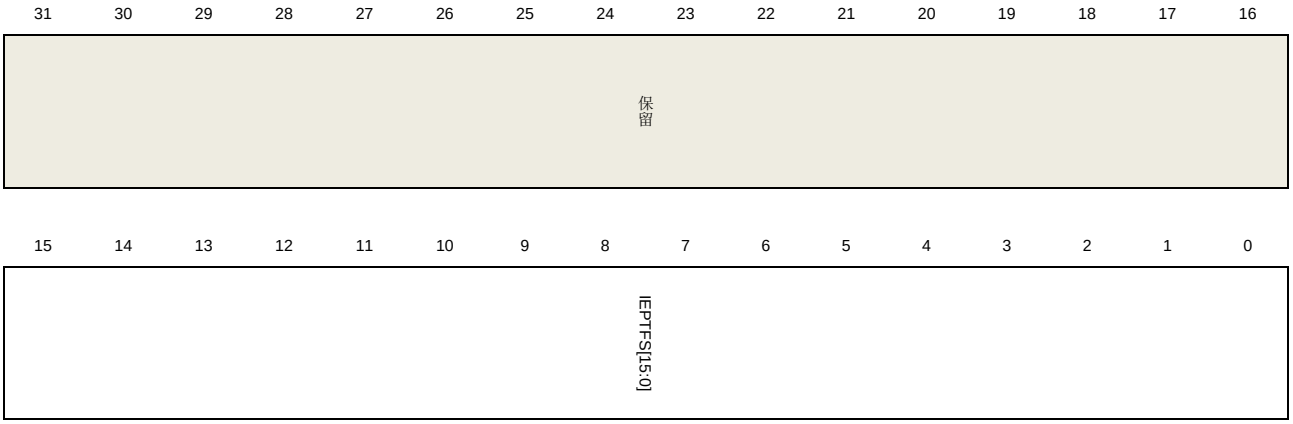
位/位域	名称	描述
31:0	DMAADDR[31:0]	DMA地址 该域定义端点的DMA地址，DMA使用该地址为IN端点提取包数据，或为OUT端点写入包数据。

设备 IN 端点 x 发送 FIFO 状态寄存器（USBHS_DIEPxTFSTAT）（x = 0..7， x 是端点编号）

地址偏移：0x0918 + (x * 0x20)
复位值：0x0000 0200

该寄存器包含每个端点的 Tx FIFO 的信息。

该寄存器采用字（32位）访问



r

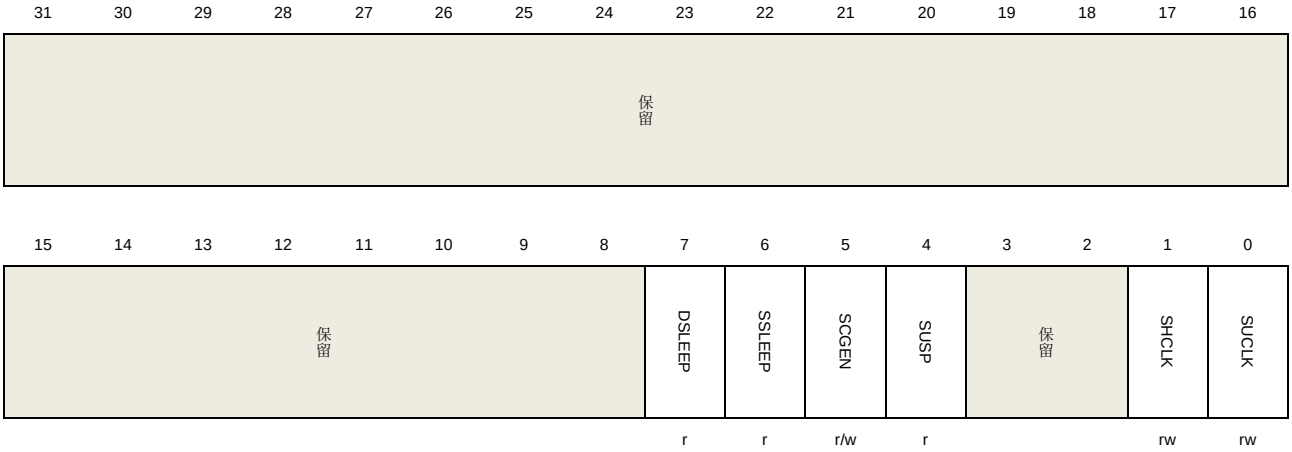
位/位域	名称	描述
31:16	保留	必须保留为复位值。
15:0	IEPTFS[15:0]	IN端点的Tx FIFO可用空间 IN端点的Tx FIFO可用空间用32位字为单位 0：FIFO是满的 1：1字可用 … n：n字可用

49.7.4. 电源和时钟控制寄存器（USBHS_PWRCLKCTL）

地址偏移：0x0E00

复位值：0x0000 0000

该寄存器采用字（32位）访问



位/位域	名称	描述
31:8	保留	必须保留为复位值。
7	DSLEEP	PHY处于深度睡眠状态
6	SSLEEP	PHY处于浅睡眠状态
5	SCGEN	当该位被设置时，内部门控时钟被启用。
4	SUSP	PHY处于挂起状态
3:2	保留	必须保留为复位值
1	SHCLK	停止HCLK 停止HCLK，节省电量 0：HCLK未停止 1：HCLK停止

0	SUCLK	停止USB时钟 停止USB时钟，节省电量 0：USB时钟未停止 1：USB时钟停止

50. EtherCAT®从站控制器（ESC）（适用于 GD32H75E）

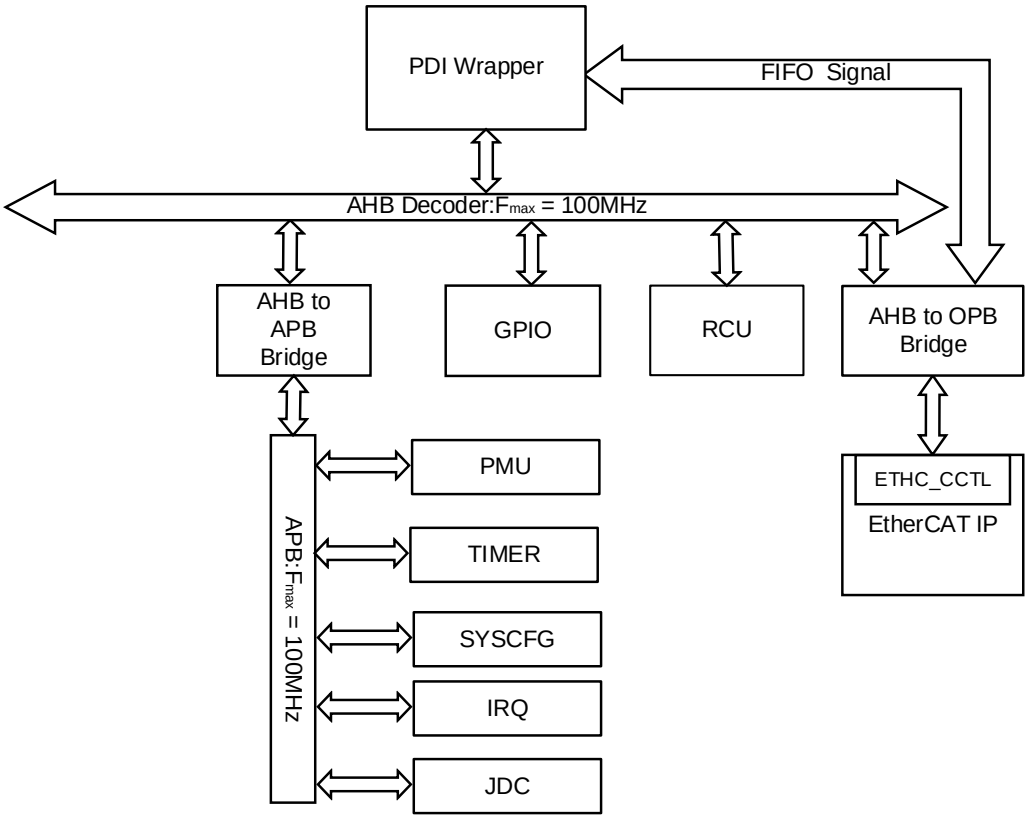
50.1. 系统和总线架构

50.1.1. 总线架构

ESC 的总线架构如下所示。基于 AMBA 5 AHB-LITE 的 AHB 矩阵是一个多层 AHB，它允许系统中一个主设备与多个从设备之间的并行访问路径。AHB 解码器上的一个主设备，包括 PDI 包装器的 AHB 总线。AHB 解码器由五个从设备组成，包括 AHB 到 OPB 桥， AHB 到 APB 桥，GPIO 和 RCU。

AHB 与包括一个 AHB 至 APB 桥接器在内的 AHB 外设相连，该桥接器在 AHB 解码器和一条 APB 总线之间提供完全同步的连接。这条 APB 总线与所有 APB 外设相连，包括电源管理单元（PMU）、定时器（TIMER）、系统配置（SYSCFG）和中断请求（IRQ）。

图 50-1. ESC 的总线架构



50.1.2. 存储器映射

本节将介绍 ESC 的存储器映射分布如下*表 50-1. ESC 存储映射表*所示。

表 50-1. ESC 存储映射表

预先定义的地址空间	地址范围	外设
EtherCAT	0x0000	Type
	0x0001	Revision
	0x0002 - 0x0003	Build
	0x0004	FMMUs Supported
	0x0005	SyncManagers Supported
	0x0006	RAM Size
	0x0007	Port Descriptor
	0x0008 - 0x0009	ESC Features Supported
	0x0010 - 0x0011	Configured Station Address
	0x0012 - 0x0013	Configured Station Alias
	0x0013 - 0x001F	保留
	0x0020	Write Register Enable
	0x0021	Write Register Protection
	0x0022 - 0x002F	保留
	0x0030	ESC Write Enable
	0x0031	ESC Write Protection
	0x0032 - 0x003F	Reserved
	0x0040	ESC Reset ECAT
	0x0041	ESC Reset PDI
	0x0042 - 0x0FF	保留
	0x0100 - 0x0103	ESC DL Control
	0x0104 - 0x0107	保留
	0x0108 - 0x0109	Physical Read/Write Offset
	0x0110 - 0x0111	ESC DL Status
	0x0112 - 0x011F	保留
	0x0120 - 0x0121	AL Control
	0x0122 - 0x012F	保留
	0x0130 - 0x0131	AL Status
	0x0132 - 0x0133	保留
	0x0134 - 0x0135	AL Status Code
	0x0136 - 0x0137	Reserved
	0x0138	RUN LED Override
	0x0139	保留
	0x0140	PDI Control
	0x0141	ESC Configuration
	0x0142 - 0x0143	ASIC Configuration
	0x0144 - 0x0145	RESERVED Register
	0x0146 - 0x014F	保留
	0x0150	PDI Configuration
	0x0151	Sync/Latch PDI Configuration

预先定义的 地址空间	地址范围	外设
	0x0152 - 0x0153	Extended PDI Configuration
	0x0154 - 0x01FF	保留
	0x0200 - 0x0201	ECAT Event Mask
	0x0202 - 0x0203	保留
	0x0204 - 0x0207	PDI AL Event Mask
	0x0208 - 0x0209	保留
	0x0210 - 0x0211	ECAT Event Request
	0x0212 - 0x021F	保留
	0x0220 - 0x0223	AL Event Request
	0x0223 - 0x022F	保留
	0x0300 - 0x0307	RX Error Counter
	0x0308 - 0x030B	Forwarded RX Error Counter
	0x030C	ECAT Processing Unit Error Counter
	0x030D	PDI Error Counter
	0x030E	PDI Error Code
	0x030F	保留
	0x0310 - 0x0313	Lost Link Counter
	0x0314 - 0x03FF	保留
	0x0400 - 0x0401	Watchdog Divider
	0x0410 - 0x0411	Watchdog Time PDI
	0x0420 - 0x0421	Watchdog Time Process Data
	0x0440 - 0x0441	Watchdog Status Process Data
	0x0442	Watchdog Counter Process Data
	0x0443	Watchdog Counter PDI
	0x0444 - 0x04FF	保留
	0x0500	EEPROM Configuration
	0x0501	EEPROM PDI Access State
	0x0502 - 0x0503	EEPROM Control/Status
	0x0504 - 0x0507	EEPROM Address
	0x0508 - 0x050B	EEPROM Data
	0x050C- 0x050F	保留
	0x0510 - 0x0511	MII Management Control/Status
	0x0512	PHY Address
	0x0513	PHY Register Address
	0x0514 - 0x0515	PHY DATA
	0x0516	MII Management ECAT Access State
	0x0517	MII Management PDI Access State
	0x0518 - 0x051B	PHY Port Status
	0x051C - 0x05FF	保留
	0x0600 - 0x06FF	FMMU
	0x0700 - 0x07FF	Reserved

预先定义的 地址空间	地址范围	外设
	0x0800 - 0x087F	SyncManager
	0x0880 - 0x08FF	保留
	0x0900 - 0x09FF	Distributed Clocks (DC)
	0x0A00 - 0x0AFF	保留
	0x0E00 - 0x0E07	Product ID
	0x0E08 - 0x0E0F	Vendor ID
	0x0E10 - 0x0EFF	保留
	0x0F00 - 0x0F03	Digital I/O Output Data
	0x0F04 - 0x0F0F	保留
	0x0F10 - 0x0F17	General Purpose Outputs
	0x0F18 - 0x0F1F	General Purpose Inputs
	0x0F20 - 0x0F7F	保留
	0x0F80 - 0x0FFF	User RAM
	0x1000 - 0x2FFF	Process Data RAM
外设	0x3300 - 0x33FF	AHB2OPB Bridge
	0x3400 - 0x34FF	RCU
	0x3500 - 0x35FF	GPIO
	0x3700 - 0x37FF	PMU
	0x3800 - 0x38FF	TIMER
	0x3900 - 0x39FF	SYSCFG
	0x3A00 - 0x3AFF	IRQ

50.1.3. AHB 直接/间接访问

ESC 可以通过 AHB 到 OPB 桥以三种方式直接/间接访问 ESC 寄存器和内核 PRAM：直接可访问的 ESC 寄存器、间接可访问的 ESC 内核寄存器和间接可访问的 ESC 内核 PRAM。该桥向上游侧提供 AHB 从接口，向下游侧提供 OPB 接口，其中上游 AHB 侧的频率比下游 ESC 内核快，并且时钟相位同步，具有 N:1（最大 N=16）的频率比。

直接 AHB 传输访问 ESC 控制器寄存器

直接访问 ESC 寄存器用于将数据/命令传输到间接访问 ESC 内核寄存器。当 AHB 总线访问地址范围为 0x0000-0x0FFF 时，将启动对 ESC 内核的单个寄存器读取操作。开始读取周期时，CCTL_BUSY 设置为 1，读取周期结束时，CCTL_BUSY 设置为 0。可以在 AHB 总线上读取有效数据。有效数据与总线上的低位对齐。当 AHB 总线访问地址范围为 0x0000-0x0FFF 时，将启动对 ESC 内核的单个寄存器写入操作。写入周期开始时，CCTL_BUSY 被设置。写入周期结束时，CCTL_BUSY 设置为 0。寄存器中的有效数据被写入，有效数据与总线上的低位对齐。

间接传输访问 ESC 内核寄存器

ESC 可以通过 ESC_CCTL_DATA 和 ESC_CCTL_CMD 寄存器间接访问 ESC 内核寄存器。当读取 ESC 内核寄存器时，需要执行以下步骤：首先写入 ESC CCTL 命令寄存器（CCTL_CMD）一次；将 ESC_CCTL Busy（CCTL_BUSY）位设置为 1；将 ESC_CCTL 地址（CCTL_ADDR）

字段设置为所需的寄存器地址；将读写（CCTL_RW）设置为 1；并将 ESC_CCTL 大小（CCTL_SIZE）字段设置为所需的大小。当 CCTL_BUSY 位被清除时，就可以读取有效数据，即可以从 ESC_CCTL 数据寄存器（ESC_CCTL_DATA）中读取数据。

在写入 ESC 内核寄存器时，只需一次写入 ESC_CCTL 命令寄存器（ESC_CCTL_CMD）。将 ESC_CCTL 繁忙（CCTL_BUSY）设置为 1；将 ESC_CCTL 地址（CCTL_ADDR）字段设置为所需的寄存器地址；清除读写（CCTL_RW）位，并将 ESC_CCTL 大小（CCTL_SIZE）字段设置为所需的大小。写入周期的完成由 ESC_CCTL 繁忙（CCTL_BUSY）位被清除为零来指示。

在上述读写操作中，有效数据始终与 ESC_CCTL 数据寄存器（CCTL_DATA）的低位对齐。有效数据可以参考下表。

表 50-2. 有效数据的对齐

CCTL_SIZE	ESC CCTL_ADDR[1:0]	ESC CCTL_DATA valid bytes
1	00/01/10/11	[7:0]/ [15:8]/ [23:16]/ [31:24]
2	00/10	[15:0]/[31:16]
4	00	[31:0]

间接传输访问 ESC 内核 PRAM

当通过 AHB 对内核 PRAM 发起读操作时，写入 PRAM 起始地址和读长度到 ESC_PRAM_ALR 寄存器后，将 PRAM_BUSY_READ 设置为 1，模块开始发起多个 OPB 读操作，从内核 PRAM 读取数据并写入 TX FIFO。所有 OPB 读操作完成后，清除 PRAM_BUSY_READ。当数据从 ESC 内核传输到 TX FIFO 时，PRAM 读长度 PRAM_LEN_READ 和 PRAM 读地址 PRAM_ADDR_READ 会更新以显示进程。根据起始地址确定第一次读数据的有效字节。根据起始地址和操作长度确定最后一次读数据的有效字节。如果需要，可以通过将 ESC_PRAM_CR[PRAM_STOP_READ]位设置为 1 来停止读命令。如果 OPB 读周期已经开始，停止命令在当前读操作完成后生效。停止命令生效后，TX FIFO 中的数据会被清除。

通过 AHB 向 PRAM 写入时，先写入 RAM 起始地址和写入长度到 ESC_PRAM_ALW 寄存器，然后向 ESC_PRAM_CW 寄存器的 PRAM_BUSY_WRITE 位写入 1，模块启动多个 OPB 写操作。从 RX FIFO 读取数据并写入内核 PRAM。所有 OPB 写操作完成后，清除 PRAM_BUSY_WRITE。写操作支持等待机制，PRAM_BUSY_WRITE 位被设置为 1，但是当 RX FIFO 为空时，OPB 模块不会立即启动 OPB 传输操作，直到 RX FIFO 中有数据才会启动此操作。在每次 OPB 确认响应后，它将检测 RX FIFO 的状态。如果 RX FIFO 为空，它将进入等待状态并进入下一个传输操作，直到有数据。当数据从 RX FIFO 传输到内核时，PRAM 写入长度 PRAM_LEN_WRITE 和 PRAM 写入地址 PRAM_ADDR_WRITE 将更新以显示进程。根据起始地址确定第一次数据写入的有效字节。根据起始地址和操作长度，确定最后一次数据写入的有效字节。

如果需要，可以通过将 PRAM_STOP_WRITE 位设置为 1 来停止写命令。如果 OPB 写周期开始，停止命令在读取操作完成后生效。停止命令生效后，RX FIFO 中的数据将被清除。

50.1.4. 忙碌状态下的寄存器保护

如果 BUSY 相关的寄存器位，例如 CCTL_BUSY、PRAM_BUSY_READ 和 PRAM_BUSY_WRITE，被设置为 1，您可以设置 BRP 位以防止寄存器被重写。当 CCTL_BUSY

设置为 1 时，ESC_CCTL_DATA/ESC_CCTL_CMD 将受到 AHB 写操作的保护，防止被重写。当 PRAM_BUSY_READ 设置为 1 时，ESC_PRAM_ALR 将受到 AHB 写操作的保护，防止被重写。当 PRAM_BUSY_WRITE 设置为 1 时，ESC_PRAM_ALW 将受到 AHB 写操作的保护，防止被覆盖。

当 BRP 位设置为 1 时，当与 BUSY 相关的寄存器位设置为 1 时，用户对相应寄存器的 AHB 写操作将丢失，然后 ESC_OPB_CS 寄存器中的 WDLF 标志将设置为 1，并且当 WDIE 位设置为 1 时，将触发中断。

当 BRP 位设置为 0，当与 BUSY 相关的寄存器位设置为 1 时，用户对相应寄存器的 AHB 写操作将导致当前 OPB 传输错误，然后 ESC_OPB_CS 寄存器中的 WEF 标志将被设置为 1，并且当 WEIE 位设置为 1 时将触发中断。建议尽快处理以避免更多错误。

50.1.5. OPB 传输超时功能

OPB 传输超时功能可以通过将 ESC_OPB_CS 寄存器中的 TOEN 位设置为 1 来启用。超时间隔可以通过 ESC_OPB_CS 寄存器中的 TO_CNT 位进行配置。当计数器超过 TO_CNT 编程值时，ESC_OPB_CS 寄存器中的 TOF 标志将被设置为 1。如果 ESC_OPB_CS 寄存器中的 TOIE 位设置为 1，则会生成超时中断。超时中断响应处理：

如果使用 ESC CCTL 直接读写模式，当超时中断响应发生后，TOF 标志会被设置。直接终止传输以避免总线被占用。

如果以太网控制器(ESC)的通信控制列表(CCTL)在间接读写模式下使用，超时中断响应发生后会设置 TOF 标志。可以通过写入 1 到 ESC CCTL_STOP 位来停止这个操作。

当 PRAM 以间接读写模式使用时，超时中断响应发生后会设置 TOF 标志。您可以通过写入 1 到 PRAM_STOP_WRITE/PRAM_STOP_READ 位来停止此操作。

50.1.6. ESC 内核控制器（ESC_CCTL）

ESC 内核控制器（ESC_CCTL）的主要目的如下：

- 配置以太网控制器（ESC）内核寄存器的间接传输访问。
- 配置 ESC 内核 PRAM 的间接传输访问

50.1.7. ESC 内核控制器寄存器定义

ESC 内核控制寄存器基地址：0x3300

ESC_CCTL_DATA 数据寄存器(ESC_CCTL_DATA)

地址偏移：0x00
复位值：0x0000 0000

这个寄存器必须通过字（32位）访问。

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

CCTL_DATA[15:0]
rw

位/位域	名称	描述
31:0	CCTL_DATA[31:0]	ESC CCTL 数据 此字段指示从 ESC 内核读取或写入的值。读取或写入取决于 ESC_CCTL_CMD[CCTL_RW]位。如果 CCTL_RW 位为 1，则此值为从 ESC 核心读取的数据；如果 CCTL_RW 位为 0，则此值为写入 ESC 核心的数据。 此字段的最低位始终指示已写入或读取的有效数据。

ESC CCTL 命令寄存器（ESC_CCTL_CMD）

地址偏移：0x04

复位值：0x0000 0000

这个寄存器必须通过字（32位）访问。

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

位/位域	名称	描述
31	CCTL_BUSY	CCTL 忙。如果将该位写为零，则无效。 0：没有进行读写（根据 CCTL_RW）操作。 1：正在进行读写操作。 **注意：**当读写操作完成时，此位将被清除。然后，HOST 可以从 ESC_CCTL_DATA 寄存器读取有效数据或写入有效数据。当此位为 0 时，可以修改 ESC_CCTL_CMD 和 ESC_CCTL_DATA 寄存器。
30	CCTL_RW	读写操作 0：写操作 1：读操作
29	CCTL_STOP	停止读操作或写操作。如果将该位写为零，则无效 0：无效果 1：停止 ESC 核心寄存器的读或写操作 **注意：**当 CCTL_BUSY 被清除时，CCTL_STOP 将被清除
28:19	保留	必须保持在复位值
18:16	CCTL_SIZE[2:0]	此字段指定 ESC CCTL 大小（字节）。1、2 和 4 是有效的，其他值是无效的。更多细节请参阅*表 50-2. 有效数据的对齐*。
15:0	CCTL_ADDR[15:0]	此字段指定将被访问的 ESC 核心寄存器的地址。

ESC PRAM FIFO 数据读取寄存器（ESC_PRAM_FIFO_DR）

地址偏移：0x10

复位值：0x0000 0000

这个寄存器必须通过字（32位）访问。

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

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

Bits	Fields	Descriptions
31:0	PRAM_FIFO_DATA_READ[31:0]	从 ESC PRAM 读取的数据。 数据的有效值是根据起始地址和传输长度来确定的。

ESC PRAM 地址和长度读取寄存器（ESC_PRAM_ALR）

地址偏移：0x14

复位值：0x0000 0000

这个寄存器必须通过字（32位）访问。

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

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

位/位域	名称	描述
31:16	PRAM_LEN_READ[15: 0]	从 ESC PRAM 读取的数据长度，以字节为单位。当数据被读入 FIFO 时，此字段会递减。 **注意：**当 PRAM_BUSY_READ 为 1 时，此字段不能被修改。
15:0	PRAM_ADDR_READ [15:0]	ESC PRAM 数据读取地址。当数据被读入 FIFO 时，此字段会递增。 **注意：**当 PRAM_BUSY_READ 为 1 时，此字段不能被修改。

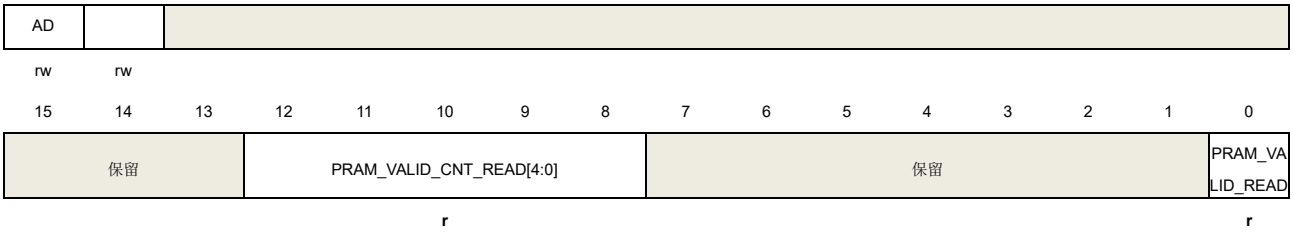
ESC PRAM 命令读取寄存器（ESC_PRAM_CR）

地址偏移：0x18

复位值：0x0000 0000

这个寄存器必须通过字（32位）访问。

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

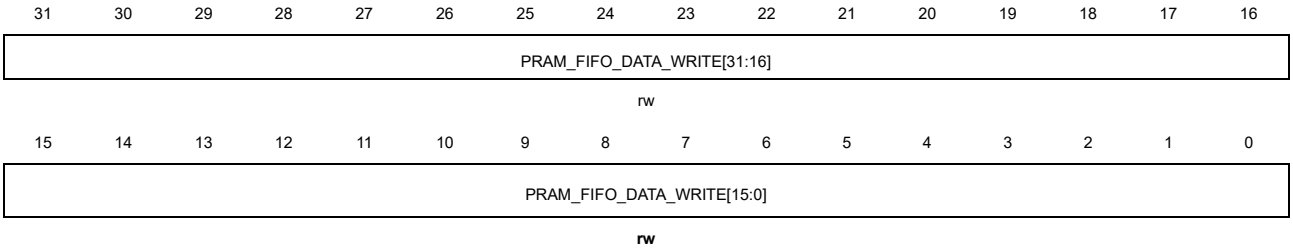


位/位域	名称	描述
31	PRAM_BUSY_READ	PRAM 是否正在被读取。如果该位被写为零，则无效。 0: 没有 PRAM 读取操作 1: PRAM 正在被读取 **注意：**当读取操作完成时，此位会被清除。
30	PRAM_STOP_READ	停止 PRAM 读取操作。如果该位被写为零，则无效。 0: 无效果 1: 停止 PRAM 读取操作 **注意：**在此位被设置为 1 之后，PRAM_BUSY_READ 将被清除，并且 RX FIFO 将被复位。然后此位将自动清除。
29:13	保留	必须保持复位值。
12:8	PRAM_DATA_CNT_READ[4:0]	PRAM 数据读取有效计数 当数据由 PRAM 读取到 RX FIFO 时，此计数增加。MCU 读取 RX FIFO,此计数减少。
7:1	保留	必须保持复位值。
0	PRAM_VALID_DATA_READ	PRAM 有效数据读取 0: 没有有效数据可读。 1: 有有效数据可读。

ESC PRAM FIFO 数据写寄存器（ESC_PRAM_FIFO_DW）

地址偏移：0x20
复位值：0x0000 0000

这个寄存器需要通过字（32位）访问。



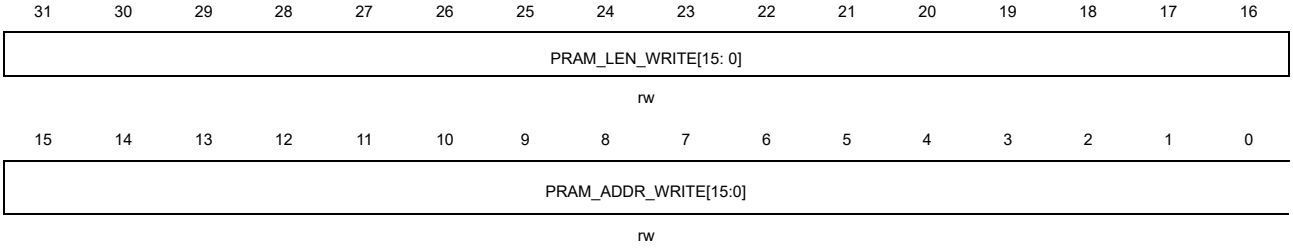
位/位域	名称	描述
31:0	PRAM_FIFO_DATA_WRITE[31:0]	写入 ESC PRAM 的数据。 数据的有效值是根据起始地址和传输长度来确定的。

ESC PRAM 地址和长度写寄存器（ESC_PRAM_ALW）

地址偏移：0x24

复位值：0x0000 0000

这个寄存器必须通过字（32位）访问。



位/位域	名称	描述
31:16	PRAM_LEN_WRITE[15: 0]	写入 ESC PRAM 的数据长度，以字节为单位。当数据被写入 FIFO 时，此字段会递减。 **注意：**当 PRAM_BUSY_WRITE 为 1 时，此字段不能被修改。
15:0	PRAM_ADDR_WRITE[15:0]	ESC PRAM 数据写入地址。当数据从 FIFO 中读取时，此字段会递增。 **注意：**当 PRAM_BUSY_WRITE 为 1 时，此字段不能被修改。

ESC PRAM 命令写寄存器（ESC_PRAM_CW）

地址偏移：0x28

复位值：0x0000 1001

这个寄存器必须通过字（32位）访问。



位/位域	名称	描述
31	PRAM_BUSY_WRITE	PRAM 是否正在被写入。如果该位被写为零，则无效。 0: 没有 PRAM 写入操作 1: PRAM 正在被写入 **注意：**当写入操作完成时，此位会被清除。
30	PRAM_STOP_WRITE	停止 PRAM 写操作。如果该位写为 0，则没有效果 0: 无效果 1: 停止 PRAM 写操作 **注意：**当该位被设置为 1 后，PRAM_BUSY_WRITE 会被清除，并且 TX FIFO 会

被复位。然后该位会自动清除。

29:13	保留	必须保持复位值。
12:8	PRAM_VALID_CNT_WRITE[4:0]	PRAM 数据写入有效计数 当数据由 MCU 写入 TXFIFO,此计数减少。TX FIFO 写入到 PRAM 时，此计数增加。
7:1	保留	必须保持复位值。
0	PRAM_VALID_DATA_WRITE	PRAM 有效数据写入 0: 没有有效数据要写入 1: 有有效数据要写入。

ESC OPB 控制和状态寄存器（ESC_OPB_CS）

地址偏移：0x30

复位值：0x0000 04A0

这个寄存器必须通过字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RAAF	RAAIE	WDLF	WDLIE	TOF	TOIE	WEF	WEIE	ESC CCTLIVF	ESC CCTLIVIE	保留					
rc_w1	rw	rc_w1	rw	rc_w1	rw	rc_w1	rw	rc_w1	rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					BRP	TO_CNT[8:0]									TOEN
					rw	rw									rw

位/位域	名称	描述
31	RAAF	保留地址访问标志。当访问的地址位于保留的地址段内时，此标志会被触发。通过向它写入 1，可以清除此标志。 0: 访问的地址是一个有效的地址段 1: 访问的地址是一个保留的地址段
30	RAAIE	保留地址访问中断使能 0: 中断被禁止 1: 每当 RAAF 位被设置时，就会发生中断
29	WDLF	写数据丢失标志。当 BRP 启用时，如果忙标志高，则 AHB 写操作会导致数据丢失，此标志将被触发。此标志可以通过写入 1 来清除。 0: 没有数据丢失 1: 此次传输的数据已丢失
28	WDLIE	写数据丢失中断使能 0: 中断被禁止 1: 每当 RAAF 位被设置时，就会发生中断
27	TOF	超时标志，当单次传输时间超过设定值时，此标志将被触发。此标志可以通过写入 1 来清除。

		0: 没有发生 ESC 核心超时传输 1: 发生了 ESC 核心超时传输
26	TOIE	超时中断使能 0: 中断被禁止 1: 每当 RAAF 位被设置时，就会发生中断 t
25	WEF	写错误标志。当 BRP 禁用时，如果忙标志高，AHB 写操作将导致当前 OPB 传输错误，此标志将被触发。此标志可以通过写入 1 来清除
24	WEIE	写错误中断使能。 0: 中断被禁止 1: 每当 WEF 位被设置时，就会发生中断
23	ESC CCTLIVF	CCTL_SIZE 和 CCTL_ADDR 非法值标志。写入寄存器的 CCTL_SIZE 和 CCTL_ADDR 值不符合要求，导致此次传输无法进行，此标志将被触发。此标志可以通过写入 1 来清除。
22	ESC CCTLIVIE	CCTL_SIZE 和 CCTL_ADDR 非法值中断使能。 0: 中断被禁止 1: 每当 IESC CCTLVF 位被设置时，就会发生中断
21:11	保留	必须保持复位值。.
10	BRP	当忙标志高时，防止寄存器变化 1: 保护生效 0: 保护不生效
9:1	TO_CNT[8:0]	超时计数器 此字段表示在未收到 ACK 响应的情况下传输超时的位数；程序员数据不能小于手册推荐（60）的最小值。
0	TOEN	超时使能 此字段表示该功能是否生效 1: 启用超时特性 0: 禁用超时特性。

50.1.8. 系统配置控制器（SYSCFG）

系统配置控制器（SYSCFG）的主要目的如下：

- 配置微控制器HCLK频率比例
- 配置SPI扩展模式
- 提供芯片ID和版本

50.1.9. 系统配置寄存器定义

系统配置基地址：0x3900

系统配置寄存器 0（SYSCFG_CFG0）

地址偏移：0x00
复位值：0x001F 0000

该寄存器可以按字（32位）访问。

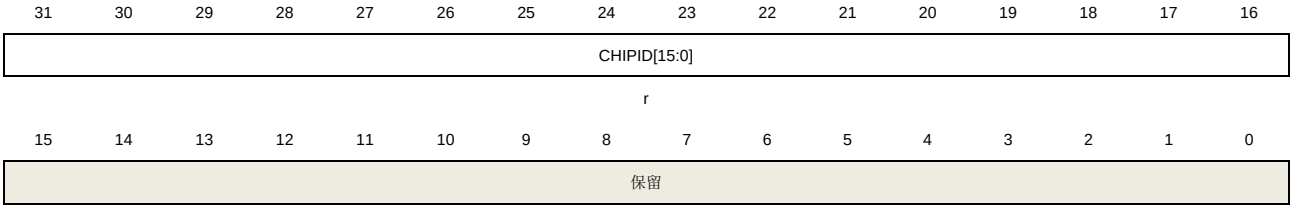


位/位域	名称	描述
31:21	保留	必须保持复位值。.
20:16	MCUFREQ[4:0]	MCU HCLK 频率比率 00000: MCU_HCLK_FREQ >=100MHz 00001: 100MHz/2 <= MCU_HCLK_FREQ < 100MHz 00010: 100MHz/3 <= MCU_HCLK_FREQ < 100MHz/2 …. 11111: 100MHz/32 <= MCU_HCLK_FREQ < 100MHz/31
15:3	保留	必须保持复位值。.
2	SPIEXTMOD	PDI 类型与 SPI 结合 0: GPIO 1: MII (OSPI: 不输出 clk_25m; 其他: 输出 clk_25M)
1:0	保留	必须保持复位值。.

系统配置芯片 ID 寄存器（SYSCFG_CHIPID）

地址偏移：0x90
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	CHIPID[15:0]	芯片 ID

15:0	保留	必须保持复位值。.

系统配置芯片版本寄存器（SYSCFG_CHIPVER）

地址偏移：0x94

复位值：0x0000 0000

该寄存器可以按字（32位）访问。



位/位域	名称	描述
31:8	保留	必须保持复位值。.
15:0	CHIPVER[7:0]	芯片版本

系统配置保留寄存器（SYSCFG_RESERVED）

地址偏移：0xF0

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。
15:1	RESERVED[15:1]	保留寄存器
0	OEEXTCFG	OE_EXT输出时钟配置位 0：打开OE_EXT时钟输出 1：关闭OE_EXT时钟输出 **注：**该bit默认配置为0打开OE_EXT时钟输出，可能会导致EMI辐射超标，关注EMI辐射问题时将该bit置1关闭OE_EXT时钟输出。

50.2. 电源管理单元（PMU）

50.2.1. 简介

功耗设计是 ESC 系列产品比较注重的问题之一。电源管理单元（PMU）提供了四种设备级别的节能模式和三种模块级别的节能模式，设备级别的节能模式包括 MOD0、MOD1、MOD2 和 MOD3，模块级别的节能模式包括 EtherCAT 时钟管理、PHY 电源管理和 LED 引脚电源管理。这些模式降低了功耗，并允许应用程序在设备运行时间、速度和功耗之间的冲突需求之间实现最佳权衡。PMU 还支持唤醒事件检测和电源管理事件（PME）通知。

50.2.2. 主要特征

- EtherCAT 时钟管理；
- PHY 功耗管理，包括 PHY A 和 B 的能量检测(ED)掉电管理和通用掉电管理；
- LED 引脚掉电管理；
- 四种设备级别的节能模式，包括 MOD0、MOD1、MOD2 和 MOD3；
- PHY 唤醒事件检测，包括 PHY ED 上电唤醒和 PHY LAN 魔术包唤醒；
- 中断唤醒通知。

50.2.3. 功能说明

设备就绪

PMU_CTL0 寄存器中的 RDY 位可以指示设备是否就绪。主机可以读取此位以获取设备的就绪状态。

- 在上电后，如果 EtherCAT 设备复位或数字复位，并且 RDY 位置位，这表明设备已成功读取 EEPROM 中的内容，并根据读取的内容进行了配置；
- 将 RCU_RSTCFG 寄存器中的 ESCRST 位置为 1，会复位 EtherCAT 内核，从而使 EtherCAT 重新读取 EEPROM 并根据读取的配置内容重新配置设备。在这个过程中，RDY 位会短暂地变为低电平；
- 当设备进入节能模式 MOD1、MOD2 或 MOD3 时，RDY 位将变为低电平。一旦设备从节能模式唤醒回到 MOD0，并且 PLL 稳定后，RDY 位将重新设置为高电平。

注意：设备支持电压检测。当供电电压达到预定值时，RDY 位将置位。

PHY 唤醒事件检测

该设备支持两种类型的 PHY 唤醒事件检测：

- PHY ED 上电唤醒事件；
- PHY LAN 魔术包唤醒。

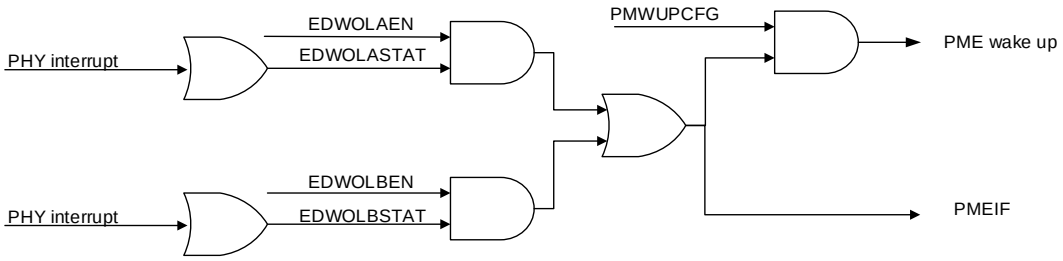
PME 唤醒通知

PME 模块负责处理 PMU_CTL 寄存器中的 EDWOLASTAT 和 EDWOLBSTAT 位的锁存功能。

可以参考*图 50-2. PME 中断待处理*，了解 PME 中断控制的逻辑。如果置位了 EDWOLAEN 或 EDWOLBEN，当端口 A 或 B 的 PHY 发生能量检测 / WoL 事件时，中断状态寄存器中的 PMEIF 位将被置位。

当置位 PMWUPCFG 时，PME 事件可以在某些设备级节能模式下自动唤醒系统。

图 50-2. PME 中断待处理



模块级节能模式

设备支持三种类型的模块级节能模式：

- PMU_CTL0 寄存器中的 ECATCLKDIS 位可以用来禁用 EtherCAT 内核时钟；
- PHY 电源管理：
  - PHYA 和 PHYB 掉电管理，支持自动能量检测掉电管理；
  - 通用掉电管理。
- LED 输出管理：
  - PMU_CTL0 寄存器中的 LEDOUTDIS 位可以用来禁用 LED 输出；
  - PMU_CTL0 寄存器中 LEDMODCFG 位可以用来配置 LED 的工作模式（仅当 LEDOUTDIS 置位时生效）；
  - PMU_CTL0 寄存器中 LEDINACT 位可用于配置 LED 在推挽模式下工作时的非活动状态（仅当 LEDOUTDIS 置位时才有效）。

设备级节能模式

设备级复位后，设备以全功能运行，所有时钟都处于激活状态。用户可以通过关闭未使用功能的时钟来实现更低的功耗。此外，还提供了四种设备级节能模式，以实现更低的功耗，它们是 MOD0、MOD1、MOD2 和 MOD3。

MOD0

设备级复位后，设备以 MOD0 模式运行，并处于全功能状态，所有时钟都处于激活状态。

MOD1

在 MOD1 模式下，设备将禁用所有来自 PLL 时钟的衍生时钟。如果通过 PHY 或外部供电，网络时钟仍然保持启用状态。晶体振荡器和 PLL 仍然保持启用。这种模式可以通过手动或自动方式退出。

这种模式适用于 PHY 的通用掉电管理、PHY 的 WoL（局域网唤醒）模式以及 PHY 的能量监

测掉电管理。

MOD2

在 MOD2 模式下，设备将禁用所有来自 PLL 时钟的衍生时钟。如果通过 PHY 或外部供电，网络时钟仍然保持启用状态。允许禁用 PLL（如果两个 PHY 都处于 ED 或通用掉电管理，PLL 将被禁用）。XTAL 和 PLL 仍然保持启用状态。这种模式可以通过手动或自动方式退出。

MOD2 适用于 PHY 的通用掉电管理、PHY 的 WoL（局域网唤醒）模式以及 PHY 的能量监测掉电管理。

MOD3

在 MOD3 模式下，设备将禁用所有由 PLL 时钟衍生的时钟。PLL 被禁用。外部网络时钟被关闭。晶体振荡器被禁用。这种模式只能手动退出。

MOD3 适用于 PHY 的通用掉电管理。

在设置此电源状态之前，主设备应置位 POWERDOWN 位。将 PHY 置于通用掉电模式。

表 50-3. 节能模式总结

Mode	PLL	System clocks	Network clocks	XTAL
MOD0	ON	ON	usable	ON
MOD1	ON	OFF	usable	ON
MOD2	OFF	OFF	usable	ON
MOD3	OFF	OFF	OFF	OFF

进入设备级节能模式

要从 MOD0 转换到 MOD1、MOD2 或 MOD3，可以遵循以下步骤进行：

1. 配置 PMWUPCFG 位；
2. 配置 PHY 唤醒检测，关于 PHY 唤醒检测可以参考 ***PHY 唤醒事件检测***；
3. 配置 PHY 唤醒通知，关于 PHY 唤醒通知可以参考 ***PME 唤醒通知***；
4. 确保设备已成功进入节能模式（确保无需发送数据包，接收器已禁用等）；
5. 设置 PMSLPEN 位。

注意：

- 进入省电模式后，寄存器 PMU_CTL0 中的 RDY 位将被设置为低电平；
- 进入省电模式后，主机接口将无效。

退出设备级别的节能模式

设备级别的节能模式可以手动或自动退出。

如果设置了 PMWUPCFG 位，则启用了 PME 唤醒功能，可能会自动唤醒 PME。关于 PME 唤醒，可以参考 ***PME 唤醒通知***。

主机可以通过如下操作手动唤醒设备：

- 对设备执行 SPI/SQI 周期（SCS#低，SCK 高）。尽管在设备被唤醒之前，所有的读写操

作都被忽略，但主机仍应通过读取 PMU_PDIREFVAL 寄存器来指示唤醒设备。在设备被唤醒之前，不应尝试读写任何其他地址。

注意：

- 通过读取 PMU_PDIREFVAL 寄存器可以确定主接口的工作状态。一旦读取到正确的值，主接口将进入就绪状态。然后，RDY 位将指示设备完全唤醒；
- 在自动或手动唤醒后，一旦设备返回到 MOD0 并且 PLL 重新稳定，设备就绪（RDY）位就会被设置，PMMODCFG 和 PMSLPEN 将被清除（置为 0）；
- 如果一切正常，设备的唤醒时间应该小于 2 毫秒。

50.2.4. 寄存器定义

PMU 基地址：0x0000 3700

控制寄存器 0（PMU_CTL0）

地址偏移：0x00
复位值：0x0000 C000

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
PMMODCFG[1:0]		保留		PMWUPCFG	LEDOUTDIS	LEDMODCFG	LEDINACT	保留		ECATCLKDIS	保留			EDWOLBSTAT	EDWOLASTAT
rw				rw	rw	rw	rw			rw				r_w1	r_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EDWOLBEN	EDWOLBEN	保留													RDY
rw	rw														r

位/位域	名称	描述
31:30	PMMODCFG[1:0]	电源管理模式配置位 当 PMSLPEN 被设置时，这些位可以用来配置电源管理模式。这些位在设备唤醒时会被清除。 00：MOD0 01：MOD1 10：MOD2 11：MOD3
29:28	保留	必须保持复位值。
27	PMWUPCFG	电源管理唤醒模式配置位 0：由主机唤醒 1：由 PME 或者主机唤醒
26	LEDOUTDIS	LEDs 输出失能位 当此位被置位时，LEDs 的输出将被失能。在漏极开路/源极开路模式下工作时，LEDs 不会被驱动；在推挽模式下工作时，LEDs 仍将被驱动，但 LEDs 的状态将变为非激活状态。 0：LEDs 输出被启用 1：LEDs 输出被禁用
25	LEDMODCFG	LEDs 工作模式配置位（仅当 LEDOUTDIS 置位时生效） 0：LEDs 的工作模式为漏极开路/源极开路 1：LEDs 的工作模式为推挽
24	LEDINACT	推挽模式下 LEDs 非激活状态配置位（仅当 LEDOUTDIS 置位时生效） 0：0 表示非激活状态

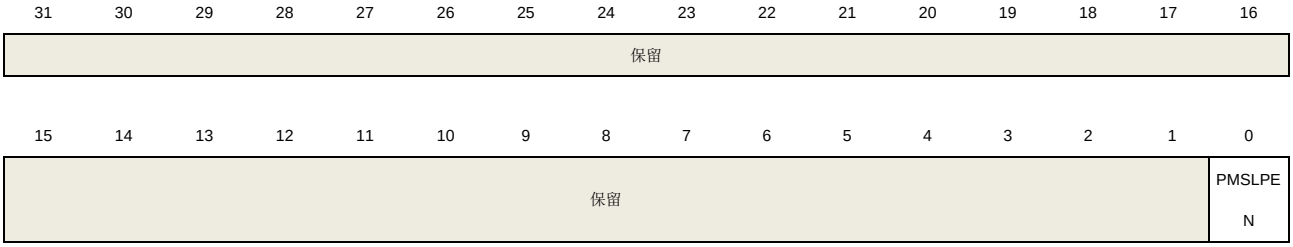
		1：1 表示非激活状态
23:22	保留	必须保持复位值。
21	ECATCLKDIS	EtherCAT 内核时钟失能位 如果 ECATCLKDIS 被置位，EtherCAT 内核时钟将被禁用。要设置这个位需要连续写入一两次。写入 0 将清除计数。 0：启用 EtherCAT 内核时钟 1：禁用 EtherCAT 内核时钟
20:18	保留	必须保持复位值。
17	EDWOLBSTAT	能量检测/ WoL 端口 B 状态位 要清除这个位，首先需要清除 PHY 上的事件。 0：端口 B PHY 未发生能量检测/ WoL 事件 1：端口 B PHY 发生能量检测/ WoL 事件
16	EDWOLASTAT	能量检测/ WoL 端口 A 状态位 要清除这个位，首先需要清除 PHY 上的事件。 0：端口 A PHY 未发生能量检测/ WoL 事件 1：端口 A PHY 发生能量检测/ WoL 事件
15	EDWOLBEN	能量检测/ WoL 端口 B 使能位 当能量检测/ WoL 事件发生在端口 B PHY 上并且该位被置位时，INTC_FLAG 寄存器中的 PMEIF 位将被设置。 0：关闭能量检测/ WoL 端口 B 1：使能能量检测/ WoL 端口 B
14	EDWOLAEN	能量检测/ WoL 端口 A 使能位 当能量检测/ WoL 事件发生在端口 A PHY 上并且该位被置位时，INTC_FLAG 寄存器中的 PMEIF 位将被设置。 0：关闭能量检测/ WoL 端口 A 1：使能能量检测/ WoL 端口 A
13:1	保留	必须保持复位值。
0	RDY	设备就绪位 这个位指示设备是否就绪。主机可以在上电、EtherCAT 设备复位、模块复位、数字复位或退出节能模式后读取这个位来获取设备的就绪状态。 0：设备未就绪 1：设备已就绪 **注意：** ■ 该位的上升沿将设置 INTC_FLAG 寄存器中的 READYIF 位，并可以触发一个中断； ■ 当这个位被清除时，除了 PMU_CTL、PMU_PDIVAL 和 RCU_RSTCFG 寄存器之外，对任何内部资源的读取访问是被禁止的； ■ 在该位被置位之前，对任何地址的写操作都是无效的。

控制寄存器 1（PMU_CTL1）

地址偏移：0x04

复位值：0x0000 0000

该寄存器可以按字（32 位）访问。



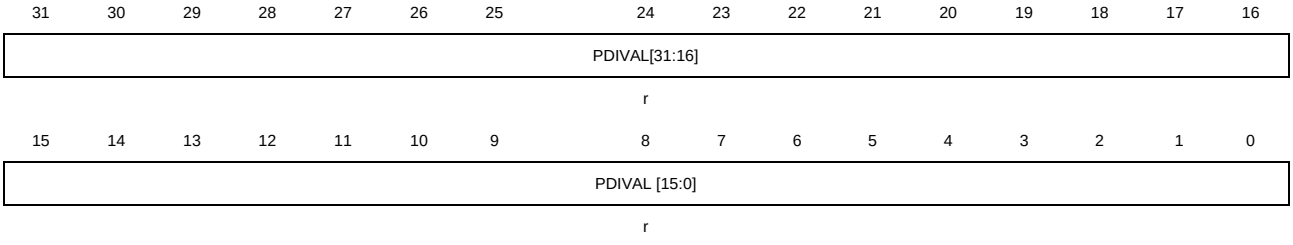
位/位域	名称	描述
31:1	保留	必须保持复位值。
0	PMSLPEN	电源管理睡眠模式使能位 当 PMSLPEN 置位时，设备将进入由 PMMODCFG 位配置的电源管理模式。 0：禁用电源管理睡眠模式 1：启用电源管理睡眠模式 **注意：** ■ 当设备唤醒时，该位将被清除； ■ 当 PMMODCFG 为 0b00 时，不应设置这个位，尽管硬件不会阻止写操作。

数据接口参考值寄存器（PMU_PDIREFVAL）

地址偏移：0x1C

复位值：0x7654 3210

该寄存器可以按字（32 位）访问。



位/位域	名称	描述
31:0	PDIVAL[31:0]	当 PDI 就绪时，读取这个寄存器将返回复位值，否则，将返回其他无效值（非复位值）。用于 PDI 接口测试。

50.3. 复位和时钟单元（RCU）

50.3.1. 复位控制单元

简介

ESC复位控制包括两种控制方式：系统复位和模块复位。系统复位包括上电复位（POR）、外部引脚复位（RSTN）和EtherCAT系统复位，可以复位设备中的所有电路。模块复位包括数字复位、PHY复位和EtherCAT内核复位，可以复位相应的模块。

主要特征

- 系统复位，复位设备中的所有电路。
- 多模块复位，复位PHY之外的数字电路。
- 单模块复位，复位以太网内核和PHY。

功能说明

系统复位

系统复位可以复位整个设备，包括上电复位（POR）、外部引脚复位（RSTN）和以太网系统级复位，具体描述如下：

上电复位：当设备刚上电或者电源断开后重上电时，发生上电复位。

RSTN引脚复位：将RSTN输入引脚驱动为低电平可启动外部引脚复位。

EtherCAT系统复位：EtherCAT系统复位是由三个连续的特殊帧/命令序列启动。

模块复位

模块复位会影响一个或多个模块，并可以为各模块生成复位信号，描述如下：

多模块复位：通过设置配置寄存器（RCU_RSTCFG）的DRST位来执行数字复位。数字复位会复位除PHY之外的所有设备子模块。

单模块复位：单模块复位仅复位指定的模块。单模块复位不会锁定配置引脚，包括端口A的PHY复位、端口B的PHY复位和EtherCAT控制器复位。单模块复位描述如下：

可通过置位复位配置寄存器（RCU_RSTCFG）中的PHYARST位或置位PHY控制寄存器（PHY_MII_CTL）中的MR_MAIN_REST位实现端口A的PHY复位。端口A的PHY复位后，PHYARST位和软复位Bit会自动清除。该复位不会影响设备的其他模块。可以通过复位配置寄存器（RCU_RSTCFG）中的PHYARST位或PHY控制寄存器（PHY_MII_CTL）中的MR_MAIN_REST位是否清除来检查端口A的PHY复位是否完成。

可通过置位复位配置寄存器（RCU_RSTCFG）中的PHYBRST位或置位PHY控制寄存器（PHY_MII_CTL）中的MR_MAIN_REST位实现端口B的PHY复位。端口B的PHY复位后，PHYBRST位和软复位Bit会自动清除。该复位不会影响设备的其他模块。可以通过复位配置寄存器（RCU_RSTCFG）中的PHYBRST位或PHY控制寄存器（PHY_MII_CTL）中的

MR_MAIN_REST位是否清除来检查端口B的PHY复位是否完成。

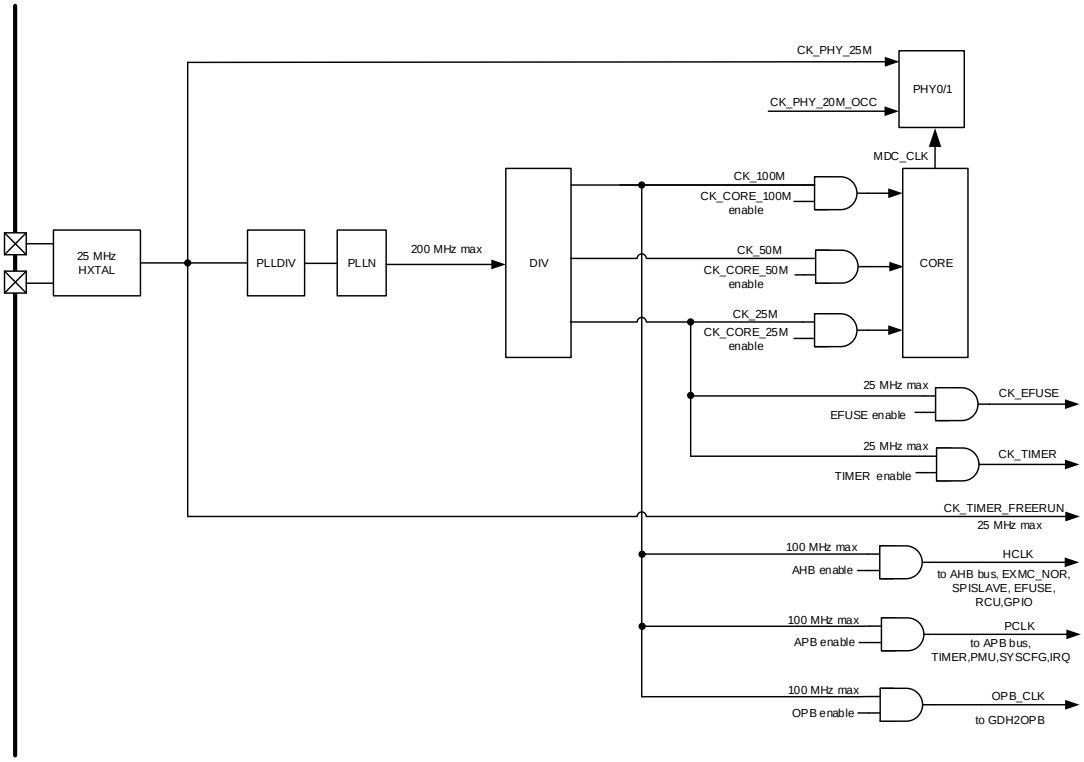
可以通过清除配置寄存器（RCU_RSTCFG）中的ESCRST位实现EtherCAT控制器的单独复位，这将复位EtherCAT内核及其寄存器。

50.3.2. 时钟控制单元（CCTL）

简介

EtherCAT时钟控制单元主要由外部高速晶体振荡器（HXTAL）和锁相环（PLL）组成。时钟通常由25MHz无源晶体振荡器的OSCIN和OSCOUT提供，或者由单端25MHz时钟源驱动器的OSCIN引脚提供。

图 50-3. 时钟树



主要特征

- 25 MHz高速晶振振荡器（HXTAL）。
- 锁相环（PLL）

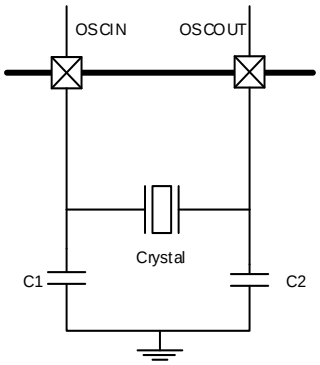
功能说明

高速晶体振荡器（HXTAL）

系统需要一个固定频率为25 MHz的时钟源，供内部时钟振荡器和锁相环使用。通常是通过将25 MHz晶体振荡器连接到芯片的OSCIN和OSCOUT引脚来提供。如果选择使用单端25 MHz时钟源，也可以通过驱动OSCIN输入引脚来提供这个时钟。如果选择了单端时钟源，为了设备正常工作，时钟输入必须持续运行。省电模式允许振荡器或外部时钟输入暂

停。

图 50-4. HXTAL 时钟源



时钟配置寄存器（RCU_CLKCFG）中的HXTALSTB位指示高速外部晶振是否稳定。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。此时，HXTAL时钟可以被直接用作系统时钟源或者PLL输入时钟。

锁相环

PLL输入是一个25MHz的HXTAL时钟，经过PLLN进行8倍频后，得到CK_PLL（200MHz）。CK_PLL时钟通过PLLDIV进行2/4/8分频，得到100MHz、50MHz、25MHz时钟用于EtherCAT内核且相关的时钟可以在内核使能寄存器（RCU_COREEN）中打开。相应的模块时钟可以在AHB使能寄存器（RCU_AHBEN）、APB使能寄存器（RCU_APBEN）和内核使能寄存器（RCU_COREEN）中打开。

50.3.3. RCU 寄存器

RCU基地址：0x3400

AHB 使能寄存器（RCU_AHBEN）

地址偏移：0x00
复位值：0x0000 000F

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														GPIOEN	OPBEN
														rw	rw

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

1：开启 GPIO 时钟

0 OPBEN OPB 时钟使能
由软件置位或复位
0：关闭 OPB 时钟
1：开启 OPB 时钟

APB 使能寄存器（RCU_APBEN）

地址偏移：0x04
复位值：0x0000 0077

该寄存器可以按字（32位）访问。



位/位域	名称	描述
31:7	保留	必须保持复位值。
6	TIMERFUNEN	TIMER 功能时钟使能 由软件置位或复位 0：关闭 TIMER 功能时钟 1：开启 TIMER 功能时钟
5	TIMEREN	TIMER 时钟使能 由软件置位或复位 0：关闭 TIMER 时钟 1：开启 TIMER 时钟（CK_TIMER）
4	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
3	保留	必须保持复位值。
2	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
1	IRQEN	IRQ 时钟使能 由软件置位或复位 0：关闭 IRQ 时钟

1：开启 IRQ 时钟

0	保留	必须保持复位值。

内核使能寄存器（RCU_COREEN）

地址偏移：0x08
复位值：0x0000 0007

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													CORE100MEN	CORE50MEN	CORE25MEN
													rw	rw	rw

位/位域	名称	描述
31:3	保留	必须保持复位值。
2	CORE100MEN	EtherCAT 内核 100M 时钟使能 由软件置位或复位 0：关闭 CK_CORE_100M 时钟 1：开启 CK_CORE_100M 时钟
1	CORE50MEN	EtherCAT 内核 50M 时钟使能 由软件置位或复位 0：关闭 CK_CORE_50M 时钟 1：开启 CK_CORE_50M 时钟
0	CORE25MEN	EtherCAT 内核 25M 时钟使能 由软件置位或复位 0：关闭 CK_CORE_25M 时钟 1：开启 CK_CORE_25M 时钟

时钟配置寄存器（RCU_CLKCFG）

地址偏移：0x0C
复位值：0x0528 0400

注意：PLLBWCTL、PLLN和PLLDIV只能在PLL_CFG_KEY = 1时读取和写入。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				PLLBWCTL[3:0]				保留		PLLN[5:0]					
				rw						rw					

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

保留	PLLDIV[2:0]	保留	PLLSTB	保留	HXTALSTB	保留
	rw		r		r	

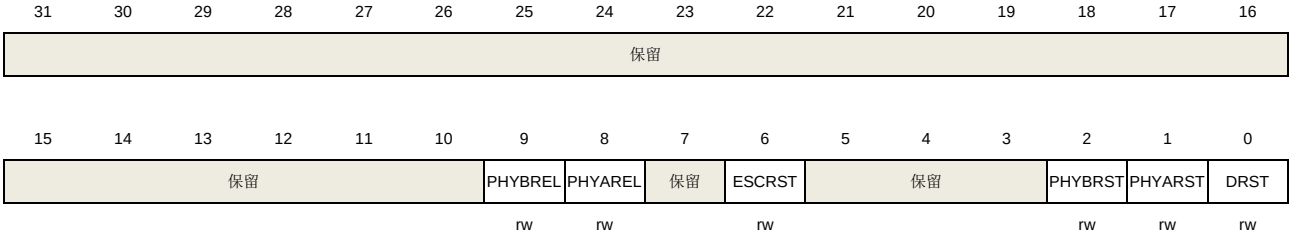
位/位域	名称	描述
31:28	保留	必须保持复位值。
27:24	PLLBWCTL[3:0]	PLL 带宽控制信号
23:22	保留	必须保持复位值。
21:16	PLLN[5:0]	PLL时钟倍频因子 由软件置位或复位 000000：保留 000001：保留 … 000110：保留 000111：保留 001000：倍频因子为8 001001：倍频因子为9 001010：倍频因子为10 … 111110：倍频因子为62 111111：倍频因子为63
15:11	保留	必须保持复位值。
10:8	PLLDIV[2:0]	PLL时钟频率分频因子 由软件置位或复位 000：分频因子为1 001：分频因子为2 … 110：分频因子为7 111：分频因子为8
7:6	保留	必须保持复位值。
5	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
4	保留	必须保持复位值。
3	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

2:0	保留	必须保持复位值。

复位配置寄存器（RCU_RSTCFG）

地址偏移：0x10
复位值：0x0000 0000

该寄存器可以按字（32位）访问。



位/位域	名称	描述
31:10	保留	必须保持复位值。
9	PHYBREL	端口B的PHY释放 0：端口B的PHY保持复位状态 1：端口B的PHY从复位状态释放 **注意：**当phyrst_mode = 1时，该位有效
8	PHYAREL	端口A的PHY释放 0：端口A的PHY保持复位状态 1： 端口A的PHY从复位状态释放 **注意：**当phyrst_mode = 1时，该位有效
7	保留	必须保持复位值。
6	ESCRST	EtherCAT复位 将该位置1将复位EtherCAT内核。当EtherCAT内核从复位状态释放时，该位由硬件自动清零。当该位被置位时，所有写入该位的操作都会被忽略。
5:3	保留	必须保持复位值。
2	PHYBRST	端口B复位 将此位置1将复位端口B PHY。当端口B PHY从复位状态释放时，该位由硬件自动清零。当该位被置位时，所有写入该位的操作都会被忽略。
1	PHYARST	端口 A PHY 复位 将此位置 1 将复位端口 A PHY。当端口 A PHY 从复位状态释放时，该位由硬件自动清零。当该位被置位时，所有写入该位的操作都会被忽略。
0	DRST	数字复位 将此位置1将复位整个芯片（除了锁相环，端口B PHY和端口A PHY）。当芯片从复位状态被释放时，该位被硬件自动清零。当该位被置位时，所有写入该位的操作都会被忽略。

PLL 配置密钥寄存器（RCU_PLL_CFG_KEY）

地址偏移：0x14
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															PLL_CFG_KEY
															rw

位/位域	名称	描述
31:1	保留	必须保持复位值。
0	PLL_CFG_KEY	当寄存器写入 0x78b465a1 时，该位为 1 并且可以读写 RCU_CLKCFG 寄存器中的 PLLBWCTL、PLLN 和 PLLDIV 位。

复位标志寄存器（RCU_PRSTF）

地址偏移：0x18
复位值：0x0000 0002

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														PRSTF	PRSTC
														r	rw

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	PRSTF	Pin 复位标志 该位在引脚复位时置位，通过置位 PRSTC 清零。 0：未发生 Pin 复位 1：发生 Pin 复位
0	PRSTC	Pin 复位标志清除 写 1 清除 PRSTF 标志

50.4. 中断控制器（INTC）

50.4.1. 简介

器件提供了可编程的多层中断结构，通过中断控制器（INTC）进行控制。中断事件是由各个子模块在内部生成，经过配置可由IRQ引脚输出单个外部主机中断。

50.4.2. 主要特征

- IRQ中断缓冲器模式、极性和置为无效间隔时间皆可配置；
- IRQ中断可配置为开漏输出，实现多个器件共用中断；
- 所有内部中断均可屏蔽且能够触发IRQ中断；
- 器件支持生成以下8种类型中断：
  - 软件中断；
  - 器件就绪中断；
  - 以太网PHY中断；
  - 定时器中断；
  - PME中断；
  - AHB2OPB桥接中断；
  - EtherCAT中断；
  - 时钟输出测试模式。

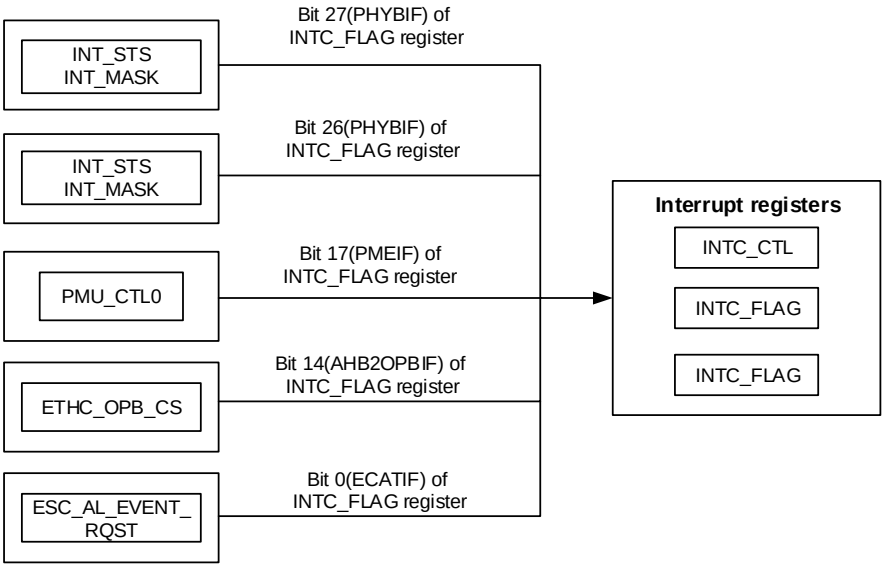
50.4.3. 功能说明

器件的中断按照中断源是否在子模块的寄存器中进行使能和清除操作，可划分为以下两种类型：

- 第一种类型包含软件、器件就绪和定时器中断，通过INTC_FLAG寄存器和INTC_EN寄存器对中断直接进行访问与配置（含监控、使能 / 禁止和清除）；
- 第二种类型包含以太网PHY、电源管理、AHB2OPB桥接和EtherCAT中断，INTC_FLAG寄存器可以提供中断事件指示，但没有具体的中断源信息，需要软件轮询子模块的中断寄存器才能确定中断源。只有在处理了中断并将中断源清除后，才能清除INTC_FLAG寄存器对应中断。

中断事件能够触发外部IRQ中断引脚输出，通过配置INTC_CTL寄存器可以实现使能 / 禁止IRQ中断引脚输出以及配置IRQ中断缓冲器模式、极性和置为无效间隔时间。INTC_CTL寄存器的DEAS位域用于配置中断请求无效间隔时间，其确保了最小的IRQ输出中断置为无效间隔周期，且置为无效间隔始终从IRQ引脚置为无效时开始。中断寄存器和中断源控制寄存器之间的关系，如*图50-5. 中断框图*所示。

图 50-5. 中断框图



软件中断

中断控制器提供了对通用的软件中断的控制。当INTC_EN寄存器的SWIE位从0切换为1时，INTC_FLAG寄存器的SWIF位会被置位。此中断提供了一种较为简单的软件产生中断的方法，用于常规软件设计。

为了使软件中断事件能够触发外部IRQ中断引脚，必须置位INTC_CTL寄存器的IRQEN位使能IRQ输出。

设备准备就绪中断

中断控制器提供了对器件就绪中断的控制。当INTC_EN寄存器的READYIE位从0切换为1时，INTC_FLAG寄存器的READYIF位用于指示器件已准备就绪，并在上电或复位后可接受访问。

为了使器件就绪中断事件能够触发外部IRQ中断引脚，必须置位INTC_CTL寄存器的IRQEN位使能IRQ输出。

以太网 PHY 中断

中断控制器提供了对以太网PHY中断的控制。当INTC_EN寄存器的PHYAIE位从0切换为1时，INTC_FLAG寄存器的PHYAIF和PHYBIF位用于指示来自以太网PHY的中断事件。关于以太网PHY中断源的详细信息，请参考*以太网PHYS*。

为了使以太网PHY中断事件能够触发外部IRQ中断引脚，必须置位INTC_CTL寄存器的IRQEN位使能IRQ输出。

定时器中断

中断控制器提供了对定时器中断的控制。当定时器的计数寄存器值从0变为0xFFFF时，将产生此中断。当INTC_EN寄存器的TIMIE位从0切换为1时，INTC_FLAG寄存器的TIMIF位用于指示来自定时器的中断事件。

为了使定时器中断事件能够触发外部IRQ中断引脚，必须置位INTC_CTL寄存器的IRQEN位使

能IRQ输出。

PME 中断

中断控制器提供了对功耗管理中断的控制。当INTC_EN寄存器的PMEIE位从0切换为1时，INTC_FLAG寄存器的PMEIF位用于指示来自PMU的中断事件。关于电源管理中断源的详细信息，请参考*电源管理单元（PMU）*。

为了使功耗管理中断事件能够触发外部IRQ中断引脚，必须置位INTC_CTL寄存器的IRQEN位使能IRQ输出。

AHB2OPB 桥中断

中断控制器提供了对AHB2OPB桥接中断的控制。当INTC_EN寄存器的AHB2OPBIE位从0切换为1时，INTC_FLAG寄存器的AHB2OPBIF位用于指示来自BUS的AHB2OPB桥接中断事件。关于AHB2OPB桥接中断源的详细信息，请参考*系统和总线架构*。

为了使AHB2OPB桥接中断事件能够触发外部IRQ中断引脚，必须置位INTC_CTL寄存器的IRQEN位使能IRQ输出。

EtherCAT 中断

中断控制器提供了对EtherCAT中断的控制。当INTC_EN寄存器的ECATIE位从0切换为1时，INTC_FLAG寄存器的ECATIF位用于指示EtherCAT中断事件。关于EtherCAT中断源的详细信息，请参考*EtherCAT*。

为了使EtherCAT中断事件能够触发外部IRQ中断引脚，必须置位INTC_CTL寄存器的IRQEN位使能IRQ输出。

时钟输出测试模式

为了进行系统调试，观察时钟情况，可通过设置INTC_CTL寄存器的IRQCKOUT位为1，实现IRQ引脚输出晶振时钟。此时，IRQ引脚须配置为推挽输出模式（IRQMODE=1），以求达到最佳效果。

50.4.4. INTC 寄存器

INTC 基地址：0x0000 3A00

控制寄存器（INTC_CTL）

地址偏移：0x00
复位值：0x0000 0000

这个寄存器需要以字（32 位）为单位进行访问。



位/位域	名称	描述
31:24	DEAS[7:0]	中断无效间隔时间 该位域用于配置中断请求无效间隔时间，单位为 10us。 该位域配置为 0 将导致器件禁用 DEAS 时间间隔、复位间隔计数器并发送出任何挂起的中断。该位域配置为非 0 值，则后续产生的中断都将遵循该设置值。
23:15	保留	必须保持复位值。
14	DEASC	中断无效间隔时间清除 该位域配置为 1，将会清零无效间隔计数器，同时会开启新的无效间隔计数（无论当前是否处于激活的无效间隔时间）。该位置位后，将由硬件自动清零。当器件产生软件复位时，该位不会被复位。 0：无影响 1：清零无效间隔计数器
13	DEASSTAT	中断无效间隔状态 该位用于指示中断无效间隔的状态。 0：不处于无效间隔时间（中断会发送至 IRQ 引脚） 1：处于无效间隔时间（中断不会发送至 IRQ 引脚）
12	IRQSTAT	内部 IRQ 线状态 该位用于指示内部 IRQ 线的状态，其不受 IRQEN 位配置影响。 0：无使能的中断处于激活状态 1：存在使能的中断处于激活状态
11:9	保留	必须保持复位值。
8	IRQEN	IRQ 引脚输出使能 该位控制 IRQ 引脚的中断输出功能。 0：禁止 IRQ 引脚输出

		1：使能 IRQ 引脚输出
7:5	保留	必须保持复位值。
4	IRQPOL	IRQ 输出极性 当器件产生软件复位时，该位不会被复位。当设备发生系统复位(如上电复位、引脚复位和 EtherCAT 系统复位)时，此位复位。 0：IRQ 有效输出为低电平 1：IRQ 有效输出为高电平 **注意：**当 IRQ 配置为开漏输出（IRQMODE=0）时，该位被忽略且 IRQ 有效输出始终为低电平。
3:2	保留	必须保持复位值。
1	IRQCKOUT	IRQ 时钟输出 0：无时钟输出 1：IRQ 引脚输出晶振时钟（用于系统调试，观察时钟） **注意：**当该位配置为 1 时，IRQ 引脚须配置为推挽输出模式（IRQMODE=1）。
0	IRQMODE	IRQ 引脚输出模式 当器件产生软件复位时，该位不会被复位。当设备发生系统复位(如上电复位、引脚复位和 EtherCAT 系统复位)时，此位复位。 0：输出开漏模式 1：输出推挽模式

标志寄存器（INTC_FLAG）

地址偏移：0x04
复位值：0x0000 0000

这个寄存器必须以字（32 位）为单位进行访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWIF	READYIF	保留		PHYBIF	PHYAIF	保留						TIMIF	保留	PMEIF	保留
rc_w1	rc_w1			r	r							rc_w1		rc_w1	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	AHB2OPBIF	保留													ECATIF
	r														r

位/位域	名称	描述
31	SWIF	软件中断标志 当 INTC_EN 寄存器的 SWIE 位为 1 时，将产生此中断。软件写 1 清 0。
30	READYIF	器件就绪中断标志 此中断用于指示器件已准备就绪，并在上电或复位后接受访问。软件写 1 清 0。
29:28	保留	必须保持复位值。
27	PHYBIF	PHY B 中断标志

		该位用于指示来自 PHY B 的中断事件。
26	PHYAIF	PHY A 中断标志 该位用于指示来自 PHY A 的中断事件。
25:20	保留	必须保持复位值。
19	TIMIF	定时器中断标志 当定时器的计数寄存器从 0 变为 0xFFFF 时，将产生此中断。软件写 1 清 0。
18	保留	必须保持复位值。
17	PMEIF	PME 中断标志 当电源管理单元检测到 PMU_CTL 寄存器中配置的功耗管理事件时，将产生此中断。 软件写 1 清 0。 **注意：**配置中断请求无效间隔不适用于 PME 中断。
16:15	保留	必须保持复位值。
14	AHB2OPBIF	AHB2OPB 桥接中断标志 该位用于指示来自 BUS 的 AHB2OPB 桥接中断事件。
13:1	保留	必须保持复位值。
0	ECATIF	EtherCAT 中断标志 该位用于指示 EtherCAT 中断事件。

使能寄存器（INTC_EN）

地址偏移：0x08
复位值：0x0000 0000

这个寄存器必须通过字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWIE	READYIE	保留		PHYBIE	PHYAIE	保留						TIMIE	保留	PMEIE	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	AHB2OPBIE	保留													ECATIE
	rw														rw

位/位域	名称	描述
31	SWIE	软件中断使能 0：禁止软件中断 1：使能软件中断
30	READYIE	器件就绪中断使能 0：禁止器件就绪中断 1：使能器件就绪中断

29:28	保留	必须保持复位值。
27	PHYBIE	PHY B 中断使能 0：禁止以太网 PHY B 中断 1：使能以太网PHY B中断
26	PHYAIE	PHY A 中断使能 0：禁止以太网 PHY A 中断 1：使能以太网PHY A中断
25:20	保留	必须保持复位值。
19	TIMIE	定时器中断使能 0：禁止定时器中断 1：使能定时器中断
18	保留	必须保持复位值。
17	PMEIE	PME 中断使能 0：禁止功耗管理中断 1：使能功耗管理中断
16:15	保留	必须保持复位值。
14	AHB2OPBIE	AHB2OPB 桥接中断使能 0：禁止 AHB2OPB 桥接中断 1：使能AHB2OPB桥接中断
13:1	保留	必须保持复位值。
0	ECATIE	EtherCAT 中断使能 0：禁止 EtherCAT 中断 1：使能EtherCAT中断

50.5. 通用输入/输出（GPIO）

50.5.1. 概述

最多支持 35 个通用输入 / 输出引脚（GPIO）的模式配置。每个 GPIO 端口将根据芯片的当前工作模式确定该端口的当前功能，包括输入 / 输出模式。每个 GPIO 引脚可以配置为上拉 / 下拉或浮空。当引脚处于输出模式时，引脚可以配置为推挽/漏极开路/源开路输出。

50.5.2. 特征

- 每个引脚具有弱上拉 / 下拉功能；
- 推挽 / 漏极开漏 / 源极开漏输出使能控制；
- 根据芯片模式配置选定引脚的功能；

50.5.3. 功能概述

GPIO 引脚配置

在复位期间，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，等待 EEPROM 加载，加载完成后。根据 ESC PDI_TYPE 决定芯片复位后引脚的初始状态，当 PDI_TYPE 等于 0x80 时，选择 SPI 模式。
GPIO 管脚依据工作状态控制为输入或输出状态。
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 引脚配置为输出引脚，用户可以配置输出驱动模式：推挽或漏极开漏以及源极开漏模式。上下拉模式以及输出模式配置支持通过 SPI 通信写入。

外部中断/事件线路

只支持一个外部中断输出接口。中断输出配置由*中断控制器（INTC）*的内部寄存器决定，输出模式也由内部 Bit 位决定。

备用功能

当芯片处于不同模式时，每个引脚具有不同的功能。

SPI 模式：当 PDI_TYPE 等于 0x80 且 MCU_PDI_TYPE 的引脚等于 0 时，AFIO 调整为 SPI 模式。

表 50-4. GPIO 配置表

模式名称	寄存器 / 信号	描述

spi8w_gpio	pdi_type / mcu_pdi_type / spi_ext_mode / sip_mode / line_mode	pdi_type == 0x80;
		mcu_pdi_type == 1'b0;
		sip_mode == 1'b0;
		line_mode == 2'b11;
		LINKACTLED0 引脚必须外部下拉
		LINKACTLED1 引脚必须外部下拉

模式名称	寄存器 / 信号	描述

spi8w_mii_down	pdi_type / mcu_pdi_type / spi_ext_mode / sip_mode / line_mode / chip_mode	pdi_type == 0x80;
		mcu_pdi_type == 1'b0;
		sip_mode == 1'b0;
		line_mode == 2'b10;
		chip_mode == 2'b10
		LINKACTLED1 引脚必须在外部下拉
		LINKACTLED1 引脚必须在外部上拉

spi8w_mii_up	pdi_type / mcu_pdi_type / spi_ext_mode / sip_mode / line_mode / chip_mode	pdi_type == 0x80;
		mcu_pdi_type == 1'b0;
		sip_mode == 1'b0;
		line_mode == 2'b10;
		chip_mode == 2'b11;
		LINKACTLED1 引脚必须在外部上拉
		LINKACTLED1 引脚必须在外部上拉

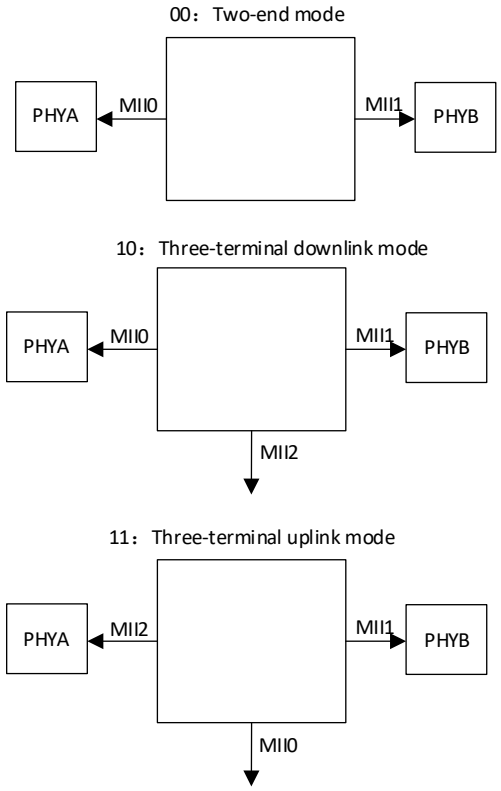
此外，当处于 SPI（2/4/8 线）+MII 模式且 chip_mode[1:0]不等于 0x11 时，EtherCAT 端口 0 连接到内部 PHY A。

当 chip_mode[1:0]等于 0x11 时，EtherCAT 端口 0 连接到 MII 引脚，端口 2 连接到内部 PHY A。

当 chip_mode[1:0]等于 0x10 时，EtherCAT 端口 2 连接到 MII 引脚。

当 chip_mode[1:0]等于 0x00 时，在这种情况下，SPI + GPIO 模式的输出不受影响，而 SPI + MII 模式中 MII 信号不输出。

图 50-6. 端口连接 PHYS



备注：

1. 部分引脚在上电复位期间或当 RST#设置为无效时被锁定，并在被锁定后自动切换。
2. MII_LINKPOL 信号在复位后被锁存，以确定 MII_LINK 引脚的极性。如果 MII_LINKPOL 等于 0，表示 MII_LINK 引脚低电平有效，表明已建立 100 Mbps 全双工链路。MII_LINKPOL 等于 1，表示 MII_LINK 引脚高电平有效，表明已建立 100 Mbps 全双工链路。
3. SYNC1_LATCH1/SYNC0_LATCH0 pad 的 omode / io_en 由 ESC 内部寄存器决定。
4. 如下锁存器信号必须在以下面模式中进行上下拉配置，并且不能配置为 X 状态。

   (1).如果 spi_ext_mode 等于 1 且 inphy_bypass 等于 0，则 chip_mode [1:0]必须设置为下拉状态。

   (2).IO16 引脚必须配置为上拉或下拉状态。

   (3).EESIZE 引脚必须配置为上拉或下拉状态。

   (4).IO17 引脚必须配置为下拉和下拉状态。

   (5).当 inphy_bypass 等于 1 时，MII_LINKPOL 引脚必须设置为上拉或下拉状态。

PDI_TYPE：参考以 EtherCAT 寄存器 ***ESC PDI 控制寄存器（ESC_PDI_CTL）***。

line_mode：SPI 输出到 GPIO 是由 SPI 输入指令决定的。

spi_ext_mode：参考***系统配置寄存器 0（SYSCFG_CFG0）***Bit 2。

chip_mode [1:0]：复位后，LINKACTLED1/LINKACTLED0 的引脚被锁定。

inphy_bypass：由工厂设置的寄存器位被配置为 1，不能修改。

模拟配置

当 GPIO 引脚用于模拟配置时：

- 弱上拉和下拉电阻禁用；
- 输出缓冲器禁用；
- 施密特触发器输入禁用；
- 读取端口输入状态寄存器“0”。

*图 50-7. 模拟配置的基本结构*是 I / O 端口的模拟模式配置。

图 50-7. 模拟配置的基本结构



备用功能(AF)配置

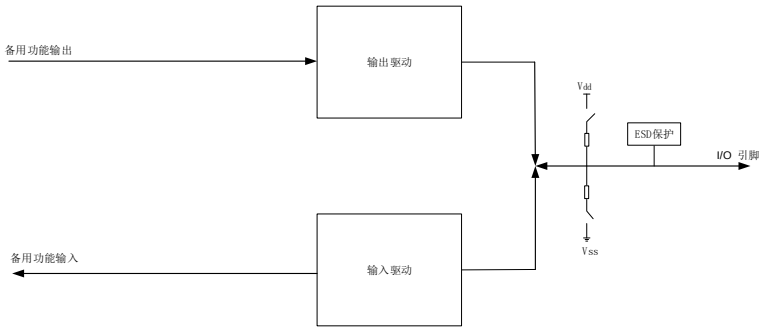
为了适应不同的器件封装，GPIO 端口支持软件配置将一些备用功能应用到其他引脚上。

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。

*图 50-8. 备用功能配置的基本结构*是 I / O 端口备用功能配置图。

图 50-8. 备用功能配置的基本结构



备注：

OSPI 模式下，由于 SPI PIN 脚占用数量较多，在 OSPI+GPIO 模式下不能使用 pdi_gpio15；OSPI+MII 模式下不能使用 ESC_CLKOUT_25M。

50.5.4. 寄存器定义

GPIO 基地址：0x3500

端口输出模式寄存器 0（GPIO0_OMODE0）

地址偏移：0x00
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
OM015[1:0]	OM014[1:0]	OM013[1:0]	OM012[1:0]	OM011[1:0]	OM010[1:0]	保留	OM008[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
OM007[1:0]	OM006[1:0]	OM005[1:0]	OM004[1:0]	保留	OM002[1:0]	OM001[1:0]	OM000[1:0]
rw	rw	rw	rw		rw	rw	rw

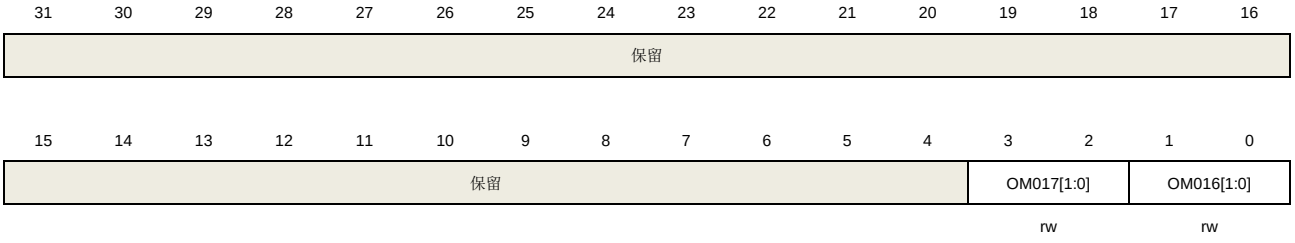
位/位域	名称	描述
31:30	OM015[1:0]	Pin IO11输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
29:28	OM014[1:0]	Pin OE_EXT输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
27:26	OM013[1:0]	Pin IO4输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
25:24	OM012[1:0]	Pin IO5输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
23:22	OM011[1:0]	Pin IO6输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
21:20	OM010[1:0]	Pin LATCH_IN输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
19:18	保留	必须保持复位值。
17:16	OM008[1:0]	Pin WD_STATE输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
15:14	OM007[1:0]	Pin IO7输出模式位

		该位由软件置位和清除。 参考 OM000[1:0]的描述
13:12	OM006[1:0]	Pin IO8输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
11:10	OM005[1:0]	Pin EOF输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
9:8	OM004[1:0]	Pin SOF输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
7:6	保留	必须保持复位值
5:4	OM002[1:0]	Pin IO18输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
3:2	OM001[1:0]	Pin IO17输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
1:0	OM000[1:0]	Pin IO16输出模式位 该位由软件置位和清除。 00: 输出推挽模式（复位值） 01:输出开漏模式 10:输出开源模式 11:保留

端口输出模式寄存器 1（GPIO0_OMODE1）

地址偏移：0x04
复位值：0x0000 0000

该寄存器可以按字（32位）访问。



位/位域	名称	描述
31:4	保留	必须保持复位值
3:2	OM017[1:0]	Pin IO13输出模式位

		该位由软件置位和清除。 参考 OM000[1:0]的描述
1:0	OM016[1:0]	Pin IO12输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述

端口输出模式寄存器 2（GPIO1_OMOD0）

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
OM115		OM114		OM113		保留		OM111		OM110		保留		OM108	
rw		rw		rw				rw		rw				rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OM107		OM106		OM105		保留		OM103		OM102		OM101		OM100	
rw		rw		rw				rw		rw		rw		rw	

位/位域	名称	描述
31:30	OM115[1:0]	Pin LINKACTLED0输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
29:28	OM114[1:0]	Pin LINKACTLED1输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
27:26	OM113[1:0]	Pin EESIZE输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
25:24	保留	必须保持复位值
23:22	OM111[1:0]	Pin EESCL输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
21:20	OM110[1:0]	Pin EESDA输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
19:18	保留	必须保持复位值
17:16	OM108[1:0]	Pin IO2输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述

15:14	OM107[1:0]	Pin IO1输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
13:12	OM106[1:0]	Pin IO0输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
11:10	OM105[1:0]	Pin WD_TRIG输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
9:8	保留	必须保持复位值
7:6	OM103[1:0]	Pin IO9输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
5:4	OM102[1:0]	Pin IO15输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
3:2	OM101[1:0]	Pin IO14输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
1:0	OM100[1:0]	Pin IO10输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述

端口输出模式寄存器 3（GPIO1_OMOD1）

地址偏移：0x0C
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												OM117		OM116	
												rw		rw	

位/位域	名称	描述
31:4	保留	必须保持复位值
3:2	OM117[1:0]	Pin OUTVALID输出模式位 该位由软件置位和清除。

参考 OM000[1:0]的描述

1:0	OM116[1:0]	Pin IO3输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述

端口上拉/下拉寄存器 0（GPIO0_PUD0）

地址偏移：0x10
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
PUD015[1:0]		PUD014[1:0]		PUD013[1:0]		PUD012[1:0]		PUD011[1:0]		PUD010[1:0]		PUD009[1:0]		PUD008[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
PUD007[1:0]		PUD006[1:0]		PUD005[1:0]		PUD004[1:0]		保留		PUD002[1:0]		PUD001[1:0]		PUD000[1:0]	
rw		rw		rw		rw				rw		rw		rw	

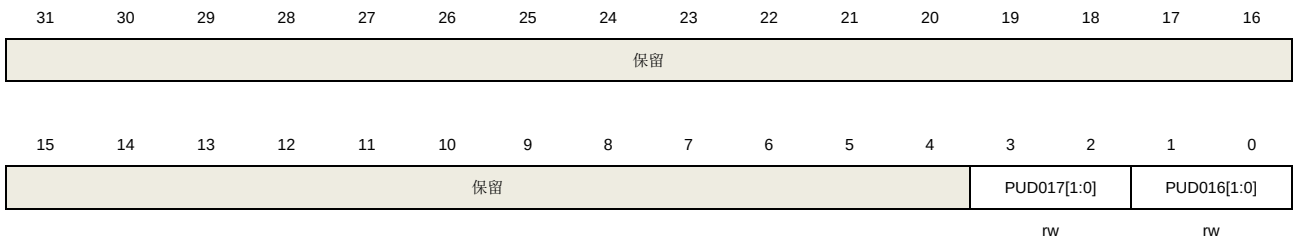
位/位域	名称	描述
31:30	PUD015[1:0]	Pin IO11上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
29:28	PUD014[1:0]	Pin OE_EXT上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
27:26	PUD013[1:0]	Pin IO4上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
25:24	PUD012[1:0]	Pin IO5上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
23:22	PUD011[1:0]	Pin IO6上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
21:20	PUD010[1:0]	Pin LATCH_IN上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
19:18	PUD009[1:0]	Pin SYNC1_LATCH1上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
17:16	PUD008[1:0]	Pin WD_STATE上拉或下拉位

		该位由软件置位和清除。 参照 PUD000[1:0]的描述
15:14	PUD007[1:0]	Pin IO7上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
13:12	PUD006[1:0]	Pin IO8上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
11:10	PUD005[1:0]	Pin EOF上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
9:8	PUD004[1:0]	Pin SOF上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
7:6	保留	必须保持复位值
5:4	PUD002[1:0]	Pin IO18上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
3:2	PUD001[1:0]	Pin IO17上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
1:0	PUD000[1:0]	Pin IO16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：模拟模式

端口上拉/下拉寄存器 1（GPIO0_PUD1）

地址偏移：0x14
复位值：0x0000 0000

该寄存器可以按字（32位）访问。



位/位域	名称	描述
31:4	保留	必须保持复位值
3:2	PUD017[1:0]	Pin IO13上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
1:0	PUD016[1:0]	Pin IO12上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述

端口上拉/下拉寄存器 2（GPIO1_PUD0）

地址偏移：0x18
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
PUD115[1:0]	PUD114[1:0]	PUD113[1:0]	PUD112[1:0]	PUD111[1:0]	PUD110[1:0]	PUD109[1:0]	PUD108[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
PUD107[1:0]	PUD106[1:0]	PUD105[1:0]	PUD104[1:0]	PUD103[1:0]	PUD102[1:0]	PUD101[1:0]	PUD100[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	PUD115[1:0]	Pin LINKACTLED0上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
29:28	PUD114[1:0]	Pin LINKACTLED1上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
27:26	PUD113[1:0]	Pin EESIZE上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
25:24	PUD112[1:0]	Pin IRQ上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
23:22	PUD111[1:0]	Pin EESCL上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
21:20	PUD110[1:0]	Pin EESDA上拉或下拉位 该位由软件置位和清除。

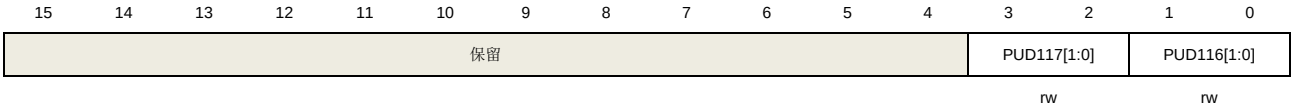
		参照 PUD000[1:0]的描述
19:18	PUD109[1:0]	Pin TESTMODE上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
17:16	PUD108[1:0]	Pin IO2上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
15:14	PUD107[1:0]	Pin IO1上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
13:12	PUD106[1:0]	Pin IO0上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
11:10	PUD105[1:0]	Pin WD_TRIG上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
9:8	PUD104[1:0]	Pin SYNC0_LATCH0上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
7:6	PUD103[1:0]	Pin IO9上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
5:4	PUD102[1:0]	Pin IO15上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
3:2	PUD101[1:0]	Pin IO14上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
1:0	PUD100[1:0]	Pin IO10上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述

端口上拉/下拉寄存器 3（GPIO1_PUD1）

地址偏移：0x1C
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															

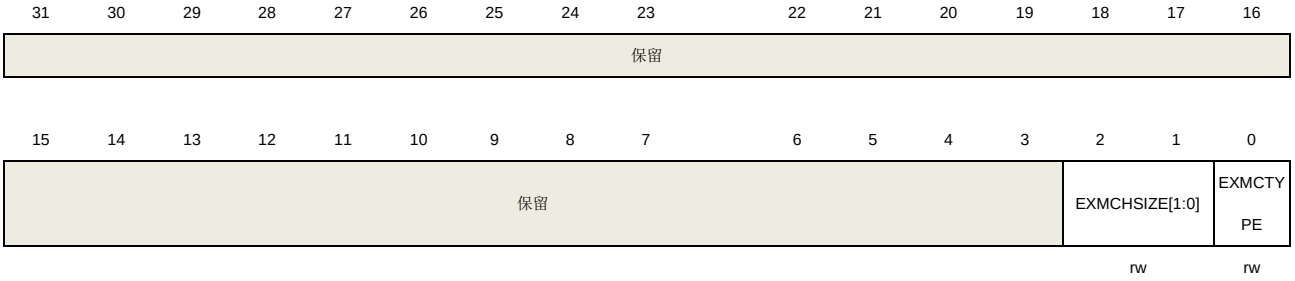


位/位域	名称	描述
31:4	保留	必须保持复位值
3:2	PUD117[1:0]	Pin OUTVALID上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
1:0	PUD116[1:0]	Pin IO3上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述

EXMC 控制寄存器（EXMC_CTL）

地址偏移：0x20
复位值：0x0000 0004

该寄存器可以按字（32位）访问。



位/位域	名称	描述
31:3	保留	必须保持复位值
2:1	EXMCHSIZE[1:0]	EXMC hsize 00: 8-bit 01:16-bit 10:32-bit 11: 保留
0	EXMCTYPE	EXMC 类型 0: 8-bit EXMC 1: 16-bit EXMC

50.6. 定时器（TIMER）

50.6.1. 基本定时器

简介

基本定时器模块有一个 16 位计数器，可以用作无符号计数器。基本定时器可以配置为生成中断。基本定时器的分辨率为 100μs。

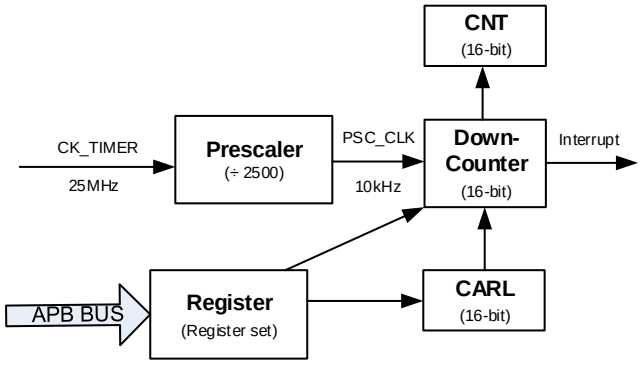
主要特征

- 计数器宽度：16 位；
- 计数时钟的来源仅为内部时钟；
- 计数模式：向下计数；
- 分辨率：100 微秒；
- 自动重装载功能；
- 中断输出：更新事件。

结构框图

*图50-9. 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 50-9. 基本定时器结构框图



功能说明

时钟源选择

基本定时器的时钟源只能是来自RCU内部25MHz的CK_TIMER。

CK_TIMER被进行2500分频生成10kHz的计数器时钟（PSC_CLK）。计数器的分辨率为100μs。

计数器向下计数模式

当TIMERx_CTL0寄存器中的CEN位置1时，基本定时器会将TIMER_CTL0寄存器中的预加载值（CPLOAD）加载到TIMER_CNT寄存器中。软件可以随时写入新的预加载值。如果CEN位置1，TIMER_CNT寄存器将立即被加载到新的预加载值，并从该值开始继续向下计数。

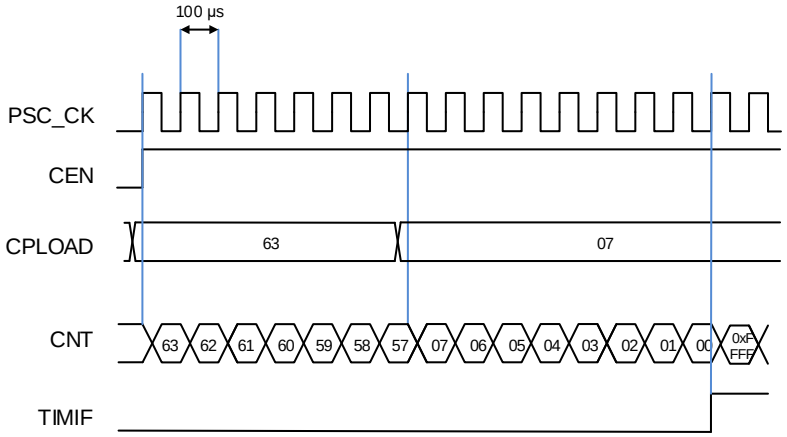
如果发生复位或TIMER_CTL0寄存器中的CEN位从1变为0，则预加载值将初始化为0xFFFF。

在复位时，计数值（CNT）将初始化为0xFFFF。

当TIMER_CTL0寄存器中的CEN位置1时，计数值主要会从预加载值持续递减到0。一旦计数器达到0，计数值会回绕到0xFFFF继续递减，并且INTC_FLAG寄存器中的TIMIF位置1。如果INTC_EN寄存器中的TIMIE位置1，基本定时器就会产生中断。如果INTC_FLAG寄存器中的TIMIF位置1，它只能通过写入1来清除该位。

*图50-10. 向下计数时序图*展示了当预加载值从0x63转换为0x07时计数器行为的一个示例。

图 50-10. 向下计数时序图



寄存器定义

基本定时器基地址：0x0000 3800

控制寄存器 0（TIMER_CTL0）

地址偏移：0x00
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
保留		CEN	保留												
		rw													

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

位/位域	名称	描述
31:30	保留	必须保持复位值
29	CEN	计数器使能 0: 计数器禁能 1: 计数器使能
28:16	保留	必须保持复位值

15:0	CPLOAD[15:0]	计数器预加载值 这些位定义了计数器的预加载值。 当CEN位置1，计数器从当前值开始向下计数。 **注意：**当CEN位从1到0，这些位将被转化为0xFFFF。

计数器寄存器（TIMER_CNT）

地址偏移：0x04
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
保留															

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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CNT[15:0]	这些位是当前的计数值。

50.6.2. 自由运行计数器

简介

自由运行计数器有一个32位的计数器，可以用作无符号计数器。计数器时钟为25MHz。

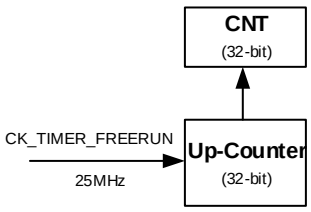
主要特征

- 计数器宽度：32 位；
- 计数时钟的来源仅为内部时钟；
- 计数模式：向上计数。

结构框图

*图50-11. 自由运行计数器结构框图*提供了FRC内部配置的细节。

图 50-11. 自由运行计数器结构框图



功能说明

时钟源

自由运行计数器的时钟源只能是来自RCU内部25MHz的CK_TIMER_FREERUN。

向上计数模式

计数器从0开始连续递增至0xFFFFFFFF，计数器频率为25MHz。一旦计数器达到最大值，它会重新从0开始计数。自由运行计数器不产生中断。如果发生复位，计数器将初始化为0。

当前计数值可以从FRC_CNT寄存器中读取。

计数器在复位事件后最多需要160纳秒才能清除。

寄存器定义

自由运行计数器基地址：0x0000 3808

计数器寄存器（FRC_CNT）

地址偏移：0x00
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
CNT [31:16]															
r															

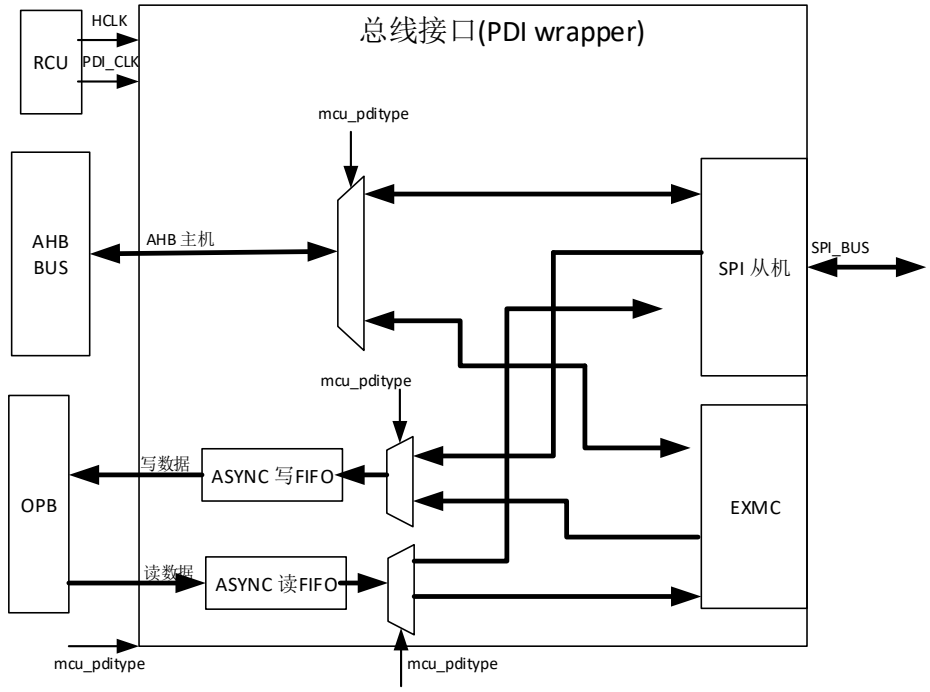
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT [15:0]															
r															

位/位域	名称	描述
31:0	CNT[31:0]	这些位是当前的计数值。 如果发生复位，这些位将初始化为 0，并且计数器从 0 开始向上计数。 当计数器达到最大值，这些位变为 0，计数器继续从 0 开始向上计数。 **注意:** 计数器在复位事件后最多需要 160 纳秒才能清除。

50.7. 总线接口（PDI Wrapper）

在 EtherCAT 中，EXMC 和 SPI SLAVE 被封装成一个用于系统集成的包装器。PDI Wrapper 只能用于在 SPI 选择数据。内部集成了两个异步 FIFO，每个都是 16X32 位。只能工作 SPI SLAVE，由 MCU_PDITYPE 引脚选择。

图 50-12. PDI Wrapper 框图



SPI 从机通过 AHB 通道访问寄存器，并通过异步读 FIFO 和异步写 FIFO 访问内核 RAM 数据。MCU_PDITYPE 引脚选择工作访问接口模块。当 MCU_PDITYPE 引脚为 0 时， SPI 从机可以访问内部数据。PDI_CLK 来自 SPI_SCK。HCLK 是一个 100MHz 的系统时钟，为 ASYNC_FIFO、SPI_SLAVE 提供时钟。

50.7.1. SPI/QSPI/OSPI 从机

简介

EtherCAT 支持 SPI/QSPI/OSPI 从模块。

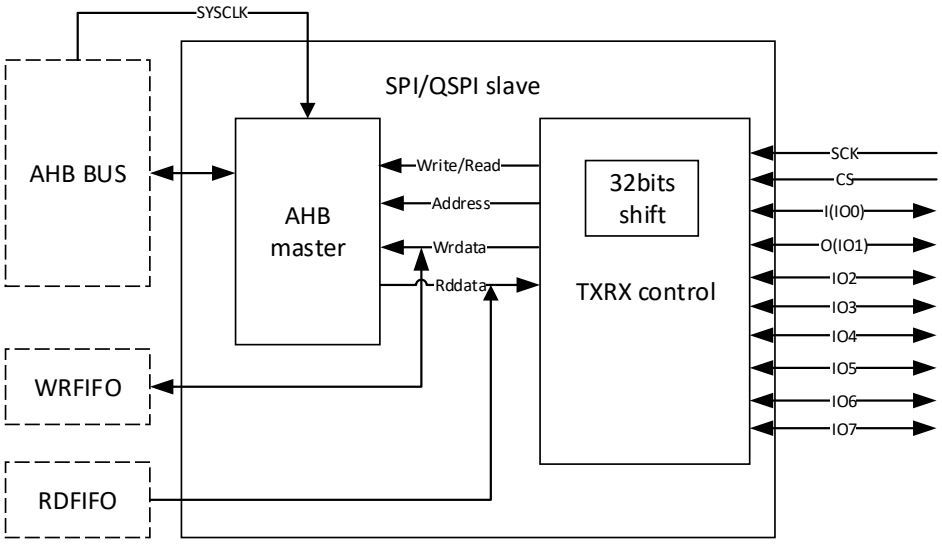
主要特征

- 支持最高SPI时钟频率为75MHz；
- 仅支持从模式；
- 支持先进先出缓冲区访问。

block 图

SPI 的 block 图如*图 50-13. SPI 的 block 图*所示。

图 50-13. SPI 的 block 图



SPI 信号描述

引脚描述

SPI/QSPI 从机模块包含两种引脚模式：4 线模式和 6 线模式。所有模式都包含公共引脚，SCK 和 CS。

表 50-5. 4 线模式

引脚名称	描述
SCK	SPI 时钟
CS	从机片选
I	输入引脚，接收 SPI/QSPI 主机数据
O	输出引脚，传输数据到 SPI/QSPI 主机

表 50-6. 6 线模式

引脚名称	描述
SCK	SPI/QSPI CLK
CS	从机片选
SIO0	输入输出引脚，接收 SPI/QSPI 主机数据和传输数据到 SPI/QSPI 主机
SIO1	输入输出引脚，接收 SPI/QSPI 主机数据和传输数据到 SPI/QSPI 主机

表 50-7. 4 线模式下的 OSPI

引脚名称	描述
SCK	SPI/QSPI 时钟
CS	从机片选
SIO0	输入输出引脚，接收 SPI/QSPI 主机数据和传输数据到 SPI/QSPI 主机
SIO1	输入输出引脚，接收 SPI/QSPI 主机数据和传输数据到 SPI/QSPI 主机
SIO2	输入输出引脚，接收 SPI/QSPI 主机数据和传输数据到 SPI/QSPI 主机

引脚名称	描述
SIO3	输入输出引脚，接收 SPI/QSPI 主机数据和传输数据到 SPI/QSPI 主机
SIO4	输入输出引脚，接收 SPI/QSPI 主机数据和传输数据到 SPI/QSPI 主机
SIO5	输入输出引脚，接收 SPI/QSPI 主机数据和传输数据到 SPI/QSPI 主机
SIO6	输入输出引脚，接收 SPI/QSPI 主机数据和传输数据到 SPI/QSPI 主机
SIO7	输入输出引脚，接收 SPI/QSPI 主机数据和传输数据到 SPI/QSPI 主机

SPI/QSPI/OSPI 从机控制器

概述

SPI 从机接口可以使用较少的引脚访问寄存器和 FIFO。在 SPI 模式下支持单、双和四线位通道，时钟频率高达 75MHz。QSPI 模式始终使用四位通道，并且也以高达 75MHz 的速率运行。OSPI 模式始终使用八位通道，并且也以高达 75MHz 的速率运行。

描述

以下是 SPI/QSPI/OSPI 客户端提供的功能概述：

- 快速读取：4 线（时钟、选择、数据输入和数据输出）读取速度高达 75MHz。串行命令、地址和数据。首次访问使用 dummy 字节。支持单次和多次寄存器读取，具有递增、递减或静态寻址功能。
- 双/四输出读取：4 线或 6 线（时钟、选择、数据输入/输出）以高达 75MHz 的速度读取。串行命令和地址，数据并行。首次访问使用 dummy 字节。具有递增、递减或静态寻址的单次和多次寄存器读取。
- 双/四 I/O 读取：4 线或 6 线（时钟、选择、数据输入/输出）以高达 75MHz 的速度读取。串行命令，并行地址和数据。首次访问时的 dummy 字节（或字节）。具有递增、递减或静态寻址的单一和多重寄存器读取。
- QSPI 读取：6 线（时钟、选择、数据输入/输出）最高可达 75MHz 的写入速度。并行命令、地址和数据。首次访问时使用哑字节。支持单次和多次寄存器读取，具有递增、递减或静态寻址功能。
- OSPI 读取：10 线（时钟、选择、数据输入/输出）最高可写入 75MHz。并行命令、地址和数据。首次访问时的 dummy 字节。具有增量、递减或静态寻址的单次和多次寄存器读取。
- 四线（时钟、选择、数据输入和数据输出）写入速度高达 75MHz。串行命令、地址和数据。具有递增、递减或静态寻址的单寄存器和多寄存器写入。
- 双/四数据写入：4 或 6 线（时钟、选择、数据输入/输出）以高达 75MHz 的速度写入。串行命令和地址，平行数据。具有递增、递减或静态寻址的单寄存器和多寄存器写入。
- 双/四地址/数据写入：4 或 6 线（时钟、选择、数据输入/输出）以高达 75MHz 的速度写入。串行命令，并行地址和数据。单次和多次寄存器写入，具有递增、递减或静态寻址。
- QSPI 写入：6 线（时钟、选择、数据输入/输出）最高可达 75MHz。并行命令、地址和数据。支持单次和多次寄存器写入，具有递增、递减或静态寻址功能。
- OSPI 写：10 线（时钟、选择、数据输入/输出）最高写入速度可达 75MHz。并行命令、地址和数据。支持单次和多次寄存器写入，具有递增、递减或静态寻址功能。

操作描述

在 SCK 输入时钟的上升沿，IO [7:0]引脚上的输入数据被采样。在 IO [7:0]引脚上，输出数据在时钟的下降沿被驱动。SCK 输入时钟可以是高电平脉冲或低电平脉冲。当 CS 片选输入为高电平时，IO [7:0]输入被忽略，IO [7:0]输出被置为三态。

在 SPI 模式下，8 位指令在 CS 有效电平后的第一个输入时钟上升沿开始。指令总是通过 I/IO0 串行输入。

对于读写指令，两个地址字节跟在指令字节后面。根据指令的不同，地址字节可以串行输入，或者每个时钟输入 2 位或 4 位。尽管所有寄存器都以双字节形式访问，但地址字段被认为是字节地址。地址字段的 15 位和 14 位指定地址是自动递减（10b）还是自动递增（01b）以进行连续访问。（如果访问内部 FIFO，将忽略 15 位和 14 位）

对于所有读指令，地址字节之后跟有 dummy 字节周期。在 dummy 字节周期期间，设备不驱动输出。dummy 字节（们）是串行输入的，或者是每时钟 2 位、4 位或 8 位。数据是串行输入的，或者是每时钟 2 位或 4 位。

对于读写指令，一个或多个 32 位数据字段跟随在 dummy 字节之后（如果有的话，否则它们跟随地址字节）。数据是串行输入的，或者每个时钟周期输入 2 位、4 位或 8 位数据。

QSPI 模式是通过启用四线 I/O（EQIO）指令从 SPI 模式进入的。一旦进入 QSPI 模式，所有后续的命令、地址、空字节和数据字节都是每个时钟 4 位。QSPI 模式可以使用重置四线 I/O（RSTQIO）指令退出。

OSPI 模式是通过 SPI 使用“启用八位 I/O（EOIO）”指令进入的。一旦进入 OSPI 模式，所有进一步的命令、地址、dummy 字节和数据字节都是每个时钟 8 位。OSPI 模式可以使用“重置八位 I/O（RSTQIO）”指令退出。

所有指令、地址和数据都以最高有效位（MSB）或最高有效二位（di-bit）或最高有效半字节（nibble）为先进行传输。地址以最高有效字节（MSB）为先进行传输。数据以最低有效字节（LSB）为先进行传输（小端模式）。

SPI 接口支持高达 75MHz 的输入时钟。

SPI 接口支持连续命令之间至少 50 纳秒的时间（CS 至少 50 纳秒的非有效时间）。

*表 50-8. SPI 指令*列出了 SPI 模式下支持的指令。*表 50-9. QSPI 指令*列出了 QSPI 指令。*表 50-10. OSPI 指令*列出了 OSPI 指令。不支持的指令不得使用。

表 50-8. SPI 指令

指令	描述	位宽	指令码值	地址字节	dummy 字节	数据字节	最大频率
配置							
EQIO	使能 QSPI	1-0-0	38h	0	0	0	75MHz
EOIO	使能 OSPI	1-0-0	3Ah	0	0	0	75MHz
RSTIO	使能 SPI	1-0-0	FFh	0	0	0	75MHz

指令	描述	位宽	指令码值	地址字节	dummy 字节	数据字节	最大频率
Read							
READ	读	1-1-1	0Bh	2	1	4 to ∞	75MHz
SDOR	SPI 双线输出读	1-1-2	3Bh	2	1	4 to ∞	75MHz
SDIOR	SPI 双线 I/O 读	1-2-2	BBh	2	2	4 to ∞	75MHz
SQOR	SPI 四线输出读	1-1-4	6Bh	2	1	4 to ∞	75MHz
SQIOR	SPI 四线 I/O 读	1-4-4	EBh	2	4	4 to ∞	75MHz
Write							
WRITE	写	1-1-1	02h	2	0	4 to ∞	75MHz
SDDW	SPI 双线输出写	1-1-2	32h	2	0	4 to ∞	75MHz
SDADW	SPI 双线 I/O 写	1-2-2	B2h	2	0	4 to ∞	75MHz
SQDW	SPI 四线输出写	1-1-4	62h	2	0	4 to ∞	75MHz
SQADW	SPI 四线 I/O 写	1-4-4	E2h	2	0	4 to ∞	75MHz

表 50-9. QSPI 指令

指令	描述	位宽	指令码值	地址字节	dummy 字节	数据字节	最大频率
Configuration							
RSTQIO	复位 QSPI	4-0-0	FFh	0	0	4 to ∞	75MHz
Read							
READ	读	4-4-4	0Bh	2	3	4 to ∞	75MHz
Write							
WRITE	写	4-4-4	02h	2	0	4 to ∞	75MHz

表 50-10. OSPI 指令

指令	描述	位宽	指令码值	地址字节	dummy 字节	数据字节	最大频率
配置							
RSTOIO	复位 OSPI	8-0-0	FFh	0	0	4 to ∞	75MHz
Read							
READ	读	8-8-8	0Bh	2	8	4 to ∞	75MHz
Write							
WRITE	写	8-8-8	02h	2	0	4 to ∞	75MHz

注意：位宽格式是：命令位宽，地址/ dummy 位宽，数据位宽。例如，1-2-4 表示命令使用 1 条线路，地址/dummy 使用 2 条线路，数据使用 4 条线路。

SPI 配置命令

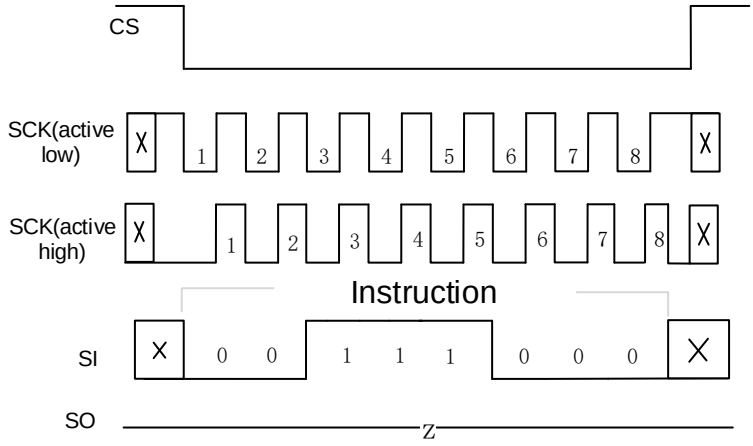
使能 QSPI

使能 QSPI 指令将操作模式更改为 QSPI。此指令仅在 SPI 总线协议中支持，并且仅支持最高 75MHz 的时钟频率。此指令在 QSPI 总线协议中不支持。

首先通过将 CS 置为有效信号来选择 SPI 客户端接口。8 位 EQIO 指令，38h，逐位通过 I/IO[0] 引脚输入，每个时钟输入一位。将 CS 输入变为无效状态以结束周期。

图 50-14. 使能 QSPI 展示了使能 QSPI 指令。

图 50-14. 使能 QSPI



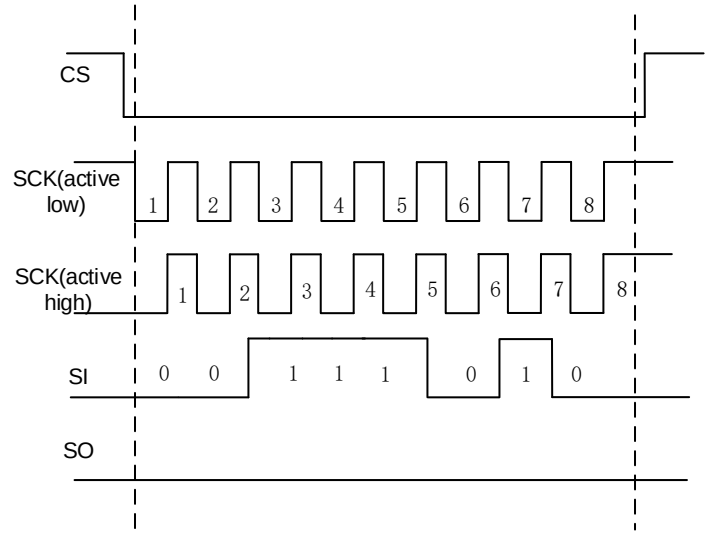
使能 OSPI

启用 OSPI 指令将操作模式更改为 OSPI。这个指令仅在 SPI 总线协议中支持，且仅支持最高 75MHz 的时钟频率。OSPI 总线协议不支持此指令。

首先通过将 CS 置为有效来选择 SPI 客户端接口。8 位 EOIO 指令，3Ah，逐位输入到 I/IO[0] 引脚，每个时钟一位。将 CS 输入变为无效状态以结束周期。

图 50-15. 使能 OSPI 展示了使能 OSPI 指令。

图 50-15. 使能 OSPI



复位 QSPI

复位 QSPI/OSPI 指令将操作模式更改为 SPI。这个指令支持在时钟频率高达 75MHz 的 SPI/QSPI/OSPI 总线协议中使用。

SPI/QSPI/OSPI 客户端接口是通过首先使将 CS 置为有效来选择的。8 位 RSTQIO 指令，FFh，在 SPI 模式下，通过 I/IO[0]引脚，每个时钟输入一位；在 QSPI 模式下，通过 IO[3:0]引脚，每个时钟输入四位。CS 输入被设为无效状态以结束周期。

图 50-16. SPI 模式复位 SPI 展示了 SPI 模式下的重置 SPI 指令。

图 50-17. QSPI 模式复位 QSPI 展示了 QSPI 模式下的复位 QSPI 指令。

图 50-18. OSPI 模式复位 OSPI 展示了 OSPI 模式下的复位 OSPI 指令。

图 50-16. SPI 模式复位 SPI

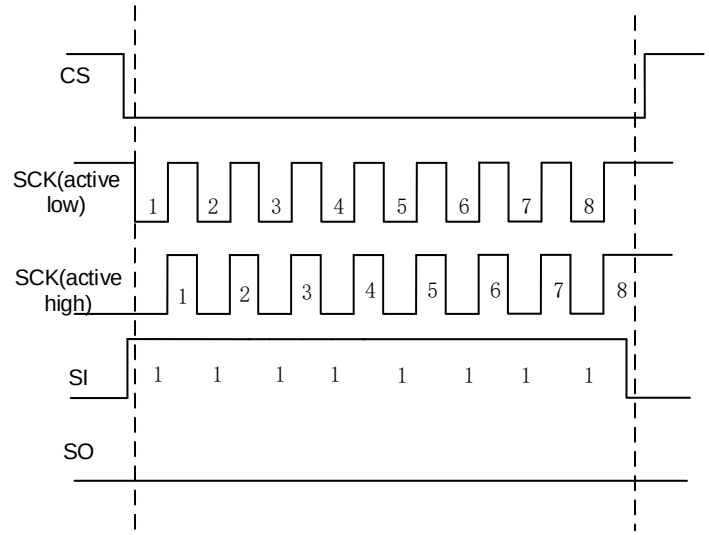


图 50-17. QSPI 模式复位 QSPI

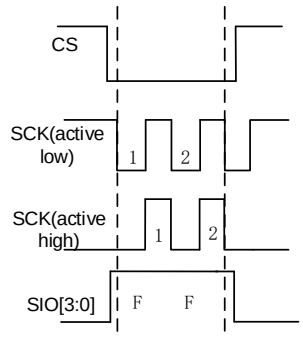
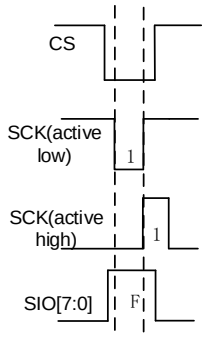


图 50-18. OSPI 模式复位 OSPI



SPI 读命令

SPI/QSPI 客户端支持各种读命令。以下适用于所有读命令。

多次读取

除了首次读取外，通过将 CS 置为有效时继续时钟脉冲来执行额外的读取。地址的高两位指定自动递增（address[15:14]=01b）或自动递减（address[15:14]=10b）。内部 DWORD 地址根据这些位进行增加、减少或保持不变。保持固定的内部地址对于寄存器轮询很有必要。

读取

读指令输入指令代码和地址以及 dummy 字节，每个时钟位输入一个位，并以每个时钟位输出数据。在 QSPI 模式下，指令代码和地址以及 dummy 字节每个时钟位输入四个位，数据以每个时钟位输出四个位。该指令在 SPI 和 QSPI 总线协议中得到支持，时钟频率可达 75MHz。

SPI/QSPI/OSPI客户端接口的选择首先通过将CS置为有效来实现。在SPI模式下，将8位的READ指令0Bh输入到I/O[0]引脚，然后是两个字节的地址和1个字节的dummy字节。在QSPI模式下，将8位的FASTREAD指令输入到IO[3:0]引脚，然后是两个字节的地址和3个字节的dummy字节。地址字节指定了设备内的一个字节地址。在OSPI模式下，将8位的FASTREAD指令输入到IO[7:0]引脚，然后是两个字节的地址和8个字节的dummy字节。地址字节指定了设备内的一个字节地址。

在最后一个dummy位（或半字节）的上升沿之后的下降沿，O/IO[1]引脚开始驱动，从所选寄存器的最低有效位的高位开始。对于QSPI模式，IO[3:0]从所选寄存器的最低有效位的高位开始驱

动。对于OSPI模式，IO[7:0]从所选寄存器的最低有效位的高位开始驱动。其余的寄存器位在随后的时钟下降沿上被移出。

将CS输入被置为无效状态以结束周期。此时，O/IO [7:0] 引脚被置为高阻态。

*图 50-19. SPI 读取*展示了 SPI 模式下典型的单个和多个寄存器的快速读取。

*图 50-20. QSPI 读取*展示了 QSPI 模式下典型的单个和多个寄存器的快速读取。

*图 50-21. OSPI 读取*展示了 OSPI 模式下典型的单个和多个寄存器的快速读取。

图 50-19. SPI 读取

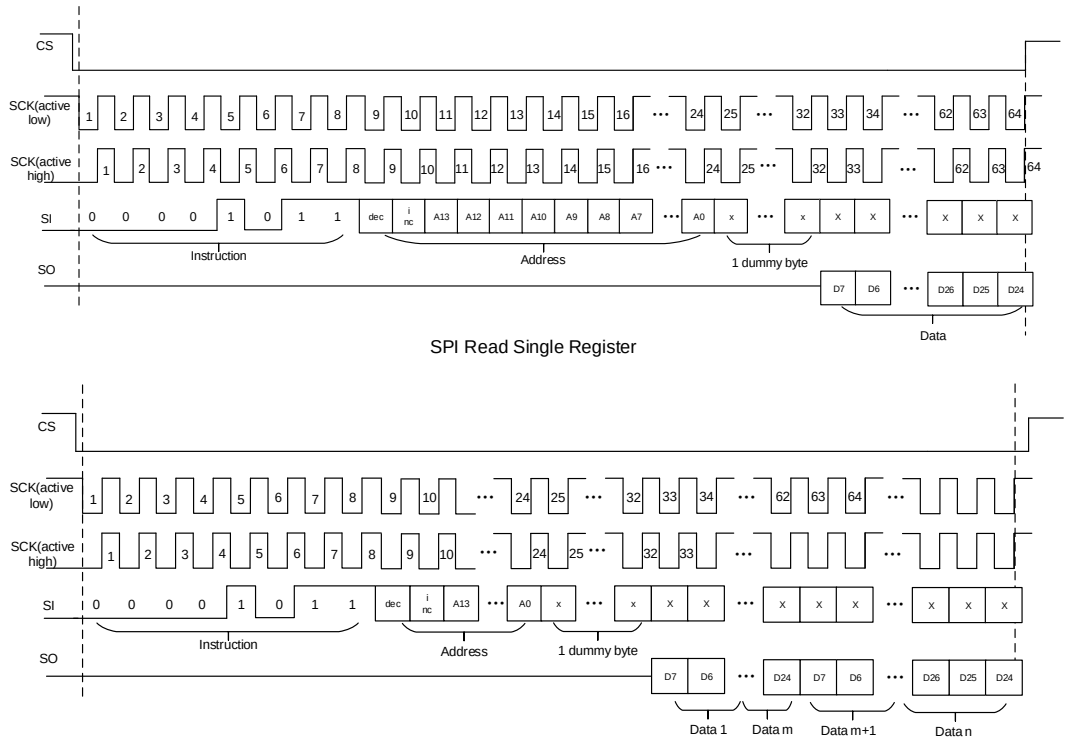
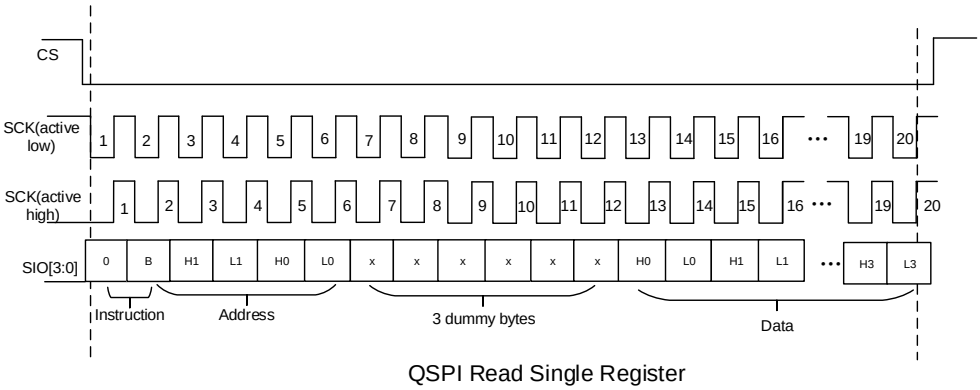
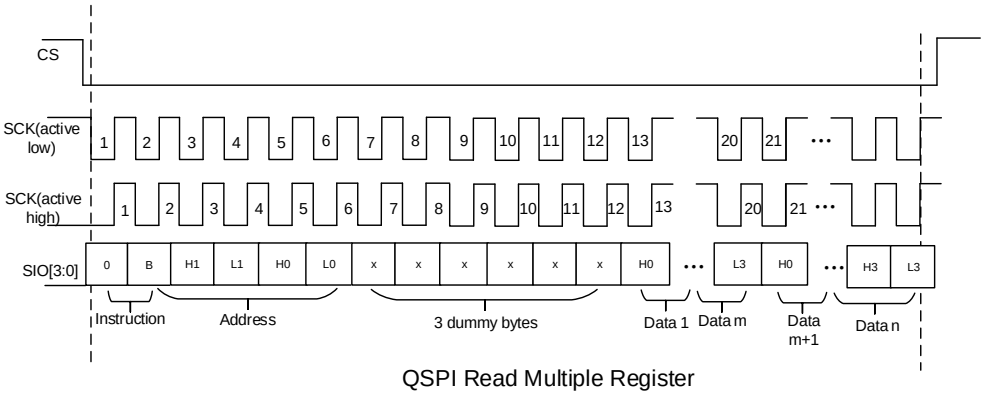


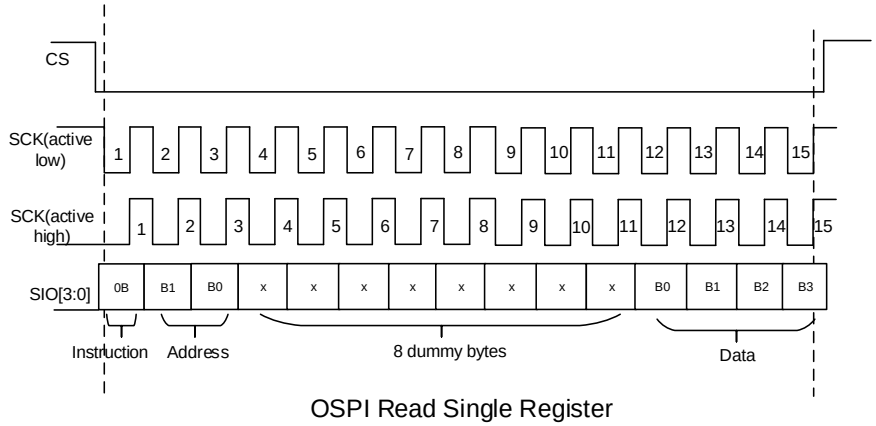
图 50-20. QSPI 读取



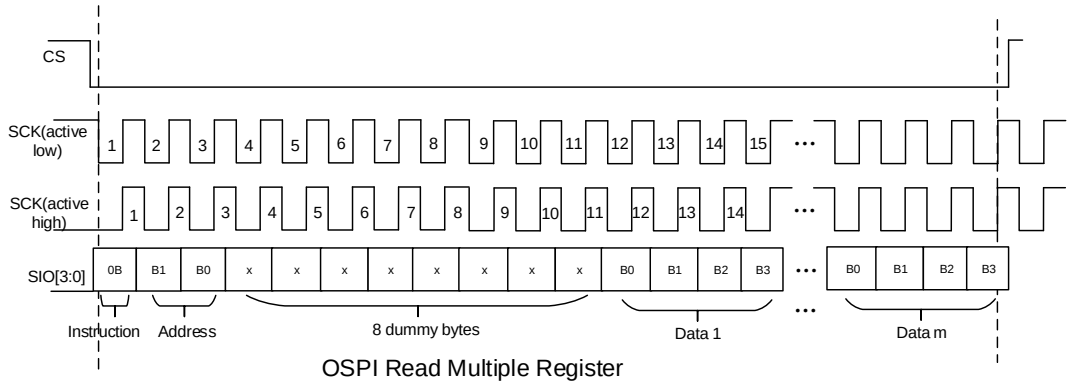


QSPI Read Multiple Register

图 50-21. OSPI 读取



OSPI Read Single Register



OSPI Read Multiple Register

双输出读取

SPI双输出读指令以每时钟一位的方式输入指令码和地址以及dummy字节，并以每时钟两位的方式输出数据。此指令仅支持SPI总线协议，并且仅支持高达75MHz的时钟频率。此指令不支持QSPI总线协议。

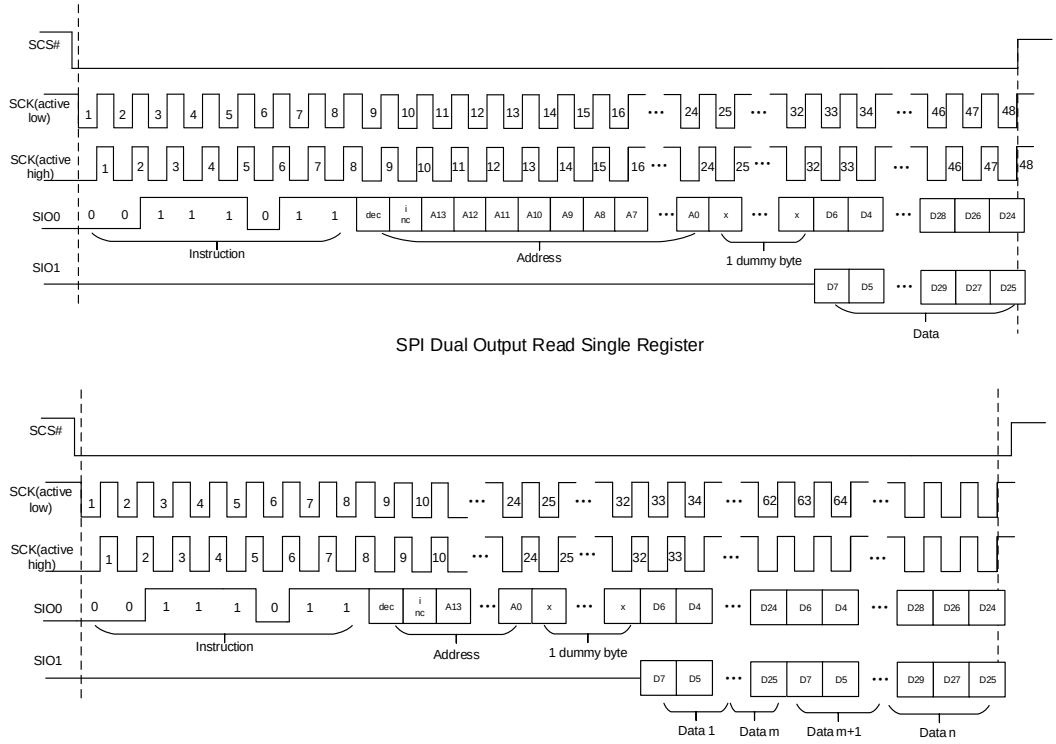
首先通过将CS置为有效信号来选择SPI客户端接口。然后输入8位SDOR指令，即3Bh，到IO [0]引脚，接着是两个字节的地址和一个字节的dummy数据。地址字节指定了设备内的一个字节地址。

在最后一个dummy二进制位的上升沿之后的下降沿，从选定寄存器的最低有效位的最高有效位开始驱动IO[1:0]引脚。其余的寄存器二进制位在随后的下降沿上被移出。

CS输入被置为无效状态以结束周期。此时IO [1:0] 引脚被置为高阻态。

*图 50-22. SPI 双输出读取*展示了典型的单个和多个寄存器的双输出读取。

图 50-22. SPI 双输出读取



四路输出读取

SPI四线输出读取指令以每时钟一位的方式输入指令码和地址以及dummy字节，并以每时钟四位的方式输出数据。这种指令只支持SPI总线协议，且时钟频率最高为75MHz。这种指令在QSPI总线协议中不支持。

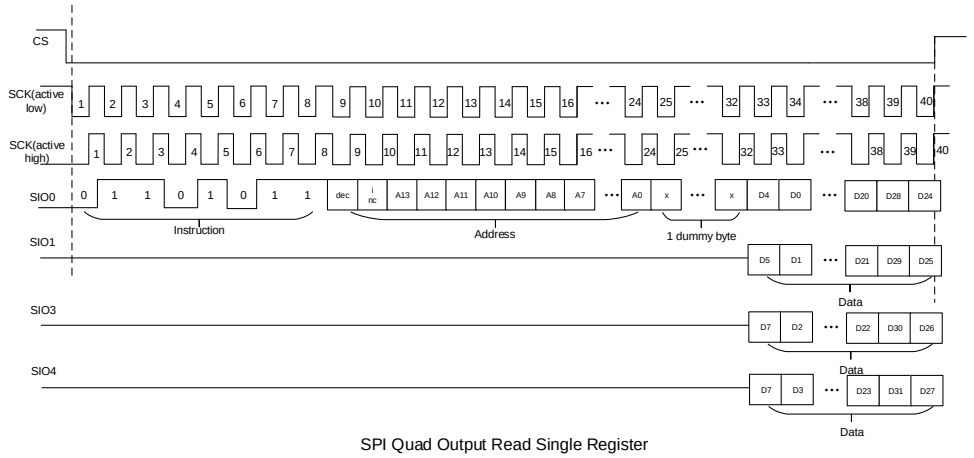
首先通过使CS置为有效状态来选择SPI客户端接口。然后输入8位的SQOR指令，6Bh，到IO [0]引脚，接着是两个字节的地址和一个字节的dummy码。地址字节指定设备内的字节地址。

在最后一个dummy位的上升沿之后的下降沿，IO [3:0]引脚开始驱动，从选定寄存器的最低有效位的位开始。其余的寄存器半字节被依次移出。

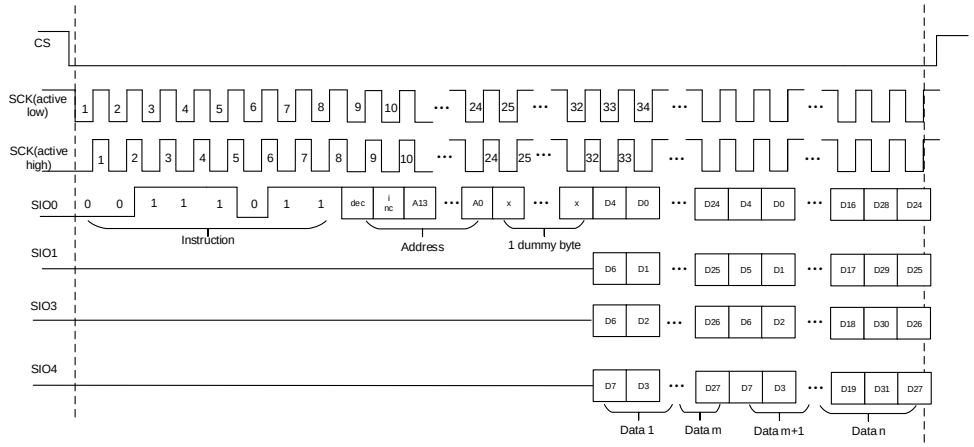
CS输入被置为无效状态以结束周期。此时，IO [3:0]引脚被三态化。

*图 50-23. SPI 四线输出读取*展示了典型的单寄存器和多寄存器四线输出读取。

图 50-23. SPI 四线输出读取



SPI Quad Output Read Single Register



SPI Quad Output Read Multiple Register

双输入 / 输出读取

SPI双I/O读指令每时钟输入指令代码一位，地址和dummy字节两位，并每时钟输出数据两位。这种指令只支持在SPI总线协议中，且时钟频率最高可达75MHz。这种指令在QSPI总线协议中不支持。

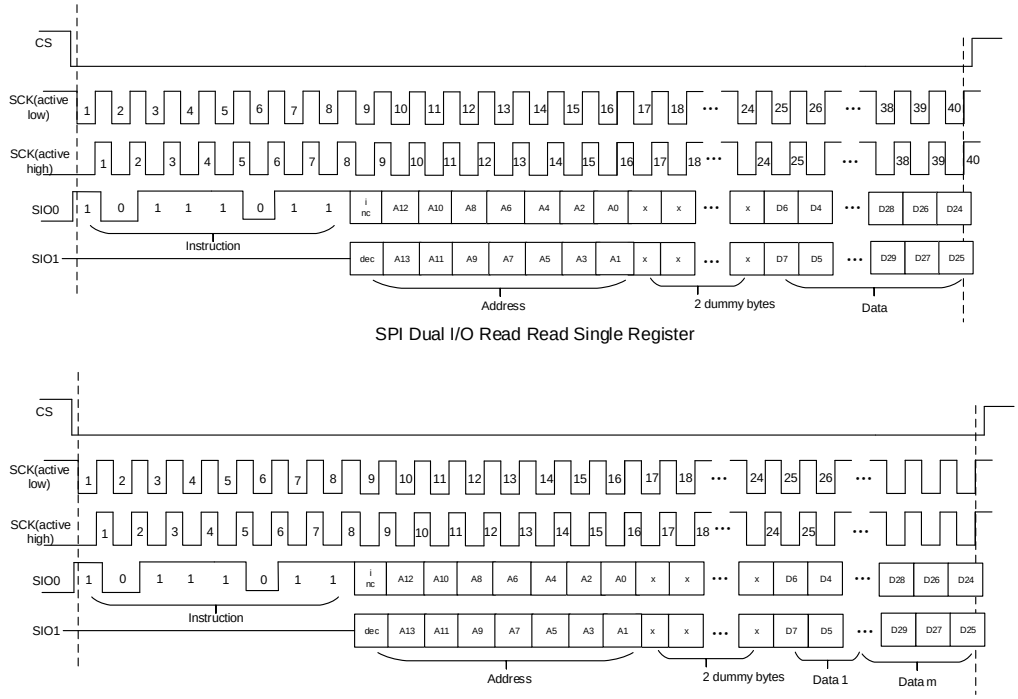
首先通过将CS置为有效来选择SPI客户端接口。然后，将8位的SDIOR指令，即BBh，输入到IO [0]引脚，接着将两个地址字节和2个dummy字节输入到IO [1:0]引脚。这些地址字节指定了设备内的一个字节地址。

在最后一个dummy二进制位的上升沿之后的下降沿上，IO [1:0]引脚开始驱动，从选定寄存器的最低有效位的高位开始。其余的寄存器二进制位在随后的下降时钟沿上被移出。

CS输入被置为无效以结束周期。此时，IO [1:0]引脚处于三态。

*图 50-24. SPI 双 I/O 读取*展示了典型的单个和多个寄存器的双 I/O 读取。

图 50-24. SPI 双 I/O 读取



四路输入 / 输出读取

SPI四线I/O读取指令以每个时钟一位输入指令码，地址和dummy字节以每个时钟四位输入，并以每个时钟四位输出数据。此指令仅在SPI总线协议中支持，且时钟频率最高为75MHz。此指令在QSPI总线协议中不支持。

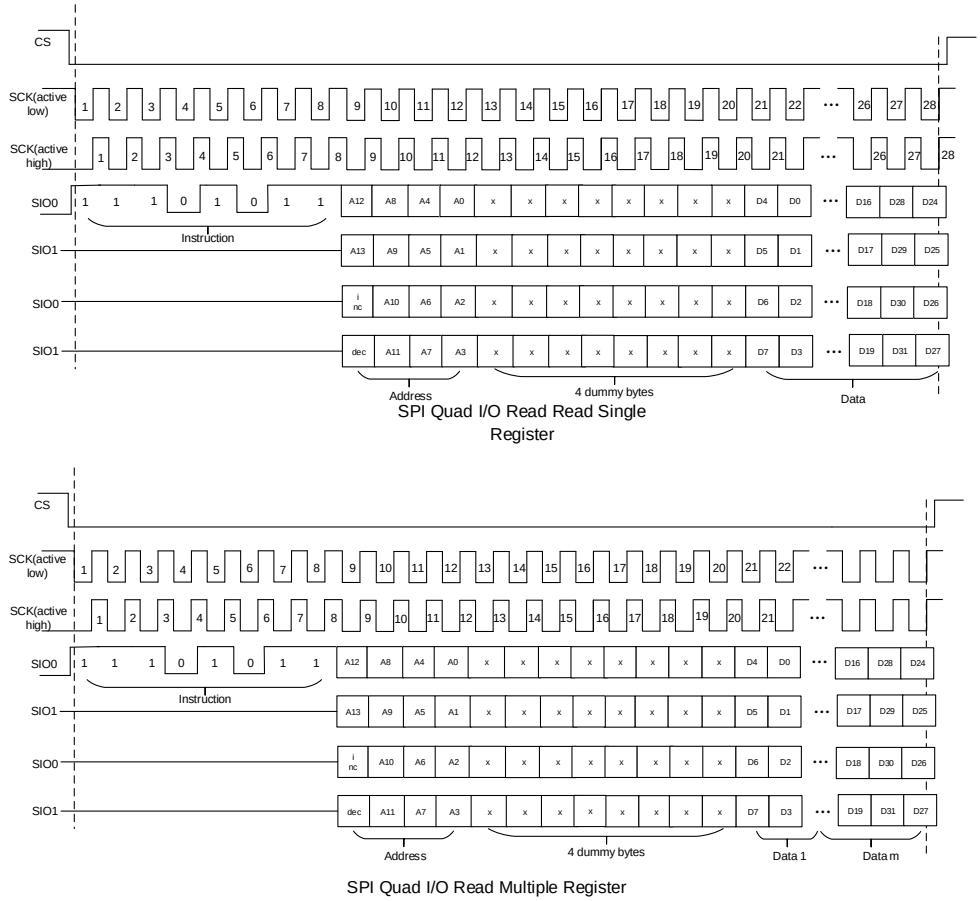
首先通过将CS置为有效来选择SPI客户端接口。然后输入8位的SQIOR指令EBh到IO [0]引脚，紧接着输入两个地址字节和4个dummy字节到IO [3:0]引脚。地址字节指定了设备内的一个字节地址。

在最后一个dummy半字节的上升沿之后的下降沿，IO [3:0] 引脚开始驱动，从所选寄存器的最低有效位的高位开始。其余的寄存器半字节在随后的下降沿上依次移出。

CS输入被置为非有效状态以结束周期。此时，IO [3:0]引脚被置为高阻态。

*图 50-25. SPI 四线 I/O 读取*展示了典型的单个和多个寄存器的双输出读取。

图 50-25. SPI 四线 I/O 读取



SPI 写命令

SPI/QSPI客户端支持多个写命令。以下适用于所有写命令。

多次写入

通过保持时钟脉冲和输入数据的激活状态，可以连续进行多次读取。地址的高两位指定自动递增（address[15:14]=01b）或自动递减（address[15:14]=10b）。内部DWORD地址根据这些位进行递增、递减或保持不变。保持固定的内部地址可能对寄存器“位操作”或其他重复写入很有用。

写入

写指令输入指令代码和地址以及数据字节，每个时钟输入一位。在QSPI模式下，指令代码和地址及数据字节每个时钟输入四位。这种指令在SPI和QSPI总线协议中都得到支持，时钟频率可高达75MHz。

SPI/QSPI客户端接口是通过首先将CS置为有效来选择的。在SPI模式下，8位写入指令02h输入到I/IO[0]引脚，然后是两个字节的地址。在QSPI模式下，8位写入指令02h输入到IO[3:0]引脚，然后是两个字节的地址。在OSPI模式下，8位写入指令02h输入到IO[7:0]引脚，然后是两个字节的地址。地址字节指定设备内的一个字节地址。

数据跟随地址字节。对于SPI模式，数据输入到I/IO[0]引脚，从最低有效位（LSB）的最高位开

始。对于QSPI模式，数据以半字节（nibble）宽度通过IO[3:0]输入，从最低有效位的最高位开始。对于OSPI模式，数据以半字节宽度通过IO[7:0]输入，从最低有效位的最高位开始。其余的位或半字节在随后的时钟边缘上移入。当32位数据输入完毕后，数据写入寄存器。如果CS返回高电平时，没有写入32位数据，则写入被认为是无效的，寄存器不受影响。

CS 输入被置为非有效状态以结束周期。

*图 50-26. SPI 写入*展示了 SPI 模式下典型的单个和多个寄存器写入。

*图 50-27. QSPI 写入*展示了 QSPI 模式下典型的单个和多个寄存器写入。

*图 50-28. OSPI 写入*展示了 OSPI 模式下典型的单个和多个寄存器写入。

图 50-26. SPI 写入

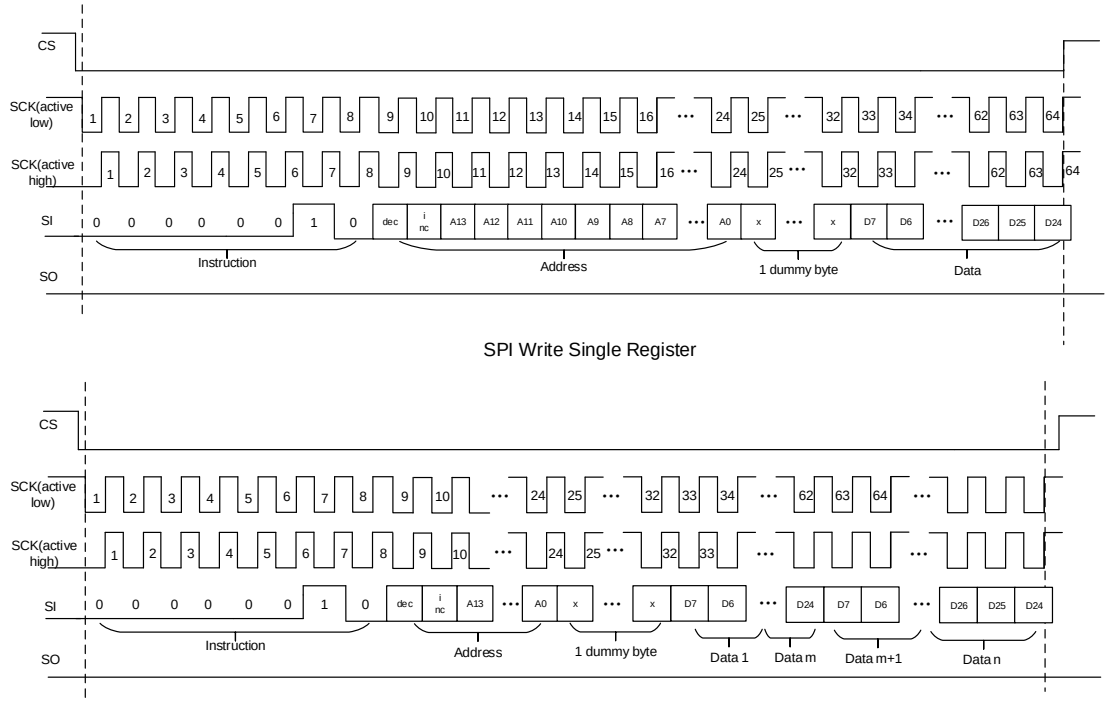
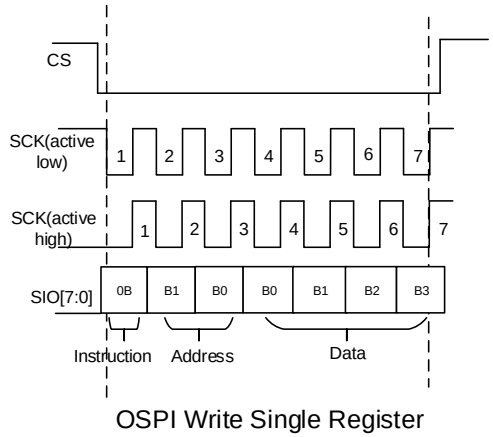


图 50-27. QSPI 写入



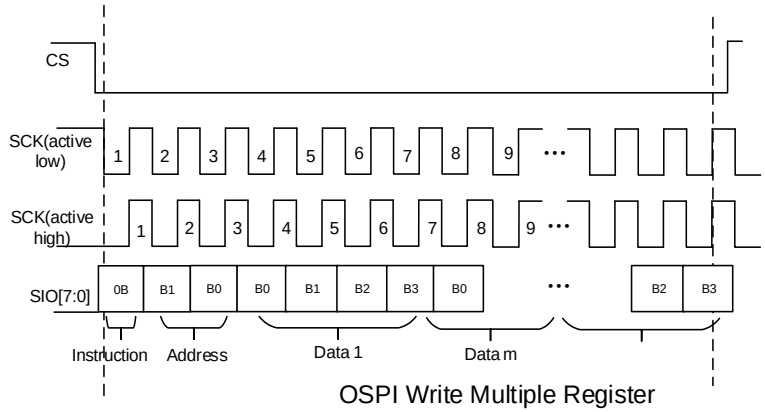
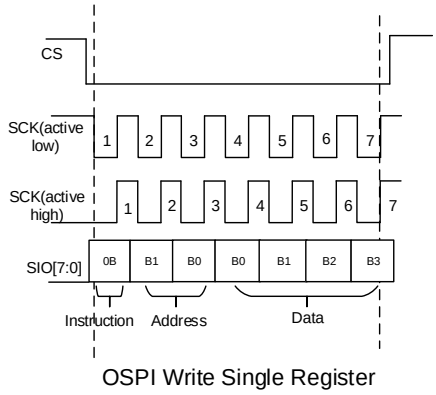
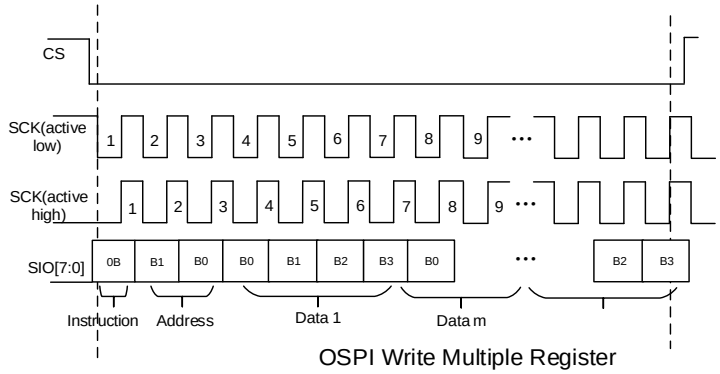


图 50-28. OSPI 写入





双数据读取

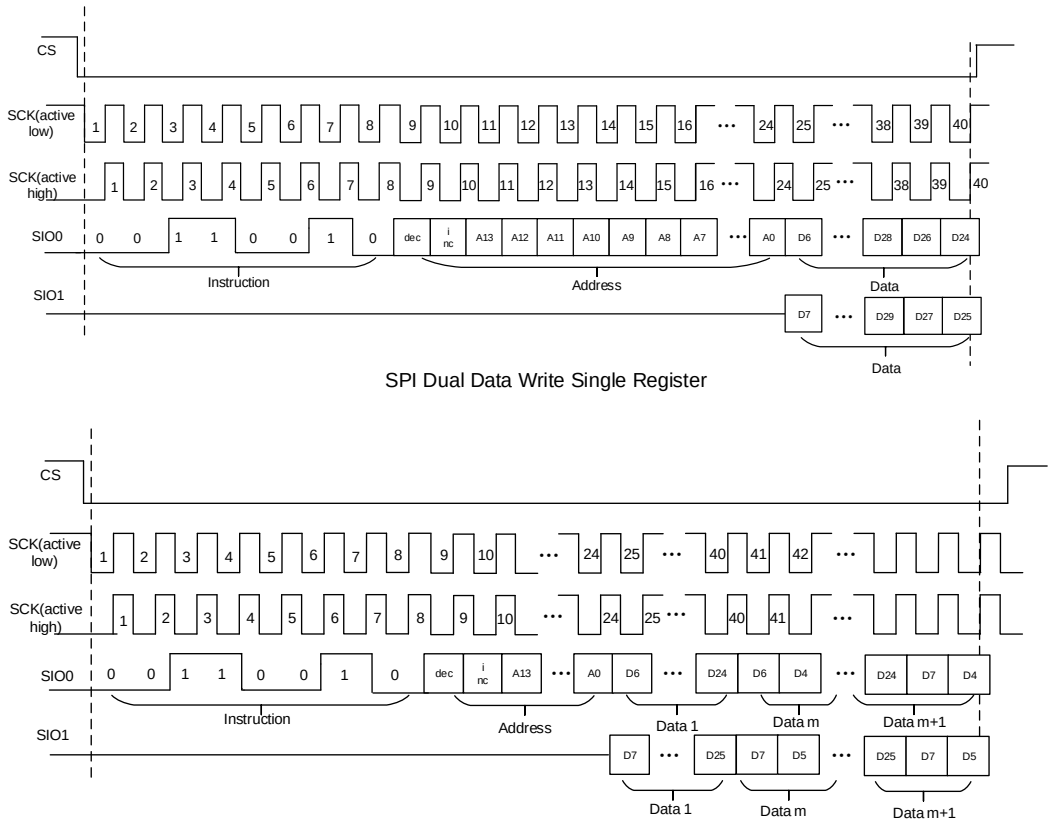
SPI双数据写入指令以每个时钟一位输入指令码和地址字节，以每两个时钟位输入数据。该指令仅在SPI总线协议中可以被支持，且时钟频率最高可达75MHz。在QSPI总线协议中不支持此指令。

首先通过将CS置为有效来选择SPI客户端接口。将8位的SDDW指令，32h，输入到IO [0]引脚，然后是两个字节的地址。地址字节指定设备内的字节地址。数据跟在地址字节后面。数据输入到IO [1:0]引脚，从最低有效字节的最左边的位开始。其余的二进制位在随后的时钟边缘上移入。在输入32位后，数据被写入寄存器。如果CS返回高电平时没有写入32位，则写入被认为是无效的，寄存器不受影响。

CS 输入被置为非有效状态以结束周期。

*图 50-29. SPI 双数据写入*展示了典型的单个和多个寄存器双数据写入。

图 50-29. SPI 双数据写入



四线数据读取

SPI四线数据写指令以每个时钟一位输入指令码和地址字节，并以每个时钟四位输入数据。此指令仅在SPI总线协议中支持，且时钟频率最高可达75MHz。此指令在QSPI总线协议中不受支持。

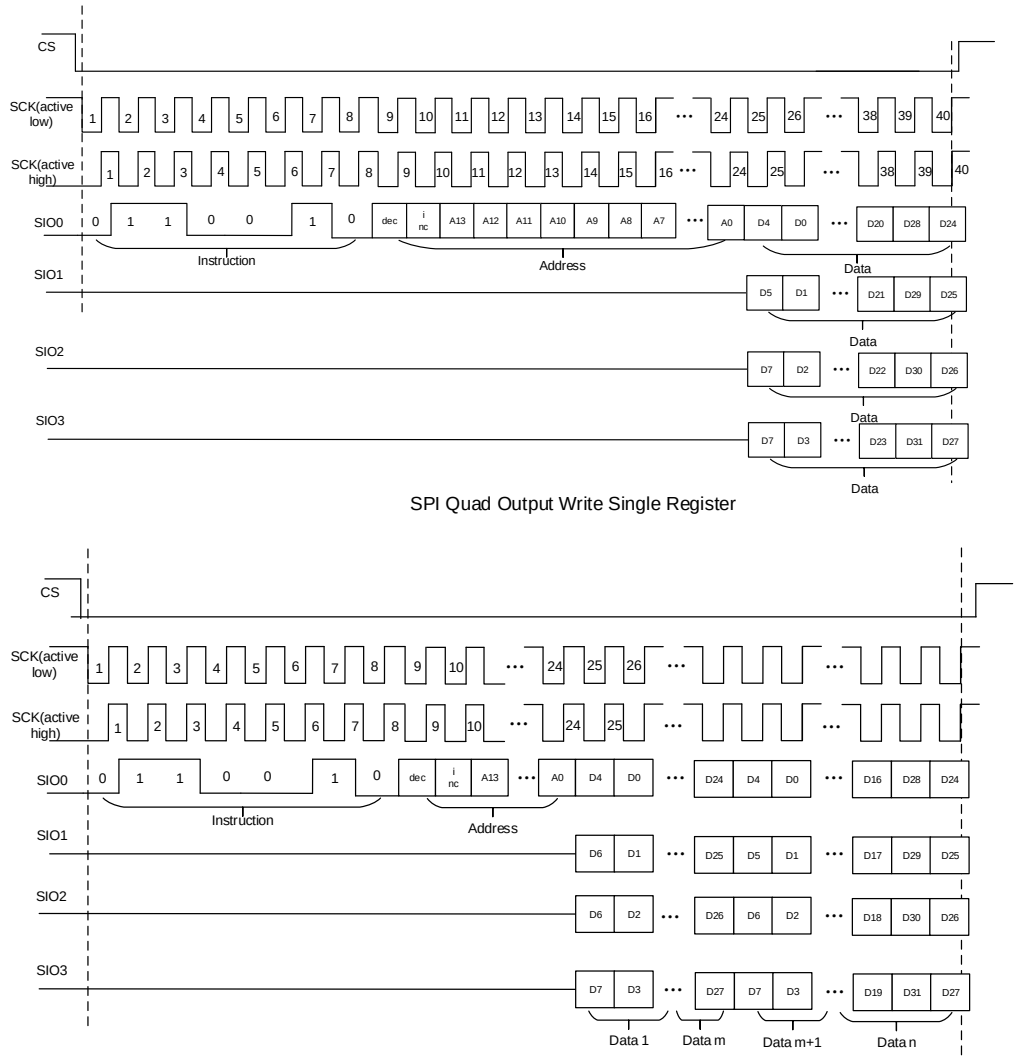
首先通过将CS置为有效来选择SPI从机接口。然后输入8位SQDW指令，62h，到IO[0]引脚，紧接着是两个地址字节。地址字节指定了设备内的一个字节地址。

数据跟随地址字节。数据从LSB的最高位开始，通过IO[3:0]引脚输入。剩余的半字节在随后的时钟边缘被移入。32位数据输入完成后，数据被写入寄存器。如果在CS返回高电平时没有写入32位，则写入被认为是无效的，寄存器不受影响。

CS输入被置为无效状态以结束周期。

*图 50-30. SPI 四线数据写入*展示了典型的单个和多个寄存器的四线数据写入。

图 50-30. SPI 四线数据写入



双地址 / 数据写入

SPI双地址/数据写指令以每个时钟位输入指令码，以每个时钟两位输入地址和数据字节。这种指令只支持SPI总线协议，且时钟频率最高为75MHz。这种指令在QSPI总线协议中不支持。

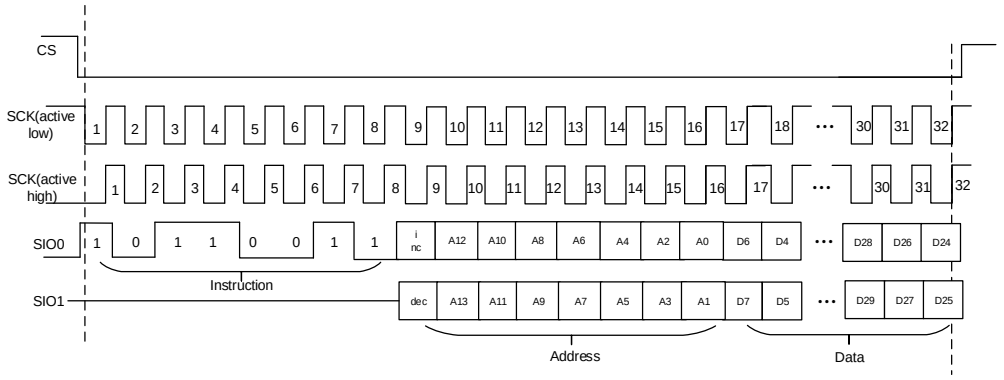
首先通过将CS置为有效来选择SPI客户端接口。然后，将8位的SDADW指令，即B2h，输入到IO[0]引脚，接着将两个地址字节输入到IO[1:0]引脚。这些地址字节指定了设备内的一个字节地址。

数据紧随地址字节之后。数据从IO[1:0]引脚输入，从最低有效字节的最高位开始。剩余的二进制位在随后的时钟边沿上被移入。当32位数据输入完成后，数据被写入寄存器。如果CS返回高电平时没有写入32位数据，则认为写入无效，寄存器不受影响。

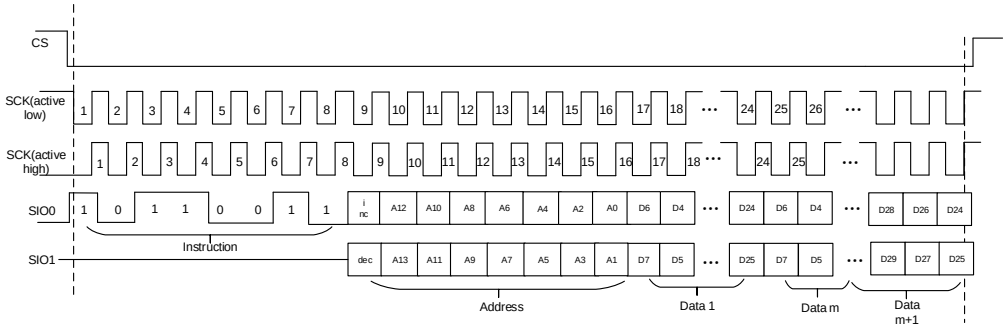
CS 输入被置为非活动状态以结束周期。

*图 50-31. SPI 双地址/数据写入*展示了典型的单个和多个寄存器的双地址/数据写入。

图 50-31. SPI 双地址/数据写入



SPI Dual Address / Data Write Single Register



SPI Dual Address / Data Write Multiple Register

四路地址 / 数据写入

SPI四地址/数据写指令以每时钟一位的方式输入指令码，以及每时钟四位的方式输入地址和数据字节。这种指令仅在SPI总线协议中支持，且仅支持最高75MHz的时钟频率。在QSPI总线协议中不支持此指令。

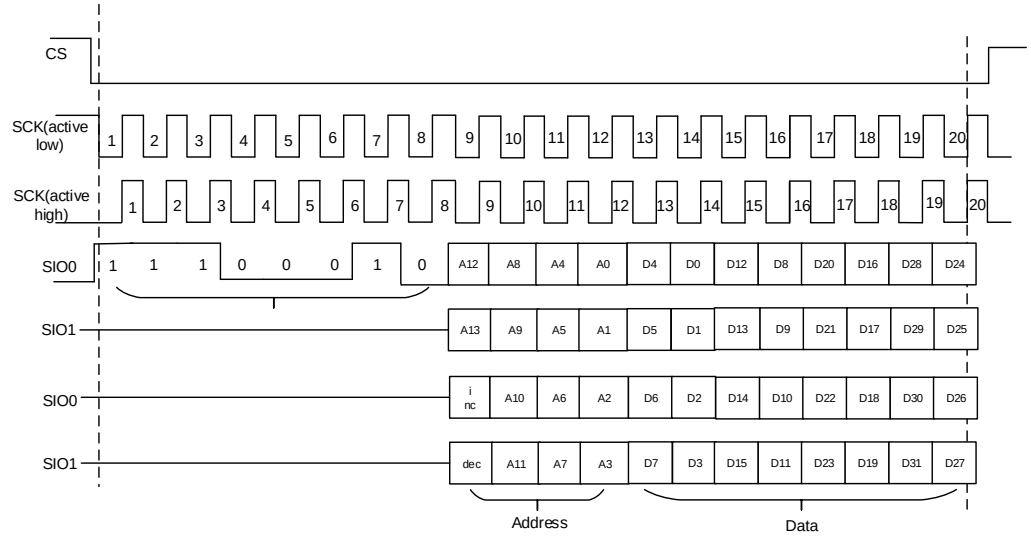
首先通过将CS置为有效来选择SPI客户端接口。然后输入8位的SQADW指令，E2h，到IO[0]引脚，随后输入两个地址字节到IO[3:0]引脚。这些地址字节指定了设备内的字节地址。

数据紧跟在地址字节之后。数据从最低位的高有效位开始，通过IO[3:0]引脚输入。其余的半字节在随后的时钟边缘上被移入。在32位数据输入完毕后，数据被写入寄存器。如果CS返回高电平时没有写入32位，则写入被认为是无效的，寄存器不受影响。

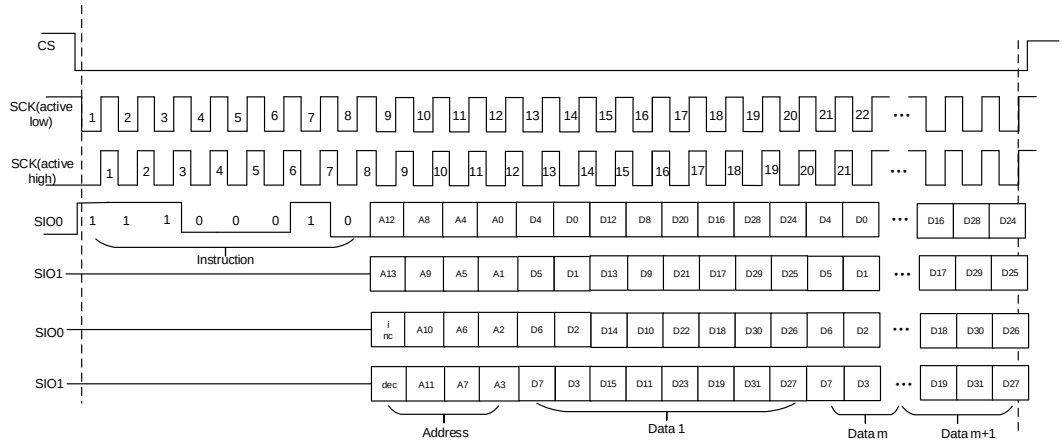
CS 输入被置为非有效状态以结束周期。

*图 50-32. SPI 四路地址/数据写入*展示了典型的单个和多个寄存器四线地址/数据写入。

图 50-32. SPI 四路地址/数据写入



SPI Quad I/O Read Read Single Register



SPI Quad Address/Data Write Multiple Register

SPI 唤醒系统

当芯片进入低功耗模式时，用户可以通过SPI/QSPI/OSPI访问BYTE_TEST和READY寄存器以退出低功耗模式。

要确定主机接口是否正常工作，应轮询字节序测试寄存器（BYTE_TEST）。一旦读取到正确的模式，就可以认为接口已经正常工作。此时，可以轮询设备就绪（READY）寄存器，以确定设备是否完全唤醒。

SPI 访问 FIFO

SPI/QSPI/OSPI支持访问寄存器和FIFO。在访问FIFO的过程中，用户需要确保FIFO在访问时不会超出边界，否则会发生数据丢失。

BUS 模块提供了一些 FIFO 计数寄存器，包括 PRAM_RD_AVAIL_CNT 和 PRAM_WR_AVAIL_CNT，这些寄存器可以通过SPI总线读出。当用户读取TXFIFO时，读取的数据量应该小于或等于PRAM_RD_AVAIL_CNT。当用户写入RXFIFO时，写入的数据量应该小于或等于PRAM_WR_AVAIL_CNT。

在读取TXFIFO时，建议先读取PRAM_RD_AVAIL_CNT，然后再从TXFIFO中读取PRAM_RD_AVAIL_CNT数量的数据。

在写入RXFIFO时，建议首先读取PRAM_WR_AVAIL_CNT，然后写入PRAM_WR_AVAIL_CNT数据到RXFIFO。

50.8. 以太网 PHYS

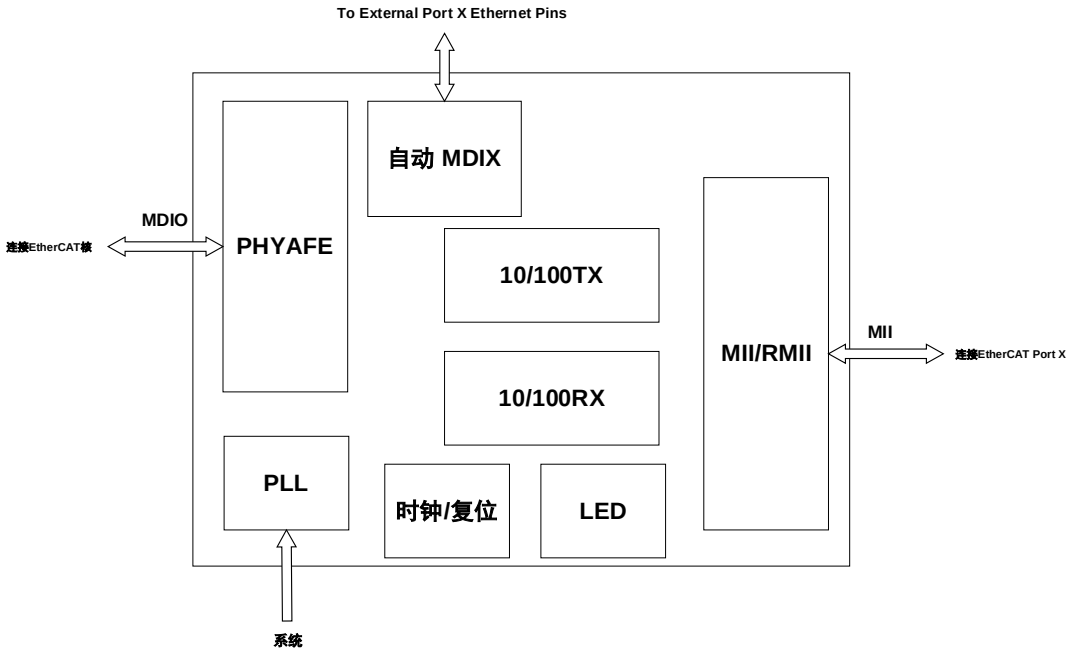
50.8.1. 简介

ESC 包含两个电压型 PHY A 和 B，它们在功能上是相同的。PHY A 连接到 EtherCAT 端口 0 或 2。PHY B 连接到 EtherCAT 端口 1。这些 PHY 通过内部 MII 接口与它们各自的 MAC 进行接口。可以配置为全双工 100 Mbps（100BASE-TX）以太网操作。所有 PHY 寄存器遵循 IEEE 802.3 规定的 MII 管理寄存器，并且是完全可配置的。

50.8.2. 主要特征

- 完全符合IEEE 802.3 100 Base-TX标准，并支持EEE。
- 自动协商和并行检测能力，用于自动速度和双工选择
- 支持MII接口
- 10Base-T的自动极性校正
- 支持即插即用的Auto-MDIX功能
- 可编程环回模式用于诊断
- 支持不同应用的可编程LED输出，开机自检LED
- 支持WOL（Wake-On-Lan）功能

图 50-33. PHY 功能模块图



50.8.3. 功能概述

操作模式

在 100BASE-TX 中，发送端将来自 MAC 接口的数据流进行 4B/5B 编码、串行化、扰乱和 MLT3 编码。在接收端，来自介质的数据流被恢复，从 MLT3 解码，去扰乱，并行化，并进行 5B/4B 解码，解码成 4 位数据。

当没有数据需要传输时，系统会通过低功耗空闲（LPI）信号通知其链路伙伴，然后进入发送器省电模式。另一方面，当从链路伙伴接收到 LPI 信号时，系统会进入接收机省电模式。只有周期的信令用于保持链路活跃。

MII 接口

MII (Media Independent Interface)是 IEEE 802.3u 中定义的 MAC 和 PHY 之间的接口。对于 10Mbps 传输，时钟频率为 2.5MHz。对于 100Mbps 传输，时钟频率为 25MHz。MAC 以 TXCLK 和 RXCLK 同步传输和接收数据，这些时钟由 PHY 生成。

TXEN 生效时，TXD[3:0]被 PHY 接受进行传输。在 TXEN 有效时，TXER 有效表示传输编码错误。TXEN 无效时，TXER 生效，TXD[3:0]等于 0001 的组合显示请求进入（或保持）低功耗状态。TXEN、TXER 和 TXD[3:0]与 TXCLK 同步采样。

当 RXDV 生效时，RXD[3:0]将从 PHY 恢复的数据传输到 MAC。RXER 表示接收错误。RXDV 无效，RXER 有效，且 RXD[3:0]等于 0001 的组合，通知 LPI 客户端（例如 MAC）链路伙伴处于低功耗状态。CRS 在 PHY 发送或接收时生效。COL 在 PHY 检测到冲突时生效。RXDV、RXER 和 RXD 与 RXCLK 同步。

SMI 接口

串行管理接口（SMI）可用于在站点管理（STA）和物理层（PHY）之间传输控制和状态信息。用户还可以通过 SMI 访问 PHY 的内部寄存器设置。MDIO 是双向信号，主要由命令（读/写）字段和数据字段组成，并与 MDC 同步。当没有驱动信号时，MDIO 引脚应该上拉。

自动 MDI/MDIX 和极性配置

自动 MDI/MDIX 配置目的在消除两个设备之间外部交叉电缆的需要。PHY 可以自动进行 MDI/MDIX 配置，以使传输和接收正常工作。MDI/MDIX 配置也可以通过手动设置寄存器来确定。

PHY 可以自动较正电缆对上的极性错误。

环回模式

环回模式提供了发送和接收的诊断功能，可以对发送和接收的数据路径进行测试。

Wake-On-LAN

Wake-On-LAN 通过使用一种特殊的网络消息来实现，这种消息被称为“魔术包”。魔术包包含目标设备的 MAC 地址。监听设备等待一个合法的、针对它的魔术包，然后激活系统

唤醒程序。

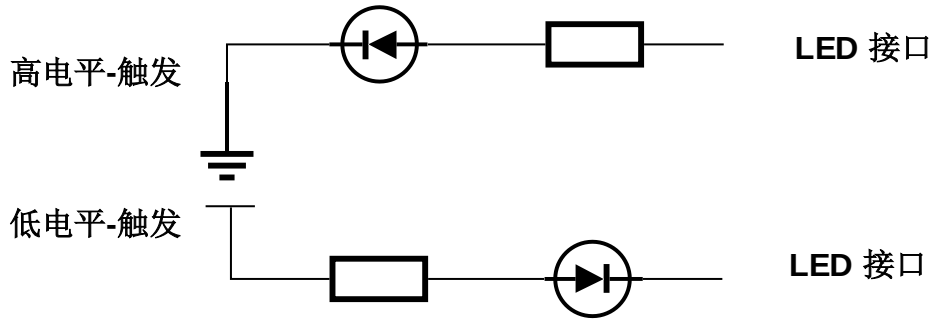
当启用了远程唤醒功能时，物理层在接收到合法的魔术包后会发送一个中断。

LED 模式

有 3 个 LED 接口用于控制链路状态、速度和双工模式指示的 LED 状态。

有两种 LED 连接类型：高电平触发和低电平触发，如图 82 所示：

图 50-34. LED 连接图



当 LED 接口被连接为高电平触发类型，接口 PHY_LED_POL 应该被置为 0。

当 LED 接口被连接为低电平触发类型，接口 PHY_LED_POL 应该被置为 1。

LED 状态信息的定义如下：

- 链接LED：
  - 打开：链接已激活
  - 关闭: 链路已断开
  - 刷新：数据传输
- 速度LED
  - 打开：100M
  - 关闭：10M
  - 刷新：N/A
- 双工LED
  - 打开：全双工
  - 关闭：半双工
  - 刷新：碰撞

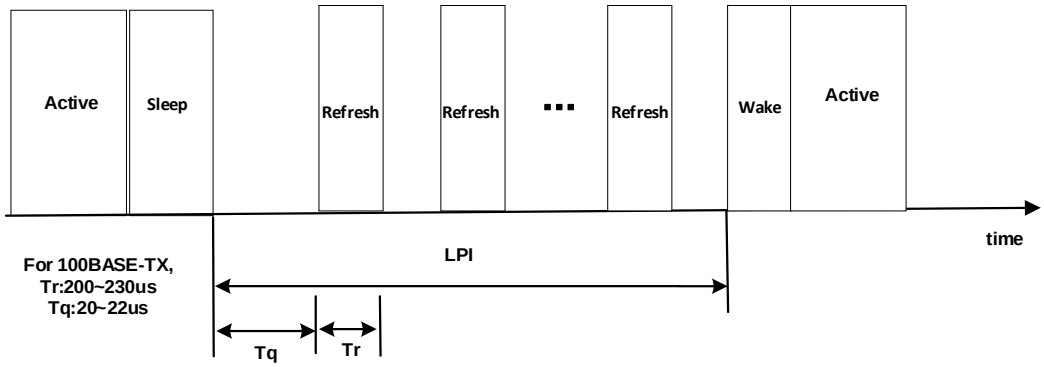
LED 的刷新周期是 33 毫秒。

LPI 信令

当 LPI 客户端发出 LPI 请求时，物理层会发送 Sleep 符号，以告知其链路伙伴本地物理层即将进入 LPI 状态。物理层在发送 Sleep 符号后进入 LPI 状态。在 LPI 状态下，只定期传输刷新符号。当 LPI 客户端请求退出 LPI 状态时，物理层会发送唤醒符号，以请求链路伙伴唤醒以进行进一步传输。

当接收到来自其链路伙伴的 Sleep 符号时，PHY 知道远端 PHY 将进入 LPI 状态。在远端 PHY 停止传输后，本地 PHY 可以关闭一些电路以节省电力。在低 LPI 状态下，物理层使用 Refresh 符号来更新其过滤系数并调整时序。当从其链路伙伴接收到 Wake 符号时，物理层在指定的恢复时间之前从 LPI 状态回到正常操作。

图 50-35.100Base-TX LPI



50.8.4. PHY 寄存器

Page0 寄存器

PHY 控制寄存器（PHY_MII_CTL）

地址偏移：0x00
复位值：0x3100

该寄存器可以按字节（8位）、半字（16位）和字（32位）进行访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR_MAIN_REST	LOOPBACK_EN	FORCE_SPEED(LSB)	MR_AUTONEG_EN	POWERDOWN	保留	MR_RESTART_AUTONEG	FORCE_DUPLEX	COL_TEST	FORCE_SPEED(MSB)	UNIDIRECTIONAL_EN	保留				
rw	rw	rw	rw	rw		rw	rw	rw	rw	rw					

位/位域	名称	描述
15	MR_MAIN_REST	主要复位 将 PHY 的状态和控制寄存器复位为默认值并自清除 1：复位 0：正常
14	LOOPBACK_EN	环回使能 在环回模式下，工作模式将被设置为全双工模式，自动关闭自协商功能
13	FORCE_SPEED(LSB)	LSB 强制速度 位 13 和位 6 结合速度选择。它仅在 MR_AUTONEG_EN = 0 时有效。启用光纤模式后，内部速率自动设置为 100M，忽略该字段设置。 00：10M 01：100M

		1X：保留
12	MR_AUTONEG_EN	自协商使能 该位的结果应与 I_FXEN 和 EN_FX 进行 OR 'ed，以确定内部最终的自动协商使能信号(自动协商将在光纤中禁用模式)。注意 LOOPBACK_EN 设置为 1，内部自动协商功能也将由 PHY 自动禁用，并且不会反映到该位上。 1: 自协商使能 0: 禁用自动协商（根据 FORCE_SPEED 和 FORCE_DUPLEX 的设置，强制为 10M 或 100M 模式）。
11	POWERDOWN	下电模式 高电平有效编程 PHY 进入断电(断电状态下行模拟 TX，模拟 RX，模拟 AD)
10	保留	必须保持复位值
9	MR_RESTART_AUTONEG	重启自协商 此位写入 1 时，该为由 PHY 自动清除，写入 0 时不受影响 1：重启自协商 0：正常
8	FORCE_DUPLEX	强制工作模式 此位仅在 MR_AUTONEG_EN = 0 时有效。该位的结果应与 LOOPBACK_EN 进行匹配，以确定内部最终双工能力 1：全双工（默认） 0：半双工
7	COL_TEST	冲突测试 当该位被置为有效时，PHY 响应 TX_EN 有效将响应 512 BT 内的 COL 信号，当连接到 MII 或连接到 GMII 时，响应 TX_EN 的有效将在 4BT 或 16BT 内生效
6	FORCE_SPEED(MSB)	MSB 强制速度 参考 bit13
5	UNIDIRECTIONAL_EN	单向使能 启用从 MII / GMII 接口编码和传输数据的能力，而不管 PHY 是否确定已建立有效链路。 此功能仅在禁用“自动协商”且工作模式为全双工时有效
4:0	保留	必须保持复位值

PHY 状态寄存器（PHY_MII_STATUS）

地址偏移：0x01
复位值：0x79C9

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
100BASE_T4	100BASE-X_FULL_DPX	100BASE-X_HALF_DPX	10BASE-T_FULL_DPX	10BASE-T_HALF_DPX	100BASE_T2_FULL_DPX	100BASE_T2_HALF_DPX	EXTENDED_STATUS	UNIDIRECTIONAL_ABLT	MF_PRB_SUP	MR_AUTONEG_CPLT	REMOTE_FAULT	AUTONEG_ABLT	LINK_STATUS	JABBER_DETECT	EXTENDED_CAP

r r r r r r r r r r r r r rc_r r r

位/位域	名称	描述
15	100BASE_T4	100BASE T4性能 不支持，始终保持为 0
14	100BASE-X_FULL_DPX	支持100BASE TX全双工 1：PHY支持100BASE TX全双工 0：PHY 不支持 100BASE TX 全双工
13	100BASE-X_HALF_DPX	支持100BASE TX半双工 1：PHY支持100BASE TX半双工 0：PHY 不支持 100BASE TX 半双工
12	10BASE-T_FULL_DPX	支持10BASE-T全双工 1：PHY支持10BASE-T全双工 0：PHY 不支持 10BASE-T 全双工
11	10BASE-T_HALF_DPX	支持10BASE-T半双工 1：PHY支持10BASE-T半双工 0：PHY 不支持 10BASE-T 半双工
10	100BASE _T2_FULL_DPX	100BASE T2全双工 不支持，始终保持为 0
9	100BASE_T2_HALF_DPX	100BASE T2半双工 不支持，始终保持为 0
8	EXTENDED_STATUS	扩展状态 1：寄存器15中的扩展状态信息 0：在寄存器 15 中没有扩展状态信息
7	UNIDIRECTIONAL_ABLT	单向能力 1：PHY能够从MII/GMII传输，无论PHY是否确定已经建立了有效的链路 0：只有当PHY确定已经建立了有效的链路时，PHY才能从MII/GMII传输
6	MF_PRB_SUP	消除前导码能力 1：PHY可以接受带有前导码消除的管理帧 0：PHY不接受带有前导码消除的管理帧
5	MR_AUTONEG_CPLT	完成自协商 1：自协商完成 0：自协商未完成
4	REMOTE_FAULT	远程故障检测 1：远程故障 0：正常
3	AUTONEG_ABLT	自协商能力 1：PHY 可以自协商

		0：PHY 不可以自协商
2	LINK_STATUS	连接状态 1：建立连接 0：断开连接 注意：当PHY被设置为环回模式，***PHY控制寄存器（PHY_MII_CTL）***BIT14 (LOOPBACK_EN被设置为1)，PHY被强制连接。为了使MAC正常工作，链路状态应根据此位进行更新。 这个位是用一个高锁存功能来实现的，这样，链路故障条件的发生将导致这个位被清除并保持清除，直到它被读取。
1	JABBER_DETECT	检测到无效情况 1：检测到 RX 和 TX 无效情况 0：正常
0	EXTENDED_CAP	扩展寄存器功能 1：扩展寄存器功能 0：仅基本寄存器设置功能

PHY 标识寄存器（PHY_ID_REG）

地址偏移：0x02
复位值：0x0044

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

位/位域	名称	描述
15:0	PHY_ID [15:0]	PHY标识bit位[31-16] OUI (bits 3-18). OUI =00-11-05

PHY 版本寄存器（PHY_VER_REG）

地址偏移：0x03
复位值：0x1400

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

位/位域	名称	描述
15:10	PHY_ID[5:0]	PHY 标识 bit 位 [5-0] OUI bits 19-24
9:4	PHY_MANF[5:0]	制造商型号

制造商的型号(bit5-0)，其中[5:4]为架构版本

3:0 PHY_REV_NO[3:0] 版本号(bits3-0)寄存器 ***PHY 标识寄存器（PHY_ID_REG）***，bit 0 是 PHY 标识符 LS 位

自动协商广播寄存器（PHY_AUTONEG_ADV）

地址偏移：0x04

复位值：0x0101

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEXT_PAGE_EN	保留	REMOTE_FAULT_EN	EXTENED_NEXT_PAGE	ASYM_PAUSE	PAUSE	100BASE_T4	100BASE TX_FULL_DPX	100BASE TX_HALF_DPX	10BASE-T_FULL_DPX	10BASE-T_HALF_DPX	SLCT_FLD[4:0]				
rw		rw	rw	rw	rw	r	rw	rw	rw	rw	r				

位/位域	名称	描述
15	NEXT_PAGE_EN	使能下一页 1：设置使用下一页 0：未使用下一页
14	保留	必须保持复位值
13	REMOTE_FAULT_EN	使能远程故障检测 1：检测到自动协商故障 0：无远程故障
12	EXTENED_NEXT_PAGE	扩展下一页 PHY 不支持时，此位始终保持为 0
11	ASYM_PAUSE	非对称式暂停功能 A6 技术能力 1：有非对称式暂停能力 0：无非对称式暂停能力
10	PAUSE	暂停功能 A5 技术能力 1：有暂停能力 0：无暂停能力
9	100BASE_T4	100BASE-T4 功能 PHY 不支持时，此位始终保持为 0
8	100BASETX_FULL_DPX	100BASE-X 全双工功能 1：支持全双工 0：不支持全双工 注意：当自动协商被禁用时，该位上的值将反映在 ***PHY 控制寄存器(PHY_MII_CTL)*** BIT6，BIT13 (FORCE_SPEED) 和 BIT8(FORCE_DUPLEX) 中的值，当 FORCE_SPEED 为 01 和 FORCE_DUPLEX 为 1 时，该位将为 1，反之亦然。
7	100BASETX_HALF_DPX	100BASE-X 半双工功能

1：支持半双工
0：不支持半双工
注意：当自动协商被禁用时，该位上的值将反映在 ***PHY 控制寄存器（PHY_MII_CTL）*** BIT6，BIT13（FORCE_SPEED）和 BIT8(FORCE_DUPLEX) 中的值，当 FORCE_SPEED 为 01 和 FORCE_DUPLEX 为 0 时，该位将为 1，反之亦然。

6	10BASE-T_FULL_DPX	10BASE-T 全双工功能 1：支持全双工 0：不支持全双工 注意：当自动协商被禁用时，该位上的值将反映在 ***PHY 控制寄存器（PHY_MII_CTL）*** BIT6，BIT13（FORCE_SPEED）和 BIT8(FORCE_DUPLEX) 中的值，当 FORCE_SPEED 为 00 和 FORCE_DUPLEX 为 1 时，该位将为 1，反之亦然。
5	10BASE-T_HALF_DPX	10BASE-T 半双工功能 1：支持半双工 0：不支持半双工 注意：当自动协商被禁用时，该位上的值将反映在 ***PHY 控制寄存器（PHY_MII_CTL）*** BIT6，BIT13（FORCE_SPEED）和 BIT8(FORCE_DUPLEX) 中的值，当 FORCE_SPEED 为 00 和 FORCE_DUPLEX 为 0 时，该位将为 1，反之亦然。
4:0	SLCT_FLD[4:0]	识别消息类型 始终保持为 5'h01

自动协商链路伙伴(LP)能力寄存器（PHY_LP_ABILITY）

地址偏移：0x05
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEXT_PAGE	ACKNOWLEDGE	REMOTE_FAULT	EXTENED_NEXT_PAGE	ASYM_PAUSE	PAUSE	100BASET4	100BASETX_FULL_DPX	100BASETX_HALF_DPX	10BASE-T_FULL_DPX	10BASE-T_HALF_DPX	SELECTOR_FIELD[4:0]				
r	r	r	r	r	r	r	r	r	r	r	r				

位/位域	名称	描述
15	NEXT_PAGE	链路伙伴下一页请求 1：链路伙伴请求下一页功能 0：基本页请求
14	ACKNOWLEDGE	链路伙伴确认收到 1：链路伙伴确认收到成功 0：收到失败
13	REMOTE_FAULT	链路伙伴检测到远端故障 1：检测到自适应故障 0：无远端故障
12	EXTENED_NEXT_PAGE	扩展下一页

11	ASYM_PAUSE	链路伙伴非对称式暂停功能 A6 技术能力 1：有非对称式暂停能力 0：无非对称式暂停能力
10	PAUSE	链路伙伴对称暂停功能 A5 技术能力 1：有对称式暂停能力 0：无对称式暂停能力
9	100BASET4	A4 技术能力 链路伙伴 100BASE-T4 功能
8	100BASETX_FULL_DPX	链路伙伴 100BASE-X 全双工功能 1：支持全双工 0：不支持全双工 注意：当自适应关闭时，该bit的值始终为1
7	100BASETX_HALF_DPX	链路伙伴 100BASE-X 半双工功能 1：支持半双工 0：不支持半双工 注意：当自适应关闭时，该bit的值始终为1
6	10BASE-T_FULL_DPX	链路伙伴 10BASE-T 全双工功能 1：支持全双工 0：不支持全双工 注意：当自适应关闭时，该bit的值始终为1
5	10BASE-T_HALF_DPX	链路伙伴 10BASE-T 半双工功能 1：支持半双工 0：不支持半双工 注意：当自适应关闭时，该bit的值始终为1
4:0	SELECTOR_FIELD[4:0]	链路伙伴识别消息类型 始终保持为 5'h01

自动协商扩展寄存器（PHY_AUTONEG_EXP）

地址偏移：0x06
复位值：0x0064

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									NXT_PAGE_LOC_ABLE	NXT_PAGE_STO_LOC	PARA_DET_FAULT	LNK_PTNER_NP_ABLE	MR_NP_ABLE	PAGE_RECEIVED	LNK_PTNER_AN_ABLE
									r	r	r	r	r	r	r

位/位域	名称	描述

15:7	保留	必须保持复位值
6	NXT_PAGE_LOC_ABLE	接收下一页位置功能 1：收到下一页存储位置由bit 5确定 0：收到下一页存储位置不由bit 5确定
5	NXT_PAGE_STO_LOC	收到的下一页存储位置 1：链接伙伴下一页存储在*自动协商下一页接收寄存器（PHY_AUTONEG_NEXT_PAGE_RECEIVE）* 0：链接伙伴下一页存储在*自动协商链路伙伴(LP)能力寄存器（PHY_LP_ABILITY）*
4	PARA_DET_FAULT	Parallel Detect Fault并行故障检测 1：本地设备检测到并行故障 0：无故障
3	LNK_PTNER_NP_ABLE	链接伙伴下一页功能 1：链接伙伴是下一页的功能 0：链接伙伴不是下一页的功能
2	MR_NP_ABLE	下一页功能 1：前设备是下一页功能 0：当前设备不是下一页功能
1	PAGE_RECEIVED	收到新页面 1：已收到新页面 0：未收到新页面
0	LNK_PTNER_AN_ABLE	链接伙伴自适应功能 1：链接伙伴是自适应功能 0：链接伙伴不是自适应功能

自动协商下一页发送寄存器（PHY_AUTONEG_NEXT_PAGE_TRANSMIT）

地址偏移：0x07

复位值：0x2001

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEXT_PAGE	保留	MESSAGE_PAGE	ACKNOWLEDGE2	TOGGLE	MSG_UFMT_CODE_FIELD[10:0]										
rw		rw	rw	r	rw										

位/位域	名称	描述
15	NEXT_PAGE	最后一页 证明这是否是最后一页 1：附加的下一页在这之后 0：下一页是最后一页
14	保留	必须保持复位值

13	MESSAGE_PAGE	消息页或未格式化页 指示这是消息页或未格式化的页 1：消息页 0：未格式化页
12	ACKNOWLEDGE2	执行信息功能 1：执行信息 0：不执行信息
11	TOGGLE	切换 切换位由硬件自动计算，软件可忽略
10:0	MSG_UFMT_CODE_FIELD[10:0]	消息/未格式化代码字段

自动协商下一页接收寄存器（PHY_AUTONEG_NEXT_PAGE_RECEIVE）

地址偏移：0x08
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

位/位域	名称	描述
15:0	NP_RX[15:0]	从链路伙伴接收到下一页

MMD 访问控制寄存器（MMD_CTL）

地址偏移：0x0D
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）进行访问。

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

位/位域	名称	描述
15:14	FUN[1:0]	功能 00：地址 01：数据，没有后增量 10：数据，读和写的后增量 11：数据，只对写操作进行后增量
13:5	保留	必须保持复位值
4:0	DEVAD[4:0]	设备地址

MMD 访问数据地址寄存器（MMD_ADDR_DATA）

地址偏移：0x0E
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

位/位域	名称	描述
15:0	ADDR_DATA[15:0]	数据地址 当bit 13.15:14==0时，为地址寄存器，否则为数据寄存器。

PHY 扩展状态寄存器（PHY_EXTENDED_STATUS）

地址偏移：0x0F
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

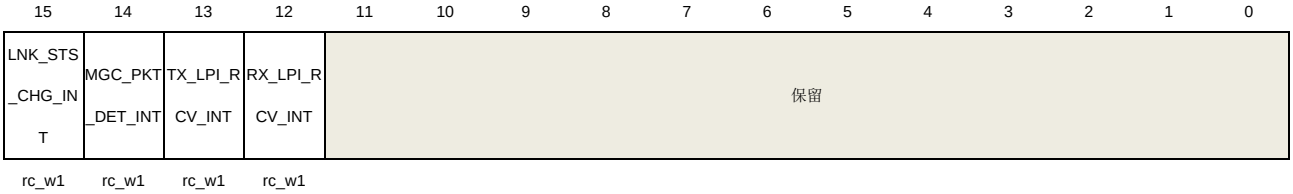
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1000BASE-X_FULL_DPX	1000BASE-X_HALF_DPX	1000BASE-T_HALF_DPX	1000BASE-T_HALF_DPX	保留											
r	r	r	r												

位/位域	名称	描述
15	1000BASE-X_FULL_DPX	1000BASE TX全双工功能 1：PHY是1000BASE-X全双工 0：PHY不是1000BASE-X全双工
14	1000BASE-X_HALF_DPX	1000BASE TX半双工功能 1：PHY是1000BASE-X半双工 0：PHY不是1000BASE-X半双工
13	1000BASE-T_HALF_DPX	1000BASE-T全双工功能 1：PHY是1000BASE-T全双工 0：PHY不是1000BASE-T全双工
12	1000BASE-T_HALF_DPX	1000BASE-T半双工功能 1：PHY是1000BASE-T半双工 0：PHY不是1000BASE-T半双工
11:0	保留	必须保持复位值。

中断状态寄存器（INT_STS）

地址偏移：0x10
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

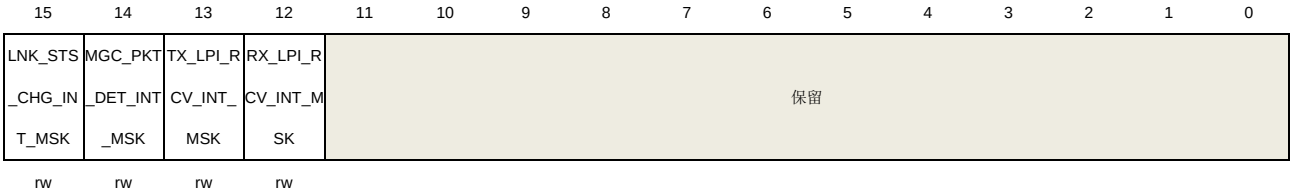


位/位域	名称	描述
15	LNK_STS_CHG_INT	连接状态变化中断 0：正常 1：连接状态改变
14	MGC_PKT_DET_INT	魔术包检测中断 0：正常 1：检测到魔术包
13	TX_LPI_RCV_INT	TX LPI接收中断 0：正常 1：TX LPI接收
12	RX_LPI_RCV_INT	RX LPI接收中断 0：正常 1：RX LPI接收
11:0	保留	必须保持复位值。

中断掩码寄存器（INT_MASK）

地址偏移：0x11
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）进行访问。



位/位域	名称	描述
15	LNK_STS_CHG_INT_MSK	连接状态变化中断掩码 当此位设置为1且INT_STS寄存器中的LINK_STS_CHG_INT位被设置为1，INT_SMI_INT_N/EXT_SMI_INT_N接口被设置为0(低电平触发)，表示发生了连接状态变化中断。

1：正常
0：中断掩码

14	MGC_PKT_DET_INT_MSK	魔术包检测中断掩码 当此位设置为1且INT_STS寄存器中的MGC_PKT_DET_INT位被设置为1，INT_SMI_INT_N/EXT_SMI_INT_N接口被设置为0(低电平触发)，表示发生了魔术包检测中断。 1：正常 0：中断掩码
13	TX_LPI_RCV_INT_MSK	TX LPI接收中断掩码 当此位设置为1且INT_STS寄存器中的TX_LPI_RCV_INT位被设置为1，INT_SMI_INT_N/EXT_SMI_INT_N接口被设置为0(低电平触发)，表示发生了TX LPI接收中断。 1：正常 0：中断掩码
12	RX_LPI_RCV_INT_MSK	RX LPI接收中断掩码 当此位设置为1且int_sts寄存器中的RX_LPI_RCV_INT位被设置为1，INT_SMI_INT_N/EXT_SMI_INT_N接口被设置为0(低电平触发)，表示发生了RX LPI接收中断。 1：正常 0：中断掩码
11:0	保留	必须保持复位值。

环回控制寄存器（PHY_LB_CTL）

地址偏移：0x12
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）进行访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RG_LB_XMII2MAC	保留	RG_LB_PCS2MAC	RG_LB_PMA2MAC	RG_LB_AFE2MAC	RG_LB_EPG2EPC	保留		RG_LB_MII2PHY	保留						
rw	rw	rw	rw	rw	rw			rw							

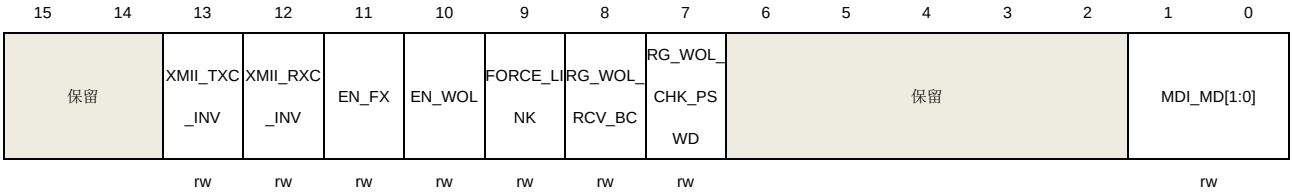
位/位域	名称	描述
15	RG_LB_XMII2MAC	XMII2MAC环回使能
14	保留	必须保持复位值
13	RG_LB_PCS2MAC	PCS2MAC环回使能
12	RG_LB_PMA2MAC	PMA2MAC环回使能
11	RG_LB_AFE2MAC	AFE2MAC环回使能
10	RG_LB_EPG2EPC	EPG2EPC环回使能

9:8	保留	必须保持复位值
7	RG_LB_MMII2PHY	RMII2PHY 环回使能 仅在使用内部phy时有效。
6:0	保留	必须保持复位值。

PHY 全局配置寄存器（PHY_GLOBAL_CONFIG）

地址偏移：0x13
复位值：0x0102

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



位/位域	名称	描述
15:14	保留	必须保持复位值。
13	XMII_TXC_INV	XMII TXCLK 反转 0：TXCLK 在 XMII 接口上没有反转 1：TXCLK在XMII接口上反转
12	XMII_RXC_INV	XMII RXCLK 反转 0：RXCLK 在 XMII 接口上没有反转 1：RXCLK在XMII接口上反转
11	EN_FX	光纤使能 该位将与 EPHY_FXEN 进行 OR 处理以确定 EPHY 的介质类型。下面显示了 OR 'ed 结果的定义 1：光纤模式 0：双绞线模式 注意：当 EPHY 被编程为光纤模式时，Auto MDIX 应该被自动禁用，并强制进入 MDI 模式
10	EN_WOL	使能 Wake-On-Lan Wake-On-Lan
9	FORCE_LINK	强制建立连接 1：强制两个10/100M模块连接 0：是否强制建立连接分别取决于10/100M配置寄存器中的FORCE_LINK_10和/或FORCE_LINK_100 注意：FORCE LINK仅在PHY编程为强制10或100M模式时有效。当PHY被编程为自动协商模式时，程序1'b1到该位没有作用(如果在自动协商开启之前将该位编程为1'b1，则H/W应自动禁用该位)

8	RG_WOL_RCV_BC	使能WOL模式接收广播魔术包 1：使能接收广播魔术包 0：其他
7	RG_WOL_CHK_PSWD	使能WOL模式下的SecureOn密码检查功能 1：使能SecureOn密码检查 0：其他
6:2	保留	必须保持复位值
1:0	MDI_MD[1:0]	MDI/MDIX 模式 DUPCOLLED 和 RXER 将会被锁存 上电复位并存储在这两个 bit 位 00：强制 MDI 模式 01：模式 10：自动 MDI/MDIX 检测(默认) 11：保留 当 PHY 处于环回模式或光纤模式时，MDI/MDIX 模式将自动设置为“强制 MDI 模式”

MAC 地址寄存器 0（RG_MAC_AADR_0）

地址偏移：0x16
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAC_ADDR_BYTE0[7:0]								MAC_ADDR_BYTE1[7:0]							
rw								rw							

位/位域	名称	描述
15:8	MAC_ADDR_BYTE0[7:0]	传输顺序中的 MAC 地址 Byte0
7:0	MAC_ADDR_BYTE1[7:0]	传输顺序中的 MAC 地址 Byte1

MAC 地址寄存器 1（RG_MAC_AADR_1）

地址偏移：0x17
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）进行访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAC_ADDR_BYTE2[7:0]								MAC_ADDR_BYTE3[7:0]							
rw								rw							

位/位域	名称	描述
15:8	MAC_ADDR_BYTE2[7:0]	传输顺序中的 MAC 地址 Byte2
7:0	MAC_ADDR_BYTE3[7:0]	传输顺序中的 MAC 地址 Byte3

MAC 地址寄存器 2（RG_MAC_AADR_2）

地址偏移：0x18
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAC_ADDR_BYTE4[7:0]								MAC_ADDR_BYTE5[7:0]							
rw								rw							

位/位域	名称	描述
15:8	MAC_ADDR_BYTE4[7:0]	传输顺序中的 MAC 地址 Byte4
7:0	MAC_ADDR_BYTE5[7:0]	传输顺序中的 MAC 地址 Byte5

PHY 状态寄存器（PHY_STATUS）

地址偏移：0x19
复位值：0x8800

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINE_NOSIG	保留		MR_AUTONEG_CPLT	A_TXSAVE80	A_TXSAVE50	JABTX_STATUS	MDIX_STATUS	mac_addr_byte5	POLARITY_REV	FINAL_PAUSE_OUT	FINAL_PAUSE_IN	FINAL_DUPLEX	FINAL_SPEED[1:0]		FINAL_LINK
r			r	r	r	r	r	r	r	r	r	r	r		r

位/位域	名称	描述
15	LINE_NOSIG	无媒介信号 0：媒介中检测到信号 1：媒介中未检测到信号
14:13	保留	必须保持复位值
12	MR_AUTONEG_CPLT	完成自协商 1：完成 0：未完成
11	A_TXSAVE80	10M 发送 80%振幅状态 1：80%TX 振幅状态 0：正常振幅
10	A_TXSAVE50	10M 发送 50%振幅状态 1：50%TX 振幅状态 0：正常振幅
9	JABRX_STATUS	实时 RX Jabber 状态 1：RX Jabber

0：没有 RX Jabber

8	JABTX_STATUS	实时 TX Jabber 状态 1：TX Jabber 0：没有 RX Jabber
7	MDIX_STATUS	MDIX 状态 1：MDIX 0：MDI
6	POLARITY_REV	极性状态 1：保留 0：正常
5	FINAL_PAUSE_OUT	暂停输出功能 启用自动协商后，该位由 *自动协商广播寄存器（PHY_AUTONEG_ADV）*BIT [11:10] 和*自动协商链路伙伴(LP)能力寄存器（PHY_LP_ABILITY）*BIT[11:10]在连接后确定。 当自动协商被禁用时，这个位始终被设置为 0。 1：存在暂停输出功能 0：不存在暂停输出功能
4	FINAL_PAUSE_IN	暂停输入功能 启用自动协商后，该位由*自动协商广播寄存器（PHY_AUTONEG_ADV）*BIT [11:10] 和*自动协商链路伙伴(LP)能力寄存器（PHY_LP_ABILITY）*BIT[11:10]在连接后确定。 当自动协商被禁用时，这个位始终被设置为 0。 1：存在暂停输入功能 0：不存在暂停输入功能
3	FINAL_DUPLEX	双工状态 在连接之前，S/W 应该忽略这个状态此时无意义 1：全双工 0：半双工
2:1	FINAL_SPEED[1:0]	速度状态 在连接之前，S/W 应该忽略这个状态此时无意义 1：100M 0：10M
0	FINAL_LINK	连接状态 1：建立连接 0：断开连接

Wake-On-Lan 安全密码寄存器 0（RG_WOL_PASSWORD_0）

地址偏移：0x1A
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

RG_WOL_PASSWORD_BYTE0[7:0]	RG_WOL_PASSWORD_BYTE1[7:0]
rw	rw

位/位域	名称	描述
15:8	RG_WOL_PASSWORD_BYTE0[7:0]	SecureON 传输顺序中的密码 Byte0
7:0	RG_WOL_PASSWORD_BYTE1[7:0]	SecureON 传输顺序中的密码 Byte1

Wake-On-Lan 安全密码寄存器 1（RG_WOL_PASSWORD_1）

地址偏移：0x1B
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RG_WOL_PASSWORD_BYTE2[7:0]								RG_WOL_PASSWORD_BYTE3[7:0]							
rw								rw							

位/位域	名称	描述
15:8	RG_WOL_PASSWORD_BYTE2[7:0]	SecureON 传输顺序中的密码 Byte2
7:0	RG_WOL_PASSWORD_BYTE3[7:0]	SecureON 传输顺序中的密码 Byte3

Wake-On-Lan 安全密码寄存器 2（RG_WOL_PASSWORD_2）

地址偏移：0x1C
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RG_WOL_PASSWORD_BYTE4[7:0]								RG_WOL_PASSWORD_BYTE5[7:0]							
rw								rw							

位/位域	名称	描述
15:8	RG_WOL_PASSWORD_BYTE4[7:0]	SecureON 传输顺序中的密码 Byte4
7:0	RG_WOL_PASSWORD_BYTE5[7:0]	SecureON 传输顺序中的密码 Byte5

Page 选择寄存器（PHY_PAGE_SEL）

地址偏移：0x1F
复位值：0x003D

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

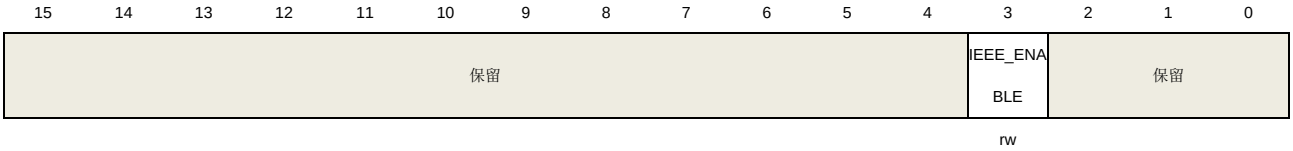
位/位域	名称	描述
15:13	保留	必须保持复位值
12:8	PAGE_SEL[4:0]	页选择寄存器
7:0	保留	必须保持复位值

Page1 寄存器

EEE 配置寄存器（EEE_CFG）

地址偏移：0x17
复位值：0x0033

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



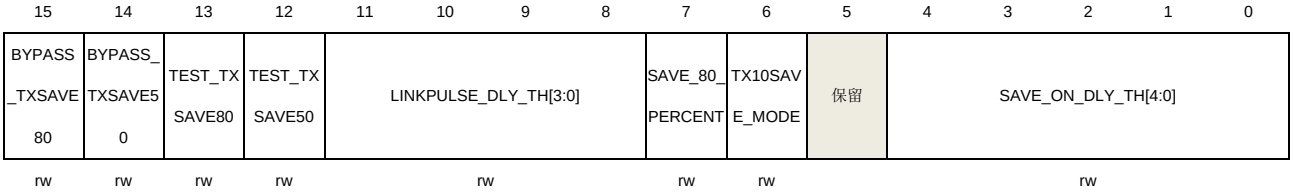
位/位域	名称	描述
15:4	保留	必须保持复位值。
3	IEEE_ENABLE	使能 EEE 使能 EPHY TX 在无数据传输时自动进入 LPI 状态
2:0	保留	必须保持复位值。

Page2 寄存器

10M 节能控制寄存器（PHY_10M_PWRSAVE）

地址偏移：0x17
复位值：0x04C8

该寄存器可以按字节（8位）、半字（16位）和字（32位）进行访问。



位/位域	名称	描述
15	BYPASS_TXSAVE80	分流器正常省电 80%通道
14	BYPASS_TXSAVE50	分流器正常省电 50%通道
13	TEST_TXSAVE80	省电 80%测试输入 仅 BYPASS_TXSAVE80 使能时有效

12	TEST_TXSAVE50	省电 50%测试输入 仅 BYPASS_TXSAVE50 使能时有效
11:8	LINKPULSE_DLY_TH[3:0]	链路脉冲延时产生阈值 TX 链路脉冲延迟传输的阈值。 延迟的时间是编程的值 x 2 个周期，其中 1 个周期等于 40ns。 这里设置的默认值是延迟 320ns
7	SAVE_80_PERCENT	省电 80% 1：使能 0：未使能
6	TX10SAVE_MODE	10M Power Saving Mode10M 省电模式 1：10M 省电模式使能 0：正常模式没有省电
5	保留	必须保持复位值
4:0	SAVE_ON_DLY_TH[4:0]	省电模式关于延时阈值

模拟传输数据测试和控制寄存器（PHY_TXDATA_CTRL）

地址偏移：0x18
复位值：0x1000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14 13 12	11 10	9	8	7 6	5	4 3 2 1 0
保留	FIR_10_SEL[2:0]	保留	BYPASS_TX_10_DATA	BYPASS_TX_10_DATA	TEST_TX_100_DATA[1:0]	保留	TEST_TX_10_DATA [4:0]
	rw		rw	rw	rw		rw

位/位域	名称	描述
15	保留	必须保持复位值
14:12	FIR_10_SEL[2:0]	10M TX 滤波选择
11:10	保留	必须保持复位值
9	BYPASS_TX_100_DATA	分流器 100M 数据传输 1：将 100M 分流器数据设为模拟并强制 TEST_TX_100_DATA 为模拟 0：正常 100M 数据发送到模拟块
8	BYPASS_TX_10_DATA	分流器 10M 数据传输 1：将 10M 分流器数据设为模拟并强制 TEST_TX_10_DATA 为模拟 0：正常 10M 数据发送到模拟块
7:6	TEST_TX_100_DATA[1:0]	100M 测试数据 当 BYPASS_TX_100_DATA 设置为 1 时，反馈到模拟块

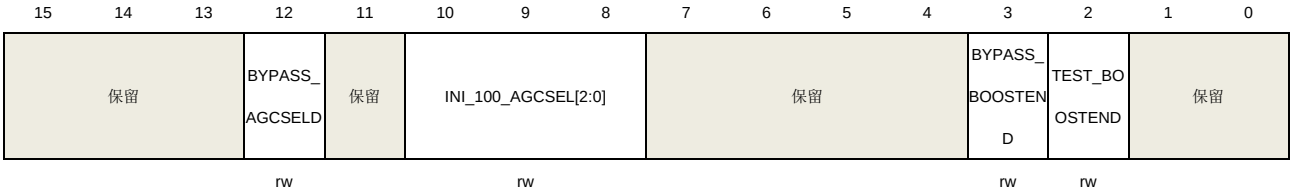
5	保留	必须保持复位值
4:0	TEST_TX_10_DATA[4:0]	10M 测试数据 当 BYPASS_TX_10_DATA 设置为 1 时，反馈到模拟块

Page3 寄存器

DSPSM 控制寄存器（PHY_DSPSM_CTRL）

地址偏移：0x11
复位值：0x8510

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



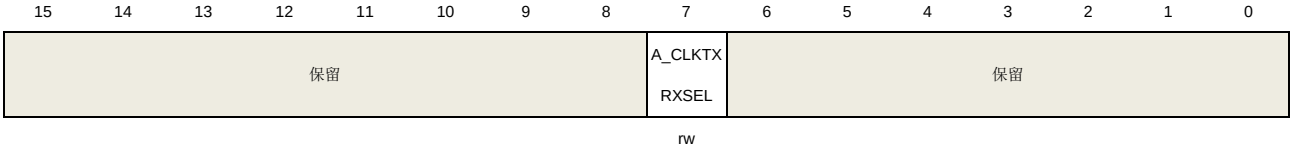
位/位域	名称	描述
15:13	保留	必须保持复位值
12	BYPASS_AGCSELD	AGCSEL 分流器 1：分流器正常 agcselD 0：正常
11	保留	必须保持复位值
10:8	INI_100_AGCSEL[2:0]	100M 模式 AGCSEL 初始值 这 3 位的功能根据 BYPASS_AGCSELD BYPASS 的设置不同 AGCSELD=1:内部 AGCSEL 固定为 INI_100_AGCSEL BYPASS AGCSELD=0:在 DSPRST 和内部 boost 为 0 时加载 INI_100_AGCSEL，当内部 boost 为低时保持不变。因此， DSPSM 可以覆盖[100,000]和[1,000]到[1,111]范围内的[boosten, AGCSEL]。
7:4	保留	必须保持复位值
3	BYPASS_BOOSTEND	BoostenD 分流器模式 内部 BoostenD 将被跳过和 TEST_BOOSTEND 将用于调试目的的控制 1：分流器模式 0：正常模式
2	TEST_BOOSTEND	BoostenD 测试输入 当 BYPASS_BOOSTEND=1 时，强制 BoostenD 为该值
1:0	保留	必须保持复位值

Page6 寄存器

模拟 ADC 控制寄存器（PHY_ADC_CTL）

地址偏移：0x10
复位值：0x5563

该寄存器可以按字节（8位）、半字（16位）和字（32位）进行访问。

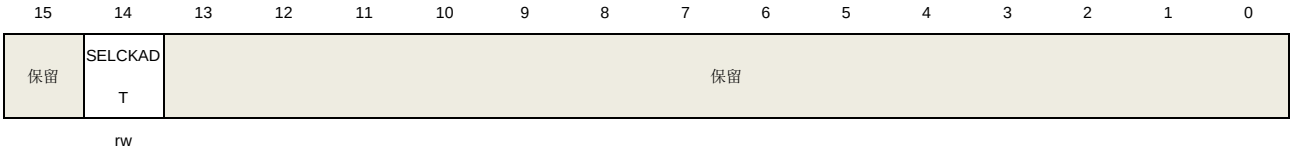


位/位域	名称	描述
15:8	保留	必须保持复位值
7	A_CLKTXRXSEL	RXCLK125 到 TXCLK125 分流器 1：内部 RXCLK125 选择 TXCLK125 0：内部 RXCLK125 选择 RXCLK125(默认)
6:0	保留	必须保持复位值

模拟 Pre-Gain 和 PLL 配置寄存器（PHY_PGPLL_CTL）

地址偏移：0x12
复位值：0x0D00

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



位/位域	名称	描述
15	保留	必须保持复位值
14	SELCKADT	测试模式 ADC 时钟选择 该位仅在 PHY 处于 AFE 测试模式时可用，非 AFE 测试模式时，向该位写入值无效 1：选择 CKADTEST 作为 ADC 输入时钟 0：选择 RXCLK125 作为 ADC 输入时钟
13:0	保留	必须保持复位值。

Page9 寄存器

内置包生成器和校验命令寄存器（EPGC_CMD）

地址偏移：0x10
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）进行访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RG_EPG_EN	RG_EPG_DATA_TYPE[2:0]			RG_EPG_IPG_LEN_FIXED	RG_EPG_DASA_FIXED	RG_EPG_MODE[1:0]		RG_EPG_LPI_INDICATION	RG_EPC_LOC	保留		RG_EPC_CLR_CNT	RG_EPG_CLR_CNT	RG_EPG_PAUSE	RG_EPG_GO
rw	rw	rw	rw	rw	rw	rw		rw	rw			rw	rw	rw	rw

位/位域	名称	描述
15	RG_EPG_EN	使能内置包发生器 此位用于使能内置包生成器以进行调试。当它被使用时，内部tx数据路径将切换到内置包生成器
14:12	RG_EPG_DATA_TYPE[2:0]	内置包发生器模式 3'b000：全0； 3'b001：全1； 3'b010：全5s； 3'b011：全As； 3'b100：字节增量； 3'b101：随机； 3'b110：字节减量；
11	RG_EPG_IPG_LEN_FIXED	固定包间间隙长度 1：报文间隙长度由RG_EPG_IPG_LEN确定 0：包间间隙长度随机由硬件
10	RG_EPG_DASA_FIXED	固定DA/SA 1：DA固定为00-01-02-03-04-05和SA固定为0a-0b-0c-0d-0e-0f 0：DA/SA 由 RG EPG DATA TYPE 配置
9:8	RG_EPG_MODE[1:0]	内置包发生器模式 2'b00：单发模式； 2'b01：突发模式； 2'b1x：持续模式；
7	RG_EPG_LPI_INDICATION	TX LPI指示 1：传输LPI 0：正常
6	RG_EPC_LOC	EPC位置 1：EPC位于TX路径 0：EPC 位于 RX 路径
5:4	保留	必须保持复位值
3	RG_EPC_CLR_CNT	内置包检查计数器清除 高位有效清除 rx 包检查统计计数器，包括总包数计数器和 crc 错误计数器，它是自清除。
2	RG_EPG_CLR_CNT	内置包生成计数器 清除高位有效清除tx包生成统计计数器，包括总包生成计数器，它是自清

除。

1	RG_EPG_PAUSE	内置包生成器结合RG_EPG_GO控制包生成，{ RG_EPG_PAUSE, RG_EPG_GO } = 2'b01：开始 2'b11：暂停 2'b00：结束 注意：在单发模式下(RG_EPG_MODE[1:0]= 2'b0)，只有start命令有效。在突发模式下(RG_EPG_MODE[1:0]= 2'b01)，这三个命令都是有效的，pause命令将暂停包的生成，内部突发包数计数器将保持，随后的start命令将继续当前突发的生成。在继续模式下(RG_EPG_MODE[1:0]= 2'b10)，所有三个命令都是有效的，并且暂停命令的行为将与停止命令相同。
0	RG_EPG_GO	内置包生成器结合RG_EPG_PAUSE控制包生成，请参考RG_EPG_PAUSE获取控制命令定义。 注意：在单发模式和连续模式下，当包生成任务完成时，该位将自动清除。当处于连续模式时，只有将0写入该位才能清除它。

内置包生成器包长度寄存器（EPG_PKT_LEN）

地址偏移：0x11
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

位/位域	名称	描述
15:12	保留	必须保持复位值
11	RG_EPG_PKT_LEN_FIXED	固定报文长度使能 1：数据包长度固定，由 rg_epg_pkt_len 决定 0：包长度随机由硬件决定
10:0	RG_EPG_PKT_LEN[10:0]	报文长度 包的总长度，包括 DA/SA，数据和 FCS。只有当 rg_epg_pkt_len_fixed 为 1(按字节计数)时才有效，否则，数据包长度将随机由硬件决定

内置包生成器包间间隔寄存器 (EPG_IPG_CFG)

地址偏移：0x12
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）进行访问。

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

RG_EPG_IPG_LEN [15:0]

rw

位/位域	名称	描述
15:0	RG_EPG_IPG_LEN [15:0]	Inter-Packet-Gap 长度

内置包生成器突发数量高位数据寄存器(EPG_BURST_PKT_NUM_CFG_HIGH)

地址偏移：0x13
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

rw

位/位域	名称	描述
15:0	RG_EPG_BURST_PKT_NUMBER_HIGH [15:0]	突发高位数据只在突发模式下有效(RG_EPG_MODE==10)

内置包生成器突发数量低位数据寄存器(EPG_BURST_PKT_NUM_CFG_LOW)

地址偏移：0x14
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

rw

位/位域	名称	描述
15:0	RG_EPG_BURST_PKT_NUMBER_LOW [15:0]	突发低位数据只在突发模式下有效(RG_EPG_MODE==10)

TX 字节计数器高位数据寄存器（TX_BYTE_CNT_HIGH）

地址偏移：0x15
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

r

位/位域	名称	描述
15:0	TX_BYTE_CNT_HIGH[15:0]	内置包发生器包生成字节计数器高位数据(bit31:16)

该计数器将通过 RG_EPG_CLR_CNT 来清除

TX 字节计数器低位数据寄存器（TX_BYTE_CNT_LOW）

地址偏移：0x16
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

位/位域	名称	描述
15:0	TX_BYTE_CNT_LOW[15:0]	内置包发生器包生成字节计数器低位数据(bit 15:0) 该计数器将通过 RG_EPG_CLR_CNT 来清除

TX 总数据包计数器高位寄存器（TX_PKT_CNT_HIGH）

地址偏移：0x17
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

位/位域	名称	描述
15:0	TX_PKT_CNT_HIGH[15:0]	内置包生成器总包生成计数器高位数据(bit31:16) 该计数器将通过 RG_EPG_CLR_CNT 来清除

TX 总数据包计数器低位寄存器（TX_PKT_CNT_LOW）

地址偏移：0x18
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

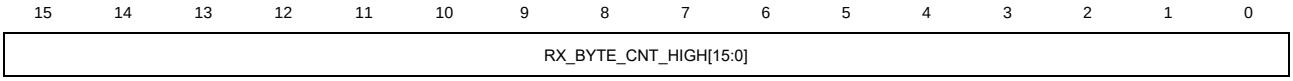
位/位域	名称	描述
15:0	TX_PKT_CNT_LOW[15:0]	内置包生成器总包生成计数器低位数据(bit15:0) 该计数器将通过 RG_EPG_CLR_CNT 来清除

RX 字节计数器高位数据寄存器（RX_BYTE_CNT_HIGH）

地址偏移：0x19

复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



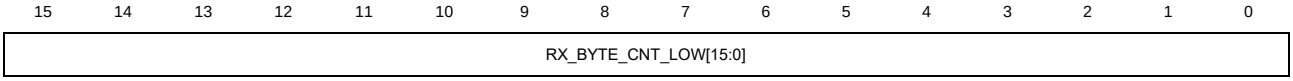
r

位/位域	名称	描述
15:0	RX_BYTE_CNT_HIGH[15:0]	数据包接收字节计数器高位数据(bit31:16) 该计数器将通过 RG_EPC_CLR_CNT 来清除

RX 字节包计数器低位数据寄存器（RX_BYTE_CNT_LOW）

地址偏移：0x1A
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）进行访问。



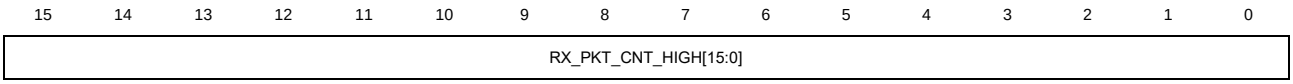
r

位/位域	名称	描述
15:0	RX_BYTE_CNT_LOW[15:0]	数据包接收字节计数器低位数据(bit15:0) 该计数器将通过 RG_EPC_CLR_CNT 来清除

RX 总数据包计数器高位寄存器（RX_PKT_CNT_HIGH）

地址偏移：0x1B
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



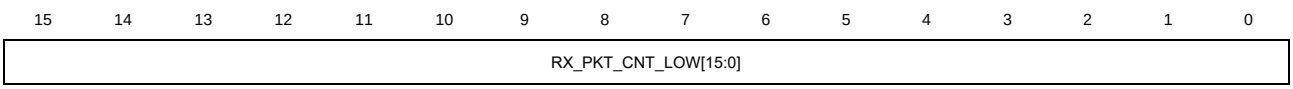
r

位/位域	名称	描述
15:0	RX_PKT_CNT_HIGH[15:0]	总包接收计数器高位数据(bit31:16) 该计数器将通过 RG_EPC_CLR_CNT 来清除

RX 总数据包计数器低位寄存器（RX_PKT_CNT_LOW）

地址偏移：0x1C
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



r

位/位域	名称	描述
15:0	RX_PKT_CNT_LOW[15:0]	总包接收计数器低位数据(bit15:0) 该计数器将通过 RG_EPC_CLR_CNT 来清除

RX CRC 错误数据包计数器高位数据寄存器（RX_CRC_ERR_CNT_HIGH）

地址偏移：0x1D
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）进行访问。

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

位/位域	名称	描述
15:0	RX_CRC_ERR_CNT_HIGH[15:0]	错误包接收计数器高位数据(bit31:16) 该计数器将通过 RG_EPC_CLR_CNT 来清除

RX CRC 错误数据包计数器低数据寄存器（RX_CRC_ERR_CNT_LOW）

地址偏移：0x1E
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

位/位域	名称	描述
15:0	RX_CRC_ERR_CNT_LOW[15:0]	错误包接收计数器低位数据(bit15:0) 该计数器将通过 RG_EPC_CLR_CNT 来清除

MDIO 寄存器

PCS 控制寄存器 1（PCS_CTL_1）

设备地址：0x3
地址偏移：0x00
复位值：0x0400

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

位/位域	名称	描述
15	PCS_RST	PCS 复位 将 AN 和 PCS MMD 寄存器复位为默认值，并在 PHY 中引起软件复位，复位完成后将自动清除
14:11	保留	必须保持复位值
10	CLOCK_STOP_EN	使能接收时钟停止 设置为 1 表示在发送 LPI 信号时停止接收 xMII 时钟，否则它将保持时钟活动
9:0	保留	必须保持复位值

PCS 状态寄存器 1（PCS_STS_1）

设备地址：0x3
地址偏移：0x01
复位值：0x0040

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				TX_LPI_REC	RX_LPI_REC	TX_LPI_INT	RX_LPI_INT	保留	CLOCK_STOP_CAP	保留					
				r	r	r	r		r						

位/位域	名称	描述
15:12	保留	必须保持复位值
11	TX_LPI_REC	TX PCS 收到 LPI 它是用一个锁存高功能实现的，这样 TX LPI 信号使这个位变成 1 并保持 1 直到它被读取。 1：TX PCS 收到 LPI 0：LPI 未接收
10	RX_LPI_REC	RX PCS 收到 LPI 它是用一个锁存高功能实现的，这样 RX LPI 信号使这个位变成 1 并保持 1 直到它被读取。 1：RX PCS 收到 LPI 0：LPI 未接收
9	TX_LPI_INT	TX PCS 接收 LPI 指示 1：TX PCS 目前正在接收 LPI 0：PCS 目前没有收到 LPI
8	RX_LPI_INT	RX PCS 接收 LPI 指示 1：RX PCS 目前正在接收 LPI 0：PCS 目前没有收到 LPI

7	保留	必须保持复位值
6	CLOCK_STOP_CAP	传输 xMII 时钟停止能力 1：RS 可能会在 LPI 期间停止发送 xMII 时钟 0：传输 xMII 时钟不可停止
5:0	保留	必须保持复位值

PCS 设备标识符寄存器（PCS_ID）

设备地址：0x3
地址偏移：0x02
复位值：0x0044

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

位/位域	名称	描述
15:0	PHY_ID[15:0]	PHY 标识 bit[31-16] OUI (bits 3-18) .OUI =00-11-05

PCS 设备版本寄存器（PCS_VER）

设备地址：0x3
地址偏移：0x03
复位值：0x1400

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

位/位域	名称	描述
15:10	PHY_ID[5:0]	PHY 标识 bit[15-10] OUI bits 19-24
9:4	PHY_MAN_MODEL[5:0]	制造商型号 制造商的型号(bits 5-0)，其中[5:4]=架构版本
3:0	PHY_REV_NO[3:0]	版本号(bits3-0) *PCS 设备标识符寄存器（PCS_ID）* bit 0 是 PHY 标识符的 LS 位

PCS 数据包寄存器 0（PCS_PKG_0）

设备地址：0x3
地址偏移：0x05

复位值：0x0089

该寄存器可以按字节（8位）、半字（16位）和字（32位）进行访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								AN_PST	TC_PST	DTE_XS_PST	PHY_XS_PST	PCS_PST	WIS_PST	PMD/PMA_PST	CLAUSS22_REG_PST
								r	r	r	r	r	r	r	r

位/位域	名称	描述
15:8	保留	必须保持复位值
7	AN_PST	自动协商存在于包中 始终为 1
6	TC_PST	TC 存在包中 始终为 0
5	DTE_XS_PST	DTE XS 存在包中 始终为 0
4	PHY_XS_PST	PHY XS 存在包中 始终为 0
3	PCS_PST	PCS 存在包中 始终为 0
2	WIS_PST	WIS 存在包中 始终为 0
1	PMD/PMA_PST	PMD/PMA 存在包中 始终为 0
0	CLAUSS22_REG_PST	自动协商存在于包中 始终为 1

PCS 数据包寄存器 1（PCS_PKG_1）

设备地址：0x3
地址偏移：0x06
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

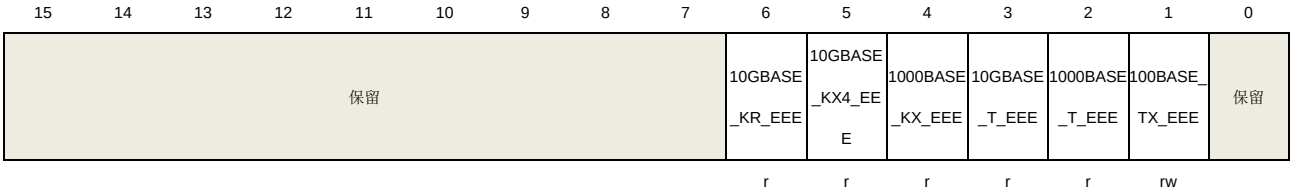
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VENDER_SPEC_DEV2_PST	VENDER_SPEC_DEV1_PST	CL22_EXT_PST	保留												
r	r	r													

位/位域	名称	描述
15	VENDER_SPEC_DEV2_PST	特殊设备 2 存在于包中 始终为 0
14	VENDER_SPEC_DEV1_PST	特殊设备 1 存在于包中 始终为 0
13	CL22_EXT_PST	Clause 22 扩展在于包中 始终为 0
12:0	保留	必须保持复位值

EEE 能力寄存器（EEE_CAP）

设备地址：0x3
地址偏移：0x14
复位值：0x0002

该寄存器可以按字节（8位）、半字（16位）和字（32位）进行访问。



位/位域	名称	描述
15:7	保留	必须保持复位值
6	10GBASE_KR_EEE	10GBASE-KR EEE 不支持，始终为 0
5	10GBASE_KX4_EEE	10GBASE-KR EEE 不支持，始终为 0
4	1000BASE_KX_EEE	10GBASE-KR EEE 不支持，始终为 0
3	10GBASE_T_EEE	10GBASE-KR EEE 不支持，始终为 0
2	1000BASE_T_EEE	1000Base-T EEE 1：EEE支持1000BASE-T 0：EEE不支持1000Base-T
1	100BASE_TX_EEE	100Base-TX EEE 1：EEE支持100BASE-TX 0：EEE不支持100BASE-TX

0	保留	必须保持复位值

EEE 唤醒错误计数器寄存器（EEE_WAKE_ERR_CNT）

设备地址：0x3
地址偏移：0x16
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

位/位域	名称	描述
15:0	EEE_WAKE_ERR_CNT[15:0]	EEE唤醒错误计数器 用于统计PHY未在指定类型的时间内完成正常的序列时的时间故障。它被读取时被清除

AN 控制寄存器（AN_CTL）

设备地址：0x7
地址偏移：0x00
复位值：0x1000

该寄存器可以按字节（8位）、半字（16位）和字（32位）进行访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AN_RST	保留	XNP_CTL	AN_EN	保留		RESTART_AUTONEG	保留								
rw		r	rw			rw									

位/位域	名称	描述
15	AN_RST	AN 复位 将 AN 和 PCS MMD 寄存器复位为默认值，并在 PHY 中引起软件复位，复位完成后将自动清除
14	保留	必须保持复位值
13	XNP_CTL	扩展下一页控制 0 不支持，保留 0
12	AN_EN	使能自动协商 它是 bit12 在 ***PHY 控制寄存器（PHY_MII_CTL）*** 1：使能自动协商 0：未使能自动协商
11:10	保留	必须保持复位值

9	RESTART_AUTONEG	重启自动协商 它是 bit9 在 ***PHY 控制寄存器（PHY_MII_CTL）***，该位为自清除 1：重启自动协商 0：正常运行
8:0	保留	必须保持复位值

AN 状态寄存器（AN_STS）

设备地址：0x7
地址偏移：0x01
复位值：0x0008

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									PAGE_RECEIVED	AN_COMPLETE	AN_ABILITY	AN_ABILITY	LINK_STATUS	保留	LP_AN_ABILITY
									r	r	r	r	r		r

位/位域	名称	描述
15:7	保留	必须保持复位值
6	PAGE_RECEIVED	页接收 设置为1表示已经接收到一个新的链接码字并将其存储在an_lp_xnp_ability寄存器中。在读取AN状态寄存器或自动协商扩展寄存器时，页面接收位将复位为0。它是***自动协商扩展寄存器（PHY_AUTONEG_EXP）***中的bit1。 1：已收到一个新的链接码字 0：没有收到链接码字
5	AN_COMPLETE	完成自动协商 设置为1表示自动协商已经完成，清除***AN控制寄存器（AN_CTL）***中的第12位使能自动协商时返回0。
4	AN_ABILITY	远程故障 设置为1，表示检测到远程故障情况。当启用光纤模式时，远端故障定义为远端故障。它实现了一个锁存高功能，这样一个远程故障的发生导致位4成为设置并保持设置，直到它被清除。在读取AN状态寄存器或状态寄存器时，它将被复位为0。它是***PHY状态寄存器（PHY_MII_STATUS）***中的bit4。
3	AN_ABILITY	自动协商能力 在双绞线模式下总是设置为1，表示PHY具有自动协商的能力。它是***PHY状态寄存器（PHY_MII_STATUS）***中的bit3。
2	LINK_STATUS	连接状态 设置为1，表示建立了有效的链路。它将通过一个锁存低功能来实现，这样链路失败条件的发生会导致链路状态位被清除并保持清除，直到它被

		读取。它是 ***PHY状态寄存器（PHY_MII_STATUS）***中的bit2。
1	保留	必须保持复位值
0	LP_AN_ABILITY	链路伙伴自动协商能力 设置为1，表示链路伙伴能够参与自动协商功能。

自动协商设备标识寄存器(AN_ID)

设备地址：0x7
地址偏移：0x02
复位值：0x0044

该寄存器可以按字节（8位）、半字（16位）和字（32位）进行访问。

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

位/位域	名称	描述
15:0	PHY_ID[15:0]	PHY标识符bit[31-16] OUI (bits 3-18). OUI =00-11-05

自动协商设备版本寄存器（AN_VER）

设备地址：0x7
地址偏移：0x03
复位值：0x1400

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

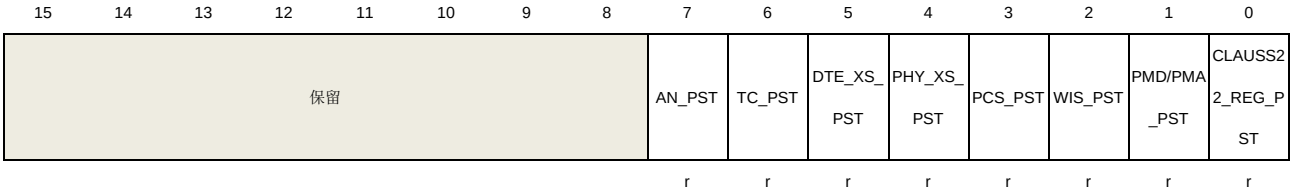
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHY_ID[5:0]						PHY_MAN_MODE[5:0]						PHY_REV_NO[3:0]			
r						r						r			

位/位域	名称	描述
15:10	PHY_ID[5:0]	PHY 标识符 bit [15-10] OUI bits 19-24
9:4	PHY_MAN_MODE[5:0]	制造商型号 制造商的型号(bits 5-0)，其中[5:4]=架构版本
3:0	PHY_REV_NO[3:0]	版本号(bits3-0) ***PCS 设备标识符寄存器（PCS_ID）***bit 0 是 PHY 标识符的 LS 位

AN 数据包寄存器 0（AN_PKG_0）

设备地址：0x7
地址偏移：0x05
复位值：0x0089

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

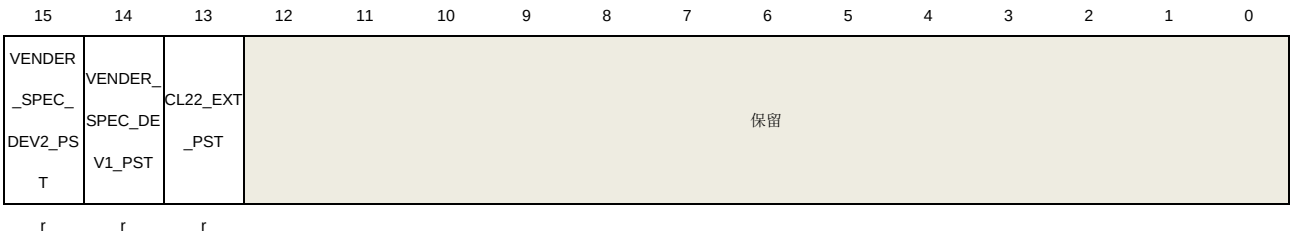


位/位域	名称	描述
15:8	保留	必须保持复位值
7	AN_PST	自动协商存在于包中 始终为 1
6	TC_PST	TC 存在包中 始终为 0
5	DTE_XS_PST	DTE XS 存在包中 始终为 0
4	PHY_XS_PST	PHY XS 存在包中 始终为 0
3	PCS_PST	PCS 存在包中 始终为 0
2	WIS_PST	WIS 存在包中 始终为 0
1	PMD/PMA_PST	PMD/PMA 存在包中 始终为 0
0	CLAUSS22_REG_PST	自动协商存在于包中 始终为 1

AN 数据包寄存器 1（AN_PKG_1）

设备地址：0x7
地址偏移：0x06
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



位/位域	名称	描述

15	VENDER_SPEC_DEV2_PST	特殊设备 2 存在于包中 始终为 0
14	VENDER_SPEC_DEV1_PST	特殊设备 1 存在于包中 始终为 0
13	CL22_EXT_PST	Clause 22 扩展在于包中 始终为 0
12:0	保留	必须保持复位值

自动协商设备标识寄存器（AN_ID）

设备地址：0x7
地址偏移：0x0E
复位值：0x0044

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

位/位域	名称	描述
15:0	PHY_ID[15:0]	PHY标识符bit[31-16] OUI (bits 3-18). OUI =00-11-05

自动协商设备版本寄存器（AN_VER）

设备地址：0x7
地址偏移：0x0F
复位值：0x1400

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

位/位域	名称	描述
15:10	PHY_ID[5:0]	PHY 标识符 bit [15-10] OUI bits 19-24
9:4	PHY_MAN_MODE[5:0]	制造商型号 制造商的型号(bits 5-0)，其中[5:4]=架构版本
3:0	PHY_REV_NO[3:0]	版本号(bits3-0) ***PCS 设备标识符寄存器（PCS_ID）*** bit 0 是 PHY 标识符的 LS 位

自动协商广播寄存器（AN_ADV）

设备地址：0x7
地址偏移：0x10
复位值：0x0101

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEXT_PAGE_EN	保留	REMOTE_FAULT_EN	EXTENED_NEXT_PAGE	ASYM_PAUSE	PAUSE	100BASE_T4	100BASETX_FULL_DPX	100BASETX_HALF_DPX	10BASE-T_FULL_DPX	10BASE-T_HALF_DPX	SLCT_FLD[4:0]				
rw		rw	rw	rw	rw	r	rw	rw	rw	rw	r				

位/位域	名称	描述
15	NEXT_PAGE_EN	下一页使能 1：设置为使用下一页 0：不要使用下一页
14	保留	必须保持复位值
13	REMOTE_FAULT_EN	远程故障检测使能 1：自动协商故障检测 0：没有远程故障
12	EXTENED_NEXT_PAGE	扩展下一页 PHY 不支持，应该始终写 0
11	ASYM_PAUSE	非对称暂停功能 A6 技术能力 1：支持非对称暂停能力 0：不支持非对称暂停
10	PAUSE	暂停功能 A5 技术能力 1：支持暂停功能 0：不支持暂停功能
9	100BASE_T4	100BASE-T4 功能 PHY 不支持，应该始终写 0
8	100BASETX_FULL_DPX	100BASE-X 全双工功能 1：支持全双工 0：不支持全双工 注意：当自动协商被禁用时，该位上的值将反映在 ***PHY 控制寄存器（PHY_MII_CTL）***BIT6, BIT13 (FORCE_SPEED)和 BIT8(FORCE_DUPLEX)中编程的值，当 FORCE_SPEED 为 2'b01 和 FORCE_DUPLEX 是 1'b1，那么这个位将是 1'b1，反之亦然。
7	100BASETX_HALF_DPX	100BASE-X 半双工功能 1：支持半双工

		0：不支持半双工 注意：当自动协商被禁用时，该位上的值将反映在 ***PHY 控制寄存器（PHY_MII_CTL）***BIT6, BIT13 (FORCE_SPEED)和 BIT8 (FORCE_DUPLEX)中编程的值。当 FORCE_SPEED 为 2'b01，FORCE_DUPLEX 为 1'b0 时，此位将为 1'b1，反之亦然。
6	10BASE-T_FULL_DPX	10BASE-T 全双工功能 1：支持全双工 0：不支持全双工 注意:当自动协商被禁用时，该位上的值将反映在 ***PHY 控制寄存器（PHY_MII_CTL）***BIT6, BIT13 (FORCE_SPEED)和 BIT8 (FORCE_DUPLEX)中编程的值。当 FORCE_SPEED 为 2'b00, FORCE_DUPLEX 为 1'b1 时，此位将为 1'b1，反之亦然。
5	10BASE-T_HALF_DPX	10BASE-T 半双工功能 1：支持半双工 0：不支持半双工 注意:当自动协商被禁用时，该位上的值将反映在 ***PHY 控制寄存器（PHY_MII_CTL）***BIT6, BIT13 (FORCE_SPEED)和 BIT8 (FORCE_DUPLEX)中编程的值。当 FORCE_SPEED 为 2'b00, FORCE_DUPLEX 为 1'b0 时，此位将为 1'b1，反之亦然。
4:0	SLCT_FLD[4:0]	识别消息类型 始终强制为 5’h01

自动协商链路伙伴(LP)性能寄存器（AN_LP_ABILITY）

设备地址：0x7
地址偏移：0x13
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEXT_PAGE	ACKNOWLEDGE	REMOTE_FAULT	EXTENED_NEXT_PAGE	ASYM_PAUSE	PAUSE	100BASET4	100BASE TX_FULL_DPX	100BASE TX_HALF_DPX	10BASE-T_FULL_DPX	10BASE-T_HALF_DPX	SELECTOR_FIELD[4:0]				
r	r	r	r	r	r	r	r	r	r	r	r				

位/位域	名称	描述
15	NEXT_PAGE	链接伙伴下一页请求 1：链接伙伴正在请求下一页功能 0：请求基页
14	ACKNOWLEDGE	链接伙伴确认接收 1：链接伙伴确认接收成功 0：没有接收
13	REMOTE_FAULT	链接伙伴检测到远程故障

		1：自动协商故障检测 0：没有远程故障
12	EXTENED_NEXT_PAGE	扩展下一页
11	ASYM_PAUSE	链接伙伴非对称暂停功能 A6技术能力 1：支持非对称暂停能力 0：不支持非对称暂停能力
10	PAUSE	链接伙伴对称暂停功能 A5技术能力 1：支持对称暂停能力 0：不支持对称暂停能力
9	100BASET4	A4技术能力 链接伙伴 100BASE-T4 能力
8	100BASETX_FULL_DPX	链接伙伴100BASE-X全双工能力 1：支持全双工 0：不支持 注意：当自动协商被禁用时，该位的值将一直被设置为1'b1
7	100BASETX_HALF_DPX	链接伙伴100BASE-X半双工能力 1：支持半双工 0：不支持 注意：当自动协商被禁用时，该位的值将一直被设置为1'b1
6	10BASE-T_FULL_DPX	链接伙伴10BASE-T全双工能力 1：支持全双工 0：不支持 注意：当自动协商被禁用时，该位的值将一直被设置为1'b1
5	10BASE-T_HALF_DPX	链接伙伴10BASE-T半双工能力 1：支持半双工 0：不支持 注意：当自动协商被禁用时，该位的值将一直被设置为1'b1
4:0	SELECTOR_FIELD[4:0]	链路伙伴标识消息类型应该是5'h01

自动协商 XNP 发送寄存器（AN_XNP_TRANSMIT）

设备地址：0x7
地址偏移：0x16
复位值：0x2001

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

NEXT_PAGE	保留	MESSAGE_PAGE	ACKNOWLEDGE2	TOGGLE	MSG_UFMT_CODE_FIELD[10:0]
rw		rw	rw	r	rw

位/位域	名称	描述
15	NEXT_PAGE	最后一页 指示这是否是最后一页 1：附加的下一页将紧随其后 0：下一页是最后一页
14	保留	必须保持复位值
13	MESSAGE_PAGE	消息页或未格式化页 指示这是消息页或未格式化的页 1：消息页面 0：无格式页面
12	ACKNOWLEDGE2	依从信息的能力 1：依从信息 0：无法依从信息
11	TOGGLE	切换 切换位由SW自动计算，软件可以忽略
10:0	MSG_UFMT_CODE_FIELD[10:0]	消息/未格式化代码字段

自动协商链路伙伴 XNP 性能寄存器（AN_LP_XNP_ABILITY）

设备地址：0x7
地址偏移：0x19
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

位/位域	名称	描述
15	NP_RX[15:0]	从链路伙伴收到下一页

主-从控制寄存器(MS_CTL)

设备地址：0x7
地址偏移：0x20
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

MS_MAU_CFG_EN	MS_MAU_CFG_VAL	PORT_TYPE	保留
rw	rw	rw	

位/位域	名称	描述
15	MS_MAU_CFG_EN	启用主从手动配置 1：启用主从手动配置 0：关闭主从手动配置
14	MS_MAU_CFG_VAL	主从手动配置值 1：配置PHY为主设备 0：配置PHY为从设备
13	PORT_TYPE	端口类型 1：多端口设备 0：单端口设备
12:0	保留	必须保持复位值

主-从状态寄存器(MS_STS)

设备地址：0x7
地址偏移：0x21
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

位/位域	名称	描述
15	MS_CFG_FAULT	检测到主从配置错误 1：检测到主从配置错误 0：未检测到主从配置错误
14	MS_CFG_RES	主从配置解析 1：本地配置解析为主 0：本地配置解析为从
13:0	保留	必须保持复位值

EEE 广播寄存器（EEE_ADV）

设备地址：0x7
地址偏移：0x3C

复位值：0x0002

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									10GBASE_KR_EEE	10GBASE_KX4_EEE	1000BASE_KX_EEE	10GBASE_T_EEE	1000BASE_T_EEE	100BASE_TX_EEE	保留
									r	r	r	r	rw	rw	

位/位域	名称	描述
15:7	保留	必须保持复位值
6	10GBASE_KR_EEE	10GBASE-KR EEE 不支持，始终为0
5	10GBASE_KX4_EEE	10GBASE-KR EEE 不支持，始终为0
4	1000BASE_KX_EEE	10GBASE-KR EEE 不支持，始终为0
3	10GBASE_T_EEE	10GBASE-KR EEE 不支持，始终为0
2	1000BASE_T_EEE	1000Base-T EEE 1：广播1000BASE-T具有EEE能力 0：不广播1000BASE-T具有EEE能力
1	100BASE_TX_EEE	100Base-TX EEE 1：广播1000BASE-TX具有EEE能力 0：不广播1000BASE-TX具有EEE能力
0	保留	必须保持复位值

EEE 链路伙伴(LP)性能寄存器（EEE_LP_ABILITY）

设备地址：0x7
地址偏移：0x3D
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									10GBASE_KR_EEE	10GBASE_KX4_EEE	1000BASE_KX_EEE	10GBASE_T_EEE	1000BASE_T_EEE	100BASE_TX_EEE	保留
									r	r	r	r	r	r	

位/位域	名称	描述

15:7	保留	必须保持复位值
6	10GBASE_KR_EEE	10GBASE-KR EEE 1：链接伙伴广播10GBASE-KR的EEE功能 0：链接伙伴不广播10GBASE-KR的EEE功能
5	10GBASE_KX4_EEE	10GBASE-KX4 EEE 1：链接伙伴广播10GBASE-KX4的EEE功能 0：链接伙伴不广播10GBASE-KX4的EEE功能
4	1000BASE_KX_EEE	1000BASE-KX EEE 1：链接伙伴广播1000BASE-KX的EEE功能 0：链接伙伴不广播1000BASE-KX的EEE功能
3	10GBASE_T_EEE	10GBASE-T EEE 1：链接伙伴广播10GBASE-T的EEE功能 0：链接伙伴不广播10GBASE-T的EEE功能
2	1000BASE_T_EEE	1000BASE-T EEE 1：链接伙伴广播1000BASE-T的EEE功能 0：链接伙伴不广播1000BASE-T的EEE功能
1	100BASE_TX_EEE	100BASE-TX EEE 1：链接伙伴广播100BASE-TX的EEE功能 0：链接伙伴不广播100BASE-TX的EEE功能
0	保留	必须保持复位值

50.9. EtherCAT

50.9.1. 简介

ESC 是一种 EtherCAT®从站控制器(ESC)。它作为 EtherCAT 现场总线和从站应用之间的接口，负责 EtherCAT 通信。ESC 支持广泛的应用。EtherCAT 控制器拥有 8K 字节的过程数据内存(PDRAM)和 8 个现场总线内存管理单元(FMMUs)，每个单元的任务是将逻辑地址映射到物理地址。EtherCAT 从站控制器还包括 8 个 SyncManager，允许 EtherCAT 和本地应用之间进行数据交换。每个 SyncManager 的方向和操作模式由 EtherCAT 主设备配置。有两种工作模式可用：缓冲模式和邮箱模式。在缓冲模式下，微控制器和 EtherCAT 主可以同时写入设备。ESC 中的缓冲区始终包含最新数据。如果新数据在旧数据可以读取之前到达，旧数据将丢失。在邮箱模式下，微控制器和 EtherCAT 主通过握手访问缓冲区，确保不会丢失数据。

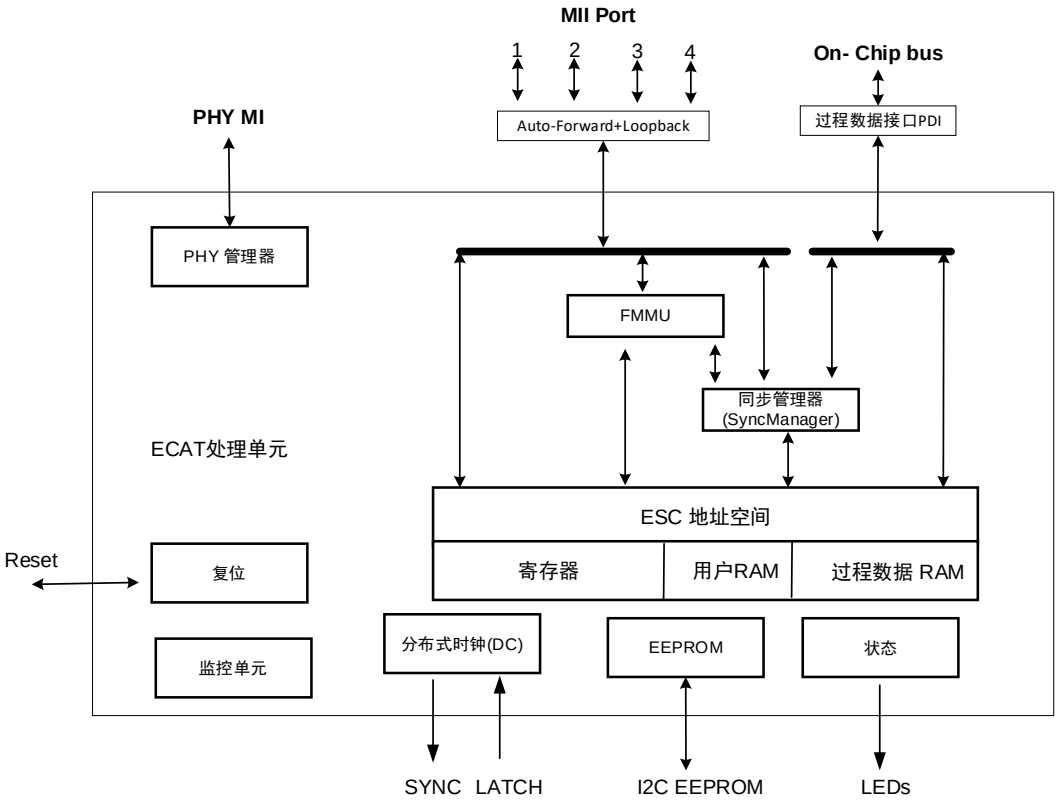
50.9.2. 主要特征

- 端口支持：2个内部PHY接口和1个外部MII。
- 8个现场总线存储管理单元（FMMUs）。
- 8KB过程数据内存（PDRAM）。
- 支持64位分布式时钟，与其他EtherCAT设备同步。
- 8个SyncManager实体。
- DC同步偏差小于1微秒。

结构框图

ESC 的功能模块图如*图 50-36. EtherCAT 系统框图*所示。

图 50-36. EtherCAT 系统框图



EtherCAT 从站控制器功能块

■ EtherCAT接口

EtherCAT 接口或端口将 ESC 连接到其他 EtherCAT 从站和主站。MAC 层是 ESC 的组成部分。物理层可能是以太网。对于以太网端口，内部以太网 PHY 连接到 ESC 的 MII 端口。EtherCAT 的传输速度固定为 100Mbit/s，采用全双工通信。链路状态和通信状态会报告给监控设备。ESC 使用三个端口，端口 0/1/2。

■ EtherCAT处理单元

EtherCAT 处理单元（EPU）接收、分析并处理 EtherCAT 数据流。它在逻辑上位于端口 0 和端口 3 之间。EtherCAT 处理单元的主要目的是启用和协调对 ESC 内部寄存器和存储器空间的访问，这些寄存器和存储器空间既可以从 EtherCAT 主设备访问，也可以通过 PDI 从本地应用程序访问。主设备和从设备应用程序之间的数据交换类似于双端口内存（进程内存），通过特殊功能增强，例如一致性检查（SyncManager）和数据映射（FMMU）。EtherCAT 处理单元包含 EtherCAT 从设备的主要功能块，除了自动转发、环回功能和 PDI。

■ 自动转发器

自动转发器接收以太网帧，进行帧检查，并将帧转发到环回功能。自动转发器生成接收帧的时间戳。

■ 环回功能

环回功能将以太网帧转发到下一个逻辑端口，如果某个端口没有连接，或者端口不可用，或者

该端口的环路处于闭合状态。端口 0 的环回功能将帧转发到 EtherCAT 处理单元。环回设置可以由 EtherCAT 主控器控制。

- FMMU

现场总线内存管理单元用于将逻辑地址映射到 ESC 的物理地址。

- 同步管理器

同步管理器负责在 EtherCAT 主设备和从设备之间进行一致的数据交换和邮箱通信。每个同步管理器的通信方向都可以配置。读或写事务可能会为 EtherCAT 主设备和附加的微控制器生成事件。同步管理器负责区分 ESC 和双端口内存之间的主要区别，因为它们将地址映射到不同的缓冲区，并根据同步管理器的状态阻止访问。这也是 PDI 带宽限制的根本原因。

- 监控

监控单元包含错误计数器和看门狗。看门狗用于观察通信并在出现错误时返回到安全状态。错误计数器用于错误检测和分析。

- PHY管理器

PHY 管理单元通过 MII 管理接口与以太网 PHY 通信。这可以由主设备或从设备使用。MII 管理接口由 ESC 本身用于在接收到增强型链路检测机制的错误后选择性地重新启动自动协商，以及用于可选的 MII 链路检测和配置功能。

- 分布式时钟

分布式时钟（DC）允许精确同步地生成输出信号和输入采样，以及生成事件的时间戳。这种同步可能覆盖整个 EtherCAT 网络。

- EEPROM

需要一个非易失性存储器来存储 EtherCAT 从站信息（ESI），通常是一个 I²C EEPROM。

- 状态/指示灯

状态模块提供 ESC 和应用程序状态信息。它控制外部 LED，如应用程序运行 LED/错误 LED 和端口链接/活动 LED。

50.9.3. 功能描述

过程数据接口（PDI）

进程数据接口（PDI）实现了从应用程序与 ESC 之间的连接。定义了几种类型的 PDI，包括串行和并行微控制器接口以及数字 I/O 接口。由于 EtherCAT 和 PDI 对内存、寄存器以及尤其是同步管理器的访问高度依赖，内部 PDI 接口最多可以实现约 12.5Mbyte/s 的吞吐量。

表 50-11. EtherCAT 的 PDI

PDI数量(PDI控制寄存器0x0140)	PDI名称
0	接口失效
128	On-chip bus

PDI 选择和配置

通常，PDI 的选型和配置是 SII EEPROM 的 ESC 配置区域的一部分。ESC 在上电时选择并配置 PDI。在这种情况下，ESC 配置区域应该反映实际设置，尽管不是由 ESC 本身决定的。释放复位后，PDI 处于激活状态，这使得微控制器能够实现 EEPROM 仿真。在 EEPROM 未加载时，注意数字输出信号和 DC 同步信号，以实现正确的输出行为。

PDI 寄存器功能通过写入确认

一些 ESC 的功能是通过写入或读取单个字节地址、同步管理器缓冲区变更或 AL 事件请求确认来触发的。随着微控制器数据总线宽度的增加，这可能会导致限制甚至问题。

由于大多数微控制器在使用字节使能信号进行写访问时没有限制，但许多微控制器在读取访问时不使用字节使能信号，它们期望获得整个数据总线的宽度的读取数据。读取单个字节是不可能的。这可能导致问题，特别是意外读取触发某些 ESC 功能的字节地址时。考虑 SyncManager 缓冲区区域从 0x1000-0x1005。一个 32 位微控制器应用程序可能会逐字节读取缓冲区。对 0x1000 的第一次访问将打开缓冲区，并且还会读取 0x1001-0x1003。第二次访问将读取 0x1001，并且还将读取 0x1000/0x1002-0x1003。当地址 0x1004 要被读取时，问题就出现了，因为这也会读取 0x1005。0x1005 的数据被丢弃了，但缓冲区被关闭了。当微控制读取 0x1005 时，它总是得到 0 - 数据似乎被损坏了。类似的问题也出现在 DC 同步信号确认（寄存器 0x098E 和 0x098F）上。32 位微控制器总是会同时确认 SYNC0 和 SYNC1，不可能分别确认。

这个问题可以通过使能 PDI 寄存器功能确认写入来解决。在这种模式下，所有原本由读访问触发的功能现在都由相应的写访问触发——这些写访问使用字节使能，因此可以限制在特定的字节。

此功能通过 IP 核配置启用。在使用此功能之前，必须通过微控制器应用程序检查 PDI 信息寄存器 0x014E[0]的当前状态。

此功能影响 SyncManager 缓冲区的读取以及从 PDI 端读取某些寄存器。EtherCAT 主设备端没有任何变化。请参阅*同步管理器(SyncManager)*，了解同步管理器的行为。以下寄存器受到 PDI 寄存器功能通过写入确认的影响：

表 50-12. 通过写入确认 PDI 寄存器功能的受影响寄存器

地址	名称	触发函数
任意	SyncManager内存和地址	读取SyncManager缓冲区，然后写入缓冲区结束地址以确认缓冲区读取。
0x0120:0x0121	AL控制	AL控制发生变化后，读取0x0120:0x0121，然后写入0x0120确认读取。
0x0440	看门狗状态进程数据	读取0x0440，然后写入0x0440以清除AL事件请求0x0220[6]
0x0806+X*16	SyncManager激活	读取0x0806+X*16，然后写入0x0806(SyncManager 0)，只为清除所有SyncManager的AL事件请求0x0220[4]

地址	名称	触发函数
0x098E	SYNC0状态	读取0x098E，然后写入0x098E以确认数据同步状态0x098E[0]
0x098F	SYNC1状态	读取0x098E，然后写入0x098E以确认数据同步状态0x098F[0]
0x09B0:0x09B7	Latch0时间上升沿锁存	读取0x09B0: 0x09B7，然后写入0x09B0，清除DC Latch0状态0x09AE[0]
0x09B8:0x09BF	Latch0时间下降沿锁存	读取0x09B8: 0x09BF，然后写入0x09B8以清除DC Latch0状态0x09AE[1]
0x09C0:0x09C7	Latch1时间上升沿锁存	读取0x09C0: 0x09C7，然后写入0x09C0，清除DC Latch1状态0x09AF[0]
0x09C8:0x09CF	Latch1时间下降沿锁存	读取0x09C8: 0x09CF，然后写入0x09C8以清除DC Latch1状态0x09AF[1]

FMMU

现场总线存储器管理单元（FMMU）通过内部地址映射的方式将逻辑地址转换为物理地址。因此，FMMU 允许使用逻辑地址来访问跨越多个从设备的分段数据：一个数据报文在几个任意分布的 EtherCAT 中寻址数据。每个 FMMU 通道将一个连续的逻辑地址空间映射到从设备的连续物理地址空间。FMMU 支持的访问类型可以配置为读取、写入或读写。

■ FMMU设置的限制

ESC 中的 FMMU 受到限制。如果使用位映射（逻辑起始位≠0，逻辑停止位≠7，或物理起始位≠0），则同一方向（读或写）的两个 FMMU 的逻辑地址范围必须至少相隔 3 个未配置任何同类型 FMMU 的逻辑字节。

■ 附加的FMMU特性

- 每个逻辑地址字节最多可以由一个FMMU（读）加上一个FMMU（写）映射，或者由一个FMMU（读/写）映射。如果两个或更多的FMMU（具有相同的方向 - 读或写）被配置用于相同的逻辑字节，则使用编号较低的FMMU（较低的配置地址空间），其他FMMU将被忽略。
- 一个或多个FMMU可能指向相同的物理内存，它们都被使用。碰撞不会发生。
- 使用一个读写FMMU或使用两个FMMU——一个用于读取，另一个用于写入——对于相同的逻辑地址来说是一样的。
- 读写型FMMU不能与同步管理器一起使用，因为独立的读写同步管理器不能配置为使用相同的（或重叠的）物理地址范围。
- 位读取支持在任何地址进行。没有映射到逻辑地址的位在EtherCAT数据报中不会被改变。例如，这允许将来自多个ESC的位映射到同一个逻辑字节中。
- 一个针对未在ESC中配置的逻辑地址空间的帧/数据报，不会改变ESC中的数据，并且ESC中的数据也不会被放置在帧/数据报中。

同步管理器(SyncManager)

ESC 的内存可以用来在 EtherCAT 主控器和连接到 PDI 的微控制器上本地应用程序之间交换数据，没有任何限制。使用内存进行通信有一些问题，这些问题可以通过 ESC 内的同步管理器来解决：

- 数据一致性无法保证。必须在软件中实现信号量以协调地交换数据。
- 数据安全无法保证。必须在软件中实现安全机制。
- EtherCAT 主站和应用程序都必须轮询内存，以了解对方何时完成访问。

同步管理器能够在 EtherCAT 主设备和本地应用程序之间实现一致且安全的数据处理，并生成中断以通知双方变化。

同步管理器由 EtherCAT 主控制器配置。通信方向是可配置的，通信模式（缓冲模式和邮箱模式）也是可配置的。同步管理器使用位于内存区域的缓冲区来交换数据。对这个缓冲区的访问由同步管理器的硬件控制。

缓冲区必须从起始地址开始访问，否则访问将被拒绝。访问起始地址后，整个缓冲区都可以被访问，包括再次访问起始地址，可以一次性访问整个缓冲区，也可以分多次访问。当访问结束地址时，缓冲区访问结束，缓冲区状态随后改变，并且如果配置了，将生成中断或看门狗触发脉冲。在一个帧内，结束地址不能被访问两次。

同步管理器支持两种通信模式：

- 缓冲模式
  - 缓冲模式允许双方，即EtherCAT主控器和本地应用程序，随时访问通信缓冲区。消费者总是获得生产者写入的最新一致的缓冲区，而生产者可以随时更新缓冲区的内容。如果缓冲区的写入速度比读取速度快，旧数据将被丢弃。
  - 缓冲模式通常用于循环处理数据。
- 邮箱模式
  - 邮箱模式实现了一种数据交换的握手机制，以确保数据不会丢失。每一方，无论是EtherCAT主控器还是本地应用程序，只有在另一方完成访问后才能获得对缓冲区的访问。首先，主机向缓冲区写入数据。然后，缓冲区被锁定以防止写入，直到从机读取完毕。之后，主机再次获得写入访问权限，而缓冲区则对从机锁定。
  - 邮箱模式通常用于应用层协议。

同步管理器仅在帧的帧检验序列（FCS）正确时才接受由主机引起的缓冲区变化，因此，缓冲区的变化在帧结束不久后生效。

同步管理器的配置寄存器位于从寄存器地址 0x0800 开始的位置。

分布式时钟（DC）

EtherCAT 从控制器分布式时钟（DC）单元支持以下功能：

- 从机（与主机）之间的时钟同步
- 同步输出信号的生成（SyncSignals）
- 输入事件的精确时间戳（LatchSignals）
- 产生同步中断

- 同步数字输出更新
- 同步数字输入采样

本设备支持 64 位分布式时钟，详细介绍如下。

EtherCAT 提供了两个输入引脚（SYNC 和 LATCH），用于对外部事件进行时间戳。记录了上升沿和下降沿的时间戳。这些引脚与 SYNC0 和 LATCH0 输出引脚共享，分别用于指示时间事件的发生。SYNC/SYNC0 和 LATCH/LATCH0 引脚的功能由 Sync/Latch PDI 配置寄存器中的 SYNC0/LATCH0 和 SYNC/LATCH 配置位决定。

当设置为 SYNC0/LATCH0 功能时，输出类型（推挽式与开漏/开源）和输出极性由同步/锁存 PDI 配置寄存器中的 SYNC0 输出驱动器/极性位和 LATCH0 输出驱动器/极性位决定。

EtherCAT 状态机(ESM)

EtherCAT 状态机（ESM）负责协调主机和从机。

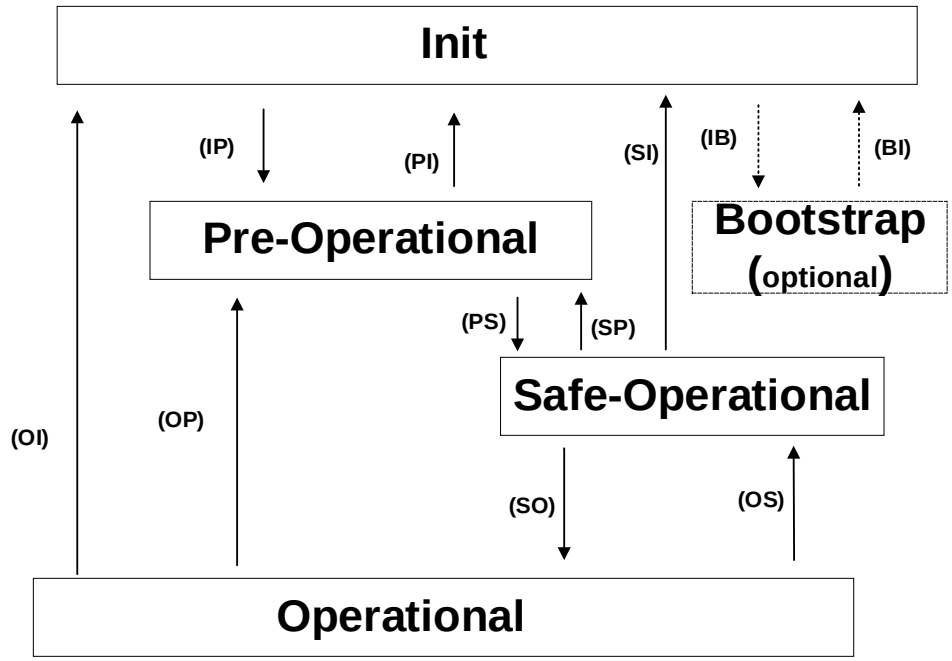
应用程序在启动时和运行期间的状态变化通常由主机的请求来启动。这些状态变化在相关操作执行后由本地应用程序确认。本地应用程序也可能发生未经请求的状态变化。

EtherCAT 从站应支持四种状态，外加一种可选状态：

- 初始化（复位后的状态）
- 预操作
- 安全操作
- 运营
- 启动（可选）

状态和允许的状态更改显示在*图 50-37. EtherCAT 状态机*。

图 50-37. EtherCAT 状态机



注意：并非所有状态变化都是可能的，从“初始化”到“操作状态”的转换需要以下序列：初始化 -> 预操作状态 -> 保存操作状态 -> 操作状态。

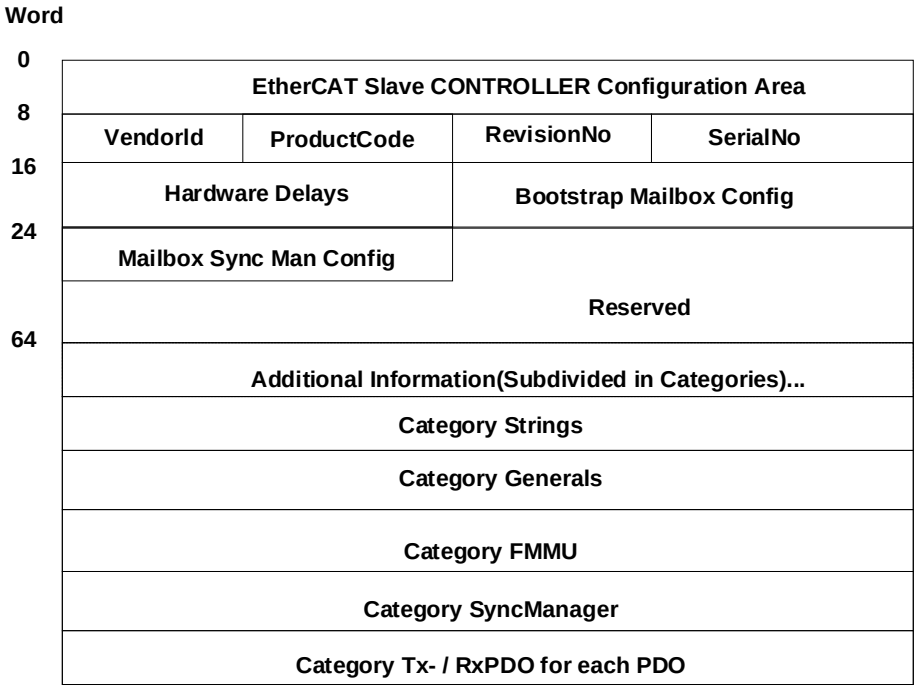
每个状态定义了所需的服务。在确认状态变更之前，所有为请求状态提供或停止所需的服务都必须得到满足。

EEPROM

EtherCAT 从站控制器使用一个强制性的 RAM（通常是一个带有 I²C 接口的串行 EEPROM）来存储 EtherCAT 从站信息（ESI）。根据 ESC 的不同，支持从 1 Kbit 到 4 Mbit 的 EEPROM 大小。

EEPROM 结构如*图 50-38. EEPROM 布局*所示。EEPROM 布局，ESI 使用字地址。

图 50-38. EEPROM 布局



至少存储在地址范围从字 0 到 63（0x00 到 0x3F）的信息是强制性的，以及通用类别（绝对最小的 EEPROM 大小是 2Kbit，具有许多类别的复杂设备应该配备 32Kbit EEPROM 或更大的存储器）。ESC 配置区域由 ESC 用于配置。所有其他部分由主机或本地应用程序使用。

复位

EtherCAT 模块提供了两个寄存器，*ESC 复位 ECAT 寄存器（ESC_RST_ECAT）*和*ESC 复位 PDI 寄存器（ESC_RST_PDI）*，它们可以分别被 EtherCAT 主站和从站访问触发一个复位请求。

中断

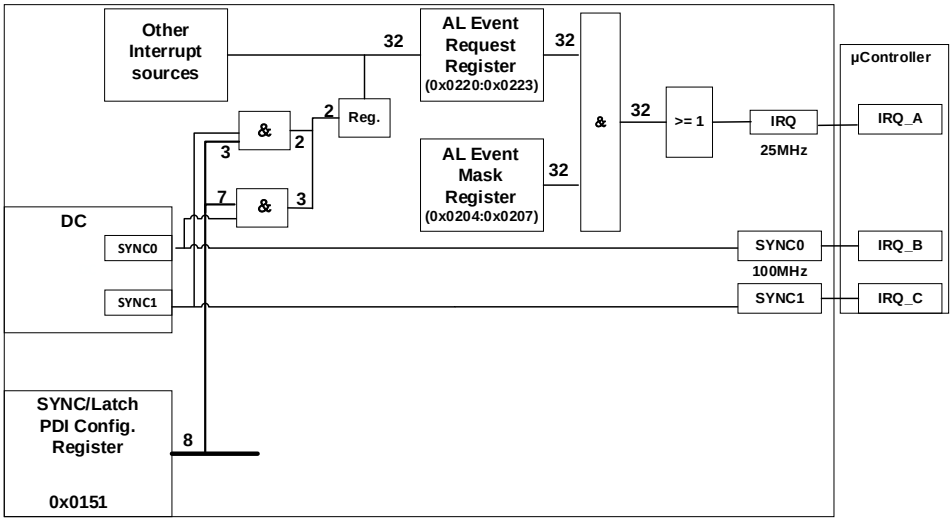
EtherCAT 支持两种类型的中断：针对微控制器的 AL 事件请求，以及针对 EtherCAT 主控的 EtherCAT 事件请求。此外，分布式时钟同步信号也可以作为微控制器的中断使用。

事件请求（PDI 中断）

AL 事件请求可以通过 PDI 中断请求信号（IRQ/SPI_IRQ 等）向微控制器发出信号。对于 IRQ 生成，AL 事件请求寄存器（0x0220:0x0223）与 AL 事件掩码寄存器（0x0204:0x0207）通过

逻辑与操作组合，然后所有结果位通过逻辑或操作组合成一个中断信号。IRQ 信号的输出驱动特性可以使用 SYNC/LATCH PDI 配置寄存器（0x0151）进行配置。AL 事件掩码寄存器允许选择对微控制器和应用程序处理相关的中断。

图 50-39. PDI 中断屏蔽和中断信号



DC 同步信号可以用两种方式用于中断生成：

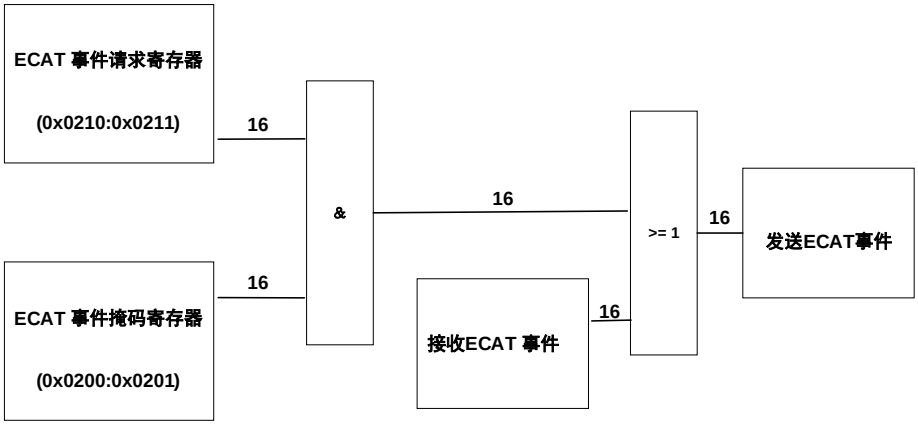
- DC同步信号被映射到AL事件请求寄存器（配置为SYNC/LATCH PDI配置寄存器0x0151.3/7）。在这种情况下，所有来自ESC到微控制器的中断被合并为一个IRQ信号，并且分布式时钟LATCH0/1输入仍然可以使用。IRQ信号的抖动约为40ns。
- DC同步信号直接连接到微控制器的中断输入。微控制器可以更快地响应DC同步信号中断（无需读取AL请求寄存器），但它需要更多的中断输入。同步信号的抖动约为12ns。DC Latch功能的可用性仅限于一个Latch输入，或者根本不可用（如果两个DC同步输出都被使用）。

ECAT 事件请求（ECAT 中断）

ECAT 事件请求用于通知 EtherCAT 主站从站事件。ECAT 事件利用 EtherCAT 数据报中的 IRQ 字段。ECAT 事件请求寄存器（0x0210:0x0211）与 ECAT 事件掩码寄存器（0x0200:0x0201）通过逻辑与操作结合。由此产生的中断位与传入的 ECAT IRQ 字段通过逻辑或操作结合，并写入传出的 ECAT IRQ 字段。ECAT 事件掩码寄存器允许选择对 EtherCAT 主控制器相关并由主应用程序处理的中断。

注意：主站无法区分哪个从站（甚至可能是多个）是中断的来源。

图 50-40. EtherCAT 中断屏蔽



清除意外中断

事件请求寄存器和清除中断的寄存器操作旨在独立访问，即通过单独的以太网帧或单独的 PDI 访问。否则可能会错过中断和/或数据。

LED

EtherCAT 从站控制器支持 LED（RUNLED）以反映链路状态和 AL 状态。ESC 的 LED 输出由 AL 状态寄存器（0x0130）控制，并支持以下状态，这些状态会自动转换为闪烁代码。

EtherCAT 内核配置通过 RUN LED 覆盖寄存器直接控制 RUN LED。

50.9.4. ESC 寄存器定义

ESC 类型寄存器（ESC_TYPE）

地址偏移：0x0000
复位值：0xBC

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

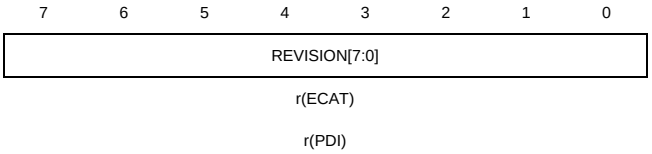
7	6	5	4	3	2	1	0
TYPE[7:0]							
r(ECAT)							
r(PDI)							

位/位域	名称	描述
7:0	TYPE[7:0]	EtherCAT 控制器类型 0xBC: GDSCN

ESC 版本寄存器（ESC_REVISION）

地址偏移：0x0001
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

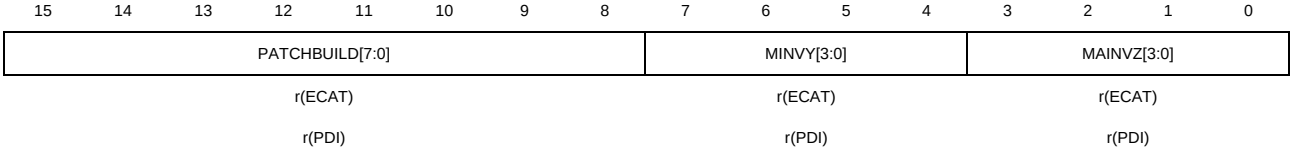


位/位域	名称	描述
7:0	REVISION[7:0]	EtherCAT 控制器版本

ESC 编译寄存器（ESC_BUILD）

地址偏移：0x0002
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

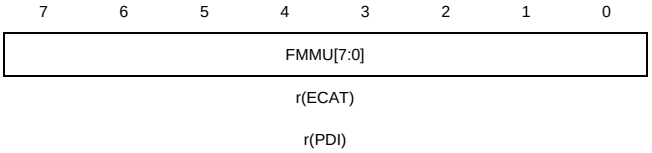


位/位域	名称	描述
15:8	PATCHBUILD[7:0]	补丁级别/开发版本: 0x00:原始发布 0x01-0x0F:原始版本的补丁发布 0x10-0xFF:开发版本
7:4	MINVY[3:0]	次要版本 Y
3:0	MAINVZ[3:0]	维护版本 Z

ESC FMMU 数目寄存器（ESC_FMMUS）

地址偏移：0x0004
复位值：0x08

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

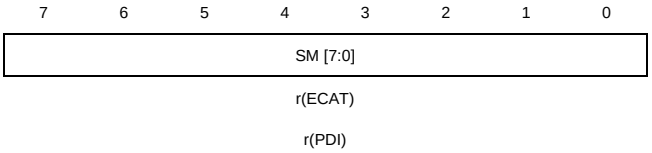


位/位域	名称	描述
7:0	FMMU[7:0]	该字段详细说明 EtherCAT 从控制器支持的 FMMU 通道(或实体)的数量。设备提供 8 个通道(或实体)。

ESC 同步管理器的数目寄存器（ESC_SYNCMS）

地址偏移：0x0005
复位值：0x08

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

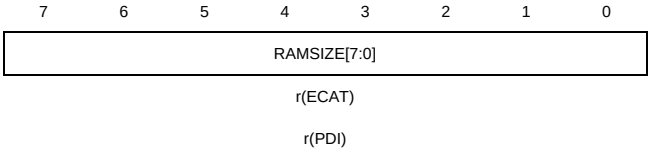


位/位域	名称	描述
7:0	SM[7:0]	支持的 SyncManager 实体数，设备提供 8 个实体。

ESC RAM 大小寄存器（ESC_RAMSIZE）

地址偏移：0x0006
复位值：0x08

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

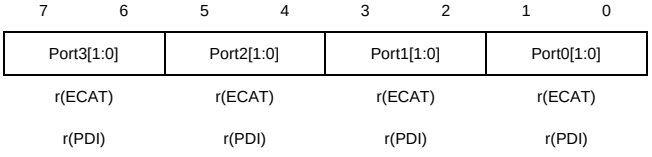


位/位域	名称	描述
7:0	RAMSIZE[7:0]	进程数据内存大小支持 8Kbyte

ESC 端口描述寄存器（ESC_PORT_DESC）

地址偏移：0x0007
复位值：0x3F

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



位/位域	名称	描述
7:6	Port3[1:0]	端口 3 配置 该字段详细说明端口3的配置。 00：未实现 01：未配置 10：保留

11：MII

5:4	Port2[1:0]	端口 2 配置 该字段详细说明端口2的配置。 00：未实现 01：未配置 10：保留 11：MII
3:2	Port1[1:0]	端口 1 配置 该字段详细说明端口1的配置。 00：未实现 01：未配置 10：保留 11：MII
1:0	Port0[1:0]	端口 0 配置 该字段详细说明端口0的配置。 00：未实现 01：未配置 10：保留 11：MII

ESC 功能支持寄存器（ESC_FEATURE）

地址偏移：0x0008
复位值：0x01CC

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				FS11	FS10	FS9	FS8	FS7	FS6	保留		FS3	FS2	FS1	FS0
				r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)			r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)

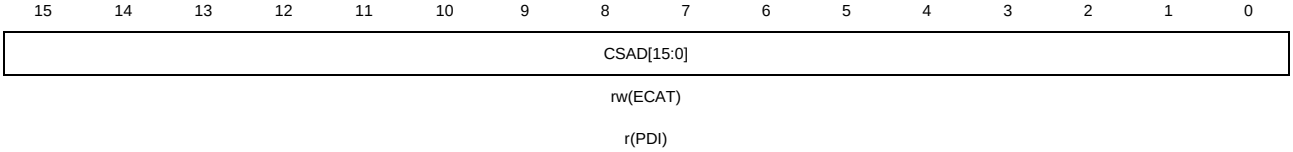
位/位域	名称	描述
15:12	保留	必须保持复位值
11	FS11	固定 FMMU/SyncManager 配置 0：可变配置 1：固定配置
10	FS10	EtherCAT 读/写命令支持(BRW,APRW,FPRW) 0：支持 1：不支持
9	FS9	EtherCAT LRW 命令支持 0：支持 1：不支持

8	FS8	NOTE: This feature refers to registers 0x981[7:3] and 0x0984 增强型 DC SYNC 激活 0：不可用 1：可用 注意：该功能请参 ***ESC 激活寄存器（ESC_REG_ACT）***[7:3]和 ***ESC 激活状态寄存器（ESC_ACT_STAT）***
7	FS7	单独处理 FCS 错误 0：不支持 1：支持，具有错误 FCS 的帧以及额外的半字节将在转发的接收计数器中单独计数
6	FS6	增强型链路检测 MII 0：不可用 1：可用
5:4	保留	必须保持复位值
3	FS3	分布式时钟（宽度） 0：32 位 1：64 位
2	FS2	分布式时钟 0：不可用 1：可用
1	FS1	未使用寄存器接口 0：允许 1：不支持
0	FS0	FMMU 操作 0：针对位操作 1：针对字节操作

ESC 配置的站点地址寄存器（ESC_STATION_ADDR）

地址偏移：0x0010
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

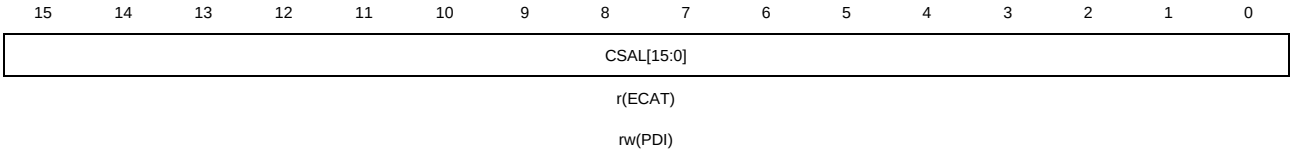


位/位域	名称	描述
15:0	CSAD[15:0]	用于节点寻址的地址(FPRD/FPWR/FPRW/FRMW 命令)。

ESC 配置的站点别名寄存器（ESC_STATION_ALS）

地址偏移：0x0012
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

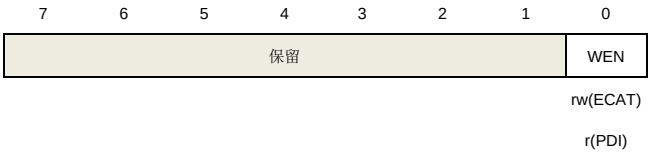


位/位域	名称	描述
15:0	CSAL[15:0]	该字段包含用于节点寻址（FPRD/FPWR/FPRW/FRMW命令）的别名地址。该别名的使用由*ESC DL控制寄存器（ESC_DL_CTL）*的Bit 24位激活。 注意：仅在上电或复位后第一次装载 EEPROM 时接受 EEPROM 值。

写使能寄存器（ESC_REG_WEN）

地址偏移：0x0020
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

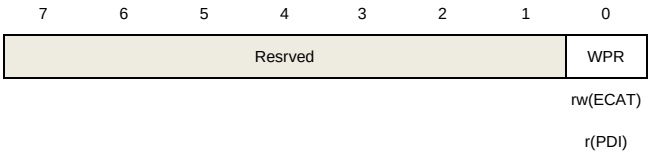


位/位域	名称	描述
7:1	保留	必须保持复位值
0	WEN	如果使能了 ESC 写保护，必须在同一以太网帧中写入该寄存器（值无关），之后才允许对该站执行其他写操作。ESC 写保护在该帧后仍有效（如果 ESC 写寄存器保护寄存器没有变化）

ESC 写保护寄存器（ESC_REG_WP）

地址偏移：0x0021
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



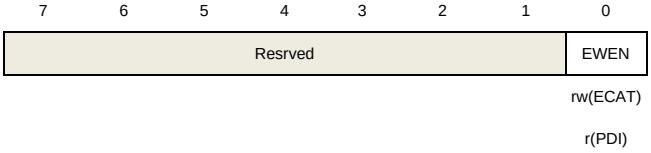
位/位域	名称	描述

7:1	保留	必须保持复位值
0	WPR	写寄存器保护 0：禁止保护 1：使能保护 除了 0020 和 0030h 之外，寄存器 0000h-0F7Fh 均受写保护。

ESC 写寄存器使能寄存器（ESC_WEN）

地址偏移：0x0030
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

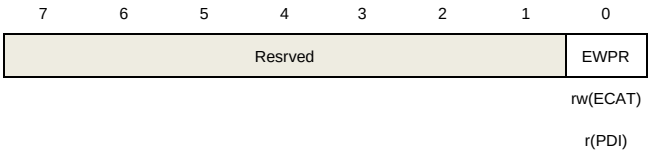


位/位域	名称	描述
7:1	保留	必须保持复位值
0	EWEN	如果使能了ESC写保护，必须在同一以太网帧中写入该寄存器（值无关），之后才允许对该站执行其他写操作。ESC写保护在该帧后仍有效（如果ESC写寄存器保护寄存器没有变化）

ESC 写寄存器保护寄存器（ESC_WP）

地址偏移：0x0031
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



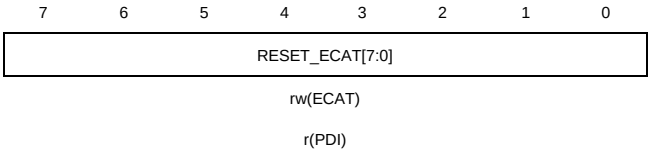
位/位域	名称	描述
7:1	保留	必须保持复位值
0	EWPR	ESC写寄存器保护 0：禁止保护 1：使能保护 除了0030h之外，所有区域均受写保护。

ESC 复位 ECAT 寄存器（ESC_RST_ECAT）

地址偏移：0x0040
复位值：0x0

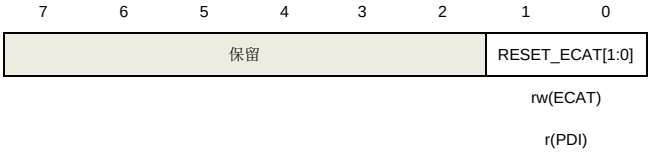
该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

写：



位/位域	名称	描述
7:0	RESET_ECAT[7:0]	通过 3 个连续的命令向该寄存器中写入 52h（“R”）、45h（“E”）和 53h（“S”）之后，将触发复位。

读:



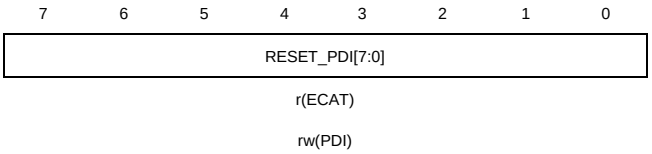
位/位域	名称	描述
7:2	保留	必须保持复位值
1:0	RESET_ECAT[1:0]	复位步骤进度 00：初始化/复位状态 01：在写入 52h（“R”）后，之前的状态是 00 10：在写入 45h（“E”）后，之前的状态是 01 11：在写入 45h（“S”）后，之前的状态是 10 这个值不能被检查到，因为ESC在达到这个状态时进入复位，导致状态00。

ESC 复位 PDI 寄存器（ESC_RST_PDI）

地址偏移：0x0041
复位值：0x0

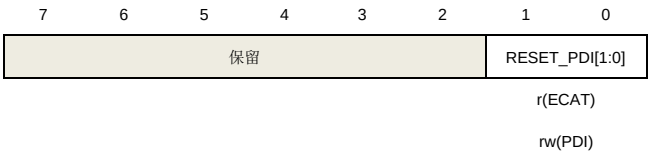
该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

写：



位/位域	名称	描述
7:0	RESET_PDI[7:0]	通过 3 个连续的命令向该寄存器中写入 52h（“R”）、45h（“E”）和 53h（“S”）之后，将触发复位。

读：

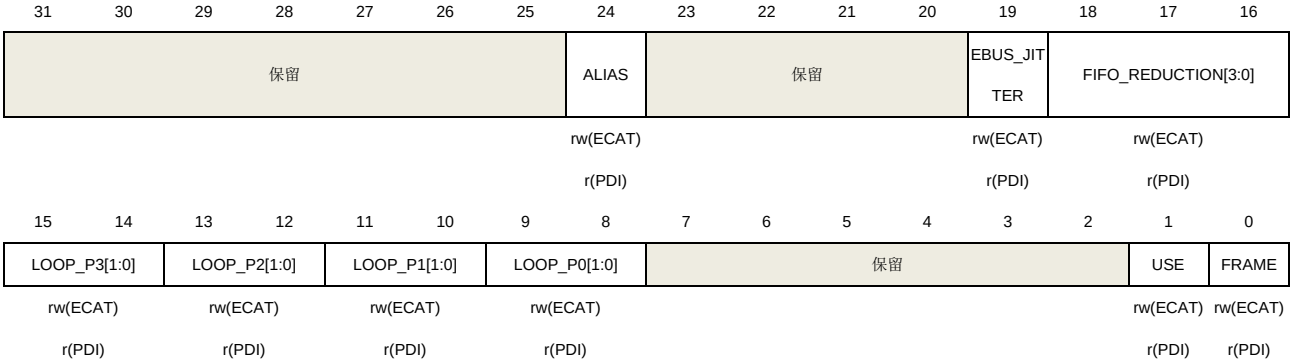


位/位域	名称	描述
7:2	保留	必须保持复位值
1:0	RESET_PDI[1:0]	复位步骤进度 00：初始化/复位状态 01：在写入 52h（“R”）后，之前的状态是 00 10：在写入 45h（“E”）后，之前的状态是 01 11：在写入 53h（“S”）后，之前的状态是 10 这个值不能被检查到，因为ESC在达到这个状态时进入复位，导致状态00。

ESC DL 控制寄存器（ESC_DL_CTL）

地址偏移：0x0100
复位值：0x7C001

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



位/位域	名称	描述
31:25	保留	必须保持复位值
24	ALIAS	站别名 0：忽略站别名 1：别名可用于所有已配置地址命令类型（FPRD和FPWR等）
23:19	保留	必须保持复位值
18:16	FIFO_REDUCTION[3:0]	接收 FIFO 大小/接收延时缩短 （ESC 延迟启动转发，直至 FIFO 至少半满） Value: MII: 0: -40 ns (-80 ns) 1: -40 ns (-80 ns) 2: -40 ns

		3: -40 ns 4: 无变化 5: 无变化 6: 无变化 7: 默认值 注意: EEPROM值仅在上电或复位后的第一次EEPROM负载时使用
15:14	LOOP_P3[1:0]	环路端口 3 00：自动 01：自动关闭 10：开启 11：关闭
13:12	LOOP_P2[1:0]	环路端口 2 00：自动 01：自动关闭 10：开启 11：关闭
11:10	LOOP_P1[1:0]	环路端口 1 00：自动 01：自动关闭 10：开启 11：关闭
9:8	LOOP_P0[1:0]	环路端口 0 00：自动 01：自动关闭 10：开启 11：关闭 注意:环路打开意味着在这个端口上发送/接收是启用的，环路关闭意味着发送/接收是禁用的，帧被转发到下一个内部打开的端口。 Auto：环路在断开时关闭，在连接时打开。 Auto Close：环路在断开时关闭，在链路恢复后(或在关闭端口收到有效的以太网帧)再次写入 01 打开。 Open:无论链路状态如何环路打开。 Closed:无论链路状态如何环路关闭。
7:2	保留	必须保持复位值
1	USE	临时使用寄存器 0x0100:0x0103[8:15]设置 0：永久使用 1：临时使用约1s，然后恢复之前的设置
0	FRAME	转发规则 0：转发非 EtherCAT 帧:处理 EtherCAT 帧，不处理不修改转发非 EtherCAT 帧。任何帧的源 MAC 地址都不会改变。

1：破坏非EtherCAT帧:处理EtherCAT帧，破坏非EtherCAT帧。源MAC地址由处理单元为每帧更改(SOURCE_MAC[1]设置为1---本地管理地址)。

ESC 物理读写偏移寄存器（ESC_PHYSICAL_RW_OFFSET）

地址偏移：0x0108
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RW_OFFSET[15:0]															
rw(ECAT)															
r(PDI)															

位/位域	名称	描述
15:0	RW_OFFSET[15:0]	这个寄存器用于设备寻址模式(FPRW, APRW, BRW)下的读写命令。内部读地址直接从 EtherCAT 数据报报头的偏移地址字段中获取，而内部写地址是通过在偏移地址字段中添加物理读/写偏移值来计算的。内部读地址= ADR，内部写地址= ADR + R/W-偏移

ESC DL 状态寄存器（ESC_DL_STAT）

地址偏移：0x0110
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DL_STAT US15	DL_STAT US14	DL_STAT US13	DL_STAT US12	DL_STAT US1`	DL_STAT US10	DL_STAT US9	DL_STAT US8	DL_STAT US7	DL_STAT US6	DL_STAT US5	DL_STAT US4	保留	DL_STAT US2	DL_STAT US1	DL_STAT US0
r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)		r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)

位/位域	名称	描述
15	DL_STATUS15	端口 3 上的通信 0：无稳定通信 1：已建立通信
14	DL_STATUS14	环路端口 3 0：开启 1：关闭
13	DL_STATUS13	端口 2 上的通信 0：无稳定通信 1：已建立通信
12	DL_STATUS12	环路端口 2 0：开启

		1：关闭
11	DL_STATUS11	端口 1 上的通信 0：无稳定通信 1：已建立通信
10	DL_STATUS10	环路端口 1 0：开启 1：关闭
9	DL_STATUS9	端口 0 上的通信 0：无稳定通信 1：已建立通信
8	DL_STATUS8	环路端口 0 0：开启 1：关闭
7	DL_STATUS7	端口 3 上的物理链路 0：无链路 1：检测到链路
6	DL_STATUS6	端口 2 上的物理链路 0：无链路 1：检测到链路
5	DL_STATUS5	端口 1 上的物理链路 0：无链路 1：检测到链路
4	DL_STATUS4	端口 0 上的物理链路 0：无链路 1：检测到链路
3	保留	必须保持复位值
2	DL_STATUS2	增强型链路检测 0：所有端口均禁止 1：至少一个端口激活 注意：仅在上电复位后第一次装载EEPROM时接受EEPROM值。
1	DL_STATUS1	PDI 看门狗状态 0：看门狗超时 1：看门狗重载
0	DL_STATUS0	PDI 工作/EEPROM 正确装载 0：EEPROM 未装载，PDI 未工作（未访问过程数据 RAM） 1：EEPROM正确装载，PDI工作（访问过程数据RAM）

ESC AL 控制寄存器（ESC_AL_CTL）

地址偏移：0x0120
复位值：0x1

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											ACK	STATE[3:0]			
											rw(ECAT)	rw(ECAT)			
											rw(PDI)	rw(PDI)			

位/位域	名称	描述
15:5	保留	必须保持复位值
4	ACK	错误指示应答 0：AL 状态寄存器中无错误指示应答 1：AL状态寄存器中有错误指示应答
3:0	STATE[3:0]	启动器件状态机的状态转换 1h：请求启动状态 2h：请求预工作状态 3h：请求自举状态 4h：请求安全工作状态 8h：请求工作状态 **注意**:PDI寄存器功能通过写命令确认被禁用:从PDI读取AL控制清除AL事件请求0x0220[0]。无法从PDI写入此寄存器。PDI寄存器功能通过写命令确认已启用:从PDI写AL控制清除AL事件请求0x0220[0]。从PDI写入这个寄存器是可以的；忽略写值(写0)。

注意：如果器件仿真关闭（0x0141[0]=0），则该寄存器的行为将类似于邮箱。ECAT写入该寄存器后，PDI必须读取该寄存器。否则，ECAT无法再次写入该寄存器。复位后，该寄存器可通过ECAT写入。(关于邮箱功能，AL控制寄存器的低字节和高字节都触发读/写功能，例如，读取0x121足以使该寄存器能够再次被写入。)如果器件仿真开启，AL控制寄存器可始终被写入，其内容复制到AL状态寄存器。

ESC AL 状态寄存器（ESC_AL_STAT）

地址偏移：0x0130
复位值：0x1

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											ERROR_IND	AL_STATE[3:0]			
											r(ECAT)	r(ECAT)			
											rw(PDI)	rw(PDI)			

位/位域	名称	描述
15:5	保留	必须保持复位值
4	ERROR_IND	错误指示 0：器件处于请求的状态或者标志由命令清零 1：本地操作后，器件未进入请求的状态或更改的状态
3:0	AL_STATE[3:0]	器件状态机的实际状态 1h：启动状态 2h：预工作状态 3h：自举状态 4h：安全工作状态 8h：工作状态

注意：如果设备仿真关闭(0x0141[0]=0)，AL状态寄存器只能从PDI写入，否则AL状态寄存器将反映AL控制寄存器的值。避免从PDI读取AL状态寄存器。

ESC AL 状态代码寄存器（ESC_AL_STAT_CODE）

地址偏移：0x0134
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AL_STATUS_CODE[15:0]															
r(ECAT)															
rw(PDI)															

位/位域	名称	描述
15:0	AL_STATUS_CODE[15:0]	状态代码

ESC 运行 LED 覆盖寄存器（ESC_RUN_LED_OVRD）

地址偏移：0x0138
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

7	6	5	4	3	2	1	0
保留			OVERRIDEN	LED_CODE[3:0]			
			rw(ECAT)	rw(ECAT)			
			rw(PDI)	rw(PDI)			

位/位域	名称	描述
7:5	保留	必须保持复位值
4	OVERRIDEN	运行改写

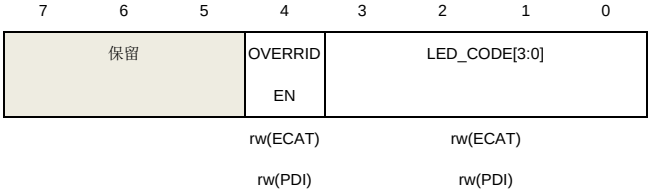
		0：禁止改写 1：使能改写	
3:0	LED_CODE[3:0]	LED 代码： 0x0：熄灭 0x1-0xC：闪烁 1x – 12x 0xD：快速闪烁 0xE：非等时闪烁 0xF：点亮	AL 状态： 启动 1 安全工作 4 预工作 2 自举 3 工作 8

注意：改变AL状态寄存器(0x0130)的有效值将禁用RUN LED覆盖(0x0138[4]=0)。在这个寄存器中读取的值总是反映当前的LED输出。

ESC 错误 LED 覆盖寄存器（ESC_ERR_LED_OVRD）

地址偏移：0x0139
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



位/位域	名称	描述
7:5	保留	必须保持复位值
4	OVERRIDEN	运行改写 0：禁止改写 1：使能改写
3:0	LED_CODE[3:0]	LED 代码： 0x0：关闭 0x1-0xC：闪烁 1x – 12x 0xD：快速闪烁 0xE：非等时闪烁 0xF：点亮

注意：新的错误条件将禁用ERR LED覆盖(0x0139[4]=0)。在这个寄存器中读取的值总是反映当前的LED输出。

ESC PDI 控制寄存器（ESC_PDI_CTL）

地址偏移量：0x0140
复位值：0x80

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

7	6	5	4	3	2	1	0
PDI_CTL[7:0]							
r(ECAT)							
r(PDI)							

位/位域	名称	描述
7:0	PDI_CTL[7:0]	过程数据接口 04h：数字 I/O 80h：片上总线 其他：保留

ESC 配置寄存器（ESC_CFG）

地址偏移：0x0141
复位值：取决于配置

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

7	6	5	4	3	2	1	0
EH_LINK_P3	EH_LINK_P2	EH_LINK_P1	EH_LINK_P0	CLK_LATCH_EN	CLK_SYNC_EN	EH_LINK_ALL	CTL_AL_STATUS
r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)

位/位域	名称	描述
7	EH_LINK_P3	增强型链路端口 3（默认为 1，EEPROM 加载后字为 0） 0：禁止（如果 bit 1 = 0） 1：使能
6	EH_LINK_P2	增强型链路端口 2（默认为 1，EEPROM 加载后字为 0） 0：禁止（如果 bit 1 = 0） 1：使能
5	EH_LINK_P1	增强型链路端口 1（默认为 1，EEPROM 加载后字为 0） 0：禁止（如果 bit 1 = 0） 1：使能
4	EH_LINK_P0	增强型链路端口 0（默认为 1，EEPROM 加载后字为 0） 0：禁止（如果 bit 1 = 0） 1：使能
3	CLK_LATCH_EN	分布式时钟锁存输入单元 0：禁止（节能） 1：使能
2	CLK_SYNC_EN	分布式时钟 SYNC 输出单元 0：禁止（节能）

		1：使能
1	EH_LINK_ALL	0：禁止（如果 bit[7:4] = 0）（默认为 1，EEPROM 加载后字为 0） 1：使能所有端口（覆写bit[7:4]）
0	CTL_AL_STATUS	器件仿真（控制 AL 状态寄存器） 0：AL 状态寄存器必须由 PDI 设置 1：AL 状态寄存器设置为写入 AL 控制寄存器的值 **注意**：对于数字I/O 模式，编程的值应为1；对于具有主机控制器的应用，值为0。

注意：Bits 1、4、5、6和7的EEPROM值只在上电或复位后的第一次EEPROM加载时传输到这个寄存器中。

ESC PDI 信息寄存器（ESC_PDI_INFO）

地址偏移：0x014E
复位值：取决于配置

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												CONFIGURE_EN	ACTIVE_EN	LOAD_EN	PDI_WREN
												r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)

位/位域	名称	描述
15:4	保留	必须保持复位值
3	CONFIGURE_EN	PDI 配置无效(默认为 0): 0：PDI 配置 ok 1：PDI配置无效
2	ACTIVE_EN	PDI 激活(默认为 0): 0：PDI 未激活 1：PDI激活
1	LOAD_EN	ESC 配置区从 EEPROM 加载(默认 0): 0：不加载 1：加载
0	PDI_WREN	PDI 功能写确认: 0：关闭 1：使能 **注意**：复位值取决于配置

ESC PDI 配置寄存器（ESC_PDI_CFG）

地址偏移：0x0150

复位值：取决于配置

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

7 6	5 4	3	2	1	0
OUT_DATA[1:0]	IN_DATA[1:0]	WATCHDOG	MODE	OUTVALID_MOD	POLARITY
r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)

位/位域	名称	描述
7:6	OUT_DATA[1:0]	输出数据采样选择 00：帧结束 01：保留 10：DC SYNC0 事件 11：DC SYNC1 事件 如果0x0150[1]=1，则在进程数据看门狗触发事件时更新输出数据(忽略0x0150[7:6])
5:4	IN_DATA[1:0]	输入数据采样选择 00：帧结束 01：LATCH_IN 的上升沿 10：DC SYNC0 事件 11：DC SYNC1事件
3	WATCHDOG	看门狗行为(默认0)： 0：输出在看门狗超时后立即复位 1：输出在看门狗超时后的下一个输出事件时复位
2	MODE	单向/双向模式(默认 0)： 0：单向模式：单独配置引脚的输入/输出方向 1：双向模式：所有I/O引脚均是双向的，忽略方向配置
1	OUTVALID_MOD	OUTVALID 模式(默认 0)： 0：输出事件信号传输 1：OUTVALID上的过程数据看门狗触发信号（WD_TRIG）传输。如果看门狗被触发，则会更新输出数据。改写0x0150[7:6]。
0	POLARITY	OUTVALID 极性(默认 0)： 0：高电平有效 1：低电平有效

注意：所有bit都可以通过EEPROM配置。

ESC 同步/锁存器配置寄存器（ESC_SL_CFG）

地址偏移量：0x0151
复位值：取决于配置

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

7	6	5	4	3	2	1	0
SYNC1_MAP	SL1_CONFIG	SYNC1_POLARITY[0:1]		SYNC0_MAP	SL0_CONFIG	SYNC1_POLARITY[1:0]	
r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)		r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	

位/位域	名称	描述
7	SYNC1_MAP	SYNC1 映射到 AL 事件请求寄存器 0x0220[3]： 0：禁止 1：使能
6	SL1_CONFIG	SYNC1/LATCH1 配置(默认 1) 0：LATCH1 输入 1：SYNC1 输出 注意：ESC具有SYNC[1:0]输出和LATCH[1:0]输入，独立于此配置
5:4	SYNC1_POLARITY[0:1]	SYNC1 输出驱动器/极性(默认 10) 00：推挽式（低电平有效） 01：漏极开路（低电平有效） 10：推挽式（高电平有效） 11：源极开路（高电平有效）
3	SYNC0_MAP	SYNC0 映射到 AL 事件请求寄存器 0x0220[2]： 0：禁止 1：使能
2	SL0_CONFIG	SYNC0/LATCH0 配置(默认 1) 0：LATCH0 输入 1：SYNC0 输出 注意：ESC具有SYNC[1:0]输出和LATCH[1:0]输入，独立于此配置
1:0	SYNC0_POLARITY[1:0]	SYNC0 输出驱动器/极性(默认 10) 00：推挽式（低电平有效） 01：漏极开路（低电平有效） 10：推挽式（高电平有效） 11：源极开路（高电平有效）

ESC PDI 扩展配置寄存器（ESC_PDI_EXT_CFG）

地址偏移：0x0152
复位值：取决于配置

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIR15	DIR15	DIR15	DIR15	DIR15	DIR15	DIR15	DIR8	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0
r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)

位/位域	名称	描述
15	DIR15	I/O 方向[31:30]成对配置为输入或输出： 参考DIR0描述
14	DIR14	I/O 方向[29:28]成对配置为输入或输出： 参考DIR0描述
13	DIR13	I/O 方向[27:26]成对配置为输入或输出： 参考DIR0描述
12	DIR12	I/O 方向[25:24]成对配置为输入或输出： 参考DIR0描述
11	DIR11	I/O 方向[23:22]成对配置为输入或输出： 参考DIR0描述
10	DIR10	I/O 方向[21:20]成对配置为输入或输出： 参考DIR0描述
9	DIR9	I/O 方向[19:18]成对配置为输入或输出： 参考DIR0描述
8	DIR8	I/O 方向[17:16]成对配置为输入或输出： 参考DIR0描述
7	DIR7	I/O 方向[15:14]成对配置为输入或输出： 参考DIR0描述
6	DIR6	I/O 方向[13:12]成对配置为输入或输出： 参考DIR0描述
5	DIR5	I/O 方向[11:10]成对配置为输入或输出： 参考DIR0描述
4	DIR4	I/O 方向[9:8]成对配置为输入或输出： 参考DIR0描述
3	DIR3	I/O 方向[7:6]成对配置为输入或输出： 参考DIR0描述
2	DIR2	I/O 方向[5:4]成对配置为输入或输出： 参考DIR0描述
1	DIR1	I/O 方向[3:2]成对配置为输入或输出： 参考DIR0描述
0	DIR0	I/O 方向[1:0]成对配置为输入或输出： 0：输入 1：输出 备注:在双向模式下保留，设置为0。预留不可用的I/O配置位，将EEPROM值设置为

0。

ESC 事件屏蔽寄存器（ESC_EVT_MSK）

地址偏移：0x0200
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT_REQ[15:0]															
rw(ECAT)															
r(PDI)															

位/位域	名称	描述
15:0	EVENT_REQ[15:0]	ECAT 事件请求寄存器事件的 ECAT 事件屏蔽，用于映射到 EtherCAT 帧的 ECAT 事件字段。 0：不映射相应的 ECAT 事件请求寄存器位 1：映射相应的 ECAT 事件请求寄存器位

ESC PDI AL 事件寄存器（ESC_PDI_AL_EVT）

地址偏移：0x0204
复位值：0x00FFFF0F

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AL_EVENT_REQ[31:16]															
r(ECAT)															
rw(PDI)															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AL_EVENT_REQ[15:0]															
r(ECAT)															
rw(PDI)															

位/位域	名称	描述
31:0	AL_EVENT_REQ[31:0]	AL 事件请求寄存器事件的 AL 事件屏蔽，用于映射到 PDI IRQ 信号。 0：不映射相应的 AL 事件请求寄存器位 1：映射相应的 AL 事件请求寄存器位

ESC 事件请求寄存器（ESC_EVT_REQ）

地址偏移：0x0210
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

保留	CHANNEL 7	CHANNEL 6	CHANNEL 5	CHANNEL 4	CHANNEL 3	CHANNEL 2	CHANNEL 1	CHANNEL 0	AL_STAT US	DL_STAT US	保留	DC_LATC H
	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)		r(ECAT) r(PDI)

位/位域	名称	描述
15:12	保留	必须保持复位值
11	CHANNEL7	SyncManager 状态镜像 0：无同步通道 7 事件 1：同步通道7事件待处理
10	CHANNEL6	SyncManager 状态镜像 0：无同步通道 6 事件 1：同步通道6事件待处理
9	CHANNEL5	SyncManager 状态镜像 0：无同步通道 5 事件 1：同步通道5事件待处理
8	CHANNEL4	SyncManager 状态镜像 0：无同步通道 4 事件 1：同步通道4事件待处理
7	CHANNEL3	SyncManager 状态镜像 0：无同步通道 3 事件 1：同步通道3事件待处理
6	CHANNEL2	SyncManager 状态镜像 0：无同步通道 2 事件 1：同步通道2事件待处理
5	CHANNEL1	SyncManager 状态镜像 0：无同步通道 1 事件 1：同步通道1事件待处理
4	CHANNEL0	SyncManager 状态镜像 0：无同步通道 0 事件 1：同步通道0事件待处理
3	AL_STATUS	AL 状态事件： 0：AL 状态无变化 1：AL 状态变化 （通过从ECAT读出AL状态0x0130:0x0131来清除位）
2	DL_STATUS	DL 状态事件： 0：DL 状态无变化 1：DL 状态变化 （通过从ECAT读出DL状态0x0110:0x0111来清除位）

1	保留	必须保持复位值
0	DC_LATCH	DC 锁存事件： 0：DC 锁存输入无变化 1：至少有一个 DC 锁存输入变化 （该位的清零方式是通过ECAT读取ECAT控制的锁存单元的DC锁存事件的时间，以使0x09AE:0x09AF指示无事件）

ESC AL 事件请求寄存器（ESC_AL_EVT_REQ）

地址偏移：0x0220

复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								SYNC15_INT	SYNC14_INT	SYNC13_INT	SYNC12_INT	SYNC11_INT	SYNC10_INT	SYNC9_INT	SYNC8_INT
								r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNC7_INT	SYNC6_INT	SYNC5_INT	SYNC4_INT	SYNC3_INT	SYNC2_INT	SYNC1_INT	SYNC0_INT	保留	PDATA	EEPROM	SYNC_CHANGE	DC_SYNC1	DC_SYNC0	DC_LATCH	AL_CONTROL
r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)		r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)

位/位域	名称	描述
31:24	保留	必须保持复位值
23	SYNC15_INT	SyncManager 中断（SyncManager 寄存器偏移 0x5，bit 0 或 bit 1） 0：无 SyncManager 15 中断 1：SyncManager 15中断待处理
22	SYNC14_INT	SyncManager 中断（SyncManager 寄存器偏移 0x5，bit 0 或 bit 1） 0：无 SyncManager 14 中断 1：SyncManager 14中断待处理
21	SYNC13_INT	SyncManager 中断（SyncManager 寄存器偏移 0x5，bit 0 或 bit 1） 0：无 SyncManager 13 中断 1：SyncManager 13中断待处理
20	SYNC12_INT	SyncManager 中断（SyncManager 寄存器偏移 0x5，bit 0 或 bit 1） 0：无 SyncManager 12 中断 1：SyncManager 12中断待处理
19	SYNC11_INT	SyncManager 中断（SyncManager 寄存器偏移 0x5，bit 0 或 bit 1） 0：无 SyncManager 11 中断 1：SyncManager 11中断待处理
18	SYNC10_INT	SyncManager 中断（SyncManager 寄存器偏移 0x5，bit 0 或 bit 1）

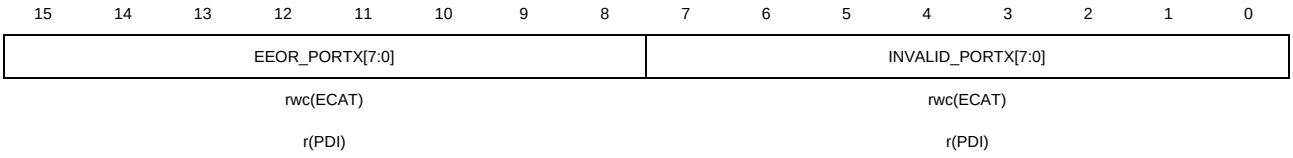
		0：无 SyncManager 10 中断 1：SyncManager 10中断待处理
17	SYNC9_INT	SyncManager 中断（SyncManager 寄存器偏移 0x5，bit 0 或 bit 1） 0：无 SyncManager 9 中断 1：SyncManager 9中断待处理
16	SYNC8_INT	SyncManager 中断（SyncManager 寄存器偏移 0x5，bit 0 或 bit 1） 0：无 SyncManager 8 中断 1：SyncManager 8中断待处理
15	SYNC7_INT	SyncManager 中断（SyncManager 寄存器偏移 0x5，bit 0 或 bit 1） 0：无 SyncManager 7 中断 1：SyncManager 7中断待处理
14	SYNC6_INT	SyncManager 中断（SyncManager 寄存器偏移 0x5，bit 0 或 bit 1） 0：无 SyncManager 6 中断 1：SyncManager 6中断待处理
13	SYNC5_INT	SyncManager 中断（SyncManager 寄存器偏移 0x5，bit 0 或 bit 1） 0：无 SyncManager 5 中断 1：SyncManager 5中断待处理
12	SYNC4_INT	SyncManager 中断（SyncManager 寄存器偏移 0x5，bit 0 或 bit 1） 0：无 SyncManager 4 中断 1：SyncManager 4中断待处理
11	SYNC3_INT	SyncManager 中断（SyncManager 寄存器偏移 0x5，bit 0 或 bit 1） 0：无 SyncManager 3 中断 1：SyncManager 3中断待处理
10	SYNC2_INT	SyncManager 中断（SyncManager 寄存器偏移 0x5，bit 0 或 bit 1） 0：无 SyncManager 2 中断 1：SyncManager 2中断待处理
9	SYNC1_INT	SyncManager 中断（SyncManager 寄存器偏移 0x5，bit 0 或 bit 1） 0：无 SyncManager 1 中断 1：SyncManager 1中断待处理
8	SYNC0_INT	SyncManager 中断（SyncManager 寄存器偏移 0x5，bit 0 或 bit 1） 0：无 SyncManager 0 中断 1：SyncManager 0中断待处理
7	保留	必须保持复位值
6	PDATA	看门狗过程数据： 0：未超时 1：已超时 （该位的清零方式是通过PDI读取看门狗状态过程数据0x0440。）
5	EEPROM	EEPROM 仿真

		0：无命令待处理 1：EEPROM 命令待处理 （该位的清零方式是通过PDI应答EEPROM控制/状态寄存器0x0502:0x0503[10:8]中的命令。）
4	SYNC_CHANGE	SyncManager 激活寄存器变化（SyncManager 激活寄存器） 0：任何 SyncManager 中均无变化 1：至少一个 SyncManager 发生变化 （该位的清零方式是通过PDI读取相应的SyncManager激活寄存器0x0806。）
3	DC_SYNC1	DC SYNC1 的状态（如果 0x0151 寄存器的 bit 7 = 1） （该位的清零方式是通过PDI读取SYNC1状态0x098F，仅在确认模式下使用。）
2	DC_SYNC0	DC SYNC0 的状态（如果 0x0151 寄存器的 bit 3 = 1） （该位的清零方式是通过PDI读取SYNC0状态0x098E，仅在确认模式下使用。）
1	DC_LATCH	DC 锁存事件 0：DC 锁存输入无变化 1：至少有一个 DC 锁存输入变化 （该位的清零方式是通过PDI读取PDI控制的锁存单元的DC锁存事件的时间，以使0x09AE:0x09AF指示无事件。）
0	AL_CONTROL	AL 控制事件 0：AL 控制寄存器无变化 1：已写入 AL 控制寄存器（AL 控制事件仅在 PDI 仿真关闭时生成(ESC 配置 0 寄存器 0x0141[0]=0)） （该位的清零方式是通过PDI读取AL控制寄存器0x0120:0x0121）

PortX 接收错误计数器寄存器（RX_PORTX_ERR）（X = 0,1,2,3）

地址偏移：0x0300 + X * 2
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



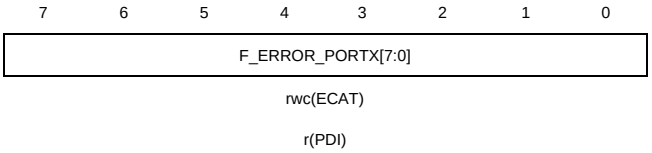
位/位域	名称	描述
15:8	EEOR_PORTX[7:0]	端口 X 接收错误计数器 当达到FFh时，计数停止。端口X接收错误计数
7:0	INVALID_PORTX[7:0]	端口 X 无效帧计数器 当达到FFh时，计数停止。端口X无效帧计数

注意：如果写入任意一个的RX错误计数器0x0300-0x030B(最好是0x0300)，则清除错误计数器0x0300- 0x030B，忽略写值(写0)，只有当端口环路打开时才会计算错误。

Port X 转发接收错误计数器寄存器（FWD_PORTX_ERR）（X = 0,1,2,3）

地址偏移：0x0308 + X
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



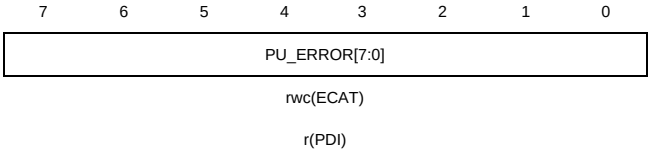
位/位域	名称	描述
7:0	F_ERROR_PORTX[7:0]	当达到 FFh 时，计数停止。端口 X 转发的接收错误计数器

注意：如果写入任意一个的RX错误计数器0x0300-0x030B(最好是0x0300)，则清除错误计数器0x0300- 0x030B，忽略写值(写0)，只有当端口环路打开时才会计算错误。

ESC 处理单元错误计数器寄存器（ESC_PU_ERR）

地址偏移：0x030C
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



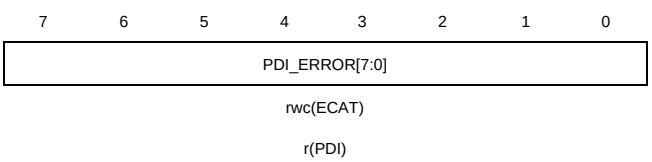
位/位域	名称	描述
7:0	PU_ERROR[7:0]	ECAT处理单元错误计数器(当到达0xFF时停止计数)。计数通过处理单元的帧的错误。

注意：如果写入错误计数器0x030C，则错误计数器0x030C会被清除。写入的值将被忽略（写入0）。

ESC PDI 错误计数器寄存器（ESC_PDI_ERR）

地址偏移：0x030D
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



位/位域	名称	描述
7:0	PDI_ERROR[7:0]	PDI 错误计数器(当到达 0xFF 时停止计数)。计数 PDI 访问是否有接口错误。

注意：如果写入错误计数器0x030D，则清除错误计数器0x030D和错误代码0x030E:0x030F。写入值将被忽略（写入0）。

ESC PortX 丢失链路计数器寄存器（ESC_PORTX_LOST_LINK）（X = 0,1,2,3）

地址偏移：0x0310 + X
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

7	6	5	4	3	2	1	0
LOST_LINK[7:0]							
rwc(ECAT)							
r(PDI)							

位/位域	名称	描述
7:0	LOST_LINK[7:0]	端口 X 的链路丢失计数器（当计数达到 0xFF 时停止计数）。仅当端口处于开启状态且环路模式为自动时进行计数。

注意：如果写入其中一个已实现的丢失链接计数器0x0310-0x0313（最好是0x0310），则0x0310-0x0313的丢失链接计数器将被清除。写入值将被忽略（写入0）。

ESC 看门狗分频器寄存器（ESC_WDG_DIV）

地址偏移：0x0400
复位值：0x09C2

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WATCHDOG_DIV[15:0]															
rw(ECAT)															
r(PDI)															

位/位域	名称	描述
15:0	WATCHDOG_DIV[15:0]	看门狗分频器：25 MHz 计时周期数（减 2），表示基本看门狗递增次数。（默认值为 100 µs = 2498）

ESC 看门狗定时器 PDI 寄存器（ESC_WDG_TIME_PDI）

地址偏移：0x0410
复位值：0x03E8

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

rw(ECAT)

r(PDI)

位/位域	名称	描述
15:0	WATCHDOG_TIME[15:0]	看门狗时间 PDI：基本看门狗递增次数。（100 µs 看门狗分频器默认值对应于 100 ms 看门狗。）

注意：如果看门狗时间设置为0x0000，则看门狗将被禁用。每次PDI访问后，看门狗都会重新开始计数。

ESC 看门狗定时器过程数据寄存器（ESC_WDG_TIME_PDATA）

地址偏移：0x0420
复位值：0x03E8

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WATCHDOG_TPD[15:0]															
rw(ECAT)															
r(PDI)															

位/位域	名称	描述
15:0	WATCHDOG_TPD[15:0]	看门狗时间过程数据：基本看门狗递增次数。（100 µs 看门狗分频器默认值对应于 100 ms 看门狗。）

ESC 看门狗状态过程数据寄存器（ESC_WDG_STAT_PDATA）

地址偏移：0x0440
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															WATCHDOG_STA
															r(ECAT)
															rw(PDI)

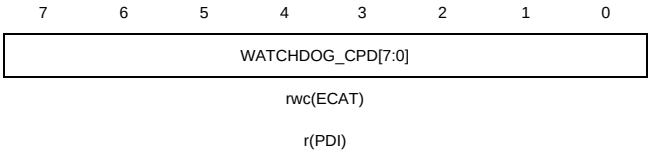
位/位域	名称	描述
15:1	保留	必须保持复位值
0	WATCHDOG_STA	看门狗过程数据的状态（由 SyncManager 触发） 0：看门狗过程数据超时 1：看门狗过程数据激活或禁止 注意：PDI 寄存器功能通过写命令确认被禁用:从 PDI 读取该寄存器清除 AL 事件请求 0x0220[6]。无法从 PDI 写入此寄存器。PDI 寄存器功能通过写命令确认已启用:从 PDI 写入该寄存器清除 AL 事件请求 0x0220[6]。从 PDI 写入这个寄存器是可能的;

忽略写值(写 0)。

ESC 看门狗计数器过程数据寄存器（ESC_WDG_CNT_PDATA）

地址偏移：0x0442
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

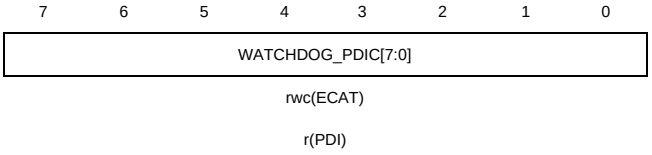


位/位域	名称	描述
7:0	WATCHDOG_CPD[7:0]	看门狗计数器过程数据（当达到 FFh 时，计数停止）。计数以确定过程数据看门狗是否超时。

ESC 看门狗计数器 PDI 寄存器（ESC_WDG_CNT_PDI）

地址偏移：0x0443
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



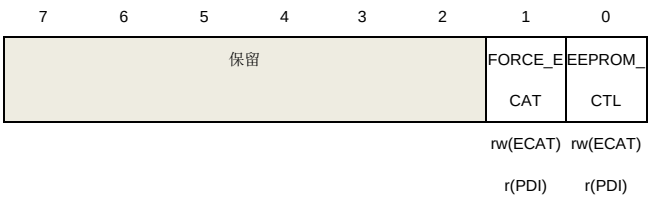
位/位域	名称	描述
7:0	WATCHDOG_PDIC[7:0]	看门狗 PDI 计数器（当达到 FFh 时，计数停止）。计数以确定 PDI 看门狗是否超时。

注意：如果写入了任何一个看门狗计数器0x0442-0x0443，那么0x0442-0x0443看门狗计数器都会被清零。写入值将被忽略（写入0）。

ESC EEPROM 配置寄存器（ESC_EEPROM_CFG）

地址偏移：0x0500
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

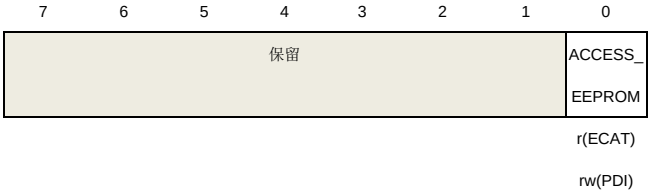


位/位域	名称	描述
7:2	保留	必须保持复位值
1	FORCE_ECAT	强制 ECAT 访问 0：不改变 0x0501 bit 0 1：复位 0x0501bit 0
0	EEPROM_CTL	PDI EEPROM 控制 0：无 1：有（PDI 具有 EEPROM 控制）

ESC EEPROM PDI 访问寄存器（ESC_EEPROM_ACC）

地址偏移：0x0501
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



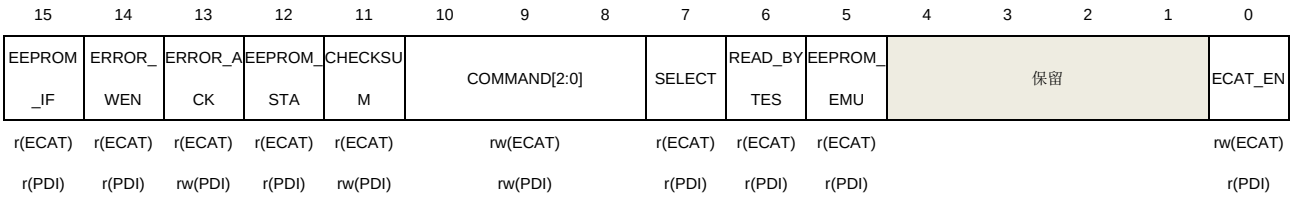
位/位域	名称	描述
7:1	保留	必须保持复位值
0	ACCESS_EEPROM	EEPROM 访问 0：PDI 释放 EEPROM 访问 1：PDI采用EEPROM访问(PDI具有EEPROM控制)

注意：仅当(0x0500[0]=1 或 0x0501[0]=1)且0x0500[1]=0时，才允许写入访问。

ESC EEPROM 控制 / 状态寄存器（ESC_EEPROM_CTL）

地址偏移：0x0502
复位值：取决于配置

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



位/位域	名称	描述
15	EEPROM_IF	繁忙（默认 0） 0：EEPROM 接口空闲

		1：EEPROM 接口繁忙
14	ERROR_WEN	错误写使能（默认 0） 0：无错误 1：未进行写使能的写命令 **注意**：通过向命令寄存器bit[10:8]位写入“000”（或任何有效命令）将错误位清零。
13	ERROR_ACK	错误应答/命令（默认 0） 0：无错误 1：丢失 EEPROM 应答或无效命令 仅 EEPROM 仿真：如果发生临时故障，则 PDI 写为 1。 **注意**：通过向命令寄存器bit[10:8]位写入“000”（或任何有效命令）将错误位清零。
12	EEPROM_STA	EEPROM 装载状态（默认 0） 0：EEPROM 已装载，器件信息可用 1：EEPROM未装载，器件信息不可用（正在装载EEPROM或者EEPROM装载失败）
11	CHECKSUM	ESC 配置区域中的校验和错误 0：校验和正确 1：校验和错误 仅针对ESC的EEPROM仿真:如果重载命令发生CRC失败，PDI将写1。
10:8	COMMAND[2:0]	命令寄存器（默认 0） 写：启动命令 读：当前执行的命令 000：无命令/EEPROM 空闲（清零错误位） 001：读 010：写 100：重载 其他：保留/无效命令（不发出） 仅 EEPROM 仿真:执行后，PDI 写入命令值以指示操作准备就绪。 **注意**：执行命令后（EEPROM 繁忙结束），命令寄存器位将自清零。向命令寄存器位写入“000”还会将错误位bit 14:13清零。如果错误应答/命令待处理（bit13），则命令寄存器位bit10:8将被忽略。
7	SELECT	选择的 EEPROM 算法 0：1 个地址字节（1 Kbit - 16 Kbit EEPROM） 1：2个地址字节（32 Kbit - 4 Mbit EEPROM）
6	READ_BYTES	支持的 EEPROM 字节数（默认 0） 0：4 字节 1：8字节
5	EEPROM_EMU	EEPROM 仿真 0：正常工作（使用 I2C 接口） 1：PDI仿真EEPROM（未使用I2C）
4:1	保留	必须保持复位值

0	ECAT_EN	ECAT 写使能（默认 0） 0：禁止写请求 1：使能写请求 如果 PDI 有 EEPROM 控制，这个位总是 1。 **注意**：执行命令后（EEPROM 繁忙结束），命令寄存器位将自清零。向命令寄存器位写入“000”还会将错误位bit 14:13清零。如果错误应答/命令待处理（bit13），则命令寄存器位bit10:8将被忽略。

ESC EEPROM 地址寄存器（ESC_EEPROM_ADDR）

地址偏移：0x0504
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

位/位域	名称	描述
31:0	EEPROM_ADDR[31:0]	EEPROM 地址 Bit 0：第一个字（16 位） Bit 1：第二个字 ..... 实际使用的 EEPROM 地址位： [9:0]：EEPROM 大小最大为 16 Kbit [17:0]：EEPROM 大小为 32 Kbit - 4 Mbit [31:0]：EEPROM仿真

注意：写入访问取决于EEPROM接口（ECAT/PDI）的分配。如果EEPROM接口忙碌（0x0502[15]=1），则写入访问将被阻止。

ESC EEPROM 数据寄存器（ESC_EEPROM_DATA）

地址偏移：0x0508
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

EERPOM_WRITE[15:0]
rw(ECAT)
rw(PDI)

位/位域	名称	描述
31:16	EEPROM_READ[15:0]	EEPROM读数据（要从EEPROM读取的数据的高字节）
15:0	EERPOM_WRITE[15:0]	EEPROM 读/写数据 要从EEPROM读取的数据的低字节或者要写入EEPROM的数据

ESC MII 管理控制/状态寄存器（ESC_MII_MNG_CS）

地址偏移：0x0510

复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MII_IF	COMMAND_ERROR	READ_EEOR	保留			COMMAND[1:0]		PHY_ADDRP0[4:0]					MII_CONFIG	PDI_CTL	WRITE_EN
r(ECAT) r(PDI)	r(ECAT) r(PDI)	rw(ECAT) rw(PDI)				rw(ECAT) rw(PDI)		r(ECAT) r(PDI)					r(ECAT) r(PDI)	r(ECAT) r(PDI)	rw(ECAT) r(PDI)

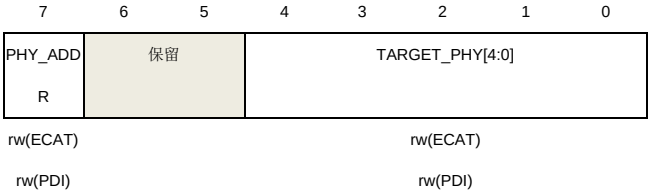
位/位域	名称	描述
15	MII_IF	繁忙 0：MI 控制状态机空闲 1：MI控制状态机工作
14	COMMAND_ERROR	命令错误 0：最后一条命令成功 1：无效命令或未进行写使能的写命令 通过无效命令或通过向命令寄存器bits[9:8]写入“00”清零。
13	READ_EEOR	读错误 0：无读错误 1：出现读错误（PHY 或寄存器不可用） 通过写入该寄存器清零。
12:10	保留	必须保持复位值
8:9	COMMAND[1:0]	命令寄存器 写：启动命令 读：当前执行的命令 00：无命令/MI 空闲（清零错误位） 01：读 10：写 11：保留（不发出） 注意：写使能bit 0在下一帧的SOF处自清除，命令bits[10:8]在命令执行后自清除(繁

		忙结束)。将“00”写入命令寄存器也将清除错误bits[14:13]。命令执行完成后，命令位将被清除。
7:3	PHY_ADDRP0[4:0]	端口 0 的 PHY 地址 （如果PHY地址是连续的，则等于PHY地址偏移量）
2	MII_CONFIG	MII 链路检测和配置： 0：禁用所有端口 1：至少启用一个MII端口，请参考PHY端口状态（0x0518 ff.）了解详细信息。
1	PDI_CTL	管理接口可以通过PDI（寄存器0x0516-0x0517）进行控制： 0：仅ECAT控制 1：PDI控制可能
0	WRITE_EN	写入使能： 0：写入禁用 1：写入启用 如果 PDI 控制了 MI（管理接口），这个位总是 1。 **注意：**写入使能位0在下一帧的SOF（帧开始）时自动清零，命令位[10:8]在命令执行后（忙碌结束）自动清零。向命令寄存器写入“00”也会清除错误位[14:13]。命令位在命令执行后被清除。

ESC PHY 地址寄存器（ESC_PHY_ADDR）

地址偏移：0x0512
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



位/位域	名称	描述
7	PHY_ADDR	目标PHY地址转换： 0：启用 1：禁用 详情请参考0x0512[4:0]和0x0510[7:3]。
6:5	保留	必须保持复位值。
4:0	TARGET_PHY[4:0]	目标PHY地址转换 当0x0512[7]=0时： 0-3：目标PHY地址0-3用于访问端口0-3的PHY，当PHY地址正确配置时 4-31：当访问PHY时，端口0的配置PHY地址（PHY地址偏移）将被加到目标PHY地址值4-31上 当0x0512[7]=1时：

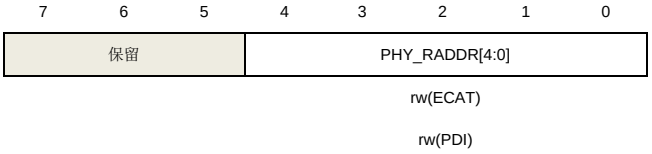
0-31：在不进行转换的情况下使用目标PHY地址访问PHY。

注意：写入访问取决于MII（ECAT/PDI）的分配。如果管理接口忙（0x0510[15]=1），则写入访问被阻止。

ESC PHY 寄存器地址寄存器（ESC_PHY_REG_ADDR）

地址偏移：0x0513
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

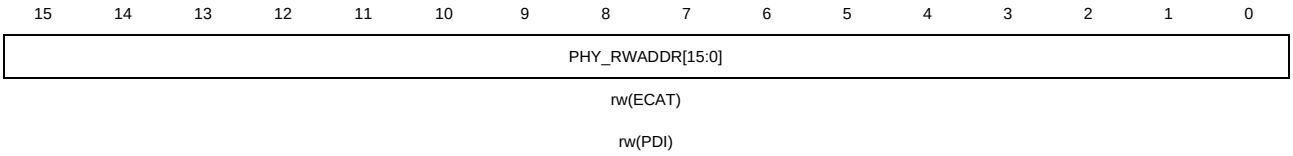


位/位域	名称	描述
7:5	保留	必须保持复位值。
4:0	PHY_RADDR[4:0]	要读取/写入的PHY寄存器的地址

ESC PHY 数据寄存器（ESC_PHY_DATA）

地址偏移：0x0514
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

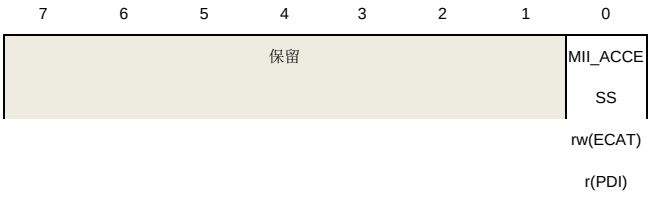


位/位域	名称	描述
15:0	PHY_RWADDR[15:0]	PHY读/写数据

MII 管理 ECAT 访问状态寄存器（MIIM_ECAT_ACC_STAT）

地址偏移：0x0516
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



位/位域	名称	描述
7:1	保留	必须保持复位值。
0	MII_ACCESS	访问MII管理： 0：ECAT使能PDI接管MII管理接口 1：ECAT声明对MII管理接口的独占访问权

注意：只有当0x0517[0]=0时，才允许写入访问。

MII 管理 PDI 访问状态寄存器（MIIM_PDI_ACC_STAT）

地址偏移：0x0517
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

7	6	5	4	3	2	1	0
保留						PDI_ACCESS	MII_ACCESS
						rw(ECAT) r(PDI)	r(ECAT) rw(PDI)

位/位域	名称	描述
7:2	保留	必须保持复位值。
1	PDI_ACCESS	强制PDI访问状态： 0：不改变0x0517[0]位 1：将0x0517[0]位复位为0
0	MII_ACCESS	访问MII管理： 0：ECAT可以访问MII管理 1：PDI可以访问MII管理

PHY Port X 状态寄存器（PHY_PORTX_STAT）（X = 0,1,2,3）

地址偏移：0x0518 + X
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

7	6	5	4	3	2	1	0
保留		PHY_UPDATE	PARTNER_ERROR	READ_ERROR	STATUS_ERROR	LINK_STATUS	PHYLINK
		rwc(ECAT) rwc(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	rwc(ECAT) rwc(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)

位/位域	名称	描述
7:6	保留	必须保持复位值。
5	PHY_UPDATE	PHY配置更新：

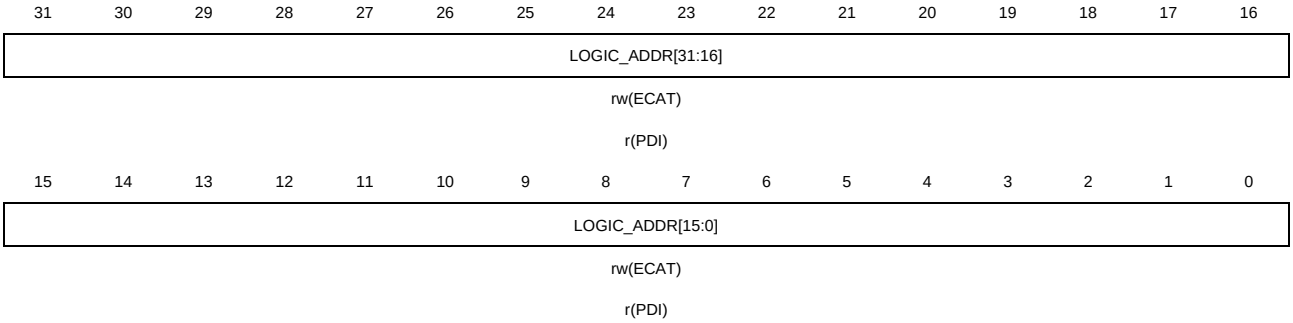
0：无更新
1：PHY配置已更新
通过向任何一个PHY端口X状态寄存器写入任意值来清除。

4	PARTNER_ERROR	链路伙伴错误： 0：未检测到错误 1：链路伙伴错误
3	READ_ERROR	读取错误： 0：未发生读取错误 1：发生了读取错误 通过向任何一个PHY端口X状态寄存器写入任意值来清除。
2	STATUS_ERROR	链路状态错误： 0：无错误 1：链路错误，链路被禁止
1	LINK_STATUS	链路状态（100 Mbit/s，全双工，自动协商）： 0：无链路 1：检测到链路
0	PHYLINK	物理链路状态： 0：无物理链路 1：检测到物理链路

FMMUX 逻辑起始地址寄存器（FMMUX_LOGIC_ADDR）（X = 0…F）

地址偏移量：0x0600 + X * 0x10
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

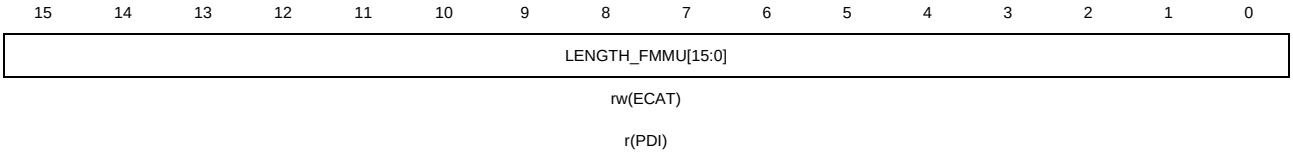


位/位域	名称	描述
31:0	LOGIC_ADDR[31:0]	EtherCAT地址空间内的逻辑起始地址。

FMMUX 长度寄存器（FMMUX_LENGTH）（X = 0…F）

地址偏移量：0x0604 + X * 0x10
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

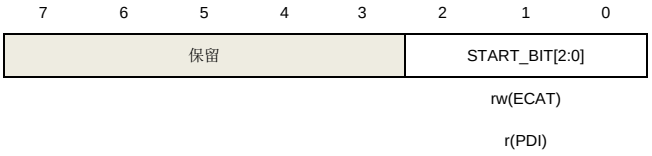


位/位域	名称	描述
15:0	LENGTH_FMMU[15:0]	从第一个逻辑FMMU字节到最后一个FMMU字节加1的偏移量（例如，如果使用了两个字节，则此参数应包含2）

FMMUX 逻辑起始位寄存器（FMMUX_STRA_BIT）（X = 0...F）

地址偏移：0x0606 + X * 0x10
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

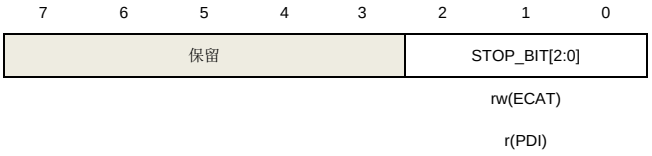


位/位域	名称	描述
7:3	保留	必须保持复位值。
2:0	START_BIT[2:0]	需要映射的逻辑起始位（位从最低有效位0开始计数到最高有效位7）

FMMUX 逻辑停止位寄存器（FMMUX_STOP_BIT）（X = 0...F）

地址偏移：0x0607 + X * 0x10
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



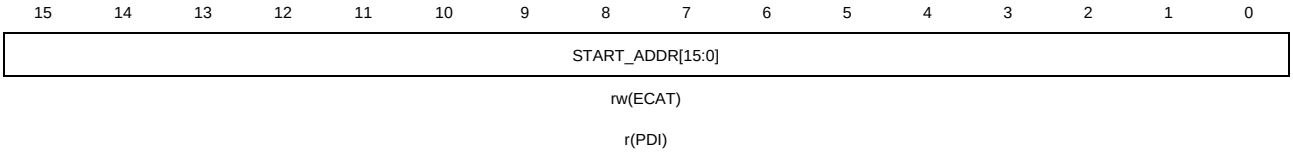
位/位域	名称	描述
7:3	保留	必须保持复位值。
2:0	STOP_BIT[2:0]	需要映射的最后一个逻辑位（位从最低有效位0开始计数到最高有效位7）

FMMUX 物理起始地址寄存器（FMMUX_ADDR）（X = 0...F）

地址偏移：0x0608 + X * 0x10

复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

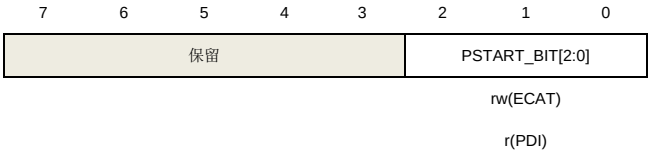


位/位域	名称	描述
15:0	START_ADDR[15:0]	物理起始地址（映射到逻辑起始地址）

FMMUX 物理起始位寄存器（FMMUX_PSBIT）（X = 0…F）

地址偏移：0x060A + X * 0x10
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

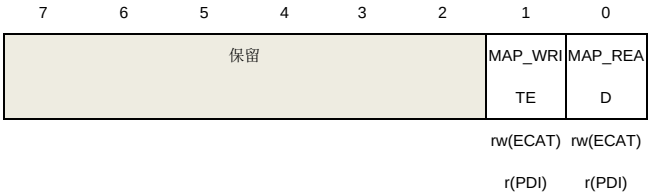


位/位域	名称	描述
7:3	保留	必须保持复位值。
2:0	PSTART_BIT[2:0]	物理起始位作为逻辑起始位映射的目标（位从最低有效位0开始计数到最高有效位7）

FMMUX 类型寄存器（FMMUX_TYPE）（X = 0…F）

地址偏移量：0x060B + X * 0x10
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



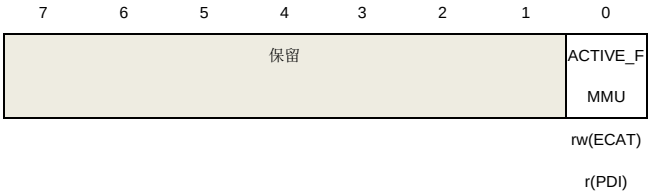
位/位域	名称	描述
7:2	保留	必须保持复位值。
1	MAP_WRITE	0: 忽略写访问的映射 1: 写访问时使用映射
0	MAP_READ	0: 忽略读访问的映射 1: 读访问时使用映射

FMMUX 激活寄存器（FMMUX_ACTIVE）（X = 0…F）

地址偏移：0x060C + X * 0x10

复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



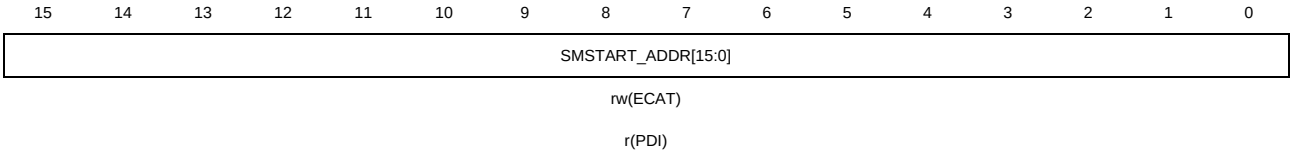
位/位域	名称	描述
7:1	保留	必须保持复位值。
0	ACTIVE_FMMU	0: FMMU未激活 1: FMMU已激活。FMMU根据配置的映射检查逻辑地址的块是否需要映射

SyncManager X 物理起始地址寄存器（SMX_START_ADDR）（X = 0…F）

地址偏移量：0x0800 + X * 8

复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



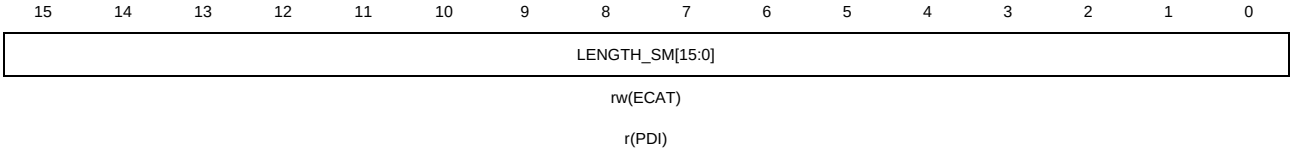
位/位域	名称	描述
15:0	SMSTART_ADDR[15:0]	将由SyncManager处理的第一个字节

SyncManager X 长度寄存器（SMX_LENGTH）（X = 0…F）

地址偏移：0x0802 + X * 8

复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



位/位域	名称	描述
15:0	LENGTH_SM[15:0]	分配给SyncManager的字节数（应大于1，否则SyncManager不会被激活。如果设置

为1，只有在配置了看门狗触发器时才会生成）

SyncManager X 控制寄存器（SMX_CTL）（X = 0…F）

地址偏移量：0x0804 + X * 8
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

7	6	5	4	3	2	1	0
保留	TRIGGER	AL_EVENT_INT	ECAT_EVENT_INT	DIR[1:0]		OPERA_MODE[1:0]	
	rw(ECAT) r(PDI)	rw(ECAT) r(PDI)	rw(ECAT) r(PDI)	rw(ECAT) r(PDI)		rw(ECAT) r(PDI)	

位/位域	名称	描述
7	保留	必须保持复位值。
6	TRIGGER	看门狗触发器启用： 0: 禁用 1: 启用
5	AL_EVENT_INT	中断在AL事件请求寄存器中： 0: 禁用 1: 启用
4	ECAT_EVENT_INT	中断在以ECAT事件请求寄存器中： 0: 禁用 1: 启用
3:2	DIR[1:0]	方向： 00: 读：ECAT读访问，PDI写访问。 01: 写：ECAT写访问，PDI读访问。 10: 保留 11: 保留
1:0	OPERA_MODE[1:0]	操作模式： 00: 缓冲（3缓冲模式） 01: 保留 10: 邮箱（单缓冲模式） 11: 保留

SyncManager X 状态寄存器（SMX_STAT）（X = 0…F）

地址偏移量：0x0805 + X * 8
复位值：0x30

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

7	6	5	4	3	2	1	0

WRITE_BUFFER	READ_BUFFER	BUFFER_STATUS[1:0]	MAILBOX_STATUS	保留	READ_INT	WRITE_INT
r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)		r(ECAT) r(PDI)	r(ECAT) r(PDI)

位/位域	名称	描述
7	WRITE_BUFFER	正在使用的写缓冲区（已打开）
6	READ_BUFFER	正在使用的读缓冲区（已打开）
5:4	BUFFER_STATUS[1:0]	缓冲模式：缓冲状态（最后写入的缓冲区）： 00: 第一个缓冲区 01: 第二个缓冲区 10: 第三个缓冲区 11: （没有缓冲区被写入） 邮箱模式：保留
3	MAILBOX_STATUS	邮箱模式：邮箱状态： 0: 邮箱为空 1: 邮箱已满 缓冲模式：保留
2	保留	必须保持复位值。
1	READ_INT	中断读取： 1: 在缓冲区完全且成功读取后中断 0: 在缓冲区写入第一个字节后清除中断 **注意：**如果已在SM控制寄存器中启用，此中断将被信号到写入端。
0	WRITE_INT	中断写入： 1: 在缓冲区完全且成功写入后中断 0: 在缓冲区读取第一个字节后清除中断 **注意：**如果已在SM控制寄存器中启用，此中断将被信号到读取端。

SyncManager X 激活寄存器（SMX_ACTIVE）（X = 0…F）

地址偏移：0x0806 + X * 8
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

7	6	5	4	3	2	1	0
LATCH_PDI	LATCH_ECAT	保留				REQUEST	SM_EN
rw(ECAT) r(PDI)	rw(ECAT) r(PDI)					rw(ECAT) r(PDI)	r(ECAT) rw(PDI)

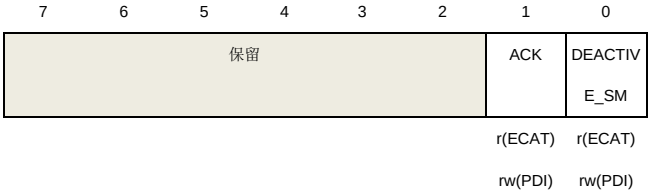
位/位域	名称	描述
7	LATCH_PDI	Latch事件PDI：

		0: 否 1: 当PDI发出缓冲区交换或当PDI访问缓冲区起始地址时生成Latch事件
6	LATCH_ECAT	Latch事件ECAT： 0: 否 1: 当以太网CAT主控器发出缓冲区交换时生成Latch事件
5:2	保留	必须保持复位值。
1	REQUEST	重复请求： 重复请求的切换意味着需要进行邮箱重试（主要与ECAT读取邮箱一起使用）
0	SM_EN	同步管理器启用/禁用： 0: 禁用：在没有同步管理器控制的情况下访问内存 1: 启用：同步管理器处于活动状态，并控制配置中设置的内存区域 **注意**：PDI寄存器功能通过写命令确认被禁用：从PDI读取此寄存器将清除所有已更改激活状态的同步管理器的AL事件请求0x0220[4]。从PDI写入此寄存器是不可能的。

PDI 控制寄存器（SMX_PDI_CTL）(X = 0…F)

地址偏移：0x0807 + X * 8
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



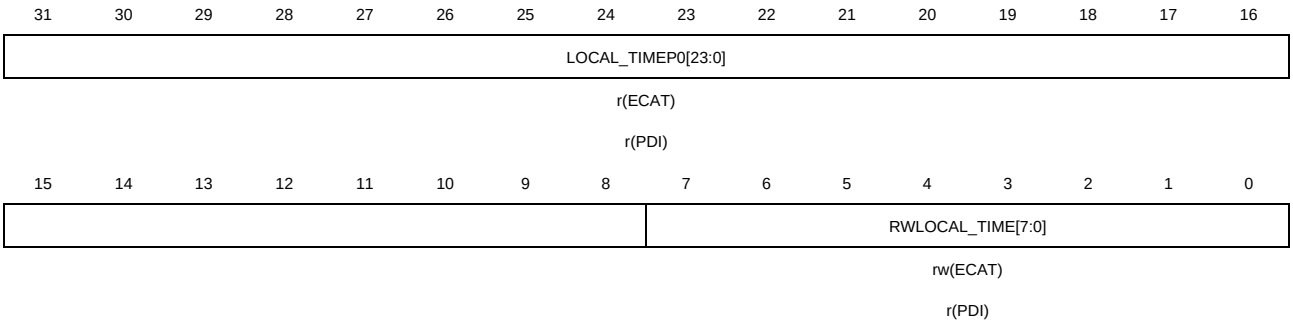
位/位域	名称	描述
7:2	保留	必须保持复位值。
1	ACK	重复确认： 如果此值设置为与重复请求所设置的值相同，PDI将确认执行之前的重复请求。
0	DEACTIVE_SM	停用同步管理器： 读取： 0: 正常操作，同步管理器已激活。 1: 同步管理器已停用并重置。同步管理器锁定对内存区域的访问。 写入： 0: 激活同步管理器 1: 请求停用同步管理器 **注意**：写入1将延迟直到当前正在处理的帧结束。

ESC 接收时间端口 0 寄存器（ESC_RCV_TIMEP0）

地址偏移：0x0900

复位值：未定义

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



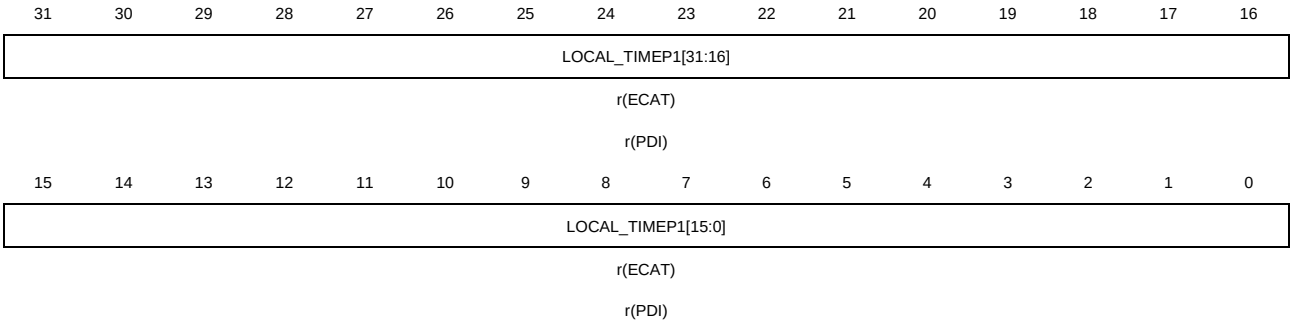
位/位域	名称	描述
31:8	LOCAL_TIMEP0[23:0]	包含对寄存器0x0900的写访问的最后一个接收帧开始时的本地时间
7:0	RWLOCAL_TIME[7:0]	写入：使用BWR或FPWR对寄存器0x0900的写访问会在每个端口的接收帧开始时（前导码的第一个比特）锁定本地时间。 读取：包含对此寄存器的写访问的最后一个接收帧开始时的本地时间。 **注意：**FPWR需要像任何FPWR命令一样进行地址匹配才能访问此寄存器。所有具有地址匹配的写命令将增加工作计数器（例如，APWR），但它们不会触发接收时间锁定。

注意：在写入此寄存器的同一帧中无法读取时间戳。

ESC 接收时间端口 1 寄存器（ESC_RCV_TIMEP1）

地址偏移：0x0904
复位值：未定义

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



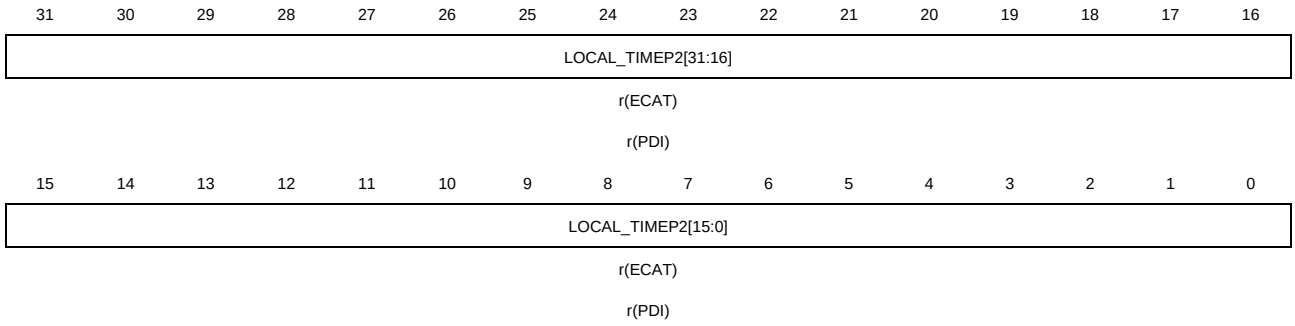
位/位域	名称	描述
31:0	LOCAL_TIMEP1[31:0]	在端口1接收到的包含对寄存器0x0900的BWR或FPWR的帧的开始（前导码的第一个比特）的本地时间。

ESC 接收时间端口 2 寄存器（ESC_RCV_TIMEP2）

地址偏移：0x0908

复位值：未定义

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

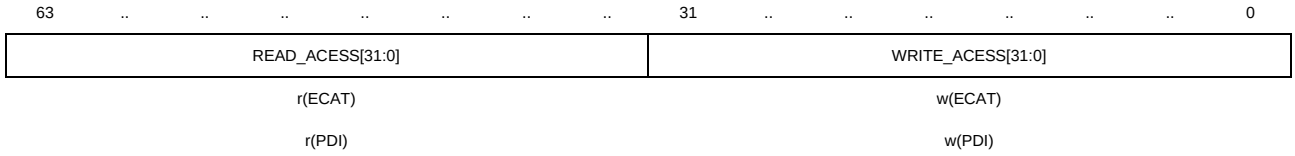


位/位域	名称	描述
31:0	LOCAL_TIMEP2[31:0]	在端口2接收到的包含对寄存器0x0900的BWR或FPWR的帧的开始（前导码的第一个比特）的本地时间。

ESC 系统时间寄存器（ESC_SYS_TIME）

地址偏移：0x0910
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



位/位域	名称	描述
63:0	READ_ACESS[63:0]	ECAT读取访问： 当帧通过参考时钟时的系统时间的本地副本（即，包括系统时间延迟）。时间在帧的开始时锁定（以太网SOF分隔符） PDI读取访问： 系统时间的本地副本。在读取第一个字节（0x0910）时锁定时间。
31:0	WRITE_ACESS[31:0]	写入访问： 写入的值将与系统时间的本地副本进行比较。结果是时间控制循环的输入。 **注意：**如果至少写入了第一个字节（0x0910），写入的值将在帧的末尾与锁定的（SOF）系统时间的本地副本进行比较。

注意：此寄存器的写访问权限取决于ESC配置（系统时间PDI控制关闭=ECAT/开启=PDI；ECAT控制是通用的）。

ESC 接收时间 ECAT 处理单元寄存器（ESC_RCVT_ECAT_PU）

地址偏移：0x0918
复位值：未定义

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

63	..	..	..	..	..	..	..	31	..	..	..	..	..	..	0
RECVE_TIME[63:0]															
r(ECAT)															
r(PDI)															

位/位域	名称	描述
63:0	RECVE_TIME[63:0]	在ECAT处理单元接收到的包含对寄存器0x0900的写访问的帧的开始（前导码的第一个比特）的本地时间 **注意：**如果端口0是开放的，此寄存器以64位值的形式反映端口0的接收时间。任何有效的ECAT写访问到寄存器0x0900都会触发锁定，而不仅仅是像寄存器0x0900那样的BWR/FPWR命令。

ESC 系统时间偏移寄存器（ESC_SYS_TIME_OFFSET）

地址偏移：0x0920
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

63	..	..	..	..	..	..	..	31	..	..	..	..	..	..	0
OFFSET_TIME[63:0]															
rw(ECAT)															
rw(PDI)															

位/位域	名称	描述
63:0	OFFSET_TIME[63:0]	本地时间与系统时间之间的差异。偏移量被加到本地时间上。

ESC 系统时间延迟寄存器（ESC_SYS_TIME_DELAY）

地址偏移：0x0928
复位值：0x0

该寄存器可以按字节(8位)、半字(16位)和字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DELAY_TIME[31:0]															
rw(ECAT)															
rw(PDI)															

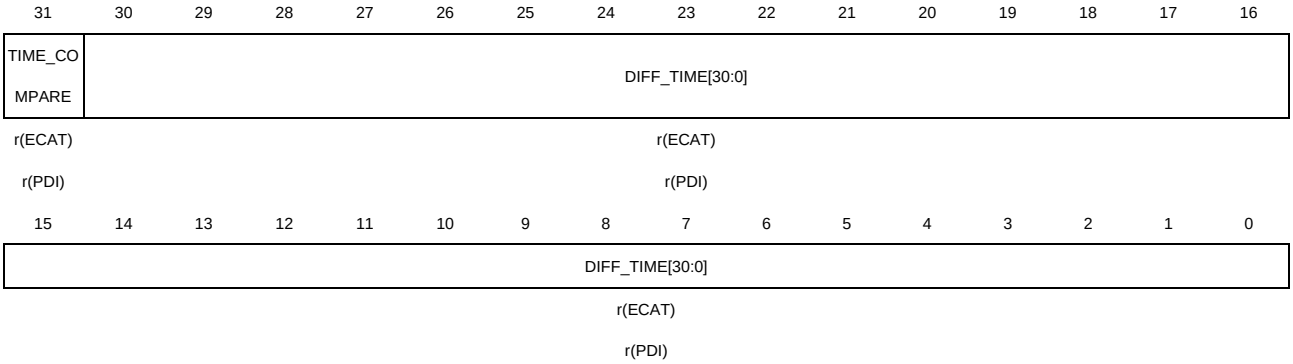
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DELAY_TIME[31:0]															
rw(ECAT)															
rw(PDI)															

位/位域	名称	描述
31:0	DELAY_TIME[31:0]	参考时钟与ESC之间的延迟

ESC 系统时间差值寄存器（ESC_SYS_TIME_DIFF）

地址偏移：0x092C
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



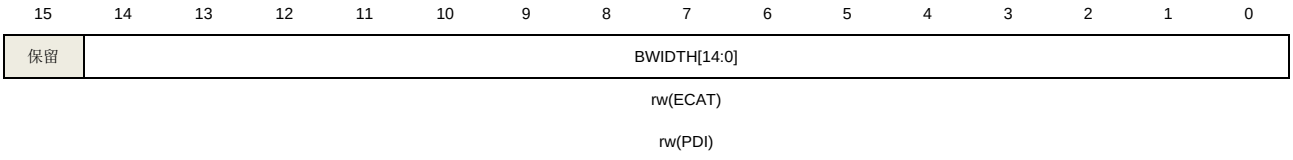
位/位域	名称	描述
31	TIME_COMPARE	0: 本地系统时间副本小于接收到的系统时间 1: 本地系统时间副本大于或等于接收到的系统时间
30:0	DIFF_TIME[30:0]	系统时间的本地副本与接收到的系统时间值之间的平均差异 差异 = 接收到的系统时间 - 系统时间的本地副本

注意：当读取位[7:0]时，位[31:8]会在ECAT/PDI内部被锁定（独立地），这保证了读取到的值是一致的。

ESC 速度计数器起始值寄存器（ESC_SPD_CNT_START）

地址偏移：0x0930
复位值：0x1000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



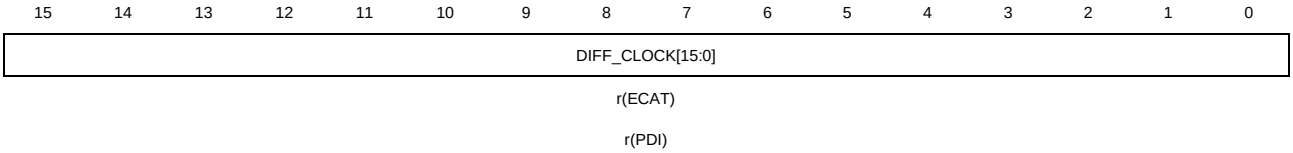
位/位域	名称	描述
15	保留	必须保持复位值。
14:0	BWIDTH[14:0]	调整系统时间本地副本的带宽（较大值 -> 较小带宽和更平滑的调整） 写访问会重置系统时间差异（0x092C:0x092F）和速度计数器差异（0x0932:0x0933）。有效值：0x0080至0x3FFF

注意：此寄存器的写访问权限取决于ESC配置（系统时间PDI控制关闭=ECAT / 开启=PDI；ECAT控制是通用的）。

ESC 速度计数器差值寄存器（ESC_SPD_CNT_DIFF）

地址偏移：0x0932
复位值：0x0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



位/位域	名称	描述
15:0	DIFF_CLOCK[15:0]	本地时钟周期与参考时钟周期之间的偏差表示（表示方式：二进制补码） 范围：±（速度计数器起始值 - 0x7F）

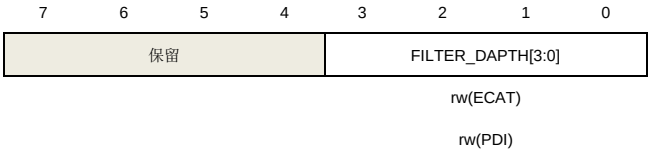
注意：系统时间差稳定在低值后，时钟偏差可以按以下方式计算：

偏差 = 速度计数器差值 / 5(速度计数器起始 + 速度计数器差值 + 2)(速度计数器起始 - 速度计数器差值 + 2)

ESC 系统时间差值滤波器深度寄存器（ESC_SYS_TIME_DIFF_FD）

地址偏移：0x0934
复位值：0x04

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



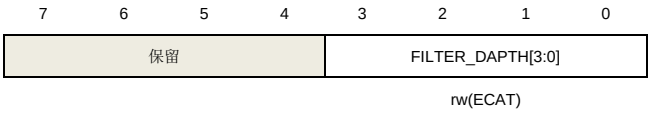
位/位域	名称	描述
7:4	保留	必须保持复位值。
3:0	FILTER_DAPTH[3:0]	用于平均接收到的系统时间偏差的滤波深度

注意：此寄存器的写访问权限取决于ESC配置（系统时间PDI控制关闭=ECAT/开启=PDI；ECAT控制是通用的）。

ESC 速度计数器滤波器深度寄存器（ESC_SPD_CNT_FD）

地址偏移：0x0935
复位值：0x12

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



rw(PDI)

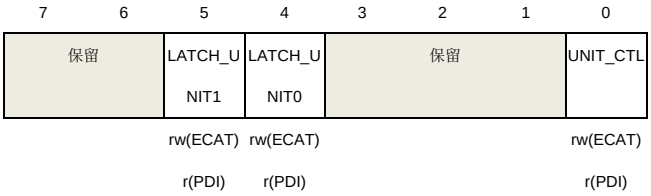
位/位域	名称	描述
7:4	保留	必须保持复位值。
3:0	FILTER_DAPTH[3:0]	用于平均时钟周期偏差的滤波深度。

注意：写入此寄存器的权限取决于ESC配置（系统时间PDI控制关闭=ECAT/开启=PDI；ECAT控制是通用的）。

ESC 循环单元控制寄存器（ESC_CYC_UNIT_CTL）

地址偏移：0x0980
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



位/位域	名称	描述
7:6	保留	必须保持复位值。
5	LATCH_UNIT1	锁存单元1： 0：ECAT控制 1：PDI控制 **注意：**根据此设置，锁存中断被路由到ECAT /PDI
4	LATCH_UNIT0	锁存单元0： 0：ECAT控制 1：PDI控制 注意：根据此设置，锁存中断被路由到ECAT / PDI。如果系统时间由PDI控制，则始终为1（PDI控制）。
3:1	保留	必须保持复位值。
0	UNIT_CTL	循环单元和SYNC0输出单元控制： 0：ECAT控制 1：PDI控制

ESC 激活寄存器（ESC_REG_ACT）

地址偏移：0x0981
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

7	6	5	4	3	2	1	0
SYNCSIGNAL	CONFIG	START_TIME	EXTEN	AUTO_ACTIVE	SYNC1	SYNC0	SYNC_OUT
rw(ECAT) rw(PDI)	rw(ECAT) rw(PDI)	rw(ECAT) rw(PDI)	rw(ECAT) rw(PDI)	rw(ECAT) rw(PDI)	rw(ECAT) rw(PDI)	rw(ECAT) rw(PDI)	rw(ECAT) rw(PDI)

位/位域	名称	描述
7	SYNCSIGNAL	同步信号调试脉冲（Vasily位）： 0：未激活 1：根据0x0981[2:1]立即仅在SYNC0-1上生成一个脉冲，用于调试 此位是自清除的，总是读取为0。 所有脉冲同时生成，循环时间被忽略。使用配置的脉冲长度。
6	CONFIG	近期配置（大约）： 0：½ DC宽度的一半特征(2^{31} ns 或2^{63} ns) 1：大约2.1秒(2^{31} ns)
5	START_TIME	开始时间合理性检查： 0：禁用。如果达到开始时间，则生成同步信号。 1：如果开始时间不在近期未来（见0x0981[6]）内，则立即生成同步信号。
4	EXTEN	开始时间循环操作的扩展（0x0990:0x0993）： 0：不扩展 1：将32位写入的开始时间扩展到64位
3	AUTO_ACTIVE	通过写入开始时间循环操作自动激活（0x0990:0x0997）： 0：禁用 1：启用自动激活。在写入开始时间后，0x0981[0]会自动设置。
2	SYNC1	SYNC1生成： 0：停用 1：生成SYNC1脉冲
1	SYNC0	SYNC0 生成： 0：停用 1：生成SYNC0 脉冲
0	SYNC_OUT	同步输出单元激活： 0：停用 1：激活

注意：写入此寄存器取决于0x0980[0]的设置。

ESC 同步信号寄存器的脉冲长度（ESC_PULSE_SM）

地址偏移：0x0982
复位值：取决于配置

该寄存器可以按字节（8位）、半字（16位）和字（32位）进行访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLEN_SM[15:0]															
r(ECAT)															
r(PDI)															

位/位域	名称	描述
15:0	PLEN_SM[15:0]	同步信号的脉冲长度（以10ns为单位） 0：确认模式：同步信号将在读取SYNC[1:0]状态寄存器时清除

ESC 激活状态寄存器（ESC_ACT_STAT）

地址偏移量：0x0984
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

7	6	5	4	3	2	1	0
保留					START_TIME	SYNC1_ACTIVE	SYNC0_ACTIVE
					r(ECAT)	r(ECAT)	r(ECAT)
					r(PDI)	r(PDI)	r(PDI)

位/位域	名称	描述
7:3	保留	必须保持复位值。
2	START_TIME	同步输出单元激活时的循环操作开始时间（0x0990:0x0997）合理性检查结果： 0：开始时间在近期内 1：开始时间不在近期内（0x0981[6]）
1	SYNC1_ACTIVE	SYNC1激活状态： 0：第一个SYNC1脉冲未待处理 1：第一个SYNC1脉冲待处理
0	SYNC0_ACTIVE	SYNC0激活状态： 0：第一个SYNC0脉冲未待处理 1：第一个SYNC0脉冲待处理

ESC SYNC0 状态寄存器（ESC_SYNC0_STAT）

地址偏移：0x098E
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

7	6	5	4	3	2	1	0
保留							SYNC0_STATE
							r(ECAT)

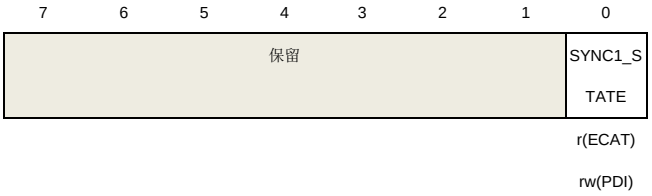
rw(PDI)

位/位域	名称	描述
7:1	保留	必须保持复位值。
0	SYNC0_STATE	确认模式下的SYNC0状态。 在确认模式下，通过从PDI读取此寄存器来清除SYNC0，仅在确认模式下使用 **注意：**PDI寄存器功能通过写命令确认被禁用：从PDI读取此寄存器将清除AL事件请求0x0220[2]。从PDI写入此寄存器是不可能的。 PDI寄存器功能通过写命令确认被启用：从PDI写入此寄存器将清除AL事件请求0x0220[2]。从PDI写入此寄存器是可能的；写入的值被忽略（写入0）。

ESC SYNC1 状态寄存器（ESC_SYNC1_STAT）

地址偏移：0x098F
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

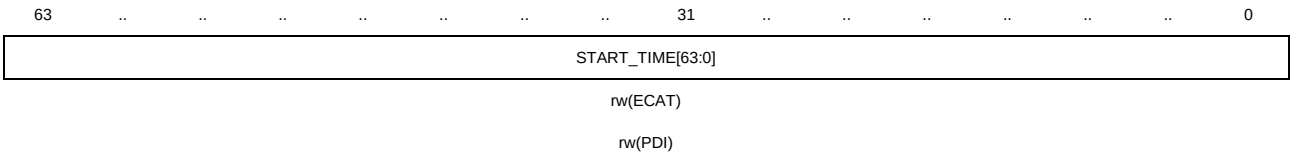


位/位域	名称	描述
7:1	保留	必须保持复位值。
0	SYNC1_STATE	确认模式下的SYNC1状态。 在确认模式下，通过从PDI读取此寄存器来清除SYNC1，仅在确认模式下使用。 **注意：**PDI寄存器功能通过写命令确认被禁用：从PDI读取此寄存器将清除AL事件请求0x0220[3]。从PDI写入此寄存器是不可能的。 PDI寄存器功能通过写命令确认被启用：从PDI写入此寄存器将清除AL事件请求0x0220[3]。从PDI写入此寄存器是可能的；写入的值被忽略（写入0）。

ESC 起始时间循环操作寄存器（ESC_START_TIME_CO）

地址偏移：0x0990
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



位/位域	名称	描述

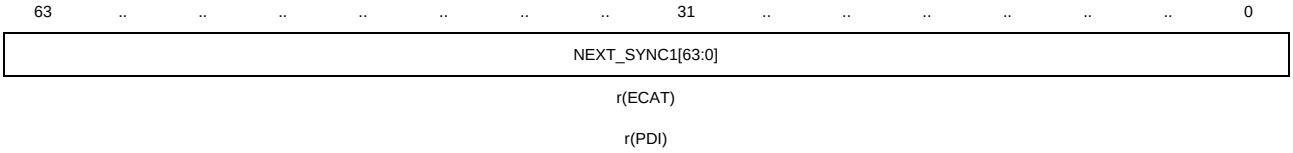
63:0	START_TIME[63:0]	写入：循环操作的开始时间（系统时间）以ns为单位 读取：下一个SYNC0脉冲的系统时间以ns为单位

注意：当读取位[7:0]时，位[63:8]会在ECAT / PDI内部独立地被锁存，这保证了读取到一致的值。

ESC 下一个 SYNC1 脉冲寄存器（ESC_NEXT_SYNC1_PLUSE）

地址偏移：0x0998
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）进行访问。



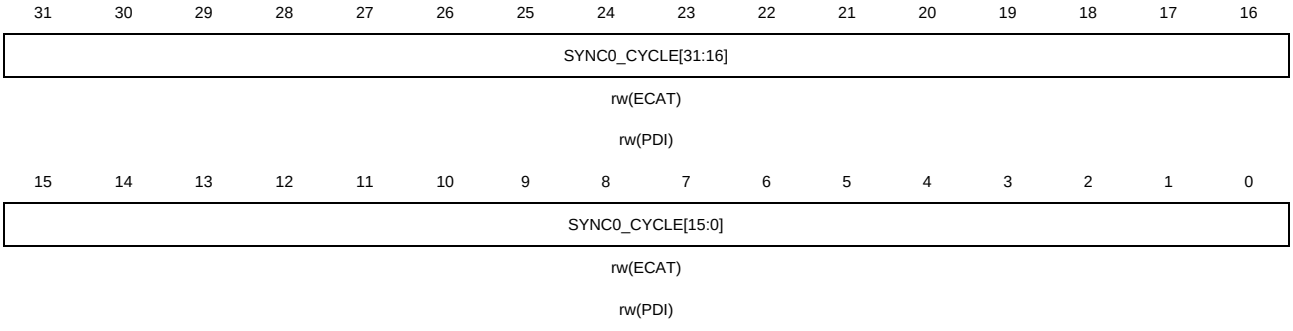
位/位域	名称	描述
63:0	NEXT_SYNC1[63:0]	下一个SYNC1脉冲的系统时间以ns为单位。

注意：当读取位[7:0]时，位[63:8]会在ECAT/PDI内部独立地锁存，这保证了读取到的值是一致的。

ESC SYNC0 周期时间寄存器（ESC_SYNC0_CYC_TIME）

地址偏移：0x09A0
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



位/位域	名称	描述
31:0	SYNC0_CYCLE[31:0]	两个连续SYNC0脉冲之间的时间以ns为单位。 0：单次触发模式，只产生一个SYNC0脉冲。 **注意：**写入此寄存器取决于0x0980[0]的设置。循环操作的最小值：60 [ns]。

ESC SYNC1 周期时间寄存器（ESC_SYNC1_CYC_TIME）

地址偏移：0x09A4
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SYNC1_CYCLE[31:16]															
rw(ECAT) rw(PDI)															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNC1_CYCLE [15:0]															
rw(ECAT) rw(PDI)															

位/位域	名称	描述
31:0	SYNC1_CYCLE[31:0]	SYNC0脉冲和SYNC1脉冲之间的时间以ns为单位。 **注意：**写入此寄存器取决于0x0980[0]的设置。

ESC LATCH0 控制寄存器（ESC_LATCH0_CTL）

地址偏移：0x09A8
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

7	6	5	4	3	2	1	0
保留						LATCH0_NEGATIVE	LATCH0_POSITIVE
						rw(ECAT) rw(PDI)	rw(ECAT) rw(PDI)

位/位域	名称	描述
7:2	保留	必须保持复位值。
1	LATCH0_NEGATIVE	Latch0下降沿： 0：连续Latch激活 1：单事件（只有第一个事件激活）
0	LATCH0_POSITIVE	Latch0上升沿： 0：连续Latch激活 1：单事件（只有第一个事件激活）

写入访问取决于0x0980[4]的设置。

ESC LATCH1 控制寄存器（ESC_LATCH1_CTL）

地址偏移量：0x09A9
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

7	6	5	4	3	2	1	0

保留	LATCH1_NEGATIVE	LATCH1_POSITIVE
	rw(ECAT) rw(PDI)	rw(ECAT) rw(PDI)

位/位域	名称	描述
7:2	保留	必须保持复位值。
1	LATCH1_NEGATIVE	Latch1下降沿： 0：连续Latch激活 1：单事件（只有第一个事件激活）
0	LATCH1_POSITIVE	Latch1上升沿： 0：连续Latch激活 1：单事件（只有第一个事件激活）

注意：写访问权限取决于0x0980[5]的设置。

ESC LATCH0 状态寄存器（ESC_LATCH0_STAT）

地址偏移：0x09AE
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）进行访问。

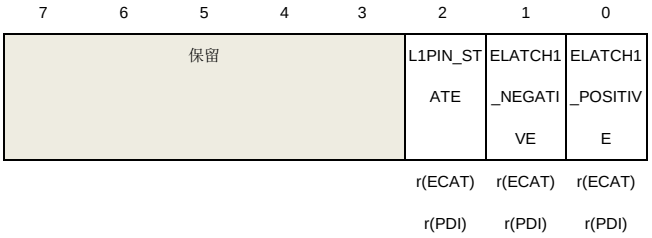
7	6	5	4	3	2	1	0
保留					L0PIN_STATE	ELATCH0_NEGATIVE	ELATCH0_POSITIVE
					r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)

位/位域	名称	描述
7:3	保留	必须保持复位值。
2	L0PIN_STATE	Latch0引脚状态
1	ELATCH0_NEGATIVE	事件 Latch0下降沿。 0：未检测到下降沿或连续模式 1：仅在单事件模式下检测到下降沿。 通过读取 Latch0 时间下降沿清除标志。
0	ELATCH0_POSITIVE	事件 Latch0上升沿。 0：未检测到上升沿或连续模式 1：仅在单事件模式下检测到上升沿。 通过读取 Latch0 时间上升沿清除标志

ESC LATCH1 状态寄存器（ESC_LATCH1_STAT）

地址偏移：0x09AF
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

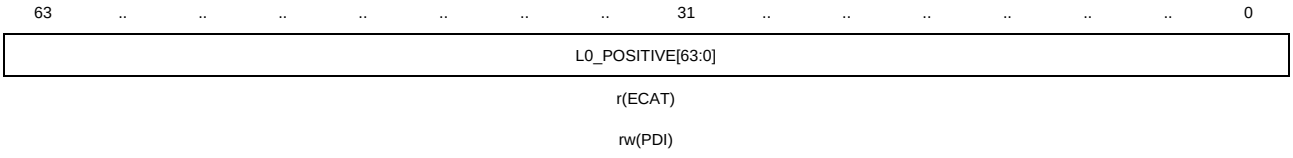


位/位域	名称	描述
7:3	保留	必须保持复位值。
2	L1PIN_STATE	Latch1引脚状态
1	ELATCH1_NEGATIVE	事件 Latch1下降沿。 0：未检测到下降沿或连续模式 1：仅在单事件模式下检测到下降沿。 通过读取 Latch1 时间下降沿清除标志。
0	ELATCH1_POSITIVE	事件 Latch1上升沿。 0：未检测到上升沿或连续模式 1：仅在单事件模式下检测到上升沿。 通过读取 Latch1 时间上升沿清除标志

ESC LATCH0 时间上升沿寄存器（ESC_LATCH0_TIME_PE）

地址偏移：0x09B0
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



位/位域	名称	描述
63:0	L0_POSITIVE[63:0]	在Latch0信号的上升沿的系统时间。 **注意：**PDI寄存器功能通过写命令确认被禁用：如果0x0980[4]=1，从PDI读取此寄存器将清除Latch0状态0x09AE[0]。从PDI写入此寄存器是不可能的。 PDI寄存器功能通过写命令确认被启用：如果0x0980[4]=1，从PDI写入此寄存器将清除Latch0状态0x09AE[0]。从PDI写入此寄存器是可能的；写入的值被忽略（写入0）。

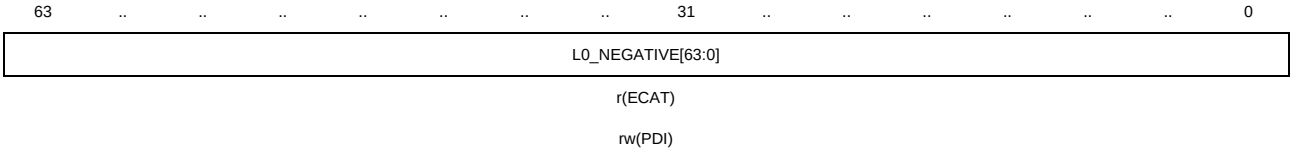
注意：当读取位[7:0]时，位[63:8]会在ECAT/PDI内部独立地被锁存，这保证了读取到的值是一

致的。从ECAT读取此寄存器时，如果0x0980[4]=0，则会清除Latch0状态0x09AE[0]。从ECAT写入此寄存器是不可能的。

ESC LATCH0 时间下降沿寄存器（ESC_LATCH0_TIME_NE）

地址偏移：0x09B8
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



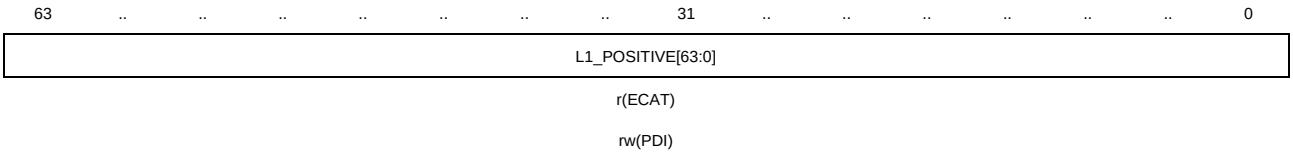
位/位域	名称	描述
63:0	L0_NEGATIVE[63:0]	在Latch0信号的下降沿的系统时间。 **注意：**PDI寄存器功能通过写命令确认被禁用：如果0x0980[4]=1，从PDI读取此寄存器将清除Latch0状态0x09AE[1]。从PDI写入此寄存器是不可能的。 PDI寄存器功能通过写命令确认被启用：如果0x0980[4]=1，从PDI写入此寄存器将清除Latch0状态0x09AE[1]。从PDI写入此寄存器是可能的；写入的值被忽略（写入0）。

注意：当读出位[7:0]时，位[63:8]会在内部被锁定（ECAT/PDI独立），这保证了读取到一致的值。如果0x0980[4]=0，从ECAT读取此寄存器会清除Latch0状态0x09AE[1]。从ECAT写入此寄存器是不可能的。

ESC LATCH1 时间上升沿寄存器（ESC_LATCH1_TIME_PE）

地址偏移：0x09C0
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



位/位域	名称	描述
63:0	L1_POSITIVE[63:0]	在Latch1信号的上升沿的系统时间。 **注意：**PDI寄存器功能通过写命令确认被禁用：如果0x0980[5]=1，从PDI读取此寄存器将清除Latch0状态0x09AF[0]。从PDI写入此寄存器是不可能的。 PDI寄存器功能通过写命令确认被启用：如果0x0980[5]=1，从PDI写入此寄存器将清除Latch0状态0x09AF[0]。从PDI写入此寄存器是可能的；写入的值被忽略（写入0）。

注意：当读取位[7:0]时，位[63:8]会在ECAT/PDI内部独立地被锁存，这保证了读取到的值是一致的。如果0x0980[5]=0，从ECAT读取此寄存器会清除Latch0状态0x09AF[0]。从ECAT写入此

寄存器是不可能的。

ESC LATCH1 时间下降沿寄存器（ESC_LATCH1_TIME1_NE）

地址偏移：0x09C8
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）进行访问。

63 .. .. 31 0
L1_NEGATIVE[63:0]
r(ECAT)
rw(PDI)

位/位域	名称	描述
63:0	L1_NEGATIVE[63:0]	在Latch1信号的下降沿的系统时间。 **注意：**PDI寄存器功能通过写命令确认被禁用：如果0x0980[5]=1，从PDI读取此寄存器将清除Latch0状态0x09AF[1]。从PDI写入此寄存器是不可能的。 PDI寄存器功能通过写命令确认被启用：如果0x0980[5]=1，从PDI写入此寄存器将清除Latch0状态0x09AF[1]。从PDI写入此寄存器是可能的；写入的值被忽略（写入0）。

注意：当读取位[7:0]时，位[63:8]会在ECAT/PDI内部独立地被锁存，这保证了读取到的值是一致的。如果0x0980[5]=0，从ECAT读取此寄存器将清除Latch0状态0x09AF[1]。从ECAT写入此寄存器是不可能的。

ESC 缓冲区变化事件时间寄存器（ESC_BUF_CET）

地址偏移：0x09F0
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

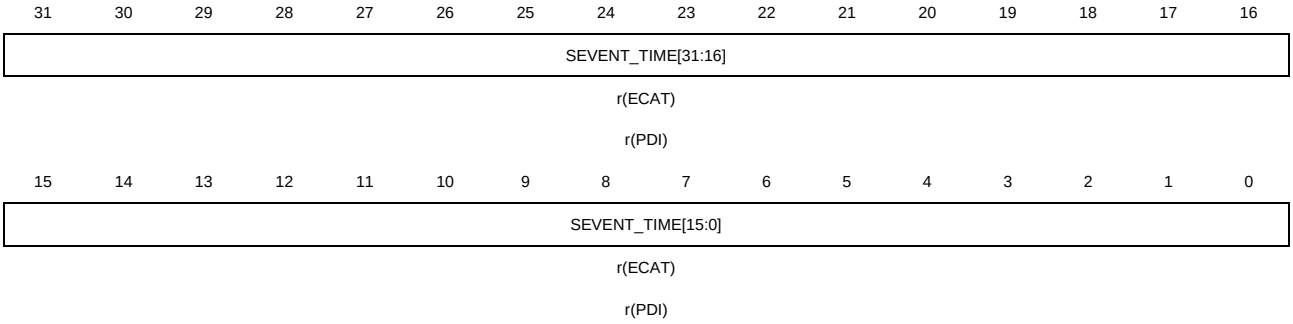
位/位域	名称	描述
31:0	EVENT_TIME[31:0]	帧开始时的本地时间，该帧至少导致一个同步管理器（SyncManager）有效一个（ECAT）事件。

注意：当读取位[7:0]时，位[31:8]会在内部被ECAT/PDI独立地锁存，这保证了读取到的值是一致的。

ESC PDI 缓冲区起始事件时间寄存器（ESC_PDI_BUF_SET）

地址偏移：0x09F8
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



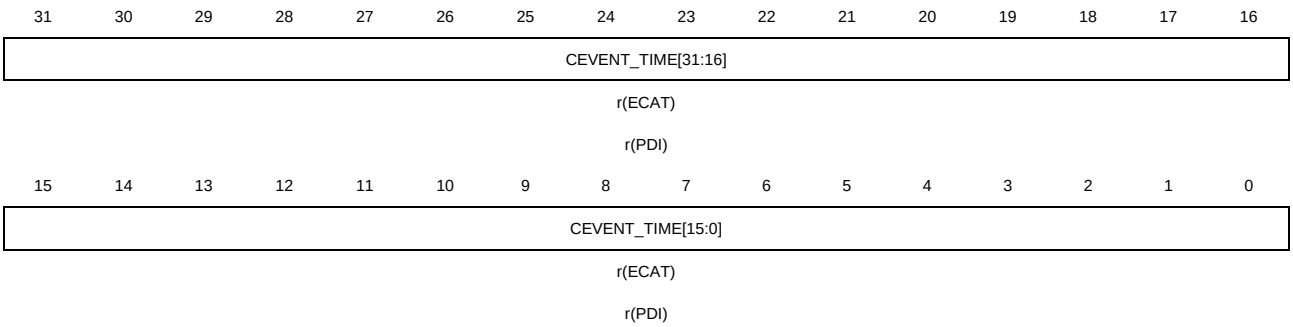
位/位域	名称	描述
31:0	PDI_SEVENT_TIME[31:0]	当至少一个同步管理器（SyncManager）有效PDI缓冲区开始事件时的本地时间。

注意：当读取位[7:0]时，位[31:8]会在内部被ECAT/PDI独立地锁定，这保证了读取到一致的值。

ESC PDI 缓冲区变化事件时间寄存器（ESC_PDI_BUF_CET）

地址偏移：0x09FC
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。



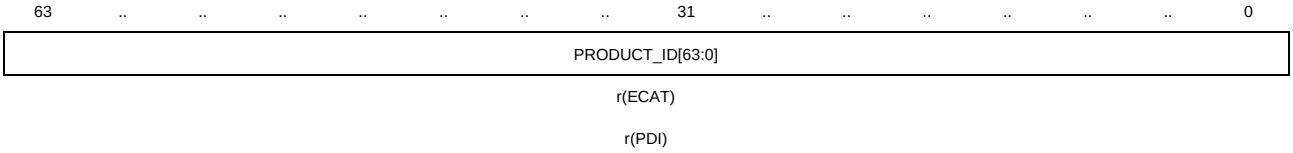
位/位域	名称	描述
31:0	PDI_CEVENT_TIME[31:0]	当至少一个SyncManager有效PDI缓冲区更改事件时的本地时间

注意：当读取位[7:0]时，位[31:8]会在ECAT/PDI内部独立地被锁存，这保证了读取到一致的值。

ESC 产品 ID 寄存器（ESC_PRODUCT_ID）

地址偏移：0x0E00
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）进行访问。

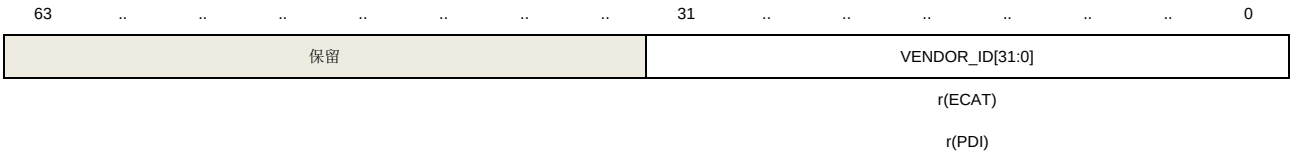


位/位域	名称	描述
63:0	PRODUCT_ID[63:0]	产品ID

ESC 供应商 ID 寄存器（ESC_VENDOR_ID）

地址偏移：0x0E08
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）进行访问。

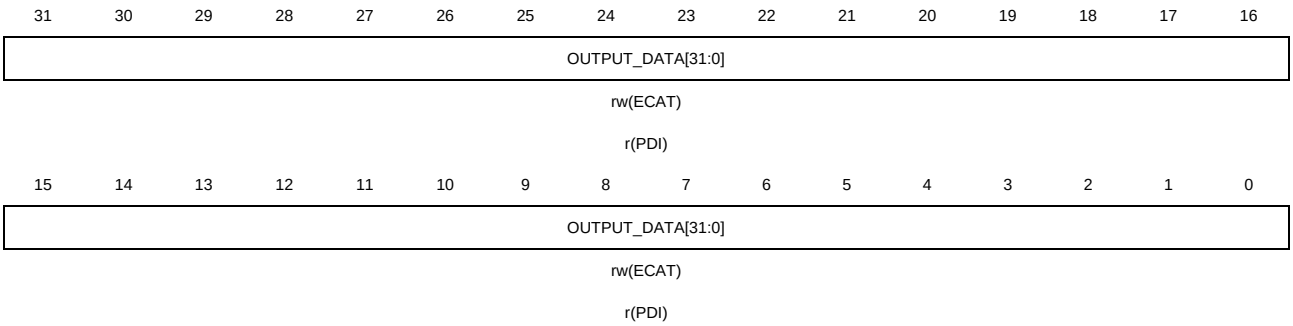


位/位域	名称	描述
63:32	保留	必须保持复位值。
31:0	VENDOR_ID[31:0]	供应商ID: [23:0] 公司 [31:24] 部门 **注意：**测试供应商ID的[31:28]=0xE

ESC 数字 I/O 输出数据寄存器（ESC_DIG_DATA）

地址偏移：0x0F00
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）进行访问。



位/位域	名称	描述
31:0	OUTPUT_DATA[31:0]	输出数据

注意：寄存器大小取决于PDI设置和/或设备配置。此寄存器支持位写入（使用逻辑寻址）。

ESC 通用输出寄存器（ESC_GP_OUTPUT）

地址偏移：0x0F10
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）进行访问。

63	31 0
GP_OUTPUT[63:0]	
rw(ECAT)	
rw(PDI)	

位/位域	名称	描述
63:0	GP_OUTPUT [63:0]	通用输出数据

注意：寄存器大小取决于PDI设置和/或设备配置。

ESC 通用输入寄存器（ESC_GP_INPUT）

地址偏移：0x0F18
复位值：0x0

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

63	31 0
GP_INPUT[63:0]	
r(ECAT)	
r(PDI)	

位/位域	名称	描述
63:0	GP_INPUT[63:0]	通用输入数据

注意：寄存器大小取决于PDI设置和/或设备配置。

ESC 用户 RAM 寄存器（ESC_USER_RAM）

地址偏移：0x0F80

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

1023 0
USER_RAM [1023:0]
rw(ECAT)
rw(PDI)

位/位域	名称	描述
----	USER_RAM [1023:0]	用户RAM（128字节）

ESC PDI 数字 I/O 输入数据寄存器（ESC_PDI_DATA）

地址偏移：0x1000
复位值：未定义

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

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

位/位域	名称	描述
31:0	INPUT_DATA[31:0]	输入数据

注意：只有在正确加载了EEPROM（寄存器0x0110[0] = 1）的情况下，才能访问进程数据RAM。

输入数据的大小取决于PDI设置和/或设备配置。如果配置了带有输入的数字I/O PDI，则数字I/O输入数据将写入这些地址的进程数据RAM中。

ESC 进程数据 RAM（ESC_PDRAM）

地址偏移：0x1000
复位值：未定义

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

8191	..	..	..	..	..	..	..	..	..	..	..	..	..	..	0
PDRAM[8191:0]															
rw(ECAT)															
rw(PDI)															

位/位域	名称	描述
----	PDRAM [8191:0]	过程数据RAM(8 KB)

注意：只有当EEPROM正确加载时（寄存器0x0110[0] = 1），才能访问处理数据RAM。

51. 附录

表 51-1. 寄存器功能位访问属性

功能位访问属性	描述
读/写(rw)	软件可以对这个位进行读写。
只读(r)	软件只能对这个位进行读。
只写(w)	软件只能对这个位进行写。读取该位将返回复位值。
读/写 1 清零(rc_w1)	软件可以读该位，对该位写入 1 可以清除这个位。写入 0 对位值没有影响。
读/写 0 清零(rc_w0)	软件可以读该位，对该位写入 0 可以清除这个位。写入 1 对位值没有影响。
翻转(t)	软件可以通过写 1 来翻转该位。写入 0 对位值没有效果。
只读/写 1 触发(rt_w1)	软件可以读该位，写入 1 触发事件，但对位值没有影响。

表 51-2. 术语

术语	描述
字	32 位长度数据
半字	16 位长度数据.
字节	8 位长度数据
IAP(应用内编程)	IAP 是在用户程序运行时对微控制器的闪存重新编程的能力。
ICP(在线编程)	ICP 是当设备安装在用户应用板上时，一个使用 JTAG 协议，SWD 协议或引导加载程序的微控制器的闪存编程能力。
选项字节	存储在闪存中的产品配置位
AHB	高级高性能总线
APB	高级外设总线
RAZ	读为 0
WI	写忽略
RAZ/WI	读为 0/写忽略

52. 版本历史

表 52-1. 版本历史

版本号	描述	日期
1.0	首次发布	2023 年 5 月 6 日
1.1	1. 更新 ***HPDF 结构框图***、***HPDF 时钟***和***阈值监视器***章节框图； 2. 更新***控制器局域网络（CAN）***章节简介描述； 3. 更新 ***I/O 补偿单元代码配置寄存器（SYSCFG_CPSCCCFG）***寄存器位域描述； 4. 更新***备用功能选择寄存器 0（GPIOx_AFSEL0，x=A...H，J，K）***和***备用功能选择寄存器 0（GPIOx_AFSEL0，x=A...H，J，K）***寄存器位域描述； 5. 更新***图 20 1. ADC 模块框图***； 6. 更新 USART 模块***接收 FIFO***章节描述； 7. 更新 GPIO 和 AFIO 模块***输入滤波***和***输入滤波寄存器（GPIOx_IFL，x=A…H，J，K）***章节描述； 8. 更新 CPDM 模块***配置寄存器（CPDM_CFG）***寄存器位域描述； 9. 更新 CMP 模块***状态寄存器（CMP_STAT）***和***中断标志位清除寄存器（CMP_IFC）***部分描述； 10. 更新 TRNG 模块***控制寄存器（TRNG_CTL）***寄存器位域描述； 11. 更新***电源管理单元（PMU）***章节内容 12. 更新 ***全局 USB 控制和状态寄存器（USBHS_GUSBCS）***寄存器； 13. 更新 ***SAI 寄存器***访问属性； 14. 更新 IPA 模块***中断标志清除寄存器（IPA_INTC）***寄存器。	2023 年 7 月 21 日
1.2	1. 更新 RTDEC 模块***图 46 3. CTR 下的 AES-128 解密流***； 2. 更新***三角函数加速器（TMU）***章节内容； 3. 更新***低功耗数字温度传感器（LPDTS）***章节内容； 4. 更新 ***RSPDIF 模块图 33-11. RSPDIF 状态机***； 5. 更新***控制器局域网络（CAN）***章节描述； 6. 更新***系统及存储器架构***章节描述； 7. 更新 RCU 模块***图 6-2. 时钟树***及部分描述； 8. 更新***时钟配置寄存器 1（RCU_CFG1）***复位值； 9. 更新 SDIO 模块***图 36-13. 8 位数据总线宽度***； 10. 更新***比较器（CMP）***章节描述； 11. 更新***通用同步异步收发器（USART）***章节描述； 12. 更新***端口输入状态寄存器(GPIOx_ISTAT，x=A…H，***	2024 年 1 月 4 日

版本号	描述	日期
	J，K）寄存器位域描述。	
1.3	1. 更新***系统及存储器架构***章节 1.8.21、1.8.24、1.8.27 寄存器的 TSCFG4[4:0]位域描述； 2. 更新***电源管理单元（PMU）***章节内容； 3. 更新***中断/事件控制器（EXTI）***章节表 8 2.中断向量表描述； 4. 更新***触发选择控制器（TRIGSEL）***章节内容 5. 更新***通用和备用输入/输出接口（GPIO 和 AFIO）***章节简介内容描述； 6. 更新***真随机数生成器（TRNG）***章节图 12 1. TRNG 模块框图； 7. ***三角函数加速器（TMU）***模块新增 TMU 精度内容描述； 8. 更新***表 18 2. DMAMUX 请求路由输入信号映射***； 9. 更新***表 20 7. 12 位 ADC 部分举例 N 和 M 的最大输出值（灰色部分表示截断）***； 10. 更新***看门狗定时器（WDGT）***模块 WWDGT 寄存器基地址； 11. 更新***定时器（TIMER）***模块表 24-9、24-13 和 24-17 中内容描述； 12. 更新***内部集成电路总线接口（I2C）***模块模拟滤波器延时时间； 13. 更新 ***TFT-LCD 接口（TLI）***模块 32.7.23 寄存器 FLL[13:0]位域描述； 14. 更新 ***S/PD 数字音频接口接收器（RSPDIF）***模块内容描述； 15. 更新***控制器局域网络（CAN）***模块 CAN 匹配过程的描述,修改 CAN_ERR1 寄存器的复位值； 16. 更新***实时解密（RTDEC）***模块内容描述； 17. 更新***硬件信号量（HWSEM）***模块 INTEN 寄存器复位值，INTC 寄存器访问属性； 18. 更新***通用串行总线高速接口（USBHS）***模块 49.7.1、49.7.2 章节描述	2024 年 5 月 10 日
1.3	1. 更新***系统及存储器架构***章节 1.8.21、1.8.24、1.8.27 寄存器的 TSCFG4[4:0]位域描述； 2. 更新***电源管理单元（PMU）***章节内容； 3. 更新***中断/事件控制器（EXTI）***章节表 8 2.中断向量表描述； 4. 更新***触发选择控制器（TRIGSEL）***章节内容 5. 更新***通用和备用输入/输出接口（GPIO 和 AFIO）***章节简介内容描述； 6. 更新***真随机数生成器（TRNG）***章节图 12 1. TRNG 模块框图；	2024 年 5 月 10 日

版本号	描述	日期
	7. ***三角函数加速器（TMU）***模块新增 TMU 精度内容描述； 8. 更新***表 18 2. DMAMUX 请求路由输入信号映射***； 9. 更新***表 20 7. 12 位 ADC 部分举例 N 和 M 的最大输出值（灰色部分表示截断）***； 10. 更新***看门狗定时器（WDGT）***模块 WWDGT 寄存器基地址； 11. 更新***定时器（TIMER）***模块表 24-9、24-13 和 24-17 中内容描述； 12. 更新***内部集成电路总线接口（I2C）***模块模拟滤波器延时时间； 13. 更新 ***TFT-LCD 接口（TLI）***模块 32.7.23 寄存器 FLL[13:0]位域描述； 14. 更新 ***S/PD 数字音频接口接收器（RSPDIF）***模块内容描述； 15. 更新***控制器局域网络（CAN）***模块 CAN 匹配过程的描述,修改 CAN_ERR1 寄存器的复位值； 16. 更新***实时解密（RTDEC）***模块内容描述； 17. 更新***硬件信号量（HWSEM）***模块 INTEN 寄存器复位值，INTC 寄存器访问属性； 18. 更新***通用串行总线高速接口（USBHS）***模块 49.7.1、49.7.2 章节描述。	
1.4	1. ***24.5.5*** 章节更新计数器自动重载寄存器（TIMERx_CAR）（TIMERx, x=5,6）为 32bit； 2. ***24.2.5*** 章节更新 DMA 配置寄存器（TIMERx_DMACFG）、DMA 发送缓冲区寄存器（TIMERx_DMATB）偏移地址为 0xE0、0xE4； 3. ***6.3.45*** 章节更新 USBHS0SW / USBHS1SW 改位 USBHS0SEL/USBHS1SEL； 4. ***24.3.5*** 章节 CH0IF 增加当通道 0 在输入模式下时，通过读 TIMERx_CH0CV 寄存器可以清零该位； 5. ***44.3.2*** 章节，更新 CMP 输出到引脚部分进行描述，增加连接到 TIMER 的注意事项，图 44-2 双边迟滞修改为单边迟滞； 6. ***27.5.14*** 章节 bit31 增加描述“当前”即“下一个将要发送或当前接收的数据； 7. ***6.2.1*** 章节时钟树新增注释； 8. ***SYSCFG*** 章节 SYSCFG_TIMERCISEL6 bit7~4 添加 0001: RSPDIF symbol_ck 描述； 9. ***23.4.4*** 章节更新带平滑校准的 RTC 中 SCPF 位描述； 10. ***表 18-3*** 新增"注意：EXTI_x(x=0…15)仅 EXTI 中断事件发生时会产生 DMA 请求；	2024 年 11 月 28 日

版本号	描述	日期
	11. ***RCU*** 章节和 ***EXTI*** 章节，更新当 systick 使用外部时钟源时，由 2 分频改为 8 分频； 12. ***HPDF*** 章节更新 HPDF_CHxCTL 寄存器 SMSD 位域描述； 13. 更新***图 18-8 中央对齐计数模式计数器时序图***； 14. ***42.5.6*** 章节更新 CAN_ERR1 寄存器复位值，调换 bit15 和 bit14 的功能； 15. ***17.3.5*** 章节***图17-7*** 中 BRTCF 改为 MBTCF,BRTCIE 改为 MBTCIE； 16. ***38.3.6*** 章节***表 38-6*** DLAT 改为同步； 17. ***Table 35-5*** 更新 YUV 和 YCbCr 模式期望系数； 18. 更新***表 44-1*** 最后一项引脚对应的 AF 值部分修改； 19. ***PMU*** 章节删除 DVSCFG 及其相关内容，删除供电模式 3,4,5； 20. ***43.3.1*** 章节增加 MDC 分频系数的描述； 21. ***21.4.18*** 章节更新 DAC 寄存器 bit 位读写描述； 22. ***表 20-5*** 和***表 20-6*** 更新时钟时间； 23. ***DBG*** 模块 JEDEC ROM 地址修改为 0xE00FD000； 24. ***10.4*** 章节寄存器：修改寄存器的读访问方式、删除 PB2 的描述。 25. 更新 ***SDIO***、***TMU*** 章节部分描述	
1.5	1. ***MDMA*** 模块***章节 17.4.4*** 中，将 MDMA_CHxSTAT1 寄存器 bit11 BZERR 位的描述更新为当块的大小或 BTLEN+1 不是源或目标数据大小的整数倍时，该位由硬件置 1。 2. ***MDMA*** 模块***章节 17.4.6*** 中，MDMA_CHxCFG 寄存器 24:18 BTLEN 位域的描述改为注意：BTLEN+1 必须是 DWIDTH 和 SWIDTH 的倍数。 3. ***PMU*** 模块中，体现了 BOR 并修改了 BOR 对 VDD 域的监测。 4. ***CAN*** 模块***章节 42.5.13*** 和章节 42.5.30 中，增加了对 CAN_BT 和 CAN_FDBT 寄存器中波特率分频值的描述：波特率分频系数为 BAUDPSC[9:0]＋1，数据位时间的波特率分频系数为 DBAUDPSC[9:0] ＋1。 5. ***CAN*** 模块***章节 42.3.9*** 总线集成状态小节下增加了一条注释：建议保持 EFDIS 位为 0 以使能边沿滤波，避免误检测总线空闲条件。 6. ***CAN*** 模块***章节 42.2*** 中，增加了 CANCLK 可配置特征，并附加了一条注释：CANCLK 时钟源的选择具有使用限制，请查看 GD32H7xx 系列的勘误手册。 7. ***CAN*** 模块更新内部计数器时钟源，增加使用限制。 ***CAN*** 模块增加对于标志帧的过滤器使用说明。	2025 年 01 月 16 日

版本号	描述	日期
1.6	1. 更新***表 1 5. 引导模式详细描述描述***; 2. 更新***表 3 1. GD32H7xx 闪存基地址和构成描述***; 3. 更新***电源管理单元（PMU）***章节部分描述 4. 更新***复位和时钟单元（RCU）***章节 PLL 控制寄存器（RCU_PLLALL）PLL0VCOSEL/PLL1VCOSEL/PLL2VCOSEL 位域描述； 5. 更新***模数转换器（ADC）***章节单端和差分输入通道小节描述； 6. 更新***通用同步异步收发器（USART）***章节半双工通信模式小节描述； 7. 更新***内部集成电路总线接口（I2C）***章节主机发送模式下和主机接收模式下的软件流程的软件流程小节描述； 8. 更新***串行外设接口/片上音频接口（SPI/I2S）***章节 I2S 停止流程小节描述； 9. 更新***串行外设接口/片上音频接口（SPI/I2S）***章节控制寄存器 0（SPI_CTL0）MSTART 位域描述； 10. 更新***串行音频接口（SAI）***章节主要特征部分描述； 11. 更新***管理数据输入/输出接口（MDIO）***章节表 37 1. MDIO 引脚定义和引脚数值寄存器（MDIO_PIN）描述； 12. 更新***控制器局域网络（CAN）***章节波特率小节描述； 13. 更新***控制器局域网络（CAN）***章节错误寄存器 1（CAN_ERR1）BDERR 和 BRERR 位域描述； 14. 更新***实时解密（RTDEC）***章节 RTDEC 原理小节描述； 15. 更新***实时解密（RTDEC）***章节区域 x 配置寄存器（RTDEC_AREx_CFG）比特 1 和中断使能寄存器（RTDEC_INTEN）比特 0 位域名称； 16. 更新***硬件信号量（HWSEM）***章节锁定信号量小节描述和控制寄存器（HWSEM_CTLx）（x=0…31）复位值； 17. 更新***通用串行总线高速接口（USBHS）***章节设备状态寄存器（USBHS_DSTAT）SPST 位域描述	2025 年 07 月 01 日
1.7	1. ***内部集成电路总线接口(I2C)***: 更新 26.4.4 节 ADDMSK2[2:0]描述和 26.4.12 节 I2C_CTL2 复位值；在 26.4.6 节增加 TIMEOUT 寄存器配置注意事项(I2C 使能时不能被改写)；在 26.3.10 和 26.4.5 节增加 TIMING 寄存器在 I2CEN 使能后可继续配置的说明，增加 SMBus 从机发送器使用 PEC 功能时必须在检测到 START 信号前配置	2026 年 01 月 20 日

版本号	描述	日期
	BYTENUM[7:0]；在 26.3.8 和 26.4.2 节增加主机传输时 BYTENUM 不要配置为 0 的说明，增加 START 位置 1 时 add10 等寄存器不能被修改的限制。 2. ***串行外设接口/片上音频接口(SPI/I2S)***：在 27.3.9 节和 27.5.10 节将 CRC 配置条件从"DZ=32 位或 DZ=16"改为"DZ=32 位"，增加 CRC 功能配置和 CRCPOLY 修改时机说明，增加 SPI 从模式 CRC 计算详细描述及清除步骤；在 27.3.6 节更新 NSS 电平保持描述；在 27.4.4 节更新 I2S 时钟源配置描述(删除 6 倍建议，增加低于 PCLK 的 1/6 倍限制说明)；在 27.3.2 和 27.5.15 节删除 IO23_DRV 相关描述与寄存器位；在 27.2.2 节删除数据位顺序描述；在 27.5.4 节 BIT31 增加在 MFD、MRU 和 MRB 模式下禁止使用 AFCTL 的说明；在 27.5.1 节更新 bit16 软件清 0 条件描述。 3. ***真随机数生成器(TRNG)***：在 11.3 节更新模块系统框图(图 11.1)；在 12.3、12.3.1、12.3.3 节将随机数"熵"描述修改为"随机数质量"。 4. ***哈希处理器(HAU):*** 在 14.5.2 节删除与英文不对应的 DMA1 CH7 描述；在 14.7.1 节修正 HAU_CTL 寄存器 bit5-4 描述(位翻转与字节翻转错误)。 5. ***控制器局域网络(CAN)***：在 42.3.10 节更新再同步有效跳变沿描述(改为"从隐性位到显性位")。 6. ***实时解密(RTDEC)***：在 46.2 节更新特性描述与英文版一致；在 46.5.1 节更新 ARE_FMVER 寄存器位域描述；在 46.5 小节和 46.5.10 节 RTDEC_INTF 寄存器增加终止事件时非加密区域读取返回 0 的描述。 7. ***S/PD 数字音频接口接收器(RSPDIF)***：在 33.3.9 节修改图 33.15 RSPDIF 上溢错误(RXSTEOMEN=1 且 RXDF[1:0]=2'b10)图示及描述。 8. ***电源管理单元(PMU)***：在 4.4.5 节(5.4.4 控制寄存器 2)交换 VCEN 和 VCRSEL 位置；在 5.3.3 节 VCORE 电源域将"0.9V"改为"Vcore"。 9. ***定时器(TIMER):*** 在第 24 章将 CHxMOMCTL 修改为 MCHxCOMCTL；在 24.3.5 节 TIMERx_CTL1 第 7 位改为保留。 10. ***管理数据输入/输出接口(MDIO)***：在 37.3 和 37.4 节更新引脚名称(MDIO 改为 MDIO_MDIO，MDC 改为 MDIO_MDC)。 11. ***数字摄像头接口(DCI)***：在 31.3 节更新信号名称以保持一致性。	

版本号	描述	日期
	12. ***编码器分频输出控制器(EDOUT)***: 在 41.6.4 节更新 EDOUT_OCNT 寄存器范围描述(从"-32768 to 32767"改为"-32767(0x8001) to 32767(0x7FFF)")。 13. ***通用同步异步收发器(USART)***: 在 25.4.13 节更新 USART_FCS 寄存器 bit26、bit25、bit5 位域描述；在 25.3.4 节删除接收采样噪声错误"起始位"条件；在 25.3.12 节更新智能卡 T=0 模式帧间隔位数(从 2.5 位改为 2 位)；在 25.4.8 节 USART_STAT 寄存器 REA 标志位增加在 UART3/4/6/7 中保留的备注；在 25.4.6 节增加 USART_RT 在过采样率为 16 时不能配置为 0xFFFFFF 的注意事项；在 25.3.10 节增加 IrDA 低功耗分频配置描述。 14. ***系统及存储器架构(SYSTEM)***: 在 1.3.1 和 1.8.20 节将 ftww 改为 fTWW 并增加"fTWW 数值详见芯片数据手册"说明；在 1.7.2 节更新图 1-8 中 TARG0 和 TARG1 访问宽度为 32 位；在 1.8.6 节将"差错"描述改为"比特错误"。 15. ***低功耗数字温度传感器(LPDTS)***: 在 40.5.2 节 T0 传感器数据寄存器(LPDTS_SDATA)增加 val 0x01 的 bit 说明。 16. ***串行音频接口(SAI)***: 在 34.3.4 节增加 SAI 从机子模块同步模式使用注意事项。 17. ***复位和时钟单元(RCU)***: 在 6.3.2、6.3.32、6.3.33 节更新 PLL VCO 时钟倍频因子描述；在 6.2.1 节更新图 6-2 时钟树中 FCLK HCLK 描述；在 6.3.42 节将 IRC16MEN 修正为 IRC64MEN。 18. ***主机直接存储器访问控制器(MDMA)***: 在 17.3.3 节缓冲区传输模式增加 note；在 17.4.6 节 MDMA_CHxCFG 寄存器更新 BTLEN[6:0]描述。 19. ***通用和备用输入/输出接口(GPIO/AFIO)***: 在 10.3.1 节更新 GPIO 引脚默认模式描述。 20. ***调试(DBG)***: 在 19.2.3 节更新安全 JTAG 读密钥时描述。 21. ***时钟校准控制器(CTC)***: 在 7.3.4 节修正公式 7-2。 22. ***时钟相位延迟模块(CPDM)***: 在 30.3.1 节将 DLSTCNT[11:0]改为 DLSTCNT[6:0]。 23. ***高性能数字滤波器(HPDF)***: 在阈值监视器快速模式章节更新滤波器输出结果所需串行样本数公式。 24. ***闪存控制器(FMC)***: 在 3.3.10 节 DCRP 区域将"DRCP"修正为"DCRP"。 25. ***中断/事件控制器(EXTI)***: 在 8.6.4 和 8.6.6 节修正 bit22 描述(从 12 改为 22)；在 8.6.3 节将 EXTI_RTEN0 寄存器中 RTEN27 和 RTEN20 位访	

版本号	描述	日期
	问权限从"w"改为"rw"。 26. ***通用串行总线高速接口(USBHS)***: 在第 49 章修改 GAHBCS 的 DMAEN 使能位描述。 27. ***RAM ECC 监视器单元(RAMECCMU)***: 将"差错"描述改为"比特错误"。	
1.8	1. 文档名称更新为 GD32H73x_75x user manual, 合并 GD32H75E user manual 相关内容。 2. ***看门狗定时器（WDGT）***章节：删除寄存器半字访问的相关描述。 3. ***通用和备用输入/输出接口（GPIO 和 AFIO）***模块：在输入滤波小节添加注意事项说明。 4. ***三角函数加速器（TMU）***模块：在控制和状态寄存器（TMU_CS）中删除 ITRTNUM[3:0] 位域的描述。 5. ***真随机数生成器（TRNG）***模块：在控制寄存器（TRNG_CTL）的 CONDRST 位域和状态寄存器（TRNG_STAT）的 SEIF、SECS 位域中补充注意事项说明。 6. ***系统及存储器架构***章节：更新 TIMERx 配置寄存器 2（SYSCFG_TIMERxCFG2, x=0,7）中 TSCFG15 位域的描述。 7. ***加密处理器（CAU）***章节：在操作模式小节增加注意事项说明。 8. ***控制器局域网络（CAN）***章节：将 CAN_ERR1 寄存器的复位值更新为 0x000X000X，完善退出 Pretended Networking mode 的描述，并更新 CAN_CTL2 寄存器 RRFRMS 位的说明。 9. ***模数转换器（ADC）***章节：补充差分通道的使用说明，并完善 ADC 常规序列的软件流程描述。	2026 年 03 月 31 日

Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the “Product”), is owned by the Company according to the laws of the People’s Republic of China and other applicable laws. The Company reserves all rights under such laws and no Intellectual Property Rights are transferred (either wholly or partially) or licensed by the Company (either expressly or impliedly) herein. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

To the maximum extent permitted by applicable law, the Company makes no representations or warranties of any kind, express or implied, with regard to the merchantability and the fitness for a particular purpose of the Product, nor does the Company assume any liability arising out of the application or use of any Product. Any information provided in this document is provided only for reference purposes. It is the sole responsibility of the user of this document to determine whether the Product is suitable and fit for its applications and products planned, and properly design, program, and test the functionality and safety of its applications and products planned using the Product. The Product is designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only, and the Product is not designed or intended for use in (i) safety critical applications such as weapons systems, nuclear facilities, atomic energy controller, combustion controller, aeronautic or aerospace applications, traffic signal instruments, pollution control or hazardous substance management; (ii) life-support systems, other medical equipment or systems (including life support equipment and surgical implants); (iii) automotive applications or environments, including but not limited to applications for active and passive safety of automobiles (regardless of front market or aftermarket), for example, EPS, braking, ADAS (camera/fusion), EMS, TCU, BMS, BSG, TPMS, Airbag, Suspension, DMS, ICMS, Domain, ESC, DCDC, e-clutch, advanced-lighting, etc.. Automobile herein means a vehicle propelled by a self-contained motor, engine or the like, such as, without limitation, cars, trucks, motorcycles, electric cars, and other transportation devices; and/or (iv) other uses where the failure of the device or the Product can reasonably be expected to result in personal injury, death, or severe property or environmental damage (collectively "Unintended Uses"). Customers shall take any and all actions to ensure the Product meets the applicable laws and regulations. The Company is not liable for, in whole or in part, and customers shall hereby release the Company as well as its suppliers and/or distributors from, any claim, damage, or other liability arising from or related to all Unintended Uses of the Product. Customers shall indemnify and hold the Company, and its officers, employees, subsidiaries, affiliates as well as its suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Product.

Information in this document is provided solely in connection with the Product. The Company reserves the right to make changes, corrections, modifications or improvements to this document and the Product described herein at any time without notice. The Company shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2026 GigaDevice Semiconductor Inc. – All rights reserved